SG2000
技术参考手册

版本: 1.0-alpha
发布日期: 2023-11-30

© 2023 算能

1 修订记录

Revision	Date	Description
1.0	2023/11/30	初稿

目 录

1 修订记录 2
2 系统概述 18
2.1 介绍 18
2.2 系统框架图 18
2.3 特性介绍 19
2.3.1 处理器内核 19
2.3.2 TPU 19
2.3.3 视频编解码 19
2.3.4 视频接口 20
2.3.5 ISP 与图像处理 20
2.3.6 硬件加速引擎 21
2.3.7 音频编解码 21
2.3.8 网络接口 21
2.3.9 安全系统模块 21
2.3.10 智能安全运行环境 21
2.3.11 外围接口 22
2.3.12 外部存储器接口 22
2.3.13 芯片物理规格 23
3 系统架构 24
3.1 处理器子系统 24
3.2 中断子系统 25
3.3 地址空间映射 28
3.4 封装与管脚分布 31
3.4.1 封装 31
3.4.2 管脚信息描述 33
3.4.3 管脚默认状态(IFGBA) 38
4 启动和升级 41
4.1 概述 41
4.2 启动模式和对应的信号锁存值对应关系 41
4.3 镜像烧写模式. 41
4.4 安全启动 41
5 复位 42
5.1 概述 42
5.2 复位控制 42
5.2.1 上电复位 42
5.2.2 系统硬复位 42

5.2.3 软复位 43
5.2.4 处理器子系统软复位 43
5.2.5 功能子系统及功能模块软复位 43
5.3 复位配置寄存器 43
5.3.1 复位配置寄存器概览 43
5.3.2 复位配置寄存器描述 43
6 时钟 47
6.1 概述 47
6.2 功能框图 47
6.3 时钟资源与分频结构 47
6.4 PLL 配置 48
6.5 整数倍频 PLL 49
6.5.1 分数倍频 PLL 50
6.6 CLK_DIV 时钟分频配置 50
6.6.1 IP/SYS 时钟源、时钟分频配置 51
6.6.2 MCLK0/MCLK1 52
6.6.3 Clk_A24M 52
6.7 PLL CRG 寄存器概览 52
6.8 PLL CRG 寄存器描述 53
6.8.1 PLL_G2 寄存器描述 53
6.8.2 PLL_G6 寄存器描述 59
6.9 CLK_DIV CRG 寄存器概览 62
6.10 CLK_DIV CRG 寄存器描述 63
7 系统控制器 81
7.1 概述 81
7.2 功能描述 81
7.2.1 全局复位使能 81
7.2.2 系统 DMA 通道映射 81
7.2.3 DDR AXI Urgent/Qos 配置 82
7.3 系统控制寄存器 82
7.3.1 系统控制寄存器概览 82
7.3.2 系统控制寄存器描述 84
8 管脚复用与控制 89
8.1 管脚复用 PINMUX 89
8.1.1 接口功能与信号/引脚/FMUX 寄存器的对应关系 89
8.1.2 FMUX 寄存器描述 121
8.2 管脚控制 PINCTRL 134
8.2.1 引脚控制寄存器概览 134
8.2.2 PINCTRL 寄存器描述 136

9 DMA 控制器 158
9.1 概述 158
9.2 特点 159
9.3 功能描述 159
9.3.1 外设请求线 159
9.3.2 访问空间 159
9.3.3 基本传输 159
9.3.4 链表传输 161
9.3.5 中断和状态 161
9.3.6 通道安全配置 163
9.4 工作方式 163
9.4.1 时钟和复位 163
9.4.2 初始化 163
9.4.3 基本传输 163
9.4.4 链表传输 164
9.4.5 中断处理 164
9.5 DMAC 寄存器 164
10 定时器 215
10.1 概述 215
10.2 特点 215
10.3 功能描述 215
10.4 工作方式 215
10.4.1 初始化 215
10.4.2 中断处理 216
10.4.3 时钟选择 216
10.5 Timer 寄存器概览 216
10.6 Timer 寄存器描述 216
11 看门狗 219
11.1 概述 219
11.2 特点 219
11.3 功能描述 219
11.3.1 应用框图 219
11.3.2 功能原理 220
11.4 工作方式 220
11.4.1 计数时钟频率配置 220
11.4.2 系统初始化配置 220
11.4.3 中断处理过程 221
11.4.4 关闭 WatchDog 221
11.5 WDT 寄存器概览 221

11.6 WDT 寄存器描述 222
12 实时时钟 225
12.1 概述 225
12.2 特点 225
12.3 功能描述 225
12.4 工作方式 226
12.4.1 计数时钟频率 226
12.4.2 RTC 复位 226
12.4.3 RTC 初始化 227
12.4.4 模拟秒时钟初始化 228
12.4.5 中断处理 228
12.4.6 休眠与唤醒 229
12.4.7 关电与上电 229
12.5 RTC 寄存器概览 229
12.6 RTC 寄存器描述 232
12.6.1 RTC_CORE_REG 232
12.6.2 RTC_MACRO_REG 243
12.6.3 RTC_CTRL 245
12.7 电源管理与低功耗模式 256
12.8 概述 256
12.8.1 MCU only 257
12.8.2 Active mode 257
12.8.3 RTC only 257
12.9 时钟控制 257
12.9.1 关闭不需要的时钟分频器 257
12.9.2 调整模块的工作频率 258
12.9.3 模块级低功耗控制 258
12.10 关闭不使用的 PLL 258
12.11 DDR 低功耗控制 258
12.12 电压调节 258
13 8051 子系统 260
13.1 概述 260
13.2 特点 260
13.3 功能描述 261
13.4 工作方式 261
13.4.1 Power domain 控制流程 261
13.4.2 8051 初始化 262
13.4.3 中断处理 262
13.4.4 MAILBOX 262
13.5 8051 子系统寄存器概览 263

13.6 8051 子系统寄存器 263
14 存储器接口 276
14.1 DDR 控制器 276
14.1.1 概述 276
14.1.2 特点 276
14.1.3 功能描述 276
14.1.4 工作方式 280
14.1.5 AXI 寄存器 281
14.1.6 DDRC 寄存器 314
14.2 SPI NOR Flash 控制器 316
14.2.1 概述 316
14.2.2 特点 316
14.2.3 功能描述 316
14.2.4 工作流程 320
14.2.5 寄存器概览 321
14.2.6 寄存器描述 322
14.3 SPI NAND Flash 控制器 325
14.3.1 概述 325
14.3.2 特点 325
14.3.3 功能描述 326
14.3.4 工作流程 329
14.3.5 数据结构(NAND Flash/SPI NAND Flash) 330
14.3.6 寄存器概览 330
14.3.7 寄存器描述 331
15 网络接口 333
15.1 Ethernet MAC 333
15.1.1 概述 333
15.1.2 功能描述 333
15.1.3 总体数据流 333
15.1.4 单网口功能配置描述 334
15.1.5 以太网收包中断管理功能 334
15.1.6 配置 PHY 芯片工作状态 334
15.1.7 工作模式切换 335
15.1.8 典型应用 335
15.1.9 寄存器偏移地址说明 335
15.1.10 GMAC 寄存器概览 335
15.1.11 GMAC 寄存器描述 336
15.2 Ethernet PHY 339
15.2.1 概述 339
15.2.2 功能描述 339

15.2.3 功能配置图描述 340

16 视频接口 341

16.1 VI 341
16.2 概述 341
16.2.1 特点 341
16.2.2 模式功能描述 342
16.2.3 图像储存模式 347
16.2.4 VI 寄存器概览 347
16.2.5 VI 寄存器描述 349
16.3 VDP（Video Display Processor） 362
16.3.1 概述 362
16.3.2 架构描述 362
16.3.3 工作方式 363
16.3.4 功能描述 365
16.3.5 VDP 寄存器概览 368
16.3.6 VDP 寄存器描述 372
16.4 MIPI Rx 413
16.4.1 概述 413
16.4.2 特点 413
16.4.3 功能描述 414
16.4.4 MIPI Rx 寄存器概览 432
16.4.5 MIPI Rx 寄存器描述 435
16.5 MIPI Tx 456
16.5.1 概述 456
16.5.2 特点 456
16.5.3 功能描述 456
16.5.4 MIPI Tx 寄存器概览 463
16.5.5 MIPI Tx 寄存器描述 463
16.5.6 MIPI Tx PHY 寄存器概览 465
16.5.7 MIPI Tx PHY 寄存器描述 466

17 音频接口 469

17.1 AIAO 469
17.1.1 概述 469
17.1.2 特点 469
17.1.3 功能描述 470
17.1.4 工作方式 474
17.1.5 AIAO 寄存器概览 474
17.1.6 AIAO 寄存器描述 475
17.2 Audio Codec 489
17.2.1 概述 489

17.2.2 特点 489
17.2.3 Audio Codec 寄存器概览 489

18 外围设备 496

18.1 I2C 496
18.1.1 概述 496
18.1.2 功能描述 496
18.1.3 功能框图 496
18.1.4 I2C 协定时序 497
18.1.5 工作方式 497
18.1.6 I2C 寄存器概览 499
18.1.7 I2C 寄存器描述 501
18.2 UART 508
18.2.1 概述 508
18.2.2 特点 509
18.2.3 功能描述 509
18.2.4 工作方式 510
18.2.5 UART 寄存器概览 515
18.2.6 UART 寄存器描述 516
18.3 SPI 521
18.3.1 概述 521
18.3.2 特点 522
18.3.3 功能描述 522
18.3.4 工作方式 523
18.3.5 三种串行外设总线时序图 527
18.3.6 寄存器概览 530
18.3.7 寄存器描述 531
18.4 eMMC/SD/SDIO 控制器 542
18.4.1 功能描述 542
18.4.2 应用说明 554
18.4.3 寄存器概览 570
18.4.4 寄存器描述 570
18.5 GPIO 584
18.5.1 概述 584
18.5.2 特点 585
18.5.3 工作方式 585
18.5.4 GPIO 寄存器概览 586
18.5.5 GPIO 寄存器描述 586
18.6 USB DRD 589
18.6.1 概述 589
18.6.2 功能描述 589
18.6.3 USBC 功能与寄存器描述 591

18.6.4 Host 初始化程序说明 630
18.6.5 Host 寄存器说明 630
18.6.6 Device 初始化程序 639
18.6.7 Device 寄存器说明 639
18.7 SARADC 649
18.7.1 概述 649
18.7.2 特点 649
18.7.3 工作方式 650
18.7.4 SARADC 寄存器概览 650
18.7.5 SARADC 寄存器描述 650
18.8 Temperature sensor 652
18.8.1 概述 652
18.8.2 工作方式 652
18.8.3 Temperature sensor 寄存器概览 655
18.8.4 Temperature sensor 寄存器描述 655
18.9 PWM 659
18.9.1 概述 659
18.9.2 特点 659
18.9.3 工作方式 659
18.9.4 PWM 寄存器概览 661
18.9.5 PWM 寄存器描述 662
18.10 Key scan 667
18.10.1 概述 667
18.10.2 工作方式 667
18.10.3 基本设定 668
18.10.4 使用 FIFO 模式. 668
18.10.5 使用 snapshot array 模式. 669
18.10.6 Key scan 寄存器概览 669
18.10.7 Key scan 寄存器描述 669
18.11 Wiegand 673
18.11.1 概述 673
18.11.2 工作方式 674
18.11.3 Wiegand 寄存器概览 676
18.11.4 Wiegand 寄存器描述 677
18.12 IRRX 紅外接口 681
18.12.1 概述 681
18.12.2 特點 681
18.12.3 工作方式 681
18.12.4 IRRX 寄存器概览 681
18.12.5 IRRX 寄存器描述 683

19 安全子系统模块 692

19.1 CryptoDMA 692
19.1.1 概述 692
19.1.2 功能特性 693
19.1.3 DMA 功能描述 693
19.1.4 对称密钥算法分组加密模式功能描述 693
19.1.5 CryptoDMA 寄存器概览 695
19.1.6 CryptoDMA 寄存器描述 697
19.2 安全除错防火墙 707
19.2.1 概述 707
19.2.2 状态查讯及密码输入介面(I2C) 707
19.2.3 状态查讯及密码输入流程 708
19.3 Efuse 控制器 709

图 目 录

图表 2-1 架构图 18
图表 3-1 封装外观尺寸, 上视图 32
图表 3-2.封装外观尺寸, 底视图 32
图表 3-3 管脚分布图. 33
图表 5-1 复位管理模块框图 42
图表 6-1 时钟管理框图 47
图表 6-2 时钟源分频示意图. 48
图表 9-1 DMAC 硬件控制流程示意图 158
图表 9-2 DMA 传输阶层 160
图表 9-3 链表相对位址与数据格式 161
图表 9-4 中断状态和来源示意图 162
图表 11-1 WatchDog 应用框图 220
图表 12-1 使用 PWM 控制 DCDC 电压范例. 259
图表 13-1 8051 子系统架构 261
图表 14-1 SoC/DRAM 互联示意图 277
图表 14-2 Standard SPI 接口模式写操作时序图 316
图表 14-3 Standard SPI 接口模式读操作时序图 317
图表 14-4 Dual-Input SPI 接口时序 317
图表 14-5 Dual-IO SPI 接口时序 318
图表 14-6 Quad-Input SPI 模式时序图 318
图表 14-7 Quad-IO SPI 模式时序图 319
图表 14-8 Standard SPI 接口模式写操作时序 326
图表 14-9 Standard SPI 接口模式读操作时序 326
图表 14-10 SPI Nand x2 接口模式操作时序 327
图表 14-11 SPI Nand x4 接口模式操作时序 327
图表 15-1 eMAC 总体数据流 334
图表 15-2 内建 10/100 Ethernet PHY 框图 340
图表 16-1 VI 功能框图 341
图表 16-2 BT.1120 水平接口时序 342
图表 16-3 BT.1120 垂直接口时序 343
图表 16-4 BT.656 水平接口时序 344
图表 16-5 BT.601 vhs 同步模式 344
图表 16-6 BT.601 vde 同步模式 345
图表 16-7 BT.601 vsde 同步模式 345
图表 16-8 BT.601 Y/C 分开 16bit 模式 345
图表 16-9 BT.601 Y/C 合并 8bit 模式 345
图表 16-10 DC 同步码模式 346
图表 16-11 DC 同步信号模式 -vhs mode 346
图表 16-12DC 同步信号模式 -VDE mode 346
图表 16-13DC 同步信号模式 -VSDE mode 346

图表 16-14 Bayer 12 bit 图像储存方式 347
图表 16-15 YCbCr 8bit 图像储存方式 347
图表 16-16 VI 寄存器概览 347
图表 16-17 VDP 功能框图 362
图表 16-18 VDP 时钟生成器 363
图表 16-19 LVDS JEIDA/VESA mode 的 data format 368
图表 16-20 MIPI Rx 功能框图及在系统中的位置 413
图表 16-21 MIPI Rx 支持接口类型 414
图表 16-22 数据包的传输机制 415
图表 16-23 CSI-2 长包格式 416
图表 16-24 CSI-2 短包格式 416
图表 16-25 YUV422 8-bit 资料传输顺序 417
图表 16-26 YUV422-8bit 资料封包传输对应 417
图表 16-27 YUV422 8-bit Frame 格式 418
图表 16-28 YUV422-10bit 资料传输顺序 418
图表 16-29 YUV422-10bit 资料封包传输对应 419
图表 16-30 YUV422-10bit Frame 格式 419
图表 16-31 RAW8 资料传输顺序 420
图表 16-32 RAW8 Frame 格式 420
图表 16-33 RAW10 资料传输顺序 420
图表 16-34 RAW10 Frame 格式 421
图表 16-35 RAW12 资料传输顺序 421
图表 16-36 RAW12 Frame 格式 421
图表 16-37 RAW16 资料传输顺序 422
图表 16-38 RAW16 Frame 格式 422
图表 16-39 MIPI 接口图像格式 423
图表 16-40 MIPI 接口宽动态数据传输 (使用 DT) 424
图表 16-41 MIPI 接口宽动态数据传输 (使用 ID) 425
图表 16-42 MIPI 接口宽动态数据传输 (寄存器设定) 426
图表 16-43 Sub-LVDS 接口数据格式 426
图表 16-44 Sub-LVDS 同步码举例 427
图表 16-45 Sub-LVDS Multi Lane 传输模式 427
图表 16-46 Sub-LVDS 单个像素时序图 428
图表 16-47 Sub-LVDS 线性模式时序图 428
图表 16-48 Sub-LVDS 宽动态模式一 429
图表 16-49 Sub-LVDS 宽动态模式二 429
图表 16-50 HiSPi Packetized-SP 模式 430
图表 16-51 HiSPi Streaming-SP 模式 431
图表 16-52 HiSPi 同步码支援模式 431
图表 16-53 HiSPi Packetized-SP 宽动态传输 432
图表 16-54 HiSPi Streaming-SP 宽动态传输 432
图表 16-55 MIPI Rx PHY 寄存器概览 433

图表 16-56 MIPI Rx CSI 控制器寄存器概览 434
图表 16-57 MIPI Rx Sub-LVDS 控制器寄存器总览 435
图表 16-58 MIPI TX 功能框图 456
图表 16-59 RGB16-bit 格式 (Data type = 0x0E) 458
图表 16-60 RGB 18-bit 格式 (Data type = 0x1E) 459
图表 16-61 RGB 24-bit 格式 (Data type = 0x3E) 460
图表 16-62 RGB 30-bit 格式 (Data type = 0x0D) 461
图表 16-63 MIPI TX 时序标记 461
图表 16-64 MIPI TX Video mode Burst Transmission 时序 462
图表 17-1 AIAO 框图 469
图表 17-2 通过 I2S 接口与内置 Audio Codec 连接示意图 471
图表 17-3 AIAO 以主模式通过 I2S 接口与外置 Audio Codec 连接示意图 471
图表 17-4 AIAO 以从模式通过 I2S 接口与外置 Audio Codec 连接示意图 472
图表 17-5 I2S 接口时序 473
图表 17-6 PCM 接口标准模式时序 473
图表 17-7 PCM 接口左对齐模式时序 474
图表 18-1 I2C 功能框图 497
图表 18-2 I2C 协定时序 497
图表 18-3 I2C 非 DMA 方式下的数据传输软件流程 498
图表 18-4 I2C DMA 方式下的数据传输软件流程 499
图表 18-5 UART 应用框图 510
图表 18-6 UART 传输数据结构 510
图表 18-7 UART 数据发送流程图 512
图表 18-8 UART 数据接收流程图 513
图表 18-9 SPI 应用框图 522
图表 18-10 对接外部 SPI/SSP slave 时的数据传输流程 525
图表 18-11 对接外部 Microwire slave 时的数据传输流程 526
图表 18-12 Motorola SPI 单帧传输格式 (SCPH = 0) 527
图表 18-13 Motorola SPI 连续帧传输格式 (SCPH = 0) 528
图表 18-14 Motorola SPI 单帧传输格式 (SCPH = 1) 528
图表 18-15 Motorola SPI 连续帧传输格式 (SCPH = 1) 528
图表 18-16 TI SSP 单帧传输格式 529
图表 18-17 TI SSP 连续帧传输格式 529
图表 18-18 NS Microwire 单帧传输格式 530
图表 18-19 NS Microwire 连续帧传输格式 530
图表 18-20 SDMMC 控制器的功能框图 544
图表 18-21 典型应用 545
图表 18-22 eMMC/SD/SDIO 指令格式 545
图表 18-23 eMMC/SD/SDIO 响应格式 546
图表 18-24 eMMC/SD/SDIO 1-bit 数据封包格式 546
图表 18-25 eMMC/SD/SDIO 4-bit 数据封包格式 547
图表 18-26 eMMC/SD/SDIO 8-bit 数据封包格式 548

图表 18-27 4-bit dual data rate 数据封包格式 549
图表 18-28 8-bit dual data rate 数据封包格式 550
图表 18-29 非数据传输指令:透过信号线 CMD 来完成指令传送与接收响应 551
图表 18-30 单块与多块读操作 551
图表 18-31 单块与多块写操作 552
图表 18-32 时钟关控程序 554
图表 18-33 时钟配置流程图 556
图表 18-34 指令传输程序 558
图表 18-35 指令完成程序 560
图表 18-36 中止指令程序 561
图表 18-37 非同步中止指令的程序图 562
图表 18-38 同步中止指令的程序图 563
图表 18-39 非 DMA 数据传输模式的程序 565
图表 18-40 SDMA 数据传输模式程序 567
图表 18-41 ADMA 数据传输模式程序 569
图表 18-42 USB DRD 内部的系统框图 590
图表 18-43 温度量测时间、计数与中断发生关系图 654
图表 18-44 PWM Continuous mode 659
图表 18-45 PWM Pulse count mode 660
图表 18-46 PWM Continuous Shift Mode 661
图表 18-47 Keyscan 架构框图 667
图表 18-48 wiegand 信号总线传输 0/1 的方式 673
图表 18-49 Wiegand 26 格式 673
图表 18-50 Wiegand 34 format 674
图表 18-51 常见的磁卡, 磁扣的数字意义 674
图表 18-52 wiegand 架构框图 675
图表 18-53 IRRX 工作方式 681
图表 19-1 安全子系模块 692
图表 19-2 ECB 模式 694
图表 19-3 CBC 模式 694
图表 19-4 CTR 模式 695
图表 19-5 eFuse CTRL 模块架构 710

表 目 录

表格 3-1 ARM Cortex-A53 处理器中断号与中断源映射表 25
表格 3-2 主处理器 RISCV C906 @ 1.0Ghz 中断号与中断源映射表 26
表格 3-3 协处理器 RISCV C906 @ 700Mhz 中断号与中断源映射表 27
表格 3-4 管脚信息表 33
表格 3-5 管脚默认状态表 38
表格 6-1 PLL 配置参数表 48
表格 6-2 整数 PLL 配置参数表 49
表格 6-3 分数 PLL 参数配置表 50
表格 6-4 时钟源及预设分频参数表 50
表格 8-1 接口功能与信号/引脚/FMUX 寄存器的对应关系表 89
表格 8-2 FMUX 寄存器配置信息表 121
表格 8-3 引脚和引脚控制寄存器对应信息表 134
表格 8-4 引脚控制寄存器配置表 136
表格 9-1 DMAC 访问空间类型 159
表格 10-1 Timer 寄存器概览 (基址 0x030A0000) 216
表格 11-1 WDT 寄存器概览 221
表格 12-1 RTC_REG 寄存器概览 (基址 0x05026000) 229
表格 12-2 RTC_MACRO_REG 寄存器概览 (基址 0x05026400) 231
表格 12-3 RTC_CTRL 寄存器概览 (基址 0x05025000) 231
表格 13-1 RTC_CTRL 寄存器概览 (基址 0x05025000) 263
表格 14-1 DDR2/DDR3 命令信号比较表 277
表格 14-2 DDR2 命令真值表 278
表格 14-3 DDR3 命令真值表 278
表格 15-1 GMAC 寄存器概览 335
表格 17-1 AIAO 子系统寄存器概览 (基址 0x0410_8000) 474
表格 17-2 I2S_TDM_0/1/2/3 寄存器概览 (基址 0x0410_0000 + n*0x10000) 475
表格 18-1 I2C 时钟选择与相关寄存器配置关系表 497
表格 18-2 芯片的 6 组 I2C 模块基地址 499
表格 18-3 芯片的 I2C 寄存器概览 500
表格 18-4 芯片的 6 组 UART 模块基地址 515
表格 18-5 芯片的 UART 寄存器概览 515
表格 18-6 芯片的 4 组 SPI 模块基地址 530
表格 18-7 SPI 寄存器概览 531
表格 18-8 SDMMC 控制器对应的功能讯号与脚位对应表 543
表格 18-9 SD3.0 支持的速度和电压 552
表格 18-10 eMMC 支持速度模式与电压 553
表格 18-11 SDMMC 寄存器概览 570
表格 18-12 芯片的 4 组 GPIO 模块基地址 586
表格 18-13 GPIO 寄存器概览 586
表格 18-14 Tempsensor 中断信号描述表 653

表格 18-15 PWM 寄存器概览 661
表格 19-1 状态查询 I2C 介面寄存器地址 707

2 系统概述

2.1 介绍

SG2000 是面向 AIot 领域推出的高性能、低功耗芯片，内置多个强劲内核：2xC906、1xArm Cortex A53、1x8051，用户可以根据需求快速切换内核。同时集成硬件编解码器：H.264 视频压缩编解码器, H.265 视频压缩编码器和 ISP，支持配置了专业级的视频图像 ISP：HDR 宽动态、3D 降噪、除雾、镜头畸变校正等多种图像增强和矫正算法。

芯片更集成了自研 TPU，可提供 0.5TOPS IN8 的算力。 特殊设计的 TPU 调度引擎能有效地为所有的张量处理器核心提供极高的带宽数据流。

同时支持丰富的外设接口：6xI2C、2xSDIO3.0、2xI2S、16xPWM、1xUSB2.0 等，满足多种场景需求。

2.2 系统框架图

图表 2-1 架构图

2.3 特性介绍

2.3.1 处理器内核

- 主处理器 RISCV C906 @ 1.0Ghz .
  - 32KB I-cache, 64KB D-Cache
  - 集成矢量(Vector)及浮点运算单元 (FPU)
- 主处理器 ARM Cortex-A53 @ 1.0Ghz
  - 32KB I-cache, 32KB D-Cache
  - 128KB L2 cache
  - 支持 Neon 以及浮点运算 FPU
- 协处理器 RISCV C906 @ 700Mhz
  - 集成浮点运算单元 (FPU)

2.3.2 TPU

- 内建 TPU , 算力达到 0.5TOPS INT8
- 支援主流的神经网络架构: Caffe， Pytorch， TensorFlow(Lite)， ONNX 和 MXNet
- 可实现行人侦测 (Pedestrian Detection) , 人脸侦测 (Face Detection) , 人脸识别 (Face recognition) , 活体侦测 (Face anti-spoofing) 及其他视频结构化应用.

2.3.3 视频编解码

- H.264 Baseline/Main/High profile
- H.265 Main profile
- H.264/H.265 均支援 I 帧及 P 帧
- MJPEG/JPEG baseline
- H.264 编解码最大分辨率 : 2880x1620 (5M)
- H.265 编码最大分辨率 : 2880x1620 (5M)
- H.264 编解码性能
  - 2880x1620@30fps+720x576@30fps
  - 1920x1080@30fps 编码 + 1920x1080@30fps 解码
- H.265 编码性能
  - 2880x1620@30fps+720x576@30fps
- JPEG 最大编解码性能
  - 2880x1620@30fps
- 支持 CBR/VBR/FIXQP 等多种码率控制模式.

- 支持 感兴趣区域 (ROI) 编码

2.3.4 视频接口

- 输入
  - 支持同时三路视频输入(mipi 2L+2L + DVP or MIPI 4L)
  - 支持 MIPI, Sub-LVDS, HiSPI 等串行接口.
  - 支持 8/10/12 bit RGB Bayer 视频输入.
  - 支持 BT.601,BT.656,BT.1120 视频输入.
  - 支持 AHD 多路混合 BT 格式.
  - 支持 SONY, OnSemi, OmniVision 等高清 CMOS sensor
  - 提供可编程频率输出供 sensor 作为参考时钟.
  - 支持最大宽度为 2880 , 最大分辨率 5M (2688x1944, 2880x1620)

- 输出
  - 支持多种串行与并行屏显规格.
  - 支持 MIPI，LVDS 等串行接口
  - 支持 BT.601, BT.656, BT.1120, RGB565/666/888, 8080 输出接口.
  - 支持 SPI 输出接口.

2.3.5 ISP 与图像处理

- 图像视频 90 度、180 度、270 度旋转
- 图像视频 Mirror、Flip 功能
- 视频 2 层 OSD 叠加
- 视频 1/32～32x 缩放功能
- 3A（AE/AWB/AF）算法
- 固定模式噪声消除、坏点校正
- 镜头阴影校正、镜头畸变校正、紫边校正
- 方向自适应 demosaic
- Gamma 校正、(区域/全域)动态对比度增强、颜色管理和增强
- 区域自适应去雾
- Bayer 降噪、3D 降噪、细节增强及锐化增强
- Local Tone mapping
- Sensor 自带宽动态和 2 帧宽动态
- 两轴数字图像防抖
- 镜头畸变校正

- 提供 PC 端 ISP tuning tools

2.3.6 硬件加速引擎

- 软硬体混合模式支持部分 OpenCV 库.
- 软硬体混合模式支持部分 IVE 库.

2.3.7 音频编解码

- 集成 Audio CODEC, 支持 16 bit 音源/语音 输入和输出.
- 集成双声道麦克风输入.
- 集成双声道输出. (需要外挂功放才能推动喇叭)
- 同时支持以 I2S/PCM/TDM 接口连接外部 audio CODEC. 内建 audio PLL 支援 MCLK 输出.
- 软件音频编解码协议 (G.711, G.726, ADPCM)
- 软件支持音频 3A (AEC, ANR, AGC) 功能.

2.3.8 网络接口

- 以太网模块提供 1 个 Ethernet MAC , 实现网路数据的接收与发送.
- Ethernet MAC 搭配内建 10/100Mbps Fast Ethernet Transceiver 可工作在 10/100Mbps 全双工或半双工模式，也可通过 RMII 外挂 PHY.

2.3.9 安全系统模块

- 硬件实现 AES/DES/SM4 多种加解密算法
- 硬件实现 HASH(SHA1/SHA256) 哈希算法
- 硬件实现随机数发生器
- 内部集成 2Kbit eFuse 逻辑空间

2.3.10 智能安全运行环境

- 支持信任链建立: 提供安全环境的基础，为可信环境的根本，如硬件安全设置、信任根

- 支持安全启动，提供安全硬件、软件保护功能
- 支持资料加密安全: 数据加密程序，运算核心加密
- 支持软、固件验签流程:确认软件可信性及完整性，包括 开机及载入验签程序
- 支持安全储存及传输:保护外部数据储存及交换
- 支持安全更新

2.3.11 外围接口

- 集成 POR, Power sequence
- 6 个单端 ADC (3 no die domain)
- 6 个 I2C (1 no die domain)
- 4 个 SPI
- 5 组 UART (no die domain)
- 4 组(16 通道) PWM
- 2 个 SDIO 接口.
  - 一个支持 3V 连接 SD 3.0 Card(支持最大容量 SDXC 2TB, 支持速度为 UHS-I)
  - 一个支持 1.8V/3.0V 连接其他 SDIO 3.0 设备.(支持速度为 UHS-I)
- 110 GPIO 接口 (25 no die domain)
- 集成 keyscan 及 Wiegand
- 集成 MAC PHY 支援 10/100Mbps 全双工或半双工模式. 也可通过 RMII 外挂 PHY.
- 一个 USB Host / device 接口
- 两组 I2S

2.3.12 外部存储器接口

- 内建 DRAM
  - DDR3 16bitx1, 最高速率达 1866Mbps, 容量 4Gbit (256MB)
- SPI NOR flash 接口 (1.8V / 3.0V)
  - 支持 1, 2, 4 线模式.
  - 最大支持 256MByte.
- SPI Nand flash 接口 (1.8V / 3.0V)
  - 支持 1KB/2KB/4KB page (对应的最大容量 16GB/32GB/64GB)
  - 使用器件本身内建的 ECC 模块.
- eMMC 4.5 接口 (1.8V/3.0V) SD0 EMMC 共电. 因为 SD 卡 default 3V, 所以有 SD 卡时, 不适合接 1.8V eMMC.

- 4 bit 接口.
- 支持 HS200
- 最大支持容量 2TB

2.3.13 芯片物理规格

- 功耗
  - 1080P + Video encode + AI : ~ 500mW
  - 其余场景 : TBD
- 工作电压
  - 内核电压为 0.9V
  - IO 电压为 1.8V 及 3.0V
  - DDR 电压 1.35V.
- 封装
  - 使用 LFBGA 封装, 封装尺寸为 10mmx10mmx1.3mm. 管脚间距为 0.65mm. 管脚总数为 205 个 。

3 系统架构

3.1 处理器子系统

本芯片有两个主处理器：RISC-V C906 和 ARM Cortex-A53，通过管脚 GPIO_RTX 的外围电路控制进行切换（具体参考 datasheet），有两种工作模式：

- RISC-V C906@1GHz + RISC-V C906@700MHz。
- ARM Cortex-A53@1GHz + RISC-V C906@700MHz。

采用 ARM Cortex-A53 处理器，具有以下特点：

- 处理器工作频率最高可达 1.0GHz。
- 支持 Neon。
- 支持浮点运算单元 FPU。
- L1 Cache 包含 32KB Instruction Cache 和 32KB Data Cache。
- 128KB L2 cache。
- 支持 MMU（Memory Management Unit）。
- 处理器内部集成中断控制器
- 支持 JTAG 调试接口。

采用 RISCV C906 处理器，具有以下特点：

- 处理器工作频率最高可达 1.0GHz。
- 集成矢量执行单元，浮点协处理器。
- L1 Cache 包含 32KB Instruction Cache 和 64KB Data Cache。
- 支持 MMU（Memory Management Unit）。
- 处理器内部集成中断控制器
- 支持 JTAG 调试接口。

3.2 中断子系统

表格 3-1 ARM Cortex-A53 处理器中断号与中断源映射表

中断号	中断源	中断号	中断源	中断号	中断源
32	TEMPSENS 中断	70	SPI1 中断	108	TRNG 中断
33	RTC Alarm 中断	71	SPI2 中断	109	ddr_axi_mon 中断
34	RTC Longpress 中断	72	SPI3 中断	110	ddr_pi_phy 中断
35	VBAT DET 中断	73	SPI4 中断	111	SPI_NOR 中断
36	JPEG 中断	74	Watch Dog0 中断	112	EPHY 中断
37	H264 中断	75	KEYSCAN 中断	113	IVE 中断
38	H265 中断	76	GPIO0 中断	114	保留
39	VC SBM 中断	77	GPIO1 中断	115	保留
40	ISP 中断	78	GPIO2 中断	116	SARADC 中断
41	SC_TOP 中断	79	GPIO3 中断	117	mbox 中断
42	CSI_MAC0 中断	80	Wiegand0 中断	151	npmuirq[0]
43	CSI_MAC1 中断	81	Wiegand1 中断	152	ctiirq[0]
44	LDC 中断	82	Wiegand2 中断	159	nexterrirq
45	System DMA 中断	83	RTC MBOX 中断	116	SARADC 中断
46	USB 中断	84	N/A		
47	Ethnet0 中断	85	RTC IRRX 中断		
48	Ethnet0 中断	86	RTC GPIO 中断		
49	EMMC Wakup 中断	87	RTC UART 中断		
50	EMMC 中断	88	RTC SPI_NOR 中断		
51	SD0 Wakup 中断	89	RTC I2C 中断		
52	SD0 中断	90	RTC WDG 中断		
53	SD1 Wakup 中断	91	TPU 中断		
54	SD1 中断	92	TDMA 中断		
55	SPI_NAND 中断	93	保留		
56	I2S0 中断	94	保留		
57	I2S1 中断	95	Timer0 中断		
58	I2S2 中断	96	Timer1 中断		
59	I2S3 中断	97	Timer2 中断		
60	UART0 中断	98	Timer3 中断		
61	UART1 中断	99	Timer4 中断		
62	UART2 中断	100	Timer5 中断		
63	UART3 中断	101	Timer6 中断		
64	UART4 中断	102	Timer7 中断		
65	I2C0 中断	103	peri_firewall 中断		
66	I2C1 中断	104	hsperi_firewall 中断		
67	I2C2 中断	105	ddr_fw 中断		
68	I2C3 中断	106	rom_firewall 中断		

69	I2C4 中断	107	SPACC 中断		

表格 3-2 主处理器 RISCV C906 @ 1.0Ghz 中断号与中断源映射表

中断号	中断源	中断号	中断源	中断号	中断源
16	TEMPSENS 中断	52	I2C3 中断	88	hsperi_firewall 中断
17	RTC Alarm 中断	53	I2C4 中断	89	ddr_fw 中断
18	RTC Longpress 中断	54	SPI1 中断	90	rom_firewall 中断
19	VBAT DET 中断	55	SPI2 中断	91	SPACC 中断
20	JPEG 中断	56	SPI3 中断	92	TRNG 中断
21	H264 中断	57	SPI4 中断	93	ddr_axi_mon 中断
22	H265 中断	58	Watch Dog1 中断	94	ddr_pi_phy 中断
23	VC SBM 中断	59	KEYSCAN 中断	95	SPI_NOR 中断
24	ISP 中断	60	GPIO0 中断	96	EPHY 中断
25	SC_TOP 中断	61	GPIO1 中断	97	IVE 中断
26	CSI_MAC0 中断	62	GPIO2 中断	98	保留
27	CSI_MAC1 中断	63	GPIO3 中断	99	保留
28	LDC 中断	64	Wiegand0 中断	100	SARADC 中断
29	System DMA 中断	65	Wiegand1 中断	101	mbox 中断
30	USB 中断	66	Wiegand2 中断		
31	Ethnet0 中断	67	RTC MBOX 中断		
32	Ethnet0 中断	68	N/A		
33	EMMC Wakup 中断	69	RTC IRRX 中断		
34	EMMC 中断	70	RTC GPIO 中断		
35	SD0 Wakup 中断	71	RTC UART 中断		
36	SD0 中断	72	RTC SPI_NOR 中断		
37	SD1 Wakup 中断	73	RTC I2C 中断		
38	SD1 中断	74	RTC WDG 中断		
39	SPI_NAND 中断	75	TPU 中断		
40	I2S0 中断	76	TDMA 中断		
41	I2S1 中断	77	保留		
42	I2S2 中断	78	保留		
43	I2S3 中断	79	Timer0 中断		
44	UART0 中断	80	Timer1 中断		
45	UART1 中断	81	Timer2 中断		
46	UART2 中断	82	Timer3 中断		
47	UART3 中断	83	Timer4 中断		
48	UART4 中断	84	Timer5 中断		
49	I2C0 中断	85	Timer6 中断		
50	I2C1 中断	86	Timer7 中断		
51	I2C2 中断	87	peri_firewall 中断		

表格 3-3 协处理器 RISCV C906 @ 700Mhz 中断号与中断源映射表

中断号	中断源	中断号	中断源
16	JPEG 中断	54	保留
17	H264 中断	55	Timer4 中断
18	H265 中断	56	Timer5 中断
19	VC SBM 中断	57	Timer6 中断
20	ISP 中断	58	Timer7 中断
21	SC_TOP 中断	59	SPACC 中断
22	CSI_MAC0 中断	60	IVE 中断
23	CSI_MAC1 中断	61	mbox 中断
24	LDC 中断		
25	System DMA 中断		
26	I2S0 中断		
27	I2S1 中断		
28	I2S2 中断		
29	I2S3 中断		
30	UART0 中断		
31	UART1 中断		
32	I2C0 中断		
33	I2C1 中断		
34	I2C2 中断		
35	I2C3 中断		
36	I2C4 中断		
37	SPI1 中断		
38	SPI2 中断		
39	Watch Dog2 中断		
40	KEYSCAN 中断		
41	GPIO0 中断		
42	GPIO1 中断		
43	GPIO2 中断		
44	GPIO3 中断		
45	Wiegand0 中断		
46	RTC MBOX 中断		
47	RTC IRRX 中断		
48	RTC GPIO 中断		
49	RTC UART 中断		
50	RTC I2C 中断		
51	RTC WDG 中断		
52	TDMA 中断		
53	保留		

3.3 地址空间映射

起始地址 [31:0]	結束地址 [31:0]	空間功能	空間大小 (Byte)
0x01000000	0x017FFFFF	保留	8M
0x01800000	0x018FFFFF	保留	
0x01900000	0x01900FFF	ap_mailbox	4K
0x01901000	0x01901FFF	ap_system_ctrl	4K
0x01902000	0x019EFFFF	保留	
0x01F00000	0x01F0FFFF	保留	64K
0x01F10000	0x01FFFFFF	保留	
0x02000000	0x02FFFFFF	保留	64K
0x03000000	0x03000FFF	TOP_MISC 控制寄存器	4K
0x03001000	0x03001FFF	PINMUX 控制寄存器	4K
0x03002000	0x03002FFF	CLKGEN/PLL 控制寄存器	4K
0x03003000	0x03003FFF	RSTGEN 控制寄存器	4K
0x03004000	0x03005FFF	保留	
0x03006000	0x03006FFF	保留	4K
0x03007000	0x03008FFF	保留	
0x03009000	0x03009FFF	保留	4K
0x0300A000	0x0300AFFF	保留	4K
0x0300B000	0x0300FFFF	保留	
0x03010000	0x03010FFF	WATCH DOG0 控制寄存器	4K
0x03011000	0x03011FFF	WATCH DOG1 控制寄存器	4K
0x03012000	0x03012FFF	WATCH DOG2 控制寄存器	4K
0x03020000	0x03020FFF	GPIO0 控制寄存器	4K
0x03021000	0x03021FFF	GPIO1 控制寄存器	4K
0x03022000	0x03022FFF	GPIO2 控制寄存器	4K
0x03023000	0x03023FFF	GPIO3 控制寄存器	4K
0x03024000	0x0302FFFF	保留	
0x03030000	0x03030FFF	WGN0 控制寄存器	4K
0x03031000	0x03031FFF	WGN1 控制寄存器	4K
0x03032000	0x03032FFF	WGN2 控制寄存器	4K
0x03033000	0x0303FFFF	保留	
0x03040000	0x0304FFFF	KEYSCAN 控制寄存器	64K
0x03050000	0x0305FFFF	EFUSE 控制寄存器	64K
0x03060000	0x03060FFF	PWM0 控制寄存器	4K
0x03061000	0x03061FFF	PWM1 控制寄存器	4K
0x03062000	0x03062FFF	PWM2 控制寄存器	4K
0x03063000	0x03063FFF	PWM3 控制寄存器	4K
0x03064000	0x0309FFFF	保留	

0x030A0000	0x030AFFFF	TIMER 控制寄存器	64K
0x030C0000	0x030CFFFF	保留	
0x030D0000	0x030D0FFF	保留	4K
0x030D1000	0x030D1FFF	保留	4K
0x030D2000	0x030D2FFF	保留	4K
0x030D3000	0x030DFFFF	保留	
0x030E0000	0x030EFFFF	TEMPSEN 控制寄存器	64K
0x030F0000	0x030FFFFF	SARADC 控制寄存器	64K
0x04000000	0x0400FFFF	I2C0 控制寄存器	64K
0x04010000	0x0401FFFF	I2C1 控制寄存器	64K
0x04020000	0x0402FFFF	I2C2 控制寄存器	64K
0x04030000	0x0403FFFF	I2C3 控制寄存器	64K
0x04040000	0x0404FFFF	I2C4 控制寄存器	64K
0x04050000	0x0405FFFF	保留	
0x04060000	0x0406FFFF	SPI_NAND 控制寄存器	64K
0x04070000	0x0407FFFF	ETH0 控制寄存器	
0x04080000	0x040FFFFF	保留	
0x04100000	0x04107FFF	I2S0 控制寄存器	64K
0x04108000	0x0410FFFF	I2S Global 控制寄存器	64K
0x04110000	0x0411FFFF	I2S1 控制寄存器	64K
0x04120000	0x0412FFFF	I2S2 控制寄存器	64K
0x04130000	0x0413FFFF	I2S3 控制寄存器	64K
0x04140000	0x0414FFFF	UART0 控制寄存器	64K
0x04150000	0x0415FFFF	UART1 控制寄存器	64K
0x04160000	0x0416FFFF	UART2 控制寄存器	64K
0x04170000	0x0417FFFF	UART3 控制寄存器	64K
0x04180000	0x0418FFFF	SPI0 控制寄存器	64K
0x04190000	0x0419FFFF	SPI1 控制寄存器	64K
0x041A0000	0x041AFFFF	SPI2 控制寄存器	64K
0x041B0000	0x041BFFFF	SPI3 控制寄存器	64K
0x041C0000	0x041CFFFF	UART4 控制寄存器	64K
0x041D0000	0x041DFFFF	AUDSRC 控制寄存器	64K
0x041E0000	0x042FFFFF	保留	
0x04300000	0x0430FFFF	eMMC 控制寄存器	64K
0x04310000	0x0431FFFF	SD0 控制寄存器	64K
0x04320000	0x0432FFFF	SD1 控制寄存器	
0x04330000	0x0433FFFF	DMA 控制寄存器	64K
0x04340000	0x0434FFFF	USB 控制寄存器	64K
0x04350000	0x043FFFFF	保留	
0x04400000	0x0441FFFF	ROM 记忆体空间	128K
0x04420000	0x04FFFFFF	保留	

0x05000000	0x05000FFF	保留	4KB
0x05020000	0x05020FFF	RTCSYS_Timer 控制寄存器	4KB
0x05021000	0x05021FFF	RTCSYS_GPIO 控制寄存器	4KB
0x05022000	0x05022FFF	RTCSYS_UART 控制寄存器	4KB
0x05023000	0x05023FFF	RTCSYS_INTR 控制寄存器	4KB
0x05024000	0x05024FFF	RTCSYS_MBOX 控制寄存器	4KB
0x05025000	0x05025FFF	RTCSYS_CTRL 控制寄存器	4KB
0x05026000	0x05026FFF	RTCSYS_CORE	4KB
0x05027000	0x05027FFF	RTCSYS_IO 控制寄存器	4KB
0x05028000	0x05028FFF	RTCSYS_OSC 控制寄存器	4KB
0x05029000	0x05029FFF	保留	4KB
0x0502A000	0x0502AFFF	RTCSYS_32kless 控制寄存器	4KB
0x0502B000	0x0502BFFF	RTCSYS_I2C 控制寄存器	4KB
0x0502C000	0x0502CFFF	RTCSYS_SAR 控制寄存器	4KB
0x0502D000	0x0502DFFF	RTCSYS_WDT 控制寄存器	4KB
0x0502E000	0x0502EFFF	RTCSYS_IRRX 控制寄存器	4KB
0x05200000	0x053FFFFF	RTCSYS_SRAM	8KB
0x05400000	0x057FFFFF	RTCSYS_SPINOR	4MB
0x08000000	0x08001FFF	保留	8K
0x08004000	0x08005FFF	DDR Controler 控制寄存器	8K
0x08006000	0x08007FFF	保留	8K
0x08008000	0x08009FFF	DDR AXI Monitor 控制寄存器	8K
0x0800A000	0x0800BFFF	DDR Global 控制寄存器	8K
0x08010000	0x08011FFF	保留	8K
0x08012000	0x08013FFF	保留	8K
0x08014000	0x09FFFFFF	保留	
0x0A000000	0x0A07FFFF	ISP 控制寄存器	512K
0x0A080000	0x0A0803FF	sc_top 控制寄存器	1K
0x0A080400	0x0A080BFF	保留	2K
0x0A080C00	0x0A080CFF	osd enc 控制寄存器	256B
0x0A080D00	0x0A080FFF	保留	768B
0x0A081000	0x0A081FFF	保留	4K
0x0A082000	0x0A082FFF	img_v 控制寄存器	4K
0x0A083000	0x0A083FFF	img_d 控制寄存器	4K
0x0A084000	0x0A084FFF	sc_d 控制寄存器	4K
0x0A085000	0x0A085FFF	sc_v1 控制寄存器	4K
0x0A086000	0x0A086FFF	sc_v2 控制寄存器	4K
0x0A087000	0x0A087FFF	sc_v3 控制寄存器	4K
0x0A088000	0x0A088FFF	DISP 控制寄存器	4K
0x0A089000	0x0A089FFF	保留	4K
0x0A08A000	0x0A08AFFF	dsi_mac 控制寄存器	4K

0x0A08B000	0x0A08BFFF	cmdq 控制寄存器	4K
0x0A08C000	0x0A08CFFF	保留	4K
0x0A08D000	0x0A08DFFF	保留	4K
0x0A08E000	0x0A09FFFF	保留	72K
0x0A0A0000	0x0A0AFFFF	IVE 控制寄存器	64K
0x0A0A0000	0x0A0BFFFF	保留	64K
0x0A0C0000	0x0A0C1FFF	ldc 控制寄存器	8K
0x0A0C2000	0x0A0C3FFF	VI0/MIPI_RX0 控制寄存器	8K
0x0A0C4000	0x0A0C5FFF	VI1/MIPI_RX1 控制寄存器	8K
0x0A0C6000	0x0A0C7FFF	VI2/MIPI_RX2 控制寄存器	8K
0x0A0C8000	0x0A0C9FFF	VIPSYS 控制寄存器	8K
0x0A0CA000	0x0A0CFFFF	保留	24K
0x0A0D0000	0x0A0D0FFF	CSI_PHY 控制寄存器	4K
0x0A0D1000	0x0A0D1FFF	DSI_PHY 控制寄存器	4K
0x0A0D2000	0x0AFFFFFF	保留	
0x0B000000	0x0B00FFFF	JPEG codec 控制寄存器	64K
0x0B010000	0x0B01FFFF	H.264 codec 控制寄存器	64K
0x0B020000	0x0B02FFFF	H.265 codec 控制寄存器	64K
0x0B030000	0x0BFFFFFF	保留	
0x0C000000	0x0FFFFFFF	保留	
0x10000000	0x1FFFFFFF	SPI_NOR 记忆体空间	256M
0x30000000	0x7FFFFFFF	保留	
0x80000000	0xFFFFFFFF	DDR 记忆体空间	4Gb

*对保留地址空间进行读写操作，将可能产生无法预期之结果

3.4 封装与管脚分布

3.4.1 封装

使用 IFBGA 封装, 封装尺寸为 10mmx10mmx1.9mm. 管脚间距为 0.65mm. 管脚总数为 205 个 。

	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	1.3
STAND OFF	A1	0.16	---	0.26
SUBSTRATE THICKNESS	A2		0.26	REF
MOLD THICKNESS	A3		0.7	REF
BODY SIZE	D		10	BSC
	E		10	BSC
BALL DIAMETER			0.3	
BALL OPENING			0.275	
BALL WIDTH	b	0.27	---	0.37
BALL PITCH	e		0.65	BSC
BALL COUNT	n		205	
EDGE BALL CENTER TO CENTER	D1		9.1	BSC
	E1		9.1	BSC
BODY CENTER TO CONTACT BALL	SD		---	BSC
	SE		---	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.2	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

*图表 **3-1** 封装外观尺寸, 上视图*

	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	1.3
STAND OFF	A1	0.16	---	0.26
SUBSTRATE THICKNESS	A2		0.26	REF
MOLD THICKNESS	A3		0.7	REF
BODY SIZE	D		10	BSC
	E		10	BSC
BALL DIAMETER			0.3	
BALL OPENING			0.275	
BALL WIDTH	b	0.27	---	0.37
BALL PITCH	e		0.65	BSC
BALL COUNT	n		205	
EDGE BALL CENTER TO CENTER	D1		9.1	BSC
	E1		9.1	BSC
BODY CENTER TO CONTACT BALL	SD		---	BSC
	SE		---	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.2	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

*图表 **3-2.**封装外观尺寸, 底视图*

管脚分布

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A		PAD_MIPI_TXM4	VSSC	PAD_MIPIRX0N		PAD_MIPIRX3P	PAD_MIPIRX4P		VIVO_D2	VIVO_D3		VIVO_D10	USB_VBUS_DET	USB_DP		A
B	PAD_MIPI_TXP3	PAD_MIPI_TXM3	PAD_MIPI_TXP4	PAD_MIPIRX0P	PAD_MIPIRX1N	PAD_MIPIRX2N	PAD_MIPIRX4N	PAD_MIPIRX5N	VIVO_D1	VIVO_D5	VIVO_D7	VIVO_D9	USB_ID	USB_DM	PAD_ETH_RXM___EPHY_TXP	B
C	PAD_MIPI_TXP2	PAD_MIPI_TXM2	CAM_PD0	CAM_MCLK0	PAD_MIPIRX1P	PAD_MIPIRX2P	PAD_MIPIRX3N	PAD_MIPIRX5P	VIVO_CLK	VIVO_D6	VIVO_D8	USB_VBUS_EN	VSSC	PAD_ETH_RXP___EPHY_TXN	GPIO_RTX___EPHY_RTX	C
D	PAD_MIPI_TXP1	PAD_MIPI_TXM1	CAM_MCLK1	IIC3_SCL	VSSC	VDD18A_MIPI	VDD18A_MIPI	VSSC	VSSC	VIVO_D4	VDD18A_PLL_N	VCC18A_USB	VSSC	PAD_ETH_TXM___EPHY_RXP	PAD_ETH_TXP___EPHY_RXN	D
E	PAD_MIPI_TXP0	PAD_MIPI_TXM0		CAM_PD1	CAM_RST0		VSSC	VDDCAP18_VIVO	VSSC	VIVO_D0	VDDIO18_1	VDD18A_ETH	ADC1	ADC2	ADC3	E
F		PAD_AUD_AOUTL	VSSC	IIC3_SDA	VDD18A_AUD	VSSC	VSSC		VDDIO_VIVO	VDDIO_SD1	VSSC		VSSC	SD1_D2		F
G	PAD_AUD_AVREF	PAD_AUD_AOUTR	VSS18A_AUD	PAD_VOUT	VSSC	VSSC	VSSC	VSSC	VSSC	VSSC	VDD33A_ETH		SD1_D3	SD1_CLK	SD1_CMD	G
H	PAD_AUD_AINL_MIC	VSS18A_AUD	VIN3V	VDDIO_SD0	VSSC	VSSC	VSSC	VSSC	VSSC	VCC33A_USB	VDD09A_ETH	RSTN	PWM0_BUCK	SD1_D1	SD1_D0	H
J	PAD_AUD_AINR_MIC	VSS18A_AUD	VSS18A_AUD	VDDC		VDD09A_MIPI	VSSC	VSSC	VSSC	VDDC	VDDC	VDDC	IIC2_SCL	IIC2_SDA		J
K		SD0_CD	SD0_D1	VDDC	VDDC	VDDC	VSSC	VSSC	VSSC	VDDC	VDDC	VDDC	UART2_RX	UART2_CTS	UART2_TX	K
L	SD0_CLK	SD0_D0	SD0_CMD	VDDC	VDDC	VSSC	VSSC	VSSC	VSSC	VDDC	VDDC_RTC		PTEST	CLK32K	UART2_RTS	L
M	SD0_D3	SD0_D2	VDDIO_EMMC	UART0_RX	UART0_TX	JTAG_CPU_TRST	VSSC	VDDQ_DRAM	VDDQ_DRAM	VDDIO_RTC	PWR_ON	PWR_GPIO2	PWR_GPIO0	CLK25M		M
N	SD0_PWR_EN	VSSC	SPK_EN	JTAG_CPU_TCK	VDDQ_DRAM	JTAG_CPU_TMS	VSSC	VSSC	VDDQ	VDD18A	PWR_WAKEUP1	PWR_WAKEUP0	PWR_GPIO1	VDDBKUP	RTC_XIN	N
P	EMMC_DAT3	EMMC_DAT0	EMMC_DAT2	EMMC_RSTN	AUX0	IIC0_SDA	VSSC	VDDQ	VDDQ	PWR_SEQ3	PWR_VBAT_DET	PWR_SEQ1	PWR_BUTTON1	VSSC	RTC_XOUT	P
R		EMMC_DAT1	EMMC_CMD	EMMC_CLK		IIC0_SCL	VSSC	VDDQ	VDDQ	GPIO_ZQ___PAD_ZQ	PWR_RSTN	PWR_SEQ2	XTAL_XIN	XTAL_XOUT		R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

图表 3-3 管脚分布图.

3.4.2 管脚信息描述

表格 3-4 管脚信息表

Pin Num	Pin Name	IO Type	IOGroup	PowerDomain	Note
C4	CAM_MCLK0	1.8V GPIO	G11	VDD18A_MIPI	
C5	CAM_PD0	1.8V GPIO	G11	VDD18A_MIPI	
C6	CAM_RST0	1.8V GPIO	G11	VDD18A_MIPI	
C7	CAM_MCLK1	1.8V GPIO	G11	VDD18A_MIPI	
C8	CAM_PD1	1.8V GPIO	G11	VDD18A_MIPI	
D4	IIC3_SCL	1.8V GPIO	G11	VDD18A_MIPI	
F4	IIC3_SDA	1.8V GPIO	G11	VDD18A_MIPI	

L1	SD0_CLK	18OD33 IO	G10	VDDIO_SD0	
L3	SD0_CMD	18OD33 IO	G10	VDDIO_SD0	
L2	SD0_D0	18OD33 IO	G10	VDDIO_SD0	
K3	SD0_D1	18OD33 IO	G10	VDDIO_SD0	
M2	SD0_D2	18OD33 IO	G10	VDDIO_SD0	
M1	SD0_D3	18OD33 IO	G10	VDDIO_SD0	
K2	SD0_CD	18OD33 IO	G7	VDDIO_EMMC	
N1	SD0_PWR_EN	18OD33 IO	G7	VDDIO_EMMC	
N3	SPK_EN	18OD33 IO	G7	VDDIO_EMMC	
M5	UART0_TX	18OD33 IO	G7	VDDIO_EMMC	
M4	UART0_RX	18OD33 IO	G7	VDDIO_EMMC	
P4	EMMC_RSTN	18OD33 IO	G7	VDDIO_EMMC	
P3	EMMC_DAT2	18OD33 IO	G7	VDDIO_EMMC	
R4	EMMC_CLK	18OD33 IO	G7	VDDIO_EMMC	
P2	EMMC_DAT0	18OD33 IO	G7	VDDIO_EMMC	
P1	EMMC_DAT3	18OD33 IO	G7	VDDIO_EMMC	
R3	EMMC_CMD	18OD33 IO	G7	VDDIO_EMMC	
R2	EMMC_DAT1	18OD33 IO	G7	VDDIO_EMMC	
N6	JTAG_CPU_TMS	18OD33 IO	G7	VDDIO_EMMC	
N4	JTAG_CPU_TCK	18OD33 IO	G7	VDDIO_EMMC	
M6	JTAG_CPU_TRST	18OD33 IO	G7	VDDIO_EMMC	
R6	IIC0_SCL	18OD33 IO	G7	VDDIO_EMMC	
P6	IIC0_SDA	18OD33 IO	G7	VDDIO_EMMC	
P5	AUX0	18OD33 IO	G7	VDDIO_EMMC	
P11	PWR_VBAT_DET	1.8V GPIO	GRTC	VDDIO_RTC	
R11	PWR_RSTN	1.8V GPIO	GRTC	VDDIO_RTC	

P12	PWR_SEQ1	1.8V GPIO	GRTC	VDDIO_RTC	
R12	PWR_SEQ2	1.8V GPIO	GRTC	VDDIO_RTC	
P10	PWR_SEQ3	1.8V GPIO	GRTC	VDDIO_RTC	
N12	PWR_WAKEUP0	1.8V GPIO	GRTC	VDDIO_RTC	
N11	PWR_WAKEUP1	1.8V GPIO	GRTC	VDDIO_RTC	
P13	PWR_BUTTON1	1.8V GPIO	GRTC	VDDIO_RTC	
M11	PWR_ON	1.8V GPIO	GRTC	VDDIO_RTC	
R13	XTAL_XIN	Xtal	GRTC	VDDIO_RTC	
M13	PWR_GPIO0	1.8V GPIO	GRTC	VDDIO_RTC	
N13	PWR_GPIO1	1.8V GPIO	GRTC	VDDIO_RTC	
M12	PWR_GPIO2	1.8V GPIO	GRTC	VDDIO_RTC	
L14	CLK32K	1.8V GPIO	GRTC	VDDIO_RTC	
M14	CLK25M	1.8V GPIO	GRTC	VDDIO_RTC	
J13	IIC2_SCL	1.8V GPIO	GRTC	VDDIO_RTC	
J14	IIC2_SDA	1.8V GPIO	GRTC	VDDIO_RTC	
K15	UART2_TX	1.8V GPIO	GRTC	VDDIO_RTC	
L15	UART2_RTS	1.8V GPIO	GRTC	VDDIO_RTC	
K13	UART2_RX	1.8V GPIO	GRTC	VDDIO_RTC	
K14	UART2_CTS	1.8V GPIO	GRTC	VDDIO_RTC	
G13	SD1_D3	18OD33 IO	GRTC	VDDIO_SD1	SD1_D3 or VO[32]Check 0x0502_70E4
F14	SD1_D2	18OD33 IO	GRTC	VDDIO_SD1	SD1_D2 or VO[33] Check 0x0502_70E4
H14	SD1_D1	18OD33 IO	GRTC	VDDIO_SD1	SD1_D1 or VO[34] Check 0x0502_70E4
H15	SD1_D0	18OD33 IO	GRTC	VDDIO_SD1	SD1_D0 or VO[35] Check 0x0502_70E4

G15	SD1_CMD	18OD33 IO	GRTC	VDDIO_SD1	SD1_CMD or VO[36] Check 0x0502_70E4
G14	SD1_CLK	18OD33 IO	GRTC	VDDIO_SD1	SD1_CLK or VO[37]Check 0x0502_70E4
H12	RSTN	1.8V GPIO	G1	VDDIO18_1	
H13	PWM0_BUCK	1.8V GPIO	G1	VDDIO18_1	
E15	ADC3	1.8V GPIO	G1	VDDIO18_1	
E14	ADC2	1.8V GPIO	G1	VDDIO18_1	
E13	ADC1	1.8V GPIO	G1	VDDIO18_1	
B13	USB_ID	1.8V GPIO	G1	VDDIO18_1	
C12	USB_VBUS_EN	1.8V GPIO	G1	VDDIO18_1	
A13	USB_VBUS_DET	1.8V GPIO	G1	VDDIO18_1	
D15	PAD_ETH_TXP___EPHY_RXN	ETH GPIO(1.8V)		VDD18A_EPHY	
D14	PAD_ETH_TXM___EPHY_RXP	ETH GPIO (1.8V)		VDD18A_EPHY	
C14	PAD_ETH_RXP___EPHY_TXN	ETH GPIO (1.8V)		VDD18A_EPHY	
B15	PAD_ETH_RXM___EPHY_TXP	ETH GPIO (1.8V)		VDD18A_EPHY	
A12	VIVO_D10	18OD33 IO	G12	VDDIO_VIVO	
B12	VIVO_D9	18OD33 IO	G12	VDDIO_VIVO	
C11	VIVO_D8	18OD33 IO	G12	VDDIO_VIVO	
B11	VIVO_D7	18OD33 IO	G12	VDDIO_VIVO	
C10	VIVO_D6	18OD33 IO	G12	VDDIO_VIVO	
B10	VIVO_D5	18OD33 IO	G12	VDDIO_VIVO	
D10	VIVO_D4	18OD33 IO	G12	VDDIO_VIVO	
A10	VIVO_D3	18OD33 IO	G12	VDDIO_VIVO	
A9	VIVO_D2	18OD33 IO	G12	VDDIO_VIVO	
B9	VIVO_D1	18OD33 IO	G12	VDDIO_VIVO	

E10	VIVO_D0	18OD33 IO	G12	VDDIO_VIVO	
C9	VIVO_CLK	18OD33 IO	G12	VDDIO_VIVO	
B8	PAD_MIPIRX5N	1.8V GPIO	G12	VDD18A_MIPI	
C8	PAD_MIPIRX5P	1.8V GPIO	G12	VDD18A_MIPI	
B7	PAD_MIPIRX4N	1.8V GPIO	G12	VDD18A_MIPI	
A7	PAD_MIPIRX4P	1.8V GPIO	G12	VDD18A_MIPI	
C7	PAD_MIPIRX3N	1.8V GPIO	G12	VDD18A_MIPI	
A6	PAD_MIPIRX3P	1.8V GPIO	G12	VDD18A_MIPI	
B6	PAD_MIPIRX2N	1.8V GPIO	G12	VDD18A_MIPI	
C6	PAD_MIPIRX2P	1.8V GPIO	G12	VDD18A_MIPI	
B5	PAD_MIPIRX1N	1.8V GPIO	G12	VDD18A_MIPI	
C5	PAD_MIPIRX1P	1.8V GPIO	G12	VDD18A_MIPI	
A4	PAD_MIPIRX0N	1.8V GPIO	G12	VDD18A_MIPI	
B4	PAD_MIPIRX0P	1.8V GPIO	G12	VDD18A_MIPI	
A2	PAD_MIPI_TXM4	1.8V GPIO	G12	VDD18A_MIPI	
B3	PAD_MIPI_TXP4	1.8V GPIO	G12	VDD18A_MIPI	
B2	PAD_MIPI_TXM3	1.8V GPIO	G12	VDD18A_MIPI	
B1	PAD_MIPI_TXP3	1.8V GPIO	G12	VDD18A_MIPI	
C2	PAD_MIPI_TXM2	1.8V GPIO	G12	VDD18A_MIPI	
C1	PAD_MIPI_TXP2	1.8V GPIO	G12	VDD18A_MIPI	
D2	PAD_MIPI_TXM1	1.8V GPIO	G12	VDD18A_MIPI	
D1	PAD_MIPI_TXP1	1.8V GPIO	G12	VDD18A_MIPI	
E2	PAD_MIPI_TXM0	1.8V GPIO	G12	VDD18A_MIPI	
E1	PAD_MIPI_TXP0	1.8V GPIO	G12	VDD18A_MIPI	
H1	PAD_AUD_AINL_MIC	AUDIO GPIO (1.8V)		VDD18A_MIPI	

J1	PAD_AUD_AINR_MIC	AUDIO GPIO (1.8V)		VDD18A_MIPI	
F2	PAD_AUD_AOUTL	AUDIO GPIO (1.8V)		VDD18A_MIPI	
G2	PAD_AUD_AOUTR	AUDIO GPIO (1.8V)		VDD18A_MIPI	
C15	GPIO_RTX___EPHY_RTX	1.8V GPIO	G12	VDDIO18_1	
R10	GPIO_ZQ___PAD_ZQ	1.8V GPIO	GRTC	VDDIO_RTC	
#N/A	PKG_TYPE0	1.8V GPIO	G1	VDDIO18_1	no pin out
#N/A	PKG_TYPE1	1.8V GPIO	G1	VDDIO18_1	no pin out
#N/A	PKG_TYPE2	1.8V GPIO	G1	VDDIO18_1	no pin out
#N/A	MUX_SPI1_MISO	NA	NA	NA	Result is feed to PAD_MIPIRX3N func7
#N/A	MUX_SPI1_MOSI	NA	NA	NA	Result is feed to PAD_MIPIRX3P func7
#N/A	MUX_SPI1_CS	NA	NA	NA	Result is feed to PAD_MIPIRX4P func7
#N/A	MUX_SPI1_SCK	NA	NA	NA	Result is feed to PAD_MIPIRX4N func7

3.4.3 管脚默认状态(IFGBA)

表格 3-5 管脚默认状态表

PinNum	PinName	Default Function	Default Function Select	Default Pull (PU/PD/NO PULL)	Default Function Direction (During reset)	IO TRAP (become input during reset)	IO Type	PowerDomain	IOBLK_ Group	Fail-safe
C4	CAM_MCLK0	XGPIOA[0]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	Fail-safe
C3	CAM_PD0	XGPIOA[1]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	
E5	CAM_RST0	XGPIOA[2]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	Fail-safe
D3	CAM_MCLK1	XGPIOA[3]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	Fail-safe
E4	CAM_PD1	XGPIOA[4]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	
D4	IIC3_SCL	XGPIOA[5]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	Fail-safe
F4	IIC3_SDA	XGPIOA[6]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G11	Fail-safe
L1	SD0_CLK	SDIO0_CLK	0	PD (61Kohm)-> DRV_LO + PD	Output (Low)		18OD33	VDDIO_SD0	G10	
L3	SD0_CMD	SDIO0_CMD	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0	G10	
L2	SD0_D0	SDIO0_D[0]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0	G10	
K3	SD0_D1	SDIO0_D[1]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0	G10	
M2	SD0_D2	SDIO0_D[2]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0	G10	
M1	SD0_D3	SDIO0_D[3]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0	G10	
K2	SD0_CD	SDIO0_CD	0	PU (60Kohm)	Input		18OD33	VDDIO_EMMC	G7	
N1	SD0_PWR_EN	XGPIOA[14]	3	PD (61Kohm) -> DRV_HI/LO + PD	Input (Trap) -> Output	V	18OD33	VDDIO_EMMC	G7	
N3	SPK_EN	XGPIOA[15]	3	PD (61Kohm)	Input		18OD33	VDDIO_EMMC	G7	
M5	UART0_TX	UART0_TX	0	PU (60Kohm) -> DRV_HI + PU	Input (Trap) -> Output	V	18OD33	VDDIO_EMMC	G7	
M4	UART0_RX	UART0_RX	0	PU (60Kohm)	Input		18OD33	VDDIO_EMMC	G7	
P4	EMMC_RSTN	EMMC_RSTN	0	PD (61Kohm) -> DRV_HI + PD	Output		18OD33	VDDIO_EMMC	G7	
P3	EMMC_DAT2	SPINOR_HOLD_X	1	PD (61Kohm) -> DRV_HI + PD	Input		18OD33	VDDIO_EMMC	G7	
R4	EMMC_CLK	SPINOR_SCK	1	DRV_LO + PD (61Ko	Output (Low)		18OD33	VDDIO_EMMC	G7	
P2	EMMC_DAT0	SPINOR_MOSI	1	PU (60Kohm)	Input (Trap)	V	18OD33	VDDIO_EMMC	G7	
P1	EMMC_DAT3	SPINOR_WP_X	1	PU (60Kohm) ->9 DRV_HI + PU	Input (Trap)	V	18OD33	VDDIO_EMMC	G7	
R3	EMMC_CMD	SPINOR_MISO	1	PU (60Kohm)	Input		18OD33	VDDIO_EMMC	G7	
R2	EMMC_DAT1	SPINOR_CS_X	1	DRV_HI + PD (61Kol	Output (High)		18OD33	VDDIO_EMMC	G7	
N6	JTAG_CPU_TMS	CR_4WTMS	0	PD (61Kohm)	Input		18OD33	VDDIO_EMMC	G7	
N4	JTAG_CPU_TCK	CR_4WTCK	0	PD (61Kohm)	Input		18OD33	VDDIO_EMMC	G7	
M6	JTAG_CPU_TRST	JTAG_CPU_TRST	0	PU (60Kohm)	Input		18OD33	VDDIO_EMMC	G7	
R6	IIC0_SCL	CR_4WTDI	0	PU (60Kohm)	Input		18OD33	VDDIO_EMMC	G7	
P6	IIC0_SDA	CR_4WTDO	0	PU (60Kohm)	Output (CR)		18OD33	VDDIO_EMMC	G7	
P5	AUX0	XGPIOA[30]	3	PD (61Kohm)	Input		18OD33	VDDIO_EMMC	G7	
P11	PWR_VBAT_DET	PWR_VBAT_DET	0	NO PULL	Input		1.8V_IO	VDDIO_RTC	GRTC	
R11	PWR_RSTN	PWR_RSTN	0	PU (60Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
P12	PWR_SEQ1	PWR_SEQ1	0	PD (87Kohm) -> DRV + PD	Output (L->H)		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
R12	PWR_SEQ2	PWR_SEQ2	0	PD (87Kohm) -> DRV + PD	Output (L->H)		1.8V_IO	VDDIO_RTC	GRTC	
P10	PWR_SEQ3	PWR_SEQ3	0	PD (87Kohm) -> DRV	Output (L->H)		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
L13	PTEST	PWR_PTEST	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
N12	PWR_WAKEUP0	PWR_WAKEUP0	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
N11	PWR_WAKEUP1	PWR_WAKEUP1	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
P13	PWR_BUTTON1	PWR_BUTTON1	0	PU (60Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
M11	PWR_ON	PWR_ON	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
M13	PWR_GPIO0	PWR_GPIO[0]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
N13	PWR_GPIO1	PWR_GPIO[1]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	
M12	PWR_GPIO2	PWR_GPIO[2]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	
L14	CLK32K	CLK32K	0	PD (87Kohm)	Output (Clock)		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
M14	CLK25M	CLK25M	0	PD (87Kohm)	Output (Clock)		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
J13	IIC2_SCL	PWR_GPIO[12]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
J14	IIC2_SDA	PWR_GPIO[13]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
K15	UART2_TX	PWR_GPIO[14]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
L15	UART2_RTS	PWR_GPIO[15]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
K13	UART2_RX	PWR_GPIO[16]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
K14	UART2_CTS	PWR_GPIO[17]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
G13	SD1_D3	PWR_SPINOR1_CS_X	6	DRV_HI + PD (61Kohm)	Output (High)		18OD33	VDDIO_SD1	GRTC	
F14	SD1_D2	PWR_SPINOR1_HOLD_X	6	PD (61Kohm) -> DRV_HI + PD	Input		18OD33	VDDIO_SD1	GRTC	
H14	SD1_D1	PWR_SPINOR1_WP_X	6	61Kohm) -> DRV_HI	Input		18OD33	VDDIO_SD1	GRTC	
H15	SD1_D0	PWR_SPINOR1_MISO	6	PD (61Kohm)	Input		18OD33	VDDIO_SD1	GRTC	
G15	SD1_CMD	PWR_SPINOR1_MOSI	6	PD (61Kohm)	Input		18OD33	VDDIO_SD1	GRTC	
G14	SD1_CLK	PWR_SPINOR1_SCK	6	RV_LO + PD (61Kohr	Output (Low)		18OD33	VDDIO_SD1	GRTC	
H12	RSTN	RSTN	0	PU (60Kohm)	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
H13	PWM0_BUCK	XGPIOB[0]	3	NO PULL	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
E15	ADC3	XGPIOB[1]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
E14	ADC2	XGPIOB[2]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
E13	ADC1	XGPIOB[3]	3	PD (87Kohm)	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
B13	USB_ID	USB_ID	0	PU (60Kohm)	Input		1.8V_IO	VDDIO18_1	G1	Fail-safe
C12	USB_VBUS_EN	USB_VBUS_EN	0	DRV_LO + PD (87Ko	Output (Low)		1.8V_IO	VDDIO18_1	G1	Fail-safe
A13	USB_VBUS_DET	USB_VBUS_DET	0	NO PULL	Input		1.8V_IO	VDDIO18_1	G1	
D15	PAD_ETH_TXP__	XGPIOB[25]	3	NO PULL			ETH_GPIO	VDD18A_EPHY	G12	
D14	PAD_ETH_TXM__	XGPIOB[24]	3	NO PULL			ETH_GPIO	VDD18A_EPHY	G12	
C14	PAD_ETH_RXP__	XGPIOB[27]	3	NO PULL			ETH_GPIO	VDD18A_EPHY	G12	
B15	PAD_ETH_RXM__	XGPIOB[26]	3	NO PULL			ETH_GPIO	VDD18A_EPHY	G12	
A12	VIVO_D10	XGPIOB[11]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
B12	VIVO_D9	XGPIOB[12]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
C11	VIVO_D8	XGPIOB[13]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
B11	VIVO_D7	XGPIOB[14]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
C10	VIVO_D6	XGPIOB[15]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
B10	VIVO_D5	XGPIOB[16]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	

算能科技

A9	VIVO_D2	XGPIOB[19]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
B9	VIVO_D1	XGPIOB[20]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
E10	VIVO_D0	XGPIOB[21]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
C9	VIVO_CLK	XGPIOB[22]	3	PD (61Kohm)	Input		18OD33	VDDIO_VIVO	G12	
B8	PAD_MIPIRX5N	XGPIOC[0]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C8	PAD_MIPIRX5P	XGPIOC[1]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B7	PAD_MIPIRX4N	CR_SCL0	0	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
A7	PAD_MIPIRX4P	CR_SDA0	0	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C7	PAD_MIPIRX3N	XGPIOC[4]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
A6	PAD_MIPIRX3P	XGPIOC[5]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B6	PAD_MIPIRX2N	XGPIOC[6]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C6	PAD_MIPIRX2P	XGPIOC[7]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B5	PAD_MIPIRX1N	XGPIOC[8]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C5	PAD_MIPIRX1P	XGPIOC[9]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
A4	PAD_MIPIRX0N	XGPIOC[10]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B4	PAD_MIPIRX0P	XGPIOC[11]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
A2	PAD_MIPI_TXM4	XGPIOC[18]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B3	PAD_MIPI_TXP4	XGPIOC[19]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B2	PAD_MIPI_TXM3	XGPIOC[20]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
B1	PAD_MIPI_TXP3	XGPIOC[21]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C2	PAD_MIPI_TXM2	XGPIOC[16]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
C1	PAD_MIPI_TXP2	XGPIOC[17]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
D2	PAD_MIPI_TXM1	XGPIOC[14]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
D1	PAD_MIPI_TXP1	XGPIOC[15]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
E2	PAD_MIPI_TXM0	XGPIOC[12]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
E1	PAD_MIPI_TXP0	XGPIOC[13]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
H1	PAD_AUD_AINL_N	XGPIOC[23]	3	NO PULL	Input		AUDIO_GPI	VDD18A_MIPI	G11	
J1	PAD_AUD_AINR_N	XGPIOC[22]	3	NO PULL	Input		AUDIO_GPI	VDD18A_MIPI	G11	
F2	PAD_AUD_AOUTL	XGPIOC[25]	3	NO PULL	Output		AUDIO_GPI	VDD18A_MIPI	G11	
G2	PAD_AUD_AOUTR	XGPIOC[24]	3	NO PULL	Output		AUDIO_GPI	VDD18A_MIPI	G11	
C15	GPIO_RTX___EPH	XGPIOB[23]	3	PD (87Kohm)	Input	V	1.8V_IO	VDDIO18_1	G12	Fail-safe
R10	GPIO_ZQ___PAD_	PWR_GPIO[24]	3	PD (87Kohm)	Input	V	1.8V_IO	VDDIO_RTC	GRTC	Fail-safe

4 启动和升级

4.1 概述

芯片由内置 ROM (BOOTROM) 启动. 芯片在复位时会通过侦测两个管脚(EMMC_DAT3, EMMC_DAT0) 上是否有弱上拉或弱下拉以确认目前选用使用的存储器件类型.
安全启动的芯片, 则会在开机及芯片升级时都会验签以确保执行或升级的软件是安全的.

4.2 启动模式和对应的信号锁存值对应关系

- 支持由 SPI Nor Flash 启动 (EMMC_DAT3 pull down, EMMC_DAT0 pull up)
- 支持由 SPI Nand Flash 启动 (EMMC_DAT3 pull down, EMMC_DAT0 pull down)
- 支持由 eMMC 启动. (EMMC_DAT3 pull up, EMMC_DAT0 pull up)

Note. 因为 SD0 与 eMMC domain 共用 IO power. 因为 SD 卡 default 3.0V, 而 eMMC 多为 1.8V. 所以基本上不支援 eMMC. 除非 SD0 不接 SD 卡.

4.3 镜像烧写模式.

- 支持通过 SD 卡烧写镜像。
- 支持通过 USB device mode 烧写镜像。
- 若 flash 中已有镜像, 软件支援由网路升级软件。

4.4 安全启动

- 支持安全启动及升级。
- AES/DES/SM4 硬件加解密。
- SHA/TRNG/Secure Efuse 安全硬件。

5 复位

5.1 概述

复位管理模块对整个芯片、子系统、各功能模块的复位与时序，进行统一的管理

5.2 复位控制

图表 5-1 复位管理模块框图

5.2.1 上电复位

上电复位(POR)由实时时钟模块配合全局电源管理与晶振时序生成。详情参照章节 12

5.2.2 系统硬复位

系统硬复位(System Hard Reset) 由 Reset Ctrl Level2 生成，对芯片全局包含子系统及各功能模块进行硬复位，来源由以下途径：

- 上电复位
- 看门狗复位
- 过热保护复位
- 外部复位管脚(RSTN)
    - 内建去抖动电路，RSTN 高低电平有效信号须达 6.56ms。

5.2.3 软复位

软复位控制通过配置相应的复位配置寄存器(Reset CRG)，由 Reset Ctrl Level3 实现，包含：

- 系统软复位 :复位全芯片，除少部分电路及 RTC 内部电路
- 处理器子系统复位 :复位处理器及处理器子系统。
- 功能子系统复位 :复位各功能子系统及功能模块
- 功能模块复位 : 复位各功能模块

5.2.4 处理器子系统软复位

操作寄存器 SOFT_AC_RSTN_0 可对处理器及子系统做软复位，配置寄存器写 0 后，复位控制器会等待 24us 延时后才触发相应处理器复位。这段期间处理器应结束对总线之访问，以避免复位后总线挂死。触发复位后对应之复位信号会持续 8us 后自动解除，处理器及处理器子系统完成复位并开始启动。

5.2.5 功能子系统及功能模块软复位

操作寄存器 SOFT_RSTN_0~3，可对各功能模块进行软复位。复位配置为低电平有效，复位信号并不会自动清除，故软件配置相应寄存器为 0 触发复位后，亦需配置为 1 解除复位。复位前须确保各功能模块及功能子系统内置 DMA 对总线访问与处理器对模块之访问处于闲置状态。否则将使复位失败易造成系统挂死。

5.3 复位配置寄存器

5.3.1 复位配置寄存器概览

Base address 0x03003000

Name	Address Offset	Description
SOFT_RSTN_0	0x000	soft-reset ctrl register 0
SOFT_RSTN_1	0x004	soft-reset ctrl register 1
SOFT_RSTN_2	0x008	soft-reset ctrl register 2
SOFT_RSTN_3	0x00c	soft-reset ctrl register 3
SOFT_CPUAC_RSTN	0x020	CPU auto clear soft-reset ctrl register
SOFT_CPU_RSTN	0x024	CPU soft-reset ctrl register

5.3.2 复位配置寄存器描述

SOFT_RSTN_0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	reg_soft_reset_x_ddr	R/W	DDR system software reset (active low)	0x1
3	reg_soft_reset_x_h264c	R/W	H264 IP software reset (active low)	0x1
4	reg_soft_reset_x_jpeg	R/W	JPEG IP software reset (active low)	0x1
5	reg_soft_reset_x_h265c	R/W	H265 IP software reset (active low)	0x1
6	reg_soft_reset_x_vipsys	R/W	VIP system software reset (active low)	0x1
7	reg_soft_reset_x_tdma	R/W	TPU_DMA IP software reset (active low)	0x1
8	reg_soft_reset_x_tpu	R/W	TPU IP software reset (active low)	0x1
9	reg_soft_reset_x_tpusys	R/W	TPU system software reset (active low)	0x1
10	Reserved			
11	reg_soft_reset_x_usb	R/W	USB IP software reset (active low)	0x1
12	reg_soft_reset_x_eth0	R/W	ETH0 IP software reset (active low)	0x1
13	reg_soft_reset_x_eth1	R/W	ETH1 IP software reset (active low)	0x1
14	reg_soft_reset_x_nand	R/W	NAND IP software reset (active low)	0x1
15	reg_soft_reset_x_emmc	R/W	EMMC IP software reset (active low)	0x1
16	reg_soft_reset_x_sd0	R/W	SD0 IP software reset (active low)	0x1
17	Reserved			
18	reg_soft_reset_x_sdma	R/W	SDMA IP software reset (active low)	0x1
19	reg_soft_reset_x_i2s0	R/W	I2S0 IP software reset (active low)	0x1
20	reg_soft_reset_x_i2s1	R/W	I2S1 IP software reset (active low)	0x1
21	reg_soft_reset_x_i2s2	R/W	I2S2 IP software reset (active low)	0x1
22	reg_soft_reset_x_i2s3	R/W	I2S3 IP software reset (active low)	0x1
23	reg_soft_reset_x_uart0	R/W	UART0 IP software reset (active low)	0x1
24	reg_soft_reset_x_uart1	R/W	UART1 IP software reset (active low)	0x1
25	reg_soft_reset_x_uart2	R/W	UART2 IP software reset (active low)	0x1
26	reg_soft_reset_x_uart3	R/W	UART3 IP software reset (active low)	0x1
27	reg_soft_reset_x_i2c0	R/W	I2C0 IP software reset (active low)	0x1
28	reg_soft_reset_x_i2c1	R/W	I2C1 IP software reset (active low)	0x1
29	reg_soft_reset_x_i2c2	R/W	I2C2 IP software reset (active low)	0x1
30	reg_soft_reset_x_i2c3	R/W	I2C3 IP software reset (active low)	0x1
31	reg_soft_reset_x_i2c4	R/W	I2C4 IP software reset (active low)	0x1

SOFT_RSTN_1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_pwm0	R/W	PWM0 IP software reset (active low)	0x1
1	reg_soft_reset_x_pwm1	R/W	PWM1 IP software reset (active low)	0x1
2	reg_soft_reset_x_pwm2	R/W	PWM2 IP software reset (active low)	0x1
3	reg_soft_reset_x_pwm3	R/W	PWM3 IP software reset (active low)	0x1
7:4	Reserved			
8	reg_soft_reset_x_spi0	R/W	SPI0 IP software reset (active low)	0x1
9	reg_soft_reset_x_spi1	R/W	SPI1 IP software reset (active low)	0x1
10	reg_soft_reset_x_spi2	R/W	SPI2 IP software reset (active low)	0x1
11	reg_soft_reset_x_spi3	R/W	SPI3 IP software reset (active low)	0x1
12	reg_soft_reset_x_gpio0	R/W	GPIO0 IP software reset (active low)	0x1
13	reg_soft_reset_x_gpio1	R/W	GPIO1 IP software reset (active low)	0x1
14	reg_soft_reset_x_gpio2	R/W	GPIO2 IP software reset (active low)	0x1
15	reg_soft_reset_x_efuse	R/W	EFUSE IP software reset (active low)	0x1
16	reg_soft_reset_x_wdt	R/W	WDT0 IP software reset (active low)	0x1
17	reg_soft_reset_x_ahb_rom	R/W	ROM IP software reset (active low)	0x1
18	reg_soft_reset_x_spic	R/W	SPIC IP software reset (active low)	0x1
19	reg_soft_reset_x_tempsen	R/W	TEMPSEN IP software reset (active low)	0x1
20	reg_soft_reset_x_saradc	R/W	SARADC IP software reset (active low)	0x1
25:21	Reserved			

Bits	Name	Access	Description	Reset
26	reg_soft_reset_x_combo_phy0	R/W	USB_PHY IP software reset (active low)	0x1
28:27	Reserved			
29	reg_soft_reset_x_spi_nand	R/W	NAND IP software reset (active low)	0x1
30	reg_soft_reset_x_se	R/W	SE IP software reset (active low)	0x1
31	Reserved			

SOFT_RSTN_2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	Reserved			
10	reg_soft_reset_x_uart4	R/W	UART4 IP software reset (active low)	0x1
11	reg_soft_reset_x_gpio3	R/W	GPIO3 IP software reset (active low)	0x1
12	reg_soft_reset_x_system	R/W	SYSTEM software reset (active low)	0x1
13	reg_soft_reset_x_timer	R/W	TIMER IP software reset (active low)	0x1
14	reg_soft_reset_x_timer0	R/W	TIMER0 IP software reset (active low)	0x1
15	reg_soft_reset_x_timer1	R/W	TIMER1 IP software reset (active low)	0x1
16	reg_soft_reset_x_timer2	R/W	TIMER2 IP software reset (active low)	0x1
17	reg_soft_reset_x_timer3	R/W	TIMER3 IP software reset (active low)	0x1
18	reg_soft_reset_x_timer4	R/W	TIMER4 IP software reset (active low)	0x1
19	reg_soft_reset_x_timer5	R/W	TIMER5 IP software reset (active low)	0x1
20	reg_soft_reset_x_timer6	R/W	TIMER6 IP software reset (active low)	0x1
21	reg_soft_reset_x_timer7	R/W	TIMER7 IP software reset (active low)	0x1
22	reg_soft_reset_x_wgn0	R/W	WGN0 IP software reset (active low)	0x1
23	reg_soft_reset_x_wgn1	R/W	WGN1 IP software reset (active low)	0x1
24	reg_soft_reset_x_wgn2	R/W	WGN2 IP software reset (active low)	0x1
25	reg_soft_reset_x_keyscan	R/W	KEYSCAN IP software reset (active low)	0x1
26	Reserved			
27	reg_soft_reset_x_auddac	R/W	AUDDAC IP software reset (active low)	0x1
28	reg_soft_reset_x_auddac_apb	R/W	AUDDAC APB software reset (active low)	0x1
29	reg_soft_reset_x_audadc	R/W	AUDADC IP software reset (active low)	0x1
30	Reserved			
31	reg_soft_reset_x_vcsys	R/W	VCSYS SYS software reset (active low)	0x1

SOFT_RSTN_3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_ethphy	R/W	ETHPHY IP software reset (active low)	0x1
1	reg_soft_reset_x_ethphy_apb	R/W	ETHPHY APB REG software reset (active low)	0x1
2	reg_soft_reset_x_audsrc	R/W	AUDSRC IP software reset (active low)	0x1
3	reg_soft_reset_x_vip_cam0	R/W	VIP CAM0 IP software reset (active low)	0x1
4	reg_soft_reset_x_wdt1	R/W	WDT1 IP software reset (active low)	0x1
5	reg_soft_reset_x_wdt2	R/W	WDT2 IP software reset (active low)	0x1
31:6	Reserved			

SOFT_CPUAC_RSTN

Offset Address: 0x020

Write Lock: SOFT_CPUAC_RSTN_wr_lock

Bits	Name	Access	Description	Reset
0	reg_auto_clear_reset_x_cpucore0	R/W	CPUCORE0 auto_clear_reset (active low)	0x1
1	reg_auto_clear_reset_x_cpucore1	R/W	CPUCORE1 auto_clear_reset (active low)	0x1

Bits	Name	Access	Description	Reset
2	reg_auto_clear_reset_x_cpucore2	R/W	CPUCORE2 auto_clear_reset (active low)	0x1
3	reg_auto_clear_reset_x_cpucore3	R/W	CPUCORE3 auto_clear_reset (active low)	0x1
4	reg_auto_clear_reset_x_cpusys0	R/W	CPUSYS0 auto_clear_reset (active low)	0x1
5	reg_auto_clear_reset_x_cpusys1	R/W	CPUSYS1 auto_clear_reset (active low)	0x1
6	reg_auto_clear_reset_x_cpusys2	R/W	CPUSYS2 auto_clear_reset (active low)	0x1
31:7	Reserved			

SOFT_CPU_RSTN

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_cpucore0	R/W	CPUCORE0 soft reset (active low)	0x0
1	reg_soft_reset_x_cpucore1	R/W	CPUCORE1 soft reset (active low)	0x0
2	reg_soft_reset_x_cpucore2	R/W	CPUCORE2 soft reset (active low)	0x0
3	reg_soft_reset_x_cpucore3	R/W	CPUCORE3 soft reset (active low)	0x0
4	reg_soft_reset_x_cpusys0	R/W	CPUSYS0 soft reset (active low)	0x0
5	reg_soft_reset_x_cpusys1	R/W	CPUSYS1 soft reset (active low)	0x0
6	reg_soft_reset_x_cpusys2	R/W	CPUSYS2 soft reset (active low)	0x0
31:7	Reserved			

6 时钟

6.1 概述

时钟管理模块对芯片时钟进行管理，包括：

- 时钟输入的管理和控制
- PLL 时钟源及相关倍频和分频控制
- 时钟分频和控制
- 生成各模块的工作时钟
- 时钟输出的管理和控制

6.2 功能框图

图表 6-1 时钟管理框图

XTAL_XIN 为 PLL 输入时钟，固定连接 25MHz 晶体；RTC_XIN 为 RTC 输入时钟，固定连接 32.768KHz 晶体。

6.3 时钟资源与分频结构

系统时钟来源主要来自外部 XTAL、使用 XTAL 倍频之 PLL 及外部协定输入时钟。如图表 3-3 所示，各 IP 一般有 XTAL ,及来自一个 PLL 或以上的时钟资，各自经过分频电路后，

以选择作为 IP 或子系统之时钟。

图表 6-2 时钟源分频示意图.

6.4 PLL 配置

如表格 6-1 所示，芯片内置 9 PLLs (不计入 Analog IP 内建 PLL)，分为整数倍频跟分数倍频两类 PLL。

表格 6-1 PLL 配置参数表

PLL	配置寄存器	下电控制寄存器	预设频率	PLL 型态
FPLL	fpll_csr	fpll_pwd (default On)	1500MHz	整数倍频
MIPIMPLL	mipimpll_csr	mipimpll_pwd (default On)	900MHz	整数倍频
MPLL	mpll_csr mpll_ssc_syn_ctrl mpll_ssc_syn_set mpll_ssc_syn_span mpll_ssc_syn_setp	mpll_pwd (default On)	1000MHz	整/分数倍频
TPLL	tpll_csr tpll_ssc_syn_ctrl	tpll_pwd (default On)	1400MHz	整/分数倍频

PLL	配置寄存器	下电控制寄存器	预设频率	PLL 型态
	tpll_ssc_syn_set tpll_ssc_syn_span tpll_ssc_syn_setp			
APLL	apll_csr apll_ssc_syn_ctrl apll_ssc_syn_set apll_ssc_syn_span apll_ssc_syn_setp	apll_pwd(default On)	1050MHz	整/分数倍频
CAM0PLL	cam0pll_csr cam0pll_ssc_syn_ctrl cam0pll_ssc_syn_set cam0pll_ssc_syn_span cam0pll_ssc_syn_setp	cam0pll_pwd(default On)	1050MHz	整/分数倍频
CAM1PLL	cam1pll_csr cam1pll_ssc_syn_ctrl cam1pll_ssc_syn_set cam1pll_ssc_syn_span cam1pll_ssc_syn_setp	cam1pll_pwd(default On)	1025MHz	整/分数倍频
DISPPLL	disppll_csr disppll_ssc_syn_ctrl disppll_ssc_syn_set disppll_ssc_syn_span disppll_ssc_syn_setp	disppll_pwd(default On)	1200MHz	整/分数倍频

6.5 整数倍频 PLL

调整整数 PLL 流程:

1. 调整时须将相关使用此 PLL 之时钟关闭或是选择时钟源为 XTAL 或其他 PLL
2. 配置 *_pll_csr 寄存器，按整数 PLL 参数表配置
3. 清除 *_pll_pwd

表格 6-2 整数 PLL 配置参数表

PLL 参数	范围	注意事项
PLL_REF	25MHz~2500MHz	
PLL_VCO	800MHz~2500MHz	
Pre_div_sel	1~127	PLL_VCO = PLL_REF*Div_sel/Pre_div_sel PLL_FOUT = PLL_VCO/Post_div_sel
Div_sel	6~127	
Post_div_sel	1~127	
Ictrl	0~7	0.2< 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <=0.35

PLL 参数	范围	注意事项
Mode	0~3	

6.5.1 分数倍频 PLL

调整分数 PLL 流程:

1. 时须将相关使用此 PLL 之 IP clock 关闭或是选择 XTAL 或其他稳定时钟源
2. 配置 *_ssc_syn_src_en 以使能 synthesizer 时钟
3. 依据 PLL 频率需求配置 *_ssc_syn_set ,
4. Toggle *_ssc_syn_up ,使配置生效
5. 配置 *_pll_csr 寄存器，按整数 PLL 参数表配置
6. 清除 *_pll_pwd

表格 6-3 分数 PLL 参数配置表

PLL 参数	范围	注意事项
ssc_freq_in	DSIPLL: 1.2GHz DDRPLL: 1.5GHz Others 600MHz	Default clock gated , need enable
ssc_syn_set	> 4.x * 2^26	ssc_freq_in*div_sel*2^26/PLL_VCO
PLL_REF	100M~2500MHz	
PLL_VCO	800MHz~2500MHz	
Div_sel	6~127	PLL_VCO = PLL_REF*div_sel FOUT = PLL_VCO/Post_div_sel
Post_div_sel	1~127	
Ictrl	0~7	0.1< 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <=0.24
Mode	0~3	

6.6 CLK_DIV 时钟分频配置

主要时钟分频资源表，下表标示各时钟所可以配置之时钟源与预设之时钟频率及分频。软件可以在 Boot 之后将时钟源由 XTAL 切换到 PLL，并进行时钟分频配置调整

表格 6-4 时钟源及预设分频参数表

CLK_NAME	XTAL	DIV	SW	PLL SRC/DIV/FREQ	DIV_IN0 SRC				DIV_IN1 SRC
clk_a53	Y	Y	Y	fpll/(2)/750M	tpll	a0pll	mipimpll	mpll	fpll
clk_cpu_axi0	Y	Y		fpll/(3)/500M	fpll	disppll			

clk_cpu_gic	Y	Y		fpll/(5)/300M	fpll				
clk_tpu	Y	Y		fpll/(3)/500M	tpll	a0pll	mipimpll	fpll	
clk_emmc	Y	Y		fpll/(15)/100M	fpll	disppll			
clk_sd0	Y	Y		fpll/(15)/100M	fpll	disppll			
clk_sd1	Y	Y		fpll/(15)/100M	fpll	disppll			
clk_spi_nand	Y	Y		fpll/(8)/187.5M	fpll	disppll			
clk_sdma_aud0	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud1	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud2	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud3	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_pwm	Y	Y		fpll/(10)/150M	fpll	disppll			
clk_uart	Y	Y		xtal/(1)/25M	xtal	disppll			
clk_axi4	Y	Y		fpll/(5)/300M	fpll	disppll			
clk_axi6	Y	Y		fpll/(15)/100M	fpll				
clk_axi_vip	Y	Y		fpll/(6)/250M	mipimpll	cam0pll	disppll	fpll	
clk_src_vip_sys_0	Y	Y		fpll/(6)/250M	mipimpll	cam0pll	disppll	fpll	
clk_src_vip_sys_1	Y	Y		fpll/(5)/300M	mipimpll	cam0pll	disppll	fpll	
clk_axi_video_codec	Y	Y		mipimpll/(2)/450M	a0pll	mipimpll	cam1pll	fpll	
clk_vc_src0	Y	Y		mipimpll/(2)/450M	disppll	mipimpll	cam1pll	fpll	
clk_spi	Y	Y		fpll/(8)/187.5M	fpll				
clk_i2c	Y	Y		clk_axi6/(1)/100M	clk_axi6				
clk_src_vip_sys_2	Y	Y		disppll/(2)/600M	mipimpll	cam0pll	disppll	fpll	
clk_audsrc	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_ap_debug	Y	Y		fpll/(5)/300M	fpll				
clk_src_rtc_sys_0	Y	Y		fpll/(5)/300M	fpll				
clk_c906_0	Y	Y	Y	fpll/(2)/750M	tpll	a0pll	mipimpll	mpll	fpll
clk_c906_1	Y	Y	Y	fpll/(3)/500M	tpll	a0pll	disppll	mpll	fpll
clk_src_vip_sys_3	Y	Y		mipimpll/(2)/450M	mipimpll	cam0pll	disppll	fpll	
clk_src_vip_sys_4	Y	Y		disppll/(3)/400M	mipimpll	cam0pll	disppll	fpll	

6.6.1 IP/SYS 时钟源、时钟分频配置

1. 关闭 IP 时钟；若不能关闭的时钟，应先配置到稳定分频时钟
   甲、CPU 变频: 配置 clk_sel_0 选择切换至 SRC1 , 避免频率太低
   乙、IP 变频:配置 clk_byp_0/1 切换时钟至 XTAL
2. 配置要调整之时钟源及分频器配置

3. 需要配置分频器寄存器[3]，该时钟分频器配置方能生效
4. 将时钟源选择到配置完成之时钟分频器

6.6.2 MCLK0/MCLK1

1. MCLK0/MCLK1 是提供外部 sensor 参考时钟。
2. 透过配置 CAM0PLL、clk_cam0_src_div 及 clk_cam0_src_div 以提供适合的 MCLK0/MCLK1 输出频率。

6.6.3 Clk_A24M

1. 在性能可接受状况可以使用 clk_a24m 作为音频时钟。
2. 透过配置 apll_frac_div_ctrl、apll_frac_div_m 及 apll_frac_div_n 输出需要的音频时钟源。
3. clk_a24m 之频率为 900MHz * N/M/2 。

6.7 PLL CRG 寄存器概览

PLL_G2 base address : 0x03002800

Name	Address Offset	Description
pll_g2_ctrl	0x000	Group2 PLL Ctrl register
pll_g2_status	0x004	Group2 PLL Status register
mipimpll_csr	0x008	MIPIMPLL Ctrl register
apll0_csr	0x00c	APLL0 Ctrl register
disppll_csr	0x010	DISPPLL Ctrl register
cam0pll_csr	0x014	CAM0PLL Ctrl register
cam1pll_csr	0x018	CAM1PLL Ctrl register
pll_g2_ssc_syn_ctrl	0x040	Group2 PLL Synthesizer ctrl register
apll_ssc_syn_ctrl	0x050	APLL synthesizer ctrl register
apll_ssc_syn_set	0x054	APLL synthesizer set register
disppll_ssc_syn_ctrl	0x060	DISPPLL synthesizer ctrl register
disppll_ssc_syn_set	0x064	DISPPLL synthesizer set register
cam0pll_ssc_syn_ctrl	0x070	CAM0PLL synthesizer ctrl register
cam0pll_ssc_syn_set	0x074	CAM0PLL synthesizer set register
cam1pll_ssc_syn_ctrl	0x080	CAM1PLL synthesizer ctrl register
cam1pll_ssc_syn_set	0x084	CAM1PLL synthesizer set register
apll_frac_div_ctrl	0x090	APLL frac divider ctrl register
apll_frac_div_m	0x094	APLL frac divider M parameter
apll_frac_div_n	0x098	APLL frac divider N parameter
mipimpll_clk_csr	0x0a0	MIPIMPLL clock Ctrl register
a0pll_clk_csr	0x0a4	a0pll clock Ctrl register
disppll_clk_csr	0x0a8	disppll clock Ctrl register
cam0pll_clk_csr	0x0ac	cam0pll clock Ctrl register
cam1pll_clk_csr	0x0b0	cam1pll clock Ctrl register
clk_cam0_src_div	0x0c0	clk_cam0_src_div
clk_cam1_src_div	0x0c4	clk_cam1_src_div

PLL_G6 base address : 0x03002900

Name	Address Offset	Description
pll_g6_ctrl	0x000	Group6 PLL Ctrl register
pll_g6_status	0x004	Group6 PLL Status register
mpll_csr	0x008	MPLL Ctrl register
tpll_csr	0x00c	TPLL Ctrl register
fpll_csr	0x010	FPLL Ctrl register
pll_g6_ssc_syn_ctrl	0x040	Group6 PLL Synthesizer ctrl register
dpll_ssc_syn_ctrl	0x050	dpll synthesizer ctrl register
dpll_ssc_syn_set	0x054	dpll synthesizer set register
dpll_ssc_syn_span	0x058	dpll synthesizer span register
dpll_ssc_syn_step	0x05c	dpll synthesizer step register
mpll_ssc_syn_ctrl	0x060	mpll synthesizer ctrl register
mpll_ssc_syn_set	0x064	mpll synthesizer set register
mpll_ssc_syn_span	0x068	mpll synthesizer span register
mpll_ssc_syn_step	0x06c	mpll synthesizer step register
tpll_ssc_syn_ctrl	0x070	tpll synthesizer ctrl register
tpll_ssc_syn_set	0x074	tpll synthesizer set register
tpll_ssc_syn_span	0x078	tpll synthesizer span register
tpll_ssc_syn_step	0x07c	tpll synthesizer step register

6.8 PLL CRG 寄存器描述

6.8.1 PLL_G2 寄存器描述

pll_g2_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mipimpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	apll0_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	disppll_pwd	R/W	pll power down	0x0
11:9	Reserved			
12	cam0pll_pwd	R/W	pll power down	0x0
15:13	Reserved			
16	cam1pll_pwd	R/W	pll power down	0x0
31:17	Reserved			

pll_g2_status

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mipimpll_val	RO	pll setting update status	

Bits	Name	Access	Description	Reset
1	updating_apll0_val	RO	pll setting update status	
2	updating_disppll_val	RO	pll setting update status	
3	updating_cam0pll_val	RO	pll setting update status	
4	updating_cam1pll_val	RO	pll setting update status	
15:5	Reserved			
16	mipimpll_lock	RO	pll lock status	
17	apll0_lock	RO	pll lock status	
18	disppll_lock	RO	pll lock status	
19	cam0pll_lock	RO	pll lock status	
20	cam1pll_lock	RO	pll lock status	
31:21	Reserved			

mipimpll_csr

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mipimpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mipimpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	mipimpll_sel_mode	R/W	pll mode setting	0x0
23:17	mipimpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mipimpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

apll0_csr

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	apll0_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	apll0_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	apll0_sel_mode	R/W	pll mode setting	0x0
23:17	apll0_div_sel	R/W	pll div_sel setting	0x0
26:24	apll0_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

disppll_csr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	disppll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	disppll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	disppll_sel_mode	R/W	pll mode setting	0x0
23:17	disppll_div_sel	R/W	pll div_sel setting	0x0
26:24	disppll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

cam0pll_csr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
6:0	cam0pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0

Bits	Name	Access	Description	Reset
7	Reserved			
14:8	cam0pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam0pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam0pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam0pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

cam1pll_csr

Offset Address: 0x018

Bits	Name	Access	Description	Reset
6:0	cam1pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	cam1pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam1pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam1pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam1pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

pll_g2_ssc_syn_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_mipimpll_sel_syn_clk	R/W	mipimpll gen synthesizer clock source 0:450M 1:900M	0x1
1	reg_dsi_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
2	reg_apll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
3	reg_disppll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
4	reg_cam0pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
5	reg_cam1pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
31:6	Reserved			

apll_ssc_syn_ctrl

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_apll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_apll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

apll_ssc_syn_set

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_apll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

disppll_ssc_syn_ctrl

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_disppll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			

Bits	Name	Access	Description	Reset
6	reg_disppll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

disppll_ssc_syn_set

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_disppll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

cam0pll_ssc_syn_ctrl

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_cam0pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam0pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

cam0pll_ssc_syn_set

Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_cam0pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

cam1pll_ssc_syn_ctrl

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_cam1pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam1pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

cam1pll_ssc_syn_set

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_cam1pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

apll_frac_div_ctrl

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	reg_apll_frac_div_clk_en	R/W	a24m clock src enable	0x0
1	reg_apll_frac_div_en	R/W	a24m clock div enable	0x0
2	reg_apll_frac_div_up	W1T		
3	reg_apll_frac_reg_out_en	R/W	a24m clock output enable	0x0
31:4	Reserved			

apll_frac_div_m

Offset Address: 0x094

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_m	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

apll_frac_div_n

Offset Address: 0x098

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_n	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

mipimpll_clk_csr

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_mipimpll_pdiv_pd	R/W	pd post div	0x0
1	reg_mipimpll_d2_pd	R/W	pd div2 div	0x1
2	reg_mipimpll_d3_pd	R/W	pd div3 div	0x1
3	reg_mipimpll_d5_pd	R/W	pd div5 div	0x1
4	reg_mipimpll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_mipimpll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_mipimpll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_mipimpll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_mipimpll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_mipimpll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

a0pll_clk_csr

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_a0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_a0pll_d2_pd	R/W	pd div2 div	0x1
2	reg_a0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_a0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_a0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_a0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_a0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_a0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_a0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_a0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

disppll_clk_csr

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	reg_disppll_pdiv_pd	R/W	pd post div	0x0
1	reg_disppll_d2_pd	R/W	pd div2 div	0x1
2	reg_disppll_d3_pd	R/W	pd div3 div	0x1
3	reg_disppll_d5_pd	R/W	pd div5 div	0x1
4	reg_disppll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_disppll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_disppll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_disppll_d3_auto_pd	R/W	auto pd div3 clk	0x1

Bits	Name	Access	Description	Reset
11	reg_disppll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_disppll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

cam0pll_clk_csr

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam0pll_d2_pd	R/W	pd div2 div	0x1
2	reg_cam0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

cam1pll_clk_csr

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	reg_cam1pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam1pll_d2_pd	R/W	pd div2 div	0x1
2	reg_cam1pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam1pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam1pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam1pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam1pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam1pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam1pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam1pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

clk_cam0_src_div

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	reg_cam0_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1
3:1	Reserved			
4	reg_cam0_div_dis	R/W	[4] Divider Reset Control 0: Assert Reset	0x0
7:5	Reserved			
9:8	reg_cam0_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam0_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

clk_cam1_src_div

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	reg_cam1_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1
3:1	Reserved			
4	reg_cam1_div_dis	R/W	[4] divider disable	0x0
7:5	Reserved			
9:8	reg_cam1_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam1_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

6.8.2 PLL_G6 寄存器描述

pll_g6_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	tpll_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	fpll_pwd	R/W	pll power down	0x0
31:9	Reserved			

pll_g6_status

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mpll_val	RO	pll setting update status	
1	updating_tpll_val	RO	pll setting update status	
2	updating_fpll_val	RO	pll setting update status	
15:3	Reserved			
16	mpll_lock	RO	pll lock status	
17	tpll_lock	RO	pll lock status	
18	fpll_lock	RO	pll lock status	
31:19	Reserved			

mpll_csr

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	mpll_sel_mode	R/W	pll mode setting	0x0

Bits	Name	Access	Description	Reset
23:17	mpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

tpll_csr

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	tpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	tpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	tpll_sel_mode	R/W	pll mode setting	0x0
23:17	tpll_div_sel	R/W	pll div_sel setting	0x0
26:24	tpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

fpll_csr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	fpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	fpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	fpll_sel_mode	R/W	pll mode setting	0x0
23:17	fpll_div_sel	R/W	pll div_sel setting	0x0
26:24	fpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

pll_g6_ssc_syn_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_fpll_sel_syn_clk	R/W	fpll gen synthesizer clock source 0:750M 1:1.5G	0x1
1	reg_ddr_ssc_syn_src_en	R/W	ddr pll synthesizer clock enable	0x1
2	reg_mpll_ssc_syn_src_en	R/W	mpll synthesizer clock enable	0x0
3	reg_tpll_ssc_syn_src_en	R/W	tpll synthesizer clock enable	0x0
31:4	Reserved			

dpll_ssc_syn_ctrl

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_dpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_dpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_dpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_dpll_ssc_syn_bypass	R/W		0x0
5	reg_dpll_ssc_syn_extpulse	R/W		0x0
6	reg_dpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

dpll_ssc_syn_set

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_dpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

dpll_ssc_syn_span

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	reg_dpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

dpll_ssc_syn_step

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	reg_dpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

mpll_ssc_syn_ctrl

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_mpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_mpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_mpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_mpll_ssc_syn_bypass	R/W		0x1
5	reg_mpll_ssc_syn_extpulse	R/W		0x0
6	reg_mpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

mpll_ssc_syn_set

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_mpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

mpll_ssc_syn_span

Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	reg_mpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

mpll_ssc_syn_step

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	reg_mpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

tpll_ssc_syn_ctrl

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_tpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_tpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_tpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_tpll_ssc_syn_bypass	R/W		0x1
5	reg_tpll_ssc_syn_extpulse	R/W		0x0
6	reg_tpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

tpll_ssc_syn_set

Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_tpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

tpll_ssc_syn_span

Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	reg_tpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

tpll_ssc_syn_step

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
23:0	reg_tpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

6.9 CLK_DIV CRG 寄存器概览

Clock Gen base address : 0x03002000

Name	Address Offset	Description
clk_en_0	0x000	clock enable register 0
clk_en_1	0x004	clock enable register 1
clk_en_2	0x008	clock enable register 2
clk_en_3	0x00c	clock enable register 3
clk_en_4	0x010	clock enable register 4
clk_sel_0	0x020	clock source selection register 0
clk_byp_0	0x030	clock bypass to xtal register 0
clk_byp_1	0x034	clock bypass to xtal register 1
div_clk_a53_0	0x040	divider register of clk_a53_0
div_clk_a53_1	0x044	divider register of clk_a53_1
div_clk_cpu_axi0	0x048	divider register of clk_cpu_axi0
div_clk_cpu_gic	0x050	divider register of clk_cpu_gic
div_clk_tpu	0x054	divider register of clk_tpu
div_clk_emmc	0x064	divider register of clk_emmc

Name	Address Offset	Description
div_clk_100k_emmc	0x06c	divider register of clk_100k_emmc
div_clk_sd0	0x070	divider register of clk_sd0
div_clk_100k_sd0	0x078	divider register of clk_100k_sd0
div_clk_sd1	0x07c	divider register of clk_sd1
div_clk_100k_sd1	0x084	divider register of clk_100k_sd1
div_clk_spi_nand	0x088	divider register of clk_spi_nand
div_clk_500m_eth0	0x08c	divider register of clk_500m_eth0
div_clk_gpio_db	0x094	divider register of clk_gpio_db
div_clk_sdma_aud0	0x098	divider register of clk_sdma_aud0
div_clk_sdma_aud1	0x09c	divider register of clk_sdma_aud1
div_clk_sdma_aud2	0x0a0	divider register of clk_sdma_aud2
div_clk_sdma_aud3	0x0a4	divider register of clk_sdma_aud3
div_clk_cam0_200	0x0a8	divider register of clk_cam0_200
div_clk_axi4	0x0b8	divider register of clk_axi4
div_clk_axi6	0x0bc	divider register of clk_axi6
div_clk_dsi_esc	0x0c4	divider register of clk_dsi_esc
div_clk_axi_vip	0x0c8	divider register of clk_axi_vip
div_clk_src_vip_sys_0	0x0d0	divider register of clk_src_vip_sys_0
div_clk_src_vip_sys_1	0x0d8	divider register of clk_src_vip_sys_1
div_clk_disp_src_vip	0x0e0	divider register of clk_disp_src_vip
div_clk_axi_video_codec	0x0e4	divider register of clk_axi_video_codec
div_clk_vc_src0	0x0ec	divider register of clk_vc_src0
div_clk_1m	0x0fc	divider register of clk_1m
div_clk_spi	0x100	divider register of clk_spi
div_clk_i2c	0x104	divider register of clk_i2c
div_clk_src_vip_sys_2	0x110	divider register of clk_src_vip_sys_2
div_clk_audsrc	0x118	divider register of clk_audsrc
div_clk_pwm_src_0	0x120	divider register of clk_pwm_src_0
div_clk_ap_debug	0x128	divider register of clk_ap_debug
div_clk_rtcsys_src_0	0x12c	divider register of clk_rtcsys_src_0
div_clk_c906_0_0	0x130	divider register of clk_c906_0_0
div_clk_c906_0_1	0x134	divider register of clk_c906_0_1
div_clk_c906_1_0	0x138	divider register of clk_c906_1_0
div_clk_c906_1_1	0x13c	divider register of clk_c906_1_1
div_clk_src_vip_sys_3	0x140	divider register of clk_src_vip_sys_3
div_clk_src_vip_sys_4	0x144	divider register of clk_src_vip_sys_4

6.10 CLK_DIV CRG 寄存器描述

clk_en_0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	clk_en_0_0	R/W	Clock Enable for clk_a53 (1: Enable; 0: Gate)	0x1
1	clk_en_0_1	R/W	Clock Enable for clk_cpu_axi0 (1: Enable; 0: Gate)	0x1
2	clk_en_0_2	R/W	Clock Enable for clk_cpu_gic (1: Enable; 0: Gate)	0x1
3	clk_en_0_3	R/W	Clock Enable for clk_xtal_a53 (1: Enable; 0: Gate)	0x1
4	clk_en_0_4	R/W	Clock Enable for clk_tpu (1: Enable; 0:	0x1

Bits	Name	Access	Description	Reset
			Gate)	
5	Reserved			
6	clk_en_0_6	R/W	Clock Enable for clk_ahb_rom (1: Enable; 0: Gate)	0x1
7	clk_en_0_7	R/W	Clock Enable for clk_ddr_axi_reg (1: Enable; 0: Gate)	0x1
8	clk_en_0_8	R/W	Clock Enable for clk_rtc_25m (1: Enable; 0: Gate)	0x1
9	clk_en_0_9	R/W	Clock Enable for clk_tempsen (1: Enable; 0: Gate)	0x1
10	clk_en_0_10	R/W	Clock Enable for clk_saradc (1: Enable; 0: Gate)	0x1
11	clk_en_0_11	R/W	Clock Enable for clk_efuse (1: Enable; 0: Gate)	0x1
12	clk_en_0_12	R/W	Clock Enable for clk_apb_efuse (1: Enable; 0: Gate)	0x1
13	Reserved			
14	clk_en_0_14	R/W	Clock Enable for clk_xtal_misc (1: Enable; 0: Gate)	0x1
15	clk_en_0_15	R/W	Clock Enable for clk_axi4_emmc (1: Enable; 0: Gate)	0x1
16	clk_en_0_16	R/W	Clock Enable for clk_emmc (1: Enable; 0: Gate)	0x1
17	clk_en_0_17	R/W	Clock Enable for clk_100k_emmc (1: Enable; 0: Gate)	0x1
18	clk_en_0_18	R/W	Clock Enable for clk_axi4_sd0 (1: Enable; 0: Gate)	0x1
19	clk_en_0_19	R/W	Clock Enable for clk_sd0 (1: Enable; 0: Gate)	0x1
20	clk_en_0_20	R/W	Clock Enable for clk_100k_sd0 (1: Enable; 0: Gate)	0x1
21	clk_en_0_21	R/W	Clock Enable for clk_axi4_sd1 (1: Enable; 0: Gate)	0x1
22	clk_en_0_22	R/W	Clock Enable for clk_sd1 (1: Enable; 0: Gate)	0x1
23	clk_en_0_23	R/W	Clock Enable for clk_100k_sd1 (1: Enable; 0: Gate)	0x1
24	clk_en_0_24	R/W	Clock Enable for clk_spi_nand (1: Enable; 0: Gate)	0x1
25	clk_en_0_25	R/W	Clock Enable for clk_500m_eth0 (1: Enable; 0: Gate)	0x1
26	clk_en_0_26	R/W	Clock Enable for clk_axi4_eth0 (1: Enable; 0: Gate)	0x1
28:27	Reserved			
29	clk_en_0_29	R/W	Clock Enable for clk_apb_gpio (1: Enable; 0: Gate)	0x1
30	clk_en_0_30	R/W	Clock Enable for clk_apb_gpio_intr (1: Enable; 0: Gate)	0x1
31	clk_en_0_31	R/W	Clock Enable for clk_gpio_db (1: Enable; 0: Gate)	0x1

clk_en_1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	clk_en_1_0	R/W	Clock Enable for clk_ahb_sf (1: Enable; 0: Gate)	0x1

Bits	Name	Access	Description	Reset
1	clk_en_1_1	R/W	Clock Enable for clk_sdma_axi (1: Enable; 0: Gate)	0x1
2	clk_en_1_2	R/W	Clock Enable for clk_sdma_aud0 (1: Enable; 0: Gate)	0x1
3	clk_en_1_3	R/W	Clock Enable for clk_sdma_aud1 (1: Enable; 0: Gate)	0x1
4	clk_en_1_4	R/W	Clock Enable for clk_sdma_aud2 (1: Enable; 0: Gate)	0x1
5	clk_en_1_5	R/W	Clock Enable for clk_sdma_aud3 (1: Enable; 0: Gate)	0x1
6	clk_en_1_6	R/W	Clock Enable for clk_apb_i2c (1: Enable; 0: Gate)	0x1
7	clk_en_1_7	R/W	Clock Enable for clk_apb_wdt (1: Enable; 0: Gate)	0x1
8	clk_en_1_8	R/W	Clock Enable for clk_apb_pwm (1: Enable; 0: Gate)	0x1
9	clk_en_1_9	R/W	Clock Enable for clk_apb_spi0 (1: Enable; 0: Gate)	0x1
10	clk_en_1_10	R/W	Clock Enable for clk_apb_spi1 (1: Enable; 0: Gate)	0x1
11	clk_en_1_11	R/W	Clock Enable for clk_apb_spi2 (1: Enable; 0: Gate)	0x1
12	clk_en_1_12	R/W	Clock Enable for clk_apb_spi3 (1: Enable; 0: Gate)	0x1
13	clk_en_1_13	R/W	Clock Enable for clk_187p5m (1: Enable; 0: Gate)	0x1
14	clk_en_1_14	R/W	Clock Enable for clk_uart0 (1: Enable; 0: Gate)	0x1
15	clk_en_1_15	R/W	Clock Enable for clk_apb_uart0 (1: Enable; 0: Gate)	0x1
16	clk_en_1_16	R/W	Clock Enable for clk_uart1 (1: Enable; 0: Gate)	0x1
17	clk_en_1_17	R/W	Clock Enable for clk_apb_uart1 (1: Enable; 0: Gate)	0x1
18	clk_en_1_18	R/W	Clock Enable for clk_uart2 (1: Enable; 0: Gate)	0x1
19	clk_en_1_19	R/W	Clock Enable for clk_apb_uart2 (1: Enable; 0: Gate)	0x1
20	clk_en_1_20	R/W	Clock Enable for clk_uart3 (1: Enable; 0: Gate)	0x1
21	clk_en_1_21	R/W	Clock Enable for clk_apb_uart3 (1: Enable; 0: Gate)	0x1
22	clk_en_1_22	R/W	Clock Enable for clk_uart4 (1: Enable; 0: Gate)	0x1
23	clk_en_1_23	R/W	Clock Enable for clk_apb_uart4 (1: Enable; 0: Gate)	0x1
24	clk_en_1_24	R/W	Clock Enable for clk_apb_i2s0 (1: Enable; 0: Gate)	0x1
25	clk_en_1_25	R/W	Clock Enable for clk_apb_i2s1 (1: Enable; 0: Gate)	0x1
26	clk_en_1_26	R/W	Clock Enable for clk_apb_i2s2 (1: Enable; 0: Gate)	0x1
27	clk_en_1_27	R/W	Clock Enable for clk_apb_i2s3 (1: Enable; 0: Gate)	0x1
28	clk_en_1_28	R/W	Clock Enable for clk_axi4_usb (1: Enable; 0: Gate)	0x1
29	clk_en_1_29	R/W	Clock Enable for clk_apb_usb (1: Enable;	0x1

Bits	Name	Access	Description	Reset
			0: Gate)	
31:30	Reserved			

clk_en_2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_en_2_1	R/W	Clock Enable for clk_axi4 (1: Enable; 0: Gate)	0x1
2	clk_en_2_2	R/W	Clock Enable for clk_axi6 (1: Enable; 0: Gate)	0x1
3	clk_en_2_3	R/W	Clock Enable for clk_dsi_esc (1: Enable; 0: Gate)	0x1
4	clk_en_2_4	R/W	Clock Enable for clk_axi_vip (1: Enable; 0: Gate)	0x1
5	clk_en_2_5	R/W	Clock Enable for clk_src_vip_sys_0 (1: Enable; 0: Gate)	0x1
6	clk_en_2_6	R/W	Clock Enable for clk_src_vip_sys_1 (1: Enable; 0: Gate)	0x1
7	clk_en_2_7	R/W	Clock Enable for clk_disp_src_vip (1: Enable; 0: Gate)	0x1
8	clk_en_2_8	R/W	Clock Enable for clk_axi_video_codec (1: Enable; 0: Gate)	0x1
9	clk_en_2_9	R/W	Clock Enable for clk_vc_src0 (1: Enable; 0: Gate)	0x1
10	clk_en_2_10	R/W	Clock Enable for clk_h264c (1: Enable; 0: Gate)	0x1
11	clk_en_2_11	R/W	Clock Enable for clk_h265c (1: Enable; 0: Gate)	0x1
12	clk_en_2_12	R/W	Clock Enable for clk_jpeg (1: Enable; 0: Gate)	0x1
13	clk_en_2_13	R/W	Clock Enable for clk_apb_jpeg (1: Enable; 0: Gate)	0x1
14	clk_en_2_14	R/W	Clock Enable for clk_apb_h264c (1: Enable; 0: Gate)	0x1
15	clk_en_2_15	R/W	Clock Enable for clk_apb_h265c (1: Enable; 0: Gate)	0x1
16	clk_en_2_16	R/W	Clock Enable for clk_cam0 (1: Enable; 0: Gate)	0x1
17	clk_en_2_17	R/W	Clock Enable for clk_cam1 (1: Enable; 0: Gate)	0x1
18	clk_en_2_18	R/W	Clock Enable for clk_csi_mac0_vip (1: Enable; 0: Gate)	0x1
19	clk_en_2_19	R/W	Clock Enable for clk_csi_mac1_vip (1: Enable; 0: Gate)	0x1
20	clk_en_2_20	R/W	Clock Enable for clk_isp_top_vip (1: Enable; 0: Gate)	0x1
21	clk_en_2_21	R/W	Clock Enable for clk_img_d_vip (1: Enable; 0: Gate)	0x1
22	clk_en_2_22	R/W	Clock Enable for clk_img_v_vip (1: Enable; 0: Gate)	0x1
23	clk_en_2_23	R/W	Clock Enable for clk_sc_top_vip (1: Enable; 0: Gate)	0x1
24	clk_en_2_24	R/W	Clock Enable for clk_sc_d_vip (1: Enable; 0: Gate)	0x1
25	clk_en_2_25	R/W	Clock Enable for clk_sc_v1_vip (1:	0x1

Bits	Name	Access	Description	Reset
			Enable; 0: Gate)	
26	clk_en_2_26	R/W	Clock Enable for clk_sc_v2_vip (1: Enable; 0: Gate)	0x1
27	clk_en_2_27	R/W	Clock Enable for clk_sc_v3_vip (1: Enable; 0: Gate)	0x1
28	clk_en_2_28	R/W	Clock Enable for clk_ldc_vip (1: Enable; 0: Gate)	0x1
29	clk_en_2_29	R/W	Clock Enable for clk_bt_vip (1: Enable; 0: Gate)	0x1
30	clk_en_2_30	R/W	Clock Enable for clk_disp_vip (1: Enable; 0: Gate)	0x1
31	clk_en_2_31	R/W	Clock Enable for clk_dsi_mac_vip (1: Enable; 0: Gate)	0x1

clk_en_3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	clk_en_3_0	R/W	Clock Enable for clk_lvds0_vip (1: Enable; 0: Gate)	0x1
1	clk_en_3_1	R/W	Clock Enable for clk_lvds1_vip (1: Enable; 0: Gate)	0x1
2	clk_en_3_2	R/W	Clock Enable for clk_csi0_rx_vip (1: Enable; 0: Gate)	0x1
3	clk_en_3_3	R/W	Clock Enable for clk_csi1_rx_vip (1: Enable; 0: Gate)	0x1
4	clk_en_3_4	R/W	Clock Enable for clk_pad_vi_vip (1: Enable; 0: Gate)	0x1
5	clk_en_3_5	R/W	Clock Enable for clk_1m (1: Enable; 0: Gate)	0x1
6	clk_en_3_6	R/W	Clock Enable for clk_spi (1: Enable; 0: Gate)	0x1
7	clk_en_3_7	R/W	Clock Enable for clk_i2c (1: Enable; 0: Gate)	0x1
8	clk_en_3_8	R/W	Clock Enable for clk_pm (1: Enable; 0: Gate)	0x1
9	clk_en_3_9	R/W	Clock Enable for clk_timer0 (1: Enable; 0: Gate)	0x1
10	clk_en_3_10	R/W	Clock Enable for clk_timer1 (1: Enable; 0: Gate)	0x1
11	clk_en_3_11	R/W	Clock Enable for clk_timer2 (1: Enable; 0: Gate)	0x1
12	clk_en_3_12	R/W	Clock Enable for clk_timer3 (1: Enable; 0: Gate)	0x1
13	clk_en_3_13	R/W	Clock Enable for clk_timer4 (1: Enable; 0: Gate)	0x1
14	clk_en_3_14	R/W	Clock Enable for clk_timer5 (1: Enable; 0: Gate)	0x1
15	clk_en_3_15	R/W	Clock Enable for clk_timer6 (1: Enable; 0: Gate)	0x1
16	clk_en_3_16	R/W	Clock Enable for clk_timer7 (1: Enable; 0: Gate)	0x1
17	clk_en_3_17	R/W	Clock Enable for clk_apb_i2c0 (1: Enable; 0: Gate)	0x1
18	clk_en_3_18	R/W	Clock Enable for clk_apb_i2c1 (1: Enable; 0: Gate)	0x1
19	clk_en_3_19	R/W	Clock Enable for clk_apb_i2c2 (1: Enable; 0: Gate)	0x1

Bits	Name	Access	Description	Reset
20	clk_en_3_20	R/W	Clock Enable for clk_apb_i2c3 (1: Enable; 0: Gate)	0x1
21	clk_en_3_21	R/W	Clock Enable for clk_apb_i2c4 (1: Enable; 0: Gate)	0x1
22	clk_en_3_22	R/W	Clock Enable for clk_wgn (1: Enable; 0: Gate)	0x1
23	clk_en_3_23	R/W	Clock Enable for clk_wgn0 (1: Enable; 0: Gate)	0x1
24	clk_en_3_24	R/W	Clock Enable for clk_wgn1 (1: Enable; 0: Gate)	0x1
25	clk_en_3_25	R/W	Clock Enable for clk_wgn2 (1: Enable; 0: Gate)	0x1
26	clk_en_3_26	R/W	Clock Enable for clk_keyscan (1: Enable; 0: Gate)	0x1
27	clk_en_3_27	R/W	Clock Enable for clk_ahb_sf1 (1: Enable; 0: Gate)	0x1
28	Reserved			
29	clk_en_3_29	R/W	Clock Enable for clk_src_vip_sys_2 (1: Enable; 0: Gate	0x1
30	clk_en_3_30	R/W	Clock Enable for clk_pad_vi1_vip (1: Enable; 0: Gate)	0x1
31	clk_en_3_31	R/W	Clock Enable for clk_cfg_reg_vip (1: Enable; 0: Gate)	0x1

clk_en_4

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	clk_en_4_0	R/W	Clock Enable for clk_cfg_reg_vc (1: Enable; 0: Gate)	0x1
1	clk_en_4_1	R/W	Clock Enable for clk_audsrc (1: Enable; 0: Gate)	0x1
2	clk_en_4_2	R/W	Clock Enable for clk_apb_audsrc (1: Enable; 0: Gate)	0x1
3	Reserved			
4	clk_en_4_4	R/W	Clock Enable for clk_pwm_src (1: Enable; 0: Gate)	0x1
5	clk_en_4_5	R/W	Clock Enable for clk_ap_debug(1: Enable; 0: Gate)	0x1
6	clk_en_4_6	R/W	Clock Enable for clk_rtcsys_src_0 (1: Enable; 0: Gate)	0x1
7	clk_en_4_7	R/W	Clock Enable for clk_pad_vi2_vip (1: Enable; 0: Gate)	0x1
8	clk_en_4_8	R/W	Clock Enable for clk_csi_be_vip (1: Enable; 0: Gate)	0x1
9	clk_en_4_9	R/W	Clock Enable for clk_vip_ip0_en	0x1
10	clk_en_4_10	R/W	Clock Enable for clk_vip_ip1_en	0x1
11	clk_en_4_11	R/W	Clock Enable for clk_vip_ip2_en	0x1
12	clk_en_4_12	R/W	Clock Enable for clk_vip_ip3_en	0x1
13	clk_en_4_13	R/W	Clock Enable for clk_c906_0_en	0x1
14	clk_en_4_14	R/W	Clock Enable for clk_c906_1_en	0x1
15	clk_en_4_15	R/W	Clock Enable for clk_src_vip_sys_3_en	0x1
16	clk_en_4_16	R/W	Clock Enable for clk_src_vip_sys_4_en	0x1
17	clk_en_4_17	R/W	Clock Enable for clk_ive_vip_en	0x1
18	clk_en_4_18	R/W	Clock Enable for clk_raw_vip_en	0x1
19	clk_en_4_19	R/W	Clock Enable for clk_osdc_vip_en	0x1
20	clk_en_4_20	R/W	Clock Enable for clk_fbc_vip_en	0x1

Bits	Name	Access	Description	Reset
21	clk_en_4_21	R/W	Clock Enable for clk_cam0_vip_en	0x1
31:22	Reserved			

clk_sel_0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	clk_sel_0_0	R/W	Clock Select for A53's clock clk_a53 1: Select div_clk_a53_0 as clock source 0: Select div_clk_a53_1 as clock source	0x0
22:1	Reserved			
23	clk_sel_0_23	R/W	Clock Select for C906's clock clk_c906_0 1: Select div_clk_c906_0_0 as clock source 0: Select div_clk_c906_0_1 as clock source	0x0
24	clk_sel_0_24	R/W	Clock Select for C906's clock clk_c906_1 1: Select div_clk_c906_1_0 as clock source 0: Select div_clk_c906_1_1 as clock source	0x0
31:25	Reserved			

clk_byp_0

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	clk_byp_0_0	R/W	Clock Bypass to xtal for A53's clock clk_a53	0x1
1	clk_byp_0_1	R/W	Clock Bypass to xtal for A53's clock clk_cpu_axi0	0x1
2	clk_byp_0_2	R/W	Clock Bypass to xtal for A53's clock clk_cpu_gic	0x1
3	clk_byp_0_3	R/W	Clock Bypass to xtal for TPU's clock clk_tpu	0x1
4	Reserved			
5	clk_byp_0_5	R/W	Clock Bypass to xtal for EMMC's clock clk_emmc	0x1
6	clk_byp_0_6	R/W	Clock Bypass to xtal for SD's clock clk_sd0	0x1
7	clk_byp_0_7	R/W	Clock Bypass to xtal for SD's clock clk_sd1	0x1
8	clk_byp_0_8	R/W	Clock Bypass to xtal for SPI_NAND's clock clk_spi_nand	0x1
9	clk_byp_0_9	R/W	Clock Bypass to xtal for ETH0's clock clk_500m_eth0	0x1
10	Reserved			
11	clk_byp_0_11	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud0	0x1
12	clk_byp_0_12	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud1	0x1
13	clk_byp_0_13	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud2	0x1
14	clk_byp_0_14	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud3	0x1
15	clk_byp_0_15	R/W	Clock Bypass to xtal for PWM's clock clk_pwm_src	0x1

Bits	Name	Access	Description	Reset
16	clk_byp_0_16	R/W	Clock Bypass to xtal for TOP's clock clk_cam0_200	0x1
18:17	Reserved			
19	clk_byp_0_19	R/W	Clock Bypass to xtal for FABRIC_AXI4's clock clk_axi4	0x1
20	clk_byp_0_20	R/W	Clock Bypass to xtal for FABRIC_AXI6's clock clk_axi6	0x1
21	clk_byp_0_21	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_dsi_esc	0x1
22	clk_byp_0_22	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_axi_vip	0x1
23	clk_byp_0_23	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_0	0x1
24	clk_byp_0_24	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_1	0x1
25	clk_byp_0_25	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_disp_src_vip	0x1
26	clk_byp_0_26	R/W	Clock Bypass to xtal for Video_subsys's clock clk_axi_video_codec	0x1
27	clk_byp_0_27	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src0	0x1
29:28	Reserved			
30	clk_byp_0_30	R/W	Clock Bypass to xtal for SPI's clock clk_spi	0x1
31	clk_byp_0_31	R/W	Clock Bypass to xtal for IIC's clock clk_i2c	0x1

clk_byp_1

Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_byp_1_1	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_2	0x1
2	clk_byp_1_2	R/W	Clock Bypass to xtal for AUDSRC's clock clk_audsrc	0x1
3	clk_byp_1_3	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src2	0x1
4	clk_byp_1_4	R/W	Clock Bypass to xtal for clk_ap_debug	0x1
5	clk_byp_1_5	R/W	Clock Bypass to xtal for clk_src_rtc_sys_0	0x1
6	clk_byp_1_6	R/W	Clock Bypass to xtal for c906_0	0x1
7	clk_byp_1_7	R/W	Clock Bypass to xtal for c906_1	0x1
8	clk_byp_1_8	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_3	0x1
9	clk_byp_1_9	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_4	0x1
31:10	Reserved			

div_clk_a53_0

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	div_clk_a53_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide	0x00000201

Bits	Name	Access	Description	Reset
			Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	

div_clk_a53_1

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	div_clk_a53_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x00000001

div_clk_cpu_axi0

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	div_clk_cpu_axi0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_cpu_gic

Offset Address: 0x050

Bits	Name	Access	Description	Reset
31:0	div_clk_cpu_gic	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_tpu

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	div_clk_tpu	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll	0x00000301

Bits	Name	Access	Description	Reset
			3 : fpll	

div_clk_emmc

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	div_clk_emmc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_100k_emmc

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_emmc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_sd0

Offset Address: 0x070

Bits	Name	Access	Description	Reset
31:0	div_clk_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_100k_sd0

Offset Address: 0x078

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_sd1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
31:0	div_clk_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register	0x00000001

Bits	Name	Access	Description	Reset
			[20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	

div_clk_100k_sd1

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_spi_nand

Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	div_clk_spi_nand	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_500m_eth0

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	div_clk_500m_eth0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_gpio_db

Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	div_clk_gpio_db	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_sdma_aud0

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register	0x00000001

Bits	Name	Access	Description	Reset
			[20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	

div_clk_sdma_aud1

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_sdma_aud2

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_sdma_aud3

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_cam0_200

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	div_clk_cam0_200	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : xtal 1 : disppll	0x00000001

div_clk_axi4

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_axi6

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	div_clk_axi6	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_dsi_esc

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	div_clk_dsi_esc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_axi_vip

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000001

div_clk_src_vip_sys_0

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000301

Bits	Name	Access	Description	Reset
			[9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	

div_clk_src_vip_sys_1

Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000301

div_clk_disp_src_vip

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
31:0	div_clk_disp_src_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_axi_video_codec

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_video_codec	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : mipimpll 2 : cam1pll 3 : fpll	0x00000101

div_clk_vc_src0

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
31:0	div_clk_vc_src0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src	0x00000101

Bits	Name	Access	Description	Reset
			0 : apll 1 : mipimpll 2 : cam1pll 3 : fpll	

div_clk_1m

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
31:0	div_clk_1m	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_spi

Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	div_clk_spi	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_i2c

Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	div_clk_i2c	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_src_vip_sys_2

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000201

div_clk_audsrc

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	div_clk_audsrc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0:	0x00000001

Bits	Name	Access	Description	Reset
			Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	

div_clk_pwm_src_0

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	div_clk_pwm_src	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_ap_debug

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	div_clk_ap_debug	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_rtcsys_src_0

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	div_clk_src_rtc_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_c906_0_0

Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src	0x00000201

Bits	Name	Access	Description	Reset
			0 : tpll 1 : apll 2 : mipimpll 3 : mpll	

div_clk_c906_0_1

Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x00000001

div_clk_c906_1_0

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x00000001

div_clk_c906_1_1

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x00000001

div_clk_src_vip_sys_3

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0:	0x00000001

Bits	Name	Access	Description	Reset
			Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	

div_clk_src_vip_sys_4

Offset Address: 0x144

Bits	Name	Access	Description	Reset
	div_clk_src_vip_sys_4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000201

7 系统控制器

7.1 概述

系统控制器通过寄存器实现对芯片的一些系统控制即使能，包含系统软复位，时钟控制等。复位时钟已在其他章节说明，本章节描述一些其他系统功能模块之配置及状态寄存器

7.2 功能描述

7.2.1 全局复位使能

系统全局软复位、Debug 复位跟 Watch Dog 复位需要透过配置 reg_sw_root_reset_en 寄存器以使能。详细在 reg_sw_root_reset_en 有各比特之说明。

7.2.2 系统 DMA 通道映射

本芯片系统 DMA 内建 8 通道，各自配置 0~7 通道请求介面。而 0~7 的通道请求由系统控制寄存器 sdma_dma_ch_remap0、sdma_dma_ch_remap1 映射至下表外设介面之一。注意不能将多个通道配置为同一外设介面。

配置步骤：
配置 DMA 信道映像寄存器 sdma_dma_ch_remap0、sdma_dma_ch_remap1、update_dma_remp_0_3、update_dma_remp_4_7 写 1，使映射生效。

编号	DMA 介面	编号	DMA 介面
0	dma_rx_req_i2s0	24	dma_rx_req_i2c0
1	dma_tx_req_i2s0	25	dma_tx_req_i2c0
2	dma_rx_req_i2s1	26	dma_rx_req_i2c1
3	dma_tx_req_i2s1	27	dma_tx_req_i2c1
4	dma_rx_req_i2s2	28	dma_rx_req_i2c2
5	dma_tx_req_i2s2	29	dma_tx_req_i2c2
6	dma_rx_req_i2s3	30	dma_rx_req_i2c3
7	dma_tx_req_i2s3	31	dma_tx_req_i2c3
8	dma_rx_req_n_uart0	32	dma_rx_req_i2c4
9	dma_tx_req_n_uart0	33	dma_tx_req_i2c4

编号	DMA 介面	编号	DMA 介面
10	dma_rx_req_n_uart1	34	dma_rx_req_tdm0
11	dma_tx_req_n_uart1	35	dma_tx_req_tdm0
12	dma_rx_req_n_uart2	36	dma_rx_req_tdm1
13	dma_tx_req_n_uart2	37	dma_req_audsrc
14	dma_rx_req_n_uart3	38	dma_req_spi_nand
15	dma_tx_req_n_uart3	39	dma_req_spi_nor
16	dma_rx_req_spi0	40	dma_rx_req_n_uart4
17	dma_tx_req_spi0	41	dma_tx_req_n_uart4
18	dma_rx_req_spi1	42	dma_req_spi_nor1
19	dma_tx_req_spi1		
20	dma_rx_req_spi2		
21	dma_tx_req_spi2		
22	dma_rx_req_spi3		
23	dma_tx_req_spi3		

7.2.3 DDR AXI Urgent/Qos 配置

可以由控制系统控制器的 ddr_axi_urgent_ow、ddr_axi_urgent、ddr_axi_qos_0、ddr_axi_qos_1 配置来自子系统之 AXI 传输优先级，详细说明参照 DDR 控制器章节。

7.3 系统控制寄存器

7.3.1 系统控制寄存器概览

Bass address 0x03000000

Name	Address Offset	Description
conf_info	0x004	conf_info
sys_ctrl_reg	0x008	sys_ctrl_reg
usb_phy_ctrl_reg	0x048	usb_phy_ctrl_reg
sdma_dma_ch_remap0	0x154	sdma_dma_ch_remap0
sdma_dma_ch_remap1	0x158	sdma_dma_ch_remap1
top_timer_clk_sel	0x1a0	top_timer_clk_sel
top_wdt_ctrl	0x1a8	top_timer_clk_sel
ddr_axi_urgent_ow	0x1b8	ddr_axi_urgent_ow
ddr_axi_urgent	0x1bc	ddr_axi_urgent
ddr_axi_qos_0	0x1d8	ddr_axi_qos_0
ddr_axi_qos_1	0x1dc	ddr_axi_qos_1
sd_pwrsw_ctrl	0x1f4	sd_pwrsw_ctrl
sd_pwrsw_time	0x1f8	sd_pwrsw_time
ddr_axi_qos_ow	0x23c	ddr_axi_qos_ow
sd_ctrl_opt	0x294	additional control register for sd

Name	Address Offset	Description
sdma_dma_int_mux	0x298	Mux sdma channel interrupt to different processors

7.3.2 系统控制寄存器描述

conf_info

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	boot_sel	RO	[2:0] boot device selection 0: SPI_NAND 1: reserved 2: SPI_NOR 3: EMMC [7:3] : resreved	
8	io_sta_usbid	RO	IO status from USBID PAD	
9	io_sta_usbvbus	RO	IO status from USB_VBUS_DET PAD	
23:10	Reserved			
31:24	io_sta_trap	RO	io_sta_trap[0] : io_boot_rom_din io_sta_trap[1] : io_boot_dev0_din io_sta_trap[2] : io_boot_dev1_din io_sta_trap[3] : io_trap_sd0_pwr_din io_sta_trap[4] : io_pkg_type0_din io_sta_trap[5] : io_pkg_type1_din io_sta_trap[6] : io_pkg_type2_din io_sta_trap[7] : io_trap_zq_din	

sys_ctrl_reg

Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	Reserved			
5:2	reg_sw_root_reset_en	R/W	bit0 : wdt reset enable bit1 : cdbgrstreq enable bit2 : reserved bit3 : reg_soft_reset_x_system enable	0x0
31:6	Reserved			

usb_phy_ctrl_reg

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_usb_phy_external_vbusvalid	R/W	external vbus status	0x0
1	reg_usb_drive_vbus	R/W	drive vbus power	0x0
4:2	Reserved			
5	toreg_usb_id_en	RO	usb id pullup status	
6	reg_usb_phy_idpad_c_ow	R/W	usb id overwrite enable	0x0
7	reg_usb_phy_idpad_c_sw	R/W	usb id overwrite value	0x0
8	io_usb_phy_idpad_c	RO	usb id external IO pin status	
9	toreg_usb_phy_idpad_c	RO	usb id pin status	
31:10	Reserved			

sdma_dma_ch_remap0

Offset Address: 0x154

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch0	R/W	dma channel 0 mapping	0x0
7:6	Reserved			

Bits	Name	Access	Description	Reset
13:8	reg_dma_remap_ch1	R/W	dma channel 1 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch2	R/W	dma channel 2 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch3	R/W	dma channel 3 mapping	0x0
30	Reserved			
31	update_dma_remp_0_3	W1T	write 1 to update dma channel0~3 mapping	

sdma_dma_ch_remap1

Offset Address: 0x158

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch4	R/W	dma channel 4 mapping	0x0
7:6	Reserved			
13:8	reg_dma_remap_ch5	R/W	dma channel 5 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch6	R/W	dma channel 6 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch7	R/W	dma channel 7 mapping	0x0
30	Reserved			
31	update_dma_remp_4_7	W1T	write 1 to update dma channel4~7 mapping	

top_timer_clk_sel

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	reg_timer_clk_sel	R/W	timer0~7 clock selection. 0: xtal clock,1:32k clock	0x0
31:8	Reserved			

top_wdt_ctrl

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
2:0	reg_wdt_rst_sys_en	R/W	enable wdt0~wdt2 to reset system	0x7
3	Reserved			
6:4	reg_wdt_rst_cpu_en	R/W	enable wdt0~wdt2 to reset cpu	0x0
7	Reserved			
10:8	reg_wdt_clk_sel	R/W	top_wdt clock selection. 0: xtal clock,1:32k clock	0x0
31:11	Reserved			

ddr_axi_urgent_ow

Offset Address: 0x1b8

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1_ow	R/W	ddr axi port1 awurgent overwrite enable	0x1
1	reg_arurgent_m1_ow	R/W	ddr axi port1 arurgent overwrite enable	0x1
2	reg_awurgent_m2_ow	R/W	ddr axi port2 awurgent overwrite enable	0x1
3	reg_arurgent_m2_ow	R/W	ddr axi port2 arurgent overwrite enable	0x1
4	reg_awurgent_m3_ow	R/W	ddr axi port3 awurgent overwrite	0x1

Bits	Name	Access	Description	Reset
			enable	
5	reg_arurgent_m3_ow	R/W	ddr axi port3 arurgent overwrite enable	0x1
6	reg_awurgent_m4_ow	R/W	ddr axi port4 awurgent overwrite enable	0x1
7	reg_arurgent_m4_ow	R/W	ddr axi port4 arurgent overwrite enable	0x1
8	reg_awurgent_m5_ow	R/W	ddr axi port5 awurgent overwrite enable	0x1
9	reg_arurgent_m5_ow	R/W	ddr axi port5 arurgent overwrite enable	0x1
10	reg_awurgent_m6_ow	R/W	ddr axi port6 awurgent overwrite enable	0x1
11	reg_arurgent_m6_ow	R/W	ddr axi port6 arurgent overwrite enable	0x1
31:12	Reserved			

ddr_axi_urgent

Offset Address: 0x1bc

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1	R/W	ddr axi port1 awurgent overwrite value	0x0
1	reg_arurgent_m1	R/W	ddr axi port1 arurgent overwrite value	0x0
2	reg_awurgent_m2	R/W	ddr axi port2 awurgent overwrite value	0x0
3	reg_arurgent_m2	R/W	ddr axi port2 arurgent overwrite value	0x0
4	reg_awurgent_m3	R/W	ddr axi port3 awurgent overwrite value	0x0
5	reg_arurgent_m3	R/W	ddr axi port3 arurgent overwrite value	0x0
6	reg_awurgent_m4	R/W	ddr axi port4 awurgent overwrite value	0x0
7	reg_arurgent_m4	R/W	ddr axi port4 arurgent overwrite value	0x0
8	reg_awurgent_m5	R/W	ddr axi port5 awurgent overwrite value	0x0
9	reg_arurgent_m5	R/W	ddr axi port5 arurgent overwrite value	0x0
10	reg_awurgent_m6	R/W	ddr axi port6 awurgent overwrite value	0x0
11	reg_arurgent_m6	R/W	ddr axi port6 arurgent overwrite value	0x0
31:12	Reserved			

ddr_axi_qos_0

Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m1	R/W	ddr axi port1 awqos setting	0x0
7:4	reg_arqos_m1	R/W	ddr axi port1 arqos setting	0x0
11:8	reg_awqos_m2	R/W	ddr axi port2 awqos setting	0x0
15:12	reg_arqos_m2	R/W	ddr axi port2 arqos setting	0x0
19:16	reg_awqos_m3	R/W	ddr axi port3 awqos setting	0x0
23:20	reg_arqos_m3	R/W	ddr axi port3 arqos setting	0x0
27:24	reg_awqos_m4	R/W	ddr axi port4 awqos setting	0x0
31:28	reg_arqos_m4	R/W	ddr axi port4 arqos setting	0x0

ddr_axi_qos_1

Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m5	R/W	ddr axi port5 awqos setting	0x0
7:4	reg_arqos_m5	R/W	ddr axi port5 arqos setting	0x0
11:8	reg_awqos_m6	R/W	ddr axi port6 awqos setting	0x0
15:12	reg_arqos_m6	R/W	ddr axi port6 arqos setting	0x0
31:16	Reserved			

sd_pwrsw_ctrl

Offset Address: 0x1f4

Bits	Name	Access	Description	Reset
0	reg_en_pwrsw	R/W	18/33 IO power switch enable	0x0
1	reg_pwrsw_vsel	R/W	18/33 IO power switch enable 0: 3.3v 1: 1.8v	0x1
2	reg_pwrsw_disc	R/W	18/33 IO power switch discharge enable	0x0
3	reg_pwrsw_auto	R/W	18/33 IO power switch auto protect enable	0x1
31:4	Reserved			

sd_pwrsw_time

Offset Address: 0x1f8

Bits	Name	Access	Description	Reset
15:0	reg_tpwrup	R/W	18/33 IO power switch, power up protection time is 500x40ns = 20us	0x1f4
31:16	reg_tpwrdn	R/W	18/33 IO power switch, power down protection time is 500x40ns = 20us	0x1f4

ddr_axi_qos_ow

Offset Address: 0x23c

Bits	Name	Access	Description	Reset
0	reg_awqos_m1_ow	R/W	ddr axi port1 awqos overwrite enable	0x1
1	reg_arqos_m1_ow	R/W	ddr axi port1 arqos overwrite enable	0x1
2	reg_awqos_m2_ow	R/W	ddr axi port2 awqos overwrite enable	0x1
3	reg_arqos_m2_ow	R/W	ddr axi port2 arqos overwrite enable	0x1
4	reg_awqos_m3_ow	R/W	ddr axi port3 awqos overwrite enable	0x1
5	reg_arqos_m3_ow	R/W	ddr axi port3 arqos overwrite enable	0x1
6	reg_awqos_m4_ow	R/W	ddr axi port4 awqos overwrite enable	0x1
7	reg_arqos_m4_ow	R/W	ddr axi port4 arqos overwrite enable	0x1
8	reg_awqos_m5_ow	R/W	ddr axi port5 awqos overwrite enable	0x1
9	reg_arqos_m5_ow	R/W	ddr axi port5 arqos overwrite enable	0x1
10	reg_awqos_m6_ow	R/W	ddr axi port6 awqos overwrite enable	0x1
11	reg_arqos_m6_ow	R/W	ddr axi port6 arqos overwrite enable	0x1
31:12	Reserved			

sd_ctrl_opt

Offset Address: 0x294

Bits	Name	Access	Description	Reset
0	reg_sd0_carddet_ow	R/W	sd0 card detect over write enable	0x0
1	reg_sd0_carddet_sw	R/W	sd0 card detect over write value	0x0
7:2	Reserved			
8	reg_sd1_carddet_ow	R/W	sd1 card detect over write enable	0x0
9	reg_sd1_carddet_sw	R/W	sd1 card detect over write value	0x0
10	reg_sd1_phy_sel	R/W	sd1 phy sel	0x0
15:11	Reserved			
16	reg_sd0_pwr_en_polarity	R/W	off chip sd0 pwr en polarity 0: SD_LDO power ctrl high is power on , low is power off 1: SD_LDO power ctrl high is power off , low is power on	0x0
31:17	Reserved			

sdma_dma_int_mux

Offset Address: 0x298

Bits	Name	Access	Description	Reset
8:0	reg_dma_int_mux_cpu0	R/W	This register is used to mux separate sdma channel interrupts to CPU0. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x1FF
9	Reserved			
18:10	reg_dma_int_mux_cpu1	R/W	This register is used to mux separate sdma channel interrupts to CPU1. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
19	Reserved			
28:20	reg_dma_int_mux_cpu2	R/W	This register is used to mux separate sdma channel interrupts to CPU2. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
31:29	Reserved			

8 管脚复用与控制

8.1 管脚复用 PINMUX

8.1.1 接口功能与信号/引脚/FMUX 寄存器的对应关系

表格 8-1 接口功能与信号/引脚/FMUX 寄存器的对应关系表

Interface	Signal Name	Direction	PinName	Function Number
ADC	(ADC1)	I	ADC1	0
	(ADC2)	I	ADC2	0
	(ADC3)	I	ADC3	0
No die domain ADC	PWR_GPIO[2](PWRSAR.VIN1)	I	PWR_GPIO2	0
	PWR_GPIO[1](PWRSAR1.VIN2)	I	PWR_GPIO1	0
	PWR_VBAT_DET(PWRSAR.VIN3)	I	PWR_VBAT_DET	0
Audio	(PAD_AUD_AINL_MIC)	I	PAD_AUD_AINL_MIC	0
	(PAD_AUD_AINR_MIC)	I	PAD_AUD_AINR_MIC	0
	(PAD_AUD_AOUTL)	O	PAD_AUD_AOUTL	0
	(PAD_AUD_AOUTR)	O	PAD_AUD_AOUTR	0
Ethernet	(PAD_ETH_RXM)	I/O	PAD_ETH_RXM___EPHY_TXP	0
	(PAD_ETH_RXP)	I/O	PAD_ETH_RXP___EPHY_TXN	0
	(PAD_ETH_TXM)	I/O	PAD_ETH_TXM___EPHY_RXP	0
	(PAD_ETH_TXP)	I/O	PAD_ETH_TXP___EPHY_RXN	0
	EPHY_LNK_LED	O	SD0_CLK	6

	EPHY_LNK_LED	O	PWR_WAKEUP0	6
	EPHY_LNK_LED	O	PWR_WAKEUP1	6
	EPHY_LNK_LED	O	SD1_CMD	5
	EPHY_LNK_LED	O	ADC3	5
	EPHY_LNK_LED	O	PWR_GPIO1	3
	EPHY_SPD_LED	O	SD0_CMD	6
	EPHY_SPD_LED	O	PWR_BUTTON1	6
	EPHY_SPD_LED	O	PWR_ON	6
	EPHY_SPD_LED	O	SD1_CLK	5
	EPHY_SPD_LED	O	ADC2	5
	EPHY_SPD_LED	O	PWR_GPIO2	3
DSI/LVDS	(PAD_MIPI_TXM0)	I/O	PAD_MIPI_TXM0	0
	(PAD_MIPI_TXP0)	I/O	PAD_MIPI_TXP0	0
	(PAD_MIPI_TXM1)	I/O	PAD_MIPI_TXM1	3
	(PAD_MIPI_TXP1)	I/O	PAD_MIPI_TXP1	3
	(PAD_MIPI_TXM2)	I/O	PAD_MIPI_TXM2	3
	(PAD_MIPI_TXP2)	I/O	PAD_MIPI_TXP2	3
	(PAD_MIPI_TXM3)	I/O	PAD_MIPI_TXM3	3
	(PAD_MIPI_TXP3)	I/O	PAD_MIPI_TXP3	3
	(PAD_MIPI_TXM4)	I/O	PAD_MIPI_TXM4	3
	(PAD_MIPI_TXP4)	I/O	PAD_MIPI_TXP4	3
CSI/sLVDS /HiSPI	(PAD_MIPIRX0N)	I/O	PAD_MIPIRX0N	0

	(PAD_MIPIRX0P)	I/O	PAD_MIPIRX0P	0
	(PAD_MIPIRX1N)	I/O	PAD_MIPIRX1N	0
	(PAD_MIPIRX1P)	I/O	PAD_MIPIRX1P	0
	(PAD_MIPIRX2N)	I/O	PAD_MIPIRX2N	0
	(PAD_MIPIRX2P)	I/O	PAD_MIPIRX2P	0
	(PAD_MIPIRX3N)	I/O	PAD_MIPIRX3N	3
	(PAD_MIPIRX3P)	I/O	PAD_MIPIRX3P	3
	(PAD_MIPIRX4N)	I/O	PAD_MIPIRX4N	3
	(PAD_MIPIRX4P)	I/O	PAD_MIPIRX4P	3
	(PAD_MIPIRX5N)	I/O	PAD_MIPIRX5N	0
	(PAD_MIPIRX5P)	I/O	PAD_MIPIRX5P	0
Aux clock out	AUX0	O	AUX0	0
	AUX0	O	SD0_D1	2
	AUX0	O	CLK32K	1
	AUX0	O	UART0_RX	5
	AUX0	O	JTAG_CPU_TMS	5
	AUX1	O	CAM_MCLK0	2
	AUX1	O	SD0_D2	2
	AUX1	O	CLK25M	1
	AUX1	O	UART0_TX	5
	AUX1	O	JTAG_CPU_TCK	5
	AUX2	O	EMMC_RSTN	4

	AUX2	O	CAM_MCLK1	2
	CLK25M	O	CLK25M	0
	CLK32K	O	CLK32K	0
Camera Interface	CAM_HS0	O	CAM_PD0	4
	CAM_HS0	O	CAM_MCLK1	4
	CAM_HS0	O	SD1_CLK	4
	CAM_HS0	O	PAD_MIPI_TXP0	6
	CAM_HS0	O	PAD_ETH_RXP___EPHY_TXN	5
	CAM_HS0	O	VIVO_D7	7
	CAM_HS0	O	VIVO_D5	7
	CAM_HS0	O	VIVO_D3	7
	CAM_HS0	O	PAD_MIPI_TXP3	7
	CAM_MCLK0	O	PAD_MIPI_TXP4	4
	CAM_MCLK0	O	PAD_MIPI_TXP0	4
	CAM_MCLK0	O	SD0_D3	1
	CAM_MCLK0	O	UART0_RX	1
	CAM_MCLK0	O	JTAG_CPU_TMS	1
	CAM_MCLK0	O	VIVO_CLK	5
	CAM_MCLK0	O	CAM_MCLK0	0
	CAM_MCLK0	O	SD1_D3	4
	CAM_MCLK0	O	SD1_D2	4
	CAM_MCLK0	O	USB_VBUS_DET	4

	CAM_MCLK0	O	PAD_MIPIRX3N	4
	CAM_MCLK0	O	PAD_MIPIRX0P	4
	CAM_MCLK0	O	MUX_SPI1_CS	2
	CAM_MCLK0	O	PAD_ETH_RXM___EPHY_TXP	2
	CAM_MCLK0	O	CLK32K	6
	CAM_MCLK0	O	ADC3	1
	CAM_MCLK0	O	PAD_ETH_TXP___EPHY_RXN	5
	CAM_MCLK0	O	VIVO_D10	5
	CAM_MCLK0	O	PAD_MIPIRX5N	5
	CAM_MCLK0	O	PAD_MIPIRX4N	5
	CAM_MCLK0	O	GPIO_RTX___EPHY_RTX	5
	CAM_MCLK1	O	SD1_D1	4
	CAM_MCLK1	O	SD1_D0	4
	CAM_MCLK1	O	PAD_MIPI_TXM4	4
	CAM_MCLK1	O	PAD_MIPI_TXM0	4
	CAM_MCLK1	O	MUX_SPI1_SCK	2
	CAM_MCLK1	O	PAD_ETH_RXP___EPHY_TXN	2
	CAM_MCLK1	O	CLK25M	6
	CAM_MCLK1	O	ADC2	1
	CAM_MCLK1	O	USB_VBUS_DET	5
	CAM_MCLK1	O	PAD_ETH_TXM___EPHY_RXP	5
	CAM_MCLK1	O	VIVO_D9	5

	CAM_MCLK1	O	PAD_MIPIRX5P	5
	CAM_MCLK1	O	PAD_MIPIRX4P	5
	CAM_MCLK1	O	PAD_MIPIRX0N	5
	CAM_MCLK1	O	SD0_D0	1
	CAM_MCLK1	O	UART0_TX	1
	CAM_MCLK1	O	JTAG_CPU_TCK	1
	CAM_MCLK1	O	VIVO_D2	5
	CAM_MCLK1	O	CAM_MCLK1	0
	CAM_VS0	O	CAM_RST0	4
	CAM_VS0	O	CAM_PD1	4
	CAM_VS0	O	SD1_CMD	4
	CAM_VS0	O	PAD_MIPI_TXM0	6
	CAM_VS0	O	PAD_ETH_RXM___EPHY_TXP	5
	CAM_VS0	O	VIVO_D8	7
	CAM_VS0	O	VIVO_D6	7
	CAM_VS0	O	VIVO_D4	7
	CAM_VS0	O	PAD_MIPI_TXM3	7
Parallel Video Out	VO_CLK0	O	PAD_MIPI_TXP2	2
	VO_CLK1	O	VIVO_CLK	2
	VO_D[0]	O	PAD_MIPI_TXM2	2
	VO_D[1]	O	PAD_MIPI_TXP1	2
	VO_D[2]	O	PAD_MIPI_TXM1	2

	VO_D[3]	O	PAD_MIPI_TXP0	2
	VO_D[4]	O	PAD_MIPI_TXM0	2
	VO_D[5]	O	PAD_MIPIRX0P	2
	VO_D[6]	O	PAD_MIPIRX0N	2
	VO_D[7]	O	PAD_MIPIRX1P	2
	VO_D[8]	O	PAD_MIPIRX1N	2
	VO_D[9]	O	PAD_MIPIRX2P	2
	VO_D[10]	O	PAD_MIPIRX2N	2
	VO_D[11]	O	PAD_MIPIRX5P	2
	VO_D[12]	O	PAD_MIPIRX5N	2
	VO_D[13]	O	VIVO_D0	2
	VO_D[14]	O	VIVO_D1	2
	VO_D[15]	O	VIVO_D2	2
	VO_D[16]	O	VIVO_D3	2
	VO_D[17]	O	VIVO_D4	2
	VO_D[18]	O	VIVO_D5	2
	VO_D[19]	O	VIVO_D6	2
	VO_D[20]	O	VIVO_D7	2
	VO_D[21]	O	VIVO_D8	2
	VO_D[22]	O	VIVO_D9	2
	VO_D[23]	O	VIVO_D10	2
	VO_D[24]	O	PAD_MIPI_TXM4	2

	VO_D[25]	O	PAD_MIPI_TXP4	2
	VO_D[26]	O	PAD_MIPI_TXM3	2
	VO_D[27]	O	PAD_MIPI_TXP3	2
	VO_D[28]	O	JTAG_CPU_TMS	7
	VO_D[29]	O	JTAG_CPU_TCK	7
	VO_D[30]	O	JTAG_CPU_TRST	6
	VO_D[31]	O	AUX0	5
	VO_D[32]	O	SD1_D3	0
	VO_D[33]	O	SD1_D2	0
	VO_D[34]	O	SD1_D1	0
	VO_D[35]	O	SD1_D0	0
	VO_D[36]	O	SD1_CMD	0
	VO_D[37]	O	SD1_CLK	0
Parallel Video In	VI0_CLK	I	PAD_MIPIRX4N	1
	VI0_D[0]	I	PAD_MIPIRX4P	1
	VI0_D[1]	I	PAD_MIPIRX3N	1
	VI0_D[10]	I	PAD_MIPI_TXP0	1
	VI0_D[11]	I	PAD_MIPI_TXM1	1
	VI0_D[12]	I	PAD_MIPI_TXP1	1
	VI0_D[13]	I	PAD_MIPI_TXM2	1
	VI0_D[14]	I	PAD_MIPI_TXP2	1
	VI0_D[15]	I	PAD_MIPI_TXM4	0

	VI0_D[16]	I	PAD_MIPI_TXP4	0
	VI0_D[17]	I	PAD_MIPI_TXM3	0
	VI0_D[18]	I	PAD_MIPI_TXP3	0
	VI0_D[2]	I	PAD_MIPIRX3P	1
	VI0_D[3]	I	PAD_MIPIRX2N	1
	VI0_D[4]	I	PAD_MIPIRX2P	1
	VI0_D[5]	I	PAD_MIPIRX1N	1
	VI0_D[6]	I	PAD_MIPIRX1P	1
	VI0_D[7]	I	PAD_MIPIRX0N	1
	VI0_D[8]	I	PAD_MIPIRX0P	1
	VI0_D[9]	I	PAD_MIPI_TXM0	1
	VI1_CLK	I	VIVO_CLK	1
	VI1_D[0]	I	VIVO_D0	1
	VI1_D[1]	I	VIVO_D1	1
	VI1_D[10]	I	VIVO_D10	1
	VI1_D[11]	I	PAD_MIPIRX5N	1
	VI1_D[12]	I	PAD_MIPIRX5P	1
	VI1_D[13]	I	PAD_MIPIRX4N	2
	VI1_D[14]	I	PAD_MIPIRX4P	2
	VI1_D[15]	I	PAD_MIPIRX3N	2
	VI1_D[16]	I	PAD_MIPIRX3P	2
	VI1_D[17]	I	PAD_MIPIRX2N	4

	VI1_D[18]	I	PAD_MIPIRX2P	4
	VI1_D[2]	I	VIVO_D2	1
	VI1_D[3]	I	VIVO_D3	1
	VI1_D[4]	I	VIVO_D4	1
	VI1_D[5]	I	VIVO_D5	1
	VI1_D[6]	I	VIVO_D6	1
	VI1_D[7]	I	VIVO_D7	1
	VI1_D[8]	I	VIVO_D8	1
	VI1_D[9]	I	VIVO_D9	1
	VI2_CLK	I	VIVO_CLK	0
	VI2_D[0]	I	VIVO_D0	0
	VI2_D[1]	I	VIVO_D1	0
	VI2_D[2]	I	VIVO_D2	0
	VI2_D[3]	I	VIVO_D3	0
	VI2_D[4]	I	VIVO_D4	0
	VI2_D[5]	I	VIVO_D5	0
	VI2_D[6]	I	VIVO_D6	0
	VI2_D[7]	I	VIVO_D7	0
eMMC	EMMC_CLK	O	EMMC_CLK	0
	EMMC_CMD	I/O	EMMC_CMD	0
	EMMC_DAT[0]	I/O	EMMC_DAT0	0
	EMMC_DAT[1]	I/O	EMMC_DAT1	0

	EMMC_DAT[2]	I/O	EMMC_DAT2	0
	EMMC_DAT[3]	I/O	EMMC_DAT3	0
	EMMC_RSTN	O	EMMC_RSTN	0
SPI_NAND	SPINAND_CLK	O	EMMC_CLK	2
	SPINAND_CS	O	EMMC_DAT1	2
	SPINAND_HOLD	I/O	EMMC_DAT2	2
	SPINAND_MISO	I/O	EMMC_CMD	2
	SPINAND_MOSI	I/O	EMMC_DAT0	2
	SPINAND_WP	I/O	EMMC_DAT3	2
SPI_NOR	SPINOR_CS_X	O	EMMC_DAT1	1
	SPINOR_HOLD_X	I/O	EMMC_DAT2	1
	SPINOR_MISO	I/O	EMMC_CMD	1
	SPINOR_MOSI	I/O	EMMC_DAT0	1
	SPINOR_SCK	O	EMMC_CLK	1
	SPINOR_WP_X	I/O	EMMC_DAT3	1
I2C	IIC1_SCL	I/O	PAD_MIPIRX4P	4
	IIC1_SCL	I/O	PAD_MIPIRX0N	4
	IIC1_SCL	I/O	VIVO_D9	6
	IIC1_SCL	I/O	SD0_CMD	1
	IIC1_SCL	I/O	SD0_D2	1
	IIC1_SCL	I/O	VIVO_D3	5
	IIC1_SCL	I/O	PAD_MIPI_TXP2	4

	IIC1_SCL	I/O	SD1_D3	2
	IIC1_SCL	I/O	MUX_SPI1_MOSI	2
	IIC1_SCL	I/O	PAD_ETH_TXP___EPHY_RXN	2
	IIC1_SCL	I/O	PAD_MIPI_TXP4	6
	IIC1_SCL	I/O	PAD_MIPI_TXP3	6
	IIC1_SCL	I/O	SD1_D2	1
	IIC1_SDA	I/O	PAD_MIPIRX4N	4
	IIC1_SDA	I/O	PAD_MIPIRX1P	4
	IIC1_SDA	I/O	VIVO_D10	6
	IIC1_SDA	I/O	SD0_CLK	1
	IIC1_SDA	I/O	SD0_D1	1
	IIC1_SDA	I/O	VIVO_D4	5
	IIC1_SDA	I/O	PAD_MIPI_TXM2	4
	IIC1_SDA	I/O	SD1_D0	2
	IIC1_SDA	I/O	MUX_SPI1_MISO	2
	IIC1_SDA	I/O	PAD_ETH_TXM___EPHY_RXP	2
	IIC1_SDA	I/O	PAD_MIPI_TXM4	6
	IIC1_SDA	I/O	PAD_MIPI_TXM3	6
	IIC1_SDA	I/O	SD1_D1	1
	IIC2_SCL	I/O	IIC2_SCL	0
	IIC2_SCL	I/O	PAD_MIPI_TXP3	4
	IIC2_SCL	I/O	PAD_MIPI_TXP1	4

	IIC2_SCL	I/O	PWR_GPIO1	6
	IIC2_SCL	I/O	VIVO_D8	6
	IIC2_SDA	I/O	PAD_MIPI_TXM3	4
	IIC2_SDA	I/O	PAD_MIPI_TXM1	4
	IIC2_SDA	I/O	IIC2_SDA	0
	IIC2_SDA	I/O	PWR_GPIO2	6
	IIC2_SDA	I/O	VIVO_D7	6
	IIC3_SCL	I/O	VIVO_D0	5
	IIC3_SCL	I/O	IIC3_SCL	0
	IIC3_SCL	I/O	SD1_CMD	2
	IIC3_SDA	I/O	VIVO_D1	5
	IIC3_SDA	I/O	IIC3_SDA	0
	IIC3_SDA	I/O	SD1_CLK	2
	IIC4_SCL	I/O	ADC3	2
	IIC4_SCL	I/O	CAM_RST0	6
	IIC4_SCL	I/O	PWR_WAKEUP0	5
	IIC4_SCL	I/O	PWR_WAKEUP1	5
	IIC4_SCL	I/O	PAD_MIPIRX2N	5
	IIC4_SCL	I/O	VIVO_D1	7
	IIC4_SDA	I/O	ADC2	2
	IIC4_SDA	I/O	CAM_PD1	6
	IIC4_SDA	I/O	PWR_BUTTON1	5

	IIC4_SDA	I/O	PWR_ON	5
	IIC4_SDA	I/O	PAD_MIPIRX2P	5
	IIC4_SDA	I/O	VIVO_D0	7
no die I2C	PWR_IIC_SCL	I/O	PWR_GPIO1	5
	PWR_IIC_SDA	I/O	PWR_GPIO2	5
IIS	IIS1_BCLK	I/O	PAD_AUD_AINL_MIC	4
	IIS1_DI	I	PAD_AUD_AOUTR	4
	IIS1_DI	I	PAD_AUD_AINR_MIC	6
	IIS1_DO	O	PAD_AUD_AINR_MIC	4
	IIS1_DO	O	PAD_AUD_AOUTR	6
	IIS1_LRCK	I/O	PAD_AUD_AOUTL	4
	IIS1_MCLK	I/O	AUX0	4
	IIS1_MCLK	I/O	PAD_MIPIRX5P	4
	IIS1_MCLK	I/O	CAM_PD0	1
	IIS1_MCLK	I/O	CAM_PD1	1
	IIS1_MCLK	I/O	IIC2_SDA	5
	IIS2_BCLK	I/O	UART2_TX	6
	IIS2_BCLK	I/O	PAD_AUD_AINL_MIC	5
	IIS2_BCLK	I/O	PAD_ETH_TXM___EPHY_RXP	7
	IIS2_DI	I	UART2_RX	6
	IIS2_DI	I	PAD_AUD_AINR_MIC	5
	IIS2_DI	I	PAD_ETH_RXM___EPHY_TXP	7

	IIS2_DO	O	UART2_RTS	6
	IIS2_DO	O	PAD_AUD_AOUTR	5
	IIS2_DO	O	PAD_ETH_RXP___EPHY_TXN	7
	IIS2_LRCK	I/O	UART2_CTS	6
	IIS2_LRCK	I/O	PAD_AUD_AOUTL	5
	IIS2_LRCK	I/O	PAD_ETH_TXP___EPHY_RXN	7
	IIS2_MCLK	I/O	IIC2_SDA	6
Keyscan	KEY_COL0	I/O	CLK32K	5
	KEY_COL0	I/O	MUX_SPI1_MOSI	5
	KEY_COL1	I/O	CLK25M	5
	KEY_COL1	I/O	MUX_SPI1_MISO	5
	KEY_COL2	I/O	IIC2_SCL	7
	KEY_COL2	I/O	ADC1	4
	KEY_COL3	I/O	IIC2_SDA	7
	KEY_COL3	I/O	UART2_RX	4
	KEY_ROW0	I/O	PAD_MIPIRX4N	6
	KEY_ROW0	I/O	UART2_RTS	4
	KEY_ROW1	I/O	PAD_MIPIRX4P	6
	KEY_ROW1	I/O	UART2_CTS	4
	KEY_ROW2	I/O	PAD_MIPIRX1P	6
	KEY_ROW2	I/O	MUX_SPI1_SCK	5
	KEY_ROW3	I/O	PAD_MIPIRX1N	6

	KEY_ROW3	I/O	MUX_SPI1_CS	5
	KEY_ROW3	I/O	UART2_TX	4
PWM	PWM[0]	I/O	PWM0_BUCK	0
	PWM[1]	I/O	GPIO_RTX___EPHY_RTX	4
	PWM[1]	I/O	VIVO_D10	0
	PWM[2]	I/O	VIVO_D2	6
	PWM[2]	I/O	CLK32K	4
	PWM[2]	I/O	GPIO_ZQ___PAD_ZQ	4
	PWM[2]	I/O	VIVO_D9	0
	PWM[3]	I/O	VIVO_D1	6
	PWM[3]	I/O	VIVO_D8	0
	PWM[3]	I/O	CLK25M	4
	PWM[4]	I/O	UART0_TX	2
	PWM[4]	I/O	SD1_D3	7
	PWM[5]	I/O	UART0_RX	2
	PWM[5]	I/O	SD1_D2	7
	PWM[6]	I/O	JTAG_CPU_TCK	2
	PWM[6]	I/O	SD1_D1	7
	PWM[7]	I/O	JTAG_CPU_TMS	2
	PWM[7]	I/O	SD1_D0	7
	PWM[8]	I/O	PWR_GPIO0	4
	PWM[8]	I/O	MUX_SPI1_MOSI	4

	PWM[8]	I/O	SD1_CMD	7
	PWM[8]	I/O	UART2_RTS	1
	PWM[8]	I/O	PAD_MIPI_TXM2	5
	PWM[9]	I/O	PWR_GPIO1	4
	PWM[9]	I/O	MUX_SPI1_MISO	4
	PWM[9]	I/O	SD1_CLK	7
	PWM[9]	I/O	UART2_CTS	1
	PWM[9]	I/O	PAD_MIPI_TXP2	5
	PWM[10]	I/O	PWR_GPIO2	4
	PWM[10]	I/O	MUX_SPI1_SCK	4
	PWM[10]	I/O	UART2_RX	1
	PWM[10]	I/O	SD0_D3	5
	PWM[10]	I/O	PAD_MIPI_TXM1	5
	PWM[11]	I/O	MUX_SPI1_CS	4
	PWM[11]	I/O	UART2_TX	1
	PWM[11]	I/O	SD0_D2	5
	PWM[11]	I/O	PAD_MIPI_TXP1	5
	PWM[12]	I/O	ADC3	4
	PWM[12]	I/O	PAD_ETH_TXM___EPHY_RXP	4
	PWM[12]	I/O	SD0_D1	5
	PWM[12]	I/O	PAD_MIPI_TXM4	5
	PWM[13]	I/O	ADC2	4

	PWM[13]	I/O	PAD_ETH_TXP___EPHY_RXN	4
	PWM[13]	I/O	SD0_D0	5
	PWM[13]	I/O	PAD_MIPI_TXP4	5
	PWM[14]	I/O	PAD_ETH_RXM___EPHY_TXP	4
	PWM[14]	I/O	IIC2_SCL	1
	PWM[14]	I/O	SD0_CMD	5
	PWM[14]	I/O	PAD_MIPI_TXM3	5
	PWM[14]	I/O	PAD_MIPI_TXM0	5
	PWM[15]	I/O	PAD_ETH_RXP___EPHY_TXN	4
	PWM[15]	I/O	IIC2_SDA	1
	PWM[15]	I/O	SD0_CLK	5
	PWM[15]	I/O	PAD_MIPI_TXP3	5
	PWM[15]	I/O	PAD_MIPI_TXP0	5
CA53 JTAG (2W) RISCV JTAG (4W) I2C0	CR_4WTCK	I/O	JTAG_CPU_TCK	0
	CR_4WTMS	I/O	JTAG_CPU_TMS	0
	CR_2WTCK	I/O	PAD_MIPIRX3P	0
	CR_2WTCK	I/O	PAD_MIPI_TXP1	0
	CR_2WTCK	I/O	PWR_GPIO2	7
	CR_SCL0	I/O	PAD_MIPIRX4N	0
	CR_SCL0	I/O	PAD_MIPI_TXP2	0
	CR_SCL0	I/O	CLK32K	2
	CR_2WTMS	I/O	PAD_MIPIRX3N	0

	CR_2WTMS	I/O	PAD_MIPI_TXM1	0
	CR_2WTMS	I/O	CLK25M	2
	CR_SDA0	I/O	PAD_MIPIRX4P	0
	CR_SDA0	I/O	PAD_MIPI_TXM2	0
	CR_SDA0	I/O	PWR_GPIO1	7
	CR_4WTDI	I/O	IIC0_SCL	0
	CR_4WTDO	I/O	IIC0_SDA	0
	JTAG_CPU_TRST	I	JTAG_CPU_TRST	0
System	PWR_BUTTON1	I	PWR_BUTTON1	0
	PWR_ON	I	PWR_ON	0
	PWR_RSTN	I	PWR_RSTN	0
	PWR_SEQ1	O	PWR_SEQ1	0
	PWR_SEQ2	O	PWR_SEQ2	0
	PWR_SEQ3	O	PWR_SEQ3	0
	PWR_WAKEUP0	I	PWR_WAKEUP0	0
	PWR_WAKEUP1	I	PWR_WAKEUP1	0
	RSTN	I	RSTN	0
	USB_ID	I	USB_ID	0
	USB_VBUS_DET	I	USB_VBUS_DET	0
	USB_VBUS_EN	O	USB_VBUS_EN	0
no die IR	PWR_IR0	I	PWR_WAKEUP0	1
	PWR_IR1	I	PWR_WAKEUP1	1

SPI_NOR1	PWR_SPINOR1_CS_X	O	SD1_D3	6
	PWR_SPINOR1_HOLD_X	I/O	SD1_D2	6
	PWR_SPINOR1_MISO	I/O	SD1_D0	6
	PWR_SPINOR1_MOSI	I/O	SD1_CMD	6
	PWR_SPINOR1_SCK	O	SD1_CLK	6
	PWR_SPINOR1_WP_X	I/O	SD1_D1	6
RMII	RMII0_IRQ	I/O	VIVO_D10	4
	RMII0_MDC	I/O	VIVO_D4	4
	RMII0_MDIO	I/O	VIVO_D8	4
	RMII0_REFCLKI	I/O	VIVO_D6	4
	RMII0_RXD0	I	VIVO_D5	4
	RMII0_RXD1	I	VIVO_D7	4
	RMII0_RXDV	I	VIVO_D1	4
	RMII0_TXCLK	O	VIVO_D0	4
	RMII0_TXD0	O	VIVO_D3	4
	RMII0_TXD1	O	VIVO_D2	4
	RMII0_TXEN	O	VIVO_CLK	4
SD1	SD1_CLK	O	PAD_MIPI_TXM4	1
	SD1_CMD	I/O	PAD_MIPI_TXP4	1
	SD1_D0	I/O	PAD_MIPI_TXM3	1
	SD1_D1	I/O	PAD_MIPI_TXP3	1
	SD1_D2	I/O	PAD_MIPI_TXM2	7

	SD1_D3	I/O	PAD_MIPI_TXP2	7
	PWR_SD1_CLK	O	SD1_CLK	0
	PWR_SD1_CMD	I/O	SD1_CMD	0
	PWR_SD1_D0	I/O	SD1_D0	0
	PWR_SD1_D1	I/O	SD1_D1	0
	PWR_SD1_D2	I/O	SD1_D2	0
	PWR_SD1_D3	I/O	SD1_D3	0
SD0	SDIO0_CD	I	SD0_CD	0
	SDIO0_CLK	O	SD0_CLK	0
	SDIO0_CMD	I/O	SD0_CMD	0
	SDIO0_D[0]	I/O	SD0_D0	0
	SDIO0_D[1]	I/O	SD0_D1	0
	SDIO0_D[2]	I/O	SD0_D2	0
	SDIO0_D[3]	I/O	SD0_D3	0
	SDIO0_PWR_EN	O	SD0_PWR_EN	0
SPI	SPI0_CS_X	O	SD0_D3	2
	SPI0_CS_X	O	PAD_MIPI_TXP2	6
	SPI0_SCK	O	SD0_CLK	2
	SPI0_SCK	O	PAD_MIPI_TXM2	6
	SPI0_SDI	I	SD0_D0	2
	SPI0_SDI	I	PAD_MIPI_TXP1	6
	SPI0_SDO	I/O	SD0_CMD	2

	SPI0_SDO	I/O	PAD_MIPI_TXM1	6
	SPI1_CS_X	O	MUX_SPI1_CS	6
	SPI1_CS_X	O	PAD_ETH_RXM___EPHY_TXP	6
	SPI1_SCK	O	MUX_SPI1_SCK	6
	SPI1_SCK	O	PAD_ETH_RXP___EPHY_TXN	6
	SPI1_SDI	I	MUX_SPI1_MISO	6
	SPI1_SDI	I	PAD_ETH_TXM___EPHY_RXP	6
	SPI1_SDO	I/O	MUX_SPI1_MOSI	6
	SPI1_SDO	I/O	PAD_ETH_TXP___EPHY_RXN	6
	SPI2_CS_X	O	SD1_D3	1
	SPI2_SCK	O	SD1_CLK	1
	SPI2_SDI	I	SD1_D0	1
	SPI2_SDO	I/O	SD1_CMD	1
	SPI3_CS_X	O	VIVO_D5	5
	SPI3_SCK	O	VIVO_D6	5
	SPI3_SDI	I	VIVO_D7	5
	SPI3_SDO	I/O	VIVO_D8	5
UART	UART0_RX	I	UART0_RX	0
	UART0_TX	O	UART0_TX	0
	UART1_CTS	I	JTAG_CPU_TCK	4
	UART1_RTS	O	JTAG_CPU_TMS	4
	UART1_RX	I	SD0_D2	4

	UART1_RX	I	UART0_RX	4
	UART1_RX	I	PWR_BUTTON1	4
	UART1_RX	I	PWR_ON	4
	UART1_RX	I	JTAG_CPU_TCK	6
	UART1_RX	I	IIC0_SDA	1
	UART1_TX	O	SD0_D1	4
	UART1_TX	O	UART0_TX	4
	UART1_TX	O	PWR_WAKEUP0	4
	UART1_TX	O	PWR_WAKEUP1	4
	UART1_TX	O	JTAG_CPU_TMS	6
	UART1_TX	O	IIC0_SCL	1
	UART2_CTS	I	VIVO_D4	6
	UART2_CTS	I	UART2_CTS	0
	UART2_RTS	O	VIVO_D3	6
	UART2_RTS	O	UART2_RTS	0
	UART2_RX	I	UART2_RX	0
	UART2_RX	I	IIC2_SCL	4
	UART2_RX	I	IIC0_SDA	2
	UART2_RX	I	SD1_D1	2
	UART2_RX	I	VIVO_D5	6
	UART2_RX	I	PWR_GPIO1	1
	UART2_RX	I	VIVO_D9	7

	UART2_RX	I	VIVO_CLK	7
	UART2_TX	O	UART2_TX	0
	UART2_TX	O	IIC2_SDA	4
	UART2_TX	O	IIC0_SCL	2
	UART2_TX	O	SD1_D2	2
	UART2_TX	O	VIVO_D6	6
	UART2_TX	O	PWR_GPIO0	1
	UART2_TX	O	VIVO_D10	7
	UART2_TX	O	VIVO_D2	7
	UART3_CTS	I	MUX_SPI1_CS	1
	UART3_CTS	I	PAD_ETH_RXM___EPHY_TXP	1
	UART3_CTS	I	SD1_D3	5
	UART3_RTS	O	MUX_SPI1_MISO	1
	UART3_RTS	O	PAD_ETH_TXM___EPHY_RXP	1
	UART3_RTS	O	SD1_D0	5
	UART3_RX	I	SD0_D3	4
	UART3_RX	I	MUX_SPI1_MOSI	1
	UART3_RX	I	PAD_ETH_TXP___EPHY_RXN	1
	UART3_RX	I	SD1_D1	5
	UART3_RX	I	ADC2	7
	UART3_TX	O	SD0_D0	4
	UART3_TX	O	MUX_SPI1_SCK	1

	UART3_TX	O	PAD_ETH_RXP___EPHY_TXN	1
	UART3_TX	O	SD1_D2	5
	UART3_TX	O	ADC3	7
	UART4_CTS	I	UART2_CTS	5
	UART4_RTS	O	UART2_RTS	5
	UART4_RX	I	UART2_RX	5
	UART4_TX	O	UART2_TX	5
no die UART	PWR_UART0_RX	I	PWR_GPIO0	2
	PWR_UART0_TX	O	PWR_WAKEUP0	2
	PWR_UART1_RX	I	UART2_RX	2
	PWR_UART1_TX	O	UART2_TX	2
Wiegand	WG0_D0	I/O	PAD_MIPIRX5N	6
	WG0_D0	I/O	PWR_WAKEUP1	7
	WG0_D0	I/O	SD0_D0	6
	WG0_D0	I/O	IIC0_SCL	5
	WG0_D1	I/O	PAD_MIPIRX5P	6
	WG0_D1	I/O	PWR_ON	7
	WG0_D1	I/O	SD0_D1	6
	WG0_D1	I/O	IIC0_SDA	5
	WG1_D0	I/O	VIVO_D0	6
	WG1_D0	I/O	SD0_D2	6
	WG1_D0	I/O	IIC0_SDA	6

	WG1_D0	I/O	UART2_RTS	7
	WG1_D1	I/O	VIVO_CLK	6
	WG1_D1	I/O	SD0_D3	6
	WG1_D1	I/O	AUX0	6
	WG1_D1	I/O	UART2_CTS	7
	WG2_D0	I/O	ADC3	6
	WG2_D0	I/O	PWR_WAKEUP0	7
	WG2_D0	I/O	UART2_TX	7
	WG2_D1	I/O	ADC2	6
	WG2_D1	I/O	PWR_BUTTON1	7
	WG2_D1	I/O	UART2_RX	7
GPIO	XGPIOA[0]	I/O	CAM_MCLK0	3
	XGPIOA[1]	I/O	CAM_PD0	3
	XGPIOA[2]	I/O	CAM_RST0	3
	XGPIOA[3]	I/O	CAM_MCLK1	3
	XGPIOA[4]	I/O	CAM_PD1	3
	XGPIOA[5]	I/O	IIC3_SCL	3
	XGPIOA[6]	I/O	IIC3_SDA	3
	XGPIOA[7]	I/O	SD0_CLK	3
	XGPIOA[8]	I/O	SD0_CMD	3
	XGPIOA[9]	I/O	SD0_D0	3
	XGPIOA[10]	I/O	SD0_D1	3

	XGPIOA[11]	I/O	SD0_D2	3
	XGPIOA[12]	I/O	SD0_D3	3
	XGPIOA[13]	I/O	SD0_CD	3
	XGPIOA[14]	I/O	SD0_PWR_EN	3
	XGPIOA[15]	I/O	SPK_EN	3
	XGPIOA[16]	I/O	UART0_TX	3
	XGPIOA[17]	I/O	UART0_RX	3
	XGPIOA[18]	I/O	JTAG_CPU_TCK	3
	XGPIOA[19]	I/O	JTAG_CPU_TMS	3
	XGPIOA[20]	I/O	JTAG_CPU_TRST	3
	XGPIOA[21]	I/O	EMMC_RSTN	3
	XGPIOA[22]	I/O	EMMC_CLK	3
	XGPIOA[23]	I/O	EMMC_CMD	3
	XGPIOA[24]	I/O	EMMC_DAT1	3
	XGPIOA[25]	I/O	EMMC_DAT0	3
	XGPIOA[26]	I/O	EMMC_DAT2	3
	XGPIOA[27]	I/O	EMMC_DAT3	3
	XGPIOA[28]	I/O	IIC0_SCL	3
	XGPIOA[29]	I/O	IIC0_SDA	3
	XGPIOA[30]	I/O	AUX0	3
	XGPIOB[0]	I/O	PWM0_BUCK	3
	XGPIOB[1]	I/O	ADC3	3

	XGPIOB[10]	I/O	MUX_SPI1_CS	3
	XGPIOB[11]	I/O	VIVO_D10	3
	XGPIOB[12]	I/O	VIVO_D9	3
	XGPIOB[13]	I/O	VIVO_D8	3
	XGPIOB[14]	I/O	VIVO_D7	3
	XGPIOB[15]	I/O	VIVO_D6	3
	XGPIOB[16]	I/O	VIVO_D5	3
	XGPIOB[17]	I/O	VIVO_D4	3
	XGPIOB[18]	I/O	VIVO_D3	3
	XGPIOB[19]	I/O	VIVO_D2	3
	XGPIOB[2]	I/O	ADC2	3
	XGPIOB[20]	I/O	VIVO_D1	3
	XGPIOB[21]	I/O	VIVO_D0	3
	XGPIOB[22]	I/O	VIVO_CLK	3
	XGPIOB[23]	I/O	GPIO_RTX___EPHY_RTX	3
	XGPIOB[24]	I/O	PAD_ETH_TXM___EPHY_RXP	3
	XGPIOB[25]	I/O	PAD_ETH_TXP___EPHY_RXN	3
	XGPIOB[26]	I/O	PAD_ETH_RXM___EPHY_TXP	3
	XGPIOB[27]	I/O	PAD_ETH_RXP___EPHY_TXN	3
	XGPIOB[3]	I/O	ADC1	3
	XGPIOB[4]	I/O	USB_ID	3
	XGPIOB[5]	I/O	USB_VBUS_EN	3

	XGPIOB[6]	I/O	USB_VBUS_DET	3
	XGPIOB[7]	I/O	MUX_SPI1_MOSI	3
	XGPIOB[8]	I/O	MUX_SPI1_MISO	3
	XGPIOB[9]	I/O	MUX_SPI1_SCK	3
	XGPIOC[0]	I/O	PAD_MIPIRX5N	3
	XGPIOC[1]	I/O	PAD_MIPIRX5P	3
	XGPIOC[2]	I/O	PAD_MIPIRX4N	3
	XGPIOC[3]	I/O	PAD_MIPIRX4P	3
	XGPIOC[4]	I/O	PAD_MIPIRX3N	3
	XGPIOC[5]	I/O	PAD_MIPIRX3P	3
	XGPIOC[6]	I/O	PAD_MIPIRX2N	3
	XGPIOC[7]	I/O	PAD_MIPIRX2P	3
	XGPIOC[8]	I/O	PAD_MIPIRX1N	3
	XGPIOC[9]	I/O	PAD_MIPIRX1P	3
	XGPIOC[10]	I/O	PAD_MIPIRX0N	3
	XGPIOC[11]	I/O	PAD_MIPIRX0P	3
	XGPIOC[12]	I/O	PAD_MIPI_TXM0	3
	XGPIOC[13]	I/O	PAD_MIPI_TXP0	3
	XGPIOC[14]	I/O	PAD_MIPI_TXM1	3
	XGPIOC[15]	I/O	PAD_MIPI_TXP1	3
	XGPIOC[16]	I/O	PAD_MIPI_TXM2	3
	XGPIOC[17]	I/O	PAD_MIPI_TXP2	3

	XGPIOC[18]	I/O	PAD_MIPI_TXM4	3
	XGPIOC[19]	I/O	PAD_MIPI_TXP4	3
	XGPIOC[20]	I/O	PAD_MIPI_TXM3	3
	XGPIOC[21]	I/O	PAD_MIPI_TXP3	3
	XGPIOC[22]	I/O	PAD_AUD_AINR_MIC	3
	XGPIOC[23]	I/O	PAD_AUD_AINL_MIC	3
	XGPIOC[24]	I/O	PAD_AUD_AOUTR	3
	XGPIOC[25]	I/O	PAD_AUD_AOUTL	3
No die domain GPIO	PWR_GPIO[0]	I/O	PWR_GPIO0	0
	PWR_GPIO[3]	I/O	PWR_SEQ1	3
	PWR_GPIO[4]	I/O	PWR_SEQ2	3
	PWR_GPIO[5]	I/O	PWR_SEQ3	3
	PWR_GPIO[6](	I/O	PWR_WAKEUP0	3
	PWR_GPIO[7]	I/O	PWR_WAKEUP1	3
	PWR_GPIO[8]	I/O	PWR_BUTTON1	3
	PWR_GPIO[9]	I/O	PWR_ON	3
	PWR_GPIO[10]	I/O	CLK32K	3
	PWR_GPIO[11]	I/O	CLK25M	3
	PWR_GPIO[12]	I/O	IIC2_SCL	3
	PWR_GPIO[13]	I/O	IIC2_SDA	3
	PWR_GPIO[14]	I/O	UART2_TX	3
	PWR_GPIO[15]	I/O	UART2_RTS	3

	PWR_GPIO[16]	I/O	UART2_RX	3
	PWR_GPIO[17]	I/O	UART2_CTS	3
	PWR_GPIO[18]	I/O	SD1_D3	3
	PWR_GPIO[19]	I/O	SD1_D2	3
	PWR_GPIO[20]	I/O	SD1_D1	3
	PWR_GPIO[21]	I/O	SD1_D0	3
	PWR_GPIO[22]	I/O	SD1_CMD	3
	PWR_GPIO[23]	I/O	SD1_CLK	3
	PWR_GPIO[24]	I/O	GPIO_ZQ___PAD_ZQ	3
Debug	DBG[0]	O	SD0_CLK	7
	DBG[0]	O	CLK32K	7
	DBG[0]	O	PAD_MIPIRX5N	7
	DBG[1]	O	SD0_CMD	7
	DBG[1]	O	CLK25M	7
	DBG[1]	O	PAD_MIPIRX5P	7
	DBG[10]	O	IIC0_SCL	7
	DBG[10]	O	PAD_MIPIRX0N	7
	DBG[11]	O	IIC0_SDA	7
	DBG[11]	O	PAD_MIPIRX0P	7
	DBG[12]	O	AUX0	7
	DBG[12]	O	PAD_MIPI_TXM0	7
	DBG[13]	O	MUX_SPI1_MOSI	7

	DBG[13]	O	PAD_MIPI_TXP0	7
	DBG[14]	O	MUX_SPI1_MISO	7
	DBG[14]	O	PAD_MIPI_TXM1	7
	DBG[15]	O	MUX_SPI1_SCK	7
	DBG[15]	O	PAD_MIPI_TXP1	7
	DBG[16]	O	MUX_SPI1_CS	7
	DBG[18]	O	PAD_MIPI_TXM4	7
	DBG[19]	O	PAD_MIPI_TXP4	7
	DBG[2]	O	SD0_D0	7
	DBG[3]	O	SD0_D1	7
	DBG[4]	O	SD0_D2	7
	DBG[5]	O	SD0_D3	7
	DBG[6]	O	UART0_TX	7
	DBG[6]	O	PAD_MIPIRX2N	7
	DBG[7]	O	UART0_RX	7
	DBG[7]	O	PAD_MIPIRX2P	7
	DBG[8]	O	PAD_MIPIRX1N	7
	DBG[9]	O	PAD_MIPIRX1P	7
Others	MUX_SPI1_CS	I/O	PAD_MIPIRX4P	7
	MUX_SPI1_MISO	I/O	PAD_MIPIRX3N	7
	MUX_SPI1_MOSI	I/O	PAD_MIPIRX3P	7
	MUX_SPI1_SCK	I/O	PAD_MIPIRX4N	7

	PKG_TYPE0	I	PKG_TYPE0	0
	PKG_TYPE1	I	PKG_TYPE1	0
	PKG_TYPE2	I	PKG_TYPE2	0

8.1.2 FMUX 寄存器描述

表格 8-2 FMUX 寄存器配置信息表

Pin Num	Pin Name	Function_select_register	fmux_default	Description
6	SD0_CLK	FMUX_GPIO_REG_IOCTRL_SD0_CLK 0x0300_101C	0x0	IO SD0_CLK function select : 0 : SDIO0_CLK (default) 1 : IIC1_SDA 2 : SPI0_SCK 3 : XGPIOA[7] 5 : PWM[15] 6 : EPHY_LNK_LED 7 : DBG[0] Others : Reserved
7	SD0_CMD	FMUX_GPIO_REG_IOCTRL_SD0_CMD 0x0300_1020	0x0	IO SD0_CMD function select : 0 : SDIO0_CMD (default) 1 : IIC1_SCL 2 : SPI0_SDO 3 : XGPIOA[8] 5 : PWM[14] 6 : EPHY_SPD_LED 7 : DBG[1] Others : Reserved
8	SD0_D0	FMUX_GPIO_REG_IOCTRL_SD0_D0 0x0300_1024	0x0	IO SD0_D0 function select : 0 : SDIO0_D[0] (default) 1 : CAM_MCLK1 2 : SPI0_SDI 3 : XGPIOA[9] 4 : UART3_TX 5 : PWM[13] 6 : WG0_D0 7 : DBG[2] Others : Reserved

10	SD0_D1	FMUX_GPIO_REG_IOCTRL_SD0_D1 0x0300_1028	0x0	IO SD0_D1 function select : 0 : SDIO0_D[1] (default) 1 : IIC1_SDA 2 : AUX0 3 : XGPIOA[10] 4 : UART1_TX 5 : PWM[12] 6 : WG0_D1 7 : DBG[3] Others : Reserved
11	SD0_D2	FMUX_GPIO_REG_IOCTRL_SD0_D2 0x0300_102C	0x0	IO SD0_D2 function select : 0 : SDIO0_D[2] (default) 1 : IIC1_SCL 2 : AUX1 3 : XGPIOA[11] 4 : UART1_RX 5 : PWM[11] 6 : WG1_D0 7 : DBG[4] Others : Reserved
12	SD0_D3	FMUX_GPIO_REG_IOCTRL_SD0_D3 0x0300_1030	0x0	IO SD0_D3 function select : 0 : SDIO0_D[3] (default) 1 : CAM_MCLK0 2 : SPI0_CS_X 3 : XGPIOA[12] 4 : UART3_RX 5 : PWM[10] 6 : WG1_D1 7 : DBG[5] Others : Reserved
14	SD0_CD	FMUX_GPIO_REG_IOCTRL_SD0_CD 0x0300_1034	0x0	IO SD0_CD function select : 0 : SDIO0_CD (default) 3 : XGPIOA[13] Others : Reserved
15	SD0_PWR _EN	FMUX_GPIO_REG_IOCTRL_SD0_PWR_E N 0x0300_1038	0x3	IO SD0_PWR_EN function select : 0 : SDIO0_PWR_EN 3 : XGPIOA[14] (default) Others : Reserved
17	SPK_EN	FMUX_GPIO_REG_IOCTRL_SPK_EN 0x0300_103C	0x3	IO SPK_EN function select : 3 : XGPIOA[15] (default) Others : Reserved

18	UART0_TX	FMUX_GPIO_REG_IOCTRL_UART0_TX 0x0300_1040	0x0	IO UART0_TX function select : 0 : UART0_TX (default) 1 : CAM_MCLK1 2 : PWM[4] 3 : XGPIOA[16] 4 : UART1_TX 5 : AUX1 7 : DBG[6] Others : Reserved
19	UART0_RX	FMUX_GPIO_REG_IOCTRL_UART0_RX 0x0300_1044	0x0	IO UART0_RX function select : 0 : UART0_RX (default) 1 : CAM_MCLK0 2 : PWM[5] 3 : XGPIOA[17] 4 : UART1_RX 5 : AUX0 7 : DBG[7] Others : Reserved
20	EMMC_DAT2	FMUX_GPIO_REG_IOCTRL_EMMC_DAT2 0x0300_104C	0x1	IO EMMC_DAT2 function select : 0 : EMMC_DAT[2] 1 : SPINOR_HOLD_X (default) 2 : SPINAND_HOLD 3 : XGPIOA[26] Others : Reserved
21	EMMC_CLK	FMUX_GPIO_REG_IOCTRL_EMMC_CLK 0x0300_1050	0x1	IO EMMC_CLK function select : 0 : EMMC_CLK 1 : SPINOR_SCK (default) 2 : SPINAND_CLK 3 : XGPIOA[22] Others : Reserved
22	EMMC_DAT0	FMUX_GPIO_REG_IOCTRL_EMMC_DAT0 0x0300_1054	0x1	IO EMMC_DAT0 function select : 0 : EMMC_DAT[0] 1 : SPINOR_MOSI (default) 2 : SPINAND_MOSI 3 : XGPIOA[25] Others : Reserved
23	EMMC_DAT3	FMUX_GPIO_REG_IOCTRL_EMMC_DAT3 0x0300_1058	0x1	IO EMMC_DAT3 function select : 0 : EMMC_DAT[3] 1 : SPINOR_WP_X (default) 2 : SPINAND_WP 3 : XGPIOA[27] Others : Reserved

24	EMMC_CMD	FMUX_GPIO_REG_IOCTRL_EMMC_CMD 0x0300_105C	0x1	IO EMMC_CMD function select : 0 : EMMC_CMD 1 : SPINOR_MISO (default) 2 : SPINAND_MISO 3 : XGPIOA[23] Others : Reserved
25	EMMC_DAT1	FMUX_GPIO_REG_IOCTRL_EMMC_DAT1 0x0300_1060	0x1	IO EMMC_DAT1 function select : 0 : EMMC_DAT[1] 1 : SPINOR_CS_X (default) 2 : SPINAND_CS 3 : XGPIOA[24] Others : Reserved
26	JTAG_CPU_TMS	FMUX_GPIO_REG_IOCTRL_JTAG_CPU_TMS 0x0300_1064	0x0	IO JTAG_CPU_TMS function select : 0 : CR_4WTMS (default) 1 : CAM_MCLK0 2 : PWM[7] 3 : XGPIOA[19] 4 : UART1_RTS 5 : AUX0 6 : UART1_TX 7 : VO_D[28] Others : Reserved
27	JTAG_CPU_TCK	FMUX_GPIO_REG_IOCTRL_JTAG_CPU_TCK 0x0300_1068	0x0	IO JTAG_CPU_TCK function select : 0 : CR_4WTCK (default) 1 : CAM_MCLK1 2 : PWM[6] 3 : XGPIOA[18] 4 : UART1_CTS 5 : AUX1 6 : UART1_RX 7 : VO_D[29] Others : Reserved
28	IIC0_SCL	FMUX_GPIO_REG_IOCTRL_IIC0_SCL 0x0300_1070	0x0	IO IIC0_SCL function select : 0 : CR_4WTDI (default) 1 : UART1_TX 2 : UART2_TX 3 : XGPIOA[28] 5 : WG0_D0 7 : DBG[10] Others : Reserved

29	IIC0_SDA	FMUX_GPIO_REG_IOCTRL_IIC0_SDA 0x0300_1074	0x0	IO IIC0_SDA function select : 0 : CR_4WTDO (default) 1 : UART1_RX 2 : UART2_RX 3 : XGPIOA[29] 5 : WG0_D1 6 : WG1_D0 7 : DBG[11] Others : Reserved
30	AUX0	FMUX_GPIO_REG_IOCTRL_AUX0 0x0300_1078	0x3	IO AUX0 function select : 0 : AUX0 3 : XGPIOA[30] (default) 4 : IIS1_MCLK 5 : VO_D[31] 6 : WG1_D1 7 : DBG[12] Others : Reserved
38	PWR_VBAT_DET	FMUX_GPIO_REG_IOCTRL_PWR_VBAT_DET 0x0300_107C	0x0	IO PWR_VBAT_DET function select : 0 : PWR_VBAT_DET (default) Others : Reserved
39	PWR_RSTN	FMUX_GPIO_REG_IOCTRL_PWR_RSTN 0x0300_1080	0x0	IO PWR_RSTN function select : 0 : PWR_RSTN (default) Others : Reserved
40	PWR_SEQ1	FMUX_GPIO_REG_IOCTRL_PWR_SEQ1 0x0300_1084	0x0	IO PWR_SEQ1 function select : 0 : PWR_SEQ1 (default) 3 : PWR_GPIO[3] Others : Reserved
41	PWR_SEQ2	FMUX_GPIO_REG_IOCTRL_PWR_SEQ2 0x0300_1088	0x0	IO PWR_SEQ2 function select : 0 : PWR_SEQ2 (default) 3 : PWR_GPIO[4] Others : Reserved
43	PWR_WAKEUP0	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090	0x0	IO PWR_WAKEUP0 function select : 0 : PWR_WAKEUP0 (default) 1 : PWR_IR0 2 : PWR_UART0_TX 3 : PWR_GPIO[6] 4 : UART1_TX 5 : IIC4_SCL 6 : EPHY_LNK_LED 7 : WG2_D0 Others : Reserved

44	PWR_BUTTON1	FMUX_GPIO_REG_IOCTRL_PWR_BUTTON1 0x0300_1098	0x0	IO PWR_BUTTON1 function select : 0 : PWR_BUTTON1 (default) 3 : PWR_GPIO[8] 4 : UART1_RX 5 : IIC4_SDA 6 : EPHY_SPD_LED 7 : WG2_D1 Others : Reserved
45	XTAL_XIN	FMUX_GPIO_REG_IOCTRL_XTAL_XIN 0x0300_10A0	0x0	IO XTAL_XIN function select : 0 : PWR_XTAL_CLKIN (default) Others : Reserved
47	PWR_GPIO0	FMUX_GPIO_REG_IOCTRL_PWR_GPIO0 0x0300_10A4	0x0	IO PWR_GPIO0 function select : 0 : PWR_GPIO[0] (default) 1 : UART2_TX 2 : PWR_UART0_RX 4 : PWM[8] Others : Reserved
48	PWR_GPIO1	FMUX_GPIO_REG_IOCTRL_PWR_GPIO1 0x0300_10A8	0x0	IO PWR_GPIO1 function select : 0 : PWR_GPIO[1] (default) 1 : UART2_RX 3 : EPHY_LNK_LED 4 : PWM[9] 5 : PWR_IIC_SCL 6 : IIC2_SCL 7 : CR_SDA0 Others : Reserved
49	PWR_GPIO2	FMUX_GPIO_REG_IOCTRL_PWR_GPIO2 0x0300_10AC	0x0	IO PWR_GPIO2 function select : 0 : PWR_GPIO[2] (default) 2 : PWR_SECTICK 3 : EPHY_SPD_LED 4 : PWM[10] 5 : PWR_IIC_SDA 6 : IIC2_SDA 7 : CR_2WTCK Others : Reserved
51	SD1_D3	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0	0x6	IO SD1_D3 function select : 0 : PWR_SD1_D3_VO32 1 : SPI2_CS_X 2 : IIC1_SCL 3 : PWR_GPIO[18] 4 : CAM_MCLK0 5 : UART3_CTS 6 : PWR_SPINOR1_CS_X (default) 7 : PWM[4] Others : Reserved

52	SD1_D2	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4	0x6	IO SD1_D2 function select : 0 : PWR_SD1_D2_VO33 1 : IIC1_SCL 2 : UART2_TX 3 : PWR_GPIO[19] 4 : CAM_MCLK0 5 : UART3_TX 6 : PWR_SPINOR1_HOLD_X (default) 7 : PWM[5] Others : Reserved
53	SD1_D1	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8	0x6	IO SD1_D1 function select : 0 : PWR_SD1_D1_VO34 1 : IIC1_SDA 2 : UART2_RX 3 : PWR_GPIO[20] 4 : CAM_MCLK1 5 : UART3_RX 6 : PWR_SPINOR1_WP_X (default) 7 : PWM[6] Others : Reserved
54	SD1_D0	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC	0x6	IO SD1_D0 function select : 0 : PWR_SD1_D0_VO35 1 : SPI2_SDI 2 : IIC1_SDA 3 : PWR_GPIO[21] 4 : CAM_MCLK1 5 : UART3_RTS 6 : PWR_SPINOR1_MISO (default) 7 : PWM[7] Others : Reserved
55	SD1_CMD	FMUX_GPIO_REG_IOCTRL_SD1_CMD 0x0300_10E0	0x6	IO SD1_CMD function select : 0 : PWR_SD1_CMD_VO36 1 : SPI2_SDO 2 : IIC3_SCL 3 : PWR_GPIO[22] 4 : CAM_VS0 5 : EPHY_LNK_LED 6 : PWR_SPINOR1_MOSI (default) 7 : PWM[8] Others : Reserved

56	SD1_CLK	FMUX_GPIO_REG_IOCTRL_SD1_CLK 0x0300_10E4	0x6	IO SD1_CLK function select : 0 : PWR_SD1_CLK_VO37 1 : SPI2_SCK 2 : IIC3_SDA 3 : PWR_GPIO[23] 4 : CAM_HS0 5 : EPHY_SPD_LED 6 : PWR_SPINOR1_SCK (default) 7 : PWM[9] Others : Reserved
58	PWM0_BUCK	FMUX_GPIO_REG_IOCTRL_PWM0_BUCK 0x0300_10EC	0x3	IO PWM0_BUCK function select : 0 : PWM[0] 3 : XGPIOB[0] (default) Others : Reserved
59	ADC1	FMUX_GPIO_REG_IOCTRL_ADC1 0x0300_10F8	0x3	IO ADC1 function select : 3 : XGPIOB[3] (default) 4 : KEY_COL2 Others : Reserved
60	USB_VBUS_DET	FMUX_GPIO_REG_IOCTRL_USB_VBUS_DET 0x0300_1108	0x0	IO USB_VBUS_DET function select : 0 : USB_VBUS_DET (default) 3 : XGPIOB[6] 4 : CAM_MCLK0 5 : CAM_MCLK1 Others : Reserved
62	PAD_ETH_TXP___EPHY_RXN	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXP 0x0300_1124	0x3	IO PAD_ETH_TXP function select : 1 : UART3_RX 2 : IIC1_SCL 3 : XGPIOB[25] (default) 4 : PWM[13] 5 : CAM_MCLK0 6 : SPI1_SDO 7 : IIS2_LRCK Others : Reserved
63	PAD_ETH_TXM___EPHY_RXP	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXM 0x0300_1128	0x3	IO PAD_ETH_TXM function select : 1 : UART3_RTS 2 : IIC1_SDA 3 : XGPIOB[24] (default) 4 : PWM[12] 5 : CAM_MCLK1 6 : SPI1_SDI 7 : IIS2_BCLK Others : Reserved

64	PAD_ETH_RXP___EPHY_TXN	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXP 0x0300_112C	0x3	IO PAD_ETH_RXP function select : 1 : UART3_TX 2 : CAM_MCLK1 3 : XGPIOB[27] (default) 4 : PWM[15] 5 : CAM_HS0 6 : SPI1_SCK 7 : IIS2_DO Others : Reserved
65	PAD_ETH_RXM___EPHY_TXP	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXM 0x0300_1130	0x3	IO PAD_ETH_RXM function select : 1 : UART3_CTS 2 : CAM_MCLK0 3 : XGPIOB[26] (default) 4 : PWM[14] 5 : CAM_VS0 6 : SPI1_CS_X 7 : IIS2_DI Others : Reserved
72	PAD_MIPIRX4N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C	0x0	IO PAD_MIPIRX4N function select : 0 : CR_SCL0 (default) 1 : VI0_CLK 2 : VI1_D[13] 3 : XGPIOC[2] 4 : IIC1_SDA 5 : CAM_MCLK0 6 : KEY_ROW0 7 : MUX_SPI1_SCK Others : Reserved
73	PAD_MIPIRX4P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170	0x0	IO PAD_MIPIRX4P function select : 0 : CR_SDA0 (default) 1 : VI0_D[0] 2 : VI1_D[14] 3 : XGPIOC[3] 4 : IIC1_SCL 5 : CAM_MCLK1 6 : KEY_ROW1 7 : MUX_SPI1_CS Others : Reserved
74	PAD_MIPIRX3N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174	0x3	IO PAD_MIPIRX3N function select : 0 : CR_2WTMS 1 : VI0_D[1] 2 : VI1_D[15] 3 : XGPIOC[4] (default) 4 : CAM_MCLK0 7 : MUX_SPI1_MISO Others : Reserved

75	PAD_MIPIRX3P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178	0x3	IO PAD_MIPIRX3P function select : 0 : CR_2WTCK 1 : VI0_D[2] 2 : VI1_D[16] 3 : XGPIOC[5] (default) 7 : MUX_SPI1_MOSI Others : Reserved
76	PAD_MIPIRX2N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2N 0x0300_117C	0x3	IO PAD_MIPIRX2N function select : 1 : VI0_D[3] 2 : VO_D[10] 3 : XGPIOC[6] (default) 4 : VI1_D[17] 5 : IIC4_SCL 7 : DBG[6] Others : Reserved
77	PAD_MIPIRX2P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2P 0x0300_1180	0x3	IO PAD_MIPIRX2P function select : 1 : VI0_D[4] 2 : VO_D[9] 3 : XGPIOC[7] (default) 4 : VI1_D[18] 5 : IIC4_SDA 7 : DBG[7] Others : Reserved
78	PAD_MIPIRX1N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1N 0x0300_1184	0x3	IO PAD_MIPIRX1N function select : 1 : VI0_D[5] 2 : VO_D[8] 3 : XGPIOC[8] (default) 6 : KEY_ROW3 7 : DBG[8] Others : Reserved
79	PAD_MIPIRX1P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1P 0x0300_1188	0x3	IO PAD_MIPIRX1P function select : 1 : VI0_D[6] 2 : VO_D[7] 3 : XGPIOC[9] (default) 4 : IIC1_SDA 6 : KEY_ROW2 7 : DBG[9] Others : Reserved
80	PAD_MIPIRX0N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0N 0x0300_118C	0x3	IO PAD_MIPIRX0N function select : 1 : VI0_D[7] 2 : VO_D[6] 3 : XGPIOC[10] (default) 4 : IIC1_SCL 5 : CAM_MCLK1 7 : DBG[10] Others : Reserved

81	PAD_MIPIRX0P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0P 0x0300_1190	0x3	IO PAD_MIPIRX0P function select : 1 : VI0_D[8] 2 : VO_D[5] 3 : XGPIOC[11] (default) 4 : CAM_MCLK0 7 : DBG[11] Others : Reserved
83	PAD_MIPI_TXM2	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4	0x3	IO PAD_MIPI_TXM2 function select : 0 : CR_SDA0 1 : VI0_D[13] 2 : VO_D[0] 3 : XGPIOC[16] (default) 4 : IIC1_SDA 5 : PWM[8] 6 : SPI0_SCK 7 : SD1_D2 Others : Reserved
84	PAD_MIPI_TXP2	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8	0x3	IO PAD_MIPI_TXP2 function select : 0 : CR_SCL0 1 : VI0_D[14] 2 : VO_CLK0 3 : XGPIOC[17] (default) 4 : IIC1_SCL 5 : PWM[9] 6 : SPI0_CS_X 7 : SD1_D3 Others : Reserved
85	PAD_MIPI_TXM1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC	0x3	IO PAD_MIPI_TXM1 function select : 0 : CR_2WTMS 1 : VI0_D[11] 2 : VO_D[2] 3 : XGPIOC[14] (default) 4 : IIC2_SDA 5 : PWM[10] 6 : SPI0_SDO 7 : DBG[14] Others : Reserved
86	PAD_MIPI_TXP1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0	0x3	IO PAD_MIPI_TXP1 function select : 0 : CR_2WTCK 1 : VI0_D[12] 2 : VO_D[1] 3 : XGPIOC[15] (default) 4 : IIC2_SCL 5 : PWM[11] 6 : SPI0_SDI 7 : DBG[15] Others : Reserved

87	PAD_MIPI_TXM0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4	0x3	IO PAD_MIPI_TXM0 function select : 1 : VI0_D[9] 2 : VO_D[4] 3 : XGPIOC[12] (default) 4 : CAM_MCLK1 5 : PWM[14] 6 : CAM_VS0 7 : DBG[12] Others : Reserved
88	PAD_MIPI_TXP0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8	0x3	IO PAD_MIPI_TXP0 function select : 1 : VI0_D[10] 2 : VO_D[3] 3 : XGPIOC[13] (default) 4 : CAM_MCLK0 5 : PWM[15] 6 : CAM_HS0 7 : DBG[13] Others : Reserved
2	PAD_AUD_AINL_MIC	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AINL_MIC 0x0300_11BC	0x3	IO PAD_AUD_AINL_MIC function select : 3 : XGPIOC[23] (default) 4 : IIS1_BCLK 5 : IIS2_BCLK Others : Reserved
4	PAD_AUD_AOUTR	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AOUTR 0x0300_11C8	0x3	IO PAD_AUD_AOUTR function select : 3 : XGPIOC[24] (default) 4 : IIS1_DI 5 : IIS2_DO 6 : IIS1_DO Others : Reserved
67	GPIO_RTX___EPHY_RTX	FMUX_GPIO_REG_IOCTRL_GPIO_RTX 0x0300_11CC	0x3	IO GPIO_RTX function select : 3 : XGPIOB[23] (default) 4 : PWM[1] 5 : CAM_MCLK0 Others : Reserved
35	GPIO_ZQ___PAD_ZQ	FMUX_GPIO_REG_IOCTRL_GPIO_ZQ 0x0300_11D0	0x3	IO GPIO_ZQ function select : 3 : PWR_GPIO[24] (default) 4 : PWM[2] Others : Reserved
#N/A	PKG_TYPE0	FMUX_GPIO_REG_IOCTRL_PKG_TYPE0 0x0300_1104	0x0	IO PKG_TYPE0 function select : 0 : PKG_TYPE0 (default) Others : Reserved
#N/A	PKG_TYPE1	FMUX_GPIO_REG_IOCTRL_PKG_TYPE1 0x0300_110C	0x0	IO PKG_TYPE1 function select : 0 : PKG_TYPE1 (default) Others : Reserved
#N/A	PKG_TYPE2	FMUX_GPIO_REG_IOCTRL_PKG_TYPE2 0x0300_1110	0x0	IO PKG_TYPE2 function select : 0 : PKG_TYPE2 (default) Others : Reserved

#N/A	MUX_SPI1_MISO	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MISO 0x0300_1114	0x3	IO MUX_SPI1_MISO function select : 1 : UART3_RTS 2 : IIC1_SDA 3 : XGPIOB[8] (default) 4 : PWM[9] 5 : KEY_COL1 6 : SPI1_SDI 7 : DBG[14] Others : Reserved
#N/A	MUX_SPI1_MOSI	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MOSI 0x0300_1118	0x3	IO MUX_SPI1_MOSI function select : 1 : UART3_RX 2 : IIC1_SCL 3 : XGPIOB[7] (default) 4 : PWM[8] 5 : KEY_COL0 6 : SPI1_SDO 7 : DBG[13] Others : Reserved
#N/A	MUX_SPI1_CS	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_CS 0x0300_111C	0x3	IO MUX_SPI1_CS function select : 1 : UART3_CTS 2 : CAM_MCLK0 3 : XGPIOB[10] (default) 4 : PWM[11] 5 : KEY_ROW3 6 : SPI1_CS_X 7 : DBG[16] Others : Reserved
#N/A	MUX_SPI1_SCK	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_SCK 0x0300_1120	0x3	IO MUX_SPI1_SCK function select : 1 : UART3_TX 2 : CAM_MCLK1 3 : XGPIOB[9] (default) 4 : PWM[10] 5 : KEY_ROW2 6 : SPI1_SCK 7 : DBG[15] Others : Reserved

8.2 管脚控制 PINCTRL

8.2.1 引脚控制寄存器概览

表格 8-3 引脚和引脚控制寄存器对应信息表

Pin Num	Pin Name	IO_cfg_register	Address
6	SD0_CLK	IOBLK_G10_REG_SD0_CLK	0x0300_1A00
7	SD0_CMD	IOBLK_G10_REG_SD0_CMD	0x0300_1A04
8	SD0_D0	IOBLK_G10_REG_SD0_D0	0x0300_1A08
10	SD0_D1	IOBLK_G10_REG_SD0_D1	0x0300_1A0C
11	SD0_D2	IOBLK_G10_REG_SD0_D2	0x0300_1A10
12	SD0_D3	IOBLK_G10_REG_SD0_D3	0x0300_1A14
14	SD0_CD	IOBLK_G7_REG_SD0_CD	0x0300_1900
15	SD0_PWR_EN	IOBLK_G7_REG_SD0_PWR_EN	0x0300_1904
17	SPK_EN	IOBLK_G7_REG_SPK_EN	0x0300_1908
18	UART0_TX	IOBLK_G7_REG_UART0_TX	0x0300_190C
19	UART0_RX	IOBLK_G7_REG_UART0_RX	0x0300_1910
20	EMMC_DAT2	IOBLK_G7_REG_EMMC_DAT2	0x0300_1918
21	EMMC_CLK	IOBLK_G7_REG_EMMC_CLK	0x0300_191C
22	EMMC_DAT0	IOBLK_G7_REG_EMMC_DAT0	0x0300_1920
23	EMMC_DAT3	IOBLK_G7_REG_EMMC_DAT3	0x0300_1924
24	EMMC_CMD	IOBLK_G7_REG_EMMC_CMD	0x0300_1928
25	EMMC_DAT1	IOBLK_G7_REG_EMMC_DAT1	0x0300_192C
26	JTAG_CPU_TMS	IOBLK_G7_REG_JTAG_CPU_TMS	0x0300_1930
27	JTAG_CPU_TCK	IOBLK_G7_REG_JTAG_CPU_TCK	0x0300_1934
28	IIC0_SCL	IOBLK_G7_REG_IIC0_SCL	0x0300_193C
29	IIC0_SDA	IOBLK_G7_REG_IIC0_SDA	0x0300_1940
30	AUX0	IOBLK_G7_REG_AUX0	0x0300_1944
38	PWR_VBAT_DET	IOBLK_GRTC_REG_PWR_VBAT_DET	0x0502_7000
39	PWR_RSTN	IOBLK_GRTC_REG_PWR_RSTN	0x0502_7004
40	PWR_SEQ1	IOBLK_GRTC_REG_PWR_SEQ1	0x0502_7008
41	PWR_SEQ2	IOBLK_GRTC_REG_PWR_SEQ2	0x0502_700C
43	PWR_WAKEUP0	IOBLK_GRTC_REG_PWR_WAKEUP0	0x0502_7018
44	PWR_BUTTON1	IOBLK_GRTC_REG_PWR_BUTTON1	0x0502_7020
45	XTAL_XIN	IOBLK_GRTC_REG_XTAL_XIN	0x0502_7028
47	PWR_GPIO0	IOBLK_GRTC_REG_PWR_GPIO0	0x0502_702C
48	PWR_GPIO1	IOBLK_GRTC_REG_PWR_GPIO1	0x0502_7030
49	PWR_GPIO2	IOBLK_GRTC_REG_PWR_GPIO2	0x0502_7034

51	SD1_D3	IOBLK_GRTC_REG_SD1_D3	0x0502_7058
52	SD1_D2	IOBLK_GRTC_REG_SD1_D2	0x0502_705C
53	SD1_D1	IOBLK_GRTC_REG_SD1_D1	0x0502_7060
54	SD1_D0	IOBLK_GRTC_REG_SD1_D0	0x0502_7064
55	SD1_CMD	IOBLK_GRTC_REG_SD1_CMD	0x0502_7068
56	SD1_CLK	IOBLK_GRTC_REG_SD1_CLK	0x0502_706C
58	PWM0_BUCK	IOBLK_G1_REG_PWM0_BUCK	0x0300_1804
59	ADC1	IOBLK_G1_REG_ADC1	0x0300_1810
60	USB_VBUS_DET	IOBLK_G1_REG_USB_VBUS_DET	0x0300_1820
62	PAD_ETH_TXP___EPHY_RXN	#N/A	#N/A
63	PAD_ETH_TXM___EPHY_RXP	#N/A	#N/A
64	PAD_ETH_RXP___EPHY_TXN	#N/A	#N/A
65	PAD_ETH_RXM___EPHY_TXP	#N/A	#N/A
72	PAD_MIPIRX4N	IOBLK_G12_REG_PAD_MIPIRX4N	0x0300_1C38
73	PAD_MIPIRX4P	IOBLK_G12_REG_PAD_MIPIRX4P	0x0300_1C3C
74	PAD_MIPIRX3N	IOBLK_G12_REG_PAD_MIPIRX3N	0x0300_1C40
75	PAD_MIPIRX3P	IOBLK_G12_REG_PAD_MIPIRX3P	0x0300_1C44
76	PAD_MIPIRX2N	IOBLK_G12_REG_PAD_MIPIRX2N	0x0300_1C48
77	PAD_MIPIRX2P	IOBLK_G12_REG_PAD_MIPIRX2P	0x0300_1C4C
78	PAD_MIPIRX1N	IOBLK_G12_REG_PAD_MIPIRX1N	0x0300_1C50
79	PAD_MIPIRX1P	IOBLK_G12_REG_PAD_MIPIRX1P	0x0300_1C54
80	PAD_MIPIRX0N	IOBLK_G12_REG_PAD_MIPIRX0N	0x0300_1C58
81	PAD_MIPIRX0P	IOBLK_G12_REG_PAD_MIPIRX0P	0x0300_1C5C
83	PAD_MIPI_TXM2	IOBLK_G12_REG_PAD_MIPI_TXM2	0x0300_1C70
84	PAD_MIPI_TXP2	IOBLK_G12_REG_PAD_MIPI_TXP2	0x0300_1C74
85	PAD_MIPI_TXM1	IOBLK_G12_REG_PAD_MIPI_TXM1	0x0300_1C78
86	PAD_MIPI_TXP1	IOBLK_G12_REG_PAD_MIPI_TXP1	0x0300_1C7C
87	PAD_MIPI_TXM0	IOBLK_G12_REG_PAD_MIPI_TXM0	0x0300_1C80
88	PAD_MIPI_TXP0	IOBLK_G12_REG_PAD_MIPI_TXP0	0x0300_1C84
2	PAD_AUD_AINL_MIC	#N/A	#N/A
4	PAD_AUD_AOUTR	#N/A	#N/A
67	GPIO_RTX___EPHY_RTX	IOBLK_G12_REG_GPIO_RTX	0x0300_1C8C
35	GPIO_ZQ___PAD_ZQ	IOBLK_GRTC_REG_GPIO_ZQ	0x0502_70E0
#N/A	PKG_TYPE0	IOBLK_G1_REG_PKG_TYPE0	0x0300_181C
#N/A	PKG_TYPE1	IOBLK_G1_REG_PKG_TYPE1	0x0300_1824
#N/A	PKG_TYPE2	IOBLK_G1_REG_PKG_TYPE2	0x0300_1828
#N/A	MUX_SPI1_MISO	#N/A	#N/A
#N/A	MUX_SPI1_MOSI	#N/A	#N/A
#N/A	MUX_SPI1_CS	#N/A	#N/A
#N/A	MUX_SPI1_SCK	#N/A	#N/A

8.2.2 PINCTRL 寄存器描述

表格 8-4 引脚控制寄存器配置表

Register Name	Field Name	Bit	Default	Field Description
IOBLK_G1_REG_PWM0_BUCK	IOBLK_G1_REG_PWM0_BUCK_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PWM0_BUCK_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PWM0_BUCK_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G1_REG_PWM0_BUCK_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_PWM0_BUCK_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_PWM0_BUCK_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_PWM0_BUCK_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_PWM0_BUCK_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G1_REG_ADC1	IOBLK_G1_REG_ADC1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_ADC1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_ADC1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G1_REG_ADC1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_ADC1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_ADC1_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_ADC1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_ADC1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G1_REG_PKG_TYPE0	IOBLK_G1_REG_PKG_TYPE0_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE0_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE0_DS0	5	0x0	输出驱动能力档位 bit 0

	IOBLK_G1_REG_PKG_TYPE0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_PKG_TYPE0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_PKG_TYPE0_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_PKG_TYPE0_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_PKG_TYPE0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G1_REG_USB_VBUS_DET	IOBLK_G1_REG_USB_VBUS_DET_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_USB_VBUS_DET_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_USB_VBUS_DET_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G1_REG_USB_VBUS_DET_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_USB_VBUS_DET_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_USB_VBUS_DET_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_USB_VBUS_DET_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_USB_VBUS_DET_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G1_REG_PKG_TYPE1	IOBLK_G1_REG_PKG_TYPE1_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE1_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G1_REG_PKG_TYPE1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_PKG_TYPE1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_PKG_TYPE1_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_PKG_TYPE1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_PKG_TYPE1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)

IOBLK_G1_REG_PKG_TYPE2	IOBLK_G1_REG_PKG_TYPE2_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE2_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G1_REG_PKG_TYPE2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G1_REG_PKG_TYPE2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G1_REG_PKG_TYPE2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G1_REG_PKG_TYPE2_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G1_REG_PKG_TYPE2_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G1_REG_PKG_TYPE2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_SD0_CD	IOBLK_G7_REG_SD0_CD_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SD0_CD_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SD0_CD_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_SD0_CD_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_SD0_CD_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_SD0_CD_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_SD0_CD_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_SD0_PWR_EN	IOBLK_G7_REG_SD0_PWR_EN_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SD0_PWR_EN_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SD0_PWR_EN_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_SD0_PWR_EN_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_SD0_PWR_EN_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_SD0_PWR_EN_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_SD0_PWR_EN_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)

IOBLK_G7_REG_SPK_EN	IOBLK_G7_REG_SPK_EN_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SPK_EN_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_SPK_EN_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_SPK_EN_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_SPK_EN_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_SPK_EN_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_SPK_EN_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_UART0_TX	IOBLK_G7_REG_UART0_TX_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_UART0_TX_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_UART0_TX_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_UART0_TX_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_UART0_TX_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_UART0_TX_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_UART0_TX_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_UART0_RX	IOBLK_G7_REG_UART0_RX_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_UART0_RX_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_UART0_RX_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_UART0_RX_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_UART0_RX_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_UART0_RX_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_UART0_RX_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_DAT2	IOBLK_G7_REG_EMMC_DAT2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效

	IOBLK_G7_REG_EMMC_DAT2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_DAT2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_EMMC_DAT2_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_DAT2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_DAT2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_CLK	IOBLK_G7_REG_EMMC_CLK_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_CLK_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_CLK_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_CLK_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_EMMC_CLK_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_CLK_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_CLK_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_DAT0	IOBLK_G7_REG_EMMC_DAT0_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT0_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_DAT0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_EMMC_DAT0_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_DAT0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_DAT0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_DAT3	IOBLK_G7_REG_EMMC_DAT3_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT3_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT3_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_DAT3_DS1	6	0x1	输出驱动能力档位 bit 1

	IOBLK_G7_REG_EMMC_DAT3_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_DAT3_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_DAT3_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_CMD	IOBLK_G7_REG_EMMC_CMD_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_CMD_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_CMD_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_CMD_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_EMMC_CMD_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_CMD_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_CMD_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_EMMC_DAT1	IOBLK_G7_REG_EMMC_DAT1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_EMMC_DAT1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_EMMC_DAT1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_EMMC_DAT1_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_EMMC_DAT1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_EMMC_DAT1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_JTAG_CPU_TMS	IOBLK_G7_REG_JTAG_CPU_TMS_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_JTAG_CPU_TMS_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_JTAG_CPU_TMS_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_JTAG_CPU_TMS_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_JTAG_CPU_TMS_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_JTAG_CPU_TMS_ST0	8	0x0	输入施密特触发器强度控制. bit 0

	IOBLK_G7_REG_JTAG_CPU_TMS_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_JTAG_CPU_TCK	IOBLK_G7_REG_JTAG_CPU_TCK_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_JTAG_CPU_TCK_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_JTAG_CPU_TCK_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_JTAG_CPU_TCK_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_JTAG_CPU_TCK_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_JTAG_CPU_TCK_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_JTAG_CPU_TCK_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_IIC0_SCL	IOBLK_G7_REG_IIC0_SCL_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_IIC0_SCL_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_IIC0_SCL_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_IIC0_SCL_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_IIC0_SCL_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_IIC0_SCL_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_IIC0_SCL_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G7_REG_IIC0_SDA	IOBLK_G7_REG_IIC0_SDA_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_IIC0_SDA_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_IIC0_SDA_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_IIC0_SDA_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_IIC0_SDA_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_IIC0_SDA_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_IIC0_SDA_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)

IOBLK_G7_REG_AUX0	IOBLK_G7_REG_AUX0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_AUX0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G7_REG_AUX0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G7_REG_AUX0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G7_REG_AUX0_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G7_REG_AUX0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G7_REG_AUX0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_CLK	IOBLK_G10_REG_SD0_CLK_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_CLK_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_CLK_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_CLK_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G10_REG_SD0_CLK_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_CLK_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_CLK_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_CMD	IOBLK_G10_REG_SD0_CMD_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_CMD_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_CMD_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_CMD_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G10_REG_SD0_CMD_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_CMD_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_CMD_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_D0	IOBLK_G10_REG_SD0_D0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效

	IOBLK_G10_REG_SD0_D0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_D0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G10_REG_SD0_D0_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_D0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_D0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_D1	IOBLK_G10_REG_SD0_D1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_D1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G10_REG_SD0_D1_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_D1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_D1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_D2	IOBLK_G10_REG_SD0_D2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_D2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G10_REG_SD0_D2_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_D2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_D2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G10_REG_SD0_D3	IOBLK_G10_REG_SD0_D3_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D3_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G10_REG_SD0_D3_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G10_REG_SD0_D3_DS1	6	0x1	输出驱动能力档位 bit 1

	IOBLK_G10_REG_SD0_D3_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_G10_REG_SD0_D3_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G10_REG_SD0_D3_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX4N	IOBLK_G12_REG_PAD_MIPIRX4N_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4N_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4N_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX4N_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX4N_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX4N_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX4N_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4N_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX4P	IOBLK_G12_REG_PAD_MIPIRX4P_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4P_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4P_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX4P_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX4P_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX4P_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX4P_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX4P_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX3N	IOBLK_G12_REG_PAD_MIPIRX3N_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX3N_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效

	IOBLK_G12_REG_PAD_MIPIRX3N_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX3N_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX3N_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX3N_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX3N_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX3N_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX3P	IOBLK_G12_REG_PAD_MIPIRX3P_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX3P_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX3P_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX3P_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX3P_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX3P_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX3P_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX3P_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX2N	IOBLK_G12_REG_PAD_MIPIRX2N_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX2N_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX2N_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX2N_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX2N_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX2N_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX2N_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效

	IOBLK_G12_REG_PAD_MIPIRX2N_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX2P	IOBLK_G12_REG_PAD_MIPIRX2P_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX2P_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX2P_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX2P_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX2P_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX2P_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX2P_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX2P_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX1N	IOBLK_G12_REG_PAD_MIPIRX1N_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1N_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1N_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX1N_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX1N_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX1N_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX1N_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1N_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX1P	IOBLK_G12_REG_PAD_MIPIRX1P_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1P_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1P_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX1P_DS1	6	0x1	输出驱动能力档位 bit 1

	IOBLK_G12_REG_PAD_MIPIRX1P_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX1P_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX1P_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX1P_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX0N	IOBLK_G12_REG_PAD_MIPIRX0N_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0N_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0N_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX0N_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX0N_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX0N_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX0N_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0N_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPIRX0P	IOBLK_G12_REG_PAD_MIPIRX0P_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0P_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0P_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPIRX0P_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPIRX0P_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPIRX0P_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPIRX0P_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPIRX0P_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXM2	IOBLK_G12_REG_PAD_MIPI_TXM2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效

	IOBLK_G12_REG_PAD_MIPI_TXM2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXM2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM2_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPI_TXM2_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXP2	IOBLK_G12_REG_PAD_MIPI_TXP2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPI_TXP2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXP2_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP2_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXM1	IOBLK_G12_REG_PAD_MIPI_TXM1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXM1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM1_ST1	9	0x0	输入施密特触发器强度控制. bit 1

	IOBLK_G12_REG_PAD_MIPI_TXM1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXP1	IOBLK_G12_REG_PAD_MIPI_TXP1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPI_TXP1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXP1_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXM0	IOBLK_G12_REG_PAD_MIPI_TXM0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXM0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXM0_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPI_TXM0_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXM0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_PAD_MIPI_TXP0	IOBLK_G12_REG_PAD_MIPI_TXP0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP0_DS0	5	0x0	输出驱动能力档位 bit 0

	IOBLK_G12_REG_PAD_MIPI_TXP0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_PAD_MIPI_TXP0_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_PAD_MIPI_TXP0_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_PAD_MIPI_TXP0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_G12_REG_GPIO_RTX	IOBLK_G12_REG_GPIO_RTX_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_GPIO_RTX_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_G12_REG_GPIO_RTX_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_G12_REG_GPIO_RTX_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_G12_REG_GPIO_RTX_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_G12_REG_GPIO_RTX_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_G12_REG_GPIO_RTX_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_G12_REG_GPIO_RTX_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_VBAT_DET	IOBLK_GRTC_REG_PWR_VBAT_DET_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_VBAT_DET_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_VBAT_DET_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_VBAT_DET_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_VBAT_DET_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_VBAT_DET_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_VBAT_DET_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_VBAT_DET_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)

IOBLK_GRTC_REG_PWR_RSTN	IOBLK_GRTC_REG_PWR_RSTN_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_RSTN_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_RSTN_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_RSTN_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_RSTN_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_RSTN_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_RSTN_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_RSTN_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_SEQ1	IOBLK_GRTC_REG_PWR_SEQ1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_SEQ1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_SEQ1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_SEQ1_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_SEQ1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_SEQ2	IOBLK_GRTC_REG_PWR_SEQ2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_SEQ2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_SEQ2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_SEQ2_ST1	9	0x0	输入施密特触发器强度控制. bit 1

	IOBLK_GRTC_REG_PWR_SEQ2_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_SEQ2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PTEST	IOBLK_GRTC_REG_PTEST_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PTEST_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PTEST_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PTEST_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PTEST_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PTEST_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PTEST_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PTEST_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_WAKEUP0	IOBLK_GRTC_REG_PWR_WAKEUP0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_WAKEUP0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_WAKEUP0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_WAKEUP0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_WAKEUP0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_WAKEUP0_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_WAKEUP0_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_WAKEUP0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_BUTTON1	IOBLK_GRTC_REG_PWR_BUTTON1_PU	2	0x1	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_BUTTON1_PD	3	0x0	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_BUTTON1_DS0	5	0x0	输出驱动能力档位 bit 0

	IOBLK_GRTC_REG_PWR_BUTTON1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_BUTTON1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_BUTTON1_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_BUTTON1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_BUTTON1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_XTAL_XIN	IOBLK_GRTC_REG_XTAL_XIN_XDS0	14	0x0	晶振输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_XTAL_XIN_XDS1	15	0x0	晶振输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_XTAL_XIN_XDS2	16	0x1	晶振输出驱动能力档位 bit 2
IOBLK_GRTC_REG_PWR_GPIO0	IOBLK_GRTC_REG_PWR_GPIO0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_GPIO0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_GPIO0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_GPIO0_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_GPIO0_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_GPIO1	IOBLK_GRTC_REG_PWR_GPIO1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_GPIO1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_GPIO1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_GPIO1_ST1	9	0x0	输入施密特触发器强度控制. bit 1

	IOBLK_GRTC_REG_PWR_GPIO1_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_PWR_GPIO2	IOBLK_GRTC_REG_PWR_GPIO2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_PWR_GPIO2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_PWR_GPIO2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_PWR_GPIO2_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_PWR_GPIO2_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_PWR_GPIO2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_D3	IOBLK_GRTC_REG_SD1_D3_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D3_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D3_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_D3_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_D3_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_GRTC_REG_SD1_D3_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_SD1_D3_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_D2	IOBLK_GRTC_REG_SD1_D2_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D2_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D2_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_D2_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_D2_DS2	7	0x0	输出驱动能力档位 bit 2

	IOBLK_GRTC_REG_SD1_D2_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_SD1_D2_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_D1	IOBLK_GRTC_REG_SD1_D1_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D1_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D1_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_D1_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_D1_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_GRTC_REG_SD1_D1_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_SD1_D1_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_D0	IOBLK_GRTC_REG_SD1_D0_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D0_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_D0_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_D0_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_D0_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_GRTC_REG_SD1_D0_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_SD1_D0_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_CMD	IOBLK_GRTC_REG_SD1_CMD_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_CMD_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_CMD_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_CMD_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_CMD_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_GRTC_REG_SD1_CMD_ST0	8	0x0	输入施密特触发器强度控制. bit 0

	IOBLK_GRTC_REG_SD1_CMD_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_SD1_CLK	IOBLK_GRTC_REG_SD1_CLK_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_CLK_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_SD1_CLK_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_SD1_CLK_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_SD1_CLK_DS2	7	0x0	输出驱动能力档位 bit 2
	IOBLK_GRTC_REG_SD1_CLK_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_SD1_CLK_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)
IOBLK_GRTC_REG_GPIO_ZQ	IOBLK_GRTC_REG_GPIO_ZQ_PU	2	0x0	上拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_GPIO_ZQ_PD	3	0x1	下拉电阻使能. 0=无效; 1=有效
	IOBLK_GRTC_REG_GPIO_ZQ_DS0	5	0x0	输出驱动能力档位 bit 0
	IOBLK_GRTC_REG_GPIO_ZQ_DS1	6	0x1	输出驱动能力档位 bit 1
	IOBLK_GRTC_REG_GPIO_ZQ_ST0	8	0x0	输入施密特触发器强度控制. bit 0
	IOBLK_GRTC_REG_GPIO_ZQ_ST1	9	0x0	输入施密特触发器强度控制. bit 1
	IOBLK_GRTC_REG_GPIO_ZQ_HE	10	0x0	弱电平维持器(Bus holder) 使能 . 0=无效 ; 1=有效
	IOBLK_GRTC_REG_GPIO_ZQ_SL	11	0x0	输出电平转换速率限制. 0=无效(较快) ; 1=有效(较慢)

9 DMA 控制器

9.1 概述

DMA(Direct Memory Access)不需 CPU 逐笔数据干涉， 存储器与设备间可直接数据传输， 此机制大大减少 CPU 控制时间， 并提高数据传输的速率， 非常适合使用在大量数据传输。 芯片工作时往往同时需要多通道的数据传输，一个通道需要一个 DMA 硬件支持， DMAC (DMA 控制器) 负责了多通道的控制。 图表 9-1 为 DMAC 硬件控制流程图， 来源与目的设备可在不同的 AXI 汇流。

图表 9-1 DMAC 硬件控制流程示意图

9.2 特点

DMAC 特点如下:

a. 最大可同时建立 8 个 DMA 通道。
b. 传输的数据来源与数据目的， 皆可设置为内存或设备。
c. 只允许单向传输配置。
d. 提供 DMA 传输暂停、恢复、取消。
e. 支持 DMA Burst 长度配置。
f. 提供 DMA 通道优先等级配置。
g. 当通道数据是在设备间传输时，流量控制可交由设备控制。
h. 支持硬件链表功能。
i. 支持通道上锁，在上锁通道未完成前的其他通道请求会被忽略。

9.3 功能描述

9.3.1 外设请求线

DMA 内建 8 组 DMA 通道，每个通道之外设请求，都需要配置以映射到外设。请参照 7.2.2 说明，在 DMA 通道使能前进行配置。

9.3.2 访问空间

表格 9-1 DMAC 访问空间类型

空间类型	描述
Memory	SRAM
	非安全 DDR 空间
外设	UART0~UART3
	I2C0~I2C4
	SPI0~SPI3
	SPI_NAND
	I2S0~I2S3

9.3.3 基本传输

DMA 数据传输是以块状来设定， 由突发传输来完成， 突发传输的长度是可以设定的，但经常发生的是块状数据量并不完美的是突发传输长度的整数倍数， 传输的最后一个交易会出现长度小于设定的突发传输长度，此时会需要使用单笔传输的请求来完成。

最大支持的 8 个 DMA 通道的来源与目的可以有以下四种组合：

a. 内存记忆体到内存记忆体。
b. 内存记忆体到设备。
c. 设备到内存记忆体。
d. 设备到设备。

图表 9-2 DMA 传输阶层

单独传输数据量可由以下暂存器写入的值来计算

- 来源传输数据量(bytes) :
  src_single_size_bytes = CHx_CTL.SRC_TR_WIDTH/8
- 来源突发传输数据量(bytes):
  src_burst_size_bytes = CHx_CTL.SRC_MSIZE * src_single_size_bytes
- 目标传输数据量(bytes):
  dst_single_size_bytes = CHx_CTL.DST_TR_WIDTH/8
- 目标突发传输数据量(bytes):
  dst_burst_size_bytes = CHx_CTL.DST_MSIZE * dst_single_size_bytes

传输流程控制权可由 DMA 控制器或来源装置或目的装置来控制, 当进行块状数据传输时, 传输数据量计算如下:

- 由 DMA 控制器控制传输流程:
  blk_size_bytes_dma = CHx_BLOCK_TS.BLOCK_TS * src_single_size_bytes
- 由来源装置控制传输流程:
  blk_size_bytes_src = (来源装置发出区块突发传输次数 * src_burst_size_bytes) + (来源装置区块单独传输次数 * src_single_size_bytes)
- 由目的装置控制传输流程:

blk_size_bytes_dst = (目的装置发出区块突发传输次数 * dst_burst_size_bytes) + (目的装置区块单独传输次数 * dst_single_size_bytes)

9.3.4 链表传输

链表传输使用在需要进行多个不连续位址的块状传输，在每个块状资料后面会带着一个炼表信息，存放着下一个节点的信息， 使得数据传输时不需要 CPU 的干预, 能直接进行下一个不连续空间的块状传输。图表 3-6 为链表信息的配置格式, 必须符合信息格式才能进入链表传输工作。

0x3C	31 Reserved 0		
0x38	31 Reserved 0		
0x34	31 CHx_LLP_STATUS [63:32] 0		
0x30	31 CHx_LLP_STATUS [31:0] 0		
0x2C	31 Write back for CHx_DSTAT 0		
0x28	31 Write back for CHx_SSTAT 0		
0x24	31 CHx_CTL [63:32] 0		
0x20	31 CHx_CTL [31:0] 0		
0x1C	31 CHx_LLP [63:32] 0		
0x18	31 CHx_LLP [31:5] 6	5 Reserved 0	
0x14	31 Reserved 0		
0x10	31 CHx_BLOCK_TS [31:0] 0		
0x0C	31 CHx_DAR [63:32] 0		
0x08	31 CHx_DAR [31:0] 0		
0x04	31 CHx_SAR [63:32] 0		
0x00	31 CHx_SAR [31:0] 0		

图表 9-3 链表相对位址与数据格式

9.3.5 中断和状态

DMAC 的中断来源为以下：

a. DMA 传输完成触发。
b. 块状传输完成触发。
c. 单笔传输完成触发。
d. 内部错误触发。
e. 通道中止触发或通道取消触发。

图表 9-4 为中断状态和来源示意图

图表 9-4 中断状态和来源示意图

9.3.6 通道安全配置

通道安全可用各通道的 awprot 值与 arprot 值来实现，遵从 AXI 总线协议，当通道为安全通道时，arprot 或 awprot 值需为 0x0， 若为其他值则为非安全通道。

9.4 工作方式

9.4.1 时钟和复位

DMAC 的时钟经由 CLK_EN_1[1]写入 0x1 后使能， 时钟方可正常工作。将 REG_SOFT_RESET_X_SDMA_INIT 写入 0x0 可将 DMAC 复位， 写入 0x1 将 DMAC 离开复位。

9.4.2 初始化

进行复位后, 可按照以下步骤进行初始化

1. 外设配置: 在 DMA 通道映射章节说明了 DMA 的外设请求线的的配置方法，根据使用的场景做外设请求线的绑定配置。
2. 确认通道关闭：将 DMA 支持的 8 个通道的使能暂存器 DMAC_ChEnReg 写入 0x0，确认通道关闭
3. 确认中断来源: 将暂存器 DMAC_COMMONREG_INTSIGNAL_ENABLEREG 与 CHx_INSTATUS_ENABLEREG 写入 0x0 关闭所有中断来源， 再将需求的中断来源写入 0x1 使能。
4. 配置通道优先权: 通道同时数据传输时会按照优先权的等级来裁判通过的顺序，
   通道暂存器 CH_PRIOR 写入的值越大，优先等级越高，优先通过。

9.4.3 基本传输

最大可支持 8 个通道同时传输, 在初始化后需使能 DMAC 通道方可开始进行数据传输, 可参考以下步骤进行内存到内存的数据传输

- 读取暂存器 DMAC_ChEnReg 获得闲置通道
- 分别将 0x0 写入通道暂存器 SRC_MULTBLK_TYPE, DST_MULTBLK_TYPE 配置为连续块状传输
- 将 0x0 写入暂存器 TT_FC, 将通道配置为内存到内存数据传输。
- 将传输的信息写入暂存器 CHx_SAR, CHx_ADR, CHx_BLOCK_TS, CHx_CTL
- 将 0x1 写入暂存器 DMAC_ChEnReg，将选定的 DMA 通道使能
- 软件可透过中断或轮询的方式获得 BLOCK_TFR_DONE 状态，当其值上升为 1 时, 即

表示数据传输已完成, 再将 0x0 写入 DMAC_ChEnReg 关闭通道, 使其恢复为闲置通道。

9.4.4 链表传输

链表传输不限定节点的数目， 除了结束的节点, 每个节点必须存有指向下一个节点的信息，炼表传输可参考以下步骤完成

1. 读取暂存器 DMAC_ChEnReg 获得闲置通道。
2. 分别将 0x3 写入通道暂存器 SRC_MULTBLK_TYPE, DST_MULTBLK_TYPE 配置为链表传输
3. 配置暂存器 CHx_LLP, CHx_CTL.ShadowReg_Or_LLI_Valid, CHx_CTL.LLI_Last，将指向第一个节点所需的信息写入。
4. 将 0x1 写入暂存器 DMAC_ChEnReg，将选定的 DMA 通道使能
5. 软件可透过中断或轮询的方式获得 BLOCK_TFR_DONE 状态，当其值上升为 1 时, 即表示最后一个节点数据传输已完成, 再将 0x0 写入 DMAC_ChEnReg 关闭通道, 使其恢复为闲置通道。

9.4.5 中断处理

中断触发后的处理步骤如下

1. 找出中断来源: 读取暂存器 CHx_INTSTATUS 与 DMAC_INSTATUSREG 来获得其值为 0x1 的中断来源。 当中断发生后，会被记录为 0x1 在对应的选定位元上，若同时有多个中断发生， 软件可依其优先等级先服务。
2. **清除中断: 写 0x1 到 CHx_INTCLEARREG 或 DMAC_INTCLEARREG 的选定位元上，此时**记**录在** CHx_INTSTATUS 与 DMAC_INSTATUSREG 的中断状态会回复为 0x0，重新记录下一次的中断条件发生。

9.5 DMAC 寄存器

DMAC_IDREG

Offset Address: 0x000

Bits	Name	Access	Description	Reset
63:0	DMAC_IDREG	RO	DMAC ID Number	

DMAC_COMPVERREG

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	DMAC_COMPVER	RO	DMAC Component Version Number.	

DMAC_CFGREG

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	DMAC_EN	R/W	This bit is used to enable the DW_axi_dmac. ■ 0: DW_axi_dmac disabled ■ 1: DW_axi_dmac enabled NOTE: If this bit DMAC_EN bit is cleared while any channel is still active, then this bit still returns 1 to indicate that there are channels still active until DW_axi_dmac hardware has terminated all activity on all channels, at which point this bit returns zero (0).	0x0
1	INT_EN	R/W	This bit is used to globally enable the interrupt generation. ■ 0: DW_axi_dmac Interrupts are disabled ■ 1: DW_axi_dmac Interrupt logic is enabled.	0x0
31:2	Reserved			

DMAC_CHENREG

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	CH1_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-1. ■ 0: DW_axi_dmac Channel-1 is disabled ■ 1: DW_axi_dmac Channel-1 is enabled The bit 'DMAC_ChEnReg.CH1_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
1	CH2_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-2. ■ 0: DW_axi_dmac Channel-2 is disabled ■ 1: DW_axi_dmac Channel-2 is enabled The bit 'DMAC_ChEnReg.CH2_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
2	CH3_EN	R/W	This bit is used to enable the	0x0

Bits	Name	Access	Description	Reset
			DW_axi_dmac Channel-3. ■ 0: DW_axi_dmac Channel-3 is disabled ■ 1: DW_axi_dmac Channel-3 is enabled The bit 'DMAC_ChEnReg.CH3_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	
3	CH4_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-4. ■ 0: DW_axi_dmac Channel-4 is disabled ■ 1: DW_axi_dmac Channel-4 is enabled The bit 'DMAC_ChEnReg.CH4_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
4	CH5_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-5. ■ 0: DW_axi_dmac Channel-5 is disabled ■ 1: DW_axi_dmac Channel-5 is enabled The bit 'DMAC_ChEnReg.CH5_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
5	CH6_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-6. ■ 0: DW_axi_dmac Channel-6 is disabled ■ 1: DW_axi_dmac Channel-6 is enabled The bit 'DMAC_ChEnReg.CH6_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the	0x0

Bits	Name	Access	Description	Reset
			destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	
6	CH7_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-7. ■ 0: DW_axi_dmac Channel-7 is disabled ■ 1: DW_axi_dmac Channel-7 is enabled The bit 'DMAC_ChEnReg.CH7_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
7	CH8_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-8. ■ 0: DW_axi_dmac Channel-8 is disabled ■ 1: DW_axi_dmac Channel-8 is enabled The bit 'DMAC_ChEnReg.CH8_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
8	CH1_EN_WE	WO	DW_axi_dmac Channel-1 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
9	CH2_EN_WE	WO	DW_axi_dmac Channel-2 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
10	CH3_EN_WE	WO	DW_axi_dmac Channel-3 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
11	CH4_EN_WE	WO	DW_axi_dmac Channel-4 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0

Bits	Name	Access	Description	Reset
12	CH5_EN_WE	WO	DW_axi_dmac Channel-5 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
13	CH6_EN_WE	WO	DW_axi_dmac Channel-6 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
14	CH7_EN_WE	WO	DW_axi_dmac Channel-7 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
15	CH8_EN_WE	WO	DW_axi_dmac Channel-8 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
16	CH1_SUSP	R/W	Channel-1 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH1_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH1_SUSP bit to 1 and polls CH1_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH1_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH1_SUSP bit to 0, after DW_axi_dmac sets CH1_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
17	CH2_SUSP	R/W	Channel-2 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully	0x0

Bits	Name	Access	Description	Reset
			until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH2_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH2_SUSP bit to 1 and polls CH2_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH2_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH2_SUSP bit to 0, after DW_axi_dmac sets CH2_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
18	CH3_SUSP	R/W	Channel-3 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH3_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH3_SUSP bit to 1 and polls CH3_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH3_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH3_SUSP bit to 0, after DW_axi_dmac sets CH3_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
19	CH4_SUSP	R/W	Channel-4 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully	0x0

Bits	Name	Access	Description	Reset
			until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH4_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH4_SUSP bit to 1 and polls CH4_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH4_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH4_SUSP bit to 0, after DW_axi_dmac sets CH4_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
20	CH5_SUSP	R/W	Channel-5 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH5_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH5_SUSP bit to 1 and polls CH5_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH5_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH5_SUSP bit to 0, after DW_axi_dmac sets CH5_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
21	CH6_SUSP	R/W	Channel-6 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully	0x0

Bits	Name	Access	Description	Reset
			until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH6_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH6_SUSP bit to 1 and polls CH6_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH6_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH6_SUSP bit to 0, after DW_axi_dmac sets CH6_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
22	CH7_SUSP	R/W	Channel-7 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH7_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH7_SUSP bit to 1 and polls CH7_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH7_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH7_SUSP bit to 0, after DW_axi_dmac sets CH7_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
23	CH8_SUSP	R/W	Channel-8 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully	0x0

Bits	Name	Access	Description	Reset
			until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH8_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH8_SUSP bit to 1 and polls CH8_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH8_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH8_SUSP bit to 0, after DW_axi_dmac sets CH8_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
24	CH1_SUSP_WE	WO	This bit is used as a write enable to the Channel-1 Suspend bit. The read back value of this register bit is always 0.	0x0
25	CH2_SUSP_WE	WO	This bit is used as a write enable to the Channel-2 Suspend bit. The read back value of this register bit is always 0.	0x0
26	CH3_SUSP_WE	WO	This bit is used as a write enable to the Channel-3 Suspend bit. The read back value of this register bit is always 0.	0x0
27	CH4_SUSP_WE	WO	This bit is used as a write enable to the Channel-4 Suspend bit. The read back value of this register bit is always 0.	0x0
28	CH5_SUSP_WE	WO	This bit is used as a write enable to the Channel-5 Suspend bit. The read back value of this register bit is always 0.	0x0
29	CH6_SUSP_WE	WO	This bit is used as a write enable to the Channel-6 Suspend bit. The read back value of this register bit is always 0.	0x0
30	CH7_SUSP_WE	WO	This bit is used as a write enable to the Channel-7 Suspend bit. The read back value of this register bit is always 0.	0x0
31	CH8_SUSP_WE	WO	This bit is used as a write enable to the Channel-8 Suspend bit. The read back value of this register bit is always 0.	0x0
32	CH1_ABORT	R/W	Channel-1 Abort Request. Software sets this bit to 1 to request channel abort. If this bit	0x0

Bits	Name	Access	Description	Reset
			is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH1_Status.CH_ABORTED bit to 1).	
33	CH2_ABORT	R/W	Channel-2 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH2_Status.CH_ABORTED bit to 1).	0x0
34	CH3_ABORT	R/W	Channel-3 Abort Request. Software sets this bit to 1 to request channel abort. If this bit	0x0

Bits	Name	Access	Description	Reset
			is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH3_Status.CH_ABORTED bit to 1).	
35	CH4_ABORT	R/W	Channel-4 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH4_Status.CH_ABORTED bit to 1).	0x0
36	CH5_ABORT	R/W	Channel-5 Abort Request. Software sets this bit to 1 to request channel abort. If this bit	0x0

Bits	Name	Access	Description	Reset
			is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH5_Status.CH_ABORTED bit to 1).	
37	CH6_ABORT	R/W	Channel-6 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH6_Status.CH_ABORTED bit to 1).	0x0
38	CH7_ABORT	R/W	Channel-7 Abort Request. Software sets this bit to 1 to request channel abort. If this bit	0x0

Bits	Name	Access	Description	Reset
			is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH7_Status.CH_ABORTED bit to 1).	
39	CH8_ABORT	R/W	Channel-8 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH8_Status.CH_ABORTED bit to 1).	0x0
40	CH1_ABORT_WE	R/W	This bit is used to write enable the Channel-1 Abort bit. The read back value of this register bit is	0x0

Bits	Name	Access	Description	Reset
			always 0.	
41	CH2_ABORT_WE	R/W	This bit is used to write enable the Channel-2 Abort bit. The read back value of this register bit is always 0.	0x0
42	CH3_ABORT_WE	R/W	This bit is used to write enable the Channel-3 Abort bit. The read back value of this register bit is always 0.	0x0
43	CH4_ABORT_WE	R/W	This bit is used to write enable the Channel-4 Abort bit. The read back value of this register bit is always 0.	0x0
44	CH5_ABORT_WE	R/W	This bit is used to write enable the Channel-5 Abort bit. The read back value of this register bit is always 0.	0x0
45	CH6_ABORT_WE	R/W	This bit is used to write enable the Channel-6 Abort bit. The read back value of this register bit is always 0.	0x0
46	CH7_ABORT_WE	R/W	This bit is used to write enable the Channel-7 Abort bit. The read back value of this register bit is always 0.	0x0
47	CH8_ABORT_WE	R/W	This bit is used to write enable the Channel-8 Abort bit. The read back value of this register bit is always 0.	0x0
63:48	RSVD_DMAC_CHENREG	RO	DMAC_CHENREG Reserved bits	

DMAC_INTSTATUSREG

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CH1_IntStat	RO	Channel 1 Interrupt Status Bit.	
1	CH2_IntStat	RO	Channel 2 Interrupt Status Bit.	
2	CH3_IntStat	RO	Channel 3 Interrupt Status Bit.	
3	CH4_IntStat	RO	Channel 4 Interrupt Status Bit.	
4	CH5_IntStat	RO	Channel 5 Interrupt Status Bit.	
5	CH6_IntStat	RO	Channel 6 Interrupt Status Bit.	
6	CH7_IntStat	RO	Channel 7 Interrupt Status Bit.	
7	CH8_IntStat	RO	Channel 8 Interrupt Status Bit.	
15:8	Reserved			
16	CommonReg_IntStat	RO	Common Register Interrupt Status Bit.	
31:17	Reserved			

DMAC_COMMONREG_INTCLEARREG

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	Clear_SLVIF_CommonReg_DEC_ERR_IntStat	WO	Slave Interface Common Register Decode Error Interrupt clear Bit.	0x0

Bits	Name	Access	Description	Reset
			This bit is used to clear the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	
1	Clear_SLVIF_CommonReg_WR2RO_ERR_IntStat	WO	Slave Interface Common Register Write to Read only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
2	Clear_SLVIF_CommonReg_RD2WO_ERR_IntStat	WO	Slave Interface Common Register Read to Write only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
3	Clear_SLVIF_CommonReg_WrOnHold_ERR_IntStat	WO	Slave Interface Common Register Write On Hold Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	Clear_SLVIF_UndefinedReg_DEC_ERR_IntStat	WO	Slave Interface Undefined register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSTATUS_ENABLEREG

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_CommonReg_DEC_ERR_IntStat	R/W	Slave Interface Common Register Decode Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat	0x0

Bits	Name	Access	Description	Reset
			in DMAC_CommonReg_IntStatusReg.	
1	Enable_SLVIF_CommonReg_WR2RO_ERR_IntStat	R/W	Slave Interface Common Register Write to Read only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntStat	R/W	Slave Interface Common Register Read to Write only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntStat	R/W	Slave Interface Common Register Write On Hold Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntStat	R/W	Slave Interface Undefined register Decode Error Interrupt Status enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSIGNAL_ENABLEREG

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_CommonReg_DEC_ERR_IntSignal	R/W	Slave Interface Common Register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
1	Enable_SLVIF_CommonReg_WR2RO	R/W	Slave Interface Common Register Write	0x0

Bits	Name	Access	Description	Reset
	_ERR_IntSignal		to Read only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntSignal	R/W	Slave Interface Common Register Read to Write only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntSignal	R/W	Slave Interface Common Register Write On Hold Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
7:4	Reserved			
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntSignal	R/W	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSTATUSREG

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	SLVIF_CommonReg_DEC_ERR_IntStat	RO	Slave Interface Common Register Decode Error Interrupt Status Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to an invalid address in the common register	

Bits	Name	Access	Description	Reset
			space (0x000 to 0x0FF) resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode Errors. ■ 1: Slave Interface Decode Error detected. The Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
1	SLVIF_CommonReg_WR2RO_ERR_IntStat	RO	Slave Interface Common Register Write to Read Only Error Interrupt Status bit. This error occurs if write operation is performed to a Read Only register in the common register space (0x000 to 0x0FF). ■ 0: No Slave Interface Write to Read Only Errors. ■ 1: Slave Interface Write to Read Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
2	SLVIF_CommonReg_RD2WO_ERR_IntStat	RO	Slave Interface Common Register Read to Write only Error Interrupt Status bit. This error occurs if Read operation is performed to a Write Only register in the common register space (0x000 to 0x0FF). ■ 0: No Slave Interface Read to Write Only Errors. ■ 1: Slave Interface Read to Write Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in	

Bits	Name	Access	Description	Reset
			DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: ■ 0x1 (Active_CommonReg_RD2WO_ERR): Slave Interface Read to Write Only Error detected ■ 0x0 (Inactive_CommonReg_RD2WO_ERR): No Slave Interface Read to Write Only Errors	
3	SLVIF_CommonReg_WrOnHold_ERR_IntStat	RO	Slave Interface Common Register Write On Hold Error Interrupt Status Bit. This error occurs if an illegal write operation is performed on a common register; this happens if a write operation is performed on a common register except DMAC_RESETREG with DMAC_RST field set to 1 when DW_axi_dmac is in Hold mode. ■ 0: No Slave Interface Common Register Write On Hold Errors. ■ 1: Slave Interface Common Register Write On Hold Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
7:4	Reserved			
8	SLVIF_UndefinedReg_DEC_ERR_IntStat	RO	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to undefined address range (>0x8FF if 8	

Bits	Name	Access	Description	Reset
			channels are configured, >0x4FF if 4 channels are configured etc.) resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode Errors. ■ 1: Slave Interface Decode Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
31:9	Reserved			

DMAC_RESETREG

Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	DMAC_RST	R/W	DMAC Reset Request bit Software writes 1 to this bit to reset the DW_axi_dmac and polls this bit to see it as 0. DW_axi_dmac resets all the modules except the slave bus interface module and clears this bit to 0. NOTE: Software is not allowed to write 0 to this bit.	0x0
31:1	Reserved			

CHx_SAR

Offset Address: 0x100

Bits	Name	Access	Description	Reset
63:0	SAR	R/W	Current Source Address of DMA transfer. Updated after each source transfer. The SINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer.	0x0

CHx_DAR

Offset Address: 0x108

Bits	Name	Access	Description	Reset
63:0	DAR	R/W	Current Destination Address of DMA transfer. Updated after each destination transfer.	0x0

Bits	Name	Access	Description	Reset
			The DINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer.	

CHx_BLOCK_TS

Offset Address: 0x110

Bits	Name	Access	Description	Reset
21:0	BLOCK_TS	R/W	Block Transfer Size. The number programmed into BLOCK_TS field indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH to be transferred in a DMA block transfer. Block Transfer Size = BLOCK_TS+1 When the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of who is the flow controller. When the source or destination peripheral is assigned as the flow controller, the value before the transfer starts saturates at DMAX_CHx_MAX_BLK_SIZE, but the actual block size can be greater.	0x0
31:22	Reserved			

CHx_CTL

Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	SMS	R/W	Source Master Select. Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. ■ 0: AXI master 1 ■ 1: AXI Master 2	0x0
1	Reserved			
2	DMS	R/W	Destination Master Select. Identifies the Master Interface layer from which the destination device (peripheral or memory) is accessed. ■ 0: AXI master 1 ■ 1: AXI Master 2	0x0
3	Reserved			
4	SINC	R/W	Source Address Increment. Indicates whether to increment the source address on every source transfer. If the device is fetching data from a source	0x0

Bits	Name	Access	Description	Reset
			peripheral FIFO with a fixed address, then set this field to 'No change'. ■ 0: Increment ■ 1: No Change	
5	Reserved			
6	DINC	R/W	Destination Address Increment. Indicates whether to increment the destination address on every destination transfer. If the device is writing data from a source peripheral FIFO with a fixed address, then set this field to 'No change'. ■ 0: Increment ■ 1: No Change	0x0
7	Reserved			
10:8	SRC_TR_WIDTH	R/W	Source Transfer Width. Mapped to AXI bus arsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0
13:11	DST_TR_WIDTH	R/W	Destination Transfer Width. Mapped to AXI bus awsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0
17:14	SRC_MSIZE	R/W	Source Burst Transaction Length. Number of data items, each of width CHx_CTL.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from the corresponding hardware or software handshaking interface. The maximum value of DST_MSIZE is limited by DMAX_CHx_MAX_MULT_SIZE.	0x0
21:18	DST_MSIZE	R/W	Destination Burst Transaction Length. Number of data items, each of width CHx_CTL.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from the corresponding hardware or software handshaking interface.Note: This Value is not related to the AXI awlen signal.	0x0
25:22	AR_CACHE	R/W	AXI 'ar_cache' signal	0x0
29:26	AW_CACHE	R/W	AXI 'aw_cache' signal	0x0
30	NonPosted_LastWrite_En	R/W	Non Posted Last Write Enable This bit decides whether posted writes can be used throughout the block transfer. ■ 0: Posted writes may be used throughout the block transfer. ■ 1: Posted writes may be used till the	0x0

Bits	Name	Access	Description	Reset
			end of the block (inside a block) and the last write in the block must be non-posted. This is to synchronize block completion interrupt generation to the last write data reaching the end memory/peripheral.	
31	Reserved			
34:32	AR_PROT	R/W	AXI 'ar_prot' signal	0x0
37:35	AW_PROT	R/W	AXI 'aw_prot' signal	0x0
38	ARLEN_EN	R/W	Source Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.ARLEN as AXI Burst length for source data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for source data transfer.	0x0
46:39	ARLEN	R/W	Source Burst Length AXI Burst length used for source data transfer. The specified burst length is used for source data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of ARLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH	0x0
47	AWLEN_EN	R/W	Destination Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.AWLEN as AXI Burst length for destination data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value which is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for destination data transfer.	0x0
55:48	AWLEN	RO	Destination Burst Length AXI Burst length used for destination data transfer. The specified burst length is used for destination data transfer till	

Bits	Name	Access	Description	Reset
			the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of AWLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH.	
56	SRC_STAT_EN	R/W	Source Status Enable Enable the logic to fetch status from source peripheral of channel x pointed to by the content of CHx_SSTATAR register and stores it in CHx_SSTAT register. This value is written back to the CHx_SSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0
57	DST_STAT_EN	R/W	Destination Status Enable Enable the logic to fetch status from destination peripheral of channel x pointed to by the content of CHx_DSTATAR register and stores it in CHx_DSTAT register. This value is written back to the CHx_DSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0
58	IOC_BlkTfr	R/W	Interrupt On completion of Block Transfer This bit is used to control the block transfer completion interrupt generation on a block by block basis for shadow register or linked list based multi-block transfers. Writing 1 to this register field enables CHx_IntStatusReg.BLOCK_TFR_DONE_IntStat field if this interrupt generation is enabled in CHx_IntStatus_EnableReg register and the external interrupt output is is asserted if this interrupt generation is enabled in CHx_IntSignal_EnableReg register.	0x0
61:59	Reserved			
62	SHADOWREG_OR_LLI_LAST	R/W	Last Shadow Register/Linked List Item.	0x0

Bits	Name	Access	Description	Reset
			Indicates whether shadow register content or the linked list item fetched from the memory is the last one or not. ■ 0: Not last Shadow Register/LLI ■ 1: Last Shadow Register/LLI	
63	SHADOWREG_OR_LLI_VALID	R/W	Shadow Register content/Linked List Item valid. Indicates whether the content of shadow register or the linked list item fetched from the memory is valid. ■ 0: Shadow Register content/LLI is invalid. ■ 1: Last Shadow Register/LLI is valid.	0x0

CHx_CFG

Offset Address: 0x120

Bits	Name	Access	Description	Reset
1:0	SRC_MULTBLK_TYPE	RO	Source Multi Block Transfer Type. These bits define the type of multi-block transfer used for source peripheral. ■ 00: Contiguous ■ 01: Reload ■ 10: Shadow Register ■ 11: Linked List If the type selected is Contiguous, the CHx_SAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Reload, the CHx_SAR register is reloaded from the initial value of SAR at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_SAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_SAR register is loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A	

Bits	Name	Access	Description	Reset
			new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: ■ 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Source Transfer ■ 0x1 (RELOAD): Reload Multiblock Type used for Source Transfer ■ 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Source Transfer ■ 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Source Transfer	
3:2	DST_MULTBLK_TYPE	R/W	Destination Multi Block Transfer Type. These bits define the type of multi-block transfer used for destination peripheral. ■ 00: Contiguous ■ 01: Reload ■ 10: Shadow Register ■ 11: Linked List If the type selected is Contiguous, the CHx_DAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Reload, the CHx_DAR register is reloaded from the initial value of DAR at the end of every	0x0

Bits	Name	Access	Description	Reset
			block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_DAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_DAR register is loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: ■ 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Destination Transfer ■ 0x1 (RELOAD): Reload Multiblock Type used for Destination Transfer ■ 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Destination Transfer ■ 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Destination Transfer	
31:4	Reserved			

Bits	Name	Access	Description	Reset
34:32	TT_FC	R/W	Transfer Type and Flow Control. The following transfer types are supported. ■ Memory to Memory ■ Memory to Peripheral ■ Peripheral to Memory ■ Peripheral to Peripheral Flow Control can be assigned to the DW_axi_dmac, the source peripheral, or hte destination peripheral. Values: ■ 0x0 (MEM_TO_MEM_DMAC): Transfer Type is memory to memory and Flow Controller is DW_axi_dmac ■ 0x1 (MEM_TO_PER_DMAC): Transfer Type is memory to peripheral and Flow Controller is DW_axi_dmac ■ 0x2 (PER_TO_MEM_DMAC): Transfer Type is peripheral to memory and Flow Controller is DW_axi_dmac ■ 0x3 (PER_TO_PER_DMAC): Transfer Type is peripheral to peripheral and Flow Controller is DW_axi_dmac ■ 0x4 (PER_TO_MEM_SRC): Transfer Type is peripheral to Memory and Flow Controller is Source peripheral ■ 0x5 (PER_TO_PER_SRC): Transfer Type is peripheral to peripheral and Flow Controller is Source peripheral ■ 0x6 (MEM_TO_PER_DST): Transfer Type is memory to peripheral and Flow Controller is Destination peripheral ■ 0x7 (PER_TO_PER_DST): Transfer Type is peripheral to peripheral and Flow Controller is Destination peripheral	0x0
35	HS_SEL_SRC	R/W	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for source requests on this channel. ■ 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ■ 1: Software handshaking interface. Hardware-initiated transaction requests are ignored.	0x0

Bits	Name	Access	Description	Reset
			If the source peripheral is memory, then this bit is ignored.	
36	HS_SEL_DST	R/W	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for destination requests on this channel. ■ 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ■ 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.	0x0
37	SRC_HWHS_POL	RO	Source Hardware Handshaking Interface Polarity. ■ 0: ACTIVE HIGH ■ 1: ACTIVE LOW	
38	DST_HWHS_POL	RO	Destination Hardware Handshaking Interface Polarity. ■ 0: ACTIVE HIGH ■ 1: ACTIVE LOW	
39	SRC_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the source of Channelx if the CHx_CFG.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1 Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x0
43:40	Reserved			
44	DST_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the destination of Channelx if the CHx_CFG.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1	0x0

Bits	Name	Access	Description	Reset
			Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	
48:45	Reserved			
51:49	CH_PRIOR	R/W	Channel Priority A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: DMAX_NUM_CHANNELS-1 A programmed value outside this range will cause erroneous behavior.	0x0
52	LOCK_CH	R/W	Channel Lock bit When the channel is granted control of the master bus interface and if the CHx_CFG.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in this case, the read-back value is always 0. Locking the channel locks AXI Read Address, Write Address and Write Data channels on the corresponding master interface. Note: Channel locking feature is supported only for memoryto-memory transfer at Block Transfer and DMA Transfer levels. Hardware does not check for the validity of channel locking setting, hence the software must take care of enabling the channel locking only for memory-to-memory transfers at Block Transfer or DMA Transfer levels. Illegal programming of channel locking might result in unpredictable behavior.	0x0
54:53	LOCK_CH_L	R/W	Channel Lock Level This bit indicates the duration over which	0x0

Bits	Name	Access	Description	Reset
			CHx_CFG.LOCK_CH bit applies. ■ 00: Over complete DMA transfer ■ 01: Over DMA block transfer ■ 1x: Reserved This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in that case, the read-back value is always 0.	
58:55	SRC_OSR_LMT	R/W	Source Outstanding Request Limit ■ Maximum outstanding request supported is 16. ■ Source Outstanding Request Limit = SRC_OSR_LMT + 1	0x0
62:59	DST_OSR_LMT	R/W	Destination Outstanding Request Limit ■ Maximum outstanding request supported is 16. ■ Source Outstanding Request Limit = DST_OSR_LMT + 1	0x0
63	Reserved			

CHx_LLP

Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	LMS	R/W	LLI master Select This bit identifies the AXI layer/interface where the memory device that stores the next linked list item resides. - 0: AXI Master 1 ■ 1: AXI Master 2 This field does not exist if the configuration parameter DMAX_CHx_LMS is not set to NO_HARDCODE.	0x0
5:1	Reserved			
63:6	LOC	R/W	Starting Address Memory of LLI block Starting Address In Memory of next LLI if block chaining is enabled. The six LSBs of the starting address are not stored because the address is assumed to be aligned to a 64-byte boundary. LLI access always uses the burst size (arsize/awsize) that is same as the data bus width and cannot be changed or programmed to anything other than this. Burst length (awlen/arlen) is chosen based on the data bus width so that the access does not cross one complete LLI structure of 64 bytes. DW_axi_dmac will fetch the entire LLI (40 bytes) in	0x0

Bits	Name	Access	Description	Reset
			one AXI burst if the burst length is not limited by other settings.	

CHx_STATUSREG

Offset Address: 0x130

Bits	Name	Access	Description	Reset
21:0	CMPLTD_BLK_TFR_SIZE	RO	Completed Block Transfer Size. This bit indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH transferred for the previous block transfer.	
31:22	Reserved			
46:32	DATA_LEFT_IN_FIFO	RO	Data Left in FIFO. This bit indicates the total number of data left in DW_axi_dmac channel FIFO after completing the current block transfer.	
63:47	Reserved			

CHx_SWHSSRCREG

Offset Address: 0x138

Bits	Name	Access	Description	Reset
0	SWHS_REQ_SRC	R/W	Software Handshake Request for Channel Source. This bit is used to request dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller or not.	0x0
1	SWHS_REQ_SRC_WE	WO	Write Enable bit for Software Handshake Request for Channel Source.	0x0
2	SWHS_SGLREQ_SRC	R/W	Software Handshake Single Request for Channel Source. This bit is used to request SINGLE (AXI burst length = 1) dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.	0x0

Bits	Name	Access	Description	Reset
3	SWHS_SGLREQ_SRC_WE	WO	Write Enable bit for Software Handshake Single Request for Channel Source.	0x0
4	SWHS_LST_SRC	R/W	Software Handshake Last Request for Channel Source. This bit is used to request LAST dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx or if the source of Channelx is not the flow controller.	0x0
5	SWHS_LST_SRC_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Source.	0x0
31:6	Reserved			

CHx_SWHSDSTREG

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	SWHS_REQ_DST	R/W	Software Handshake Request for Channel Destination. This bit is used to request dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
1	SWHS_REQ_DST_WE	WO	Write Enable bit for Software Handshake Request for Channel Destination.	0x0
2	SWHS_SGLREQ_DST	R/W	Software Handshake Single Request for Channel Destination. This bit is used to request SINGLE (AXI burst length = 1) dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
3	SWHS_SGLREQ_DST_WE	WO	Write Enable bit for Software Handshake Single Request for Channel Destination.	0x0
4	SWHS_LST_DST	R/W	Software Handshake Last Request for Channel Destination. This bit is used to request LAST dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
5	SWHS_LST_DST_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Destination.	0x0

Bits	Name	Access	Description	Reset
31:6	Reserved			

CHx_BLK_TFR_RESUMEREQREG

Offset Address: 0x148

Bits	Name	Access	Description	Reset
0	BLK_TFR_RESUMEREQ	WO	Block Transfer Resume Request during Linked-List or Shadow-Register-based multi-block transfer.	0x0
31:1	Reserved			

CHx_AXI_IDREG

Offset Address: 0x150

Bits	Name	Access	Description	Reset
14:0	AXI_READ_ID_SUFFIX	R/W	AXI Read ID Suffix These bits form part of the ARID output of AXI3/AXI4 master interface.	0x0
15	Reserved			
30:16	AXI_WRITE_ID_SUFFIX	R/W	AXI Write ID Suffix. These bits form part of the AWID output of AXI3/AXI4 master interface.	0x0
31	Reserved			

CHx_AXI_QOSREG

Offset Address: 0x158

Bits	Name	Access	Description	Reset
3:0	AXI_AWQOS	R/W	AXI AWQOS. These bits form the awqos output of AXI4 master interface.	0x0
7:4	AXI_ARQOS	R/W	AXI ARQOS. These bits form the arqos output of AXI4 master interface.	0x0
31:8	Reserved			

CHx_SSTAT

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	SSTAT	RO	Source Status Source status information retrieved by hardware from the address pointed to by the contents of the CHx_SSTATAR register.	

CHx_DSTAT

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	DSTAT	RO	Destination Status Destination status information retrieved by hardware from the address pointed to by the contents	

Bits	Name	Access	Description	Reset
			of the CHx_DSTATAR register.	

CHx_SSTATAR

Offset Address: 0x170

Bits	Name	Access	Description	Reset
63:0	SSTATAR	R/W	Source Status Fetch Address Pointer from where hardware can fetch the source status information, which is registered in the CHx_SSTAT register and written out to the CHx_SSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multi-block transfer is enabled for either source or destination peripheral of the channel. Source peripheral should update the source status information, if any, at the location pointed to by CHx_SSTATAR to utilize this feature.	0x0

CHx_DSTATAR

Offset Address: 0x178

Bits	Name	Access	Description	Reset
63:0	DSTATAR	R/W	Destination Status Fetch Address Pointer from where hardware can fetch the Destination status information, which is registered in the CHx_DSTAT register and written out to the CHx_DSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multiblock transfer is enabled for either source or destination peripheral of the channel. Destination peripheral should update the destination status information, if any, at the location pointed to by CHx_DSTATAR to utilize this feature.	0x0

CHx_INTSTATUS_ENABLEREG

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntStat	R/W	Block Transfer Done Interrupt Status Enable. ■ 0: Disable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG	0x0

Bits	Name	Access	Description	Reset
			■ 1: Enable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG	
1	Enable_DMA_TFR_DONE_IntStat	R/W	DMA Transfer Done Interrupt Status Enable. ■ 0: Disable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG	0x0
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntStat	R/W	Source Transaction Completed Status Enable. ■ 0: Disable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG	0x0
4	Enable_DST_TRANSCOMP_IntStat	R/W	Destination Transaction Completed Status Enable. ■ 0: Disable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG	0x0
5	Enable_SRC_DEC_ERR_IntStat	R/W	Source Decode Error Status Enable. ■ 0: Disable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG	0x0
6	Enable_DST_DEC_ERR_IntStat	R/W	Destination Decode Error Status Enable. ■ 0: Disable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG	0x0
7	Enable_SRC_SLV_ERR_IntStat	R/W	Source Slave Error Status Enable. ■ 0: Disable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG	0x0
8	Enable_DST_SLV_ERR_IntStat	R/W	Destination Slave Error Status Enable. ■ 0: Disable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of	0x0

Bits	Name	Access	Description	Reset
			Destination Slave Error Interrupt in CHx_INTSTATUSREG	
9	Enable_LLI_RD_DEC_ERR_IntStat	R/W	LLI Read Decode Error Status Enable. ■ 0: Disable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG	0x0
10	Enable_LLI_WR_DEC_ERR_IntStat	R/W	LLI WRITE Decode Error Status Enable. ■ 0: Disable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG	0x0
11	Enable_LLI_RD_SLV_ERR_IntStat	R/W	LLI Read Slave Error Status Enable. ■ 0: Disable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG	0x0
12	Enable_LLI_WR_SLV_ERR_IntStat	R/W	LLI WRITE Slave Error Status Enable. ■ 0: Disable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG	0x0
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	R/W	Shadow register or LLI Invalid Error Status Enable. ■ 0: Disable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG	0x0
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntStat	R/W	Slave Interface Multi Block type Error Status Enable. ■ 0: Disable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG	0x0
15	Reserved			
16	Enable_SLVIF_DEC_ERR_IntStat	R/W	Slave Interface Decode Error Status Enable. ■ 0: Disable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave	0x0

Bits	Name	Access	Description	Reset
			Interface Decode Error Interrupt in CHx_INTSTATUSREG	
17	Enable_SLVIF_WR2RO_ERR_IntStat	R/W	Slave Interface Write to Read Only Error Status Enable. ■ 0: Disable the generation of Slave Interface Write to Read only Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave Interface Write to Read Only Error Interrupt in CHx_INTSTATUSREG	0x0
18	Enable_SLVIF_RD2RWO_ERR_IntStat	R/W	Slave Interface Read to write Only Error Status Enable. ■ 0: Disable the generation of Slave Interface Read to Write only Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave Interface Read to Write Only Error Interrupt in CHx_INTSTATUSREG	0x0
19	Enable_SLVIF_WRONCHEN_ERR_IntStat	R/W	Slave Interface Write On Channel Enabled Error Status Enable. ■ 0: Disable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG	0x0
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	R/W	Shadow Register Write On Valid Error Status Enable. ■ 0: Disable the generation of Shadow Register Write On Valid Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Shadow register Write On Valid Error Interrupt in CHx_INTSTATUSREG	0x0
21	Enable_SLVIF_WRONHOLD_ERR_IntStat	R/W	Slave Interface Write On Hold Error Status Enable. ■ 0: Disable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG	0x0
26:22	Reserved			
27	Enable_CH_LOCK_CLEARED_IntStat	R/W	Channel Lock Cleared Status Enable. ■ 0: Disable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG	0x0

Bits	Name	Access	Description	Reset
			■ 1: Enable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG	
28	Enable_CH_SRC_SUSPENDED_IntStat	R/W	Channel Source Suspended Status Enable. ■ 0: Disable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG	0x0
29	Enable_CH_SUSPENDED_IntStat	R/W	Channel Suspended Status Enable. ■ 0: Disable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG	0x0
30	Enable_CH_DISABLED_IntStat	R/W	Channel Disabled Status Enable. ■ 0: Disable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG	0x0
31	Enable_CH_ABORTED_IntStat	R/W	Channel Aborted Status Enable. ■ 0: Disable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG ■ 1: Enable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG	0x0

CHx_INTSTATUS

Offset Address: 0x188

Bits	Name	Access	Description	Reset
0	BLOCK_TFR_DONE_IntStat	RO	Block Transfer Done. This indicates to the software that the DW_axi_dmac has completed the requested block transfer. The DW_axi_dmac sets this bit to 1 when the transfer is successfully completed. ■ 0: Block Transfer not completed. ■ 1: Block Transfer completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register.	
1	DMA_TFR_DONE_IntStat	RO	DMA Transfer Done. This indicates to the software that the DW_axi_dmac has completed the requested DMA transfer. The DW_axi_dmac sets this bit to 1 along with setting CHx_INTSTATUS.BLOCK_TFR_DONE bit to 1 when the last	

Bits	Name	Access	Description	Reset
			block transfer is completed. ■ 0: DMA Transfer not completed. ■ 1: DMA Transfer Completed This bit is cleared to 0 on writing 1	
2	Reserved			
3	SRC_TRANSCOMP_IntStat	RO	Source Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	
4	DST_TRANSCOMP_IntStat	RO	Destination Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	
5	SRC_DEC_ERR_IntStat	RO	Source Decode Error. Decode Error detected by Master Interface during source data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. ■ 0: No Source Decode Errors. ■ 1: Source Decode Error detected.	
6	DST_DEC_ERR_IntStat	RO	Destination Decode Error. Decode Error detected by Master Interface during destination data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. ■ 0: No destination Decode Errors. ■ 1: Destination Decode Error Detected	
7	SRC_SLV_ERR_IntStat	RO	Source Slave Error. Slave Error detected by Master Interface during source data transfer. This error occurs if the slave	

Bits	Name	Access	Description	Reset
			interface from which the data is read issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. ■ 0: No Source Slave Errors ■ 1: Source Slave Error Detected	
8	DST_SLV_ERR_IntStat	RO	Destination Slave Error. Slave Error detected by Master Interface during destination data transfer. This error occurs if the slave interface to which the data is written issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. ■ 0: No Destination Slave Errors ■ 1: Destination Slave Errors Detected	
9	LLI_RD_DEC_ERR_IntStat	RO	LLI Read Decode Error. Decode Error detected by Master Interface during LLI read operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. ■ 0: NO LLI Read Decode Errors. ■ 1: LLI Read Decode Error detected	
10	LLI_WR_DEC_ERR_IntStat	RO	LLI WRITE Decode Error. Decode Error detected by Master Interface during LLI writeback operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0.	

Bits	Name	Access	Description	Reset
			■ 0: NO LLI Write Decode Errors. ■ 1: LLI write Decode Error detected.	
11	LLI_RD_SLV_ERR_IntStat	RO	LLI Read Slave Error. Slave Error detected by Master Interface during LLI read operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. ■ 0: No LLI Read Slave Errors. ■ 1: LLI read Slave Error detected.	
12	LLI_WR_SLV_ERR_IntStat	RO	LLI WRITE Slave Error. Slave Error detected by Master Interface during LLI writeback operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. ■ 0: No LLI write Slave Errors. ■ 1: LLI Write SLAVE Error detected.	
13	SHADOWREG_OR_LLI_INVALID_ERR_IntStat	RO	Shadow register or LLI Invalid Error. This error occurs if CHx_CTL.ShadowReg_Or_LLI_Valid bit is seen to be 0 during DW_axi_dmac Shadow Register / LLI fetch phase. This error condition causes the DW_axi_dmac to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid Shadow Register availability. In the case of LLI pre-fetching, ShadowReg_Or_LLI_Invalid_ERR Interrupt is not generated even if ShadowReg_Or_LLI_Valid bit is seen to be 0 for the pre-fetched LLI. In this case, DW_axi_dmac re-attempts the LLI fetch operation after completing the current block transfer and generates ShadowReg_Or_LLI_Invalid_ERR	

Bits	Name	Access	Description	Reset
			Interrupt only if ShadowReg_Or_LLI_Valid bit is still seen to be 0. ■ 0: No Shadow Register / LLI Invalid errors. ■ 1: Shadow Register / LLI Invalid error detected.	
14	SLVIF_MULTIBLKTYPE_ERR_IntStat	RO	Slave Interface Multi Block type Error. This error occurs if multi-block transfer type programmed in CHx_CFG register (SRC_MLTBLK_TYPE and DST_MLTBLK_TYPE) is invalid. This error condition causes the DW_axi_dmac to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid multiblock transfer type availability. ■ 0: No Multi-block transfer type Errors. ■ 1: Multi-block transfer type Error detected.	
15	Reserved			
16	SLVIF_DEC_ERR_IntStat	RO	Slave Interface Decode Error. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to invalid address in Channelx register space resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode errors. ■ 1: Slave Interface Decode Error detected.	
17	SLVIF_WR2RO_ERR_IntStat	RO	Slave Interface Write to Read Only Error. This error occurs if write operation is performed to a Read Only register. ■ 0: No Slave Interface Write to Read Only Errors. ■ 1: Slave Interface Write to Read Only Error detected.	
18	SLVIF_RD2RWO_ERR_IntStat	RO	Slave Interface Read to write Only Error. This error occurs if read operation is performed to a Write Only register. ■ 0: No Slave Interface Read to Write Only Errors. ■ 1: Slave Interface Read to Write Only Error detected.	

Bits	Name	Access	Description	Reset
19	SLVIF_WRONCHEN_ERR_IntStat	RO	Slave Interface Write On Channel Enabled Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a register when the channel is enabled and if it is not allowed for the corresponding register as per the DW_axi_dmac specification. ■ 0: No Slave Interface Write On Channel Enabled Errors. ■ 1: Slave Interface Write On Channel Enabled Error detected.	
20	SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	RO	Shadow Register Write On Valid Error. This error occurs if shadow register based multi-block transfer is enabled and software tries to write to the shadow register when CHx_CTL.ShadowReg_Or_LLI_Valid bit is 1. ■ 0: No Slave Interface Shadow Register Write On Valid Errors. ■ 1: Slave Interface Shadow Register Write On Valid Error detected.	
21	SLVIF_WRONHOLD_ERR_IntStat	RO	Slave Interface Write On Hold Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a channel register when DW_axi_dmac is in Hold mode. ■ 0: No Slave Interface Write On Hold Errors. ■ 1: Slave Interface Write On Hold Error detected.	
26:22	Reserved			
27	CH_LOCK_CLEARED_IntStat	RO	Channel Lock Cleared. This indicates to the software that the locking of the corresponding channel in DW_axi_dmac is cleared. ■ 0: Channel locking is not cleared. ■ 1: Channel locking is cleared.	
28	CH_SRC_SUSPENDED_IntStat	RO	Channel Source Suspended. This indicates to the software that the corresponding channel source data transfer in DW_axi_dmac is suspended. ■ 0: Channel source is not suspended ■ 1: Channel Source is suspended.	
29	CH_SUSPENDED_IntStat	RO	Channel Suspended.	

Bits	Name	Access	Description	Reset
			This indicates to the software that the corresponding channel in DW_axi_dmac is suspended. ■ 0: Channel is not suspended. ■ 1: Channel is suspended.	
30	CH_DISABLED_IntStat	RO	Channel Disabled. This indicates to the software that the corresponding channel in DW_axi_dmac is disabled. ■ 0: Channel is not disabled. ■ 1: Channel is disabled. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled.	
31	CH_ABORTED_IntStat	RO	Channel Aborted. This indicates to the software that the corresponding channel in DW_axi_dmac is aborted. ■ 0: Channel is not aborted ■ 1: Channel is aborted	

CHx_INTSIGNAL_ENABLEREG

Offset Address: 0x190

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntSignal	R/W	Block Transfer Done Interrupt Signal Enable. ■ 0: Disable the propagation of Block Transfer Done Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Block Transfer Done Interrupt to generate a port level interrupt	0x0
1	Enable_DMA_TFR_DONE_IntSignal	R/W	DMA Transfer Done Interrupt Signal Enable. ■ 0: Disable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt ■ 1: Enable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt	0x0
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntSignal	R/W	Source Transaction Completed Signal Enable. ■ 0: Disable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt	0x0

Bits	Name	Access	Description	Reset
4	Enable_DST_TRANSCOMP_IntSignal	R/W	Destination Transaction Completed Signal Enable. ■ 0: Disable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt	0x0
5	Enable_SRC_DEC_ERR_IntSignal	R/W	Source Decode Error Signal Enable. ■ 0: Disable the propagation of Source Decode Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Source Decode Error Interrupt to generate a port level interrupt	0x0
6	Enable_DST_DEC_ERR_IntSignal	R/W	Destination Decode Error Signal Enable. ■ 0: Disable the propagation of Destination Decode Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Destination Decode Error Interrupt to generate a port level interrupt	0x0
7	Enable_SRC_SLV_ERR_IntSignal	R/W	Source Slave Error Signal Enable. ■ 0: Disable the propagation of Source Slave Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Source Slave Error Interrupt to generate a port level interrupt	0x0
8	Enable_DST_SLV_ERR_IntSignal	R/W	Destination Slave Error Signal Enable. ■ 0: Disable the propagation of Destination Slave Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Destination Slave Error Interrupt to generate a port level interrupt	0x0
9	Enable_LLI_RD_DEC_ERR_IntSignal	R/W	LLI Read Decode Error Signal Enable. ■ 0: Disable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt	0x0
10	Enable_LLI_WR_DEC_ERR_IntSignal	R/W	LLI WRITE Decode Error Signal Enable. ■ 0: Disable the propagation of LLI WRITE Decode Error Interrupt to generate a port level	0x0

Bits	Name	Access	Description	Reset
			interrupt ■ 1: Enable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt	
11	Enable_LLI_RD_SLV_ERR_IntSignal	R/W	LLI Read Slave Error Signal Enable. ■ 0: Disable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt	0x0
12	Enable_LLI_WR_SLV_ERR_IntSignal	R/W	LLI WRITE Slave Error Signal Enable. ■ 0: Disable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt	0x0
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntSignal	R/W	Shadow register or LLI Invalid Error Signal Enable. ■ 0: Disable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt	0x0
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntSignal	R/W	Slave Interface Multi Block type Error Signal Enable. ■ 0: Disable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt	0x0
15	Reserved			
16	Enable_SLVIF_DEC_ERR_IntSignal	R/W	Slave Interface Decode Error Signal Enable. ■ 0: Disable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt	0x0
17	Enable_SLVIF_WR2RO_ERR_IntSignal	R/W	Slave Interface Write to Read Only Error Signal Enable. ■ 0: Disable the propagation of Slave	0x0

Bits	Name	Access	Description	Reset
			Interface Write to Read only Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Write to Read Only Error Interrupt to generate a port level interrupt	
18	Enable_SLVIF_RD2RWO_ERR_IntSignal	R/W	Slave Interface Read to write Only Error Signal Enable. ■ 0: Disable the propagation of Slave Interface Read to Write only Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Read to Write Only Error Interrupt to generate a port level interrupt	0x0
19	Enable_SLVIF_WRONCHEN_ERR_IntSignal	R/W	Slave Interface Write On Channel Enabled Error Signal Enable. ■ 0: Disable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt	0x0
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntSignal	R/W	Shadow Register Write On Valid Error Signal Enable. ■ 0: Disable the propagation of Shadow Register Write On Valid Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Shadow register Write On Valid Error Interrupt to generate a port level interrupt	0x0
21	Enable_SLVIF_WRONHOLD_ERR_IntSignal	R/W	Slave Interface Write On Hold Error Signal Enable. ■ 0: Disable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt	0x0
26:22	Reserved			
27	Enable_CH_LOCK_CLEARED_IntSignal	R/W	Channel Lock Cleared Signal Enable. ■ 0: Disable the propagation of Channel Lock Cleared Interrupt to generate a port level	0x0

Bits	Name	Access	Description	Reset
			interrupt ■ 1: Enable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt	
28	Enable_CH_SRC_SUSPENDED_IntSignal	R/W	Channel Source Suspended Signal Enable. ■ 0: Disable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt	0x0
29	Enable_CH_SUSPENDED_IntSignal	R/W	Channel Suspended Signal Enable. ■ 0: Disable the propagation of Channel Suspended Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Channel Suspended Interrupt to generate a port level interrupt	0x0
30	Enable_CH_DISABLED_IntSignal	R/W	Channel Disabled Signal Enable. ■ 0: Disable the propagation of Channel Disabled Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Channel Disabled Interrupt to generate a port level interrupt	0x0
31	Enable_CH_ABORTED_IntSignal	R/W	Channel Aborted Signal Enable. ■ 0: Disable the propagation of Channel Aborted Interrupt to generate a port level interrupt ■ 1: Enable the propagation of Channel Aborted Interrupt to generate a port level interrupt	0x0

CHx_INTCLEARREG

Offset Address: 0x198

Bits	Name	Access	Description	Reset
0	Clear_BLOCK_TFR_DONE_IntStat	WO	Block Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CH1_INTSTATUSREG	0x0
1	Clear_DMA_TFR_DONE_IntStat	WO	DMA Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
2	Reserved			
3	Clear_SRC_TRANSCOMP_IntStat	WO	Source Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0

Bits	Name	Access	Description	Reset
4	Clear_DST_TRANSCOMP_IntStat	WO	Destination Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
5	Clear_SRC_DEC_ERR_IntStat	WO	Source Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
6	Clear_DST_DEC_ERR_IntStat	WO	Destination Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
7	Clear_SRC_SLV_ERR_IntStat	WO	Source Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
8	Clear_DST_SLV_ERR_IntStat	WO	Destination Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
9	Clear_LLI_RD_DEC_ERR_IntStat	WO	LLI Read Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
10	Clear_LLI_WR_DEC_ERR_IntStat	WO	LLI WRITE Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
11	Clear_LLI_RD_SLV_ERR_IntStat	WO	LLI Read Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
12	Clear_LLI_WR_SLV_ERR_IntStat	WO	LLI WRITE Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
13	Clear_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	WO	Shadow register or LLI Invalid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
14	Clear_SLVIF_MULTIBLKTYPE_ERR_IntStat	WO	Slave Interface Multi Block type Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
15	Reserved			
16	Clear_SLVIF_DEC_ERR_IntStat	WO	Slave Interface Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0

Bits	Name	Access	Description	Reset
17	Clear_SLVIF_WR2RO_ERR_IntStat	WO	Slave Interface Write to Read Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
18	Clear_SLVIF_RD2RWO_ERR_IntStat	WO	Slave Interface Read to write Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
19	Clear_SLVIF_WRONCHEN_ERR_IntStat	WO	Slave Interface Write On Channel Enabled Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
20	Clear_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	WO	Shadow Register Write On Valid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
21	Clear_SLVIF_WRONHOLD_ERR_IntStat	WO	Slave Interface Write On Hold Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
26:22	Reserved			
27	Clear_CH_LOCK_CLEARED_IntStat	WO	Channel Lock Cleared Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
28	Clear_CH_SRC_SUSPENDED_IntStat	WO	Channel Source Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
29	Clear_CH_SUSPENDED_IntStat	WO	Channel Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
30	Clear_CH_DISABLED_IntStat	WO	Channel Disabled Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
31	Reserved			

10 定时器

10.1 概述

系统配置 8 个 Timer 模块。用作定时、计数功能，可以供应用程序实现定时和计数，也可以供操作系统实现系统时钟。

10.2 特点

定时器具有以下特点：

- 32bit 减法定时器/计数器。
- 支持 2 种计数模式：自由运行模式、用户定义计数模式。
- 系统可读取当前的计数值。

当计数值递减到 0 时，会产生一个中断。

10.3 功能描述

Timer基于一个 32bit 减法计数器。计数器的值在每个计数时钟的上升沿减1。当计数值递减到零，Timer将产生一个中断。

Timer有以下 2 种计数模式：

- 自由运行模式: 定时器持续计数，当计数值减到0 时又自动回转到其最大值，并继续计数。计数长度最大值为0xFFFF_FFFF。
- 用户定义计数模式: 定时器持续计数，当计数值减到0 时从 TimerNLoadCount(N=1~8) 寄存器中再次载入初值并继续计数。

对定时器载入计数初值的方法如下：

通过写 TimerNLoadCount(N=1~8) 寄存器可对定时器载入计数初值。

10.4 工作方式

10.4.1 初始化

步骤 1　写 TimerNLoadCount(N=1~8) 寄存器，为 Timer 载入计数初值。

步骤 2　设置 TimerNControlReg[2:0](N=1~8) 寄存器，选择 Timer 计数模式，屏蔽 Timer 中断，启动 Timer 开始计数。

10.4.2 中断处理

定时器产生中断时，操作步骤如下：
步骤 1 读 TimerNEOI(N=1~8) 寄存器，清除 TimerN 中断。
步骤 2 执行等待该中断的进程。
步骤 3 进程完成后，恢复执行执行被中断的程序。

10.4.3 时钟选择

系统 Timer 可选 25MHz /32KHz 计数时钟。使用 reg_timer_clk_sel 做选择。

10.5 Timer 寄存器概览

Timer 寄存器通过总线存取。

Timer 寄存器概览如 表格 10-1 所示。

表格 10-1 Timer 寄存器概览 (基址 0x030A0000)

Name	Address Offset	Description
Timer1LoadCount	0x000	Value to be loaded into Timer1
Timer1CurrentValue	0x004	Current Value of Timer1
Timer1ControlReg	0x008	Control Register for Timer1
Timer1EOI	0x00c	Clears the interrupt from Timer1
Timer1IntStatus	0x010	Contains the interrupt status for Timer1
Timer2 Registers	0x014~ 0x024	共 5 个寄存器，内容与 Timer1 相同。
Timer3 Registers	0x028~ 0x038	共 5 个寄存器，内容与 Timer1 相同。
Timer4 Registers	0x03c~ 0x04c	共 5 个寄存器，内容与 Timer1 相同。
Timer5 Registers	0x050~ 0x060	共 5 个寄存器，内容与 Timer1 相同。
Timer6 Registers	0x064~ 0x074	共 5 个寄存器，内容与 Timer1 相同。
Timer7 Registers	0x078~ 0x088	共 5 个寄存器，内容与 Timer1 相同。
Timer8 Registers	0x08c~ 0x09c	共 5 个寄存器，内容与 Timer1 相同。
TimersIntStatus	0x0a0	Contains the interrupt status of all timers in the component.
TimersEOI	0x0a4	Returns all zeroes (0) and clears all active interrupts.
TimersRawIntStatus	0x0a8	Contains the unmasked interrupt status of all timers in the component.

10.6 Timer 寄存器描述

Timer1LoadCount

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	Timer1LoadCount	R/W	Timer1 Load Count Register Value to be loaded into Timer1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	0x0

Timer1CurrentValue

Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	Timer1CurrentValue	RO	Timer1 Current Value Current Value of Timer1. This register is supported only when timer_1_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value.	

Timer1ControlReg

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	Timer1ControlReg	R/W	[2] Timer interrupt mask for Timer1 0 – not masked 1 – masked [1] Timer mode for Timer1 0 – free-running mode 1 – user-defined count mode [0] Timer enable bit for Timer1 0 – disable 1 – enable	0x0
31:3	Reserved			

Timer1EOI

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	Timer1EOI	RO	Reading from this register returns all zeroes (0) and clears the interrupt from Timer1.	
31:1	Reserved			

Timer1IntStatus

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	Timer1IntStatus	RO	Contains the interrupt status for Timer1.	
31:1	Reserved			

TimersIntStatus

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	TimersIntStatus	RO	Contains the interrupt status of all timers. Reading from this register does not	

Bits	Name	Access	Description	Reset
			clear any active interrupts: 0 – either timer_intr or timer_intr_n is not active after masking 1 – either timer_intr or timer_intr_n is active after masking	
31:8	Reserved			

TimersEOI

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	TimersEOI	RO	Reading this register returns all zeroes (0) and clears all active interrupts.	
31:8	Reserved			

TimersRawIntStatus

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	TimersRawIntStatus	RO	The register contains the unmasked interrupt status of all timers. 0 – either timer_intr or timer_intr_n is not active prior to masking 1 – either timer_intr or timer_intr_n is active prior to masking	
31:8	Reserved			

11 看门狗

11.1 概述

系统配置 4 个看门狗 (WatchDog) 模块。用于系统发生异常情况下，于一定时间后发出中断或复位信号，以中断或复位整个系统。

11.2 特点

WatchDog 具备以下特点：

- 配置一个 32bit 递减计数器。
- 支持计数初值（即时间间隔）可配置。
- 支持 WatchDog 重启保护，防止误动作导致 WatchDog 被重启。
- 支持复位信号产生。
- 支持超时中断产生。

11.3 功能描述

系统通过系统总线给 WatchDog 配置寄存器参数值。WatchDog 定时发出 WDT_INTR 中断请求给系统，并在系统没有响应中断的情况下（如：死机），发出 WDT_SYS_RST 复位信号，使系统复位，达到监控系统运行的目的。

11.3.1 应用框图

图表 11-1 WatchDog 应用框图

11.3.2 功能原理

WatchDog 的计数初值由寄存器 WDT_TORR 载入，运行基于 1个 32bit 减法计数器。在 WatchDog 时钟使能情况下，计数值在每个计数时钟的上升沿减1。当计数值递减到0，WatchDog 将产生一个中断。然后在下一个计数时钟上升沿，计数器又从寄存器 WDT_TORR 中重新载入计数初值，开始递减计数。
如果计数器的计数值第二次计数递减到0时，CPU还没有清除WatchDog 中断，则 WatchDog 将发出复位信号 WDT_SYS_RST，计数器停止计数。
用户可以透过设置寄存器 WDT_CR[1] 决定是否在计数器的计数值第一次计数递减到 0 时立刻发出复位信号 WDT_SYS_RST。

11.4 工作方式

11.4.1 计数时钟频率配置

WatchDog 计数时钟为 25MHz/32KHz 时钟。使用 reg_wdt_clk_sel 做选择。

11.4.2 系统初始化配置

系统上电复位后WatchDog 计数器处于停止计数状态，在系统初始化过程中需要将

WatchDog初始化并启动其运行。WatchDog 的初始化过程如下：
步骤 1　写寄存器WDT_TORR，设置 WatchDog 计数初值。
步骤 2　写寄存器WDT_CR[1]，设置 WatchDog 计数超时响应模式。
步骤 3　写寄存器 WDT_CR[0]，启动 WatchDog 计数。

11.4.3 中断处理过程

系统收到 WatchDog 发出的中断后，应及时清除其中断状态。
WatchDog 中断处理的过程如下：
步骤 1　读寄存器 WDT_EOI，清除WatchDog 的中断状态。
步骤 2　向寄存器 WDT_CRR 写入 0x76，重启 WatchDog。

11.4.4 关闭 WatchDog

向寄存器 WDT_CR[0] 写入 0 或 1 控制 WatchDog 的状态：
- 0 : WDT 关闭。
- 1 : WDT 开启。启动后只有系统复位能够使 WDT 关闭。

11.5 WDT 寄存器概览

WDT 寄存器通过总线存取。四個 WDT 基地址為:
- WDT0　: 0x03010000
- WDT1　: 0x03011000
- WDT2　: 0x03012000
- RTCSYS_WDT　: 0x0502D000

WDT 寄存器概览如表格 11-1 所示。

表格 11-1 WDT 寄存器概览

Name	Address Offset	Description
WDT_CR	0x000	Control register
WDT_TORR	0x004	Timeout range register
WDT_CCVR	0x008	Current counter value register
WDT_CRR	0x00c	Counter restart register
WDT_STAT	0x010	Interrupt status register
WDT_EOI	0x014	Interrupt clear register
WDT_TOC	0x01C	Time Out Count

11.6 WDT 寄存器描述

WDT_CR

Offset Address: 0x000

Bits	Name	Access	Description	Reset
4:0	WDT_CR	R/W	[4:2] Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to 256 pclk cycles. 000 – 2 pclk cycles 001 – 4 pclk cycles 010 – 8 pclk cycles 011 – 16 pclk cycles 100 – 32 pclk cycles 101 – 64 pclk cycles 110 – 128 pclk cycles 111 – 256 pclk cycles [1] Response mode. Selects the output response generated to a timeout. 0 = Generate a system reset. 1 = First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset. [0] WDT enable. This bit is used to enable and disable the WatchDog. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. Once this bit has been enabled, it can be cleared only by a system reset. 0 = WDT disabled. 1 = WDT enabled.	0x0
6	TOR_MODE	R/W	The Mode of Timeout Period	0x0
7	ITOR_MODE	R/W	The Mode of Timeout Period for initialization	0x0
31:5	Reserved			

WDT_TORR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
3:0	WDT_TORR	R/W	[3:0] TOP(TimeOut Period). This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values is limited by 32-bit width. If TOP is programmed to select a range that is greater than the counter	0x0

Bits	Name	Access	Description	Reset
			width, the timeout period is truncated to fit to the counter width. This affects only the non-user specified values as users are limited to these boundaries during configuration. The range of values available for a 32-bit watchdog counter are: TOR_MODE = 0 T = 2^(16 + WDT_TORR) TOR_MODE = 1 T = WDT_TOC <<(WDT_TORR +1)	
7:4	WDT_ITORR		Initial TimeOut Period ITOR_MODE = 0 T = 2^(16 + WDT_ITORR) ITOR_MODE = 1 T = WDT_TOC <<(WDT_ITORR +1)	
31:4	Reserved			

WDT_CCVR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	WDT_CCVR	RO	This register, when read, is the current value of the internal counter.	

WDT_CRR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	WDT_CRR	R/W	[7:0] Counter Restart Register This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.	0x0
31:8	Reserved			

WDT_STAT

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	WDT_STAT	RO	[0] Interrupt Status Register This register shows the interrupt status of the WDT. 1 = Interrupt is active regardless of polarity. 0 = Interrupt is inactive.	
31:1	Reserved			

WDT_EOI

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	WDT_EOI	RO	[0] Interrupt Clear Register Clears the watchdog interrupt. This can	

Bits	Name	Access	Description	Reset
			be used to clear the interrupt without restarting the watchdog counter.	
31:1	Reserved			

WDT_TOC

Offset Address: 0x01C

Bits	Name	Access	Description	Reset
15:0	WDT_TOC	R/W	Time out counter	0x0

12 实时时钟

12.1 概述

实时时钟 RTC (Real Time Clock) 是芯片内一个独立供电的模块，内部包含一个 32KHz 振荡器与上电复位(Power-On-Reset/POR)的子模块，可用于时间显示及定时报警(Alarm)产生。此外由硬件状态机提供芯片上电、下电及复位的触发与时序控制。

12.2 特点

RTC 具有以下特点：

- 提供芯片系统复位信号 (System Reset)
- 提供 32,768 Hz 计数时钟 (误差 < ±1%)
- 提供 32-bit 秒钟计数器及硬件校准电路，可让秒钟精准度达到 5ppm 水平
- 支持报警时钟配置，并产生报警中断
- 提供 2KB 空间 SRAM，可保存软件代码或暂存数据
- 支持电池低压检测，并产生中断
- 支持按键触发芯片从休眠唤醒
- 支持软件触发芯片休眠、复位或过热重启
- 支持 Watchdog 触发芯片系统复位
- 支持报警触发芯片从休眠唤醒
- 上下电时序、复位时间间隔可配置
- 提供 1 个超低功耗模拟秒钟计数器 (32 bit to count from 1970~2106 year)

12.3 功能描述

RTC 是芯片内一个独立供电的恒电模块，当 RTC 首次上电时，内部 POR 模块产生一个低电平脉冲，32KHz 振荡器开始起振。POR 低电平维持时间大于 13 个 32KHz 时钟周期，RTC 进入初始状态。当状态机检测到电池电压处于正常状态，则开始依默认值时序完成芯片上电流程并解除系统复位信号。软件在首次开机后需要将 RTC 初始化，并配置计数初始值。当系统要执行关机或进入休眠时，可通过配置系统控制寄存器(RTC_CTRL)触发 RTC 状态机依配置时序完成芯片下电流程。

当芯片处于下电关机或休眠状态时，只要维持 RTC 供电就会持续工作状态，以保存必要的软件代码或用户数据于 SRAM 和信息寄存器 (RTC_INFO0 ~ RTC_INFO3, RTC_NOPOR_INFO0 ~ RTC_NOPOR_INFO3)；计数器会持续计数，同时也会侦测按键是否

触发芯片开机或自休眠唤醒，当收到触发则会由状态机依配置时序完成芯片上电流程并解除系统复位信号。处理器内核在重启后，软件可藉由回读先前写入信息寄存器的内容来判断芯片状态，RTC 内也提供两个状态寄存器 (RTC_ST_ON_REASON, RTC_ST_OFF_REASON)，分别记录前一次芯片下电或上电、复位的触发条件，提供软件更细致内容，例如是否曾经发生非预期事件：强制复位、芯片过热或电池/电源脱落等。另外，当 RTC 收到 Watchdog 时，会触发状态机依配置时序发出系统复位信号，将芯片重启。

RTC 的计数时钟采用 32KHz 时钟，运行基于一个 32-bit 的加法计数器提供秒计数，计数初值由寄存器 RTC_SET_SEC_CNTR_VALUE 载入。软件可通过寄存器 RTC_SEC_CNTR_VALUE 回读秒数值，转换为具体的年、月、日、时、分。

32KHz 时钟及秒脉衝週期可經由软件流程做校准(calibration)，或開啟硬件模塊周期性执行自动校准。

软件可通过配置 32-bit 寄存器 RTC_ALARM_TIME，并将 RTC_ALARM_ENABLE 写 1 使能 Alarm，当秒计数值 RTC_SEC_CNTR_VALUE 递加到与 RTC_ALARM_TIME 值相等时，RTC 将产生报警中断。中断状态会维持直到 RTC_ALARM_ENABLE 写 0。

此外 RTC 提供电池低压检测，当电池电压低于一定水平时，RTC 将产生中断，软件收到低压中断后，可即时执行关机程序并触发 RTC 完成下电流程，以防止系统发生异常错误。

12.4 工作方式

12.4.1 计数时钟频率

RTC 的秒计数器，计数最大时间为：

2^32 = 49710 天 = 136 年

12.4.2 RTC 复位

RTC 作为芯片上电、下电控制单元，本身并无法单独被软复位，除了首次上电的 POR 外，可支持在异常时，透过按键 RSTN 强制全芯片复位(包含 RTC)，RSTN 按键解除后，RTC 所有寄存器恢复为默认值，状态机也回到初始。若状态机检测到电池电压处于正常状态，则开始依默认值时序完成芯片上电流程并解除系统复位信号。

12.4.3 RTC 初始化

RTC 在芯片第一次上电后，系统需要将 RTC 初始化。不过在初始化之前，需要先对 32KHz 振荡器时钟及秒时间进行校准。校准模块使用 25MHz 晶振时钟采样 32KHz 时钟，配合软件流程操作，在粗调(coarse tune)模式下，以 25MHz 晶振时钟采样一个 32KHz 时钟周期，软件依采样脉冲数调整配置寄存器 RTC_ANA_CALIB[8:0]，加快或减慢 32KHz 振荡时钟周期时间，以提高 32KHz 时钟精准度。完成粗调后可进一步进入细调(fine tune)模式，默认值以 25MHz 晶振时钟采样 256 个 32KHz 时钟周期，软件依采样脉冲数取平均值，可取得 1 秒钟计数 32KHz 时钟所需脉冲数，并将值写入寄存器 RTC_SEC_PULSE_GEN_INT、RTC_SEC_PULSE_GEN_FRAC，完成秒校准。

32KHz 时钟粗调校准软件流程如下：

1. 配置寄存器 RTC_ANA_SEL_FTUNE 為 0，RTC_ANA_CALIB 初始值為 0x100
2. 以下利用 bindary search 方式實現校准：
   FTUNE = RTC_ANA_CALIB; offset = 0x80
3. 配置寄存器 RTC_FC_COARSE_EN 为 1，开始粗调。轮询 RTC_FC_COARSE_TIME 的值，直到大于前一次读取值後，配置 RTC_FC_COARSE_EN 為 0
4. 读取 RTC_FC_COARSE_VALUE，取得 25MHz 时钟采样一个 32KHz 时钟周期的计数值。
   if (RTC_FC_COARSE_VALUE > 770) FTUNE = FTUNE + offset;
   if (RTC_FC_COARSE_VALUE < 755) FTUNE = FTUNE - offset;
   将 FTUNE 值寫回寄存器 RTC_ANA_CALIB
   offset = offset >> 1;
5. 当 RTC_FC_COARSE_VALUE 的值介于 755~770 之间时，32KHz 时钟精度已达到 32,768Hz ± 1%之内，结束粗调。否则等待 0.5ms，重复步骤 3~5，最多重复 8 次即可完成

32KHz 时钟细调校准软件流程如下：

1. 配置寄存器 RTC_SEL_SEC_PULSE 為 0。配置 RTC_FC_FINE_EN 为 1，开始细调
2. 轮询 RTC_FC_FINE_TIME 的值，直到大于前一次读取值
3. 读取 RTC_FC_FINE_VALUE，取得 25MHz 时钟采样 256 个 32KHz 时钟周期的计数值
4. 32KHz 时钟频率可由以下算式取得：
   Frequency = 256 / (RTC_FC_FINE_VALUE x 40ns)
   例如：256 / (195310 x 40) = 32768.4194357
5. 取整数部份 32768，写入寄存器 RTC_SEC_PULSE_GEN_INT，小数部份取 8-bit = 0.4194357 x 256 = 107，写入寄存器 RTC_SEC_PULSE_GEN_FRAC
6. 配置 RTC_FC_FINE_EN 为 0，结束细调

时钟校准流程可视系统需要，由软件做一次性执行或周期性执行。除了软件校准流程

外，RTC 也同时支持硬件周期性执行自动校准。

完成时钟校准后，进一步初始化 RTC。以下只列出必要的初始化过程，其余大部份参数寄存器，仅在需要优化芯片上下电或开关机时序时才有必要配置，一般建议使用默认值。

1. 配置寄存器 RTC_POR_DB_MAGIC_KEY 值 0x5AF0，使能 POR 去抖动(debounce)，以防止 RTC 模块供电电压因发生短暂压降而造成 POR 误触发。去抖动时间约 1ms
2. 配置 RTC_SET_SEC_CNTR_VALUE，设置 RTC 时间计数初始值
3. 对 RTC_SET_SEC_CNTR_TRIG 写入 1，将计数初始值加载到 RTC 秒计数器
4. 轮询 RTC_SEC_CNTR_VALUE 寄存器值，直到读取值等于 RTC_SET_SEC_CNTR_VALUE 值
5. 配置 RTC_PWR_DET_COMP[0]为 1，使能电池低压检测，并配置 RTC_PWR_DET_SEL[0]为 1，以发出低压检测中断，当电池电压低于 threshold 值，状态信号进入低电平。可通过配置寄存器 RTC_PWR_DET_COMP[12:8]调整 threshold 值。
6. RTC 首次上電後必須將寄存器 reg_rtcsys_rstn_src_sel 由默認值 0 改配置為 1，使 RTC 子系統在芯片下電後(suspend or powerdown)維持在工作狀態。此寄存器值若為 0，則 RTC 子系統會隨芯片下電被復位。
7. RTC 首次上電後必須將寄存器 RTC_EN_AUTO_POWER_UP 由默認值 1 改配置為 0，若維持默認值則當芯片進入下電狀態(powerdown)，RTC 檢測 PWR_VBAT_DET 為高電平就會自動進入上電狀態。

12.4.4 模拟秒时钟初始化

1. 配置 RTC_MACRO_RG_DEFD = 16’hC80 (32000 个 32KHz 时钟周期)
2. 配置 RTC_MACRO_DA_SOC_READY 为 1
3. 配置 RTC_MACRO_DA_CLEAR_ALL 为 1
4. 配置 RTC_MACRO_DA_CLEAR_ALL 为 0
5. 配置 RTC_MACRO_RG_SET_T 为需求 Counter 值
6. 配置 RTC_MACRO_DA_LATCH_PASS 为 1
7. 配置 RTC_MACRO_DA_LATCH_PASS 为 0
8. 配置 RTC_MACRO_DA_SOC_READY 为 0
9. 读取 RTC_MACRO_RO_T 来得到 counter 值

12.4.5 中断处理

RTC 会发出报警中断及低压检测中断的状态信号，当系统收到报警中断后，表示定时时间到，用户可以执行相对应的自定义操作。将寄存器 RTC_ALARM_ENABLE 设为 0 可

清除中断状态。如果需要继续设定新的定时时间，则将新值写入寄存器 RTC_ALARM_TIME 后，再次将 RTC_ALARM_ENABLE 设为 1。

12.4.6 休眠与唤醒

系统软件通过配置寄存器 req_suspend 为 1，可使芯片下电进入休眠状态(suspend)，配置寄存器 RTC_EN_PWR_WAKEUP 选择可触发芯片唤醒(wakeup)的来源。需注意在配置 req_suspend 之前，必需先将寄存器 RTC_PG_REG 写 0，让 DDR IO 进入恒定状态(retent)，以避免芯片在下电或上电过程中因误动作造成 DDR 资料遭到破坏。当芯片唤醒后，系统软件必需在执行 DDR 初始化流程之前，将寄存器 RTC_PG_REG 写 1，解除 DDR IO 的保护状态。

此外若要采用按键(PWR_ON, PWR_BUTTON, PWR_WAKEUP)唤醒，在进入休眠状态前，必需先配置相关 IO PINMUX 寄存器，将 IO 锁定为 RTC input function。

12.4.7 关电与上电

系统软件通过配置寄存器 req_shdn 为 1，可使芯片包含 DDR 下电进入关机状态(poweroff)，配置寄存器 RTC_EN_PWR_UP 选择可触发芯片上电开机(powerup)的来源。

12.5 RTC 寄存器概览

RTC 寄存器包含多个部份，RTC_CORE_REG、RTC_MACRO_REG 和 RTC_CTRL_REG，寄存器基址不同，都通过总线存取。

RTC_CORE_REG 寄存器概览如 表格 12-1 所示。

表格 12-1 RTC_REG 寄存器概览 (基址 0x05026000)

Name	Address Offset	Description
RTC_ANA_CALIB	0x000	32K 振荡器控制
RTC_SEC_PULSE_GEN	0x004	秒脉冲产生器整数位与小数位
RTC_ALARM_TIME	0x008	设定定时报警时间
RTC_ALARM_ENABLE	0x00c	使能报警
RTC_SET_SEC_CNTR_VALUE	0x010	设定秒计数器值
RTC_SET_SEC_CNTR_TRIG	0x014	加载秒计数器值
RTC_SEC_CNTR_VALUE	0x018	读取目前秒计数器值
RTC_INFO0	0x01c	信息寄存器 0
RTC_INFO1	0x020	信息寄存器 1

Name	Address Offset	Description
RTC_INFO2	0x024	信息寄存器 2
RTC_INFO3	0x028	信息寄存器 3
RTC_NOPOR_INFO0	0x02c	无复位信息寄存器 0
RTC_NOPOR_INFO1	0x030	无复位信息寄存器 1
RTC_NOPOR_INFO2	0x034	无复位信息寄存器 2
RTC_NOPOR_INFO3	0x038	无复位信息寄存器 3
RTC_DB_PWR_VBAT_DET	0x040	PWR_VBAT_DET 去抖动时间
RTC_DB_BUTTON1	0x048	PWR_BUTTON1 去抖动时间
RTC_DB_PWR_ON	0x04c	PWR_ON 去抖动时间
RTC_7SEC_RESET	0x050	設定長按 PWR_BUTTON 秒數強制 reset
RTC_THM_SHDN_AUTO_REBOOT	0x064	选择 REQ_THM_SHDN 动作
RTC_POR_DB_MAGIC_KEY	0x068	使能 POR 长时去抖动
RTC_DB_SEL_PWR	0x06c	选择 PWR_BUTTON 去抖动模式
RTC_UP_SEQ0	0x070	上电 PWR_SEQ0 输出时序
RTC_UP_SEQ1	0x074	上电 PWR_SEQ1 输出时序
RTC_UP_SEQ2	0x078	上电 PWR_SEQ2 输出时序
RTC_UP_SEQ3	0x07c	上电 PWR_SEQ3 输出时序
RTC_UP_IF_EN	0x080	上电 ISO 解除时序
RTC_UP_RSTN	0x084	上电系统复位解除时序
RTC_UP_MAX	0x088	上电流程完成时序
RTC_DN_SEQ0	0x090	下电 PWR_SEQ0 输出时序
RTC_DN_SEQ1	0x094	下电 PWR_SEQ1 输出时序
RTC_DN_SEQ2	0x098	下电 PWR_SEQ2 输出时序
RTC_DN_SEQ3	0x09c	下电 PWR_SEQ3 输出时序
RTC_DN_IF_EN	0x0a0	下电 ISO 打开时序
RTC_DN_RSTN	0x0a4	下电系统复位发出时序
RTC_DN_MAX	0x0a8	下电流程完成时序
RTC_PWR_CYC_MAX	0x0b0	Power-cycle 完成时序
RTC_WARM_RST_MAX	0x0b4	Warm-reset 完成时序
RTC_EN_7SEC_RST	0x0b8	設定 PWR_BUTTON1 7SEC reset 模式
RTC_EN_PWR_WAKEUP	0x0bc	设定休眠唤醒来源
RTC_EN_SHDN_REQ	0x0c0	使能 REQ_SHDN
RTC_EN_THM_SHDN	0x0c4	使能 REQ_THM_SHDN
RTC_EN_PWR_CYC_REQ	0x0c8	使能 REQ_PWR_CYC
RTC_EN_WARM_RST_REQ	0x0cc	使能 REQ_WARM_RST
RTC_EN_PWR_VBAT_DET	0x0d0	使能状态机参考 PWR_VBAT_DET
FSM_STATE	0x0d4	RTC 状态机值
RTC_EN_WDG_RST_REQ	0x0e0	使能 REQ_WDG_RST
RTC_EN_SUSPEND_REQ	0x0e4	使能 REQ_SUSPEND
RTC_DB_REQ_WDG_RST	0x0e8	REQ_WDG_RST 去抖动时间
RTC_DB_REQ_SUSPEND	0x0ec	REQ_SUSPEND 去抖动时间
RTC_PG_REG	0x0f0	Power Good 寄存器
RTC_ST_ON_REASON	0x0f8	上电状态寄存器
RTC_ST_OFF_REASON	0x0fc	下电状态寄存器
RTC_EN_WAKEUP_REQ	0x120	使能 REQ_WAKEUP
RTC_PWR_WAKEUP_POLARITY	0x128	選擇 PWR_WAKEUP 低電平
RTC_DB_SEL_REQ	0x130	选择去抖动模式
RTC_PWR_DET_SEL	0x140	选择低电压检测信号来源

RTC_MACRO_REG 寄存器概览如 表格 12-3 所示。

表格 12-2 RTC_MACRO_REG 寄存器概览 (基址 0x05026400)

Name	Address Offset	Description
RTC_PWR_DET_COMP	0x044	低电压检测控制
RTC_MACRO_DA_CLEAR_ALL	0x080	DA_CLEAR_ALL
RTC_MACRO_DA_SET_ALL	0x084	DA_SEL_ALL
RTC_MACRO_DA_LATCH_PASS	0x088	DA_LATCH_PASS
RTC_MACRO_DA_SOC_READY	0x08c	DA_SOC_READY
RTC_MACRO_PD_SLDO	0x090	PD_SLDO
RTC_MACRO_RG_DEFD	0x094	RG_DEFD
RTC_MACRO_RG_SET_T	0x098	RG_SET_T
RTC_MACRO_RO_CLK_STOP	0x0a0	RO_CLK_STOP
RTC_MACRO_RO_DEFQ	0x0a4	RO_DEFQ
RTC_MACRO_RO_T	0x0a8	RO_T

RTC_CTRL 寄存器概览如 表格 12-3 所示。

表格 12-3 RTC_CTRL 寄存器概览 (基址 0x05025000)

Name	Address Offset	Description
rtc_ctrl_unlockkey	0x004	rtc_ctrl_unlockkey
rtc_ctrl0	0x008	rtc_ctrl0
rtc_ctrl_status0	0x00c	rtc_ctrl_status0
rtc_ctrl_status1	0x010	rtc_ctrl_status1
rtc_ctrl_status2gpio	0x014	rtc_ctrl_status2gpio
rtcsys_rst_ctrl	0x018	rtcsys_rst_ctrl
rtcsys_clkmux	0x01c	rtcsys_clkmux
rtcsys_mcu51_ctrl0	0x020	rtcsys_mcu51_ctrl0
rtcsys_mcu51_ctrl1	0x024	rtcsys_mcu51_ctrl1
rtcsys_pmu	0x028	rtcsys_pmu
rtcsys_status	0x02c	rtcsys_status
rtcsys_clkbyp	0x030	rtcsys_clkbyp
rtcsys_clk_en	0x034	rtcsys_clk_en
rtcsys_wkup_ctrl	0x038	rtcsys_wkup_ctrl
rtcsys_clkdiv	0x03c	rtcsys_clkdiv
fc_coarse_en	0x040	fc_coarse_en
fc_coarse_cal	0x044	fc_coarse_cal
fc_fine_en	0x048	fc_fine_en
fc_fine_period	0x04c	fc_fine_period
fc_fine_cal	0x050	fc_fine_cal
rtcsys_pmu2	0x054	rtcsys_pmu2
rtcsys_clkdiv1	0x058	rtcsys_clkdiv1
rtcsys_mcu51_dbg	0x05c	rtcsys_mcu51_dbg
sw_reg0	0x060	sw_reg0
sw_reg1_por	0x064	sw_reg1_por
fab_lp_ctrl	0x068	fab_lp_ctrl
fab_option	0x06c	fab_option
rtcsys_mcu51_ictrl1	0x07c	rtcsys_mcu51_ictrl1
rtc_ip_pwr_req	0x080	rtc_ip_pwr_req
rtc_ip_iso_ctrl	0x084	rtc_ip_iso_ctrl

Name	Address Offset	Description
rtcsys_spare_reg0	0x088	rtcsys_spare_reg0
rtcsys_spare_reg1	0x08c	rtcsys_spare_reg1
rtcsys_spare_ro	0x090	rtcsys_spare_ro
rtcsys_wkup_ctrl1	0x094	rtcsys_wkup_ctrl1
rtcsys_sram_ctrl	0x098	rtcsys_sram_ctrl
rtcsys_io_ctrl	0x09c	rtcsys_io_ctrl
rtcsys_wdt_ctrl	0x0a0	rtcsys_wdt_ctrl
rtcsys_irrx_clk_ctrl	0x0a4	rtcsys_irrx_clk_ctrl
rtcsys_rtc_wkup_ctrl	0x0a8	rtcsys_rtc_wkup_ctrl
rtcsys_por_rst_ctrl	0x0ac	rtcsys_por_rst_ctrl

12.6 RTC 寄存器描述

12.6.1 RTC_CORE_REG

RTC_ANA_CALIB

Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	RTC_ANA_CALIB	R/W	调整类比模块 32K 振荡器频率	0x100
17:16	RTC_ANA_ISEL	R/W	调整类比模块 32K XTAL 电流 00 = 2uA, 01 = 1.5uA, 11 = 0.5uA	0x3
30:18	Reserved			
31	RTC_ANA_SEL_FTUNE	R/W	选择 32K OSC 校准值来源 0 = 由寄存器 RTC_ANA_CALIB 控制 1 = 由硬件电路控制	0x1

RTC_SEC_PULSE_GEN

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	RTC_SEC_PULSE_GEN_FRAC	R/W	秒脉冲产生器小数部位	0x0
23:8	RTC_SEC_PULSE_GEN_INT	R/W	秒脉冲产生器整数部位 当计数器递加值大于整数位值时，产生秒脉冲信号，使秒计数器加 1。	0x8000
30:24	Reserved			
31	RTC_SEL_SEC_PULSE	R/W	选择秒脉冲信号来源 0 = 秒脉冲信号于内部产生 1 = 秒脉冲信号由外部硬件电路产生 当设为 1 时，寄存器 RTC_SEL_PULSE_GEN_FRAC & RTC_SEL_PULSE_GEN_INT 没有作用。	0x1

RTC_ALARM_TIME

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	RTC_ALARM_TIME	R/W	设定定时报警时间	0xffffffff

RTC_ALARM_ENABLE

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	RTC_ALARM_ENABLE	R/W	报警使能 设 1 使能报警，设 0 关闭报警或清除报警中断状态。	0x0
31:1	Reserved			

RTC_SET_SEC_CNTR_VALUE

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RTC_SET_SEC_CNTR_VALUE	R/W	设定秒计数器值	0x0

RTC_SET_SEC_CNTR_TRIG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	RTC_SET_SEC_CNTR_TRIG	W1C	加载秒计数器值 设 1 以让 RTC_SET_SEC_CNTR_VALUE 设定值生效，写 1 后寄存器会自动清为 0。	
31:1	Reserved			

RTC_SEC_CNTR_VALUE

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RTC_SEC_CNTR_VALUE	RO	读取目前秒计数器值	

RTC_INFO0

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RTC_INFO0	R/W	信息寄存器 0	0xABCD 1234

RTC_INFO1

Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	RTC_INFO1	R/W	信息寄存器 1	0xDEAD BEEF

RTC_INFO2

Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	RTC_INFO2	R/W	信息寄存器 2	0xABCD 1234

RTC_INFO3

Offset Address: 0x028

Bits	Name	Access	Description	Reset
31:0	RTC_INFO3	R/W	信息寄存器 3	0xDEAD BEEF

RTC_NOPOR_INFO0

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO0	R/W	无复位信息寄存器 0	Random

RTC_NOPOR_INFO1

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO1	R/W	无复位信息寄存器 1	Random

RTC_NOPOR_INFO2

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO2	R/W	无复位信息寄存器 2	Random

RTC_NOPOR_INFO3

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO3	R/W	无复位信息寄存器 3	Random

RTC_APB_BUSY_SEL

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	Reserved	R/W		
4	rtc_apb_32k_busy_sel	R/W	选择 rtc pclk busy 信号来源 (当 busy 时让 pclk 保持全速) 0 = pclk busy 信号由硬件电路产生 1 = pclk busy 信号由寄存器 rtc_apb_32k_force_busy 控制	0x0
7:5	Reserved			
8	rtc_apb_32k_force_busy	R/W	1 = pclk 永远维持全速运作 0 = pclk 只在 psel 时回到全速运作	0x0
31:9	Reserved			

RTC_DB_PWR_VBAT_DET

Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_VBAT_DET	R/W	PWR_VBAT_DET 去抖动时间 (单位: 32K 时钟)	0x2
31:16	Reserved			

RTC_DB_BUTTON1

Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	RTC_DB_BUTTON1	R/W	PWR_BUTTON1 去抖动时间 (单位: 32K 时钟) 默认值 0x100 约为 8ms。	0x100
31:16	Reserved			

RTC_DB_PWR_ON

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_ON	R/W	PWR_ON 去抖动时间 (单位: 32K 时钟)	0x100
31:16	Reserved			

RTC_7SEC_RESET

Offset Address: 0x050

Bits	Name	Access	Description	Reset
7:0	RTC_7SEC_RESET	R/W	長按 PWR_BUTTON1 reset 去抖動時間 (單位: 秒) 此寄存器只會被 POR	0x7
15:8	Reserved			
31:16	RTC_7SEC_UNLOCK_KEY	WO	同時寫入 0xDC78 可解除 RTC_7SEC_RESET 防寫	0x0

RTC_THM_SHDN_AUTO_REBOOT

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	RTC_THM_SHDN_AUTO_REBOOT	R/W	选择收到 REQ_THM_SHDN 时： 0 = 启动下电流程 1 = 启动 Power-cycle 流程 (先下电再上电)	0x0
31:1	Reserved			

RTC_POR_DB_MAGIC_KEY

Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	RTC_POR_DB_MAGIC_KEY	R/W	写入 0x5AF0 会使时 POR 去抖动(约 1ms)	Random
31:16	Reserved			

RTC_DB_SEL_PWR

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	Reserved			
1	DB_SEL_PWR_BUTTON1	R/W	选择 PWR_BUTTON1 去抖动模式 0 = 状态机采用 PWR_BUTTON1 去抖动信号下降沿触发 1 = 状态机采用 PWR_BUTTON1 去抖动信号低电平触发	0x1
2	DB_SEL_PWR_ON	R/W	选择 PWR_ON 去抖动模式 0 = 状态机采用 PWR_ON 去抖动信号上升沿触发 1 = 状态机采用 PWR_ON 去抖动信号高电平触发	0x1
3	DB_SEL_PWR_WAKEUP0	R/W	选择 PWR_WAKEUP0 去抖动模式 0 = 状态机采用 PWR_WAKEUP0 去抖动信号上升沿触发 1 = 状态机采用 PWR_ WAKEUP0 去抖动信号高电平触发	0x1
4	DB_SEL_PWR_WAKEUP1	R/W	选择 PWR_WAKEUP1 去抖动模式 0 = 状态机采用 PWR_WAKEUP1 去抖动信号上升沿触发 1 = 状态机采用 PWR_ WAKEUP1 去抖动信号高电平触发	0x1
31:5	Reserved			

RTC_UP_SEQ0

Offset Address: 0x070

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ0	R/W	上电流程 PWR_SEQ0 输出由 0 转 1 时间(单位: 32K 时钟)	0x0
31:16	Reserved			

RTC_UP_SEQ1

Offset Address: 0x074

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ1	R/W	上电流程 PWR_SEQ1 输出由 0 转 1 时间(单位: 32K 时钟)	0x40
31:16	Reserved			

RTC_UP_SEQ2

Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ2	R/W	上电流程 PWR_SEQ2 输出由 0 转 1 时间(单位: 32K 时钟)	0x80
31:16	Reserved			

RTC_UP_SEQ3

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ3	R/W	上电流程 PWR_SEQ3 输出由 0 转 1 时间(单位: 32K 时钟)	0xc0
31:16	Reserved			

RTC_UP_IF_EN

Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	RTC_UP_IF_EN	R/W	上电流程解除下电区 ISO 时间(单位: 32K 时钟)	0x100
31:16	Reserved			

RTC_UP_RSTN

Offset Address: 0x084

Bits	Name	Access	Description	Reset
15:0	RTC_UP_RSTN	R/W	上电流程系统复位解除时间(单位: 32K 时钟)	0x140
31:16	Reserved			

RTC_UP_MAX

Offset Address: 0x088

Bits	Name	Access	Description	Reset
15:0	RTC_UP_MAX	R/W	完整上电流程完成时间(单位: 32K 时钟) RTC_UP_SEQ0~RTC_UP_MAX 为上电流程各阶段时序绝对时间，建议使用默认值。	0x180
31:16	Reserved			

RTC_DN_SEQ0

Offset Address: 0x090

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ0	R/W	下电流程 PWR_SEQ0 输出由 1 转 0 时间(单位: 32K 时钟)	0x140
31:16	Reserved			

RTC_DN_SEQ1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ1	R/W	下电流程 PWR_SEQ1 输出由 1 转 0 时间(单位: 32K 时钟)	0x100
31:16	Reserved			

RTC_DN_SEQ2

Offset Address: 0x098

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ2	R/W	下电流程 PWR_SEQ2 输出由 1 转 0 时间(单位: 32K 时钟)	0xc0
31:16	Reserved			

RTC_DN_SEQ3

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ3	R/W	下电流程 PWR_SEQ3 输出由 1 转 0 时间(单位: 32K 时钟)	0x80
31:16	Reserved			

RTC_DN_IF_EN

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
15:0	RTC_DN_IF_EN	R/W	下电流程打开下电区信号 ISO 时间(单位: 32K 时钟)	0x40
31:16	Reserved			

RTC_DN_RSTN

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RTC_DN_RSTN	R/W	下电流程发出系统复位时间(单位: 32K 时钟)	0x0
31:16	Reserved			

RTC_DN_MAX

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
15:0	RTC_DN_MAX	R/W	完整下电或休眠流程完成时间(单位: 32K 时钟) RTC_DN_SEQ0~RTC_DN_MAX 为下电流程各阶段时序绝对时间，建议使用默认值。	0x180
31:16	Reserved			

RTC_PWR_CYC_MAX

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
15:0	RTC_PWR_CYC_MAX	R/W	完整 Power-cycle 流程完成时间(单位: 32K 时钟) Power-cycle 时间包含完整下电流程加上电流程。	0x4000
31:16	Reserved			

RTC_WARM_RST_MAX

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
15:0	RTC_WARM_RST_MAX	R/W	完整 WARM_RESET 流程完成时间(单位: 32K 时钟) 相当于系统复位低电平时间。	0x40
31:16	Reserved			

RTC_EN_7SEC_RST

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
0	RTC_EN_7SEC_RST	R/W	使能長按 PWR_BUTTON1 7 秒觸發 RTC 強制復位	0x0
1	RTC_7SEC_RST_MODE	R/W	7 秒強制復位方式 0 = 低電平模式, 1 = 短脈衝模式 當 7 秒強制復位發生時，選擇產生短脈復位信號或保持復位直到放開 PWR_BUTTON1 按鍵。	0x0
2	DB_SEL_PWR_BUTTON1_7SEC	R/W	0 = 若 PWR_BUTTON1 按鍵已觸發上電流程後重置 7 秒復位計數器 1 = NOP	0x0
3	SEL_7SEC_RST_RTCSYS	R/W	0 = 7 秒強制復位信號會重置 RTC 子系統 sec reset will not reset rtcsys 1 = 不會重置 RTC 子系統	0x1
15:4	Reserved			
31:16	RTC_EN_7SEC_UNLOCK_KEY	WO	同時寫入 0xDC78 可解除[3:0]防寫	0x0

RTC_EN_PWR_WAKEUP

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
6:0	RTC_EN_PWR_WAKEUP	R/W	设定可触发自休眠状态唤醒的来源 0 = 不能触发唤醒 1 = 可以触发唤醒 [0] = PWR_WAKEUP0 [1] = PWR_WAKEUP1 [2] = PWR_ON [3] = REQ_POWERUP [4] = PWR_BUTTON1 [5] = Alarm [6] = REQ_WAKEUP	0x0
7	Reserved			

Bits	Name	Access	Description	Reset
14:8	RTC_EN_PWR_UP	R/W	设定可触发上電開機的来源 0 = 不能触发上電 1 = 可以触发上電 [8] = PWR_WAKEUP0 [9] = PWR_WAKEUP1 [10] = PWR_ON [11] = REQ_POWERUP [12] = PWR_BUTTON1 [13] = Alarm [14] = REQ_WAKEUP	0x14
31:15	Reserved			

RTC_EN_SHDN_REQ

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	RTC_EN_SHDN_REQ	R/W	使能软件请求下电(REQ_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_THM_SHDN

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	RTC_EN_THM_SHDN	R/W	使能请求过热下电或重启 (REQ_THM_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_PWR_CYC_REQ

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_CYC_REQ	R/W	使能请求 Power-cycle (REQ_PWR_CYC) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_WARM_RST_REQ

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
0	RTC_EN_WARM_RST_REQ	R/W	使能请求系统软重启 (REQ_WARM_RST) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_PWR_VBAT_DET

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_VBAT_DET_UP	R/W	使能状态机参考电池低压检测状态 (PWR_VBAT_DET) 0 = disable, 1 = enable 若此值设 1，当任何按键欲触发上电或唤醒时，状态机会判断低电压检测输出值，若低压检测输出为低电平(电池电压过低或无电源)，则 RTC 状态机	0x1

Bits	Name	Access	Description	Reset
			维持目前状态不变。	
1	RTC_EN_PWR_VBAT_DET_DN	R/W	使能电池低压状态触发下电 0 = disable, 1 = enable 若此值设 1，当芯片正处于正常上电或已进入休眠，RTC 状态机会判断低压检测输出值，若低压检测输出由高电平转成低电平时(电池电压过低或电源脱落)，会触发状态机执行下电流程。	0x1
2	RTC_EN_AUTO_POWER_UP	R/W	使能 RTC 狀態機自動進入上電狀態 1 = 當進入 Power-down 而 PWR_VBAT_DET 為高電平時，則自動上電 0 = 當進入 Power-down 時停留此狀態，直到任一上電來源被觸發	0x1
31:3	Reserved			

FSM_STATE

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
3:0	FSM_STATE	RO	RTC 状态机值 4’h0 = ST_OFF (系统下电完成) 4’h1 = ST_UP (上电流程进行中) 4’h2 = ST_DN (下电流程进行中) 4’h3 = ST_ON (系统上电完成) 4’h4 = ST_PWR_CYC2 (Power-cycle 下电流程结束) 4’h6 = ST_PWR_CYC (Power-cycle 下电中) 4’h7 = ST_WARM_RESET (系统复位中) 4’h9 = ST_SUSP (系统已休眠) 4’hB = ST_PRE_SUSP (休眠下电流程进行中)	
31:4	Reserved			

RTC_EN_WDG_RST_REQ

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
0	RTC_EN_WDG_RST_REQ	R/W	使能 Watchdog 请求系统复位 (REQ_WDG_RST) 0 = disable, 1 = enable	0x0
1	RTC_EN_SUS_WDG_RST_REQ	R/W	当休眠状态时，使能 Watchdog 请求系统复位 0 = disable, 1 = enable	0x1
31:2	Reserved			

RTC_EN_SUSPEND_REQ

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
0	RTC_EN_SUSPEND_REQ	R/W	使能请求休眠(REQ_SUSPEND) 0 = disable, 1 = enable	0x0

Bits	Name	Access	Description	Reset
31:1	Reserved			

RTC_PG_REG

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
3:0	RTC_PG_REG	R/W	芯片 Power Good 状态 1 = 芯片有电(Power Good)，IO 信号可通过 0 = 芯片下电，IO 信号保持暂存状态(retent) [0] = 控制 DDR IO [3:1] = reserved 此寄存器信号用来控制芯片与 DDR 的 IO 接口是否为正常通过或为 retent 状态。当系统要进入休眠前，软件必须先将此寄存器值设为 0，让 DDR IO 维持固定状态。当系统自休眠状态唤醒之后，必须先将此寄存器值设为 1，让 DDR 回复正常操作。当进入下电状态时，此寄存器值会自动被清为全 1。	0xF
31:4	Reserved			

RTC_ST_ON_REASON

Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
3:0	ST_ON_REASON_LAST_STATE	RO	RTC 状态机由以下状态回到上电完成(ST_ON) 4'h0 = 由下电(ST_OFF)回到上电 4'h3 = 由上电进入 Power-cycle 或 Warm-reset 再回到上电 4'h9 = 由休眠返回上电 当系统重启后，软件可藉由读取此寄存器判断芯片开机上电原因。	
15:4	Reserved			
31:16	ST_ON_REASON_LAST_INPUT	RO	状态机返回上电状态的触发原因(记录信号状态值) [0] = PWR_VBAT_DET (0: 触发下电) [1] = PWR_ON (1: 按键触发上电) [2] = RTC_EN_AUTO_POWER_UP [3] = PWR_BUTTON1 (0: 按键触发上电) [4] = PWR_BUTTON1_7SEC [5] = PWR_WAKEUP0 (1: 按键触发唤醒) [6] = PWR_WAKEUP1 (1: 按键触发唤醒) [7] = Alarm (1: 发生定时警报) [8] = REQ_PWR_CYC (1: 软件触发 Power-cycle) [9] = REQ_THM_SHDN (1: 软件触发下电/power-cycle) [10] = REQ_WARM_RST (1: 软件触发 reset) [11] = REQ_WDG_RST (1: Watchdog 触	

Bits	Name	Access	Description	Reset
			发 reset) [12] = REQ_SHDN (1: 软件触发下电) [13] = REQ_SUSPEND (1: 软件触发休眠) [14] = REQ_WAKEUP (1: 事件触发唤醒) [15] = REQ_POWERUP	

RTC_ST_OFF_REASON

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
3:0	ST_OFF_REASON_LAST_STATE	RO	RTC 状态机曾由以下状态进入下电完成(ST_OFF) 4'h3 = 由上电(ST_ON)进入下电 4'h9 = 由休眠(ST_SUSP)进入下电 其他 = 7 秒強制復位發生 当系统重启后，软件可藉由读取此寄存器判断芯片前次进入下电原因。	
15:4	Reserved			
31:16	ST_OFF_REASON_LAST_INPUT	RO	状态机曾进入下电状态的触发原因(记录信号状态值) [14:0] 与 ST_ON_REASON_LAST_INPUT 相同 [15] = 0: 曾发生 7 秒強制復位	

RTC_EN_WAKEUP_REQ

Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	RTC_EN_WAKEUP_REQ	R/W	使能事件请求自休眠状态唤醒 0 = disable, 1 = enable	0x0
1	RTC_EN_POWERUP_REQ	R/W	使能事件请求上電 0 = disable, 1 = enable	0x0
31:2	Reserved			

RTC_PWR_WAKEUP_POLARITY

Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	PWR_WAKEUP0_POLARITY	R/W	選擇 PWR_WAKEUP0 有效極性 1 = 高電平有效 0 = 低電平有效	0x1
1	PWR_WAKEUP1_POLARITY	R/W	選擇 PWR_WAKEUP1 有效極性 1 = 高電平有效 0 = 低電平有效	0x1
31:2	Reserved			

RTC_DB_SEL_REQ

Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	DB_SEL_REQ_SHDN	R/W	选择软件信号 REQ_SHDN 去抖动模式 0 = 由寄存器值上升沿触发 1 = 由寄存器脉冲信号触发	0x1
1	DB_SEL_REQ_THM_SHDN	R/W	选择信号 REQ_THM_SHDN 去抖动模	0x1

Bits	Name	Access	Description	Reset
			式 0 = 由信号高位触发 1 = 由信号上升沿解发	
2	DB_SEL_REQ_PWR_CYC	R/W	选择软件信号 REQ_PWR_CYC 去抖动模式 0 = 由寄存器值上升沿触发 1 = 由寄存器脉冲信号触发	0x1
3	DB_SEL_REQ_WARM_RST	R/W	选择软件信号 REQ_WARM_RST 去抖动模式 0 = 由寄存器值上升沿触发 1 = 由寄存器脉冲信号触发	0x1
4	DB_SEL_REQ_WDG_RST	R/W	选择信号 REQ_WDG_RST 去抖动模式 0 = 由信号高位触发 1 = 由信号上升沿解发	0x1
5	DB_SEL_REQ_SUSPEND	R/W	选择软件信号 REQ_SUSPEND 去抖动模式 0 = 由寄存器值上升沿触发 1 = 由寄存器脉冲信号触发	0x1
6	DB_SEL_REQ_WAKEUP	R/W	选择信号 REQ_WAKEUP 去抖动模式 0 = 由信号高位触发 1 = 由信号上升沿解发	0x1
7	DB_SEL_REQ_POWERUP	R/W	选择信号 REQ_POWERUP 去抖动模式 0 = 由信号高位触发 1 = 由信号上升沿解发	0x1
31:8	Reserved			

RTC_PWR_DET_SEL

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	pwr_det_o_sel_comp	R/W	选择低电压侦测的状态信号输出来源 0 = 直接来自 IO PWR_VBAT_DET 1 = 来自类比低压检测电路输出 低压检测状态值可由寄存器 RTC_CTRL_STATUS0[0]读出	0x0
1	pwr_det_i_sel_comp	R/W	选择 RTC 状态机低电压触发下电信号来源 0 = 直接来自 IO PWR_VBAT_DET 1 = 来自类比低压检测电路输出	0x0
31:2	Reserved			

12.6.2 RTC_MACRO_REG

RTC_PWR_DET_COMP

Offset Address: 0x44

Bits	Name	Access	Description	Reset
0	pwr_det_comp_enable	R/W	使能类比模块低电压检测 1 = enable 0 = disable	0x0

Bits	Name	Access	Description	Reset
7:1	Reserved			
12:8	pwr_det_comp_sel	R/W	设置低电压检测电压比较 threshold Threshold = 1.20V + (pwr_det_comp_sel * 12.5mV)	0xf
31:13	Reserved			

RTC_MACRO_DA_CLEAR_ALL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	DA_CLEAR_ALL	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_SET_ALL

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	DA_SEL_ALL	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_LATCH_PASS

Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	DA_LATCH_PASS	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_SOC_READY

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	DA_SOC_READY	R/W		0x0
31:1	Reserved			

RTC_MACRO_PD_SLDO

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	PD_SLDO	R/W		0x0
31:1	Reserved			

RTC_MACRO_RG_DEFD

Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RG_DEFD	R/W		0x7fff
31:16	Reserved			

RTC_MACRO_RG_SET_T

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	RG_SET_T	R/W		0x0

RTC_MACRO_RO_CLK_STOP

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	RO_CLK_STOP	RO		
31:1	Reserved			

RTC_MACRO_RO_DEFQ

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RO_DEFQ	RO		
31:16	Reserved			

RTC_MACRO_RO_T

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	RO_T	RO		

12.6.3 RTC_CTRL

RTC_CTRL0_UNLOCKKEY

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	rtc_ctrl0_unlockkey	R/W	要配置寄存器 RTC_CTRL0 之前，必须先对此寄存器写值 0xAB18 以解锁防写 若 unlockkey_clear 设为 1，则对 RTC_CTRL0 做一次写入动作后，此寄存器值会自动清为 0，RTC_CTRL0 回复防写。	0x0000
31:16	Reserved			

RTC_CTRL0

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	req_shdn	R/W	请求下电 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_SHDN_REQ 必须先设为 1 才会有效。	0x0
1	req_sw_thm_shdn	R/W	软件模式请求过热下电或重启 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_THM_SHDN 必须先设为 1 才会有效。	0x0
2	hw_thm_shdn_en	R/W	使能硬件模式请求过热下电或重启 0 = diable, 1 = enable	0x0
3	req_pwr_cyc	R/W	请求 Power-cycle 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_PWR_CYC_REQ 必须先设为 1 才会有效。	0x0
4	req_warm_rst	R/W	请求 Warm-reset 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_WARM_RST_REQ 必须先设为 1 才会有效。	0x0

Bits	Name	Access	Description	Reset
5	req_sw_wdg_rst	R/W	软件模式请求 Watchdog 复位 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_WDG_RST_REQ 必须先设为 1 才会有效。	0x0
6	hw_wdg_rst_en	R/W	使能硬件模式请求 Watchdog 复位 0 = diable, 1 = enable	0x0
7	req_suspend	R/W	请求休眠 0 = 无动作, 1 = 对 RTC 发出请求 寄存器 RTC_EN_SUSPEND_REQ 必须先设为 1 才会有效。	0x0
8	unlockkey_clear	R/W	使能自动清除寄存器解锁	0x0
9	Reserved			
10	reg_rtc_mode	R/W	32K 時鐘來源 0 = OSC32K, 1 = XTAL32K	0x0
11	reg_clk32k_cg_en	R/W	32K 時鐘開關 0 = 關閉, 1 = 打開	0x1
31:12	Reserved			

RTC_CTRL_STATUS0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	rtc_pwr_vbat_det_o	RO	低电压检测状态信号输出	
1	rtc_pwr_button0_o	RO	PWR_BUTTON0 IO 信号输出	
2	rtc_pwr_button1_o	RO	PWR_BUTTON1 IO 信号输出	
3	Reserved			
4	rtc_pwr_on_o	RO	PWR_ON IO 信号输出	
5	rtc_pwr_wakeup0_o	RO	PWR_WAKEUP0 IO 信号输出	
6	rtc_pwr_wakeup1_o	RO	PWR_WAKEUP1 IO 信号输出	
7	rtc_mode_o	RO	RTC_MODE IO 信号输出	
20:8	Reserved			
21	rtc_alarm_o	RO	Alarm 状态	
22	hw_thm_shdn_sta_i	RO	过热重启状态信号	
23	hw_wdg_rst_sta_i	RO	Watchdog 复位状态信号	
24	sys_reset_x_i	RO		
25	cg_en_out_clk_32k	RO		
29:26	rtc_fsm_st	RO	RTC 状態機值	
31:30	Reserved			

RTC_CTRL_STATUS1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	rtc_sec_value_o	RO	RTC 秒计数器值	

rtc_ctrl_status2gpio

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	status2gpio_en	R/W		0x0

Bits	Name	Access	Description	Reset
31:8	Reserved			

rtcsys_rst_ctrl

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_soft_rstn_mcu	R/W	0 : reest MCU	0x0
2	reg_soft_rstn_sdio	R/W	0 : reset SD1	0x1
3	reg_soft_rstn_uart	R/W	0 : reset Uart	0x1
4	reg_soft_rstn_spinor	R/W	0 : reset spinor1	0x1
5	reg_soft_rstn_ictl	R/W	0 : reset dw_ictl	0x1
6	reg_soft_rstn_mbox	R/W	0 : reset mbox	0x1
7	reg_soft_rstn_fab_hs2rtc	R/W	0 : reset hs2rtc	0x1
8	reg_soft_rstn_fab_rtc2ap	R/W	0 : reset rtc2ap	0x1
9	reg_soft_rstn_fab_sram	R/W	0 : reset ahb sram logic	0x1
10	reg_soft_rstn_apb	R/W	no load	0x1
11	reg_soft_rstn_apb_timer	R/W	0 : reset dw timer apb logic	0x1
12	reg_soft_rstn_timer0	R/W	0 : reset dw timer0	0x1
13	reg_soft_rstn_timer1	R/W	0 : reset dw timer1	0x1
14	reg_soft_rstn_osc	R/W	0 : reset osc	0x1
15	reg_soft_rstn_gpio	R/W	0 : reset gpio	0x1
16	reg_soft_rstn_i2c	R/W	0 : reset i2c	0x1
17	reg_soft_rstn_saradc	R/W	0 : reset saradc	0x1
18	reg_soft_rstn_wdt	R/W	0 : reset wdt	0x1
19	reg_soft_rstn_irrx	R/W	0 : reset irrx	0x1
20	reg_soft_rstn_f32kless	R/W	0: reset f32kless	0x1
31:21	Reserved			

rtcsys_clkmux

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
3:0	reg_sdio_clk_mux	R/W	clk_sd1_pre 0 : fpll/4 1: osc_div	0x0
7:4	reg_fab_clk_mux	R/W	clk_fab_pre 0 : 32K, 1: fpll/5, 2: osc_div	0x0
9:8	reg_timer0_clk_mux	R/W	0: xtal 1: 32K	0x0
11:10	reg_timer1_clk_mux	R/W	0: xtal 1: 32K	0x0
13:12	reg_apb_clk_mux	R/W	00 : cgdiv and refer to apbactive 01 : force clk_apb, clk_fab 1:1 (default) 10 : force clk_apb, clk_fab 1:2 11 : force clk_apb, clk_fab 1:4	0x1
15:14	Reserved			
17:16	reg_i2c_clk_mux	R/W	0: xtal 1: osc div	0x0
19:18	reg_sd_mclk_clk_mux	R/W	0: 100Khz from OSC, 1: 32K	0x0
20	reg_saradc_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
21	reg_irrx_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
31:22	Reserved			

rtcsys_mcu51_ctrl0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
4:0	reg_51_rom_addr_size	R/W	Determines how many of the sixteen internal ROM address bits (irom_addr) are used (0 = no internal ROM present);	0xc
5	reg_51_mem_ea_n	R/W	0 : external rom exist, 1: external rom not exist	0x0
6	reg_51_xdata_mode	R/W	0 : fetch xdata with clock gating 1 : fetch xdata wo clock gating (to support 51 timer and 51 uart)	0x0
7	reg_51_rom_addr_def	R/W	0: mercury define , max internal rom = 2^reg_51_rom_addr_size -1 internal rom offset = 4K*reg_51irom_ioffset 1: mars define , max internal rom = 2K*reg_51_rom_addr_size -1 internal rom offset = 2K*reg_51irom_ioffset	0x0
10:8	Reserved			
31:11	reg_51xdata_ioffset0	R/W	Set offset address[31:12] to select mcu8051 boot device	0x05200

rtcsys_mcu51_ctrl1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
4:0	reg_51irom_ioffset	R/W	boot rom offset to rtcsys_sram	0x0
5	Reserved			
9:6	reg_51_pf_mode	R/W	reg_51_pf_mode	0x0
10	Reserved			
31:11	reg_51xdata_doffset0	R/W	Set offset address[31:12] to select mcu8051 xdata	0x05200

rtcsys_pmu

Offset Address: 0x028

Bits	Name	Access	Description	Reset
3:0	Reserved			
4	reg_dis_pmu_ldo_ctrl	R/W	disable pmu ldo ctrl 0: enable pmu to ctrl RTC_LDO sleep mode 1: disable pmu to ctrl RTC_LDO sleep mode	0x0
5	reg_wdt_clkoff_by_pmu	R/W	wdt_clk gate by pmu when mcu into idle mode 1. wdt clock gate by pmu	0x0
6	reg_force_osc_off	R/W	1 : force osc off	0x0
7	reg_force_osc_on	R/W	1 : force osc on	0x0
8	reg_pmu_sleep_mode	R/W	pmu enter light sleep mode when mcu idle 1 : enable pmu light sleep mode when mcu idle (pmu control osc_req/ sram slp) 0 disable pmu light sleep mode	0x0
9	reg_pmu_lowpwr_mode	R/W	mcu_pmu into sleep state when rtc at suspend state & mcu idle & reg_pmu_sleep_mode enable 1 : enable mcu_pmu into sleep mode (trigger rtc ldo step down power) 0 disable mcu_pmu sleep mode	0x0

Bits	Name	Access	Description	Reset
13:10	reg_pmu_stable_cnt	R/W	Stable timer when mcu_pmu leave sleep state, clock unit : 31.25us (32khz), wait for 1~16 tick cycle	0x3
14	reg_xtal_off_by_pmu	R/W	pmu control xtal request 1: xtal request disable by pmu sleep mode	0x0
15	reg_rtcsys_clk25m_req	R/W	xtal request1 for rtcsys 0: disable 25m xtal request1(rtcsys) 1: enable 25m xtal request1 (rtcsys)	0x1
19:16	reg_rtc_vbat_det_db_cnt	R/W	vbat det int debounce time (cycle unit : 32K)	0x2
20	reg_rtc_vbat_det_db_en	R/W	0: disable vbat det int debounce 1: enable vbat det int debounce	0x1
21	reg_ahb_sram_auto_slp_en	R/W	1: enable ahb sram into slp md when bus idle	0x0
23:22	reg_ahb_sram_busy_sel	R/W	2'd0: cs | cs_d1 2'd1: cs | cs_d1 | cs_d2 2'd2: cs | cs_d1 | cs_d2 | cs_d3 3'd3: cs | cs_d1 | cs_d2 | cs_d3 | cs_d4	0x0
24	reg_rtc_stint_clr	W1P	clear rtc state change interrupt	
25	reg_vbat_det_int_clr	W1P	clear vbet det interrupt	
26	reg_rtcsys_clk25m_hw_req	R/W	xtal request1 for rtcsys from hw ip 0: disable 25m xtal request1 from hw ip(rtcsys) 1: enable 25m xtal request1 from hw ip(rtcsys)	0x0
27	Reserved			
28	reg_vbat_det_force_clk	R/W	1: when vbat det happen, change rtcsys bus clock to OSC	0x0
29	reg_mcu_clkoff_by_pmu	R/W	mcu_clk gate by pmu when into idle mode 1. mcu clock gate by pmu	0x1
30	reg_xtal_off_by_susp	R/W	ISO off control xtal request 1: xtal request disable by ISO_OFF	0x0
31	reg_osc_off_by_susp	R/W	ISO off control osc request 1: osc request disable by ISO_OFF	0x0

rtcsys_status

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	reg_rtcsys_status	RO	[0] enable rtc2apb ahb path 0: rtcsys ip can only access 0x05000000+16MB 1: rtcsys ip can access full range address [1] flag of vbat_det_force_clk	

rtcsys_clkbyp

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	reg_clk_byp	R/W	[0] : clk_fab , 0: clk_fab_pre, 1: xtal (default) [1] : clk_sdio, 1: clk_sd1_pre, 1: xtal (default) [31:2]: NA	0xffffffff

rtcsys_clk_en

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_clk_en	R/W	[0]: NA [1]: clk_sd1 (sd1 card clock) [2]: clk_fab_sd1 (sd1 core clock) [3]: clk_mcu [4]: clk_hs2rtc_mst [5]: clk_rtc2ap_slv [6]: clk_spinor1 [7]: clk_fab_sram (AHB sram) [8]: NA [9]: clk_apb_timer [10]: clk_timer0 [11]: clk_timer1 [12]: clk_apb_uart [13]: clk_uart [14]: clk_apb_ictrl [15]: clk_apb_mbox [16]: clk_apb_gpio [17]: clk_apb_osc [18]: clk_gpio_db [19]: clk_apb_i2c [20]: clk_i2c [21]: NA [22]: clk_sd1_tmclk [23]: clk_apb_saradc [24]: clk_saradc [25]: clk_apb_wdt [26]: clk_wdt [27]: clk_irrx [31:28]: NA	0xffffffff

rtcsys_wkup_ctrl

Offset Address: 0x038

Bits	Name	Access	Description	Reset
14:0	reg_rtcsys_wkint_mask	R/W	mask int to RTC_CORE.REQ_WAKEUP/ MCU_PMU [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15	reg_vbat_det_wkup_mask	R/W	1: mask vbat det int	0x1
16	reg_sw_wkint_req	R/W	mcu sw wakeup interrupt to RTC_CORE 1: interrupt active	0x0
23:17	Reserved			
24	reg_wkint2rtc_mask	R/W	1: mask wakeup int (rtcsys int) to RTC core	0x1
31:25	Reserved			

rtcsys_clkdiv

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	reg_div_clk_osc_fab_div_val	R/W	Clock Divider Factor	0x1
4	reg_div_clk_osc_fab_dis	R/W	Clock gate	0x0
5	reg_div_clk_osc_fab_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
15:6	Reserved			
19:16	reg_div_clk_osc_i2c_div_val	R/W	Clock Divider Factor	0x1
20	reg_div_clk_osc_i2c_dis	R/W	Clock gate	0x0
21	reg_div_clk_osc_i2c_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
23:22	Reserved			
29:24	reg_div_clk_osc_saradc_div_val	R/W	Clock Divider Factor	0x1
30	reg_div_clk_osc_saradc_dis	R/W	Clock gate	0x0
31	reg_div_clk_osc_saradc_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0

fc_coarse_en

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	fc_coarse_en	R/W	使能 32K 粗调 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_coarse_cal

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	fc_coarse_value	RO	32K 粗调计数器值 (单位: 25MHz 时钟) 25MHz 时钟计数一个 32K 时钟周期。	
31:16	fc_coarse_time	RO	32K 粗调完成次数	

fc_fine_en

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	fc_fine_en	R/W	使能 32K 细调 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_fine_period

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	fc_fine_period	R/W	32K 细调计数周期 (单位: 32K 时钟) 设定每次使用 25MHz 时钟计数几个 32K 时钟周期。	0x0100
31:16	Reserved			

fc_fine_cal

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	fc_fine_value	RO	32K 细调计数器值 (单位: 25MHz 时钟) 25MHz 时钟计数一个 fc_fine_period	

Bits	Name	Access	Description	Reset
			周期。	
31:24	fc_fine_time	RO	32K 细调完成次数	

rtcsys_pmu2

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	reg_rtc_sys_wkint_db_en	R/W	PMU wakeup int debounce enable	0x1
4:1	reg_rtc_sys_wkint_db_cnt	R/W	PMU wakeup int debounce cycle (32K)	0x2
31:5	Reserved			

rtcsys_clkdiv1

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_div_clk_osc_irrx_div_val	R/W	Clock Divider Factor	0x0
22	Reserved			
23	reg_div_clk_osc_irrx_dis	R/W	Clock gate	0x1
31:24	Reserved			

rtcsys_mcu51_dbg

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
3:0	reg_51_dbg_sel	R/W	select mcu51 debug bus (check mcu design review ppt)	0x0
4	reg_51_dbg_snap_shot	W1P	snap shot mcu51 internal register to dbg register (reg_rtcsys_dbg)	
5	reg_51_dbg_step_en	R/W	0: disable mcu debug function 1: enable mcu debug function, and mcu stop at current PC	0x0
6	reg_51_dbg_step	W1P	1: mcu jump to next PC	
7	reg_51_dbg_jump	W1P	1: mcu jump to target pc value (reg_51_dbg_jump2pc)	
15:8	Reserved			
31:16	reg_51_dbg_jump2pc	R/W	16 bit mcu target pc value	0x0

sw_reg0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	sw_reg0	R/W	reg for SW	0x0
31:8	Reserved			

sw_reg1_por

Offset Address: 0x064

Bits	Name	Access	Description	Reset
7:0	sw_reg1_por	R/W	reg for SW could only be reset by power reset	0x0
31:8	Reserved			

fab_lp_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
7:0	rtcsys_fab_busy_sel	R/W	select signal to request sys_ctrl to speed up fab clock	0xDF
9:8	rtcsys_fab_busy_ctrl	R/W	rtcsys_fab_busy signal is combi or register out	0x0
11:10	apdbg_busy_ctrl	R/W	apdbg_busy signal is combi or register out	0x0
13:12	reg_apb_busy_ctrl	R/W	apb bridge_busy signal is combi or register out	0x3
15:14	reg_mcu_busy_ctrl	R/W	mcu_busy signal is combi or register out	0x3
31:16	Reserved			

rtcsys_mcu51_ictrl1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	reg_51_int1_src_mask	R/W	select rtcsys_int src to mcu int1_n 1: mask, 0: un-mask [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	0xffff
31:16	reg_51_int1_final_status	R0	mcu int1_n status [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	

rtc_ip_pwr_req

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_sd1_pwr_req	R/W	power fence control	0x1

Bits	Name	Access	Description	Reset
			1: power on, 0: power off [0]: sd1	
1	reg_sd1_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
2	reg_mcu_pwr_req	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
3	reg_mcu_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
15:4	Reserved			
16	reg_sd1_pwr_ack	R0	power fence power status 1: power on, 0: power off [0]: sd1	
17	reg_sd1_pwr_ack_2nd	R0	power fence power status 1: power on, 0: power off [0]: sd1	
18	reg_mcu_pwr_ack	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
19	reg_mcu_pwr_ack_2nd	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
31:20	Reserved			

rtc_ip_iso_ctrl

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	reg_sd1_iso_en	R/W	sd1 iso enablel 1: iso enable, 0: iso disable	0x0
1	reg_mcu_iso_en	R/W	mcu iso enablel 1: iso enable, 0: iso disable	0x0
15:2	Reserved			
17:16	reg_ip_por_en	R/W	1: pwr_island reset assert when power ack is 0	0x3
31:18	Reserved			

rtcsys_wkup_ctrl1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_rtcsys_wkint_final_status	RO	wkint final status [0]: sd1_wakeup_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: NA [7]: NA	
31:8	Reserved			

rtcsys_sram_ctrl

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	reg_ahb_sram_slp	R/W	1 : ahb sram into sleep mode	0x0
1	reg_ahb_sram_sd	R/W	1 : ahb sram into shut down mode	0x0
2	reg_ahb_sram_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
3	reg_sdio_sram_slp	R/W	1 : sdio sram into sleep mode	0x0
4	reg_sdio_sram_sd	R/W	1 : sdio sram into shut down mode	0x0
5	reg_sdio_sram_ctrl_ov	R/W	0 : sram's sd pin = 1'b0 1: sram ctrol by register reg_sdio_sram_sd	0x1
6	reg_mcu_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
7	reg_mcu_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
8	reg_mcu_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
9	reg_rtc_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
10	reg_rtc_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
11	reg_rtc_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
27:12	Reserved			
28	reg_mcu_sram_force_ce	R/W	1: force mcu_iram cs = 1	0x1
31:29	Reserved			

rtcsys_io_ctrl

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	reg_i2c_mux_opt0	R/W	0: pwr_gpio6/8 control by dw_gpio 1: pwr_gpio6 is PWR_IIC_SDA pwr_gpio8 is PWR_IIC_SCL	0x0
31:1	Reserved			

rtcsys_wdt_ctrl

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_rtc_hw_wdg_rst_en	R/W	0: disable rtc wdt trigger warm reset or pwrcyc reset 1: enable rtc wdt trigger warm reset or pwrcyc reset	0x0
1	reg_rtc_wdt_ctrl_mask_en	R/W	no load	0x1
31:2	Reserved			

rtcsys_irrx_clk_ctrl

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_irrx_clk_sw_force_on	R/W	force on clk ctrl of irrx	0x1
1	reg_irrx_xtal_req_en	R/W	enable irrx clk ctrl requet XTAL	0x0
2	reg_irrx_osc_req_en	R/W	enable irrx clk ctrl requet OSC	0x0
3	reg_irrx_ldo_req_en	R/W	enable irrx clk ctrl requet LDO	0x0
7:4	Reserved			
15:8	reg_irrx_xtal_filter_cyc	R/W	irrx xtal filter cycle (default 2ms)	0x40
19:16	reg_irrx_clk_ctrl_st	RO	irrx clock ctrol state	

Bits	Name	Access	Description	Reset
31:20	Reserved			

rtcsys_rtc_wkup_ctrl

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	reg_rtc_wkint_mask	R/W	wakeup source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15:8	Reserved			
23:16	reg_rtc_puint_mask	R/W	power-up source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
31:24	Reserved			

rtcsys_por_rst_ctrl

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_rtcsys_reset_en	R/W	0: not allow rtcsys reset by pwr cyc/ wdt warm reset 1 : allow rtcsys reset by pwr cyc/ wdt warm reset	0x0
1	reg_rtcsys_rstn_src_sel	R/W	select rtcsys rstn src 0: rtc_core fsm (reset with die domain) 1: por_pwr_rstn	0x0
31:2	Reserved			

12.7 电源管理与低功耗模式

12.8 概述

芯片支持三种主要的电源模式.

a. MCU only (32k-less)
b. Active
c. RTC only

12.8.1 MCU only

芯片内建低功耗的 MCU. 此时需要提供 VDDIO_RTC 电源 (VDDIO_SD1 视情况). 在这个 No die domain 里有一个 MCU 的小系统, 有自己的 clock, timer, uart, i2c, gpio, adc 等周边. 能够取代原本外挂的 MCU. MCU 能够在收到及过滤外部输入后, 唤醒主系统回到 active mode.

系统中有校准过的 oscillator 可以快速醒来, 快速睡着. 以进一步节省每次被唤醒所使用的功耗.
MCU 在等机时, 功耗约为 100uA (TBD)

12.8.2 Active mode.

Active mode 是芯片完全醒来的工作的状态. 但仍有其他如 dynamic frequency scalling 或是 dynamic clock gating 的省电技巧.

12.8.3 RTC only

RTC only 时, 系统处于最低功耗状况. 这时只有 VDDBKUP 上有功耗. 在主电源或电池消失时, 为了能够继续数时间而存在. 这时的功耗约为 6uA (TBD).
以市售常见可充电的 1mAH Li 电池. 能够使用约 166hr 约 7 天.
若改用 5mAH Li 电池. 则能延长到约 833hr 约 35 天.
因为 RTC only 是在主电源消失时的模式. 所以 RTC 目前只能计数时间. 无法唤醒系统.

12.9 时钟控制

12.9.1 关闭不需要的时钟分频器

参考时钟配置章节，根据各模块所需时钟源，将没使用到之时钟分频器关闭。达到节省功耗之目的。

12.9.2 调整模块的工作频率

根据各模块所需之时钟规格，选择较低之时钟源。分频配置较多以降低模块工作频率。须注意单一模块之频率降低不一定能降低整体之功耗。

12.9.3 模块级低功耗控制

- 模拟模块：MIPI/USB/ETH/AUD 相关寄存器设置，将不使用之模块关闭或是进入低功耗模式。
- 数字模块：根据硬件与规格，关闭不需要的数字模块之时钟

12.10 关闭不使用的 PLL

参考 PLL 配置，可以将不需要使用之 PLL 做 Powerdown，以达到功耗之节省

12.11 DDR 低功耗控制

在总线没有存取一段时间之后, DDR 控制器会自动进入 Self refresh 和 Power down 的状态. 以减少系统功耗.

有些场景因为存取断断续续无法找到足够大的空隙进入 self refresh. 这时还能考虑藉由内建存取资料量的统计寄存器. 以确认带宽是否过剩可以考虑直接降频.
DDR 控制器支援动态频率调整. 但由于调整频率会暂时停止 DDR 的存取一段时间. 所以为了减少中断的时间可能造成实时应用缓冲 underflow / overflow. 只限于 50% 和 100% 两个档位.

12.12 电压调节

芯片预设使用 PWM0 去控制 VDDC 电压调控之用. 以下是一个使用 PWM 去影向 DCDC 输出电压的例子.

```
VOUT = Vfb + R1(I2-I3)
     = Vfb + R1(Vfb/R2-(Vpwm-Vfb)/R3)
     = Vfb + R1(Vfb/R2-(VDDIOxduty-Vfb)/R3)
```

Vfb	R1	R2	R3	VDDIO
0.6	50,000	100,000	200,000	1.8

Duty	Vpwm	Vout
0.00%	0.00	1.05
5.00%	0.09	1.03
10.00%	0.18	1.01
15.00%	0.27	0.98
20.00%	0.36	0.96
25.00%	0.45	0.94
30.00%	0.54	0.92
35.00%	0.63	0.89
40.00%	0.72	0.87
45.00%	0.81	0.85
50.00%	0.90	0.83
55.00%	0.99	0.80
60.00%	1.08	0.78
65.00%	1.17	0.76
70.00%	1.26	0.74
75.00%	1.35	0.71
80.00%	1.44	0.69
85.00%	1.53	0.67
90.00%	1.62	0.65
95.00%	1.71	0.62
100.00%	1.80	0.60

```
5 RC = 5x1Kx2.2uF = 11ms
(Vpwm settle time)

Vpwm = VDDIO x PWM duty cycle
```

图表 12-1 使用 PWM 控制 DCDC 电压范例.

13 8051 子系统

13.1 概述

8051 子系统位于 RTC 独立供电的模块内，子系统配置一个 8051 和 I2C/ UART/ SPI NOR/ SD 控制器与 Timer/ WDT/中断管理/Mail box IP。系统软件可以使用 8051 在系统休眠时管理唤醒条件和唤醒系统，并且透过周边控制器与外部装置沟通。

13.2 特点

子系统内配置：

- 8051 微处理器具有以下特点
  - 支持标准 8051 指令集
  - 频率范围 25MHz ~ 300 MHz
  - 除错功能：单步执行/ jump2pc/ snapshot PSW, DPTR, PC
  - 支持 32bit 数据存取
  - 复位向量可配置到系统 AHB SRAM/ DRAM/ SPINOR
  - 支持 WFI (Clock gating)
  - 支持 code banking(最大 64x64 KB)
- 提供 8KB 空间 AHB SRAM，可被 8051 当成指令 TCM 或暂存数据
- 提供 2 组 32bit 计数器，用作定时、计数功能，可以供应用程序实现定时和计数，也可以供操作系统实现系统时钟。
- 提供 1 组 WDT，用于系统发生异常情况下，于一定时间后发出中断或复位信号，以中断或复位整个系统。
- 提供 1 组片外 SPINOR 控制
- 提供中断控制器，用于管理中断来源
- 提供 2 组 Mailbox，让 ACPU 和 8051 可以互相沟通
- 提供 1 组 I2C
- 提供 1 组 UART

13.3 功能描述

图表 13-1 8051 子系统架构

子系统内分成两个 power domain: AO domain ; MCU domain(绿色区域)系统可以透过寄存器来选择关闭 MCU 所属 power domain 来达到省电需求。
在系统休眠状态下，可以透过 MCU 处理中断并且通过配置寄存器唤醒系统，也可以透过 I2C/ UART 与外部装置沟通。

13.4 工作方式

13.4.1 Power domain 控制流程

子系统内分成三个 power domain, RTC domain (Always on), MCU domain。MCU domain 可以通过配置寄存器进行上下电流程，流程如下：

MCU power_off>
1. 软复位寄存器 =0
2. 配置寄存器 reg_mcu_iso_en = 1
3. 配置寄存器 reg_mcu_pwr_req = 0

MCU power_on=>
1. 配置寄存器 reg_mcu_pwr_req = 1
2. 轮询寄存器 reg_mcu_pwr_ack = 1
3. 配置寄存器 reg_mcu_iso_en = 0
4. 软复位寄存器 =1

13.4.2 8051 初始化

8051 在系统初始是处于复位状态，使用 8051 可以透过 ACPU 来完成下面的软件流程:

1. 8051 复位状态处于复位状态(寄存器 reg_soft_rstn_mcu =0)
2. 配置寄存器 reg_mcu_rom_addr_size 来决定指令 TCM size。
3. 配置寄存器 reg_51irom_ioffset 来决定 TCM 执行在 AHB SRAM 上的位置
4. 配置寄存器 reg_soft_rstn_mcu =1 解除 8051 复位状态

13.4.3 中断处理

8051 可以透过 int0_n, int1_n 接口接收外部电平触发中断，int0_n/ int1_n 分别从 ictl (中断控制)和配置寄存器 reg_51_int1_src_mask 来选择输出中断讯号给 8051。

中断號	中断名称	中断叙述
0	Vbat_det	系统掉电中断
1	mbox_int0	Mailbox 中断
2	NA	保留
3	irrx	遙控器接收中断
4	gpio_int	PWR GPIO 中断
5	uart_int	PWR UART 中断
6	spinor1_int	SPINOR1 中断
7	timer_int0	TIMER0 中断
8	timer_int1	TIMER1 中断
9	Irq_ap2rtc[0]	系统中断
10	Irq_ap2rtc[1]	系统中断
11	I2c_int	PWR I2C 中断
12	st_change_int	RTC 状态变更中断
13	hw_thm_shdn	系统过热中断
14	saradc	SARADC 中断
15	wdt_int	看门狗中断

图表 13-12 8051 子系统中断列表

13.4.4 MAILBOX

Mailbox 提供 2 组 spinlock 功能栏位，和 4 组 32bit 信息栏位，让 ACPU/8051 可以相互传输信息

13.5 8051 子系统寄存器概览

表格 13-1 RTC_CTRL 寄存器概览 (基址 0x05025000)

Name	Address Offset	Description
rtc_ctrl_version	0x000	rtc_ctrl_version
rtc_ctrl_unlockkey	0x004	rtc_ctrl_unlockkey
rtc_ctrl0	0x008	rtc_ctrl0
rtc_ctrl_status0	0x00c	rtc_ctrl_status0
rtc_ctrl_status1	0x010	rtc_ctrl_status1
rtc_ctrl_status2gpio	0x014	rtc_ctrl_status2gpio
rtcsys_rst_ctrl	0x018	rtcsys_rst_ctrl
rtcsys_clkmux	0x01c	rtcsys_clkmux
rtcsys_mcu51_ctrl0	0x020	rtcsys_mcu51_ctrl0
rtcsys_mcu51_ctrl1	0x024	rtcsys_mcu51_ctrl1
rtcsys_pmu	0x028	rtcsys_pmu
rtcsys_status	0x02c	rtcsys_status
rtcsys_clkbyp	0x030	rtcsys_clkbyp
rtcsys_clk_en	0x034	rtcsys_clk_en
rtcsys_wkup_ctrl	0x038	rtcsys_wkup_ctrl
rtcsys_clkdiv	0x03c	rtcsys_clkdiv
fc_coarse_en	0x040	fc_coarse_en
fc_coarse_cal	0x044	fc_coarse_cal
fc_fine_en	0x048	fc_fine_en
fc_fine_period	0x04c	fc_fine_period
fc_fine_cal	0x050	fc_fine_cal
rtcsys_pmu2	0x054	rtcsys_pmu2
rtcsys_clkdiv1	0x058	rtcsys_clkdiv
rtcsys_mcu51_dbg	0x05c	rtcsys_mcu51_dbg
sw_reg0	0x060	sw_reg0
sw_reg1_por	0x064	sw_reg1_por
fab_lp_ctrl	0x068	fab_lp_ctrl
fab_option	0x06c	fab_option
rtcsys_mcu51_ictrl1	0x07c	rtcsys_mcu51_ictrl1
rtc_ip_pwr_req	0x080	rtc_ip_pwr_req
rtc_ip_iso_ctrl	0x084	rtc_ip_iso_ctrl
rtcsys_wkup_ctrl1	0x094	rtcsys_wkup_ctrl1
rtcsys_sram_ctrl	0x098	rtcsys_sram_ctrl
rtcsys_io_ctrl	0x09c	rtcsys_io_ctrl
rtcsys_wdt_ctrl	0x0a0	rtcsys_wdt_ctrl
rtcsys_irrx_clk_ctrl	0x0a4	rtcsys_irrx_clk_ctrl
rtcsys_rtc_wkup_ctrl	0x0a8	rtcsys_rtc_wkup_ctrl
rtcsys_por_rst_ctrl	0x0ac	rtcsys_por_rst_ctrl

13.6 8051 子系统寄存器

rtc_ctrl_unlockkey

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	rtc_ctrl_unlockkey	R/W	"rtc_ctrl0" could be write when unlockkey is set to be 0xAB18.	0x0000

Bits	Name	Access	Description	Reset
			"ptest_adc2ram_ctrl" could be write when unlockkey is set to be 0x0423. If unlockkey_clear is set to 1, the rtc_ctrl0_unlockkey will be clear after a apb write to rtc_ctrl0 or ptest_adc2ram_ctrl	
31:16	Reserved			

rtc_ctrl0

Offset Address: 0x008
Write Lock: wr_lock_rtc_ctrl0

Bits	Name	Access	Description	Reset
0	req_shdn	W1P	Mask: Enabled	
1	req_sw_thm_shdn	R/W	Mask: Enabled	0x0
2	hw_thm_shdn_en	R/W	Mask: Enabled	0x0
3	req_pwr_cyc	W1P	Mask: Enabled	
4	req_warm_rst	W1P	Mask: Enabled	
5	req_sw_wdg_rst	R/W	Mask: Enabled	0x0
6	hw_wdg_rst_en	R/W	Mask: Enabled	0x0
7	req_suspend	W1P	Mask: Enabled	
8	unlockkey_clear	R/W	Mask: Enabled	0x0
9	Reserved			
10	reg_rtc_mode	R/W	Mask: Enabled	0x0
11	reg_clk32k_cg_en	R/W	Mask: Enabled	0x1
31:12	Reserved			

rtc_ctrl_status0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	rtc_pwr_vbat_det_o	RO		
1	rtc_pwr_button0_o	RO		
2	rtc_pwr_button1_o	RO		
3	rtc_pwr_button1_7sec_o	RO		
4	rtc_pwr_on_o	RO		
5	rtc_pwr_wakeup0_o	RO		
6	rtc_pwr_wakeup1_o	RO		
7	rtc_mode_o	RO		
19:8	Reserved			
20	rtc_rstn_o	RO		
21	rtc_alarm_o	RO		

Bits	Name	Access	Description	Reset
22	hw_thm_shdn_sta_i	RO		
23	hw_wdg_rst_sta_i	RO		
24	sys_reset_x_i	RO		
25	cg_en_out_clk_32k	RO		
29:26	rtc_fsm_st	RO		
31:30	Reserved			

rtc_ctrl_status1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	rtc_sec_value_o	RO		

rtc_ctrl_status2gpio

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	status2gpio_en	R/W		0x0
31:8	Reserved			

rtcsys_rst_ctrl

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_soft_rstn_mcu	R/W	0 : reest MCU	0x0
2	reg_soft_rstn_sdio	R/W	0 : reset SD1	0x1
3	reg_soft_rstn_uart	R/W	0 : reset Uart	0x1
4	reg_soft_rstn_spinor	R/W	0 : reset spinor1	0x1
5	reg_soft_rstn_ictl	R/W	0 : reset dw_ictl	0x1
6	reg_soft_rstn_mbox	R/W	0 : reset mbox	0x1
7	reg_soft_rstn_fab_hs2rtc	R/W	0 : reset hs2rtc	0x1
8	reg_soft_rstn_fab_rtc2ap	R/W	0 : reset rtc2ap	0x1
9	reg_soft_rstn_fab_sram	R/W	0 : reset ahb sram logic	0x1
10	reg_soft_rstn_apb	R/W	no load	0x1
11	reg_soft_rstn_apb_timer	R/W	0 : reset dw timer apb logic	0x1
12	reg_soft_rstn_timer0	R/W	0 : reset dw timer0	0x1
13	reg_soft_rstn_timer1	R/W	0 : reset dw timer1	0x1
14	reg_soft_rstn_osc	R/W	0 : reset osc	0x1
15	reg_soft_rstn_gpio	R/W	0 : reset gpio	0x1
16	reg_soft_rstn_i2c	R/W	0 : reset i2c	0x1
17	reg_soft_rstn_saradc	R/W	0 : reset saradc	0x1
18	reg_soft_rstn_wdt	R/W	0 : reset wdt	0x1
19	reg_soft_rstn_irrx	R/W	0 : reset irrx	0x1
20	reg_soft_rstn_f32kless	R/W	0 : reset f32kless	0x1
31:21	Reserved			

rtcsys_clkmux

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
3:0	reg_sdio_clk_mux	R/W	clk_sd1_pre 0 : fpll/4 1: osc_div	0x0
7:4	reg_fab_clk_mux	R/W	clk_fab_pre 0 : 32K, 1: fpll/5, 2: osc_div	0x0

Bits	Name	Access	Description	Reset
9:8	reg_timer0_clk_mux	R/W	0: xtal 1: 32K	0x0
11:10	reg_timer1_clk_mux	R/W	0: xtal 1: 32K	0x0
13:12	reg_apb_clk_mux	R/W	00 : cgdiv and refer to apbactive 01 : force clk_apb, clk_fab 1:1 (default) 10 : force clk_apb, clk_fab 1:2 11 : force clk_apb, clk_fab 1:4	0x1
15:14	Reserved			
17:16	reg_i2c_clk_mux	R/W	0: xtal 1: osc div	0x0
19:18	reg_sd_mclk_clk_mux	R/W	0: 100Khz from OSC, 1: 32K	0x0
20	reg_saradc_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
21	reg_irrx_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
31:22	Reserved			

rtcsys_mcu51_ctrl0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
4:0	reg_51_rom_addr_size	R/W	Determines how many of the sixteen internal ROM address bits (irom_addr) are used (0 = no internal ROM present);	0xc
5	reg_51_mem_ea_n	R/W	0 : external rom exist, 1: external rom not exist	0x0
6	reg_51_xdata_mode	R/W	0 : fetch xdata with clock gating 1 : fetch xdata wo clock gating (to support 51 timer and 51 uart)	0x0
7	reg_51_rom_addr_def	R/W	0: mercury define , max internal rom = 2^reg_51_rom_addr_size -1 internal rom offset = 4K*reg_51irom_ioffset 1: mars define , max internal rom = 2K*reg_51_rom_addr_size -1 internal rom offset = 2K*reg_51irom_ioffset	0x0
10:8	Reserved			
31:11	reg_51xdata_ioffset0	R/W	Set offset address[31:11] to select mcu8051 boot device	0x0A400

rtcsys_mcu51_ctrl1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
4:0	reg_51irom_ioffset	R/W	boot rom offset to rtcsys_sram	0x0
5	Reserved			
9:6	reg_51_pf_mode	R/W	reg_51_pf_mode	0x0
10	Reserved			
31:11	reg_51xdata_doffset0	R/W	Set offset address[31:11] to select mcu8051 xdata	0x0A400

rtcsys_pmu

Offset Address: 0x028

Bits	Name	Access	Description	Reset
3:0	Reserved			
4	reg_dis_pmu_ldo_ctrl	R/W	disable pmu ldo ctrl	0x0

Bits	Name	Access	Description	Reset
			0: enable pmu to ctrl RTC_LDO sleep mode 1: disable pmu to ctrl RTC_LDO sleep mode	
5	reg_wdt_clkoff_by_pmu	R/W	wdt_clk gate by pmu when mcu into idle mode 1. wdt clock gate by pmu	0x0
6	reg_force_osc_off	R/W	1 : force osc off	0x0
7	reg_force_osc_on	R/W	1 : force osc on	0x0
8	reg_pmu_sleep_mode	R/W	pmu enter light sleep mode when mcu idle 1 : enable pmu light sleep mode when mcu idle (pmu control osc_req/ sram slp) 0 disable pmu light sleep mode	0x0
9	reg_pmu_lowpwr_mode	R/W	mcu_pmu into sleep state when rtc at suspend state & mcu idle & reg_pmu_sleep_mode enable 1 : enable mcu_pmu into sleep mode (trigger rtc ldo step down power) 0 disable mcu_pmu sleep mode	0x0
13:10	reg_pmu_stable_cnt	R/W	Stable timer when mcu_pmu leave sleep state, clock unit : 31.25us (32khz), wait for 1~16 tick cycle	0x3
14	reg_xtal_off_by_pmu	R/W	pmu control xtal request 1: xtal request disable by pmu sleep mode	0x0
15	reg_rtcsys_clk25m_req	R/W	xtal request1 for rtcsys 0: disable 25m xtal request1(rtcsys) 1: enable 25m xtal request1 (rtcsys)	0x1
19:16	reg_rtc_vbat_det_db_cnt	R/W	vbat det int debounce time (cycle unit : 32K)	0x2
20	reg_rtc_vbat_det_db_en	R/W	0: disable vbat det int debounce 1: enable vbat det int debounce	0x1
21	reg_ahb_sram_auto_slp_en	R/W	1: enable ahb sram into slp md when bus idle	0x0
23:22	reg_ahb_sram_busy_sel	R/W	2'd0: cs | cs_d1 2'd1: cs | cs_d1 | cs_d2 2'd2: cs | cs_d1 | cs_d2 | cs_d3 3'd3: cs | cs_d1 | cs_d2 | cs_d3 | cs_d4	0x0
24	reg_rtc_stint_clr	W1P	clear rtc state change interrupt	
25	reg_vbat_det_int_clr	W1P	clear vbet det interrupt	
26	reg_rtcsys_clk25m_hw_req	R/W	xtal request1 for rtcsys from hw ip 0: disable 25m xtal request1 from hw ip(rtcsys) 1: enable 25m xtal request1 from hw ip(rtcsys)	0x0
27	Reserved			
28	reg_vbat_det_force_clk	R/W	1: when vbat det happen, change rtcsys bus clock to OSC	0x0
29	reg_mcu_clkoff_by_pmu	R/W	mcu_clk gate by pmu when into idle mode 1. mcu clock gate by pmu	0x1
30	reg_xtal_off_by_susp	R/W	ISO off control xtal request 1: xtal request disable by ISO_OFF	0x0

Bits	Name	Access	Description	Reset
31	reg_osc_off_by_susp	R/W	ISO off control osc request 1: osc request disable by ISO_OFF	0x0

rtcsys_status

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	reg_rtcsys_status	RO	[0] enable rtc2apb ahb path 0: rtcsys ip can only access 0x05000000+16MB 1: rtcsys ip can access full range address [1] flag of vbat_det_force_clk	

rtcsys_clkbyp

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	reg_clk_byp	R/W	[0] : clk_fab , 0: clk_fab_pre, 1: xtal (default) [1] : clk_sdio, 1: clk_sd1_pre, 1: xtal (default) [31:2]: NA	0xffffffff

rtcsys_clk_en

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_clk_en	R/W	[0]: NA [1]: clk_sd1 (sd1 card clock) [2]: clk_fab_sd1 (sd1 core clock) [3]: clk_mcu [4]: clk_hs2rtc_mst [5]: clk_rtc2ap_slv [6]: clk_spinor1 [7]: clk_fab_sram (AHB sram) [8]: NA [9]: clk_apb_timer [10]: clk_timer0 [11]: clk_timer1 [12]: clk_apb_uart [13]: clk_uart [14]: clk_apb_ictrl [15]: clk_apb_mbox [16]: clk_apb_gpio [17]: clk_apb_osc [18]: clk_gpio_db [19]: clk_apb_i2c [20]: clk_i2c [21]: NA [22]: clk_sd1_tmclk [23]: clk_apb_saradc [24]: clk_saradc [25]: clk_apb_wdt [26]: clk_wdt [27]: clk_irrx [31:28]: NA	0xffffffff

rtcsys_wkup_ctrl

Offset Address: 0x038

Bits	Name	Access	Description	Reset
14:0	reg_rtcsys_wkint_mask	R/W	mask int to RTC_CORE.REQ_WAKEUP/ MCU_PMU [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15	reg_vbat_det_wkup_mask	R/W	1: mask vbat det int	0x1
16	reg_sw_wkint_req	R/W	mcu sw wakeup interrupt to RTC_CORE 1: interrupt active	0x0
23:17	Reserved			
24	reg_wkint2rtc_mask	R/W	1: mask wakeup int (rtcsys int) to RTC core	0x1
31:25	Reserved			

rtcsys_clkdiv

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	reg_div_clk_osc_fab_div_val	R/W	Clock Divider Factor	0x1
4	reg_div_clk_osc_fab_dis	R/W	Clock gate	0x0
5	reg_div_clk_osc_fab_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
15:6	Reserved			
19:16	reg_div_clk_osc_i2c_div_val	R/W	Clock Divider Factor	0x1
20	reg_div_clk_osc_i2c_dis	R/W	Clock gate	0x0
21	reg_div_clk_osc_i2c_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
23:22	Reserved			
29:24	reg_div_clk_osc_saradc_div_val	R/W	Clock Divider Factor	0x1
30	reg_div_clk_osc_saradc_dis	R/W	Clock gate	0x0
31	reg_div_clk_osc_saradc_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0

fc_coarse_en

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	fc_coarse_en	R/W	使能 32K 粗調 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_coarse_cal

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	fc_coarse_value	RO	32K 粗調計數器值 (單位: 25MHz 時鐘) 25MHz 時鐘計數一個 32K 時鐘週期。	
31:16	fc_coarse_time	RO	32K 粗調完成次數	

fc_fine_en

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	fc_fine_en	R/W	使能 32K 細調 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_fine_period

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	fc_fine_period	R/W	32K 細調計數週期 (單位: 32K 時鐘) 設定每次使用 25MHz 時鐘計數幾個 32K 時鐘週期。	0x0100
31:16	Reserved			

fc_fine_cal

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	fc_fine_value	RO	32K 細調計數器值 (單位: 25MHz 時鐘) 25MHz 時鐘計數一個 fc_fine_period 週期。	
31:24	fc_fine_time	RO	32K 細調完成次數	

rtcsys_pmu2

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	reg_rtc_sys_wkint_db_en	R/W	PMU wakeup int debounce enable	0x1
4:1	reg_rtc_sys_wkint_db_cnt	R/W	PMU wakeup int debounce cycle (32K)	0x2
31:5	Reserved			

rtcsys_clkdiv1

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_div_clk_osc_irrx_div_val	R/W	Clock Divider Factor	0x0
22	Reserved			
23	reg_div_clk_osc_irrx_dis	R/W	Clock gate	0x1
31:24	Reserved			

rtcsys_mcu51_dbg

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
3:0	reg_51_dbg_sel	R/W	select mcu51 debug bus (check mcu design review ppt)	0x0
4	reg_51_dbg_snap_shot	W1P	snap shot mcu51 internal register to dbg register (reg_rtcsys_dbg)	
5	reg_51_dbg_step_en	R/W	0: disable mcu debug function 1: enable mcu debug function, and mcu stop at current PC	0x0
6	reg_51_dbg_step	W1P	1: mcu jump to next PC	
7	reg_51_dbg_jump	W1P	1: mcu jump to target pc value (reg_51_dbg_jump2pc)	

Bits	Name	Access	Description	Reset
15:8	Reserved			
31:16	reg_51_dbg_jump2pc	R/W	16 bit mcu target pc value	0x0

sw_reg0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	sw_reg0	R/W	reg for SW	0x0
31:8	Reserved			

sw_reg1_por

G

Offset Address: 0x064

Bits	Name	Access	Description	Reset
7:0	sw_reg1_por	R/W	reg for SW could only be reset by power reset	0x0
31:8	Reserved			

fab_lp_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
7:0	rtcsys_fab_busy_sel	R/W	select signal to request sys_ctrl to speed up fab clock	0xDF
9:8	rtcsys_fab_busy_ctrl	R/W	rtcsys_fab_busy signal is combi or register out	0x0
11:10	apdbg_busy_ctrl	R/W	apdbg_busy signal is combi or register out	0x0
13:12	reg_apb_busy_ctrl	R/W	apb bridge_busy signal is combi or register out	0x3
15:14	reg_mcu_busy_ctrl	R/W	mcu_busy signal is combi or register out	0x3
31:16	Reserved			

fab_option

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
1:0	rtcsys_fab_option	R/W	the ahb_h2h2 design option	0x0
31:2	Reserved			

rtcsys_mcu51_ictrl1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	reg_51_int1_src_mask	R/W	select rtcsys_int src to mcu int1_n 1: mask, 0: un-mask [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int	0xffff

Bits	Name	Access	Description	Reset
			[12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	
31:16	reg_51_int1_final_status	RO	mcu int1_n status [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	

rtc_ip_pwr_req

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_sd1_pwr_req	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
1	reg_sd1_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
2	reg_mcu_pwr_req	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
3	reg_mcu_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
15:4	Reserved			
16	reg_sd1_pwr_ack	RO	power fence power status 1: power on, 0: power off [0]: sd1	
17	reg_sd1_pwr_ack_2nd	RO	power fence power status 1: power on, 0: power off [0]: sd1	
18	reg_mcu_pwr_ack	RO	power fence power status 1: power on, 0: power off [1]: mcu subsys	
19	reg_mcu_pwr_ack_2nd	RO	power fence power status 1: power on, 0: power off [1]: mcu subsys	
31:20	Reserved			

rtc_ip_iso_ctrl

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	reg_sd1_iso_en	R/W	sd1 iso enablel	0x0

Bits	Name	Access	Description	Reset
			1: iso enable, 0: iso disable	
1	reg_mcu_iso_en	R/W	mcu iso enablel 1: iso enable, 0: iso disable	0x0
15:2	Reserved			
17:16	reg_ip_por_en	R/W	1: pwr_island reset assert when power ack is 0	0x3
31:18	Reserved			

rtcsys_wkup_ctrl1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_rtcsys_wkint_final_status	RO	wkint final status [0]: sd1_wakeup_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: NA [7]: NA	
31:8	Reserved			

rtcsys_sram_ctrl

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	reg_ahb_sram_slp	R/W	1 : ahb sram into sleep mode	0x0
1	reg_ahb_sram_sd	R/W	1 : ahb sram into shut down mode	0x0
2	reg_ahb_sram_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
3	reg_sdio_sram_slp	R/W	1 : sdio sram into sleep mode	0x0
4	reg_sdio_sram_sd	R/W	1 : sdio sram into shut down mode	0x0
5	reg_sdio_sram_ctrl_ov	R/W	0 : sram's sd pin = 1'b0 1: sram ctrol by register reg_sdio_sram_sd	0x1
6	reg_mcu_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
7	reg_mcu_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
8	reg_mcu_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
9	reg_rtc_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
10	reg_rtc_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
11	reg_rtc_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
12	reg_ahb_sram_1_slp	R/W	1 : ahb sram into sleep mode	0x0
13	reg_ahb_sram_1_sd	R/W	1 : ahb sram into shut down mode	0x0
14	reg_ahb_sram_1_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
15	reg_ahb_sram_2_slp	R/W	1 : ahb sram into sleep mode	0x0
16	reg_ahb_sram_2_sd	R/W	1 : ahb sram into shut down mode	0x0
17	reg_ahb_sram_2_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb	0x1

Bits	Name	Access	Description	Reset
			sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	
27:18	Reserved			
28	reg_mcu_sram_force_ce	R/W	1: force mcu_iram cs = 1	0x1
31:29	Reserved			

rtcsys_io_ctrl

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	reg_i2c_mux_opt0	R/W	0: pwr_gpio6/8 control by dw_gpio 1: pwr_gpio6 is PWR_IIC_SDA pwr_gpio8 is PWR_IIC_SCL	0x0
31:1	Reserved			

rtcsys_wdt_ctrl

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_rtc_hw_wdg_rst_en	R/W	0: disable rtc wdt trigger warm reset or pwrcyc reset 1: enable rtc wdt trigger warm reset or pwrcyc reset	0x0
1	reg_rtc_wdt_ctrl_mask_en	R/W	no load	0x1
31:2	Reserved			

rtcsys_irrx_clk_ctrl

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_irrx_clk_sw_force_on	R/W	force on clk ctrl of irrx	0x1
1	reg_irrx_xtal_req_en	R/W	enable irrx clk ctrl requet XTAL	0x0
2	reg_irrx_osc_req_en	R/W	enable irrx clk ctrl requet OSC	0x0
3	reg_irrx_ldo_req_en	R/W	enable irrx clk ctrl requet LDO	0x0
7:4	Reserved			
15:8	reg_irrx_xtal_filter_cyc	R/W	irrx xtal filter cycle (default 2ms)	0x40
19:16	reg_irrx_clk_ctrl_st	RO	irrx clock ctrol state	
31:20	Reserved			

rtcsys_rtc_wkup_ctrl

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	reg_rtc_wkint_mask	R/W	wakeup source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15:8	Reserved			
23:16	reg_rtc_puint_mask	R/W	power-up source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int	0xff

Bits	Name	Access	Description	Reset
			[2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	
31:24	Reserved			

rtcsys_por_rst_ctrl

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_rtcsys_reset_en	R/W	0: not allow rtcsys reset by pwr cyc/ wdt warm reset 1 : allow rtcsys reset by pwr cyc/ wdt warm reset	0x0
1	reg_rtcsys_rstn_src_sel	R/W	select rtcsys rstn src 0: rtc_core fsm (reset with die domain) 1: por_pwr_rstn	0x0
31:2	Reserved			

14 存储器接口

14.1 DDR 控制器

14.1.1 概述

DDR 控制器实现了动态储存器(DRAM)的数据访问，它将 SoC 中各主设备的数据访问命令转换为符合 JEDEC 标准的动态储存器命令，加以适当排程，进而提升动态储存器总线的使用效率。

14.1.2 特点

- 功能特点：
- 支持:
  - DDR2 最高数据速率 1333 Mbps。
  - DDR3 最高数据速率 1866 Mbps。
- 支持接口数据位宽 16-bit。
- 支持单通道，单 rank。
- 支持自动刷新控制。
- 支持优先级控制。
- 支持数据流量统计。
- 支持低功耗模式。
- 支持地址映射。
- 支持管脚复用。

14.1.3 功能描述

14.1.3.1 应用框图

DRAM 接口支持 16 比特数据位宽，32 比特位宽。图表 14-1 为主芯片与单片 DRAM 器件的互联示意图：

图表 14-1 SoC/DRAM 互联示意图

Command 由数个信号组成，依 DRAM 型别会有所差异，表格 4-1 对 DDR2 / DDR3 命令信号做个比较：

表格 14-1 DDR2/DDR3 命令信号比较表

功能	DDR2	DDR3
DDR CKE	D2x_CKE	D3x_CKE
DDR CLKN	D2x_CK_N	D3x_CK_N
DDR CLKP	D2x_CK_P	D3x_CK_P
DDR CSB	D2x_CS_N	D3x_CS_N
DDR RESETN	D2x_RESET_N	D3x_RESET_N
DDR RASN	D2x_RAS_N	D3x_RAS_N
DDR CASN	D2x_CAS_N	D3x_CAS_N
DDR WEN	D2x_WE_N	D3x_WE_N
DDR ACTN	N/A	N/A
DDR BA	D2x_BAn	D3x_BAn
DDR BG	N/A	N/A
DDR MA	D2x_An (n = 0 – 15)	D3x_An (n = 0 – 15)
DDR ODT	D2x_ODT	D3x_ODT
说明: 管脚名称会随封装不同有些许差异。		

14.1.3.2 功能原理

基于 DRAM 的存储特性，JEDEC 制定一套标准，标准规范了访问 DRAM 数据及控制 DRAM 状态所需的命令及时序。适当的配置 DDR 寄存器，DDR 控制器就能发出满足 JEDEC 标准的时序，完成读、写及低耗电控制等动作。

14.1.3.2.1 命令真值表

DDR 接口时序满足 JDEDC 标准，表格 14-2 及表格 14-3。分别为 DDR2 及 DDR3 支持命令的真值表，供用户参考，其余信息可参照 JEDEC 标准。

表格 14-2 DDR2 命令真值表

Function	CKE Pre	CKE Cur	CS#	RAS#	CAS#	WE#	BA0 - BA2	A11 - A15	A10 / AP	A0 - A9
Mode Register Set	H	H	L	L	L	L	BA	OP		
Refresh	H	H	L	L	L	H	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA		
Write	H	H	L	H	L	L	BA	RFU	L	CA
Write with Auto Precharge	H	H	L	H	L	L	BA	RFU	H	CA
Read	H	H	L	H	L	H	BA	RFU	L	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X
H：表示高电平；L：表示低电平；V：表示有效；X：表示不关心。 RFU：表示保留给未来使用。										

表格 14-3 DDR3 命令真值表

Function	CKE Pre	CKE Cur	CS#	RAS#	CAS#	WE#	BA0 - BA2	A13 - A15	A12 / BC#	A10 / AP	A0 - A9, 11
Mode Register Set	H	H	L	L	L	L	BA	OP			
Refresh	H	H	L	L	L	H	V	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA			
Write **(Fixed BL8 or BC4)**	H	H	L	H	L	L	BA	RFU	V	L	CA

Function	CKE Pre	CKE Cur	CS #	RAS#	CAS#	WE#	BA0 - BA2	A13 - A15	A12 / BC #	A10 / AP	A0 - A9, 11
Write (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	L	CA
Write (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	L	CA
Write with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	L	BA	RFU	V	H	CA
Write with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	H	CA
Write with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	H	CA
Read (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	L	CA
Read (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	L	CA
Read (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	L	CA
Read with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	H	CA
Read with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	H	CA
Read with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
ZQ Calibration Long	H	H	L	H	H	L	X	X	X	H	X
ZQ Calibration Short	H	H	L	H	H	L	X	X	X	L	X

H：表示高电平；L：表示低电平；V：表示有效；X：表示不关心。
RFU：表示保留给未来使用。

14.1.3.2.2 自动刷新

DDR 控制器具备控制自动刷新功能的能力，控制自动刷新的目为减少访问数据的延迟或是减少刷新命令对 DRAM 带宽的冲击，尽量在 DRAM 空闲时发刷新命令，具体可用手段有：

- 等间隔刷新：每间隔 tREFI 时间就发刷新命令。
- 取巧式刷新：DDR 控制器内部会统计 tREFI 过期数目，再利用空闲时间连续发。

14.1.3.2.3 低功耗管理

DDR 控制器支持低功耗模式：

- 普通低功耗模式：透过寄存器设定一空闲时间，当普通低功耗模式使能后，且 DDR 控制器满足空闲时间，自动控制 DRAM 进入普通低耗模式。

- 自刷新模式：自刷新为一功耗更低的模式，透过寄存器设定一空闲时间，当自刷新模式使能后，且 DDR 控制器满足空闲时间，自动控制 DRAM 进入自刷新模式。

14.1.3.2.4 仲裁机制

DDR 控制器主要依据 DRAM 各项控制时序优化系统的带宽使用率，另通过优先级调度算法对各命令进行调度。另外，DDRC 还实现了 timeout 控制、real-time 控制这两种调度辅助手段（根据业务需要来使能这两种控制手段，可以同时使能，也可以单独使能），对命令的请求进行控制。

- 连续地址访问限制

  限制级数为：0~15 个 DRAM 读/写指令，每个 AXI 端口的配置是独立的。DDR 控制器默认对连续地址有高优先权，以优化 DRAM 利用率，此机制限制每个 AXI 埠最大可连续存取 DRAM 的长度。
- 优先级调度

  优先级等级为：0~15，数值越高表示优先权越高，每个 AXI 埠的读/写优先级配置是独立的。
- Timeout 控制

  对于每个 AXI 埠的读/写传输，可配置 timeout 寄存器以避免过长时间的等待，在等待时间到达后，强制屏蔽尚未到达等待时间、或未配置 timeout 属性的 AXI 端口。
- Real-time 控制

  对于 real-time function，可配置硬件缓冲阀值，若缓冲不足时自动提高优先级至最高，并可限制其他 AXI 埠生成新的传输。

14.1.3.2.5 流量统计和命令 latency 统计功能

DDR 控制器支持流量统计功能：可以统计各 AXI 埠读写流量，用以收集当前流量信息以决定是否需要进行流量控制。可以统计 DRAM 总体读/写流量统计。
DDR 控制器支持 AXI latency 统计功能，支持指定/不指定传输的累积 latency 统计。

14.1.3.2.6 地址映射方式

DDR 控制器实现了将系统总线的访问地址转换为 DRAM 的访问地址。可实现 RBC (row_bank_column)、BRC，并支持 Bank interleave 在 row/column bit。

14.1.4 工作方式

14.1.4.1 软复位

不支持软复位。

14.1.4.2 DDR 初始化配置流程

本控制器初始化流程以软件包型式提供。

14.1.5 AXI 寄存器

14.1.5.1 AXI 寄存器概览

基址 0x0800_4000

Name	Address Offset	Description
AXI_CTRL0_1	0x4b4	AXI1 read timeout control
AXI_CTRL1_1	0x4b8	AXI1 write timeout control
AXI_CTRL0_2	0x564	AXI2 read timeout control
AXI_CTRL1_2	0x568	AXI2 write timeout control
AXI_CTRL0_3	0x614	AXI3 read timeout control
AXI_CTRL1_3	0x618	AXI3 write timeout control

基址 0x0800_8000

Name	Address Offset	Description
AXI_MON0_CTRL	0x000	AXI monitor 0 control
AXI_MON0_INPUT	0x004	AXI monitor 0 input selection
AXI_MON0_FILTER0	0x010	AXI monitor 0 filter settings
AXI_MON0_FILTER1	0x014	AXI monitor 0 filter settings
AXI_MON0_FILTER2	0x018	AXI monitor 0 filter settings
AXI_MON0_FILTER3	0x01c	AXI monitor 0 filter settings
AXI_MON0_FILTER4	0x020	AXI monitor 0 filter settings
AXI_MON0_FILTER5	0x024	AXI monitor 0 filter settings
AXI_MON0_FILTER6	0x028	AXI monitor 0 filter settings
AXI_MON0_FILTER7	0x02c	AXI monitor 0 filter settings
AXI_MON0_FILTER8	0x030	AXI monitor 0 filter settings
AXI_MON0_RPT0	0x040	AXI monitor 0 cycle count
AXI_MON0_RPT1	0x044	AXI monitor 0 hit count
AXI_MON0_RPT2	0x048	AXI monitor 0 byte count
AXI_MON0_RPT3	0x04c	AXI monitor 0 latency count
AXI_MON1_CTRL	0x080	AXI monitor 1 control
AXI_MON1_INPUT	0x084	AXI monitor 1 input selection
AXI_MON1_FILTER0	0x090	AXI monitor 1 filter settings
AXI_MON1_FILTER1	0x094	AXI monitor 1 filter settings
AXI_MON1_FILTER2	0x098	AXI monitor 1 filter settings
AXI_MON1_FILTER3	0x09c	AXI monitor 1 filter settings
AXI_MON1_FILTER4	0x0a0	AXI monitor 1 filter settings
AXI_MON1_FILTER5	0x0a4	AXI monitor 1 filter settings
AXI_MON1_FILTER6	0x0a8	AXI monitor 1 filter settings
AXI_MON1_FILTER7	0x0ac	AXI monitor 1 filter settings
AXI_MON1_FILTER8	0x0b0	AXI monitor 1 filter settings
AXI_MON1_RPT0	0x0c0	AXI monitor 1 cycle count
AXI_MON1_RPT1	0x0c4	AXI monitor 1 hit count
AXI_MON1_RPT2	0x0c8	AXI monitor 1 byte count
AXI_MON1_RPT3	0x0cc	AXI monitor 1 latency count
AXI_MON2_CTRL	0x100	AXI monitor 2 control

Name	Address Offset	Description
AXI_MON2_INPUT	0x104	AXI monitor 2 input selection
AXI_MON2_FILTER0	0x110	AXI monitor 2 filter settings
AXI_MON2_FILTER1	0x114	AXI monitor 2 filter settings
AXI_MON2_FILTER2	0x118	AXI monitor 2 filter settings
AXI_MON2_FILTER3	0x11c	AXI monitor 2 filter settings
AXI_MON2_FILTER4	0x120	AXI monitor 2 filter settings
AXI_MON2_FILTER5	0x124	AXI monitor 2 filter settings
AXI_MON2_FILTER6	0x128	AXI monitor 2 filter settings
AXI_MON2_FILTER7	0x12c	AXI monitor 2 filter settings
AXI_MON2_FILTER8	0x130	AXI monitor 2 filter settings
AXI_MON2_RPT0	0x140	AXI monitor 2 cycle count
AXI_MON2_RPT1	0x144	AXI monitor 2 hit count
AXI_MON2_RPT2	0x148	AXI monitor 2 byte count
AXI_MON2_RPT3	0x14c	AXI monitor 2 latency count
AXI_MON3_CTRL	0x180	AXI monitor 3 control
AXI_MON3_INPUT	0x184	AXI monitor 3 input selection
AXI_MON3_FILTER0	0x190	AXI monitor 3 filter settings
AXI_MON3_FILTER1	0x194	AXI monitor 3 filter settings
AXI_MON3_FILTER2	0x198	AXI monitor 3 filter settings
AXI_MON3_FILTER3	0x19c	AXI monitor 3 filter settings
AXI_MON3_FILTER4	0x1a0	AXI monitor 3 filter settings
AXI_MON3_FILTER5	0x1a4	AXI monitor 3 filter settings
AXI_MON3_FILTER6	0x1a8	AXI monitor 3 filter settings
AXI_MON3_FILTER7	0x1ac	AXI monitor 3 filter settings
AXI_MON3_FILTER8	0x1b0	AXI monitor 3 filter settings
AXI_MON3_RPT0	0x1c0	AXI monitor 3 cycle count
AXI_MON3_RPT1	0x1c4	AXI monitor 3 hit count
AXI_MON3_RPT2	0x1c8	AXI monitor 3 byte count
AXI_MON3_RPT3	0x1cc	AXI monitor 3 latency count
AXI_MON4_CTRL	0x200	AXI monitor 4 control
AXI_MON4_INPUT	0x204	AXI monitor 4 input selection
AXI_MON4_FILTER0	0x210	AXI monitor 4 filter settings
AXI_MON4_FILTER1	0x214	AXI monitor 4 filter settings
AXI_MON4_FILTER2	0x218	AXI monitor 4 filter settings
AXI_MON4_FILTER3	0x21c	AXI monitor 4 filter settings
AXI_MON4_FILTER4	0x220	AXI monitor 4 filter settings
AXI_MON4_FILTER5	0x224	AXI monitor 4 filter settings
AXI_MON4_FILTER6	0x228	AXI monitor 4 filter settings
AXI_MON4_FILTER7	0x22c	AXI monitor 4 filter settings
AXI_MON4_FILTER8	0x230	AXI monitor 4 filter settings
AXI_MON4_RPT0	0x240	AXI monitor 4 cycle count
AXI_MON4_RPT1	0x244	AXI monitor 4 hit count
AXI_MON4_RPT2	0x248	AXI monitor 4 byte count
AXI_MON4_RPT3	0x24c	AXI monitor 4 latency count
AXI_MON5_CTRL	0x280	AXI monitor 5 control
AXI_MON5_INPUT	0x284	AXI monitor 5 input selection
AXI_MON5_FILTER0	0x290	AXI monitor 5 filter settings
AXI_MON5_FILTER1	0x294	AXI monitor 5 filter settings
AXI_MON5_FILTER2	0x298	AXI monitor 5 filter settings
AXI_MON5_FILTER3	0x29c	AXI monitor 5 filter settings
AXI_MON5_FILTER4	0x2a0	AXI monitor 5 filter settings
AXI_MON5_FILTER5	0x2a4	AXI monitor 5 filter settings
AXI_MON5_FILTER6	0x2a8	AXI monitor 5 filter settings
AXI_MON5_FILTER7	0x2ac	AXI monitor 5 filter settings
AXI_MON5_FILTER8	0x2b0	AXI monitor 5 filter settings

Name	Address Offset	Description
AXI_MON5_RPT0	0x2c0	AXI monitor 5 cycle count
AXI_MON5_RPT1	0x2c4	AXI monitor 5 hit count
AXI_MON5_RPT2	0x2c8	AXI monitor 5 byte count
AXI_MON5_RPT3	0x2cc	AXI monitor 5 latency count
AXI_MON6_CTRL	0x300	AXI monitor 6 control
AXI_MON6_INPUT	0x304	AXI monitor 6 input selection
AXI_MON6_FILTER0	0x310	AXI monitor 6 filter settings
AXI_MON6_FILTER1	0x314	AXI monitor 6 filter settings
AXI_MON6_FILTER2	0x318	AXI monitor 6 filter settings
AXI_MON6_FILTER3	0x31c	AXI monitor 6 filter settings
AXI_MON6_FILTER4	0x320	AXI monitor 6 filter settings
AXI_MON6_FILTER5	0x324	AXI monitor 6 filter settings
AXI_MON6_FILTER6	0x328	AXI monitor 6 filter settings
AXI_MON6_FILTER7	0x32c	AXI monitor 6 filter settings
AXI_MON6_FILTER8	0x330	AXI monitor 6 filter settings
AXI_MON6_RPT0	0x340	AXI monitor 6 cycle count
AXI_MON6_RPT1	0x344	AXI monitor 6 hit count
AXI_MON6_RPT2	0x348	AXI monitor 6 byte count
AXI_MON6_RPT3	0x34c	AXI monitor 6 latency count
AXI_MON7_CTRL	0x380	AXI monitor 7 control
AXI_MON7_INPUT	0x384	AXI monitor 7 input selection
AXI_MON7_FILTER0	0x390	AXI monitor 7 filter settings
AXI_MON7_FILTER1	0x394	AXI monitor 7 filter settings
AXI_MON7_FILTER2	0x398	AXI monitor 7 filter settings
AXI_MON7_FILTER3	0x39c	AXI monitor 7 filter settings
AXI_MON7_FILTER4	0x3a0	AXI monitor 7 filter settings
AXI_MON7_FILTER5	0x3a4	AXI monitor 7 filter settings
AXI_MON7_FILTER6	0x3a8	AXI monitor 7 filter settings
AXI_MON7_FILTER7	0x3ac	AXI monitor 7 filter settings
AXI_MON7_FILTER8	0x3b0	AXI monitor 7 filter settings
AXI_MON7_RPT0	0x3c0	AXI monitor 7 cycle count
AXI_MON7_RPT1	0x3c4	AXI monitor 7 hit count
AXI_MON7_RPT2	0x3c8	AXI monitor 7 byte count
AXI_MON7_RPT3	0x3cc	AXI monitor 7 latency count
AXI_MON8_CTRL	0x400	AXI monitor 8 control
AXI_MON8_INPUT	0x404	AXI monitor 8 input selection
AXI_MON8_FILTER0	0x410	AXI monitor 8 filter settings
AXI_MON8_FILTER1	0x414	AXI monitor 8 filter settings
AXI_MON8_FILTER2	0x418	AXI monitor 8 filter settings
AXI_MON8_FILTER3	0x41c	AXI monitor 8 filter settings
AXI_MON8_FILTER4	0x420	AXI monitor 8 filter settings
AXI_MON8_FILTER5	0x424	AXI monitor 8 filter settings
AXI_MON8_FILTER6	0x428	AXI monitor 8 filter settings
AXI_MON8_FILTER7	0x42c	AXI monitor 8 filter settings
AXI_MON8_FILTER8	0x430	AXI monitor 8 filter settings
AXI_MON8_RPT0	0x440	AXI monitor 8 cycle count
AXI_MON8_RPT1	0x444	AXI monitor 8 hit count
AXI_MON8_RPT2	0x448	AXI monitor 8 byte count
AXI_MON8_RPT3	0x44c	AXI monitor 8 latency count
AXI_MON9_CTRL	0x480	AXI monitor 9 control
AXI_MON9_INPUT	0x484	AXI monitor 9 input selection
AXI_MON9_FILTER0	0x490	AXI monitor 9 filter settings
AXI_MON9_FILTER1	0x494	AXI monitor 9 filter settings
AXI_MON9_FILTER2	0x498	AXI monitor 9 filter settings
AXI_MON9_FILTER3	0x49c	AXI monitor 9 filter settings

Name	Address Offset	Description
AXI_MON9_FILTER4	0x4a0	AXI monitor 9 filter settings
AXI_MON9_FILTER5	0x4a4	AXI monitor 9 filter settings
AXI_MON9_FILTER6	0x4a8	AXI monitor 9 filter settings
AXI_MON9_FILTER7	0x4ac	AXI monitor 9 filter settings
AXI_MON9_FILTER8	0x4b0	AXI monitor 9 filter settings
AXI_MON9_RPT0	0x4c0	AXI monitor 9 cycle count
AXI_MON9_RPT1	0x4c4	AXI monitor 9 hit count
AXI_MON9_RPT2	0x4c8	AXI monitor 9 byte count
AXI_MON9_RPT3	0x4cc	AXI monitor 9 latency count
AXI_MON10_CTRL	0x500	AXI monitor 10 control
AXI_MON10_INPUT	0x504	AXI monitor 10 input selection
AXI_MON10_FILTER0	0x510	AXI monitor 10 filter settings
AXI_MON10_FILTER1	0x514	AXI monitor 10 filter settings
AXI_MON10_FILTER2	0x518	AXI monitor 10 filter settings
AXI_MON10_FILTER3	0x51c	AXI monitor 10 filter settings
AXI_MON10_FILTER4	0x520	AXI monitor 10 filter settings
AXI_MON10_FILTER5	0x524	AXI monitor 10 filter settings
AXI_MON10_FILTER6	0x528	AXI monitor 10 filter settings
AXI_MON10_FILTER7	0x52c	AXI monitor 10 filter settings
AXI_MON10_FILTER8	0x530	AXI monitor 10 filter settings
AXI_MON10_RPT0	0x540	AXI monitor 10 cycle count
AXI_MON10_RPT1	0x544	AXI monitor 10 hit count
AXI_MON10_RPT2	0x548	AXI monitor 10 byte count
AXI_MON10_RPT3	0x54c	AXI monitor 10 latency count
AXI_MON11_CTRL	0x580	AXI monitor 11 control
AXI_MON11_INPUT	0x584	AXI monitor 11 input selection
AXI_MON11_FILTER0	0x590	AXI monitor 11 filter settings
AXI_MON11_FILTER1	0x594	AXI monitor 11 filter settings
AXI_MON11_FILTER2	0x598	AXI monitor 11 filter settings
AXI_MON11_FILTER3	0x59c	AXI monitor 11 filter settings
AXI_MON11_FILTER4	0x5a0	AXI monitor 11 filter settings
AXI_MON11_FILTER5	0x5a4	AXI monitor 11 filter settings
AXI_MON11_FILTER6	0x5a8	AXI monitor 11 filter settings
AXI_MON11_FILTER7	0x5ac	AXI monitor 11 filter settings
AXI_MON11_FILTER8	0x5b0	AXI monitor 11 filter settings
AXI_MON11_RPT0	0x5c0	AXI monitor 11 cycle count
AXI_MON11_RPT1	0x5c4	AXI monitor 11 hit count
AXI_MON11_RPT2	0x5c8	AXI monitor 11 byte count
AXI_MON11_RPT3	0x5cc	AXI monitor 11 latency count

14.1.5.2 AXI 寄存器描述

基址 0x0800_4000

AXI_CTRL0_1

Offset Address: 0x4b4

Bits	Name	Access	Description	Reset
9:0	axi1_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0

Bits	Name	Access	Description	Reset
11:10	Reserved			
12	axi1_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_1

Offset Address: 0x4b8

Bits	Name	Access	Description	Reset
9:0	axi1_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi1_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

AXI_CTRL0_2

Offset Address: 0x564

Bits	Name	Access	Description	Reset
9:0	axi2_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_2

Offset Address: 0x568

Bits	Name	Access	Description	Reset
9:0	axi2_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

AXI_CTRL0_3

Offset Address: 0x614

Bits	Name	Access	Description	Reset
9:0	axi3_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest	0x0

Bits	Name	Access	Description	Reset
			priority.	
11:10	Reserved			
12	axi3_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_3

Offset Address: 0x618

Bits	Name	Access	Description	Reset
9:0	axi3_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi3_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

基址 0x0800_8000

AXI_MON0_CTRL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	axi_mon0_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon0_clear	R/W	Clear all the counter.	0x0
2	axi_mon0_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon0_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon0_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon0_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON0_INPUT

Offset Address: 0x004

Bits	Name	Access	Description	Reset
5:0	axi_mon0_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON0_FILTER0

Offset Address: 0x010

Bits	Name	Access	Description	Reset
9:0	axi_mon0_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask	0x0

Bits	Name	Access	Description	Reset
			bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	
31:10	Reserved			

AXI_MON0_FILTER1

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON0_FILTER2

Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON0_FILTER3

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON0_FILTER4

Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON0_FILTER5

Offset Address: 0x024

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON0_FILTER6

Offset Address: 0x028

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON0_FILTER7

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon0_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon0_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON0_FILTER8

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	axi_mon0_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon0_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon0_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon0_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON0_RPT0

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	axi_mon0_cycle_count	RO	AXI monitor 0 cycle count, counting after func_en assert	

AXI_MON0_RPT1

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_count	RO	AXI monitor 0 hit count, counting after func_en assert	

AXI_MON0_RPT2

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	axi_mon0_byte_count	RO	AXI monitor 0 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON0_RPT3

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_latency_count	RO	AXI monitor 0 latency count, counting after func_en assert, += oustanding	

AXI_MON1_CTRL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	axi_mon1_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon1_clear	R/W	Clear all the counter.	0x0
2	axi_mon1_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon1_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0

Bits	Name	Access	Description	Reset
5	axi_mon1_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon1_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON1_INPUT

Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	axi_mon1_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON1_FILTER0

Offset Address: 0x090

Bits	Name	Access	Description	Reset
9:0	axi_mon1_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON1_FILTER1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON1_FILTER2

Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON1_FILTER3

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON1_FILTER4

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON1_FILTER5

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON1_FILTER6

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON1_FILTER7

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon1_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon1_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON1_FILTER8

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	axi_mon1_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon1_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon1_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon1_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON1_RPT0

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	axi_mon1_cycle_count	RO	AXI monitor 1 cycle count, counting after func_en assert	

AXI_MON1_RPT1

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_count	RO	AXI monitor 1 hit count, counting after func_en assert	

AXI_MON1_RPT2

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	axi_mon1_byte_count	RO	AXI monitor 1 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON1_RPT3

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	axi_mon1_latency_count	RO	AXI monitor 1 latency count, counting after func_en assert, += oustanding	

AXI_MON2_CTRL

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	axi_mon2_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon2_clear	R/W	Clear all the counter.	0x0
2	axi_mon2_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon2_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon2_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon2_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON2_INPUT

Offset Address: 0x104

Bits	Name	Access	Description	Reset
5:0	axi_mon2_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON2_FILTER0

Offset Address: 0x110

Bits	Name	Access	Description	Reset
9:0	axi_mon2_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON2_FILTER1

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON2_FILTER2

Offset Address: 0x118

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON2_FILTER3

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON2_FILTER4

Offset Address: 0x120

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON2_FILTER5

Offset Address: 0x124

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON2_FILTER6

Offset Address: 0x128

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON2_FILTER7

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon2_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon2_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON2_FILTER8

Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	axi_mon2_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon2_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0

Bits	Name	Access	Description	Reset
10:8	axi_mon2_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon2_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON2_RPT0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	axi_mon2_cycle_count	RO	AXI monitor 2 cycle count, counting after func_en assert	

AXI_MON2_RPT1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_count	RO	AXI monitor 2 hit count, counting after func_en assert	

AXI_MON2_RPT2

Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	axi_mon2_byte_count	RO	AXI monitor 2 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON2_RPT3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_latency_count	RO	AXI monitor 2 latency count, counting after func_en assert, += oustanding	

AXI_MON3_CTRL

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	axi_mon3_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon3_clear	R/W	Clear all the counter.	0x0
2	axi_mon3_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon3_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon3_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon3_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON3_INPUT

Offset Address: 0x184

Bits	Name	Access	Description	Reset
5:0	axi_mon3_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON3_FILTER0

Offset Address: 0x190

Bits	Name	Access	Description	Reset
9:0	axi_mon3_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON3_FILTER1

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON3_FILTER2

Offset Address: 0x198

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON3_FILTER3

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON3_FILTER4

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON3_FILTER5

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON3_FILTER6

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON3_FILTER7

Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon3_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon3_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON3_FILTER8

Offset Address: 0x1b0

Bits	Name	Access	Description	Reset
0	axi_mon3_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon3_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon3_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon3_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON3_RPT0

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	axi_mon3_cycle_count	RO	AXI monitor 3 cycle count, counting after func_en assert	

AXI_MON3_RPT1

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_count	RO	AXI monitor 3 hit count, counting after func_en assert	

AXI_MON3_RPT2

Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	axi_mon3_byte_count	RO	AXI monitor 3 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON3_RPT3

Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	axi_mon3_latency_count	RO	AXI monitor 3 latency count, counting after func_en assert, += oustanding	

AXI_MON4_CTRL

Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	axi_mon4_en	R/W	If set to 1, enables the AXI monitor function.	0x0

Bits	Name	Access	Description	Reset
1	axi_mon4_clear	R/W	Clear all the counter.	0x0
2	axi_mon4_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon4_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon4_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon4_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON4_INPUT

Offset Address: 0x204

Bits	Name	Access	Description	Reset
5:0	axi_mon4_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON4_FILTER0

Offset Address: 0x210

Bits	Name	Access	Description	Reset
9:0	axi_mon4_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON4_FILTER1

Offset Address: 0x214

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON4_FILTER2

Offset Address: 0x218

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON4_FILTER3

Offset Address: 0x21c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON4_FILTER4

Offset Address: 0x220

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON4_FILTER5

Offset Address: 0x224

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON4_FILTER6

Offset Address: 0x228

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON4_FILTER7

Offset Address: 0x22c

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon4_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon4_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON4_FILTER8

Offset Address: 0x230

Bits	Name	Access	Description	Reset
0	axi_mon4_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon4_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon4_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon4_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON4_RPT0

Offset Address: 0x240

Bits	Name	Access	Description	Reset
31:0	axi_mon4_cycle_count	RO	AXI monitor 4 cycle count, counting after func_en assert	

AXI_MON4_RPT1

Offset Address: 0x244

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_count	RO	AXI monitor 4 hit count, counting after	

Bits	Name	Access	Description	Reset
			func_en assert	

AXI_MON4_RPT2

Offset Address: 0x248

Bits	Name	Access	Description	Reset
31:0	axi_mon4_byte_count	RO	AXI monitor 4 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON4_RPT3

Offset Address: 0x24c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_latency_count	RO	AXI monitor 4 latency count, counting after func_en assert, += oustanding	

AXI_MON5_CTRL

Offset Address: 0x280

Bits	Name	Access	Description	Reset
0	axi_mon5_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon5_clear	R/W	Clear all the counter.	0x0
2	axi_mon5_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon5_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon5_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon5_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON5_INPUT

Offset Address: 0x284

Bits	Name	Access	Description	Reset
5:0	axi_mon5_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON5_FILTER0

Offset Address: 0x290

Bits	Name	Access	Description	Reset
9:0	axi_mon5_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON5_FILTER1

Offset Address: 0x294

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON5_FILTER2

Offset Address: 0x298

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON5_FILTER3

Offset Address: 0x29c

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON5_FILTER4

Offset Address: 0x2a0

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON5_FILTER5

Offset Address: 0x2a4

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON5_FILTER6

Offset Address: 0x2a8

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON5_FILTER7

Offset Address: 0x2ac

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon5_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon5_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON5_FILTER8

Offset Address: 0x2b0

Bits	Name	Access	Description	Reset
0	axi_mon5_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0

Bits	Name	Access	Description	Reset
3:1	Reserved			
7:4	axi_mon5_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon5_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon5_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON5_RPT0

Offset Address: 0x2c0

Bits	Name	Access	Description	Reset
31:0	axi_mon5_cycle_count	RO	AXI monitor 5 cycle count, counting after func_en assert	

AXI_MON5_RPT1

Offset Address: 0x2c4

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_count	RO	AXI monitor 5 hit count, counting after func_en assert	

AXI_MON5_RPT2

Offset Address: 0x2c8

Bits	Name	Access	Description	Reset
31:0	axi_mon5_byte_count	RO	AXI monitor 5 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON5_RPT3

Offset Address: 0x2cc

Bits	Name	Access	Description	Reset
31:0	axi_mon5_latency_count	RO	AXI monitor 5 latency count, counting after func_en assert, += oustanding	

AXI_MON6_CTRL

Offset Address: 0x300

Bits	Name	Access	Description	Reset
0	axi_mon6_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon6_clear	R/W	Clear all the counter.	0x0
2	axi_mon6_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon6_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon6_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon6_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON6_INPUT

Offset Address: 0x304

Bits	Name	Access	Description	Reset
5:0	axi_mon6_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0

Bits	Name	Access	Description	Reset
31:6	Reserved			

AXI_MON6_FILTER0

Offset Address: 0x310

Bits	Name	Access	Description	Reset
9:0	axi_mon6_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON6_FILTER1

Offset Address: 0x314

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON6_FILTER2

Offset Address: 0x318

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON6_FILTER3

Offset Address: 0x31c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON6_FILTER4

Offset Address: 0x320

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON6_FILTER5

Offset Address: 0x324

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON6_FILTER6

Offset Address: 0x328

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON6_FILTER7

Offset Address: 0x32c

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon6_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon6_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON6_FILTER8

Offset Address: 0x330

Bits	Name	Access	Description	Reset
0	axi_mon6_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon6_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon6_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon6_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON6_RPT0

Offset Address: 0x340

Bits	Name	Access	Description	Reset
31:0	axi_mon6_cycle_count	RO	AXI monitor 6 cycle count, counting after func_en assert	

AXI_MON6_RPT1

Offset Address: 0x344

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_count	RO	AXI monitor 6 hit count, counting after func_en assert	

AXI_MON6_RPT2

Offset Address: 0x348

Bits	Name	Access	Description	Reset
31:0	axi_mon6_byte_count	RO	AXI monitor 6 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON6_RPT3

Offset Address: 0x34c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_latency_count	RO	AXI monitor 6 latency count, counting after func_en assert, += oustanding	

AXI_MON7_CTRL

Offset Address: 0x380

Bits	Name	Access	Description	Reset
0	axi_mon7_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon7_clear	R/W	Clear all the counter.	0x0
2	axi_mon7_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon7_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon7_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon7_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON7_INPUT

Offset Address: 0x384

Bits	Name	Access	Description	Reset
5:0	axi_mon7_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON7_FILTER0

Offset Address: 0x390

Bits	Name	Access	Description	Reset
9:0	axi_mon7_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON7_FILTER1

Offset Address: 0x394

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON7_FILTER2

Offset Address: 0x398

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON7_FILTER3

Offset Address: 0x39c

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON7_FILTER4

Offset Address: 0x3a0

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON7_FILTER5

Offset Address: 0x3a4

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON7_FILTER6

Offset Address: 0x3a8

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON7_FILTER7

Offset Address: 0x3ac

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon7_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon7_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON7_FILTER8

Offset Address: 0x3b0

Bits	Name	Access	Description	Reset
0	axi_mon7_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon7_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon7_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon7_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON7_RPT0

Offset Address: 0x3c0

Bits	Name	Access	Description	Reset
31:0	axi_mon7_cycle_count	RO	AXI monitor 7 cycle count, counting after func_en assert	

AXI_MON7_RPT1

Offset Address: 0x3c4

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_count	RO	AXI monitor 7 hit count, counting after func_en assert	

AXI_MON7_RPT2

Offset Address: 0x3c8

Bits	Name	Access	Description	Reset
31:0	axi_mon7_byte_count	RO	AXI monitor 7 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON7_RPT3

Offset Address: 0x3cc

Bits	Name	Access	Description	Reset
31:0	axi_mon7_latency_count	RO	AXI monitor 7 latency count, counting after func_en assert, += oustanding	

AXI_MON8_CTRL

Offset Address: 0x400

Bits	Name	Access	Description	Reset
0	axi_mon8_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon8_clear	R/W	Clear all the counter.	0x0
2	axi_mon8_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon8_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon8_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon8_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON8_INPUT

Offset Address: 0x404

Bits	Name	Access	Description	Reset
5:0	axi_mon8_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON8_FILTER0

Offset Address: 0x410

Bits	Name	Access	Description	Reset
9:0	axi_mon8_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0

Bits	Name	Access	Description	Reset
31:10	Reserved			

AXI_MON8_FILTER1

Offset Address: 0x414

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON8_FILTER2

Offset Address: 0x418

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON8_FILTER3

Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON8_FILTER4

Offset Address: 0x420

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON8_FILTER5

Offset Address: 0x424

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON8_FILTER6

Offset Address: 0x428

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON8_FILTER7

Offset Address: 0x42c

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon8_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon8_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON8_FILTER8

Offset Address: 0x430

Bits	Name	Access	Description	Reset
0	axi_mon8_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon8_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon8_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon8_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON8_RPT0

Offset Address: 0x440

Bits	Name	Access	Description	Reset
31:0	axi_mon8_cycle_count	RO	AXI monitor 8 cycle count, counting after func_en assert	

AXI_MON8_RPT1

Offset Address: 0x444

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_count	RO	AXI monitor 8 hit count, counting after func_en assert	

AXI_MON8_RPT2

Offset Address: 0x448

Bits	Name	Access	Description	Reset
31:0	axi_mon8_byte_count	RO	AXI monitor 8 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON8_RPT3

Offset Address: 0x44c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_latency_count	RO	AXI monitor 8 latency count, counting after func_en assert, += oustanding	

AXI_MON9_CTRL

Offset Address: 0x480

Bits	Name	Access	Description	Reset
0	axi_mon9_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon9_clear	R/W	Clear all the counter.	0x0
2	axi_mon9_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon9_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon9_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon9_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON9_INPUT

Offset Address: 0x484

Bits	Name	Access	Description	Reset
5:0	axi_mon9_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON9_FILTER0

Offset Address: 0x490

Bits	Name	Access	Description	Reset
9:0	axi_mon9_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON9_FILTER1

Offset Address: 0x494

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON9_FILTER2

Offset Address: 0x498

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON9_FILTER3

Offset Address: 0x49c

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON9_FILTER4

Offset Address: 0x4a0

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON9_FILTER5

Offset Address: 0x4a4

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON9_FILTER6

Offset Address: 0x4a8

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON9_FILTER7

Offset Address: 0x4ac

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon9_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon9_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON9_FILTER8

Offset Address: 0x4b0

Bits	Name	Access	Description	Reset
0	axi_mon9_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon9_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon9_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon9_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON9_RPT0

Offset Address: 0x4c0

Bits	Name	Access	Description	Reset
31:0	axi_mon9_cycle_count	RO	AXI monitor 9 cycle count, counting after func_en assert	

AXI_MON9_RPT1

Offset Address: 0x4c4

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_count	RO	AXI monitor 9 hit count, counting after func_en assert	

AXI_MON9_RPT2

Offset Address: 0x4c8

Bits	Name	Access	Description	Reset
31:0	axi_mon9_byte_count	RO	AXI monitor 9 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON9_RPT3

Offset Address: 0x4cc

Bits	Name	Access	Description	Reset
31:0	axi_mon9_latency_count	RO	AXI monitor 9 latency count, counting after func_en assert, += oustanding	

AXI_MON10_CTRL

Offset Address: 0x500

Bits	Name	Access	Description	Reset
0	axi_mon10_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon10_clear	R/W	Clear all the counter.	0x0
2	axi_mon10_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon10_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon10_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon10_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON10_INPUT

Offset Address: 0x504

Bits	Name	Access	Description	Reset
5:0	axi_mon10_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON10_FILTER0

Offset Address: 0x510

Bits	Name	Access	Description	Reset
9:0	axi_mon10_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON10_FILTER1

Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON10_FILTER2

Offset Address: 0x518

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON10_FILTER3

Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON10_FILTER4

Offset Address: 0x520

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON10_FILTER5

Offset Address: 0x524

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON10_FILTER6

Offset Address: 0x528

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON10_FILTER7

Offset Address: 0x52c

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon10_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon10_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON10_FILTER8

Offset Address: 0x530

Bits	Name	Access	Description	Reset
0	axi_mon10_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon10_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon10_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon10_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON10_RPT0

Offset Address: 0x540

Bits	Name	Access	Description	Reset
31:0	axi_mon10_cycle_count	RO	AXI monitor 10 cycle count, counting after func_en assert	

AXI_MON10_RPT1

Offset Address: 0x544

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_count	RO	AXI monitor 10 hit count, counting after func_en assert	

AXI_MON10_RPT2

Offset Address: 0x548

Bits	Name	Access	Description	Reset
31:0	axi_mon10_byte_count	RO	AXI monitor 10 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON10_RPT3

Offset Address: 0x54c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_latency_count	RO	AXI monitor 10 latency count, counting after func_en assert, += oustanding	

AXI_MON11_CTRL

Offset Address: 0x580

Bits	Name	Access	Description	Reset
0	axi_mon11_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon11_clear	R/W	Clear all the counter.	0x0
2	axi_mon11_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon11_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon11_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon11_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON11_INPUT

Offset Address: 0x584

Bits	Name	Access	Description	Reset
5:0	axi_mon11_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON11_FILTER0

Offset Address: 0x590

Bits	Name	Access	Description	Reset
9:0	axi_mon11_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB	0x0

Bits	Name	Access	Description	Reset
			boundary	
31:10	Reserved			

AXI_MON11_FILTER1

Offset Address: 0x594

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON11_FILTER2

Offset Address: 0x598

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON11_FILTER3

Offset Address: 0x59c

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON11_FILTER4

Offset Address: 0x5a0

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON11_FILTER5

Offset Address: 0x5a4

Bits	Name	Access	Description	Reset
23:0	axi_mon11_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON11_FILTER6

Offset Address: 0x5a8

Bits	Name	Access	Description	Reset
23:0	axi_mon11_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON11_FILTER7

Offset Address: 0x5ac

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon11_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon11_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON11_FILTER8

Offset Address: 0x5b0

Bits	Name	Access	Description	Reset
0	axi_mon11_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon11_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon11_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon11_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON11_RPT0

Offset Address: 0x5c0

Bits	Name	Access	Description	Reset
31:0	axi_mon11_cycle_count	RO	AXI monitor 11 cycle count, counting after func_en assert	

AXI_MON11_RPT1

Offset Address: 0x5c4

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_count	RO	AXI monitor 11 hit count, counting after func_en assert	

AXI_MON11_RPT2

Offset Address: 0x5c8

Bits	Name	Access	Description	Reset
31:0	axi_mon11_byte_count	RO	AXI monitor 11 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON11_RPT3

Offset Address: 0x5cc

Bits	Name	Access	Description	Reset
31:0	axi_mon11_latency_count	RO	AXI monitor 11 latency count, counting after func_en assert, += oustanding	

14.1.6 DDRC 寄存器

14.1.6.1 DDRC 寄存器概览

基址 0x0800_4000

Name	Address Offset	Description
DRAM_REF_CTRL	0x064	DRAM refresh parameter
DRAM_MRD0	0x0dc	DRAM MR value
DRAM_MRD1	0x0e0	DRAM MR value

14.1.6.2 DDRC 寄存器描述

基址 0x0800_4000

DRAM_REF_CTRL

Offset Address: 0x064

Bits	Name	Access	Description	Reset
9:0	t_rfc	R/W	Specify tRFC Unit: ddr core clock cycles	0x8c
15:10	Reserved			
27:16	t_refi	R/W	Specify tREFI Unit: 32 ddr core clocks	0x62
31:28	Reserved			

DRAM_MRD0

Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
15:0	ddr_mr1	R/W	DDR3: Write value for MR1 register	0x510
31:16	ddr_mr0	R/W	DDR3: Write value for MR0 register	0x0

DRAM_MRD1

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
15:0	ddr_mr3	R/W	DDR3: Write value for MR3 register	0x0
31:16	ddr_mr2	R/W	DDR3: Write value for MR2 register	0x0

14.2 SPI NOR Flash 控制器

14.2.1 概述

提供片外 SPI NOR Flash 数据存取。

14.2.2 特点

- 支持外接 1 片选。
- 支持 Dual/Qual 读写操作。
- 支持多种规格器件。
  - 支持 3Byte 地址器件及 4Byte 地址器件。
  - 支时最大 256MB 容量器件。
- 支持 BOOT 功能。

14.2.3 功能描述

14.2.3.1 接口描述

SPI NOR Flash 控制器 可以支持三种SPI NOR接口类型，类型分别为Standard SPI、Dual SPI接口模式及Qual SPI接口模式。

- Standard SPI接口模式:

Standard SPI接口模式具有1bit数据输入线和1bit数据输出线。图表为Standard SPI接口模式写操作时序图，为Standard SPI接口模式读操作时序图。

图表 14-2 Standard SPI 接口模式写操作时序图

时序说明：

- ✓ command/address/dummy cycles 以单bit串行方式在DO 线上输出。
- ✓ Data 以单bit串行方式在DO线上输出。

图表 14-3 Standard SPI 接口模式读操作时序图

时序说明：

- ✓ command/address/dummy cycles 以单bit串行方式在DO 线上输出。
- ✓ Data 以单 bit 串行方式在 DI 线上输入。

- Dual-Input SPI接口模式:

Dual Input SPI接口模式，数据输入阶段并行2bit数据线。图表4-4 为 Dual Input SPI接口模式操作时序图。

图表 14-4 Dual-Input SPI 接口时序

时序说明：

- ✓ Command/address/dummy cycles以单Bit串行方式在DO 线上输出。
- ✓ Data以2 Bits方式在DO/DI线上或输入（读）。

- Dual-IO SPI接口模式:

Dual IO SPI接口模式，地址输出、数据输入阶段并行2bit数据线。图表4-4 为 Dual IO SPI接口模式操作时序图。

图表 14-5 Dual-IO SPI 接口时序

时序说明：

- ✓ Command 以单Bit串行方式在DO 线上输出。
- ✓ address/dummy cycles/Data以2 Bits方式在DO/DI线上输出（写）或输入（读）。

- Quad-Input SPI接口模式:

Quad Input SPI接口模式，数据输入阶段并行4bit数据线。图表4-6为 Quad Input SPI接口模式操作时序图。

图表 14-6 Quad-Input SPI 模式时序图

时序说明：

- ✓ Command/address/dummy cycles 以单Bit串行方式在DO 线上输出。
- ✓ Data以4 Bits方式在DO/DI/WPN/HOLDN 输入（读）。

- Quad-IO SPI接口模式:

Quad IO SPI接口模式，地址输出、数据输入阶段并行4bit数据线。图表4-7为 Quad IO SPI接口模式操作时序图。

图表 14-7 Quad-IO SPI 模式时序图

时序说明：

- ✓ Command 以单Bit串行方式在DO 线上输出。
- ✓ address/dummy cycles/Data 以4 Bits方式在DO/DI/WPN/HOLDN 输出（写）或输入（读）。

14.2.3.2 Boot 功能

SPI NOR Boot 数据位于芯片地址 0x1000_0000~0x1FFF_FFFF，直接映射到 SPI NOR Flash 的连续地址空间 0x0000_0000~0x0FFF_FFFF。SPI_NOR Flash 最大可支持到 256MB，其中若需使用大于 16MB SPI_NOR Flash，需利用到 4 bytes 地址模式。芯片复位状态为 3bytes 地址模式，需透过配置使能 4bytes 地址模式，故 SPI_NOR Flash 需要能支持 3bytes/4bytes 地址模式。

14.2.3.3 寄存器操作

软件配置操作相关寄存器，如操作命令、地址等，最后配置 reg_go_busy 寄存器下发操作，控制器根据软件配置值，下发操作给器件。

14.2.3.4 DMA 操作

- DMMR 读模式

在 SPI_NOR Flash Controller 处于 DMMR 模式下，SPI_NOR Flash 空间直接映射至芯片地址空间 0x1000_0000~0x1FFF_FFFF。系统 DMA 可使用 memory-to-memory 模式，将 SPI_NOR 数据搬移至 DDR。

- Non-DMMR 读写模式

指令、地址和数据都须透过 FF_PORT 发送接收。需配置控制器寄存器选择读读指令、写指令、指令长度、数据长度后透过 CPU 或 DMA 写 FF_PORT 以发出指令及地址，透

过写、读 FF_PORT 以发送、接收数据。

14.2.4 工作流程

14.2.4.1 初始化流程

步骤1. 如果需要调整 Timing 参数，根据器件配置 SPI Clock Divider。
步骤2. 配置中断控制寄存器。

14.2.4.2 器件状态寄存器操作

步骤1. 配置传输数据长度。
步骤2. 传输模式相关配置。
步骤3. 配置 reg_go_busy。
步骤4. 将传输内容写入缓存。
步骤5. 检测 INT_STS，等待操作完成。

14.2.4.3 SPI NOR Flash 地址模式切换流程

对于 SPI NOR Flash 器件，支持 3 Byte 与 4 Byte 两种 Flash 地址模式，可以在芯片启动起来之后通过配置寄存器动态切换地址模式。在芯片启动起来之后切换 Flash 地址模式的步骤如下:

步骤1. 无 Flash 操作或保证之前对 Flash 器件之操作完成。
步骤2. 根据器件要求，用寄存器操作方式配置器件的相关寄存器发特定命令配置 Flash 进入 4 Byte 模式。
步骤3. 配置 SPI NOR Flash 控制器[reg_byte4en]式为 4 Byte 模式，完成 3 Byte 模式到 4Byte 模式切换。

14.2.4.4 DMA 读操作流程

步骤1. 禁止 dmmr 模式，禁止 dma_en。
步骤2. 写 FF_PT 为 1，以清空 FIFO 及重置读写指标。
步骤3. 使能 dmmr 模式。
步骤4. 配置系统 DMA 为 mem-to-mem 搬移。 DST_TR_WIDTH = 0x2 (transaction width is 32bit)、DST_MSIZE = 0x0 (burst transaction length =1)、BLOCK_TS = TRAN_NUM/4 -1 。 实际上之 BLOCK_TS/ DST_TR_WIDTH/ DST_MSIZE 需根据实际传输长度做适当的配置。
步骤5. 使能系统 DMA 选定通道。开始搬移

步骤6. 等待ＤＭＡ相应通道中断。表示ＤＭＡ读取完成。

14.2.4.5 DMA 写操作流程

步骤1. 禁止 dmmr 模式，禁止 dma_en。
步骤2. 写 FF_PT 为 1，以清空 FIFO 及重置读写指标。
步骤3. 配置系统 DMA 通道映射，将选定的 DMA 通道映射至 39:dma_req_spi_nor。
步骤4. 配置系统 DMA 为 peri-to-mem 搬移。 DST_TR_WIDTH = 0x2 (transaction width is 32bit)、DST_MSIZE = 0x0 (burst transaction length =1)、BLOCK_TS = TRAN_NUM/4 -1 。 实际上之 BLOCK_TS/ DST_TR_WIDTH/ DST_MSIZE 需根据实际传输长度做适当的配置。
步骤5. 使能系统 DMA 选定通道。
步骤6. 配置 SPI_NOR 寄存器 TRAN_NUM, 不包含指令及地址
步骤7. 配置 SPI_NOR 寄存器 TRAN_CSR，tran_mode = 0x2 (tx only)、fast_mode、bus_width, addr_bn、dma_en=0 及 reg_go_busy。 Ex: TRAN_CSR = 0x0000BC2A
步骤8. 对 SPI_NOR 寄存器 FF_PORT 写入 command and address
步骤9. 查询 SPI_NOR 寄存器 rdat_ff_pt 为 0 , 确保 command 及 address 发送完毕。
步骤10. 配置 SPI_NOR 寄存器 TRAN_CSR 使能 dma_en，
步骤11. 检测 SPI_NOR 寄存器 INT_STS，等待操作完成，表示缓冲区内容已写入器件。

14.2.4.6 其它注意事项

- 对器件操作未完成前，不可改动相关寄存器配置，否则可能导致操作不正常。

14.2.5 寄存器概览

Base Address 0x10000000

Name	Address Offset	Description
SPI_CTRL	0x000	SPI_NOR 操作控制
CE_CTRL	0x004	CE 操作控制
DLY_CTRL	0x008	时延控制
DMMR_CTRL	0x00c	DMMR 模式控制
TRAN_CSR	0x010	传输控制

Name	Address Offset	Description
TRAN_NUM	0x014	传输帧数
FF_PORT	0x018	FIFO 写入读出口
FF_PT	0x020	FIFO 指标状态
INT_STS	0x028	中断状态
INT_EN	0x02c	中断使能

14.2.6 寄存器描述

SPI_CTRL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
10:0	sck_div	R/W	SPI Clock Divider SCK frequency = HCLK frequency / (2(SckDiv+ 1))	0x9
11	Reserved			
12	cpha	R/W	Clock Phase 0：片选有效后 SCK 第一个时钟沿开始采样数据 1：片选有效后 SCK 第二个时钟沿开始采样数据	0x0
13	cpol	R/W	Clock Polarity 0：SCK 空闲时为低电平 1：SCK 空闲时为高电平	0x0
14	hold_o	R/W	HOLD 管脚输出电平	0x1
15	wp_o	R/W	WP 管脚输出电平	0x1
19:16	frame_len	R/W	发送和接收的帧长度。0 表示帧长度为16 bits；不支持帧长度为1。	0x8
20	lsb_first	R/W	LSBF: Least Significant Bit First 0: Frame MSB first 1: Frame LSB first	0x0
21	srst	R/W	写 1 复位各状态机和中断标志位	0x0
31:22	Reserved			

CE_CTRL

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	ce_manual	R/W	CEManual 控制 CE 脚的电平值	0x0
1	ce_manual_en	R/W	CE Manual Enable 0： CE 脚的电平值受硬件状态机控制 1： CE 脚的电平值受 CEManual 寄存器控制	0x0
31:2	Reserved			

DLY_CTRL

Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	frame_interval	R/W	控制相邻两帧数据的帧间距：T = TSCK * FmIntvl（帧间距内无 SCK 脉冲）。相邻两帧数据的帧间距，为 0 时无帧间距。	0x0
7:4	Reserved			
11:8	cet	R/W	CET 控制一次传输开始前 CE 相对 SCK 第一个时钟沿，提前有效的时间和传输结束后相对于 SCK 最后一个时钟沿继续保持有效的时间。该时间计算为 T = TSCK * (CET+1)	0x3
13:12	smp_en_dly	R/W	接收采样时延选项。于 SCK 正缘往后 delay sample cycle（IP 工作时钟）以进行采样	0x0
14	rx_pipe_ctrl	R/W	接收采样时钟沿选项。 0: 正常采样 1: 采 SCK 负缘采样，以达高速传输	0x0
31:15	Reserved			

DMMR_CTRL

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	dmmr_mode	R/W	dmmr_mode.该位为 1 时 AHB 上的读地址将被直接映射到 SPI Flash 中,控制器自动从 SPI Flash 相应地址读取数据而无需软件设置相关命令和地址.此时可将 SPI Flash 当作 ROM 使用.DMMR 为 1 时,IP 内寄存器可写但不可读.	0x1
31:1	Reserved			

TRAN_CSR

Offset Address: 0x010

Bits	Name	Access	Description	Reset
1:0	tran_mode	R/W	Transfer Mode 00: No Tx, No Rx 01: Rx only 10: Tx only 11: Tx and Rx TranMode 表示除命令和地址外的传输数据的收发模式。	0x0
2	Reserved			
3	fast_mode	R/W	FastMode: 0: Normal Mode 1: Fast Mode	0x0
5:4	bus_width	R/W	Bus Width 00: 1 bit bus 01: 2 bit bus 10: 4 bit bus 11: Reserved	0x0
6	dma_en	R/W	0: DMA Disable 1: DMA Enable 当 TranMode 为 11 时（发送接收同时进行）不支持 DMA 传输	0x0
7	miso_cked	R/W	miso_i 引脚的电平值	0x0

Bits	Name	Access	Description	Reset
10:8	addr_bn	R/W	Address Byte Number 表示当前 Flash 传输地址字段的字节数，0 表示无地址字段	0x3
11	with_cmd	R/W	With Command 0: 当前传输不带命令 1: 当前传输带命令	0x1
13:12	ff_trg_lvl	R/W	FFTrgLvl 控制 FIFO 在何种条件下产生中断和 DMA 请求。 00：1 Byte 01：2 Bytes 10：4 Bytes 11：8 Bytes 对于 Transmit，当 FIFO 中空闲 Byte 数大于等于 FFTrgLvl 定义的 Byte 数时生成中断和 DMA 请求； 对于 Receive，当 FIFO 中有效 Byte 数大于等于 FFTrgLvl 定义的 Byte 数时生成中断和 DMA 请求。	0x3
14	Reserved			
15	go_busy	R/W	向该位写 0 不起作用，写 1 将该位置 1 并启动一次传输，传输结束后该位自动清零。发起新的传输前，软件应查询该寄存器，该寄存器为 0 时才可发起新传输。	0x0
19:16	dummy_cyc	R/W	dummy cycle count	0x0
20	byte4en	R/W	4 bytes address cycle enable in dmmr_mode	0x0
21	byte4cmd	R/W	4 bytes address cmd enable in dmmr_mode	0x0
31:22	Reserved			

TRAN_NUM

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	rdat_tran_num	R/W	非 dmmr_mode 下 TRAN_NUM 为一次传输中收发的帧数	0x0
31:16	Reserved			

FF_PORT

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	rdat_ff_port	R/W	FIFO 写入读出地址	0x0

FF_PT

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	rdat_ff_pt	R/W	读为 FIFO 中有效数据字节数，写为 FIFO 清空	0x0
7:4	Reserved			
9:8	wrcnt	R/W	当前 fifo ,写入字节偏移指标状态	0x0
12:10	rdpt	R/W	当前 fifo ,读取字节 偏移指标状态	0x0
31:13	Reserved			

INT_STS

Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	tran_done_int	R/W	每成功发送一帧数据生成一次该中断	0x0
1	Reserved			
2	rdff_int	R/W	每成功接收一帧数据生成一次该中断	0x0
3	wrff_int	R/W	CPU 收到该中断后向 FIFO 写帧数据	0x0
4	rx_frame_int	R/W	CPU 收到该中断后从 FIFO 读帧数据	0x0
5	tx_frame_int	R/W	该中断标志着一次传输的完成	0x0
31:6	Reserved			

INT_EN

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	tran_done_int_en	R/W	使能中断 tran_done_int	0x0
1	Reserved			
2	rdff_int_en	R/W	使能中断 rdff_int	0x0
3	wrff_int_en	R/W	使能中断 wrff_int	0x0
4	rx_frame_int_en	R/W	使能中断 rx_frame_int	0x0
5	tx_frame_int_en	R/W	使能中断 tx_frame_int	0x0
31:6	Reserved			

14.3 SPI NAND Flash 控制器

14.3.1 概述

提供片外 SPI NAND Flash 数据存取。

14.3.2 特点

- 支持外接 1 片选。
- 支持 SPI NAND Flash x1/x2/x4 读写操作。
- 支时多种规格 SPI NAND Flash 器件。
  - 支持页 2KB、4KB 器件。
  - 支时 64Pages/Block、128Pages/Block 器件。
- 支持 SPI NAND 的 BOOT 功能。

14.3.3 功能描述

14.3.3.1 接口描述

SPI NAND Flash 控制器 可以支持三种SPI NAND接口类型，类型分别为Standard SPI、X2接口模式及X4接口模式。

Standard SPI接口模式:
Standard SPI接口模式具有1bit数据输入线和1bit数据输出线。图表 14-8为Standard SPI接口模式写操作时序图，图表 14-9为Standard SPI接口模式读操作时序图。

图表 14-8 Standard SPI 接口模式写操作时序

时序说明：

- ✓ command/address/dummy cycles 以单bit串行方式在DO 线上输出。
- ✓ Data 以单bit串行方式在DO线上输出。

图表 14-9 Standard SPI 接口模式读操作时序

时序说明：

- ✓ command/address/dummy cycles 以单bit串行方式在DO 线上输出。

Data 以单 bit 串行方式在 DI 线上输入。

X2接口模式:

Standard SPI接口模式共用2bit数据输入输出输出线。图表 14-10为X2接口模式操作时序图。

图表 14-10 SPI Nand x2 接口模式操作时序

时序说明：

- ✓ command/address/dummy cycles 以单Bit串行方式在DO 线上输出。
- ✓ Data以2 Bits方式在DO/DI线上输出（写）或输入（读）。

X4接口模式:
Standard SPI接口模式共用4bit数据输入输出输出线。图表 14-11为X4接口模式操作时序图。

图表 14-11 SPI Nand x4 接口模式操作时序

时序说明：

- ✓ command/address/dummy cycles 以单Bit串行方式在DO 线上输出。
- ✓ Data以4 Bits方式在DO/DI/WPN/HOLDN线上输出（写）或输入（读）。

14.3.3.2 SPI NAND FLASH 地址说明

在下发 SPI NAND Flash 的读写操作时，操作时根据具体的操作下发行列地址。

- 写操作：在 LOAD 操作时配置列地址，在 PROGRAM 操作时配置行地址。
- 读操作：在 PAGE READ TO CACHE 操作时配置行地址，在 READ 操作时配置列地址。
- 地址下发由控制器完成，软件需要根据操作指令配置 reg_trx_cmd_idx，及地址配置 reg_trx_cmd_cnt0、reg_trx_cmd_cnt1。

14.3.3.3 Boot 功能

由于 SPI NAND Flash 地址空间不连续且存在坏块的可能，因此 Boot 数据不能直接映射到 Flash 中。
支持自适应 Boot 功能，能够根据 Block0 的数据自动适配器件的信息。控制器要求物理 Block0 必须为好块，其他块为坏块则可以自动跳过。

14.3.3.4 寄存器操作

软件配置操作相关寄存器，如操作命令、地址等，最后配置 reg_trx_start 寄存器下发操作，控制器根据软件配置值，下发操作给器件。如果还需要向器件传输数据，那会使用内部的 DMA 操作来传输数据。

14.3.3.5 内置 DMA 操作方式

支持使用内置系统 DMA 模式进行读写操作以提高访问速度。通过此种方式，可以通过总线直接访问片内或片外存储空间。
步骤1. 配置 DMA 使用通道。
步骤2. 配置来源及目标地址。
步骤3. 配置传输格式及数据长度。

14.3.3.6 TIMEOUT 功能

软件设置最大 1 秒 TIMEOUT 机制，做为器件无正常响应时之保護。

14.3.4 工作流程

14.3.4.1 初始化流程

步骤1. （如果需要调整 Timing 参数）根据器件配置时序寄存器 reg_trx_sck_h 及 reg_trx_sck_l。
步骤2. 配置中断控制寄存器 reg_trx_done_int_en。

14.3.4.2 器件寄存器操作流程

步骤1. 配置传输数据长度 reg_trx_cmd_cont_size 及 reg_trx_data_size。
步骤2. 配置器件指令其相关内容 reg_trx_cmd_id、 reg_trx_cmd_cont0 及 reg_trx_cmd_cont1。
步骤3. 配置 reg_trx_start 寄存器下发操作。
步骤4. 检测到 reg_trx_done_int，表示操作完成。

14.3.4.3 擦除操作流程

对于 flash 操作，在编程操作前，都必须进行擦除，在擦除操作前还必须完成 WREN 操作。
步骤1. 配置传输数据长度 reg_trx_cmd_cont_size。
步骤2. 配置器件指令其相关内容 reg_trx_cmd_id、 reg_trx_cmd_cont0。
步骤3. 配置 reg_trx_start 寄存器下发操作。
步骤4. 检测到 reg_trx_done_int，表示操作完成。

14.3.4.4 内置 DMA 读操作流程

步骤1. 配置系统 DMA 寄存器，参照 9.5。
步骤2. 配置传输数据长度 reg_trx_cmd_cont_size 、reg_trx_data_size。
步骤3. 配置器件指令其相关内容 reg_trx_cmd_id、 reg_trx_cmd_cont0 及 reg_trx_cmd_cont1。
步骤4. 配置 reg_trx_start 寄存器下发操作。
步骤5. 检测到 reg_trx_done_int，表示器件内容完成读取且已写入缓冲区。

14.3.4.5 内置 DMA 写操作流程

步骤1. 配置系统 DMA 寄存器，参照 9.5。
步骤2. 配置传输数据长度 reg_trx_cmd_cont_size 、reg_trx_data_size。

步骤3. 配置器件指令其相关内容 reg_trx_cmd_id、 reg_trx_cmd_cont0 及 reg_trx_cmd_cont1。
步骤4. 配置 reg_trx_start 寄存器下发操作。
步骤5. 检测到 reg_trx_done_int，表示缓冲区内容已写入器件 cache 完成。

14.3.4.6 其它注意事项

- 部分 SPI NAND Flash 器件要求在使用之前或者异常复位后必须先进行器件的 RESET 操作；所以为器件相容性考量，开始使用或异常复位后的第一个传输指令需为 RESET。
- 对器件操作未完成前，不可改动相关寄存器配置，否则可能导致操作不正常。

14.3.5 数据结构(NAND Flash/SPI NAND Flash)

14.3.5.1 2KB page_size

对 2KB page_size 的配置，软件可用的冗余区常见大小为 64Byte。数据在 Buffer 和 Flash 中的结构下所示。冗余区大小与实际使用器件相关。

用户数据	Data(2048)	OOB(64)

14.3.5.2 4KB page_size

对 4KB page_size 的配置，软件可用的冗余区常见大小为 256Byte。数据在 Buffer 和 Flash 中的结构下所示。冗余区大小与实际使用器件相关。

用户数据	Data(4096)	OOB(256)

14.3.6 寄存器概览

Name	Address Offset	Description
reg_ctrl	0x000	transmission control
reg_timing_ctrl	0x004	timing control
reg_trx_size	0x008	number of content size
reg_int_en	0x010	interrupt enable
reg_int_clr	0x014	interrupt clear
reg_int_sts	0x01c	interrupt status
reg_cont0	0x030	content 0
reg_cont1	0x034	content 1
reg_cmplt_cnt	0x058	number of transferred bytes
reg_tx_data	0x060	tx data

Name	Address Offset	Description
reg_rx_data	0x064	rx data

14.3.7 寄存器描述

reg_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_trx_start	W1T	trigger spi transmission start	
31:1	Reserved			

reg_timing_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_trx_time_start	R/W	time for cs assert to 1st command bit unit: sck period	0x0
3:2	Reserved			
7:4	reg_trx_time_end	R/W	time for last data bit to cs de-assert unit: sck period	0x0
15:8	Reserved			
19:16	reg_trx_sck_h	R/W	time for sck high unit: source clock period	0x0
23:20	reg_trx_sck_l	R/W	time for sck low unit: source clock period	0x1
31:24	Reserved			

reg_trx_size

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_trx_cmd_cont_size	R/W	numbers of command content byte	0x0
3	Reserved			
5:4	reg_trx_dummy_size	R/W	numbers of dummy byte	0x0
15:6	Reserved			
28:16	reg_trx_data_size	R/W	numbers of data byte	0x0
31:29	Reserved			

reg_int_en

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_en	R/W	trx_done interrupt enable	0x1
31:1	Reserved			

reg_int_clr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_clr	W1T	trx_done interrupt clear	
31:1	Reserved			

reg_int_sts

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_trx_done_int	RO	trx_done interrpt	
31:1	Reserved			

reg_cont0

Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	reg_trx_cmd_idx	R/W	spi flash command value	0x0
31:8	reg_trx_cmd_cont0	R/W	spi flash address, or other contents	0x0

reg_cont1

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_trx_cmd_cont1	R/W	spi flash address, or other contents	0x0

reg_cmplt_cnt

Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_cmplt_cnt	RO	number of transferred bytes	
31:13	Reserved			

reg_tx_data

Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	reg_tx_data	RO	spi tx data	

reg_rx_data

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_rx_data	RO	spi rx data	

15 网络接口

15.1 Ethernet MAC

15.1.1 概述

芯片支持 1 个 Ethernet MAC, 实现网路数据的接收与发送。
这个 Ethernet MAC 搭配内建 10/100Mbps Fast Ethernet Transceiver 可工作在 10/100Mbps 全双工或半工双模式。支持通过 RMII 外挂 PHY。

15.1.2 功能描述

以太网模块有如下功能特点：

- Ethernet MAC0 搭配内建 10/100Mbps Fast Ethernet Transceiver 搭配内建 Ethernet PHY 支持 10/100Mbit/s 速率。
- 支持通过 RMII 外挂 PHY。
- 支持全双工或半双工工作模式。
- 支持对输入帧进行 CRC 校验。
- 支持对输出帧添加 CRC 校验。
- 支持短帧填充功能。
- 支持端口全双工模式下的内环回。
- 支持对接收和发送帧进行统计计数。
- 支持收发包缓存。
- 支持 COE（Checksum Offload Engine）校验和卸载引擎功能。

15.1.3 总体数据流

以太网交换接口的总体数据流如图表 15-1 所示。

图表 15-1 eMAC 总体数据流

15.1.4 单网口功能配置描述

15.1.4.1 以太网收发帧管理功能

CPU 先配置 Ethernet MAC 接收与发送 Descriptor List 缓存区与 Descriptor List 内容撰写例如,收发帧地址与封包种类大小参数的设定.

接收时，Ethernet MAC 接收收到的各种数据包，并根据 CPU 配置在接收 Descriptor List 讯息, 例如封包缓存资讯，包括封包缓存起始地址、封包缓存深度等，将收到的封包存放到 DDR 中。再通知 CPU 做后续处理动作。

发送时，Ethernet MAC 根据 CPU 配置在发送 Descriptor List 的封包缓存信息，例如包括封包缓存起始地址、封包长度以及其他的封包信息等，将存于 DDR 的封包搬运过来，自行组装成包，然后发送到网络接口上。再通知 CPU 封包已传送完毕。

15.1.5 以太网收包中断管理功能

15.1.5.1 中断产生

设置接收方向中断, 配置 Re_Int_Enable bit[6] = 1, CPU 查询 接收中断状态 Reg_Int_Status bit[6] 。

15.1.5.2 中断清除

CPU 查询 接收中断状态 Reg_Int_Status bit[6], 写 1 清除中断状态。

15.1.6 配置 PHY 芯片工作状态

Ethernet MAC 提供 MDIO 接口介面对 PHY 芯片的设定。MDIO 接口介面分为读操作和写操作，主要控制 MDIO 接口介面的器存器为 Reg_MdioAddr 和 Reg_MdioData.

- 读操作的配置步骤如下：

  配置 MDIO 控制寄存器以下设定:

  Reg_MdioAddr bit[15:11] 设定 PHY 芯片地址。请依据 PHY 芯片或板端规划.

  Reg_MdioAddr bit[10:6] 设定要读写的 PHY 内部寄存器地址。

  Reg_MdioAddr bit[1] 写入 0(读的动作命令)。

最后 设定 Reg_MdioAddr bit[0] = 1, 来启动 读的动作.
MDIO 接口介面会将读回的数据接收到 Reg_MdioData bit[15:0], 并将 Reg_Mdi0Addr bit[0] 改变为 0.

- 写操作的配置步骤如下：

配置 MDIO 控制寄存器以下设定:
Reg_MdioAddr bit[15:11] 设定 PHY 芯片地址。请依据 PHY 芯片或板端规划.
Reg_MdioAddr bit[10:6] 设定要读写的 PHY 内部寄存器地址。
Reg_MdioAddr bit[1] 写入 1(写的动作命令)。
最后 设定 Reg_MdioAddr bit[0] = 1, 来启动 写的动作.
MDIO 接口介面会在写的动作完成后, 将 Reg_Mdi0Addr bit[0] 改变为 0.

15.1.7 工作模式切换

Ethernet MAC 的工作模式：Ethernet MAC0 是支援 内建 EPHY 功能使用. 采用的工作模式是 RMII(10/100M).

速度与模式切换寄存器设定如下步骤 :
配置 ETH0 Reg_MacConfig bit[14] = (100M:1, 10M:0)；

注意事项 : 芯片正常工作时不可进行此项配置，建议在初始化时进行配置。

15.1.8 典型应用

15.1.9 寄存器偏移地址说明

Ethernet MAC 0 寄存器偏移地址空间:

- ETH0_MAC : 0x0451_000~0x0451_FFFF

15.1.10 GMAC 寄存器概览

表格 15-1 GMAC 寄存器概览

Name	Address Offset	Description
Reg_MacConfig	0x000	本机 MAC 工作状态寄存器
Reg_MdioAddr	0x010	MDIO 操作寄存器.
Reg_MdioData	0x014	MDIO 数据读写寄存器

Name	Address Offset	Description
Reg_MacAddr0_High	0x040	本机 MAC 地址寄存器#0 高 16bit
Reg_MacAddr0_Low	*0x044*	*本机 MAC 地址寄存器#0 低 32bit*
Reg_MacAddr1_High	0x048	本机 MAC 地址寄存器#0 高 16bit
Reg_MacAddr1_Low	0x04c	本机 MAC 地址寄存器#0 低 32bit
Reg_Tx_Packet_Num_Good_Bad	0x118	发送成功的好包与错包帧数统计寄存器
Reg_Tx_Bcast_Packets_Good	0x11c	发送成功的好包的广播帧数统计寄存器
Reg_Tx_Mcast_Packets_Good	0x120	发送成功的好包的多播帧数统计寄存器
Reg_Tx_Ucast_Packets_Good_Bad	0x13c	发送成功的好包与错包的单播帧数统计寄存器
Reg_Tx_Mcast_Packets_Good_Bad	0x140	发送成功的好包与错包的多播帧数统计寄存器
Reg_Tx_Bcast_Packets_Good_Bad	0x144	发送成功的好包与错包的广播帧数统计寄存器
Reg_Rx_Packets_Num_Good_Bad	0x180	接收成功的好包与错包帧数统计寄存器
Reg_Rx_Bcast_Packets_Good	0x18c	接收成功的好包的广播帧数统计寄存器
Reg_Rx_Mcast_Packets_Good	0x190	接收成功的好包的多播帧数统计寄存器
Reg_Rx_CRC_Error_Packets	0x194	接收 CRC 检测错误的帧数统计寄存器
Reg_Rx_Ucast_Packets_Good	0x1c4	接收成功的好包的单播帧数统计寄存器
Reg_Int_Enable	0x101c	中断使能寄存器
Reg_Int_Status	0x1014	中断状态寄存器

15.1.11 GMAC 寄存器描述

Reg_MacConfig

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	RX_EN	R/W	MAC 接收使能寄存器	0x0
3	TX_EN	R/W	MAC 发送使能寄存器	0x0
6:4	Reserved			
7	APCS_EN	R/W	自动 Pad 或 CRC 剥离控制使能寄存器	0x0
9:8	Reserved			
10	CHKS_EN	R/W	IP checksum 辅助始能寄存器	0x0
11	DUPLEX_MODE	R/W	全/半双工模式寄存器, 1 (使能全双工模式)	0x0
12	LPBK_MODE	R/W	环回模式控制寄存器	0x0
13	Reserved			
14	SPEED_MODE	R/W	速度模式寄存器 1'b1:100M, 1'b0:10M	0x0

Bits	Name	Access	Description	Reset
16:15	Reserved			
19:17	IPG_VAL	R/W	发送封包间距控制寄存器.	0x0
22:20	Reserved			
23	WD_DISABLE	R/W	看门狗不使能寄存器	0x0
24	Reserved			
25	CRC_STRIP_EN	R/W	CRC 剥离 Type 型态封包控制使能寄存器	0x0
31:26	Reserved			

Reg_MdioAddr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	GO	R/W	MDIO 操作完成指示. 1:启动操作 0:操作完成.	0x0
1	CMD	R/W	MDIO 操作命令类型 写(1'b1), 读(1'b0)	0x0
5:2	Reserved			
10:6	RegAddr	R/W	外部 PHY 地址配置寄存器	0x0
15:11	PhyAddr	R/W	PHY 器件内部寄存器位址寄存器	0x0
31:16	Reserved			

Reg_MdioData

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	MdioData	R/W	MDIO 写或从 PHY 回读数据寄存器	0x0
31:16	Reserved			

Reg_MacAddr0_High

Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	Addr0_High	R/W	MAC 地址寄存器#0 bit[47:32]	0x0
30:16	Reserved			
31	Addr0_EN	R/W	Addr0 Enable	0x0

Reg_MacAddr0_Low

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	Addr0_Low	R/W	MAC 地址寄存器#0 bit[31:0]	0x0

Reg_MacAddr1_High

Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	Addr1_High	R/W	MAC 地址寄存器#1 bit[47:32]	0x0
23:16	Reserved			
29:24	Addr1_MASK	R/W	Addr1 Mask Byte	0x0
30	Addr1_TYPE	R/W	1'b1: compare SA 1'b0: compare DA	0x0
31	Addr1_EN	R/W	Addr1 Enable	0x0

Reg_MacAddr1_Low

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	Addr1_Low	R/W	MAC 地址寄存器#1 bit[31:0]	0x0

Reg_Tx_Packet_Num_Good_Bad

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	TxPktNumGB	RO	发送成功的好包与错包帧数统计寄存器	

Reg_Tx_Bcast_Packets_Good

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	TxBcG	RO	发送成功的好包的广播帧数统计寄存器	

Reg_Tx_Mcast_Packets_Good

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	TxMcG	RO	发送成功的好包的多播帧数统计寄存器	

Reg_Tx_Ucast_Packets_Good_Bad

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	TxUcGB	RO	发送成功的好包与错包的单播帧数统计寄存器	

Reg_Tx_Mcast_Packets_Good_Bad

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	TxMcGB	RO	发送成功的好包与错包的多播帧数统计寄存器	

Reg_Tx_Bcast_Packets_Good_Bad

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	TxBcGB	RO	发送成功的好包与错包的广播帧数统计寄存器	

Reg_Rx_Packets_Num_Good_Bad

Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	RxPktGB	RO	接收成功的好包与错包帧数统计寄存器	

Reg_Rx_Bcast_Packets_Good

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	RxBcG	RO	接收成功的好包的广播帧数统计寄存器	

Reg_Rx_Mcast_Packets_Good

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	RxMcG	RO	接收成功的好包的多播帧数统计寄存器	

Reg_Rx_CRC_Error_Packets

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	RxCrcERR	RO	接收 CRC 检测错误的帧数统计寄存器	

Reg_Rx_Ucast_Packets_Good

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	RxUcG	RO	接收成功的好包的单播帧数统计寄存器	

Reg_Int_Enable

Offset Address: 0x101c

Bits	Name	Access	Description	Reset
0	TxInt_EN0	R/W	发送中断使能寄存器	0x0
5:1	Reserved			
6	RxInt_EN0	R/W	接收中断使能寄存器	0x0
31:7	Reserved			

Reg_Int_Status

Offset Address: 0x1014

Bits	Name	Access	Description	Reset
0	TxInt_ST0	RO	发送中断状态寄存器	
5:1	Reserved			
6	RxInt_ST0	RO	接收中断状态寄存器	
31:7	Reserved			

15.2 Ethernet PHY

15.2.1 概述

蕊片提供一组内建 Ethernet 10/100 Base-TX compliant PHY 接口。

15.2.2 功能描述

- 支持 IEEE 802.3 10/100 Base-TX compliant.
- 支持传收送速度与全半双工工作模式自动协商功能.
- 支持以太网路线交叉线或直连线自动移动翻转功能.

- 支持以太网路 WOL(Wake on Lan) 功能.

15.2.3 功能配置图描述

图表 15-2 内建 10/100 Ethernet PHY 框图

10/100Mbps 传送及收发功能在标准 category5(CAT5) 双绞线上, 传送接收讯号经由 transformer 连接于 RJ45 标准接口.

16 视频接口

16.1 VI

16.2 概述

视频输入单元 VI (Video Input) 为芯片接收摄像头视频数据模块，可以支持通过 MIPI Rx (包含 MIPI、Sub-LVDS、HiSPi) 接口或 BT.656、BT.601、BT.1120 接口和 DC (Digital Camera) 接收视频数据，处理后送到下一级的视频处理模块 (ISP)。VI 的功能框图如图表 16-1 所示。

VI 分为两个物理子模块，由移动行业处理器接口接收端模块 MIPI Rx 和视频输入处理模块 VI Proc 组成。MIPI Rx 模块接收处理不同的视频数据，而 VI Proc 模块则会将不同格式的视频信号统整为 ISP 模块所需的单一视频信号送出。

图表 16-1 VI 功能框图

16.2.1 特点

- MIPI Rx 最多同时支持两路 sensor 数据输入。
  - 单一 sensor 最大支持 5M(2688x1944, 2880x1620) @60fps HDR 输入 or @30fps 线性输入。
  - 双路 sensor 最大支持 FHD(1920x1080) @60fps HDR or linear 输入

• MIPI Rx 输入支持最大数据位宽为 12bit。
• 支持 BT.656、BT.601、BT.1120(仅支持逐行模式)
• 支持 BT.656/BT.1120 多通道融合输入(1,2,4 通道，仅支持逐行模式)

DC 接口
• 支持 MIPI CSI-2、Sub-LVDS、HiSPi 接口。
• 支持 MIPI 接口输入 YUV422 格式。
• 支持 两帧高动态范围(HDR)影像输入。

16.2.2 模式功能描述

16.2.2.1 典型应用

VI 可以支持多种时序输入以及不同接口，并针对不同编码方式做视频输入采集，系统可以利用暂存器配置不同的功能模式，使之适应不同的视频接口。
VI 模块最多可支持三路输入，典型输入如下

• 1 路 5M(2688x1944, 2880x1620) @60fps HDR 输入或是线性 @30fps 输入
• 2 路 FHD(1920x1080) @60fps HDR or 线性输入 + 1 路 BT 输入 (BT.656, BT.601 or BT.1120)
• 1 路 5M(2688x1944, 2880x1620) @60fps HDR or @30fps 线性输入 + 1 路 BT 输入

16.2.2.2 功能原理

16.2.2.3 BT.1120 接口时序

VI 支持 Y/C 分开输入的 BT.1120 接口时序，传输视频讯号之前会先传送同步码，同步码为数据流中使用特殊字节 SAV 和 EAV 分别表示有效行数据的开始和结束。同步码后使用 16bit 来传输视频讯号，其中 8bit 用来传输亮度，另外 8bit 用来传输色度，如图表 16-2 和 图表 16-3 所示。

图表 16-2 BT.1120 水平接口时序

图表 16-3 BT.1120 垂直接口时序

同步码格式为 4 个字节(byte)的组合，依序为 {0xFF, 0x00, 0x00, EAV/SAV}, 第四个字节详解如下， SG2000 只支持渐进式逐行扫描格式(Progressive)，故 bit 6 值为 0 。

	SAV/EAV bit			Protection Bit				
7 (Fixed)	**6 (F)**	**5 (V)**	**4 (H)**	**3 (P3)**	**2 (P2)**	**1 (P1)**	**0 (P0)**	注解
1	0	0	0	0	0	0	0	SAV_VLD
1	0	0	1	1	1	0	1	EAV_VLD
1	0	1	0	1	0	1	1	SAV_BLK
1	0	1	1	0	1	1	0	EAV_BLK

SAV_VLD : 有效行区，行同步信号结束，有效画素开始。
EVA_VLD : 有效行区，行同步信号开始，有效画素结束。
SAV_BLK : 消隐行区，行同步信号结束。
EAV_BLK : 消隐行区，行同步信号开始。

16.2.2.4 BT.656 接口时序

VI 同样支持 Y/C 合并输入的 BT.656 接口时序，传输时同样利用同步码 SAV 和 EAV 表示有效行数据的开始和结束，但只利用 8bit 传输视频讯号，并利用分时的方式传输亮度

和色度，如图表 16-4 所示。

图表 16-4 BT.656 水平接口时序

BT.656 和 BT.1120 差别只在影像传输用 16 bit (BT.1120)和 8bit(BT.656), 其余垂直时序和同步码格式都一致。

16.2.2.5 BT.601 接口时序

除了利用同步码 BT.1120 和 BT.656，VI 支持利用多种不同的同步信号来传输的 BT.601 接口时序。实际的视频资料可利用寄存器设定为 Y/C 分开输入的 16bit 模式或是 Y/C 合并分时输入的 8bit 模式，而同步方式则可利用寄存器选择 vhs，vde，或是 vsde 三种模式，详细时序如下图。

图表 16-5 BT.601 vhs 同步模式

vhs 模式输入同步信号为帧同步信号(vs), 行同步信号(hs)，系统必须设定帧后消隐行数(vs_back_porch)，影像高度(img_ht), 行后消隐像素数(hs_back_porch)和影像宽度(img_wd)。

图表 16-6 BT.601 vde 同步模式

vde 模式同步信号为行有效信号(vde)和 行有效信号(hde)。在此模式下，系统不用设置时序相序相关的参数。VI 模组会依据 hde/vde 信号收资料,以依 vde 信号做帧更新。

图表 16-7 BT.601 vsde 同步模式

vsde 模式同步信号为帧同步信号(vs)和 有效像素旗标(de)。在此模式下，系统不用设置时序相序相关的参数。VI 模组会依据 de 信号收资料,以依 vs 信号做帧更新。

图表 16-8 BT.601 Y/C 分开 16bit 模式

图表 16-9 BT.601 Y/C 合并 8bit 模式

16.2.2.6 数字摄像头 (DC) 接口时序

VI 支持传送 RAW 格式仿 BT 传输的数字摄像头 (DC) 接口时序，在 DC 接口可支持 8bit，10bit，12bit，和 16bit 四种不同模式，也可利用寄存器设定利用接收同步码或是类似 BT.601 的三种不同同步模式来接收视频信号。

图表 16-10 DC 同步码模式

图表 16-11 DC 同步信号模式-vhs mode

图表 16-12DC 同步信号模式-VDE mode

图表 16-13DC 同步信号模式-VSDE mode

16.2.3 图像储存模式

储存在 DRAM 的图像区分为 Bayer 12bit 与 YCbCr 8 bit 两种格式。其中 Y/Cb/Cr 为分开存储在三个不同的 DRAM 位置。两种(12bit/8bit)格式的图像在 DRAM 的排列方式如下图所示。

图表 16-14 Bayer 12 bit 图像储存方式

Bayer 12bit		Pixel 0	Pixel 1	Pixel 2	Pixel 3				
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
DRAM Address	Addr N+0	4	5	6	7	8	9	10	11
	Addr N+1	4	5	6	7	8	9	10	11
	Addr N+2	0	1	2	3	0	1	2	3
	Addr N+3	4	5	6	7	8	9	10	11
	Addr N+4	4	5	6	7	8	9	10	11
	Addr N+5	0	1	2	3	0	1	2	3

图表 16-15 YCbCr 8bit 图像储存方式

YCbCr 8bit		Pixel 0	Pixel 1	Pixel 2	Pixel 3				
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
DRAM Address	Addr N+0	0	1	2	3	4	5	6	7
	Addr N+1	0	1	2	3	4	5	6	7
	Addr N+2	0	1	2	3	4	5	6	7
	Addr N+3	0	1	2	3	4	5	6	7

16.2.4 VI 寄存器概览

芯片中有两套相同的 VI 模块，内部寄存器偏移地址都相同，而基址分别为 0x0A0C2000 和 0x0A0C4000。另外有一套只支持 BT 接口的 VI 模块，基址为 0x0A0C6000。

图表 16-16 VI 寄存器概览

Name	Address Offset	Description
REG_00	0x000	MODE
REG_10	0x010	TTL_MODE_0
REG_14	0x014	TTL_MODE_1
REG_18	0x018	TTL_MODE_2
REG_1C	0x01c	TTL_MODE_3

Name	Address Offset	Description
REG_20	0x020	TTL_MODE_4
REG_24	0x024	TTL_MODE_5
REG_28	0x028	TTL_MODE_6
REG_30	0x030	TTL_MODE_7
REG_40	0x040	HDR_MODE_0
REG_44	0x044	HDR_MODE_1
REG_48	0x048	HDR_MODE_2
REG_50	0x050	BLC_MODE
REG_54	0x054	BLC_MODE_0
REG_58	0x058	BLC_MODE_1
REG_60	0x060	VI_PINMUX_0
REG_64	0x064	VI_PINMUX_1
REG_68	0x068	VI_PINMUX_2
REG_6C	0x06c	VI_PINMUX_3
REG_70	0x070	VI_PINMUX_4
REG_74	0x074	VI_PINMUX_5
REG_80	0x080	BT_PATH_0
REG_88	0x088	BT_PATH_2
REG_8C	0x08c	BT_PATH_3
REG_90	0x090	BT_PATH_4
REG_94	0x094	BT_PATH_5
REG_98	0x098	BT_PATH_6
REG_9C	0x09c	BT_PATH_7
REG_A0	0x0a0	BT_PATH_8
REG_A4	0x0a4	BT_PATH_A
REG_B0	0x0b0	CROP_0
REG_B4	0x0b4	CROP_1
REG_D0	0x0d0	MODE_CTRL
REG_D4	0x0d4	SYNC_CODE_0
REG_D8	0x0d8	SYNC_CODE_1
REG_DC	0x0dc	SYNC_CODE_2
REG_E0	0x0e0	SYNC_CODE_3
REG_E4	0x0e4	SYNC_CODE_4
REG_E8	0x0e8	SYNC_CODE_5
REG_EC	0x0ec	SYNC_CODE_6
REG_F0	0x0f0	SYNC_CODE_7
REG_F4	0x0f4	SYNC_CODE_8
REG_F8	0x0f8	SYNC_CODE_9
REG_FC	0x0fc	VS_GEN
REG_100	0x100	SYNC_CODE_A
REG_104	0x104	SYNC_CODE_B
REG_108	0x108	HDR_PATTEN_2
REG_110	0x110	HISPI_MODE_CTRL_0
REG_114	0x114	HISPI_MODE_CTRL_1
REG_118	0x118	HISPI_MODE_CTRL_2
REG_11C	0x11c	HISPI_MODE_CTRL_3
REG_120	0x120	HISPI_MODE_CTRL_4
REG_124	0x124	HISPI_MODE_CTRL_5

16.2.5 VI 寄存器描述

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_sensor_mac_mode	R/W	Sensor mode 3'b000: Disable 3'b001: CSI 3'b010: Sub-LVDS 3'b011: TTL	0x0
3	reg_bt_demux_enable	R/W	BT Demux enable	0x0
4	reg_csi_ctrl_enable	R/W	CSI controller enable	0x0
5	reg_csi_vs_inv	R/W	CSI VS inverse	0x1
6	reg_csi_hs_inv	R/W	CSI HS inverse	0x1
7	Reserved			
8	reg_sublvds_ctrl_enable	R/W	Sub-LVDS controller enable	0x0
9	reg_sublvds_vs_inv	R/W	Sub-LVDS VS inverse	0x1
10	reg_sublvds_hs_inv	R/W	Sub-LVDS HS inverse	0x1
11	reg_sublvds_hdr_inv	R/W	Sub-LVDS HDR inverse	0x1
31:12	Reserved			

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_ttl_ip_en	R/W	TTL enable	0x0
2:1	reg_ttl_sensor_bit	R/W	TTL bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit 2'b11: 16-bit	0x0
3	Reserved			
5:4	reg_ttl_bt_fmt_out	R/W	TTL BT output format 2'b00: {Cb,Y},{Cr,Y} 2'b01: {Cr,Y},{Cb,Y} 2'b10: {Y,Cb},{Y,Cr} 2'b11: {Y,Cr},{Y,Cb}	0x2
7:6	Reserved			
11:8	reg_ttl_fmt_in	R/W	TTL input format 4'b0000: bt_2x with sync pattern, 9-bit BT656 4'b0001: bt_1x with sync pattern, 17-bit BT1120 4'b0010: bt_2x without sync pattern, 11-bit BT601 (vhs mode) 4'b0011: bt_1x without sync pattern, 19-bit BT601 (vhs mode) 4'b0100: bt_2x without sync pattern, 11-bit BT601 (vde mode) 4'b0101: bt_1x without sync pattern, 19-bit BT601 (vde mode) 4'b0110: bt_2x without sync pattern, 11-bit BT601 (vsde mode) 4'b0111: bt_1x without sync pattern,	0x0

Bits	Name	Access	Description	Reset
			19-bit BT601 (vsde mode) 4'b100x: sensor with sync pattern 4'b101x: sensor without sync pattern, use vs + hs (vhs mode) 4'b110x: sensor without sync pattern, use vde + hde (vde mode) 4'b111x: sensor without sync pattern, use vs + hde (vsde mode)	
13:12	reg_ttl_bt_data_seq	R/W	TTL bt data sequence 2'b00: Cb0-Y0-Cr0-Y1 2'b01: Cr0-Y0-Cb0-Y1 2'b10: Y0-Cb0-Y1-Cr0 2'b11: Y0-Cr0-Y1-Cb0	0x0
14	reg_ttl_vs_inv	R/W	TTL vs inverse	0x0
15	reg_ttl_hs_inv	R/W	TTL hs inverse	0x0
31:16	Reserved			

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_ttl_vs_bp	R/W	TTL vsync back porch setting	0x0
15:12	Reserved			
27:16	reg_ttl_hs_bp	R/W	TTL hsync back porch setting	0x0
31:28	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_ttl_img_wd	R/W	TTL image width setting	0x0
15:12	Reserved			
27:16	reg_ttl_img_ht	R/W	TTL image height setting	0x0
31:28	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_0	R/W	TTL sync code 0	0x0
31:16	reg_ttl_sync_1	R/W	TTL sync code 1	0x0

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_2	R/W	TTL sync code 2	0x0
31:16	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sav_vld	R/W	TTL valid line SAV	0x0
31:16	reg_ttl_sav_blk	R/W	TTL blanking line SAV	0x0

REG_28

Offset Address: 0x028

Bits	Name	Access	Description	Reset
15:0	reg_ttl_eav_vld	R/W	TTL valid line EAV	0x0
31:16	reg_ttl_eav_blk	R/W	TTL blanking line EAV	0x0

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_vi_sel	R/W	VI input mode select 3'h1: RAW 3'h2: BT601 3'h3: BT656 3'h4: BT1120 else: reserved	0x0
3	reg_vi_from	R/W	VI input from VI0 or VI1 1'b0: from VI0 1'b1: from VI1	0x0
4	reg_vi_clk_inv	R/W	VI clock inverse	0x0
5	reg_vi_v_sel_vs	R/W	1'b1: vs_in signal as vs 1'b0: vs_in signal as vde	0x1
6	reg_vi_vs_dbg	R/W	vsync source select	0x0
7	Reserved			
8	reg_pad_vi0_clk_inv	R/W	vi0 clk inverse	0x0
9	reg_pad_vi1_clk_inv	R/W	vi1 clk inverse	0x0
10	reg_pad_vi2_clk_inv	R/W	vi2 clk inverse	0x0
31:11	Reserved			

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_hdr_en	R/W	Sensor mac hdr manual mode enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
1	reg_sensor_mac_hdr_vsinv	R/W	Sensor mac vsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
2	reg_sensor_mac_hdr_hsinv	R/W	Sensor mac hsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3	reg_sensor_mac_hdr_deinv	R/W	Sensor mac de output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
4	reg_sensor_mac_hdr_hdr0inv	R/W	Sensor mac hdr[0] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
5	reg_sensor_mac_hdr_hdr1inv	R/W	Sensor mac hdr[1] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
6	reg_sensor_mac_hdr_blcinv	R/W	Sensor mac blc output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

Bits	Name	Access	Description	Reset
7	Reserved			
8	reg_sensor_mac_hdr_mode	R/W	Sensor mac hdr mode 1'b1 stands for HiSPi S-SP HDR mode, remove HDR blanking line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:9	Reserved			

REG_44

Offset Address: 0x044

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_hdr_shift	R/W	Sensor mac hdr long exposure shift (long exposure lines before 1st short exposure line) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:13	Reserved			
28:16	reg_sensor_mac_hdr_vsize	R/W	Sensor mac hdr vsize Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:29	Reserved			

REG_48

Offset Address: 0x048

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_info_line_num	R/W	Info line number Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:13	Reserved			
16	reg_sensor_mac_rm_info_line	R/W	Remove info line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:17	Reserved			

REG_50

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_blc0_en	R/W	BLC0 mode enable	0x0
1	reg_sensor_mac_blc1_en	R/W	BLC1 mode enable	0x0
31:2	Reserved			

REG_54

Offset Address: 0x054

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc0_start	R/W	BLC0 start line number	0x0
15:13	Reserved			
28:16	reg_sensor_mac_blc0_size	R/W	BLC0 line size	0x4
31:29	Reserved			

REG_58

Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc1_start	R/W	BLC1 start line number	0x0
15:13	Reserved			
28:16	reg_sensor_mac_blc1_size	R/W	BLC1 line size	0x4
31:29	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
5:0	reg_vi_vs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_hs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_vde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_hde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_64

Offset Address: 0x064

Bits	Name	Access	Description	Reset
5:0	reg_vi_d0_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d1_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d2_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d3_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_68

Offset Address: 0x068

Bits	Name	Access	Description	Reset
5:0	reg_vi_d4_sel	R/W	vi pin select [5]: from VI1 or VI0	0x0

Bits	Name	Access	Description	Reset
			[4:0]: from which VI pad count	
7:6	Reserved			
13:8	reg_vi_d5_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d6_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d7_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_6C

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
5:0	reg_vi_d8_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d9_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d10_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d11_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
5:0	reg_vi_d12_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d13_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d14_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d15_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0

Bits	Name	Access	Description	Reset
31:30	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
2:0	reg_vi_bt_d0_sel	R/W	vi bt pin select from which VI2 pad count	0x0
3	Reserved			
6:4	reg_vi_bt_d1_sel	R/W	vi bt pin select from which VI2 pad count	0x1
7	Reserved			
10:8	reg_vi_bt_d2_sel	R/W	vi bt pin select from which VI2 pad count	0x2
11	Reserved			
14:12	reg_vi_bt_d3_sel	R/W	vi bt pin select from which VI2 pad count	0x3
15	Reserved			
18:16	reg_vi_bt_d4_sel	R/W	vi bt pin select from which VI2 pad count	0x4
19	Reserved			
22:20	reg_vi_bt_d5_sel	R/W	vi bt pin select from which VI2 pad count	0x5
23	Reserved			
26:24	reg_vi_bt_d6_sel	R/W	vi bt pin select from which VI2 pad count	0x6
27	Reserved			
30:28	reg_vi_bt_d7_sel	R/W	vi bt pin select from which VI2 pad count	0x7
31	Reserved			

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_bt_clr_sync_lost_1t	R/W	Clear sync_lost signal	0x0
1	reg_bt_ip_en	R/W	BT path enable	0x0
2	reg_bt_ddr_mode	R/W	BT DDR mode	0x0
3	reg_bt_hs_gate_by_vde	R/W	HS gating by VDE	0x0
4	reg_bt_vs_inv	R/W	vsync inverse	0x0
5	reg_bt_hs_inv	R/W	hsync inverse	0x0
6	reg_bt_vs_as_vde	R/W	input vsync as vde	0x0
7	reg_bt_hs_as_hde	R/W	input hsync as hde	0x0
14:8	reg_bt_sw_en_clk	R/W	Clock gating software enable [0]: delay control clock enable [1]: timing demultiplexer clock enable [2]: timing gen clock enable [3]: rx decode 0 clock enable [4]: rx decode 1 clock enable [5]: rx decode 2 clock enable [6]: rx decode 3 clock enable	0x0
15	Reserved			
17:16	reg_bt_demux_ch	R/W	Demux setting 2'h0: No demux 2'h1: Demux 2	0x0

Bits	Name	Access	Description	Reset
			2'h2: Demux 3 2'h3: Demux 4	
19:18	Reserved			
22:20	reg_bt_fmt_sel	R/W	3'b000 : bt_2x with sync pattern, 9-bit BT656 (clock + 8-bit data) 3'b001 : bt_1x with sync pattern, 17-bit BT1120 (clock + 16-bit data) 3'b010 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hs) (vhs_mode) 3'b011 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hs) (vhs_mode) 3'b100 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vde + hde) (vde_mode) 3'b101 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vde + hde) (vde_mode) 3'b110 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hde) (vsde_mode) 3'b111 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hde) (vsde_mode)	0x0
31:23	Reserved			

REG_88

Offset Address: 0x088

Bits	Name	Access	Description	Reset
11:0	reg_bt_img_wd_m1	R/W	BT image width	0x0
15:12	Reserved			
27:16	reg_bt_img_ht_m1	R/W	BT image height	0x0
31:28	Reserved			

REG_8C

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
11:0	reg_bt_vs_bp_m1	R/W	BT vsync back porch	0x0
15:12	Reserved			
27:16	reg_bt_hs_bp_m1	R/W	BT hsync back porch	0x0
31:28	Reserved			

REG_90

Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_bt_vs_fp_m1	R/W	BT vsync front porch	0x0
15:8	reg_bt_hs_fp_m1	R/W	BT hsync front porch	0x0
31:16	Reserved			

REG_94

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_bt_sync_0	R/W	BT sync code byte 0	0x0

Bits	Name	Access	Description	Reset
15:8	reg_bt_sync_1	R/W	BT sync code byte 1	0x0
23:16	reg_bt_sync_2	R/W	BT sync code byte 2	0x0
31:24	Reserved			

REG_98

Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_0	R/W	BT valid SAV sync code for demux 0	0x0
15:8	reg_bt_sav_blk_0	R/W	BT blank SAV sync code for demux 0	0x0
23:16	reg_bt_eav_vld_0	R/W	BT valid EAV sync code for demux 0	0x0
31:24	reg_bt_eav_blk_0	R/W	BT blank EAV sync code for demux 0	0x0

REG_9C

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_1	R/W	BT valid SAV sync code for demux 1	0x0
15:8	reg_bt_sav_blk_1	R/W	BT blank SAV sync code for demux 1	0x0
23:16	reg_bt_eav_vld_1	R/W	BT valid EAV sync code for demux 1	0x0
31:24	reg_bt_eav_blk_1	R/W	BT blank EAV sync code for demux 1	0x0

REG_A0

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_2	R/W	BT valid SAV sync code for demux 2	0x0
15:8	reg_bt_sav_blk_2	R/W	BT blank SAV sync code for demux 2	0x0
23:16	reg_bt_eav_vld_2	R/W	BT valid EAV sync code for demux 2	0x0
31:24	reg_bt_eav_blk_2	R/W	BT blank EAV sync code for demux 2	0x0

REG_A4

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_3	R/W	BT valid SAV sync code for demux 3	0x0
15:8	reg_bt_sav_blk_3	R/W	BT blank SAV sync code for demux 3	0x0
23:16	reg_bt_eav_vld_3	R/W	BT valid EAV sync code for demux 3	0x0
31:24	reg_bt_eav_blk_3	R/W	BT blank EAV sync code for demux 3	0x0

REG_B0

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_x	R/W	Pixels before reg_sensor_mac_crop_start_x will be cropped in each line if enable reg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_x	R/W	Pixels after reg_sensor_mac_crop_end_x will be cropped in each line if enable reg_sensor_mac_crop_en.	0xFFF
30:29	Reserved			
31	reg_sensor_mac_crop_en	R/W	enable crop function	0x0

REG_B4

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_y	R/W	Lines before reg_sensor_mac_crop_start_y will be cropped in each frame if enable reg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_y	R/W	Lines after reg_sensor_mac_crop_end_y will be cropped in each frame if enable reg_sensor_mac_crop_en.	0xFFF
31:29	Reserved			

REG_D0

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	reg_ttl_as_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
7:1	Reserved			
9:8	reg_ttl_as_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_ttl_as_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_ttl_as_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_ttl_as_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

REG_D4

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_ttl_as_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

REG_D8

Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

REG_DC

Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

REG_E0

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

REG_E4

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

REG_E8

Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

REG_EC

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802
31:28	Reserved			

REG_F0

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01

Bits	Name	Access	Description	Reset
31:28	Reserved			

REG_F4

Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

REG_F8

Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

REG_FC

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_ttl_as_slvds_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

REG_100

Offset Address: 0x100

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3
31:28	Reserved			

REG_104

Offset Address: 0x104

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			

Bits	Name	Access	Description	Reset
27:16	reg_ttl_as_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

REG_108

Offset Address: 0x108

Bits	Name	Access	Description	Reset
13:0	reg_ttl_as_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_ttl_as_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

REG_110

Offset Address: 0x110

Bits	Name	Access	Description	Reset
0	reg_ttl_as_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_ttl_as_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_ttl_as_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

REG_114

Offset Address: 0x114

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00
31:28	Reserved			

REG_118

Offset Address: 0x118

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

REG_11C

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

REG_120

Offset Address: 0x120

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

REG_124

Offset Address: 0x124

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

16.3 VDP（Video Display Processor）

16.3.1 概述

VDP 模块可迭加图形数据在视频数据之上，再经由不同的显示通输出。视频数据可从内存读取或是接收 VPSS 模块输出的视频 。图形数据则必须由内存读取。

16.3.2 架构描述

VDP 的整体架构如图

图表 16-17 VDP 功能框图

SG2000 仅支持一路 BT.656/BT.601/BT.1120/MCU-I80/MCU-M68/LVDS/MIPI DSI/serial TTL

输出

- 总线数据读取，数据处理，包含视频层 V, 图形层 G。
- 视频层可接收不同的影像格式 (YUV422,YUV420,YUV444,RBG packet,NV12/NV21,YUYV-packet)，并将之转换成显示通道所需的时序及格式。
- 图形层支持 ARGB8888,ARGB4444, ARGB1555, 8-bit/4-bit LUT,font base 等格式的 OSD。
- 支持 OSD 厌缩格式
- Blend : 单层图形层 G 视频图形迭加器。

VDP 的特点如下 :

- MIPI Tx 输出接口 : 最大支持 1080P@60fps RGB24-bit 输出
- LVDS 输出 : 最大支持 720P@60fps RGB24-bit 输出
- 数字输出接口 : (BT format only support progressive mode)
    - 支持 ITU-R BT.1120 输出
    - 支持 ITU-R BT.656 输出
    - 支持 ITU-R BT.601 输出
    - MCU I80/M68 8-bit 输出
    - 支持 serial TTL 平行输出
- 视频层 : V 层
- 图形层 : G 层
- 迭加特性 : V/G 256 阶线性迭加, 并支持 Gamma 校正。
- VDP 包含一个显示通道，一个时序中断以及一个低带宽中断。

16.3.3 工作方式

16.3.3.1 时钟配置

VDP 搭配专属的时钟生成器，架构如下

图表 16-18 VDP 时钟生成器

- clk_syn : 1GHz or 1.2Ghz

- FREQ_SYN : 6-bit 整数，26-bit 小数的频率生成器
- PLL : 产生 VDP 时钟及 MIPI Tx/ LVDS 串行所需时钟

16.3.3.2 复位

VDP 复位包含一个硬件复位及一个软件复位。
在进行 AXI 总线复位之前 :
- 将所有 AXI 存取关闭。
- 确认 AXI 存取动作已结束，再配置总线复位。

16.3.3.3 输出接口

VDP 支持下列三种接口输出
- MIPI DSI
- LVDS
- ITU-R BT.1120/ITU-R BT.656/ITU-R BT.601/MCU I80/MCU M68/Serial TTL 输出

16.3.3.4 中断

VDP 支持两类中断
- 垂直时序中断
- 低带宽中断

16.3.3.5 垂直时序中断

VDP 包含一个显示通道，对应有一个垂直时序中断:
- 支持帧开始/结束中断。
- 支持中断屏蔽可配。
- 每个中断可写 1 清除。

16.3.3.6 低带宽中断

VDP 支持 polling 方式提供低带宽状态:
- 中断屏蔽可配。
- 中断写 1 清除。

16.3.4 功能描述

16.3.4.1 视频层功能

16.3.4.2 视频层 V 特性

- 支持输入影像格式 : 400, planar-420,planar-422,planar-444, RGB packet.
- 最小输入分辨率为 64x64,最大输入分辨率为 1920x1080。
- 最小输出分辨率为 64x64,最大输出分辨率为 1920x1080。
- 支持输入数据位宽: 8-bit 。
- YUV420 水平/垂直分辨率为 2 的倍数。
- YUV422 水平分辨率为 2 的倍数。
- YUV400/Planar-444(RGB or YUV)/RGB packet 无分辨率限制。
- 源起始地址可配，地址 32-byte 对齐。
  - 源 stride 可配，32-byte 对齐
- 支持色彩空间转换，支持对比度/亮度调节。
- 支持视频层 BT.601, BT.709 色域转换。
- 支持显示位置/大小可配，在屏幕任意位置显示。

16.3.4.3 图形层 G 特性

- 支持输入像素格式：ARGB8888,ARGB4444,ARGB1555,LUT8,LUT4,LUT1。
- 最小输入分辨率为 64x64,最大输入分辨率为 1920x1080。
  - 最小输出分辨率为 64x64,最大输出分辨率为 1920x1080。
- 源起始地址可配，地址 32-byte 对齐。
  - 源 stride 可配，32-byte 对齐。
  - 支持色彩空间转换。
  - 支持显示位置可配，在屏幕任意位置显示。
  - 支持 0~255 alpha。
  - 支持 colorkey 处理。

16.3.4.4 叠加处理

VDP 只支持 一层视频层，一层图形层迭加。

16.3.4.5 叠加特性

支持 8 window 迭加。

16.3.4.6 显示通道特性

- 可做为高清，标清的输出通道。
  - 只能选择一个输出接口输出。
  - 支持典型 1080P@60fps，720P@60fps 输出时序。

16.3.4.7 时序配置

VDP 输出接口可依不同对接芯片接口，支持配置各种典型及非典型时序。
所有时序参数必须在接口打开之前完成配置。

16.3.4.8 高清输出接口 MIPI Tx

- 支持 RGB888，RGB666,RGB565,RGB10-10-10 输出。
- 支持 1080P@60fps 4-channel 显示。
- 支持 720P@60fps 2/4-channel 显示

16.3.4.9 高清输出接口 BT.1120 特性

- 支持 YUV422 8bit 输出。
- 支持数据钳位的 clip，根据接口协议，Y 钳位范围 16~235，C 钳位范围 16~240。
- 支持以下典型输出时序：720P、1080P。
- 16bit 数据，默认 Y 在高位，C 在低位，YC 位置可互换。
- 支持色彩讯号 Cb/Cr 时序互换。
- 支持输出随路时钟反相
- 只支持 Progressive 时序。

16.3.4.10 标清输出接口 BT.656 特性

- 支持 YUV422 8bit 输出。
- 支持数据钳位的 clip，根据接口协议，Y 钳位范围 16~235，C 钳位范围 16~240。
- 支持 Y/C 时序互换，色彩讯号 Cb/Cr 时序互换。
- 只支持 Progressive 时序

16.3.4.11 BT.601 特性

16.3.4.12

- 支持 YUV422 8-bit 输出。
- YC 数据范围 0 ~ 255。
- 支持 VS/HS 同步讯号输出
- 支持 Y/C 时序互换，色彩讯号 Cb/Cr 时序互换。
- 只支持 Progressive 时序。

16.3.4.13 MCU 特性

- 支持 8-bit data, 4-bit control 输出。
- 支持 I80/M68 格式输出。

16.3.4.14 LCD LVDS 输出接口

支持 1-link LVDS 输出。
支持 6-bit/8-bit RGB 串行输出。
支持 VESA/JEIDA 格式输出。

图表 16-19 LVDS JEIDA/VESA mode 的 data format

16.3.5 VDP 寄存器概览

VDP DISP 寄存器位置为 0X0A08_8000 ~0x0A08_83FF

Name	Address Offset	Description
REG_00	0x000	
REG_01	0x004	
REG_02	0x008	
REG_03	0x00c	
REG_04	0x010	
REG_05	0x014	
REG_06	0x018	
REG_07	0x01c	
REG_13	0x034	
REG_14	0x038	
REG_15	0x03c	
REG_16	0x040	
REG_17	0x044	

Name	Address Offset	Description
REG_18	0x048	
REG_19	0x04c	
REG_20	0x050	
REG_21	0x054	
REG_22	0x058	
REG_23	0x05c	
REG_24	0x060	
REG_25	0x064	
REG_26	0x068	
REG_27	0x06c	
REG_28	0x070	
REG_29	0x074	
REG_30	0x078	
REG_31	0x07c	
REG_32	0x080	
REG_33	0x084	
REG_34	0x088	
REG_35	0x08c	
REG_36	0x090	
REG_37	0x094	
REG_39	0x09c	
REG_40	0x0a0	
REG_41	0x0a4	
REG_42	0x0a8	
REG_43	0x0ac	
REG_CATCH	0x0c0	
REG_OSD_FIFO_M0	0x130	
REG_OSD_FIFO_M1	0x134	
REG_OSD_FIFO_CNT0	0x138	
REG_OSD_FIFO_CNT1	0x13c	
REG_OSD_FIFO_CNT2	0x140	
REG_OSD_FIFO_CNT3	0x144	
REG_GAMMA_CTRL	0x180	
REG_GAMMA_WR_LUT	0x184	
REG_MCU_IF_CTRL	0x200	
REG_HW_MCU_AUTO	0x210	
REG_HW_MCU_CMD	0x214	
REG_HW_MCU_CMD_0	0x218	
REG_HW_MCU_CMD_1	0x21c	
REG_HW_MCU_CMD_2	0x220	
REG_HW_MCU_CMD_3	0x224	

Name	Address Offset	Description
REG_HW_MCU_CMD_4	0x228	
REG_HW_MCU_CMD_5	0x22c	
REG_HW_MCU_CMD_6	0x230	
REG_HW_MCU_CMD_7	0x234	
REG_HW_MCU_OV	0x238	
REG_SRGB_CTRL	0x240	
COV_W0_CFG	0x280	
COV_W0_SIZE	0x284	
COV_W0_COLOR	0x288	
COV_W1_CFG	0x28c	
COV_W1_SIZE	0x290	
COV_W1_COLOR	0x294	
COV_W2_CFG	0x298	
COV_W2_SIZE	0x29c	
COV_W2_COLOR	0x2a0	
COV_W3_CFG	0x2a4	
COV_W3_SIZE	0x2a8	
COV_W3_COLOR	0x2ac	
REG_TGEN_LITE_SIZE	0x304	
REG_TGEN_LITE_VS	0x308	
REG_TGEN_LITE_HS	0x314	

VDP OSD 寄存器位置为 0X0A08_8800 ~0x0A08_89FF

Name	Address Offset	Description
VGOP_REG_0	0x000	OW0_Setting_0
VGOP_REG_1	0x004	OW0_Setting_1
VGOP_REG_2	0x008	OW0_Setting_2
VGOP_REG_3	0x00c	OW0_Setting_3
VGOP_REG_4	0x010	OW0_Setting_3_1
VGOP_REG_5	0x014	OW0_Setting_4
VGOP_REG_6	0x018	OW0_Setting_5
VGOP_REG_10	0x020	OW1_Setting_0
VGOP_REG_11	0x024	OW1_Setting_1
VGOP_REG_12	0x028	OW1_Setting_2
VGOP_REG_13	0x02c	OW1_Setting_3
VGOP_REG_14	0x030	OW1_Setting_3_1
VGOP_REG_15	0x034	OW1_Setting_4
VGOP_REG_16	0x038	OW1_Setting_5
VGOP_REG_20	0x040	OW2_Setting_0
VGOP_REG_21	0x044	OW2_Setting_1
VGOP_REG_22	0x048	OW2_Setting_2
VGOP_REG_23	0x04c	OW2_Setting_3
VGOP_REG_24	0x050	OW2_Setting_3_1
VGOP_REG_25	0x054	OW2_Setting_4
VGOP_REG_26	0x058	OW2_Setting_5
VGOP_REG_30	0x060	OW3_Setting_0

Name	Address Offset	Description
VGOP_REG_31	0x064	OW3_Setting_1
VGOP_REG_32	0x068	OW3_Setting_2
VGOP_REG_33	0x06c	OW3_Setting_3
VGOP_REG_34	0x070	OW3_Setting_3_1
VGOP_REG_35	0x074	OW3_Setting_4
VGOP_REG_36	0x078	OW3_Setting_5
VGOP_REG_40	0x080	OW4_Setting_0
VGOP_REG_41	0x084	OW4_Setting_1
VGOP_REG_42	0x088	OW4_Setting_2
VGOP_REG_43	0x08c	OW4_Setting_3
VGOP_REG_44	0x090	OW4_Setting_3_1
VGOP_REG_45	0x094	OW4_Setting_4
VGOP_REG_46	0x098	OW4_Setting_5
VGOP_REG_50	0x0a0	OW5_Setting_0
VGOP_REG_51	0x0a4	OW5_Setting_1
VGOP_REG_52	0x0a8	OW5_Setting_2
VGOP_REG_53	0x0ac	OW5_Setting_3
VGOP_REG_54	0x0b0	OW5_Setting_3_1
VGOP_REG_55	0x0b4	OW5_Setting_4
VGOP_REG_56	0x0b8	OW5_Setting_5
VGOP_REG_60	0x0c0	OW6_Setting_0
VGOP_REG_61	0x0c4	OW6_Setting_1
VGOP_REG_62	0x0c8	OW6_Setting_2
VGOP_REG_63	0x0cc	OW6_Setting_3
VGOP_REG_64	0x0d0	OW6_Setting_3_1
VGOP_REG_65	0x0d4	OW6_Setting_4
VGOP_REG_66	0x0d8	OW6_Setting_5
VGOP_REG_70	0x0e0	OW7_Setting_0
VGOP_REG_71	0x0e4	OW7_Setting_1
VGOP_REG_72	0x0e8	OW7_Setting_2
VGOP_REG_73	0x0ec	OW7_Setting_3
VGOP_REG_74	0x0f0	OW7_Setting_3_1
VGOP_REG_75	0x0f4	OW7_Setting_4
VGOP_REG_76	0x0f8	OW7_Setting_5
VGOP_REG_80	0x100	Common
VGOP_REG_81	0x104	idx_sram_ctl
VGOP_REG_82	0x108	idx_sram_ctl
VGOP_REG_83	0x10c	clr_key_content
VGOP_REG_84	0x110	const_clr
VGOP_REG_85	0x114	debug
VGOP_REG_86	0x120	fb_thr
VGOP_REG_87	0x124	fb0_setting
VGOP_REG_88	0x128	fb0_init_st
VGOP_REG_89	0x12c	fb0_st_ro
VGOP_REG_90	0x134	fb1_setting
VGOP_REG_91	0x138	fb1_init_st
VGOP_REG_92	0x13c	fb1_st_ro
BW_LIMIT	0x140	
VGOP_DEC_00	0x150	vgop_dec_ctrl
VGOP_DEC_01	0x154	vgop_dec_debug
VGOP_LUT16_0	0x160	vgop_lut16_0_1
VGOP_LUT16_1	0x164	vgop_lut16_2_3
VGOP_LUT16_2	0x168	vgop_lut16_4_5
VGOP_LUT16_3	0x16c	vgop_lut16_6_7

Name	Address Offset	Description
VGOP_LUT16_4	0x170	vgop_lut16_8_9
VGOP_LUT16_5	0x174	vgop_lut16_10_11
VGOP_LUT16_6	0x178	vgop_lut16_12_13
VGOP_LUT16_7	0x17c	vgop_lut16_14_15

16.3.6 VDP 寄存器描述

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_disp_sel	R/W	disp soruce selection 0 : from DMA 1 : from scaler_disp Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
8:3	Reserved			
9	reg_vs_pol_lite	R/W	vsync polarity for tgen_lite 0 : high active 1 : low active	0x0
10	reg_hs_pol_lite	R/W	hsync polarity for tgen lite 0 : high active 1 : low active	0x0
11	reg_tgen_en_lite	R/W	tgen lite enable signal 0 : disable 1 : enable	0x0
15:12	reg_fmt_sel	R/W	disp source from DMA 4'h0 : YUV420 planar 4'h1 : YUV422 planar 4'h2 : RGB888 planar 4'h3 : RGB packed {R,G,B} 4'h4 : RGB packed {B,G,R} 4'h5 : Y only 4'h6 : reserved 4'h7 : reserved 4'h8 : NV21 {V,U} 4'h9 : NV12 {U,V} 4'ha : YUV422-SP1{V,U} 4'hb : YUV422-SP2{U,V} 4'hc : YUV2-1 {U,Y,V,Y} 4'hd : YUV2-2 {V,Y,U,Y} 4'he : YUV2-3 {Y,U,Y,V} 4'hf : YUV2-4 {Y,V,Y,U} Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:16	Reserved			

REG_01

Offset Address: 0x004

Bits	Name	Access	Description	Reset
13:0	reg_vtt	R/W	vtotal Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x167
15:14	Reserved			
29:16	reg_htt	R/W	htotal Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1df
31:30	Reserved			

REG_02

Offset Address: 0x008

Bits	Name	Access	Description	Reset
13:0	reg_vs_str	R/W	vsync start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_vs_stp	R/W	vsync end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x3
31:30	Reserved			

REG_03

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
13:0	reg_vfde_str	R/W	vfde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x6e
15:14	Reserved			
29:16	reg_vfde_stp	R/W	vfde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x15d
31:30	Reserved			

REG_04

Offset Address: 0x010

Bits	Name	Access	Description	Reset
13:0	reg_vmde_str	R/W	vmde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x6e
15:14	Reserved			
29:16	reg_vmde_stp	R/W	vmde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x15d
31:30	Reserved			

REG_05

Offset Address: 0x014

Bits	Name	Access	Description	Reset
13:0	reg_hs_str	R/W	hsync start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_hs_stp	R/W	hsync end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0xa
31:30	Reserved			

REG_06

Offset Address: 0x018

Bits	Name	Access	Description	Reset
13:0	reg_hfde_str	R/W	hfde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x14
15:14	Reserved			
29:16	reg_hfde_stp	R/W	hfde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x153
31:30	Reserved			

REG_07

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
13:0	reg_hmde_str	R/W	hmde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x14
15:14	Reserved			
29:16	reg_hmde_stp	R/W	hmde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x153
31:30	Reserved			

REG_13

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_src_y_base_0	R/W	source Y/R/packet DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_14

Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	reg_src_y_base_1	R/W	Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
31:8	Reserved			

REG_15

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	reg_src_u_base_0	R/W	source U/G DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_16

Offset Address: 0x040

Bits	Name	Access	Description	Reset
7:0	reg_src_u_base_1	R/W	Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:8	Reserved			

REG_17

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	reg_src_v_base_0	R/W	source V/B DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_18

Offset Address: 0x048

Bits	Name	Access	Description	Reset
7:0	reg_src_v_base_1	R/W	Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:8	Reserved			

REG_19

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
23:0	reg_src_y_pitch	R/W	source Y/R/packet line pitch, must be 32-byte aligned Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_20

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	reg_src_c_pitch	R/W	source UV/GB line pitch, must be 32-byte aligned Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
24	reg_64b_align	R/W	read address 64-byte align enable, only effective when reg_burst_ln se to 3,7,11 or 15	0x0
31:25	Reserved			

REG_21

Offset Address: 0x054

Bits	Name	Access	Description	Reset
15:0	reg_src_x_str	R/W	source crop x start position Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:16	reg_src_y_str	R/W	source crop y start position Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_22

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	reg_src_wd	R/W	effect when source is from DRAM Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:16	reg_src_ht	R/W	effect when source is from DRAM Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_23

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c00	R/W	display output csc coef C00, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_out_csc_c01	R/W	display output csc coef C01, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30	Reserved			
31	reg_out_csc_en	R/W	display output Color Space Convert Enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_24

Offset Address: 0x060

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c02	R/W	display output csc coef C02, s.3.10 Shadow: Yes	0x59c

Bits	Name	Access	Description	Reset
			Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
15:14	Reserved			
29:16	reg_out_csc_c10	R/W	display output csc coef C10, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
31:30	Reserved			

REG_25

Offset Address: 0x064

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c11	R/W	display output csc coef C11, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x2160
15:14	Reserved			
29:16	reg_out_csc_c12	R/W	display output csc coef C12, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x22db
31:30	Reserved			

REG_26

Offset Address: 0x068

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c20	R/W	display output csc coef C20, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_out_csc_c21	R/W	display output csc coef C21, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x717
31:30	Reserved			

REG_27

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c22	R/W	display output csc coef C22, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved			

REG_28

Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	reg_out_csc_sub_0	R/W	display output csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_out_csc_sub_1	R/W	display output csc sub value, 0~255	0x80

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
23:16	reg_out_csc_sub_2	R/W	display output csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
31:24	Reserved			

REG_29

Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	reg_out_csc_add_0	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_out_csc_add_1	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
23:16	reg_out_csc_add_2	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_30

Offset Address: 0x078

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c00	R/W	display input csc coef C00, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_in_csc_c01	R/W	display input csc coef C01, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30	Reserved			
31	reg_in_csc_en	R/W	display input Color Space Convert Enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_31

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c02	R/W	display input csc coef C02, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x59c
15:14	Reserved			
29:16	reg_in_csc_c10	R/W	display input csc coef C10, s.3.10 Shadow: Yes	0x400

Bits	Name	Access	Description	Reset
			Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
31:30	Reserved			

REG_32

Offset Address: 0x080

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c11	R/W	display input csc coef C11, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x2160
15:14	Reserved			
29:16	reg_in_csc_c12	R/W	display input csc coef C12, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x22db
31:30	Reserved			

REG_33

Offset Address: 0x084

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c20	R/W	display input csc coef C20, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_in_csc_c21	R/W	display input csc coef C21, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x717
31:30	Reserved			

REG_34

Offset Address: 0x088

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c22	R/W	display input csc coef C22, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved			

REG_35

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
7:0	reg_in_csc_sub_0	R/W	display input csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_in_csc_sub_1	R/W	display input csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
23:16	reg_in_csc_sub_2	R/W	display input csc sub value, 0~255 Shadow: Yes	0x80

Bits	Name	Access	Description	Reset
			Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
31:24	Reserved			

REG_36

Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_in_csc_add_0	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_in_csc_add_1	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
23:16	reg_in_csc_add_2	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_37

Offset Address: 0x094

Bits	Name	Access	Description	Reset
4:0	Reserved			
5	reg_fix_mc	R/W	mde window output fix color enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:6	Reserved			

REG_39

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
15:0	Reserved			
25:16	reg_fde_r	R/W	fde window r color	0x0
31:26	Reserved			

REG_40

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
9:0	reg_fde_g	R/W	fde window g color	0x0
15:10	Reserved			
25:16	reg_fde_b	R/W	fde window b color	0x3ff
31:26	Reserved			

REG_41

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
9:0	reg_mde_r	R/W	mde window r color	0x3ff
15:10	Reserved			
25:16	reg_mde_g	R/W	mde window g color	0x0

Bits	Name	Access	Description	Reset
31:26	Reserved			

REG_42

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
9:0	reg_mde_b	R/W	mde window b color	0x0
15:10	Reserved			
17:16	reg_out_bit	R/W	2 : 8-bit 3 : 6-bit others : 10-bit	0x2
19:18	reg_drop_md	R/W	2 : rounding 3 : direct drop others : noise dither	0x1
31:20	Reserved			

REG_43

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_disp_bw_fail	RO	data not ready for display, Low Bandwidth interrupt	
1	reg_clr_disp_bw_fail	W1T	clear reg_disp_bw_fail	
2	reg_osd_bw_fail	RO	OSD data not ready for display, Low Bandwidth interrupt	
3	reg_clr_osd_bw_fail	W1T	clear reg_osd_bw_fail	
31:4	Reserved			

REG_CATCH

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	reg_qos_sel_rr	R/W	qos criteria selection 0 : data in buffer + outstanding 1 : data in buffer only	0x1
31:3	Reserved			

REG_OSD_FIFO_M0

Offset Address: 0x130

Bits	Name	Access	Description	Reset
11:0	reg_osd_thres_0	R/W	osd fifo monitor threshold value 0	0x10
15:12	Reserved			
27:16	reg_osd_thres_1	R/W	osd fifo monitor threshold value 1	0x18
31:28	Reserved			

REG_OSD_FIFO_M1

Offset Address: 0x134

Bits	Name	Access	Description	Reset
11:0	reg_osd_thres_2	R/W	osd fifo monitor threshold value 2	0x20
15:12	Reserved			
27:16	reg_osd_thres_3	R/W	osd fifo monitor threshold value 3	0x28
31:28	Reserved			

REG_OSD_FIFO_CNT0

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_0	RO	osd fifo data less than threshold value 0 count	

REG_OSD_FIFO_CNT1

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_1	RO	osd fifo data less than threshold value 1 count	

REG_OSD_FIFO_CNT2

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_2	RO	osd fifo data less than threshold value 2 count	

REG_OSD_FIFO_CNT3

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_3	RO	osd fifo data less than threshold value 3 count	

REG_GAMMA_CTRL

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	reg_gamma_acc_lut	R/W	apb access gamma lut table	0x0
1	reg_gamma_acc_wr	R/W	0 : read , 1 : write	0x0
2	reg_gamma_en	R/W	gamma enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3	reg_gamma_pre_osd	R/W	1 : gamma -> osd 0 : osd -> gamma Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:4	Reserved			

REG_GAMMA_WR_LUT

Offset Address: 0x184

Bits	Name	Access	Description	Reset
7:0	reg_gamma_b_wdat	R/W	B gamma Lut write data	0x0
15:8	reg_gamma_g_wdat	R/W	G gamma Lut write data	0x0
23:16	reg_gamma_r_wdat	R/W	R gamma Lut write data	0x0
30:24	reg_gamma_addr	R/W	gamma lut address	0x0
31	reg_gamma_acc_w1t	W1T	write 1 to access lut (wrtie)	

REG_MCU_IF_CTRL

Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	reg_i80_if_en	R/W	display S.W. mcu interface enable (read prepared data from DRAM)	0x0
1	reg_i80_sw_mode_en	R/W	mcu sw step by step mode enable	0x0

Bits	Name	Access	Description	Reset
2	reg_i80_hw_if_en	R/W	display H.W. mcu interface enable (read RGB data fromDRAM) this mode is mutually exclusive with reg_i80_if_en, and the control is in REG_HW_MCU related register group	0x0
3	Reserved			
7:4	reg_i80_ctrl_ini	R/W	control signal initial value	0xf
9:8	Reserved			
10	reg_i80_ip_clr_w1t	W1T	write 1 to clear mcu interface IP	
11	reg_i80_run_w1t	W1T	write 1 to start mcu interface IP	
31:12	Reserved			

REG_HW_MCU_AUTO

Offset Address: 0x210

Bits	Name	Access	Description	Reset
0	reg_mcu_hw_trig	R/W	rising edge to start H.W MCU	0x0
1	reg_mcu_hw_stop	R/W	rising edge to stop H.W. MCU	0x0
3:2	reg_cs_h_hw_blk	R/W	csx signal state in video blanking duration 2'b00 : csx high in h-blanking & v-blanking 2'b01 : csx high in v-blanking 2'b1x : csx high when not sending video data	0x2
4	reg_mcu_565	R/W	H.W. MCU data out format 0 : RGB565, 2 TX cycle per pix 1 : RGB888, 3 TX cycle per pix (RGB666 also with this mode)	0x1
5	reg_pre_cmd_en	R/W	1 : insert cmd before H.W. MCU data send use reg_sw_tx related config, if this bit set to high a. please set reg_sw_tx related config b. set hfde_str number > reg_sw_tx_num	0x0
6	reg_hw_mcu_start_flag	RWC	H.W. MCU interface start to send data, write 1 to clear	
7	reg_hw_mcu_stop_flag	RWC	H.W. MCU interace stop to send data, write 1 to clear	
31:8	Reserved			

REG_HW_MCU_CMD

Offset Address: 0x214

Bits	Name	Access	Description	Reset
0	reg_mcu_sw_trig	R/W	rising edge to trig S.W. command command number is set in reg_mcu_sw_tx_num (max 16 command per trig) command queue is set in reg_sw_tx0 ~ reg_sw_txf	0x0
1	reg_cs_h_sw_idle	R/W	csx signal state after S.W. trig finish 1'b0 : keep low 1'b1 : go high	0x0
2	Reserved			

Bits	Name	Access	Description	Reset
3	reg_mcu_sw_tx_done	RWC	S.W. TX done flag, write 1 to clear	
7:4	reg_mcu_sw_tx_num	R/W	H.W. MCU engine send s.w. command number(0 ~ 15) -> (1~16)	0x0
11:8	reg_hw_mcu_state	RO	H.W. MCU status 0 : IDLE 1 : SW_TX 2 : WAIT_VS 3 : BLK 4 : HW_TX0 5 : HW_TX1 6 : HW_TX2	
15:12	reg_sw_tx_cnt_ro	RO	S.W. TX number by H.W. when finish, this value is set to 0	
31:16	Reserved			

REG_HW_MCU_CMD_0

Offset Address: 0x218

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_0	R/W	S.W. TX data [9:8] : OP code 2'b00 : command write 2'b01 : data write 2'b10 : command read 2'b11 : data read [7:0] TX data	0x0
15:10	Reserved			
25:16	reg_sw_tx_1	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_1

Offset Address: 0x21c

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_2	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_3	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_2

Offset Address: 0x220

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_4	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_5	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_3

Offset Address: 0x224

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_6	R/W		0x0
15:10	Reserved			

Bits	Name	Access	Description	Reset
25:16	reg_sw_tx_7	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_4

Offset Address: 0x228

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_8	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_9	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_5

Offset Address: 0x22c

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_a	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_b	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_6

Offset Address: 0x230

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_c	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_d	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_7

Offset Address: 0x234

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_e	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_f	R/W		0x0
31:26	Reserved			

REG_HW_MCU_OV

Offset Address: 0x238

Bits	Name	Access	Description	Reset
0	reg_i80_wrx_sw_ov	R/W	S.W. over-write i80 WRX enable	0x0
1	reg_i80_rdx_sw_ov	R/W		0x0
2	reg_i80_cdx_sw_ov	R/W		0x0
3	reg_i80_csx_sw_ov	R/W		0x0
4	reg_i80_dat_sw_ov	R/W		0x0
7:5	Reserved			
8	reg_i80_wrx_sw_dat	R/W		0x0
9	reg_i80_rdx_sw_dat	R/W		0x0

Bits	Name	Access	Description	Reset
10	reg_i80_cdx_sw_dat	R/W		0x0
11	reg_i80_csx_sw_dat	R/W		0x0
15:12	Reserved			
23:16	reg_i80_dat_sw_dat	R/W		0x0
31:24	Reserved			

REG_SRGB_CTRL

Offset Address: 0x240

Bits	Name	Access	Description	Reset
0	reg_srgb_ttl_en	R/W	serial RGB TTL interface enable	0x0
1	reg_srgb_ttl_4t	R/W	serial RGB output cycle per pixel 0 : 3T 1 : 4T	0x0
3:2	Reserved			
5:4	reg_srgb_ttl_g_num	R/W	sw overite serial RGB out G pixel idx, effective when reg_srgb_ttl_sw_seq = 1	0x0
7:6	reg_srgb_ttl_b_num	R/W	sw overite serial RGB out B pixel idx, effective when reg_srgb_ttl_sw_seq = 1	0x0
8	reg_srgb_ttl_sw_seq	R/W	sw redefine serial RGB output squence	0x0
31:9	Reserved			

COV_W0_CFG

Offset Address: 0x280

Bits	Name	Access	Description	Reset
11:0	reg_cover_w0_x_str	R/W	cover window 0 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w0_y_str	R/W	cover window 0 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w0_enable	R/W	cover window 0 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W0_SIZE

Offset Address: 0x284

Bits	Name	Access	Description	Reset
11:0	reg_cover_w0_wd_m1	R/W	cover window 0 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w0_ht_m1	R/W	cover window 0 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

Bits	Name	Access	Description	Reset
31:28	Reserved			

COV_W0_COLOR

Offset Address: 0x288

Bits	Name	Access	Description	Reset
7:0	reg_cover_w0_r	R/W	cover window 0 r color	0x0
15:8	reg_cover_w0_g	R/W	cover window 0 g color	0x0
23:16	reg_cover_w0_b	R/W	cover window 0 b color	0x0
31:24	Reserved			

COV_W1_CFG

Offset Address: 0x28c

Bits	Name	Access	Description	Reset
11:0	reg_cover_w1_x_str	R/W	cover window1 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w1_y_str	R/W	cover window1 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w1_enable	R/W	cover window 1 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W1_SIZE

Offset Address: 0x290

Bits	Name	Access	Description	Reset
11:0	reg_cover_w1_wd_m1	R/W	cover window 1 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w1_ht_m1	R/W	cover window 1 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W1_COLOR

Offset Address: 0x294

Bits	Name	Access	Description	Reset
7:0	reg_cover_w1_r	R/W	cover window 1 r color	0x0
15:8	reg_cover_w1_g	R/W	cover window 1 g color	0x0
23:16	reg_cover_w1_b	R/W	cover window 1 b color	0x0
31:24	Reserved			

COV_W2_CFG

Offset Address: 0x298

Bits	Name	Access	Description	Reset
11:0	reg_cover_w2_x_str	R/W	cover window 2 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w2_y_str	R/W	cover window 2 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w2_enable	R/W	cover window 2 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W2_SIZE

Offset Address: 0x29c

Bits	Name	Access	Description	Reset
11:0	reg_cover_w2_wd_m1	R/W	cover window 2 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w2_ht_m1	R/W	cover window 2 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W2_COLOR

Offset Address: 0x2a0

Bits	Name	Access	Description	Reset
7:0	reg_cover_w2_r	R/W	cover window 2 r color	0x0
15:8	reg_cover_w2_g	R/W	cover window 2 g color	0x0
23:16	reg_cover_w2_b	R/W	cover window 2 b color	0x0
31:24	Reserved			

COV_W3_CFG

Offset Address: 0x2a4

Bits	Name	Access	Description	Reset
11:0	reg_cover_w3_x_str	R/W	cover window 3 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w3_y_str	R/W	cover window 3 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

Bits	Name	Access	Description	Reset
30:28	Reserved			
31	reg_cover_w3_enable	R/W	cover window 3 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W3_SIZE

Offset Address: 0x2a8

Bits	Name	Access	Description	Reset
11:0	reg_cover_w3_wd_m1	R/W	cover window 3 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w3_ht_m1	R/W	cover window 3 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W3_COLOR

Offset Address: 0x2ac

Bits	Name	Access	Description	Reset
7:0	reg_cover_w3_r	R/W	cover window 3 r color	0x0
15:8	reg_cover_w3_g	R/W	cover window 3 g color	0x0
23:16	reg_cover_w3_b	R/W	cover window 3 b color	0x0
31:24	Reserved			

REG_TGEN_LITE_SIZE

Offset Address: 0x304

Bits	Name	Access	Description	Reset
13:0	reg_vtt_lite	R/W	vtotal Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x167
15:14	Reserved			
29:16	reg_htt_lite	R/W	htotal Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1df
31:30	Reserved			

REG_TGEN_LITE_VS

Offset Address: 0x308

Bits	Name	Access	Description	Reset
13:0	reg_vs_str_lite	R/W	vsync start Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_vs_stp_lite	R/W	vsync end Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x3

Bits	Name	Access	Description	Reset
31:30	Reserved			

REG_TGEN_LITE_HS

Offset Address: 0x314

Bits	Name	Access	Description	Reset
13:0	reg_hs_str_lite	R/W	hsync start Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_hs_stp_lite	R/W	hsync end Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0xa
31:30	Reserved			

VDP_OSD 寄存器描述

VGOP_REG_0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
3:0	reg_ow0_format	R/W	OW0 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	reg_ow0_h_start	R/W	OW0 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow0_h_end	R/W	OW0 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
11:0	reg_ow0_v_start	R/W	OW0 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update	0x0

Bits	Name	Access	Description	Reset
			Shadow Read Select: shdw_read_sel	
15:12	Reserved			
27:16	reg_ow0_v_end	R/W	OW0 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow0_dram_str_adr_l	R/W	OW0 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_4

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	reg_ow0_dram_str_adr_h	R/W	OW0 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_5

Offset Address: 0x014

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow0_dram_strip	R/W	OW0 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow0_crop_pixels	R/W	OW0 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_6

Offset Address: 0x018

Bits	Name	Access	Description	Reset
3:0	Reserved			

Bits	Name	Access	Description	Reset
13:4	reg_ow0_dram_hsize	R/W	OW0 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow0_dram_vsize	R/W	OW0 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_10

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	reg_ow1_format	R/W	OW1 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_11

Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_ow1_h_start	R/W	OW1 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow1_h_end	R/W	OW1 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_12

Offset Address: 0x028

Bits	Name	Access	Description	Reset
11:0	reg_ow1_v_start	R/W	OW1 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow1_v_end	R/W	OW1 V end pixel, unit: 1line Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
31:28	Reserved			

VGOP_REG_13

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow1_dram_str_adr_l	R/W	OW1 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_14

Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	reg_ow1_dram_str_adr_h	R/W	OW1 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_15

Offset Address: 0x034

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow1_dram_strip	R/W	OW1 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow1_crop_pixels	R/W	OW1 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_16

Offset Address: 0x038

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow1_dram_hsize	R/W	OW1 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsiz	0x0

Bits	Name	Access	Description	Reset
			e} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
15:14	Reserved			
27:16	reg_ow1_dram_vsize	R/W	OW1 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_20

Offset Address: 0x040

Bits	Name	Access	Description	Reset
3:0	reg_ow2_format	R/W	OW2 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_21

Offset Address: 0x044

Bits	Name	Access	Description	Reset
11:0	reg_ow2_h_start	R/W	OW2 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow2_h_end	R/W	OW2 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_22

Offset Address: 0x048

Bits	Name	Access	Description	Reset
11:0	reg_ow2_v_start	R/W	OW2 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow2_v_end	R/W	OW2 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_23

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow2_dram_str_adr_l	R/W	OW2 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_24

Offset Address: 0x050

Bits	Name	Access	Description	Reset
7:0	reg_ow2_dram_str_adr_h	R/W	OW2 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_25

Offset Address: 0x054

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow2_dram_strip	R/W	OW2 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow2_crop_pixels	R/W	OW2 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_26

Offset Address: 0x058

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow2_dram_hsize	R/W	OW2 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			

Bits	Name	Access	Description	Reset
27:16	reg_ow2_dram_vsize	R/W	OW2 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_30

Offset Address: 0x060

Bits	Name	Access	Description	Reset
3:0	reg_ow3_format	R/W	OW3 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_31

Offset Address: 0x064

Bits	Name	Access	Description	Reset
11:0	reg_ow3_h_start	R/W	OW3 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow3_h_end	R/W	OW3 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_32

Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	reg_ow3_v_start	R/W	OW3 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow3_v_end	R/W	OW3 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_33

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow3_dram_str_adr_l	R/W	OW3 DRAM Start address[31:0]	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	

VGOP_REG_34

Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	reg_ow3_dram_str_adr_h	R/W	OW3 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_35

Offset Address: 0x074

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow3_dram_strip	R/W	OW3 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow3_crop_pixels	R/W	OW3 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_36

Offset Address: 0x078

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow3_dram_hsize	R/W	OW3 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow3_dram_vsize	R/W	OW3 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

Bits	Name	Access	Description	Reset
31:28	Reserved			

VGOP_REG_40

Offset Address: 0x080

Bits	Name	Access	Description	Reset
3:0	reg_ow4_format	R/W	OW4 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_41

Offset Address: 0x084

Bits	Name	Access	Description	Reset
11:0	reg_ow4_h_start	R/W	OW4 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow4_h_end	R/W	OW4 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_42

Offset Address: 0x088

Bits	Name	Access	Description	Reset
11:0	reg_ow4_v_start	R/W	OW4 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow4_v_end	R/W	OW4 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_43

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow4_dram_str_adr_l	R/W	OW4 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_44

Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_ow4_dram_str_adr_h	R/W	OW4 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_45

Offset Address: 0x094

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow4_dram_strip	R/W	OW4 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow4_crop_pixels	R/W	OW4 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_46

Offset Address: 0x098

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow4_dram_hsize	R/W	OW4 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow4_dram_vsize	R/W	OW4 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_50

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
3:0	reg_ow5_format	R/W	OW5 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_51

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
11:0	reg_ow5_h_start	R/W	OW5 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow5_h_end	R/W	OW5 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_52

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
11:0	reg_ow5_v_start	R/W	OW5 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow5_v_end	R/W	OW5 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_53

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow5_dram_str_adr_l	R/W	OW5 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_54

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
7:0	reg_ow5_dram_str_adr_h	R/W	OW5 DRAM Start address[39:32] Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
31:8	Reserved			

VGOP_REG_55

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow5_dram_strip	R/W	OW5 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow5_crop_pixels	R/W	OW5 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_56

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow5_dram_hsize	R/W	OW5 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow5_dram_vsize	R/W	OW5 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_60

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
3:0	reg_ow6_format	R/W	OW6 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555	0x0

Bits	Name	Access	Description	Reset
			4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
31:4	Reserved			

VGOP_REG_61

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
11:0	reg_ow6_h_start	R/W	OW6 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow6_h_end	R/W	OW6 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_62

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
11:0	reg_ow6_v_start	R/W	OW6 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow6_v_end	R/W	OW6 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_63

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow6_dram_str_adr_l	R/W	OW6 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_64

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
7:0	reg_ow6_dram_str_adr_h	R/W	OW6 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_65

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow6_dram_strip	R/W	OW6 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow6_crop_pixels	R/W	OW6 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_66

Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow6_dram_hsize	R/W	OW6 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow6_dram_vsize	R/W	OW6 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_70

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
3:0	reg_ow7_format	R/W	OW7 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

Bits	Name	Access	Description	Reset
31:4	Reserved			

VGOP_REG_71

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_ow7_h_start	R/W	OW7 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow7_h_end	R/W	OW7 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_72

Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_ow7_v_start	R/W	OW7 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow7_v_end	R/W	OW7 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_73

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow7_dram_str_adr_l	R/W	OW7 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_74

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
7:0	reg_ow7_dram_str_adr_h	R/W	OW7 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_75

Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow7_dram_strip	R/W	OW7 DRAM Strip	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
22:16	reg_ow7_crop_pixels	R/W	OW7 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_76

Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow7_dram_hsize	R/W	OW7 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vsize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow7_dram_vsize	R/W	OW7 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_80

osd_common_ctrl

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_ow0_en	R/W	OSD Window0 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
1	reg_ow1_en	R/W	OSD Window1 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
2	reg_ow2_en	R/W	OSD Window2 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
3	reg_ow3_en	R/W	OSD Window3 enable Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
4	reg_ow4_en	R/W	OSD Window4 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
5	reg_ow5_en	R/W	OSD Window5 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
6	reg_ow6_en	R/W	OSD Window6 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
7	reg_ow7_en	R/W	OSD Window7 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
8	reg_vgop_hscal	R/W	VGOP H scale up (x2) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
9	reg_vgop_vscal	R/W	VGOP V scale up (x2) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
10	reg_clr_key_en	R/W	Color Key enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
11	Reserved			
15:12	reg_vgop_arlen	R/W	AXI-R burst length (INCR mode) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
30:16	Reserved			
31	reg_vgop_sw_rst	R/W	vgop software reset Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_81

LUT256 SRAM

Offset Address: 0x104

Bits	Name	Access	Description	Reset
15:0	reg_idx_wdata	R/W	Idx sram wdata (ARGB4444)	0x0
23:16	reg_idx_adr	R/W	Idx sram address	0x0
31:24	Reserved			

VGOP_REG_82

LUT256 SRAM

Offset Address: 0x108

Bits	Name	Access	Description	Reset
15:0	reg_idx_rdata	R	idx sram rdata	
16	reg_idx_wr	R/W	idx sram write	0x0

Bits	Name	Access	Description	Reset
17	reg_idx_rd	R/W	idx sram read	0x0
18	reg_vgop_db_clr	R/W	vgop debug flag clear	0x0
31:19	Reserved			

VGOP_REG_83

color key content
Offset Address: 0x10c

Bits	Name	Access	Description	Reset
23:0	reg_clr_key	R/W	Color Key (RGB888) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:24	Reserved			

VGOP_REG_84

constant color
Offset Address: 0x110

Bits	Name	Access	Description	Reset
15:0	reg_const_argb0	R/W	Constant0 ARGB4444 for Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_const_argb1	R/W	Constant1 ARGB4444 for Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_85

debug
Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_vgop_debug	R	debug	

VGOP_REG_86

fb_thr
Offset Address: 0x120

Bits	Name	Access	Description	Reset
5:0	reg_fb_clr_hi_thr	R/W	font_box brightness strong threshold , unit: 4 levels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x30
7:6	Reserved			
13:8	reg_fb_clr_lo_thr	R/W	font_box brightness wrak threshold , unit: 4 levels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x20
15:14	Reserved			
16	reg_fb_init	R/W	fb strong status init Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
17	reg_fb_font_is_dark	R/W	fb inv control	0x0

Bits	Name	Access	Description	Reset
			1 : font color is dark 0 : font color is light Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
19:18	Reserved			
25:20	reg_fb_diff_fnum	R/W	font_box brightness detect interval 0: stop detection update 1: always detection update 2~63: detection update if n-1 continous frames detection value is all difference Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
31:26	Reserved			

VGOP_REG_87

fb0_setting
Offset Address: 0x124

Bits	Name	Access	Description	Reset
6:0	reg_fb0_width	R/W	font_box_0 width , unit: 1 pixel, 1~128 pixels (reg_fb_width+1) * (reg_fb_num+1) must be equal to attached ow width Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0f
7	Reserved			
12:8	reg_fb0_pix_thr	R/W	font_box_0 strong pixel pixel threshold, unit: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x10
14:13	reg_fb0_sample_rate	R/W	font_box_0 sample rqtio 0: per pixel 1: 2 pixels 2: 4 pixels 3: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
15	Reserved			
20:16	reg_fb0_num	R/W	font_box_0 box numbers , 1~32 boxes Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1f
23:21	Reserved			
26:24	reg_fb0_attached_idx	R/W	font_box_0 attached ow number Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
27	Reserved			
28	reg_fb0_en	R/W	font_box_0 enable (active while attached ow is alive and	0x0

Bits	Name	Access	Description	Reset
			set as font base fromat) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
31:29	Reserved			

VGOP_REG_88

fb0_init_st
Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_fb0_init_st	R/W	font_box_0 init strong value Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_89

fb0_st_ro
Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	reg_fb0_record	R	font_box_0 strong value record	

VGOP_REG_90

fb1_setting
Offset Address: 0x134

Bits	Name	Access	Description	Reset
6:0	reg_fb1_width	R/W	font_box_1 width , unit: 1 pixel, 1~128 pixels (reg_fb_width+1) * (reg_fb_num+1) must be equal to attached ow width Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0f
7	Reserved			
12:8	reg_fb1_pix_thr	R/W	font_box_1 strong pixel pixel threshold, unit: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x10
14:13	reg_fb1_sample_rate	R/W	font_box_1 sample rqtio 0: per pixel 1: 2 pixels 2: 4 pixels 3: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
15	Reserved			
20:16	reg_fb1_num	R/W	font_box_1 box numbers , 1~32 boxes Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1f
23:21	Reserved			
26:24	reg_fb1_attached_idx	R/W	font_box_1 attached ow number Shadow: Yes	0x7

Bits	Name	Access	Description	Reset
			Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	
27	Reserved			
28	reg_fb1_en	R/W	font_box_1 enable (active while attached ow is alive and set as font base fromat) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:29	Reserved			

VGOP_REG_91

fb1_init_st

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	reg_fb1_init_st	R/W	font_box_1 init strong value Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_92

fb1_st_ro

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	reg_fb1_record	R	font_box_1 strong value record	

BW_LIMIT

Offset Address: 0x140

Bits	Name	Access	Description	Reset
9:0	reg_bwl_win	R/W	B.W. limit window period, unit : clk_axi cycle	0x0
15:10	Reserved			
25:16	reg_bwl_vld	R/W	B.W. limit valid number in reg_bwl_win, unit : Byte	0x0
30:26	Reserved			
31	reg_bwl_en	R/W	B.W. limit enable	0x0

VGOP_DEC_00

vgop_dec_ctrl

Offset Address: 0x150

Bits	Name	Access	Description	Reset
0	reg_odec_en	R/W	vgop decoder enable 0: bypass 1: decoder mode Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
1	reg_odec_int_en	R/W	vgop_decoder INT enable	0x0
2	reg_odec_int_clr	R/W	vgop_decoder INT clear	0x0
3	reg_odec_wdt_en	R/W	vgop wdt enable	0x0
7:4	reg_odec_dbg_ridx	R/W	vgop_decoder debug idx	0x0
8	reg_odec_done	R	vgop_decoder done	
11:9	Reserved			

Bits	Name	Access	Description	Reset
14:12	reg_odec_attached_idx	R/W	vgop decoder attached ow number Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15	Reserved			
18:16	reg_odec_wdt_fdiv_bit	R/W	wdt precision	0x0
23:19	Reserved			
31:24	reg_odec_int_vec	R	vgop_decoder INT vector 1: odec done 2: Watch Dog time out 4: detect reduntant data input	

VGOP_DEC_01

vgop_dec_debug
Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_odec_dbg_rdata	R	vgop decoder debug rdata	

VGOP_LUT16_0

vgop_lut16_0_1
Offset Address: 0x160

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt0	R/W	LUT 16 content 0 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt1	R/W	LUT 16 content 1 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_1

vgop_lut16_2_3
Offset Address: 0x164

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt2	R/W	LUT 16 content 2 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt3	R/W	LUT 16 content 3(ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_2

vgop_lut16_4_5
Offset Address: 0x168

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt4	R/W	LUT 16 content 4 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt5	R/W	LUT 16 content 5 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update	0x0

Bits	Name	Access	Description	Reset
			Shadow Read Select: shdw_read_sel	

VGOP_LUT16_3

vgop_lut16_6_7
Offset Address: 0x16c

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt6	R/W	LUT 16 content 6 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt7	R/W	LUT 16 content 7 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_4

vgop_lut16_8_9
Offset Address: 0x170

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt8	R/W	LUT 16 content 8 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt9	R/W	LUT 16 content 9 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_5

vgop_lut16_10_11
Offset Address: 0x174

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt10	R/W	LUT 16 content 10 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt11	R/W	LUT 16 content 11 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_6

vgop_lut16_12_13
Offset Address: 0x178

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt12	R/W	LUT 16 content 12 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt13	R/W	LUT 16 content 13 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_7

vgop_lut16_14_15

Offset Address: 0x17c

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt14	R/W	LUT 16 content 14 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt15	R/W	LUT 16 content 15 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

16.4 MIPI Rx

16.4.1 概述

MIPI Rx (Mobile Industry Processor Interface Receiver) 模块主要功能为接收由 CMOS sensor 所传送的视频数据，支持 MIPI D-PHY、sub-LVDS (Low-Voltage Differential Signal)、HiSPi (High-Speed Serial Pixel Interface) 等不同的串行视频信号输入，并将其处理转化为内部视频时序，传递给下一级的视频处理模块 (ISP)。
MIPI Rx 模块中可细分为 PHY 和 Controller 两部分，其中 PHY 模块集成了模拟和数字两个部分，主要将串行信号转换为并行信号，而 Controller 模块则负责解码不同的视频数据格式，传送给后端的视频处理模块 (ISP)。功能框图及在系统中的位置如图表 16-20 所示。

图表 16-20 MIPI Rx 功能框图及在系统中的位置

16.4.2 特点

• 支持 MIPI DPHY-ver2.1

- 可同时支持 2 路 sensor 输入
- 单一 sensor 最大支持 5M(2688x1944, 2880x1620) @30fps HDR or @60fps 线性输入
- 双路 sensor 最大支持 FHD(1920x1080) @60fps HDR or linear 输入
- 单路最多支持 4-Lane MIPI D-PHY 接口，最大支持 1.5Gbps/Lane
- 单路最多支持 4-Lane sub-LVDS/ HiSPi 接口，最大支持 1.5Gbps/Lane
- 支持 RAW8/ RAW10/ RAW12/RAW16 数据类型的解析
- 支持 YUV422 8-bit/ YUV422 10-bit 数据类型的解析
- 最多支持 2 帧 WDR，支持多种 WDR 时序
- 支持 sub-LVDS/ HiSPi 模式像素/同步码大小端配置
- 支持 Lane 数和 Lane 顺序可配置

16.4.3 功能描述

16.4.3.1 典型应用

在使用图像传感器的应用中，可根据不同的接口选择(MIPI/ Sub-LVDS/ HiSPi)，设置 MIPI Rx 模块寄存器，同时 MIPI Rx 也支持不同速度和不同分辨率的传输需求，并兼容多种图像传感器格式。
MIPI Rx 模块中包含 1 组 D-PHY，每组分别有六个差分对，一个 D-PHY 可支持一对差分时钟，最多配上四对数据差分信号，或同时支持两组一对差分时钟配上两对数据差分信号，因此 MIPI Rx 可以同时支持 2 路 Sensor 输入。另外 MIPI Rx 可支持不同的差分对排序以及时钟差分对位置，可以通过寄存器配置时钟的来源和差分对排序方式。
MIPI Rx 只针对接口的时序转换和解码，不处理影像处理的部分。所以在满足带宽的前提下可支持任意分辨率和帧率。MIPI Rx 的带宽有两部分限制：PHY 的接口数据率和内部处理速度。输入接口最大支持 1.5Gbps/Lane，内部处理速度最大为 600M*1pixels/s。

	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI DPHY	200mV	200mV	750MHz	1.5Gbps
Sub-LVDS	900mV	150mV	750MHz	1.5Gbps
HiSPi(HiVCM)	900mV	280mV	750MHz	1.5Gbps
HiSPi(SLVDS)	200mV	200mV	750MHz	1.5Gbps

图表 16-21 MIPI Rx 支持接口类型

16.4.3.2 MIPI 接口数据格式

MIPI 规范由不同的工作组负责开发和维护，分别对应不同领域的应用。MIPI Rx 支持 D-PHY 和 CSI-2 (Camera Serial Interface)。D-PHY 规定了物理层的传输规范，CSI-2 规定了 Camera 输出数据包的格式和协议。

- D-PHY
  D-PHY 是 MIPI 联盟发布的高速物理层标准，规定了接口层的物理特性和传输协议。D-PHY 采用了 200mV 源同步的低压差分信号技术，每个 Lane 的数据绿率范围支持到 2500Mbps。D-PHY 可以工作在低功耗 (Low Power, LP) 和高速 (High Speed, HS) 两种模式下。
- CSI-2
  CSI-2 是针对摄像头的数据协议，规定了主机与外设通信的数据包格式。
  CSI-2 可以支持不同像素格式的图像应用，数据传输的最小粒度是字节。为增加 CSI-2 的性能，可以选择数据 Lane 的数量，CSI-2 协议规订了发送端将像素数据打包成字节的机制，并指明多个数据 Lane 分配和管理的方式。字节数据以数据包的形式组织，数据包在 SoT 与 EoT 之间传输。 接收端根据协议解析相应的数据包，恢复出原始的像素数据。
  MIPI Rx 支持 RAW8/ RAW10/ RAW12/RAW16/ YUV422-8bit/ YUV422-10bit 数据类型的解析。
  CSI-2 的数据包分为长包和短包两种，包含有校验码，能进行误码纠正和错误检测。

长包和短包都是在 SoT 和 EoT 之间传输，在数据传送的间隙，D-PHY 处于 LP 模式。CSI-2 数据包的传输机制如图所示。 PH 和 PF 分别表示 Packet Header 和 Packet Footer。

图表 16-22 数据包的传输机制

长包用于传输有效像素数据，分为五个部分：Data ID，Word Count，ECC，PAYLOAD，CHECKSUM。
Data ID 包含 Virtual Channel 和 Data Type。Virtual Channel 控制传输所用的通道，可以使用不同的通道传输不童的数据。Data Type 则指定传输数据的类型。
Word Count 表示接收端需要接收到的数据量。
ECC 是纠错码，可以纠正或检测 Data Type 和 Word Count 的误码。
PAYLOAD 为实际需要传输的像素数据。
CHECKSUM 是利用线性反馈移位寄存器所产生的校验和，用于 PAYLOAD 资料的校验。
长包的结构如图表 16-23 所示。

图表 16-23 CSI-2 长包格式

短包的作用是同步传输信息，包含 Data ID，Data Field 和 ECC 三个部分，其格式如图表 16-24 所示。

图表 16-24 CSI-2 短包格式

MIPI Rx 共支持六种视频资料格式的传输，包含 YUV422-8bit、YUV422-10bit、RAW8、RAW10、RAW12 和 RAW16。不同的资料格式的传输方式如下所述。

YUV422-8bit 的传输模式为 UYVY 的形式，如图表 16-25。

图表 16-25 YUV422 8-bit 资料传输顺序

封包相对于视频信号的对应则如图表 16-26 所示。

图表 16-26 YUV422-8bit 资料封包传输对应

整张 Frame 的传输格式就会成为如图表 16-27 所示。

图表 16-27 YUV422 8-bit Frame 格式

YUV422-10bit 的传输模式同样为 UYVY，传输顺序如图表 16-28 所示。

图表 16-28 YUV422-10bit 资料传输顺序

封包相对于视频信号的对应则如图表 16-29 所示。

图表 16-29 YUV422-10bit 资料封包传输对应

整张 Frame 的传输格式就会成为如图表 16-30 所示。

图表 16-30 YUV422-10bit Frame 格式

RAW8 的传输顺序如图表 16-31 所示。

图表 16-31 RAW8 资料传输顺序

整张 Frame 的传输格式就会成为如图表 16-32 所示。

图表 16-32 RAW8 Frame 格式

RAW10 的传输顺序如图表 16-33 所示。

图表 16-33 RAW10 资料传输顺序

整张 Frame 的传输格式就会成为如图表 16-34 所示。

FS															
	Packer Header, PH	P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs	Packer Footer, PF	
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		
		P1	P2	*P3*	P4	LSBs	P5	....	P637	P638	*P639*	P640	LSBs		FE

图表 16-34 RAW10 Frame 格式

RAW12 的传输顺序如图表 16-35 所示。

图表 16-35 RAW12 资料传输顺序

整张 Frame 的传输格式就会成为如图表 16-36 所示。

FS															
	Packer Header, PH	P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs	Packer Footer, PF	
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		
		P1	P2	LSBs	*P3*	P4	LSBs	....	P638	LSBs	*P639*	P640	LSBs		FE

图表 16-36 RAW12 Frame 格式

RAW16 的传输顺序如图表 16-37 所示。

Line Start	Packet Header	**P1[15:8]**	**P1[7:0]**	P2[15:8]	P2[7:0]	***P3[15:8]***	***P3[7:0]***	P4[15:8]
Line End	***P637[7:0]***	P638[15:8]	P638[7:0]	***P639[15:8]***	***P639[7:0]***	P640[15:8]	P640[7:0]	Packet Footer

图表 16-37 RAW16 资料传输顺序

整张 Frame 的传输格式就会成为如图表 16-38 所示。

FS	Packer Header, PH										Packer Footer, PF	
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		
	P1	**P1**	P2	P2	***P3***	....	***P639***	***P639***	P640	P640		FE

图表 16-38 RAW16 Frame 格式

16.4.3.3 MIPI 接口线性模式

MIPI 接口的线性模式传输格式如图表 16-39 所示。每张图的传输都是以短包 Frame Start (FS)作为起式，短包 Frame End (FE)作为结束。中间的视频内容则是以行为单位，每个长包传输一条完整的视频行。长包格式如 MIPI 标准所规范每一行有 32bit 的数据包头 (PH, Pecket Header)，其中包含了当前行的 Virtual Channel 和 Data Type 等信息。

图表 16-39 MIPI 接口图像格式

16.4.3.4 MIPI 接口宽动态模式

MIPI Rx 支持四种 MIPI 接口的宽动态 (WDR) 模式，分别为：
1. 使用 DT (Data Type) 区分长短曝光数据
2. 使用识别码 ID (Identification Code) 区分长短曝光数据
3. 利用寄存器设定长短曝光数据延迟区间

使用 DT 的 WDR 传输方式如图表 16-40 所示，不同曝光长度共用一组 FS/FE 短包，而长包的包头中包含有 DT 信息，可利用不同的 DT 来区分长短曝光数据，其中真正的数据格式 DT 和代表长短曝光数据的两组 DT 都可以利用寄存器来设定，MIPI Rx 便能够解析出正确的宽动态时序传送给后方的视频处理模块。

图表 16-40 MIPI 接口宽动态数据传输 (使用 DT)

使用 ID 的 WDR 传输方式如图表 16-41 所示，不同曝光长度共用一组 FS/FE 短包，而传输数据中每个长包的前四个像素用来传输代表不同曝光长度的 ID (Identification Code)，代表长短曝的 ID 可利用寄存器设置，MIPI Rx 会利用 ID 分别不同曝光视频信号，并将前四个像素去除后再传送给视频处理模块。

图表 16-41 MIPI 接口宽动态数据传输 (使用 ID)

最后一种支持的 WDR 传输方式则是没有任何的 DT 或 ID 来表示传输的长包是长曝或短曝的内容，使用者必须自行设定寄存器表明长曝和短曝之间的曝光行数差异，MIPI Rx 会解析出相对应的时序给视频处理模块，实际传输时序如图表 16-42 所示。

图表 16-42 MIPI 接口宽动态数据传输 (寄存器设定)

16.4.3.5 Sub-LVDS 接口数据格式

超低电压差分信号 sub-LVDS (Low-Voltage Differential Signal) 普遍应用于前端摄像头，通过同步码区分有效视频信号的区间以及宽动态模式的长短曝。

MIPI Rx 的 PHY 将差分串行数据转换为并行数据，接着 MIPI Rx 的控制器把并行数据根据不同的模式和同步码解码出像素数据。

MIPI Rx 支持 8bit、10bit 和 12bit 三种位元宽度的 Sub-LVDS 传输模式，接口数据格式如图表 16-43 所示，所有有效的视频信号都会在 SAV 和 EAV 同步码中间，其中同步码都是由四个字段构成，每个字段的位宽与后方的像素位宽相同，而前三个字段为固定的基准码字，第四个字段可用来区分有效区间的起始或结束，Sub-LVDS 同步码格式如图表 16-44 所示，同步码根据不同的厂商会使用不同的数值，图表 16-44 只是其中一种实现方式，不同的数值可以在寄存器中设置。

图表 16-43 Sub-LVDS 接口数据格式

12-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFFh	000h	000h	AB0h
	End sync code (EAV)				B60h
Effective line	Start sync code (SAV)				800h
	End sync code (EAV)				9D0h
10-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	3FFh	000h	000h	2ACh
	End sync code (EAV)				2D8h
Effective line	Start sync code (SAV)				200h
	End sync code (EAV)				274h
8-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFh	00h	00h	ABh
	End sync code (EAV)				B6h
Effective line	Start sync code (SAV)				80h
	End sync code (EAV)				9Dh

图表 16-44 Sub-LVDS 同步码举例

而 Sub-LVDS 的同步码和像素资讯在不同 Lane 的传输模式如图表 16-45 所示，每一条 Lane 都会传输相同的同步码，紧接着才是像素数据，像素数据会根据使用的通道数以像素位宽为单位照顺序排列。

图表 16-45 Sub-LVDS Multi Lane 传输模式

Sub-LVDS 中同步码和像素数据是串行的，而 MIPI Rx 支持数据的大小端可利用寄存器设置，以大端模式为例，输出单个像素点的时序如图表 16-46 所示。

图表 16-46 Sub-LVDS 单个像素时序图

16.4.3.6 Sub-LVDS 接口线性模式

Sub-LVDS 在线性模式中，利用同步码标明一张图像数据中每条线的起始和结束，而在同步码 SAV 和 EAV 以外的就不是有效视频数据，如图表 16-47 所示。

SAV blanking line	V-blanking	EAV blanking line	H-blanking
	V-blanking		H-blanking
	V-blanking		H-blanking
	V-blanking		H-blanking
SAV effective line	Effective Pixel	EAV effective line	H-blanking
	Effective Pixel		H-blanking
	⋮		⋮
	Effective Pixel		H-blanking
	Effective Pixel		H-blanking
SAV blanking line	V-blanking	EAV blanking line	H-blanking
	⋮		⋮
	V-blanking		H-blanking

图表 16-47 Sub-LVDS 线性模式时序图

16.4.3.7 Sub-LVDS 接口宽动态模式

MIPI Rx 可支持两种 Sub-LVDS 接口宽动态模式，第一种模式中如图表 16-48，长短曝的视频信号分别都有 SAV 和 EAV 同步码包住，MIPI Rx 可利用不同的同步码解析出该视频信号是长曝还是短曝。第二种模式则是如图表 16-49 将长曝和短曝用同一组 SAV 和

EAV 包住，必须在寄存器中设置每条线的宽度和 blanking 长度，MIPI Rx 必须利用这些寄存器的设定和同步码解析出长曝和短曝分别的时序，接着再传送到视频处理模块。

<table>
<tr><td rowspan="3">SAV blanking</td><td>Blanking</td><td rowspan="3">EAV blanking</td><td rowspan="9">Blanking</td><td rowspan="4">SAV blanking</td><td>Blanking</td><td rowspan="4">EAV blanking</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>Blanking</td><td>Blanking</td></tr>
<tr><td rowspan="5">SAV effective</td><td>LEF Effective Pixel</td><td rowspan="5">EAV effective</td><td rowspan="5">SAV effective</td><td>SEF Effective Pixel</td><td rowspan="5">EAV effective</td></tr>
<tr><td>LEF Effective Pixel</td><td>SEF Effective Pixel</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>LEF Effective Pixel</td><td>SEF Effective Pixel</td></tr>
<tr><td>LEF Effective Pixel</td><td>SEF Effective Pixel</td></tr>
<tr><td rowspan="3">SAV blanking</td><td>Blanking</td><td rowspan="3">EAV blanking</td><td>SAV blanking</td><td>Blanking</td><td>EAV blanking</td></tr>
<tr><td>⋮</td><td></td><td></td><td></td></tr>
<tr><td>Blanking</td><td></td><td></td><td></td></tr>
</table>

图表 16-48 Sub-LVDS 宽动态模式一

<table>
<tr><td rowspan="3">SAV blanking</td><td>Blanking</td><td rowspan="12">Blanking</td><td rowspan="4">Blanking</td><td rowspan="3">EAV blanking</td></tr>
<tr><td>⋮</td></tr>
<tr><td>Blanking</td></tr>
<tr><td>SAV only LEF</td><td>LEF Effective Pixel</td><td rowspan="2">EAV only LEF</td></tr>
<tr><td>SAV only LEF</td><td>LEF Effective Pixel</td></tr>
<tr><td rowspan="4">SAV LEF & SEF</td><td rowspan="2">⋮</td><td>Blanking</td></tr>
<tr><td>SEF Effective Pixel</td><td rowspan="4">EAV LEF & SEF</td></tr>
<tr><td>SEF Effective Pixel</td></tr>
<tr><td>LEF Effective Pixel</td><td rowspan="4">⋮</td></tr>
<tr><td>LEF Effective Pixel</td></tr>
<tr><td rowspan="3">SAV only SEF</td><td>Blanking</td><td rowspan="3">EAV only SEF</td></tr>
<tr><td>⋮</td><td>SEF Effective Pixel</td></tr>
<tr><td>SEF Effective Pixel</td></tr>
<tr><td></td><td>Blanking</td><td>Blanking</td><td></td></tr>
</table>

图表 16-49 Sub-LVDS 宽动态模式二

16.4.3.8 HiSPi 接口数据格式

High-Speed Serial Pixel (HiSPi) 接口也被使用于某些摄像头，和 Sub-LVDS 类似使用同步码来区分有效的视频信息和在宽动态模式中区分长短曝。HiSPi 规范中定义了四种不同的打包模式，分别为 Packetized-SP、Streaming-SP、Streaming-S 和 ActiveStart-SP8 四种不同的传输模式。

MIPI Rx 支持其中较为常见的 Packetized-SP 和 Streaming-SP 两种传输方式。

16.4.3.9 HiSPi 接口线性模式

MIPI Rx 支持两种不同的 HiSPi 模式，在 Packetized-SP 模式中如图表 16-50 所示，图像传感器使用 SOF 表示有效视频信号的第一行，并用 EOF 表示有效视频信号最后一行的结束。其他有效视频信号则是使用 SOL 和 EOL 作为一行的起始和结束。

<table>
<tr><td>SOF</td><td>Effective Pixel</td><td rowspan="5">EOL</td><td>H-blanking</td></tr>
<tr><td rowspan="4">SOL</td><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>Effective Pixel</td><td>EOF</td><td>H-blanking</td></tr>
<tr><td colspan="4">V-blanking</td></tr>
</table>

图表 16-50 HiSPi Packetized-SP 模式

在 Streaming-SP 模式中如图表 16-51 所示，图像传感器并不传送代表结束的 EOL 或是 EOF，因此 MIPI Rx 控制器必须利用寄存器的设置得知有效视频信号的数量，才能解析出正确的视频信号传送给视频处理模块 (ISP)。另外 Streaming-SP 模式也支持表示空白行数的 SAV 信号，两种不同传输方式所支持的同步码整理如图表 16-52 所示。

<table>
<tr><td>SOF</td><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td rowspan="4">SOL</td><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>SOV</td><td colspan="2">V-blanking</td></tr>
</table>

图表 16-51 HiSPi Streaming-SP 模式

同步码	Packetized-SP	Streaming-SP
SOF	Required	Required
SOL	Required	Required
EOF	Required	Unsupported
EOL	Required	Unsupported
SAV	Unsupported	Required

图表 16-52 HiSPi 同步码支援模式

16.4.3.10 HiSPi 接口宽动态模式

HiSPi 接口宽动态模式同样分为两种不同的模式，第一种 Packetized-SP 如图表 16-53 所示，长曝和短曝会利用不同的同步码来区别，其中在 SOF_L 和 EOF_L 中的才是有效的长曝视频信号，同样在 SOF_S 和 EOF_S 中的才是有效的短曝视频信号。在长曝的最后和短曝的起始几行并不是有效的像素区，而是以固定值填充。

<table>
<tr><td>SOF_L</td><td>LEF Effective Pixel</td><td rowspan="9">EOL_F</td><td>H-blanking</td><td rowspan="3">SOL_S</td><td>Padding Row</td><td rowspan="9">EOL_S</td><td>H-blanking</td></tr>
<tr><td rowspan="9">SOL_L</td><td>LEF Effective Pixel</td><td>H-blanking</td><td>Padding Row</td><td>H-blanking</td></tr>
<tr><td rowspan="2">⋮</td><td rowspan="7">⋮</td><td>Padding Row</td><td rowspan="7">⋮</td></tr>
<tr><td>Padding Row</td></tr>
<tr><td rowspan="2">SOF_S</td><td>SEF Effective Pixel</td></tr>
<tr><td>LEF Effective Pixel</td><td rowspan="6">SOL_S</td><td>SEF Effective Pixel</td></tr>
<tr><td>LEF Effective Pixel</td><td>EOF_F</td><td rowspan="2">⋮</td></tr>
<tr><td>Padding Row</td><td rowspan="5">EOL_F</td></tr>
<tr><td>Padding Row</td></tr>
<tr><td>Padding Row</td><td>H-blanking</td><td>SEF Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>Padding Row</td><td>H-blanking</td><td>SEF Effective Pixel</td><td>EOF_S</td><td>H-blanking</td></tr>
<tr><td colspan="8">V-blanking</td></tr>
</table>

图表 16-53 HiSPi Packetized-SP 宽动态传输

第二种 Streaming-SP 的宽动态传输则如图表 16-54 所示，其中长曝和短曝的同步码和线性模式相同，因此需要利用寄存器设置长曝短曝之间的曝光行数差距，MIPI Rx 才能解析出正确的宽动态视频信号。

<table>
<tr><td>SOF</td><td>LEF Effective Pixel</td><td rowspan="9">EOL</td><td>H-blanking</td><td rowspan="9">SOL</td><td>Padding Row</td><td rowspan="9">EOL</td><td>H-blanking</td></tr>
<tr><td rowspan="8">SOL</td><td>LEF Effective Pixel</td><td>H-blanking</td><td>Padding Row</td><td>H-blanking</td></tr>
<tr><td rowspan="2">⋮</td><td rowspan="7">⋮</td><td>Padding Row</td><td rowspan="7">⋮</td></tr>
<tr><td>Padding Row</td></tr>
<tr><td>SEF Effective Pixel</td></tr>
<tr><td>SEF Effective Pixel</td></tr>
<tr><td>LEF Effective Pixel</td><td rowspan="6">⋮</td></tr>
<tr><td>LEF Effective Pixel</td></tr>
<tr><td>Padding Row</td></tr>
<tr><td>Padding Row</td></tr>
<tr><td>Padding Row</td><td>H-blanking</td><td>SEF Effective Pixel</td><td>H-blanking</td></tr>
<tr><td>Padding Row</td><td>H-blanking</td><td>SEF Effective Pixel</td><td>EOF</td><td>H-blanking</td></tr>
<tr><td>SOV</td><td>V-blanking</td><td>EOV</td><td></td><td>SOV</td><td>V-blanking</td><td>EOV</td><td></td></tr>
</table>

图表 16-54 HiSPi Streaming-SP 宽动态传输

16.4.4 MIPI Rx 寄存器概览

在芯片中最多可以同时使用两组 MIPI Rx 模块，其中主要分为三组寄存器，第一个部分是控制 PHY 模块的寄存器，基址为 0x0A0D0000，第二部分是控制 CSI 模块的寄存器，

基址为 0x0A0C2400 和 0x0A0C4400，第三部分则是控制 Sub-LVDS 和 HiSPi 模块的寄存器，基址为 0x0A0C2200 和 0x0A0C4200。

图表 16-55 MIPI Rx PHY 寄存器概览

Name	Address Offset	Description
REG_00	0x000	PD_CTRL
REG_04	0x004	ANALOG_CTRL
REG_30	0x030	SENSOR_MODE_CTRL
REG_34	0x034	ANALOG_CAL_0
REG_38	0x038	ANALOG_CAL_1
REG_3C	0x03c	ANALOG_CAL_2
REG_40	0x040	ANALOG_CAL_3
REG_44	0x044	ANALOG_CAL_4
REG_48	0x048	ANALOG_CAL_5
REG_80	0x080	CLOCK_INVERCE_CTL
REG_A0	0x0a0	
REG_A4	0x0a4	
REG_A8	0x0a8	
REG_AC	0x0ac	

Offset = 0x0A0D0300

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_24	0x024	SLVDS_CTRL_1
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7
REG_D2_0	0x180	D2_REG_CTRL_CALIB_0
REG_D2_1	0x184	D2_REG_CTRL_CALIB_1

Name	Address Offset	Description
REG_D2_3	0x18c	D2_CALIB_RESULT_0
REG_D2_4	0x190	D2_CALIB_RESULT_1
REG_D2_5	0x194	D2_CALIB_RESULT_2
REG_D2_6	0x198	D2_CALIB_RESULT_3
REG_D2_7	0x19c	D2_CALIB_RESULT_4
REG_D2_8	0x1a0	D2_CALIB_RESULT_5
REG_D2_9	0x1a4	D2_CALIB_RESULT_6
REG_D2_A	0x1a8	D2_CALIB_RESULT_7
REG_D3_0	0x1c0	D3_REG_CTRL_CALIB_0
REG_D3_1	0x1c4	D3_REG_CTRL_CALIB_1
REG_D3_3	0x1cc	D3_CALIB_RESULT_0
REG_D3_4	0x1d0	D3_CALIB_RESULT_1
REG_D3_5	0x1d4	D3_CALIB_RESULT_2
REG_D3_6	0x1d8	D3_CALIB_RESULT_3
REG_D3_7	0x1dc	D3_CALIB_RESULT_4
REG_D3_8	0x1e0	D3_CALIB_RESULT_5
REG_D3_9	0x1e4	D3_CALIB_RESULT_6
REG_D3_A	0x1e8	D3_CALIB_RESULT_7

Offset = 0x0A0D0600

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7

图表 16-56 MIPI Rx CSI 控制器寄存器概览

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL

Name	Address Offset	Description
REG_04	0x004	INTR_CTRL
REG_08	0x008	HDR_CTRL_0
REG_0C	0x00c	HDR_CTRL_1
REG_10	0x010	HDR_CTRL_2
REG_14	0x014	BLC_CTRL
REG_18	0x018	HDR_CTRL_3
REG_1C	0x01c	HDR_CTRL_4
REG_20	0x020	HDR_CTRL_5
REG_24	0x024	HDR_CTRL_6
REG_40	0x040	CSI_STATUS
REG_60	0x060	
REG_70	0x070	CSI_VS_GEN
REG_74	0x074	HDR_DT_CTRL

图表 16-57 MIPI Rx Sub-LVDS 控制器寄存器总览

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL
REG_04	0x004	SYNC_CODE_0
REG_08	0x008	SYNC_CODE_1
REG_0C	0x00c	SYNC_CODE_2
REG_10	0x010	SYNC_CODE_3
REG_14	0x014	SYNC_CODE_4
REG_18	0x018	SYNC_CODE_5
REG_1C	0x01c	SYNC_CODE_6
REG_20	0x020	SYNC_CODE_7
REG_24	0x024	SYNC_CODE_8
REG_28	0x028	SYNC_CODE_9
REG_2C	0x02c	VS_GEN
REG_30	0x030	LANE_MODE
REG_50	0x050	SYNC_CODE_A
REG_54	0x054	SYNC_CODE_B
REG_58	0x058	HDR_PATTEN_2
REG_60	0x060	HISPI_MODE_CTRL_0
REG_64	0x064	HISPI_MODE_CTRL_1
REG_68	0x068	HISPI_MODE_CTRL_2
REG_6C	0x06c	HISPI_MODE_CTRL_3
REG_70	0x070	HISPI_MODE_CTRL_4
REG_74	0x074	HISPI_MODE_CTRL_5
REG_80	0x080	DBG_SEL

16.4.5 MIPI Rx 寄存器描述

第一部分为 MIPI Rx PHY 寄存器叙述

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
13:0	Reserved			

Bits	Name	Access	Description	Reset
14	reg_mipirx_pd_ibias	R/W	Power down analog ibias	0x1
15	Reserved			
21:16	reg_mipirx_pd_rxlp	R/W	Power down analog RXLP	0x3f
31:22	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_mipirx_sel_clk_channel	R/W	Analog macro clock lane select	0x0
30:22	Reserved			
31	reg_mipimpll_clk_csi_en	R/W	Gating test clock from mipimpll	0x0

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_sensor_phy_mode	R/W	Sensor PHY mode enable select 0: 1C4D 1: 1C2D + 1C2D else: reserved	0x0
31:3	Reserved			

REG_34

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal0	RO	Analog lane 0 calibration result	

REG_38

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal1	RO	Analog lane 1 calibration result	

REG_3C

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal2	RO	Analog lane 2 calibration result	

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal3	RO	Analog lane 3 calibration result	

REG_44

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal4	RO	Analog lane 4 calibration result	

REG_48

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal5	RO	Analog lane 5 calibration result	

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_ad_d0_clk_inv	R/W	AD clock lane0 inverse	0x0
1	reg_ad_d1_clk_inv	R/W	AD clock lane1 inverse	0x0
2	reg_ad_d2_clk_inv	R/W	AD clock lane2 inverse	0x0
3	reg_ad_d3_clk_inv	R/W	AD clock lane3 inverse	0x0
4	reg_ad_d4_clk_inv	R/W	AD clock lane4 inverse	0x0
5	reg_ad_d5_clk_inv	R/W	AD clock lane5 inverse	0x0
31:6	Reserved			

REG_A0

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vtt	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_vs_str	R/W		0x0
31:30	Reserved			

REG_A4

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vs_stp	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_htt	R/W		0x0
31:30	Reserved			

REG_A8

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
13:0	reg_cam0_hs_str	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_hs_stp	R/W		0x0
31:30	Reserved			

REG_AC

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0_vs_pol	R/W		0x0
1	reg_cam0_hs_pol	R/W		0x0
2	reg_cam0_tgen_en	R/W		0x0
31:3	Reserved			

Offset = 0x0A0D0300

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3	Reserved			
6:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2
7	Reserved			
10:8	reg_csi_lane_d2_sel	R/W	Data lane 2 select	0x3
11	Reserved			
14:12	reg_csi_lane_d3_sel	R/W	Data lane 3 select	0x4
31:15	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
10	reg_csi_lane_d2_pnswap	R/W	Data lane 2 pn swap	0x0
11	reg_csi_lane_d3_pnswap	R/W	Data lane 3 pn swap	0x0
15:12	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
3:0	reg_deskew_lane_en	R/W	Deskew lane enable 4'h0: No lane 4'h1: 1-lane 4'h3: 2-lane 8'hf: 4-lane	0x0
31:4	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode	0x2

Bits	Name	Access	Description	Reset
			2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	
7:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
15:8	Reserved			
27:16	reg_slvds_sav_1st	R/W	Sub-LVDS sync code 1st symbol	0xfff
31:28	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sav_2nd	R/W	Sub-LVDS sync code 2nd symbol	0x0
15:12	Reserved			
27:16	reg_slvds_sav_3rd	R/W	Sub-LVDS sync code 3rd symbol	0x0
31:28	Reserved			

REG_D0_0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D2_0

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	reg_d2_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d2_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d2_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d2_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d2_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d2_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d2_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D2_1

Offset Address: 0x184

Bits	Name	Access	Description	Reset
0	reg_d2_calib_en	R/W	Calibration software enable	0x0
1	reg_d2_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d2_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d2_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D2_3

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D2_4

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D2_5

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D2_6

Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D2_7

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D2_8

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D2_9

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D2_A

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D3_0

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
0	reg_d3_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d3_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d3_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d3_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d3_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d3_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d3_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D3_1

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
0	reg_d3_calib_en	R/W	Calibration software enable	0x0
1	reg_d3_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d3_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d3_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D3_3

Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D3_4

Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D3_5

Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D3_6

Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D3_7

Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D3_8

Offset Address: 0x1e0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D3_9

Offset Address: 0x1e4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D3_A

Offset Address: 0x1e8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_7	RO	Calibration result phase 224~255	

Offset = 0x0A0D0600

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3:2	Reserved			
5:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2
31:6	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3:2	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
15:10	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
1:0	reg_deskew_lane_en	R/W	Deskew lane enable 2'h0: No lane 2'h1: 1-lane 2'h3: 2-lane	0x0
31:2	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
5:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
31:6	Reserved			

REG_D0_0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

第二部分为 MIPI Rx CSI 控制器寄存器。

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_mode	R/W	Lane mode 3'b000: 1-lane 3'b001: 2-lane 3'b011: 4-lane 3'b111: 8-lane	0x0
3	reg_csi_ignore_ecc	R/W	Ignore ecc result 1'b0: normal 1'b1: still processing even ecc error	0x0
4	reg_csi_vc_check	R/W	VC check enable 1'b0: do not check VC 1'b1: only process packets that meet vc_set[3:0]	0x0
7:5	Reserved			
11:8	reg_csi_vc_set	R/W	VC set only use when reg_csi_vc_check assert	0x0
12	reg_csi_line_start_sent	R/W	LS and LE packet sent 1'b0: create hsync signal by controller 1'b1: use LS and LE to create hsync signal	0x0
31:13	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_mask	R/W	Interrupt mask control	0x0
15:8	reg_csi_intr_clr	W1T	Interrupt clear	
16	reg_csi_hdr_en	R/W	HDR mode enable	0x0
17	reg_csi_hdr_mode	R/W	HDR mode selection 1'b0: HDR VC mode 1'b1: HDR ID mode	0x0
18	reg_csi_id_rm_else	R/W	Remove non reconized ID line 1'b0: dont remove 1'b1: remove	0x1

Bits	Name	Access	Description	Reset
19	reg_csi_id_rm_ob	R/W	Remove ob line 1'b0: don't remove 1'b1: remove	0x1
31:20	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_lef	R/W	ID for LEF ob n0	0x221
31:16	reg_csi_n0_ob_sef	R/W	ID for SEF ob n0	0x222

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_lef	R/W	ID for LEF active n0	0x241
31:16	reg_csi_n1_ob_lef	R/W	ID for LEF ob n1	0x231

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
15:0	reg_csi_n1_ob_sef	R/W	ID for SEF ob n1	0x232
31:16	reg_csi_n1_lef	R/W	ID for LEF active n1	0x251

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
5:0	reg_csi_blc_dt	R/W	Data type for optical black line	0x37
7:6	Reserved			
8	reg_csi_blc_en	R/W	Optical black line mode enable	0x0
11:9	Reserved			
14:12	reg_csi_blc_format_set	R/W	Optical black line data format set 3'd0: YUV422 8bit 3'd1: YUV422 10bit 3'd2: RAW8 3'd3: RAW10 3'd4: RAW12 3'd5: RAW16 else: reserved	0x2
31:15	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
3:0	reg_csi_vc_map_ch00	R/W	VC mapping to ISP channel 00	0x0
7:4	reg_csi_vc_map_ch01	R/W	VC mapping to ISP channel 01	0x1
11:8	reg_csi_vc_map_ch10	R/W	VC mapping to ISP channel 10	0x2
15:12	reg_csi_vc_map_ch11	R/W	VC mapping to ISP channel 11	0x3
31:16	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef	R/W	ID for SEF active n0	0x242

Bits	Name	Access	Description	Reset
31:16	reg_csi_n1_sef	R/W	ID for SEF active n1	0x252

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef2	R/W	ID for SEF2 active n0	0x244
31:16	reg_csi_n1_sef2	R/W	ID for SEF2 active n1	0x254

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_sef2	R/W	ID for SEF2 ob n0	0x224
31:16	reg_csi_n1_ob_sef2	R/W	ID for SEF2 ob n1	0x234

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_csi_ecc_no_error	RO	ECC no error	
1	reg_csi_ecc_corrected_error	RO	ECC corrected error	
2	reg_csi_ecc_error	RO	ECC error	
3	Reserved			
4	reg_csi_crc_error	RO	CRC error	
5	reg_csi_wc_error	RO	WC error	
7:6	Reserved			
8	reg_csi_fifo_full	RO	CSI FIFO full	
15:9	Reserved			
21:16	reg_csi_decode_format	RO	CSI decode format from header bit[0]: YUV422 8bit bit[1]: YUV422 10bit bit[2]: RAW8 bit[3]: RAW10 bit[4]: RAW12 bit[5]: RAW16	
31:22	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_status	RO	Interrupt status bit[0]: ecc error bit[1]: crc error bit[2]: hdr id error bit[3]: word count error bit[4]: fifo full	
31:8	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
1:0	reg_csi_vs_gen_mode	R/W	2'b00: vs gen by FS 2'b01: vs gen by FE else: vs gen by FS & FE	0x2

Bits	Name	Access	Description	Reset
3:2	Reserved			
4	reg_csi_vs_gen_by_vcset	R/W	Vsync generation setting 1'b0: generated by all vc short packet 1'b1: only generated by indicated vc short packet	0x0
31:5	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
0	reg_csi_hdr_dt_mode	R/W	CSI HDR DT mode enable	0x0
3:1	Reserved			
9:4	reg_csi_hdr_dt_format	R/W	CSI HDR DT mode video format data type	0x0
11:10	Reserved			
17:12	reg_csi_hdr_dt_lef	R/W	CSI HDR DT mode LEF data type	0x0
19:18	Reserved			
25:20	reg_csi_hdr_dt_sef	R/W	CSI HDR DT mode SEF data type	0x0
31:26	Reserved			

第三部分为 MIPI Rx Sub-LVDS 控制寄存器。

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
9:8	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000

Bits	Name	Access	Description	Reset
15:12	Reserved			
27:16	reg_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802

Bits	Name	Access	Description	Reset
31:28	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01
31:28	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

REG_28

Offset Address: 0x028

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

REG_2C

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
11:0	reg_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_slvds_lane_mode	R/W	Sub-LVDS lane mode 2'b0: 1-lane	0x3

Bits	Name	Access	Description	Reset
			2'b1: 2-lane 2'b3: 4-lane 2'b7: 8-lane	
3	Reserved			
11:4	reg_slvds_sync_source	R/W	Sub-LVDS output sync source select	0x1
31:12	Reserved			

REG_50

Offset Address: 0x050

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3
31:28	Reserved			

REG_54

Offset Address: 0x054

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			
27:16	reg_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

REG_58

Offset Address: 0x058

Bits	Name	Access	Description	Reset
13:0	reg_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

REG_64

Offset Address: 0x064

Bits	Name	Access	Description	Reset
11:0	reg_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00
31:28	Reserved			

REG_68

Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

REG_6C

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
7:0	reg_dbg_sel	R/W	Debug signal select	0x0
31:8	Reserved			

16.5 MIPI Tx

16.5.1 概述

The Display Serial Interface（DSI）接口是移动行业处理器接口联盟（Mobile Industry Processor Interface alliance, MIPI联盟）定义的一种高速串行接口，主要用于处理器和显示模块之间的连接。MIPI Tx 接口实现 DSI 接口，支持 MIPI D-PHY V1.0 串行信号输出。MIPI Tx 包含模拟 PHY 和数字 Controller 两部分，系统架构及功能框图如图所示。

图表 16-58 MIPI TX 功能框图

16.5.2 特点

MIPI Tx 有以下特点：

- 支持 1/2/4 Data Lane MIPI D-PHY 接口，顺序，PN 极可配置。
- 高速模式最大支持 2500Mbps/Lane。
- 仅 Data Lane0 支持低速发送及接收、BTA (Bus Turn-Around)功能，低速模式速度最高 10Mbps。
- 支持 DSI RGB16/18/24/30 bit 数据类形输出。
- 支持 DSI video mode 和 command mode, video mode 支持 Burst mode、 Non-burst mode with Sync Events 和 Non-burst mode with Sync pulses。

16.5.3 功能描述

MIPI Tx 包括 Tx D-PHY 和 Tx Controller 两部分

- Tx D-PHY 支持 MIPI D-PHY ver2.1 协议，主要实现了物理层的传输规范。
- Tx Controller 根据 MIPI DSI 协议对数据格式进行封装。

16.5.3.1 Tx D-PHY

Tx D-PHY 有两种工作模式，高速(High Speed, HS)和低速(Low Power, LP)：

- Video mode 的数据通过高速模式传输。
- Command mode 的数据通过低速模式来传输。

高速模式每个通道(Lane)的数率范围为 80~2500Mbps, 低速模式的最大速率为 10Mbps。
高速模式最大支持 4 个 data lane，实际使用的 data lane 个数可以为 1/2/4，顺序及极性可配置。低速模式的发送、接收及 Bus Turn-Around(BTA)仅配置后的 data lane0 支持。

16.5.3.2 Tx Controller

16.5.3.3 数据包的发送

当有复数个高速封包而要传送时，Tx D-PHY会依据Tx Controller送出来高速资料传输需求自动进行HS和LP 模式的切换。Tx Controller 支持HS传输结时是否发送EoT 包（End of Transmission，EoT）。

16.5.3.4 数据类型

控制器支持 DSI RGB16/18/24/30 bit 的发送，各种数据类型的组成格式如图所示。

图表 16-59 RGB16-bit 格式 (Data type = 0x0E)

图表 16 -60 RGB 18-bit 格式 (Data type = 0x1E)

注意 : RGB 18-bit 仅支持 data type = 0x0E, 不支持 loosely Packet 模式(data type = 0x2E)

图表 16-61 RGB 24-bit 格式 (Data type = 0x3E)

图表 16-62 RGB 30-bit 格式 (Data type = 0x0D)

16.5.3.5 接口时序

时序标记如下

图表 16-63 MIPI TX 时序标记

VSA：帧同步行数
VBP：帧后消隐行数
VACT：帧有效数据行

VFP：帧前消隐行
VSS：帧同步信号

HSS：行同步信号
HBP：行后消隐区
RGB：行有效数据
HFP：行前消隐区

BLLP：Blanking or Low-Power Interval

Video 垂直时序为
帧同步信号(VSS)，帧后消隐区(VBP)，有效行区(VACT)，帧前消隐区(VFP)，帧同步信号。
水平时序为
行同步信号 (HSS)，行后消隐区(HBP), 有效画素(HACT)，行前消隐区(HFP)，行同步信号。
有效同步信号为 VSS, HSS, 有效画素为 有效行区(VACT) 和 有效画素(HACT) 之交集。

Video mode Burst Transmission 时序如下

图表 16-64 MIPI TX Video mode Burst Transmission 时序

MIPI Tx 只传输有效同步信号及数据，其余时间进入 BLLP 区以降低功耗。

16.5.4 MIPI Tx 寄存器概览

MIPI Tx Control 寄存器位置为 0x0A08A000

Name	Address Offset	Description
DSI_MAC_REG_00	0x000	DSI_MAC_EN
DSI_MAC_REG_01	0x004	HS_REG_00
DSI_MAC_REG_02	0x008	HS_REG_01
DSI_MAC_REG_03	0x00c	ESC_REG
DSI_MAC_REG_04	0x010	ESC_REG_TX0
DSI_MAC_REG_05	0x014	ESC_REG_TX1
DSI_MAC_REG_06	0x018	ESC_REG_TX2
DSI_MAC_REG_07	0x01c	ESC_REG_TX3
DSI_MAC_REG_08	0x020	ESC_REG_RX0
DSI_MAC_REG_09	0x024	ESC_REG_RX1

16.5.5 MIPI Tx 寄存器描述

DSI_MAC_REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_sw_spkt_en	RWS	trigger soft short packet transmission (only work while reg_video_mode=0)	
1	reg_esc_en	RWS	Escape mode enable	
2	reg_video_mode	RWS	HS video mode enable 0: HS command mode or Escape mode 1: HS Video mode	
3	Reserved			
4	reg_sw_spkt_done	R	soft Short packet transmission finish , and clear after reg_sw_spkt_en deassert	
5	reg_esc_done_d0	R	Escape mode finish	
31:6	Reserved			

DSI_MAC_REG_01

Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_sw_spkt	R/W	Soft Short packet for HS mode [7:0] Data ID [15:8]: Data 0 [23:16] Data 1 (* ECC is generated by HW)	0x0
25:24	reg_lane_mode	R/W	lane active setting while HS mode 0: D0 Lane active 1: D0/D1 Lane Active 2: D0/D1/D2/D3 Lane Active 3: reserved	0x1
26	reg_eotp_en	R/W	enable insert eopt short packet	0x1

Bits	Name	Access	Description	Reset
			followed each transmission	
27	reg_skew_cal	R/W	skew calibration	0x0
28	reg_skew_cal_ini	R/W	skew calibration initialize	0x0
29	reg_hs_c_conti	R/W	c lane continous mode	0x1
31:30	reg_rgb_clr_fmt	R/W	RGB Packed format 0: RGB888 1: RGB666 2: RGB565 3: RGB10-10-10	0x0

DSI_MAC_REG_02

Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_lpkt_wc	R/W	Video line data paket byte counts (packed data must be byte alignment)	0x0
26:16	reg_event_delay	R/W	time shift for each video packet event (unit: pixel time)	0x80
31:27	Reserved			

DSI_MAC_REG_03

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	reg_esc_mode	R/W	Escape mode 0: TX trigger 1: TX LPDT 2: ULPS 3: ULPS EXIT 4: RX 5: reserved 6: C ULPS 7: C ULPS EXIT	0x1
3	Reserved			
7:4	reg_esc_trig	R/W	Trigger command for Escape TX trigger mode	0x0
11:8	reg_tx_bc	R/W	TX LPDT byte counts 0: 1 byte 1: 2 bytes ..... 15: 16 bytes	0x0
15:12	reg_bta_rx_bc	R	Byte counts for RX received 0: no received 1: 1 byte ..... 8: 8 bytes	
31:16	Reserved			

DSI_MAC_REG_04

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte0	R/W	TX byte 0	0x0
15:8	reg_tx_byte1	R/W	TX byte 1	0x0
23:16	reg_tx_byte2	R/W	TX byte 2	0x0

Bits	Name	Access	Description	Reset
31:24	reg_tx_byte3	R/W	TX byte 3	0x0

DSI_MAC_REG_05

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte4	R/W	TX byte 4	0x0
15:8	reg_tx_byte5	R/W	TX byte 5	0x0
23:16	reg_tx_byte6	R/W	TX byte 6	0x0
31:24	reg_tx_byte7	R/W	TX byte 7	0x0

DSI_MAC_REG_06

Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte8	R/W	TX byte 8	0x0
15:8	reg_tx_byte9	R/W	TX byte 9	0x0
23:16	reg_tx_bytea	R/W	TX byte 10	0x0
31:24	reg_tx_byteb	R/W	TX byte 11	0x0

DSI_MAC_REG_07

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
7:0	reg_tx_bytec	R/W	TX byte 12	0x0
15:8	reg_tx_byted	R/W	TX byte 13	0x0
23:16	reg_tx_bytee	R/W	TX byte 14	0x0
31:24	reg_tx_bytef	R/W	TX byte 15	0x0

DSI_MAC_REG_08

Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	reg_bta_rx_byte0	R	RX byte 0	
15:8	reg_bta_rx_byte1	R	RX byte 1	
23:16	reg_bta_rx_byte2	R	RX byte 2	
31:24	reg_bta_rx_byte3	R	RX byte 3	

DSI_MAC_REG_09

Offset Address: 0x024

Bits	Name	Access	Description	Reset
7:0	reg_bta_rx_byte4	R	RX byte 4	
15:8	reg_bta_rx_byte5	R	RX byte 5	
23:16	reg_bta_rx_byte6	R	RX byte 6	
31:24	reg_bta_rx_byte7	R	RX byte 7	

16.5.6 MIPI Tx PHY 寄存器概览

MIPI Tx PHY 寄存器位置为 0x0A0D_1000

Name	Address Offset	Description
REG_00	0x000	DSI PHY REG_00
REG_01	0x004	DSI PHY REG_01
REG_02	0x008	DSI PHY REG_02
REG_03	0x00c	DSI PHY REG_03
REG_04	0x010	DSI PHY REG_04
REG_05	0x014	DSI PHY REG_05
REG_23	0x08c	DSI PHY REG_23
REG_24	0x090	DSI PHY REG_24
REG_25	0x094	DSI PHY REG_25
REG_26	0x098	DSI PHY REG_26
REG_27	0x09c	DSI PHY REG_27
REG_28	0x0a0	DSI PHY REG_28
REG_2D	0x0b4	DSI PHY REG_2D

16.5.7 MIPI Tx PHY 寄存器描述

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_clk_lane_en	R/W	MIPI Tx clock Lane Enable	0x0
1	reg_dat_0_lane_en	R/W	MIPI Tx data Lane 0 enable	0x0
2	reg_dat_1_lane_en	R/W	MIPI Tx data Lane 1 enable	0x0
3	reg_dat_2_lane_en	R/W	MIPI Tx data Lane 2 enable	0x0
4	reg_dat_3_lane_en	R/W	MIPI Tx data Lane 3 enable	0x0
31:5	Reserved			

REG_01

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	reg_t_clk_prepare	R/W	CLK-PREPARE timing parameter, unit : 8UI	0x5
15:8	reg_t_clk_zero	R/W	CLK-ZERO timing parameter, unit : 8UI	0x24
23:16	reg_t_clk_pre	R/W	CLK-PRE timing parameter, unit : 8UI	0x8
31:24	reg_t_clk_post	R/W	CLK-POST timing parameter, unit : 8UI	0x20

REG_02

Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	reg_t_clk_trail	R/W	CLK-TRAIL timing parameter, unit : 8UI	0x1
31:8	Reserved			

REG_03

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
19:0	reg_t_esc_init	R/W	ESC-INIT timing parameter, unit : T_LPX	0x1000
31:20	Reserved			

REG_04

Offset Address: 0x010

Bits	Name	Access	Description	Reset
19:0	reg_t_esc_wakeup	R/W	ESC-WAKEUP timing parameter, unit : T_LPX	0x4e20
31:20	Reserved			

REG_05

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	reg_t_hs_pre_on	R/W	HS-PRE-ON timing parameter, unit : 8UI	0x1
15:8	reg_t_hs_prepare	R/W	HS-PREPARE timing parameter, unit : 8UI	0x6
23:16	reg_t_hs_zero	R/W	HS-ZERO timing parameter, unit : 8UI	0x20
31:24	reg_t_hs_trail	R/W	HS-TRAIL timing parameter, unit : 8UI	0x1

REG_23

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_en_ssc	R/W	MIPI Tx Spred Spectrum Enable	0x0
3:2	reg_ssc_mode	R/W	Spread Spectrum Mode	0x0
31:4	Reserved			

REG_24

Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	reg_set	R/W	Frequency Synthesizer divider, format 6.26	0x11F51 4F9

REG_25

Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	reg_span	R/W	SSC Span parameter	0x0
31:16	Reserved			

REG_26

Offset Address: 0x098

Bits	Name	Access	Description	Reset
23:0	reg_step	R/W	SSC Step parameter	0x0
31:24	Reserved			

REG_27

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
2:0	reg_dsi_lane_0_sel	R/W	MIPI Tx PAD0 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x1
3	Reserved			
6:4	reg_dsi_lane_1_sel	R/W	MIPI Tx PAD1 Config 0 : clock lane 1 : Data Lane 0	0x2

Bits	Name	Access	Description	Reset
			2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	
7	Reserved			
10:8	reg_dsi_lane_2_sel	R/W	MIPI Tx PAD2 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x0
11	Reserved			
14:12	reg_dsi_lane_3_sel	R/W	MIPI Tx PAD3 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x3
15	Reserved			
18:16	reg_dsi_lane_4_sel	R/W	MIPI Tx PAD4 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x4
23:19	Reserved			
28:24	reg_sel_clk_lane	R/W	MIPI Tx Clock lane Config 5'h01 : PAD0 as MIPI Tx Clock Lane 5'h02 : PAD1 as MIPI Tx Clock Lane 5'h04 : PAD2 as MIPI Tx Clock Lane 5'h08 : PAD3 as MIPI Tx Clock Lane 5'h10 : PAD4 as MIPI Tx Clock Lane	0x4
31:29	Reserved			

REG_28

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_dsi_lane_0_pn_swap	R/W	MIPI Tx PAD0 PN swap enable	0x0
1	reg_dsi_lane_1_pn_swap	R/W	MIPI Tx PAD1 PN swap enable	0x0
2	reg_dsi_lane_2_pn_swap	R/W	MIPI Tx PAD2 PN swap enable	0x0
3	reg_dsi_lane_3_pn_swap	R/W	MIPI Tx PAD3 PN swap enable	0x0
31:4	Reserved			

REG_2D

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
0	reg_lvds_en	R/W	MIPI Tx modue in LVDS Tx mode	0x0
31:1	Reserved			

17 音频接口

17.1 AIAO

17.1.1 概述

音频输入输出接口 (Audio Input/Audio Output)用于和芯片内置 Audio Codec 或芯片外置的 Audio Codec 及数位麦克风做对接，完成音频数据的发送和接收，实现录音、放音、对讲等功能。芯片内部将 AIAO 相关模块集成在一个子系统内，内置的 Audio Codec ADC/DAC，支持立体声输入和输出。对外支持两组 I2S 接口，内部则集成 4 组 I2S TX/RX 模块，可同时接收发送音频数据，并可支持多声道数据同时收送。基本模块框图如下图所示：

图表 17-1 AIAO 框图

17.1.2 特点

AIAO 接口支持主模式(Master-mode)、从模式(Slave-mode) I2S 和 PCM 模式，并支持多通道 TDM 模式。接收和发送音频数据通过 DMA 方式存取 DDR 空间。具体特点如下：

- 高弹性化可配置的时序参数，帧周期、帧同步信号持续时间和极性(polarity)皆可配置
- 信号发出和采样的时钟缘可配置
- 支持主模式和从模式立体声 I2S 模式音频数据的发送和接收
- 支持主模式和从模式单声道和立体声 PCM 模式音频数据的发送和接收
- 支持主模式和从模式多声道 TDM 模式音频数据的发送和接收
- 接收和发送可以单独或同时使能
- 数据采用 DMA 操作，可通过软件开辟的缓冲区做循环存取

17.1.2.1 PCM 接口

- 支持 16-bit 线性 PCM 编码的发送和接收
- PCM 接口帧同步信号可支持短脉冲(1 个时钟周期)和长脉冲(时钟周期数可配置)
- 接口时序支持标准模式和左对齐模式

17.1.2.2 I2S 接口

- 支持 16/24-bit 数据的发送和接收
- 支持 8kHz ~ 192kHz 采样率
- I2S 接口帧同步信号可支持低电平左声道或高电平左声道
- 接口时序支持标准模式和左对齐模式

17.1.3 功能描述

AIAO 子系统透过内部 PINMUX 将内置 Audio codec、I2S 管脚和 TX/RX 模块做连接，通过软件依应用适当配置寄存器，达到不同的连接方式。

17.1.3.1 典型应用

典型应用如下：

- 支持 I2S 从模式对接内置 Audio Codec ADC，或者 I2S/PCM/TDM 主/从模式对接外部 ADC 作音频采集。
- 支持 I2S 主模式对接内置 Audio Codec DAC，或者 I2S/PCM/TDM 主/从模式对接外部 DAC 作音频播放。

图表 17-2 通过 I2S 接口与内置 Audio Codec 连接示意图

图表 17-3 AIAO 以主模式通过 I2S 接口与外置 Audio Codec 连接示意图

图表 17-4 AIAO 以从模式通过 I2S 接口与外置 Audio Codec 连接示意图

17.1.3.2 功能原理

音源经由内置或外置 Audio Codec ADC 进行 Analog-to-Digital 转换成音频数据，通过 I2S 或 PCM 接口由对接的 RX 模块接收，并经由 DMA 存入循环缓冲区，再由 CPU 取走做存储，从而完成录音功能。TX 模块则经由 DMA 从循环缓冲区读取音频数据，通过 I2S 或 PCM 接口将音频数据传送给对接的内置或外置 Audio Codec DAC，进行 Digital-to-Analog 转换进行音源播放。

对接外部 I2S 接口时，支持的 I2S 时序如图表 17-5

(*fs_offset* = 1, *fs_idef* = 1, *fs_polarity* = 0, *slot_num* = 1, *slot_size* = 31, *data_size* = 23, *fb_offset* = 0, *slot_en* = 16'h3)
(*tx_sample_edge* = 0, *rx_sample_edge* = 1)

图表 17-5 I2S 接口时序

图表 17-5 以音频数据宽度 24-bit 为例，数据采用 MSB First 方式传输，MSB 相对 LRCK 信号延迟一个 BCLK 周期，数据和 LRCK 信号使用 BCLK 的下降缘发出，在 BCLK 的上升缘采样(tx_sample_edge = 0, rx_sample_edge = 1)。

对接外部 PCM 接口时，支持 PCM 标准时序和数据左对齐时序，标准模式时序如图表 17-6，左对齐模式时序如图表 17-7。

图表 17-6 PCM 接口标准模式时序

图表 11-6 以立体声音频数据宽度 16-bit 为例，数据采用 MSB First 方式传输，MSB 相对 LRCK 信号延迟一个 BCLK 周期，数据和 LRCK 信号使用 BCLK 的下降缘发出，在 BCLK 的上升缘采样(tx_sample_edge = 0, rx_sample_edge = 1)。

图表 17-7 PCM 接口左对齐模式时序

左对齐模式时，数据和 LRCK 信号在同拍开始发出。

17.1.4 工作方式

AIAO 内部集成的 Audio Codec 与外部通过 I2S 接口对接的 Audio Codec 能通过软件配置同时工作，连接方式如图 11-2 ~ 11-4。软件依应用在使能数据传输前，需优先配置 AIAO 子系统寄存器 i2s_tdm_sclk_in_sel, i2s_tdm_fs_in_sel, i2s_tdm_sdi_in_sel, i2s_tdm_sdo_out_sel，以连接各接口与对应操作的 TX/RX 模块 (I2S_TDM_0~I2S_TDM_3)。

17.1.4.1 时钟门控及时钟配置

若 AIAO 操作在主模式，必须先将做为 MCLK/BCLK 时钟来源的 TX/RX 模块寄存器 maste_mode 设为 1，视采样率配置分频寄存器 I2S_CLK_CTRL1(mclk_div, bclk_div)，再将寄存器 I2S_CLK_CTRL0(aud_en)设为 1，打开时钟门控。

17.1.4.2 软复位

AIAO 集成的 4 个 TX/RX 模块皆有独立的软复位，当使能 TX/RX 模块进行数据传输前，必须先配置寄存器 FIFO_RESET 和 I2S_RESET 进行软复位。

17.1.5 AIAO 寄存器概览

AIAO 子系统寄存器概览如表格 17-1 所示。

表格 17-1 AIAO 子系统寄存器概览 (基址 0x0410_8000)

Name	Address Offset	Description
i2s_tdm_sclk_in_sel	0x000	选择从模式 TX/RX 模块 SCLK 来源
i2s_tdm_fs_in_sel	0x004	选择从模式 TX/RX 模块帧同步信号来源
i2s_tdm_sdi_in_sel	0x008	选择 RX 模块 SDI 信号来源
i2s_tdm_sdo_out_sel	0x00c	选择子系统 SDO 输出来源
i2s_bclk_oen_sel	0x030	BCLK IO 的输出控制
audio_pdm_ctrl	0x040	使能 PDM 模式
i2s_sys_int_en	0x060	使能 I2S 子系統中斷信號
i2s_sys_ints	0x064	I2S 子系統中斷信號狀態

I2S_TDM_0~I2S_TDM_3 模块寄存器概览如表格 17-2 所示。

*表格 17-2 I2S_TDM_0/1/2/3 寄存器概览 (基址 0x0410_0000 + n*0x10000)*

Name	Address Offset	Description
BLK_MODE_SETTING	0x000	TX/RX 模块操作控制
FRAME_SETTING	0x004	音频帧时序控制
SLOT_SETTING1	0x008	声道与数据控制
SLOT_SETTING2	0x00c	使能声道
DATA_FORMAT	0x010	设定储存数据格式
BLK_CFG	0x014	TX/RX 模块功能配置
I2S_ENABLE	0x018	使能 TX/RX 模块
I2S_RESET	0x01c	TX/RX 模块复位
I2S_INT_EN	0x020	使能中断
I2S_INT	0x024	中断状态
FIFO_THRESHOLD	0x028	FIFO threshold
I2S_LRCK_MASTER	0x02c	使能 LRCK 主模式持续输出模式
FIFO_RESET	0x030	FIFO 复位
RX_STATUS	0x040	RX 模块内部状态寄存器
TX_STATUS	0x048	TX 模块内部状态寄存器
I2S_CLK_CTRL0	0x060	时钟控制
I2S_CLK_CTRL1	0x064	主模式时钟分频
I2S_PCM_SYNTH	0x068	PCM 帧同步信号合成模式控制
RX_RD_PORT	0x080	RX FIFO read port
TX_WR_PORT	0x0c0	TX FIFO write port

17.1.6 AIAO 寄存器描述

17.1.6.1 AIAO 子系统寄存器描述

i2s_tdm_sclk_in_sel

Select sclk source

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sclk_in_sel	R/W	当 i2s_tdm_0 操作在从模式时，选择 sclk 输入来源 3'b000 = reserved 3'b001 = 来自 i2s_tdm_1 bclk_out 3'b010 = 来自 i2s_tdm_2 bclk_out 3'b011 = 来自 i2s_tdm_3 bclk_out 3'b100 = 来自内置 Audio Codec ADC BCLK 3'b101 = 来自 IO I2S1_BCLK 3'b110 = 来自 IO I2S2_BCLK	0x4

Bits	Name	Access	Description	Reset
			3'b111 = reserved	
3	Reserved			
6:4	i2s_tdm_1_sclk_in_sel	R/W	当 i2s_tdm_1 操作在从模式时，选择 sclk 输入来源 3'b000 = 来自 i2s_tdm_0 bclk_out 3'b001 = 来自 audio_pdm 模块 sclk_out 3'b010 = 来自 i2s_tdm_2 bclk_out 3'b011 = 来自 i2s_tdm_3 bclk_out 3'b100 = 来自内置 Audio Codec ADC BCLK 3'b101 = 来自 IO I2S1_BCLK 3'b110 = 来自 IO I2S2_BCLK 3'b111 = reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_sclk_in_sel	R/W	当 i2s_tdm_2 操作在从模式时，选择 sclk 输入来源 3'b000 = 来自 i2s_tdm_0 bclk_out 3'b001 = 来自 i2s_tdm_1 bclk_out 3'b010 = reserved 3'b011 = 来自 i2s_tdm_3 bclk_out 3'b100 = 来自内置 Audio Codec ADC BCLK 3'b101 = 来自 IO I2S1_BCLK 3'b110 = 来自 IO I2S2_BCLK 3'b111 = reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sclk_in_sel	R/W	当 i2s_tdm_3 操作在从模式时，选择 sclk 输入来源 3'b000 = 来自 i2s_tdm_0 bclk_out 3'b001 = 来自 i2s_tdm_1 bclk_out 3'b010 = 来自 i2s_tdm_2 bclk_out 3'b011 = reserved 3'b100 = 来自内置 Audio Codec ADC BCLK 3'b101 = 来自 IO I2S1_BCLK 3'b110 = 来自 IO I2S2_BCLK 3'b111 = reserved	0x7
31:15	Reserved			

i2s_tdm_fs_in_sel

Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_fs_in_sel	R/W	当 i2s_tdm_0 操作在从模式时，选择 LRCK/FS 输入来源 3'b000 = reserved 3'b001 = 来自 i2s_tdm_1 lrck_out 3'b010 = 来自 i2s_tdm_2 lrck_out 3'b011 = 来自 i2s_tdm_3 lrck_out 3'b100 = 来自内置 Audio Codec ADC LRCK 3'b101 = 来自 IO I2S1_LRCK 3'b110 = 来自 IO I2S2_LRCK 3'b111 = reserved	0x4

Bits	Name	Access	Description	Reset
3	Reserved			
6:4	i2s_tdm_1_fs_in_sel	R/W	当 i2s_tdm_1 操作在从模式时，选择 LRCK/FS 输入来源 3'b000 = 来自 i2s_tdm_0 lrck_out 3'b001 = 来自 audio_pdm 模块 lrck_out 3'b010 = 来自 i2s_tdm_2 lrck_out 3'b011 = 来自 i2s_tdm_3 lrck_out 3'b100 = 来自内置 Audio Codec ADC LRCK 3'b101 = 来自 IO I2S1_LRCK 3'b110 = 来自 IO I2S2_LRCK 3'b111 = reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_fs_in_sel	R/W	当 i2s_tdm_2 操作在从模式时，选择 LRCK/FS 输入来源 3'b000 = 来自 i2s_tdm_0 lrck_out 3'b001 = 来自 i2s_tdm_1 lrck_out 3'b010 = reserved 3'b011 = 来自 i2s_tdm_3 lrck_out 3'b100 = 来自内置 Audio Codec ADC LRCK 3'b101 = 来自 IO I2S1_LRCK 3'b110 = 来自 IO I2S2_LRCK 3'b111 = reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_fs_in_sel	R/W	当 i2s_tdm_3 操作在从模式时，选择 LRCK/FS 输入来源 3'b000 = 来自 i2s_tdm_0 lrck_out 3'b001 = 来自 i2s_tdm_1 lrck_out 3'b010 = 来自 i2s_tdm_2 lrck_out 3'b011 = reserved 3'b100 = 来自内置 Audio Codec ADC LRCK 3'b101 = 来自 IO I2S1_LRCK 3'b110 = 来自 IO I2S2_LRCK 3'b111 = reserved	0x7
31:15	Reserved			

i2s_tdm_sdi_in_sel

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdi_in_sel	R/W	选择 i2s_tdm_0 RX 模块 SDI 信号来源 3'b000 = reserved 3'b001 = 来自 i2s_tdm_1 sdo 3'b010 = 来自 i2s_tdm_2 sdo 3'b011 = 来自 i2s_tdm_3 sdo 3'b100 = 来自内置 Audio Codec ADC SDO 3'b101 = 来自 IO I2S1_SDI 3'b110 = 来自 IO I2S2_SDI 3'b111 = reserved	0x4
3	Reserved			
6:4	i2s_tdm_1_sdi_in_sel	R/W	选择 i2s_tdm_1 RX 模块 SDI 信号来源	0x5

Bits	Name	Access	Description	Reset
			3'b000 = 来自 i2s_tdm_0 sdo 3'b001 = 来自 audio_pdm 模块 sdo 3'b010 = 来自 i2s_tdm_2 sdo 3'b011 = 来自 i2s_tdm_3 sdo 3'b100 = 来自内置 Audio Codec ADC SDO 3'b101 = 来自 IO I2S1_SDI 3'b110 = 来自 IO I2S2_SDI 3'b111 = reserved	
7	Reserved			
10:8	i2s_tdm_2_sdi_in_sel	R/W	选择 i2s_tdm_2 RX 模块 SDI 信号来源 3'b000 = 来自 i2s_tdm_0 sdo 3'b001 = 来自 i2s_tdm_1 sdo 3'b010 = reserved 3'b011 = 来自 i2s_tdm_3 sdo 3'b100 = 来自内置 Audio Codec ADC SDO 3'b101 = 来自 IO I2S1_SDI 3'b110 = 来自 IO I2S2_SDI 3'b111 = reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sdi_in_sel	R/W	选择 i2s_tdm_3 RX 模块 SDI 信号来源 3'b000 = 来自 i2s_tdm_0 sdo 3'b001 = 来自 i2s_tdm_1 sdo 3'b010 = 来自 i2s_tdm_2 sdo 3'b011 = reserved 3'b100 = 来自内置 Audio Codec ADC SDO 3'b101 = 来自 IO I2S1_SDI 3'b110 = 来自 IO I2S2_SDI 3'b111 = reserved	0x7
31:15	Reserved			

i2s_tdm_sdo_out_sel

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdo_out_sel	R/W	只予许配置为默认值 0x4	0x4
3	Reserved			
6:4	i2s_tdm_1_sdo_out_sel	R/W	选择 I2S1_SDO 信号来源 3'b100 = I2S1_SDO 来自 i2s_tdm_0 TX 模块 sdo 3'b101 = I2S1_SDO 来自 i2s_tdm_1 TX 模块 sdo 3'b110 = I2S1_SDO 来自 i2s_tdm_2 TX 模块 sdo 3'b111 = I2S1_SDO 来自 i2s_tdm_3 TX 模块 sdo 其他值 = reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_sdo_out_sel	R/W	选择 I2S2_SDO 信号来源 3'b100 = I2S2_SDO 来自 i2s_tdm_0 TX 模块 sdo 3'b101 = I2S2_SDO 来自 i2s_tdm_1 TX	0x6

Bits	Name	Access	Description	Reset
			模块 sdo 3'b110 = I2S2_SDO 来自 i2s_tdm_2 TX 模块 sdo 3'b111 = I2S2_SDO 来自 i2s_tdm_3 TX 模块 sdo 其他值 = reserved	
11	Reserved			
14:12	i2s_tdm_3_sdo_out_sel	R/W	只予许配置为默认值 0x7 内置 Audio Codec DAC 的 SDI 固定来自 i2s_tdm_3 TX 模块 sdo	0x7
31:15	Reserved			

i2s_bclk_oen_sel

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	i2s0_bclk_oen_sel	R/W	Reserved	0x0
1	i2s1_bclk_oen_sel	R/W	选择 I2S1_BCLK IO 的 oen 控制方式 0 = 由 i2s_tdm_1 TX 模块控制 1 = 统一由寄存器 i2s1_bclk_oen_ext 控制	0x0
2	i2s2_bclk_oen_sel	R/W	选择 I2S2_BCLK IO 的 oen 控制方式 0 = 由 i2s_tdm_2 TX 模块控制 1 = 统一由寄存器 i2s2_bclk_oen_ext 控制	0x0
3	i2s3_bclk_oen_sel	R/W	只予许配置为默认值 0x0	0x0
7:4	Reserved			
8	i2s0_bclk_oen_ext	R/W	Reserved	0x0
9	i2s1_bclk_oen_ext	R/W	由子系统寄存器控制 I2S1_BCLK IO 的 oen 0 = 关闭 IO 输出 1 = 使能 IO 输出	0x0
10	i2s2_bclk_oen_ext	R/W	由子系统寄存器控制 I2S2_BCLK IO 的 oen 0 = 关闭 IO 输出 1 = 使能 IO 输出	0x0
11	i2s3_bclk_oen_ext	R/W	只予许配置为默认值 0x0	0x0
15:12	Reserved			
16	i2s_bclk_oen_no_delay	R/W	只予许配置为默认值 0x0	0x0
31:17	Reserved			

audio_pdm_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	Reserved			
1	audio_pdm_sel_i2s1	R/W	使能 PDM 模式 0 = 一般工作模式 1 = I2S2 IO 操作于 PDM 模式，使用 i2s_tdm_1 RX 模块接收数据 当此值设 1 时，I2S2_BCLK IO 的 oen 固定由寄存器 i2s2_bclk_oen_ext 做控制	0x0
31:2	Reserved			

i2s_sys_int_en

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	i2s0_int_en	R/W	使能 I2S0 interrupt	0x1
1	i2s1_int_en	R/W	使能 I2S1 interrupt	0x1
2	i2s2_int_en	R/W	使能 I2S2 interrupt	0x1
3	i2s3_int_en	R/W	使能 I2S3 interrupt	0x1
7:4	Reserved			
8	i2s_subsys_int_en	R/W	使能 I2S_SUBSYS interrupt	0x1
31:9	Reserved			

i2s_sys_ints

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	i2s0_int	RO	I2S0 interrupt status 當發生 I2S0 中斷，可進一步讀取 I2S0 寄存器 I2S_INT 值，判斷中斷狀態	
1	i2s1_int	RO	I2S1 interrupt status 當發生 I2S1 中斷，可進一步讀取 I2S1 寄存器 I2S_INT 值，判斷中斷狀態	
2	i2s2_int	RO	I2S2 interrupt status 當發生 I2S2 中斷，可進一步讀取 I2S2 寄存器 I2S_INT 值，判斷中斷狀態	
3	i2s3_int	RO	I2S3 interrupt status 當發生 I2S3 中斷，可進一步讀取 I2S3 寄存器 I2S_INT 值，判斷中斷狀態	
31:4	Reserved			

17.1.6.2 I2S_TDM 模块寄存器描述

BLK_MODE_SETTING

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	tx_mode	R/W	选择传输模式 0 = 接收模式(RX), 1 = 发送模式(TX)	0x0
1	master_mode	R/W	选择主模式或从模式 0 = 从模式 1 = 主模式，BCLK 与 LRCK 由内部产生	0x0
2	rx_sample_edge	R/W	选择接收信号的 BCLK 时钟采样缘 0 = 下降沿采样 1 = 上升沿采样	0x1
3	tx_sample_edge	R/W	选择发送数据 SDO 的 BCLK 时钟缘 0 = 下降沿变化 1 = 上升沿变化	0x0

Bits	Name	Access	Description	Reset
6:4	Reserved			
7	dma_mode	R/W	数据搬移模式 0 = 软件模式, 1 = DMA 模式	0x1
8	Reserved	R/W	内部调试使用，只予许配置为默认值 0x0	0x0
9	Reserved			
10	slave_tx_fs_direct_in	R/W	选择用从模式发送数据时 LRCK 的来源 0 = LRCK 信号由 TX 模块内部产生 1 = LRCK 信号直接来自 master 此寄存器仅供调试使用。	0x0
11	Reserved			
12	pcm_synth_mode	R/W	主模式 PCM 帧同步信号使用频率合成模式 0 = PCM 帧同步信号周期由 frame_length 控制 1 = PCM 帧同步信号周期由 ck_coef_n/ck_coef_m 控制	0x0
15:13	Reserved			
31:16	reg_dummy	R/W	Dummy register	0xff00

FRAME_SETTING

Offset Address: 0x004

Bits	Name	Access	Description	Reset
8:0	frame_length	R/W	音频帧宽度(BCLK 时钟) 0~511 = 1~512 bits	0x1F
11:9	Reserved			
12	fs_polarity	R/W	LRCK 信号极性(polarity) 0 = 低电平启始(有效) 1 = 高电平启始(有效) I2S 模式，设 0 表示左声道优先，LRCK 输出低电平，设 1 表示左声道优先，LRCK 输出高电平。PCM 模式，设 1 表示帧同步脉冲信号为高电平。	0x0
13	fs_offset	R/W	帧音源数据 MSB 延迟 0 = 与 LRCK 信号同拍 1 = 较 LRCK 信号延迟一拍	0x1
14	fs_idef	R/W	选择 LRCK 信号模式 0 = 作为 PCM/TDM 帧同步 1 = 作为 I2S 左右声道	0x1
15	Reserved			
23:16	fs_active_length	R/W	LRCK 信号有效宽度 0~255 = 1~256 bits.	0x0F
31:24	Reserved			

SLOT_SETTING1

Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	slot_num	R/W	每帧音频内的声道数 0~15 = 1~16 声道 配置 1 表示两声道(左/右)。	0x1
7:4	Reserved			

Bits	Name	Access	Description	Reset
13:8	slot_size	R/W	声道宽度 0~63 = 1~64 bits	0x0F
15:14	Reserved			
20:16	data_size	R/W	声道内的有效数据宽度 0~31 = 1~32 bits 在一般情况(如 I2S 和 PCM 模式)，配置数据宽度和声道宽度相同值。	0x0F
23:21	Reserved			
28:24	fb_offset	R/W	声道内的数据 MSB 延迟 0 = 数据和声道时序同拍, 1~31 = 延迟 1~31 bits 此值控制各声道内第 1 个有效比特的时序。	0x0
31:29	Reserved			

SLOT_SETTING2

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	slot_en	R/W	使能有效的声道数据 配置 slot_en[n]为 1，代表第 n 个声道资料有效，配置 0 则为无效。在 TX 模式下，若某个声道设为无效，则该声道发送数据全为 0。在 RX 模式下，若某个声道设为无效，则该声道数据被接收后会以全 0 取代。	0x0003
31:16	Reserved			

DATA_FORMAT

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	data_format	R/W	只予许配置为默认值 0x0	0x0
2:1	word_length	R/W	选择音频数据资料内存宽度 2'b00 = 8-bit, 2'b01 = 16-bit, 2'b10 = 32-bit, 2'b11 = reserved 若收送数据有效比特数小于此值，则将数据向右对齐，MSB 补 0。	0x1
3	pad_slot_no	R/W	填充声道注记到 RX 数据中 0 = 无 1 = 将 4-bit 声道顺序填充到音频数据 MSB 中 当接收 24-bit 数据，可将比特 31~28 依序填入声道顺序再储存为 32-bit，提供软件后处理运用。	0x0
4	skip_rx_inactive_slot	R/W	在 RX 模式时忽略设为无效的声道数据 0 = 当接收被设为无效的声道数据时，会替换为全 0 储存 1 = 当接收被设为无效的声道数据时，会直接忽略不存入 RX FIFO 内。	0x0
5	skip_tx_inactive_slot	R/W	在 TX 模式时忽略设为无效的声道数据	0x0

Bits	Name	Access	Description	Reset
			0 = 当发送被设为无效的声道数据时，数据会从 TX FIFO 读出，但被替换为全 0 后发送 1 = 当发送被设为无效的声道数据时，不会从 TX FIFO 读出，而会直接发送全 0 比特。	
6	tx_source_left_align	R/W	TX 模式内存数据为左对齐 0 = 数据右对齐 1 = 数据左对齐 若内存宽度大于音源数据宽度，例如内存宽度为 32-bit，数据宽度为 24-bit，此值配置为 0 时，会取[23:0]发送，此值配置为 1 时，会取[31:8]发送。	0x0
31:7	Reserved			

BLK_CFG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	force_complete	R/W	强制结束模式 0 = 当 i2s_enable 由 1 设 0 后，直到一帧完整音频数据接收或发送完后才结束 1 = 当 i2s_enable 由 1 设 0 后，强制结束数据接收或发送 建议使用默认值 0x0。	0x0
1	dma_req_force_stop	R/W	dma_req 强制停止模式 0 = 当 i2s_enable 由 1 设 0 后，已发送的 dma_req 会保持高电平直到收到 DMA 返回 dma_ack 为止 set from 1 to 0 1 = 当 i2s_enable 由 1 设 0 后，取消已发送的 dma_req 建议使用默认值 0x1。	0x1
3:2	Reserved			
4	auto_disable_with_ch_en	R/W	I2S FIFO 传输自动停止 0 = 正常工作 1 = FIFO 操作与系统 DMA 连动 当 DMA channel 在 TX/RX 模块结束前(i2s_enable 设 0)已结束传输，则 TX/RX FIFO 会连动停止，在 RX 模式下，数据会继续接收但不会写入 FIFO，在 TX 模式下，会发送全 0 数据，不会再由 FIFO 读值。	0x0
5	Reserved			
6	rx_start_wait_dma_en	R/W	RX 模式等待 DMA 使能 0 = RX 模式在 i2s_enable 由 0 设 1 后开始运作 1 = RX 模式在 i2s_enable 由 0 设 1 后，会等待 DMA 被使能后才开始运作	0x0
7	Reserved			
8	rx_blk_clk_force_en	R/W	RX 模块时钟门控恒开	0x0

Bits	Name	Access	Description	Reset
			0 = 使能时钟门控自动关闭 1 = 时钟保持恒开	
9	rx_fifo_dma_clk_force_en	R/W	RX FIFO 控制模块时钟门控恒开 0 = 使能时钟门控自动关闭 1 = 时钟保持恒开	0x1
15:10	Reserved			
16	tx_blk_clk_force_en	R/W	TX 模块时钟门控恒开 0 = 使能时钟门控自动关闭 1 = 时钟保持恒开	0x0
17	tx_fifo_dma_clk_force_en	R/W	TX FIFO 控制模块时钟门控恒开 0 = 使能时钟门控自动关闭 1 = 时钟保持恒开	0x1
31:18	Reserved			

I2S_ENABLE

block enable

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	i2s_enable	R/W	使能 TX/RX 模块 0 = 关闭 TX/RX 1 = 使能 TX/RX 模块 在配置此值为 1 前，必须先配置 tx_fifo_reset/rx_fifo_reset 及 i2s_reset_tx/i2s_reset_rx。	0x0
31:1	Reserved			

I2S_RESET

sw reset

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	i2s_reset_rx	R/W	RX 模块软复位 0 = 无复位, 1 = 复位 对此值写 1 再写 0 以复位 RX 模块。由于跨时钟域，写 1 后必须等待一定时间再写 0，以让复位信号生效。	0x0
1	i2s_reset_tx	R/W	TX 模块软复位 0 = 无复位, 1 = 复位 对此值写 1 再写 0 以复位 TX 模块。由于跨时钟域，写 1 后必须等待一定时间再写 0，以让复位信号生效。	0x0
31:2	Reserved			

I2S_INT_EN

interrupt enable

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int_en	R/W	使能 RX FIFO data available interrupt	0x0
1	rx_fifo_overflow_int_en	R/W	使能 RX FIFO overflow interrupt	0x1
2	rx_fifo_underflow_int_en	R/W	使能 RX FIFO underflow interrupt	0x1
3	Reserved			
4	tx_fifo_avail_int_en	R/W	使能 TX FIFO data available interrupt	0x0
5	tx_fifo_overflow_int_en	R/W	使能 TX FIFO overflow interrupt	0x1

Bits	Name	Access	Description	Reset
6	tx_fifo_underflow_int_en	R/W	使能 TX FIFO underflow interrupt	0x1
7	Reserved			
8	i2s_int_en	R/W	使能 I2S IP interrupt 所有 I2S 中斷信號會合併為 1-bit, 並反應在 I2S 子系統寄存器 i2s_sys_ints 中	0x1
31:9	Reserved			

I2S_INT

interrupt status

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int	RO	RX FIFO 数据有效中断状态 当 RX FIFO 满度大于 rx_fifo_threshold 时，此值为 1。写 1 以清除中断状态。(rx_fifo_avail_int_en 需设 1)	
1	rx_fifo_overflow_int	RO	RX FIFO 溢出中断状态 写 1 以清除中断状态。 (rx_fifo_overflow_int_en 需设 1)	
2	rx_fifo_underflow_int	RO	RX FIFO 下溢中断状态 写 1 以清除中断状态。 (rx_fifo_underflow_int_en 需设 1)	
3	Reserved			
4	tx_fifo_avail_int	RO	TX FIFO 有空中断状态 当 TX FIFO 空度大于 tx_fifo_threshold 时，此值为 1。写 1 以清除中断状态。(tx_fifo_avail_int_en 需设 1)	
5	tx_fifo_overflow_int	RO	TX FIFO 溢出中断状态 写 1 以清除中断状态。 (tx_fifo_overflow_int_en 需设 1)	
6	tx_fifo_underflow_int	RO	TX FIFO 下溢中断状态 写 1 以清除中断状态。 (tx_fifo_underflow_int_en 需设 1)	
7	Reserved			
8	rx_fifo_avail_int_raw	RO	RX FIFO 数据有效中断状态 当 RX FIFO 满度大于 rx_fifo_threshold 时，此值为 1。写 1 以清除中断状态。(rx_fifo_avail_int_en 需设 1)	
9	rx_fifo_overflow_int_raw	RO	RX FIFO 溢出原始中断状态 写 1 以清除中断状态。	
10	rx_fifo_underflow_int_raw	RO	RX FIFO 下溢原始中断状态 写 1 以清除中断状态。	
11	Reserved			
12	tx_fifo_avail_int_raw	RO	TX FIFO 有空原始中断状态 当 TX FIFO 空度大于 tx_fifo_threshold 时，此值为 1。写 1 以清除中断状态。	
13	tx_fifo_overflow_int_raw	RO	TX FIFO 溢出原始中断状态 写 1 以清除中断状态。	
14	tx_fifo_underflow_int_raw	RO	TX FIFO 下溢原始中断状态 写 1 以清除中断状态。	
31:15	Reserved			

FIFO_THRESHOLD

Offset Address: 0x028

Bits	Name	Access	Description	Reset
4:0	rx_fifo_threshold	R/W	RX FIFO threshold 在 DMA 传输模式，当 RX FIFO 满度大于等于此值时，对 DMA 发出读请求信号。	0x7
15:5	Reserved			
20:16	tx_fifo_threshold	R/W	TX FIFO threshold 在 DMA 传输模式，当 TX FIFO 空度大于等于此值时，对 DMA 发出写请求信号。	0x7
23:21	Reserved			
28:24	tx_fifo_high_threshold	R/W	TX FIFO high threshold 在使能 TX 模块后，TX FIFO 会对 DMA 发出写请求，直到 FIFO 满度大于等于此值时，TX 模块才会开始对外发送数据。	0x1F
31:29	Reserved			

I2S_LRCK_MASTER

block enable

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	i2s_lrck_master_enable	R/W	使能 i2s_tdm 模块做为 BCLK/LRCK 主模式输出产生器 0 = 当操作在主模式(master_mode 设 1)，BCLK 和 LRCK 信号会在使能 TX/RX 后(i2s_enable 设 1)才对外输出 1 = 当操作在主模式，且 aud_en 设 1，将此值设 1 后 BCLK 和 LRCK 就会开始输出，此时 i2s_tdm 模块做为单纯的主模式 BCLK/LRCK 信号产生器。	0x0
31:1	Reserved			

FIFO_RESET

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	rx_fifo_reset	R/W	RX 通道 FIFO 软复位 写 1 后写 0 可对 RX FIFO 做软复位	0x0
15:1	Reserved			
16	tx_fifo_reset	R/W	TX 通道 FIFO 软复位 写 1 后写 0 可对 TX FIFO 做软复位	0x0
31:17	Reserved			

RX_STATUS

Offset Address: 0x040

Bits	Name	Access	Description	Reset
8:0	rx_frame_size_cnt	RO	RX 模块内部帧宽度计数器值	
9	rx_i2s_disable_req	RO	内部信号	
10	i2s_rx_start_wait	RO	内部信号	
16:11	rx_data_size_cnt	RO	RX 模块内部有效数据宽度计数器值	

Bits	Name	Access	Description	Reset
22:17	rx_slot_size_cnt	RO	RX 模块内部声道宽度计数器值	
23	i2s_reset_rx_sclk	RO	RX 模块时钟域的 i2s_reset_rx 信号状态	
28:24	rx_slot_num_cnt	RO	RX 模块内部声道数计数器值	
29	receive_start_sclk	RO	RX 模块时钟域的 i2s_enable 信号状态	
30	rx_blk_active	RO	RX 模块工作中	
31	rx_dma_req	RO	RX 模块发出 dma_req	

TX_STATUS

Offset Address: 0x048

Bits	Name	Access	Description	Reset
8:0	tx_frame_size_cnt	RO	TX 模块内部帧宽度计数器值	
9	tx_i2s_disable_req	RO	内部信号	
10	i2s_tx_start_wait	RO	内部信号	
16:11	tx_data_size_cnt	RO	TX 模块内部有效数据宽度计数器值	
22:17	tx_slot_size_cnt	RO	TX 模块内部声道宽度计数器值	
23	i2s_reset_tx_sclk	RO	TX 模块时钟域的 i2s_reset_tx 信号状态	
28:24	tx_slot_num_cnt	RO	TX 模块内部声道数计数器值	
29	transmit_start_sclk	RO	TX 模块时钟域的 i2s_enable 信号状态	
30	tx_blk_active	RO	TX 模块工作中	
31	tx_dma_req	RO	TX 模块发出 dma_req	

I2S_CLK_CTRL0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	aud_clk_sel	R/W	选择音源时钟来源(aud_clk) 0 = 来自内部 audio PLL 1 = 来自外部的 mclk_in	0x0
1	Reserved			
2	mclk_out_inv	R/W	主模式 mclk_out 时钟信号对外接口取反 0 = mclk_out 无取反 1 = mclk_out 取反	0x0
3	bclk_out_inv	R/W	主模式 bclk_out 时钟信号对外接口取反 0 = bclk_out 无取反 1 = bclk_out 取反	0x0
4	bclk_in_inv	R/W	从模式 bclk_in 时钟信号取反 0 = bclk_in 无取反 1 = bclk_in 取反	0x0
5	Reserved			
6	bclk_out_clk_force_en	R/W	主模式 bclk_out 时钟恒输出 1 = 恒输出 0 = 当 TX/RX 模式使能，且数据开始传输时才输出	0x1
7	mclk_out_en	R/W	使能 mclk_out IO 输出 0 = disable 1 = enable	0x0

Bits	Name	Access	Description	Reset
8	aud_en	R/W	使能时钟产生器和主模式时钟输出 0 = 关闭时钟产生器 1 = 使能时钟产生器并输出时钟信号 当在从模式时，请配置此值为 0。当在主模式时，在配置此值为 1 之前，必须先依数据采样率配置分频器 mclk_div/bclk_div。	0x0
31:9	Reserved			

I2S_CLK_CTRL1

Offset Address: 0x064

Bits	Name	Access	Description	Reset
15:0	mclk_div	R/W	mclk 时钟自音源时钟分频 1 = 除 1, 2 = 除 2, 3 = 除 3, …	0x1
31:16	bclk_div	R/W	bclk 时钟自 mclk 时钟分频 (仅主模式有效) 1 = 除 1, 2 = 除 2, 3 = 除 3, …	0x2

I2S_PCM_SYNTH

Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	ck_coef_n	R/W	PCM 帧同步信号周期分频系数 N 当 pcm_synth_mode 设 1 时，帧同步信号周期 = BCLK * (N/M)，N 值必须小于 M 值。	0x1
15:12	Reserved			
31:16	ck_coef_m	R/W	PCM 帧同步信号周期分频系数 M 当 pcm_synth_mode 设 1 时，帧同步信号周期 = BCLK * (N/M)，N 值必须小于 M 值。	0x40

RX_RD_PORT

Offset Address: 0x080

Bits	Name	Access	Description	Reset
31:0	rx_rd_port	RO	RX FIFO 读地址 DMA 需配置此地址做为接收数据时的读地址。	

TX_WR_PORT

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	tx_wr_port	WO	TX FIFO 写地址 DMA 需配置此地址做为发送数据时的写地址。	0x0

17.2 Audio Codec

17.2.1 概述

芯片集成高性能的 Audio Codec，包括立体声回放 DAC（90dB DR A-Weighted），支持两路单端 lineout 输出；立体声录音 ADC（90dB DR A-Weighted），支持立体声单端输入。

17.2.2 特点

- 90dB DR A-Weighted 立体声 DAC
- 立体声单端 Lineout 输出
- DAC 数字音量控制范围：-24dB～0dB
- DAC 支持 8kHz~48kHz 采样率
- 90dB DR A-Weighted 立体声 ADC
- ADC PGA 增益范围: 0db ~48dB
- ADC 支持 8kHz~48kHz 采样率
- ADC 支持 Mic in 立体声单端输入或 Line in 立体声单端输入

17.2.3 Audio Codec 寄存器概览

17.2.3.1 Audio DAC/ADC 寄存器概览

Name	Address Offset	Description
txdac_ctrl0	0x000	
txdac_ctrl1	0x004	
txdac_status	0x008	
txdac_afe0	0x00c	
txdac_afe1	0x010	
txdac_ana0	0x020	
txdac_ana1	0x024	
rxadc_ctrl0	0x100	
rxadcc_ctrl1	0x104	
rxadc_status	0x108	
rxadc_ana0	0x110	
rxadc_ana1	0x114	
rxadc_ana2	0x118	
rxadc_ana3	0x11c	
rxadc_ana4	0x120	

17.2.3.2 Audio ADC 寄存器描述

rxadc_ctrl0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_rxadc_en	R/W		0x0
1	reg_i2s_tx_en	R/W		0x0
31:2	Reserved			

rxadcc_ctrl1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
1:0	reg_rxadc_cic_opt	R/W	CIC decimation filter option 0: downsample ratio 64 1: downsample ratio 128 2: downsample ratio 256 3: downsample ratio 512	0x0
2	reg_rxadc_chn_swap	R/W	L/R input data channel swap	0x0
3	reg_rxadc_single	R/W	Only single channel supported, used when ANALOG in differential input mode	0x0
6:4	reg_rxadc_dcb_opt	R/W	DC blocking filter option 3'b000: bypass 3'b001: 1-2^(-8) 3'b010: 1-2^(-9) 3'b011: 1-2^(-10) 3'b100: 1-2^(-11) 3'b101: 1-2^(-12) other: bypass	0x5
7	Reserved			
8	reg_rxadc_igr_init	R/W	i2s keep silence when filter with initial value	0x0
9	reg_rxadc_clk_force_en	R/W	force clock enable	0x0
10	reg_rxadc_fsm_src_sel	R/W	FSM trigger source 0: adc0_vld 1: adc1_vld	0x0
31:11	Reserved			

rxadc_status

Offset Address: 0x108

Bits	Name	Access	Description	Reset
0	reg_rxadc_cic0_init_done	RO	cic_d_0 init done	
1	reg_rxadc_fir1_0_init_done	RO	fir1_0 init done	
2	reg_rxadc_fir2_0_init_done	RO	fir2_0 init done	
3	Reserved			
4	reg_rxadc_cic1_init_done	RO	cic_d_1 init done	
5	reg_rxadc_fir1_1_init_done	RO	fir1_1 init done	
6	reg_rxadc_fir2_1_init_done	RO	fir2_1 init done	
7	Reserved			
10:8	reg_rxadc_fsm	RO	ADC FSM state	
31:11	Reserved			

rxadc_ana0

Offset Address: 0x110

Bits	Name	Access	Description	Reset
12:0	reg_gstepl_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
13	reg_g6dbl_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
15:14	reg_gainl_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0
28:16	reg_gstepr_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
29	reg_g6dbr_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
31:30	reg_gainr_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0

rxadc_ana1

Offset Address: 0x114

Bits	Name	Access	Description	Reset
15:0	reg_gainl_status	RO	[12:0]: gstep1 [13]: g6dbl [15:14]: gainl	
31:16	reg_gainr_status	RO	[28:16]: gstepr [29]: g6dbr [31:30]: gainr	

rxadc_ana2

Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	reg_mutel_rxpga	R/W	Enable pin of mute left channel, active high	0x0
1	reg_muter_rxpga	R/W	Enable pin of mute right channel, active high	0x0
15:2	Reserved			

Bits	Name	Access	Description	Reset
16	reg_diff_en_rxpga	R/W	Enable pin of differential mode(Left channel only, VIN/VINB=PAD_VINL/PAD_VINR	0x0
17	reg_tristate_rxpga	R/W	?	0x0
31:18	Reserved			

rxadc_ana3

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
0	reg_addi_rxadc	R/W	ADC opamp current +50%, active High	0x0
1	reg_cksel_rxadc	R/W	PGA enable control input, active High	0x0
2	reg_en_asar_i_rxadc	R/W	Enable pin of L channel quantizer	0x1
3	reg_en_asar_q_rxadc	R/W	Enable pin of Q channel quantizer	0x1
5:4	reg_dem_type_rxadc	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: NA	0x1
7:6	Reserved			
11:8	reg_ctune_rxadc	R/W	RXADC integrator CFB selection 12*135fF + (8/4/2/1)*135fF	0xc
12	reg_en_dither_rxadc	R/W	Enable pin of dithering	0x1
13	reg_rstsdm_rxadc	R/W	Enable pin of resetting integrator	0x0
14	reg_en_vcmt_rxadc	R/W	?	0x0
15	Reserved			
17:16	reg_vldo0p9_rxadc	R/W	0.9V LDO output selection 00:0.85V 01:0.9V 10:0.95V 11:1.0V	0x1
19:18	reg_vldo12_rxadc	R/W	1.2V LDO output selection 00:1.1V 01:1.15V 10:1.2V 11:1.25V	0x1
21:20	reg_rnlvl_rxadc	R/W	Dither option	0x0
31:22	Reserved			

rxadc_ana4

Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	reg_da_en_rxpga_status	RO	DA_EN_RXPGA status	
1	reg_da_end2us_rxpga_status	RO	DA_END2US_RXPGA status	
2	reg_da_en_rxadc_status	RO	DA_EN_RXADC status	
3	reg_da_en_audbias_status	RO	DA_EN_AUDBIAS status	
15:4	Reserved			
18:16	reg_ad_dol_rxadc	RO	Left channel 3-bits output	
19	Reserved			
22:20	reg_ad_dor_rxadc	RO	Right channel 3-bits output	
31:23	Reserved			

17.2.3.3 Audio DAC 寄存器描述

txdac_ctrl0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_txdac_en	R/W	audio dac enable	0x0
1	reg_i2s_rx_en	R/W	audio dac i2s output enable	0x0
31:2	Reserved			

txdac_ctrl1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_txdac_cic_opt	R/W	CIC decimation filter option 0: upsample ratio 64 1: upsample ratio 128 2: upsample ratio 256 3: upsample ratio 512	0x0
3:2	Reserved			
5:4	reg_txdac_dem_type	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: thermal code	0x1
7:6	Reserved			
8	reg_txdac_dsm_opt	R/W	DSM order option 0: order2 1: order1	0x0
9	reg_txdac_zcd_en	R/W	enable zero corssing function move from reg_spare0	0x0
10	reg_txdac_fsm_src_sel	R/W	FSM trigger source 0: i2s1_vld 1: i2s0_vld	0x1
11	Reserved			
14:12	reg_txdac_dither_opt	R/W	Dither option 0: disable others: weight = LSB/(2^(n-1))	0x0
31:15	Reserved			

txdac_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_txdac_fsm	RO	DAC main fsm state	
3	Reserved			
6:4	reg_txdac_afe_fsm	RO	DAC AFE fsm state	
7	Reserved			
16:8	reg_txdac_gain0	RO	DAC L-channel gain	
19:17	Reserved			
28:20	reg_txdac_gain1	RO	DAC R-channel gain	
31:29	Reserved			

txdac_afe0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
5:0	reg_txdac_init_dly_cnt	R/W	DAC AFE initialize delay, min>2us	0x28
7:6	Reserved			
15:8	reg_txdac_value_tick	R/W	DAC AFE tick value for initialize/stop value ramp	0x0
27:16	reg_txdac_gain_tick	R/W	DAC AFE tick value for gain ramp	0x800
31:28	Reserved			

txdac_afe1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
8:0	reg_txdac_gain_ub_0	R/W	channel 0, L-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
15:9	Reserved			
24:16	reg_txdac_gain_ub_1	R/W	channel 1, R-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
27:25	Reserved			
28	reg_txdac_ramp_bp	R/W	bypass initial ramp procedure	0x0
31:29	Reserved			

txdac_ana0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_addi_txdac	R/W	na	0x0
3:1	Reserved			
5:4	reg_tsel_txdac	R/W	2'b00: NA 2'b01: VCM 2'b10: VDD15A	0x0

Bits	Name	Access	Description	Reset
			2'b11: undefined	
7:6	Reserved			
9:8	reg_vsel_txdac	R/W	1.5V LDO output selection 2'b00:1.35V 2'b01:1.4V 2'b10:1.45V 2'b11:1.5V	0x3
31:10	Reserved			

txdac_ana1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_da_en_txdac_ow_val	R/W	DA_EN_TXDAC overwrite value	0x0
1	reg_da_end2us_txdac_ow_val	R/W	DA_END2US_TXDAC overwrite value	0x0
15:2	Reserved			
16	reg_da_en_txdac_ow_en	R/W	DA_EN_TXDAC overwrite enable	0x0
17	reg_da_end2us_txdac_ow_en	R/W	DA_END2US_TXDAC overwrite enable	0x0
31:18	Reserved			

18 外围设备

18.1 I2C

18.1.1 概述

本芯片配置 6 个 I2C 控制器，可个别配置为 Master/Slave。IO 配置需参考 Function pin mux 做配置。

18.1.2 功能描述

I2C 控制器具有以下功能特点：

- 支持标准地址（7bit）和扩展地址（10bit）。
- 传输速率支持标准模式（100kbit/s）和快速模式（400kbit/s）。
- 支持 General Call 和 Start Byte 功能。
- 不支持 CBUS 器件。
- 支持 DMA 操作。
- 支持 64 x 8bit 的 TX FIFO 和 64 x 8bit 的 RX FIFO。

18.1.3 功能框图

图表 12-1 为 I2C 模块之功能框图。IIC_CLK 为模块时钟，芯片支持可配 25MHz 或 100MHz。CPU 透过 APB 总线配置寄存器以选择 I2C 各模式与时序，写 TXFIFO、读 RXFIFO，并触发 FSM 以发送、接收 SDA/SCL 相关 IO 信号。System DMA 亦可搭配 I2C DMA_IF，并透过 APB 总线写 TXFIFO、读 RXFIFO，以发送、接收 I2C 信号。

图表 18-1 I2C 功能框图

18.1.4 I2C 协定时序

芯片 I2C 支持一般标准之 I2C 协定时序如图表 12-2 所示。

图表 18-2 I2C 协定时序

18.1.5 工作方式

18.1.5.1 配置 I2C 时钟与时序参数

1. CLK_DIV CRG 寄存器章节，配置 clk_byp_0_31 可以选择 IIC_CLK 为 25MHz 预设时钟源或 100MHz 时钟源。
2. 配置相关时序配置需要在模块不使能状态。需将 IC_ENABLE 设置为 0，并查询 IC_ENABLE_STATUS[0] 确定为 0。
3. 参照表格 18-1，按 I2C 时钟选择，配置 I2C 时序计数寄存器

表格 18-1 I2C 时钟选择与相关寄存器配置关系表

寄存器	25M IIC_CLK	100M IIC_CLK	描述
IC_SS_SCL_HCNT	115	460	标准速度模式之 SCL 高电平时间计数
IC_SS_SCL_LCNT	135	540	标准速度模式之 SCL 低电平时间计数
IC_FS_SCL_HCNT	21	90	快速速度模式之 SCL 高电平时间计数
IC_FS_SCL_LCNT	42	160	快速速度模式之 SCL 低电平时间计数
IC_SDA_HOLD	1	1	SDA 保持时间计数，相对 SCL 负缘
IC_SDA_SETUP	6	25	SDA 设定时间计数，相对 SCL 正缘

寄存器	25M IIC_CLK	100M IIC_CLK	描述
IC_FS_SPKLEN	2	5	I2C 毛刺抑制时间计数

18.1.5.2 非 DMA 方式下的数据传输

图表 18-3 I2C 非 DMA 方式下的数据传输软件流程

18.1.5.3 DMA 方式下的数据传输

图表 18-4 I2C DMA 方式下的数据传输软件流程

18.1.6 I2C 寄存器概览

表格 18-2 芯片的 6 组 I2C 模块基地址

GPIO 模块	基地址
I2C0	0x04000000

GPIO 模块	基地址
I2C1	0x04010000
I2C2	0x04020000
I2C3	0x04030000
I2C4	0x04040000
RTCSYS_I2C	0x0502B000

表格 18-3 芯片的 I2C 寄存器概览

Name	Address Offset	Description
IC_CON	0x000	I2C Control
IC_TAR	0x004	I2C Target Address
IC_SAR	0x008	I2C Slave Address
IC_DATA_CMD	0x010	I2C Rx/Tx Data Buffer and Command
IC_SS_SCL_HCNT	0x014	Standard speed I2C Clock SCL High Count
IC_SS_SCL_LCNT	0x018	Standard speed I2C Clock SCL Low Count
IC_FS_SCL_HCNT	0x01c	Fast speed I2C Clock SCL High Count
IC_FS_SCL_LCNT	0x020	Fast speed I2C Clock SCL Low Count
IC_INTR_STAT	0x02c	I2C Interrupt Status
IC_INTR_MASK	0x030	I2C Interrupt Mask
IC_RAW_INTR_STAT	0x034	I2C Raw Interrupt Status
IC_RX_TL	0x038	I2C Receive FIFO Threshold
IC_TX_TL	0x03c	I2C Transmit FIFO Threshold
IC_CLR_INTR	0x040	Clear Combined and Individual Interrupts
IC_CLR_RX_UNDER	0x044	Clear RX_UNDER Interrupt
IC_CLR_RX_OVER	0x048	Clear RX_OVER Interrupt
IC_CLR_TX_OVER	0x04c	Clear TX_OVER Interrupt
IC_CLR_RD_REQ	0x050	Clear RD_REQ Interrupt
IC_CLR_TX_ABRT	0x054	Clear TX_ABRT Interrupt
IC_CLR_RX_DONE	0x058	Clear RX_DONE Interrupt
IC_CLR_ACTIVITY	0x05c	Clear ACTIVITY Interrupt
IC_CLR_STOP_DET	0x060	Clear STOP_DET Interrupt
IC_CLR_START_DET	0x064	Clear START_DET Interrupt
IC_CLR_GEN_CALL	0x068	Clear GEN_CALL Interrupt
IC_ENABLE	0x06c	I2C Enable
IC_STATUS	0x070	I2C Status register
IC_TXFLR	0x074	Transmit FIFO Level Register
IC_RXFLR	0x078	Receive FIFO Level Register
IC_SDA_HOLD	0x07c	SDA hold time length register
IC_TX_ABRT_SOURCE	0x080	I2C Transmit Abort Status Register
IC_SLV_DATA_NACK_ONLY	0x084	Generate SLV_DATA_NACK Register
IC_DMA_CR	0x088	DMA Control Register for transmit and receive handshaking interface
IC_DMA_TDLR	0x08c	DMA Transmit Data Level
IC_DMA_RDLR	0x090	DMA Receive Data Level
IC_SDA_SETUP	0x094	I2C SDA Setup Register
IC_ACK_GENERAL_CALL	0x098	I2C ACK General Call Register
IC_ENABLE_STATUS	0x09c	I2C Enable Status Register
IC_FS_SPKLEN	0x0a0	ISS and FS spike suppression limit
IC_HS_SPKLEN	0x0a4	HS spike suppression limit

18.1.7 I2C 寄存器描述

IC_CON

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	MASTER_MODE	R/W	enable master mode	0x1
2:1	SPEED	R/W	1: standard mode (~100 kbit/s) 2: fast mode (~400 kbit/s)	0x3
3	IC_10BITADDR_SLAVE	R/W	enable 10bit slave address mode	0x1
4	IC_10BITADDR_MASTER	R/W	enable 10bit master address mode	0x1
5	IC_RESTART_EN	R/W	enable I2C master to be able generate restart	0x1
6	IC_SLAVE_DISABLE	R/W	0: slave is enabled 1: slave is disabled	0x1
31:7	Reserved			

IC_TAR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
9:0	IC_TAR	R/W	I2C Target Address Register	0x55
10	GC_OR_START	R/W	If bit SPECIAL is set to 1, then this bit indicates whether a General Call or START byte command 0: general call 1: start byte	0x0
11	SPECIAL	R/W	Used to issue General Call or START BYTE	0x0
31:12	Reserved			

IC_SAR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	IC_SAR	R/W	I2C Slave Address Register	0x55
31:10	Reserved			

IC_DATA_CMD

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	DAT	R/W	transmitted or received data port	0x0
8	CMD	R/W	0: Write 1: Read	0x0
9	STOP	R/W	issue stop	0x0
10	RESTART	R/W	issue restart	0x0
31:11	Reserved			

IC_SS_SCL_HCNT

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_HCNT	R/W	Standard Speed I2C Clock SCL High Count Register	0x0190
31:16	Reserved			

IC_SS_SCL_LCNT

Offset Address: 0x018

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_LCNT	R/W	Standard Speed I2C Clock SCL Low Count Register	0x01d6
31:16	Reserved			

IC_FS_SCL_HCNT

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_HCNT	R/W	Fast Speed I2C Clock SCL High Count Register	0x003C
31:16	Reserved			

IC_FS_SCL_LCNT

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_LCNT	R/W	Fast Speed I2C Clock SCL Low Count Register	0x0082
31:16	Reserved			

IC_INTR_STAT

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	R_RX_UNDER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
1	R_RX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
2	R_RX_FULL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
3	R_TX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
4	R_TX_EMPTY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
5	R_RD_REQ	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
6	R_TX_ABRT	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
7	R_RX_DONE	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
8	R_ACTIVITY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
9	R_STOP_DET	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
10	R_START_DET	RO	corresponding masked interrupt staus,	

Bits	Name	Access	Description	Reset
			please reference I2C Raw Interrupt Status	
11	R_GEN_CALL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
31:12	Reserved			

IC_INTR_MASK

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	M_RX_UNDER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
1	M_RX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
2	M_RX_FULL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
3	M_TX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
4	M_TX_EMPTY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
5	M_RD_REQ	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
6	M_TX_ABRT	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
7	M_RX_DONE	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
8	M_ACTIVITY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
9	M_STOP_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
10	M_START_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
11	M_GEN_CALL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
31:12	Reserved			

IC_RAW_INTR_STAT

Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	IST_RX_UNDER	RO	when receive buffer is empty by reading from the IC_DATA_CMD register	
1	IST_RX_OVER	RO	receive buffer is oveflow (64Bytes)	
2	IST_RX_FULL	RO	receive buffer reaches or goes above	

Bits	Name	Access	Description	Reset
			the RX_TL threshold	
3	IST_TX_OVER	RO	transmit buffer is oveflow (64Bytes)	
4	IST_TX_EMPTY	RO	transmit buffer is at or below the TX_TL threshold	
5	IST_RD_REQ	RO	In slave mode, I2C hold SCL and wait for the response from processor	
6	IST_TX_ABRT	RO	In master or slave mode, when transmitter is unable to complete the action	
7	IST_RX_DONE	RO	In slave-transmitter mode, a NACK is received	
8	IST_ACTIVITY	RO	I2C activity is detected	
9	IST_STOP_DET	RO	STOP occurred	
10	IST_START_DET	RO	START or RESTART occurred	
11	IST_GEN_CALL	RO	General Call address is received	
31:12	Reserved			

IC_RX_TL

Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	RX_TL	R/W	Receive FIFO Threshold Level	0x0
31:8	Reserved			

IC_TX_TL

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
7:0	TX_TL	R/W	Transmit FIFO Threshold Level	0x0
31:8	Reserved			

IC_CLR_INTR

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	CLR_INTR	RO	read to clear corresponding all raw staus	
31:1	Reserved			

IC_CLR_RX_UNDER

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	CLR_RX_UNDER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_RX_OVER

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	CLR_RX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_TX_OVER

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	CLR_TX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_RD_REQ

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	CLR_RD_REQ	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_TX_ABRT

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	CLR_TX_ABRT	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_RX_DONE

Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	CLR_RX_DONE	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_ACTIVITY

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
0	CLR_ACTIVITY	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_STOP_DET

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	CLR_STOP_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_START_DET

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	CLR_START_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw	

Bits	Name	Access	Description	Reset
			Interrupt Status	
31:1	Reserved			

IC_CLR_GEN_CALL

Offset Address: 0x068

Bits	Name	Access	Description	Reset
0	CLR_GEN_CALL	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_ENABLE

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	ENABLE	R/W	Enables I2C controller	0x0
31:1	Reserved			

IC_STATUS

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	ST_ACTIVITY	RO	I2C Activity Status.	
1	ST_TFNF	RO	Transmit FIFO Not Full	
2	ST_TFE	RO	Transmit FIFO Completely Empty	
3	ST_RFNE	RO	Receive FIFO Not Empty	
4	ST_RFF	RO	Receive FIFO Completely Full	
5	ST_MST_ACTIVITY	RO	Master FSM Activity Status	
6	ST_SLV_ACTIVITY	RO	Slave FSM Activity Status	
31:7	Reserved			

IC_TXFLR

Offset Address: 0x074

Bits	Name	Access	Description	Reset
6:0	TXFLR	RO	I2C Transmit FIFO Level	
31:7	Reserved			

IC_RXFLR

Offset Address: 0x078

Bits	Name	Access	Description	Reset
6:0	RXFLR	RO	I2C Receive FIFO Level Register	
31:7	Reserved			

IC_SDA_HOLD

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	IC_SDA_HOLD	R/W	Sets the required SDA hold time in units of IP clock.	0x1
31:16	Reserved			

IC_TX_ABRT_SOURCE

Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	TX_ABRT_SOURCE	RO	I2C Transmit Abort Source Register	
31:16	Reserved			

IC_SLV_DATA_NACK_ONLY

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	NACK	R/W	generate a NACK in slave-receiver mode	0x0
31:1	Reserved			

IC_DMA_CR

Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	RDMAE	R/W	Receive DMA Enable	0x0
1	TDMAE	R/W	Transmit DMA Enable	0x0
31:2	Reserved			

IC_DMA_TDLR

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
5:0	DMATDL	R/W	the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value	0x0
31:6	Reserved			

IC_DMA_RDLR

Offset Address: 0x090

Bits	Name	Access	Description	Reset
5:0	DMARDL	R/W	dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1	0x0
31:6	Reserved			

IC_SDA_SETUP

Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDA_SETUP	R/W	SDA Setup time config register	0x64
31:1	Reserved			

IC_ACK_GENERAL_CALL

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	ACK_GEN_CALL	R/W	When set to 1, DW_apb_i2c responds with a ACK when it receives a General Call. When set to 0, the IP does not generate General Call interrupts	0x1
31:1	Reserved			

IC_ENABLE_STATUS

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	IC_EN	RO	I2C Enable Status Register	
1	SLV_DISABLED_WHILE_BUSY	RO	Slave Disabled While Busy (Transmit, Receive)	
2	SLV_RX_DATA_LOST	RO	Slave Received Data Lost.	
31:3	Reserved			

IC_FS_SPKLEN

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	IC_FS_SPKLEN	R/W	I2C SS and FS Spike Suppression Limit Register	0x5
31:8	Reserved			

IC_HS_SPKLEN

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	IC_HS_SPKLEN	R/W	I2C HS Spike Suppression Limit Register	0x1
31:8	Reserved			

18.2 UART

18.2.1 概述

UART（Universal Asynchronous Receiver Transmitter）是一个非同步串行的通信介面，主要功能是将来自外围设备的资料进行串并转换之后传入内部总线，以及将资料进行并串转换之后输出到外部设备。UART 的主要功能是和外部芯片的 UART 进行对接，从而实现两芯片间的通信。

本芯片提供 5 个 UART 控制器，相关概述如下表，IO 相关配置请参照管脚信息描述章节。

控制器	支持模式	IO 管脚
UART0	两线 UART	UART0_TX/UART0_RX
UART1	两/四线 UART	UART1_TX/UART1_RX/UART1_CTS/UART1_RTS
		XGPIOA[20]/ XGPIOA[21]/ XGPIOA[22]/ XGPIOA[26]
UART2	两/四线 UART	UART2_TX/UART2_RX/UART2_CTS/UART2_RTS
		XGPIOA[20]/ XGPIOA[21]/ XGPIOA[22]/ XGPIOA[26]
		IIC2_SDA/IIC2_SCL

控制器	支持模式	IO 管脚
UART3	两/四线 UART	SPI0_CS_X/SPI0_SCK/SPI0_SDI/SPI0_SDO
		VI_DATA22/VI_DATA21/VI_DATA24/VI_DATA23
		PWM3/PWM2
UART4	两线 UART	XGPIOA[22]/ XGPIOA[26]
		UART1_RTS/UART1_CTS

18.2.2 特点

UART 模组有以下特点：

- 支援 64 x 8bit 的发送 FIFO 和 64 x 8bit 的接收 FIFO。
- 支援资料位元和停止位元的位宽可程式设计。资料位元可通过程式设计设定为 5/6/7/8 比特；
- 停止位可通过程式设计设定为 1bit,1.5bit 或 2bit。
- 支援奇、偶校验方式或者无校验。
- 支援传输速率可程式设计。
- 支援接收 FIFO 中断、发送 FIFO 中断、错误中断。
- 支援初始中断状态查询和遮罩后中断状态查询。
- 支援 DMA 操作。

18.2.3 功能描述

18.2.3.1 应用框图

UART 是一普遍通用之点对点物理层传输协定，可以用来对接各种系统，包含 PC、各种周边芯片，可作为芯片与芯片中间之沟通介面。

图表 18-5 UART 应用框图

18.2.3.2 功能原理

- 波特率(buad rate)

由于 UART 介面没有参考时钟，属于异步传输方式，需要双方采用相同传输速度亦即波特率(buad rate)以进行沟通。若有误差，误差率需够小使不致误传。1 比特之速率称为波特率(buad rate)。典型之波特率有 300, 1200, 2400, 9600, 19200, 38400, 115200 bps 等。

- 帧结构(frame structure)

UART 传输数据结构以帧为单位。帧结构包含起始信号、资料信号、校验位元和结束信号。

图表 18-6 UART 传输数据结构

- 起始信号（start bit）

起始信号为一帧开始的标志。发动一个帧传输的最开头，就是在 TXD 上发送一低电平信号位。在 RXD 上，若在闲置状态接收到一低电平之信号位则判断为接收到一个侦传输之起始。

- 资料信号（data bit）

资料位元宽可以根据不同的应用要求进行调整，可为 5/6/7/8 比特资料位元宽。典型为 8 比特资料位宽。

- 校验位元（parity bit）

校验位元是 1 比特纠错信号，UART 的校验位元有奇数同位检查、偶校验和固定校验位，同时支持校验位的使能和禁止，详细描述请见 LCR 寄存器。

- 结束信号（stop bit）

结束信号即帧的停止位元，支援 1 比特、1.5 比特和 2 比特停止位元。发送一帧的结束信号就是把 TXD 发送高电平完成传输，进入闲置状态。接收一帧在计数到校验位元之后，需接收到结束信号。

18.2.4 工作方式

18.2.4.1 波特率配置

- UART 工作时钟(UART_SCLK)配置

可参考 CLK_DIV CRG 寄存器描述，配置 clk_sel_0_9~ clk_sel_0_13 选择 uart0~uart4 的工作时钟。预设为 1:XTAL 25MHz，配置为 0 即选择 UART PLL 分频时钟源。PLL 分频时钟

源预设为 187.5MHz ，若有需要可以配置分频寄存器 div_clk_187p5m 调整 PLL 分频时钟为 1500/N MHz，最高达 187.5MHz。

- UART 波特率配置

DLL、DLH 为 UART 控制器内部之波特率分频控制寄存器，DLH 为高 8 位、DLL 为低 8 位。配置 DLH、DLL 前须先配置 LCR[7]为 1。此时可配置寄存器 RBR_THR_DLL(DLL)，IER_DLH(DLH)。

配置完成后，波特率即设定完成，公式为：

$$Baud\ rate = {UART_SCLK}/{16 * (256 * DLH + DLL)}$$

- 以 UART SCLK 25MHz 为例，配置 115200 波特率，算式为：

$$(256 * DLH + DLL) = {25M}/{16 * 115200} = 13.5$$

若选择配置 DLL 为 14、DLH 为 0，则实际波特率为：

$$Baud\ rate = {25M}/{16 * 14} = 111607$$

一比特时间误差为：

$$Bit\ Error = {(115200 - 114286)}/{115200} = 3.12\%$$

累积一帧时间误差为：

$$Frame\ Error = 3.12\% * 10 = 31.2\%$$

- 以 UART SCLK 187.5MHz 为例，配置 115200 波特率，算式为：

$$(256 * DLH + DLL) = {187.5M}/{16 * 115200} = 101.7$$

若选择配置 DLL 为 102、DLH 为 0，则实际波特率为：

$$Baud\ rate = {187.5M}/{16 * 102} = 114890$$

一比特时间误差为：

$$Bit\ Error = {(115200 - 114890)}/{115200} = 0.27\%$$

累积一帧时间误差为：

$$Frame\ Error = 3.12\% * 10 = 2.7\%$$

18.2.4.2 数据发送流程图

图表 18-7 UART 数据发送流程图

18.2.4.3 数据接收流程图

图表 18-8 UART 数据接收流程图

18.2.4.4 中断或查询方式下的数据传输

18.2.4.4.1 初始化步骤

1. 向 LCR[7]写 1。使能配置 Divisor Latch Access

2. 写相应的配置值到 RBR_THR_DLL、IER_DLH 寄存器，配置传输波特率。
3. 向 LCR[7]写 0。
4. 配置 LCR，设定相应的 UART 工作模式
5. 配置 FCR，设定相应的发送及接收 FIFO 阈值。
6. 若使用中断方式，则需设定 IER，使能相应中断信号；

18.2.4.4.2 数据发送

1. 在 LCR[7]为 0 状态下，将发送数据写入 RBR_THR_DLL(Transmit Holding Register)，启动数据发送。
2. 若使用查询方式，则通过读取 USR[1](Transmit FIFO not full) 与 TFL (Transmit FIFO Level) 检测 TX_FIFO 状态，根据 TX_FIFO 的状态决定是否继续向 RBR_THR_DLL 中写入数据；
3. 若使用中断方式，则根据相应中断状态位检测；决定是否继续向 RBR_THR_DLL 中写入数据。
4. 通过检测 USR[2](Transmit FIFO Empty)，判断 UART 是否完成全部数据发送。

18.2.4.4.3 数据接收

1. 若使用查询方式，则通过读取 USR[3](Receive FIFO Not Empty) 与 RFL(Receive FIFO Level) 检测 RX_FIFO 状态，根据 RX_FIFO 的状态决定是否读取 RBR_THR_DLL(Receive Buffer Register)，取得数据。
2. 若使用中断方式，则根据相应中断状态位检测决定是否读取 RBR_THR_DLL(Receive Buffer Register)，取得数据。

18.2.4.5 DMA 方式下的数据传输

18.2.4.5.1 初始化步骤

1. 向 LCR[7]写 1。使能配置 Divisor Latch Access
2. 写相应的配置值到 RBR_THR_DLL、IER_DLH 寄存器，配置传输波特率。
3. 向 LCR[7]写 0。
4. 配置 LCR，设定相应的 UART 工作模式
5. 配置 FCR，设定相应的发送及接收 FIFO 阈值。
6. 关闭 IER 中的 ETBEI/ERBFI；

18.2.4.5.2 数据发送

1. 配置系统 DMA 通道映射。参照系统 DMA 通道映射，将选取之 UART 控制器 TX/RX 请求线号配置到对应的系统 DMA 通道。例如： UART0 TX 配置系统 DMA 通道 3 则 sdma_dma_ch_remap0[29:24]=9 。配置完成需配置 update_dma_remp_0_3 使配置生效。
2. 配置系统 DMA 数据通道，包括数据传输源和目的地址、数据传输个数、传输类型等参数。具体配置时请参见 DMA 控制器章节。
3. 通过系统 DMA 中断上报，判断数据是否发送完成。

18.2.4.5.3 数据接收

1. 配置系统 DMA 通道映射。参照 DMA 通道映射，将选取之 UART 控制器 TX/RX 请求线号配置到对应的系统 DMA 通道。例如：UART0 RX 配置系统 DMA 通道 1 则设置 sdma_dma_ch_remap0[13:8] = 8 配置完成需配置 update_dma_remp_0_3 使配置生效。
2. 配置系统 DMA 数据通道，包括数据传输源和目的地址、数据接收区地址、数据传输个数、传输类型等参数。具体配置时请参见 DMA 控制器章节。
3. 通过系统 DMA 中断上报，判断数据是否接收完成。

18.2.5 UART 寄存器概览

表格 18-4 芯片的 6 组 UART 模块基地址

GPIO 模块	基地址
UART0	0x04140000
UART1	0x04150000
UART2	0x04160000
UART3	0x04170000
UART4	0x041C0000
RTCSYS_UART	0x05022000

表格 18-5 芯片的 UART 寄存器概览

Name	Address Offset	Description
RBR_THR_DLL	0x000	Receive Buffer,Transmit Holding or Divisor Latch Low byte Register
IER_DLH	0x004	Interrupt Enable or Divisor Latch high byte Register
FCR_IIR	0x008	FIFO Control or Interrupt Identification Register
LCR	0x00c	Line Control Register
MCR	0x010	Modem Control Register

Name	Address Offset	Description
LSR	0x014	Line Status Register
MSR	0x018	Modem Status Register
LPDLL	0x020	Low Power Divisor Latch (Low) Register
LPDLH	0x024	Low Power Divisor Latch (High) Register
SRBR_STHR	0x030	Shadow Receive/Trasnmit Buffer Register
FAR	0x070	FIFO Access Register
TFR	0x074	Transmit FIFO Read
RFW	0x078	Receive FIFO Write
USR	0x07c	UART Status Register
TFL	0x080	Transmit FIFO Level
RFL	0x084	Receive FIFO Level
SRR	0x088	Software Reset Register
SRTS	0x08c	Shadow Request to Send
SBCR	0x090	Shadow Break Control Register
SDMAM	0x094	Shadow DMA Mode
SFE	0x098	Shadow FIFO Enable
SRT	0x09c	Shadow RCVR Trigger
STET	0x0a0	Shadow TX Empty Trigger
HTX	0x0a4	Halt TX
DMASA	0x0a8	DMA Software Acknowledge

18.2.6 UART 寄存器描述

RBR_THR_DLL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	RBR_THR_DLL	R/W	LCR[7] bit = 0 ： (R) Receive Buffer Register ,Data byte received on the serial input port (W)Transmit Holding Register,Data to be transmitted on the serial output port LCR[7] bit = 1 ： Lower 8 bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART	0x0
31:8	Reserved			

IER_DLH

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	IER_DLH	R/W	LCR[7] bit = 0 ： IER[0] : Enable Received Data Available Interrupt. IER[1] : Enable Transmit Holding Register Empty Interrupt. IER[2] : Enable Receiver Line Status Interrupt. IER[3] : Enable Modem Status Interrupt. IER[7] : Programmable THRE Interrupt Mode Enable	0x0

Bits	Name	Access	Description	Reset
			LCR[7] bit = 1 ： Upper 8-bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART.	
31:8	Reserved			

FCR_IIR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	FCR_IIR	R/W	(R) interrupt Identification Register [3:0] Interrupt ID 0000 : modem status 0001 : no interrupt pending 0010 : THR empty 0100 : received data available 0110 : receiver line status 0111 : busy detect 1100 : character timeout [7:6] FIFOs Enabled 00 – disabled 11 – enable (W) FIFO Control Register [0] FIFO Enable [1] RCVR FIFO Reset [2] XMIT FIFO Reset [3] DMA Mode 0 – mode 0, single DMA data transfers at a time 1 – mode 1, multi DMA data transfers are made continuously [5:4] TX Empty 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO ¼ full 11 – FIFO ½ full [7:6] RCVR Trigger 00 – 1 character in the FIFO 01 – FIFO ¼ full 10 – FIFO ½ full 11 – FIFO 2 less than full	0x1
31:8	Reserved			

LCR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	LCR	R/W	Line Control Register [1:0] Data Length Select. (00:5 bits,01:6 bits,10:7 bits,11:8 bits) [2] Number of stop bits. (0:1 stop bit,1:1.5 stop bits when Data Length Select is 0, else 2 stop bits) [3] Parity Enable [4] Even Parity Select [5] Stick Parity [6] Break Control Bit [7] Divisor Latch Access Bit	0x0

Bits	Name	Access	Description	Reset
31:8	Reserved			

MCR

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	MCR	R/W	Modem Control Register [0] reserved [1] Request to Send. This is used to directly control the Request to Send (rts_n) output [2] reserved [3] reserved [4] reserved [5] Auto Flow Control Enable. [6] reserved	0x0
31:8	Reserved			

LSR

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	LSR	RO	Line Status Register [0] Data Ready bit. there is at least one character in the RBR or the receiver FIFO. [1] Overrun error bit. This is used to indicate the occurrence of an overrun error. [2] Parity Error bit. [3] Framing Error bit.. [4] Break Interrupt bit. [5] Transmit Holding Register Empty bit. [6] Transmitter Empty bit. [7] Receiver FIFO Error bit.	
31:8	Reserved			

MSR

Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	MSR	RO	Modem Status Register [0] Delta Clear to Send. [1] reserved [2] reserved [3] reserved [4] CTS [5] reserved [6] reserved [7] reserved	
31:8	Reserved			

LPDLL

Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	LPDLL	R/W	LCR[7] bit = 1 : Low Power Divisor Latch	0x0

Bits	Name	Access	Description	Reset
			(Low) Register	
31:8	Reserved			

LPDLH

Offset Address: 0x024

Bits	Name	Access	Description	Reset
7:0	LPDLH	R/W	LCR[7] bit = 1 : Low Power Divisor Latch (High) Register	0x0
31:8	Reserved			

SRBR_STHR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	SRBR_STHR	R/W	LCR[7] bit = 0 : (R) Shadow Receive Buffer Register (W) Shadow Transmit Holding Register	0x0
31:8	Reserved			

FAR

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	FAR	R/W	FIFO Access Register,This register is use to enable a FIFO access mode for testing	0x0
31:1	Reserved			

TFR

Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	TFR	R/W	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled	0x0
31:8	Reserved			

RFW

Offset Address: 0x078

Bits	Name	Access	Description	Reset
9:0	RFW	R/W	Receive FIFO Write. These bits are only valid when FIFO access mode is enabled [7:0] Receive FIFO Write Data. [8] Receive FIFO Parity Error. [9] Receive FIFO Framing Error.	0x0
31:10	Reserved			

USR

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
4:0	USR	RO	UART Status Register [0] UART Busy. [1] Transmit FIFO Not Full. [2] Transmit FIFO Empty. [3] Receive FIFO Not Empty. [4] Receive FIFO Full.	

Bits	Name	Access	Description	Reset
31:5	Reserved			

TFL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
5:0	TFL	RO	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.	
31:6	Reserved			

RFL

Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	RFL	RO	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.	
31:6	Reserved			

SRR

Offset Address: 0x088

Bits	Name	Access	Description	Reset
2:0	SRR	R/W	Software Reset Register [0] UART Reset. [1] RCVR FIFO Reset. [2] XMIT FIFO Reset.	0x0
31:3	Reserved			

SRTS

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	SRTS	R/W	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1])	0x0
31:1	Reserved			

SBCR

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SBCR	R/W	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]).	0x0
31:1	Reserved			

SDMAM

Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDMAM	R/W	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).	0x0
31:1	Reserved			

SFE

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	SFE	R/W	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).	0x0
31:1	Reserved			

SRT

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
1:0	SRT	R/W	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).	0x0
31:2	Reserved			

STET

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
1:0	STET	R/W	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).	0x0
31:2	Reserved			

HTX

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	HTX	R/W	This register is use to halt transmissions for testing,	0x0
31:1	Reserved			

DMASA

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	DMASA	R/W	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.	0x0
31:1	Reserved			

18.3 SPI

18.3.1 概述

系统配置 4 个 SPI 控制器模块，可以作为 Master 与外部设备进行同步串行通信，实现数据的串并、并串转换。

18.3.2 特点

SPI 控制器模块的特点如下：

- 支持 Motorola SPI (全双工) 、TI SSP (全双工) 、NS MicroWire (半双工) 三种串行外设接口协议
- 独立的收/发 FIFO
- 可编程的数据帧长度：4～16 bits
- SPI 接口时钟频率可编程
- 支持 DMA 操作模式
- 支持内部环回测试模式
- 工作参考时钟可设置为 187.5MHz 或 100MHz，输出的 SPI_SCK 最大支持 46.875MHz。

18.3.3 功能描述

18.3.3.1 典型应用

SPI master 对接外部 slave 时的应用框图如图表 18-9 所示。

图表 18-9 SPI 应用框图

18.3.4 工作方式

18.3.4.1 工作模式

SPI 的工作模式分为：

- 中断或查询方式下的数据传输
- DMA 方式下的数据传输。

18.3.4.2 时钟

SPI 控制器模块参考时钟可设置为 187.5MHz 或 100MHz。
输出的 SPI_SCK 最大支持 46.875MHz。
计算方式如下：
输出的 SPI_SCK = SPI 工作参考时钟/BAUDR
SPI 工作参考时钟 : 187.5MHz 或 100MHz。
BAUDR 寄存器 : 设置介于 2 到 65534 之间的偶数.
计算范例 :
SPI 工作参考时钟 = 187.5MHz 与 BAUDR =4
输出的 SPI_SCK = 187.5MHz/4 = 46.875MHz

18.3.4.3 中断处理

SPI 控制器模块有 6 个中断，其中前 5 个是高电平有效的可屏蔽独立中断源。

- RXFINTR
  接收 FIFO 中断请求。当接收 FIFO 中有 RXFTLR+1 个或更多的有效数据时，该中断置位。
- RXOINTR
  当接收 FIFO 已满，且又有新的数据需要写入 FIFO 时，会引起 FIFO Overflow，该中断置位。此时数据被写入接收移位寄存器，而不是FIFO。
- RXUINTR
  当接收 FIFO 被读空，没有新的数据被写入接收 FIFO 之前又有新的读取请求发生，会引起 FIFO Underflow，该中断置位。此时读取的值全部为 0。可以通过读寄存器 RXUICR 清除该中断。
- TXOINTR
  当发送 FIFO 已满，且又有新的数据需要写入 FIFO 时，会引起 FIFO Overflow，该中断置位。
- TXEINTR
  发送FIFO 中断请求。当发送FIFO 中有 TXFTLR 个或更少的有效数据时，该中断置位。
- SPI_INTR
  组合中断，为以上 5 个中断经过“或”运算后的结果。

若要屏蔽此中断，必须设置寄存器 IMR 屏蔽上述 5 个中断。
如果上述 5 个独立中断中任意一个置位且使能，该中断置位。

18.3.4.4 初始化

SPI 控制器模块初始化步骤如下：
步骤 1 寄存器 SPIENR 设置“0”， 停止 SPI 模块。
步骤 2 配置寄存器 BAUDR，设定输出时钟分频除数，设定值必须为偶数。
步骤 3 设置寄存器 CTRLR0，配置传输数据位宽及传输帧格式等参数。
步骤 4 DMA 操作模式下，配置寄存器 DMACR，启用SPI的DMA 功能。操作于DMA 模式时，应设置中断相关寄存器以禁止产生中断信号。
步骤 5 中断操作模式下，设置寄存器 IMR，以产生相应的中断信号。
步骤 6 寄存器 SPIENR 设置“1”，使能 SPI 模块。

18.3.4.5 SPI 的数据传输流程

- SPI master 对接外部 SPI/SSP slave 时的流程如图表 18-10 所示。

图表 18-10 对接外部 SPI/SSP slave 时的数据传输流程

- SPI master 对接外部 Microwire slave 时的流程如图表 18-11 所示。

图表 18-11 对接外部 Microwire slave 时的数据传输流程

18.3.4.6 DMA 方式下的数据传输

SPI 模块使用两个 DMA 通道，一个用于发送，一个用于接收。 SPI DMA 模式设置的相

关寄存器有 DMACR、DMATDLR、DMARDLR。
启用 SPI DMA 模式的步骤如下：
步骤 1 获取两个 DMA 通道。
步骤 2 设置寄存器 DMACR[1:0]，使能 SPI DMA 发送接收。
步骤 3 寄存器 SPIENR 设置“1”，使能 SPI。
步骤 4 发送数据

1. 配置发送 DMA 通道相关的控制寄存器。
2. 启动 DMA 控制器，响应 SPI 发送 FIFO 的请求。
3. 通过 DMA 控制器中断上报，判断发送是否完成，如果完成则关闭 SPI 的发送 DMA 功能。

步骤 5 接收数据

1. 配置接收 DMA 通道相关的控制寄存器。
2. 启动 DMA 控制器，响应 SPI 接收FIFO 的 DMA 请求。
3. 通过 DMA 控制器中断上报，判断数据是否接收完成，如果完成则关闭 SPI 的接收 DMA 功能。

步骤 6 寄存器 SPIENR 设置“0”，停止 SPI。

18.3.5 三种串行外设总线时序图

18.3.5.1 Motorola SPI 接口

以下各图表示 Motorola SPI 各种数据传输格式。其中 SCPH 代表 SPI_SCK 相位，SCPOL 代表 SPI_SCK 极性，通过寄存器 CTRLR0[7:6] 进行设置。

(A) SCPH = 0

此模式下，SPI_CS_X 处于空闲状态时设置为高电平，传输时则设置为低电平。SPI_SCK 则是通过 SCPOL 设置而有所不同，SCPOL = 0，处于空闲状态时设置为低电平，传输时则以时钟的上升沿抓取数据，SCPOL = 1，处于空闲状态时设置为高电平，传输时则以时钟的下降沿抓取数据。

- 单帧传输格式如图表 18-12 所示。

图表 18-12 Motorola SPI 单帧传输格式 (SCPH = 0)

- 连续帧传输格式如图表 18-13 所示。

图表 18-13 Motorola SPI 连续帧传输格式 (SCPH = 0)

(B) SCPH = 1

此模式下，SPI_CS_X 处于空闲状态时设置为高电平，传输时则设置为低电平。SPI_SCK 则是通过 SCPOL 设置而有所不同，SCPOL = 0，处于空闲状态时设置为低电平，传输时则以时钟的下降沿抓取数据，SCPOL = 1，处于空闲状态时设置为高电平，传输时则以时钟的上升沿抓取数据。

- 单帧传输格式如图表 18-14 所示。

图表 18-14 Motorola SPI 单帧传输格式 (SCPH = 1)

- 连续帧传输格式如图表 18-15 所示。

图表 18-15 Motorola SPI 连续帧传输格式 (SCPH = 1)

18.3.5.2 TI 同步串行接口

SSP 模式下，SPI_CS_X 处于空闲状态时设置为高电平，传输时则设置为低电平。
SPI_SCK 处于空闲状态时设置为低电平，传输时则以时钟的下降沿抓取数据。
以下各图表示 TI SSP 数据传输格式。

- 单帧传输格式如图表 18-16 所示。
-

图表 18-16 TI SSP 单帧传输格式

- 连续帧传输格式如图表 18-17 所示。

图表 18-17 TI SSP 连续帧传输格式

18.3.5.3 National Semiconductor Microwire 接口

Microwire 模式下，SPI_CS_X 处于空闲状态时设置为高电平，传输时则设置为低电平。
SPI_SCK 处于空闲状态时设置为低电平，传输时则以时钟的上升沿抓取数据。
此模式在进行数据传输时, 必须先加控制字(control word), 外部芯片再根据控制字响应 Master 所需求的数据(data word). 控制字长度可通过寄存器 CTRLR0[15:12] 进行设置, 其它相关参数可通过寄存器 MWCR 进行设置。

以下各图表示 NS Microwire 数据传输格式。

- 单帧传输格式如图表 18-18 所示。

图表 18-18 NS Microwire 单帧传输格式

- 连续帧传输格式如图表 18-19 所示。

图表 18-19 NS Microwire 连续帧传输格式

18.3.6 寄存器概览

芯片的 4 组 SPI 模块基地址如表格 18-6 所示。

表格 18-6 芯片的 4 组 SPI 模块基地址

GPIO 模块	基地址
SPI0	0x04180000
SPI1	0x04190000
SPI2	0x041A0000
SPI3	0x041B0000

表格 18-7 是第 1 组 SPI 模块 (SPI0) 寄存器的偏移地址以及定义，SPI0～SPI3 具有相同的寄存器定义。

表格 18-7 SPI 寄存器概览

Name	Address Offset	Description
CTRLR0	0x000	Control Register 0
CTRLR1	0x004	Control Register 1
SPIENR	0x008	SPI Enable Register
MWCR	0x00c	Microwire Control Register
SER	0x010	Slave Enable Register
BAUDR	0x014	Baud Rate Select
TXFTLR	0x018	Transmit FIFO Threshold Level
RXFTLR	0x01c	Receive FIFO Threshold Level
TXFLR	0x020	Transmit FIFO Level Register
RXFLR	0x024	Receive FIFO Level Register
SR	0x028	Status Register
IMR	0x02c	Interrupt Mask Register
ISR	0x030	Interrupt Status Register
RISR	0x034	Raw Interrupt Status Register
TXOICR	0x038	Transmit FIFO Overflow Interrupt Clear Register
RXOICR	0x03c	Receive FIFO Overflow Interrupt Clear Register
RXUICR	0x040	Receive FIFO Underflow Interrupt Clear Register
MSTICR	0x044	Multi-Master Interrupt Clear Register
ICR	0x048	Interrupt Clear Register
DMACR	0x04c	DMA Control Register
DMATDLR	0x050	DMA Transmit Data Level
DMARDLR	0x054	DMA Receive Data Level
DR (36 组)	0x060	Data Register
RX_SAMPLE_DLY	0x0f0	Rx Sample Delay Register

18.3.7 寄存器描述

CTRLR0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	CTRLR0	R/W	[15:12] Control Frame Size. Selects the length of the control word for the Microwire frame format. 0000 1-bit control word 0001 2-bit control word 0010 3-bit control word 0011 4-bit control word 0100 5-bit control word 0101 6-bit control word 0110 7-bit control word 0111 8-bit control word 1000 9-bit control word 1001 10-bit control word 1010 11-bit control word 1011 12-bit control word 1100 13-bit control word 1101 14-bit control word 1110 15-bit control word 1111 16-bit control word	0x7

Bits	Name	Access	Description	Reset
			[11] Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serialslave and serial-master modes. 0 – Normal Mode Operation 1 – Test Mode Operation [10] only for slave mode. [9:8] Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. 00 –- Transmit & Receive 01 –- Transmit Only 10 –- Receive Only 11 –- EEPROM Read [7] Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high	

Bits	Name	Access	Description	Reset
			[6] Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit [5:4] Frame Format. Selects which serial protocol transfers the data. 00 –- Motorola SPI 01 –- Texas Instruments SSP 10 –- National Semiconductors Microwire 11 –- Reserved [3:0] Data Frame Size. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. 0000 Reserved – undefined operation 0001 Reserved – undefined operation 0010 Reserved – undefined operation 0011 4-bit serial data transfer 0100 5-bit serial data transfer 0101 6-bit serial data transfer 0110 7-bit serial data transfer 0111 8-bit serial data transfer 1000 9-bit serial data transfer 1001 10-bit serial data transfer 1010 11-bit serial data transfer 1011 12-bit serial data transfer 1100 13-bit serial data transfer 1101 14-bit serial data transfer 1110 15-bit serial data transfer 1111 16-bit serial data transfer	
31:16	Reserved			

CTRLR1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	CTRLR1	R/W	Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.	0x0
31:16	Reserved			

SPIENR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	SPIENR	R/W	SPI Enable. Enables and disables all SPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the SPI control registers when enabled. When disabled, the spi_sleep output is set (after delay) to inform the system that it is safe to remove the spi_clk, thus saving power consumption in the system.	0x0
31:1	Reserved			

MWCR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	MWCR	R/W	[2] Microwire Handshaking. Relevant only when the SPI is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the “busy/ready” handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled [1] Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the SPI from the external serial device. When this bit is set to 1, the data word is transmitted from the SPI to the external serial device.	0x0

Bits	Name	Access	Description	Reset
			[0] Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0 – non-sequential transfer 1 – sequential transfer	
31:3	Reserved			

SER

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	SER	R/W	Slave Select Enable Flag. This register corresponds to a slave select line (ss_x_n]) from the SPI master. When this register is set (1), the slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable this register that corresponds to the slave device with which the master wants to communicate. 1: Selected 0: Not Selected	0x0
31:1	Reserved			

BAUDR

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	BAUDR	R/W	SPI Clock Divider(SCKDV). The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out/SCKDV = Fssi_clk where SCKDV is any even value between 2 and 65534. For example: for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz	0x0
31:16	Reserved			

TXFTLR

Offset Address: 0x018

Bits	Name	Access	Description	Reset
2:0	TXFTLR	R/W	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 8; If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.	0x0
31:3	Reserved			

RXFTLR

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	RXFTLR	R/W	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is 8. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.	0x0
31:3	Reserved			

TXFLR

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	TXFLR	RO	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.	
31:4	Reserved			

RXFLR

Offset Address: 0x024

Bits	Name	Access	Description	Reset
3:0	RXFLR	RO	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.	
31:4	Reserved			

SR

Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	SR	RO	[6] Data Collision Error. This bit is set if the SPI master is actively transmitting when another master selects this device as a slave. This informs the processor that the last data transfer was	

Bits	Name	Access	Description	Reset
			halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error [5] Transmission Error. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 – No error 1 – Transmission error [4] Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full [3] Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty [2] Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty [1] Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full [0] SPI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 0 – SPI is idle or disabled 1 – SPI is actively transferring data	
31:7	Reserved			

IMR

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
5:0	IMR	R/W	[5] Multi-Master Contention Interrupt Mask. 0 – spi_mst_intr interrupt is masked 1 – spi_mst_intr interrupt is not masked [4] Receive FIFO Full Interrupt Mask 0 – spi_rxf_intr interrupt is masked 1 – spi_rxf_intr interrupt is not masked [3] Receive FIFO Overflow Interrupt Mask 0 – spi_rxo_intr interrupt is masked 1 – spi_rxo_intr interrupt is not masked [2] Receive FIFO Underflow Interrupt Mask 0 – spi_rxu_intr interrupt is masked 1 – spi_rxu_intr interrupt is not masked [1] Transmit FIFO Overflow Interrupt Mask 0 – spi_txo_intr interrupt is masked 1 – spi_txo_intr interrupt is not masked [0] Transmit FIFO Empty Interrupt Mask 0 – spi_txe_intr interrupt is masked 1 – spi_txe_intr interrupt is not masked	0x3F
31:6	Reserved			

ISR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
5:0	ISR	RO	[5] Multi-Master Contention Interrupt Status. 0 = spi_mst_intr interrupt not active after masking 1 = spi_mst_intr interrupt is active after masking [4] Receive FIFO Full Interrupt Status 0 = spi_rxf_intr interrupt is not active after masking 1 = spi_rxf_intr interrupt is full after masking [3] Receive FIFO Overflow Interrupt Status 0 = spi_rxo_intr interrupt is not active after masking 1 = spi_rxo_intr interrupt is active after masking [2] Receive FIFO Underflow Interrupt Status	

Bits	Name	Access	Description	Reset
			0 = spi_rxu_intr interrupt is not active after masking 1 = spi_rxu_intr interrupt is active after masking [1] Transmit FIFO Overflow Interrupt Status 0 = spi_txo_intr interrupt is not active after masking 1 = spi_txo_intr interrupt is active after masking [0] Transmit FIFO Empty Interrupt Status 0 = spi_txe_intr interrupt is not active after masking 1 = spi_txe_intr interrupt is active after masking	
31:6	Reserved			

RISR

Offset Address: 0x034

Bits	Name	Access	Description	Reset
5:0	RISR	RO	[5] Multi-Master Contention Raw Interrupt Status. 0 = spi_mst_intr interrupt is not active prior to masking 1 = spi_mst_intr interrupt is active prior masking [4] Receive FIFO Full Raw Interrupt Status 0 = spi_rxf_intr interrupt is not active prior to masking 1 = spi_rxf_intr interrupt is active prior to masking [3] Receive FIFO Overflow Raw Interrupt Status 0 = spi_rxo_intr interrupt is not active prior to masking 1 = spi_rxo_intr interrupt is active prior masking [2] Receive FIFO Underflow Raw Interrupt Status 0 = spi_rxu_intr interrupt is not active prior to masking 1 = spi_rxu_intr interrupt is active prior to masking [1] Transmit FIFO Overflow Raw Interrupt Status 0 = spi_txo_intr interrupt is not active prior to masking 1 = spi_txo_intr interrupt is active prior	

Bits	Name	Access	Description	Reset
			masking [0] Transmit FIFO Empty Raw Interrupt Status 0 = spi_txe_intr interrupt is not active prior to masking 1 = spi_txe_intr interrupt is active prior masking	
31:6	Reserved			

TXOICR

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	TXOICR	RO	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_txo_intr interrupt; writing has no effect.	
31:1	Reserved			

RXOICR

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	RXOICR	RO	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxo_intr interrupt; writing has no effect.	
31:1	Reserved			

RXUICR

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	RXUICR	RO	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxu_intr interrupt; writing has no effect.	
31:1	Reserved			

MSTICR

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	MSTICR	RO	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_mst_intr interrupt; writing has no effect.	
31:1	Reserved			

ICR

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	ICR	RO	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the spi_txo_intr, spi_rxu_intr, spi_rxo_intr, and the spi_mst_intr interrupts. Writing to this register has no effect.	
31:1	Reserved			

DMACR

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
1:0	DMACR	R/W	[1] Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled [0] Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 = Receive DMA disabled 1 = Receive DMA enabled	0x0
31:2	Reserved			

DMATDLR

Offset Address: 0x050

Bits	Name	Access	Description	Reset
2:0	DMATDLR	R/W	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.	0x0
31:3	Reserved			

DMARDLR

Offset Address: 0x054

Bits	Name	Access	Description	Reset
2:0	DMARDLR	R/W	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.	0x0
31:3	Reserved			

DR

Offset Address: 0x060

Bits	Name	Access	Description	Reset
15:0	DR	R/W	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer Note : The DR register in the SPI occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus.	0x0
31:16	Reserved			

RX_SAMPLE_DLY

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
7:0	RX_SAMPLE_DLY	R/W	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal. Each value represents a single ssi_clk delay on the sample of the rxd signal. NOTE: If this register is programmed with a value that exceeds the depth of the internal shift registers (DEPTH = 8), a zero (0) delay will be applied to the rxd sample.	0x0
31:8	Reserved			

18.4 eMMC/SD/SDIO 控制器

18.4.1 功能描述

18.4.1.1 功能框图

EMMC/SD/SDIO 控制器(简称 SDMMC 控制器)用于处理 SD 卡与 eMMC 的资料读写等操作，以及基于 SDIO 协议所支援的外部装置(如蓝牙、WIFI 等)。本芯片提供了三套 SDMMC 控制器。其中：

EMMC 支援符合 eMMC4.1 与 eMMC4.5 协议的设备。

SDIO0 支援符合 Secure Digital Memory (SD 3.0)协议的设备。
SDIO1 支援符合 Secure Digital I/O(SDIO 3.0)协议的设备。

芯片中 3 个 SDMMC 控制器对应的功能信号和脚位，如下表所示。

表格 18-8 SDMMC 控制器对应的功能讯号与脚位对应表

SDMMC 控制器	功能讯号	脚位
eMMC	EMMC_CLK	EMMC_CLK
	EMMC_CMD	EMMC_CMD
	EMMC_DATA0	EMMC_DAT0
	EMMC_DATA1	EMMC_DAT1
	EMMC_DATA2	EMMC_DAT2
	EMMC_DATA3	EMMC_DAT3
	EMMC_RSTn	EMMC_RSTN
SDIO0	SD_CLK	SD0_CLK
	SD_CMD	SD0_CMD
	SD_DATA0	SD0_D0
	SD_DATA1	SD0_D1
	SD_DATA2	SD0_D2
	SD_DATA3	SD0_D3
	SD_CARD_DETECT	SD0_CD
	SD_POWER_EN	SD0_PWR_EN
SDIO1	SDIO_CLK	SD1_CLK
	SDIO_CMD	SD1_CMD
	SDIO_DATA0	SD1_D0
	SDIO_DATA1	SD1_D1
	SDIO_DATA2	SD1_D2
	SDIO_DATA3	SD1_D3

图表 18-20 SDMMC 控制器的功能框图

SDMMC 的功能：

1. 支援 SD 卡，SDIO 装置与 eMMC。
2. 透过内部的 DMA 控制器，在 eMMC/SD/SDIO 与系统记忆体资料之间传输资料。
3. 支援命令与数据的 CRC 产生与检查。
4. 可透过内部除频器产生不同模式间需要的频率。
5. 提供关闭内部时钟与介面上时钟的机制来达到省电需求。
6. 提供 1-bit 与 4-bit 数据传输介面来与装置沟通。
7. 支援 block_size 等于 1~2048byte 的数据读写操作。
8. 支援 SDIO 协定，包含中断区间，suspend，resume 与 read wait 等操作。
9. 支援 AXI/AHB 介面，可透过内部 DMA 来存取系统记忆体。
10. 支援 AHB 介面，可透过 CPU 来存取内部的暂存器。

图表 18-21 典型应用

18.4.1.2 指令与响应

eMMC/SD 的总线封包主要由三者组成:指令，响应与数据。
其中指令与响应封包是透过 CMD 讯号线传输。

- 指令封包

指令封包由主控端传送给装置，用来指示一个操作的开始。封包格式由起始位元，传输位元，指令编号，指令参数，CRC 验证码和结束位元共 48 位元组成。如图表 18-22 所示。

图表 18-22 eMMC/SD/SDIO 指令格式

- 响应封包
  装置端接受到指令后，会根据不同指令类别返回响应，用来显示装置端的状态或参数。其长度为 48 位元或 136 位元。如图表 18-23 所示。

图表 18-23 eMMC/SD/SDIO 响应格式

- 数据封包: 资料封包用来交换主控端与装置间的数据，根据不同需求可选择 1-bit (DATA0), 4-bit(DATA0-DATA3)或者 7-bit(DATA0-DATA7)，在每一时钟区间，每一条数据信号线可选择传输 1-bit (single data rate)或者 2-bit(dual data rate)。其封包格分别如图表 18-24~图表 18-28 所示。

图表 18-24 eMMC/SD/SDIO 1-bit 数据封包格式

图表 18-25 eMMC/SD/SDIO 4-bit 数据封包格式

图表 18-26 eMMC/SD/SDIO 8-bit 数据封包格式

图表 18-27 4-bit dual data rate 数据封包格式

图表 18-28 8-bit dual data rate 数据封包格式

根据是否有资料传输，指令可再分为以下两种:

- 非数据传输指令:透过信号线 CMD 来完成指令传送与接收响应。

图表 18-29 非数据传输指令:透过信号线 CMD 来完成指令传送与接收响应

- 数据传输指令:除了信号线 CMD 上的交互外，还有数据线 DAT0~DAT3 的数据传输

18.4.1.3 数据传输

在主控端与装置端的数据传输主要以块(block)为单位，除了数据外还会夹带 CRC 检查位元来验证数据的正确性。比较常用的有单块数据读写和多块数据读写方式，相较于单块数据传输而言，多块数据传输有比较高的效率。其中, EMMC 与 SD 卡的块大小为 512byte，SDIO 比较特殊，可支援 1~2048byte 的块大小，使用者可根据不同的装置来定义块大小的数值。

(1) 单块与多块读操作如图表 18-30，单块传输由指令，响应，数据与 CRC 组成。多块传输最后可靠 STOP CMD 来中止传输。

图表 18-30 单块与多块读操作

(2) 单块与多块写操作如图表 18-31，传输过程会透过 DAT0 讯号线来发 BUSY 讯号，通知主控端写入装置正在进行。

图表 18-31 单块与多块写操作

18.4.1.4 SD3.0 支持的速度模式及电压切换

- 电压切换程序 (1.8V → 3.3V)

步骤一: 设定 PWRSW 成 3.3V
=> sd_pwrsw_ctrl (0x030001F4) = 0x00000009
(reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=0(3.0v), reg_en_pwrsw=1)

步骤二: 等待 1ms 来完成电压切换

- 电压切换程序 (3.3V → 1.8V)

步骤一: 设定 PWRSW 成 1.8V
=> sd_pwrsw_ctrl (0x030001F4) = 0x0000000B
(reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=1(1.8v), reg_en_pwrsw=1)

步骤二: 等待 1ms 来完成电压切换。

- 支持速度模式与电压

SD3.0 所支持速度模式与电压如下表格。

表格 18-9 SD3.0 支持的速度和电压

支持模式		支持速度	电压
DS (default speed)		25Mhz	1.8V/3.3V
HS (high speed)		50Mhz	1.8V/3.3V
SDR12		25Mhz	1.8V
SDR25		50Mhz	1.8V
DDR50		50Mhz	1.8V
SDR50		100Mhz	1.8V
SDR104		187.5Mhz	1.8V

18.4.1.5 eMMC 支持的速度模式及电压

eMMC 所支持速度模式与电压如下表。

表格 18-10 eMMC 支持速度模式与电压

支持模式	支持速度	电压
DS (default speed)	26Mhz	1.8V/3.3V
HS (high speed)	52Mhz	1.8V/3.3V
DDR52	52Mhz	1.8V
HS200	187.5Mhz	1.8V

18.4.2 应用说明

18.4.2.1 时钟阀控

图表 18-32 时钟关控程序

如图表 18-32 为时钟关控程序，主控端必须确保在汇流排上没有传输进行才能关闭时钟。

(1) 读取暂存器 PRESENT_STS
(2) 检查位元 CMD_INHIBIT 与 DAT_INHIBIT 是否都是 0
(3) 如果有任一位元不为 0, 代表传输仍在进行，需要延迟等待。
(4) 如果皆为 0，则可以设定 CLK_CTL[SD_CLK_EN]=0 来关闭时钟

18.4.2.2 软复位

当控制器操作异常时，可靠复位配置暂存器(base address = 0x0300_3000)来进行软复位。其使用的暂存器位址分别如下所示:

6. EMMC : SOFT_RSTN_0[reg_soft_reset_x_emmc] (address offset : 0x000, Bit15)
7. SDIO0 : SOFT_RSTN_0[reg_soft_reset_x_sd0] (address offset : 0x000, Bit16)
8. SDIO1 : SOFT_RSTN_0[reg_soft_reset_x_sd1] (address offset : 0x000, Bit17)

18.4.2.3 接口时钟配置

图表 18-33 为接口时钟配置流程图。SDMMC 控制器内部提供一个除频器，让使用者根据不同协议与速度模式，来调整所需要的时钟频率。其关系式为:

$$F_{SD_CLK_OUT} = F_{INT_CARD_CLK} / (2 \times clk_divisor)$$

当 SDMMC 改变频率时除了要确保没有指令与数据仍在传输中之外，还必须按照接口时钟配置流程图的步骤来设定，避免输出到 eMMC/SD 装置的时钟产生毛刺。

(1) 关闭接口时钟。
(2) 计算分频因子。
(3) 设置分频因子。将(2)计算出来的参数填入 CLK_CTL[FREQ_SEL]，并且启动打开内部时钟开关(CLK_CTL[INT_CLK_EN]=1)。
(4) 检查 CLK_CTL[INT_CLK_STABLE]来确认频率是否切换完成。
(5) 如果还未完成(CLK_CTL[INT_CLK_STABLE]=0)，则延迟等待。
(6) 如果切换时钟频率完成，则打开接口时钟。

图表 18-33 时钟配置流程图

18.4.2.4 非数据传输指令

- 指令传输程序

  指令传输程序如图表 18-34。

  (1) 首先必须检查暂存器位元 PRESENT[CMD_INHIBIT]是否为 0 来确认 CMD Line 是

否仍在使用中。

(2) 如果 CMD Line 是空闲的，则进一步确认是否为带 busy 的指令。如果非 busy 指令，则不需要检查 DATA Line 上的状况，直接执行步骤(5)，反之如果为带 busy 的指令，则必须执行步骤(3)确认是否为 Abort 指令。

(3) 如果为 Abort 指令，代表 CMD line 完成传输时，DATA line 也同时是闲置的，可直接进入步骤(5)；反之，如果不是 Abort 指令，则必须执行步骤(4)确认 DATA Line 上 busy 是否已经释放。

(4) 检查暂存器位元 PRESENT[DAT_INHIBIT]是否为 0 来确认 DATA Line 上是否仍在使用中，如果仍在使用中则持续等待直到传输结束，然后再执行步骤(5)。

(5) 根据指令需求来设定 ARGUMENT 暂存器与 CMD 暂存器的值。

图表 18-34 指令传输程序

- 指令完成程序 (Command Complete Sequence)
  指令完成程序如图表 18-35。
  (1) 首先会等待 Command 完成的中断 NORM_INT_STS[CMD_CMPL]。
  (2) 收到中断后设定 NORM_INT_STS[CMD_CMPL]=1 来清除 CMD_CMPL 中断状态。
  (3) 接着读取 RESP1_0，RESP3_2，RESP5_4，RESP7_6 等暂存器来获取响应值。
  (4) 如果是一个包含资料传输的指令，会执行步骤(5)，否则跳到步骤(8)。
  (5) 等待资料传输中断 NORM_INT_STS[XFER_CMPL]。
  (6) 收到中断后设定 NORM_INT_STS[XFER_CMPL]=1 来清除 XFER_CMPL 中断状态。
  (7) 检查 RESP1_0，RESP3_2，RESP5_4，RESP7_6 等暂存器确认是否有错误状态。如果没错误状态则执行步骤(8)，回报没有错误。有错误则执行步骤(9)来回报错误。

图表 18-35 指令完成程序

18.4.2.5 停止或中止数据传输

- 中止(Abort)指令程序
  中止指令对于 eMMC/SD 装置是靠 CMD12, 而 SDIO 装置则是 CMD52 来完成。使用时机主要有两种状况，
  (1) 停止无限模块数据传输。
  (2) 停止多模块数据传输。
  中止指令程序如图表 18-36，详细步骤如下:

图表 18-36 中止指令程序

中止指令有两种方式: 同步中止指令与非同步中止指令。

- 非同步中止指令程序

  图表 18-37 为非同步中止指令的程序图。详细步骤如下:

  (1) 根据不同传输模式执行中止指令。

  (2) 设定 SW_RESET 暂存器中的 SW_RST_CMD 与 SW_RST_DAT 来重置 CMD 与 DAT 信号线。

  (3) 检查位元 SW_RESET[SW_RST_CMD]与 SW_RESET[SW_RST_DAT]确认是否重置完成。若两者皆为 0，则结束程序。若其中之一为 1，则回到步骤(3)延迟等待。

图表 18-37 非同步中止指令的程序图

- 同步中止指令程序

  图表 18-38 为同步中止指令的程序图。详细步骤如下:

  (1) 写入位元 BG_CTL[STOP_BG_REQ]在 Block Gap 区间停止传输动作。

  (2) 等待传输完成的中断 NORM_INT_STS[XFER_CMPL]。

  (3) 收到中断后设定 NORM_INT_STS[XFER_CMPL]=1 来清除 XFER_CMPL 中断状态。

  (4) 根据不同传输模式执行中止指令。

(5) 设定 SW_RESET 暂存器中的 SW_RST_CMD 与 SW_RST_DAT 来重置 CMD 与 DAT 信号线。
(6) 检查位元 SW_RESET[SW_RST_CMD]与 SW_RESET[SW_RST_DAT]确认是否重置完成。若两者皆为 0，则结束程序。若其中之一为 1，则回到步骤(6)延迟等待。

图表 18-38 同步中止指令的程序图

18.4.2.6 非 DMA 数据传输模式

非 DMA 数据传输模式的程序如图表 18-39。详细步骤如下所示:
(1) 写入 BLK_SIZE 暂存器来设定块大小。
(2) 写入 BLK_CNT 暂存器来设定块数目。
(3) 写入 ARGUMENT 暂存器来设定指令参数。
(4) 写入 XFER_MODE 暂存器来设定传输模式。主控端可根据使用情境来决定设定。包含 Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable。

(5) 写入 CMD 暂存器来设定指令与响应的类型。
(6) 等待 Command 完成的中断 NORM_INT_STS[CMD_CMPL]。
(7) 收到中断后设定 NORM_INT_STS[CMD_CMPL]=1 来清除 CMD_CMPL 中断状态。
(8) 接着读取 RESP1_0，RESP3_2，RESP5_4，RESP7_6 等暂存器来获取响应值。
(9) 如果是读取操作将执行步骤(14)，如果是写入操作将执行步骤(10)。
(10) 等待 Buffer Write Ready 的中断 NORM_INT_STS[BUF_WRDY]。
(11) 收到中断后设定 NORM_INT_STS[BUFF_WRDY]=1 来清除 BUFF_WRDY 中断状态。
(12) 将想要写入装置的数据依序写入 BUF_DATA 暂存器。
(13) 如果还有更多的 block 要写入，则回到步骤(10)，直到最后一个 block 写完后，则进入步骤(18)。
(14) 等待 Buffer Read Ready 的中断 NORM_INT_STS[BUF_RRDY]。
(15) 收到中断后设定 NORM_INT_STS[BUFF_RRDY]=1 来清除 BUFF_RRDY 中断状态。
(16) 依序从 BUF_DATA 暂存器读取从装置端接收回来的资料。
(17) 如果还有更多的 block 要读取，则回到步骤(14)，直到最后一个 block 读完后，则进入步骤(18)。
(18) 判断是单一模块传输，多模块传输或者无限模块传输。如果是单模块或者多模块传输则跳至步骤(19)。如果是无限模块传输则跳至步骤(21)，执行中止传输的动作。
(19) 等待数据传输完成的中断 NORM_INT_STS[XFER_CMPL]。
(20) 收到中断后设定 NORM_INT_STS[CMD_XFER]=1 来清除 XFER_CMPL 中断状态。
(21) 执行中止传输程序。

图表 18-39 非 DMA 数据传输模式的程序

18.4.2.7 SDMA 数据传输模式

SDMA 数据传输模式程序如图表 18-40，详细步骤如下所示:

(1) 写入 SDMA_SA 暂存器来设定数据传输时所使用的系统记忆体起始位址。
(2) 写入 BLK_SIZE 暂存器来设定块大小。
(3) 写入 BLK_CNT 暂存器来设定块数目。
(4) 写入 ARGUMENT 暂存器来设定指令参数。
(5) 写入 XFER_MODE 暂存器来设定传输模式。主控端可根据使用情境来决定设定。包含 Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable。
(6) 写入 CMD 暂存器来设定指令与响应的类型。
(7) 等待 Command 完成的中断 NORM_INT_STS[CMD_CMPL]。
(8) 收到中断后设定 NORM_INT_STS[CMD_CMPL]=1 来清除 CMD_CMPL 中断状态。
(9) 接着读取 RESP1_0，RESP3_2，RESP5_4，RESP7_6 等暂存器来获取响应值。
(10) 等待数据传输中断与 DMA 中断
(11) 读取中断状态暂存器 NORM_INT_STS 来判断中断类型。如果是 DMA 中断则跳至步骤(12)，如果是数据传输中断则跳至步骤(14)。
(12) 设定 NORM_INT_STS[DMA_INT]=1 来清除 DMA_INT 状态值。
(13) 写入 SDMA_SA 暂存器重新设定下一次 DMA 的系统记忆体起始位址。然后跳至步骤(10)。
(14) 设定 NORM_INT_STS[DMA_INT]=1 与[NORM_INT_STS[XFER_CMPL]=1 来清除 DMA_INT 与 XFER_CMPL 状态值。然后结束程序。

图表 18-40 SDMA 数据传输模式程序

18.4.2.8 ADMA 数据传输模式

ADMA 数据传输模式程序如图表 18-41，详细步骤如下所示:

(1) 将 ADMA 描述表填入系统记忆体。

(2) 写入 ADMA_SA_L 与 ADMA_SA_H 暂存器来设定描述表所使用的系统记忆体起始位址。
(3) 写入 BLK_SIZE 暂存器来设定块大小。
(4) 写入 BLK_CNT 暂存器来设定块数目。
(5) 写入 ARGUMENT 暂存器来设定指令参数。
(6) 写入 XFER_MODE 暂存器来设定传输模式。主控端可根据使用情境来决定设定。包含 Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable。
(7) 写入 CMD 暂存器来设定指令与响应的类型。
(8) 等待 Command 完成的中断 NORM_INT_STS[CMD_CMPL]。
(9) 收到中断后设定 NORM_INT_STS[CMD_CMPL]=1 来清除 CMD_CMPL 中断状态。
(10) 接着读取 RESP1_0，RESP3_2，RESP5_4，RESP7_6 等暂存器来获取响应值。
(11) 等待数据传输中断或者 ADMA 错误(ADMA Error)中断。
(12) 读取中断状态暂存器 NORM_INT_STS 与 ERR_INT_STS 来判断中断类型。如果是 ADMA 错误中断则跳至步骤(13)，如果是数据传输中断则跳至步骤(15)。
(13) 设定 ERR_INT_STS[ADMA_ERR]=1 来清除 ADMA_ERR 状态值。
(14) 进入 ADMA 中止传输程序(ADMA Abort Transaction)，执行 Abort Command 来中止与装置端的资料传输，必要的时候可以检查 ADMA Error Status 暂存器来检查错误发生的原因。
(15) 设 NORM_INT_STS[XFER_CMPL]=1 来清 XFER_CMPL 状态值。然后结束程序。

图表 18-41 ADMA 数据传输模式程序

18.4.3 寄存器概览

表格 18-11 为 SDMMC 寄存器概览(EMMC 的基本位址为 0x0430_0000, SDIO0 为 0x0431_0000, SDIO1 为 0x0500_0000)

表格 18-11 SDMMC 寄存器概览

Name	Address Offset	Description
SDMA_SADDR	0x000	SDMA System Memory Address/ Argument2
BLK_SIZE_AND_CNT	0x004	Block Size and Block Count Register
ARGUMENT	0x008	Argument 1 Register
XFER_MODE_AND_CMD	0x00c	Transfer Mode and Command Register
RESP31_0	0x010	Response Bit 31-0 Regsiter
RESP63_32	0x014	Response Bit 63-32 Regsiter
RESP95_64	0x018	Response Bit 95-64 Regsiter
RESP127_96	0x01c	Response Bit 127-96 Regsiter
BUF_DATA	0x020	Buffer Data Port Register
PRESENT_STS	0x024	Present State Register
HOST_CTL1_PWR_BG_WUP	0x028	Host Control 1 , Power, Block Gap and Wakeup Register
CLK_CTL_SWRST	0x02c	Clock and Reset Control Register
NORM_AND_ERR_INT_STS	0x030	Normal and Error Interrupt Status Register
NORM_AND_ERR_INT_STS_EN	0x034	Normal and Error Interrupt Status Enable Register
NORM_AND_ERR_INT_SIG_EN	0x038	Normal and Error Interrupt Signal Enable Register
AUTO_CMD_ERR_AND_HOST_CTL2	0x03c	Auto CMD Error Status Register and Host Control 2 register
CAPABILITIES1	0x040	Capabilities 1 Register
CAPABILITIES2	0x044	Capabilities 2 Register
FORCE_EVENT_ERR	0x050	Force Event Register for Auto CMD Error Status
ADMA_ERR_STS	0x054	ADMA Error Status Register
ADMA_SADDR_L	0x058	ADMA System Address Register for low 32-bit
ADMA_SADDR_H	0x05c	ADMA System Address Register for high 32-bit
PRESENT_VUL_INIT_DS	0x060	Present Value Register for Initialization and Default Speed
PRESENT_VUL_HS_SDR12	0x064	Present Value Register for High-speed and SDR12
PRESENT_VUL_SDR25_SDR50	0x068	Present Value Register for SDR25 and SDR50
PRESENT_VUL_SDR104_DDR50	0x06c	Present Value Register for SDR104 and DDR50
SLOT_INT_AND_HOST_VER	0x0fc	Slot Interrupt Status and Host Controller Version Register
EMMC_CTRL	0x200	MSHC Control register
EMMC_BOOT_CTL	0x204	eMMC Boot Control Register
CDET_TOUT_CTL	0x208	Card Detect Control Register
MBIU_CTRL	0x20c	MBIU Control register
PHY_TX_RX_DLY	0x240	PHY tx and rx delay line register
PHY_DS_DLY	0x244	PHY DS delay line register
PHY_DLY_STS	0x248	PHY delay line status register
PHY_CONFIG	0x24c	PHY Configuration register

18.4.4 寄存器描述

下面为详细的寄存器描述。

SDMA_SADDR

SDMA System Memory Address/ Argument2
Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	SDMA_SA	R/W	Physical system memory address used for DMA transfer and the second argument for Auto CMD23	0x0

BLK_SIZE_AND_CNT

Block Size and Block Count Register
Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	XFER_BLK_SIZE	R/W	Block Size of data transfer. - 0x1 : 1 byte - 0x2 : 2 bytes ..... - 0x200 : 512 bytes ..... - 0x800 : 2048 bytes	0x0
14:12	SDMA_BUF_BDARY	R/W	Host SDMA buffer Boundary - 0x0 (4K bytes) - 0x1 (8K bytes) - 0x2 (16K bytes) - 0x3 (32K bytes) - 0x4 (64K bytes) - 0x5 (128K bytes) - 0x6 (256K bytes) - 0x7 (512K bytes)	0x0
15	Reserved			
31:16	BLK_CNT	R/W	Blocks Count for Current Transfer	0x0

ARGUMENT

Argument 1 Register
Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	ARGUMENT	R/W	Command Argument 1	0x0

XFER_MODE_AND_CMD

Transfer Mode and Command Register
Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	DMA_ENABLE	R/W	DMA enable 1 : DMA Data Transfer 0 : No data transfer or Non DMA data transfer	0x0
1	BLK_CNT_ENABLE	R/W	Block Count Enable. This bit is used to enable the block count regiser, which is only relevant for multiple block transfers. 1 : Enable 0 : Disable	0x0
3:2	AUTO_CMD_ENABLE	R/W	Auto CMD enable. This field determines use of auto command funcitons 0x0 : Auto command Disabled	0x0

Bits	Name	Access	Description	Reset
			0x1 : Auto CMD12 Enable 0x2 : Auto CMD23 Enable 0x3 : Reserved	
4	DAT_XFER_DIR	R/W	Data Transfer Direction Select 1 : Read (card to host) 0 : Write (host to card)	0x0
5	MULTI_BLK_SEL	R/W	Multi/single Block Select 1 : Multiple block transfer 0 : Single block transfer	0x0
6	RESP_TYPE	R/W	Response Type R1/R5 0x0 : R1 (Memory) 0x1 : R5 (SDIO)	0x0
7	RESP_ERR_CHK_ENABLE	R/W	Response Error Check Enable 1 : Enable 0 : Disable	0x0
8	RESP_INT_DISABLE	R/W	Response Interrupt Disable 1 : Disable 0 : Enable	0x0
15:9	Reserved			
17:16	RESP_TYPE_SEL	R/W	Response Type Select 0x0 : No Response 0x1 : Response Length 136 0x2 : Response Length 48 0x3 : Response Length 48 with busy	0x0
18	SUB_CMD_FLAG	R/W	Sub Command Flag 1 : Sub Command 0 : Main Command	0x0
19	CMD_CRC_CHK_ENABLE	R/W	Command CRC check enable 1 : Enable 0 : Disable	0x0
20	CMD_IDX_CHK_ENABLE	R/W	Command Index Check Enable 1 : Enable 0 : Disable	0x0
21	DATA_PRESENT_SEL	R/W	Data Present Select. It is set to 0 for following : (1) Commands using only CMD line (ex. CMD52) (2) Commands with no data transfer but using busy signal on DAT0 (ex. R1b) (3) Resume command 1 : Data Present 0 : No Data Present	0x0
23:22	CMD_TYPE	R/W	Command Type 0x0 : Normal 0x1 : Suspend (CMD52 for writing "Bus Suspend" in CCCR) 0x2 : CMD52 for writing "Function Select" in CCCR) 0x3 : Abort (CMD12, CMD52 for writing "I/O Abort" in CCCR)	0x0
29:24	CMD_IDX	R/W	Command Index	0x0
31:30	Reserved			

RESP31_0

Response Bit 31-0 Regsiter
Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RESP31_0	RO	Command Response for RSP[39:8]	

RESP63_32

Response Bit 63-32 Regsiter
Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	RESP63_32	RO	Command Response for RSP[71:40]	

RESP95_64

Response Bit 95-64 Regsiter
Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RESP95_64	RO	Command Response for RSP[103:72]	

RESP127_96

Response Bit 127-96 Regsiter
Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RESP127_96	RO	Command Response for RSP[135:104]	

BUF_DATA

Buffer Data Port Register
Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	BUF_DATA	R/W	Buffer Data	0x0

PRESENT_STS

Present State Register
Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	CMD_INHIBIT	RO	Command Inhibit (CMD) 1 : Cannot issue command 0 : Can issue command using only CMD line	
1	CMD_INHIBIT_DAT	RO	Command Inhibit (DAT) 1 : Cannot issue command wihich used the DAT line 0 : Can issue command using only DAT line	
2	DAT_LINE_ACTIVE	RO	DAT Line Active This bit indicates whether one of the DAT line on SD Bus is in use. 1 : DAT Line Active 0 : DAT Line Inactive	
3	RE_TUNE_REQ	RO	Re-Tuning Request 1 : Sampling clock need re-tuning 0 : Fixed or well tuned sampling clock	
7:4	Reserved			
8	WR_XFER_ACTIVE	RO	Write Transfer Active 1 : Transferring data 0 : No valid data	
9	RD_XFER_ACTIVE	RO	Read Transfer Active 1 : Transferring data	

Bits	Name	Access	Description	Reset
			0 : No valid data	
10	BUF_WR_ENABLE	RO	Buffer Write Enable 1 : Enable 0 : Disable	
11	BUF_RD_ENABLE	RO	Buffer Read Enable 1 : Enable 0 : Disable	
15:12	Reserved			
16	CARD_INSERTED	RO	Card Inserted 1 : Card Inserted 0 : Reset or Debouncing or No card	
17	CARD_STABLE	RO	Card State Stable 1 : No Card or Inserted 0 : Reset or Debouncing	
18	CARD_CD_STS	RO	Card Detect Pin Level 1 : Card Present (SD_CD = 0) 0 : No Card Present (SD_CD = 1)	
19	CARD_WP_STS	RO	Write Protect Switch Pin Level 1 : Write enabled (SD_WP =0) 0 : Write protected (SD_WP = 1)	
23:20	DAT_3_0_STS	RO	DAT[3:0] Line Signal Level	
24	CMD_LINE_STS	RO	CMD Line Signal Level	
31:25	Reserved			

HOST_CTL1_PWR_BG_WUP

Host Control 1 Register
Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	LEC_CTL	R/W	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. 1 : LED on 0 : LED off	0x0
1	DAT_XFER_WIDTH	R/W	Data Transfer Width. 1 : 4-bit mode 0 : 1-bit mode	0x0
2	HS_ENABLE	R/W	High Speed Enable 1 : High Speed Enable 0 : Normal Speed Enable	0x0
4:3	DMA_SEL	R/W	DMA Select. 0x0 : SDMA mode 0x1 : Reserved 0x2 : ADMA2 0x3 : ADMA2 or ADMA3	0x0
5	EXT_DAT_WIDTH	R/W	Extended Data Transfer Width 1 : 8-bit mode 0 : Selected by DAT_XFER_WIDTH	0x0
6	CRAD_DET_TEST	R/W	Card Detect Test Level 1 : Card Inserted 0 : No card	0x0
7	CARD_DET_SEL	R/W	Card Detect Signal Selection 1 : CARD_DET_TEST is selected (for test purpose) 0 : SD_CD Is selected	0x0
8	SD_BUS_PWR	R/W	SD Bus Power.	0x0

Bits	Name	Access	Description	Reset
			1 : Power on 0 : Power off	
11:9	SD_BUS_VOL_SEL	R/W	SD Bus Voltage Select 111b : 3.3V 110b : 3.0V 101b : 1.8V 100b - 000b : Reserved	0x0
15:12	Reserved			
16	STOP_BG_REQ	R/W	Stop At Block Gap Request. This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. 1 : Stop 0 : Transfer	0x0
17	CONTINUE_REQ	R/W	Continue Request. This bit is used to restart a transaction, which was stoped using the STOP_BG_REQ. 1 : Restart 0 : Not affect	0x0
18	READ_WAIT	R/W	Read Wait Control 1 : Enable Read Wait Control 0 : Disable Read Wait Control	0x0
19	INT_BG	R/W	Interrupt At Block Gap 1 : Enable 0 : Disabel	0x0
23:20	Reserved			
24	WAKEUP_ON_CARD_INT	R/W	Wakeup Event Enable On Card Interrupt. 1 : Enable 0 : Disable	0x0
25	WAKEUP_ON_CARD_INSERT	R/W	Wakeup Event Enable On Card Insertion. 1 : Enable 0 : Disable	0x0
26	WAKEUP_ON_CARD_REMV	R/W	Wakeup Event Enable On Card Removal. 1 : Enable 0 : Disable	0x0
31:27	Reserved			

CLK_CTL_SWRST

Clock and Timeout Control Register

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	INT_CLK_EN	R/W	Internal Clock Enable 1 : Oscillate 0 : Stop	0x0
1	INT_CLK_STABLE	RO	Internal Clock Stable. 1 : Ready 0 : Not Ready	
2	SD_CLK_EN	R/W	SD Clock Enable for Card 1 : Enable 0 : Disable	0x0
3	PLL_EN	R/W	PLL Enable 1 : Enable 0 : Disable	0x0

Bits	Name	Access	Description	Reset
5:4	Reserved			
7:6	UP_FREQ_SEL	R/W	Upper Bits of SDCLK Frequency Select	0x0
15:8	FREQ_SEL	R/W	SDCLK Frequency Select	0x0
19:16	TOUT_CNT	R/W	Data Timeout Counter Value 0x0 : TMCLK x 2^13 0x1 : TMCLK x 2^14 ..... 0xe : TMCLK x 2^ 27 0xf : Reserved	0x0
23:20	Reserved			
24	SW_RST_ALL	R/W	Software Reset For All	0x0
25	SW_RST_CMD	R/W	Software Reset For CMD Line	0x0
26	SW_RST_DAT	R/W	Software Reset For DATA Line	0x0
31:27	Reserved			

NORM_AND_ERR_INT_STS

Normal and Error Interrupt Status Register
Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CMD_CMPL	RWC	Command Complete	
1	XFER_CMPL	RWC	Transfer Complete	
2	BG_EVENT	RWC	Block Gap Event	
3	DMA_INT	RWC	DMA Interrupt	
4	BUF_WRDY	RWC	Buffer Write Ready	
5	BUF_RRDY	RWC	Buffer Read Ready	
6	CARD_INSERT_INT	RWC	Card Insertion	
7	CARD_REMOV_INT	RWC	Card Removal	
8	CARD_INT	RO	Card Interrupt	
9	INT_A	RO	INT_A. This status is set if INT_A is enabled and INT_A pin is in low level	
10	INT_B	RO	INT_B. This status is set if INT_B is enabled and INT_B pin is in low level	
11	INT_C	RO	INT_C. This status is set if INT_C is enabled and INT_C pin is in low level	
12	RE_TUNE_EVENT	RO	Re-Tuning Event	
13	Reserved			
14	CQE_EVENT	RO	Command Queuing Event	
15	ERR_INT	RO	Error Interrupt	
16	CMD_TOUT_ERR	RWC	Command Timeout Error	
17	CMD_CRC_ERR	RWC	Command CRC Error	
18	CMD_ENDBIT_ERR	RWC	Command End Bit Error	
19	CMD_IDX_ERR	RWC	Command Index Error	
20	DAT_TOUT_ERR	RWC	Data Timeout Error	
21	DAT_CRC_ERR	RWC	Data CRC Error	
22	DAT_ENDBIT_ERR	RWC	Data End Bit Error	

Bits	Name	Access	Description	Reset
23	CURR_LIMIT_ERR	RWC	Current Limit Error	
24	AUTO_CMD_ERR	RWC	Auto Command Error	
25	ADMA_ERR	RWC	ADMA Error	
26	TUNE_ERR	RWC	Tuning Error	
27	Reserved			
28	BOOT_ACK_ERR	RWC		
31:29	Reserved			

NORM_AND_ERR_INT_STS_EN

Normal and Error Interrupt Status Enable Register
Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	CMD_CMPL_EN	R/W	Command Complete Status Enable	0x0
1	XFER_CMPL_EN	R/W	Transfer Complete Status Enable	0x0
2	BG_EVENT_EN	R/W	Block Gap Event Status Enable	0x0
3	DMA_INT_EN	R/W	DMA Interrupt Status Enable	0x0
4	BUF_WRDY_EN	R/W	Buffer Write Ready Status Enable	0x0
5	BUF_RRDY_EN	R/W	Buffer Read Ready Status Enabel	0x0
6	CARD_INSERT_INT_EN	R/W	Card Insertion Status Enable	0x0
7	CARD_REMOV_INT_EN	R/W	Card Removal Status Enable	0x0
8	CARD_INT_EN	R/W	Card Interrupt Status Enable	0x0
9	INT_A_EN	R/W	INT_A Status Enable.	0x0
10	INT_B_EN	R/W	INT_B Status Enable.	0x0
11	INT_C_EN	R/W	INT_C Status Enable.	0x0
12	RE_TUNE_EVENT_EN	R/W	Re-Tuning Event Status Enable	0x0
13	Reserved			
14	CQE_EVENT_EN	R/W	Command Queuing Event Status Enable	0x0
15	ERR_INT_EN	R/W	Error Interrupt Status Enable	0x0
16	CMD_TOUT_ERR_EN	R/W	Command Timeout Error Status Enable	0x0
17	CMD_CRC_ERR_EN	R/W	Command CRC Error Status Enable	0x0
18	CMD_ENDBIT_ERR_EN	R/W	Command End Bit Error Status Enable	0x0
19	CMD_IDX_ERR_EN	R/W	Command Index Error Status Enable	0x0
20	DAT_TOUT_ERR_EN	R/W	Data Timeout Error Status Enable	0x0
21	DAT_CRC_ERR_EN	R/W	Data CRC Error Status Enable	0x0
22	DAT_ENDBIT_ERR_EN	R/W	Data End Bit Error Statue Enable	0x0
23	CURR_LIMIT_ERR_EN	R/W	Current Limit Error Status Enable	0x0
24	AUTO_CMD_ERR_EN	R/W	Auto Command Error Status Enable	0x0
25	ADMA_ERR_EN	R/W	ADMA Error Status Enable	0x0
26	TUNE_ERR_EN	R/W	Tuning Error Status Enable	0x0
27	Reserved			
28	BOOT_ACK_ERR_EN	R/W	Boot Ack Error Status Enable	0x0
31:29	Reserved			

NORM_AND_ERR_INT_SIG_EN

Normal and Error Interrupt Signal Enable Register
Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	CMD_CMPL_SIG_EN	R/W	Command Complete Signal Enable	0x0
1	XFER_CMPL_SIG_EN	R/W	Transfer Complete Signal Enable	0x0
2	BG_EVENT_SIG_EN	R/W	Block Gap Event Signal Enable	0x0
3	DMA_INT_SIG_EN	R/W	DMA Interrupt Signal Enable	0x0
4	BUF_WRDY_SIG_EN	R/W	Buffer Write Ready Signal Enable	0x0

Bits	Name	Access	Description	Reset
5	BUF_RRDY_SIG_EN	R/W	Buffer Read Ready Signal Enabel	0x0
6	CARD_INSERT_INT_SIG_EN	R/W	Card Insertion Signal Enable	0x0
7	CARD_REMOV_INT_SIG_EN	R/W	Card Removal Signal Enable	0x0
8	CARD_INT_SIG_EN	R/W	Card Interrupt Signal Enable	0x0
9	INT_A_SIG_EN	R/W	INT_A Signal Enable.	0x0
10	INT_B_SIG_EN	R/W	INT_B Signal Enable.	0x0
11	INT_C_SIG_EN	R/W	INT_C Signal Enable.	0x0
12	RE_TUNE_EVENT_SIG_EN	R/W	Re-Tuning EventSignal Enable	0x0
13	Reserved			
14	CQE_EVENT_SIG_EN	R/W	CQE EventSignal Enable	0x0
15	Reserved			
16	CMD_TOUT_ERR_SIG_EN	R/W	Command Timeout Error Signal Enable	0x0
17	CMD_CRC_ERR_SIG_EN	R/W	Command CRC Error Signal Enable	0x0
18	CMD_ENDBIT_ERR_SIG_EN	R/W	Command End Bit Error Signal Enable	0x0
19	CMD_IDX_ERR_SIG_EN	R/W	Command Index Error Signal Enable	0x0
20	DAT_TOUT_ERR_SIG_EN	R/W	Data Timeout Error Signal Enable	0x0
21	DAT_CRC_ERR_SIG_EN	R/W	Data CRC Error Signal Enable	0x0
22	DAT_ENDBIT_ERR_SIG_EN	R/W	Data End Bit Error Signal Enable	0x0
23	CURR_LIMIT_ERR_SIG_EN	R/W	Current Limit Error Signal Enable	0x0
24	AUTO_CMD_ERR_SIG_EN	R/W	Auto Command Error Signal Enable	0x0
25	ADMA_ERR_SIG_EN	R/W	ADMA Error Signal Enable	0x0
26	TUNE_ERR_SIG_EN	R/W	Tuning Error Signal Enable	0x0
27	Reserved			
28	BOOT_ACK_ERR_SIG_EN	R/W	Boot Ack Error Signal Enable	0x0
31:29	Reserved			

AUTO_CMD_ERR_AND_HOST_CTL2

Auto CMD Error Status Register and Host Control 2 register
Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	AUTO_CMD12_NO_EXE	RO	Auto CMD12 Not Executed	
1	AUTO_CMD_TOUT_ERR	RO	Auto CMD Timeout Error	
2	AUTO_CMD_CRC_ERR	RO	Auto CMD CRC Error	
3	AUTO_CMD_ENDBIT_ERR	RO	Auto CMD End Bit Error	
4	AUTO_CMD_IDX_ERR	RO	Auto CMD Index Error	
6:5	Reserved			
7	CMD_NOT_ISSUE_BY_CMD12	RO	Command Not Issued By Auto CMD12 Error	
15:8	Reserved			
18:16	UHS_MODE_SEL	R/W	USH Speed Mode Select (for SD) 0x0 : SDR12 0x1 : SDR25 0x2 : SDR50 0x3 : SDR104 0x4 : DDR50 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved eMMC Speed Mode Select (for eMMC) 0x0 : Default speed 0x1 : High speed 0x2 : Reserved 0x3 : HS200	0x0

Bits	Name	Access	Description	Reset
			0x4 : DDR52 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved	
19	EN_18_SIG	R/W	1.8V Signaling Enable	0x0
21:20	DRV_SEL	R/W	Driver Strength Select 0x0 : Driver Type B 0x1 : Driver Type A 0x2 : Driver Type C 0x3 : Driver Type D	0x0
22	EXECUTE_TUNE	R/W	Execute Tuning 1 : Execute Tuning 0 : Not Tuned or Tuning Completed	0x0
23	SAMPLE_CLK_SEL	R/W	Sampling Clock Select 1 : Tuned clock is used to sample data 0 : Fixed clock is used to sample data	0x0
29:24	Reserved			
30	ASYNC_INT_EN	R/W	Asynchronous Interrupt Enable. 1 : Enable 0 : Disable	0x0
31	PRESET_VAL_ENABLE	R/W	Preset Value Enable 1 : Automatic Selection by Preset Value are Enabled 0 : SDLCK and Driver Strength are controlled by Host Driver	0x0

CAPABILITIES1

Capabilities 1 Register
Offset Address: 0x040

Bits	Name	Access	Description	Reset
5:0	TOUT_CLK_FREQ	RO	Timeout Clock Frequency Not 0 : 1KHz~ 63KHz or 1Mhz~ 63Mhz	
6	Reserved			
7	TOUT_CLK_UNIT	RO	Timeout Clock Unit 1 : 1MHz 0 : 1KHz	
15:8	BASE_CLK_FREQ	RO	Base Clock Frequency for SD clock 0x0 : Get information through another method 0x1 : 1MHz 0x2 : 2MHz ..... 0xFF : 255Mhz	
17:16	MAX_BLK_LEN	RO	Max Block Length 0x0 : 512 (byte) 0x1 : 1024 0x2 : 2048 0x3 : Reserverd	
18	EMBEDDED_8BIT	RO	8-bit Support for Embedded Device	
19	ADMA2_SUPPORT	RO	ADMA2 Support	
20	Reserved			
21	HS_SUPPORT	RO	High Speed Support	
22	SDMA_SUPPORT	RO	SDMA Support	
23	SUSP_RES_SUPPORT	RO	Suspend/Resume Support	

Bits	Name	Access	Description	Reset
24	V33_SUPPORT	RO	3.3V Support	
25	V30_SUPPORT	RO	3.0V Support	
26	V18_SUPPORT	RO	1.8V Support	
27	Reserved			
28	BUS64_SUPPORT	RO	64-bit System Bus Support	
29	ASYNC_INT_SUPPORT	RO	Asynchronous Interrupt Support	
31:30	SLOT_TYPE	RO	Slot Type 0x0 : Removable Card 0x1 : Embedded Slot 0x2 : Shared Bus Slot	

CAPABILITIES2

Capabilities 2 Register
Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	SDR50_SUPPORT	RO	SDR50 Support	
1	SDR104_SUPPORT	RO	SDR104 Support	
2	DDR50_SUPPORT	RO	DDR50 Support	
3	Reserved			
4	DRV_A_SUPPORT	RO	Driver Type A Support	
5	DRV_C_SUPPORT	RO	Driver Type C Support	
6	DRV_D_SUPPORT	RO	Driver Type D Support	
7	Reserved			
11:8	RETUNE_TIMER	RO	Timer Count for Re-Tuning 0x0 : Disable n : 2^(n-1) seconds 0xB : 1024 seconds 0xC ~ 0xE : Reserved 0xF : Get Information from other source	
12	Reserved			
13	TUNE_SDR50	RO	Use Tuning for SDR50	
15:14	RETUNE_MODE	RO	Re-Tuning Modes	
23:16	CLK_MULTIPLIER	RO	Clock Multiplier	
31:24	Reserved			

FORCE_EVENT_ERR

Force Event Register for Auto CMD Error Status
Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	FORCE_AUTO_CMD12_NOT_EXE	R/W	Force Event for Auto CMD12 Not Executed	0x0
1	FORCE_AUTO_CMD_TOUT_ERR	R/W	Force Event for Auto CMD Timeout Error	0x0
2	FORCE_AUTO_CMD_CRC_ERR	R/W	Force Event for Auto CMD CRC Error	0x0
3	FORCE_AUTO_CMD_EBIT_ERR	R/W	Force Event for Auto CMD End Bit Error	0x0
4	FORCE_AUTO_CMD_IDX_ERR	R/W	Force Event for Auto CMD Index Error	0x0
6:5	Reserved			
7	FORCE_AUTO_CMD_NOT_ISSUE	R/W	Force Event for Command Not Issued By Auto CMD12 Error	0x0

Bits	Name	Access	Description	Reset
15:8	Reserved			
16	FORCE_CMD_TOUT_ERR	R/W	Force Event for Auto CMD12 Not Executed	0x0
17	FORCE_CMD_CRC_ERR	R/W	Force Event for CMD Timeout Error	0x0
18	FORCE_CMD_EBIT_ERR	R/W	Force Event for CMD End Bit Error	0x0
19	FORCE_CMD_IDX_ERR	R/W	Force Event for CMD Index Error	0x0
20	FORCE_DAT_TOUT_ERR	R/W	Force Event for DATA Timeout Error	0x0
21	FORCE_DAT_CRC_ERR	R/W	Force Event for DATA End Bit Error	0x0
22	FORCE_DAT_EBIT_ERR	R/W	Force Event for DATA Index Error	0x0
23	FORCE_CURR_LIMIT_ERR	R/W	Force Event for current limit error	0x0
24	FORCE_AUTO_CMD_ERR	R/W	Force Event for Auto CMD Error	0x0
25	FORCE_ADMA_ERR	R/W	Force Event for ADMA Error	0x0
26	FORCE_TUNING_ERR	R/W	Force Event for Tuning Error	0x0
27	Reserved			
28	FORCE_BOOT_ACK_ERR	R/W	Force Event for Response Error	0x0
31:29	Reserved			

ADMA_ERR_STS

ADMA Error Status Register
Offset Address: 0x054

Bits	Name	Access	Description	Reset
1:0	ADMA_ERR_STS	RO	ADMA Error Status 0x0 : ST_STOP (Stop DMA) 0x1 : ST_FDS (Fetch Descriptor) 0x2 : Never set this state 0x3 : ST_TFR (transfer data)	
2	ADMA_LEN_MISMATCH	RO	ADMA Length Mismatch Error	
31:3	Reserved			

ADMA_SADDR_L

ADMA System Address Register for low 32-bit
Offset Address: 0x058

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_L	R/W	ADMA System Address for low 32-bit	0x0

ADMA_SADDR_H

ADMA System Address Register for high 32-bit
Offset Address: 0x05c

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_H	R/W	ADMA System Address for high 32-bit	0x0

PRESENT_VUL_INIT_DS

Present Value Register for Initialization and Default Speed
Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_INIT_DS	RO	Present Value Register for Initialization and Default Speed	

PRESENT_VUL_HS_SDR12

Present Value Register for High-speed and SDR12
Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_HS_SDR12	RO	Present Value Register for High-speed	

Bits	Name	Access	Description	Reset
			and SDR12	

PRESENT_VUL_SDR25_SDR50

Present Value Register for SDR25 and SDR50
Offset Address: 0x068

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR25_SDR50	RO	Present Value Register for SDR25 and SDR50	

PRESENT_VUL_SDR104_DDR50

Present Value Register for SDR104 and DDR50
Offset Address: 0x06c

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR104_DDR50	RO	Present Value Register for SDR104 and DDR50	

SLOT_INT_AND_HOST_VER

Slot Interrupt Status and Host Controller Version Register
Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
7:0	INT_SLOT	RO	Interrupt Signal for Each Slot	
15:8	Reserved			
23:16	SPEC_VER	RO	Specification Version Number 00h : SD Host 1.00 01h : SD Host 2.00 02h : SD Host 3.00 03h : SD Host 4.00 04h : SD Host 4.10 05h : SD Host 4.20	
31:24	VENDOR_VER	RO	Verdor Version Number	

EMMC_CTRL

MSHC Control register
Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	EMMC_FUNC_EN	R/W	eMMC Card present	0x0
1	LATANCY_1T	R/W	Latancy 1t for cmd in	0x1
2	CLK_FREE_EN	R/W	Internal clock gating disable control	0x0
3	DISABLE_DATA_CRC_CHK	R/W	Disable Data CRC Check	0x0
7:4	Reserved			
8	EMMC_RSTN	R/W	EMMC Device Reset Signal control	0x1
9	EMMC_RSTN_OEN	R/W	Output Enable control for EMMC Device Reset Signal PAD	0x1
11:10	Reserved			
12	CQE_ALGO_SEL	R/W	Scheduler algorithm selected for execution 1 : First come First serve (FCFS_ONLY) 0 : Priority based reordering with FCFS (PRI_REORDER_PLUS_FCFS)	0x0
13	CQE_PREFETCH_DISABLE	R/W	Enable or Disable CQE's PREFETCH Feature 1 : Disable 0 : Enable	0x0

Bits	Name	Access	Description	Reset
15:14	Reserved			
16	timer_clk_sel	R/W	timer clock source selection 1 : 32K 0 : 100K	0x0
31:17	Reserved			

EMMC_BOOT_CTL

eMMC Boot Control Register
Offset Address: 0x204

Bits	Name	Access	Description	Reset
0	BOOT_MODE_ENABLE	R/W	Mandatory Boot Enable	0x0
1	BOOT_ACK_ENABLE	R/W	Boot Ack Enable	0x0
3:2	Reserved			
7:4	BOOT_TOUT_CNT	R/W	Boot Ack Timeout Counter Value	0x0
8	VALIDATE_BOOT	W	Validate Mandatory Boot Enable Bit	
31:9	Reserved			

CDET_TOUT_CTL

Card Detect Control Register
Offset Address: 0x208

Bits	Name	Access	Description	Reset
15:0	CDET_DEBUUNCE_CNT	R/W	card detect debounce counter	0x000F
31:16	Reserved			

MBIU_CTRL

MBIU Control register
Offset Address: 0x20c

Bits	Name	Access	Description	Reset
0	UNDEFL_INCR_EN	R/W	Undefined INCR Burst	0x1
1	BURST_INCR4_EN	R/W	INCR4 Burst	0x1
2	BURST_INCR8_EN	R/W	INCR8 Burst	0x1
3	BURST_INCR16_EN	R/W	INCR16 Burst	0x1
31:4	Reserved			

PHY_TX_RX_DLY

PHY tx and rx delay line register
Offset Address: 0x240

Bits	Name	Access	Description	Reset
6:0	PHY_TX_DLY	R/W	PHY tx delay line phase selection	0x0
7	Reserved			
9:8	PHY_TX_SRC	R/W	PHY tx delay line clock source selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx 2'b1x : reserved	0x0
10	PHY_TX_EVEN_ODD	R/W	PHY tx delay line clock source selection	0x0
15:11	Reserved			
22:16	PHY_RX_DLY	R/W	PHY rx delay line phase selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx 2'b1x : reserved	0x0
23	Reserved			
25:24	PHY_RX_SRC	R/W	PHY rx delay line clock source selection	0x0

Bits	Name	Access	Description	Reset
26	PHY_RX_EVEN_ODD	R/W	PHY rx delay line clock source selection	0x0
31:27	Reserved			

PHY_DS_DLY

PHY DS delay line register
Offset Address: 0x244

Bits	Name	Access	Description	Reset
6:0	PHY_DS_DLY	R/W	PHY DS delay line phase selection	0x0
7	Reserved			
9:8	PHY_DS_SRC	R/W	PHY DS delay line clock source selection	0x0
10	PHY_DS_EVEN_ODD	R/W	PHY DS delay line clock source selection	0x0
31:11	Reserved			

PHY_DLY_STS

PHY delay line status register
Offset Address: 0x248

Bits	Name	Access	Description	Reset
0	PHY_TX_LEAD_LAG	RO	PHY tx delay line lead or lag flag	
1	PHY_RX_LEAD_LAG	RO	PHY rx delay line lead or lag flag	
2	PHY_DS_LEAD_LAG	RO	PHY ds delay line lead or lag flag	
31:3	Reserved			

PHY_CONFIG

PHY Configuration register
Offset Address: 0x24c

Bits	Name	Access	Description	Reset
0	PHY_TX_BPS	R/W	PHY tx data path bypass enable 0 : Pipe enable 1 : Bypass	0x1
1	ADJ_TIMING_EN	R/W	Adjust bus timing enable	0x0
7:2	Reserved			
9:8	ADJ_NCR	R/W	Adjust NCR counter	0x0
11:10	ADJ_NCRC	R/W	Adjust NCRC counter	0x0
31:12	Reserved			

18.5 GPIO

18.5.1 概述

系统配置 4 组 GPIO (General Purpose Input/Output)，GPIO0 ~ GPIO3。每组GPIO 提供 32 个可编程的输入输出管脚。
每一个管脚的方向可以任意设置为输入或者输出，用于生成特定应用的输出信号或收集特定应用的输入信号。设置为输入管脚时，GPIO 可用作中断源；设置为输出管脚时，每一个 GPIO 都可以独立地输出 0 或 1。
GPIO 可以根据输入信号的电平或跳变值产生可屏蔽的中断。GPIOx_INTR_FLAG

（x=0~3）信号给中断控制器一个指示，表示有中断发生。

18.5.2 特点

每一个管脚的方向可以任意设置为输入或者输出。

- 设置为输入管脚时，GPIO 可用作中断源。
- 设置为输出管脚时，每一个 GPIO 都可以独立地输出 0 或 1。

18.5.3 工作方式

18.5.3.1 接口复位

在芯片上电或系统复位时，4 个 GPIO 模块会同时被复位，GPIO 管脚在复位之后默认处于输入状态。

18.5.3.2 通用输入输出

每个管脚可以任意设置为输入或者输出，步骤如下：

步骤 1 配置寄存器 GPIO_SWPORTA_DDR，设置 GPIO 是作为输入还是输出。

步骤 2 配置成输入管脚时，读取 GPIO_EXT_PORTA 寄存器可查看输入信号值；配置成输出管脚时，向 GPIO_SWPORTA_DR 寄存器写入输出值可控制GPIO 输出电平。

18.5.3.3 中断操作

每一个 GPIO 都可以用作中断源，通过 GPIO_INTEN 等 9 个寄存器进行控制。通过这些寄存器用户可以选择中断来源、中断电平极性以及边沿触发特性。

多个 GPIO 中断同时发生时，将会统一汇集成一个中断进行上报 (4 组 GPIO 各自会有一个汇集中断旗标上报) 。

中断源的特性和中断触发类别由 GPIO_INTTYPE_LEVEL、GPIO_INT_POLARITY、GPIO_INTMASK、GPIO_DEBOUNCE、GPIO_LS_SYNC 五个寄存器决定。

中断的原始状态和屏蔽后的状态通过 GPIO_RAW_INTSTATUS 和 GPIO_INTSTATUS 读取。

通过设置 GPIO_PORTA_EOI 可控制中断状态的清除。

每一个 GPIO 都可以支持中断，设置步骤如下：

步骤 1 配置寄存器 GPIO_INTTYPE_LEVEL，选择电平触发或边沿触发。

步骤 2 配置寄存器 GPIO_INT_POLARITY，选择低电平/高电平触发和下降沿/上升沿触发。

步骤 3 对寄存器 GPIO_PORTA_EOI 写入 0xFFFFFFFF，清除中断。

步骤 4 配置 GPIO_INTEN 寄存器；使能 GPIO 管脚中断功能。

18.5.4 GPIO 寄存器概览

芯片的 4 组 GPIO 模块基地址如表格 18-12 所示。

表格 18-12 芯片的 4 组 GPIO 模块基地址

GPIO 模块	基地址
GPIO0	0x03020000
GPIO1	0x03021000
GPIO2	0x03022000
GPIO3	0x03023000
RTCSYS_GPIO	0x05021000

表格 18-13 是第 1 组 GPIO 模块 (GPIO0) 寄存器的偏移地址以及定义，GPIO0～GPIO3 具有相同的寄存器定义。

表格 18-13 GPIO 寄存器概览

Name	Address Offset	Description
GPIO_SWPORTA_DR	0x000	Port A data register
GPIO_SWPORTA_DDR	0x004	Port A data direction register
GPIO_INTEN	0x030	Interrupt enable register
GPIO_INTMASK	0x034	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x038	Interrupt level register
GPIO_INT_POLARITY	0x03c	Interrupt polarity register
GPIO_INTSTATUS	0x040	Interrupt status of Port A
GPIO_RAW_INTSTATUS	0x044	Raw interrupt status of Port A (pre-masking)
GPIO_DEBOUNCE	0x048	Debounce enable register
GPIO_PORTA_EOI	0x04c	Port A clear interrupt register
GPIO_EXT_PORTA	0x050	Port A external port register
GPIO_LS_SYNC	0x060	Level-sensitive synchronization enable register

18.5.5 GPIO 寄存器描述

GPIO_SWPORTA_DR

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DR	R/W	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode and the corresponding control bit for Port A is set to Software mode. The value read back is equal to the last value written to this register.	0x0

GPIO_SWPORTA_DDR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DDR	R/W	Values written to this register independently control the direction of the corresponding data bit in Port A. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_A parameter. 0 – Input (default) 1 – Output	0x0

GPIO_INTEN

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	GPIO_INTEN	R/W	Allows each bit of Port A to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output or if Port A mode is set to Hardware. 0 – Configure Port A bit as normal GPIO signal (default) 1 – Configure Port A bit as interrupt	0x0

GPIO_INTMASK

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	GPIO_INTMASK	R/W	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking. 0 – Interrupt bits are unmasked (default) 1 – Mask interrupt	0x0

GPIO_INTTYPE_LEVEL

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	GPIO_INTTYPE_LEVEL	R/W	Controls the type of interrupt that can occur on Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive.	0x0

Bits	Name	Access	Description	Reset
			0 – Level-sensitive (default) 1 – Edge-sensitive	

GPIO_INT_POLARITY

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	GPIO_INT_POLARITY	R/W	Controls the polarity of edge or level sensitivity that can occur on input of Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 0 – Active-low (default) 1 – Active-high	0x0

GPIO_INTSTATUS

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	GPIO_INTSTATUS	RO	Interrupt status of Port A	

GPIO_RAW_INTSTATUS

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	GPIO_RAW_INTSTATUS	RO	Raw interrupt of status of Port A (premasking bits)	

GPIO_DEBOUNCE

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	GPIO_DEBOUNCE	R/W	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0 – No debounce (default) 1 – Enable debounce	0x0

GPIO_PORTA_EOI

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	GPIO_PORTA_EOI	R/W	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0 – No interrupt clear (default) 1 – Clear interrupt	0x0

GPIO_EXT_PORTA

Offset Address: 0x050

Bits	Name	Access	Description	Reset
31:0	GPIO_EXT_PORTA	RO	When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.	

GPIO_LS_SYNC

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	GPIO_LS_SYNC	R/W	[0] Synchronization level Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0 – No synchronization to pclk_intr (default) 1 – Synchronize to pclk_intr	0x0
31:1	Reserved			

18.6 USB DRD

18.6.1 概述

USB DRD 的功能为可分别担任 Host 或是 Device 的角色，可以通过软件的设定来改变，传输协议符合 USB 2.0 规格，最高传输速率可达 40MB/s 以上；Host/Device 的主要运作模式皆为分散/聚集 DMA 传输(scatter gather DMA)，详细情况会分别在 Host 与 Device 的章节中描述；USB DRD 的功能简列如下：

- 控制传输 (Control Transfer)
- 批量传输 (Bulk Transfer)
- 实时传输 (Isochronous Transfer)
- Host 可连接 USB Hub，并且支持中断传输(Interrupt Transfer)
- 通过 USB 电器特性测试(USBET)，讯号质量、兼容性良好

18.6.2 功能描述

18.6.2.1 系统框图

下方图片显示 USB DRD 内部的系统框图：

图表 18-42 USB DRD 内部的系统框图

18.6.2.2 功能特征

USB DRD 的功能简列如下：

. 符合 USB2.0 传输协议规范
. 向下兼容 USB1.1 传输协议规范
. 支持 HS/FS/LS 三种速度模式
. 支持 Host 或者是 Device 功能
. 支持四种 USB 传输协订所规范的传输型态：控制传输、批量传输、实时传输、中断传输
. 可连接 USB Hub，将单一接口扩展为多个 USB 接口
. 透过 USB Hub 扩展，最多可连接 127 Device 装置
. 支持 USB2.0 休眠/回复(suspend/resume)省电模式
. 支持键盘、鼠标等 HID 装置
. Device mode 主要用于下载更新内部软件之用，也可做其他功能，例如数据传输之用
. 最大传输速率可达 40MB/s 以上

18.6.3 USBC 功能与寄存器描述

18.6.3.1 USBC 功能描述

USB DRD 可以切换 Host 或是 Device 功能，可以择其一使用，但不能同时存在工作，其功能选择与管理就是靠 USBC 此区块来控制；另外 Host 与 device 有一些 Serial Bus 上面的事件与中断触发，也会将缓存器放置在此一区块。

18.6.3.2 USBC 寄存器摘要

Name	Address Offset	Description
GOTGCTL	0x000	Control and Status Register
GOTGINT	0x004	Interrupt Register
GAHBCFG	0x008	AHB Configuration Register
GUSBCFG	0x00c	USB Configuration Register
GRSTCTL	0x010	Reset Register
GINTSTS	0x014	Interrupt Status Register
GINTMSK	0x018	Interrupt Mask Register
GUID	0x03c	User ID Register
GLPMCFG	0x054	Core LPM Configuration Register
GPWRDN	0x058	Power Down Register

18.6.3.3 USBC 寄存器详细列表

USBC 的内存寻址位置为 0x0434_0000，在文中就以 USBC_BASE_ADDR 来表示，若要对其进行读写控制，则在内存空间中真实的寻址位置就会以 USBC_BASE_ADDR + Offset

来表示；每个寄存器都有其相对应的相对寻址(Offset)，详细内容如下方所描述。

GOTGCTL

Control and Status Register
Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	SesReqScs	RO	Mode: Device only Session Request Success (SesReqScs) The core sets this bit when a session request initiation is successful. ■ 1'b0: Session request failure ■ 1'b1: Session request success Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	SesReq	R/W	Mode: SRP-capable device Session Request (SesReq) The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is SET. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. ■ 1'b0: No session request ■ 1'b1: Session request Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	VbvalidOvEn	R/W	Mode: Host only VBUS Valid Override Enable (VbvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.VbvalidOvVal. ■ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.VbvalidOvVal. ■ 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the core. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
3	VbvalidOvVal	R/W	Mode: Host only VBUS Valid Override Value (VbvalidOvVal) This bit is used to set Override value for vbusvalid signal when GOTGCTL.VbvalidOvEn is set. ■ 1'b0: vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn =1	0x0

Bits	Name	Access	Description	Reset
			■ 1'b1: vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn =1	
4	AvalidOvEn	R/W	Mode: Host only A-Peripheral Session Valid Override Enable (AvalidOvEn) This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal. ■ 1'b1: Internally Avalid received from the PHY is overridden with GOTGCTL.AvalidOvVal. ■ 1'b0: Override is disabled and avalid signal from the respective PHY selected is used internally by the core.	0x0
5	AvalidOvVal	R/W	Mode: Host only A-Peripheral Session Valid Override Value (AvalidOvVal) This bit is used to set Override value for Avalid signal when GOTGCTL.AvalidOvEn is set. ■ 1'b0: Avalid value is 1'b0 when GOTGCTL.AvalidOvEn =1 ■ 1'b1: Avalid value is 1'b1 when GOTGCTL.AvalidOvEn =1	0x0
6	BvalidOvEn	R/W	Mode: Device only B-Peripheral Session Valid Override Enable (BvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal. ■ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal. ■ 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the force	0x0
7	BvalidOvVal	R/W	Mode: Device only B-Peripheral Session Valid Override Value (BvalidOvVal) This bit is used to set Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set. ■ 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn =1 ■ 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn =1	0x0
8	HstNegScs	RO	Mode: HNP-capable device Host Negotiation Success (HstNegScs) The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. ■ 1'b0: Host negotiation failure ■ 1'b1: Host negotiation success	
9	HNPReq	R/W	Mode: HNP Capable OTG Device HNP Request (HNPReq) The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared.	0x0

Bits	Name	Access	Description	Reset
			■ 1'b0: No HNP request ■ 1'b1: HNP request	
10	HstSetHNPEn	R/W	Mode: HNP Capable OTG Host Host Set HNP Enable (HstSetHNPEn) The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. ■ 1'b0: Host Set HNP is not enabled ■ 1'b1: Host Set HNP is enabled	0x0
11	DevHNPEn	R/W	Mode: HNP Capable OTG Device Device HNP Enabled (DevHNPEn) The application sets this bit when it successfully receives aSetFeature.SetHNPEnable command from the connected USB host. ■ 1'b0: HNP is not enabled in the application ■ 1'b1: HNP is enabled in the application	0x0
12	EHEn	R/W	Embedded Host Enable It is used to select between OTG A Device state Machine andEmbedded Host state machine. ■ 1'b1: Embedded Host State Machine is selected ■ 1'b0: OTG A Device state machine is selected Note: This field is valid only in SRP-Capable OTG Mode (OTG_MODE=0,1)	0x0
14:13	Reserved_00_14_13	RO	Reserved for future use.	
15	DbnceFltrBypass	R/W	Mode: Host and Device Debounce Filter Bypass Bypass Debounce filters for avalid, bvalid, vbusvalid, sessend, and iddig signals when enabled. ■ 1'b0: Disabled ■ 1'b1: Enabled	0x0
16	ConIDSts	RO	Mode: Host and Device Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. ■ 1'b0: The DWC_otg core is in A-Device mode ■ 1'b1: The DWC_otg core is in B-Device mode	
17	DbncTime	RO	Mode: Host only Long/Short Debounce Time (DbncTime) Indicates the debounce time of a detected connection. ■ 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 µs) ■ 1'b1: Short debounce time, used for soft connections (2.5 µs)	
18	ASesVld	RO	Mode: Host only A-Session Valid (ASesVld) Indicates the Host mode transceiver status. ■ 1'b0: A-session is not valid ■ 1'b1: A-session is valid	
19	BSesVld	RO	Mode: Device only B-Session Valid (BSesVld) Indicates the Device mode transceiver status.	

Bits	Name	Access	Description	Reset
			■ 1'b0: B-session is not valid. ■ 1'b1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected.	
20	OTGVer	R/W	OTG Version (OTGVer) Indicates the OTG revision. ■ 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. ■ 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.	0x0
21	CurMod_operation	RO	Current Mode of Operation (CurMod) Mode: Host and Device Indicates the current mode. ■ 1'b0: Device mode ■ 1'b1: Host mode	
26:22	MultValIdBC_operation	RO	Multi Valued ID pin (MultValIdBC) Mode: Host and Device Battery Charger ACA inputs in the following order: ■ Bit 26 - rid_float ■ Bit 25 - rid_gnd ■ Bit 24 - rid_a ■ Bit 23 - rid_b ■ Bit 22 - rid_c	
27	ChirpEn	RO	Chirp On Enable (ChirpEn) Mode: Device Only This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT!=1, this bit is a reserved bit.	
31:28	Reserved_00_31_28	RO	Reserved for future use.	

GOTGINT

Interrupt Register
Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	Reserved_04_1_0	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	SesEndDet	RWC	Write Behavior: One to clear Mode: Host and Device Session End Detected (SesEndDet) The core sets this bit when the utmiotg_bvalid signal is deasserted.This bit can be set only by the core and the application should write 1 to clear it.	
7:3	Reserved_04_7_3	RO	Reserved for future use.	

Bits	Name	Access	Description	Reset
8	SesReqSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Session Request Success Status Change (SesReqSucStsChng) The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
9	HstNegSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Success Status Change (HstNegSucStsChng) The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
16:10	Reserved_04_16_10	RO	Reserved for future use.	
17	HstNegDet	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Detected (HstNegDet) The core sets this bit when it detects a host negotiation request on the USB.This bit can be set only by the core and the application should write 1 to clear it.	
18	ADevTOUTChg	RWC	Write Behavior: One to clear Mode: Host and Device A-Device Timeout Change (ADevTOUTChg) The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.This bit can be set only by the core and the application should write 1 to clear it.	
19	DbnceDone	RWC	Write Behavior: One to clear Mode: Host only Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).This bit can be set only by the core and the application	

Bits	Name	Access	Description	Reset
			should write 1 to clear it.	
20	MultValIpChng	RWC	Write Behavior: One to clear This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT=1, otherwise it is reserved.	
31:21	Reserved_04_31_21	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GAHBCFG

AHB Configuration Register
Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	GlblIntrMsk	R/W	Mode: Host and device Global Interrupt Mask (GlblIntrMsk) The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. ■ 1'b0: Mask the interrupt assertion to the application. ■ 1'b1: Unmask the interrupt assertion to the application. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
4:1	HBstLen	R/W	Mode: Host and device Burst Length/Type (HBstLen) This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, which can be used by an external wrapper to interface the External DMA Controller interface to Synopsys DW_ahb_dmac or ARM PrimeCell. External DMA Mode defines the DMA burst length in terms of 32-bit words: ■ 4'b0000: 1 word ■ 4'b0001: 4 words ■ 4'b0010: 8 words ■ 4'b0011: 16 words ■ 4'b0100: 32 words ■ 4'b0101: 64 words ■ 4'b0110: 128 words ■ 4'b0111: 256 words ■ Others: Reserved Internal DMA Mode-AHB Master burst type: ■ 4'b0000 Single ■ 4'b0001 INCR 4'b0011 ■ INCR4 4'b0101 ■ INCR8 4'b0111 ■ INCR16	0x0

Bits	Name	Access	Description	Reset
			■ Others: Reserved	
5	DMAEn	R/W	Mode: Host and device DMA Enable (DMAEn) ■ 1'b0: Core operates in Slave mode ■ 1'b1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.	0x0
6	Reserved_08_6	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
7	NPTxFEmpLvl	R/W	Mode: Host and device Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: ■ 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is half empty ■ 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is completely empty Dedicated FIFO in device mode: ■ 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty ■ 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty	0x0
8	PTxFEmpLvl	R/W	Mode: Host only Periodic TxFIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. ■ 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty ■ 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	0x0
20:9	Reserved_08_20_9	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
21	RemMemSupp	R/W	Mode: Host and Device Remote Memory Support (RemMemSupp)	0x0

Bits	Name	Access	Description	Reset
			This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. ■ GAHBCFG.RemMemSupp=1 The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. ■ GAHBCFG.RemMemSupp=0 The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it doesn't wait for the sys_dma_done signal to complete the DATA transfers.	
22	NotiAllDmaWrit	R/W	Mode: Host and Device Notify all DMA Write Transactions (NotiAllDmaWrit) This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. ■ GAHBCFG.NotiAllDmaWrit = 1 DWC_otg core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. ■ GAHBCFG.NotiAllDmaWrit = 0 DWC_otg core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.	0x0
23	AHBSingle	R/W	Mode: Host and Device AHBSingleSupport (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the DWC_otg core is operating in DMA mode.	0x0

Bits	Name	Access	Description	Reset
			■ 1' b0: This is the default mode. When this bit is set to 1'b0, the remaining data in the transfer is sent using INCR burst size. ■ 1' b1: When set to 1'b1, the remaining data in a transfer is sent using Single burst size. Note: If this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the DWC_otg core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus.)	
24	InvDescEndiannes s	R/W	Mode: Host and Device Inverse Descriptor Endianness ■ 1'b0: Descriptor endianness is similar to the AHB Master endianness ■ 1'b1: - If the AHB Master endianness is Big Endian, the Descriptor Endianness is Little Endian. - If the AHB Master endianness is Little Endian, the Descriptor Endianness is Big Endian.	0x0
31:25	Reserved_08_31_ 25	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GUSBCFG

USB Configuration Register
Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	TOutCal	R/W	Mode: Host and Device HS/FS Timeout Calibration (TOutCal) The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for fullspeed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: ■ One 30-MHz PHY clock = 16 bit times ■ One 60-MHz PHY clock = 8 bit times Full-speed operation: ■ One 30-MHz PHY clock = 0.4 bit times	0x0

Bits	Name	Access	Description	Reset
			■ One 60-MHz PHY clock = 0.2 bit times ■ One 48-MHz PHY clock = 0.25 bit times Using the HS as an example, if you set ToutCal to '001' you add one 30MHz PHY clock or 16 bit times. If you set ToutCal to '010' you add two 30MHz PHY clocks or 32 bit times, and so on. The 3 bits allow you to add up to 7 PHY clocks, and the number of bit times depend on the speed, and the PHY clock you are using. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	PHYIf	RO	Mode: Host and Device PHY Interface (PHYIf) The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. ■ 1'b0: 8 bits ■ 1'b1: 16 bits This bit is writable only If UTMI+ and ULPI were selected. Otherwise, this bit returns the value for the power-on interface selected during configuration. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
4	ULPI_UTMI_Sel	R/W	Mode: Host and Device ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select either a UTMI+ interface or ULPI Interface. ■ 1'b0: UTMI+ Interface ■ 1'b1: ULPI Interface This bit is writable only If UTMI+ and ULPI was specified for High- Speed PHY Interface(s). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
5	FSIntf	R/W	Mode: Host and Device Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. ■ 1'b0: 6-pin unidirectional full-speed serial interface ■ 1'b1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a USB 1.1 FS interface was selected, then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.	0x0
6	PHYSel	R/W	Mode: Host and Device	0x0

Bits	Name	Access	Description	Reset
			USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. ■ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY ■ 1'b1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a high-speed PHY interface was not selected, this bit is always 1, with Read Only access. If both interface types were selected (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.	
7	DDRSel	R/W	Mode: Host and Device ULPI DDR Select (DDRSel) The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. ■ 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus ■ 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus This bit is valid only when OTG_HSPHY_INTERFACE = 2 or 3.	0x0
8	SRPCap	R/W	Mode: Host and Device SRP-Capable (SRPCap) The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. ■ 1'b0: SRP capability is not enabled. ■ 1'b1: SRP capability is enabled. This bit is writable only if an SRP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0. If SRP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	0x0
9	HNPCap	R/W	Mode: Host and Device HNP-Capable (HNPCap) The application uses this bit to control the DWC_otg core's HNP capabilities. ■ 1'b0: HNP capability is not enabled. ■ 1'b1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0.	0x0

Bits	Name	Access	Description	Reset
			If HNP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	
13:10	USBTrdTim	R/W	Mode: Device only USB Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to ■ 4'h5: When the MAC interface is 16-bit UTMI+. ■ 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so If you need the AHB to run at less than 30 MHz, and If USB turnaround time is not critical, these bits can be programmed to a larger value.	0x5
14	Reserved_0C_14	RO	Reserved for future use.	
15	PhyLPwrClkSel	R/W	Mode: Host and Device PHY Low-Power Clock Select (PhyLPwrClkSel) Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. ■ 1'b0: 480-MHz Internal PLL clock ■ 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30- MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS mode and at either 48 or 6 MHz in LS mode (depending on the PHY vendor). This bit drives the utmi_fsls_low_power core output signal, and is valid only For UTMI+ PHYs.	0x0
16	OtgI2CSel	R/W	Mode: Host and Device UTMIFS or I2C Interface Select (OtgI2CSel) The application uses this bit to select the I2C interface. ■ 1' b0: UTMI USB 1.1 Full-Speed interface for OTG signals ■ 1' b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface? in coreConsultant (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.	0x0
17	ULPIFsLs	R/W	Mode: Host and Device ULPI FS/LS Select (ULPIFsLs) The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is	0x0

Bits	Name	Access	Description	Reset
			selected on the ULPI PHY. ■ 1'b0: ULPI interface ■ 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3) Before setting this bit, the application needs to ensure that GUSBCFG.ULPI_UTMI_SEL = 1'b1.	
18	ULPIAutoRes	R/W	Mode: Host and Device ULPI Auto Resume (ULPIAutoRes) This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. ■ 1'b0: PHY does not use AutoResume feature. ■ 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
19	ULPIClkSusM	R/W	Mode: Host and Device ULPI Clock SuspendM (ULPIClkSusM) This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. ■ 1'b0: PHY powers down internal clock during suspend. ■ 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
20	ULPIExtVbusDrv	R/W	Mode: Host only ULPI External VBUS Drive (ULPIExtVbusDrv) This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. ■ 1'b0: PHY drives VBUS using internal charge pump (Default). ■ 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
21	ULPIExtVbusIndicator	R/W	Mode: Host only ULPI External VBUS Indicator (ULPIExtVbusIndicator) This bit indicates to the ULPI PHY to use an external VBUS overcurrent indicator. ■ 1'b0: PHY uses internal VBUS valid comparator. ■ 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
22	TermSelDLPulse	R/W	Mode: Device only TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. ■ 1'b0: Data line pulsing using utmi_txvalid (Default). ■ 1'b1: Data line pulsing using utmi_termsel.	0x0
23	Complement	R/W	Mode: Host only Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. For more information, refer to the ULPI Specification.	0x0

Bits	Name	Access	Description	Reset
			■ 1'b0: PHY does not invert ExternalVbusIndicator signal ■ 1'b1: PHY does invert ExternalVbusIndicator signal This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	
24	Indicator	R/W	Mode: Host only Indicator Pass Through Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. For more information, refer to the ULPI Specification. ■ 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. ■ 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator. This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
25	ULPI	R/W	Mode: Host only ULPI Interface Protect Disable Controls circuitry built into the PHY For protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. For more information, refer to the ULPI Specification. ■ 1'b0: Enables the interface protect circuit ■ 1'b1: Disables the interface protect circuit This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
26	IC_USBCap	RO	Mode: Host and Device IC_USB-Capable (IC_USBCap) The application uses this bit to control the DWC_otg core's IC_USB capabilities. ■ 1'b0: IC_USB PHY Interface is not selected. ■ 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0. The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.	
27	IC_USBTrafCtl	R/W	Mode: Device only IC_USB TrafficPullRemove Control (IC_USBTrafCtl) When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling, per section 6.3.4 of the IC_USB specification. This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field USBCFG.IC_USBCap is set to 1.	0x0
28	TxEndDelay	R/W	Mode: Device only	0x0

Bits	Name	Access	Description	Reset
			Tx End Delay (TxEndDelay) Writing 1'b1 to this bit enables the core to follow the TxEndDelay timings as per UTMI+ specification 1.05 section 4.1.5 for opmode signal during remote wakeup. ■ 1'b0: Normal Mode. ■ 1'b1: Tx End delay.	
29	ForceHstMode	R/W	Mode: Host and device Force Host Mode (ForceHstMode) Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode. ■ 1'b1: Force Host Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	0x0
30	ForceDevMode	R/W	Mode: Host and device Force Device Mode (ForceDevMode) Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode. ■ 1'b1: Force Device Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	0x0
31	CorruptTxPkt	R/W	Mode: Host and device Corrupt Tx packet (CorruptTxPkt) This bit is for debug purposes only. Never set this bit to 1.The application should always write 1'b0 to this bit.	0x0

GRSTCTL

Reset Register
Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	CSftRst	RO	Write Behavior: One to set Mode: Host and Device Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: ■ Clears the interrupts and all the CSR registers except the following register bits: - PCGCCTL.RstPdwnModule - PCGCCTL.GateHclk - PCGCCTL.PwrClmp	

Bits	Name	Access	Description	Reset
			- PCGCCTL.StopPPhyLPwrClkSelclk - GUSBCFG.PhyLPwrClkSel - GUSBCFG.DDRSel - GUSBCFG.PHYSel - GUSBCFG.FSIntf - GUSBCFG.ULPI_UTMI_Sel - GUSBCFG.PHYIf - GUSBCFG.TxEndDelay - GUSBCFG.TermSelDLPulse - GUSBCFG.ULPIClkSusM - GUSBCFG.ULPIAutoRes - GUSBCFG.ULPIFsLs - GGPIO - GPWRDN - GADPCTL - HCFG.FSLSPclkSel - DCFG.DevSpd - DCTL.SftDiscon ■ All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. ■ Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. ■ When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. After this bit is cleared, the application must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). The application must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. After a new clock is selected, the PHY domain has to be reset for proper operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	PIUFSSftRst	RO	Write Behavior: One to set Mode: Host and Device	

Bits	Name	Access	Description	Reset
			PIU FS Dedicated Controller Soft Reset (PIUFSSftRst) Resets the PIU FS Dedicated Controller All module state machines in FS Dedicated Controller of PIU are reset to the IDLE state. Used to reset the FS Dedicated controller in PIU in case of any PHY Errors like Loss of activity or Babble Error resulting in the PHY remaining in RX state for more than one frame boundary. This is a self clearing bit and core clears this bit after all the necessary logic is reset in the core. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	FrmCntrRst	RO	Write Behavior: One to set Mode: Host only Host Frame Counter Reset (FrmCntrRst) The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0. If the application writes 1 to the bit, it may not be able to read back the value as it gets cleared by the core in a few clock cycles. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	INTknQFlsh	RWS	Mode: Device only IN Token Sequence Learning Queue Flush (INTknQFlsh) This bit is valid only if OTG_EN_DED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.	
4	RxFFlsh	RO	Write Behavior: One to set Mode: Host and Device RxFIFO Flush (RxFFlsh) The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.	
5	TxFFlsh	RWS	Write Behavior: One to set Mode: Host and Device TxFIFO Flush (TxFFlsh) This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither	

Bits	Name	Access	Description	Reset
			writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: ■ Read - NAK Effective Interrupt ensures the core is not reading from the FIFO ■ Write - GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.	
10:6	TxFNum	R/W	Mode: Host and Device TxFIFO Number (TxFNum) This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. ■ 5'h0: - Non-periodic TxFIFO flush in Host mode - Non-periodic TxFIFO flush in device mode when in shared FIFO operation - Tx FIFO 0 flush in device mode when in dedicated FIFO mode ■ 5'h1: - Periodic TxFIFO flush in Host mode - Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation - TXFIFO 1 flush in device mode when in dedicated FIFO mode ■ 5'h2: - Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation - TXFIFO 2 flush in device mode when in dedicated FIFO mode ... ■ 5'hF: - Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation - TXFIFO 15 flush in device mode when in dedicated FIFO mode ■ 5'h10: - Flush all the transmit FIFOs in device or host mode.	0x0
29:11	Reserved_10_29_11	RO	Reserved for future use.	
30	DMAReq	RO	Mode: Host and Device DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug.	
31	AHBIdle	RO	Mode: Host and Device AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition.	

GINTSTS

Interrupt Status Register
Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	CurMod	RO	Mode: Host and Device Current Mode of Operation (CurMod) Indicates the current mode. ■ 1'b0: Device mode ■ 1'b1: Host mode Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMis	RWC	Write Behavior: One to clear Mode: Host and Device Mode Mismatch Interrupt (ModeMis) The core sets this bit when the application is trying to access: ■ A Host mode register, when the core is operating in Device mode ■ A Device mode register, when the core is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core. This bit can be set only by the core and the application should write 1 to clear it. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	OTGInt	RO	Mode: Host and Device OTG Interrupt (OTGInt) The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	Sof	RWC	Write Behavior: One to clear Mode: Host and Device Start of (micro)Frame (Sof) In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device	

Bits	Name	Access	Description	Reset
			Status register to get the current (micro)Frame number. This interrupt is seen only when the core is operating at either HS or FS.This bit can be set only by the core and the application must write 1 to clear it. Note: The register may return 1'b1 if read immediately after power on reset. If the register bit reads 1'b1 immediately after power on reset, it does not indicate that an SOF has been sent (in host mode), or SOF has been received (in device mode). The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset, the application can clear the bit.	
4	RxFLvl	RO	Mode: Host and Device RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO.	
5	NPTxFEmp	RO	Mode: Host and Device Non-periodic TxFIFO Empty (NPTxFEmp) This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl). In host mode, the application can use GINTSTS.NPTxFEmp with the OTG_EN_DED_TX_FIFO parameter set to either 1 or 0. In device mode, the application uses GINTSTS.NPTxFEmp when OTG_EN_DED_TX_FIFO=0. When OTG_EN_DED_TX_FIFO=1, the application uses DIEPINTn.TxFEmp.	
6	GINNakEff	RO	Mode: Device only Global IN Non-periodic NAK Effective (GINNakEff) Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak) set by the application has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	
7	GOUTNakEff	RO	Mode: Device only	

Bits	Name	Access	Description	Reset
			Global OUT NAK Effective (GOUTNakEff) Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak) set by the application has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).	
8	ULPICKINT_I2C CKINT	RWC	Write Behavior: One to clear Mode: Host and Device ULPI Carkit Interrupt (ULPICKINT) The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. This field is used only if the Carkit interface was enabled in coreConsultant (parameter OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. I2C Carkit Interrupt (I2CCKINT) The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	
9	I2CINT	RWC	Write Behavior: One to clear Mode: Host and Device I2C Interrupt (I2CINT) The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	
10	ErlySusp	RWC	Write Behavior: One to clear Mode: Device only Early Suspend (ErlySusp) The core sets this bit to indicate that an Idle state has been detected on the USB For 3 ms.	
11	USBSusp	RWC	Write Behavior: One to clear Mode: Device only USB Suspend (USBSusp) The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspend state when there is no activity on the linestate signal for an extended period of time.	
12	USBRst	RWC	Write Behavior: One to clear Mode: Device only USB Reset (USBRst) The core sets this bit to indicate that a reset is detected on the USB.	
13	EnumDone	RWC	Write Behavior: One to clear Mode: Device only Enumeration Done (EnumDone)	

Bits	Name	Access	Description	Reset
			The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.	
14	ISOOutDrop	RWC	Write Behavior: One to clear Mode: Device only Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	
15	EOPF	RWC	Write Behavior: One to clear Mode: Device only End of Periodic Frame Interrupt (EOPF) Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.	
16	RstrDoneInt	RWC	Mode: Host and Device Restore Done Interrupt (RstrDoneInt) The core sets this bit to indicate that the Restore command after Hibernation was completed by the core. The core continues from Suspend state into the mode dictated by the PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled (OTG_EN_PWRPOPT=2).	
17	EPMis	RO	Write Behavior: One to clear Mode: Device only Endpoint Mismatch Interrupt (EPMis) Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	
18	IEPInt	RO	Mode: Device only IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	
19	OEPInt	RO	Mode: Device only	

Bits	Name	Access	Description	Reset
			OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and Then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	
20	incompISOIN	RWC	Write Behavior: One to clear Mode: Device only Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.	
21	incompIP_incompISOOUT	RWC	Write Behavior: One to clear Incomplete Periodic Transfer (incompIP) Mode: Host only In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) Mode: Device only In Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	
22	FetSusp	RWC	Write Behavior: One to clear Mode: Device only Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data For IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application For an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: ■ Sets a Global non-periodic IN NAK handshake ■ Disables In endpoints	

Bits	Name	Access	Description	Reset
			■ Flushes the FIFO ■ Determines the token sequence from the IN Token Sequence Learning Queue ■ Re-enables the endpoints ■ Clears the Global non-periodic IN NAK handshake If the Global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received. The core generates an 'IN token received when FIFO empty' interrupt. DWC_otg then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a Global NAK handshake. Alternatively, the application can mask the “IN token received when FIFO empty” interrupt when clearing a Global IN NAK handshake.	
23	ResetDet	RWC	Write Behavior: One to clear Mode: Device only Reset detected Interrupt (ResetDet) In Device mode, this interrupt is asserted when a reset is detected on the USB in partial power-down mode when the device is in Suspend. In Host mode, this interrupt is not asserted.	
24	PrtInt	RO	Mode: Host only Host Port Interrupt (PrtInt) The core sets this bit to indicate a change in port status of one of the DWC_otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	
25	HChInt	RO	Mode: Host only Host Channels Interrupt (HChInt) The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this	

Bits	Name	Access	Description	Reset
			bit.	
26	PTxFEmp	RO	Mode: Host only Periodic TxFIFO Empty (PTxFEmp) This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).	
27	LPM_Int	RWC	Write Behavior: One to clear Mode: Host and Device LPM Transaction Received Interrupt (LPM_Int) ■ Device Mode - This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. ■ Host Mode - This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt).	
28	ConIDStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Connector ID Status Change (ConIDStsChng) The core sets this bit when there is a change in connector ID status.	
29	DisconnInt	RWC	Write Behavior: One to clear Mode: Host only Disconnect Detected Interrupt (DisconnInt) Asserted when a device disconnect is detected.	
30	SessReqInt	RWC	Write Behavior: One to clear Mode: Host and Device Session Request/New Session Detected Interrupt (SessReqInt) In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.	
31	WkUpInt	RWC	Write Behavior: One to clear Mode: Host and Device Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. ■ During Suspend (L2): - Device Mode - This interrupt is asserted only when Host Initiated Resume is detected on USB. - Host Mode - This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. ■ During LPM (L1): - Device Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.	

Bits	Name	Access	Description	Reset
			- Host Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.	

GINTMSK

Interrupt Mask Register
Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved_18_0	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMisMsk	R/W	Mode: Host and Device Mode Mismatch Interrupt Mask (ModeMisMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	OTGIntMsk	R/W	Mode: Host and Device OTG Interrupt Mask (OTGIntMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
3	SofMsk	R/W	Mode: Host and Device Start of (micro)Frame Mask (SofMsk)	0x0
4	RxFLvlMsk	R/W	Mode: Host and Device Receive FIFO Non-Empty Mask (RxFLvlMsk)	0x0
5	NPTxFEmpMsk	R/W	Mode: Host and Device Non-periodic TxFIFO Empty Mask (NPTxFEmpMsk)	0x0
6	GINNakEffMsk	R/W	Mode: Device only Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)	0x0
7	GOUTNakEffMsk	R/W	Mode: Device only Global OUT NAK Effective Mask (GOUTNakEffMsk)	0x0
8	ULPICKINTMsk_I2CCKINTMsk	R/W	ULPI Carkit Interrupt Mask (ULPICKINTMsk) Mode: Host and Device I2C Carkit Interrupt Mask (I2CCKINTMsk) Mode: Host and Device	0x0
9	I2CIntMsk	R/W	Mode: Host and Device I2C Interrupt Mask (I2CIntMsk)	0x0
10	ErlySuspMsk	R/W	Mode: Device only Early Suspend Mask (ErlySuspMsk)	0x0
11	USBSuspMsk	R/W	Mode: Device only USB Suspend Mask (USBSuspMsk)	0x0
12	USBRstMsk	R/W	Mode: Device only USB Reset Mask (USBRstMsk)	0x0
13	EnumDoneMsk	R/W	Mode: Device only Enumeration Done Mask (EnumDoneMsk)	0x0
14	ISOOutDropMsk	R/W	Mode: Device only Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)	0x0
15	EOPFMsk	R/W	Mode: Device only End of Periodic Frame Interrupt Mask (EOPFMsk)	0x0
16	RstrDoneIntMsk	R/W	Mode: Host and Device Restore Done Interrupt Mask (RstrDoneIntMsk)	0x0

Bits	Name	Access	Description	Reset
			This field is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2).	
17	EPMisMsk	R/W	Mode: Device only Endpoint Mismatch Interrupt Mask (EPMisMsk)	0x0
18	IEPIntMsk	R/W	Mode: Device only IN Endpoints Interrupt Mask (IEPIntMsk)	0x0
19	OEPIntMsk	R/W	Mode: Device only OUT Endpoints Interrupt Mask (OEPIntMsk)	0x0
20	incompISOINMsk	R/W	Mode: Device only Incomplete Isochronous IN Transfer Mask (incompISOINMsk) This bit is enabled only when device periodic endpoints are enabled in Dedicated TxFIFO mode.	0x0
21	incomplPMsk_incompISOOUTMsk	R/W	Incomplete Periodic Transfer Mask (incomplPMsk) Mode: Host only Incomplete Isochronous OUT Transfer Mask (incompISOOUTMsk) Mode: Device only	0x0
22	FetSuspMsk	R/W	Mode: Device only Data Fetch Suspended Mask (FetSuspMsk)	0x0
23	ResetDetMsk	R/W	Mode: Device only Reset detected Interrupt Mask (ResetDetMsk)	0x0
24	PrtIntMsk	R/W	Mode: Host only Host Port Interrupt Mask (PrtIntMsk)	0x0
25	HChIntMsk	R/W	Mode: Host only Host Channels Interrupt Mask (HChIntMsk)	0x0
26	PTxFEmpMsk	R/W	Mode: Host only Periodic TxFIFO Empty Mask (PTxFEmpMsk)	0x0
27	LPM_IntMsk	R/W	Mode: Host and Device LPM Transaction received interrupt Mask (LPM_IntMsk)	0x0
28	ConIDStsChngMsk	R/W	Mode: Host and Device Connector ID Status Change Mask (ConIDStsChngMsk)	0x0
29	DisconnIntMsk	R/W	Mode: Host and Device Disconnect Detected Interrupt Mask (DisconnIntMsk)	0x0
30	SessReqIntMsk	R/W	Mode: Host and Device Session Request/New Session Detected Interrupt Mask (SessReqIntMsk)	0x0
31	WkUpIntMsk	R/W	Mode: Host and Device Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk) The WakeUp bit is used for LPM state wake up in a way similar to that of wake up in suspend state.	0x0

GUID

User ID Register

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	UserID	R/W	User ID (UserID) Application-programmable ID field. Reset: Configurable	0x0

GLPMCFG

Core LPM Configuration Register

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	LPMCap	R/W	Mode: Host and Device LPM-Capable (LPMCap) The application uses this bit to control the DWC_otg core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device or hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. ■ 1b0: LPM capability is not enabled ■ 1b1: LPM capability is enabled This bit is writable only if an LPM mode was specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_LPM). Otherwise, reads return 0. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	AppL1Res	R/W	Mode: Device only LPM response programmed by application (AppL1Res) Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1’b0, then the core always responds with NYET response. If GLPMCFG.LPMCap is 1’b1, the core response is as follows. ■ 1: ACK Even though ACK is pre-programmed, the core Device responds with ACK only on successful LPM transaction. The LPM transaction is successful if: - No PID/CRC5 Errors in either EXT token or LPM token (else ERROR) - Valid bLinkState = 0001B (L1) received in LPM transaction (else STALL) - No data pending in transmit queue (else NYET). ■ 0: NYET The pre-programmed software bit is over-ridden for response to LPM token when: - The received bLinkState is not L1 (STALL response), or - An error is detected in either of the LPM token packets because of corruption (ERROR response). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
5:2	HIRD	R/W	Mode: Host and Device ■ EnBESL = 1' b0 Host-Initiated Resume Duration (HIRD) Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated	0x0

Bits	Name	Access	Description	Reset
			resume. Device Mode (Read-Only): This field is updated with the Received LPM Token HIRD bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No HIRD[3:0] THIRD (µs) 1 4’b0000 50 2 4’b0001 125 3 4’b0010 200 4 4’b0011 275 5 4’b0100 350 6 4’b0101 425 7 4’b0110 500 8 4’b0111 575 9 4’b1000 650 10 4’b1001 725 11 4’b1010 800 12 4’b1011 875 13 4’b1100 950 14 4’b1101 1025 15 4’b1110 1100 16 4’b1111 1175 Reset: 4’b0000 ■ EnBESL = 1’ b1 Best Effort Service Latency (BESL) Host Mode: The value of BESL to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode (Read-Only): This field is updated with the Received LPM Token BESL bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No BESL[3:0] TBESL (µs) 1 4’b0000 125 2 4’b0001 150 3 4’b0010 200 4 4’b0011 300 5 4’b0100 400 6 4’b0101 500 7 4’b0110 1000 8 4’b0111 2000 9 4’b1000 3000 10 4’b1001 4000 11 4’b1010 5000 12 4’b1011 6000 13 4’b1100 7000 14 4’b1101 8000 15 4’b1110 9000 16 4’b1111 10000 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
6	bRemoteWake	R/W	Mode: Host and Device	0x0

Bits	Name	Access	Description	Reset
			RemoteWakeEnable (bRemoteWake) Host Mode: The value of remote wake up to be sent in the wIndex field of LPM transaction. Device Mode (Read-Only): This field is updated with the Received LPM Token bRemoteWake bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction.	
7	EnblSlpM	R/W	Mode: Host and Device Enable utmi_sleep_n (EnblSlpM) ULPI Interface: The application uses this bit to control the utmi_sleep_n assertion to the PHY when in L1 state. For the host, this bit is valid only in “local device” mode. ■ 1b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. ■ 1b1: utmi_sleep_n assertion from the core is transferred to the external PHY. Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The Synopsys ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted. Other Interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode. ■ 1' b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. ■ 1' b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.	0x0
12:8	HIRD_Thres	R/W	Mode: Host and Device BESL or HIRD Threshold (HIRD_Thres) Device Mode: ■ EnBESL = 1' b0: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD value is greater than or equal to the value defined in this field HIRD_Thres[3:0] and HIRD_Thres[4] is set to 1’b1. ■ EnBESL = 1' b1: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when BESL value is greater than or equal to the value defined in this field BESL_Thres[3:0] and BESL_Thres[4] is set to 1’b1. ■ DCTL.DeepSleepBESLReject = 1'b1: In device initiated resume, the core expects the Host to resume service to the device within the BESL value	0x0

Bits	Name	Access	Description	Reset
			corresponding to L1 exit time specified in HIRD_Thresh[3:0]. The Device sends a NYET response when the received HIRD in LPM token is greater than HIRD threshold. ■ Note: To differentiate between Deep Sleep and Shallow sleep HIRD greater than or equal to HIRD threshold comparison is done. For differentiating between NYET or ACK response for LPM token HIRD greater than HIRD Threshold comparison is used. Host Mode: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD_Thres[4] is set to 1b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by host (TL1HubDrvResume2) on the USB bus when it detects device initiated resume. HIRD_Thres must not be programmed with a value greater than 4’b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Sl. No ----Thres[3:0] --- -Host Mode Resume Signaling Time (µs) --- - --- - --- - --- - --------- - --- - EnBESL = 1'b0 - -EnBESL = 1'b1 1 4’b0000 60 75 2 4’b0001 135 100 3 4’b0010 -210 150 4 4’b0011 285 250 5 4’b0100 360 350 6 4’b0101 435 450 7 4’b0110 510 950 8 4’b0111 585 Invalid 9 4’b1000 660 Invalid 10 4’b1001 735 Invalid 11 4’b1010 810 Invalid 12 4’b1011 885 Invalid 13 4’b1100 960 Invalid 14 4’b1101 Invalid Invalid 15 4’b1110 Invalid Invalid 16 4’b1111 Invalid Invalid The following truth table explains the difference in behavior between the UTMI and ULPI interface in different modes of operation: Bit 7 --Bit 6 --sleep_n ---l1_suspend_n --suspend_n ---Mode of Operation 0 --- -----1- --- ----1--- --- --1--- --- ------- --- --1--- --- ------- -- Normal Operation 0 --- -----0- --- ----1--- --- --1--- --- ------- --- --0--- --- ------- --L2 Suspend 1 --- -----0- --- ----1--- --- --0--- --- ------- --- --1--- --- ------- --L1 Deep Sleep 1 --- -----1- --- ----0--- --- --1--- --- ------- --- --1--- --- ------- --L1 Shallow Sleep	
14:13	CoreL1Res	RO	Mode: Host and Device LPM Response (CoreL1Res) Device Mode: The response of the core to LPM transaction	

Bits	Name	Access	Description	Reset
			received is reflected in these two bits. Host Mode: Handshake response received from local device for LPM transaction ■ 11 - ACK ■ 10 - NYET ■ 01 - STALL ■ 00 - ERROR (No handshake response)	
15	SlpSts	RO	Mode: Device only Port Sleep Status (SlpSts) This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the TL1TokenRetry timer has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input signal. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: ■ When there is any activity on the USB linestate ■ When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or softdisconnects the device. Host Mode: The host transitions to Sleep (L1) state as a side-effect of a successful LPM transaction by the core to the local port with ACK response from the device. The read value of this bit reflects the current Sleep status of the port. The core clears this bit after: ■ The core detects a remote L1 Wakeup signal, ■ The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register, or ■ The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively). Values: ■ 1b0: Core not in L1 ■ 1b1: Core in L1	
16	L1ResumeOK	RO	Mode: Host and device Sleep State Resume OK (L1ResumeOK) Indicates that the application or host can start resume from Sleep state. This bit is valid in LPM sleep (L1) state. It is set in sleep mode after a delay of 50 µs (TL1Residency). This bit is reset when SlpSts is 0. ■ 1b1: The application or core can start Resume from Sleep state	

Bits	Name	Access	Description	Reset
			■ 1b0: The application or core cannot start Resume from Sleep state	
20:17	LPM_Chnl_Indx	R/W	Mode: Host only LPM Channel Index (LPM_Chnl_Indx) The channel number on which the LPM transaction has to be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and end point number programmed in the corresponding channel into the LPM transaction.	0x0
23:21	LPM_Retry_Cnt	R/W	Mode: Host only LPM Retry Count (LPM_Retry_Cnt) When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.	0x0
24	SndLPM	RWS	Write Behavior: One to set Mode: Host only Send LPM Transaction (SndLPM) When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the Device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must be set only when the host is connected to a local port.	
27:25	LPM_RetryCnt_Sts	RO	Mode: Host only LPM Retry Count Status (LPM_RetryCnt_Sts) Number of LPM Host Retries still remaining to be transmitted for the current LPM sequence.	
28	EnBESL	R/W	Mode: Host and device Enable Best Effort Service Latency (BESL) This bit enables the BESL feature as defined in the LPM errata: ■ 1' b0: The core works as described in the following document: USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 ■ 1' b1: The core works as per the LPM Errata	0x0
29	RstrSlpSts	R/W	Mode: Device only Restore SlpSts (RstrSlpSts) When the application power-gates the core (partial power-down or hibernation), the application needs to program this bit to restore the LPM status in the core. Based on the BESL value received from the Host, the application needs to program this bit during restore process. The application should program this bit depending on whether it decided to put the core in Shallow Sleep (Clock Gating	0x0

Bits	Name	Access	Description	Reset
			Only) or Deep Sleep (Power Gating) mode: ■ 1'b0: The application puts the core in Shallow Sleep mode based on the BESL value from the Host. ■ 1'b1: The application puts the core in Deep Sleep mode based on the BESL value from the Host.	
30	HSICCon	R/W	Mode: Host and device HSIC-Connect (HSICCon) The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the Host Core configures to drive HSIC Idle state (STROBE=1&DATA=0) on the bus. It then waits for device to initiate the HSIC Connect sequence. Device Mode: Once this bit is set, the Device Core waits for HSIC Idle linestate on the bus. After receiving the Idle linestate it then initiates the HSIC Connect. This bit is valid only if OTG_ENABLE_HSIC = 1, if_sel_hsic = 1, and InvSelHSIC = 0. Otherwise, it is read-only.	0x0
31	InvSelHsic	R/W	Mode: Host and device HSIC-Invert Select HSIC (InvSelHsic) The application uses this bit to control the DWC_otg core HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core is non-HSIC-capable, it can connect to only PHYs that are not HSIC capable. If the core is HSIC-capable, it can connect only to PHYs that are HSIC capable. ■ If if_sel_hsic input signal is 1: - InvSelHsic = 1b1: HSIC capability is not enabled - InvSelHsic = 1b0: HSIC capability is enabled ■ If if_sel_hsic input signal is 0: - InvSelHsic = 1b1: HSIC capability is enabled - InvSelHsic = 1b0: HSIC capability is not enabled This bit is writable only if HSIC mode is specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled. Otherwise, reads return 0.	0x0

GPWRDN

Power Down Register
Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	PMUIntSel	R/W	Mode: Host and Device PMU Interrupt Select (PMUIntSel) When the hibernation functionality is selected (OTG_EN_PWROPT = 2), a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state, all interrupts from the	0x0

Bits	Name	Access	Description	Reset
			DWC_otg_core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation ■ 1'b0: Internal DWC_otg_core interrupt is selected ■ 1'b1: External DWC_otg_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	PMUActv	R/W	Mode: Host and Device PMU Active (PMUActv) This bit enables or disables the PMU logic. ■ 1'b0: Disable PMU module ■ 1'b1: Enable PMU module Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	Restore	R/W	Mode: Host and Device Restore The application must program this bit to enable or disable restore mode from the PMU module. ■ 1' b0: DWC_otg in normal mode of operation ■ 1' b1: DWC_otg in restore mode Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWROPT = 2. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
3	PwrDnClmp	R/W	Mode: Host and Device Power Down Clamp (PwrDnClmp) The application must program this bit to enable or disable the clamps to all the outputs of the DWC_otg core module to prevent the corruption of other active logic. ■ 1'b0: Disable PMU power clamp ■ 1'b1: Enable PMU power clamp	0x0
4	PwrDnRst_n	R/W	Mode: Host and Device Power Down ResetN (PwrDnRst_n) The application must program this bit to reset the DWC_otg core during the Hibernation exit process or during ADP when powering up the core (if the DWC_otg core was powered off during ADP process). ■ 1'b1: DWC_otg is in normal operation ■ 1'b0: Reset DWC_otg Note: This bit must not be written to during normal mode of operation.	0x1
5	PwrDnSwtch	R/W	Mode: Host and Device	0x0

Bits	Name	Access	Description	Reset
			Power Down Switch (PwrDnSwtch) This bit indicates to the DWC_otg core whether the VDD switch is in ON or OFF state. ■ 1'b0: DWC_otg is in ON state ■ 1'b1: DWC_otg is in OFF state Note: This bit must not be written to during normal mode of operation.	
6	DisableVBUS	R/W	Mode: Host and Device DisableVBUS Host Mode: The application must program this bit if HPRT0.PrtPwr was programmed to 0 before switching off the Core. This indicates to the PMU whether session was ended before entering Hibernation. ■ 1'b0: HPRT0.PrtPwr was not programed to 0. ■ 1'b1: HPRT0.PrtPwr was programmed to 0. Device Mode: The application must program this bit to inform the PMU whether the bvalid valid signal is high (session valid) or low (session end) whenever the core is switched off. ■ 1'b0: bvalid signal is High (Session Valid) ■ 1'b1: bvalid signal is Low (Session End) This bit is valid only when GPWRDN.PMUActv is 1.	0x0
7	LnStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Line State Change (LnStsChng) This interrupt is asserted when there is a linestate change detected by the PMU. The application must read GPWRDN.Linestate to determine the current linestate on USB. ■ 1'b0: No LineState change on USB ■ 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1 and OTG_EN_PWROPT = 2.	
8	LineStageChange Msk	R/W	Mode: Host and Device Mask For LineStateChange interrupt (LineStageChangeMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
9	ResetDetected	RWC	Write Behavior: One to clear Mode: Device only ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. ■ 1'b0: Reset Not Detected ■ 1'b1: Reset Detected This bit is valid only when OTG_EN_PWROPT = 2.	
10	Mask_ResetDetM sk	R/W	Mode: Device only Mask For ResetDetected interrupt (ResetDetMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
11	DisconnectDetect	RWC	Write Behavior: One to clear Mode: Host only DisconnectDetect This field indicates that Disconnect has been detected by the	

Bits	Name	Access	Description	Reset
			PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. ■ 1'b0: Disconnect not detected ■ 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWROPT = 2.	
12	DisconnectDetect Msk	R/W	Mode: Host only Mask For DisconnectDetect Interrupt (DisconnectDetectMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
13	ConnectDet	RO	Mode: Host and Device Write Behavior: One to clear ConnectDet This field indicates that a new connect has been detected ■ 1'b0: Connect not detected ■ 1'b1: Connect detected This bit is valid only when OTG_EN_PWROPT = 2.	
14	ConnDetMsk	R/W	Mode: Host and Device ConnDetMsk Mask for ConnectDet interrupt This bit is valid only when OTG_EN_PWROPT = 2.	0x0
15	SRPDetect	RWC	Mode: Host only SRPDetect This field indicates that SRP has been detected by the PMU. This field generates an interrupt. After detecting SRP during hibernation the application must not restore the core. The application must get into the initialization process. ■ 1'b0: SRP not detected ■ 1'b1: SRP detected	
16	SRPDetectMsk	R/W	Mode: Host only Mask For SRPDetect Interrupt (SRPDetectMsk)	0x0
17	StsChngInt	RWC	Write Behavior: One to clear Status Change Interrupt (StsChngInt) This field indicates a status change in either the IDDIG or BSessVld signal. ■ 1'b0: No Status change ■ 1'b1: status change detected After receiving this interrupt the application must read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application. Note: When Battery Charger is Enabled and the ULPI interface is used, if StsChngInt is received and the application reads GPWRDN register and determines that it is due to a change in the value of IDDIG, then StsChngInt may be generated once again within the next	

Bits	Name	Access	Description	Reset
			few clock cycles. This occurs because of an ambiguity in the implementation of Battery Charger Support over the ULPI interface. After receiving the StsChngInt for the second time the application can once again read the GPWRDN register. However, this time the value IDDIG (or BSesVld) will not have changed. The application then processes the second interrupt but no further action will be required as a result.	
18	StsChngIntMsk	R/W	Mode: Host and Device Mask For StsChng Interrupt (StsChngIntMsk)	0x0
20:19	LineState	RO	Mode: Host and Device LineState This field indicates the current linestate on USB as seen by the PMU module. ■ 2'b00: DM = 0, DP = 0 ■ 2'b01: DM = 0, DP = 1 ■ 2'b10: DM = 1, DP = 0 ■ 2'b11: Not-defined This bit is valid only when GPWRDN.PMUActv is 1.	
21	IDDIG	RO	Mode: Host and Device IDDIG This bit indicates the status of the IDDIG signal. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. ■ 1'b1: Device mode ■ 1'b0: Host mode This bit is valid only when GPWRDN.PMUActv is 1.	
22	BSessVld	RO	Mode: Device only B Session Valid (BSessVld) This field reflects the B session valid status signal from the PHY. ■ 1'b0: B-Valid is 0 ■ 1'b1: B-Valid is 1 This bit is valid only when GPWRDN.PMUActv is 1.	
23	ADPInt	RWC	Write Behavior: One to clear Mode: Host and Device ADP Interrupt (ADPInt) This bit is set whenever there is a ADP event	
28:24	MultValIdBC	RO	Mode: Host and Device MultValIdBC (MultValIdBC) Battery Charger ACA inputs in the following order: ■ Bit 28 - rid_float ■ Bit 27 - rid_gnd ■ Bit 26 - rid_a ■ Bit 25 - rid_b ■ Bit 24 - rid_c These bits are present only if BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.	

Bits	Name	Access	Description	Reset
			Reset: As per ACA input	
31:29	Reserved_58_31_29	RO	Reserved for future use.	

18.6.4 Host 初始化程序说明

完成【频率启动程序】和【模式切换与初始化程序】之后，需要接着执行 XHCI 的初始化程序，如下方所列，再来就可以依需求启动四种标准型态传输，启动标准传输的方法详细内容可参照 XHCI 规格书，在此就不多加赘述：

设定 GINTMSK.PrtInt 寄存器为 unmask 状态
设定 HCFG register 为 FS device 或者是 HS device 装置
设定 HPRT.PrtPwr 寄存器为 1，此一设定会打开 USB bus 上的 VBUS
等待 HPRT0.PrtConnDet 中断发生，代表有一个 device 连接到 USB downstream port
设定 HPRT.PrtRst 寄存器为 1，开始进行 USB port reset
等待至少 10ms 让 USB port reset 有足够的时间完成 handshake
设定 HPRT.PrtRst 为 0，完成 USB port reset 程序
等待 HPRT.PrtEnChng 中断发生
读取 HPRT.PrtSpd 寄存器获取 enumeration speed 数值
设定 HFIR 寄存器进而配置相对应的 PHY Clock
设定 RXFSIZE 寄存器，配置 RXFIFO 的大小
设定 GNPTXFSIZ 寄存器，配置非周期性传输 TXFIFO 的大小
设定 HPTXFSIZ 寄存器，配置周期性传输 TXFIFO 的大小

18.6.5 Host 寄存器说明

Host 寄存器的基础寻址(base address)在整个内存空间的地址为 **0x0434_0000**，在文中此基础寻址将以 HOST_BASE_ADDR 做为代表，因此 Host 控制器的每个寄存器在内存空间中的真实寻址就会是【HOST_BASE_ADDR+相对地址】。

18.6.5.1 寄存器摘要

Name	Address Offset	Description
HCFG	0x400	Host Configuration Register
HFIR	0x404	Host Frame Interval Register
HFNUM	0x408	Host Frame Number/Frame Time Remaining Register
HPTXSTS	0x410	Host Periodic Transmit FIFO/Queue Status Register
HAINT	0x414	Host All Channels Interrupt Register
HAINTMSK	0x418	Host All Channels Interrupt Mask Register
HFLBAddr	0x41c	Host Frame List Base Address Register
HCCHARn	0x500	Host Channel-n Characteristics Register

Name	Address Offset	Description
HCDMAn	0x514	Host Channel-n DMA Address Register
HCDMABn	0x51c	Host Channel-n DMA Buffer Address Register

18.6.5.2 寄存器详细列表

HCFG

Host Configuration Register
Offset Address: 0x400

Bits	Name	Access	Description	Reset
1:0	FSLSPclkSel	R/W	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode: ■ 2'b00: PHY clock is running at 30/60 MHz ■ 2'b01: PHY clock is running at 48 MHz ■ Others: Reserved When the core is in LS Host mode: ■ 2'b00: PHY clock is running at 30/60 MHz. When the UTMI+/ULPI PHY Low Power mode is not selected, use 30/60 MHz. ■ 2'b01: PHY clock is running at 48 MHz. When the UTMI+ PHY Low Power mode is selected, use 48MHz If the PHY supplies a 48 MHz clock during LS mode. ■ 2'b10: PHY clock is running at 6 MHz. In USB 1.1 FS mode, use 6 MHz when the UTMI+ PHY Low Power mode is selected and the PHY supplies a 6 MHz clock during LS mode. If you select a 6 MHz clock during LS mode, you must do a soft reset. ■ 2'b11: Reserved Notes: ■ When Core in FS mode, the internal and external clocks have the same frequency. ■ When Core in LS mode, - If FSLSPclkSel = 2’b00: Internal and external clocks have the same frequency - If FSLSPclkSel = 2’b10: Internal clock is divided by eight version of external 48 MHz clock (utmifs_clk). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
2	FSLSSupp	R/W	FS- and LS-Only Support (FSLSSupp) The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even If the connected device supports HS traffic. Do not make	0x0

Bits	Name	Access	Description	Reset
			changes to this field after initial programming. ■ 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device ■ 1'b1: FS/LS-only, even If the connected device can support HS Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
6:3	Reserved_400_6_3	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
7	Ena32KHzS	R/W	Enable 32 KHz Suspend mode (Ena32KHzS) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. When FS PHY interface is chosen and this bit is set, the core expects that the PHY clock is switched from 48 MHz to 32 KHz during Suspend.	0x0
15:8	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when HCFG.Ena32KHzS is set. It controls the Resume period when the core resumes from Suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.	0x2
22:16	Reserved_400_22_16	RO	Reserved for future use.	
23	DescDMA	R/W	Enable Scatter/gather DMA in Host mode (DescDMA) When the Scatter/Gather DMA option is selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. Note: This bit must be modified only once after a reset. The following combinations are available for programming: ■ GAHBCFG.DMAEn=0,HCFG.DescDMA=0 => Slave mode ■ GAHBCFG.DMAEn=0,HCFG.DescDMA=1 => Invalid ■ GAHBCFG.DMAEn=1,HCFG.DescDMA=0 => Buffered DMA mode ■ GAHBCFG.DMAEn=1,HCFG.DescDMA=1 => Scatter/Gather DMA mode In non-Scatter/Gather DMA mode, this bit is reserved.	0x0
25:24	FrListEn	R/W	Frame List Entries (FrListEn) The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode. ■ 2'b00: Reserved ■ 2'b01: 8 Entries ■ 2'b10: 16 Entries ■ 2'b11: 32 Entries	0x0

Bits	Name	Access	Description	Reset
			In non-Scatter/Gather DMA mode, these bits are reserved.	
26	PerSchedEna	R/W	Enable Periodic Scheduling (PerSchedEna) Applicable in Host Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is res and the core does not process any periodic channels. As soon as this bit is set, the core gets ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core gets ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non-Scatter/Gather DMA mode, this bit is reserved.	0x0
30:27	Reserved_400_30_27	RO	Reserved for future use.	
31	ModeChTimEn	R/W	Mode Change Ready Timer Enable (ModeChTimEn) This bit is used to enable/disable the Host core to wait 200 PHY clock cycles at the end of Resume to change the opmode signal to the PHY to 00 after Suspend or LPM. ■ 1'b0: The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to the change the opmode from 2'b10 to 2'b00 ■ 1'b1: The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.	0x0

HFIR

Host Frame Interval Register
Offset Address: 0x404

Bits	Name	Access	Description	Reset
15:0	FrInt	R/W	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro- SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for an FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the	0xEA60

Bits	Name	Access	Description	Reset
			FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. ■ 125 µs * (PHY clock frequency for HS) ■ 1 ms * (PHY clock frequency for FS/LS) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
16	HFIRRldCtrl	R/W	Reload Control (HFIRRldCtrl) This bit allows dynamic reloading of the HFIR register during run time. ■ 1'b0: The HFIR cannot be reloaded dynamically ■ 1'b1: The HFIR can be dynamically reloaded during runtime. This bit needs to be programmed during initial configuration and its value must not be changed during runtime.	0x0
31:17	Reserved_404_31_17	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HFNUM

Host Frame Number/Frame Time Remaining Register
Offset Address: 0x408

Bits	Name	Access	Description	Reset
15:0	FrNum	RO	Frame Number (FrNum) This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected in coreConsultant (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:16	FrRem	RO	Frame Time Remaining (FrRem) Indicates the amount of time remaining in the current microframe (HS) or Frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HPTXSTS

Host Periodic Transmit FIFO/Queue Status Register
Offset Address: 0x410

Bits	Name	Access	Description	Reset
15:0	PTxFSpcAvail	RO	Periodic Transmit Data FIFO Space Available (PTxFSpcAvail)	

Bits	Name	Access	Description	Reset
			Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words ■ 16'h0: Periodic TxFIFO is full ■ 16'h1: 1 word available ■ 16'h2: 2 words available ■ 16'hn: n words available (n: 0 ~ 32,768) ■ 16' h8000: 32,768 words available ■ Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
23:16	PTxQSpcAvail	RO	Periodic Transmit Request Queue Space Available (PTxQSpcAvail) Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. ■ 8'h0: Periodic Transmit Request Queue is full ■ 8'h1: 1 location available ■ 8'h2: 2 locations available ■ n: n locations available (n: 0~16) ■ Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:24	PTxQTop	RO	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. ■ Bit [31]: Odd/Even (micro)Frame - 1'b0: send in even (micro)Frame - 1'b1: send in odd (micro)Frame ■ Bits [30:27]: Channel/endpoint number ■ Bits [26:25]: Type - 2'b00: IN/OUT - 2'b01: Zero-length packet - 2'b10: CSPLIT - 2'b11: Disable channel command ■ Bit [24]: Terminate (last entry for the selected channel or endpoint)	

HAINT

Host All Channels Interrupt Register
Offset Address: 0x414

Bits	Name	Access	Description	Reset
15:0	HAINT	RO	Channel Interrupts (HAINT) One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	Reserved_414_31	RO	Reserved for future use	

Bits	Name	Access	Description	Reset
	_16			

HAINTMSK

Host All Channels Interrupt Mask Register
Offset Address: 0x418

Bits	Name	Access	Description	Reset
15:0	HAINTMsk	R/W	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_418_31_16	RO	Reserved for future use	

HFLBAddr

Host Frame List Base Address Register
Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	HFLBAddr	R/W	The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCCHARn

Host Channel-n Characteristics Register
Offset Address: 0x500

Bits	Name	Access	Description	Reset
10:0	MPS	R/W	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
14:11	EPNum	R/W	Endpoint Number (EPNum) Indicates the endpoint number on the device serving as the data source or sink. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
15	EPDir	R/W	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. ■ 1'b0: OUT ■ 1'b1: IN Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
16	Reserved_500_16	RO	Reserved for future use.	
17	LSpdDev	R/W	Low-Speed Device (LSpdDev) This field is set by the application to indicate that this	0x0

Bits	Name	Access	Description	Reset
			channel is communicating to a low-speed device. The application must program this bit when a low speed device is connected to the host through an FS HUB. The DWC_otg Host core uses this field to drive the XCVR_SELECT signal to 2'b11 while communicating to the LS Device through the FS hub. Note: In a peer to peer setup, the DWC_otg Host core ignores this bit even if it is set by the application software.	
19:18	EPType	R/W	Endpoint Type (EPType) Indicates the transfer type selected. ■ 2'b00: Control ■ 2'b01: Isochronous ■ 2'b10: Bulk ■ 2'b11: Interrupt	0x0
21:20	EC	R/W	Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non-periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. ■ 2'b00: Reserved. This field yields undefined results. ■ 2'b01: 1 transaction ■ 2'b10: 2 transactions to be issued for this endpoint per microframe ■ 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transaction on transaction errors. This field must be set to at least 2'b01.	0x0
28:22	DevAddr	R/W	Device Address (DevAddr) This field selects the specific device serving as the data source or sink.	0x0
29	OddFrm	R/W	Odd Frame (OddFrm) This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. ■ 1'b0: Even (micro)frame ■ 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.	0x0
30	ChDis	RWS	Write Behavior: One to set Channel Disable (ChDis)	

Bits	Name	Access	Description	Reset
			The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	
31	ChEna	RWS	Write Behavior: One to set Channel Enable (ChEna) When Scatter/Gather mode is enabled: ■ 1'b0: Indicates that the descriptor structure is not yet ready. ■ 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled: This field is set by the application and cleared by the OTG host. ■ 1'b0: Channel disabled ■ 1'b1: Channel enabled	

HCDMAn

Host Channel-n DMA Address Register
Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	DMAAddr	R/W	DMA Address (DMAAddr) This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCDMABn

Host Channel-n DMA Buffer Address Register
Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	DMABufAddr	R/W	DMA Address (DMAAddr) Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

18.6.6 Device 初始化程序

务必完成 13.8.6.1 的【频率启动程序】之后，再来进行启动与切换至 Device 功能，请依照以下步骤：
设定 DCFG 寄存器中下列位:
DescDMA 为 1，启动 descriptor DMA mode
Device Speed 为 HS 或者是 FS
非 0 传输的状态位
周期性传输的 Interval 数值
设定在 DMA 传输中的 FIFO Threshold 大小
清除 DCTL.SftDiscon 位，让装置对 Host 开始进行 Connection 动作
清除 GINTMSK 以下位
USB Port Reset mask
Enumeration done mask
Early suspend mask
USB suspend mask
SOF mask
等待 GINTSTS.USBReset 中断发生，并且开始进行 USB Reset 初始化程序
等待 GINTSTS.EnumerationDone 中断发生，代表 USB Reset 程序已经完成，并且去读取 DSTS 寄存器，获取 enumeration speed，并进行 Enumeration 初始化程序

18.6.7 Device 寄存器说明

Device 寄存器的基础寻址(address)在整个内存空间的地址为 **0x0434_0000**，在文中此基础寻址将以 DEV_BASE_ADDR 做为代表，因此每个寄存器在内存空间中的真实寻址就会是【DEV_BASE_ADDR+相对地址】。

18.6.7.1 寄存器摘要

Name	Address Offset	Description
DCFG	0x800	Device Configuration Register
DCTL	0x804	Device Control Register
DSTS	0x808	Device Status Register
DIEPMSK	0x810	Device IN Endpoint Common Interrupt Mask Register
DOEPMSK	0x814	Device OUT Endpoint Common Interrupt Mask Register
DAINT	0x818	Device All Endpoints Interrupt Register
DAINTMSK	0x81c	Device Endpoints Interrupt Mask Register
DIEPEMPMSK	0x834	Device IN Endpoint FIFO Empty Interrupt Mask Register
DEACHINT	0x838	Device Each Endpoint Interrupt Register
DEACHINTMSK	0x83c	Device Each Endpoint Interrupt Register Mask

18.6.7.2 寄存器详细列表

DCFG

Device Configuration Register
Offset Address: 0x800

Bits	Name	Access	Description	Reset
1:0	DevSpd	R/W	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. ■ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6MHz LS mode, you must do a soft reset. ■ 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
2	NZStsOUTHShk	R/W	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a non zero-length data packet during the OUT transaction of a control transfer's Status stage. ■ 1'b1: Send a STALL handshake on a non zero-length status OUT transaction and do not send the received OUT packet to the application. ■ 1'b0: Send the received OUT packet to the application (zero-length or non zero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
3	Ena32KHzSusp	R/W	Enable 32 KHz Suspend mode (Ena32KHzSusp) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. If FS PHY interface is chosen and this bit is set, the PHY clock during Suspend must be switched from 48 MHz to 32 KHz. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
10:4	DevAddr	R/W	Device Address (DevAddr) The application must program this field after every SetAddress control	0x0

Bits	Name	Access	Description	Reset
			command.	
12:11	PerFrInt	R/W	Periodic Frame Interval (PerFrInt) Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. ■ 2'b00: 80% of the (micro)frame interval ■ 2'b01: 85% ■ 2'b10: 90% ■ 2'b11: 95%	0x0
13	EnDevOutNak	R/W	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA mode. ■ 1'b0: The core does not set NAK after Bulk OUT transfer complete ■ 1'b1: The core sets NAK after Bulk OUT transfer complete This is a one time programmable bit after reset like any other DCFG register bits. This bit is valid only when OTG_EN_DESC_DMA == 1'b1.	0x0
14	XCVRDLY	R/W	Enables or disables delay between xcvr_sel and txvalid during device chirp ■ 1'b1: Enable delay between xcvr_sel and txvalid during Device chirp ■ 1'b0: No delay between xcvr_sel and txvalid during Device chirp	0x0
15	ErraticIntMsk	R/W	Mode: Device Erratic Error Interrupt Mask ■ 1'b1: Mask early suspend interrupt on erratic error ■ 1'b0: Early suspend interrupt is generated on erratic error	0x0
17:16	Reserved_800_17_16	RO	Reserved for future use.	
22:18	EPMisCnt	R/W	IN Endpoint Mismatch Count (EPMisCnt) This field is valid only in shared FIFO operation. The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.	0x8
23	DescDMA	R/W	Enable Scatter/Gather DMA in Device mode (DescDMA). When the Scatter/Gather DMA option is selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. Note: This bit must be modified only once after a reset. The following combinations are available for programming: ■ GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode ■ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid	0x0

Bits	Name	Access	Description	Reset
			■ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffer DMA mode ■ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode	
25:24	PerSchIntvl	R/W	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed only for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate For fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. When no periodic endpoints are active, the internal DMA engine services non-periodic endpoints, ignoring this field. After the specified time within a (micro)frame, the DMA switches to fetching for non-periodic endpoints. ■ 2'b00: 25% of (micro)frame. ■ 2'b01: 50% of (micro)frame. ■ 2'b10: 75% of (micro)frame. ■ 2'b11: Reserved.	0x0
31:26	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when DCFG.Ena32KHzSusp is set. It controls the resume period when the core resumes from suspend. The core counts for “ResValid” number of clock cycles to detect a valid resume when this bit is set.	0x2

DCTL

Device Control Register

Offset Address: 0x804

Bits	Name	Access	Description	Reset
0	RmtWkUpSig	R/W	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1–15 ms after setting it.If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 µs (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM	0x0

Bits	Name	Access	Description	Reset
			transaction is zero. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	SftDiscon	R/W	Soft Disconnect (SftDiscon) The application uses this bit to signal the DWC_otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. ■ 1' b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. ■ 1' b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host. Note: This bit can be also used for ULPI/FS Serial interfaces. Note: This bit is not impacted by a soft reset. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x1
2	GNPINNakSts	RO	Global Non-periodic IN NAK Status (GNPINNakSts) ■ 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. ■ 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	GOUTNakSts	RO	Global OUT NAK Status (GOUTNakSts) ■ 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. ■ 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	
6:4	TstCtl	R/W	Test Control (TstCtl) ■ 3'b000: Test mode disabled ■ 3'b001: Test_J mode ■ 3'b010: Test_K mode ■ 3'b011: Test_SE0_NAK mode ■ 3'b100: Test_Packet mode ■ 3'b101: Test_Force_Enable ■ Others: Reserved	0x0
7	SGNPInNak	RWC	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all nonperiodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation.	

Bits	Name	Access	Description	Reset
			The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.	
8	CGNPInNak	RWC	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK.	
9	SGOUTNak	RWC	Set Global OUT NAK (SGOUTNak) A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.	
10	CGOUTNak	RWC	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK.	
11	PWROnPrgDone	R/W	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is complete after a wake-up from Power Down mode.	0x0
12	Reserved_804_12	RO	Reserved for future use.	
14:13	GMC	R/W	Global Multi Count (GMC) GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic end points. ■ 2' b00: Invalid. ■ 2' b01: 1 packet. ■ 2' b10: 2 packets. ■ 2' b11: 3 packets. The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1. When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.	0x0
15	IgnrFrmNum	R/W	Ignore frame number for isochronous endpoints (IgnrFrmNum) Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and should not be programmed to 1.Non-Scatter/Gather DMA mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): This bit is not used when Threshold mode is enabled and should not be programmed to 1. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed. ■ When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. - 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame - 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts. Scatter/Gather DMA Mode	0x0

Bits	Name	Access	Description	Reset
			(GAHBCFG.DMAEn=1,DCFG.DescDMA=1): This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1. In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1. ■ 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. ■ 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor.	
16	NakOnBble	R/W	NAK on Babble Error (NakOnBble) Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received.	0x0
17	EnContOnBNA	R/W	Enable Continue on BNA (EnContOnBNA) This bit enables the DWC_otg core to continue on BNA for Bulk OUT and INTR OUT endpoints. With this feature enabled, when a Bulk OUT or INTR OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. ■ 1'b0: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor. ■ 1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt. This bit is valid only when OTG_EN_DESC_DMA == 1'b1. It is a one-time programmable after reset bit like any other DCTL register bits.	0x0
18	DeepSleepBESL Reject	R/W	Deep Sleep BESL Reject Core rejects LPM request with HIRD value greater than HIRD threshold programmed. NYET response is sent for LPM tokens with HIRD value greater than HIRD threshold. By default, the Deep Sleep BESL Reject feature is disabled.	0x0
31:19	Reserved_804_31_19	RO	Reserved for future use.	

DSTS

Device Status Register
Offset Address: 0x808

Bits	Name	Access	Description	Reset
0	SuspSts	RO	Suspend Status (SuspSts) In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspend state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend under the following conditions: ■ If there is any activity on the phy_line_state_i signal ■ If the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).	

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2:1	EnumSpd	RO	Enumerated Speed (EnumSpd) Indicates the speed at which the DWC_otg core has come up after speed detection through a chirp sequence. ■ 2'b00: High speed (PHY clock is running at 30 or 60 MHz) ■ 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) ■ 2'b10: Low speed (PHY clock is running at 6 MHz) ■ 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.	
3	ErrticErr	RO	Erratic Error (ErrticErr) The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted For at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the DWC_otg core goes into Suspend state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted because of an erratic error, the application can only perform a soft disconnect recover.	
7:4	Reserved_808_7_4	RO	Reserved for future use.	
21:8	SOFFN	RO	Frame or Microframe Number of the Received SOF (SOFFN) When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a Frame number. Note: This register may return a non zero value if read immediately after power on reset. In case the register bit reads non zero immediately after power on reset it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established.	
23:22	DevLnSts	RO	Device Line Status (DevLnSts) Indicates the current logic level USB data lines ■ Bit [23]: Logic level of D+ ■ Bit [22]: Logic level of D–	
31:24	Reserved_808_31_24	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

DIEPMSK

Device IN Endpoint Common Interrupt Mask Register

Offset Address: 0x810

Bits	Name	Access	Description	Reset
0	DiXferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk)	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	DiEPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	DiAHBErrMsk	R/W	AHB Error Mask (AHBErrMsk)	0x0
3	TimeOUTMsk	R/W	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	0x0
4	INTknTXFEmpMsk	R/W	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	0x0
5	INTknEPMisMsk	R/W	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	0x0
6	INEPNakEffMsk	R/W	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	0x0
7	Reserved_810_7	RO	Reserved for future use.	
8	TxfifoUndrnMsk	R/W	Fifo Underrun Mask (TxfifoUndrnMsk)	0x0
9	BNAInIntrMsk	R/W	BNA Interrupt Mask (BNAInIntrMsk) This bit is valid only when Device Descriptor DMA is enabled.	0x0
12:10	Reserved_810_12_10	RO	Reserved for future use.	
13	DiNAKMsk	R/W	NAK interrupt Mask (NAKMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved_810_31_14	RO	Reserved for future use.	

DOEPMSK

Device OUT Endpoint Common Interrupt Mask Register
Offset Address: 0x814

Bits	Name	Access	Description	Reset
0	XferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	EPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	AHBErrMsk	R/W	AHB Error (AHBErrMsk)	0x0
3	SetUPMsk	R/W	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	0x0
4	OUTTknEPdisMsk	R/W	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	0x0
5	StsPhseRcvdMsk	R/W	Status Phase Received Mask (StsPhseRcvdMsk)	0x0
6	Back2BackSETup Msk	R/W	Back-to-Back SETUP Packets Received Mask (Back2BackSETupMsk) Applies to control OUT endpoints only.	0x0
7	Reserved_814_7	RO	Reserved for future use.	
8	OutPktErrMsk	R/W	OUT Packet Error Mask (OutPktErrMsk)	0x0
9	BnaOutIntrMsk	R/W	BNA interrupt Mask (BnaOutIntrMsk)	0x0
11:10	Reserved_814_11 _10	RO	Reserved for future use.	
12	BbleErrMsk	R/W	Babble Error interrupt Mask (BbleErrMsk)	0x0
13	NAKMsk	R/W	NAK interrupt Mask (NAKMsk)	0x0
14	NYETMsk	R/W	NYET interrupt Mask (NYETMsk) Shadow: Yes Shadow Ctrl: vs_1t	0x0

Bits	Name	Access	Description	Reset
			Shadow Read Select: shrd_sel	
31:15	Reserved_814_31_15	RO	Reserved for future use.	

DAINT

Device All Endpoints Interrupt Register
Offset Address: 0x818

Bits	Name	Access	Description	Reset
15:0	InEpInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	OutEPInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	

DAINTMSK

Device Endpoints Interrupt Mask Register
Offset Address: 0x81c

Bits	Name	Access	Description	Reset
15:0	InEpMsk	R/W	IN EP Interrupt Mask Bits (InEpMsk) One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15 The value of this field depends on the number of IN endpoints that are configured. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	OutEpMsk	R/W	OUT EP Interrupt Mask Bits (OutEpMsk) One per OUT endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15 The value of this field depends on the number of OUT endpoints that are configured.	0x0

DIEPEMPMSK

Device IN Endpoint FIFO Empty Interrupt Mask Register
Offset Address: 0x834

Bits	Name	Access	Description	Reset
15:0	InEpTxfEmpMsk	R/W	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_834_31_16	RO	Reserved for future use.	

DEACHINT

Device Each Endpoint Interrupt Register
Offset Address: 0x838

Bits	Name	Access	Description	Reset
15:0	EchInEpInt	RO	IN Endpoint Interrupt Bits (EchInEpInt) One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	EchOutEPInt	RO	OUT Endpoint Interrupt Bits (EchOutEPInt) One bit per OUT endpoint: ■ Bit 16 for OUT endpoint 0 ... ■ Bit 31 for OUT endpoint 15	

DEACHINTMSK

Device Each Endpoint Interrupt Register Mask
Offset Address: 0x83c

Bits	Name	Access	Description	Reset
15:0	EchInEpMsk	R/W	IN EP Interrupt Mask Bits (EchInEpMsk) One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	EchOutEpMsk	R/W	OUT EP Interrupt Mask Bits (EchOutEpMsk) One per OUT Endpoint: ■ Bit 16 for IN endpoint 0 ... ■ Bit 31 for endpoint 15	0x0

18.7 SARADC

18.7.1 概述

SARADC 为模拟信号数字转换控制器。本芯片有最多 6 个独立通道可供使用

18.7.2 特点

- 控制器工作频率 12.5MHz;
- 扫描频率不能高于 320K/s；

- 12bit 采样精度，6 个独立通道；
- 可一次触发三个信道依序扫描；
- 扫描完成自动上报中断;

18.7.3 工作方式

CPU 配置扫描信道，可同时配置 6 个信道，启动 SARADC 进行通道扫描。通道扫描完成所有使能通道后，通过中断通知系统扫描完成，CPU 可以获取转换结果。

18.7.4 SARADC 寄存器概览

SARADC Base address 0x030F0000
RTCSYS_SARADC Base address 0x0502C000

Name	Address Offset	Description
saradc_ctrl	0x004	control register
saradc_status	0x008	staus register
saradc_cyc_set	0x00c	saradc waveform setting register
saradc_ch1_result	0x014	channel 1 result register
saradc_ch2_result	0x018	channel 2 result register
saradc_ch3_result	0x01c	channel 3 result register
saradc_intr_en	0x020	interrupt enable register
saradc_intr_clr	0x024	interrupt clear register
saradc_intr_sta	0x028	interrupt status register
saradc_intr_raw	0x02c	interrupt raw status register

18.7.5 SARADC 寄存器描述

saradc_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_saradc_en	RWS	when re_saradc_en is set , saradc start to measure the channels enabled in reg_saradc_sel	
3:1	Reserved			
7:4	reg_saradc_sel	R/W	select channel (1~3)	0x0
31:8	Reserved			

saradc_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	sta_saradc_busy	RO	busy rise when re_saradc_en is set	
3:1	Reserved			
7:4	sta_saradc_ch_busy	RO	per channel busy status	
15:8	Reserved			
19:16	sta_saradc_st	RO	fsm status for debug	

Bits	Name	Access	Description	Reset
24:20	sta_saradc_cycle	RO	sample cycle for debug	
31:25	Reserved			

saradc_cyc_set

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
4:0	reg_saradc_cyc_settling	R/W	saradc startup cycle = 1 + reg_saradc_cyc_settling , default is 16 cycle	0xF
7:5	Reserved			
11:8	reg_saradc_cyc_samp	R/W	saradc sample window = 1 + reg_saradc_cyc_samp , default is 4 cycle	0x3
15:12	reg_saradc_cyc_clkdiv	R/W	saradc clock divider , freq = ip_clk/(1+clk_div) , default is 25M/2 = 12.5M = 80ns	0x1
19:16	reg_saradc_cyc_comp	R/W	saradc compare cycle = 1+ reg_saradc_cyc_comp , default is 12 cycle	0xB
31:20	Reserved			

saradc_ch1_result

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch1_result	RO	ch1 measure result	
14:12	Reserved			
15	sta_saradc_ch1_valid	RO	ch1 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

saradc_ch2_result

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch2_result	RO	ch2 measure result	
14:12	Reserved			
15	sta_saradc_ch2_valid	RO	ch2 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

saradc_ch3_result

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch3_result	RO	ch3 measure result	
14:12	Reserved			
15	sta_saradc_ch3_valid	RO	ch3 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	

Bits	Name	Access	Description	Reset
31:16	Reserved			

saradc_intr_en

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_en	R/W	interrupt enable (mask)	0x0
31:1	Reserved			

saradc_intr_clr

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_clr	RWC	interrupt clear	
31:1	Reserved			

saradc_intr_sta

Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_sta	RO	interrup masked status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

saradc_intr_raw

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_raw	RO	interrupt raw status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

18.8 Temperature sensor

18.8.1 概述

本芯片内建两温度感测器，可以由 CPU 周期性监控芯片温度。可以在芯片过热无响应时触发电源管理模块进行系统关机重置以避免过热之危险。

18.8.2 工作方式

- 单次量测时间配置: reg_tempsen_accsel = 1 (1024T) 配置状况下，单次量测时间所需为 (1/(25M/12)*(1024+2+64) ~ 523.2us
- 周期量测时间配置: reg_tempsen_auto_prediv 预设为 24，使得 reg_tempsen_auto_cycle 每单位为 1us 。 两次量测时间之周期需配置

reg_tempsen_auto_cycle 大于 524。 例如每秒量测一次则配置 reg_tempsen_auto_cycle = 1000000。

- 量测通道配置: 若配置 reg_tempsen_sel = 3，则每次触发两个温度感测器同时量测。
- 配置高低温监控温度: 配置触发高温警报、低温回复之温度 tempsen_chx_temp_th
- 配置中断
- 使能温度感测器进行量测: 配置 reg_tempsen_en = 1 进行量测，等待中断

表格 18-14 Tempsensor 中断信号描述表

中断位	信号		描述
[0]	Irq_Temp0_measure		Tempsens0 量测完成
[1]	Irq_Temp1_measure		Tempsens1 量测完成
[2]	reserved		保留
[3]	reserved		保留
[4]	Irq_Temp0_over_high_level		Tempsens0 温度大于、等于高温临界值
[5]	Irq_Temp1_over_high_level		Tempsens1 温度大于、等于高温临界值
[6]	reserved		保留
[7]	reserved		保留
[8]	Irq_Temp0_under_low_level		Tempsens0 温度小于、等于低温临界值
[9]	Irq_Temp1_under_low_level		Tempsens1 温度小于、等于低温临界值
[10]	reserved		保留
[11]	reserved		保留
[12]	Irq_Temp0_over_high_cont		Tempsens0 温度大于、等于高温临界值已达 reg_tempsen_ovhl_cnt_to_irq 次数
[13]	Irq_Temp1_over_high_cont		Tempsens1 温度大于、等于高温临界值已达 reg_tempsen_ovhl_cnt_to_irq 次数
[14]	reserved		保留
[15]	reserved		保留
[16]	Irq_Temp0_under_low_cont		Tempsens0 温度小于、等于低温临界值已达 reg_tempsen_udll_cnt_to_irq 次数
[17]	Irq_Temp1_under_low_cont		Tempsens1 温度小于、等于低温临界值已达

中断位	信号		描述
			reg_tempsen_udll_cnt_to_irq 次数
[18]	reserved		保留
[19]	reserved		保留
[20]	Irq_Temp0_overheat		Tempsens0 温度大于、等于过热临界值
[21]	Irq_Temp1_overheat		Tempsens1 温度大于、等于过热临界值
[22]	reserved		保留
[23]	reserved		保留

图表 18-43 温度量测时间、计数与中断发生关系图

- 检视温度量测结果: sta_tempsen_chX_result 记载前一次完成之温度量测结果，sta_tempsen_chX_max_result 记载曾经量测到的温度最大值，tempsen_chX_temp_th_cnt 记载连续高温低温次数。
- 配置过热保护之温度 reg_tempsen_overheat_th、过热复位请求倒数时间 reg_tempsen_overheat_cycle 并使能 reg_overheat_reset_en。
  再参照 RTC 寄存器章节，配置 RTC 寄存器 hw_thm_shdn_en、RTC_EN_THM_SHDN 、RTC_THM_SHDN_AUTO_REBOOT 使能过热触发下电或重启。在发生过热时，temp sensor controller 会先发出中断并开始倒数，sta_tempsen_overheat_countdown 等于 1 时，后触发 RTC 的下电保护请求。若在此之前软件有介入并进行温控后可以配置 reg_overheat_reset_clr 清除倒数，但在下次量测结果依然过热时，过热保护中断依然会再次触发与倒数。

18.8.3 Temperature sensor 寄存器概览

BassAddress: 0x030A0000

Name	Address Offset	Description
tempsen_version	0x000	ip version number
tempsen_ctrl	0x004	control register
tempsen_status	0x008	staus register
tempsen_set	0x00c	temperature sensor macro setting
tempsen_intr_en	0x010	interrupt enable
tempsen_intr_clr	0x014	interrupt clear
tempsen_intr_sta	0x018	interupt status
tempsen_intr_raw	0x01c	interrupt raw status
tempsen_ch0_result	0x020	temperature sensor channel 0 result
tempsen_ch1_result	0x024	temperature sensor channel 1 result
tempsen_ch0_temp_th	0x040	temperature sensor channel 0 threshold
tempsen_ch1_temp_th	0x044	temperature sensor channel 1 threshold
Overheat_th	0x060	overheat threshold register
tempsen_auto_period	0x064	auto sample setting register
tempsen_overheat_ctrl	0x068	overheat control register
tempsen_overheat_countdown	0x06c	overheat status register
tempsen_ch0_temp_th_cnt	0x070	counter of channel 0 over/under threshold event
tempsen_ch1_temp_th_cnt	0x074	counter of channel 1 over/under threshold event

18.8.4 Temperature sensor 寄存器描述

tempsen_version

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	reg_ip_version	RO	verision 1.0	

tempsen_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_tempsen_en	R/W	when re_tempsen_en is set , tempsen start to measure the channel set in reg_tempsen_sel	0x0
3:1	Reserved			
7:4	reg_tempsen_sel	R/W	temperature sense channel selection	0x0
15:8	Reserved			
23:16	reg_tempsen_ovhl_cnt_to_irq	R/W	counting threshold of high temperature	0x8
31:24	reg_tempsen_udll_cnt_to_irq	R/W	counting threshold of low temperature	0x8

tempsen_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	sta_tempsen_busy	RO	busy status rise when re_tempsen_en is set	
31:1	Reserved			

tempsen_set

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	reg_tempsen_bgen	R/W	sensor macro bandgap enable	0x0
1	reg_tempsen_chopen	R/W	sensor macro chopper function enable	0x1
2	reg_tempsen_choppol	R/W	sensor macro chopper polarity when CHOPEN=0	0x1
3	reg_tempsen_clkpol	R/W	sensor macro clock polarity when DA_TEMPSEN_EN=0	0x1
5:4	reg_tempsen_chopsel	R/W	sensor macro chop period, 0:128T, 1:256T, 2:512T, 3:1024T	0x2
7:6	reg_tempsen_accsel	R/W	sensor macro accumulate period, 0:512T, 1:1024T, 2:2048T, 3:4096T	0x1
15:8	reg_tempsen_cyc_clkdiv	R/W	clock divider for sensor macro, freq = ip_clk/(1+clk_div) , default is 25M/12 =2.083M , T = 0.48us	0xB
17:16	reg_tempsen_tsel	R/W	sensor macro test selection, please keep 0	0x0
18	reg_tempsen_en_bjt_test	R/W	sensor macro test selection, please keep 0	0x0
31:19	Reserved			

tempsen_intr_en

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_en	R/W	interrupt enable	0x0

tempsen_intr_clr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_clr	RWC	interrupt clear	

tempsen_intr_sta

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_sta	RO	interrupt masked status	

tempsen_intr_raw

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_raw	RO	interrupt raw status: [3:0] ch3~ch0 measurement finish [7:4] ch3~ch0 measurement result is higher than high temperature threshold [11:8] ch3~ch0 measurement result is lower than low temperature threshold [15:12] ch3~ch0's high temperature event count is more than threshold [19:16] ch3~ch0's low temperature event count is more than threshold [23:20] ch3~ch0 measurement result is higher than overheat temperature	

tempsen_ch0_result

Offset Address: 0x020

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch0_result	RO	channel 0 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch0_max_result	RO	channel 0 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch0_max_result	RWC	write 1 to clear channel 0 max temperature measurement result	

tempsen_ch1_result

Offset Address: 0x024

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch1_result	RO	channel 1 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch1_max_result	RO	channel 1 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch1_max_result	RWC	write 1 to clear channel 1 max temperature measurement result	

tempsen_ch0_temp_th

Offset Address: 0x040

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch0_hi_th	R/W	channel 0 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch0_lo_th	R/W	channel 0 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

tempsen_ch1_temp_th

Offset Address: 0x044

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch1_hi_th	R/W	channel 1 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch1_lo_th	R/W	channel 1 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

Overheat_th

Offset Address: 0x060

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_overheat_th	R/W	overheat temperature threshold	0x0
31:13	Reserved			

tempsen_auto_period

Offset Address: 0x064

Bits	Name	Access	Description	Reset
23:0	reg_tempsen_auto_cycle	R/W	auto measure period. T_measure = reg_tempsen_auto_cycle*T_prediv	0x0

Bits	Name	Access	Description	Reset
31:24	reg_tempsen_auto_prediv	R/W	a predivider setting for auto measure period. T_prediv = (25M/(reg_tempsen_auto_prediv+1))	0x18

tempsen_overheat_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
29:0	reg_tempsen_overheat_cycle	R/W	After overheat event happens, the cycle count will be load to a counter and trigger counting down. when counting down to 1, a reset signal will be issue to power control unit.	0x100000
30	reg_overheat_reset_clr	RWC	write 1 to stop overheat reset counting.	
31	reg_overheat_reset_en	R/W	enable overheat reset counting down.	0x0

tempsen_overheat_countdown

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
29:0	sta_tempsen_overheat_countdown	RO	overheat reset counter	
30	Reserved			
31	sta_overheat_reset	RO	overheat reset signal status	

tempsen_ch0_temp_th_cnt

Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	sta_ch0_over_hi_temp_th_cnt	RO	channel 0 high temperature event count status	
15:8	sta_ch0_under_lo_temp_th_cnt	RO	channel 0 low temperature event count status	
16	reg_ch0_temp_th_cnt_clr	RWC	write 1 to clear channel 0 temperature event count	
31:17	Reserved			

tempsen_ch1_temp_th_cnt

Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	sta_ch1_over_hi_temp_th_cnt	RO	channel 1 high temperature event count status	
15:8	sta_ch1_under_lo_temp_th_cnt	RO	channel 1 low temperature event count status	
16	reg_ch1_temp_th_cnt_clr	RWC	write 1 to clear channel 1 temperature event count	
31:17	Reserved			

18.9 PWM

18.9.1 概述

芯片提供 1 组 4 路独立的 PWM 信号输出。

18.9.2 特点

PWM 的时钟来源为 100MHz 和 148.5MHz 二选一，默认为 100MHz。4 路 PWM 皆可独立操作：

- 内部有 30-bit 计数器，输出周期和高/低电平拍数皆可配置
- 支持最高 50MHz (100MHz/2) 或 74.25MHz (148.5MHz/2) 输出，最低约 0.093Hz (100MHz/(2^30-1)) 或 0.138Hz (148.5MHz/(2^32-1))
- 支持连续输出(PWMMODE = 0)和固定脉冲个数输出(PWMMODE = 1)两种模式
- 支持 4 路 PWM 同步输出模式(SHIFTMODE = 1)，4 路 PWM 输出可通过配置寄存器调控相差

18.9.3 工作方式

PWM 基本配置流程如下(以 PWM0 为例)：

1. 基于选定的时钟源，通过计算得到需输出的方波周期和低电平拍数
2. 将对应值写入寄存器 HLPERIOD0、PERIOD0
3. 若要操作在连续输出模式，配置 PWMMODE 为 0，将 PWMSTART[0]设 1 后，PWM0 开始输出，直到 PWMSTART[0]设 0 后结束输出
4. 若要操作在固定脉冲个数输出模式，配置 PWMMODE 为 1，需输出的方波数目写入寄存器 PCOUNT0。将 PWMSTART[0]设 1 后，PWM0 开始输出，达到设定的方波数后自动结束，状态寄存器 PWMDONE 由 0 转 1

图表 18-44 PWM Continuous mode

图表 18-45 PWM Pulse count mode

例如：要输出一个频率为 1MHz，低电平占比 75%，脉冲个数为 16 的波形

1. 使用默认 100MHz 时钟源，周期数(PERIOD0)配置为 100MHz / 1MHz = 100，低电平数(HLPERIOD0)配置为 100 x 75% = 75。脉冲数(PCOUNT0)配置为 16
2. PWMSTART[0]写 1 后开始输出波形
3. 读取寄存器 PWMDONE[0]直到值为 1，表示输出完成
4. 可读取寄存器 PULSECOUNT0，确认输出脉冲数状态值

若要再次使能 PWM，需要先将 PWMSTART[0]写 0 再写 1，以将计数器与状态寄存器复位。

当 4 路 PWM 要操作在同步输出模式时，首先配置 SHIFTMODE 为 1，流程如下：

1. 配置 HLPERIOD0/PERIOD0, HLPERIOD1/PERIOD1, HLPERIOD2/PERIOD2, HLPERIOD3/PERIOD3 为相同值
2. 依 4 路方波波形需要错开的 phase 差，配置适当值到寄存器 SHIFTCOUNT0, SHIFTCOUNT1, SHIFTCOUNT2, SHIFTCOUNT3
3. 配置 PWMSTART[3:0]为 4'hF，并将 SHIFTSTART 设 1，4 路的计数器同时开始计数，并在计数器值等于 SHIFTCOUNTn 时输出第 n 路的 PWM 波形

图表 18-46 PWM Continuous Shift Mode

例如：要同时输出 4 路频率皆为 1KHz，低电平占比 75%的方波，每路波形依序错开 1/4 周期

1. 使用默认 100MHz 时钟源，周期数配置为 100MHz / 1KHz = 100,000，低电平数配置为 100,000 x 75% = 75,000
2. 配置 SHIFTCOUNT0 = 0，SHIFTCOUNT1 = 100,000 x 1/4 = 25,000，SHIFTCOUNT2 = 100,000 x 2/4 = 50,000，SHIFTCOUNT3 = 100,000 x 3/4 = 75,000
3. PWMSTART[3:0]设为 4'hF，并将 SHIFTSTART 设 1，4 路 PWM 依序输出第一个脉冲

将 SHIFTSTART 设 0 后结束输出，读取寄存器 PWMDONE[3:0]直到值为 4'hF，表示 4 路皆输出完成。

18.9.4 PWM 寄存器概览

PWM 寄存器概览如表格 18-15 所示。

表格 18-15 PWM 寄存器概览

Name	Address Offset	Description
HLPERIOD0	0x000	PWM0 低电平拍数
PERIOD0	0x004	PWM0 方波周期拍数
HLPERIOD1	0x008	PWM1 低电平拍数
PERIOD1	0x00c	PWM1 方波周期拍数

Name	Address Offset	Description
HLPERIOD2	0x010	PWM2 低电平拍数
PERIOD2	0x014	PWM2 方波周期拍数
HLPERIOD3	0x018	PWM3 低电平拍数
PERIOD3	0x01c	PWM3 方波周期拍数
POLARITY	0x040	PWM 模式设定
PWMSTART	0x044	使能 PWM 输出
PWMDONE	0x048	PWM 结束状态
PWMUPDATE	0x04c	动态加载 PWM 周期参数
PCOUNT0	0x050	设定 PWM0 脉冲数
PCOUNT1	0x054	设定 PWM1 脉冲数
PCOUNT2	0x058	设定 PWM2 脉冲数
PCOUNT3	0x05c	设定 PWM3 脉冲数
PULSECOUNT0	0x060	PWM0 已输出脉冲计数器状态
PULSECOUNT1	0x064	PWM1 已输出脉冲计数器状态
PULSECOUNT2	0x068	PWM2 已输出脉冲计数器状态
PULSECOUNT3	0x06c	PWM3 已输出脉冲计数器状态
SHIFTCOUNT0	0x080	同步模式 PWM0 初始相差
SHIFTCOUNT1	0x084	同步模式 PWM1 初始相差
SHIFTCOUNT2	0x088	同步模式 PWM2 初始相差
SHIFTCOUNT3	0x08c	同步模式 PWM3 初始相差
SHIFTSTART	0x090	PWM 同步模式使能
PWM_OE	0x0d0	PWM IO output enable

18.9.5 PWM 寄存器描述

HLPERIOD0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
29:0	HLPERIOD0	R/W	PWM0 低电平拍数 (单位是 clk_pwm) 当 POLARITY[0]为 0，此值为低电平拍数，当 POLARITY[0]为 1，此值为高电平拍数。	0x1

PERIOD0

Offset Address: 0x004

Bits	Name	Access	Description	Reset
29:0	PERIOD0	R/W	PWM0 方波周期拍数 (单位是 clk_pwm) 注意 PERIOD 值必须大于 HLPERIOD 值。	0x2

HLPERIOD1

Offset Address: 0x008

Bits	Name	Access	Description	Reset
29:0	HLPERIOD1	R/W	PWM1 低电平拍数 (单位是 clk_pwm) 当 POLARITY[1]为 0，此值为低电平拍	0x1

Bits	Name	Access	Description	Reset
			数，当 POLARITY[1]为 1，此值为高电平拍数。	

PERIOD1

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
29:0	PERIOD1	R/W	PWM1 方波周期拍数 (单位是 clk_pwm) 注意 PERIOD 值必须大于 HLPERIOD 值。	0x2

HLPERIOD2

Offset Address: 0x010

Bits	Name	Access	Description	Reset
29:0	HLPERIOD2	R/W	PWM2 低电平拍数 (单位是 clk_pwm) 当 POLARITY[2]为 0，此值为低电平拍数，当 POLARITY[2]为 1，此值为高电平拍数。	0x1

PERIOD2

Offset Address: 0x014

Bits	Name	Access	Description	Reset
29:0	PERIOD2	R/W	PWM2 方波周期拍数 (单位是 clk_pwm) 注意 PERIOD 值必须大于 HLPERIOD 值。	0x2

HLPERIOD3

Offset Address: 0x018

Bits	Name	Access	Description	Reset
29:0	HLPERIOD3	R/W	PWM3 低电平拍数 (单位是 clk_pwm) 当 POLARITY[3]为 0，此值为低电平拍数，当 POLARITY[3]为 1，此值为高电平拍数。	0x1

PERIOD3

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
29:0	PERIOD3	R/W	PWM3 方波周期拍数 (单位是 clk_pwm) 注意 PERIOD 值必须大于 HLPERIOD 值。	0x2

POLARITY

Offset Address: 0x040

Bits	Name	Access	Description	Reset
3:0	POLARITY	R/W	PWM0~3 信号极性 [n] = 0: PWMn 默认为低电平输出 [n] = 1: PWMn 默认为高电平输出	0x0
7:4	Reserved			
11:8	PWMMODE	R/W	PWM0~3 操作模式 [n+8] = 0: PWMn 操作在连续出模式 [n+8] = 1: PWMn 操作在固定脉冲输出	0x0

Bits	Name	Access	Description	Reset
			模式	
15:12	Reserved			
16	SHIFTMODE	R/W	使能 PWM 同步相位输出模式 0 = PWM0~3 皆操作在一般模式 1 = PWM0~3 操作在 4 路同步输出模式	0x0
19:17	Reserved			
20	pclk_force_en	R/W	APB 时钟门控控制 0 = 使能 APB 时钟门控在闲置时自动关闭时钟 1 = APB 时钟保持恒开	0x0
31:21	Reserved			

PWMSTART

Offset Address: 0x044

Bits	Name	Access	Description	Reset
3:0	PWMSTART	R/W	使能 PWM0~3 [n] = 0: 停止 PWMn [n] = 1: 输出 PWMn 当 PWMMODE 设 0，将 bit n 写 0 再写 1 以启动 PWMn 开始输出，直到将 bit n 写 0 则停止输出。当 PWMMODE 设 1，将 bit n 写 1 启动 PWMn 开始输出，当输出的脉冲个数等于 PCOUNTn 值时自动停止输出。当 SHIFTMODE 设 1， PWMSTART[3:0]做为 PWM0~3 的信号输出使能，PWM 启动改由 SHIFTSTART 控制。	0x0
31:4	Reserved			

PWMDONE

Offset Address: 0x048

Bits	Name	Access	Description	Reset
3:0	PWMDONE	RO	PWM0~3 结束输出状态 [n] = 1: PWMn 已结束输出 当 PWMSTART[n]由 0 设 1 后，会将寄存器值清为 0。	
31:4	Reserved			

PWMUPDATE

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
3:0	PWMUPDATE	R/W	动态加载 PWM 参数 当 PWMSTART 由 0 写 1 时，寄存器值(HLPERIODn, PERIODn)会在 PWM 内部暂存住。若想在 PWM 输出当中动态改变波形，将新值写入 HLPERIODn 和 PERIODn 后，将 PWMUPDATE[n]写 1 再写 0 才能让新值生效。	0x0
31:4	Reserved			

PCOUNT0

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	PCOUNT0	R/W	PWM0 脉冲个数 (设定值必须大于 0) 只有当 PWMMODE[0]设 1 时有效。	0x1
31:24	Reserved			

PCOUNT1

Offset Address: 0x054

Bits	Name	Access	Description	Reset
23:0	PCOUNT1	R/W	PWM1 脉冲个数 (设定值必须大于 0) 只有当 PWMMODE[1]设 1 时有效。	0x1
31:24	Reserved			

PCOUNT2

Offset Address: 0x058

Bits	Name	Access	Description	Reset
23:0	PCOUNT2	R/W	PWM2 脉冲个数 (设定值必须大于 0) 只有当 PWMMODE[2]设 1 时有效。	0x1
31:24	Reserved			

PCOUNT3

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	PCOUNT3	R/W	PWM3 脉冲个数 (设定值必须大于 0) 只有当 PWMMODE[3]设 1 时有效。	0x1
31:24	Reserved			

PULSECOUNT0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT0	RO	PWM0 已输出的脉冲个数状态	
31:24	Reserved			

PULSECOUNT1

Offset Address: 0x064

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT1	RO	PWM1 已输出的脉冲个数状态	
31:24	Reserved			

PULSECOUNT2

Offset Address: 0x068

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT2	RO	PWM2 已输出的脉冲个数状态	
31:24	Reserved			

PULSECOUNT3

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT3	RO	PWM3 已输出的脉冲个数状态	
31:24	Reserved			

SHIFTCOUNT0

Offset Address: 0x080

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT0	R/W	PWM0 第一个脉冲输出的相差 (单位是 clk_pwm) 只有当 SHIFTMODE 设 1 时有效。	0x0
31:24	Reserved			

SHIFTCOUNT1

Offset Address: 0x084

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT1	R/W	PWM1 第一个脉冲输出的相差 (单位是 clk_pwm) 只有当 SHIFTMODE 设 1 时有效。	0x0
31:24	Reserved			

SHIFTCOUNT2

Offset Address: 0x088

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT2	R/W	PWM2 第一个脉冲输出的相差 (单位是 clk_pwm) 只有当 SHIFTMODE 设 1 时有效。	0x0
31:24	Reserved			

SHIFTCOUNT3

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT3	R/W	PWM3 第一个脉冲输出的相差 (单位是 clk_pwm) 只有当 SHIFTMODE 设 1 时有效。	0x0
31:24	Reserved			

SHIFTSTART

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SHIFTSTART	R/W	在同步模式时使能 PWM 输出 当 SHIFTMODE 设 1 时，此寄存器写 1 后开始输出 PWM0~3。	0x0
31:1	Reserved			

PWM_OE

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
3:0	PWM_OE	R/W	PWM0~3 IO output enable 1 = IO 为输出, 0 = IO 为输入	0x0
31:4	Reserved			

18.10 Key scan

18.10.1 概述

Keyscan 最多支援 8x8 = 64 个按键的矩阵. 若按键不需要这么多. 能自由决定要 mask 掉或保留哪些 row 或 column. 视软件需要可以选用 snapshot 模式及 FIFO 模式来取得按键资讯.

18.10.2 工作方式

图表 18-47 Keyscan 架构框图

状态机 (FSM) 在休息模式(没有按下任何键)时, 所有的 row 都输出 0 , 而 col 都是输入模

式并开启弱上拉 (弱上拉的设定在 ioblk 相映的寄存器, 不在 keyscan 模块). 当任何一个按键被按下时, col 端经过 debounce 后会看到非全 1 的值. 说明有按键被按着. 这时 FSM 会启动一次扫描. 依序让 Row [0] -> Row[7] 一次只有一个 bit 输出 0 (其余都在 HiZ 高阻态) . 每次的结果会更新入一个 array 中.

FSM 会不断的循环扫描, 直到所有 row 扫回的 col 又都是全 1 表示已没有按着的按键, 才会又会到休息模式下 (所有 row 输出 0) .

18.10.3 基本设定

KEYSCAN0 里的 reg_row_mask, reg_col_mask 及 reg_enable 是当没有使用到 8x8 的矩阵时, 可以选择性把某些 IO 档掉, 不输出, 也不参考其输入. Default 是全关的. 所以需要打开.
KEYSCAN_CONFIG2 里的 reg_db_col 决定了 column input 需要经过多少时间的 debounce 才可以使用.
KEYSCAN_CONFIG1 里的 reg_slow_div 决定了 IP 的 FSM 每个 stage 在停留的时间. 要记得这个数字一定要比 debounce 的时间大. 不然 IO 上转态后未完成 debounce 就进行判读会错乱.

KEYSCAN_CONFIG3 里的 reg_wait_cntr 则可以用来降低扫描速度. 因为只要按键按着, keyscan 模块就会不停的作扫描. 这个 counter 可以控制在每次启动新一轮扫描前, 固定等待一定的时间. 以减低扫描频率.

18.10.4 使用 FIFO 模式.

使用 FIFO 模式时, IP 每次扫进来的 64 个按键值会存在 array 里. 每次只要有任何一个 key 的状态与上一次扫描内容不一样, 他就会把这个 key 的 index 及当下的值 (0/1) push 进 FIFO 内. 所以 [5:0] 里的数字指定了是哪一个按键. [6] 表明是按下 (0) 或是放开 (1) . 当 FIFO 不为空时, 就会发出 IRQ. 这个模式的优点是省略掉软件一个一个 bit 作检查是哪个 bit 有改变的复锁工作. 缺点是 KEY_SCAN_FIFO 是个 read 会自动 pop 的 register, 要小心操作.

打开 KEYSCAN_IRQ_ENABLE 的 reg_irq_fifo_not_empty_enable
收到 IRQ 后, 读取 KEYSCAN_IRQ_FLAG 里的 reg_irq_fifo_not_empty , 然后检查 KEYSCAN_FIFO_STATUS 的 reg_fifo_not_empty. 则开始 read KEYSCAN_FIFO 里的内容.
直到清干净后, 再把 KEY_SCAN_IRQ_CLEAR 清掉. 结束 IRQ retine.

18.10.5 使用 snapshot array 模式.

使用 snapshot array 时, IP 内会目前扫进来的 64 按键的值存在一个 array 里. 若这个 array 的内容与 KEYSCAN_SNAPSHOT_ARRAY 内容不一样. 就会发 IRQ. 而软件可以触发 KEYSCAN_SNAPSHOT_TRIG, 把当前的 array 内容捉到 snapshot array 后, 再慢慢比对甚么内容与之前的认知有改变.

打开 KEYSCAN_IRQ_ENABLE 的 reg_irq_snapshot_change_enable.
收到 IRQ 后, 读取 trigger KEYSCAN_SNAPSHOT_TRIG, 然后解读 KEYSCAN_SNAPSHOT_ARRAY 的内容.
再把 KEY_SCAN_IRQ_CLEAR 清掉. 结束 IRQ retine.

18.10.6 Key scan 寄存器概览

Name	Address Offset	Description
KEYSCAN_CONFIG0	0x000	
KEYSCAN_CONFIG1	0x004	
KEYSCAN_CONFIG2	0x008	
KEYSCAN_CONFIG3	0x00c	
KEYSCAN_SNAPSHOT_ARRAY	0x014	
KEYSCAN_SNAPSHOT_TRIG	0x01c	
KEYSCAN_FIFO_STATUS	0x020	
KEYSCAN_FIFO	0x024	
KEYSCAN_IRQ_ENABLE	0x028	
KEYSCAN_IRQ_FLAG	0x02c	
KEYSCAN_IRQ_CLEAR	0x030	

18.10.7 Key scan 寄存器描述

KEYSCAN_CONFIG0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_row_mask	R/W	ROW[7:0] Mask 0 = enable 1 = disable	0xff
15:8	reg_col_mask	R/W	COL[7:0] Mask 0 = enable	0xff

Bits	Name	Access	Description	Reset
			1 = disable	
16	reg_enable	R/W	keyscan enable 0 = disable 1 = enable	0x0
31:17	Reserved			

KEYSCAN_CONFIG1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_slow_div	R/W	slow divider (MUST BE BIGGER THAN reg_db_col) Each step is IP clock frequency divide by reg_slow_div Scan frequency = IP clock freq / ((reg_slow_div+1) * (9+reg_wait_count+1)) IDLE -> ROW0 -> ROW1 -> ROW2->ROW3->ROW4->ROW5->ROW6 ->ROW7->UPDATE->WAIT->IDLE	0xff
31:24	Reserved			

KEYSCAN_CONFIG2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_db_col	R/W	column input debounce counter (IP clock cycle)	0x64
31:16	Reserved			

KEYSCAN_CONFIG3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_wait_cntr	R/W	wait interval between each scan (unit is reg_slow_div count)	0x10
31:8	Reserved			

KEYSCAN_SNAPSHOT_ARRAY

Offset Address: 0x014

Bits	Name	Access	Description	Reset
63:0	reg_cpu_snapshot_array	RO	CPU snapshot array result (0 = press, 1 = not press) [0] = Row 0 , Col 0 [1] = Row 0 , Col 1 … [7] = Row 0, Col 7 [8] = Row 1, Col 0 [63] = Row 7, Col 7 [N] = Row Y, Col X (N = Y*8 + X)	

KEYSCAN_SNAPSHOT_TRIG

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_cpu_snapshot_toggle	W1T	Write 1 to Trigger snapshot array to	

Bits	Name	Access	Description	Reset
			update When current result is different from snapshot result, irq happen To solve the IRQ, write 1 to trigger the snapshot array to copy from current array and start checking which bit is different from previous state	
31:1	Reserved			

KEYSCAN_FIFO_STATUS

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	reg_fifo_count	RO	FIFO content count 0 = Empty 1 = one content in FIFO N = N content in FIFO	
4	reg_fifo_not_empty	RO	FIFO not empty flag 0 = Empty 1 = Not empty	
31:5	Reserved			

KEYSCAN_FIFO

Offset Address: 0x024

Bits	Name	Access	Description	Reset
6:0	reg_fifo_rdata	ROC	read data from FIFO (Auto POP) - check FIFO empty-ness before read [6] 0 = press, 1 = not-press [5:0] = index Row = INT(index/8) Col = mod(index,8) 63 = Row 7 , Column 7 13 = Row 1 , Clumne 5	
31:7	Reserved			

KEYSCAN_IRQ_ENABLE

Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_enable	R/W	FIFO mode IRQ Enable 0 = Disable 1 = Enable	0x0
3:1	Reserved			
4	reg_irq_snapshot_change_enable	R/W	Snapshot mode IRQ Enable 0 = Disable 1 = Enable	0x0
31:5	Reserved			

KEYSCAN_IRQ_FLAG

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty	RO	FIFO not empty IRQ flag 0 = Empty 1 = Not empty	
3:1	Reserved			

Bits	Name	Access	Description	Reset
4	reg_irq_snapshot_change	RO	Snapshot change IRQ flag 0 = No change 1 = Change	
31:5	Reserved			

KEYSCAN_IRQ_CLEAR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_clear_w1t	W1T	FIFO not empty IRQ Clear (Write 1 clear)	
3:1	Reserved			
4	reg_irq_snapshot_change_clear_w1t	W1T	Snapshot Change IRQ Clear (Write 1 clear)	
31:5	Reserved			

18.11 Wiegand

18.11.1 概述

图表 18-48 wiegand 信号总线传输 0/1 的方式

Wiegand 介面使用两根单端的讯号, D0/D1. 当总线 idle 时都是 high, 在 D0 上出现一个 low pulse 表示传送了一个"0", 在 D1 上发现 low pulse, 就是传送一个 "1"

Wiegand 常用在门禁系统上, 有两种常用 format, Wiegand 26/34, 分别代表封包的 bit 数. 这两种 format 简介如下.

18.11.1.1 Wiegand 26

图表 18-49 Wiegand 26 格式

18.11.1.2 Wiegand 34

图表 18-50 Wiegand 34 format

F = Facility Code
U = User code

有些门禁卡背后有一串数字, 把他转成 hex 后,
Dec 0002262506 Ⓟ Hex : 22_85_EA
0x22 是他的 Facility code (Dec = 34)
0x85EA 是他的 user code (Dec = 34282)

图表 18-51 常见的磁卡, 磁扣的数字意义

后面看到的 34, 34282 是拆开后重新用十进制表示的 Facility code & user code

PS. 本 IP TX RX 均不处理 parity 的 insert 或 checking. 均由软件处理.

18.11.2 工作方式

图表 18-52 wiegand 架构框图

Wiegand 模块包含 TX 和 RX , 可用途单向或双向使用. 自己 TX 出去的时候, RX 会档住不看. 所以 RX 不会收到自己打出去的讯号. 而 TX 支援 push pull mode 或是 open drain mode.

18.11.2.1 TX

TX 前, 设好 TX 的 high time 和 low time, packet 的 bit 数, 是由 MSB 1st or LSB 1st. 然后把 data 放在 TX_BUFFER 寄存器后, 用 TX_TRIG 把 data 打出去.
TX 完整传输结束后, 可以靠 TX finish 的中断, 或是 polling TX_BUSY 的 status 来确定甚么时候可以打下一个 packet.

18.11.2.2 RX

RX 前, 设定好 debounce time, 预期接收 packet 的 bit 数. 当 D0 D1 出现 low pulse 时, RX

即开始把 data push 进 temp buffer 中. 当接收到 bit 数达到一个 packet 的预期 bit 数时, 会把 temp buffer 推到 RX_BUFFER 中, 发出中断, 请软件处理. 而 temp buffer 则继续接收下一笔 data.

如果 D0 D1 上出现 idle timeout 时, 即使 bit 数未到, 也会强迫视为一个 packet .

RX BUFFER 的高位元会记录这个 packet 一共收到的 bit 数, 是否是因为 timeout 造成的, 或是在软件读取前, 有发生 overflow .

每收到一个 packet, 可以靠 rx_buffer_rececived 的中断, 或是 RX_BUFFER_VALID 来判定 RX_BUFFER 中是否有有效的 data. RX Data 取走之后, 再 trigger RX_BUFFER_CLEAR 去清空 RX_BUFFER 以接收下一个 packet

18.11.3 Wiegand 寄存器概览

Name	Address Offset	Description
TX_CONFIG0	0x000	
TX_CONFIG1	0x004	
TX_CONFIG2	0x008	
TX_BUFFER	0x00c	
TX_TRIG	0x014	
TX_BUSY	0x018	
TX_DEBUG	0x01c	
RX_CONFIG0	0x020	
RX_CONFIG1	0x024	
RX_CONFIG2	0x028	
RX_BUFFER	0x02c	
RX_BUFFER_VALID	0x038	
RX_BUFFER_CLEAR	0x03c	
RX_DEBUG	0x040	
IRQ_ENABLE	0x044	
IRQ_FLAG	0x048	
IRQ_CLEAR	0x04c	

18.11.4 Wiegand 寄存器描述

TX_CONFIG0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
23:0	reg_tx_lowtime	R/W	TX Low width , unit = cycle	0xff
31:24	Reserved			

TX_CONFIG1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_tx_hightime	R/W	TX High width , unit = cycle	0xff
31:24	Reserved			

TX_CONFIG2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	reg_tx_bitcount	R/W	TX Frame bit count per transmit , unit = bit	0x18
7	Reserved			
8	reg_tx_msb1st	R/W	TX Transmit from MSB or LSB 0 : LSB 1st , from tx_buffer[0] --> tx_buffer[reg_tx_bitcount] 1 : MSB 1st , from tx_buffer[reg_tx_bitcount] --> tx_buffer[0]	0x0
15:9	Reserved			
16	reg_tx_opendrain	R/W	TX using push-pull mode or opendrain mode 0 : push-pull mode 1 : opendrain mode	0x0
31:17	Reserved			

TX_BUFFER

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
63:0	reg_tx_buffer	R/W	TX buffer content	0x00

TX_TRIG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_tx_trig_w1t	W1T	Trigger transmittion Write 1 trigger (please check reg_tx_busy before transmit)	
31:1	Reserved			

TX_BUSY

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_tx_busy	RO	0 = idle, allow to trigger transmission	

Bits	Name	Access	Description	Reset
			1 = busy, do not trigger any more tranmission or it will be ignore.	
31:1	Reserved			

TX_DEBUG

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	reg_tx_fsm	RO	TX Finite State Machine current state 0 : idle 1 : wait bus idle 2 : tx_start 3 : transmit low 4 : transmit high 5 : tx_stop	
7:3	Reserved			
14:8	reg_tx_pointer	RO	TX pointer current position indicate how many bit is still not yet send	
31:15	Reserved			

RX_CONFIG0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_rx_debounce	R/W	RX input debounce time (unit is cycle)	0xff
31:16	Reserved			

RX_CONFIG1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	reg_idle_timeout	R/W	Bus timeout cycle count When bus is idle for idle_timeout cycle, bus is expected to be back to idle If some bit has received but not yet accumulate to rx_bitcount, it will also treat as a complete packet.	0xfff

RX_CONFIG2

Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	reg_rx_bitcount	R/W	RX Expected Frame bit count , unit = bit	0x18
7	Reserved			
8	reg_rx_msb1st	R/W	RX Received sequence 0 : LSB 1st, 1st data is put in reg_rx_buffer[0]->[1]->[2].... 1 : MSB 1st, 1st data is put in reg_rx_buffer[reg_rx_bitcount]->[0]	0x0
11:9	Reserved			
12	reg_rx_enable	R/W	RX Enable 0 : disable 1 : Enable	0x0
31:13	Reserved			

RX_BUFFER

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
72:0	reg_rx_buffer	RO	RX Buffer [63:0] = fifo = Indicate received content [70:64] = fifo_bit_count = How many effective bit is in rx_buffer[63:0] [71] = idle_reach = This RX is terminate by bus idle timeout [72] = overflow = This RX just overwrite an un-read message	
95:73	Reserved			

RX_BUFFER_VALID

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_valid	RO	reg_rx_buffer validness 0 : not valid 1 : valid	
31:1	Reserved			

RX_BUFFER_CLEAR

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_clear_w1t	W1T	reg_rx_buffer clear (write 1 clear)	
31:1	Reserved			

RX_DEBUG

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_businidle	RO	bus in idle indication 0 : bus is not in idle 1 : bus is in idle more than reg_rx_idle_timeout cycle	
31:1	Reserved			

IRQ_ENABLE

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_enable	R/W	TX Finish IRQ Enable (to inform all data has being transmit, ready for next) 0 : Disable 1 : Enable	0x0
3:1	Reserved			
4	reg_irq_rx_overflow_enable	R/W	RX Overflow IRQ Enable 0 : Disable 1 : Enable	0x0
7:5	Reserved			
8	reg_irq_rx_received_enable	R/W	RX Received IRQ Enable 0 : Disable 1 : Enable	0x0
31:9	Reserved			

IRQ_FLAG

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish	RO	TX Finish IRQ Flag 0 : no IRQ 1 : IRQ (one tranmission has being completed)	
3:1	Reserved			
4	reg_irq_rx_overflow	RO	RX overflow IRQ Flag 0 : no IRQ 1 : IRQ (rx buffer is not pop and new data has overwrited)	
7:5	Reserved			
8	reg_irq_rx_received	RO	RX received IRQ Flag 0 : no IRQ 1 : RX buffer has new data	
31:9	Reserved			

IRQ_CLEAR

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_clear_w1t	W1T	TX Finish IRQ Clear , Write 1 to clear reg_irq_tx_finish flag	
3:1	Reserved			
4	reg_irq_rx_overflow_clear_w1t	W1T	RX Overflow IRQ Clear . Write 1 to clear reg_irq_rx_overflow flag	
7:5	Reserved			
8	reg_irq_rx_received_clear_w1t	W1T	RX Received IRQ Clear . Write 1 to clear reg_irq_rx_received flag	
31:9	Reserved			

18.12 IRRX 紅外接口

18.12.1 概述

通过 IRRX 單元接收外部紅外數据.

18.12.2 特點

1. 支持 NEC 編碼模式 (含 repeat code)
2. 支持 Philips RC5/RC6 編碼模式
3. 支持 Sony 編碼模式
4. 支持紅外線喚醒功能

18.12.3 工作方式

图表 18-53 IRRX 工作方式

軟件預先設定接收的紅外線資料格式, 当 IRRX 模塊接收到紅外線訊號後將其解碼, 符合預定格式的編碼透过中斷傳送給 CPU, CPU 再依据編碼進行相应操作.

18.12.4 IRRX 寄存器概览

Name	Address Offset	Description
IR_EN	0x000	
IR_MODE	0x004	
IR_CFG	0x008	
IR_FRAME	0x00c	
int_en	0x010	
int_clr	0x014	
int_msk	0x018	

Name	Address Offset	Description
int	0x01c	
int_raw	0x020	
IR_SYMBOL_CFG0	0x030	
IR_SYMBOL_CFG1	0x034	
IR_SYMBOL_CFG2	0x038	
IR_SYMBOL_CFG3	0x03c	
IR_SYMBOL_CFG4	0x040	
IR_SYMBOL_CFG5	0x044	
IR_SYMBOL_CFG6	0x048	
IR_SYMBOL_CFG7	0x04c	
IR_CLOCK_CTRL	0x050	
IR_DATA0	0x080	
IR_DATA1	0x084	
IR_DATA2	0x088	
IR_DATA3	0x08c	
IR_DATA4	0x090	
IR_NEC_DATA0	0x0a8	
IR_SONY_DATA0	0x0ac	
IR_SONY_DATA1	0x0b0	
IR_PHILIPS_DATA0	0x0b4	
IR_PHILIPS_DATA1	0x0b8	
IR_PRD_REC0	0x0e0	
IR_PRD_REC1	0x0e4	
IR_PRD_REC2	0x0e8	
IR_PRD_REC3	0x0ec	
IR_PRD_REC4	0x0f0	
IR_PRD_REC5	0x0f4	
SPARE_0	0xff0	
SPARE_1	0xff4	
SPARE_RO	0xff8	
DATA_CODE	0xffc	

18.12.5 IRRX 寄存器描述

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_ir_rx_en	R/W	ir receiver enable	0x0
1	reg_ir_rx_rst	R/W	ir receiver reset	0x0
15:2	Reserved			
16	reg_ir_init_done	RO	ir receiver ready	
31:17	Reserved			

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_ir_mode	R/W	ir receiver mode 0x0: pulse distance coding 0x2: bi-phase coding, RC5 0x3: bi-phase coding, RC6	0x0
7:2	Reserved			
8	reg_periodic_mode	R/W	periodic sample mode 0x1: periodic sample, reg_ic_mode is ignored	0x0
31:9	Reserved			

Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	reg_tick_prd	R/W	tick period	0x18
11:8	reg_sample_prd	R/W	sample period	0x9
14:12	reg_debounce	R/W	input signal debounce control	0x4
15	Reserved			
16	reg_import_inv	R/W	input signal polarity control	0x1
17	reg_export_inv	R/W	output data polarity control	0x0
18	reg_sony_format	R/W	tx transmit stop burst at frame end 0x1: SONY formate 0x0: NEC formate	0x0

Bits	Name	Access	Description	Reset
19	reg_repeat_support	R/W	support simplfied repeat code	0x1
20	reg_bit_edge_sel	R/W	data bit selection used in pulse distance coding 0x1: SONY formate 0x0: NEC formate	0x0
31:21	Reserved			

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_length	R/W	ir receiver data length	0x20
8	reg_slength	R/W	ir receiver repeate code length 0x1: TC9012 0x0: others	0x0
15:9	Reserved			
18:16	reg_lead_p_ratio	R/W	ratio of leading symbol period and strobe period, used in periodic sample mode	0x4
31:19	Reserved			

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_en	R/W	rx_done interrupt enable	0x1
1	reg_frame_err_int_en	R/W	frame_err interrupt enable	0x1
2	reg_frame_ovf_int_en	R/W	frame_ovf interrupt enable	0x1
3	reg_release_int_en	R/W	release interrupt enable	0x1
4	reg_repeat_int_en	R/W	repeat interrupt enable	0x1
31:5	Reserved			

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_clr	W1T	rx_done interrupt clear	
1	reg_frame_err_int_clr	W1T	frame_err interrupt clear	
2	reg_frame_ovf_int_clr	W1T	frame_ovf interrupt clear	
3	reg_release_int_clr	W1T	release interrupt clear	

Bits	Name	Access	Description	Reset
4	reg_repeat_int_clr	W1T	repeat interrupt clear	
31:5	Reserved			

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_msk	R/W	rx_done interrupt mask	0x0
1	reg_frame_err_int_msk	R/W	frame_err interrupt mask	0x0
2	reg_frame_ovf_int_msk	R/W	frame_ovf interrupt mask	0x0
3	reg_release_int_msk	R/W	release interrupt mask	0x0
4	reg_repeat_int_msk	R/W	repeat interrupt mask	0x0
31:5	Reserved			

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_rx_done_int	RO	rx_done interrpt	
1	reg_frame_err_int	RO	frame_err interrpt	
2	reg_frame_ovf_int	RO	frame_ovf interrpt	
3	reg_release_int	RO	release interrpt	
4	reg_repeat_int	RO	repeat interrpt	
31:5	Reserved			

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_raw	RO	rx_done interrupt raw value	
1	reg_frame_err_int_raw	RO	frame_err interrupt raw value	
2	reg_frame_ovf_int_raw	RO	frame_ovf interrupt raw value	
3	reg_release_int_raw	RO	release interrupt raw value	
4	reg_repeat_int_raw	RO	repeat interrupt raw value	
31:5	Reserved			

Offset Address: 0x030

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_p	R/W	lead symbol postive interval	0x383
15:12	Reserved			

Bits	Name	Access	Description	Reset
23:16	reg_ir_rx_lead_p_tol	R/W	lead symbol postive interval tolerance reg_ir_rx_lead_p + reg_ir_rx_lead_p_tol <= 12'FFF reg_ir_rx_lead_p - reg_ir_rx_lead_p_tol >= 12'000	0x48
31:24	Reserved			

Offset Address: 0x034

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_n	R/W	lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_lead_n_tol	R/W	lead symbol negtive interval tolerance reg_ir_rx_lead_n + reg_ir_rx_lead_n_tol <= 12'FFF reg_ir_rx_lead_n - reg_ir_rx_lead_n_tol >= 12'000	0x24
31:24	Reserved			

Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	reg_ir_rx_stop	R/W	stop symbol postive interval	0x37
11:8	reg_ir_rx_stop_tol	R/W	stop symbol postive interval tolerance reg_ir_rx_stop + reg_ir_rx_stop_tol <= 8'FF reg_ir_rx_stop - reg_ir_rx_stop_tol >= 8'00	0x3
15:12	Reserved			
23:16	reg_ir_rx_bit_p	R/W	data symbol postive interval	0x37

Bits	Name	Access	Description	Reset
31:24	reg_ir_rx_bit_p_tol	R/W	data symbol postive interval tolerance reg_ir_rx_bit_p + reg_ir_rx_bit_p_tol <= 8'FF reg_ir_rx_bit_p - reg_ir_rx_bit_p_tol >= 8'00	0x3

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_one	R/W	data one totol interval	0xe0
15:12	Reserved			
23:16	reg_ir_rx_bit_one_tol	R/W	data one total interval tolerance reg_ir_rx_bit_one + reg_ir_rx_bit_one_tol <= 12'FFF reg_ir_rx_bit_one - reg_ir_rx_bit_one_tol >= 12'000	0x18
31:24	Reserved			

Offset Address: 0x040

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_zero	R/W	data zero totol interval	0x6f
15:12	Reserved			
23:16	reg_ir_rx_bit_zero_tol	R/W	data zero total interval tolerance reg_ir_rx_bit_zero + reg_ir_rx_bit_zero_tol <= 12'FFF reg_ir_rx_bit_zero - reg_ir_rx_bit_zero_tol >= 12'000	0x09
31:24	Reserved			

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	reg_ir_rx_release_time	R/W	time for wait repeat code, >108ms.	0x27de
31:16	Reserved			

Offset Address: 0x048

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_p	R/W	repeate code lead symbol postive interval	0x383
15:12	Reserved			
23:16	reg_ir_rx_slead_p_tol	R/W	repeate code lead symbol postive interval tolerance reg_ir_rx_slead_p + reg_ir_rx_slead_p_tol <= 12'FFF reg_ir_rx_slead_p - reg_ir_rx_slead_p_tol >= 12'000	0x48
31:24	Reserved			

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_n	R/W	repeate code lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_slead_n_tol	R/W	repeate code lead symbol negtive interval tolerance reg_ir_rx_slead_n + reg_ir_rx_slead_n_tol <= 12'FFF reg_ir_rx_slead_n - reg_ir_rx_slead_n_tol >= 12'000	0x24
31:24	Reserved			

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_pclock_auto_ctrl	R/W	pclk auto-gating control	0x0
1	reg_lpclock_switch_en	R/W	ip clock could be switch to 32KHz	0x0
7:2	Reserved			
8	reg_skip_lead_p	R/W	skip lead pulse check	0x0
31:9	Reserved			

Offset Address: 0x080

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data0	RO	recevier data[31:0]	

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data1	RO	recevier data[63:32]	

Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data2	RO	recevier data[95:64]	

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data3	RO	recevier data[127:96]	

Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data4	RO	recevier data[159:128]	

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_nec_32bit	RO	receiver data, nec format	

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_sony_12bit	RO	receiver data, sony D7C5 format	
15:12	Reserved			
30:16	reg_ir_rx_sony_15bit	RO	receiver data, sony D7C8 format	
31	Reserved			

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_sony_20bit	RO	receiver data, sony D7C13 format	
31:20	Reserved			

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_philips_rc5	RO	receiver data, RC5 format	
31:12	Reserved			

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_philips_rc6	RO	receiver data, RC6 format	
31:20	Reserved			

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_start_p0_min	RO	start phase0 minimul width	
15:12	Reserved			
27:16	reg_start_p0_max	RO	start phase0 maximul width	
31:28	Reserved			

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_start_p1_min	RO	start phase1 minimul width	
15:12	Reserved			
27:16	reg_start_p1_max	RO	start phase1 maximul width	
31:28	Reserved			

Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_bit_p0_min	RO	bit phase0 minimul width	
15:12	Reserved			
27:16	reg_bit_p0_max	RO	bit phase0 maximul width	
31:28	Reserved			

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_bit_p1_min	RO	bit phase1 minimul width	
15:12	Reserved			

Bits	Name	Access	Description	Reset
27:16	reg_bit_p1_max	RO	bit phase1 maximul width	
31:28	Reserved			

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
11:0	reg_end_min	RO	end phase minimuml width	
15:12	Reserved			
27:16	reg_end_max	RO	end phase maximuml width	
31:28	Reserved			

Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
15:0	reg_frame_min	RO	frame minimul width	
31:16	reg_frame_max	RO	frame maximul width	

19 安全子系统模块

芯片提供独立安全子系统模块，负责提供特定安全功能。

安全子系统模块包括了下列安全功能模块
- 密码运算单元(Crypto DMA)
- 真随机产生单元 (True Random Number Generator)
- 安全除错保护单元(Secure Debug Protection)

密码运算单元(Crypto DMA)提供对称密钥加解密及哈希(Hash)的硬件加速，安全 eFuse 单元负责提供系统安全设置及安全密钥给安全子系统使用，真乱数产生单元提供具合格乱度的随机数供安全系统使用。

图表 19-1 安全子系模块

19.1 CryptoDMA

19.1.1 概述

CryptoDMA 为实现对称密钥算法、杂凑算法及 BASE64 转换的硬体加速器，支持对称算法: AES 128/192/256, DES/TDES, SM4 及 杂凑算法: SHA-1/SHA256 等密码运算。透过链表(linked-list)的指令串达到资料区块的密钥加解密或杂凑运算功能的直接记忆体存取。
- 对称算法适用于进行数据的硬体加解密加速处理，同时支持多种分组加密及区块串联处理方式，包括 ECB, CBC, CTR
- AES（Advanced Encryption Standard）算法的实现符合 FIPS 197 标准。DES (Data Encryption Standard)/TDES 算法的实现符合 ISO/IEC 18033-3

- 杂凑算法适用于进行于数据完整性查验和数字签名运算加速。SHA1 及 SHA256，符合 FIPS180-2 标准。
- BASE64 运算适用于处理文字资料内，储存二迈位资料，如 MIME 电子邮件或 URL 资料

19.1.2 功能特性

CryptoDMA 模块有如下功能特性：

- 支持对称加解密算法 AES 及分组加密模式 ECB/CBC/CTR。密钥长度支持 128 位、256 位，密钥可由安全作业系统或链表指令配置。
- 支持对称加解密算法 SM4 及分组加密模式 ECB/CBC/CTR。
- 支持对称加解密算法 DES/TDES 及分组加密模式 ECB/CBC/CTR。
- 支持哈希算法 SHA1、SHA256
- 支持 CPU 配置输入 PIO 数据和 DMA 方式读取縺表指令输入数据。
- 支持循环链表结构，支持拼接多个链表数据。
- 提供中断状态查询、中断屏蔽和中断清除功能。

19.1.3 DMA 功能描述

CryptoDMA 提供记忆体直接存取 DMA 功能，应用程式只需对目标资料区块提供链表指令，启动 CryptoDMA DMA 功能，直到收到完成的中断通知，区块加解密或哈希运算即表示结束，并将运算结果输出至目标地址。

19.1.4 对称密钥算法分组加密模式功能描述

对称密钥算法 AES/DES/SM4 皆支持 ECB/CBC/CTR 分组加密模式。

19.1.4.1 ECB 模式

ECB（Electronic CodeBook）模式中，加、解密算法是直接由各个分组的运算直接应用到各个分组数据。这个特点使得明文的加密和密文的解密可以由区块资料任一分组独立进行。

图表 19-2 ECB 模式

19.1.4.2 CBC 模式

CBC（Cipher Block Chaining），将输入明文分组先与输入向量 IV（Intialization Vector）或前一分组密文结果进行异或运算，才进行加密运算，CBC 模式下的加密操作是必须从第一笔区块资料分组开始进行，之后加密运算皆需要前一分组所得密文来进行加密。解密时可由当前密文解密后及前一组密文异或运算后得到明文。

图表 19-3 CBC 模式

19.1.4.3 CTR 模式

CTR（Counter）是利用加密或解密一组不同的数列来保证加密数据处理的独立性及安全性，一般是采用加密累加数列后和明文进行异或运算。

图表 19-4 CTR 模式

19.1.5 CryptoDMA 寄存器概览

Name	Address Offset	Description
dma_ctrl	0x000	DMA controll register
int_mask	0x004	interrupt mask
des_base_0	0x008	descriptor base low address
des_base_1	0x00c	descriptor base high address
spacc_int_raw	0x010	interrupt
secure_key_valid	0x014	key valid
des_addr_0	0x018	current descirptor low address
des_addr_1	0x01c	current descirptor high address
PIO_cmd_data_0	0x080	PIO command0
PIO_cmd_data_1	0x084	PIO command1
PIO_cmd_data_2	0x088	PIO command2
PIO_cmd_data_3	0x08c	PIO command3
PIO_cmd_data_4	0x090	PIO command4
PIO_cmd_data_5	0x094	PIO command5
PIO_cmd_data_6	0x098	PIO command6
PIO_cmd_data_7	0x09c	PIO command7

Name	Address Offset	Description
PIO_cmd_data_8	0x0a0	PIO command8
PIO_cmd_data_9	0x0a4	PIO command9
PIO_cmd_data_10	0x0a8	PIO command10
PIO_cmd_data_11	0x0ac	PIO command11
PIO_cmd_data_12	0x0b0	PIO command12
PIO_cmd_data_13	0x0b4	PIO command13
PIO_cmd_data_14	0x0b8	PIO command14
PIO_cmd_data_15	0x0bc	PIO command15
PIO_cmd_data_16	0x0c0	PIO command16
PIO_cmd_data_17	0x0c4	PIO command17
PIO_cmd_data_18	0x0c8	PIO command18
PIO_cmd_data_19	0x0cc	PIO command19
PIO_cmd_data_20	0x0d0	PIO command20
PIO_cmd_data_21	0x0d4	PIO command21
key_data_0	0x100	cipher key data 0
key_data_1	0x104	cipher key data 1
key_data_2	0x108	cipher key data 2
key_data_3	0x10c	cipher key data 3
key_data_4	0x110	cipher key data 4
key_data_5	0x114	cipher key data 5
key_data_6	0x118	cipher key data 6
key_data_7	0x11c	cipher key data 7
key_data_8	0x120	cipher key data 8
key_data_9	0x124	cipher key data 9
key_data_10	0x128	cipher key data 10
key_data_11	0x12c	cipher key data 11
key_data_12	0x130	cipher key data 12
key_data_13	0x134	cipher key data 13
key_data_14	0x138	cipher key data 14
key_data_15	0x13c	cipher key data 15
key_data_16	0x140	cipher key data 16
key_data_17	0x144	cipher key data 17
key_data_18	0x148	cipher key data 18
key_data_19	0x14c	cipher key data 19
key_data_20	0x150	cipher key data 20
key_data_21	0x154	cipher key data 21
key_data_22	0x158	cipher key data 22
key_data_23	0x15c	cipher key data 23
ini_data_0	0x180	initial vector data 0
ini_data_1	0x184	initial vector data 1
ini_data_2	0x188	initial vector data 2
ini_data_3	0x18c	initial vector data 3
ini_data_4	0x190	initial vector data 4
ini_data_5	0x194	initial vector data 5
ini_data_6	0x198	initial vector data 6
ini_data_7	0x19c	initial vector data 7
ini_data_8	0x1a0	initial vector data 8
ini_data_9	0x1a4	initial vector data 9
ini_data_10	0x1a8	initial vector data 10
ini_data_11	0x1ac	initial vector data 11
sha_data_0	0x1c0	SHA paramenter0
sha_data_1	0x1c4	SHA paramenter1
sha_data_2	0x1c8	SHA paramenter2
sha_data_3	0x1cc	SHA paramenter3
sha_data_4	0x1d0	SHA paramenter4

Name	Address Offset	Description
sha_data_5	0x1d4	SHA paramenter5
sha_data_6	0x1d8	SHA paramenter6
sha_data_7	0x1dc	SHA paramenter7

19.1.6 CryptoDMA 寄存器描述

(基址 0x02060000)

dma_ctrl

dma_ctrl
Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	dma_en	R/W	DMA 通道使能控制 0：通道禁能； 1：通道使能。	0x0
1	decriptor_mode	R/W	通道指令模式 0: PIO 模式 1:描述键表模式	0x0
15:2	Reserved			
23:16	max_read_burst	R/W	最大读取 burst 值	0x0
31:24	max_write_burs	R/W	最大写入 burst 值	0x0

int_mask

int_mask
Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	int_enc_mask	R/W	加解密中断屏蔽	0x0
1	int_hash_mask	R/W	Hash 中断屏蔽	0x0
31:2	Reserved		保留	

des_base_0

des_base_0
Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	des_base_0	R/W	描述键表位址_低位址	0x0

des_base_1

des_base_1
Offset Address: 0x00c

Bits	Name	Access	Description	Reset
31:0	des_base_1	R/W	描述键表位址_高位址	0x0

spacc_int_raw

spacc_int_raw
Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	spacc_int_raw	R/W	DMA 中断	0x0

secure_key_valid

secure_key_valid
Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	secure_key_valid	R/W	密钥有效值 **One-Way:** Enabled	0x0

des_addr_0

des_addr_0
Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	des_addr_0	R/W	描述键表位址偏移_低位址	0x0

des_addr_1

des_addr_1
Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	des_addr_1	R/W	描述键表位址偏移_高位址	0x0

PIO_cmd_data_0

PIO mode descriptor
Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_0	R/W	PIO 模式描述键表, 0：模式描述键表无效 1：模式描述键表有效	0x0
1	PIO_cmd_data_0_1	R/W	描述键表最后描述子	0x0
2	PIO_cmd_data_0_2	R/W	PIO 模式中断致能 0：中断禁能； 1：中断使能。	0x0
3	PIO_cmd_data_0_3	R/W	接续描述键表地址选择 0:使用当前地址的接续地址 1:使用接续地址寄存器值	0x0
7:4	PIO_cmd_data_0_4	R/W	保留	0x0
8	PIO_cmd_data_0_8	R/W	DMA 使用旁通	0x0
9	PIO_cmd_data_0_9	R/W	使用 AES 加解密	0x0
10	PIO_cmd_data_0_10	R/W	使用 DES 加解密	0x0
11	PIO_cmd_data_0_11	R/W	使用 SM4 加解密	0x0
12	PIO_cmd_data_0_12	R/W	使用 SHA 运算	0x0
13	PIO_cmd_data_0_13	R/W	使用 BASE64 加解码	0x0
15:14	Reserved			
19:16	PIO_cmd_data_0_16	R/W	运算密钥选择 使用 aes/des/sm4 运算: 1:描述子密钥 2:密钥 2 4:密钥 1 8:密朗 0 使用 sha 1:描述子参数 2:保留 4:保留 8:使用 SHA 参数寄存器 其馀保留	0x0
23:20	PIO_cmd_data_0_20	R/W	首次运算 IV 选择	0x0

Bits	Name	Access	Description	Reset
			使用 aes/des/sm4 运算: 1:描述子 IV 2:IV2 4:IV1 8:IV0	
26:24	PIO_cmd_data_0_24	R/W	接续运算 IV 选择 使用 aes/des/sm4 运算: 1:描述子 IV 2:IV2 4:IV1 8:IV0	0x0
27	PIO_cmd_data_0_27	R/W	密钥选择致能	0x0
31:28	PIO_cmd_data_0_28	R/W	保留	0x0

PIO_cmd_data_1

PIO mode descriptor
Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_1_0	R/W	加解密选择 1-encryption/0-decription hash 参数选择 1-need parameter/0-no need	0x0
1	PIO_cmd_data_1_1	R/W	CBC 模式 0-ECB/1-CBC	0x0
2	PIO_cmd_data_1_2	R/W	CTR 模式 1-CTR	0x0
5:3	PIO_cmd_data_1_3	R/W	密钥模式 100-128bit/010-192bit/001-256bit for aes 0-DES/1-TDES sha mode 0-SHA1/1-SHA256	0x0
31:6	PIO_cmd_data_1_6	R/W	保留	0x0

PIO_cmd_data_2

PIO mode descriptor
Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_2	R/W	保留	0x0

PIO_cmd_data_3

PIO mode descriptor
Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_3	R/W	接续描述子位址_低位址	0x0

PIO_cmd_data_4

PIO mode descriptor
Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_4	R/W	接续描述子位址_高位址	0x0

PIO_cmd_data_5

PIO mode descriptor
Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_5	R/W	DMA 来源地址_低位址	0x0

PIO_cmd_data_6

PIO mode descriptor

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_6	R/W	DMA 来源地址_高位址	0x0

PIO_cmd_data_7

PIO mode descriptor
Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_7	R/W	DMA 目标地址_低位址	0x0

PIO_cmd_data_8

PIO mode descriptor
Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_8	R/W	DMA 目标地址_高位址	0x0

PIO_cmd_data_9

PIO mode descriptor
Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_9	R/W	SHA 信息大小	0x0

PIO_cmd_data_10

PIO mode descriptor
Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_10	R/W	保留	0x0

PIO_cmd_data_11

PIO mode descriptor
Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_11	R/W	BASE64 目标信息大小	0x0

PIO_cmd_data_12

PIO mode descriptor
Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_12	R/W	保留	0x0

PIO_cmd_data_13

PIO mode descriptor
Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_13	R/W	保留	0x0

PIO_cmd_data_14

PIO mode descriptor
Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_14	R/W	保留	0x0

PIO_cmd_data_15

PIO mode descriptor

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_15	R/W	保留	0x0

PIO_cmd_data_16

PIO mode descriptor
Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_16	R/W	保留	0x0

PIO_cmd_data_17

PIO mode descriptor
Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_17	R/W	保留	0x0

PIO_cmd_data_18

PIO mode descriptor
Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_18	R/W	保留	0x0

PIO_cmd_data_19

PIO mode descriptor
Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_19	R/W	保留	0x0

PIO_cmd_data_20

PIO mode descriptor
Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_20	R/W	保留	0x0

PIO_cmd_data_21

PIO mode descriptor
Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_21	R/W	保留	0x0

key_data_0

3key
Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	key_data_0	RO	密钥	

key_data_1

key
Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	key_data_1	RO	密钥	

key_data_2

key
Offset Address: 0x108

Bits	Name	Access	Description	Reset
31:0	key_data_2	RO	密钥	

key_data_3

key
Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	key_data_3	RO	密钥	

key_data_4

key
Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	key_data_4	RO	密钥	

key_data_5

key
Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	key_data_5	RO	密钥	

key_data_6

key
Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	key_data_6	RO	密钥	

key_data_7

key
Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	key_data_7	RO	密钥	

key_data_8

key
Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	key_data_8	RO	密钥	

key_data_9

key
Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	key_data_9	RO	密钥	

key_data_10

key
Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	key_data_10	RO	密钥	

key_data_11

key
Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	key_data_11	RO	密钥	

key_data_12

key
Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	key_data_12	RO	密钥	

key_data_13

key
Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	key_data_13	RO	密钥	

key_data_14

key
Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	key_data_14	RO	密钥	

key_data_15

key
Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	key_data_15	RO	密钥	

key_data_16

key
Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	key_data_16	RO	密钥	

key_data_17

key
Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	key_data_17	RO	密钥	

key_data_18

key
Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	key_data_18	RO	密钥	

key_data_19

key
Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	key_data_19	RO	密钥	

key_data_20

key
Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	key_data_20	RO	密钥	

key_data_21

key
Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	key_data_21	RO	密钥	

key_data_22

key
Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	key_data_22	RO	密钥	

key_data_23

key
Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	key_data_23	RO	密钥	

ini_data_0

3iv
Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	ini_data_0	RO	初始参数	

ini_data_1

iv
Offset Address: 0x184

Bits	Name	Access	Description	Reset
31:0	ini_data_1	RO	初始参数	

ini_data_2

iv
Offset Address: 0x188

Bits	Name	Access	Description	Reset
31:0	ini_data_2	RO	初始参数	

ini_data_3

iv
Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	ini_data_3	RO	初始参数	

ini_data_4

iv

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	ini_data_4	RO	初始参数	

ini_data_5

iv

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	ini_data_5	RO	初始参数	

ini_data_6

iv

Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	ini_data_6	RO	初始参数	

ini_data_7

iv

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	ini_data_7	RO	初始参数	

ini_data_8

iv

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
1:0	ini_data_8	RO	初始参数	
31:2	Reserved			

ini_data_9

iv

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
1:0	ini_data_9	RO	初始参数	
31:2	Reserved			

ini_data_10

iv

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	ini_data_10	RO	初始参数	

ini_data_11

iv

Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
31:0	ini_data_11	RO	初始参数	

sha_data_0

sha parameter
Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	sha_data_0	RO	SHA 参数	

sha_data_1

sha parameter
Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	sha_data_1	RO	SHA 参数	

sha_data_2

sha parameter
Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	sha_data_2	RO	SHA 参数	

sha_data_3

sha parameter
Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	sha_data_3	RO	SHA 参数	

sha_data_4

sha parameter
Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	sha_data_4	RO	SHA 参数	

sha_data_5

sha parameter
Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	sha_data_5	RO	SHA 参数	

sha_data_6

sha parameter
Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	sha_data_6	RO	SHA 参数	

sha_data_7

sha parameter
Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	sha_data_7	RO	SHA 参数	

19.2 安全除错防火墙

为了提供除错或测试时读取或控制晶片内部相关功能，晶片提供了数种除错介面，如 JTAG、I2C 等不同的外部介面。若缺乏适当的保护机制，这些介面很轻易地会被利用以直接或间接的方式，来攻击晶片安全机制或读取内部需保密的资讯。为了保护及控制这些介面，晶片采用了安全除错防火墙措施。

19.2.1 概述

芯片支持三种主要的除错介面，

1. RISCV JTAG: RISCV 处理器内建调试接口，予许使用者透过 JTAG 接口访问 ARM 内部寄存器
2. I2C: 晶片提供除错介面
3. Test interface: 晶片提供生产测试专用调试接口

对于 JTAG/I2C 接口，安全除错防火墙针对其访问提供特定保护控制

对 Test interface 接口，安全除错介面则其访问提供一类独立的保护控制 (Test 访问)

针对这些除错类别，安全除错防火墙提供三种连接控制状态

- 开放(Open): 予许外部透过接品连入，不施加额外控制
- 保护(Protected): 不予许外部透过接品连入，直到由外部透过 I2C 接口输入对应密码解开保护
- 关闭(Closed): 不予许外部透过该接口连入，无法藉由其他方法再次开启。

19.2.2 状态查讯及密码输入介面(I2C)

安全除错防火墙提供一个独立运作的 I2C 介面，供晶片外部透过 I2C 来查讯除错介面当前状态及输入相对应密码来将在保护状态的介面重新开启。外部需指定正确的 I2C ID 来连接接防火墙接口。

I2C 介面寄存器位址如下

表格 19-1 状态查询 I2C 介面寄存器地址

byte address	bitwidth	signal name	description
0	32	i2c_REE_password [31:0]	128-位元除错介面密码
4	32	i2c_REE_password [63:32]	128-位元除错介面密码
8	32	i2c_REE_password [95:64]	128-位元除错介面密码
C	32	i2c_REE_password [127:96]	128-位元除错介面密码
10	32	Reserved	保留

byte address	bitwidth	signal name	description
14	32	Reserved	保留
18	32	Reserved	保留
1C	32	Reserved	保留
20	32	Reserved	保留
24	32	Reserved	保留
28	32	Reserved	保留
2C	32	Reserved	保留
30	32	i2c_TST_password [31:0]	128-bit password entry field for test ports
34	32	i2c_TST_password [63:32]	128-位元测试除错介面密码
38	32	i2c_TST_password [95:64]	128-位元测试除错介面密码
3C	32	i2c_TST_password [127:96]	128-位元测试除错介面密码
40	1	REE_PW_update	更新密码比较结果
44	1	Reserved	保留
48	1	Reserved	保留
4C	1	TST_PW_update	更新测试介面密码比较结果
80	32	Chip_UID0 (LSB of ID)	晶片识别序号(Device ID)
84	32	Chip_UID1 (MSB of ID)	晶片识别序号(Device ID)
88	32	MKTSEG	市场区隔编号
8C	32	DBG_MODE	晶片除错介面保护设置状态 [1:0] : 介面保护模式，0:open;1:protected:2/3:closed [3:2] : 保留 [5:4] : 保留 [7:6] : 测试介面保护模式，0:open;1:protected:2/3:closed [8]: 使用 HASH 比较
90	32	reserved	保留
94	4	DBG_PROT_STATUS	当前除错保护状态 [0]: 介面保护状态，0:open 1:closed [1]: 保留 [2]: 保留 [3]::测试介面保护状态，0:open 1:closed

19.2.3 状态查讯及密码输入流程

19.2.3.1 状态查讯流程

- (步骤一) 控制外部 I2C 发出开始讯号
- (步骤二) 由 I2C 发出防火墙 I2C ID (预设 0x56)
- (步骤三) 由 I2C 读取位址 0x04001A94，以获得当前除错介面保护状态

19.2.3.2 密码输入流程

- (步骤一) 控制外部 I2C 发出开始讯号
- (步骤二) 由 I2C 发出除错介面防火墙的 I2C ID (预设 0x56)
- (步骤三) 由 I2C 读取位址 0x04001A80/0x04001A84，以获得设备序号 Device ID，读

取 0x04001A88 以获得市场区别编号

- (步骤四) 经由设备序号及市场区别编号，准备各个类别相对应的解锁密码
- (步骤五) 由 I2C 读取位址 0x94，以获得当前除错介面保护状态，确认是否上锁
- (步骤六) 以非安全除错介面为例，由 I2C 将非安全密码写入位址 0x04001A00/0x04001A04/0x04001A08/0x04001A0C
- (步骤七) I2C 写任意值至位址 0x04001A10，更新密码比较值
- (步骤五) 由 I2C 读取位址 0x94，以获得当前除错介面保护状态，确认是否解锁

19.3 Efuse 控制器

晶片内部集成 4Kbit eFuse 空间，并以 Efuse Ctrl 对 Efuse 进行编程和读取。

Efuse Ctrl 的主要功能包括：

- 提供双 eFuse 位元 (Double bit) 保护机制，由两个实体 eFuse 位元组成单一位元逻辑有效值，等效于提供 2Kbit 暂存器空间，以提高 eFuse 烧写或资料维持的强健性
- 上电复位后（Power-On-Reset），自动加载 efuse 的内容到寄存器，提供晶片系统所需组态设定及减少对 Efuse 所需读取次数提高使用期限
- 提供 efuse 编程、读取、验证读取及上下电指令及内容安全保护机制。

Efuse 资料寄存器分成两块区域，一块为非安全区域，另一块是安全区域，非安全区域内资料予许所有模块访问，安全区域只予许安全模块访问。非安全区域储存系统组态及公开讯息，安全区域储存安全组态、密钥及密码。

图表 19-5 eFuse CTRL 模块架构