



雅仪科技——【嵌入式智能互联设备】软硬件方案供应商。
<http://blog.mcuol.com/travellinux>，获取更多资讯。

Hi3520 H.264 编解码处理器

用户指南

文档版本	02
发布日期	2010-01-19
部件编码	N/A

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

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编： 518129

网址： <http://www.hisilicon.com>

客户服务电话： +86-755-28788858

客户服务传真： +86-755-28357515

客户服务邮箱： support@hisilicon.com

版权所有 © 深圳市海思半导体有限公司 2009~2010。保留一切权利。

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

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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



目 录

About This Document	1
1 Product Description	1-1
1.1 System Architecture	1-1
1.1.1 Overview	1-1
1.1.2 Processor System.....	1-2
1.1.3 Graphics Processing	1-2
1.1.4 Video Encoding and Decoding.....	1-3
1.1.5 Encryption/Decryption Engine.....	1-3
1.1.6 Memory Controller Interface.....	1-3
1.1.7 Ethernet Interface	1-4
1.1.8 Video Interface	1-4
1.1.9 Audio Interface.....	1-4
1.1.10 MMC/SD/SDIO Controller	1-5
1.1.11 PCI Interface.....	1-5
1.1.12 USB Interface.....	1-5
1.1.13 Other Peripheral Interface	1-5
1.1.14 Hardware Feature	1-6
1.2 Application Scenario	1-6
2 Hardware	2-1
2.1 Pin Description.....	2-1
2.1.1 Power Pins.....	2-2
2.1.2 SIO Pins	2-21
2.1.3 VDAC Pins.....	2-23
2.1.4 DDRA Pins.....	2-25
2.1.5 DDRB Pins.....	2-30
2.1.6 EBI Pins	2-33
2.1.7 GMAC Pins.....	2-36
2.1.8 VO Pins	2-37
2.1.9 PCI Pins.....	2-39
2.1.10 SYS Pins.....	2-43
2.1.11 I ² C Pins.....	2-44



2.1.12 JTAG Pins	2-44
2.1.13 UART Pins	2-44
2.1.14 USB Pins	2-45
2.1.15 VI Pins	2-46
2.2 Description of the Software Multiplexing Pins	2-48
2.2.1 VI Pin Multiplexing	2-48
2.2.2 VO Pin Multiplexing	2-52
2.2.3 I2C Pin Multiplexing	2-56
2.2.4 SIO Pin Multiplexing	2-56
2.2.5 EBI Pin Multiplexing	2-57
2.2.6 GMAC Pin Multiplexing	2-57
2.2.7 PCI Pin Multiplexing	2-58
2.3 Description of Hardware Multiplexing Pins	2-59
2.3.1 EBI Pin Multiplexing	2-59
2.3.2 PCI Pin Multiplexing	2-62
2.4 Summary of IO Config Registers	2-63
2.5 Description of the IO Config Registers	2-67
2.6 Recommended Sequence of Power-On and Power-Off	2-108
2.7 External Interrupts	2-109
2.8 Electrical Specifications	2-109
2.8.1 DC/AC Parameters	2-109
2.8.2 Recommended Operating Conditions	2-110
2.9 PCB Routing Recommendations	2-111
2.10 Timing Parameters	2-111
2.10.1 Primitives of Timing Diagrams	2-111
2.10.2 Timings of the DDR2 Interface	2-111
2.10.3 Timing of the GMAC Interface	2-114
2.10.4 Timing of the VI Interface	2-119
2.10.5 Timing of the VO Interface	2-119
2.10.6 Timings of the PCI Interface	2-120
2.10.7 Timing of the I ² C Interface	2-121
2.10.8 Timing of the MMC Interface	2-122
2.10.9 Timing of the SSP Interface	2-123
2.10.10 Timing of the UART Interface	2-125
2.10.11 Timings of SIO Interfaces	2-126
2.10.12 Timing of the SMI Interface	2-128
2.11 Package and Pinout	2-128
2.11.1 Package	2-128
2.11.2 Pinout	2-131
2.11.3 Pin Arrangement Table	2-136
3 System	3-1



3.1 Reset.....	3-1
3.1.1 Overview	3-1
3.1.2 Reset Signal Control.....	3-1
3.1.3 Reset Configuration.....	3-2
3.2 Clock	3-3
3.2.1 Overview	3-3
3.2.2 Clock Control	3-3
3.2.3 Clock Configuration	3-5
3.3 Processor and Address Space Mapping of the Memory	3-16
3.3.1 Processor	3-16
3.3.2 Address Space Mapping of the Memory	3-17
3.4 Interrupt System	3-30
3.4.1 Overview	3-30
3.4.2 Features	3-30
3.4.3 Function Description	3-30
3.4.4 Register Summary	3-34
3.4.5 Register Description	3-35
3.5 Direct Memory Access Controller.....	3-40
3.5.1 Overview	3-40
3.5.2 Features	3-40
3.5.3 Function Description	3-41
3.5.4 Operating Mode.....	3-44
3.5.5 Register Summary	3-45
3.5.6 Register Description	3-48
3.6 CIPHER.....	3-67
3.6.1 Overview	3-67
3.6.2 Features	3-67
3.6.3 Function Description	3-68
3.6.4 Operating Mode.....	3-78
3.6.5 Register Summary	3-80
3.6.6 Register Description	3-81
3.7 Timer	3-95
3.7.1 Overview	3-95
3.7.2 Features	3-96
3.7.3 Function Description	3-96
3.7.4 Operating Mode.....	3-97
3.7.5 Register Summary	3-98
3.7.6 Register Description	3-98
3.8 Watchdog.....	3-107
3.8.1 Overview	3-107
3.8.2 Features	3-107
3.8.3 Signal Description	3-108



3.8.4 Function Description	3-108
3.8.5 Operating Mode.....	3-109
3.8.6 Register Summary	3-110
3.8.7 Register Description	3-111
3.9 Real Time Clock.....	3-115
3.9.1 Overview	3-115
3.9.2 Features	3-115
3.9.3 Function Description	3-115
3.9.4 Operating Mode.....	3-115
3.9.5 Register Summary	3-117
3.9.6 Regitsr Description.....	3-117
3.10 System Controller.....	3-121
3.10.1 Overview	3-121
3.10.2 Features	3-121
3.10.3 Function Description	3-121
3.10.4 Register Summary	3-127
3.10.5 Register Description	3-129
3.11 Power Management and Low-Power Mode Control	3-173
3.11.1 Overview	3-173
3.11.2 System Operating Modes	3-174
3.11.3 Clock Gating and Clock Frequency Adjustment	3-174
3.11.4 DDR Low-Power Contorl.....	3-175
4 Memory Controller	4-1
4.1 DDR Controller	4-1
4.1.1 Overview	4-1
4.1.2 Features	4-1
4.1.3 Signal Description	4-1
4.1.4 Function Description	4-4
4.1.5 Operating Mode.....	4-12
4.1.6 Register Summary	4-13
4.1.7 Register Description	4-14
4.2 SMI Controller	4-29
4.2.1 Overview	4-29
4.2.2 Features	4-29
4.2.3 Signal Description	4-30
4.2.4 Function Description	4-31
4.2.5 Operating Mode.....	4-35
4.2.6 Register Summary	4-38
4.2.7 Register Description	4-39
4.3 NAND Flash Controller	4-52
4.3.1 Overview	4-52



4.3.2 Features	4-52
4.3.3 Description of NANDC Interfaces	4-53
4.3.4 Function Description	4-54
4.3.5 Operating Modes	4-56
4.3.6 Register Summary	4-65
4.3.7 Register Description	4-66
5 GMAC	5-1
5.1 Overview	5-1
5.2 Feature	5-1
5.3 Signal Description	5-2
5.4 Function Description	5-3
5.4.1 Typical Application	5-3
5.4.2 Frame Format	5-4
5.4.3 Uplink and Downlink Frame Management	5-4
5.4.4 Flow Control of the GMAC	5-7
5.4.5 Packet Receive Interrupt Management	5-7
5.4.6 Packet Filtering	5-7
5.5 Operating Mode	5-7
5.5.1 Reading and Writing the PHY	5-7
5.5.2 Transmitting and Receiving Packets	5-7
5.6 GMAC Register Summary	5-7
5.7 Register Description	5-7
6 Video Interface	6-1
6.1 VIU	6-1
6.1.1 Overview	6-1
6.1.2 Features	6-1
6.1.3 Signal Description	6-2
6.1.4 Function Description	6-4
6.1.5 Operating Mode	6-20
6.1.6 Register Summary	6-24
6.1.7 Register Description	6-27
6.2 VOU	6-75
6.2.1 Overview	6-75
6.2.2 Features	6-75
6.2.3 Signal Description	6-77
6.2.4 Function Description	6-80
6.2.5 Operating Mode	6-84
6.2.6 Register Summary	6-88
6.2.7 Register Description	6-105
7 Audio Interface	7-1
7.1 Overview	7-1



7.2 Features	7-1
7.2.1 Features of the PCM Interface.....	7-1
7.2.2 Features of the I ² S Interface	7-1
7.3 Signal Description	7-2
7.4 Function Description	7-3
7.5 Operating Mode.....	7-8
7.6 Register Summary	7-11
7.7 Register Description	7-13
8 MMC/SD/SDIO Controller	8-1
8.1 Overview	8-1
8.2 Features	8-1
8.3 Signal Description	8-2
8.4 Function Description	8-3
8.5 Operating Mode.....	8-8
8.5.1 Pin Multiplexing.....	8-8
8.5.2 Clock Gating	8-8
8.5.3 Soft Reset	8-8
8.6 Register Summary	8-18
8.7 Register Description	8-19
9 PCI.....	9-1
9.1 Overview	9-1
9.2 Features	9-1
9.3 Signal Description	9-2
9.4 Function Description	9-4
9.5 Operating Mode.....	9-6
9.5.1 Pin Multiplexing.....	9-6
9.5.2 Clock Gating	9-7
9.5.3 Clock Configuration	9-7
9.5.4 Soft Reset	9-8
9.5.5 Configuring the Operating Mode	9-8
9.5.6 Transmitting Data Through the Window	9-8
9.5.7 Transmitting Data Through the DMA Channel	9-9
9.6 Register Summary	9-11
9.6.1 Registers at the AHB Side	9-11
9.6.2 Registers for the Header Region of the PCI Configuration Space	9-12
9.7 Register Description	9-14
9.7.1 Registers at the AHB Side	9-14
9.7.2 Registers in the PCI Configuration Space	9-29
10 USB 2.0 Host.....	10-1
10.1 Overview	10-1
10.2 Features	10-1



10.3 Function Description	10-1
10.4 Operating Mode.....	10-3
10.4.1 Interface Signals.....	10-3
10.4.2 Typical Application	10-4
10.4.3 Clock Gating	10-4
10.4.4 Soft Reset	10-5
10.5 Register Summary	10-5
10.6 Register Description.....	10-5
11 Other Peripheral Interfaces.....	11-1
11.1 I ² C Interface.....	11-1
11.1.1 Overview	11-1
11.1.2 Features	11-1
11.1.3 Signal Description	11-1
11.1.4 Function Description	11-2
11.1.5 Operating Mode.....	11-4
11.1.6 Register Summary	11-8
11.1.7 Register Descriptions.....	11-9
11.2 UART	11-31
11.2.1 Overview	11-31
11.2.2 Features	11-31
11.2.3 Signal Description	11-31
11.2.4 Function Description	11-33
11.2.5 Operating Mode.....	11-35
11.2.6 Register Summary	11-38
11.2.7 Register Description	11-39
11.3 SPI	11-52
11.3.1 Overview	11-52
11.3.2 Features	11-52
11.3.3 Signal Description	11-53
11.3.4 Function Description	11-53
11.3.5 Operating Mode.....	11-62
11.3.6 Register Summary	11-65
11.3.7 Register Description	11-65
11.4 IR.....	11-73
11.4.1 Overview	11-73
11.4.2 Features	11-73
11.4.3 Signal Description	11-73
11.4.4 Function Description	11-73
11.4.5 Operating Mode.....	11-82
11.4.6 Register Summary	11-85
11.4.7 Register Description	11-85



11.5 GPIO.....	11-98
11.5.1 Overview	11-98
11.5.2 Features	11-98
11.5.3 Signal Description	11-98
11.5.4 Function Description	11-103
11.5.5 Operating Mode.....	11-103
11.5.6 Register Summary	11-107
11.5.7 Register Description	11-108
12 Test Interface.....	12-1
12.1 Overview	12-1
12.2 Operating Modes	12-1
12.3 JTAG Debugging.....	12-1
12.3.1 JTAG Interface Signals.....	12-1
12.3.2 Debugging Mode.....	12-2
13 Video Processing Module.....	13-1
13.1 Video Codec	13-1
13.1.1 Overview	13-1
13.1.2 Features	13-1
13.2 TDE	13-2
13.2.1 Overview	13-2
13.2.2 Features	13-2
A Acronyms and Abbreviations.....	A-1



插图目录

Figure 1-1 Logic block diagram of the Hi3520	1-1
Figure 1-2 Block diagram of the Hi3520 in a 16-channel CIF DVR.....	1-7
Figure 1-3 Block diagram of the Hi3520 in a 4-channel D1 DVR	1-8
Figure 1-4 Block diagram of the Hi3520 in an 8-channel D1 DVR	1-9
Figure 2-1 Primitives of timing diagrams.....	2-111
Figure 2-2 Write timing of dqs_out relative to dq_out, CKP, and CKN	2-112
Figure 2-3 Write timing of dqs_out relative to CK.....	2-112
Figure 2-4 Write timing of cmd/addr relative to CK	2-112
Figure 2-5 DDR2 SDRAM output timing	2-113
Figure 2-6 100 Mbit/s receive timing of the MII interface	2-114
Figure 2-7 10 Mbit/s receive timing of the MII interface	2-115
Figure 2-8 100 Mbit/s transmit timing of the MII interface	2-115
Figure 2-9 10 Mbit/s transmit timing of the MII interface	2-115
Figure 2-10 Receive timing of the RGMII interface	2-116
Figure 2-11 Transmit timing of the RGMII interface	2-116
Figure 2-12 Read timing of the MDIO interface	2-117
Figure 2-13 Write timing of the MDIO interface.....	2-117
Figure 2-14 Timing parameter diagram of the MDIO interface	2-118
Figure 2-15 Transmit timing of the MDIO interface	2-118
Figure 2-16 Timing of the VI interface.....	2-119
Figure 2-17 Timing of the VO interface	2-119
Figure 2-18 Timing of the VO concatenated input interface	2-120
Figure 2-19 Timing of the PCI interface (using the inside clock of the Hi3520)	2-120
Figure 2-20 Timing of the PCI interface (using the external clock of the Hi3520)	2-121
Figure 2-21 Transfer timing of the I ² C interface	2-121
Figure 2-22 Timing of the MMC Interface.....	2-122



Figure 2-23 SSP_SPICK clock timing of the SSP interface	2-123
Figure 2-24 Timing of the SSP interface in master mode (sph = 0).....	2-123
Figure 2-25 Timing of the SSP interface in master mode (sph = 1).....	2-124
Figure 2-26 Timing of the UART interface	2-126
Figure 2-27 Receive timing of the I ² S interface	2-127
Figure 2-28 Transmit timing of the I ² S interface	2-127
Figure 2-29 Receive timing of the PCM interface.....	2-127
Figure 2-30 Transmit timing of the PCM interface	2-128
Figure 2-31 Package dimensions (top view).....	2-129
Figure 2-32 Package dimensions (bottom view)	2-129
Figure 2-33 Detail B	2-130
Figure 2-34 Package dimensions (side view)	2-130
Figure 2-35 Detail A.....	2-130
Figure 2-36 Schematic colors in each pinout diagram.....	2-132
Figure 2-37 Hi3520 pinout (rows A–T, columns 1–16).....	2-133
Figure 2-38 Hi3520 pinout (rows A–T, columns 17–32).....	2-134
Figure 2-39 Hi3520 pinout (rows U–AM, columns 1–16)	2-135
Figure 2-40 Hi3520 pinout (rows U–AM, columns 17–32)	2-136
Figure 3-1 Reset signal control diagram.....	3-1
Figure 3-2 Functional block diagram of the clock management module.....	3-4
Figure 3-3 Address space mapping during the booting from the NOR flash.....	3-18
Figure 3-4 Address distribution after the remapping is cleared during the booting from the NOR flash.	3-20
Figure 3-5 Address space mapping during the booting from the NAND flash.....	3-21
Figure 3-6 Address distribution after the remapping is cleared during the booting from the NAND flash.	3-22
Figure 3-7 Address space mapping during the booting from the DDR.....	3-24
Figure 3-8 Address distribution after the remapping is cleared during the booting from the DDR.....	3-25
Figure 3-9 Functional block diagram of the INT	3-31
Figure 3-10 Functional block diagram of the DMAC	3-41
Figure 3-11 Diagram of updating channel registers through LLI	3-42
Figure 3-12 Structure of the DMAC LLIs	3-60
Figure 3-13 3DES encryption of the 3-key operation and 2-key operation	3-68
Figure 3-14 3DES decryption of the 3-key operation and 2-key operation	3-69
Figure 3-15 ECB mode of the AES and DES algorithms	3-69



Figure 3-16 ECB mode of the 3DES algorithm.....	3-70
Figure 3-17 CBC mode of the AES and DES algorithms	3-71
Figure 3-18 CBC mode of the 3DES algorithm.....	3-72
Figure 3-19 S-bit CFB mode of the AES and DES algorithms.....	3-73
Figure 3-20 S-bit CFB mode of the 3DES algorithm	3-74
Figure 3-21 OFB mode of the AES algorithm	3-75
Figure 3-22 S-bit OFB mode of the DES algorithm	3-76
Figure 3-23 S-bit OFB mode of the 3DES algorithm	3-77
Figure 3-24 CTR mode of the AES algorithm	3-78
Figure 3-25 Application block diagram of the watchdog	3-108
Figure 3-26 Process of switching the system mode.....	3-124
Figure 3-27 Bit allocation of chip ID registers	3-127
Figure 4-1 Diagram of connecting the DDRC to one DDR2 SDRAM.....	4-5
Figure 4-2 Diagram of connecting the DDRC to two DDR2 SDRAMs.....	4-6
Figure 4-3 Connection between the SMI controller and the asynchronous static memory.....	4-32
Figure 4-4 Connection between the SMI controller and the control devices integrated with asynchronous static memory interfaces	4-32
Figure 4-5 Timing diagram (read/write) of the SMI controller in timing parameter mode	4-34
Figure 4-6 Timing diagram (page read) of the SMI controller in timing parameter mode	4-34
Figure 4-7 Timing diagram (wait read/write) of the SMI controller in asynchronous wait mode	4-35
Figure 4-8 Block diagram of NANDC interfaces	4-54
Figure 4-9 Typical timing when the NANDC reads the data of a page size from the NAND flash	4-55
Figure 4-10 Timing when the NANDC starts to be programmed.....	4-56
Figure 4-11 Data storage structure of the NAND flash with the page size of (512 + 16) bytes in 1-bit ECC mode.....	4-60
Figure 4-12 Data storage structure of the NAND flash with the page size of 2 KB (2048 + 64) bytes.....	4-61
Figure 4-13 Data structure after data is automatically stored in main areas and their corresponding spare areas	4-61
Figure 4-14 Data storage structure of the NAND flash with the page size of 2 KB (2048 + 64) bytes in 4-bit ECC mode	4-62
Figure 4-15 Data storage structure of the NAND flash with the page size of (2048 + 26 x 4) bytes in 8-bit ECC mode.....	4-62
Figure 5-1 Typical application	5-4
Figure 5-2 Ethernet II frame format	5-4
Figure 5-3 General data flow.....	5-5



Figure 5-4 Data storage format in the downlink configuration FIFO	5-6
Figure 5-5 Data storage format in the uplink configuration FIFO	5-6
Figure 5-6 Data storage format in the downlink SDRAM	5-7
Figure 5-7 Data storage format in the uplink SDRAM	5-7
Figure 6-1 Functional block diagram of the VIU	6-1
Figure 6-2 Typical application of the VIU	6-4
Figure 6-3 Vertical timing of the 525-line 60 fields/s video system	6-7
Figure 6-4 Vertical timing of the 625-line 50 fields/s video system	6-7
Figure 6-5 2-Path BT.656 TDM timing	6-8
Figure 6-6 4-Path BT.656 TDM timing	6-8
Figure 6-7 Horizontal input timing of the HD interface	6-9
Figure 6-8 Vertical input timing of the HD interface	6-9
Figure 6-9 ITU-R BT.601 horizontal timing	6-10
Figure 6-10 NTSC vertical sync timings	6-10
Figure 6-11 PAL vertical sync timings	6-11
Figure 6-12 DC horizontal timing	6-12
Figure 6-13 DC vertical pulse timing	6-12
Figure 6-14 DC vertical line valid timing	6-12
Figure 6-15 Horizontal sync timing of the 16-bit sync parallel interface	6-13
Figure 6-16 Vertical sync timing of the 16-bit sync parallel interface	6-13
Figure 6-17 Relationships between the active video area and the horizontal/vertical blanking areas	6-14
Figure 6-18 YCbCr 4:2:2 co-sited sampling format	6-14
Figure 6-19 YCbCr 4:2:2 interspersed sampling format	6-15
Figure 6-20 YCbCr4:2:2 storage mode	6-16
Figure 6-21 Big endian and little endian storage modes	6-16
Figure 6-22 Package storage mode	6-17
Figure 6-23 8-bit raw data storage mode	6-17
Figure 6-24 Position of the data of blanking areas	6-18
Figure 6-25 Mappings between the external ports and internal channels of the VIU	6-19
Figure 6-26 Clock configuration of the VIU	6-20
Figure 6-27 Workflow of the VIU	6-22
Figure 6-28 Software configuration process	6-23
Figure 6-29 Process of reading the luminance statistical value through software	6-24



Figure 6-30	Parameter diagram of the captured image.....	6-36
Figure 6-31	Sequence of storing blanking area data in registers	6-64
Figure 6-32	Multiplexing relationship between VOU interfaces.....	6-80
Figure 6-33	Processing the pixel alpha data of a graphics layer.....	6-82
Figure 6-34	Three sets of coordinates of a video layer.....	6-83
Figure 6-35	Process of configuring surface registers (recommended)	6-85
Figure 6-36	Frame update mode of surface registers.....	6-86
Figure 6-37	Field update mode of surface registers.....	6-86
Figure 6-38	Updating the on-chip coefficients	6-88
Figure 7-1	Connection diagram of the I ² S interface in master mode.....	7-4
Figure 7-2	Connections diagram of the I ² S interface in the slave mode.....	7-4
Figure 7-3	Connections of the I ² S/PCM interface used for audio recording in master mode.....	7-5
Figure 7-4	Connections of the I ² S/PCM interface used for audio recording in slave mode	7-6
Figure 7-5	Timing diagram of the I2S interface	7-6
Figure 7-6	Timing diagram of the PCM interface in standard mode	7-7
Figure 7-7	Timing diagram of the PCM interface in customized mode	7-7
Figure 7-8	Receiving 2-/4-/8-/16-channel data through the I ² S interface.....	7-7
Figure 7-9	Receiving 2-/4-/8-/16-channel data through the PCM interface	7-8
Figure 8-1	Typical application circuit of the MMC.....	8-3
Figure 8-2	Functional block diagram of the MMC.....	8-4
Figure 8-3	Non-data transfer commands of the MMC	8-5
Figure 8-4	Command format of the MMC	8-5
Figure 8-5	Response format of the card device	8-5
Figure 8-6	Single-block and multiple-block read operations.....	8-6
Figure 8-7	Single-block and multiple-block write operations	8-7
Figure 8-8	Block data format in 1-bit transfer mode	8-7
Figure 8-9	Block data format in 4-bit transfer mode	8-8
Figure 9-1	Typical application of the PCI bus	9-4
Figure 9-2	Architecture of the Hi3520 PCI module.....	9-5
Figure 9-3	Timing diagram of the memory (I/O) read operation.....	9-5
Figure 9-4	Timing diagram of the memory (I/O) write operation	9-6
Figure 9-5	Timing diagram of configuration access of the PCI bus	9-6
Figure 10-1	Logic block diagram of the USB module	10-2



Figure 10-2 Reference design of the USB 2.0 host	10-4
Figure 11-1 Circuit diagram of the typical I ² C application.....	11-2
Figure 11-2 Timing for I ² C data transfer	11-2
Figure 11-3 Frame format for I ² C data transfer	11-3
Figure 11-4 Typical application block diagram 1 of the UART.....	11-33
Figure 11-5 Typical application block diagram 2 of the UART.....	11-34
Figure 11-6 Block diagram of the RTS signal output mode	11-34
Figure 11-7 Frame format of the UART	11-35
Figure 11-8 Application block diagram when the SPI is connected to a single slave device.....	11-54
Figure 11-9 Application block diagram when the SPI is connected to two slave devices	11-54
Figure 11-10 Application block diagram of the SPI acting as a slave device	11-55
Figure 11-11 Motorola SPI single frame format (spo = 0, sph = 0).....	11-55
Figure 11-12 Motorola SPI continuous frame format (spo = 0, sph = 0).....	11-56
Figure 11-13 Motorola SPI single frame format (spo = 0, sph = 1).....	11-56
Figure 11-14 Motorola SPI continuous frame format (spo = 0, sph = 1).....	11-57
Figure 11-15 Motorola SPI single frame format (spo = 1, sph = 0).....	11-57
Figure 11-16 Motorola SPI continuous frame format (spo = 1, sph = 0).....	11-58
Figure 11-17 Motorola SPI single frame format (spo = 1, sph = 1).....	11-58
Figure 11-18 Motorola SPI continuous frame format (spo = 1, sph = 1).....	11-59
Figure 11-19 TI synchronous serial single frame format	11-59
Figure 11-20 TI synchronous serial continuous frame format.....	11-60
Figure 11-21 National Semiconductor Microwire single frame format.....	11-60
Figure 11-22 National Semiconductor Microwire continuous frame format.....	11-61
Figure 11-23 Functional block diagram of the IR module.....	11-74
Figure 11-24 Frame format for transmitting a single NEC with simple repeat code	11-76
Figure 11-25 Frame format for transmitting continuous NEC with simple repeat codes by holding the key down.....	11-77
Figure 11-26 Definitions of bit0 and bit1 in the NEC with simple repeat code.....	11-77
Figure 11-27 Code format for transmitting a single NEC with simple repeat code.....	11-77
Figure 11-28 Code format for transmitting continuous NEC with simple repeat codes by holding the key down	11-77
Figure 11-29 Frame format for transmitting a single NEC with full repeat code	11-78
Figure 11-30 Frame format for transmitting continuous NEC with full repeat codes by holding the key down	11-78
Figure 11-31 Definitions of bit0 and bit1 in the NEC with full repeat code.....	11-78



Figure 11-32 Code format for transmitting a single NEC with full repeat code	11-79
Figure 11-33 Frame format for transmitting a single TC9012 code	11-79
Figure 11-34 Frame format for transmitting continuous TC9012 code by holding the key down.....	11-80
Figure 11-35 Definitions of bit0 and bit1 of the TC9012 code.....	11-80
Figure 11-36 Code format for transmitting a single TC9012 code	11-80
Figure 11-37 Code format for transmitting continuous TC9012 codes (C0 = 1).....	11-80
Figure 11-38 Code format for transmitting continuous TC9012 codes (C0 = 0).....	11-81
Figure 11-39 Frame format for transmitting a single SONY code.....	11-81
Figure 11-40 Frame format for transmitting continuous SONY codes by holding the key down.....	11-81
Figure 11-41 Definitions of bit0 and bit1	11-82
Figure 11-42 Process of initializing the IR module	11-83
Figure 11-43 Process of reading the decoded data.....	11-84
Figure 12-1 Schematic diagram of the system for debugging of the ARM software.....	12-2



表格目录

Table 2-1 I/O type of pins	2-1
Table 2-2 Power pins	2-2
Table 2-3 SIO pins	2-22
Table 2-4 VDAC pins	2-23
Table 2-5 DDRA pins	2-26
Table 2-6 DDRB pins	2-30
Table 2-7 EBI pins	2-34
Table 2-8 GMAC pins.....	2-36
Table 2-9 VO pins	2-38
Table 2-10 PCI pins	2-39
Table 2-11 SYS pins	2-43
Table 2-12 I2C pins.....	2-44
Table 2-13 JTAG pins	2-44
Table 2-14 UART pins	2-45
Table 2-15 USB pins.....	2-45
Table 2-16 VI pins	2-46
Table 2-17 Software multiplexing pins of the VI interface.....	2-48
Table 2-18 Software multiplexing signals of the VI interface	2-50
Table 2-19 Software multiplexing pins of the VO interface	2-52
Table 2-20 Software multiplexing signals of the VO interface.....	2-53
Table 2-21 Software multiplexing pins of the I2C interface.....	2-56
Table 2-22 Software multiplexing signals of the I2C interface.....	2-56
Table 2-23 Software multiplexing pins of the SIO interface.....	2-56
Table 2-24 Software multiplexing signals of the SIO interface	2-56
Table 2-25 Software multiplexing pins of the EBI interface.....	2-57
Table 2-26 Software multiplexing signals of the EBI interface	2-57



Table 2-27 Software multiplexing pins of the GMAC interface	2-58
Table 2-28 Software multiplexing signals of the GMAC interface.....	2-58
Table 2-29 Software multiplexing pins of the PCI interface.....	2-58
Table 2-30 Software multiplexing signals of the PCI interface	2-58
Table 2-31 Hardware multiplexing pins of the EBI interface	2-59
Table 2-32 Hardware multiplexing signals of the EBI interface.....	2-60
Table 2-33 Hardware multiplexing pins of the PCI interface	2-62
Table 2-34 Hardware multiplexing signals of the PCI interface.....	2-62
Table 2-35 Summary of the IO config registers (base address: 0x200F_0000).....	2-63
Table 2-36 DC parameters (DVDD33 = 3.3 V)	2-109
Table 2-37 DC parameters (DVDD18 = 1.8 V)	2-109
Table 2-38 Recommended operating conditions.....	2-110
Table 2-39 Clock parameters	2-113
Table 2-40 Memory device parameters.....	2-113
Table 2-41 Package/Board parameters.....	2-114
Table 2-42 Timing parameters of the MII interface	2-116
Table 2-43 Timing parameters of the RGMII interface.....	2-117
Table 2-44 Timing parameters of the MDIO interface.....	2-118
Table 2-45 Transmit timing parameters of the MDIO interface.....	2-118
Table 2-46 Timing parameters of the VI interface	2-119
Table 2-47 Timing parameters of the VO interface.....	2-119
Table 2-48 Timing of the VO concatenated input interface	2-120
Table 2-49 Timing parameters of the PCI interface	2-121
Table 2-50 Timing parameters of the I ² C interface	2-122
Table 2-51 Timing parameters of the MMC	2-123
Table 2-52 Timing parameters of the SSP interface.....	2-124
Table 2-53 Timing parameters of the data receive signal URXD of the UART interface.....	2-126
Table 2-54 Timing parameters of the data transmit signal UTXD of the UART interface.....	2-126
Table 2-55 Timing parameters of the I ² S interface	2-127
Table 2-56 Timing parameters of the PCM interface.....	2-128
Table 2-57 Package parameters.....	2-130
Table 2-58 Statistics on the Hi3520 pins.....	2-131
Table 2-59 Pin arrangement	2-136



Table 3-1 Types of reset signals	3-2
Table 3-2 PLL configuration registers of the Hi3520	3-5
Table 3-3 Formulas for calculating the frequency of each PLL pin of the Hi3520	3-5
Table 3-4 Configuration parameters related to the APLL of the Hi3520	3-6
Table 3-5 Configuration parameters related to the EPLL of the Hi3520	3-7
Table 3-6 Configuration parameters related to VPLL0 of the Hi3520	3-8
Table 3-7 Configuration parameters related to VPLL1 of the Hi3520	3-9
Table 3-8 Frequency configuration of the ARM/SCLK/HCLK/PCLK	3-11
Table 3-9 Mapping between the system controller status and clock switching	3-11
Table 3-10 Clock frequency configuration of the MMC module	3-12
Table 3-11 Clock frequency configuration of the PCI module	3-12
Table 3-12 Clock frequency configuration of the SMI module	3-12
Table 3-13 Clock frequency configuration of the VO module	3-13
Table 3-14 Clock frequency configuration of the VI module	3-13
Table 3-15 Clock frequency configuration of the ETH module	3-14
Table 3-16 Clock frequency configuration of the SIO0/SIO1/SIO2 module	3-14
Table 3-17 Selection of the boot mode	3-17
Table 3-18 Address space list	3-26
Table 3-19 Allocation of the PCI address space	3-30
Table 3-20 Interrupt mapping of VIC0	3-32
Table 3-21 Interrupt mapping of VIC1	3-33
Table 3-22 Summary of INT registers	3-35
Table 3-23 DMAC hardware request signals	3-43
Table 3-24 Summary of DMAC registers (base address: 0x0x100D_0000)	3-46
Table 3-25 Mapping between the value of DBSize or SBSize and the burst size	3-63
Table 3-26 Mapping between the value of DWidth or SWidth and the transfer bit width	3-63
Table 3-27 Three protection bits of the prot field of DMAC_CX_CONTROL	3-64
Table 3-28 Flow controller and transfer type corresponding to the flow_cntrl field	3-67
Table 3-29 Summary of CIPHER register (base address: 0x100C_0000)	3-80
Table 3-30 Summary of timer registers (base addresses: 0x2000_0000, 0x2001_0000, 0x2002_0000, and 0x2003_0000)	3-98
Table 3-31 Watchdog interface signal	3-108
Table 3-32 Summary of watchdog registers (base address: 0x2004_0000)	3-110
Table 3-33 Summary of RTC registers (base address: 0x2006_0000)	3-117



Table 3-34 Relationship between the status of the system controller and the system clock	3-122
Table 3-35 Summary of system controller registers (base address: 0x2005_0000)	3-128
Table 4-1 Signals of the DDRC interface	4-1
Table 4-2 DDR2 SDRAMs supported by the Hi3520 DDRC	4-7
Table 4-3 Command truth table of the DDRC	4-8
Table 4-4 Address mapping when the DDRC is in 16-bit mode	4-10
Table 4-5 Address mapping when the DDRC is in 32-bit mode	4-11
Table 4-6 Summary of DDRC registers	4-14
Table 4-7 Interface signals of the SMI controller	4-30
Table 4-8 Configuration of timing parameters of the SMI controller (bus clock fBUSCLK = 200 MHz)	4-33
Table 4-9 Read/write timing parameters of the SMI controller	4-34
Table 4-10 Page read timing parameters of the SMI controller	4-35
Table 4-11 Memory address space supported by the SMI controller	4-36
Table 4-12 Summary of SMI controller registers (base address: 0x1010_0000)	4-38
Table 4-13 Signals of NANDC interfaces	4-53
Table 4-14 Boot configuration pins	4-57
Table 4-15 K9F2G08U0M addresses	4-58
Table 4-16 K9GAG08X0M addresses	4-59
Table 4-17 Common commands for operating the NAND flash memories	4-59
Table 4-18 Summary of NANDC registers (base address: 0x1000_0000)	4-65
Table 5-1 Signals of external interfaces	5-2
Table 5-2 Contents of the Ethernet II frame	5-4
Table 5-3 Summary of GMAC registers (base address: 0x1009_0000)	5-7
Table 5-4 Variables in the offset address of GMAC registers	5-7
Table 6-1 Signals of the VI interface	6-2
Table 6-2 Format of the ITU-R BT.656 YCbCr 4:2:2 line data	6-5
Table 6-3 Formats of SAV and EAV	6-5
Table 6-4 Valid values of SAV and EAV	6-5
Table 6-5 ITU-R BT.656 error-correcting codes	6-6
Table 6-6 Formats of SAV and EAV with channel IDs	6-9
Table 6-7 Summary of VIU registers (base address: 0x1010_0000)	6-24
Table 6-8 Signals of the VOU digital output interface	6-77
Table 6-9 Signals of the VOU analog output interface	6-79



Table 6-10 Summary of VOU registers (base address: 0x2013_0000).....	6-88
Table 6-11 Variables in the offset addresses of VOU registers	6-104
Table 7-1 SIO interface signals.....	7-2
Table 7-2 Pin multiplexing.....	7-8
Table 7-3 Summary of SIO registers.....	7-11
Table 8-1 MMC interface signals.....	8-2
Table 8-2 Load parameters of signal lines	8-3
Table 8-3 Reference configuration of MMC_CMD for non-data transfer command	8-11
Table 8-4 Reference configuration of MMC_CMD during the process of reading single-block or multiple-block data.....	8-12
Table 8-5 Reference configuration of MMC_CMD during the process of writing single-block or multiple-block data.....	8-14
Table 8-6 Reference configuration of MMC_CMDARG during a resume operation.....	8-17
Table 8-7 Summary of MMC registers (base address: 0x1003_0000).....	8-18
Table 9-1 PCI interface signals	9-2
Table 9-2 PCI interface signals	9-9
Table 9-3 Registers at the AHB side (base address: 0xB000_0000).....	9-11
Table 9-4 Summary of PCI configuration space registers.....	9-12
Table 10-1 Interface signals of the USB 2.0 host.....	10-3
Table 10-2 Summary of USB registers (base address: 0x100B_0000)	10-5
Table 11-1 Signals of the I ² C interface	11-2
Table 11-2 Typical values of I2C_SS_SCL_HCNT.....	11-4
Table 11-3 Typical values of I2C_SS_SCL_LCNT	11-4
Table 11-4 Typical values of I2C_FS_SCL_HCNT.....	11-5
Table 11-5 Typical values of I2C_FS_SCL_LCNT	11-5
Table 11-6 I ² C register summary (base address: 0x200D_0000).....	11-8
Table 11-7 Signals of the UART0 interface	11-31
Table 11-8 Signals of the UART1 interface	11-32
Table 11-9 Signals of the UART2 interface	11-32
Table 11-10 Signals of the UART3 interface	11-32
Table 11-11 Summary of UART registers.....	11-38
Table 11-12 SPI interface signals.....	11-53
Table 11-13 Typical configuration of the SPI clock dividers	11-62
Table 11-14 Summary of SPI registers (base address: 0x200E_0000).....	11-65



Table 11-15 IR interface signals.....	11-73
Table 11-16 Statistics on the code formats of the infrared receive data (NEC with simple repeat code).....	11-74
Table 11-17 Statistics on the code formats of the infrared receive data (NEC with full repeat code).....	11-75
Table 11-18 Statistics on the code formats of the infrared receive data (TC9012 and SONY).....	11-76
Table 11-19 Summary of IR registers (base address: 0x2007_0000).....	11-85
Table 11-20 GPIO interface signals	11-99
Table 11-21 Configuration of pin multiplexing	11-103
Table 11-22 Base addresses of the eight sets of GPIO pins	11-107
Table 11-23 GPIO register summary	11-108
Table 12-1 Operating modes of the Hi3520.....	12-1
Table 12-2 JTAG interface signals of the Hi3520.....	12-2
Table 12-3 Connections to the ARM926 and ARM1176	12-3



前言

概述

本文档介绍了 Hi3520 芯片的特性、逻辑结构，详细描述各个模块的功能、工作方式、相关寄存器定义，用图表的方式给出了接口时序关系和相关参数，并详细描述了芯片的管脚定义和用途以及芯片的性能参数和封装尺寸。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3520 H.264 编解码处理器	V100


读者对象

本文档主要适用于以下工程师：

- 电子产品设计维护人员
- 电子产品元器件市场销售人员

约定符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。



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

通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display” 格式	“Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。

表格内容约定

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

寄存器访问类型约定

类型	说明	类型	说明
RO	只读，不可写。	W0C	可读，写 0 清零，写 1 保持不变。
WO	只写。	W1S	可读，写 1 置 1，写 0 保持不变。



类型	说明	类型	说明
RW	可读可写。	W0S	可读, 写 0 置 1, 写 1 保持不变。
RC	读清零。	OSW	可读, 写 1 后片内自清零, 即产生一个脉冲。
W1C	可读, 写 1 清零, 写 0 保持不变。		

数值单位约定

数据容量、频率、数据速率等的表达方式说明如下。

类别	符号	对应的数值
数据容量 (如 RAM 容量)	1K	1024
	1M	1,048,576
	1G	1,073,741,824
频率、数据速率等	1k	1000
	1M	1,000,000
	1G	1,000,000,000

地址、数据的表达方式说明如下。

符号	举例	说明
0x	0xFE04、0x18	用 16 进制表示的数据值、地址值。
0b	0b000、0b00 00000000	表示 2 进制的数据值以及 2 进制序列 (寄存器描述中除外)。

修订记录

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

修订日期	版本	修订说明
2010-01-19	02	第 2 章 硬件特性 修改 2.5 复用寄存器描述中 reg2~reg17 中涉及 VOUDAT0~VOUDAT15 的复用描述。



修订日期	版本	修订说明
2009-12-23	01	<p>第 1 章 产品概述</p> <p>1.1.14 硬件特性中的工作环境温度由“0℃~70℃”；增加芯片封装表面温度和结温。</p> <p>第 2 章 硬件特性</p> <p>表 2-4 补充 RSET 管脚的计算公式和举例。</p> <p>修改表 2-32 中 NFPAGE0 和 NFPAGE1 的描述，将 00 和 11 改为保留；FUNSEL0 和 FUNSEL1 的描述中增加 01 和 10 的描述。</p> <p>表 2-38 中将工作环境温度由“0℃~70℃”；增加芯片封装表面温度和结温。</p> <p>第 3 章 系统控制器</p> <p>3.10.5 寄存器描述中 SC_PERCTRL23 bit[10:9]的描述中将 00 和 11 的含义改为保留。</p> <p>第 4 章 存储控制器</p> <p>删除对 512B、8KB page size NAND Flash 的相关内容描述。</p>
2009-11-30	00B60	<p>第 11 章 其它外设接口</p> <p>删除 GPIO 章节中关于支持漏极输出的特性。</p>
2009-11-20	00B50	<p>第 4 章 存储控制器</p> <p>修改“4.3.5 工作方式”下“时钟门控”中的步骤三涉及配置的系统控制器的bit位及配置值，由“向系统寄存器SC_PERDIS[nandc_clken]写0”改为“向系统寄存器SC_PERDIS[nandclkdis]写1”。</p> <p>第 11 章 其它外设接口</p> <p>修改“11.2.5 工作方式”下“时钟门控”中的步骤三涉及配置的系统控制器的配置值，由“配置系统控制器 SC_PERDIS[uartclkdis]=1”改为“配置系统控制器 SC_PERDIS[uartclkdis]=0xF”。</p>
2009-10-30	00B40	<p>第 4 章 存储控制器</p> <p>4.1.4.2 DDR 地址映射表 4-4、表 4-5 的 2Gbit 地址线分配进行修改。</p> <p>第 6 章 视频接口</p> <p>修改 VIn_PORT_CFG bit[11:10]的描述；删除 bit[7:6]中“说明：当数据接收为 BT.656 模式时，该 2bit 有效。”的描述。</p>



修订日期	版本	修订说明
2009-08-28	00B30	<p>第 3 章 系统控制</p> <p>管脚 XIN 改为 XIN24。</p> <p>修改 CIPHER_CTRL bit[7:6]中 3DES 算法下的密钥长度描述。</p> <p>第 4 章 存储控制器</p> <p>4.1.5.3 DDR 初始化步骤 4，把 DDR 突发长度（burst length）设定为 4 改为 8。</p> <p>第 6 章 视频接口</p> <p>6.1.5.3 时钟配置中删除“端口 0 或端口 2 工作在 4 路时分复用”的描述。</p> <p>第 9 章 PCI</p> <p>PCIAHB_SIZ_NP 寄存器的复位值由 FF000000 改为 FFC00000；PCIAHB_SIZ_PF 复位值由 FF000000 改为 FF800000。</p>
2009-07-30	00B20	<p>全规格版本，包括 1~13 章、附录。</p> <p>第 2 章硬件章节相对 B01/B02 主要修改点：</p> <ol style="list-style-type: none"> 1、AVDD 和 AVSS 管脚信号名修改为 AVDD_DAC 和 AVSS_DAC。 2、DDR 参考电压 VREFA 和 VREFB 管脚信号名改成 VREF。 3、D22 管脚功能改为 DDRA_UDM0。 4、D11 管脚功能改为 DDRA_LDM1。 5、VI 管脚复用修改：VI0DAT0~VI0DAT7 复用为 VOUDAT8~VOUDAT15，VI1DAT0~VI1DAT7 复用为 VOUDAT0~VOUDAT7。 6、增加封装尺寸描述和 PinMap 图。
2009-5-30	00B01/00B02	<p>初始版本，只有第 1 章概述和第 2 章硬件特性章节。</p>



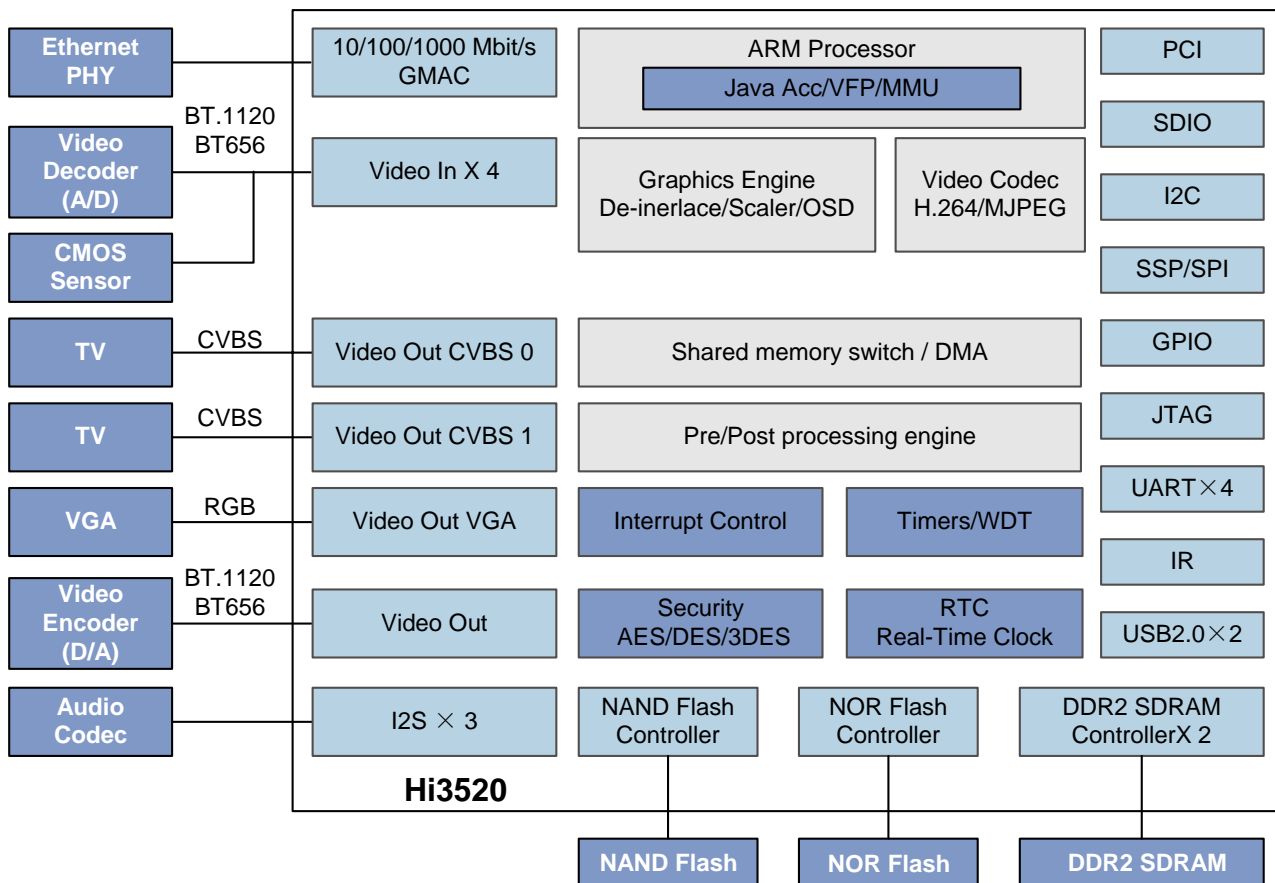
1 产品概述

1.1 体系架构

1.1.1 概述

Hi3520 芯片逻辑框图如图 1-1 所示。

图1-1 Hi3520 芯片逻辑框图





1.1.2 处理器系统

Hi3520 处理器系统基于高性能的 ARM1176 和 ARM926 双核处理器搭建，ARM1176 CPU（Central Processing Unit）处理频率达到 600MHz。ARM1176 为主控处理器，ARM926 为视音频业务控制处理器，双核架构将能够提供更加灵活和丰富的应用业务，更高性能的音视频业务。处理器系统主要包含以下几个部分：

- ARM1176 处理器：作为整个系统的主控 CPU，协同硬件加速器一起完成音视频编解码功能以及系统调度操作。处理器内嵌 16KB 指令 cache 和 16KB 数据 cache，内嵌 2KB 指令紧耦合存储器 ITCM（Instruction Tightly-Coupled Memory），工作频率可以达到 600MHz。
- 直接存储器访问控制器：DMAC（Direct Memory Access Controller）在存储器 and 外设之间、外设和外设之间、存储器和存储器之间进行数据传输。
- 中断系统：为系统提供中断管理功能。支持 32 个中断源。
- 时钟：为整个系统提供时钟管理，包括芯片主时钟管理、各个模块的门控时钟管理。
- 复位：对整个系统的复位、各功能模块的复位进行统一的管理，包括：上电复位的管理和控制、系统软复位、功能模块单独软复位控制。
- 定时器：提供 4 组 Dual-Timer，每组 Dual-Timer 提供 2 个独立的计时器。
- 看门狗：用于系统异常情况下，复位整个系统。
- 实时时钟：用于实现时间显示和定时报警功能。
- 系统控制器：提供控制系统运行的手段，它控制系统运行的模式，监控系统运行状态，管理系统中的重要模块（如时钟、复位、管脚复用等），完成对外设的某些功能的配置。

1.1.3 图形处理

Hi3520 图形处理模块对视频输入图像或者视频输出图像进行加工处理，使其能够适应各种不同的应用场景，达到更好的图像显示效果。主要特点如下：

- 支持对输入图像的 De-interlace 处理。
- 支持对输出图像的 De-interlace 处理，用于逐行显示系统，支持 60 场到 60 帧或者 60 场到 30 帧的转换。
- 支持图像色彩和对比度增强及图像去噪功能。
- 支持 clip、alpha blending、ROP、colorkey 和 Gamma 校正等功能。
- 支持图形缩放（不超过 16 倍的任意大小的缩放功能）。
- 支持前后 OSD（On Screen Display）图像叠加功能，同时支持 4 个区域的视频遮盖功能。
- 支持图像输出抗闪烁功能。
- 支持 2D 数据拷贝和数据填充等功能。



1.1.4 视频编解码处理

Hi3520 集成高性能的 H.264/MJPEG/JPEG 硬件编码器和解码器，主要特点如下：

- H.264 Main Profile @ Level 3 编解码
- H.264 BaseLine @ Level 3 编解码
- JPEG/MJPEG Baseline 编解码
- H.264 实时编码或解码的最大性能为 8 路 D1
- H.264 编码或解码可支持最大的分辨率为 1280×1024@30fps
- H.264 同编同解支持：
 - 120fps D1 编码+120fps CIF 编码+120fps D1 解码@NTSC
 - 100fps D1 编码+100fps CIF 编码+100fps D1 解码@PAL
- 支持双码流编码，双码流可分别选择 H.264 或 MJPEG 协议
- MJPEG/JPEG 最大支持 300 万像素的编码，帧率为 10f/s
- H.264 支持 CBR/VBR/ABR 码率控制，范围为 16kbit/s~20Mbit/s

1.1.5 加解密引擎

加解密引擎集成 AES (Advanced Encryption Standard) /DES (Data Encryption Standard) /3DES 多种加解密算法以及数字水印加解密技术，主要特点如下：

- DES/3DES 和 AES 算法符合 FIPS46-3/FIPS 197 标准，DES/3DES 和 AES 的工作模式均符合 FIPS -81/NIST special800-38a 标准
- 支持数字水印技术

1.1.6 存储控制器接口

存储控制器接口主要特点如下：

- 2 个 DDR2 (Double Data-Rate Controller) 接口
 - 最高工作频率为 400MHz，能够为整个系统提供充足的带宽
 - 数据位宽分别为 16bit 和 32bit
 - 16bit DDR2 SDRAM 最大支持 256MB 存储器空间
 - 32bit DDR2 SDRAM 最大支持 512MB 存储器空间
- NOR Flash 接口
 - 8bit 数据位宽
 - 2 个片选，每个片选最大支持 32MB
 - 支持 NOR Flash 启动
- NAND Flash 接口
 - 8bit 数据位宽



- 支持 SLC、MLC；1、4、8bit ECC
- 最大支持 8GB 容量
- 支持 NAND Flash 启动

1.1.7 以太网接口

符合 802.3 标准的 10/100/1000 Mbit/s 以太网接口，实现外部端口和 ARM1176 主控处理器间无阻塞的数据交换，主要特点如下：

- 支持 10/100 Mbit/s 全双工、半双工模式；支持 1000Mbit/s 全双工模式
- 提供 RGMII 和 MII 接口
- 支持 MDIO 功能
- 支持流控帧的发送和接收
- 支持 MAC 地址过滤
- 提供流量限制功能。
- 提供收发接收错包、丢包、超短包、超长包、单播包、多播包等计数调试功能

1.1.8 视频接口

视频接口包括 4 个视频输入接口、2 个数字视频输出接口、6 个模拟视频输出通道。工作频率支持 27/54/108 MHz，主要特点如下：

- 视频输入接口：
 - 支持 4 路 BT.656 YCrCb 4:2:2 接口，8bit，27/54/108MHz
 - 支持 2 路标准 SMPTE296M、BT.1120 高清时序
 - 支持 Digital Camera 数字接口，支持的最大分辨率为 1280×1024@30fps、1600×1200@20fps、2048×1536@10fps
- 视频输出接口：
 - 多路视频输出接口
 - VGA/YPbPr×1+CVBS×2
 - VGA×2
 - 支持高清输出，最大分辨率 1920×1080p@30fps，1920×1080i@60fps
 - 支持 BT.656、BT.1120、LCD 数字输出

1.1.9 音频接口

Hi3520 音频输入输出接口 SIO (Sonic Input/Output) 包括 3 个标准的 I²S (Inter-IC Sound) 音频接口，能够兼容主流的各种单路和多路级连的 Audio codec 芯片。音频接口主要特点如下：

- 每个 I²S 接口支持 16 路 8bit 或 16bit 的音频级联输入
- 支持 8/16/32 位采样精度
- 支持 8/16/32/44.1/48 kHz 采样频率



1.1.10 MMC/SD/SDIO 控制器

MMC (Multi-media Card) /SD (Secure Digital) /SDIO (Secure Digital Input/Output) 控制器用来处理对 SD/MMC 存储卡的读写等操作, 并可以通过 SDIO 协议实现对扩展外设 (如 Blue Tooth、WiFi 等) 的支持。MMC/SD/SDIO 控制器可以控制符合以下协议的设备:

- SD mem-version 2.00
- SDIO-version 1.10
- MMC-version 4.2

1.1.11 PCI 接口

Hi3520 PCI (Peripheral Component Interconnect) 总线是一种通用的本地总线 (Local Bus)。PCI 接口以其灵活的可扩展性用于实现产品的不同应用形态和应用场合, 实现多设备级联和扩展符合 PCI/miniPCI 接口的外设, 如 SATA、WiFi、PCI-to-PCI Bridge 等。PCI 接口主要特点如下:

- 符合 PCI2.3 总线协议, 并兼容 miniPCI 接口协议
- 支持主桥 (Host Bridge) 模式和从桥 (Simple Bridge) 模式, 既可以作为主控制器对外扩展 PCI 设备 (最大支持 5 个 PCI Device), 也可以作为 PCI 接口的从设备。
- PCI 支持 33MHz 或者 66MHz

1.1.12 USB 接口

Hi3520 集成了 2 个 USB (Universal Serial Bus) 2.0 Host, 主要特点如下:

- 兼容 USB 2.0
- 符合 OHCI Rev 1.0a、EHCI Rev 1.0
- 可以支持 High-speed、Full-speed、Low-speed 三种传输模式
- 支持低功耗的解决方案
- 支持 Control Transfer、Bulk Transfer、Isochronous Transfer、Interrupt Transfer 四种基本数据传输类型
- 可以通过连接 USB Hub, 最多连接 127 个设备

1.1.13 其他外设接口

其他外设接口特点如下:

- I²C
I²C (The Inter-Integrated Circuit) 控制器实现标准 I²C 主从设备功能, 兼容 Philips I²C 总线协议, 可完成对 I²C 总线上的从设备的数据发送和接收。
- UART
UART (Universal Asynchronous Receiver Transmitter) 是一个异步串行的通信接口, 可以和外部芯片的 UART 对接, 实现芯片间的通信。Hi3520 支持 4 个 UART



接口，其中 3 个支持基本的串行数据传输功能，另 1 个还支持 RTS、CTS 硬件流控功能。

- SPI

SPI (Synchronous Peripheral Interface) 控制器可以作为一个 master 或 slave 与外部的设备来进行同步串行通信。支持以下通信协议：

- A Motorola SPI (Synchronous Peripheral Interface) -compatible interface
- A Texas Instruments synchronous serial interface
- A National Semiconductor Microwire interface

- IR

IR (Infrared Remoter) 通过红外接口接收红外数据，可以支持 NEC with simple repeat code、NEC with full repeat code、SONY 和 TC9012 四种数据格式解码，并且具备接收数据错误检测和红外遥控唤醒等功能。

- GPIO

Hi3520 最大支持 8 组 GPIO (General Purpose Input/Output)，每组 GPIO 提供 8 个可编程的输入输出管脚。GPIO 与其它业务管脚复用。每个管脚可以配置为输入或者输出，用于产生特定应用的输出信号或采集特定应用的输入信号。

1.1.14 硬件特性

- 1600mW 典型功耗 (DVR 应用场景)
- 支持多级省电模式
- 90nm 工艺，1.0/1.8/3.3 V 芯片供电电压
- 768 pin FPBGA 封装，0.8mm 管脚间距，27mm×27mm 封装尺寸
- 工作环境温度：-20℃~+85℃
- 芯片封装表面温度：-20℃~+105℃
- 芯片结温：-40℃~+125℃

1.2 应用场景

Hi3520 是一款基于 ARM11 处理器内核以及视频硬件加速引擎的高性能通信媒体处理器，具有高集成、可编程、支持 MPEG-4 AVC/H.264 和 MJPEG 等多协议的优点，可广泛应用于实时视频通信、数字图像监控等领域。

Hi3520 的主要应用场景如下：

- [16 路 CIF 硬盘录像机 DVR](#)
- [4 路 D1 硬盘录像机 DVR](#)
- [8 路 D1 硬盘录像机 DVR](#)

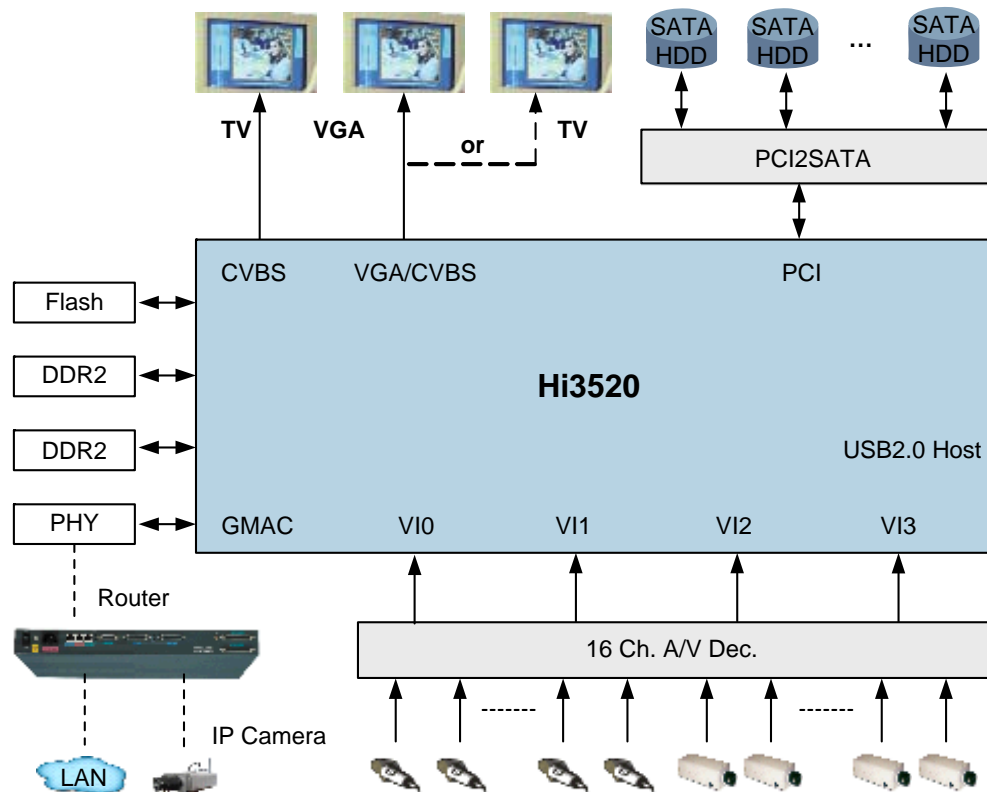


16 路 CIF 硬盘录像机 DVR

Hi3520 应用于 16CIF 硬盘录像机 DVR (Digital Video Recorder) 的应用框图如图 1-2 所示。其主要规格如下:

- 1280×1024 VGA 显示输出
- 16 路 CIF 录像
- 16 路 QCIF 网传
- 16 路 CIF 解码回放

图1-2 Hi3520 16CIF 硬盘录像机 DVR 应用框图



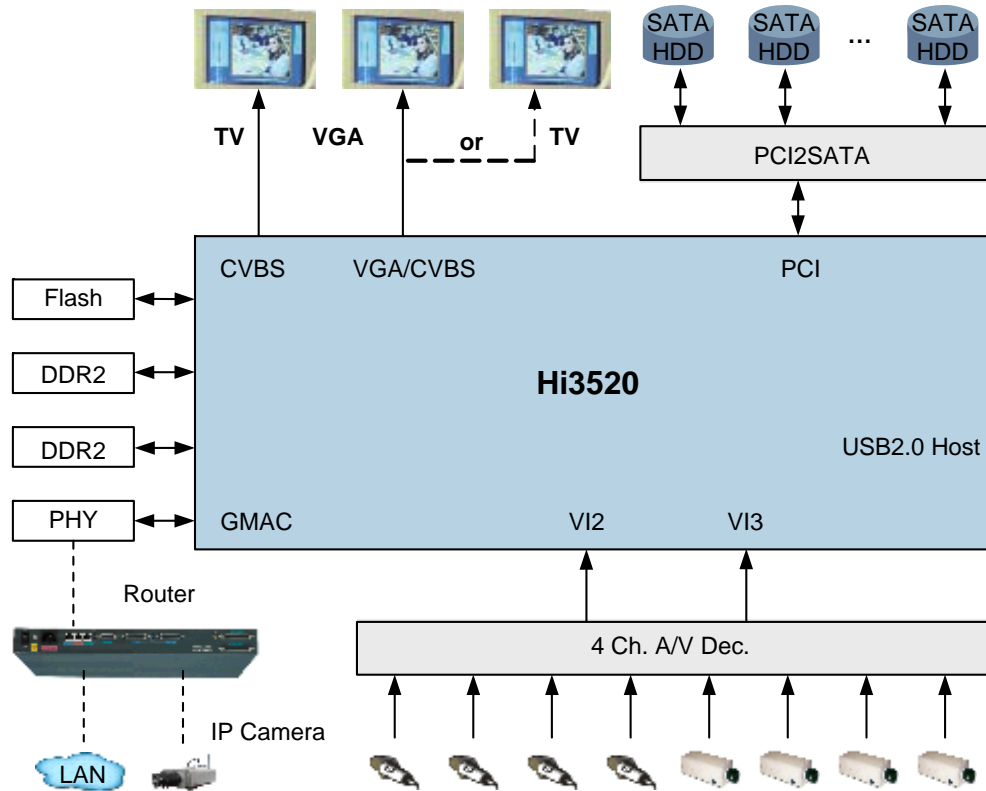


4 路 D1 硬盘录像机 DVR

Hi3520 应用于 4D1 硬盘录像机 DVR 的应用框图如图 1-3 所示。其主要规格如下：

- 1280×1024 VGA 显示输出
- 4 路 D1 录像
- 4 路 CIF 网传
- 4 路 D1 解码回放

图1-3 Hi3520 4D1 硬盘录像机 DVR 应用框图



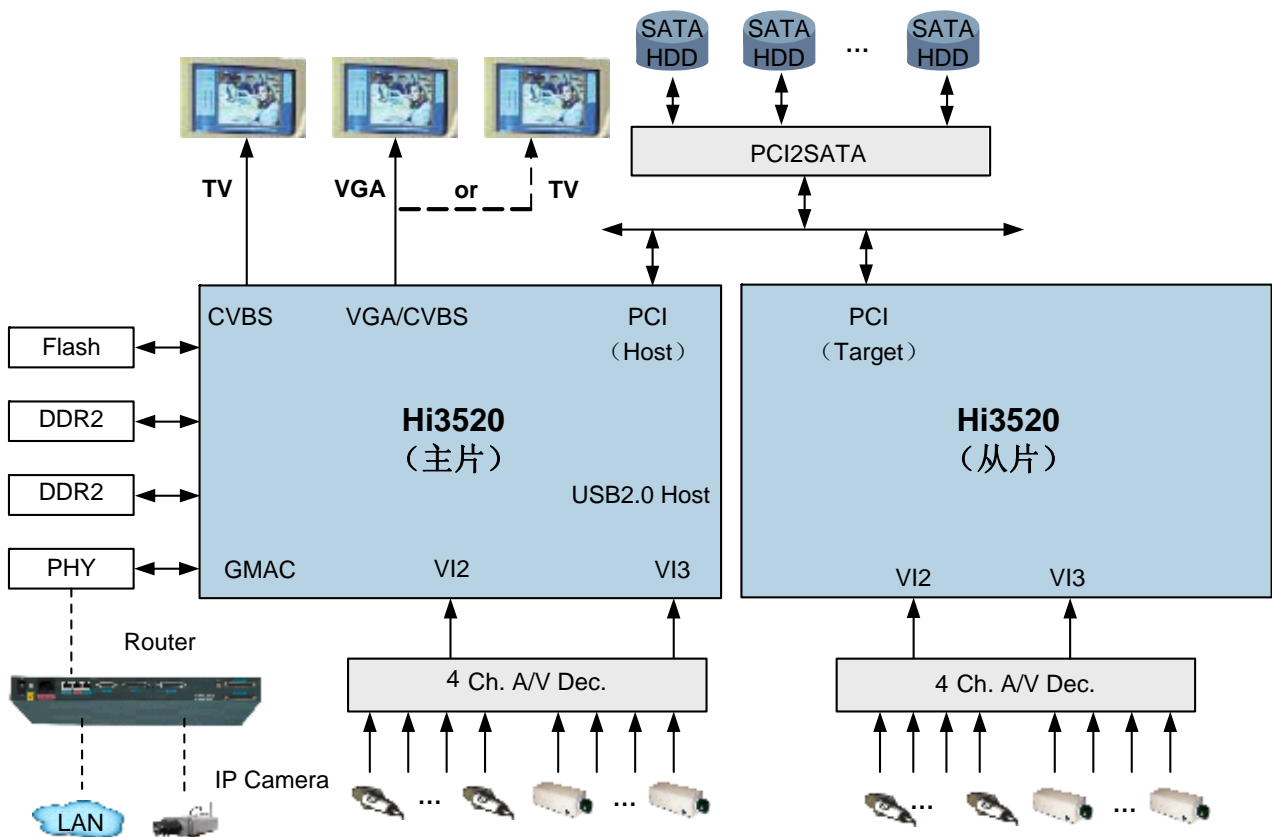


8 路 D1 硬盘录像机 DVR

Hi3520 应用于 8D1 硬盘录像机 DVR 的应用框图如图 1-4 所示。其主要规格如下：

- 1280×1024 VGA 显示输出
- 8 路 D1 录像
- 8 路 CIF 网传
- 8 路 D1 解码回放

图1-4 Hi3520 8D1 硬盘录像机 DVR 应用框图





2 硬件特性

2.1 管脚描述

管脚 I/O 类型说明如表 2-1 所示。

表2-1 管脚 I/O 类型说明

I/O	说明
I	输入信号。
I _{PD}	输入信号，内部下拉。
I _{PU}	输入信号，内部上拉。
I _S	输入信号，带施密特触发器。
I _{SPD}	输入信号，带施密特触发器，内部下拉。
I _{SPU}	输入信号，带施密特触发器，内部上拉。
O	输出信号。
O _{OD}	输出，漏极开路。
I/O	双向输入/输出信号。
I _{PD} /O	双向，输入下拉。
I _{PU} /O	双向，输入上拉。
I _{SPU} /O	双向，输入上拉，带施密特触发器。
I _{PD} /O _{OD}	双向，输入下拉，输出漏极开路。
I _{PU} /O _{OD}	双向，输入上拉，输出漏极开路。
I _S /O	双向，输入带施密特触发器。
I _S /O _{OD}	双向，输入带施密特触发器，输出漏极开路。
CIN	Crystal Oscillator，晶振输入。



I/O	说明
COUT	Crystal Oscillator, 晶振输出。
P	电源。
G	地。

2.1.1 POWER 管脚

POWER 管脚如表 2-2 所示。

表2-2 POWER 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
VDAC 电源和地					
AB8	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AE11	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AL5	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AK3	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AK4	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AJ3	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AH2	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AH7	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AG3	AVDD33_D AC	-	-	3.3	VDAC 3.3V 模拟电源。
AD10	DVDD10_D AC	-	-	1.0	VDAC 1.0V 数字电源。
AD8	DVDD33_D AC0	-	-	3.3	VDAC0 3.3V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AE9	DVDD33_D AC1	-	-	3.3	VDAC1 3.3V 数字电源。
AM1	AVSS_DAC	-	-	-	VDAC 模拟地。
AM5	AVSS_DAC	-	-	-	VDAC 模拟地。
AL2	AVSS_DAC	-	-	-	VDAC 模拟地。
AL3	AVSS_DAC	-	-	-	VDAC 模拟地。
AL4	AVSS_DAC	-	-	-	VDAC 模拟地。
AK2	AVSS_DAC	-	-	-	VDAC 模拟地。
AJ2	AVSS_DAC	-	-	-	VDAC 模拟地。
AJ4	AVSS_DAC	-	-	-	VDAC 模拟地。
AJ5	AVSS_DAC	-	-	-	VDAC 模拟地。
AH1	AVSS_DAC	-	-	-	VDAC 模拟地。
AH3	AVSS_DAC	-	-	-	VDAC 模拟地。
AH6	AVSS_DAC	-	-	-	VDAC 模拟地。
AD11	AVSS_DAC	-	-	-	VDAC 模拟地。
AB9	AVSS_DAC	-	-	-	VDAC 模拟地。
AA12	AVSS_DAC	-	-	-	VDAC 模拟地。
AA13	AVSS_DAC	-	-	-	VDAC 模拟地。
Y12	AVSS_DAC	-	-	-	VDAC 模拟地。
AC9	DVSS10_D AC	-	-	-	VDAC 1.0V 对应数字地。
AE8	DVSS33_D AC	-	-	-	VDAC 3.3V 对应数字地。
AD9	DVSS33_D AC	-	-	-	VDAC 3.3V 对应数字地。
PLL 电源和地					
U28	AVDD33_A EPLL	-	-	3.3	ARM 和 ETH PLL1 3.3V 模拟电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
T28	AVDD33_V PLL	-	-	3.3	Video PLL 3.3V 模拟电源。
V24	VDD10_AP LL	-	-	1.0	ARM PLL 1.0V 数字电源。
U24	VDD10_EPL L	-	-	1.0	ETH PLL 1.0V 数字电源。
T24	VDD10_VP LL0	-	-	1.0	Video 0 PLL 1.0V 数字电源。
T25	VDD10_VP LL1	-	-	1.0	Video 1 PLL 1.0V 数字电源。
V25	VSS_APLL	-	-	-	ARM PLL 地。
V28	VSS_APLL	-	-	-	ARM PLL 地。
U25	VSS_EPLL	-	-	-	ETH PLL 地。
R25	VSS_VPLL	-	-	-	Video PLL 地。
R28	VSS_VPLL	-	-	-	Video PLL 地。
USB 电源和地					
K5	USBVDD	-	-	1.0	USB 2.0 HOST 数字电源，外接 1.0V。
N8	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
R8	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
P9	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
M1	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
M2	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
M3	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
L4	USBVDDA3 3	-	-	3.3	USB 2.0 HOST 模拟电源，外接 3.3V。
L5	USBVSS	-	-	-	USB 2.0 HOST 数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
P8	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
R9	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
N9	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
L3	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
K1	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
K2	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
K4	USBVSSA3 3	-	-	-	USB 2.0 HOST 模拟地。
内核(CORE)电源					
AA14	DVDD10	-	-	1.0	1.0V 数字电源。
AA15	DVDD10	-	-	1.0	1.0V 数字电源。
AA16	DVDD10	-	-	1.0	1.0V 数字电源。
AA17	DVDD10	-	-	1.0	1.0V 数字电源。
AA18	DVDD10	-	-	1.0	1.0V 数字电源。
AA19	DVDD10	-	-	1.0	1.0V 数字电源。
AA20	DVDD10	-	-	1.0	1.0V 数字电源。
Y21	DVDD10	-	-	1.0	1.0V 数字电源。
W12	DVDD10	-	-	1.0	1.0V 数字电源。
W21	DVDD10	-	-	1.0	1.0V 数字电源。
V12	DVDD10	-	-	1.0	1.0V 数字电源。
V21	DVDD10	-	-	1.0	1.0V 数字电源。
U12	DVDD10	-	-	1.0	1.0V 数字电源。
U21	DVDD10	-	-	1.0	1.0V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
T12	DVDD10	-	-	1.0	1.0V 数字电源。
T21	DVDD10	-	-	1.0	1.0V 数字电源。
R12	DVDD10	-	-	1.0	1.0V 数字电源。
R21	DVDD10	-	-	1.0	1.0V 数字电源。
P12	DVDD10	-	-	1.0	1.0V 数字电源。
P21	DVDD10	-	-	1.0	1.0V 数字电源。
N12	DVDD10	-	-	1.0	1.0V 数字电源。
N21	DVDD10	-	-	1.0	1.0V 数字电源。
M13	DVDD10	-	-	1.0	1.0V 数字电源。
M14	DVDD10	-	-	1.0	1.0V 数字电源。
M15	DVDD10	-	-	1.0	1.0V 数字电源。
M16	DVDD10	-	-	1.0	1.0V 数字电源。
M17	DVDD10	-	-	1.0	1.0V 数字电源。
M18	DVDD10	-	-	1.0	1.0V 数字电源。
M19	DVDD10	-	-	1.0	1.0V 数字电源。
M20	DVDD10	-	-	1.0	1.0V 数字电源。
DDR2 电源					
J25	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H17	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H16	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H15	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H10	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H9	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
R24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
P24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
P30	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
P31	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
N24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
M24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
M28	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
L24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
K24	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
K30	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
K31	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J10	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J11	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J12	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J13	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J14	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J15	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J16	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J17	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J18	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J19	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J20	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J21	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J22	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
J23	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
G28	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E7	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E8	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E12	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E13	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E16	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E17	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E18	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E22	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E23	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E26	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E27	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E30	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
E31	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
C12	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
C19	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
C23	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
C30	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B4	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B9	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B12	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B19	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B23	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
B27	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
A4	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
A9	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
A27	DVDD18	-	-	1.8	DDR2 1.8V 数字电源。
H24	VREF	-	-	0.9	DDRA 0.9V 参考电源。
H22	VREF	-	-	0.9	DDRA 0.9V 参考电源。
H20	VREF	-	-	0.9	DDRA 0.9V 参考电源。
H18	VREF	-	-	0.9	DDRA 0.9V 参考电源。
H14	VREF	-	-	0.9	DDRA 0.9V 参考电源。
H12	VREF	-	-	0.9	DDRA 0.9V 参考电源。
K25	VREF	-	-	0.9	DDRB 0.9V 参考电源。
M25	VREF	-	-	0.9	DDRB 0.9V 参考电源。
P25	VREF	-	-	0.9	DDRB 0.9V 参考电源。
其它 IO 电源					
K8	DVDD33	-	-	3.3	3.3V 数字电源。
L8	DVDD33	-	-	3.3	3.3V 数字电源。
M8	DVDD33	-	-	3.3	3.3V 数字电源。
T8	DVDD33	-	-	3.3	3.3V 数字电源。
U8	DVDD33	-	-	3.3	3.3V 数字电源。
Y8	DVDD33	-	-	3.3	3.3V 数字电源。
AA8	DVDD33	-	-	3.3	3.3V 数字电源。
AE23	DVDD33	-	-	3.3	3.3V 数字电源。
AE22	DVDD33	-	-	3.3	3.3V 数字电源。
AE19	DVDD33	-	-	3.3	3.3V 数字电源。
AE18	DVDD33	-	-	3.3	3.3V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AE15	DVDD33	-	-	3.3	3.3V 数字电源。
AE14	DVDD33	-	-	3.3	3.3V 数字电源。
W25	DVDD33	-	-	3.3	3.3V 数字电源。
AB25	DVDD33	-	-	3.3	3.3V 数字电源。
AC25	DVDD33	-	-	3.3	3.3V 数字电源。
AH10	DVDD33	-	-	3.3	3.3V 数字电源。
AH11	DVDD33	-	-	3.3	3.3V 数字电源。
AH14	DVDD33	-	-	3.3	3.3V 数字电源。
AH15	DVDD33	-	-	3.3	3.3V 数字电源。
AH18	DVDD33	-	-	3.3	3.3V 数字电源。
AH19	DVDD33	-	-	3.3	3.3V 数字电源。
AH22	DVDD33	-	-	3.3	3.3V 数字电源。
AH23	DVDD33	-	-	3.3	3.3V 数字电源。
AH26	DVDD33	-	-	3.3	3.3V 数字电源。
AH27	DVDD33	-	-	3.3	3.3V 数字电源。
AF5	DVDD33	-	-	3.3	3.3V 数字电源。
AF28	DVDD33	-	-	3.3	3.3V 数字电源。
AE5	DVDD33	-	-	3.3	3.3V 数字电源。
AE28	DVDD33	-	-	3.3	3.3V 数字电源。
AD5	DVDD33	-	-	3.3	3.3V 数字电源。
AD13	DVDD33	-	-	3.3	3.3V 数字电源。
AD14	DVDD33	-	-	3.3	3.3V 数字电源。
AD15	DVDD33	-	-	3.3	3.3V 数字电源。
AD16	DVDD33	-	-	3.3	3.3V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AD17	DVDD33	-	-	3.3	3.3V 数字电源。
AD18	DVDD33	-	-	3.3	3.3V 数字电源。
AD19	DVDD33	-	-	3.3	3.3V 数字电源。
AD20	DVDD33	-	-	3.3	3.3V 数字电源。
AD21	DVDD33	-	-	3.3	3.3V 数字电源。
AD22	DVDD33	-	-	3.3	3.3V 数字电源。
AD23	DVDD33	-	-	3.3	3.3V 数字电源。
AC24	DVDD33	-	-	3.3	3.3V 数字电源。
AB24	DVDD33	-	-	3.3	3.3V 数字电源。
AB28	DVDD33	-	-	3.3	3.3V 数字电源。
AA5	DVDD33	-	-	3.3	3.3V 数字电源。
AA9	DVDD33	-	-	3.3	3.3V 数字电源。
AA24	DVDD33	-	-	3.3	3.3V 数字电源。
AA28	DVDD33	-	-	3.3	3.3V 数字电源。
Y5	DVDD33	-	-	3.3	3.3V 数字电源。
Y9	DVDD33	-	-	3.3	3.3V 数字电源。
Y24	DVDD33	-	-	3.3	3.3V 数字电源。
W9	DVDD33	-	-	3.3	3.3V 数字电源。
W24	DVDD33	-	-	3.3	3.3V 数字电源。
V9	DVDD33	-	-	3.3	3.3V 数字电源。
U5	DVDD33	-	-	3.3	3.3V 数字电源。
U9	DVDD33	-	-	3.3	3.3V 数字电源。
T5	DVDD33	-	-	3.3	3.3V 数字电源。
T9	DVDD33	-	-	3.3	3.3V 数字电源。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
M9	DVDD33	-	-	3.3	3.3V 数字电源。
L9	DVDD33	-	-	3.3	3.3V 数字电源。
K9	DVDD33	-	-	3.3	3.3V 数字电源。
J5	DVDD33	-	-	3.3	3.3V 数字电源。
H5	DVDD33	-	-	3.3	3.3V 数字电源。
G5	DVDD33	-	-	3.3	3.3V 数字电源。
其它数字地					
J8	VSS	-	-	-	数字地。
V8	VSS	-	-	-	数字地。
W8	VSS	-	-	-	数字地。
AE24	VSS	-	-	-	数字地。
AE21	VSS	-	-	-	数字地。
AE20	VSS	-	-	-	数字地。
AE17	VSS	-	-	-	数字地。
AE16	VSS	-	-	-	数字地。
AE13	VSS	-	-	-	数字地。
AE12	VSS	-	-	-	数字地。
L25	VSS	-	-	-	数字地。
N25	VSS	-	-	-	数字地。
Y25	VSS	-	-	-	数字地。
AA25	VSS	-	-	-	数字地。
AD25	VSS	-	-	-	数字地。
AE25	VSS	-	-	-	数字地。
H25	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
H23	VSS	-	-	-	数字地。
H21	VSS	-	-	-	数字地。
H19	VSS	-	-	-	数字地。
H13	VSS	-	-	-	数字地。
H11	VSS	-	-	-	数字地。
H8	VSS	-	-	-	数字地。
AM8	VSS	-	-	-	数字地。
AM13	VSS	-	-	-	数字地。
AM18	VSS	-	-	-	数字地。
AM23	VSS	-	-	-	数字地。
AM28	VSS	-	-	-	数字地。
AM32	VSS	-	-	-	数字地。
AK8	VSS	-	-	-	数字地。
AK12	VSS	-	-	-	数字地。
AK16	VSS	-	-	-	数字地。
AK20	VSS	-	-	-	数字地。
AK24	VSS	-	-	-	数字地。
AK27	VSS	-	-	-	数字地。
AK30	VSS	-	-	-	数字地。
AH8	VSS	-	-	-	数字地。
AH9	VSS	-	-	-	数字地。
AH12	VSS	-	-	-	数字地。
AH13	VSS	-	-	-	数字地。
AH16	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AH17	VSS	-	-	-	数字地。
AH20	VSS	-	-	-	数字地。
AH21	VSS	-	-	-	数字地。
AH24	VSS	-	-	-	数字地。
AH25	VSS	-	-	-	数字地。
AH28	VSS	-	-	-	数字地。
AH32	VSS	-	-	-	数字地。
AG28	VSS	-	-	-	数字地。
AF1	VSS	-	-	-	数字地。
AF30	VSS	-	-	-	数字地。
AE3	VSS	-	-	-	数字地。
AD12	VSS	-	-	-	数字地。
AD24	VSS	-	-	-	数字地。
AD28	VSS	-	-	-	数字地。
AC1	VSS	-	-	-	数字地。
AC5	VSS	-	-	-	数字地。
AC28	VSS	-	-	-	数字地。
AC32	VSS	-	-	-	数字地。
AB5	VSS	-	-	-	数字地。
AA3	VSS	-	-	-	数字地。
AA21	VSS	-	-	-	数字地。
AA30	VSS	-	-	-	数字地。
Y13	VSS	-	-	-	数字地。
Y14	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
Y15	VSS	-	-	-	数字地。
Y16	VSS	-	-	-	数字地。
Y17	VSS	-	-	-	数字地。
Y18	VSS	-	-	-	数字地。
Y19	VSS	-	-	-	数字地。
Y20	VSS	-	-	-	数字地。
Y28	VSS	-	-	-	数字地。
W1	VSS	-	-	-	数字地。
W5	VSS	-	-	-	数字地。
W13	VSS	-	-	-	数字地。
W14	VSS	-	-	-	数字地。
W15	VSS	-	-	-	数字地。
W16	VSS	-	-	-	数字地。
W17	VSS	-	-	-	数字地。
W18	VSS	-	-	-	数字地。
W19	VSS	-	-	-	数字地。
W20	VSS	-	-	-	数字地。
W28	VSS	-	-	-	数字地。
V5	VSS	-	-	-	数字地。
V13	VSS	-	-	-	数字地。
V14	VSS	-	-	-	数字地。
V15	VSS	-	-	-	数字地。
V16	VSS	-	-	-	数字地。
V17	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
V18	VSS	-	-	-	数字地。
V19	VSS	-	-	-	数字地。
V20	VSS	-	-	-	数字地。
U3	VSS	-	-	-	数字地。
U13	VSS	-	-	-	数字地。
U14	VSS	-	-	-	数字地。
U15	VSS	-	-	-	数字地。
U16	VSS	-	-	-	数字地。
U17	VSS	-	-	-	数字地。
U18	VSS	-	-	-	数字地。
U19	VSS	-	-	-	数字地。
U20	VSS	-	-	-	数字地。
T13	VSS	-	-	-	数字地。
T14	VSS	-	-	-	数字地。
T15	VSS	-	-	-	数字地。
T16	VSS	-	-	-	数字地。
T17	VSS	-	-	-	数字地。
T18	VSS	-	-	-	数字地。
T19	VSS	-	-	-	数字地。
T20	VSS	-	-	-	数字地。
T31	VSS	-	-	-	数字地。
T32	VSS	-	-	-	数字地。
R1	VSS	-	-	-	数字地。
R5	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
R13	VSS	-	-	-	数字地。
R14	VSS	-	-	-	数字地。
R15	VSS	-	-	-	数字地。
R16	VSS	-	-	-	数字地。
R17	VSS	-	-	-	数字地。
R18	VSS	-	-	-	数字地。
R19	VSS	-	-	-	数字地。
R20	VSS	-	-	-	数字地。
P3	VSS	-	-	-	数字地。
P5	VSS	-	-	-	数字地。
P13	VSS	-	-	-	数字地。
P14	VSS	-	-	-	数字地。
P15	VSS	-	-	-	数字地。
P16	VSS	-	-	-	数字地。
P17	VSS	-	-	-	数字地。
P18	VSS	-	-	-	数字地。
P19	VSS	-	-	-	数字地。
P20	VSS	-	-	-	数字地。
P28	VSS	-	-	-	数字地。
P29	VSS	-	-	-	数字地。
P32	VSS	-	-	-	数字地。
N13	VSS	-	-	-	数字地。
N14	VSS	-	-	-	数字地。
N15	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
N16	VSS	-	-	-	数字地。
N17	VSS	-	-	-	数字地。
N18	VSS	-	-	-	数字地。
N19	VSS	-	-	-	数字地。
N20	VSS	-	-	-	数字地。
M12	VSS	-	-	-	数字地。
M21	VSS	-	-	-	数字地。
M30	VSS	-	-	-	数字地。
M31	VSS	-	-	-	数字地。
K28	VSS	-	-	-	数字地。
J3	VSS	-	-	-	数字地。
J9	VSS	-	-	-	数字地。
J24	VSS	-	-	-	数字地。
J28	VSS	-	-	-	数字地。
H1	VSS	-	-	-	数字地。
H2	VSS	-	-	-	数字地。
H30	VSS	-	-	-	数字地。
H31	VSS	-	-	-	数字地。
F3	VSS	-	-	-	数字地。
F5	VSS	-	-	-	数字地。
E1	VSS	-	-	-	数字地。
E5	VSS	-	-	-	数字地。
E6	VSS	-	-	-	数字地。
E9	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
E10	VSS	-	-	-	数字地。
E11	VSS	-	-	-	数字地。
E14	VSS	-	-	-	数字地。
E15	VSS	-	-	-	数字地。
E19	VSS	-	-	-	数字地。
E20	VSS	-	-	-	数字地。
E21	VSS	-	-	-	数字地。
E24	VSS	-	-	-	数字地。
E25	VSS	-	-	-	数字地。
E28	VSS	-	-	-	数字地。
D4	VSS	-	-	-	数字地。
D26	VSS	-	-	-	数字地。
C3	VSS	-	-	-	数字地。
C6	VSS	-	-	-	数字地。
C10	VSS	-	-	-	数字地。
C16	VSS	-	-	-	数字地。
C27	VSS	-	-	-	数字地。
C31	VSS	-	-	-	数字地。
C32	VSS	-	-	-	数字地。
B2	VSS	-	-	-	数字地。
B6	VSS	-	-	-	数字地。
B10	VSS	-	-	-	数字地。
B14	VSS	-	-	-	数字地。
B16	VSS	-	-	-	数字地。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
B20	VSS	-	-	-	数字地。
B25	VSS	-	-	-	数字地。
B29	VSS	-	-	-	数字地。
A1	VSS	-	-	-	数字地。
A14	VSS	-	-	-	数字地。
A20	VSS	-	-	-	数字地。
A25	VSS	-	-	-	数字地。
A29	VSS	-	-	-	数字地。
A32	VSS	-	-	-	数字地。

2.1.2 SIO 管脚

SIO 管脚如表 2-3 所示。

表2-3 SIO 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
P2	ACKOUT	I _{SPU/O}	4	-	I ² S 或 PCM 接口主时钟，可作为外部 audio codec 的工作时钟。
M5	SIO0DI	I _{PU}	-	-	SIO0 数据输入。如不使用，该管脚可悬空。
N2	SIO0DO	O _T	4	-	SIO0 数据输出。
N5	SIO0RCK	I _{SPU/O}	4	-	I ² S/PCM 接收位流时钟。
N3	SIO0RFS	I _{SPU/O}	4	-	I ² S 接收左右声道选择信号（与 ADC 接口），或 PCM 帧同步信号。
N4	SIO0XCK	I _{SPU/O}	4	-	I ² S/PCM 发送位流时钟。
M4	SIO0XFS	I _{SPU/O}	4	-	I ² S 发送左右声道选择信号（与 DAC 接口），或 PCM 帧同步信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
P4	SIO1DI	I _{PU}	-	-	SIO1 数据输入。如不使用, 该管脚可悬空。
R3	SIO1RCK	I _{SPU/O}	4	-	I ² S/PCM 接收位流时钟。
P1	SIO1RFS	I _{SPU/O}	4	-	I ² S 接收左右声道选择信号 (与 ADC 接口), 或 PCM 帧同步信号。
R4	SIO2DI	I _{SPU/O}	4	-	SIO2 数据输入。如不使用, 该管脚可悬空。
N1	SIO2RCK	I _{SPU/O}	4	-	I ² S/PCM 接收位流时钟。
R2	SIO2RFS	I _{SPU/O}	4	-	I ² S 接收左右声道选择信号 (与 ADC 接口), 或 PCM 帧同步信号。

2.1.3 VDAC 管脚

VDAC 管脚如表 2-4 所示。

表2-4 VDAC 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AG5	COMPDA C0	-	-	3.3	DAC0 的补偿管脚, 外接补偿电容。 说明: 建议该管脚和 DAC 的模拟 3.3V 电源之间并联 0.01μF 的陶瓷电容和 10uF 的钽电容。
AH4	COMPDA C1	-	-	3.3	DAC1 的补偿管脚, 外接补偿电容。 说明: 建议该管脚和 DAC 的模拟 3.3V 电源之间并联 0.01μF 的陶瓷电容和 10uF 的钽电容。
AJ1	DACVGA0 B	-	-	3.3	DAC0 的 blue 信号输出, 该信号为模拟信号。
AK1	DACVGA0 G	-	-	3.3	DAC0 的 green 信号或者一路标清的 CVBS 输出, 该信号为模拟信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AL1	DACVGA0R	-	-	3.3	DAC0 的 red 信号或者一路标清的 CVBS 输出, 该信号为模拟信号。
AM4	DACVGA1B	-	-	3.3	DAC1 的 Blue 信号输出, 该信号为模拟信号。
AM3	DACVGA1G	-	-	3.3	DAC1 的 green 信号, 该信号为模拟信号。
AM2	DACVGA1R	-	-	3.3	DAC1 的 red 信号, 该信号为模拟信号。
AG4	RSETDAC0	-	-	3.3	DAC0 的扩展电阻管脚, 外接扩展电阻。 ^a 说明: DAC0 的 full-scale 输出电流调节电阻管脚, 电阻接到地和该管脚之间。
AH5	RSETDAC1	-	-	3.3	DAC1 的扩展电阻管脚, 外接扩展电阻。 ^a 说明: DAC1 的 full-scale 输出电流调节电阻管脚, 电阻接到地和该管脚之间。
AC8	VREFINDAC0	-	-	3.3	DAC0 的参考电压输入。 说明: 建议该管脚和 DAC 的模拟地之间放置 0.1μF 的陶瓷电容。
AE10	VREFINDAC1	-	-	3.3	DAC1 的参考电压输入。 说明: 建议该管脚和 DAC 的模拟地之间放置 0.1μF 的陶瓷电容。

$$RSET (\Omega) = \frac{VREFIN (V) \times 40.3}{I_{OFS} (A)} (\Omega)$$

a: $VREFIN(V)$ 是指管脚 VREFDAC0 和 VREFDAC1 的电压值, 为 1.183V; I_{OFS} 是满负荷的电流值, 可依据设计的负载和此时的满负荷需求电压推算。例如: 单板设计负载为 37.5Ω, 满负荷的电压要求是 1.2V, 可算出电流为 32mA, 此时外置电阻为 $(1.183 \times 40.3) / 32mA = 1.49K \Omega$ 。

2.1.4 DDRA 管脚

DDRA 管脚如表 2-5 所示。



表2-5 DDRA 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
A10	DDRA_AD R0	-	-	1.8	DDR 端口 A 的地址信号。
A18	DDRA_AD R1	-	-	1.8	DDR 端口 A 的地址信号。
A12	DDRA_AD R2	-	-	1.8	DDR 端口 A 的地址信号。
B18	DDRA_AD R3	-	-	1.8	DDR 端口 A 的地址信号。
B11	DDRA_AD R4	-	-	1.8	DDR 端口 A 的地址信号。
D18	DDRA_AD R5	-	-	1.8	DDR 端口 A 的地址信号。
A11	DDRA_AD R6	-	-	1.8	DDR 端口 A 的地址信号。
C17	DDRA_AD R7	-	-	1.8	DDR 端口 A 的地址信号。
C13	DDRA_AD R8	-	-	1.8	DDR 端口 A 的地址信号。
D16	DDRA_AD R9	-	-	1.8	DDR 端口 A 的地址信号。
C18	DDRA_AD R10	-	-	1.8	DDR 端口 A 的地址信号。
D14	DDRA_AD R11	-	-	1.8	DDR 端口 A 的地址信号。
D17	DDRA_AD R12	-	-	1.8	DDR 端口 A 的地址信号。
A19	DDRA_AD R13	-	-	1.8	DDR 端口 A 的地址信号。
A16	DDRA_BA 0	-	-	1.8	DDR 端口 A 的 bank0 选择信号。
A17	DDRA_BA 1	-	-	1.8	DDR 端口 A 的 bank1 选择信号。
B17	DDRA_BA 2	-	-	1.8	DDR 端口 A 的 bank2 选择信号。
D13	DDRA_CASN	-	-	1.8	DDR 端口 A 的列地址选通信号，低电平有效。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
C15	DDRA_CKE	-	-	1.8	DDR 端口 A 的时钟使能信号，高电平有效。
B15	DDRA_CKN0	-	-	1.8	DDR 端口 A 的 0 组负向差分时钟。
B13	DDRA_CKN1	-	-	1.8	DDR 端口 A 的 1 组负向差分时钟。
A15	DDRA_CKP0	-	-	1.8	DDR 端口 A 的 0 组正向差分时钟。
A13	DDRA_CKP1	-	-	1.8	DDR 端口 A 的 1 组正向差分时钟。
D15	DDRA_CS_N	-	-	1.8	DDR 端口 A 输出到 DDR2 SDRAM 的片选信号，低电平有效。
C24	DDRA_DQ0	-	-	1.8	DDR 端口 A 的数据信号。
D24	DDRA_DQ1	-	-	1.8	DDR 端口 A 的数据信号。
C25	DDRA_DQ2	-	-	1.8	DDR 端口 A 的数据信号。
A26	DDRA_DQ3	-	-	1.8	DDR 端口 A 的数据信号。
B26	DDRA_DQ4	-	-	1.8	DDR 端口 A 的数据信号。
C26	DDRA_DQ5	-	-	1.8	DDR 端口 A 的数据信号。
D23	DDRA_DQ6	-	-	1.8	DDR 端口 A 的数据信号。
A23	DDRA_DQ7	-	-	1.8	DDR 端口 A 的数据信号。
D21	DDRA_DQ8	-	-	1.8	DDR 端口 A 的数据信号。
C20	DDRA_DQ9	-	-	1.8	DDR 端口 A 的数据信号。
B22	DDRA_DQ10	-	-	1.8	DDR 端口 A 的数据信号。
D20	DDRA_DQ11	-	-	1.8	DDR 端口 A 的数据信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
C21	DDRA_DQ 12	-	-	1.8	DDR 端口 A 的数据信号。
C22	DDRA_DQ 13	-	-	1.8	DDR 端口 A 的数据信号。
D19	DDRA_DQ 14	-	-	1.8	DDR 端口 A 的数据信号。
A22	DDRA_DQ 15	-	-	1.8	DDR 端口 A 的数据信号。
C7	DDRA_DQ 16	-	-	1.8	DDR 端口 A 的数据信号。
D8	DDRA_DQ 17	-	-	1.8	DDR 端口 A 的数据信号。
C8	DDRA_DQ 18	-	-	1.8	DDR 端口 A 的数据信号。
D10	DDRA_DQ 19	-	-	1.8	DDR 端口 A 的数据信号。
D9	DDRA_DQ 20	-	-	1.8	DDR 端口 A 的数据信号。
C9	DDRA_DQ 21	-	-	1.8	DDR 端口 A 的数据信号。
B7	DDRA_DQ 22	-	-	1.8	DDR 端口 A 的数据信号。
A7	DDRA_DQ 23	-	-	1.8	DDR 端口 A 的数据信号。
D6	DDRA_DQ 24	-	-	1.8	DDR 端口 A 的数据信号。
A3	DDRA_DQ 25	-	-	1.8	DDR 端口 A 的数据信号。
D7	DDRA_DQ 26	-	-	1.8	DDR 端口 A 的数据信号。
D5	DDRA_DQ 27	-	-	1.8	DDR 端口 A 的数据信号。
C4	DDRA_DQ 28	-	-	1.8	DDR 端口 A 的数据信号。
A6	DDRA_DQ 29	-	-	1.8	DDR 端口 A 的数据信号。
B3	DDRA_DQ 30	-	-	1.8	DDR 端口 A 的数据信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
C5	DDRA_DQ31	-	-	1.8	DDR 端口 A 的数据信号。
B24	DDRA_DQSN0	-	-	1.8	DDR 端口 A 的数据选通负向差分信号，对应数据总线 DDRDQ[7:0]。
B21	DDRA_DQSN1	-	-	1.8	DDR 端口 A 的数据选通负向差分信号，对应数据总线 DDRDQ[15:8]。
B8	DDRA_DQSN2	-	-	1.8	DDR 端口 A 的数据选通负向差分信号，对应数据总线 DDRDQ[23:16]。
B5	DDRA_DQSN3	-	-	1.8	DDR 端口 A 的数据选通负向差分信号，对应数据总线 DDRDQ[31:24]。
A24	DDRA_DQSP0	-	-	1.8	DDR 端口 A 的数据选通正向差分信号，对应数据总线 DDRDQ[7:0]。
A21	DDRA_DQSP1	-	-	1.8	DDR 端口 A 的数据选通正向差分信号，对应数据总线 DDRDQ[15:8]。
A8	DDRA_DQSP2	-	-	1.8	DDR 端口 A 的数据选通正向差分信号，对应数据总线 DDRDQ[23:16]。
A5	DDRA_DQSP3	-	-	1.8	DDR 端口 A 的数据选通正向差分信号，对应数据总线 DDRDQ[31:24]。
D25	DDRA_LDM0	-	-	1.8	DDR 端口 A 的字节屏蔽信号，对应数据总线 DDRDQ[7:0]。
D22	DDRA_UDM0	-	-	1.8	DDR 端口 A 的字节屏蔽信号，对应数据总线 DDRDQ[15:8]。
C11	DDRA_ODT	-	-	1.8	DDR 端口 A 输出的 ODT 信号。
D12	DDRA_RASN	-	-	1.8	DDR 端口 A 的行地址选通信号，低电平有效。
D11	DDRA_LDM1	-	-	1.8	DDR 端口 A 的字节屏蔽信号，对应数据总线 DDRDQ[23:16]。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
A2	DDRA_UD M1	-	-	1.8	DDR 端口 A 的字节屏蔽信号，对应数据总线 DDRDQ[31:24]。
C14	DDRA_WE N	-	-	1.8	DDR 端口 A 的写使能信号，低电平有效。

2.1.5 DDRB 管脚

DDR B 管脚如表 2-6 所示。

表2-6 DDR B 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
H28	DDR B_AD R0	-	-	1.8	DDR 端口 B 的地址信号。
N29	DDR B_AD R1	-	-	1.8	DDR 端口 B 的地址信号。
H32	DDR B_AD R2	-	-	1.8	DDR 端口 B 的地址信号。
M32	DDR B_AD R3	-	-	1.8	DDR 端口 B 的地址信号。
G30	DDR B_AD R4	-	-	1.8	DDR 端口 B 的地址信号。
N32	DDR B_AD R5	-	-	1.8	DDR 端口 B 的地址信号。
G31	DDR B_AD R6	-	-	1.8	DDR 端口 B 的地址信号。
L32	DDR B_AD R7	-	-	1.8	DDR 端口 B 的地址信号。
J29	DDR B_AD R8	-	-	1.8	DDR 端口 B 的地址信号。
L30	DDR B_AD R9	-	-	1.8	DDR 端口 B 的地址信号。
N30	DDR B_AD R10	-	-	1.8	DDR 端口 B 的地址信号。
K29	DDR B_AD R11	-	-	1.8	DDR 端口 B 的地址信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
N28	DDRB_AD R12	-	-	1.8	DDR 端口 B 的地址信号。
N31	DDRB_AD R13	-	-	1.8	DDR 端口 B 的地址信号。
L31	DDRB_BA 0	-	-	1.8	DDR 端口 B 的 bank0 选择信号。
L28	DDRB_BA 1	-	-	1.8	DDR 端口 B 的 bank1 选择信号。
M29	DDRB_BA 2	-	-	1.8	DDR 端口 B 的 bank2 选择信号。
J30	DDRB_CA SN	-	-	1.8	DDR 端口 B 的列地址选通信号，低电平有效。
L29	DDRB_CK E	-	-	1.8	DDR 端口 B 的时钟使能信号，高电平有效。
J32	DDRB_CK N0	-	-	1.8	DDR 端口 B 的 0 组负向差分时钟。
J31	DDRB_CK P0	-	-	1.8	DDR 端口 B 的 0 组正向差分时钟。
G32	DDRB_CS N	-	-	1.8	DDR 端口 B 输出到 DDR2 SDRAM 的片选信号，低电平有效。
E29	DDRB_DQ 0	-	-	1.8	DDR 端口 B 的数据信号。
F28	DDRB_DQ 1	-	-	1.8	DDR 端口 B 的数据信号。
F29	DDRB_DQ 2	-	-	1.8	DDR 端口 B 的数据信号。
E32	DDRB_DQ 3	-	-	1.8	DDR 端口 B 的数据信号。
F30	DDRB_DQ 4	-	-	1.8	DDR 端口 B 的数据信号。
F31	DDRB_DQ 5	-	-	1.8	DDR 端口 B 的数据信号。
D30	DDRB_DQ 6	-	-	1.8	DDR 端口 B 的数据信号。
D29	DDRB_DQ 7	-	-	1.8	DDR 端口 B 的数据信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
B31	DDRB_DQ 8	-	-	1.8	DDR 端口 B 的数据信号。
C29	DDRB_DQ 9	-	-	1.8	DDR 端口 B 的数据信号。
A31	DDRB_DQ 10	-	-	1.8	DDR 端口 B 的数据信号。
D28	DDRB_DQ 11	-	-	1.8	DDR 端口 B 的数据信号。
B28	DDRB_DQ 12	-	-	1.8	DDR 端口 B 的数据信号。
B32	DDRB_DQ 13	-	-	1.8	DDR 端口 B 的数据信号。
A28	DDRB_DQ 14	-	-	1.8	DDR 端口 B 的数据信号。
C28	DDRB_DQ 15	-	-	1.8	DDR 端口 B 的数据信号。
D32	DDRB_DQ SN0	-	-	1.8	DDR 端口 B 的数据选通负向差分信号，对应数据总线 DDRDQ[7:0]。
B30	DDRB_DQ SN1	-	-	1.8	DDR 端口 B 的数据选通负向差分信号，对应数据总线 DDRDQ[15:8]。
D31	DDRB_DQ SP0	-	-	1.8	DDR 端口 B 的数据选通正向差分信号，对应数据总线 DDRDQ[7:0]。
A30	DDRB_DQ SP1	-	-	1.8	DDR 端口 B 的数据选通正向差分信号，对应数据总线 DDRDQ[15:8]。
G29	DDRB_LD M0	-	-	1.8	DDR 端口 B 的字节屏蔽信号，对应数据总线 DDRDQ[7:0]。
F32	DDRB_OD T	-	-	1.8	DDR 端口 B 输出的 ODT 信号。
H29	DDRB_RA SN	-	-	1.8	DDR 端口 B 的行地址选通信号，低电平有效。
D27	DDRB_UD M0	-	-	1.8	DDR 端口 B 的字节屏蔽信号，对应数据总线 DDRDQ[15:8]。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
K32	DDRB_WEN	-	-	1.8	DDR 端口 B 的写使能信号，低电平有效。

2.1.6 EBI 管脚

EBI 管脚如表 2-7 所示。

表2-7 EBI 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AK29	EBIADR0	O	8	-	EBI 地址总线 ADR0。
AM30	EBIADR1	O	8	-	EBI 地址总线 ADR1。
AM31	EBIADR2	O	8	-	EBI 地址总线 ADR2。
AL31	EBIADR3	O	8	-	EBI 地址总线 ADR3。
AJ29	EBIADR4	O	8	-	EBI 地址总线 ADR4。
AK32	EBIADR5	O	8	-	EBI 地址总线 ADR5。
AL32	EBIADR6	O	8	-	EBI 地址总线 ADR6。
AK31	EBIADR7	O	8	-	EBI 地址总线 ADR7。
AJ31	EBIADR8	O	8	-	EBI 地址总线 ADR8。
AJ30	EBIADR9	O	8	-	EBI 地址总线 ADR9。
AJ32	EBIADR10	O	8	-	EBI 地址总线 ADR10。
AH29	EBIADR11	O	8	-	EBI 地址总线 ADR11。
AH30	EBIADR12	O	8	-	EBI 地址总线 ADR12。
AH31	EBIADR13	O	8	-	EBI 地址总线 ADR13。
AG29	EBIADR14	O	8	-	EBI 地址总线 ADR14。
AG31	EBIADR15	I/O	8	-	EBI 地址总线 ADR15。
AG30	EBIADR16	I/O	8	-	EBI 地址总线 ADR16。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AG32	EBIADR17	I/O	8	-	EBI 地址总线 ADR17。
AF29	EBIADR18	I/O	8	-	EBI 地址总线 ADR18。
AF32	EBIADR19	I/O	8	-	EBI 地址总线 ADR19。
AF31	EBIADR20	I/O	8	-	EBI 地址总线 ADR20。
AE29	EBIADR21	I/O	8	-	EBI 地址总线 ADR21。
AE31	EBIADR22	I/O	8	-	EBI 地址总线 ADR22。
AE30	EBIADR23	I/O	8	-	EBI 地址总线 ADR23。
AE32	EBIADR24	I/O	8	-	EBI 地址总线 ADR24。
AL27	EBIDQ0	I/O	8	-	EBI 数据总线 DAT0。
AL28	EBIDQ1	I/O	8	-	EBI 数据总线 DAT1。
AJ27	EBIDQ2	I/O	8	-	EBI 数据总线 DAT2。
AK28	EBIDQ3	I/O	8	-	EBI 数据总线 DAT3。
AJ28	EBIDQ4	I/O	8	-	EBI 数据总线 DAT4。
AL29	EBIDQ5	I/O	8	-	EBI 数据总线 DAT5。
AM29	EBIDQ6	I/O	8	-	EBI 数据总线 DAT6。
AL30	EBIDQ7	I/O	8	-	EBI 数据总线 DAT7。
AL24	EBIWEN	O	8	-	EBI 写信号，低电平有效。
AM26	NFALE	O	4	-	Nand Flash 地址锁存信号。
AM25	NFCLE	O	4	-	Nand Flash 命令锁存信号。
AK26	NFCS0N	O	4	-	Nand Flash 片选信号 0，低电平有效。系统支持从该片选启动。
AM27	NFCS1N	I/O	4	-	Nand Flash 片选信号 1，低电平有效。
AK25	NFOEN	O	8	-	Nand Flash 的读使能信号，低电平有效。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AJ26	NFRB	I _{PU} /O	4	-	Nand Flash 的空闲指示信号。 0: 忙。 1: 空闲; 外接两片 Nand Flash 时, 需要将两片 Flash 器件的输出经过线与后连接到本管脚。
AJ25	SMICS0N	O	4	-	SMI 片选信号 0, 可配置成低电平有效或高电平有效, 默认为低电平有效。系统支持从该片选启动。
AL25	SMICS1N	I/O	4	-	SMI 片选信号 1, 可配置成低电平有效或高电平有效, 默认为低电平有效。
AJ24	SMIOEN	O	8	-	SMI 接口的读使能信号, 低电平有效。
AL26	EBIRDYN	I _{SPU}	-	-	SMI 接口的输入 Ready 指示信号, 低电平有效。如不使用, 该管脚可悬空。

2.1.7 GMAC 管脚

GMAC 管脚如表 2-8 所示。

表2-8 GMAC 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
D1	GCOL	I _{PU} /O	4	-	GMAC 碰撞指示信号。
G3	GCRS	I _{PU} /O	4	-	GMAC 载波侦听信号。
F1	GRCLK	I _{PU}	-	-	GMAC 接收侧时钟输入。
H4	GRXD0	I _{PU}	-	-	GMAC 接收数据 DAT0。
F2	GRXD1	I _{PU}	-	-	GMAC 接收数据 DAT1。
H3	GRXD2	I _{PU}	-	-	GMAC 接收数据 DAT2。
J4	GRXD3	I _{PU}	-	-	GMAC 接收数据 DAT3。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
E3	GRXDV	I _{PU}	-	-	GMAC 接收数据有效。
F4	GTCLK	I _{PU}	-	-	GMAC 发送侧时钟输入。
E2	GTCLKOUT	I _{PU} /O	8	-	GMAC 发送侧时钟输出。
D3	GTXD0	O	8	-	GAMC 发送数据 DAT0。
E4	GTXD1	O	8	-	GAMC 发送数据 DAT1。
C1	GTXD2	O	8	-	GAMC 发送数据 DAT2。
D2	GTXD3	O	8	-	GAMC 发送数据 DAT3。
G4	GTXEN	O	8	-	GMAC 发送使能信号。
C2	MDCK	O	4	-	MDIO 接口时钟输出。
B1	MDIO	I _{PU} /O	4	-	MDIO 接口的输入/输出信号。

2.1.8 VO 管脚

VO 管脚如表 2-9 所示。

表2-9 VO 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
Y29	GPIO3_0	I/O	8	-	通用 GPIO。 说明：该管脚复用为 VOBDAT7，表示为 VO RGB 输出的 blue 数据 DAT7。
W32	VO0CK	I/O	8	-	标清图像输出时钟。
W30	VO0DAT0	I/O	8	-	标清图像输出数据 DAT0。
W29	VO0DAT1	I/O	8	-	标清图像输出数据 DAT1。
V32	VO0DAT2	I/O	8	-	标清图像输出数据 DAT2。
V31	VO0DAT3	I/O	8	-	标清图像输出数据 DAT3。
V30	VO0DAT4	I/O	8	-	标清图像输出数据 DAT4。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
V29	VO0DAT5	I/O	8	-	标清图像输出数据 DAT5。
U29	VO0DAT6	I/O	8	-	标清图像输出数据 DAT6。
U30	VO0DAT7	I/O	8	-	标清图像输出数据 DAT7。
AA31	VO1CK	I/O	8	-	高清图像或者级联输出时钟。
AD29	VO1DAT0	O _T	8	-	高清图像或者级联输出 DAT0。
AD31	VO1DAT1	O _T	8	-	高清图像或者级联输出 DAT1。
AD30	VO1DAT2	I _{SPU/O}	8	-	高清图像或者级联输出 DAT2。
AD32	VO1DAT3	I _{SPU/O}	8	-	高清图像或者级联输出 DAT3。
AC29	VO1DAT4	I _{SPU/O}	8	-	高清图像或者级联输出 DAT4。
AC31	VO1DAT5	I _{SPU/O}	8	-	高清图像或者级联输出 DAT5。
AB29	VO1DAT6	I _{SPU/O}	8	-	高清图像或者级联输出 DAT6。
AC30	VO1DAT7	I/O	8	-	高清图像或者级联输出 DAT7。
AB31	VO1DAT8	I/O	8	-	高清图像或者级联输出 DAT8。
AB30	VO1DAT9	I _{PU/O}	8	-	高清图像或者级联输出 DAT9。
AB32	VO1DAT10	O _T	8	-	高清图像或者级联输出 DAT10。
AA32	VO1DAT11	I _{PU/O}	8	-	高清图像或者级联输出 DAT11。
AA29	VO1DAT12	I _{PU/O}	8	-	高清图像或者级联输出 DAT12。
Y31	VO1DAT13	I/O	8	-	高清图像或者级联输出 DAT13。
Y32	VO1DAT14	I/O	8	-	高清图像或者级联输出 DAT14。
Y30	VO1DAT15	I/O	8	-	高清图像或者级联输出 DAT15。
W31	VORGBDV	I/O	8	-	VO RGB 输出的数据有效指示。



2.1.9 PCI 管脚

PCI 管脚如表 2-10 所示。

表2-10 PCI 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AL21	PCIAD0	I/O	-	-	PCI 地址数据复用总线 0。
AM22	PCIAD1	I/O	-	-	PCI 地址数据复用总线 1。
AJ20	PCIAD2	I/O	-	-	PCI 地址数据复用总线 2。
AM21	PCIAD3	I/O	-	-	PCI 地址数据复用总线 3。
AM20	PCIAD4	I/O	-	-	PCI 地址数据复用总线 4。
AL20	PCIAD5	I/O	-	-	PCI 地址数据复用总线 5。
AJ19	PCIAD6	I/O	-	-	PCI 地址数据复用总线 6。
AK19	PCIAD7	I/O	-	-	PCI 地址数据复用总线 7。
AJ17	PCIAD8	I/O	-	-	PCI 地址数据复用总线 8。
AK17	PCIAD9	I/O	-	-	PCI 地址数据复用总线 9。
AL17	PCIAD10	I/O	-	-	PCI 地址数据复用总线 10。
AM17	PCIAD11	I/O	-	-	PCI 地址数据复用总线 11。
AJ16	PCIAD12	I/O	-	-	PCI 地址数据复用总线 12。
AL16	PCIAD13	I/O	-	-	PCI 地址数据复用总线 13。
AM16	PCIAD14	I/O	-	-	PCI 地址数据复用总线 14。
AJ15	PCIAD15	I/O	-	-	PCI 地址数据复用总线 15。
AM14	PCIAD16	I/O	-	-	PCI 地址数据复用总线 16。
AL13	PCIAD17	I/O	-	-	PCI 地址数据复用总线 17。
AK13	PCIAD18	I/O	-	-	PCI 地址数据复用总线 18。
AJ13	PCIAD19	I/O	-	-	PCI 地址数据复用总线 19。
AM12	PCIAD20	I/O	-	-	PCI 地址数据复用总线 20。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AL12	PCIAD21	I/O	-	-	PCI 地址数据复用总线 21。
AJ12	PCIAD22	I/O	-	-	PCI 地址数据复用总线 22。
AM11	PCIAD23	I/O	-	-	PCI 地址数据复用总线 23。
AJ11	PCIAD24	I/O	-	-	PCI 地址数据复用总线 24。
AM10	PCIAD25	I/O	-	-	PCI 地址数据复用总线 25。
AL10	PCIAD26	I/O	-	-	PCI 地址数据复用总线 26。
AK10	PCIAD27	I/O	-	-	PCI 地址数据复用总线 27。
AJ10	PCIAD28	I/O	-	-	PCI 地址数据复用总线 28。
AM9	PCIAD29	I/O	-	-	PCI 地址数据复用总线 29。
AL9	PCIAD30	I/O	-	-	PCI 地址数据复用总线 30。
AK9	PCIAD31	I/O	-	-	PCI 地址数据复用总线 31。
AK11	PCICBE0	I/O	-	-	PCI 字节使能和命令复用总线 0。
AL11	PCICBE1	I/O	-	-	PCI 字节使能和命令复用总线 1。
AM15	PCICBE2	I/O	-	-	PCI 字节使能和命令复用总线 2。
AL18	PCICBE3	I/O	-	-	PCI 字节使能和命令复用总线 3。
AM7	PCICLK	I/O	-	-	PCI 工作时钟。
AK15	PCIDEVSE LN	I/O	-	-	PCI 总线 DEVSEL 信号，低电平有效。
AJ9	PCIFRAM EN	I/O	-	-	PCI 总线 FRAME 信号线。
AK21	PCIGRAN T0N	I/O	-	-	PCI 总线仲裁信号 0，低电平有效。
AJ21	PCIGRAN T1N	I/O	-	-	PCI 总线仲裁信号 1，低电平有效。
AK22	PCIGRAN T2N	I/O	-	-	PCI 总线仲裁信号 2，低电平有效。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AL22	PCIGRAN T3N	I/O	-	-	PCI 总线仲裁信号 3, 低电平有效。
AJ22	PCIGRAN T4N	I/O	-	-	PCI 总线仲裁信号 4, 低电平有效。
AL14	PCIIDSEL	I _S /O	-	-	PCI 总线 IDSEL 信号。
AL8	PCIINTAN	I _S /O	-	-	PCI 中断 INTA 信号, 低电平有效。
AJ14	PCIIRDYN	I/O	-	-	PCI 总线 IRDY 信号, 低电平有效。
AM19	PCIPAR	I/O	-	-	PCI 总线 PAR 信号。
AK18	PCIPERRN	I/O	-	-	PCI 总线 PERR 信号, 低电平有效。如不使用, 该管脚需外接上拉电阻, 电阻阻值需大于 4.7k Ω 。
AL19	PCIREQ0N	I _S /O	-	-	PCI 总线申请信号 0, 低电平有效。
AL23	PCIREQ1N	I _S /O	-	-	PCI 总线申请信号 1, 低电平有效。
AK23	PCIREQ2N	I _S /O	-	-	PCI 总线申请信号 2, 低电平有效。
AJ23	PCIREQ3N	I _S /O	-	-	PCI 总线申请信号 3, 低电平有效。
AM24	PCIREQ4N	I _S /O	-	-	PCI 总线申请信号 4, 低电平有效。
AJ8	PCIRSTN	I _S /O	-	-	PCI 总线复位。
AJ18	PCISERRN	I/O	-	-	PCI 总线 SERR 信号, 低电平有效。如不使用, 该管脚需外接上拉电阻, 电阻阻值需大于 4.7k Ω 。
AL15	PCISTOPN	I/O	-	-	PCI 总线 STOP 信号, 低电平有效。
AK14	PCITRDY N	I/O	-	-	PCI 总线 TRDY 信号, 低电平有效。



2.1.10 SYS 管脚

SYS 管脚如表 2-11 所示。

表2-11 SYS 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AJ6	RSTN	I _{SPU}	-	-	系统上电复位信号输入，低电平有效。
T30	TESTMODE	I _{SPD}	-	-	功能模式和 ARM debug 模式选择。 0: 正常工作模式，ARM 可进入 debug 模式； 1: 测试模式。
AK7	WDGRST	O _T	4	-	看门狗复位输出，低电平有效，OD 输出。
U31	XIN_24	-	-	-	24MHz 晶振时钟输入或钟振时钟输入。
U32	XOUT_24	-	-	-	24MHz 晶振时钟输出。如果接钟振，该管脚悬空。

2.1.11 I2C 管脚

I²C 管脚如表 2-12 所示。

表2-12 I²C 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
G1	SCL	I _{SPU} /O	4	-	I ² C 总线时钟，OD 输出。PCB 上需要外接上拉电阻。
G2	SDA	I _{SPU} /O	4	-	I ² C 总线数据/地址，OD 输出。PCB 上需要外接上拉电阻。

2.1.12 JTAG 管脚

JTAG 管脚如表 2-13 所示。



表2-13 JTAG 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
R31	TCK	I _{SPD}	-	-	JTAG 时钟输入。
R29	TDI	I _{SPU}	-	-	JTAG 数据输入。
R30	TDO	O _T	4	-	JTAG 数据输出。
R32	TMS	I _{SPU}	-	-	JTAG 模式选择输入。
T29	TRSTN	I _{SPD}	-	-	JTAG 复位输入，低电平有效。

2.1.13 UART 管脚

UART 管脚如表 2-14 所示。

表2-14 UART 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AM6	UCTSN1	I _{PU}	-	-	UART1 的清除发送信号。
AL6	URTSN1	O _T	4	-	UART1 的请求发送信号。如不使用，该管脚可悬空。
AK5	URXD0	I _{PU}	-	-	UART0 数据接收。
AJ7	URXD1	I _{PU}	-	-	UART1 数据接收。
AK6	UTXD0	O _T	4	-	UART0 数据发送。
AL7	UTXD1	O _T	4	-	UART1 数据发送。

2.1.14 USB 管脚

USB 管脚如表 2-15 所示。

表2-15 USB 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
J2	USBDM0	I/O	-	3.3	USB 2.0 HOST 端口 0 D-差分数据总线，该信号为模拟信号。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
L2	USBDM1	I/O	-	3.3	USB 2.0 HOST 端口 1 D-差分数据总线, 该信号为模拟信号。
J1	USBDP0	I/O	-	3.3	USB 2.0 HOST 端口 0 D+差分数据总线, 该信号为模拟信号。
L1	USBDP1	I/O	-	3.3	USB 2.0 HOST 端口 1 D+差分数据总线, 该信号为模拟信号。
K3	USBREXT	I/O	-	3.3	USB 2.0 HOST 外置电阻连接端。

2.1.15 VI 管脚

VI 管脚如表 2-16 所示。

表2-16 VI 管脚

Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AC2	VI0CK	I _{PU}	-	-	VI0 接口的时钟。
AB2	VI0DAT0	I _{PU}	-	-	VI0 接口的数据 DAT0。
AB3	VI0DAT1	I _{PU}	-	-	VI0 接口的数据 DAT1。
AD1	VI0DAT2	I _{PU}	-	-	VI0 接口的数据 DAT2。
AB4	VI0DAT3	I _{PU}	-	-	VI0 接口的数据 DAT3。
AC4	VI0DAT4	I _{PU}	-	-	VI0 接口的数据 DAT4。
AC3	VI0DAT5	I _{PU}	-	-	VI0 接口的数据 DAT5。
AD2	VI0DAT6	I _{PU}	-	-	VI0 接口的数据 DAT6。
AD3	VI0DAT7	I _{PU}	-	-	VI0 接口的数据 DAT7。
AA4	VI0HS	I _{PU} /O	4	-	VI0 接口的水平同步信号。如不使用, 该管脚可悬空。
AB1	VI0VS	I _{PU} /O	4	-	VI0 接口的垂直同步信号。如不使用, 该管脚可悬空。
AE1	VI1CK	I _{PU}	-	-	VI1 接口的时钟。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AE2	VI1DAT0	I _{PU} /O	4	-	VI1 接口的数据 DAT0。
AF2	VI1DAT1	I _{PU} /O	4	-	VI1 接口的数据 DAT1。
AE4	VI1DAT2	I _{PU} /O	4	-	VI1 接口的数据 DAT2。
AD4	VI1DAT3	I _{PU} /O	4	-	VI1 接口的数据 DAT3。
AF3	VI1DAT4	I _{PU} /O	4	-	VI1 接口的数据 DAT4。
AG1	VI1DAT5	I _{PU} /O	4	-	VI1 接口的数据 DAT5。
AF4	VI1DAT6	I _{PU} /O	4	-	VI1 接口的数据 DAT6。
AG2	VI1DAT7	I _{PU} /O	4	-	VI1 接口的数据 DAT7。
T1	VI2CK	I _{PU}	-	-	VI2 接口的时钟。
T2	VI2DAT0	I _{PU} /O	4	-	VI2 接口的数据 DAT0。
U4	VI2DAT1	I _{PU} /O	4	-	VI2 接口的数据 DAT1。
U2	VI2DAT2	I _{PU} /O	4	-	VI2 接口的数据 DAT2。
U1	VI2DAT3	I _{PU} /O	4	-	VI2 接口的数据 DAT3。
V3	VI2DAT4	I _{PU} /O	4	-	VI2 接口的数据 DAT4。
V2	VI2DAT5	I _{PU} /O	4	-	VI2 接口的数据 DAT5。
V1	VI2DAT6	I _{PU} /O	4	-	VI2 接口的数据 DAT6。
V4	VI2DAT7	I _{PU} /O	4	-	VI2 接口的数据 DAT7。
T3	VI2HS	I _{PU} /O	4	-	VI2 接口的水平同步信号。如不使用，该管脚可悬空。
T4	VI2VS	I _{PU} /O	4	-	VI2 接口的垂直同步信号。如不使用，该管脚可悬空。
W2	VI3CK	I _{PU}	-	-	VI3 接口的时钟。
W3	VI3DAT0	I _{PU} /O	4	-	VI3 接口的数据 DAT0。
Y1	VI3DAT1	I _{PU} /O	4	-	VI3 接口的数据 DAT1。
W4	VI3DAT2	I _{PU} /O	4	-	VI3 接口的数据 DAT2。



Pin	管脚名称	类型	驱动 (mA)	电压 (V)	描述
AA1	VI3DAT3	I _{PU} /O	4	-	VI3 接口的数据 DAT3。
Y2	VI3DAT4	I _{PU} /O	4	-	VI3 接口的数据 DAT4。
Y3	VI3DAT5	I _{PU} /O	4	-	VI3 接口的数据 DAT5。
Y4	VI3DAT6	I _{PU} /O	4	-	VI3 接口的数据 DAT6。
AA2	VI3DAT7	I _{PU} /O	4	-	VI3 接口的数据 DAT7。

2.2 软件复用管脚描述

软件复用指 CPU 可以通过配置寄存器控制管脚的复用功能。

2.2.1 VI 管脚复用

VI 的软件复用管脚如表 2-17 所示。

表2-17 VI 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2
AA4	VI0HS	reg0	URXD2	GPIO1_3
AB1	VI0VS	reg1	UTXD2	GPIO1_4
AB2	VI0DAT0	reg2	VOUDAT8	-
AB3	VI0DAT1	reg3	VOUDAT9	-
AD1	VI0DAT2	reg4	VOUDAT10	-
AB4	VI0DAT3	reg5	VOUDAT11	-
AC4	VI0DAT4	reg6	VOUDAT12	-
AC3	VI0DAT5	reg7	VOUDAT13	-
AD2	VI0DAT6	reg8	VOUDAT14	-
AD3	VI0DAT7	reg9	VOUDAT15	-
AE2	VI1DAT0	reg10	GPIO4_0	VOUDAT0



Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2
AF2	VI1DAT1	reg11	GPIO4_1	VOUDAT1
AE4	VI1DAT2	reg12	GPIO4_2	VOUDAT2
AD4	VI1DAT3	reg13	GPIO4_3	VOUDAT3
AF3	VI1DAT4	reg14	GPIO4_4	VOUDAT4
AG1	VI1DAT5	reg15	GPIO4_5	VOUDAT5
AF4	VI1DAT6	reg16	GPIO4_6	VOUDAT6
AG2	VI1DAT7	reg17	GPIO4_7	VOUDAT7
T3	VI2HS	reg18	URXD3	GPIO3_4
T4	VI2VS	reg19	UTXD3	GPIO3_5
T2	VI2DAT0	reg20	GPIO5_0	-
U4	VI2DAT1	reg21	GPIO5_1	-
U2	VI2DAT2	reg22	GPIO5_2	-
U1	VI2DAT3	reg23	GPIO5_3	-
V3	VI2DAT4	reg24	GPIO5_4	-
V2	VI2DAT5	reg25	GPIO5_5	-
V1	VI2DAT6	reg26	GPIO5_6	-
V4	VI2DAT7	reg27	GPIO5_7	-
W3	VI3DAT0	reg28	GPIO6_0	-
Y1	VI3DAT1	reg29	GPIO6_1	-
W4	VI3DAT2	reg30	GPIO6_2	-
AA1	VI3DAT3	reg31	GPIO6_3	-
Y2	VI3DAT4	reg32	GPIO6_4	-
Y3	VI3DAT5	reg33	GPIO6_5	-
Y4	VI3DAT6	reg34	GPIO6_6	-



Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2
AA2	VI3DAT7	reg35	GPIO6_7	-

VI 的软件复用信号如表 2-18 所示。

表2-18 VI 的软件复用信号描述

信号名	方向	说明
GPIO1_3	I/O	通用 GPIO。
GPIO1_4	I/O	通用 GPIO。
GPIO3_4	I/O	通用 GPIO。
GPIO3_5	I/O	通用 GPIO。
GPIO4_0	I/O	通用 GPIO。
GPIO4_1	I/O	通用 GPIO。
GPIO4_2	I/O	通用 GPIO。
GPIO4_3	I/O	通用 GPIO。
GPIO4_4	I/O	通用 GPIO。
GPIO4_5	I/O	通用 GPIO。
GPIO4_6	I/O	通用 GPIO。
GPIO4_7	I/O	通用 GPIO。
GPIO5_0	I/O	通用 GPIO。
GPIO5_1	I/O	通用 GPIO。
GPIO5_2	I/O	通用 GPIO。
GPIO5_3	I/O	通用 GPIO。
GPIO5_4	I/O	通用 GPIO。
GPIO5_5	I/O	通用 GPIO。
GPIO5_6	I/O	通用 GPIO。
GPIO5_7	I/O	通用 GPIO。
GPIO6_0	I/O	通用 GPIO。
GPIO6_1	I/O	通用 GPIO。



信号名	方向	说明
GPIO6_2	I/O	通用 GPIO。
GPIO6_3	I/O	通用 GPIO。
GPIO6_4	I/O	通用 GPIO。
GPIO6_5	I/O	通用 GPIO。
GPIO6_6	I/O	通用 GPIO。
GPIO6_7	I/O	通用 GPIO。
URXD2	I	UART2 数据接收。
URXD3	I	UART3 数据接收。
UTXD2	O	UART2 数据发送。
UTXD3	O	UART3 数据发送。
VOUDAT0	I	级联时输入的级联图像数据 0。
VOUDAT1	I	级联时输入的级联图像数据 1。
VOUDAT10	I	级联时输入的级联图像数据 10。
VOUDAT11	I	级联时输入的级联图像数据 11。
VOUDAT12	I	级联时输入的级联图像数据 12。
VOUDAT13	I	级联时输入的级联图像数据 13。
VOUDAT14	I	级联时输入的级联图像数据 14。
VOUDAT15	I	级联时输入的级联图像数据 15。
VOUDAT2	I	级联时输入的级联图像数据 2。
VOUDAT3	I	级联时输入的级联图像数据 3。
VOUDAT4	I	级联时输入的级联图像数据 4。
VOUDAT5	I	级联时输入的级联图像数据 5。
VOUDAT6	I	级联时输入的级联图像数据 6。
VOUDAT7	I	级联时输入的级联图像数据 7。
VOUDAT8	I	级联时输入的级联图像数据 8。
VOUDAT9	I	级联时输入的级联图像数据 9。



2.2.2 VO 管脚复用

VO 的软件复用管脚如表 2-19 所示。

表2-19 VO 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2	复用信号 3
W32	VO0CK	reg36	GPIO1_5	VORDAT0	-
W30	VO0DAT0	reg37	GPIO2_0	VORDAT1	-
W29	VO0DAT1	reg38	GPIO2_1	VORDAT2	-
V32	VO0DAT2	reg39	GPIO2_2	VORDAT3	-
V31	VO0DAT3	reg40	GPIO2_3	VORDAT4	-
V30	VO0DAT4	reg41	GPIO2_4	VORDAT5	-
V29	VO0DAT5	reg42	GPIO2_5	VORDAT6	-
U29	VO0DAT6	reg43	GPIO2_6	VORDAT7	-
U30	VO0DAT7	reg44	GPIO2_7	VOGDAT0	-
AA31	VO1CK	reg45	GPIO7_2	SDIOCK	-
AD29	VO1DAT0	reg46	VGA1HS	-	-
AD31	VO1DAT1	reg47	VGA1VS	-	-
AD30	VO1DAT2	reg48	SDIOCMD	VOGDAT1	-
AD32	VO1DAT3	reg49	SDIODAT0	VOGDAT2	-
AC29	VO1DAT4	reg50	SDIODAT1	VOGDAT3	-
AC31	VO1DAT5	reg51	SDIODAT2	VOGDAT4	-
AB29	VO1DAT6	reg52	SDIODAT3	VOGDAT5	-
AC30	VO1DAT7	reg53	GPIO1_6	VOGDAT6	-
AB31	VO1DAT8	reg54	SPICK	VOGDAT7	-
AB30	VO1DAT9	reg55	SPIDI	VOBDAT0	-
AB32	VO1DAT10	reg56	SPIDO	VOBDAT1	-



Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2	复用信号 3
AA32	VO1DAT11	reg57	SPICSN0	VOBDAT2	-
AA29	VO1DAT12	reg58	SPICSN1	VOBDAT3	-
Y31	VO1DAT13	reg59	GPIO7_4	VOBDAT4	VGA0HS
Y32	VO1DAT14	reg60	GPIO7_5	VOBDAT5	VGA0VS
Y30	VO1DAT15	reg61	GPIO1_7	VOBDAT6	-
Y29	GPIO3_0	reg62	VOBDAT7	-	-
W31	VORGBDV	reg82	GPIO7_3	-	-

VO 的软件复用信号描述如表 2-20 所示。

表2-20 VO 的软件复用信号描述

信号名	方向	说明
GPIO1_5	I/O	通用 GPIO。
GPIO1_6	I/O	通用 GPIO。
GPIO1_7	I/O	通用 GPIO。
GPIO2_0	I/O	通用 GPIO。
GPIO2_1	I/O	通用 GPIO。
GPIO2_2	I/O	通用 GPIO。
GPIO2_3	I/O	通用 GPIO。
GPIO2_4	I/O	通用 GPIO。
GPIO2_5	I/O	通用 GPIO。
GPIO2_6	I/O	通用 GPIO。
GPIO2_7	I/O	通用 GPIO。
GPIO7_2	I/O	通用 GPIO。
GPIO7_3	I/O	通用 GPIO。
GPIO7_4	I/O	通用 GPIO。
GPIO7_5	I/O	通用 GPIO。



信号名	方向	说明
SDIOCK	O	SDIO/MMC 时钟。
SDIOCMD	I/O	SDIO/MMC 命令。
SDIODAT0	I/O	SDIO/MMC 数据 DAT0。
SDIODAT1	I/O	SDIO/MMC 数据 DAT1。
SDIODAT2	I/O	SDIO/MMC 数据 DAT2。
SDIODAT3	I/O	SDIO/MMC 数据 DAT3。
SPICK	I/O	SPI 的时钟信号。
SPICSN0	I/O	SPI 的片选信号 0，低电平有效。
SPICSN1	I/O	SPI 的片选信号 1，低电平有效。
SPIDI	I	SPI 输入数据。如不使用，该管脚可悬空。
SPIDO	O	SPI 的输出数据。
VGA0HS	O	VGA0 的水平同步信号。
VGA0VS	O	VGA0 的垂直同步信号。
VGA1HS	O	VGA1 的水平同步信号。
VGA1VS	O	VGA1 的垂直同步信号。
VOBDAT0	O	VO RGB 输出的 blue 数据 DAT0。
VOBDAT1	O	VO RGB 输出的 blue 数据 DAT1。
VOBDAT2	O	VO RGB 输出的 blue 数据 DAT2。
VOBDAT3	O	VO RGB 输出的 blue 数据 DAT3。
VOBDAT4	O	VO RGB 输出的 blue 数据 DAT4。
VOBDAT5	O	VO RGB 输出的 blue 数据 DAT5。
VOBDAT6	O	VO RGB 输出的 blue 数据 DAT6。
VOBDAT7	O	VO RGB 输出的 blue 数据 DAT7。
VOGDAT0	O	VO RGB 输出的 green 数据 DAT0。
VOGDAT1	O	VO RGB 输出的 green 数据 DAT1。
VOGDAT2	O	VO RGB 输出的 green 数据 DAT2。
VOGDAT3	O	VO RGB 输出的 green 数据 DAT3。
VOGDAT4	O	VO RGB 输出的 green 数据 DAT4。
VOGDAT5	O	VO RGB 输出的 green 数据 DAT5。



信号名	方向	说明
VOGDAT6	O	VO RGB 输出的 green 数据 DAT6。
VOGDAT7	O	VO RGB 输出的 green 数据 DAT7。
VORDAT0	O	VO RGB 输出的 red 数据 DAT0。
VORDAT1	O	VO RGB 输出的 red 数据 DAT1。
VORDAT2	O	VO RGB 输出的 red 数据 DAT2。
VORDAT3	O	VO RGB 输出的 red 数据 DAT3。
VORDAT4	O	VO RGB 输出的 red 数据 DAT4。
VORDAT5	O	VO RGB 输出的 red 数据 DAT5。
VORDAT6	O	VO RGB 输出的 red 数据 DAT6。
VORDAT7	O	VO RGB 输出的 red 数据 DAT7。

2.2.3 I2C 管脚复用

I²C 的软件复用管脚如表 2-21 所示。

表2-21 I²C 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1
G2	SDA	reg63	GPIO0_0
G1	SCL	reg64	GPIO0_1

I²C 的软件复用信号描述如表 2-22 所示。

表2-22 I²C 的软件复用信号描述

信号名	方向	说明
GPIO0_0	I/O	通用 GPIO。
GPIO0_1	I/O	通用 GPIO。

2.2.4 SIO 管脚复用

SIO 的软件复用管脚如表 2-23 所示。



表2-23 SIO 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1	复用信号 2
M4	SIO0XFS	reg65	GPIO0_2	-
N4	SIO0XCK	reg66	GPIO0_3	-
P2	ACKOUT	reg67	GPIO0_4	-
R4	SIO2DI	reg68	GPIO0_5	SIO1DO
R2	SIO2RFS	reg69	GPIO0_6	SIO1XFS
N1	SIO2RCK	reg70	GPIO0_7	SIO1XCK

SIO 的软件复用信号描述如表 2-24 所示。

表2-24 SIO 的软件复用信号描述

信号名	方向	说明
GPIO0_2	I/O	通用 GPIO。
GPIO0_3	I/O	通用 GPIO。
GPIO0_4	I/O	通用 GPIO。
GPIO0_5	I/O	通用 GPIO。
GPIO0_6	I/O	通用 GPIO。
GPIO0_7	I/O	通用 GPIO。
SIO1DO	O	SIO 接口 1 数据输出。
SIO1XCK	I/O	I ² S/PCM 发送位流时钟。
SIO1XFS	I/O	I ² S 发送左右声道选择信号（与 DAC 接口），或 PCM 帧同步信号。

2.2.5 EBI 管脚复用

EBI 的软件复用管脚如表 2-25 所示。



表2-25 EBI 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1
AL25	SMICS1N	reg71	GPIO3_2
AL26	EBIRDYN	reg74	IRRCV
AM27	NFCS1N	reg72	GPIO3_3
AJ26	NFRB	reg73	GPIO3_1

EBI 的软件复用管脚信号如表 2-26 所示。

表2-26 EBI 的软件复用信号描述

信号名	方向	说明
GPIO3_1	I/O	通用 GPIO。
GPIO3_2	I/O	通用 GPIO。
GPIO3_3	I/O	通用 GPIO。

2.2.6 GMAC 管脚复用

GMAC 的软件复用管脚如表 2-27 所示。

表2-27 GMAC 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1
E2	GTCLKOUT	reg75	GPIO1_0
D1	GCOL	reg76	GPIO1_1
G3	GCRS	reg77	GPIO1_2

GMAC 的软件复用信号描述如表 2-28 所示。

表2-28 GMAC 的软件复用信号描述

信号名	方向	说明
GPIO1_0	I/O	通用 GPIO。
GPIO1_1	I/O	通用 GPIO。



信号名	方向	说明
GPIO1_2	I/O	通用 GPIO。
IRRCV	I	红外遥控接收。

2.2.7 PCI 管脚复用

PCI 的软件复用管脚如表 2-29 所示。

表2-29 PCI 的软件复用管脚

Pin	Pad 信号	复用控制寄存器	复用信号 1
AJ23	PCIREQ3N	reg78	GPIO3_6
AM24	PCIREQ4N	reg79	GPIO3_7
AL22	PCIGRANT3N	reg80	GPIO7_0
AJ22	PCIGRANT4N	reg81	GPIO7_1

PCI 的软件复用信号描述如表 2-30 所示。

表2-30 PCI 的软件复用信号描述

信号名	方向	说明
GPIO3_6	I/O	通用 GPIO。
GPIO3_7	I/O	通用 GPIO。
GPIO7_0	I/O	通用 GPIO。
GPIO7_1	I/O	通用 GPIO。

2.3 硬件管脚复用描述

2.3.1 EBI 管脚复用

EBI 管脚复用如表 2-31 所示。



说明

表 2-31 中 power_on==0b1 表示芯片硬复位撤消时刻。下述对应管脚在单板设计时，必须通过上下拉电阻选择相应的模式。

表2-31 EBI 硬件管脚复用

Pin	Pad 信号	复用信号 1 (power_on==0b1)
AG31	EBIADR15	NFECC0
AG30	EBIADR16	NFECC1
AG32	EBIADR17	NFNUM0
AF29	EBIADR18	NFNUM1
AF32	EBIADR19	NFPAGE0
AF31	EBIADR20	NFPAGE1
AE29	EBIADR21	FUNSEL0
AE31	EBIADR22	FUNSEL1
AE30	EBIADR23	BOOTSEL0
AE32	EBIADR24	BOOTSEL1

EBI 的硬件复用信号描述如表 2-32 所示。

表2-32 与 EBI 复用的信号描述

信号名	方向	说明
NFECC0	I	与 NFECC1 配合使用，表示系统使用 Nand Flash boot 的 ECC 类型。 {NFECC1, NFECC0} 的含义如下： 00: 不使能； 01: 1bit 模式； 10: 4bit 模式； 11: 8bit 模式。



信号名	方向	说明
NFECC1	I	与 NFECC0 配合使用，表示系统使用 Nand Flash boot 的 ECC 类型。 {NFECC1, NFECC0}的含义如下： 00：不使能； 01：1bit 模式； 10：4bit 模式； 11：8bit 模式。
NFNUM0	I	与 NFNUM1 配合使用，表示系统使用 Nand Flash boot 的地址数目。 {NFNUM1, NFNUM0}的含义如下： 00：3 个地址周期； 01：4 个地址周期； 10：5 个地址周期； 11：6 个地址周期。
NFNUM1	I	与 NFNUM0 配合使用，表示系统使用 Nand Flash boot 的地址数目。 {NFNUM1, NFNUM0}的含义如下： 00：3 个地址周期； 01：4 个地址周期； 10：5 个地址周期； 11：6 个地址周期。
NFPAGE0	I	与 NFPAGE1 配合使用，表示系统使用 Nand Flash boot 的页面设置。 {NFPAGE1, NFPAGE0}的含义如下： 01：2KB； 10：4KB；（目前驱动暂未支持） 其他：保留。
NFPAGE1	I	与 NFPAGE0 配合使用，表示系统使用 Nand Flash boot 的页面设置。 {NFPAGE1, NFPAGE0}的含义如下： 01：2KB； 10：4KB；（目前驱动暂未支持） 其他：保留。



信号名	方向	说明
FUNSEL0	I	与 FUNSEL1 配和使用，表示 ARM debug 选择。 {FUNSEL1,FUNSEL0}的含义如下： 00: debug ARM1176; 10: debug ARM926; 01: debug ARM1176+ARM926; 11: 保留。
FUNSEL1	I	与 FUNSEL0 配和使用，表示 ARM debug 选择。 {FUNSEL1, FUNSEL0}的含义如下： 00: debug ARM1176; 10: debug ARM926; 01: debug ARM1176+ARM926; 11: 保留。
BOOTSEL0	I	与 BOOTSEL1 配和使用，表示启动模式选择。 {BOOTSEL1, BOOTSEL0}的含义如下： 00: 选择 Nor Flash 启动; 01: 选择 Nand Flash 启动; 1X: 选择 DDR 启动。
BOOTSEL1	I	与 BOOTSEL0 配和使用，表示启动模式选择。 {BOOTSEL1, BOOTSEL0}的含义如下： 00: 选择 Nor Flash 启动; 01: 选择 Nand Flash 启动; 1X: 选择 DDR 启动。

2.3.2 PCI 管脚复用

PCI 管脚复用如表 2-33 所示。



说明

表 2-33 中 `pcimode==0b1` 表示系统控制寄存器 `periphctrl11[pci_mode]` 配置为 1。

表2-33 PCI 硬件管脚复用

Pin	Pad 信号	复用信号 1 (<code>pcimode == 0b1</code>)
AL14	PCIIDSEL	PCIINTBN



Pin	Pad 信号	复用信号 1 (pcimode == 0b0)
AL19	PCIREQ0N	PCIGRANT_SLAVEN
AK21	PCIGRANT0N	PCIREQ_SLAVEN

PCI 的硬件复用信号描述如表 2-34 所示。

表2-34 与 PCI 复用的信号描述

信号名	方向	说明
PCIINTBN	I	PCI 中断 INTB 信号，低电平有效。
PCIGRANT_SLAVEN	I	从模式下，PCI 总线仲裁输入。
PCIREQ_SLAVEN	O	从模式下，PCI 总线请求输出。

2.4 IO Config（管脚复用控制）寄存器概览

IO Config（管脚复用控制）寄存器概览如表 2-35 所示。

表2-35 IO Config 寄存器概览（基地址是 0x200F_0000）

偏移地址	名称	描述	页码
0x0000	reg0	VI0HS 管脚的复用控制寄存器	2-67
0x0004	reg1	VI0VS 管脚的复用控制寄存器	2-68
0x0008	reg2	VI0DAT0 管脚的复用控制寄存器	2-68
0x000C	reg3	VI0DAT1 管脚的复用控制寄存器	2-69
0x0010	reg4	VI0DAT2 管脚的复用控制寄存器	2-69
0x0014	reg5	VI0DAT3 管脚的复用控制寄存器	2-70
0x0018	reg6	VI0DAT4 管脚的复用控制寄存器	2-70
0x001C	reg7	VI0DAT5 管脚的复用控制寄存器	2-71
0x0020	reg8	VI0DAT6 管脚的复用控制寄存器	2-71
0x0024	reg9	VI0DAT7 管脚的复用控制寄存器	2-72
0x0028	reg10	VI1DAT0 管脚的复用控制寄存器	2-72
0x002C	reg11	VI1DAT1 管脚的复用控制寄存器	2-73



偏移地址	名称	描述	页码
0x0030	reg12	VI1DAT2 管脚的复用控制寄存器	2-73
0x0034	reg13	VI1DAT3 管脚的复用控制寄存器	2-74
0x0038	reg14	VI1DAT4 管脚的复用控制寄存器	2-74
0x003C	reg15	VI1DAT5 管脚的复用控制寄存器	2-75
0x0040	reg16	VI1DAT6 管脚的复用控制寄存器	2-75
0x0044	reg17	VI1DAT7 管脚的复用控制寄存器	2-76
0x0048	reg18	VI2HS 管脚的复用控制寄存器	2-76
0x004C	reg19	VI2VS 管脚的复用控制寄存器	2-77
0x0050	reg20	VI2DAT0 管脚的复用控制寄存器	2-77
0x0054	reg21	VI2DAT1 管脚的复用控制寄存器	2-78
0x0058	reg22	VI2DAT2 管脚的复用控制寄存器	2-78
0x005C	reg23	VI2DAT3 管脚的复用控制寄存器	2-79
0x0060	reg24	VI2DAT4 管脚的复用控制寄存器	2-79
0x0064	reg25	VI2DAT5 管脚的复用控制寄存器	2-80
0x0068	reg26	VI2DAT6 管脚的复用控制寄存器	2-80
0x006C	reg27	VI2DAT7 管脚的复用控制寄存器	2-81
0x0070	reg28	VI3DAT0 管脚的复用控制寄存器	2-81
0x0074	reg29	VI3DAT1 管脚的复用控制寄存器	2-82
0x0078	reg30	VI3DAT2 管脚的复用控制寄存器	2-82
0x007C	reg31	VI3DAT3 管脚的复用控制寄存器	2-83
0x0080	reg32	VI3DAT4 管脚的复用控制寄存器	2-83
0x0084	reg33	VI3DAT5 管脚的复用控制寄存器	2-84
0x0088	reg34	VI3DAT6 管脚的复用控制寄存器	2-84
0x008C	reg35	VI3DAT7 管脚的复用控制寄存器	2-85
0x0090	reg36	VO0CK 管脚的复用控制寄存器	2-85
0x0094	reg37	VO0DAT0 管脚的复用控制寄存器	2-86
0x0098	reg38	VO0DAT1 管脚的复用控制寄存器	2-86
0x009C	reg39	VO0DAT2 管脚的复用控制寄存器	2-87



偏移地址	名称	描述	页码
0x00A0	reg40	VO0DAT3 管脚的复用控制寄存器	2-87
0x00A4	reg41	VO0DAT4 管脚的复用控制寄存器	2-88
0x00A8	reg42	VO0DAT5 管脚的复用控制寄存器	2-88
0x00AC	reg43	VO0DAT6 管脚的复用控制寄存器	2-89
0x00B0	reg44	VO0DAT7 管脚的复用控制寄存器	2-89
0x00B4	reg45	VO1CK 管脚的复用控制寄存器	2-90
0x00B8	reg46	VO1DAT0 管脚的复用控制寄存器	2-90
0x00BC	reg47	VO1DAT1 管脚的复用控制寄存器	2-91
0x00C0	reg48	VO1DAT2 管脚的复用控制寄存器	2-91
0x00C4	reg49	VO1DAT3 管脚的复用控制寄存器	2-92
0x00C8	reg50	VO1DAT4 管脚的复用控制寄存器	2-92
0x00CC	reg51	VO1DAT5 管脚的复用控制寄存器	2-93
0x00D0	reg52	VO1DAT6 管脚的复用控制寄存器	2-93
0x00D4	reg53	VO1DAT7 管脚的复用控制寄存器	2-94
0x00D8	reg54	VO1DAT8 管脚的复用控制寄存器	2-94
0x00DC	reg55	VO1DAT9 管脚的复用控制寄存器	2-95
0x00E0	reg56	VO1DAT10 管脚的复用控制寄存器	2-95
0x00E4	reg57	VO1DAT11 管脚的复用控制寄存器	2-96
0x00E8	reg58	VO1DAT12 管脚的复用控制寄存器	2-96
0x00EC	reg59	VO1DAT13 管脚的复用控制寄存器	2-97
0x00F0	reg60	VO1DAT14 管脚的复用控制寄存器	2-97
0x00F4	reg61	VO1DAT15 管脚的复用控制寄存器	2-98
0x00F8	reg62	GPIO3_0 管脚的复用控制寄存器	2-98
0x00FC	reg63	SDA 管脚的复用控制寄存器	2-99
0x0100	reg64	SCL 管脚的复用控制寄存器	2-99
0x0104	reg65	SIO0XFS 管脚的复用控制寄存器	2-100
0x0108	reg66	SIO0XCK 管脚的复用控制寄存器	2-100
0x010C	reg67	ACKOUT 管脚的复用控制寄存器	2-101
0x0110	reg68	SIO2DI 管脚的复用控制寄存器	2-101



偏移地址	名称	描述	页码
0x0114	reg69	SIO2RFS 管脚的复用控制寄存器	2-102
0x0118	reg70	SIO2RCK 管脚的复用控制寄存器	2-102
0x011C	reg71	SMICS1N 管脚的复用控制寄存器	2-103
0x0120	reg72	NFCS1N 管脚的复用控制寄存器	2-103
0x0124	reg73	NFRB 管脚的复用控制寄存器	2-104
0x0128	reg74	EBIRDYN 管脚的复用控制寄存器	2-104
0x012C	reg75	GTCLKOUT 管脚的复用控制寄存器	2-105
0x0130	reg76	GCOL 管脚的复用控制寄存器	2-105
0x0134	reg77	GCRS 管脚的复用控制寄存器	2-106
0x0138	reg78	PCIREQ3N 管脚的复用控制寄存器	2-106
0x013C	reg79	PCIREQ4N 管脚的复用控制寄存器	2-107
0x0140	reg80	PCIGRANT3N 管脚的复用控制寄存器	2-107
0x0144	reg81	PCIGRANT4N 管脚的复用控制寄存器	2-108
0x0148	reg82	VORGBDV 管脚的复用控制寄存器	2-108

2.5 复用寄存器描述

reg0

VI0HS 管脚复用控制寄存器。

	Offset Address	Register Name	Total Reset Value																						
	0x0000	reg0	0x0000_0000																						
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																								
Name	reserved															reg0									
Reset	0 0																								
	Bits	Access	Name	Description																					
	[1:0]	RW	reg0	VI0HS 管脚的具体复用情况。 00: VI0HS;																					



			01: URXD2; 10: GPIO1_3; 其它: 保留。
--	--	--	---------------------------------------

reg1

VI0VS 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0004		reg1		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg1
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg1	VI0VS 管脚的具体复用情况。 00: VI0VS; 01: UTXD2; 10: GPIO1_4; 其它: 保留。						

reg2

VI0DAT0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0008		reg2		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg2
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg2	VI0DAT0 管脚的具体复用情况。 0: VI0DAT0;						

			1: VOUDAT8。
--	--	--	-------------

reg3

VI0DAT1 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x000C		reg3		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg3
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg3	VI0DAT1 管脚的具体复用情况。 0: VI0DAT1; 1: VOUDAT9。						

reg4

VI0DAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0010		reg4		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg4
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg4	VI0DAT2 管脚的具体复用情况。 0: VI0DAT2; 1: VOUDAT10。						



reg5

VI0DAT3 管脚复用控制寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0014				reg5				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											reg5				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[0]	RW	reg5		VI0DAT3 管脚的具体复用情况。 0: VI0DAT3; 1: VOUDAT11。																											

reg6

VI0DAT4 管脚复用控制寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0018				reg6				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											reg6				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[0]	RW	reg6		VI0DAT4 管脚的具体复用情况。 0: VI0DAT4; 1: VOUDAT12。																											

reg7

VI0DAT5 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x001C		reg7		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg7
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg7	VI0DAT5 管脚的具体复用情况。 0: VI0DAT5; 1: VOUDAT13。						

reg8

VI0DAT6 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0020		reg8		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg8
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg8	VI0DAT6 管脚的具体复用情况。 0: VI0DAT6; 1: VOUDAT14。						

reg9

VI0DAT7 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0024		reg9		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg9
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg9	VI0DAT7 管脚的具体复用情况。 0: VI0DAT7; 1: VOUDAT15。						

reg10

VI1DAT0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0028		reg10		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg10
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg10	VI1DAT0 管脚的具体复用情况。 00: VI1DAT0; 01: GPIO4_0; 10: VOUDAT0; 其它: 保留。						

reg11

VI1DAT1 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x002C		reg11		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg11
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg11	VI1DAT1 管脚的具体复用情况。 00: VI1DAT1; 01: GPIO4_1; 10: VOUDAT1; 其它: 保留。						

reg12

VI1DAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0030		reg12		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg12
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg12	VI1DAT2 管脚的具体复用情况。 00: VI1DAT2; 01: GPIO4_2; 10: VOUDAT2; 其它: 保留。						

reg13

VI1DAT3 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0034		reg13		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg13
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg13	VI1DAT3 管脚的具体复用情况。 00: VI1DAT3; 01: GPIO4_3; 10: VOUDAT3; 其它: 保留。						

reg14

VI1DAT4 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0038		reg14		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg14
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg14	VI1DAT4 管脚的具体复用情况。 00: VI1DAT4; 01: GPIO4_4; 10: VOUDAT4; 其它: 保留。						

reg15

VI1DAT5 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x003C		reg15		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg15
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg15	VI1DAT5 管脚的具体复用情况。 00: VI1DAT5; 01: GPIO4_5; 10: VOUDAT5; 其它: 保留。						

reg16

VI1DAT6 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0040		reg16		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg16
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg16	VI1DAT6 管脚的具体复用情况。 00: VI1DAT6; 01: GPIO4_6; 10: VOUDAT6; 其它: 保留。						

reg17

VI1DAT7 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0044		reg17		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg17
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg17	VI1DAT7 管脚的具体复用情况。 00: VI1DAT7; 01: GPIO4_7; 10: VOUDAT7; 其它: 保留。						

reg18

VI2HS 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0048		reg18		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg18
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg18	VI2HS 管脚的具体复用情况。 00: VI2HS; 01: URXD3; 10: GPIO3_4; 其它: 保留。						

reg19

VI2VS 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x004C		reg19		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg19
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg19	VI2VS 管脚的具体复用情况。 00: VI2VS; 01: UTXD3; 10: GPIO3_5; 其它: 保留。						

reg20

VI2DAT0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0050		reg20		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg20
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg20	VI2DAT0 管脚的具体复用情况。 0: VI2DAT0; 1: GPIO5_0。						

reg21

VI2DAT1 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0054		reg21		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg21
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg21	VI2DAT1 管脚的具体复用情况。 0: VI2DAT1; 1: GPIO5_1。						

reg22

VI2DAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0058		reg22		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg22
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg22	VI2DAT2 管脚的具体复用情况。 0: VI2DAT2; 1: GPIO5_2。						

reg23

VI2DAT3 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x005C		reg23		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg23
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg23	VI2DAT3 管脚的具体复用情况。 0: VI2DAT3; 1: GPIO5_3。						

reg24

VI2DAT4 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0060		reg24		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg24
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg24	VI2DAT4 管脚的具体复用情况。 0: VI2DAT4; 1: GPIO5_4。						

reg25

VI2DAT5 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0064		reg25		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg25
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg25	VI2DAT5 管脚的具体复用情况。 0: VI2DAT5; 1: GPIO5_5。						

reg26

VI2DAT6 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0068		reg26		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg26
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg26	VI2DAT6 管脚的具体复用情况。 0: VI2DAT6; 1: GPIO5_6。						

reg27

VI2DAT7 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x006C		reg27		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg27
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg27	VI2DAT7 管脚的具体复用情况。 0: VI2DAT7; 1: GPIO5_7。						

reg28

VI3DAT0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0070		reg28		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg28
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg28	VI3DAT0 管脚的具体复用情况。 0: VI3DAT0; 1: GPIO6_0。						

reg29

VI3DAT1 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0074		reg29		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg29
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg29	VI3DAT1 管脚的具体复用情况。 0: VI3DAT1; 1: GPIO6_1。						

reg30

VI3DAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0078		reg30		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg30
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg30	VI3DAT2 管脚的具体复用情况。 0: VI3DAT2; 1: GPIO6_2。						

reg31

VI3DAT3 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x007C		reg31		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg31
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg31	VI3DAT3 管脚的具体复用情况。 0: VI3DAT3; 1: GPIO6_3。						

reg32

VI3DAT4 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0080		reg32		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg32
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg32	VI3DAT4 管脚的具体复用情况。 0: VI3DAT4; 1: GPIO6_4。						

reg33

VI3DAT5 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0084		reg33		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg33
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg33	VI3DAT5 管脚的具体复用情况。 0: VI3DAT5; 1: GPIO6_5。						

reg34

VI3DAT6 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0088		reg34		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg34
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg34	VI3DAT6 管脚的具体复用情况。 0: VI3DAT6; 1: GPIO6_6。						

reg35

VI3DAT7 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x008C		reg35		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg35
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg35	VI3DAT7 管脚的具体复用情况。 0: VI3DAT7; 1: GPIO6_7。						

reg36

VO0CK 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0090		reg36		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg36
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg36	VO0CK 管脚的具体复用情况。 00: GPIO1_5; 01: VO0CK; 10: VORDAT0; 其它: 保留。						

reg37

VO0DAT0 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0094		reg37		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg37
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg37	VO0DAT0 管脚的具体复用情况。 00: GPIO2_0; 01: VO0DAT0; 10: VORDAT1; 其它: 保留。						

reg38

VO0DAT1 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0098		reg38		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg38
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg38	VO0DAT1 管脚的具体复用情况。 00: GPIO2_1; 01: VO0DAT1; 10: VORDAT2; 其它: 保留。						

reg39

VO0DAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value																												
0x009C		reg39		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											reg39				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[1:0]	RW	reg39	VO0DAT2 管脚的具体复用情况。 00: GPIO2_2; 01: VO0DAT2; 10: VORDAT3; 其它: 保留。																													

reg40

VO0DAT3 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value																												
0x00A0		reg40		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											reg40				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[1:0]	RW	reg40	VO0DAT3 管脚的具体复用情况。 00: GPIO2_3; 01: VO0DAT3; 10: VORDAT4; 其它: 保留。																													

reg41

VO0DAT4 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00A4		reg41		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg41
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg41	VO0DAT4 管脚的具体复用情况。 00: GPIO2_4; 01: VO0DAT4; 10: VORDAT5; 其它: 保留。						

reg42

VO0DAT5 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00A8		reg42		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg42
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg42	VO0DAT5 管脚的具体复用情况。 00: GPIO2_5; 01: VO0DAT5; 10: VORDAT6; 其它: 保留。						

reg43

VO0DAT6 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00AC		reg43		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg43
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg43	VO0DAT6 管脚的具体复用情况。 00: GPIO2_6; 01: VO0DAT6; 10: VORDAT7; 其它: 保留。						

reg44

VO0DAT7 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00B0		reg44		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg44
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg44	VO0DAT7 管脚的具体复用情况。 00: GPIO2_7; 01: VO0DAT7; 10: VOGDAT0; 其它: 保留。						

reg45

VO1CK 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00B4		reg45		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg45
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg45	VO1CK 管脚的具体复用情况。 00: GPIO7_2; 01: VO1CK; 10: SDIOCK; 其它: 保留。						

reg46

VOIDAT0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00B8		reg46		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg46
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg46	VOIDAT0 管脚的具体复用情况。 0: VGA1HS; 1: VOIDAT0。						

reg47

VOIDAT1 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00BC		reg47		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg47
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg47	VO1DAT1 管脚的具体复用情况。 0: VGA1VS; 1: VO1DAT1。						

reg48

VOIDAT2 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00C0		reg48		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg48
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg48	VOIDAT2 管脚的具体复用情况。 00: SDIOCMD; 01: VO1DAT2; 10: VOGDAT1; 其它: 保留。						

reg49

VOIDAT3 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00C4		reg49		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg49
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg49	VO1DAT3 管脚的具体复用情况。 00: SDIODAT0; 01: VO1DAT3; 10: VOGDAT2; 其它: 保留。						

reg50

VO1DAT4 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00C8		reg50		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg50
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg50	VO1DAT4 管脚的具体复用情况。 00: SDIODAT1; 01: VO1DAT4; 10: VOGDAT3; 其它: 保留。						

reg51

VO1DAT5 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00CC		reg51		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg51
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg51	VO1DAT5 管脚的具体复用情况。 00: SDIODAT2; 01: VO1DAT5; 10: VOGDAT4; 其它: 保留。						

reg52

VO1DAT6 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00D0		reg52		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg52
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg52	VO1DAT6 管脚的具体复用情况。 00: SDIODAT3; 01: VO1DAT6; 10: VOGDAT5; 其它: 保留。						

reg53

VO1DAT7 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00D4		reg53		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg53
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg53	VO1DAT7 管脚的具体复用情况。 00: GPIO1_6; 01: VO1DAT7; 10: VOGDAT6; 其它: 保留。						

reg54

VO1DAT8 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00D8		reg54		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg54
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg54	VO1DAT8 管脚的具体复用情况。 00: SPICK; 01: VO1DAT8; 10: VOGDAT7; 其它: 保留。						

reg55

VO1DAT9 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00DC		reg55		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg55
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg55	VO1DAT9 管脚的具体复用情况。 00: SPIDI; 01: VO1DAT9; 10: VOBDAT0; 其它: 保留。						

reg56

VOIDAT10 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00E0		reg56		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg56
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg56	VOIDAT10 管脚的具体复用情况。 00: SPIDO; 01: VOIDAT10; 10: VOBDAT1; 其它: 保留。						

reg57

VOIDAT11 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00E4		reg57		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg57
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg57	VOIDAT11 管脚的具体复用情况。 00: SPICSN0; 01: VOIDAT11; 10: VOBDAT2; 其它: 保留。						

reg58

VOIDAT12 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00E8		reg58		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg58
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg58	VOIDAT12 管脚的具体复用情况。 00: SPICSN1; 01: VOIDAT12; 10: VOBDAT3; 其它: 保留。						

reg59

VOIDAT13 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00EC		reg59		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg59
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg59	VOIDAT13 管脚的具体复用情况。 00: GPIO7_4; 01: VOIDAT13; 10: VOBDAT4; 11: VGA0HS。						

reg60

VOIDAT14 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00F0		reg60		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg60
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg60	VOIDAT14 管脚的具体复用情况。 00: GPIO7_5; 01: VOIDAT14; 10: VOBDAT5; 11: VGA0VS。						

reg61

VOIDAT15 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x00F4		reg61		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg61
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg61	VO1DAT15 管脚的具体复用情况。 00: GPIO1_7; 01: VO1DAT15; 10: VOBDAT6; 其它: 保留。						

reg62

GPIO3_0 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00F8		reg62		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg62
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg62	GPIO3_0 管脚的具体复用情况。 0: GPIO3_0; 1: VOBDAT7。						

reg63

SDA 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x00FC		reg63		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg63
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg63	SDA 管脚的具体复用情况。 0: SDA; 1: GPIO0_0。						

reg64

SCL 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0100		reg64		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg64
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg64	SCL 管脚的具体复用情况。 0: SCL; 1: GPIO0_1。						

reg65

SIO0XFS 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0104		reg65		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg65
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg65	SIO0XFS 管脚的具体复用情况。 0: SIO0XFS; 1: GPIO0_2。						

reg66

SIO0XCK 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0108		reg66		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg66
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg66	SIO0XCK 管脚的具体复用情况。 0: SIO0XCK; 1: GPIO0_3。						

reg67

ACKOUT 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x010C		reg67		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg67
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg67	ACKOUT 管脚的具体复用情况。 0: GPIO0_4; 1: ACKOUT。						

reg68

SIO2DI 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0110		reg68		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg68
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg68	SIO2DI 管脚的具体复用情况。 00: GPIO0_5; 01: SIO2DI; 10: SIO1DO; 其它: 保留。						

reg69

SIO2RFS 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0114		reg69		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg69
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg69	SIO2RFS 管脚的具体复用情况。 00: GPIO0_6; 01: SIO2RFS; 10: SIO1XFS; 其它: 保留。						

reg70

SIO2RCK 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0118		reg70		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg70
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[1:0]	RW	reg70	SIO2RCK 管脚的具体复用情况。 00: GPIO0_7; 01: SIO2RCK; 10: SIO1XCK; 其它: 保留。						

reg71

SMICS1N 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x011C		reg71		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg71
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg71	SMICS1N 管脚的具体复用情况。 0: GPIO3_2; 1: SMICS1N。						

reg72

NFCS1N 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0120		reg72		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg72
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg72	NFCS1N 管脚的具体复用情况。 0: GPIO3_3; 1: NFCS1N。						

reg73

NFRB 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0124		reg73		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg73
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg73	NFRB 管脚的具体复用情况。 0: NFRB; 1: GPIO3_1。						

reg74

EBIRDYN 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0128		reg74		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg74
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg74	EBIRDYN 管脚的具体复用情况。 0: EBIRDYN; 1: IRRCV。						

reg75

GTCLKOUT 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x012C		reg75		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg75
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg75	GTCLKOUT 管脚的具体复用情况。 0: GPIO1_0; 1: GTCLKOUT。						

reg76

GCOL 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0130		reg76		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg76
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg76	GCOL 管脚的具体复用情况。 0: GCOL; 1: GPIO1_1。						

reg77

GCRS 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0134		reg77		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg77
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg77	GCRS 管脚的具体复用情况。 0: GCRS; 1: GPIO1_2。						

reg78

PCIREQ3N 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0138		reg78		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg78
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg78	PCIREQ3N 管脚的具体复用情况。 0: GPIO3_6; 1: PCIREQ3N。						

reg79

PCIREQ4N 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x013C		reg79		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg79
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg79	PCIREQ4N 管脚的具体复用情况。 0: GPIO3_7; 1: PCIREQ4N。						

reg80

PCIGRANT3N 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0140		reg80		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg80
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg80	PCIGRANT3N 管脚的具体复用情况。 0: GPIO7_0; 1: PCIGRANT3N。						

reg81

PCIGRANT4N 管脚复用控制寄存器。



Offset Address		Register Name		Total Reset Value					
0x0144		reg81		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg81
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg81	PCIGRANT4N 管脚的具体复用情况。 0: GPIO7_1; 1: PCIGRANT4N。						

reg82

VORGBDV 管脚复用控制寄存器。

Offset Address		Register Name		Total Reset Value					
0x0148		reg82		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								reg82
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[0]	RW	reg82	VORGBDV 管脚的具体复用情况。 0: GPIO7_3; 1: VORGBDV。						

2.6 上下电顺序推荐

推荐先上高电压，后上低电压；先关低电压，后关高电压。



2.7 外部中断

外部中断请参见“3.4 中断系统”。

2.8 电气特性

2.8.1 DC/AC 参数

DC/AC 参数如表 2-36~表 2-37 所示。

表2-36 DC 参数表 (DVDD33=3.3V)

符号	参数	最小值	典型值	最大值	单位	说明
V _{IH}	高电平输入电压	2.0	-	5.5	V	-
V _{IL}	低电平输入电压	-0.3	-	0.8	V	-
I _L	输入漏电流	-	-	±10	μA	-
I _{OZ}	三态输出漏电流	-	-	±10	μA	-
V _{OL}	低电平输出电压	-	-	0.4	V	-
R _{PU}	上拉电阻	63	92	142	kΩ	-
R _{PD}	下拉电阻	57	91	159	kΩ	-

表2-37 DC 参数表 (DVDD18=1.8V)

符号	参数	最小值	典型值	最大值	单位	说明
V _{ref}	参考电压	0.49×DVD D18	0.50×DVD D18	0.51×DVD D18	V	-
V _{IH(dc)}	高电平输入电压	V _{ref} +0.125	-	DVDD18+ 0.3	V	-
V _{IL(dc)}	低电平输入电压	-0.3	-	V _{ref} -0.125	V	-
I _{OH(dc)}	高电平输出电流	-13.4	-	-	mA	V _{oh(dc)} =1.42V
I _{OL(dc)}	低电平输出电流	13.4	-	-	mA	V _{ol(dc)} = 0.28V
V _{IH(ac)}	高电平输入电压	V _{ref} +0.25	-	-	V	-
V _{IL(ac)}	低电平输入电压	-	-	V _{ref} -0.25	V	-



符号	参数	最小值	典型值	最大值	单位	说明
R_{TT}	输入内部上拉到 $0.5 \times DVDD18$ 的 ODT	-30% -30% -30%	150 75 50	+30% +30% +30%	Ohm	-
V_{TTINT}	有效端接电压	-2.0%	$0.5 \times DVDD18$	+2.0%	V	-
I_{LS}	输入漏电流	-1.0	-	1.0	μA	无端接 SSTL 模式下

2.8.2 推荐工作条件

推荐工作条件如表 2-38 所示。

表2-38 推荐工作条件

符号	参数	最小值	典型值	最大值	单位
TOPT	操作环境温度	-20	-	85	°C
TS	芯片封装表面温度	-20	-	105	°C
TJ	芯片结温	-40	-	125	°C
DVDD10	内部 Core 电源	0.95	1	1.1	V
DVDD33	I/O 电源	3.135	3.3	3.6	V
DVDD18	DDR I/O 电源	1.7	1.8	1.9	V
VREF	DDR 参考电源	0.85	0.9	0.95	V
AVDD33_PLL	PLL 模拟电源	3.135	3.3	3.6	V
DVDD10_PLL	PLL 数字电源	0.95	1	1.1	V
AVDD33_USB	USB 模拟电源	3.135	3.3	3.465	V
DVDD10_USB	USB 数字电源	0.95	1	1.05	V
AVDD33_VDAC	VDAC 模拟电源	3.135	3.3	3.6	V
DVDD10_DAC	VDAC 数字电源	0.95	1	1.1	V
DVDD33_DAC	VDAC 数字电源	3.0	3.3	3.6	V



2.9 PCB 布线建议

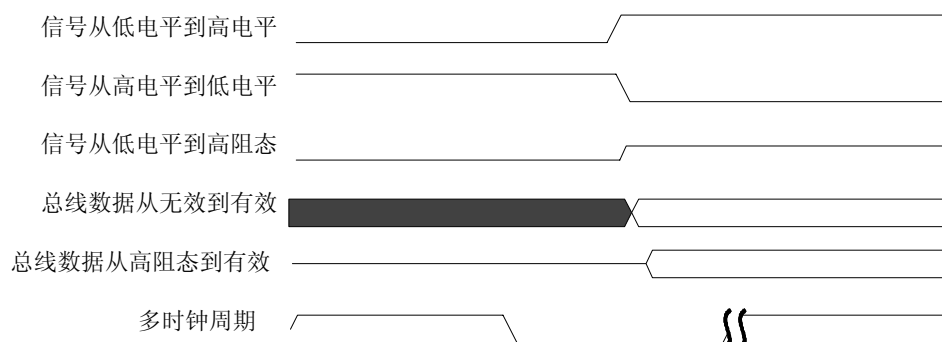
PCB 布线建议请参见《Hi3520 硬件设计 用户指南》。

2.10 时序规格

2.10.1 时序图例

图 2-1 介绍了本手册的时序图中的图元。

图2-1 时序图元说明



2.10.2 DDR2 接口时序

2.10.2.1 写操作时序

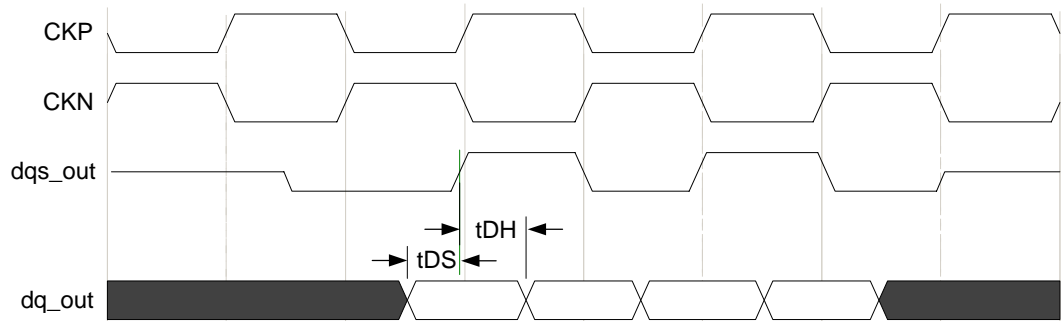
dqs_out 相对于 dq_out 的写操作时序

dqs_out 相对于 dq_out 的写操作时序中需要检查的时序参数是 tDS 和 tDH。在 DDR2-800 中，tDS=0.50ns；tDH=0.125ns。

dqs_out 相对 DDR 时钟 CKP/CKN 以及 dq_out 的时序如图 2-2 所示。



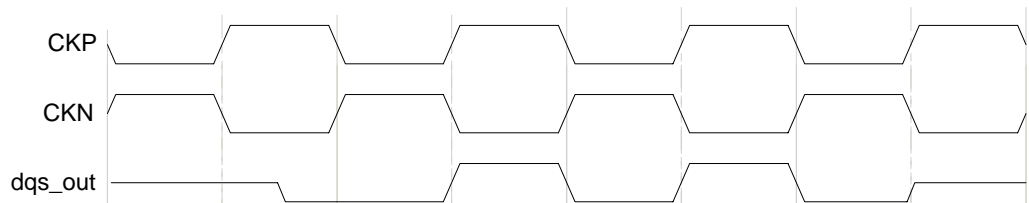
图2-2 dqs_out 相对于 dq_out 的写操作时序图



dqs_out 相对于 ck 的写操作时序

dqs_out 相对于 ck 的写操作时序中需要检查的时序参数是 tDSS 和 tDSH，这两个参数的值为 $tCK \times 0.2$ ，时序关系如图 2-3 所示。

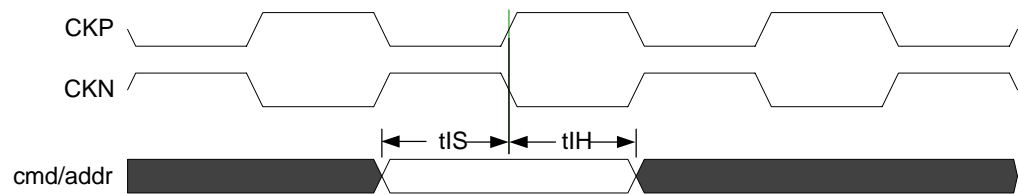
图2-3 dqs_out 相对于 ck 的写操作时序图



cmd/addr 相对于 ck 的写操作时序

cmd/addr 相对于 ck 的写操作时序如图 2-4 所示。

图2-4 cmd/addr 相对于 ck 的写操作时序图



2.10.2.2 读操作时序

cmd/addr 相对于 ck 的读操作时序

请参见“2.10.2.1 写操作时序”中“cmd/addr 相对于 ck 的写操作时序”的描述。

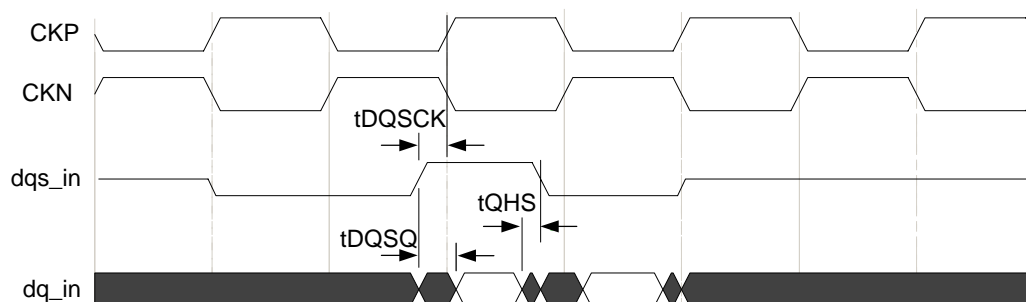


dqs_in 相对于 dq_in 的读操作时序

对于 DDR2 SDRAM 输出时序，理想情况下，DQS 和 CK 是同相位，但是有 t_{DQSCK} 的偏斜。该参数的值为 0.35ns。 t_{DQSQ} 是 dq 和 dqs 之间的抖动，是最晚有效的 dq 相对于 dqs 的抖动，该值为 0.2ns； t_{QHS} 是最早有效的 dq 相对于 dqs 的抖动，其值为 0.3ns。

DDR2 SDRAM 输出时序如图 2-5 所示。

图2-5 DDR2 SDRAM 输出时序图



2.10.2.3 时序参数

本文中描述的时序都是 DDR PHY 侧的时序。对于 Hi3520，以 DDR2-800 的时序参数为依据，具体参数信息如表 2-39~表 2-41 所示。

表2-39 Clock Parameters

参数	典型值	单位
Memory clock frequency	400.00	MHz
PLL clock jitter	0.200	ns
PLL duty cycle	45.000	%
CLK Skew	0.200	ns

表2-40 Memory Device Parameters

参数	典型值	单位
Memory Type	DDR2-800	-
tDSS: DQS falling edge-CK setup time	0.2	Tck
tDSH: DQS falling edge-CK hold time	0.2	Tck
tDS: DQ/DM-DQS setup time	0.050	ns
tDH: DQ/DM-DQS hold time	0.125	ns



参数	典型值	单位
DQS-DQ Skew, tDQSQ	0.200	ns
Data hold Skew Factor, tQHS	0.400	ns
Adr/Cmd Setup time, tIS	0.175	ns
Adr/Cmd Hold time, tIH	0.250	ns
tDQSCK: DQS output access time from CKP/CKN (min)	-0.350	ns
tDQSCK: DQS output access time from CKP/CKN (max)	0.350	ns

表2-41 Package/Board Parameters

参数	典型值	单位
Package trace skew (max/min)	0.060	ns
Board trace length mismatch (For Dqs/Dq)	0.050	ns
Board trace length mismatch (For others)	0.100	ns
Board and Package Uncertainty(Output)	0.360	ns
Board and Package Uncertainty(Input)	0.360	ns
ASIC Skew	0.200	ns

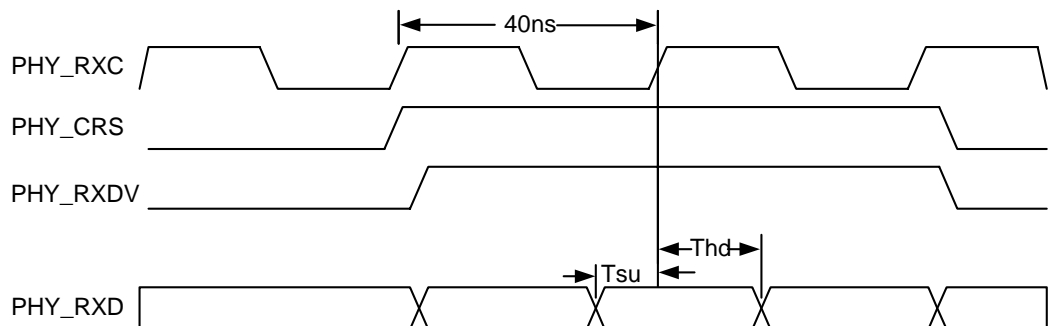
2.10.3 GMAC 接口时序

芯片提供标准的 MII 接口，连接 PHY（Physical Layer Entity Sublayer）芯片，符合 MII 接口时序标准。

MII 接口接收时序

MII 接口 100Mbit/s 接收时序如图 2-6 所示。

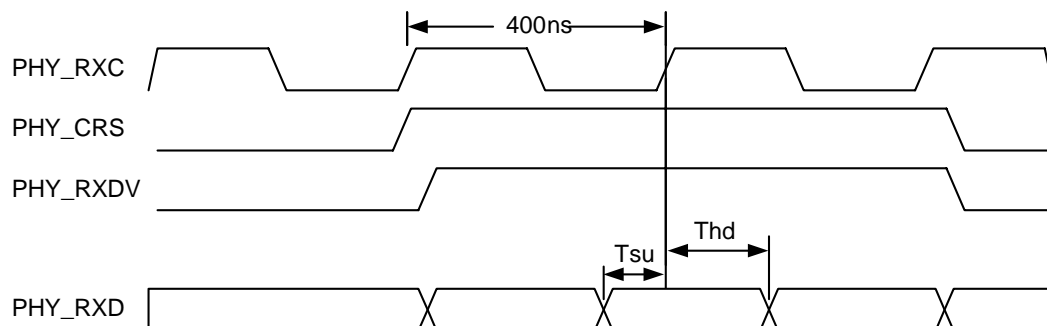
图2-6 MII 接口 100Mbit/s 接收时序





MII 接口 10Mbit/s 接收时序如图 2-7 所示。

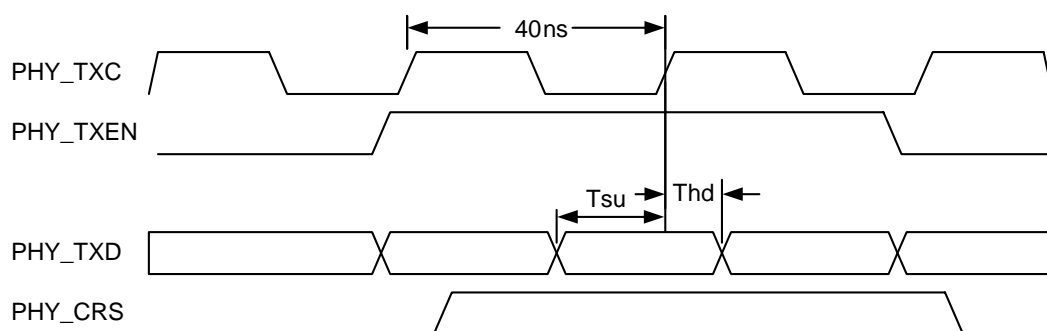
图2-7 MII 接口 10Mbit/s 接收时序



MII 接口发送时序

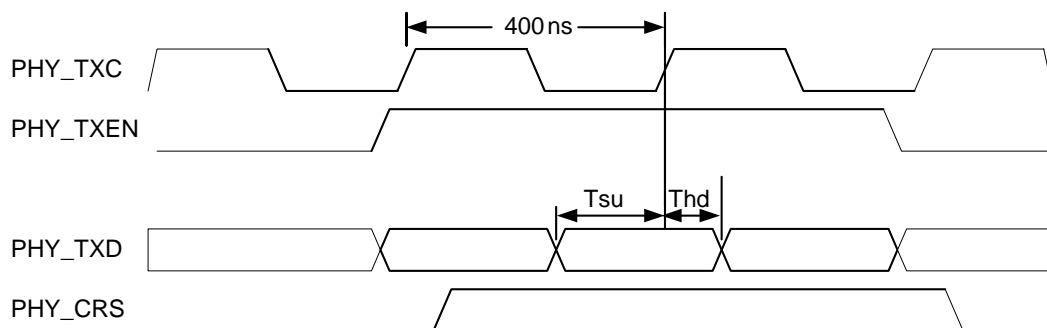
MII 接口 100Mbit/s 发送时序如图 2-8 所示。

图2-8 MII 接口 100Mbit/s 发送时序



MII 接口 10Mbit/s 发送时序如图 2-9 所示。

图2-9 MII 接口 10Mbit/s 发送时序





MII 接口时序参数

MII 接口时序参数如表 2-42 所示。

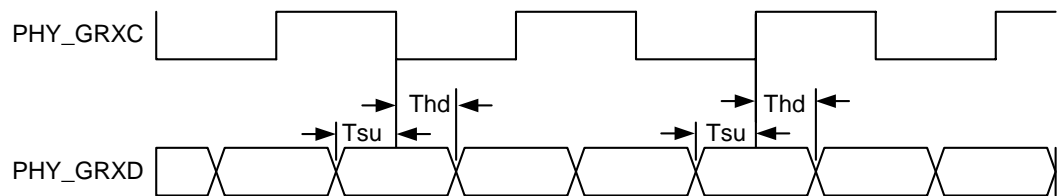
表2-42 MII 接口时序参数

参数	符号	信号	最小值	最大值	单位
MII 信号建立时间	Tsu (RX)	PHY_RXDV、 PHY_RXD[3:0]	10	-	ns
MII 信号保持时间	Thd (RX)	PHY_RXDV、 PHY_RXD[3:0]	10	-	ns
MII 信号建立时间	Tsu (TX)	PHY_TXEN、 PHY_TXD[3:0]	6	-	ns
MII 信号保持时间	Thd (TX)	PHY_TXEN、 PHY_TXD[3:0]	2	-	ns

RGMII 接口接收时序

RGMII 接口接收时序如图 2-10 所示。

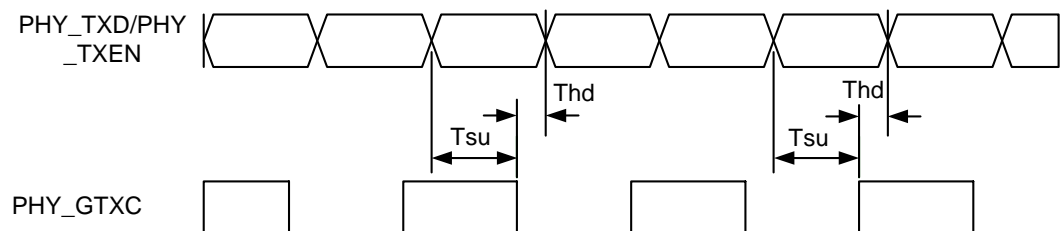
图2-10 RGMII 接口接收时序



RGMII 接口发送时序

RGMII 接口发送时序如图 2-11 所示。

图2-11 RGMII 接口发送时序





RGMII 接口时序参数

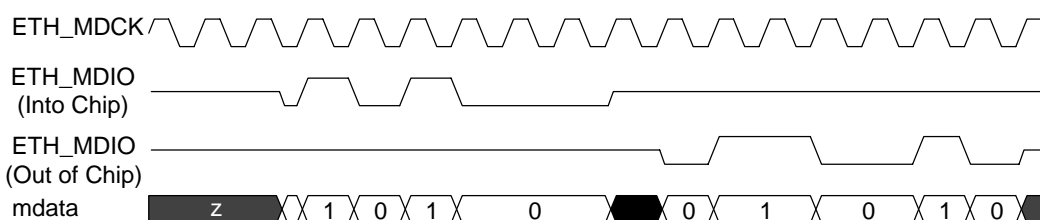
表2-43 RGMII 接口时序参数

参数	符号	信号	最小值	最大值	单位
RGMII 信号建立时间	Tsu (RX)	PHY_RXDV、 PHY_RXD[3:0]	0.8	-	ns
RGMII 信号保持时间	Thd (RX)	PHY_RXDV、 PHY_RXD[3:0]	1.2	-	ns
RGMII 信号建立时间	Tsu (TX)	PHY_TXEN、 PHY_TXD[3:0]	-0.7	0.7	ns
RGMII 信号保持时间	Thd (TX)	PHY_TXEN、 PHY_TXD[3:0]	-0.7	0.7	ns

MDIO 读时序

MDIO 接口读时序如图 2-12 所示。

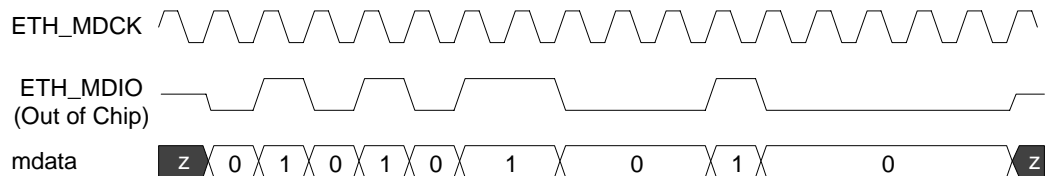
图2-12 MDIO 接口读时序



MDIO 写时序

MDIO 接口写时序如图 2-13 所示。

图2-13 MDIO 接口写时序

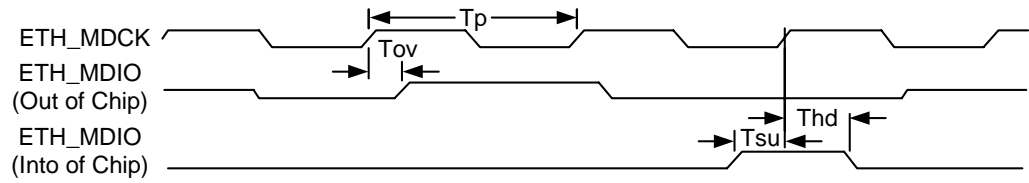


MDIO 接口时序参数

MDIO 接口时序参数示意如图 2-14 所示。



图2-14 MDIO 接口时序示意



MDIO 接口时序参数如表 2-44 所示。

表2-44 MDIO 接口时序参数

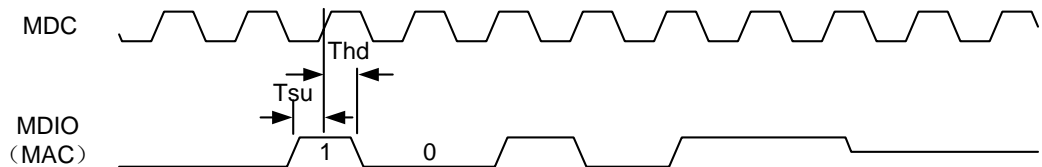
参数	符号	信号	最小值	最大值	单位
MDIO 发送数据延迟时间	T_{ov}	MDIO	37	37	ns
MDIO 时钟周期	T_p	MDC	370 或 740	370 或 740	ns
MDIO 接收数据建立时间	T_{su}	MDIO	10	-	ns
MDIO 接收数据保持时间	T_{hd}	MDIO	10	-	ns

注：MDC 时钟周期 T_p 可通过调整 MDC 频率进行改变。

MDIO 发送时序

MDIO 发送时序如图 2-15 所示。

图2-15 MDIO 发送时序



MDIO 发送时序参数如表 2-45 所示。

表2-45 MDIO 发送时序参数

参数	符号	信号	最小值	最大值	单位
MDIO 发送数据建立时间 (MDC 18MHz)	T_{su}	MDIO	32	-	ns
MDIO 发送数据保持时间 (MDC 18MHz)	T_{hd}	MDIO	24	-	ns

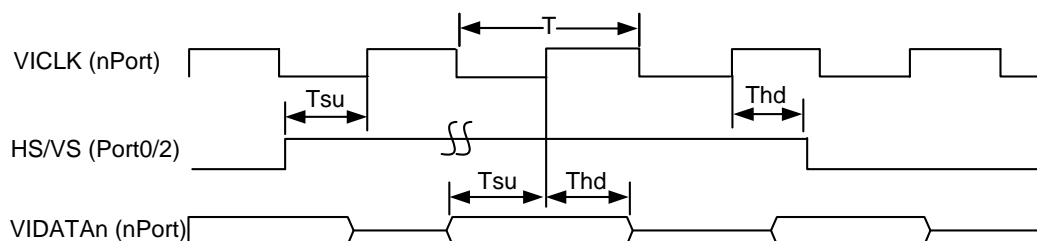


参数	符号	信号	最小值	最大值	单位
MDIO 发送数据建立时间 (MDC 2.5MHz)	Tsu	MDIO	200	-	ns
MDIO 发送数据保持时间 (MDC 2.5MHz)	Thd	MDIO	200	-	ns

2.10.4 VI 接口时序

VI 接口时序如图 2-16 所示。

图2-16 VI 接口时序图



VI 接口时序参数如表 2-46 所示。

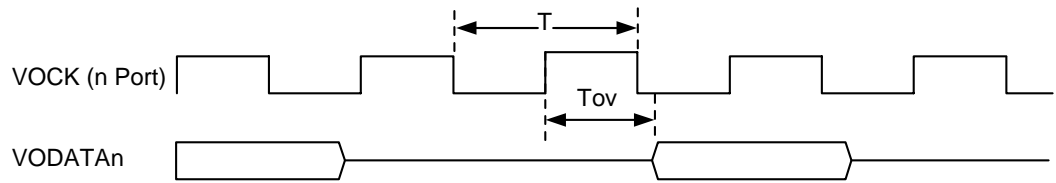
表2-46 VI 接口时序参数

参数	符号	最小值	典型值	最大值	单位
VICKL 时钟周期 (VIU 输入时钟与其支持的协议相关)	T	-	-	-	ns
输入信号建立时间要求	Tsu	2.93	-	-	ns
输入信号保持时间要求	Thd	2	-	-	ns

2.10.5 VO 接口时序

VO 接口时序如图 2-17 所示。

图2-17 VO 输出接口时序图



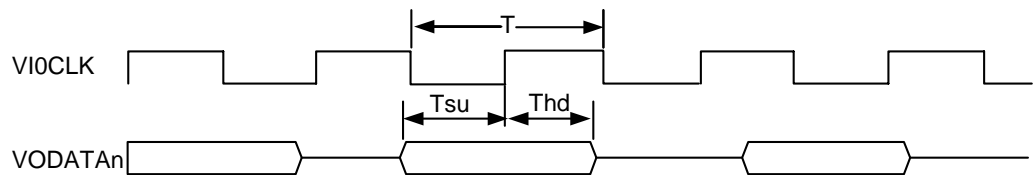
VO 接口输出时序参数如表 2-48 所示。

表2-47 VO 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
VOCK 时钟周期	T	-	37.03	-	ns
输出信号延时	Tov	$T/2 - \Delta T$	$T/2 + 1$	$T/2 + 1 + \Delta T$	ns

注： ΔT 与支持的协议相关，例如 BT.656 协议， $\Delta T=3$ ；BT.1120 协议， $\Delta T=0.11T$ 。

图2-18 VO 级联输入接口时序图



VO 接口时序参数如表 2-48 所示。

表2-48 VO 级联输入接口时序参数表

参数	符号	最小值	典型值	最大值	单位
VI0CLK 时钟周期	T	-	13.5	-	ns
输入信号建立时间要求	Tsu	3	-	-	ns
输入信号保持时间要求	Thd	2	-	-	ns

2.10.6 PCI 接口时序

PCI 接口时序如图 2-19 和图 2-20 所示。



图2-19 PCI 接口时序(采用芯片内部时钟)

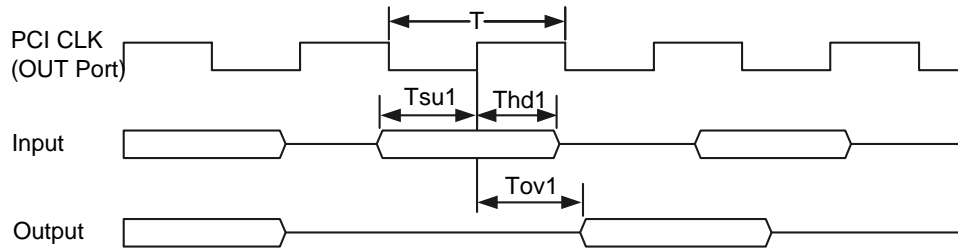
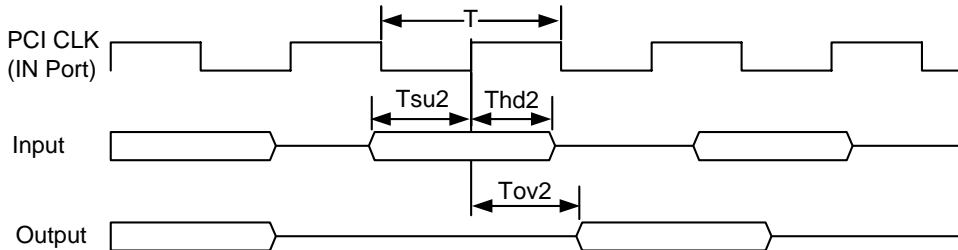


图2-20 PCI 接口时序(采用芯片外部时钟)



PCI 接口时序参数如表 2-48 所示。

表2-49 PCI 接口时序参数表

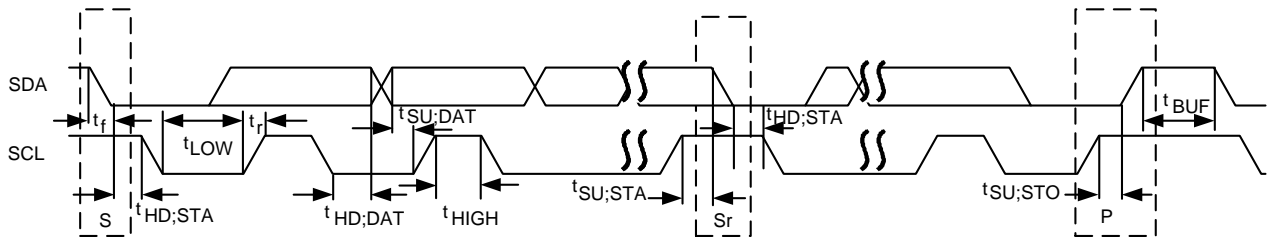
参数	符号	最小值	典型值	最大值	单位
PCI CLK 时钟周期	T	15.15	-	-	ns
输入信号建立时间	Tos1	5	-	-	ns
	Tos2	4.5	-	-	ns
输入信号保持时间	Toh1	0	-	-	ns
	Toh2	0	-	-	ns
输出信号延时	Tov1	4	-	6	ns
	Tov2	2	-	8	ns

2.10.7 I²C 接口时序

I²C 传输时序如图 2-21 所示。



图2-21 I²C 传输时序



I²C 接口时序参数如表 2-50 所示。

表2-50 I²C 接口时序参数表

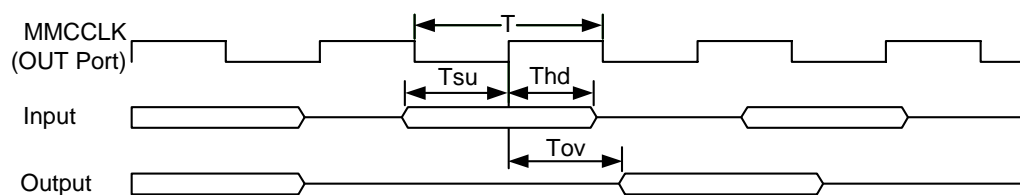
参数	符号	标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	
SCL (Serial Clock) 时钟频率	f_{SCL}	0	100	0	400	KHz
启动保持时间	$t_{HD;STA}$	4.0	-	0.6	-	μs
SCL 低电平周期	t_{LOW}	4.7	-	1.3	-	μs
SCL 高电平周期	t_{HIGH}	4.0	-	0.6	-	μs
启动建立时间	$t_{SU;STA}$	4.7	-	0.6	-	μs
数据保持时间	$t_{HD;DAT}$	0	3.45	0	0.9	μs
数据建立时间	$t_{SU;DAT}$	250	-	100	-	ns
SDA、SCL 上升时间	t_r	-	1000	$20+0.1C_b$	300	ns
SDA、SCL 下降时间	t_f	-	300	$20+0.1C_b$	300	ns
结束建立时间	$t_{SU;STO}$	4.0	-	0.6	-	μs
开始与结束之间的总线释放时间	t_{BUF}	4.7	-	1.3	-	μs
总线负载	C_b	-	400	-	400	pF
低电平噪声容限	V_{nL}	$0.1V_{DD}$	-	$0.1V_{DD}$	-	V
高电平噪声容限	V_{nH}	$0.2V_{DD}$	-	$0.2V_{DD}$	-	V

2.10.8 MMC 接口时序

MMC 接口时序如图 2-22 所示。



图2-22 MMC 接口时序



MMC 接口时序参数如表 2-51 所示。

表2-51 MMC 接口时序参数

参数	符号	最小值	典型值	最大值	单位
MMCCLK 时钟周期	T	20.8	-	-	ns
输入信号建立时间	T_{su}	5	-	-	ns
输入信号保持时间	T_{hd}	2.5	-	-	ns
输出信号延时	T_{ov}	6.5	-	14.3	ns

2.10.9 SPI 接口时序

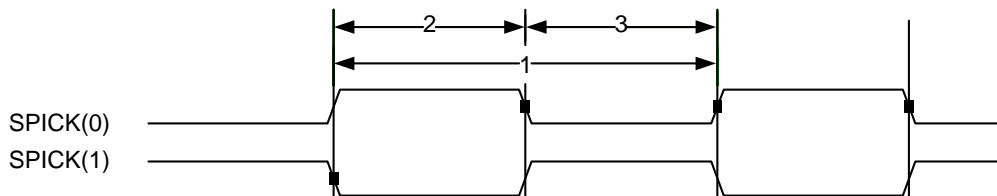
说明

图 2-23 ~ 图 2-25 中，以下缩略语或字母意义不变：

- MSB:Most Significant Bit
- LSB:Least Significant Bit
- SPICK(0):spo=0
- SPICK(1):spo=1

SPI 接口时钟时序如图 2-23 所示。

图2-23 SPICK 时序



SPI 主模式下接口时序分别如图 2-24 和图 2-25 所示。



图2-24 SPI 主模式下接口时序 (sph=0)

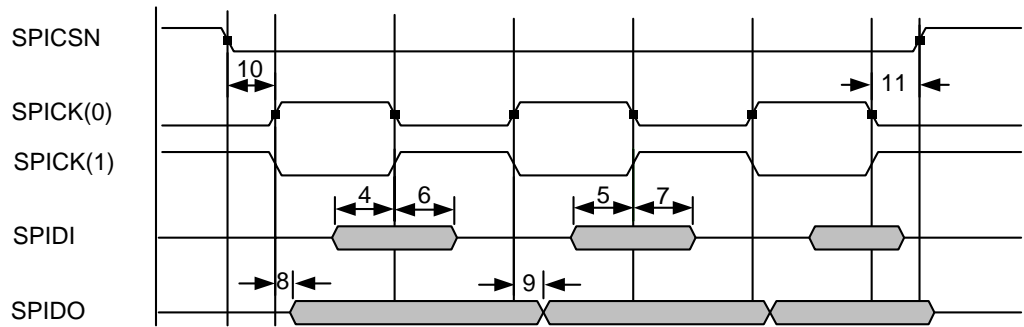
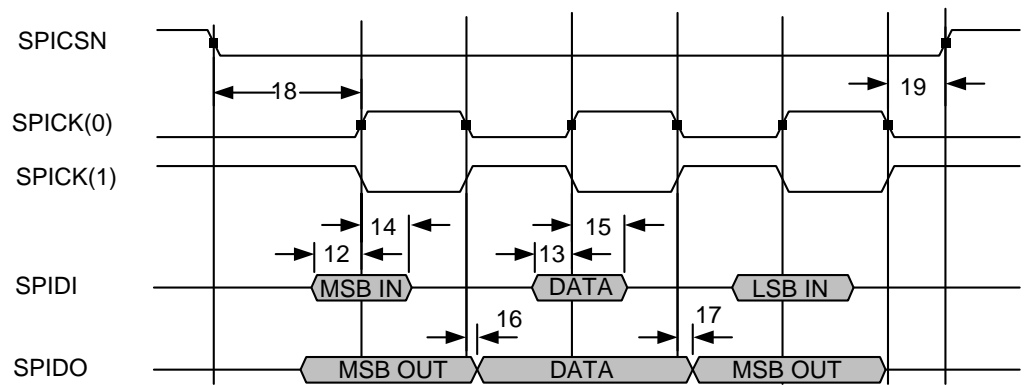


图2-25 SPI 主模式下接口时序 (sph=1)



SPI 接口时序参数如表 2-52 所示。

表2-52 SPI 接口时序参数

No	参数	符号	最小值	典型值	最大值	单位
1	Cycle time, SPICK	tc	-	-	-	ns
2	Pulse duration, SPICK high (All Master Modes)	tw1	-	-	-	ns
3	Pulse duration, SPICK low (All Master Modes)	tw2	-	-	-	ns
4	Setup time, SPIDI (input) valid before SPICK (output) falling edge	tsu1	-	-	-	ns
5	Setup time, SPIDI (input) valid before SPICK (output) rising edge	tsu2	-	-	-	ns
6	Hold time, SPIDI (input) valid after SPICK (output) falling edge	th1	-	-	-	ns



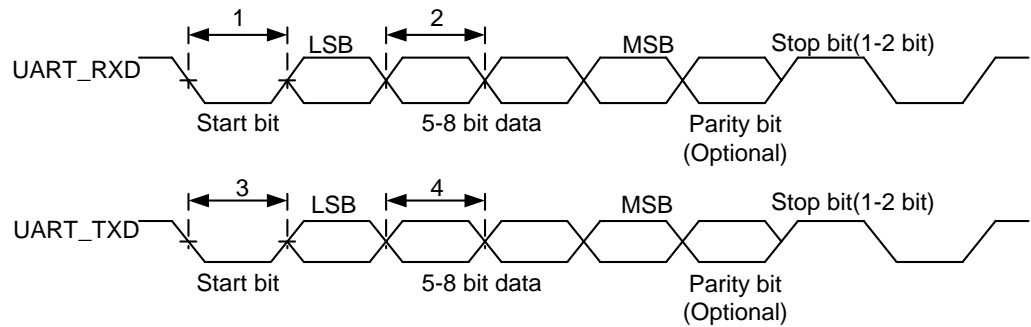
No	参数	符号	最小值	典型值	最大值	单位
7	Hold time, SPIDI (input) valid after SPICK (output) rising edge	th2	-	-	-	ns
8	Delay time, SPICK (output) rising edge to SPIDO (output) transition	td1	-	-	-	ns
9	Delay time, SPICK (output) falling edge to SPIDO (output) transition	td2	-	-	-	ns
10	Delay time, SPICSN (output) falling edge to first SPICK (output) rising or falling edge	td3	-	-	-	ns
11	Delay time, SPICK (output) rising or falling edge to SPICSN (output) rising edge	td4	-	-	-	ns
12	Setup time, SPIDI (input) valid before SPICK (output) rising edge	tsu3	-	-	-	ns
13	Setup time, SPIDI (input) valid before SPICK (output) falling edge	tsu4	-	-	-	ns
14	Hold time, SPIDI (input) valid after SPICK (output) rising edge	th3	-	-	-	ns
15	Hold time, SPIDI (input) valid after SPICK (output) falling edge	th4	-	-	-	ns
16	Delay time, SSP_SPICK (output) falling edge to SSP_SPIDO (output) transition	td5	-	-	-	ns
17	Delay time, SPICK (output) rising edge to SPIDO (output) transition	td6	-	-	-	ns
18	Delay time, SPICSN (output) falling edge to first SPICK (output) rising or falling edge	td7	-	-	-	ns
19	Delay time, SPICK (output) rising or falling edge to SPICSN (output) rising edge	td8	-	-	-	ns

2.10.10 UART 接口时序

UART 接口时序如图 2-26 所示。



图2-26 UART 接口时序



UART 接收数据信号 URXD 时序要求如表 2-53 所示。

表2-53 UART 接收数据信号时序参数

No	参数	符号	最小值	典型值	最大值	单位
1	脉冲宽度, 接收 Start Bit	UART_RXD	0.96U	-	1.05U	ns
2	脉冲宽度, 接收 Data Bit	UART_RXD	0.96U	-	1.05U	ns

注: U = UART baud time = 1/baud rate

UART 发送数据信号 UTXD 时序要求如表 2-54 所示。

表2-54 UART 发送数据信号时序参数

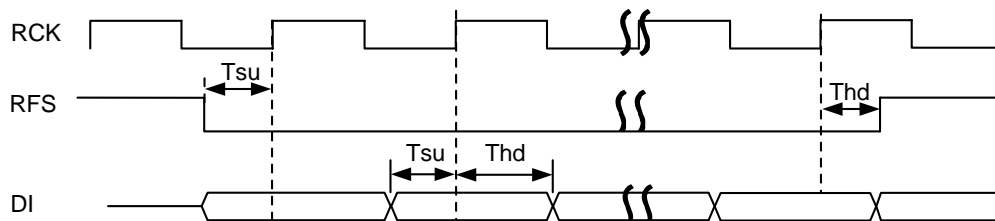
No	参数	符号	最小值	典型值	最大值	单位
3	脉冲宽度, 传送 Start Bit	UART_TXD	U-2	-	U+2	ns
4	脉冲宽度, 传送 Data Bit	UART_TXD	U-2	-	U+2	ns

注: U = UART baud time = 1/baud rate

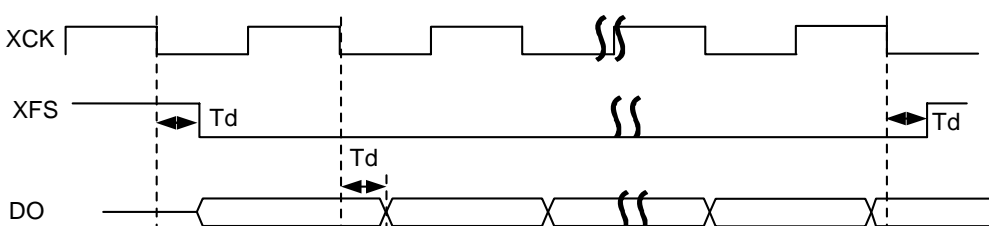
2.10.11 SIO 接口时序

I²S 模式接口时序

I²S 接口接收时序如图 2-27 所示。

图2-27 I²S 接口接收时序图

I²S 接口发送时序如图 2-28 所示。

图2-28 I²S 接口发送时序图

I²S 接口时序参数如表 2-55 所示。

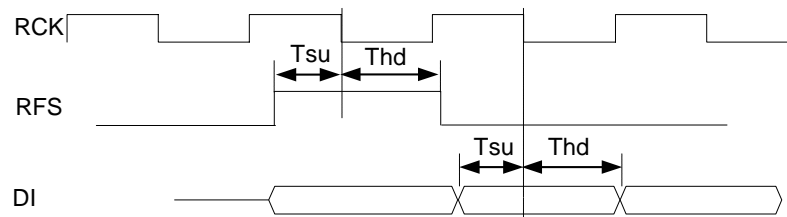
表2-55 I²S 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
输入信号建立时间	T_{su}	10	-	-	ns
输入信号保持时间	T_{hd}	10	-	-	ns
输出信号延时	T_d	0	-	8	ns

PCM 模式接口时序

PCM 接口接收时序如图 2-29 所示。

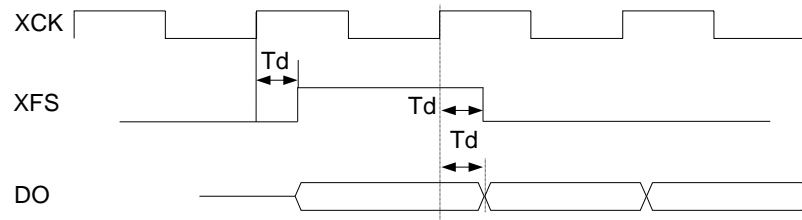
图2-29 PCM 接口接收时序图





PCM 接口发送时序如图 2-30 所示。

图2-30 PCM 接口发送时序图



PCM 接口时序参数如表 2-56 所示。

表2-56 PCM 接口时序参数表

参数	符号	最小值	典型值	最大值	单位
输入信号建立时间	T_{su}	10	-	-	ns
输入信号保持时间	T_{hd}	10	-	-	ns
输出信号延时	T_d	0	-	8	ns

2.10.12 SMI 接口时序

SMI 接口时序请参见“4.2.4 功能描述”中“功能原理”部分的内容。

2.11 封装和管脚分布

2.11.1 封装

Hi3520 芯片采用 PBGA 封装，封装尺寸为 27mm×27mm，管脚间距为 0.8mm。具体封装尺寸请参见图 2-31～图 2-35，尺寸参数请参见表 2-57。

图2-31 芯片尺寸视图（顶视图）

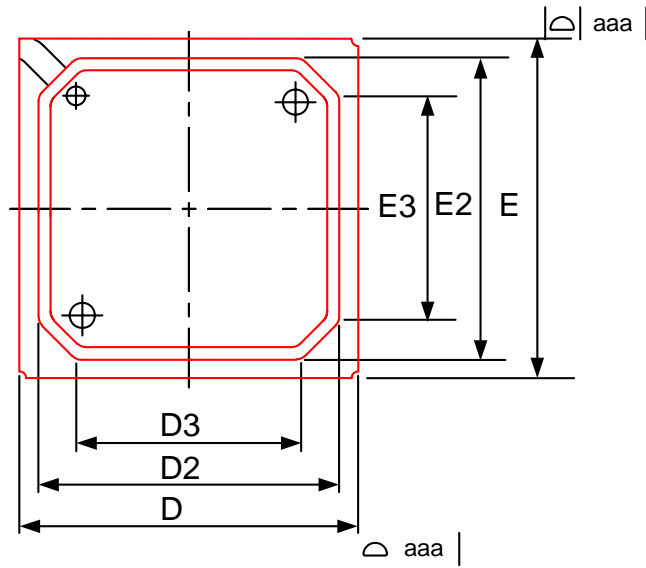


图2-32 芯片尺寸视图（底视图）

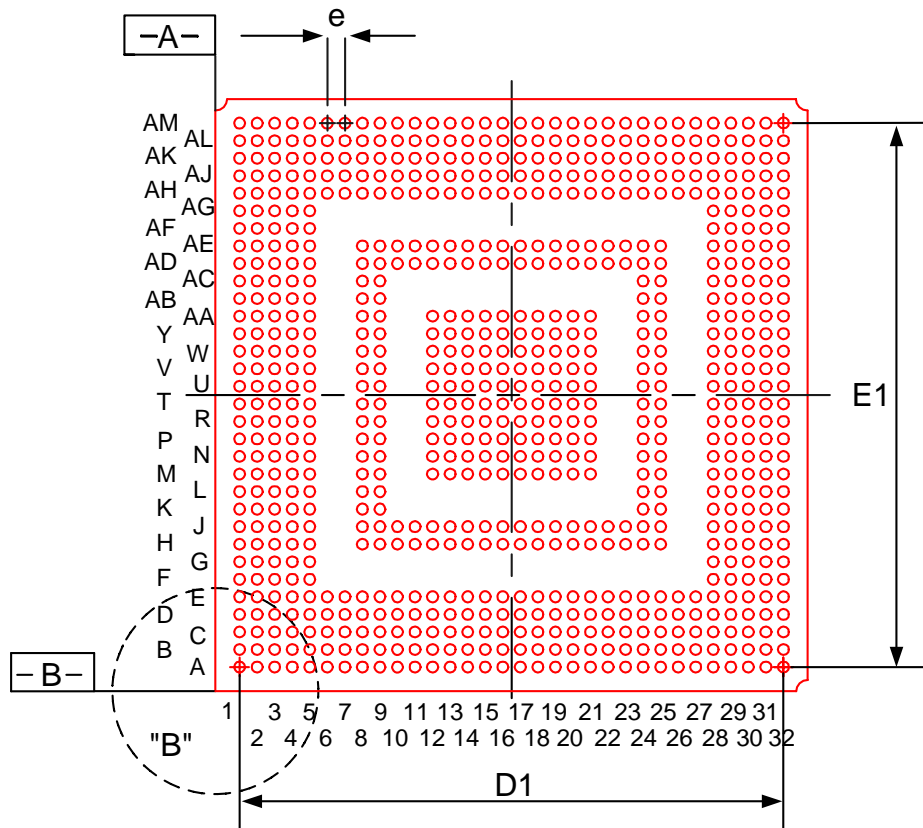




图2-33 Detail B 的放大图

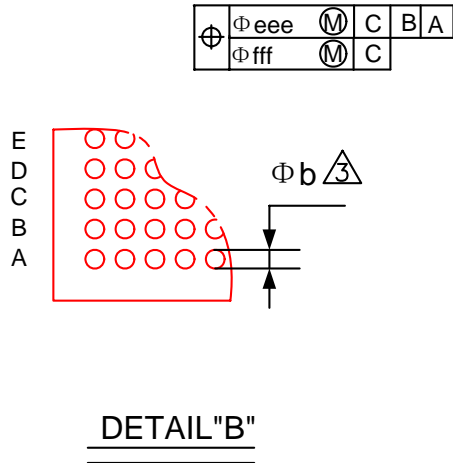


图2-34 芯片尺寸视图（侧视图）

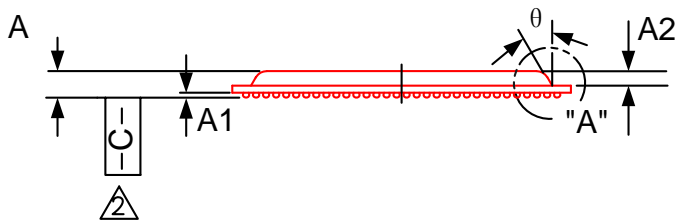


图2-35 Detail A 的放大图

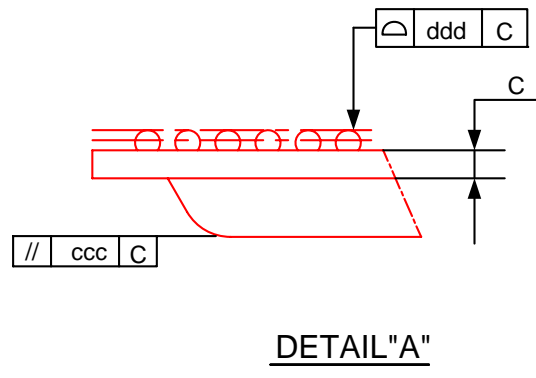


表2-57 封装参数说明表

参数	尺寸 (mm)		
	最小值	典型值	最大值
A	-	2.13	2.23



参数	尺寸 (mm)		
	最小值	典型值	最大值
A1	0.30	0.40	0.50
A2	1.12	1.17	1.22
b	0.40	0.50	0.60
c	0.51	0.56	0.61
D	26.80	27.00	27.20
D1	-	24.80	-
D2	23.80	24.00	24.20
D3	-	18.00	-
E	26.80	27.00	27.20
E1	-	24.80	-
E2	23.80	24.00	24.20
E3	--	18.00	-
e	-	0.80	-
aaa	0.20		
ccc	0.25		
ddd	0.15		
eee	0.25		
fff	0.10		
θ	30° TPY		

2.11.2 管脚分布

Hi3520 的管脚有 768 个，管脚数目统计表如表 2-58 所示。

表2-58 Hi3520 管脚数目统计表

管脚类别	数量
数字 I/O	334
数字电源	170
数字地	203
模拟 I/O	17



管脚类别	数量
模拟电源	18
模拟地	24
晶体时钟	2
总计	768

Hi3520 管脚分布图（顶视图）如图 2-37~图 2-40 所示。



说明

图 2-37 ~ 图 2-40 中的颜色示意如图 2-36 所示。

图2-36 管脚分布图中的颜色示意图

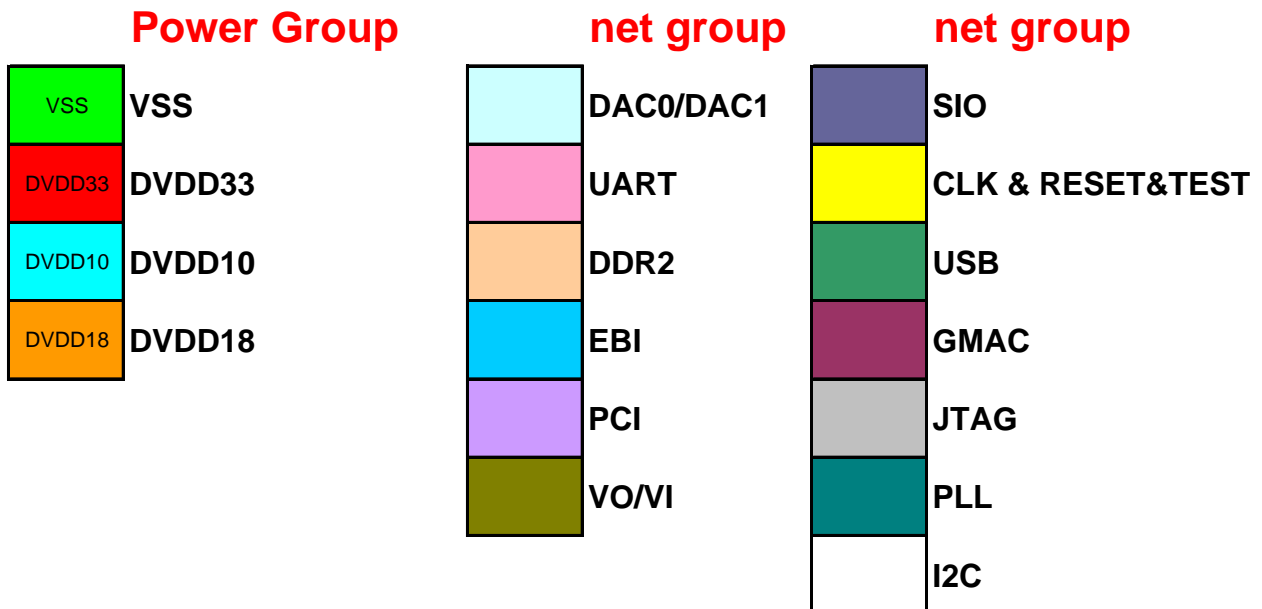


图2-37 Hi3520 管脚分布图 (1~16、A~T)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	VSS	DDRA_UD M1	DDRA_DQ 25	DVDD18	DDRA_DQ SP3	DDRA_DQ 29	DDRA_DQ 23	DDRA_DQ SP2	DVDD18	DDRA_AD R0	DDRA_AD R6	DDRA_AD R2	DDRA_CK P1	VSS	DDRA_CK P0	DDRA_BA 0
B	MDIO	VSS	DDRA_DQ 30	DVDD18	DDRA_DQ SN3	VSS	DDRA_DQ 22	DDRA_DQ SN2	DVDD18	VSS	DDRA_AD R4	DVDD18	DDRA_CK N1	VSS	DDRA_CK N0	VSS
C	GTxD2	MDCK	VSS	DDRA_DQ 28	DDRA_DQ 31	VSS	DDRA_DQ 16	DDRA_DQ 18	DDRA_DQ 21	VSS	DDRA_OD T	DVDD18	DDRA_AD R8	DDRA_WE N	DDRA_CK E	VSS
D	GCOL	GTxD3	GTxD0	VSS	DDRA_DQ 27	DDRA_DQ 24	DDRA_DQ 26	DDRA_DQ 17	DDRA_DQ 20	DDRA_DQ 19	DDRA_LD M1	DDRA_RA SN	DDRA_CA SN	DDRA_AD R11	DDRA_CS N	DDRA_AD R9
E	VSS	GTCLKOUT	GRXDV	GTxD1	VSS	VSS	DVDD18	DVDD18	VSS	VSS	VSS	DVDD18	DVDD18	VSS	VSS	DVDD18
F	GRCLK	GRXD1	VSS	GTCLK	VSS											
G	SCL	SDA	GCRS	GTxEN	DVDD33											
H	VSS	VSS	GRXD2	GRXD0	DVDD33											
J	USBDP0	USBDM0	VSS	GRXD3	DVDD33											
K	USBVSSA 33	USBVSSA 33	USBREXT	USBVSSA 33	USBVDD											
L	USBDP1	USBDM1	USBVSSA 33	USBVDDA 33	USBVSS											
M	USBVDDA 33	USBVDDA 33	USBVDDA 33	SIO0XFS	SIO0DI											
N	SIO2RCK	SIO0DO	SIO0RFS	SIO0XCK	SIO0RCK											
P	SIO1RFS	ACKOUT	VSS	SIO1DI	VSS											
R	VSS	SIO2RFS	SIO1RCK	SIO2DI	VSS											
T	VI2CK	VI2DAT0	VI2HS	VI2VS	DVDD33											

VSS	DVDD18	DVDD18	VSS	VREF	VSS	VREF	DVDD18	DVDD18
VSS	VSS	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18
DVDD33	DVDD33							
DVDD33	DVDD33							
DVDD33	DVDD33							
USBVDDA 33	USBVSSA 33							
USBVSSA 33	USBVDDA 33							
USBVDDA 33	USBVSSA 33							
DVDD33	DVDD33							

VSS	DVDD10	DVDD10	DVDD10	DVDD10
DVDD10	VSS	VSS	VSS	VSS
DVDD10	VSS	VSS	VSS	VSS
DVDD10	VSS	VSS	VSS	VSS
DVDD10	VSS	VSS	VSS	VSS



图2-38 Hi3520 管脚分布图（17~32、A~T）

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
DDRA_BA 1	DDRA_AD R1	DDRA_AD R13	VSS	DDRA_DQ SP1	DDRA_DQ 15	DDRA_DQ 7	DDRA_DQ SP0	VSS	DDRA_DQ 3	DVDD18	DDRB_DQ 14	VSS	DDRB_DQ SP1	DDRB_DQ 10	VSS	A
DDRA_BA 2	DDRA_AD R3	DVDD18	VSS	DDRA_DQ SN1	DDRA_DQ 10	DVDD18	DDRA_DQ SN0	VSS	DDRA_DQ 4	DVDD18	DDRB_DQ 12	VSS	DDRB_DQ SN1	DDRB_DQ 8	DDRB_DQ 13	B
DDRA_AD R7	DDRA_AD R10	DVDD18	DDRA_DQ 9	DDRA_DQ 12	DDRA_DQ 13	DVDD18	DDRA_DQ 0	DDRA_DQ 2	DDRA_DQ 5	VSS	DDRB_DQ 15	DDRB_DQ 9	DVDD18	VSS	VSS	C
DDRA_AD R12	DDRA_AD R5	DDRA_DQ 14	DDRA_DQ 11	DDRA_DQ 8	DDRA_UD M0	DDRA_DQ 6	DDRA_DQ 1	DDRA_LD M0	VSS	DDRB_UD M0	DDRB_DQ 11	DDRB_DQ 7	DDRB_DQ 6	DDRB_DQ SP0	DDRB_DQ SN0	D
DVDD18	DVDD18	VSS	VSS	VSS	DVDD18	DVDD18	VSS	VSS	DVDD18	DVDD18	VSS	DDRB_DQ 0	DVDD18	DVDD18	DDRB_DQ 3	E
											DDRB_DQ 1	DDRB_DQ 2	DDRB_DQ 4	DDRB_DQ 5	DDRB_OD T	F
											DVDD18	DDRB_LD M0	DDRB_AD R4	DDRB_AD R6	DDRB_CS N	G
DVDD18	VREF	VSS	VREF	VSS	VREF	VSS	VREF	VSS			DDRB_AD R0	DDRB_RA SN	VSS	VSS	DDRB_AD R2	H
DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	VSS	DVDD18			VSS	DDRB_AD R8	DDRB_CA SN	DDRB_CK P0	DDRB_CK N0	J
											VSS	DDRB_AD R11	DVDD18	DVDD18	DDRB_WE N	K
											DDRB_BA 1	DDRB_CK E	DDRB_AD R9	DDRB_BA 0	DDRB_AD R7	L
											DVDD18	DDRB_BA 2	VSS	VSS	DDRB_AD R3	M
											DDRB_AD R12	DDRB_AD R1	DDRB_AD R10	DDRB_AD R13	DDRB_AD R5	N
											VSS	VSS	DVDD18	DVDD18	VSS	P
											VSS_VPLL	TDI	TDO	TCK	TMS	R
											AVDD33_ VPLL	TRSTN	TESTMOD E	VSS	VSS	T

DVDD18	VREF	VSS	VREF	VSS	VREF	VSS	VREF	VSS
DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	DVDD18	VSS	DVDD18
							DVDD18	VREF
							DVDD18	VSS
							DVDD18	VREF
							DVDD18	VSS
							DVDD18	VREF
							DVDD18	VSS_VPLL
							VDD10_V PLL0	VDD10_V PLL1

DVDD10	DVDD10	DVDD10	DVDD10	VSS
VSS	VSS	VSS	VSS	DVDD10
VSS	VSS	VSS	VSS	DVDD10
VSS	VSS	VSS	VSS	DVDD10
VSS	VSS	VSS	VSS	DVDD10



图2-40 Hi3520 管脚分布图（17~32、U~AM）

VSS	VSS	VSS	VSS	DVDD10	VDD10_EPLL	VSS_EPLL	AVDD33_AEPLL	VO0DAT6	VO0DAT7	XIN_24	XOUT_24	U				
VSS	VSS	VSS	VSS	DVDD10	VDD10_APLL	VSS_APLL	VSS_APLL	VO0DAT5	VO0DAT4	VO0DAT3	VO0DAT2	V				
VSS	VSS	VSS	VSS	DVDD10	DVDD33	DVDD33	VSS	VO0DAT1	VO0DAT0	VORGBDV	VO0CK	W				
VSS	VSS	VSS	VSS	DVDD10	DVDD33	VSS	VSS	GPIO3_0	VO1DAT15	VO1DAT13	VO1DAT14	Y				
DVDD10	DVDD10	DVDD10	DVDD10	VSS	DVDD33	VSS	DVDD33	VO1DAT12	VSS	VO1CK	VO1DAT11	AA				
DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	VSS	VSS	DVDD33	VO1DAT6	VO1DAT9	VO1DAT8	VO1DAT10	AB	
VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	DVDD33	VSS	VSS	VSS	VO1DAT4	VO1DAT7	VO1DAT5	VSS	AC		
DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	DVDD33	VSS	VSS	VSS	VO1DAT0	VO1DAT2	VO1DAT1	VO1DAT3	AD		
VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	EBIADR21	EBIADR23	EBIADR22	EBIADR24	AE		
VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	EBIADR18	VSS	EBIADR20	EBIADR19	AF		
VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	DVDD33	VSS	VSS	VSS	EBIADR14	EBIADR16	EBIADR15	EBIADR17	AG		
VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	DVDD33	VSS	VSS	DVDD33	VSS	EBIADR11	EBIADR12	EBIADR13	VSS	AH	
PCIAD8	PCISERRN	PCIAD6	PCIAD2	PCIGRANT1N	PCIGRANT4N	PCIREQ3N	SPIOEN	SMICS0N	NFRB	EBIDQ2	EBIDQ4	EBIADR4	EBIADR9	EBIADR8	EBIADR10	AJ
PCIAD9	PCIPERRN	PCIAD7	VSS	PCIGRANT0N	PCIGRANT2N	PCIREQ2N	VSS	NFOEN	NFCS0N	VSS	EBIDQ3	EBIADR0	VSS	EBIADR7	EBIADR5	AK
PCIAD10	PCICBEN	PCIREQ0N	PCIAD5	PCIAD0	PCIGRANT3N	PCIREQ1N	EBIWEN	SMICS1N	EBIRDYN	EBIDQ0	EBIDQ1	EBIDQ5	EBIDQ7	EBIADR3	EBIADR6	AL
PCIAD11	VSS	PCIPAR	PCIAD4	PCIAD3	PCIAD1	VSS	PCIREQ4N	NFCLE	NFALE	NFCS1N	VSS	EBIDQ6	EBIADR1	EBIADR2	VSS	AM
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

2.11.3 管脚排列表

Hi3520 的管脚按位置排列如表 2-59 所示。

表2-59 管脚排列表

位置	管脚名称	位置	管脚名称
A1	VSS	U1	VI2DAT3
A2	DDRA_UDM1	U2	VI2DAT2
A3	DDRA_DQ25	U3	VSS
A4	DVDD18	U4	VI2DAT1
A5	DDRA_DQSP3	U5	DVDD33
A6	DDRA_DQ29	U8	DVDD33



位置	管脚名称	位置	管脚名称
A7	DDRA_DQ23	U9	DVDD33
A8	DDRA_DQSP2	U12	DVDD10
A9	DVDD18	U13	VSS
A10	DDRA_ADR0	U14	VSS
A11	DDRA_ADR6	U15	VSS
A12	DDRA_ADR2	U16	VSS
A13	DDRA_CKP1	U17	VSS
A14	VSS	U18	VSS
A15	DDRA_CKP0	U19	VSS
A16	DDRA_BA0	U20	VSS
A17	DDRA_BA1	U21	DVDD10
A18	DDRA_ADR1	U24	VDD10_EPLL
A19	DDRA_ADR13	U25	VSS_EPLL
A20	VSS	U28	AVDD33_AEPLL
A21	DDRA_DQSP1	U29	VO0DAT6
A22	DDRA_DQ15	U30	VO0DAT7
A23	DDRA_DQ7	U31	XIN_24
A24	DDRA_DQSP0	U32	XOUT_24
A25	VSS	V1	VI2DAT6
A26	DDRA_DQ3	V2	VI2DAT5
A27	DVDD18	V3	VI2DAT4
A28	DDRB_DQ14	V4	VI2DAT7
A29	VSS	V5	VSS
A30	DDRB_DQSP1	V8	VSS
A31	DDRB_DQ10	V9	DVDD33
A32	VSS	V12	DVDD10
B1	MDIO	V13	VSS
B2	VSS	V14	VSS
B3	DDRA_DQ30	V15	VSS
B4	DVDD18	V16	VSS
B5	DDRA_DQSN3	V17	VSS



位置	管脚名称	位置	管脚名称
B6	VSS	V18	VSS
B7	DDRA_DQ22	V19	VSS
B8	DDRA_DQSN2	V20	VSS
B9	DVDD18	V21	DVDD10
B10	VSS	V24	VDD10_APLL
B11	DDRA_ADR4	V25	VSS_APLL
B12	DVDD18	V28	VSS_APLL
B13	DDRA_CKN1	V29	VO0DAT5
B14	VSS	V30	VO0DAT4
B15	DDRA_CKN0	V31	VO0DAT3
B16	VSS	V32	VO0DAT2
B17	DDRA_BA2	W1	VSS
B18	DDRA_ADR3	W2	VI3CK
B19	DVDD18	W3	VI3DAT0
B20	VSS	W4	VI3DAT2
B21	DDRA_DQSN1	W5	VSS
B22	DDRA_DQ10	W8	VSS
B23	DVDD18	W9	DVDD33
B24	DDRA_DQSN0	W12	DVDD10
B25	VSS	W13	VSS
B26	DDRA_DQ4	W14	VSS
B27	DVDD18	W15	VSS
B28	DDRB_DQ12	W16	VSS
B29	VSS	W17	VSS
B30	DDRB_DQSN1	W18	VSS
B31	DDRB_DQ8	W19	VSS
B32	DDRB_DQ13	W20	VSS
C1	GTXD2	W21	DVDD10
C2	MDCK	W24	DVDD33
C3	VSS	W25	DVDD33
C4	DDRA_DQ28	W28	VSS



位置	管脚名称	位置	管脚名称
C5	DDRA_DQ31	W29	VO0DAT1
C6	VSS	W30	VO0DAT0
C7	DDRA_DQ16	W31	VORGBDV
C8	DDRA_DQ18	W32	VO0CK
C9	DDRA_DQ21	Y1	VI3DAT1
C10	VSS	Y2	VI3DAT4
C11	DDRA_ODT	Y3	VI3DAT5
C12	DVDD18	Y4	VI3DAT6
C13	DDRA_ADR8	Y5	DVDD33
C14	DDRA_WEN	Y8	DVDD33
C15	DDRA_CKE	Y9	DVDD33
C16	VSS	Y12	AVSS_DAC
C17	DDRA_ADR7	Y13	VSS
C18	DDRA_ADR10	Y14	VSS
C19	DVDD18	Y15	VSS
C20	DDRA_DQ9	Y16	VSS
C21	DDRA_DQ12	Y17	VSS
C22	DDRA_DQ13	Y18	VSS
C23	DVDD18	Y19	VSS
C24	DDRA_DQ0	Y20	VSS
C25	DDRA_DQ2	Y21	DVDD10
C26	DDRA_DQ5	Y24	DVDD33
C27	VSS	Y25	VSS
C28	DDRB_DQ15	Y28	VSS
C29	DDRB_DQ9	Y29	GPIO3_0
C30	DVDD18	Y30	VO1DAT15
C31	VSS	Y31	VO1DAT13
C32	VSS	Y32	VO1DAT14
D1	GCOL	AA1	VI3DAT3
D2	GTXD3	AA2	VI3DAT7
D3	GTXD0	AA3	VSS



位置	管脚名称	位置	管脚名称
D4	VSS	AA4	VI0HS
D5	DDRA_DQ27	AA5	DVDD33
D6	DDRA_DQ24	AA8	DVDD33
D7	DDRA_DQ26	AA9	DVDD33
D8	DDRA_DQ17	AA12	AVSS_DAC
D9	DDRA_DQ20	AA13	AVSS_DAC
D10	DDRA_DQ19	AA14	DVDD10
D11	DDRA_LDM1	AA15	DVDD10
D12	DDRA_RASN	AA16	DVDD10
D13	DDRA_CASN	AA17	DVDD10
D14	DDRA_ADR11	AA18	DVDD10
D15	DDRA_CSN	AA19	DVDD10
D16	DDRA_ADR9	AA20	DVDD10
D17	DDRA_ADR12	AA21	VSS
D18	DDRA_ADR5	AA24	DVDD33
D19	DDRA_DQ14	AA25	VSS
D20	DDRA_DQ11	AA28	DVDD33
D21	DDRA_DQ8	AA29	VO1DAT12
D22	DDRA_UDM0	AA30	VSS
D23	DDRA_DQ6	AA31	VO1CK
D24	DDRA_DQ1	AA32	VO1DAT11
D25	DDRA_LDM0	AB1	VI0VS
D26	VSS	AB2	VI0DAT0
D27	DDRB_UDM0	AB3	VI0DAT1
D28	DDRB_DQ11	AB4	VI0DAT3
D29	DDRB_DQ7	AB5	VSS
D30	DDRB_DQ6	AB8	AVDD33_DAC
D31	DDRB_DQSP0	AB9	AVSS_DAC
D32	DDRB_DQSN0	AB24	DVDD33
E1	VSS	AB25	DVDD33
E2	GTCLKOUT	AB28	DVDD33



位置	管脚名称	位置	管脚名称
E3	GRXDV	AB29	VO1DAT6
E4	GTXD1	AB30	VO1DAT9
E5	VSS	AB31	VO1DAT8
E6	VSS	AB32	VO1DAT10
E7	DVDD18	AC1	VSS
E8	DVDD18	AC2	VI0CK
E9	VSS	AC3	VI0DAT5
E10	VSS	AC4	VI0DAT4
E11	VSS	AC5	VSS
E12	DVDD18	AC8	VREFINDAC0
E13	DVDD18	AC9	DVSS10_DAC
E14	VSS	AC24	DVDD33
E15	VSS	AC25	DVDD33
E16	DVDD18	AC28	VSS
E17	DVDD18	AC29	VO1DAT4
E18	DVDD18	AC30	VO1DAT7
E19	VSS	AC31	VO1DAT5
E20	VSS	AC32	VSS
E21	VSS	AD1	VI0DAT2
E22	DVDD18	AD2	VI0DAT6
E23	DVDD18	AD3	VI0DAT7
E24	VSS	AD4	VI1DAT3
E25	VSS	AD5	DVDD33
E26	DVDD18	AD8	DVDD33_DAC0
E27	DVDD18	AD9	DVSS33_DAC
E28	VSS	AD10	DVDD10_DAC
E29	DDRB_DQ0	AD11	AVSS_DAC
E30	DVDD18	AD12	VSS
E31	DVDD18	AD13	DVDD33
E32	DDRB_DQ3	AD14	DVDD33
F1	GRCLK	AD15	DVDD33



位置	管脚名称	位置	管脚名称
F2	GRXD1	AD16	DVDD33
F3	VSS	AD17	DVDD33
F4	GTCLK	AD18	DVDD33
F5	VSS	AD19	DVDD33
F28	DDRB_DQ1	AD20	DVDD33
F29	DDRB_DQ2	AD21	DVDD33
F30	DDRB_DQ4	AD22	DVDD33
F31	DDRB_DQ5	AD23	DVDD33
F32	DDRB_ODT	AD24	VSS
G1	SCL	AD25	VSS
G2	SDA	AD28	VSS
G3	GCRS	AD29	VO1DAT0
G4	GTXEN	AD30	VO1DAT2
G5	DVDD33	AD31	VO1DAT1
G28	DVDD18	AD32	VO1DAT3
G29	DDRB_LDM0	AE1	VI1CK
G30	DDRB_ADR4	AE2	VI1DAT0
G31	DDRB_ADR6	AE3	VSS
G32	DDRB_CSN	AE4	VI1DAT2
H1	VSS	AE5	DVDD33
H2	VSS	AE8	DVSS33_DAC
H3	GRXD2	AE9	DVDD33_DAC1
H4	GRXD0	AE10	VREFINDAC1
H5	DVDD33	AE11	AVDD33_DAC
H8	VSS	AE12	VSS
H9	DVDD18	AE13	VSS
H10	DVDD18	AE14	DVDD33
H11	VSS	AE15	DVDD33
H12	VREF	AE16	VSS
H13	VSS	AE17	VSS
H14	VREF	AE18	DVDD33



位置	管脚名称	位置	管脚名称
H15	DVDD18	AE19	DVDD33
H16	DVDD18	AE20	VSS
H17	DVDD18	AE21	VSS
H18	VREF	AE22	DVDD33
H19	VSS	AE23	DVDD33
H20	VREF	AE24	VSS
H21	VSS	AE25	VSS
H22	VREF	AE28	DVDD33
H23	VSS	AE29	EBIADR21
H24	VREF	AE30	EBIADR23
H25	VSS	AE31	EBIADR22
H28	DDRB_ADR0	AE32	EBIADR24
H29	DDRB_RASN	AF1	VSS
H30	VSS	AF2	VI1DAT1
H31	VSS	AF3	VI1DAT4
H32	DDRB_ADR2	AF4	VI1DAT6
J1	USBDP0	AF5	DVDD33
J2	USBDM0	AF28	DVDD33
J3	VSS	AF29	EBIADR18
J4	GRXD3	AF30	VSS
J5	DVDD33	AF31	EBIADR20
J8	VSS	AF32	EBIADR19
J9	VSS	AG1	VI1DAT5
J10	DVDD18	AG2	VI1DAT7
J11	DVDD18	AG3	AVDD33_DAC
J12	DVDD18	AG4	RSETDAC0
J13	DVDD18	AG5	COMPDAC0
J14	DVDD18	AG28	VSS
J15	DVDD18	AG29	EBIADR14
J16	DVDD18	AG30	EBIADR16
J17	DVDD18	AG31	EBIADR15



位置	管脚名称	位置	管脚名称
J18	DVDD18	AG32	EBIADR17
J19	DVDD18	AH1	AVSS_DAC
J20	DVDD18	AH2	AVDD33_DAC
J21	DVDD18	AH3	AVSS_DAC
J22	DVDD18	AH4	COMPDAC1
J23	DVDD18	AH5	RSETDAC1
J24	VSS	AH6	AVSS_DAC
J25	DVDD18	AH7	AVDD33_DAC
J28	VSS	AH8	VSS
J29	DDRB_ADR8	AH9	VSS
J30	DDRB_CASN	AH10	DVDD33
J31	DDRB_CKP0	AH11	DVDD33
J32	DDRB_CKN0	AH12	VSS
K1	USBVSSA33	AH13	VSS
K2	USBVSSA33	AH14	DVDD33
K3	USBREXT	AH15	DVDD33
K4	USBVSSA33	AH16	VSS
K5	USBVDD	AH17	VSS
K8	DVDD33	AH18	DVDD33
K9	DVDD33	AH19	DVDD33
K24	DVDD18	AH20	VSS
K25	VREF	AH21	VSS
K28	VSS	AH22	DVDD33
K29	DDRB_ADR11	AH23	DVDD33
K30	DVDD18	AH24	VSS
K31	DVDD18	AH25	VSS
K32	DDRB_WEN	AH26	DVDD33
L1	USBDP1	AH27	DVDD33
L2	USBDM1	AH28	VSS
L3	USBVSSA33	AH29	EBIADR11
L4	USBVDDA33	AH30	EBIADR12



位置	管脚名称	位置	管脚名称
L5	USBVSS	AH31	EBIADR13
L8	DVDD33	AH32	VSS
L9	DVDD33	AJ1	DACVGA0B
L24	DVDD18	AJ2	AVSS_DAC
L25	VSS	AJ3	AVDD33_DAC
L28	DDRB_BA1	AJ4	AVSS_DAC
L29	DDRB_CKE	AJ5	AVSS_DAC
L30	DDRB_ADR9	AJ6	RSTN
L31	DDRB_BA0	AJ7	URXD1
L32	DDRB_ADR7	AJ8	PCIRSTN
M1	USBVDDA33	AJ9	PCIFRAMEN
M2	USBVDDA33	AJ10	PCIAD28
M3	USBVDDA33	AJ11	PCIAD24
M4	SIO0XFS	AJ12	PCIAD22
M5	SIO0DI	AJ13	PCIAD19
M8	DVDD33	AJ14	PCIIRDYN
M9	DVDD33	AJ15	PCIAD15
M12	VSS	AJ16	PCIAD12
M13	DVDD10	AJ17	PCIAD8
M14	DVDD10	AJ18	PCISERRN
M15	DVDD10	AJ19	PCIAD6
M16	DVDD10	AJ20	PCIAD2
M17	DVDD10	AJ21	PCIGRANT1N
M18	DVDD10	AJ22	PCIGRANT4N
M19	DVDD10	AJ23	PCIREQ3N
M20	DVDD10	AJ24	SMIOEN
M21	VSS	AJ25	SMICS0N
M24	DVDD18	AJ26	NFRB
M25	VREF	AJ27	EBIDQ2
M28	DVDD18	AJ28	EBIDQ4
M29	DDRB_BA2	AJ29	EBIADR4



位置	管脚名称	位置	管脚名称
M30	VSS	AJ30	EBIADR9
M31	VSS	AJ31	EBIADR8
M32	DDRB_ADR3	AJ32	EBIADR10
N1	SIO2RCK	AK1	DACVGA0G
N2	SIO0DO	AK2	AVSS_DAC
N3	SIO0RFS	AK3	AVDD33_DAC
N4	SIO0XCK	AK4	AVDD33_DAC
N5	SIO0RCK	AK5	URXD0
N8	USBVDDA33	AK6	UTXD0
N9	USBVSSA33	AK7	WDGRST
N12	DVDD10	AK8	VSS
N13	VSS	AK9	PCIAD31
N14	VSS	AK10	PCIAD27
N15	VSS	AK11	PCICBE0
N16	VSS	AK12	VSS
N17	VSS	AK13	PCIAD18
N18	VSS	AK14	PCITRDYN
N19	VSS	AK15	PCIDEVSELN
N20	VSS	AK16	VSS
N21	DVDD10	AK17	PCIAD9
N24	DVDD18	AK18	PCIPERRN
N25	VSS	AK19	PCIAD7
N28	DDRB_ADR12	AK20	VSS
N29	DDRB_ADR1	AK21	PCIGRANT0N
N30	DDRB_ADR10	AK22	PCIGRANT2N
N31	DDRB_ADR13	AK23	PCIREQ2N
N32	DDRB_ADR5	AK24	VSS
P1	SIO1RFS	AK25	NFOEN
P2	ACKOUT	AK26	NFCS0N
P3	VSS	AK27	VSS
P4	SIO1DI	AK28	EBIDQ3



位置	管脚名称	位置	管脚名称
P5	VSS	AK29	EBIADR0
P8	USBVSSA33	AK30	VSS
P9	USBVDDA33	AK31	EBIADR7
P12	DVDD10	AK32	EBIADR5
P13	VSS	AL1	DACVGA0R
P14	VSS	AL2	AVSS_DAC
P15	VSS	AL3	AVSS_DAC
P16	VSS	AL4	AVSS_DAC
P17	VSS	AL5	AVDD33_DAC
P18	VSS	AL6	URTSN1
P19	VSS	AL7	UTXD1
P20	VSS	AL8	PCIINTAN
P21	DVDD10	AL9	PCIAD30
P24	DVDD18	AL10	PCIAD26
P25	VREF	AL11	PCICBE1
P28	VSS	AL12	PCIAD21
P29	VSS	AL13	PCIAD17
P30	DVDD18	AL14	PCIIDSEL
P31	DVDD18	AL15	PCISTOPN
P32	VSS	AL16	PCIAD13
R1	VSS	AL17	PCIAD10
R2	SIO2RFS	AL18	PCICBE3
R3	SIO1RCK	AL19	PCIREQ0N
R4	SIO2DI	AL20	PCIAD5
R5	VSS	AL21	PCIAD0
R8	USBVDDA33	AL22	PCIGRANT3N
R9	USBVSSA33	AL23	PCIREQ1N
R12	DVDD10	AL24	EBIWEN
R13	VSS	AL25	SMICS1N
R14	VSS	AL26	EBIRDYN
R15	VSS	AL27	EBIDQ0



位置	管脚名称	位置	管脚名称
R16	VSS	AL28	EBIDQ1
R17	VSS	AL29	EBIDQ5
R18	VSS	AL30	EBIDQ7
R19	VSS	AL31	EBIADR3
R20	VSS	AL32	EBIADR6
R21	DVDD10	AM1	AVSS_DAC
R24	DVDD18	AM2	DACVGA1R
R25	VSS_VPLL	AM3	DACVGA1G
R28	VSS_VPLL	AM4	DACVGA1B
R29	TDI	AM5	AVSS_DAC
R30	TDO	AM6	UCTSN1
R31	TCK	AM7	PCICLK
R32	TMS	AM8	VSS
T1	VI2CK	AM9	PCIAD29
T2	VI2DAT0	AM10	PCIAD25
T3	VI2HS	AM11	PCIAD23
T4	VI2VS	AM12	PCIAD20
T5	DVDD33	AM13	VSS
T8	DVDD33	AM14	PCIAD16
T9	DVDD33	AM15	PCICBE2
T12	DVDD10	AM16	PCIAD14
T13	VSS	AM17	PCIAD11
T14	VSS	AM18	VSS
T15	VSS	AM19	PCIPAR
T16	VSS	AM20	PCIAD4
T17	VSS	AM21	PCIAD3
T18	VSS	AM22	PCIAD1
T19	VSS	AM23	VSS
T20	VSS	AM24	PCIREQ4N
T21	DVDD10	AM25	NFCLE
T24	VDD10_VPLL0	AM26	NFALE



位置	管脚名称	位置	管脚名称
T25	VDD10_VPLL1	AM27	NFCS1N
T28	AVDD33_VPLL	AM28	VSS
T29	TRSTN	AM29	EBIDQ6
T30	TESTMODE	AM30	EBIADR1
T31	VSS	AM31	EBIADR2
T32	VSS	AM32	VSS



3 系统控制

3.1 复位

3.1.1 概述

复位管理模块对整个芯片的复位、各功能模块的复位进行统一的管理，包括：

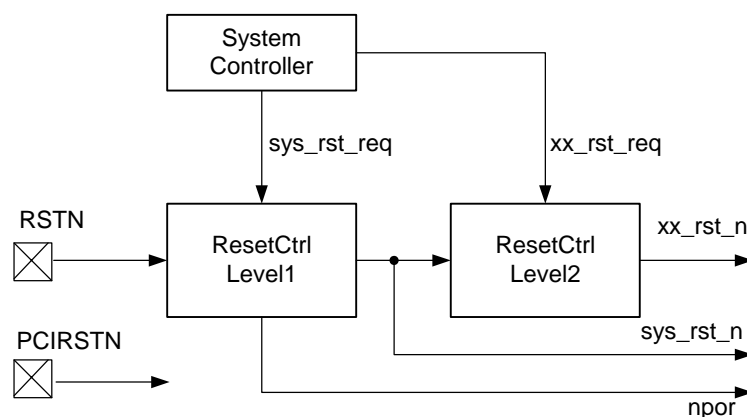
- 上电复位的管理和控制
- 系统软复位、功能模块单独软复位控制
- 复位信号同步到各模块对应时钟域

复位管理模块生成芯片内部各功能模块的复位信号。

3.1.2 复位控制

复位信号控制如图 3-1 所示。

图3-1 复位信号控制图



RSTN: 上电复位信号，源自芯片管脚 RSTN 输入。
PCIRSTN: PCI 总线复位信号，源自芯片管脚 PCIRSTN 输入。
sys_rst_req: 全局软复位请求信号，源自系统控制器。
xx_rst_req: 子模块单独软复位请求信号，源自系统控制器。
xx_rst_n、sys_rst_n、npor: 复位信号。



表3-1 复位信号分类表

复位信号类型	产生方式	用途
全局硬复位 npor	来自复位管脚 RSTN	对整芯片进行全局复位。
全局软复位 sys_rst_n	软件配置系统控制器的全局软复位寄存器	对整芯片中除了时钟复位电路和测试电路的所有模块进行全局复位。
子模块复位 xx_rst_n	软件配置系统控制寄存器的子模块复位控制寄存器	芯片各子模块的单独复位。
PCI 总线复位 PCIRSTN	来自 PCI 的管脚 PCIRSTN	当 Hi3520 处于从模式时，此管脚复位能够复位 PCI 子模块。

3.1.3 复位配置

上电复位

RSTN 是 Hi3520 芯片的功能复位输入 IO，完成上电复位过程必须同时满足以下条件：

- 上电复位 IO 输入一个低电平脉冲。
- 晶振时钟输入管脚 XIN24 输入的时钟稳定。
- 输入的上电复位信号低电平维持时间大于 12 个 XIN24 晶振时钟周期。

系统复位

实现系统复位有两种途径：

- 上电复位。
- 全局软复位，通过系统控制器 SC_SYSSTAT 控制。

软复位

软复位控制通过配置相应的系统控制器来实现，具体配置请参见 SC_SYSSTAT 的描述。



注意

- 系统软复位请求发出后，电路必须等待至少 360 个系统时钟周期才完成复位撤消。
- 各模块单独软复位不会自动撤消，比如某模块的复位是配置 1 时，模块处于复位状态，那么必须再配置为 0，该模块复位才会撤消。
- USB 2.0 Host PHY 和控制器 MMC、ETH、VI、VO 和 RTC 这些模块上电时默认均处于复位状态，正常工作前必须先撤销复位。



3.2 时钟

3.2.1 概述

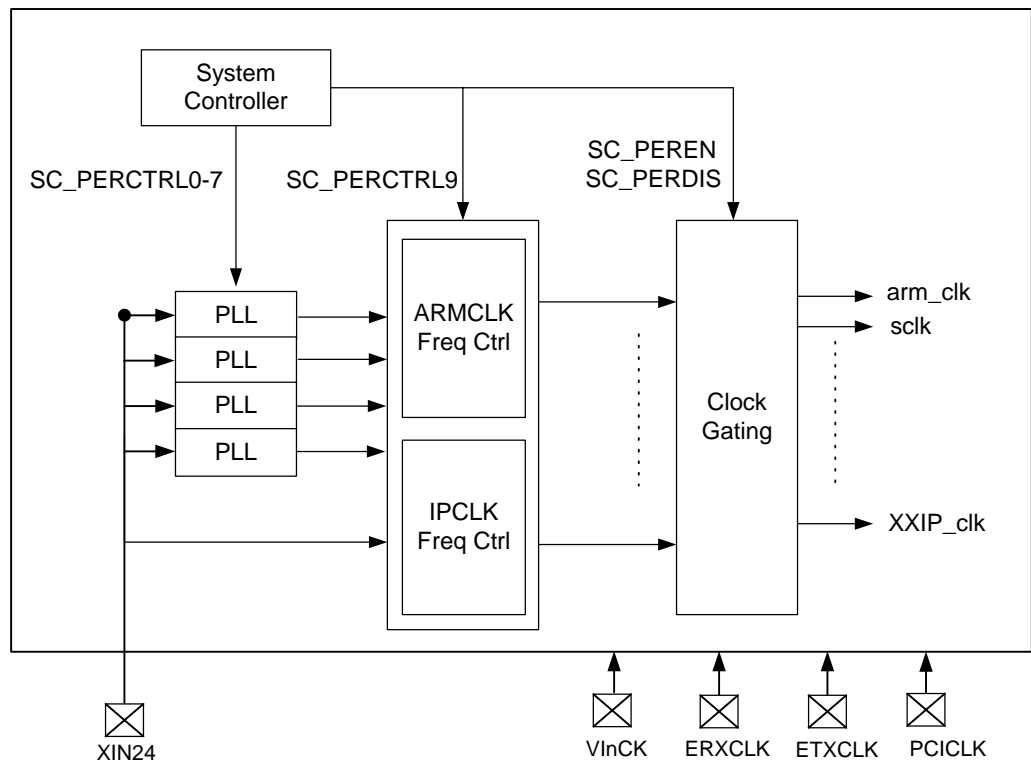
时钟管理模块对芯片时钟输入、时钟生成和控制进行统一的管理，包括：

- 时钟输入的管理和控制
- 时钟分频和控制
- 生成各模块的工作时钟

3.2.2 时钟控制框图

时钟管理模块功能框图如图 3-2 所示。

图3-2 时钟管理模块功能框图



注：图中的 VInCK 中的 n 取值范围为 0~3。

时钟管理模块有以下两部分输入：

- 源自芯片管脚的时钟输入：XIN24、VI0CK、VI1CK、VI2CK、VI3CK、ERXCLK、ETXCLK 和 PCICKL。
 - XIN24 为 PLL 输入时钟和 USB 2.0 Host PHY 输入时钟
 - VI0CK、VI1CK、VI2CK、VI3CK 为视频输入时钟
 - ERXCLK、ETXCLK 为 ETH 模块接口时钟



- PCICLK 为 PCI 模块接口时钟
- 源自系统控制器的时钟控制寄存器。
 - PLL 频率配置
 - IP 时钟频率配置
 - 时钟门控配置

时钟管理模块功能主体主要包括三部分：

- PLL 单元：用于产生 ARM 和总线时钟以及其它外设所需时钟。
- ARM 频率控制单元 ARMCLK Freq Ctrl 和模块时钟频率控制单元 IPCLK Freq Ctrl。
- 时钟门控管理单元 Clock Gating。

3.2.3 时钟配置

PLL 配置寄存器

Hi3520 内部使用了四个 PLL，每个 PLL 使用两组配置寄存器，如表 3-2 所示。每组配置寄存器的具体描述请参见系统控制器 SC_PERCTRL0~SC_PERCTRL7。

表3-2 Hi3520 PLL 配置寄存器

PLL	配置寄存器	备注
APLL	SC_PERCTRL0	APLL 主要用于输出 ARM 和总线时钟。
	SC_PERCTRL1	
EPLL	SC_PERCTRL2	EPLL 主要用于输出 ETH 的时钟。
	SC_PERCTRL3	
VPLL0	SC_PERCTRL4	VPLL0 主要用于产生 VO0 模块时钟。
	SC_PERCTRL5	
VPLL1	SC_PERCTRL6	VPLL1 主要用于产生 VO1 模块时钟。
	SC_PERCTRL7	

所有 PLL 采用管脚 XIN24 输入的晶振时钟作为输入时钟（24MHz），每个 PLL 可以输出多个时钟频率，如表 3-3 所示。

表3-3 Hi3520 PLL 各端口频率计算方法

PLL Pin	计算方法描述	注意事项
FREF	24MHz	Hi3520 要求接 24MHz 晶振时钟



PLL Pin	计算方法描述	注意事项
FOUTVCO	$FREF \times (fbdiv + frac/2^{24}) / refdiv$	PLL VCO 工作频率，要求在 800MHz（含）到 2.4GHz（含）之间。当 PLL 处于整数分频模式时，frac=0。
FOUTPOSTDIV	$FOUTVCO / pstdiv1 \times pstdiv2$	-
FOUT1ph0	$FOUTVCO / (pstdiv1 \times 2)$	-
FOUT2	$FOUTVCO / (pstdiv1 \times 4)$	-
FOUT3	$FOUTVCO / (pstdiv1 \times 6)$	-
FOUT4	$FOUTVCO / (pstdiv1 \times 8)$	-

Hi3520 使用的四个 PLL 中的 APLL 和 EPLL 的配置值固定，VPLL0 和 VPLL1 根据场景不同可配置不同的工作频率。

APLL 使用 FOUT2、FOUT3 和 FOUT4，典型场景下要求的频率依次为 600MHz、400MHz 和 300MHz，推荐配置如表 3-4 所示，即 SC_PERCTRL0=0x8900_0000、SC_PERCTRL1=0x006C_1064。

表3-4 Hi3520 APLL 配置参数

bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL0[31]	dsmpd	0: 小数分频模式; 1: 整数分频模式。	0b1
SC_PERCTRL0 [30]	bypass	0: 非旁路 (no bypass) ; 1: 旁路 (bypass) 。	0b0
SC_PERCTRL0[29:27]	postdiv2	PLL 第二级输出分频系数。	0b001
SC_PERCTRL0[26:24]	postdiv1	PLL 第一级输出分频系数。	0b001
SC_PERCTRL0[23:0]	frac	PLL 小数分频系数。	0x00_0000
SC_PERCTRL1[22]	reset	PLL 复位控制。 0: 复位; 1: 不复位。	0b1
SC_PERCTRL1[21]	pd	PLL Power Down 控制。 0: 关闭; 1: 打开。	0b1



bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL1[20]	foutvcopd	PLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b0
SC_PERCTRL1[19]	postdivpd	PLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL1[18]	fout4phasepd	PLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL1[17:12]	refdiv	PLL 参考时钟分频系数。	0x01
SC_PERCTRL1[11:0]	fbdiv	PLL 整数倍频系数。	0x064

EPLL 使用 FOUTPOSTDIV、FOUT1ph0、FOUT2 和 FOUT4，典型场景下要求的频率依次为 500MHz、250MHz、125MHz 和 62.5MHz，推荐配置如表 3-5 所示，即 SC_PERCTRL6=0x8B00_0000 和 SC_PERCTRL7=0x006C_207D。

表3-5 Hi3520 EPLL 配置参数

bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL6[31]	dsmpd	0: 小数分频模式; 1: 分数分频模式。	0x1
SC_PERCTRL6[30]	bypass	0: 非旁路 (no bypass); 1: 旁路 (bypass)。	0x0
SC_PERCTRL6[29:27]	postdiv2	PLL 第二级输出分频系数。	0x1
SC_PERCTRL6[26:24]	postdiv1	PLL 第一级输出分频系数。	0x3
SC_PERCTRL6[23:0]	frac	PLL 小数分频系数。	0x00_0000
SC_PERCTRL7[22]	reset	PLL 复位控制。 0: 复位; 1: 不复位。	0x1
SC_PERCTRL7[21]	pd	PLL Power Down 控制。 0: 关闭; 1: 打开。	0x1



bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL7[20]	foutvcopd	PLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。	0x0
SC_PERCTRL7[19]	postdivpd	PLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。	0x1
SC_PERCTRL7[18]	vfout4phasepd	PLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。	0x1
SC_PERCTRL7[17:12]	refdiv	PLL 参考时钟分频系数。	0x2
SC_PERCTRL7[11:0]	fbdiv	PLL 整数倍频系数。	0x07D

VPLL0 和 VPLL1 均使用的 PLL 的 FOUTPOSTDIV 作为频率输出端口。根据业务需要的频率，配置 SC_PERCTRL2~SC_PERCTRL5，推荐配置如表 3-6 和表 3-7 所示。

表3-6 Hi3520 VPLL0 配置参数

bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL2[31]	dsmpd	0: 小数分频模式; 1: 分数分频模式。	场景决定
SC_PERCTRL2[30]	bypass	0: 非旁路 (no bypass) ; 1: 旁路 (bypass) 。	0b0
SC_PERCTRL2[29:27]	postdiv2	PLL 第二级输出分频系数。	场景决定
SC_PERCTRL2[26:24]	postdiv1	PLL 第一级输出分频系数。	场景决定
SC_PERCTRL2 [23:0]	frac	PLL 小数分频系数。	场景决定
SC_PERCTRL3[22]	reset	PLL 复位控制。 0: 复位; 1: 不复位。	0b1
SC_PERCTRL3 [21]	pd	PLL Power Down 控制。 0: 关闭; 1: 打开。	0b1



bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL3 [20]	foutvcopd	PLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b0
SC_PERCTRL3[19]	postdivpd	PLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL3[18]	vfout4phasepd	PLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL3[17:12]	refdiv	PLL 参考时钟分频系数。	场景决定
SC_PERCTRL3 [11:0]	fbdiv	PLL 整数倍频系数。	场景决定

表3-7 Hi3520 VPLL1 配置参数

bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL4[31]	dsmpd	0: 小数分频模式; 1: 分数分频模式。	场景决定
SC_PERCTRL4[30]	bypass	0: 非旁路 (no bypass) ; 1: 旁路 (bypass) 。	0b0
SC_PERCTRL4[29:27]	postdiv2	PLL 第二级输出分频系数。	场景决定
SC_PERCTRL4[26:24]	postdiv1	PLL 第一级输出分频系数。	场景决定
SC_PERCTRL4[23:0]	frac	PLL 小数分频系数。	场景决定
SC_PERCTRL5[22]	reset	PLL 复位控制。 0: 复位; 1: 不复位。	0b1
SC_PERCTRL5[21]	pd	PLL Power Down 控制。 0: 关闭; 1: 打开。	0b1



bit 位	PLL Pin	意义描述	推荐配置值
SC_PERCTRL5[20]	foutvcopd	PLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b0
SC_PERCTRL5[19]	postdivpd	PLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL5[18]	vfout4phasepd	PLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。	0b1
SC_PERCTRL5[17:12]	refdiv	PLL 参考时钟分频系数。	场景决定
SC_PERCTRL5 [11:0]	fbdiv	PLL 整数倍频系数。	场景决定

当确定业务需要的时钟频率后，根据此频率值推算出 PLL 的配置值。以配置 VPLL1 为例，VPLL1 输出 FOUTPOSTDIV 给 VO1 模块，VO1 模块需要工作时钟 74.25MHz，则配置寄存器的过程如下：

- 设定 postdiv2=4, postdiv1=5, 则 $FOUTVCO=74.25 \times 4 \times 5=1485\text{MHz}$;
- 设定 refdiv=1, 则 $24 \times (\text{fbdiv} + \text{frac}/2^24)=1485\text{MHz}$ 。

由上面的条件推出：fbdiv=61, frac=14680064。

若 VPLL0 需要输出 54MHz, VPLL1 需要输出 108MHz, 则配置如下：

- SC_PERCTRL2=0xB400_0000
- SC_PERCTRL3=0x006C_1036
- SC_PERCTRL4=0x9C00_0000
- SC_PERCTRL5=0x006C_1036

ARM/SCLK/HCLK/PCLK 频率配置

ARM/SCLK/HCLK/PCLK 频率配置方法如表 3-8 所示。

表3-8 ARM/SCLK/HCLK/PCLK 频率配置

信号名	描述
sysmode[3:0]	ARM 频率切换。 默认为 0xB, 为晶振模式。可通过配置 SC_CTRL [2:0]控制该信号。



信号名	描述
sleep_mode	除 SCLK/IR 模块/CRG (Clock Reset Generation) 模块以外所有时钟关闭, 系统进入睡眠模式。可通过配置 SC_CTRL bit[2:0]控制该信号。
arm0freq_sel	ARM1196 频率模式。可通过配置 SC_PERCTRL9 bit[1:0]控制该信号。

系统控制器的状态和时钟切换的对应关系如表 3-9 所示。

表3-9 系统控制器状态和时钟切换对应关系

系统控制器状态	45kHz 时钟使能状态	24MHz 晶振使能状态	ARMPLL 使能状态	系统时钟状态
NORMAL	使能	使能	使能	ARM 子系统的工作时钟都来自 PLL 输出。
SLOW	使能	使能	不使能	ARM 子系统的工作时钟都来自 24MHz 晶振输入。
DOZE	使能	使能	不使能	ARM 子系统的工作时钟都来自 24MHz 晶振时钟分频得到的 45KHz 时钟。
SLEEP	使能	使能	不使能	除系统控制器和红外模块工作在 45KHz 外, 其它模块的时钟都处于关闭状态。

模块时钟频率配置

MMC 时钟频率配置方法如表 3-10 所示。

表3-10 MMC 时钟频率配置

信号名	描述
mmcclk_sel	MMC 模块工作时钟频率控制, 可通过配置 SC_PERCTRL9 bit[21:20]控制该信号。
mmcsap_sel	MMC 模块采样卡数据的时钟正反相控制, 可通过配置 SC_PERCTRL bit[22]控制该信号。



PCI 钟频率配置方式如表 3-11 所示。

表3-11 PCI 时钟频率配置

信号名	描述
pciclk_sel[3:0]	PCI 模块的工作时钟频率控制，可通过配置 SC_PERCTRL9 bit[24:23]控制该信号。

SMI 时钟频率配置方式如表 3-12 所示。

表3-12 SMI 时钟频率配置

信号名	描述
ssmcclk_sel	SMI 模块工作时钟频率选择，提供 AHB 总线的 1 或 2 分频。可通过配置 SC_PERCTRL9 bit[3]控制该信号。

VO 输出时钟配置方式如表 3-13 所示。

表3-13 VO 时钟频率配置

信号名	描述
vo0out_sel	VO0 模块 BT656 输出时钟相位选择，提供 VO0 工作时钟的正反相输出。可通过配置 SC_PERCTRL9 bit[18]控制该信号。
vo1out_sel	VO1 模块 BT1120 输出时钟相位选择，提供 VO1 工作时钟的正反相输出。可通过配置 SC_PERCTRL9 bit[19]控制该信号。
vosd_sel	VO_SD 部分 DATE 时钟与 VO 时钟的分频比控制。可通过配置 SC_PERCTRL9 bit[15:14]控制该信号。
vohd_sel	VO_HD 部分 DATE 时钟与 VO 时钟的分频比控制。可通过配置 SC_PERCTRL9 bit[17:16]控制该信号。

VI 输出时钟配置方式如表 3-14 所示。

表3-14 VI 时钟频率配置

信号名	描述
vi3div_sel[1:0]	VI3 路接口分频时钟选择，可提供 VI3 路时钟的 1/2/4 分频选择。可通过配置 SC_PERCTRL9 bit[13:12]控制该信号。
vi2div_sel[1:0]	VI2 路接口分频时钟选择，可提供 VI2 路时钟的 1/2/4 分频选择。可通过配置 SC_PERCTRL9 bit[11:10]控制该信号。



信号名	描述
vi1div_sel[1:0]	VI1 路接口分频时钟选择，可提供 VI1 路时钟的 1/2/4 分频选择。可通过配置 SC_PERCTRL9 bit[9:8]控制该信号。
vi0div_sel[1:0]	VI0 路接口分频时钟选择，可提供 VI0 路时钟的 1/2/4 分频选择。可通过配置 SC_PERCTRL9 bit[7:6]控制该信号。
vi3_vi2_sel	VI3 路接口时钟选择，可通过配置 SC_PERCTRL9 bit[5]控制该信号。
vi1_vi0_sel	VI1 路接口时钟选择，可通过配置 SC_PERCTRL9 bit[4]控制该信号。

ETH 模块时钟配置方式如表 3-15 所示。

表3-15 ETH 时钟频率配置

信号名	描述
ethmac_mode[2:0]	GMAC 时钟模式选择。可通过配置 SC_PERCTRL9 [28:26]控制该信号。
ethclk_sel	ETH txclk 产生方式。可通过配置 SC_PERCTRL9 [29]控制该信号。
ethtioclk_sel	ETH txioclk 正反相配置。可通过配置 SC_PERCTRL9 [30]控制该信号。

SIO0/SIO1/SIO2 时钟频率配置方法如表 3-16 所示。

表3-16 SIO0/SIO1/SIO2 时钟频率配置

信号名	描述
sio0clk_sel[23:0]	用于 SIO0 模块的输出接口主时钟 SIO_MCLK 的分频因子控制，可通过配置 SC_PERCTRL13 bit[23:0]控制该信号。
sio0_bclk_sel[3:0]	配置 SIO0 模块在主模式下的位流时钟 SIO_XCK (BCLK) 的频率。可通过配置 SC_PERCTRL13 bit[27:24]控制该信号。
sio0_lrclk_sel[3:0]	配置 SIO0 模块在主模式下的输出采样率时钟 SIO_RFS/SIO_XFS (FSClk) 的频率，可通过配置 SC_PERCTRL13 bit[31:28]控制该信号。
sio12clk_sel[23:0]	用于 SIO1/SIO2 模块的输出接口主时钟 SIO_MCLK 的分频因子控制，可通过配置 SC_PERCTRL14 bit[23:0]控制该信号。
sio1_bclk_sel[3:0]	配置 SIO1 模块在主模式下的位流时钟 SIO_XCK (BCLK) 的频率。可通过配置 SC_PERCTRL14 bit[27:24]控制该信号。



信号名	描述
sio1_lrclk_sel[3:0]	配置 SIO1 模块在主模式下的输出采样率时钟 SIO_RFS/SIO_XFS (FSCLK) 的频率, 可通过配置 SC_PERCRTL14 bit[31:28]控制该信号。
sio2_bclk_sel[3:0]	配置 SIO2 模块在主模式下的位流时钟 SIO_XCK (BCLK) 的频率。可通过配置 SC_PERCRTL16 bit[3:0]控制该信号。
sio2_lrclk_sel[3:0]	配置 SIO2 模块在主模式下的输出采样率时钟 SIO_RFS/SIO_XFS (FSCLK) 的频率, 可通过配置 SC_PERCRTL16 bit[7:4]控制该信号。
sio0_blk_edge	SIO0 位流时钟正反相控制。可通过配置 SC_PERCRTL16 bit[8]控制该信号。
sio1_blk_edge	SIO1 位流时钟正反相控制。可通过配置 SC_PERCRTL16 bit[8]控制该信号。
sio2_blk_edge	SIO2 位流时钟正反相控制。可通过配置 SC_PERCRTL16 bit[8]控制该信号。
ackout_sel	ACKOUT 输出时钟。可通过配置 SC_PERCRTL9 bit[25]控制该信号。

一般的应用场景会给定采样率时钟频率 FSCLK, 而比特时钟 BCLK 和主时钟 MCLK 相对于 FSCLK 分别具有可变的倍数关系, 时钟频率配置方法示例如下:

假设时钟源 ARMPLL 已经配置输出频率为 540MHz, 此时要求配置出 SIO 工作时钟频率为 FSCLK=48kHz、MCLK=256FSCLK=12.288MHz、BCLK=16FSCLK=384KHz。配置方法如下:

- ARMPLL 到 MCLK 分频比计算为: $N=12.288/540$, 则 $sioclk_sel[23:0]=N \times 2^{27}$, 根据四舍五入取整原则计算为 3054199, 因此配置 $sioclk_sel=0x002E_9A77$, 即可得到 MCLK 的正确频率。
- BCLK 由 MCLK 分频获得, 分频比为 $BCLK/MCLK=16/256=1/16$, 因此根据配置表中的对应关系, 配置 $siobclk_sel[2:0]=0b100$ (对应 16 分频) 便可得到 BCLK 的正确频率。
- FSCLK 由 BCLK 分频得到, 分频比为 $1/16$, 因此根据配置表中的对应关系, 配置 $siolrclk_sel=0b011$ (对应 16 分频) 便可得到 FSCLK 的正确频率。

时钟门控配置

时钟门控配置请参见系统控制器 SC_PEREN。

注意事项

时钟配置需要注意以下事项:

- ARM 工作时钟上电默认为晶振模式, 即选择 XIN24 输入的晶振时钟。



- PLL 在变更频率配置时，需要等待 0.1ms 才能输出稳定的时钟。更改 PLL 频率配置只能在系统处于 SLOW 模式下进行。
- 在 PLL 输出时钟未稳定的情况下，不能执行系统切换到 PLL 模式。

3.3 处理器及存储器地址空间映射

3.3.1 处理器

Hi3520 内嵌处理器 ARM1176ZJF-S (ARM0) 和 ARM926EJ-S (ARM1)。

ARM1176ZJF-S 处理器的特点如下：

- 采用 32bit ARM v5TEJ，8 级流水，兼容 32bit ARM、16bit Thumb 指令集。
- 内嵌增强型 DSP 指令。
- 支持 Java。
- 支持 VFP (Vector Floating-Point) 运算。
- 支持指令预取。
- 提供独立的 16KB 指令 Cache、16KB 数据 Cache、Cache 为 4 路组相连 (4-Way)、Cache Line 大小为 32byte；Cache 为虚拟 Index 和物理地址混合结构；数据 Cache 支持 Write-back 和 Write-through 操作；处理器 Core 和 Cache 使用 64bit 接口。
- 数据 Cache 为 Non-blocking 结构，支持 HUM (Hit-Under-Miss)。
- 包含 MMU (Memory Management Unit)，支持 VxWorks、Linux、WindowCE、PalmOS 等开放操作系统。
- 支持 ARMv6 的页表结构，支持更加复杂的存储控制。
- 独立 64bit 指令和数据总线接口，总线工作频率与 ARM1176 系统时钟频率支持 1:1、1:2 和 1:3 的配置。
- 采用小端字序模式 (little endian)。
- 支持快速中断请求 FIQ 和一般中断请求 IRQ。
- 支持 JTAG 调试接口。
- 支持内部寄存器对系统性能进行统计。
- 提供独立 2KB ITCM (Instruction TCM)。
- 最高运行频率为 600MHz。

ARM926EJ-S 处理器的特点如下：

- 采用 32bit ARM v5TEJ，5 级流水，兼容 32bit ARM、16bit Thumb 指令集。
- 内嵌增强型 DSP 指令。
- 支持 Java。
- 提供独立的 16KB 指令 Cache、16kB 数据 Cache、4 路组相连 Cache，Cache Line 大小为 32byte；数据 Cache 支持 write-back 和 write-through 操作可配置。
- Cache 支持伪随机或者 round-robin 替换算法，并可进行配置。



- 独立指令和数据总线接口，总线工作频率与 ARM926EJ-S 系统时钟频率支持 1:1 和 1:2 的配置。
- 包含 MMU，支持 VxWorks、Linux、WindowCE、PalmOS 等开放操作系统。
- 提供独立 2KB ITCM（Instruction TCM）。
- 采用小端字序模式（little endian）。
- 支持快速中断请求 FIQ（Fast Interrupt Request）和一般中断请求 IRQ（Interrupt Request）。
- 支持 JTAG 调试接口。
- 支持动态功耗管理和静态功耗管理。

3.3.2 存储器地址空间映射

3.3.2.1 启动 ARM1176

Hi3520 内部使用的地址总线为 32bit 位宽，可寻址的地址空间为 4GB。Hi3520 支持以下 3 种启动方式：

- 从 NOR Flash 启动
- 从 NAND Flash 启动
- 从 DDR 启动



说明

上面 3 种启动方式仅针对 ARM1176。

通过对管脚 EBIADR23 和 EBIADR24 进行设置，选择启动方式，具体情况如表 3-17 所示。

表3-17 启动方式选择

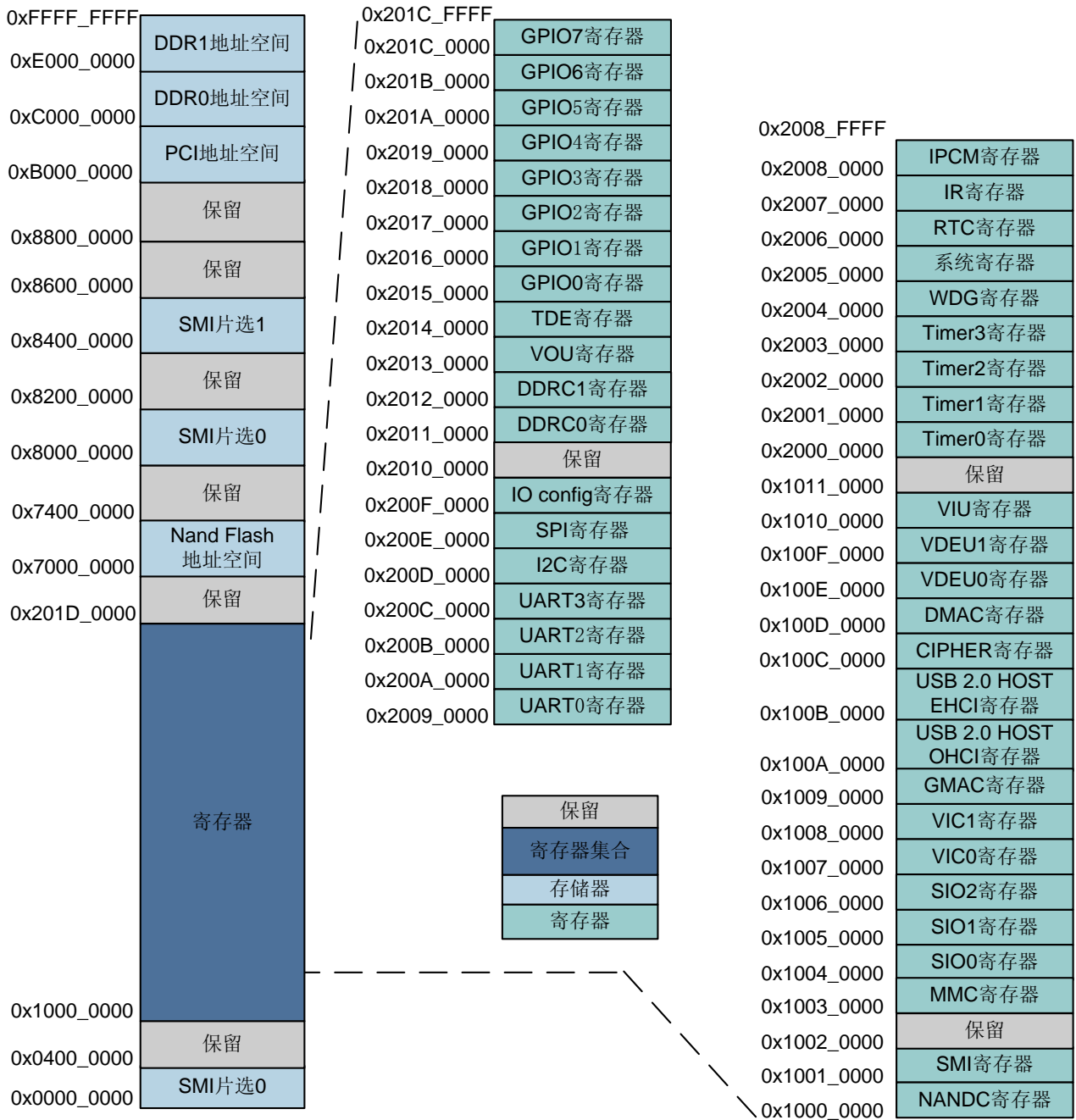
EBIADR23	EBIADR24	启动方式
下拉	下拉	从 NOR Flash 启动
下拉	上拉	从 DDR 启动
上拉	下拉	从 NAND Flash 启动
上拉	上拉	从 DDR 启动

从 NOR Flash 启动

按照表 3-17 进行正确设置后，才可选择从 NOR Flash 启动。此时外部连接的存储器一般为异步 NOR Flash，Hi3520 只支持 8bit 的 NOR Flash 存储器。

从 NOR Flash 启动时芯片内部的存储空间映射如图 3-3 所示。

图3-3 从 NOR Flash 启动时的地址空间映射





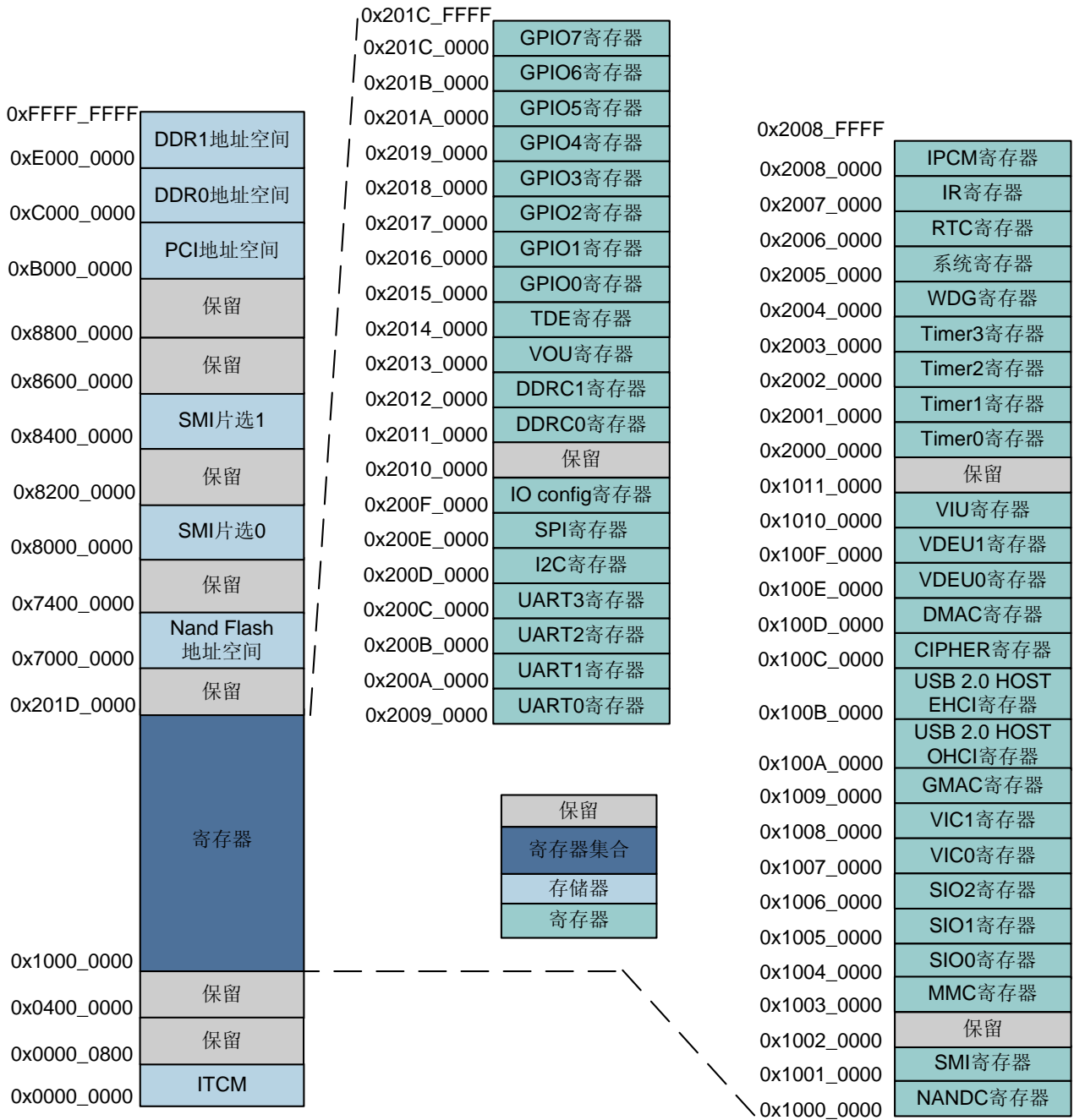
从 NOR Flash 启动时清除重映射后的地址分布如[图 3-4](#) 所示。



注意

- 系统实际上提供了 2KB 的 ITCM 地址空间，因此，软件需要保证在 ITCM 的程序和数据必须在 2KB 的范围内。
- 如果需要使用 ITCM，则必须通过 ARM 的系统控制协处理器 CP15 提供的寄存器设置 ITCM 使能，并且配置 ITCM 的大小等信息。使用 MCR 指令配置 C9 寄存器为 0xD。

图3-4 从 NOR Flash 启动时清除重映射后的地址分布



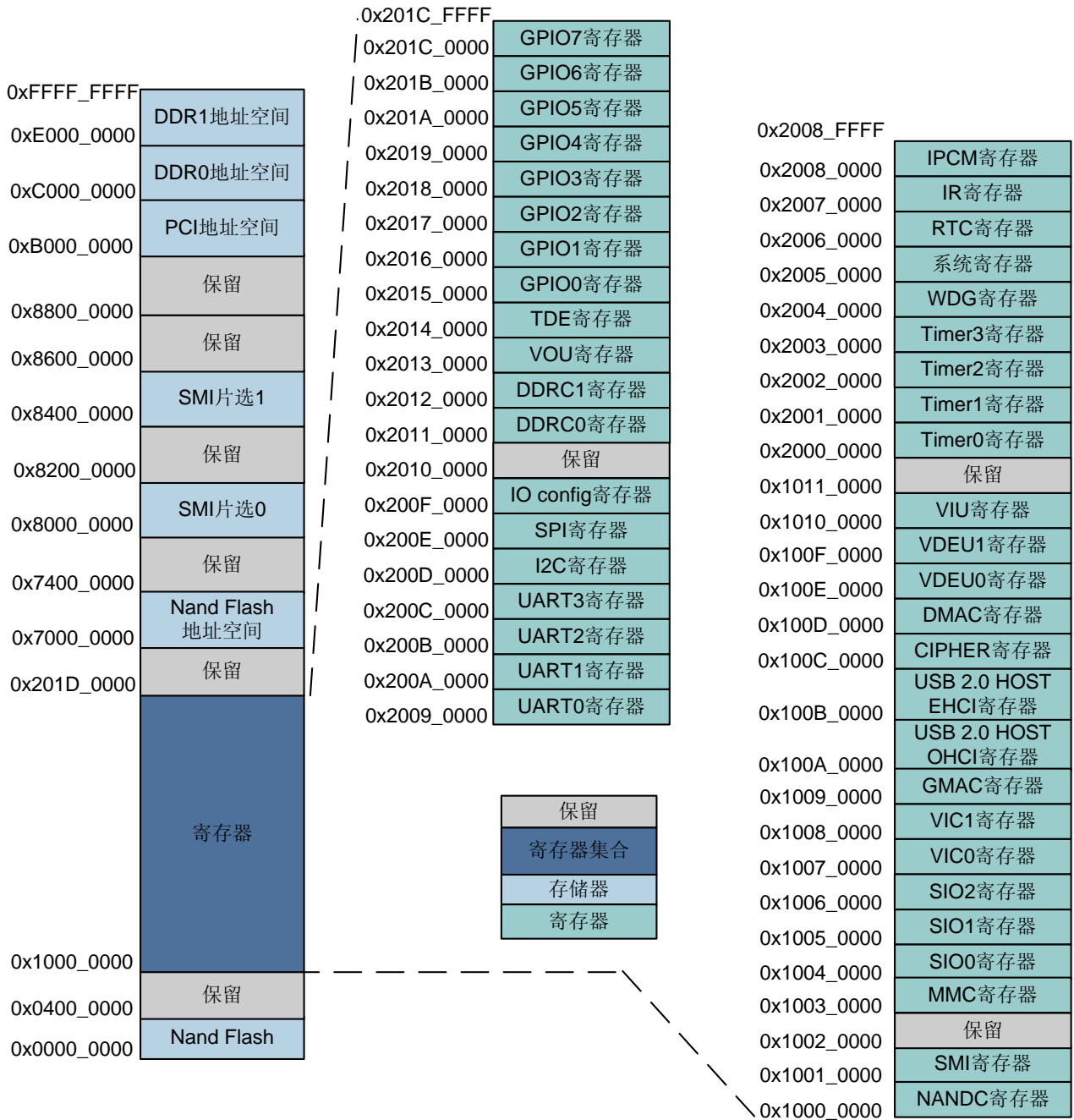
从 NAND Flash 启动

按照表 3-17 进行正确设置后，才可选择从 NAND Flash 启动。此时外部连接的存储器一般为 NAND Flash，Hi3520 只支持 8bit 的 NAND Flash 存储器。



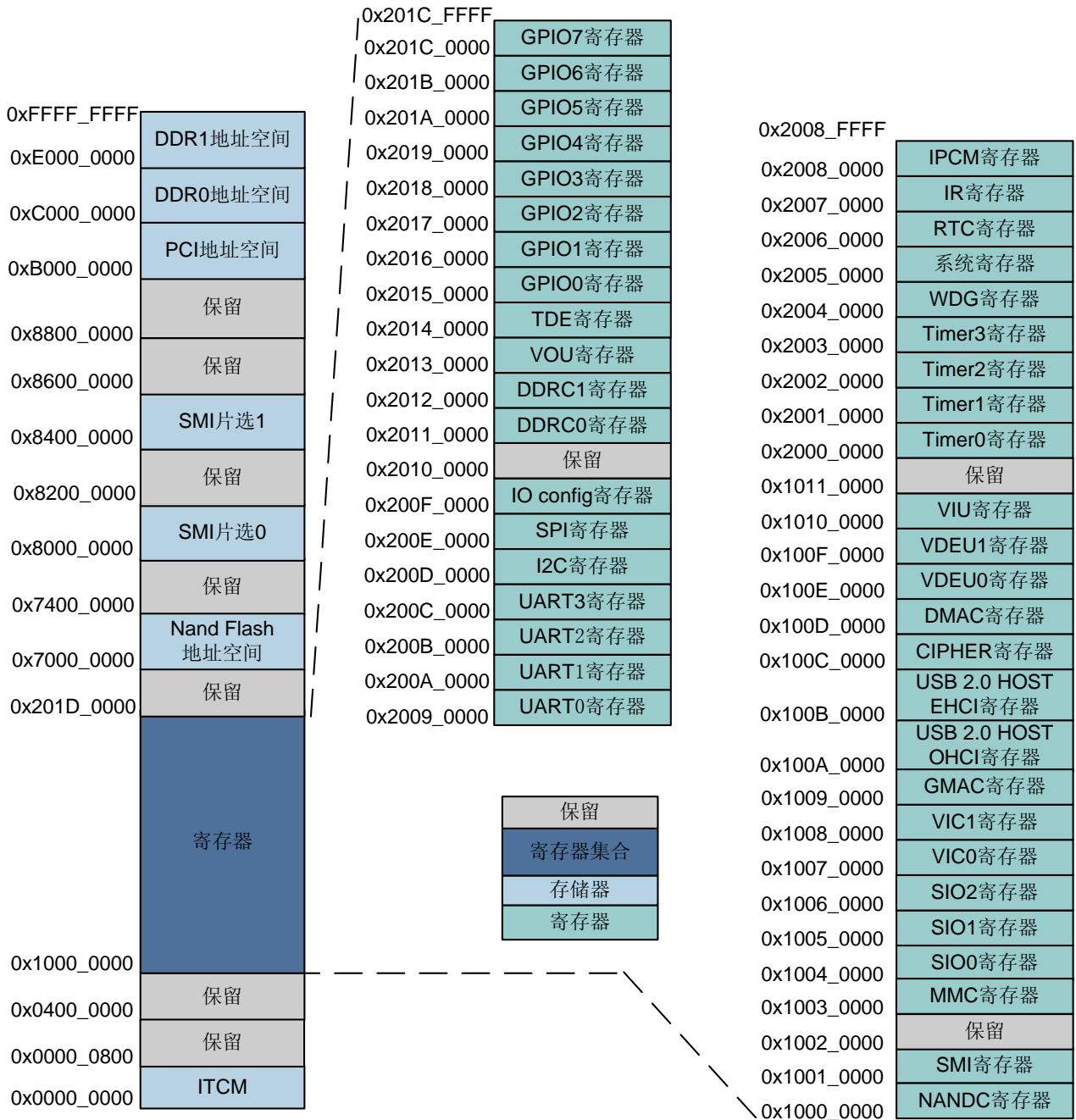
从 NAND Flash 启动启动时芯片内部的存储空间映射如图 3-5 所示。

图3-5 从 NAND Flash 启动时的地址空间映射



从 NAND Flash 启动时清除重映射后的地址分布如图 3-6 所示。

图3-6 从 NAND Flash 启动时清除重映射后的地址分布



从 DDR 启动

按照表 3-17 进行正确设置后，才可选择从 DDR 启动时。此时，芯片需要一片主芯片配合，具体步骤如下：

步骤 1 上电复位撤消，芯片的 ARM1176ZJF-S 处于复位状态。



步骤 2 主芯片 PCI 接口配置芯片，将启动程序写入 0x0000_0000(此时指向 DDR 器件)。

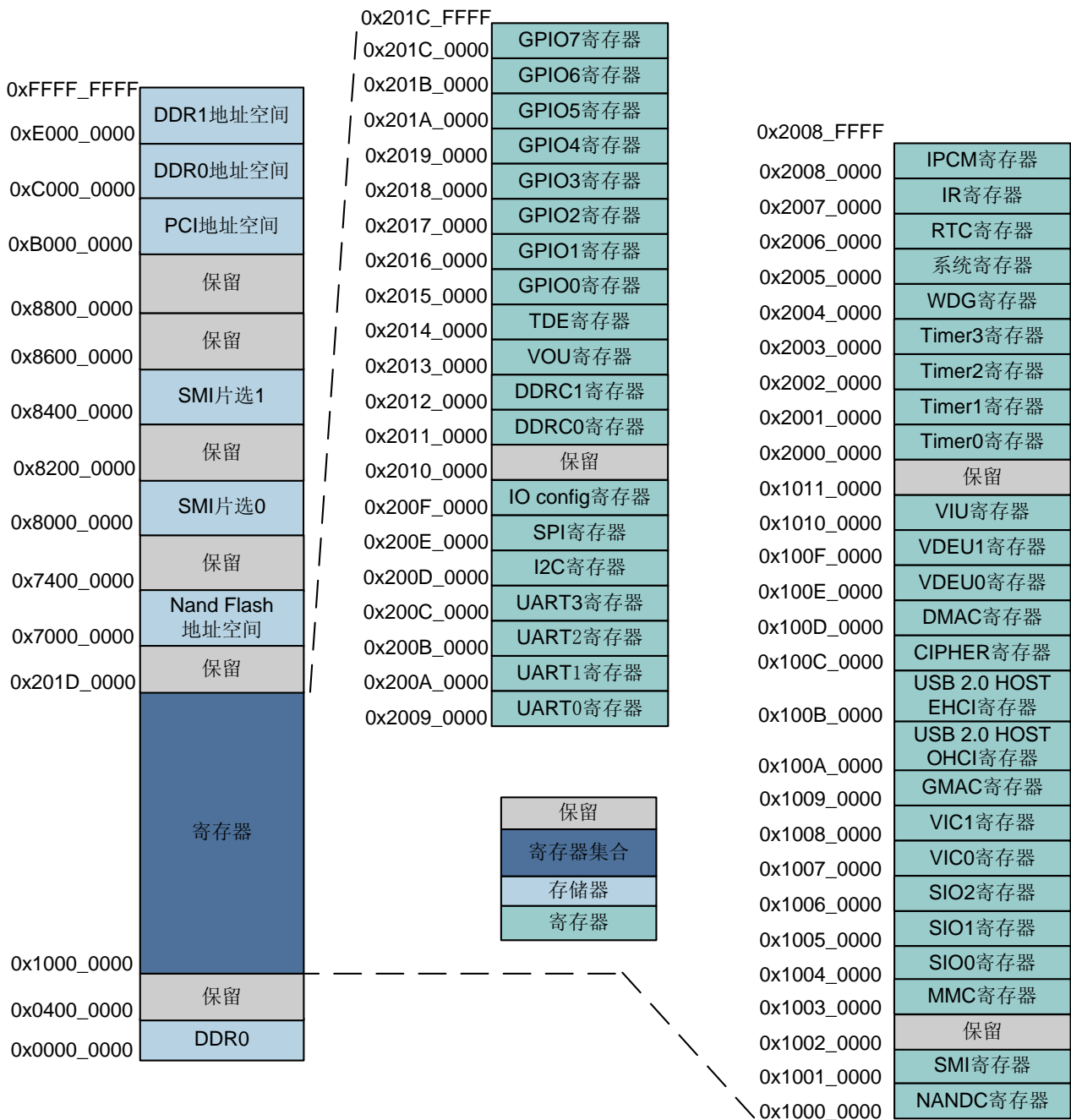
步骤 3 主芯片 PCI 接口配置芯片，将 ARM1176ZJF-S 的复位撤消。

步骤 4 芯片从 DDR 器件加载程序，进行启动。

----结束

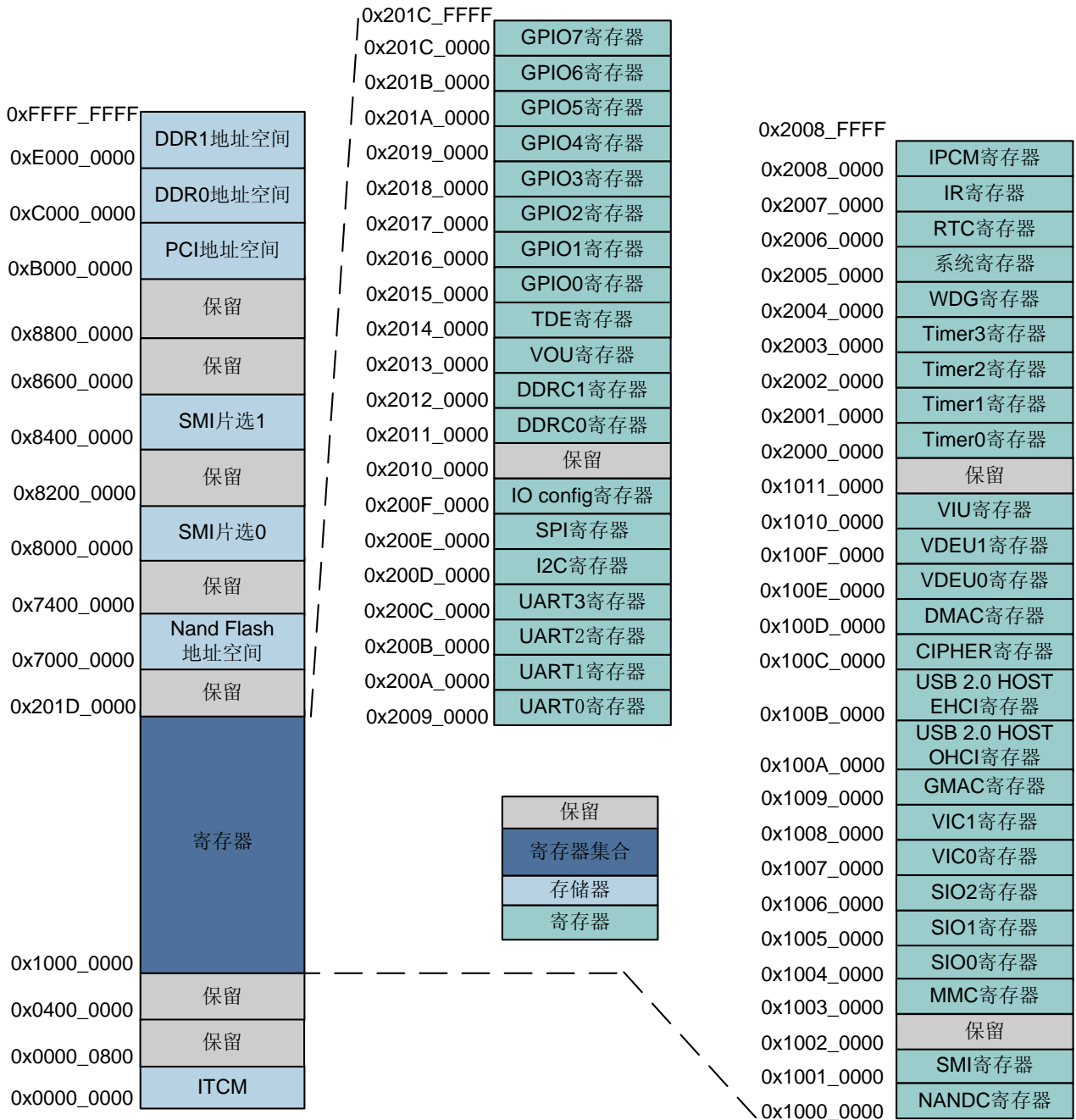
从 DDR 启动时芯片内部的存储空间映射如图 3-7 所示。

图3-7 从 DDR 启动时的地址空间映射



从 DDR 启动时清除重映射后的地址分布如图 3-8 所示。

图3-8 从 DDR 启动时清除重映射后的地址分布



3.3.2.2 启动 ARM926

ARM926 对 “0x00xx_xxxx” 地址段进行访问时，实际访问的地址是 “0xAAxx_xxxx”，AA 表示系统控制寄存器 SC_PERIPHCTRL17 bit[7:0]设置的值。对其他地址段的访问同 ARM1176 一致。



注意

ARM926 在 Hi3520 芯片系统中是从地位的处理器，只能在 ARM1176 启动后，通过配置相关寄存器才可启动。

ARM926 启动过程如下：

- 步骤 1 配置系统控制寄存器 SC_PERIPHCTRL17 bit[7:0]。例如配置成 0xC8。
- 步骤 2 向相应的地址（0xC800_0000）写入 ARM926 的启动程序。
- 步骤 3 向系统控制寄存器 SC_PERIPHCTRL8 bit[1]写 1，撤消 ARM926 的复位。
- 步骤 4 ARM926 自动从 0xC800_0000 读取相关程序进行 boot。

----结束

3.3.2.3 地址空间列表

Hi3520 内部的地址空间列表如表 3-18 所示。

表3-18 Hi3520 地址空间列表

起始地址	结束地址	功能	大小	说明
0xE000_0000	0xFFFF_FFFF	DDR1	512MB	动态存储器寻址空间。
0xC000_0000	0xDFFF_FFFF	DDR0	512MB	动态存储器寻址空间。
0xB000_0000	0xBFFF_FFFF	PCI 地址空间	256MB	此段地址空间分配请参见表 3-19。
0x8800_0000	0xAFFF_FFF	保留	640MB	-
0x8600_0000	0x87FF_FFFF	保留	32MB	-
0x8400_0000	0x85FF_FFFF	SSMC 静态片选 1	32MB	-
0x8200_0000	0x83FF_FFFF	保留	32MB	-
0x8000_0000	0x81FF_FFFF	SSMC 静态片选 0	32MB	-
0x7400_0000	0x7FFF_FFFF	保留	64MB	-
0x7000_0000	0x73FF_FFFF	NAND Flash 存储空间	64MB	-
0x201D_0000	0x6FFF_FFFF	保留	1278MB	-
0x201C_0000	0x201C_FFFF	GPIO7 寄存器	64KB	-



起始地址	结束地址	功能	大小	说明
0x201B_0000	0x201B_FFFF	GPIO6 寄存器	64KB	-
0x201A_0000	0x201A_FFFF	GPIO5 寄存器	64KB	-
0x2019_0000	0x2019_FFFF	GPIO4 寄存器	64KB	-
0x2018_0000	0x2018_FFFF	GPIO3 寄存器	64KB	-
0x2017_0000	0x2017_FFFF	GPIO2 寄存器	64KB	-
0x2016_0000	0x2016_FFFF	GPIO1 寄存器	64KB	-
0x2015_0000	0x2015_FFFF	GPIO0 寄存器	64KB	-
0x2014_0000	0x2014_FFFF	TDE 寄存器	64KB	-
0x2013_0000	0x2013_FFFF	VOU 寄存器	64KB	-
0x2012_0000	0x2012_FFFF	DDRC1 寄存器	64KB	-
0x2011_0000	0x2011_FFFF	DDRC0 寄存器	64KB	-
0x2010_0000	0x2010_FFFF	保留	64KB	-
0x200F_0000	0x200F_FFFF	IO config 寄存器	64KB	-
0x200E_0000	0x200E_FFFF	SPI 寄存器	64KB	-
0x200D_0000	0x200D_FFFF	I ² C 寄存器	64KB	-
0x200C_0000	0x200C_FFFF	UART3 寄存器	64KB	-
0x200B_0000	0x200B_FFFF	UART2 寄存器	64KB	-
0x200A_0000	0x200A_FFFF	UART1 寄存器	64KB	-
0x2009_0000	0x2009_FFFF	UART0 寄存器	64KB	-
0x2008_0000	0x2008_FFFF	IPCM	64KB	-
0x2007_0000	0x2007_FFFF	IR 寄存器	64KB	-
0x2006_0000	0x2006_FFFF	RTC 寄存器	64KB	-
0x2005_0000	0x2005_FFFF	系统控制寄存器	64KB	-
0x2004_0000	0x2004_FFFF	WDG 寄存器	64KB	-
0x2003_0000	0x2003_FFFF	Timer3 寄存器	64KB	-
0x2002_0000	0x2002_FFFF	Timer2 寄存器	64KB	-
0x2001_0000	0x2001_FFFF	Timer1 寄存器	64KB	-
0x2000_0000	0x2000_FFFF	Timer0 寄存器	64KB	-
0x1011_0000	0x1FFF_FFFF	保留	255MB	-



起始地址	结束地址	功能	大小	说明
0x1010_0000	0x1010_FFFF	VIU 寄存器	64KB	-
0x100F_0000	0x100F_FFFF	VEDU1 寄存器	64KB	-
0x100E_0000	0x100E_FFFF	VEDU0 寄存器	64KB	-
0x100D_0000	0x100D_FFFF	DMAC 寄存器	64KB	-
0x100C_0000	0x100C_FFFF	CIPHER 寄存器	64KB	-
0x100B_0000	0x100B_FFFF	USB 2.0 HOST EHCI 寄存器	64KB	-
0x100A_0000	0x100A_FFFF	USB 2.0 HOST OHCI 寄存器	64KB	-
0x1009_0000	0x1009_FFFF	GMAC 寄存器	64KB	-
0x1008_0000	0x1008_FFFF	VIC1 寄存器	64KB	-
0x1007_0000	0x1007_FFFF	VIC0 寄存器	64KB	-
0x1006_0000	0x1006_FFFF	SIO2 寄存器	64KB	-
0x1005_0000	0x1005_FFFF	SIO1 寄存器	64KB	-
0x1004_0000	0x1004_FFFF	SIO0 寄存器	64KB	-
0x1003_0000	0x1003_FFFF	SDIO 寄存器	64KB	-
0x1002_0000	0x1002_FFFF	保留	64KB	-
0x1001_0000	0x1001_FFFF	SSMC 寄存器	64KB	-
0x1000_0000	0x1000_FFFF	NANDC 寄存器	64KB	-
0x0400_0000	0x0FFF_FFFF	保留	192MB	-
0x0000_0000	0x03FF_FFFF	地址重映射选定的 存储器	64MB	复位时根据表 3-17 中管脚 EBIADR23 和 EBIADR24 的情 况定当前存储空间 所对应的器件。 通常情况下，清除 地址重映射之后， 会把 ITCM 配置到 该空间，ITCM 的 大小为 2KB，对应 地址空间范围为 0x0000_0000~ 0x0000_0800。



PCI 地址空间分配情况如表 3-19 所示。

表3-19 PCI 地址空间分配

起始地址	结束地址	功能	大小	说明
0xB000_0000	0xB000_03FF	PCI 控制寄存器	1KB	-
0xB000_0400	0xB000_FFFF	保留	63KB	-
0xB001_0000	0xB001_FFFF	IO 地址空间	64KB	当 PCI 接口处于主模式时，该段地址空间有效。
0xB002_0000	0xB7FF_FFFF	NP 地址空间	127MB	-
0xB800_0000	0xBFFF_FFFF	FP 地址空间	128MB	-

3.4 INT

3.4.1 概述

中断系统 INT（Interrupt System）为系统提供中断管理功能。Hi3520 的 INT 由 VIC0 和 VIC1 组成。

3.4.2 特点

INT 具有以下特性：

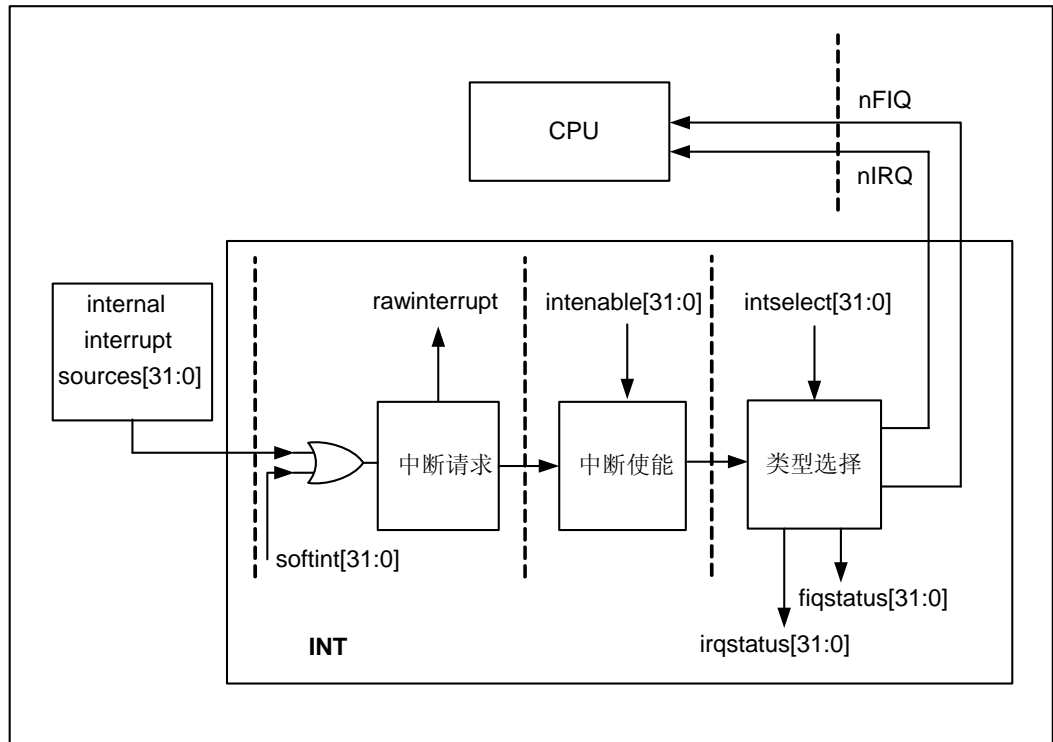
- 支持 32 个中断源，高电平触发。
- 支持在 INT 的工作时钟关断的情况下产生中断请求。
- 中断类型可配置，各中断源可根据需要配置为 IRQ（Interrupt Request）或 FIQ（Fast Interrupt Request）。
- 支持中断源屏蔽。
- 支持原始中断源状态查询和屏蔽后中断源状态查询。
- 支持访问保护功能。当设置为启用访问保护状态时，只有处于特权模式下的 CPU 才能访问 INT。

3.4.3 功能描述

功能框图

INT 的功能框图如图 3-9 所示。

图3-9 INT 功能框图



注：图中的 n 表示低电平有效。

CPU 可通过内部总线访问 INT 的功能寄存器。例如：

- 通过配置 `INT_INTENABLE` 实现对各中断源中断请求的使能或屏蔽。
- 通过配置 `INT_INTSELECT` 实现对各中断源中断请求类型的选择（通常最多只有一个中断请求会被配置为 FIQ）。
- 通过配置 `INT_SOFTINT` 和 `INT_SOFTINTCLEAR` 实现产生和清除软件中断请求。
- 通过 `INT_IRQSTATUS` 和 `INT_FIQSTATUS` 查询产生中断请求时的中断源状态。
- 通过 `INT_RAWINTR` 查询中断源及软件中断所产生的原始中断请求。

中断处理流程

当中断发生时，中断处理的步骤如下：

- 步骤 1 ARM 处理器切换到 IRQ 或 FIQ 中断模式。
- 步骤 2 跳转到初级中断处理程序并压栈。
- 步骤 3 进入 System 模式。
- 步骤 4 查询寄存器 `INT_IRQSTATUS` 或 `INT_FIQSTATUS` 确定中断来源，若存在多个等待服务的中断源则比较各中断的优先级。



步骤 5 跳转到中断源所对应的中断服务程序 ISR (Interrupt Service Routine)，通过配置寄存器 `INT_INTENABLE` 屏蔽当前处理的中断号，如果有需要可通过配置寄存器 `INT_INTENABLE` 开启相应中断使能。

步骤 6 执行 ISR。

步骤 7 清除当前中断源，若是软件中断则配置 `INT_SOFTINTCLEAR` 寄存器的相应位。

步骤 8 打开 ARM CPSR 寄存器中 IRQ 和 FIQ 屏蔽位。

步骤 9 出栈，并从中断中返回。

----结束

中断映射

VIC0 和 VIC1 的中断映射分别如表 3-20 和所示。

表3-20 VIC0 中断映射表

中断号	对应的中断源
0	WatchDog 中断
1	全局软件中断
2	COMMRX 中断
3	COMMTX 中断
4	Dual-Timer01 中断
5	Dual-Timer23 中断
6	GPIO0 中断
7	GPIO1 中断
8	GPIO2、GPIO3、GPIO4、GPIO5、GPIO6、GPIO7 组合中断
9	IR 中断
10	RTC 中断
11	SPI 中断
12	UART0 或 UART1 中断
13	UART2 或 UART3 中断
14	GMAC 中断
15	DMAC 中断
16	I ² C 中断
17	VIU 中断



中断号	对应的中断源
18	TDE 中断
19	VOU 中断
20	VEDU0 中断
21	VEDU1 中断
22	USB 2.0 HOST OHCI 中断
23	USB 2.0 HOST EHCI 中断
24	SDIO 中断
25	SIO0 中断
26	SIO1 中断
27	SIO2 中断
28	CIPHER 中断
29	PCI 中断
30	NANDC 中断
31	IPCM 中断

注：COMMRX、COMMTX 中断为 CPU 调试使用。

表3-21 VIC1 中断映射表

中断号	对应的中断源
0	WatchDog 中断
1	全局软件中断
2	COMMRX 中断
3	COMMTX 中断
4	Dual-Timer45 中断
5	Dual-Timer67 中断
6	GPIO0 中断
7	GPIO1 中断
8	GPIO2、GPIO3、GPIO4、GPIO5、GPIO6、GPIO7 组合中断
9	IR 中断



中断号	对应的中断源
10	RTC 中断
11	SPI 中断
12	UART0 或 UART1 中断
13	UART2 或 UART3 中断
14	GMAC 中断
15	DMAC 中断
16	I ² C 中断
17	VIU 中断
18	TDE 中断
19	VOU 中断
20	VEDU0 中断
21	VEDU1 中断
22	USB 2.0 HOST OHCI 中断
23	USB 2.0 HOST EHCI 中断
24	SDIO 中断
25	SIO0 中断
26	SIO1 中断
27	SIO2 中断
28	CIPHER 中断
29	PCI 中断
30	NANDC 中断
31	IPCM 中断

注：COMMRX、COMMTX 中断为 CPU 调试使用。

3.4.4 寄存器概览

VIC0 和 VIC1 的寄存器除基地址不同，其他特性都相同。二者的基地址分别是 0x1007_0000 和 0x1008_0000，以其中一组寄存器为例，概览如表 3-22 所示。



表3-22 INT 寄存器概览

偏移地址	名称	描述	页码
0x000	INT_IRQSTATUS	IRQ 中断状态寄存器	3-31
0x004	INT_FIQSTATUS	FIQ 中断状态寄存器	3-31
0x008	INT_RAWINTR	原始中断状态寄存器	3-32
0x00C	INT_INTSELECT	中断源选择寄存器	3-32
0x010	INT_INTENABLE	中断使能寄存器	3-33
0x014	INT_INTENCLEAR	中断使能清除寄存器	3-33
0x018	INT_SOFTINT	软件中断寄存器	3-34
0x01C	INT_SOFTINTCLEAR	软件中断清除寄存器	3-34
0x020	INT_PROTECTION	保护使能寄存器	3-35

3.4.5 寄存器描述

INT_IRQSTATUS

INT_IRQSTATUS 为 IRQ 中断状态寄存器。32bit 分别对应 32 个中断源，详细内容请参见表 3-20。

	Offset Address	Register Name	Total Reset Value
	0x000	INT_IRQSTATUS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	irqstatus		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	irqstatus
	Description		
	IRQ 中断源的中断状态。 0: 无中断; 1: 有中断, 并向处理器发出 IRQ 中断。		

INT_FIQSTATUS

INT_FIQSTATUS 为 FIQ 中断状态寄存器。32bit 分别对应 32 个中断源。



Offset Address		Register Name		Total Reset Value				
0x004		INT_FIQSTATUS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	fiqstatus							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RO	fiqstatus	FIQ 中断源的中断状态。 0: 无中断; 1: 有中断, 并向处理器发出 FIQ 中断。					

INT_RAWINTR

INT_RAWINTR 为原始中断状态寄存器。提供屏蔽前的中断请求状态和配置 INT_SOFTINT 产生的软件中断状态。32bit 分别对应 32 个中断源。

Offset Address		Register Name		Total Reset Value				
0x008		INT_RAWINTR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rawinterrupt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RO	rawinterrupt	中断源屏蔽前的中断请求状态。 0: 无中断; 1: 有中断。					

INT_INTSELECT

INT_INTSELECT 为中断源选择寄存器。选择对应的中断源是生成 IRQ 中断还是生成 FIQ 中断。32bit 分别对应 32 个中断源。



Offset Address		Register Name		Total Reset Value				
0x00C		INT_INTSELECT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	intselect							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	intselect	中断源的中断请求类型。 0: IRQ 中断; 1: FIQ 中断。					

INT_INTENABLE

INT_INTENABLE 为中断使能寄存器。用于使能中断请求线。复位时，由于 INT_INTENABLE 的值变为 0x0000_0000，所有中断源都被屏蔽。32bit 分别对应 32 个中断源。

Offset Address		Register Name		Total Reset Value				
0x010		INT_INTENABLE		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	intenable							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	intenable	读该寄存器时，返回各中断源的屏蔽状态。 0: 被屏蔽; 1: 未被屏蔽。 写该寄存器时，按位使能中断源。 0: 对应位的当前值不受影响; 1: 对应位被置 1，使能相应的中断请求。					

INT_INTENCLEAR

INT_INTENCLEAR 为中断使能清除寄存器。清除中断使能寄存器 INT_INTENABLE 的相应位，该寄存器为只写寄存器，无复位值。



Offset Address		Register Name		Total Reset Value																
0x014		INT_INTENCLEAR		-																
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0												
Name	intenableclear																			
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bits	Access	Name	Description																	
[31:0]	WO	intenableclear	屏蔽 INT_INTENABLE 对应的中断源。 0: INT_INTENABLE 对应位的当前值不受影响； 1: INT_INTENABLE 对应位被清 0，屏蔽相应的中断请求。																	

INT_SOFTINT

INT_SOFTINT 为软件中断寄存器。通过软件可控制中断源输入线产生软件中断。软件中断的清除可通过写软件中断清除寄存器 [INT_SOFTINTCLEAR](#) 实现，清除通常是在中断服务程序结束时进行。

Offset Address		Register Name		Total Reset Value																
0x018		INT_SOFTINT		0x0000_0000																
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0												
Name	softint																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																	
[31:0]	RW	softint	在指定的中断源上产生一个屏蔽前软件中断。 0: 对应位的当前值不受影响； 1: 对应位被置 1，产生一个软件中断请求。																	

INT_SOFTINTCLEAR

INT_SOFTINTCLEAR 为软件中断清除寄存器。清除软件中断寄存器 [INT_SOFTINT](#) 的相应位，该寄存器为只写寄存器，无复位值。



Offset Address		Register Name		Total Reset Value																
0x01C		INT_SOFTINTCLEAR		-																
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0												
Name	softintclear																			
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bits	Access	Name	Description																	
[31:0]	WO	softintclear	屏蔽 INT_SOFTINT 对应的中断请求。 0: INT_SOFTINT 的对应位不受影响; 1: INT_SOFTINT 的对应位清 0, 屏蔽相应的中断请求。																	

INT_PROTECTION

INT_PROTECTION 为保护使能寄存器。用于关断或使能被保护的寄存器访问权限。



- 复位时，该寄存器被清 0，用户模式（user mode）或特权模式（privileged mode）都可访问 INT 的寄存器。
- 当 CPU 无法产生正确的保护信息（HPROT）时，将该寄存器复位后即可允许在用户模式下访问 INT 的寄存器。

Offset Address		Register Name		Total Reset Value																
0x020		INT_PROTECTION		0x0000_0000																
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0												
Name	reserved								protection											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																	
[31:1]	-	reserved	保留。																	
[0]	RW	protection	寄存器访问保护使能。 0: 禁止寄存器访问保护，CPU 采用特权模式和用户模式都可以访问 INT 的寄存器； 1: 使能寄存器访问保护，CPU 只能采用特权模式访问 INT 的寄存器。																	



3.5 直接存储器存取控制器

3.5.1 概述

直接存储器访问（DMA）方式，是一种完全由硬件执行 I/O 交换的工作方式。在这种方式中，直接存储器访问控制器（DMAC）直接在存储器和外设、外设和外设、存储器和存储器之间进行数据传输，避免处理器干涉并减少了处理器中断处理开销。DMA（Directory Memory Access）方式一般用于高速传输成组的数据。DMAC（Directory Memory Access Controller）在收到 DMA 传输请求后根据 CPU 对通道的配置启动总线主控制器，向存储器和外设发出地址和控制信号，对传输数据的个数计数，并且以中断方式向 CPU 报告传输操作的结束或错误。

3.5.2 特点

DMAC 有如下特点：

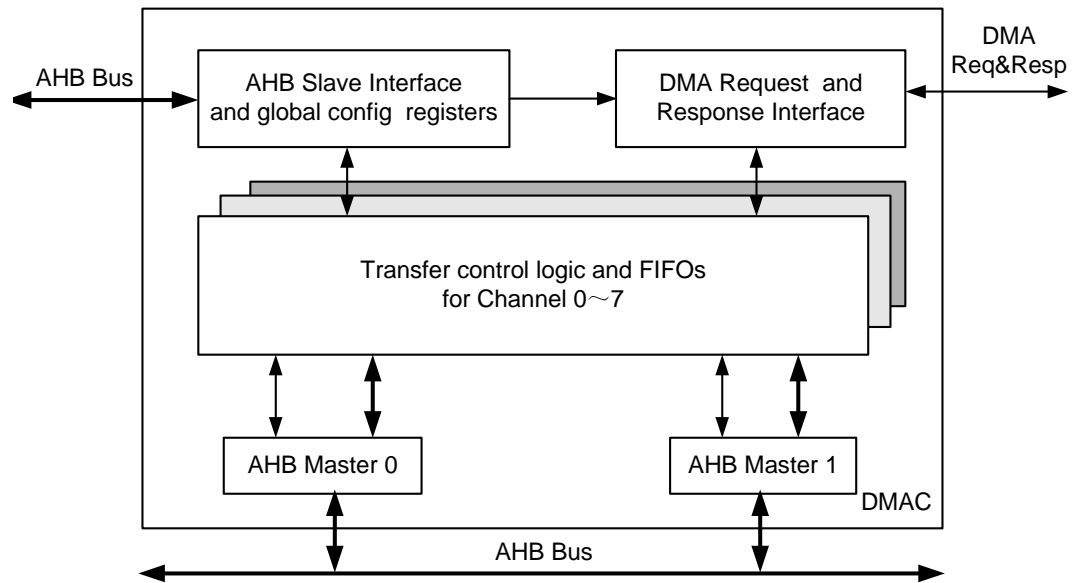
- 支持 8bit、16bit、32bit 数据位宽方式传输。
- 提供 8 个 DMA 通道，每个通道可配置用于一种单向传输。
- DMA 通道优先级固定，优先级从高到低对应的通道号依次为 0~7。当来自 2 个外设的 DMA 请求同时有效时，优先级高的通道先开始传输。
- DMAC 通道 0~通道 5 中各包含 1 个 4×32bit 的 FIFO，DMAC 通道 6~通道 7 中各包含 1 个 16×32bit 的 FIFO（First In First Out）。
- 提供 2 个总线宽度为 32bit 的 Master 总线接口用于数据传输。
- 外设可使用单次传输（single）和连续传输（burst）2 种 DMA 请求。
- 提供 16 组 DMA 请求输入，可通过配置，作为通道的源端请求或目的端请求。
- 支持软件控制的 DMA 请求。
- 支持通过编程决定 DMA burst 长度。
- 源地址和目的地址可分别配置为在 DMA 传输过程中自动递增或不递增。
- 支持 4 种数据传输方向：
 - 存储器至外设
 - 存储器至存储器
 - 外设至存储器
 - 外设至外设
- 支持链表 DMA 传输。
- 支持 DMAC 流控。
- 提供 2 个可屏蔽中断输出（分别上报给主/从 ARM），支持 DMA 错误和 DMA 传输完成中断屏蔽前后状态查询，及两者的组合中断状态查询。
- 支持 DMAC 使能禁止，用于功耗控制，支持 DMAC 时钟门控。

3.5.3 功能描述

3.5.3.1 功能框图

DMAC 的功能框图如图 3-10 所示。

图3-10 DMAC 功能框图



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

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

----结束

3.5.3.2 工作流程

DMAC 基本工作流程如下：

- 步骤 1 软件选定 DMAC 的一个通道用于 DMA 传输，配置该通道的源地址、目的地址、链表头指针、传输数据个数、源/目的端对应的外设请求线号、源/目的端使用的 Master 并启动该通道。一旦通道被启动，DMAC 硬件即开始检测与该通道相连的源外设和目的设备的 DMA 请求线上的活动。
- 步骤 2 源设备向 DMAC 发起 DMA 请求（如果源设备为存储器，DMAC 默认其 DMA 请求始终有效）。
- 步骤 3 DMAC 通道响应源设备 DMA 请求，从源设备读取数据并存入通道内部的 FIFO 中。
- 步骤 4 目的设备向 DMAC 发起 DMA 请求（如果目的设备为存储器，DMAC 默认其 DMA 请求始终有效）。



步骤 5 DMAC 通道响应目的设备 DMA 请求，从通道内部的 FIFO 中取出数据并写入目的设备。

步骤 6 步骤 2、3 和步骤 4、5 可能是并发执行的，因为源设备和目的设备有可能同时向 DMAC 发起 DMA 请求。

当出现 DMA 通道 FIFO 被写满而目的设备来不及读走、或 DMA 通道 FIFO 被读空而源设备来不及写入时，DMAC 自动阻塞源设备或目的设备的 DMA 请求，直到相应的 FIFO 空满状态解除为止。

在 DMAC 与源设备、目的设备的多次交互过程中，步骤 2~5 反复被执行，直到软件指定的数据传输全部完成时，发出传输完成中断（该中断可被屏蔽）。

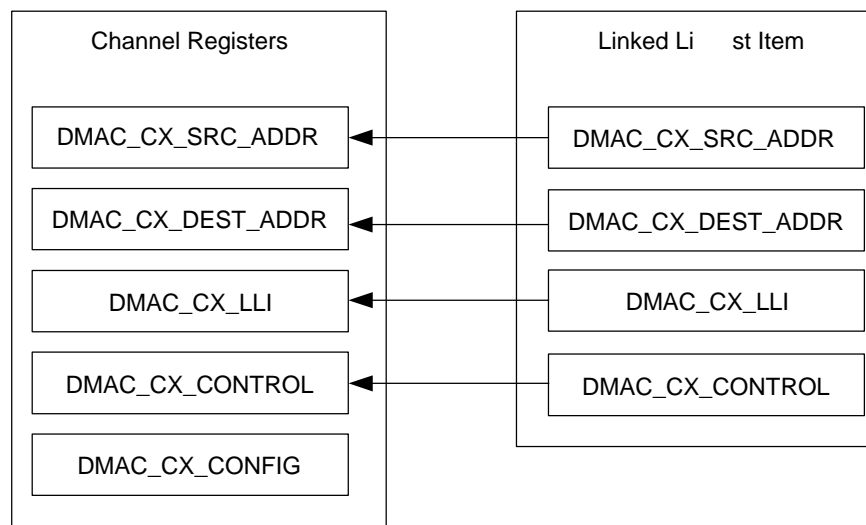
如果寄存器 `DMAC_CX_LLI` 不为 0，则以该寄存器的值为地址读取链表结点，并将读取值依次载入 `DMAC_CX_SRC_ADDR`、`DMAC_CX_DEST_ADDR`、`DMAC_CX_LLI` 以及 `DMAC_CX_CONTROL` 这四个寄存器（如图 3-11 所示），然后回到步骤 2。

若寄存器 `DMAC_CX_LLI` 的值为 0 则停止当前的 DMA 传输，通道自动关闭，传输过程结束。

----结束

LLI 更新通道寄存器示意如图 3-11 所示。

图3-11 LLI 更新通道寄存器示意图



3.5.3.3 DMA 与外设的连接关系

外设利用 DMA 请求信号向 DMAC 请求发起数据传输。

DMA 请求信号

DMAC 为每个外设提供了 2 种 DMA 请求信号，分别为：

- DMACBREQ



burst 传输请求信号。该信号引发一次 burst 传输，burst 长度为预先设定值。

- DMACSREQ

单次传输请求信号。该信号引发一次单次传输，即 DMAC 从外设读取一个数据或向外设写一个数据。

请求清除信号

DMAC 提供一个请求清除信号 DMACLR:

DMAC 向每个外设发出的 DMA 请求清除信号，用于应答外设的 DMA 请求信号。

DMAC 请求线

DMAC 的硬件请求描述如表 3-23 所示。

表3-23 DMAC 硬件请求描述

DMAC 控制器硬件请求线编号	描述
0	SIO0 接收通道
1	SIO0 发送通道
2	SIO1 接收通道
3	SIO1 发送通道
4	SIO2 接收通道
5	SIO2 发送通道
6	SPI 接收通道
7	SPI 发送通道
8	SDIO Read 通道
9	SDIO Write 通道
10	UART0 RX 通道 DMA 请求
11	UART0 TX 通道 DMA 请求
12	UART1 RX 通道 DMA 请求
13	UART1 TX 通道 DMA 请求
14	UART2 RX 通道 DMA 请求
15	UART2 TX 通道 DMA 请求



DMA 通道对应的源端和目的端请求由软件配置。例如，DMA 请求号 2 为 SIO1 的接收通道请求，若希望使用通道 3 传输 SIO1 的接收数据，则应配置 DMA 请求号 2 作为通道 3 的源端请求。

存储器没有 DMA 请求线，当 DMA 传输的一方为存储器时，DMAC 默认其 DMA 请求是始终有效的。由于 DMAC 的通道 6、7 上的传输，每次总线操作之后，都会插入 IDLE 周期，供高优先级通道的 Master 占用总线进行传输；因此，建议将存储器到存储器的传输配置到通道 6、7 进行，以免总线上其它通道较长时间无法占用总线。

3.5.4 工作方式

初始化

初始化 DMAC 步骤如下：

- 步骤 1 配置寄存器 `DMAC_CONFIG`，设置 DMAC 的 Master1、Master2 的 Endianness，向 `DMAC_CONFIG[e]` 写 1，启动 DMAC。
- 步骤 2 配置寄存器 `DMAC_INT_TC_CLR` 和 `DMAC_INT_ERR_CLR` 的所有位为 1，清除所有中断状态。
- 步骤 3 配置寄存器 `DMAC_SYNC` 相应位为 0，设置需要进行同步的 DMA 请求信号组。
- 步骤 4 依次配置并关闭各个通道。向每个通道的 `DMAC_CX_CONFIG[e]` 写 0，关闭通道。

----结束

启动通道

DMAC 初始化完成之后，需要配置并启动 DMA 通道，才可以使用 DMAC 来进行数据传输。通道的配置启动步骤如下：

- 步骤 1 读寄存器 `DMAC_ENBLD_CHNS`，找出处于非传输状态的通道，并从中选择一个通道用于配置。
- 步骤 2 向寄存器 `DMAC_INT_TC_CLR` 和 `DMAC_INT_ERR_CLR` 的相应位写 1，清除选定通道的中断状态。
- 步骤 3 配置并启动选定的通道。配置步骤如下：
 1. 配置通道寄存器 `DMAC_CX_SRC_ADDR`，设置源设备访问首地址。
 2. 配置通道寄存器 `DMAC_CX_DEST_ADDR`，设置目的设备访问首地址。
 3. 如果配置通道用于单块数据传输，则将通道寄存器 `DMAC_CX_LLI` 配置为 0。
 4. 如果配置通道用于链表数据传输，则将通道寄存器 `DMAC_CX_LLI` 配置为链表头指针。
 5. 配置通道寄存器 `DMAC_CX_CONTROL`，设置访问源/目的设备所采用的 Master、源/目的设备的位宽、burst size、地址递增以及 transfer size 等参数。
 6. 配置通道寄存器 `DMAC_CX_CONFIG`，设置本通道的 DMA 请求信号、流控方式及中断屏蔽。`DMAC_CX_CONFIG[e]` 此时应写入 0，即暂不启动该通道。
 7. 配置通道寄存器 `DMAC_CX_CONFIG`，启动该通道。注意此时该寄存器的写入值除 Channel Enable 位改为 1 外，其他位不变。



----结束

中断处理

DMA 通道配置启动传输完成之后或传输过程中出现错误，都会根据传输通道的配置来上报中断给主或者从 ARM。中断程序的处理步骤如下：

- 步骤 1** 读中断状态寄存器 `DMAC_INT_STAT`（如果是从 ARM 收到中断，则读中断状态寄存器 `DMAC_INT_STAT1`），找出发出中断请求的通道。当多个通道同时发出中断请求时，先服务优先级最高的中断。
- 步骤 2** 读寄存器 `DMAC_INT_TC_STAT`（如果是从 ARM 收到中断，则读中断状态寄存器 `DMAC_INT_STAT1`），比较选定的位是否为 1，以确定对应通道发出的中断为传输完成中断。若是，则转到步骤 4 执行；否则转到步骤 3 继续执行。
- 步骤 3** 读寄存器 `DMAC_INT_ERR_STAT`（如果是从 ARM 收到中断，则读中断状态寄存器 `DMAC_INT_STAT1`），比较选定的位是否为 1，以确定对应通道发出的中断为错误中断。若是则转到步骤 5 执行；否则退出中断处理。
- 步骤 4** 传输完成中断处理。可分为以下 3 个子步骤：
1. 配置寄存器 `DMAC_INT_TC_CLR`，对选定的位写 1，清除对应通道的中断状态。
 2. 取走或使用掉内存中 `buffer` 中的数据，如有需要（例如：需在内存中新开辟一个 `buffer`）可重新配置并启动该通道。
 3. 退出中断处理。
- 步骤 5** 错误中断处理。可分为以下 3 个子步骤：
1. 配置寄存器 `DMAC_INT_ERR_CLR`，对选定的位写 1，清除对应通道的中断状态。
 2. 给出错误信息，有需要可重新配置并启动该通道。
 3. 退出中断处理。

----结束

3.5.5 寄存器概览

DMAC 寄存器概览如表 3-24 所示。

表3-24 DMAC 寄存器概览（基址是 0x100D_0000）

偏移地址	寄存器名	功能简述	页码
0x000	DMAC_INT_STAT	DMAC 中断状态寄存器	3-44
0x004	DMAC_INT_TC_STAT	DMAC 传输完成中断状态寄存器	3-45
0x008	DMAC_INT_TC_CLR	DMAC 传输完成中断清除寄存器	3-45



偏移地址	寄存器名	功能简述	页码
0x00C	DMAC_INT_ERR_STAT	DMAC 错误中断状态寄存器	3-46
0x010	DMAC_INT_ERR_CLR	DMAC 错误中断清除寄存器	3-46
0x014	DMAC_RAW_INT_TC_STAT	DMAC 原始传输完成中断状态寄存器	3-47
0x018	DMAC_RAW_INT_ERR_STAT	DMAC 原始错误中断状态寄存器	3-47
0x01C	DMAC_ENBLD_CHNS	DMAC 通道使能状态寄存器	3-48
0x020	DMAC_SOFT_BREQ	软件 burst DMA 请求寄存器	3-49
0x024	DMAC_SOFT_SREQ	软件 Single DMA 请求寄存器	3-49
0x028	DMAC_SOFT_LBREQ	软件 Last burst DMA 请求寄存器	3-50
0x02C	DMAC_SOFT_LSREQ	软件 Last Single DMA 请求寄存器	3-51
0x030	DMAC_CONFIG	DMAC 配置寄存器	3-51
0x034	DMAC_SYNC	DMAC 请求同步寄存器	3-52
0x040	DMAC_INT_STAT1	DMAC 中断状态寄存器 1	3-52
0x044	DMAC_INT_TC_STAT1	DMAC Terminal Count 中断状态寄存器 1	3-53
0x048	DMAC_INT_ERROR_STAT1	DMAC 错误中断状态寄存器 1	3-53
0x100	DMAC_C0_SRC_ADDR	通道 0 源地址寄存器	3-52
0x104	DMAC_C0_DEST_ADDR	通道 0 目的地址寄存器	3-55
0x108	DMAC_C0_LLI	通道 0 链表项寄存器	3-56
0x10C	DMAC_C0_CONTROL	通道 0 控制寄存器	3-57
0x110	DMAC_C0_CONFIG	通道 0 配置寄存器	3-60
0x120	DMAC_C1_SRC_ADDR	通道 1 源地址寄存器	3-52
0x124	DMAC_C1_DEST_ADDR	通道 1 目的地址寄存器	3-55
0x128	DMAC_C1_LLI	通道 1 链表项寄存器	3-56
0x12C	DMAC_C1_CONTROL	通道 1 控制寄存器	3-57
0x130	DMAC_C1_CONFIG	通道 1 配置寄存器	3-60
0x140	DMAC_C2_SRC_ADDR	通道 2 源地址寄存器	3-52



偏移地址	寄存器名	功能简述	页码
0x144	DMAC_C2_DEST_ADDR	通道 2 目的地址寄存器	3-55
0x148	DMAC_C2_LLI	通道 2 链表项寄存器	3-56
0x14C	DMAC_C2_CONTROL	通道 2 控制寄存器	3-57
0x150	DMAC_C2_CONFIG	通道 2 配置寄存器	3-60
0x160	DMAC_C3_SRC_ADDR	通道 3 源地址寄存器	3-52
0x164	DMAC_C3_DEST_ADDR	通道 3 目的地址寄存器	3-55
0x168	DMAC_C3_LLI	通道 3 链表项寄存器	3-56
0x16C	DMAC_C3_CONTROL	通道 3 控制寄存器	3-57
0x170	DMAC_C3_CONFIG	通道 3 配置寄存器	3-60
0x180	DMAC_C4_SRC_ADDR	通道 4 源地址寄存器	3-52
0x184	DMAC_C4_DEST_ADDR	通道 4 目的地址寄存器	3-55
0x188	DMAC_C4_LLI	通道 4 链表项寄存器	3-56
0x18C	DMAC_C4_CONTROL	通道 4 控制寄存器	3-57
0x190	DMAC_C4_CONFIG	通道 4 配置寄存器	3-60
0x1A0	DMAC_C5_SRC_ADDR	通道 5 源地址寄存器	3-52
0x1A4	DMAC_C5_DEST_ADDR	通道 5 目的地址寄存器	3-55
0x1A8	DMAC_C5_LLI	通道 5 链表项寄存器	3-56
0x1AC	DMAC_C5_CONTROL	通道 5 控制寄存器	3-57
0x1B0	DMAC_C5_CONFIG	通道 5 配置寄存器	3-60
0x1C0	DMAC_C6_SRC_ADDR	通道 6 源地址寄存器	3-52
0x1C4	DMAC_C6_DEST_ADDR	通道 6 目的地址寄存器	3-55
0x1C8	DMAC_C6_LLI	通道 6 链表项寄存器	3-56
0x1CC	DMAC_C6_CONTROL	通道 6 控制寄存器	3-57
0x1D0	DMAC_C6_CONFIG	通道 6 配置寄存器	3-60
0x1E0	DMAC_C7_SRC_ADDR	通道 7 源地址寄存器	3-52
0x1E4	DMAC_C7_DEST_ADDR	通道 7 目的地址寄存器	3-55
0x1E8	DMAC_C7_LLI	通道 7 链表项寄存器	3-56
0x1EC	DMAC_C7_CONTROL	通道 7 控制寄存器	3-57
0x1F0	DMAC_C7_CONFIG	通道 7 配置寄存器	3-60



3.5.6 寄存器描述

DMAC_INT_STAT

DMAC_INT_STAT 为中断状态寄存器，给出经过屏蔽后的中断状态，供主 ARM 查询使用。若寄存器 [DMAC_INT_TC_STAT](#) 和 [DMAC_INT_ERR_STAT](#) 的相应位同时被屏蔽，则该寄存器的对应位被屏蔽。该寄存器的每 1 位对应 DMAC 的 1 个通道。



Offset Address	Register Name	Total Reset Value	
0x000	DMAC_INT_STAT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	int_stat	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	int_stat	DMA 各通道经屏蔽后的中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生中断； 1: 已产生中断，该中断请求可能来自该通道的错误中断或传输完成中断。

DMAC_INT_TC_STAT

DMAC_INT_TC_STAT 为传输结束状态寄存器，给出经过屏蔽后的传输完成中断状态，供主 ARM 查询使用，对应的屏蔽位为寄存器 [DMAC_CX_CONFIG\[itc\]](#)（其中 X 表示通道号 0~7）。该寄存器必须和寄存器 [DMAC_INT_STAT](#) 结合在一起使用。

Offset Address	Register Name	Total Reset Value	
0x004	DMAC_INT_TC_STAT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	int_tc_stat	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	int_tc_stat	经过屏蔽后的传输完成中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生传输完成中断； 1: 已产生传输完成中断。

DMAC_INT_TC_CLR

DMAC_INT_TC_CLR 为传输结束状态清除寄存器，用于清除传输完成中断。某 bit 配置 1 同时清除 [DMAC_INT_TC_STAT](#) 和 [DMAC_INT_TC_STAT1](#)。



Offset Address	Register Name	Total Reset Value	
0x008	DMAC_INT_TC_CLR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	int_tc_clr	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	WO	int_tc_clr	清除传输完成中断，bit[7:0]分别对应通道 7~通道 0。 0: 不清除； 1: 清除。

DMAC_INT_ERR_STAT

DMAC_INT_ERR_STAT 为错误中断状态寄存器，给出经过屏蔽后的错误中断状态，供主 ARM 查询使用，对应的屏蔽位为寄存器 [DMAC_CX_CONFIG\[ie\]](#)。该寄存器必须和 [DMAC_INT_STAT](#) 寄存器结合在一起使用。

Offset Address	Register Name	Total Reset Value	
0x00C	DMAC_INT_ERR_STAT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	int_err_stat	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	int_err_stat	经过屏蔽后的错误中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生错误中断； 1: 产生错误中断。

DMAC_INT_ERR_CLR

DMAC_INT_ERR_CLR 为错误中断清除寄存器，用于清除出错中断。某 bit 配置 1 同时清除 [DMAC_INT_ERR_STAT](#) 和 [DMAC_INT_ERR_STAT1](#)。



Offset Address		Register Name		Total Reset Value		
0x010		DMAC_INT_ERR_CLR		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved				int_err_clr	
Reset	0 0					
Bits	Access	Name	Description			
[31:8]	-	reserved	保留。			
[7:0]	WO	int_err_clr	清除出错中断，bit[7:0]分别对应通道 7~通道 0。 0: 不清除； 1: 清除。			

DMAC_RAW_INT_TC_STAT

DMAC_RAW_INT_TC_STAT 为原始传输完成中断状态寄存器，给出各通道屏蔽前的传输完成中断状态。

Offset Address		Register Name		Total Reset Value		
0x014		DMAC_RAW_INT_TC_STAT		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved				raw_int_tc_stat	
Reset	0 0					
Bits	Access	Name	Description			
[31:8]	-	reserved	保留。			
[7:0]	RO	raw_int_tc_stat	原始传输完成中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生传输完成中断； 1: 产生传输完成中断。			

DMAC_RAW_INT_ERR_STAT

DMAC_RAW_INT_ERR_STAT 为原始传输错误中断状态寄存器，给出各通道屏蔽前的错误中断状态。

Offset Address	Register Name	Total Reset Value	
0x018	DMAC_RAW_INT_ERR_STAT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	raw_int_err_stat	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	raw_int_err_stat	各通道屏蔽前的错误中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生错误中断； 1: 产生错误中断。

DMAC_ENBLD_CHNS

DMAC_ENBLD_CHNS 为使能通道寄存器，用于表明被使能的通道。

如寄存器

DMAC_ENBLD_CHNS 的某位为 1，表示对应的通道被使能。使能某个通道，由该通道的通道寄存器 DMAC_CX_CONFIG 的使能位决定。当某个通道的 DMA 传输结束时，寄存器

DMAC_ENBLD_CHNS 中与该通道对应的位被清 0。

Offset Address	Register Name	Total Reset Value	
0x01C	DMAC_ENBLD_CHNS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	enabled_channels	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	enabled_channels	通道使能状态，bit[7:0]分别对应通道 7~通道 0。 0: 禁止； 1: 使能。



DMAC_SOFT_BREQ

DMAC_SOFT_BREQ 为软件 burst 请求寄存器，用于供软件控制产生 DMA burst 请求。

读该寄存器，可得知当前正在请求 DMA burst 传输的设备。外设和该寄存器都可以产生 1 个 DMA 请求。

说明

建议不要同时使用软件 DMA 请求和硬件 DMA 请求。

	Offset Address	Register Name	Total Reset Value
	0x020	DMAC_SOFT_BREQ	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		soft_breq
Reset	0 0		
Bits	Access	Name	Description
[31:16]	-	reserved	保留。
[15:0]	RW	soft_breq	<p>用于软件控制产生 DMA burst 传输请求，每比特对应请求见表 3-23。</p> <p>写该寄存器时：</p> <p>0：无影响；</p> <p>1：产生 DMA burst 传输请求，当传输结束时该寄存器中的相应位被清 0。</p> <p>读该寄存器时：</p> <p>0：与请求线 DMACBREQ[15:0]对应的外设未发出 DMA burst 请求；</p> <p>1：与请求线 DMACBREQ[15:0]对应的外设正在请求 DMA burst 传输。</p>

DMAC_SOFT_SREQ

DMAC_SOFT_SREQ 为软件 Single 请求寄存器，用于供软件控制产生 DMA 单次传输请求。

如读该寄存器，可得知当前正在请求 DMA 单次传输的设备。通过 DMAC 的 16 个 DMA 请求输入信号和该寄存器都可以产生 1 个 DMA 请求。

说明

建议不要同时使用软件 DMA 请求和硬件 DMA 请求。



Offset Address		Register Name		Total Reset Value
0x024		DMAC_SOFT_SREQ		0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	reserved		soft_sreq	
Reset	0 0			
Bits	Access	Name	Description	
[31:16]	-	reserved	保留。	
[15:0]	RW	soft_sreq	用于软件控制产生 DMA single 传输请求，每比特对应请参见表 3-23。 写该寄存器时： 0：无影响； 1：产生 DMA single 传输请求，当传输结束时该寄存器中的相应位被清 0。 读该寄存器时： 0：与请求线 DMACBREQ[15:0]对应的外设未发出 DMA signal 请求； 1：与请求线 DMACBREQ[15:0]对应的外设正在请求 DMA signal 传输。	

DMAC_SOFT_LBREQ

DMAC_SOFT_LBREQ 为软件最后一个 burst 请求寄存器，用于供软件控制产生 DMA last burst 传输请求。

Offset Address		Register Name		Total Reset Value
0x028		DMAC_SOFT_LBREQ		0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	reserved		soft_lbreq	
Reset	0 0			
Bits	Access	Name	Description	
[31:16]	-	reserved	保留。	
[15:0]	WO	soft_lbreq	由软件发起 last burst 请求，每比特对应请求请参见表 3-23。 0：无影响； 1：产生 DMA last burst 传输请求，当传输结束时该寄存器中的相应位被清 0。	



DMAC_SOFT_LSREQ

DMAC_SOFT_LSREQ 为软件最后一个 single 请求寄存器，用于供软件控制产生 DMA last single 传输请求。

Offset Address		Register Name		Total Reset Value		
0x02C		DMAC_SOFT_LSREQ		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved				soft_lsreq	
Reset	0 0					0
Bits	Access	Name	Description			
[31:16]	-	reserved	保留。			
[15:0]	WO	soft_lsreq	由软件发起 last single 传输请求，每比特对应请求请参见表 3-23。 0: 无影响； 1: 产生一个 DMA last single 传输请求，当传输结束时该寄存器中的相应位被清 0。			

DMAC_CONFIG

DMAC_CONFIG 为配置寄存器，用于配置 DMAC 的操作。通过写该寄存器的 m1 (bit[1]) 和 m2 (bit[2])，可改变 DMAC 的 2 个 Master 接口的 endianness。复位时，DMAC 的 2 个 Master 接口被设为 little endian 模式。



说明

2 个 Master 均采用 little endian。

Offset Address		Register Name		Total Reset Value		
0x030		DMAC_CONFIG		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved			m2	m1	e
Reset	0 0					0
Bits	Access	Name	Description			
[31:3]	-	reserved	保留。			
[2]	RW	m2	Master2 endianness 配置。 0: little endian 模式； 1: big endian 模式。			



[1]	RW	m1	Master1 endianness 配置。 0: little endian 模式; 1: big endian 模式。
[0]	RW	e	DMAC 使能。 0: 禁止 DMAC; 1: 使能 DMAC。

DMAC_SYNC

DMAC_SYNC 为同步寄存器，用于启用或禁用为 DMA 请求信号提供的同步逻辑。



说明

建议各请求均不启用同步。

	Offset Address	Register Name	Total Reset Value
	0x034	DMAC_SYNC	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved dmac_sync		
Reset	0 0		
Bits	Access	Name	Description
[31:16]	-	reserved	保留。
[15:0]	RW	dmac_sync	控制是否需要对请求线进行同步，每比特对应请求请参见表 3-23。 0: 使能对应外设的 DMA 请求信号同步逻辑。 1: 禁止对应外设的 DMA 请求信号同步逻辑。

DMAC_INT_STAT1

DMAC_INT_STAT1 为中断状态寄存器，给出经过屏蔽后的中断状态，供从 ARM 查询使用。若寄存器 [DMAC_INT_STAT1](#) 和 [DMAC_INT_ERR_STAT1](#) 的相应位同时被屏蔽，则该寄存器的对应位被屏蔽。该寄存器的每 1 位对应着 DMAC 的 1 个通道。

	Offset Address	Register Name	Total Reset Value
	0x040	DMAC_INT_STAT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved int_stat		
Reset	0 0		
Bits	Access	Name	Description



[31:8]	-	reserved	保留。
[7:0]	RO	int_stat	DMA 各通道经屏蔽后的中断状态，bit[7:0]分别对应通道 7~通道 0。 0：未产生中断； 1：已产生中断，该中断请求可能来自该通道的错误中断或传输完成中断。

DMAC_INT_TC_STAT1

DMAC_INT_TC_STAT1 为传输结束状态寄存器，给出经过屏蔽后的传输完成中断状态，供从 ARM 查询使用，对应的屏蔽位为寄存器 [DMAC_CX_CONFIG\[itc\]](#)（其中 X 表示通道号 0~7）。该寄存器必须和寄存器 [DMAC_INT_STAT1](#) 结合在一起使用。

	Offset Address	Register Name	Total Reset Value
	0x044	DMAC_INT_TC_STAT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		int_tc_stat
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	int_tc_stat	经过屏蔽后的传输完成中断状态，bit[7:0]分别对应通道 7~通道 0。 0：未产生传输完成中断； 1：已产生传输完成中断。

DMAC_INT_ERR_STAT1

DMAC_INT_ERR_STAT1 为错误中断状态寄存器，给出经过屏蔽后的错误中断状态，供从 ARM 查询使用，对应的屏蔽位为寄存器 [DMAC_CX_CONFIG\[ie\]](#)。该寄存器必须和 [DMAC_INT_STAT1](#) 寄存器结合在一起使用。

Offset Address	Register Name	Total Reset Value	
0x048	DMAC_INT_ERR_STAT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	int_err_stat	
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。
[7:0]	RO	int_err_stat	经过屏蔽后的错误中断状态，bit[7:0]分别对应通道 7~通道 0。 0: 未产生错误中断； 1: 产生错误中断。

DMAC_CX_SRC_ADDR

DMAC_CX_SRC_ADDR 为源地址寄存器，给出当前待传数据的源地址（字节排序）。

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

每个寄存器在对应的通道被启动前都要由软件对其直接编程。当通道被启动后，该寄存器在下列情况下更新：

- 当源地址递增时。
- 当传完一个完整的数据块后，从链表结点中载入时。

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

源地址和目的地址必须与源设备和目的设备的传输宽度对齐。

Offset Address	Register Name	Total Reset Value	
$0x100+X \times 0x20$	DMAC_CX_SRC_ADDR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	src_addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	src_addr	DMA 源地址。



说明

- DMAC 提供了 8 个通道，每个通道都包括 5 个通道寄存器：
- [DMAC_CX_SRC_ADDR](#) 寄存器
- [DMAC_CX_DEST_ADDR](#) 寄存器
- [DMAC_CX_LLI](#) 寄存器
- [DMAC_CX_CONTROL](#) 寄存器
- [DMAC_CX_CONFIG](#) 寄存器

当 DMA 从存储器中载入链表结点时，前 4 个寄存器由 DMAC 自动更新。



注意

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

DMAC_CX_DEST_ADDR

DMAC_CX_DEST_ADDR 为目的地址寄存器，偏移地址为： $0x104+X \times 0x20$ 。其中，X 的取值为 0~7，分别对应 DMA 通道 0~通道 7。

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

- 目的地址递增。
- 传完一个完整的数据块后，从链表结点中载入。

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

	Offset Address	Register Name	Total Reset Value
	$0x104+X \times 0x20$	DMAC_CX_DEST_ADDR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	dest_addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	dest_addr	DMA 目的地址。

DMAC_CX_LLI

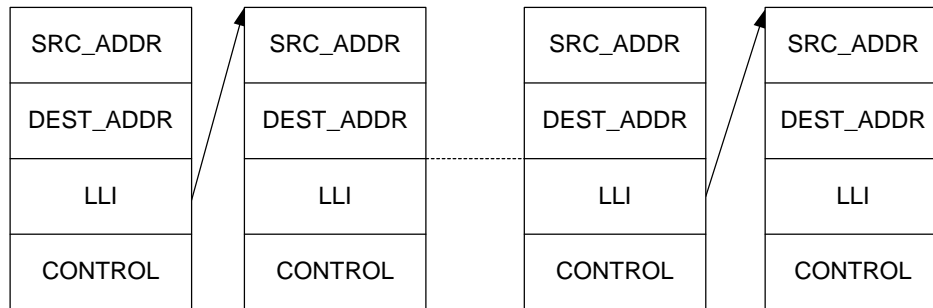
DMAC_CX_LLI 为链表寄存器，偏移地址为： $0x108+X \times 0x20$ 。其中，X 的取值为 0~7，分别对应 DMA 通道 0~通道 7。

DMAC 的链表结点数据结构为：

- 通道寄存器 **DMAC_CX_SRC_ADDR**，设置源设备首地址。
- 通道寄存器 **DMAC_CX_DEST_ADDR**，设置目的设备首地址。
- 通道寄存器 **DMAC_CX_LLI**，设置下一个结点的地址。
- 通道寄存器 **DMAC_CX_CONTROL**，设置访问源/目的的设备所采用的 Master、源/目的设备的位宽、burst size、地址递增以及 transfer size 等参数。

DMAC 链表结构示例如图 3-12 所示。

图3-12 DMAC 链表结构示例



注意

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

如果 LLI 的值为 0，表示当前结点是链表的链尾，则当本结点对应的数据块全部传完后，该通道就会被关闭。

	Offset Address	Register Name	Total Reset Value															
	$0x108+X \times 0x20$	DMAC_CXLLI	$0x0000_0000$															
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																	
Name	lli																reserved	lm
Reset	0 0																	
	Bits	Access	Name	Description														



[31:2]	RW	lli	Linked list item。下一个链表结点地址的[31:2]位，地址位[1:0]为 0。要求链表地址 4 字节对齐。
[1]	RW	reserved	保留，写入时必须写 0，读出时应屏蔽该位。
[0]	RW	lm	用于载入下一个链表结点的 Master。 0: Master1; 1: Master2。

DMAC_CX_CONTROL

DMAC_CX_CONTROL 为通道控制寄存器，偏移地址为：0x10C+X×0x20。其中，X 的取值为 0~7，分别对应 DMA 通道 0~通道 7。

通道控制寄存器包含了 DMA 通道控制信息，如传输长度、burst 长度、传输位宽等。

每个寄存器在对应的通道被启动前，都要由软件对其直接编程。当通道被启动后，该寄存器的值在传完 1 个完整的数据块后，从链表结点载入时更新。

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

	Offset Address	Register Name	Total Reset Value
	0x10C+X× 0x20	DMAC_CX_CONTROL	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	i prot di si d s dwidth swidth dbsize sbsize		transfersize
Reset	0 0		
Bits	Access	Name	Description
[31]	RW	i	传输完成中断使能位。该位用于决定当前链表结点是否触发传输完成中断。 0: 当前链表结点不触发传输完成中断; 1: 当前链表结点触发传输完成中断。
[30:28]	RW	prot	Master 发出的访问保护 HPROT[2:0]信号，这几位的具体含义请参见表 3-27 所示。
[27]	RW	di	目的地址递增。 0: 目的地址不递增; 1: 目的地址每传一个数就递增一次。 目的设备为外设时目的地址不递增；目的设备为存储器时目的地址递增。
[26]	RW	si	源地址递增。



			<p>0: 源地址不递增; 1: 源地址每传一个数就递增一次。 源设备为外设时源地址不递增; 源设备为存储器时源地址递增。</p>
[25]	RW	d	<p>设置访问目的设备的 Master。 0: SIO0、SIO1、UART0、SDIO、SPI 使用 Master1 访问; 1: NORFlash、DDR 使用 Master2 访问。</p>
[24]	RW	s	<p>设置访问源设备的 Master。 0: SIO0、SIO1、UART0、SDIO、SPI 使用 Master1 访问; 1: NORFlash、DDR 使用 Master2 访问。</p>
[23:21]	RW	dwidth	<p>目的设备传输位宽。 宽于 Master 位宽的传输位宽是非法的。 目的设备和源设备的位宽可以不一样, 硬件自动对数据进行 pack 和 unpack。 DWidth 的值和具体的位宽对应关系请参见表 3-26。</p>
[20:18]	RW	swidth	<p>源设备传输位宽。 宽于 Master 位宽的传输位宽是非法的。 目的设备和源设备的位宽可以不一样, 硬件自动对数据进行 pack 和 unpack。 SWidth 的值和具体的位宽对应关系请参见表 3-26。</p>
[17:15]	RW	dbsize	<p>目的设备 burst 长度。 表示 1 次目的设备 burst 传输所需传输的数据个数, 即当 DMACCxBREQ 有效时, 传输的数据个数。 该值必须被设为目的设备支持的 burst 大小, 或者若目的设备为存储器, 被设为到存储地址边界的存储区域大小。 DBSize 的值和具体的传输长度的对应关系请参见表 3-25。</p>
[14:12]	RW	sbsize	<p>源设备 burst 长度。 表示 1 次源设备 burst 传输所需传输的数据个数, 即当 DMACCxBREQ 有效时, 传输的数据个数。 该值必须被设为源设备支持的 burst 大小, 或者若源设备为存储器时, 被设为到存储地址边界的存储区域大小。 SBSIZE 的值和具体的传输长度的对应关系请参见表 3-25。</p>
[11:0]	RW	transfersize	<p>通过写该寄存器可设定 DMA 传输的长度, 前提是 DMAC 是流控制器。这里 transfer size 表示的源设备待传数据的个数。 读该寄存器可得到在与目的设备相连的总线上已传出的数据个数。 当该通道处于活动状态时, 读该寄存器得不到有效信息。这是因为当软件得到读出的寄存器值, 该寄存器的值已经随着</p>



			通道传输改变了。对该寄存器的读操作一般是用在通道被启动后然后又停止传输时。
--	--	--	---------------------------------------

DMAC_CX_CONTROL 寄存器的 DBSize 及 SBSize 的值与其对应的 burst 长度如表 3-25 所示。

表3-25 DBSize 及 SBSize 的值与其对应的 burst 长度

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

DMAC_CX_CONTROL 寄存器的 DWidth 和 SWidth 的值与其对应传输位宽如表 3-26 所示。

表3-26 DWidth 和 SWidth 的值与其对应传输位宽

SWidth 或 DWidth 的值	传输位宽
000	Byte (8bit)
001	Halfword (16bit)
010	Word (32bit)
其他	保留

配置寄存器 DMAC_CX_CONTROL 时需注意：

- 当源设备的传输宽度小于目的设备传输宽度时，源设备的传输宽度与 transfer size 的乘积应为目的设备传输宽度的整数倍，否则 FIFO 中的数据将会滞留并丢失。
- SWidth 和 DWidth 字段不能设置为未定义的位宽。
- transfer size 字段若被写为 0 且 DMAC 又是流控制器，则 DMAC 将不会发生任何传输动作。编程者应负责关闭此 DMA 通道并对此通道重新编程。
- 不应 DMAC_CX_CONTROL 寄存器进行普通的写入/读出测试。由于 transfer size 字段不是一个普通的可写入并读回相同值的寄存器字段。当写入时，该字段



如一个控制寄存器，因为其决定了 DMAC 应传输多少个数据；当读回时，该字段则相当于一个状态寄存器，因为其返回的剩下的待传输数据个数（以源设备位宽为单位）。

- 当 transfer size 字段的设置值大于源设备或目的设备中的 FIFO 的深度（是外设的 FIFO，不是 DMAC 的 FIFO），则 DMAC 的源地址或目的地址必须被设为不递增模式，否则有可能导致外设的 FIFO 溢出。

总线访问信息在传输发生时由 Master 接口信号提供给源设备或目的设备。这些访问信息是通过对通道寄存器编程设定的 `DMAC_CX_CONTROL[prot]` 和 `DMAC_CX_CONFIG[Lock]` 位。表 3-27 给出了使用 prot 的 3 个保护位的含义。

表3-27 DMAC_CX_CONTROL 寄存器 Prot 段属性及定义

比特	描述	目的
[2]	Cacheable or nonCacheable	指明访问是可 Cache 还是不可 Cache。 0: 不可 Cache。 1: 可 Cache。 该位控制总线信号 HPROT[3]的输出。
[1]	bufferable or nonbufferable	指明访问是可缓冲还是不可缓冲。 0: 不可缓冲。 1: 可缓冲。 该位控制总线信号 HPROT[2]的输出。
[0]	privileged or user	指明访问是用户模式还是特权模式。 0: 用户模式。 1: 特权模式。 该位控制总线信号 HPROT[1]的输出。

DMAC_CX_CONFIG

DMAC_CX_CONFIG 为通道配置寄存器，偏移地址为：0x110+X×0x20。其中，X 的取值为 0~7，分别对应 DMA 通道 0~通道 7。

该寄存器在新的链表结点被载入时不会被更新。



Offset Address	Register Name	Total Reset Value	
0x110+X×0x20	DMAC_CX_CONFIG	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved h a l t i c ie flow_cntrl reserved dest_peripheral reserved src_peripheral e		
Reset	0 0		
Bits	Access	Name	Description
[31:21]	-	reserved	保留。 写入时必须写入 0，读出时应被屏蔽。
[20]	ITC1	R/W	传输完成中断屏蔽位。 0：屏蔽本通道的传输完成对从 ARM 的中断； 1：不屏蔽本通道的传输完成从 ARM 的中断。
[19]	IE1	R/W	错误中断屏蔽位。 0：屏蔽本通道的错误对从 ARM 的中断； 1：不屏蔽本通道的错误对从 ARM 的中断。
[17]	RO	a	Active 位。 0：通道 FIFO 中没有数据； 1：通道 FIFO 中有数据。 该位可以和 Halt 位以及 Channel Enable 位一起用于无数据丢失地关闭一个 DMA 通道。
[16]	RW	l	Lock 位。 0：禁止总线上 lock 传输； 1：使能总线上 lock 传输。
[15]	RW	itc	传输完成中断屏蔽位。 0：屏蔽本通道的传输完成对主 ARM 的中断； 1：不屏蔽本通道的传输完成对主 ARM 的中断。
[14]	RW	ie	错误中断屏蔽位。 0：屏蔽本通道的错误对主 ARM 的中断； 1：不屏蔽本通道的错误对主 ARM 的中断。
[13:11]	RW	flow_cntrl	流控及传输类型字段。 该字段用于指定流控制器和传输类型。流控制器可以是 DMAC、源设备和目的设备。 传输类型可以是存储器到外设、外设到存储器、外设到外设、存储器到存储器。详细描述请参见表 3-28 所示。



[10]	-	reserved	保留。 写入时必须写入 0，读出时应被屏蔽。
[9:6]	RW	dest_peripheral	目的设备。该字段用于选择一个外设请求信号作为本通道的 DMA 目的设备的请求信号。 如果 DMA 传输的目的设备是存储器则该字段被忽略。
[5]	-	reserved	保留。 写入时必须为 0，读出时应被屏蔽。
[4:1]	RW	src_peripheral	源设备。该字段用于选择一个外设请求信号作为本通道的 DMA 源设备的请求信号。 如果 DMA 传输的源设备是存储器则该字段被忽略。
[0]	RW	e	通道使能位。读该寄存器可获得本通道的状态（也可通过读寄存器 DMACEnbldChns 获得）。 0：关闭通道； 1：启动通道。 通过清 0 可关闭通道。将该位被清 0 时，当前的总线传输会继续执行直到完成。然后通道关闭，FIFO 中剩余的数据全部丢失；当最后一个 LLI 完成或传输中出现错误时，通道也会被关闭，同时该位被清 0；如果要关闭通道，而又不使通道 FIFO 中的数据丢失，则 Halt 位也必须同时被置 1，使通道忽略后来的 DMA 请求。然后必须轮询 active 位，直到其值变为 0，表明通道 FIFO 中不再留有数据。此时才能够清除 enable 位。 通过置 1 启动通道必须先重新初始化通道，然后才能再次启动通道；若通过简单的置 1 启动通道，会引发不可预测性的后果。

注：当刚通过写 Channel Enable 位关闭一个通道时，必须要等到轮询到寄存器 `DMAC_ENBLD_CHNS` 中的相应 bit 为 0 之后，才能将 Channel Enable 位重新置 1。这是因为通道实际的关闭并没有在将 Channel Enable 位清 0 后立即生效。总线 burst 的运行延时也必须考虑到。

表 3-28 描述了 `DMAC_CX_CONFIG` 寄存器的 `flow_cntrl` 字段对应的流控和传输类型。

表3-28 流控制器及传输类型位定义

比特值	传输类型	控制器
000	存储器至存储器	DMAC
001	存储器至外设	DMAC
010	外设至存储器	DMAC
011	源设备至目的设备	DMAC
其他	保留	保留



3.6 CIPHER

3.6.1 概述

CIPHER 是一个实现 DES (Data Encryption Standard) /3DES 和 AES (Advanced Encryption Standard) 加解密处理的模块, DES/3DES 和 AES 算法的实现符合 FIPS46-3/FIPS 197 标准。DES/3DES 和 AES 的工作模式符合 FIPS -81/NIST special800-38a 标准。

CIPHER 模块适用于进行大量数据的高效加解密处理, 可支持一次实现单个分组的加解密或多个分组的加解密。

3.6.2 特点

CIPHER 模块有如下特点:

- AES 密钥长度支持 128、192、256 位。
- DES 密钥长度支持 64 位。
- 3DES 支持 3 个密钥的方式, 也支持 2 个密钥的方式。
- AES 支持 ECB (Electronic CodeBook)、CBC (Cipher Block Chaining)、1/8/128-CFB (Cipher FeedBack)、128-OFB (Output FeedBack) 和 CTR (Counter) 几种工作模式, 工作模式符合 NIST special800-38a 标准。
- DES/3DES 支持 ECB、CBC、1/8/64-CFB、1/8/64-OFB 几种工作模式, 工作模式符合 FIPS-81 标准。
- ECB、CBC、CFB、OFB 工作模式下, 支持一次实现多个分组的加解密运算, 也支持一次实现单个分组的加解密运算。
- AES 的 CTR 工作模式下, 只支持一次实现单个分组的加解密运算。
- 提供中断状态查询、中断屏蔽和中断清除功能。
- 提供强行中止模块运行功能。
- 提供对输入数据 (包括分组输入、向量输入、密钥) 和输出数据 (包括分组输出、向量输出) 的字节序调整功能。

3.6.3 功能描述

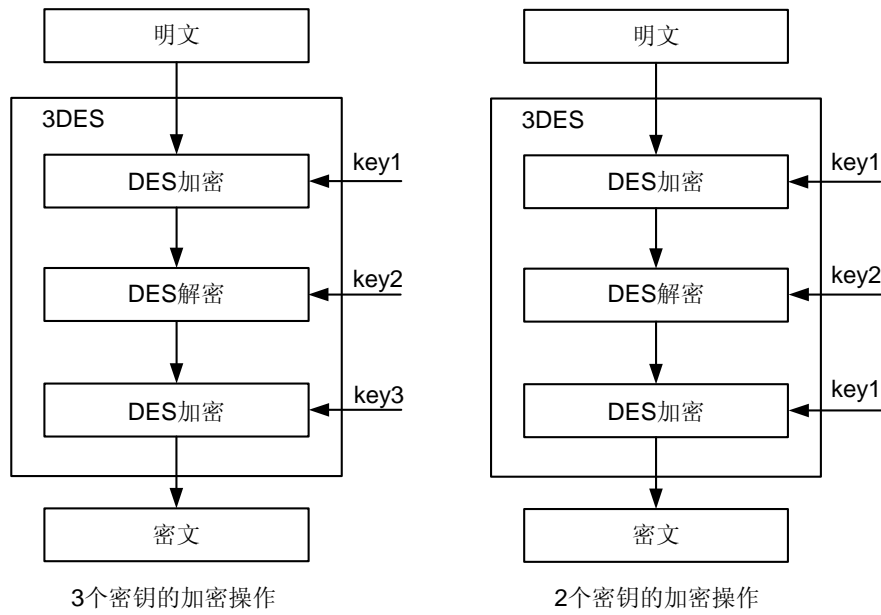
DES/3DES、AES 算法支持的几种工作模式分别符合 FIPS-81 标准和 NIST special800-38a 标准, 对于 DES/3DES 和 AES 算法, ECB、CBC 和 CFB 工作模式相同, OFB 和 CTR (只有 AES 算法中包括) 工作模式略有区别。

3DES 算法

3DES 支持 3 个密钥和 2 个密钥的运算, 2 个密钥的运算可以看作 3 个密钥的一种简化情况, 在 2 个密钥的操作中的第三个密钥 (key3) 都使用第一个密钥 (key1) 代替。

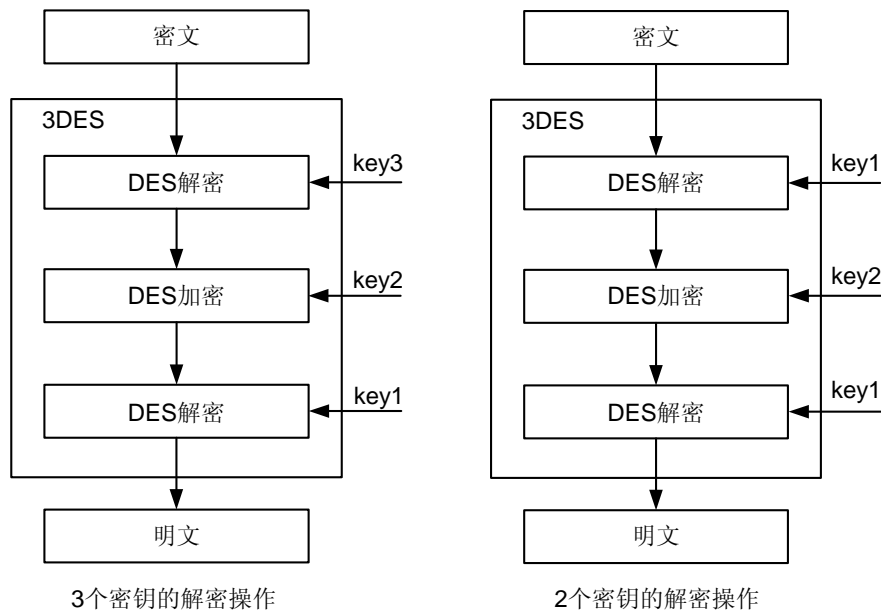
3 个密钥和 2 个密钥的 3DES 加密运算过程如图 3-13 所示。

图3-13 3个密钥和2个密钥的3DES加密操作



3个密钥和2个密钥的3DES解密运算过程如图3-14所示。

图3-14 3个密钥和2个密钥的3DES解密操作



ECB 模式

ECB (Electronic CodeBook) 模式中，加、解密算法是直接应用到各个分组数据，而且各个分组的运算均独立。这个特点使得明文的加密操作和密文的解密操作可以并行进行。AES/DES 和 3DES 的电子密码本 (ECB) 模式分别如图 3-15 和图 3-16 所示。

图3-15 AES/DES 的电子密码本 (ECB) 模式

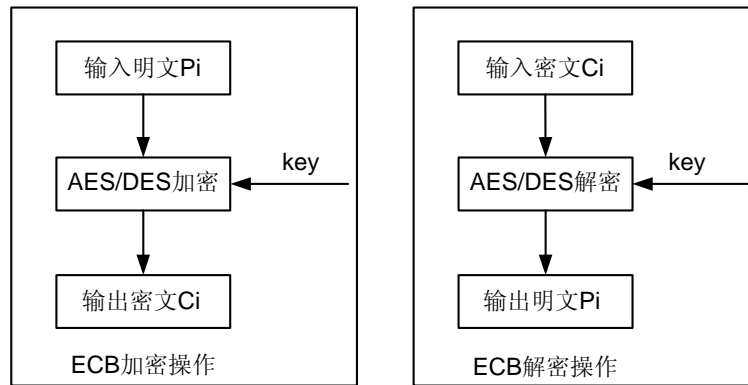
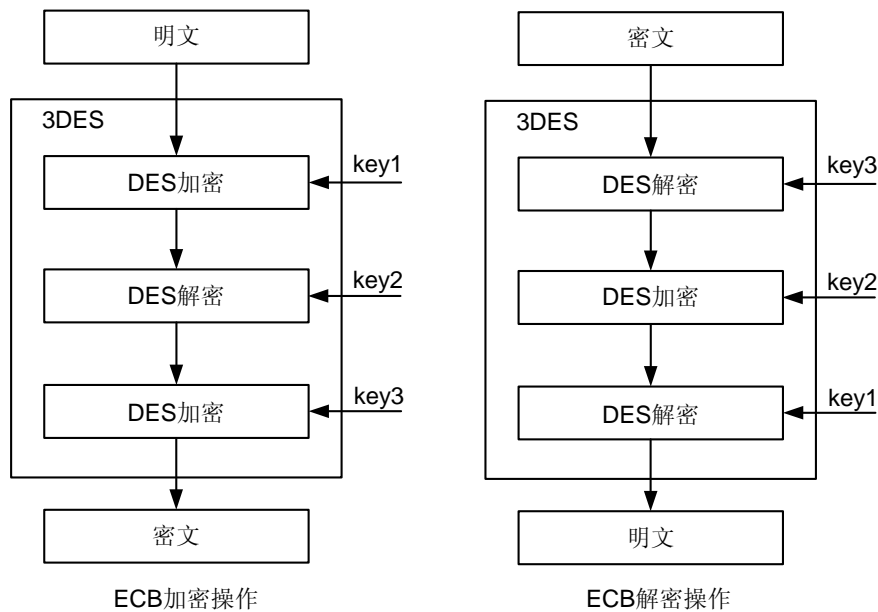


图3-16 3DES 的电子密码本 (ECB) 模式



CBC 模式

CBC (Cipher Block Chaining) 模式下, 加密的输入明文分组需要先与输入向量 IV (Initialization Vector) 进行异或操作, 才进入加密操作, 而每个明文分组的加密处理都与上一个明文分组处理的结果 (即密文) 相关, 因此 CBC 模式下的加密操作是不能进行并行处理的。但是解密操作不依赖于上一个分组的明文输出, 是可以进行并行处理的。AES/DES 和 3DES 的密码分组链接 (CBC) 模式分别如图 3-17 和图 3-18 所示。



图3-17 AES/DES 的密码分组链接 (CBC) 模式

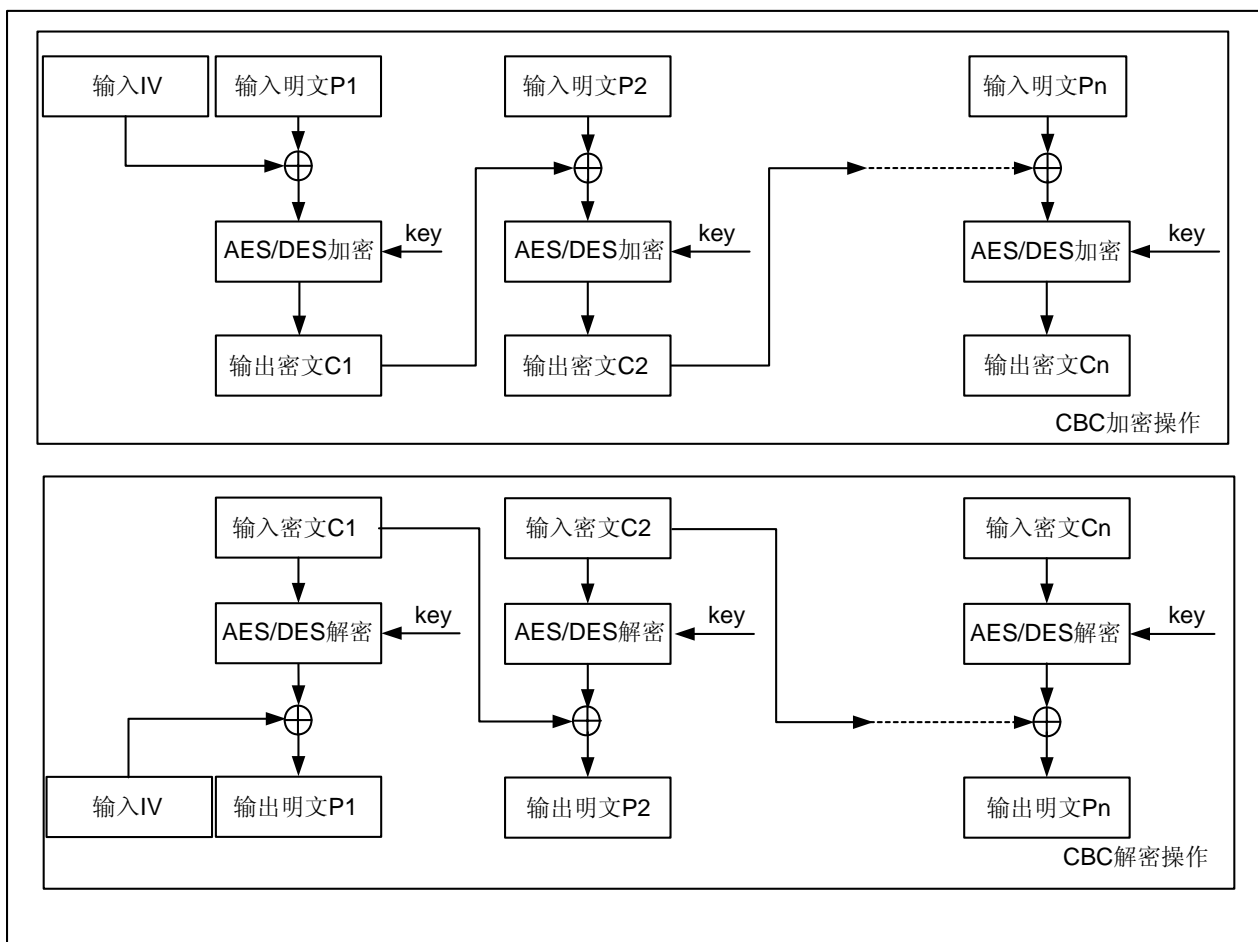
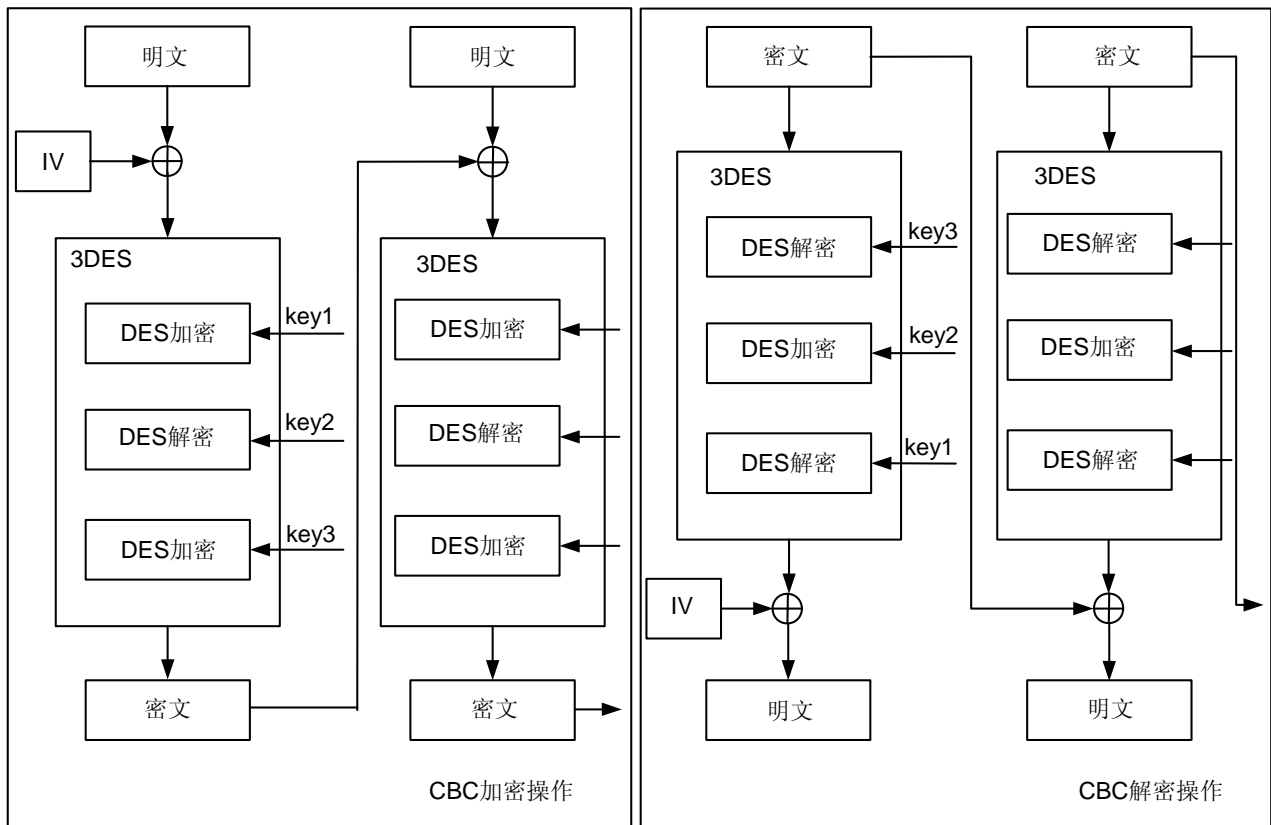


图3-18 3DES 的密码分组链接（CBC）模式



CFB 模式

CFB (Cipher FeedBack) 模式是将分组密码转换成流密码的一种工作模式，可以通过选择 CFB 的操作位数来实现。移位操作的位数用 s 位表示，关于 s 位，存在以下 2 种情况：

- 对于 DES/3DES， s 位可以是 1 位、8 位或 64 位。
- 对于 AES， s 位可以是 1 位、8 位或 128 位。

AES/DES 的 s 位密码反馈 (CFB) 模式和 3DES 的 s 位密码反馈 (CFB) 模式分别如图 3-19 和图 3-20 所示。

图3-19 AES/DES 的 s 位密码反馈 (CFB) 模式

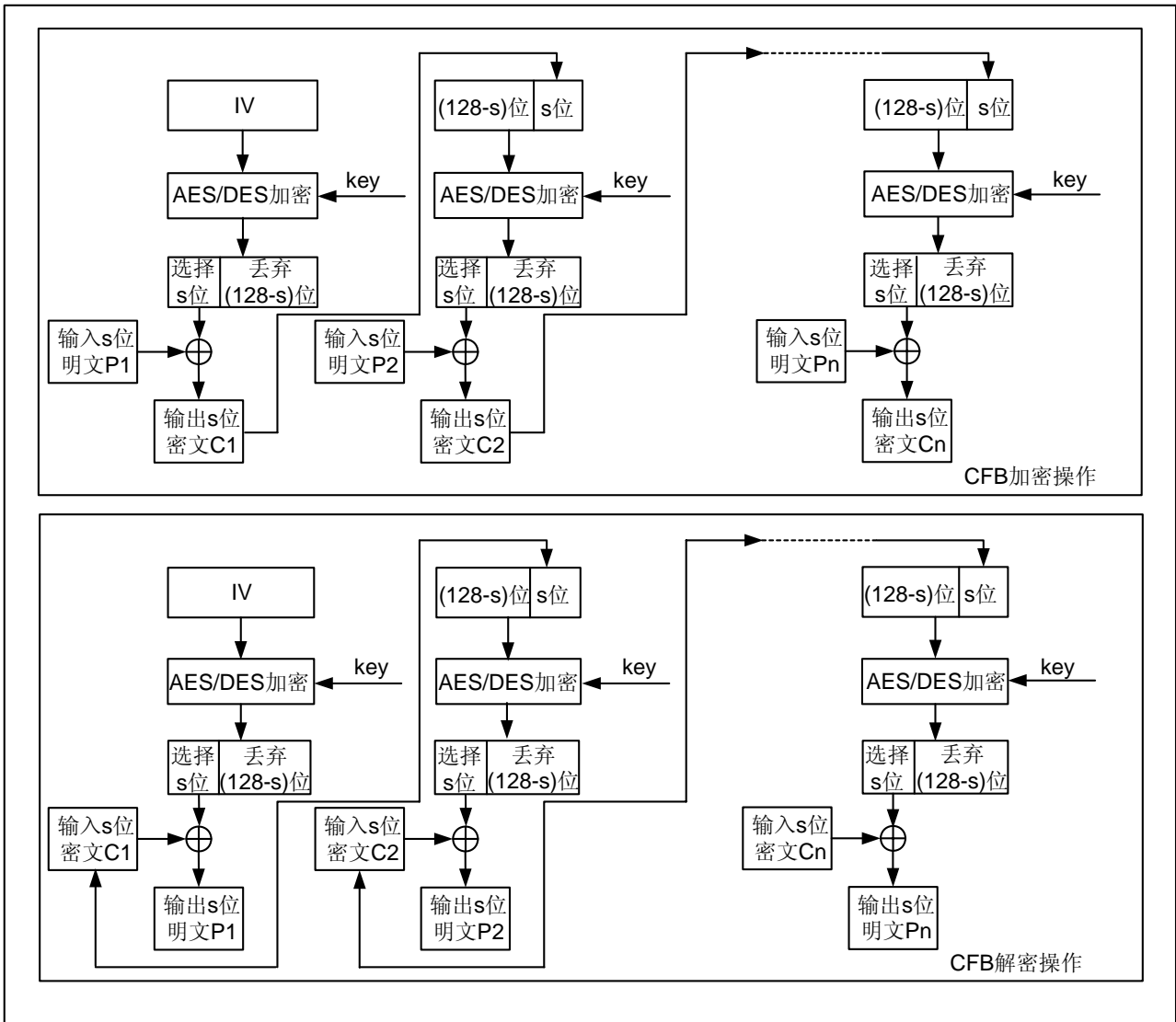
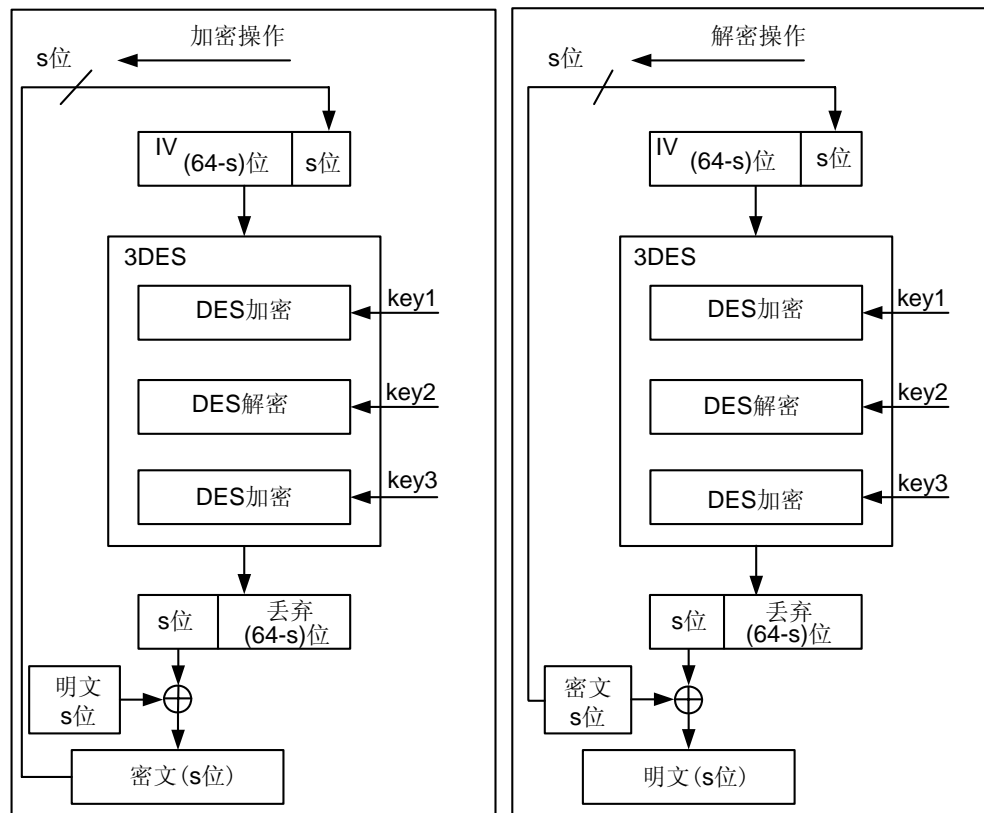


图3-20 3DES 的 s 位密码反馈（CFB）模式



OFB 模式

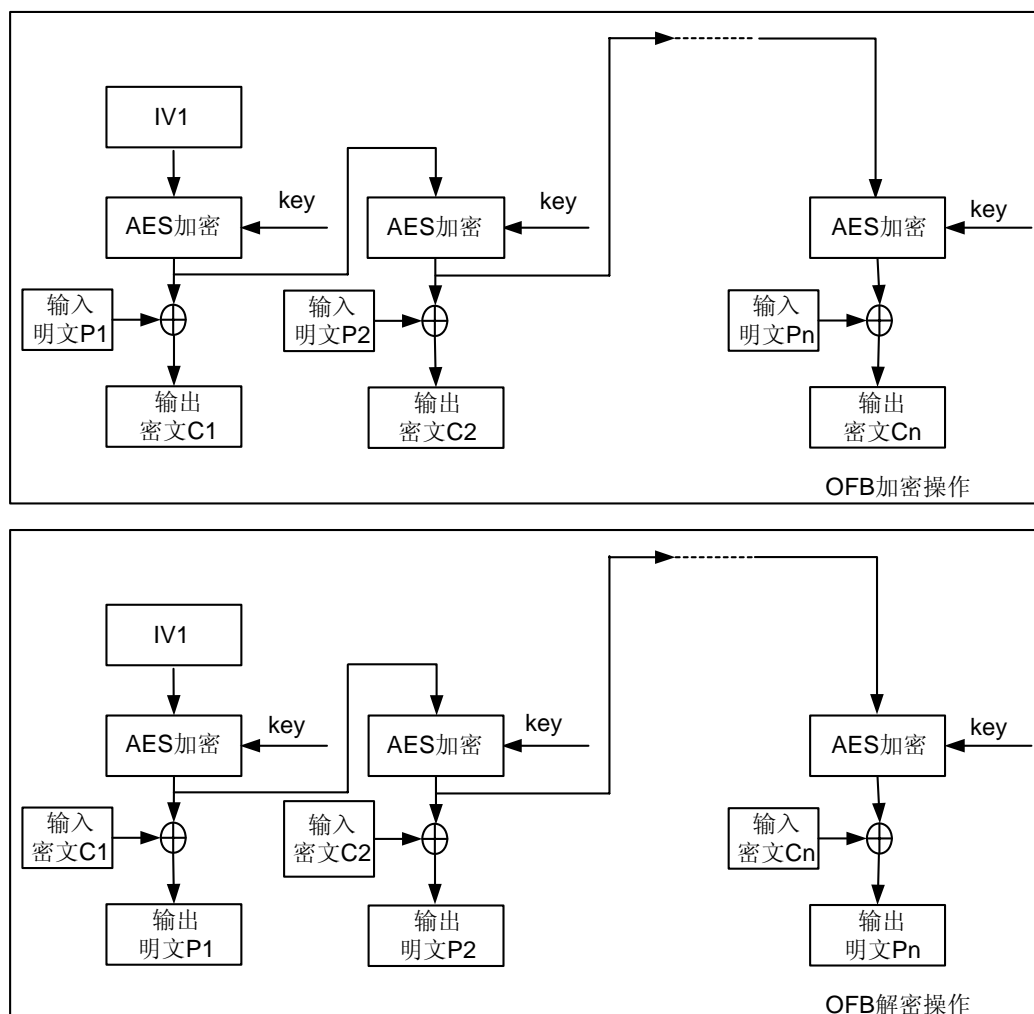
OFB（Output FeedBack）模式下，将 IV 直接作为加密操作的输入，因此对同一个密钥的操作情况下，应该使用不相同的 IV，避免降低操作的安全性。关于 s 位，存在以下 2 种情况：

- 对于 DES/3DES，s 位可以是 1 位、8 位或 64 位。
- 对于 AES，s 位只能是 128 位。



AES 的输出反馈（OFB）模式如图 3-21 所示。

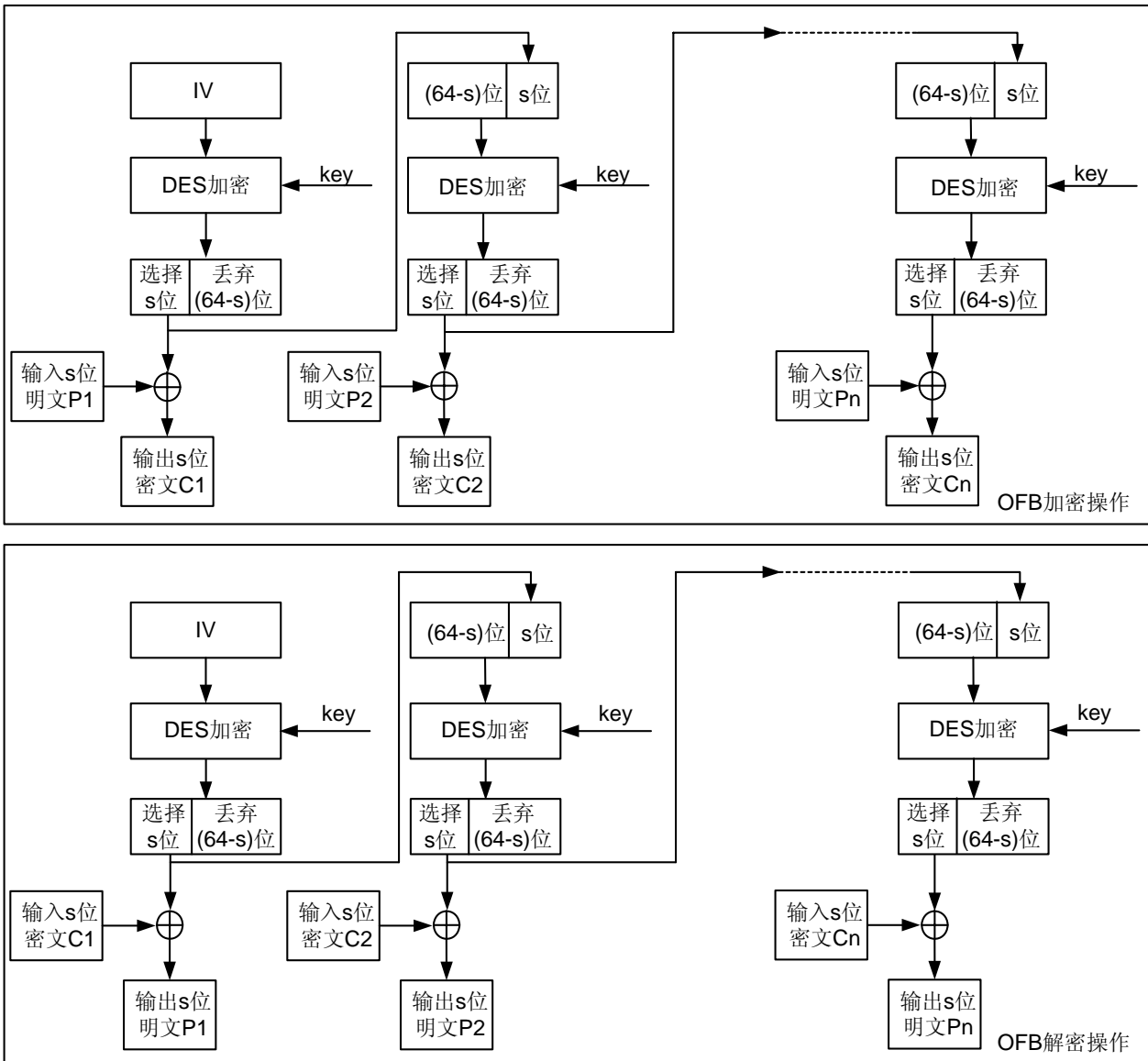
图3-21 AES 的输出反馈（OFB）模式



DES 的 s 位输出反馈模式如图 3-22 所示。

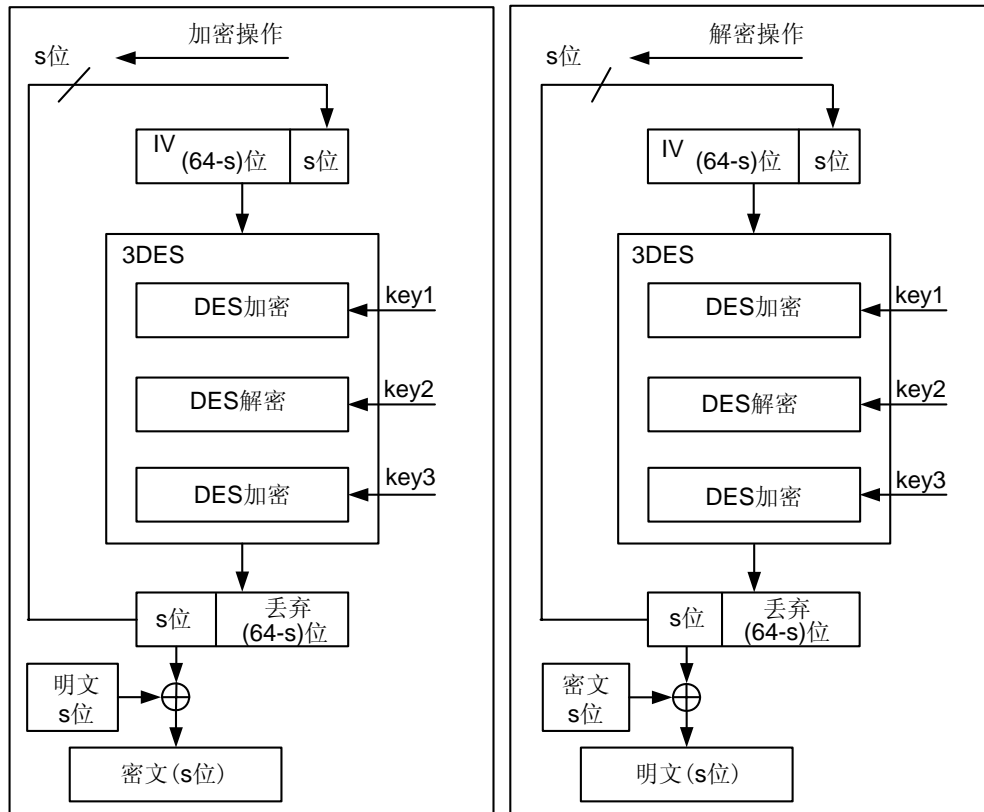


图3-22 DES 的 s 位输出反馈 (OFB) 模式



3DES 的 s 位输出反馈模式如图 3-23 所示。

图3-23 3DES 的 s 位输出反馈（OFB）模式



CTR 模式

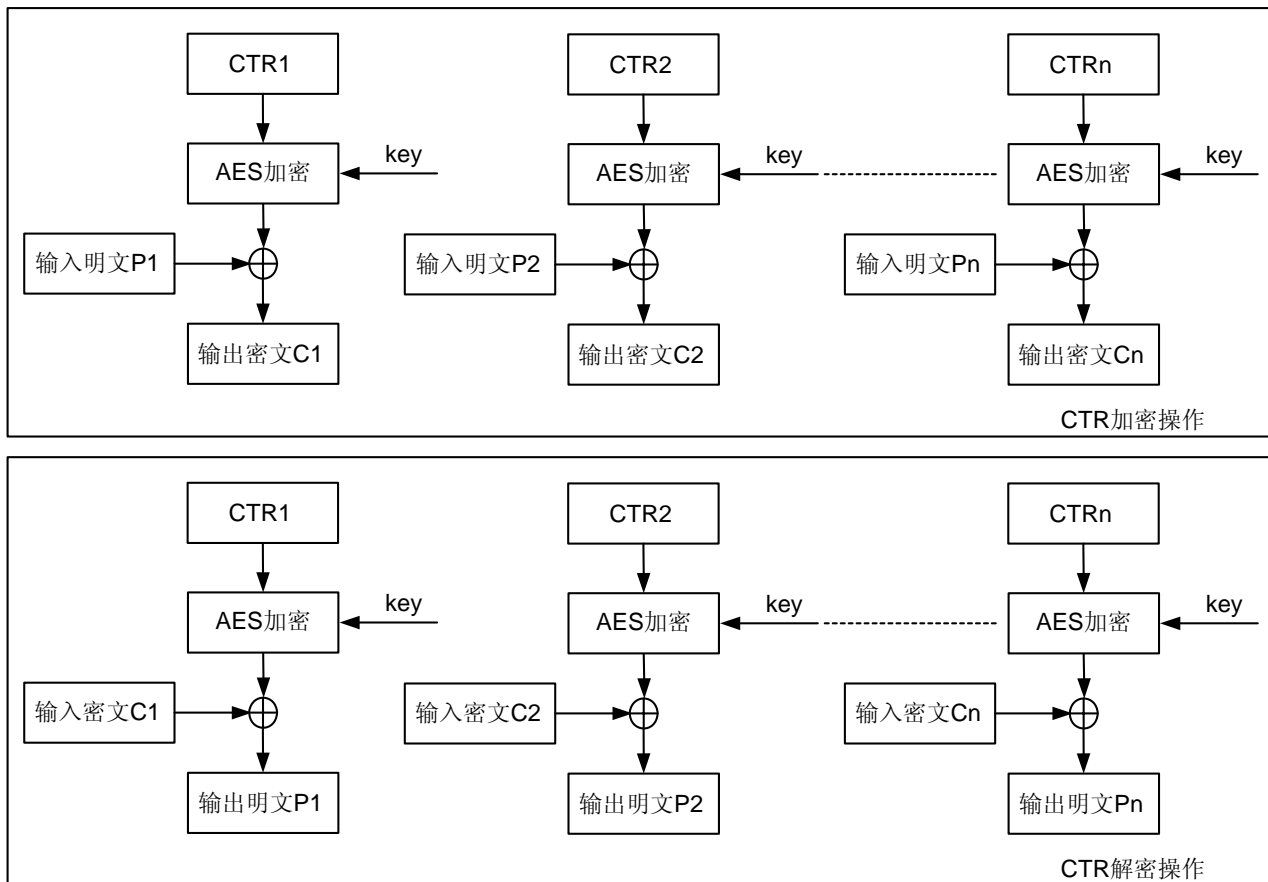
CTR (Counter) 模式下，向 AES 加密或解密处理模块输入不同的数据来保证数据处理的安全性，这种数据可以是计数的值。因此，计数值 CTRn 的选取也决定了这种方式应用的安全性。

说明

CTRn 一般采用累加计数的方式获取。

AES 的 CTR 模式如图 3-24 所示。

图3-24 AES 的 CTR 模式



3.6.4 工作方式

CIPHER 的单分组操作流程

CIPHER 的单分组操作流程如下：

- 步骤 1 读取状态寄存器 `CIPHER_BUSY`，如果为 0，则执行步骤 2，否则继续读取该寄存器。
- 步骤 2 配置控制寄存器 `CIPHER_CTRL`。
- 步骤 3 配置寄存器 `CIPHER_DIN`、`CIPHER_IVIN` 和 `CIPHER_KEY`。
- 步骤 4 配置启动寄存器 `CIPHER_ST` 为 0x1。
- 步骤 5 等待操作结束。主要有以下 2 种方式：
 - 中断方式：如果有中断，则执行步骤 6，否则继续等待中断。
 - 查询方式：读取状态寄存器 `CIPHER_BUSY`，如果状态寄存器是 0，则执行步骤 6，否则继续读取该寄存器。
- 步骤 6 读取结果寄存器 `CIPHER_DOUT`、`CIPHER_IVOUT`。

----结束



所有的工作模式都可以选择进行单分组操作流程。

如果需要使用单分组操作来实现一段数据的加、解密，则需要将上述中的流程执行多次。为了提高执行效率，需要注意以下几个方面：

- 对于同一段数据，如果操作的密钥相同，则只有第一个分组需要对 **CIPHER_KEY** 寄存器进行配置。
- 对于 CBC、CFB 和 OFB 工作模式，执行完一个分组操作之后，默认操作会自动将本次操作中的 **CIPHER_IVOUT** 的值作为下次的 **CIPHER_IVIN**，因此，启动下次操作有 2 种方法：
 - 配置 **CIPHER_CTRL**[*ivin_sel*]为 0，再配置 **CIPHER_DIN** 寄存器。
 - 读出本次的 **CIPHER_IVOUT** 中的值，配置到 **CIPHER_IVIN** 寄存器中，并配置 **CIPHER_CTRL**[*ivin_sel*]为 1，再配置 **CIPHER_CTRL** 寄存器。
- 对于 CBC、CFB 和 OFB 操作模式，如果本次操作的 **CIPHER_IVIN** 寄存器中的值与上个分组的 **CIPHER_IVOUT** 寄存器中的值没有关系，则必须重新配置 **CIPHER_IVIN** 寄存器，并将 **CIPHER_CTRL**[*ivin_sel*]置为 1。

CIPHER 的多分组操作流程

CIPHER 的多分组操作流程如下：

- 步骤 1 读取状态寄存器 **CIPHER_BUSY**，如果状态寄存器是 0，则执行步骤 2，否则继续读取该寄存器。
- 步骤 2 配置控制寄存器 **CIPHER_CTRL**。
- 步骤 3 配置数据存放的地址、数据量寄存器：**SRC_START_ADDR**、**DEST_START_ADDR** 和 **MEM_LENGTH**。
- 步骤 4 配置寄存器 **CIPHER_IVIN** 和 **CIPHER_KEY**。
- 步骤 5 配置启动寄存器 **CIPHER_ST** 为 0x1。
- 步骤 6 两种方式等待操作结束：
 - 中断方式，如果有中断，则执行步骤 7，否则继续等待中断。
 - 查询方式，读取状态寄存器 **CIPHER_BUSY**，如果状态寄存器是 0，则执行步骤 7，否则继续读取该寄存器。
- 步骤 7 读取结果寄存器 **CIPHER_IVOUT**。

----结束

除了 AES CTR 工作模式外，其它的工作模式都可以使用多分组操作流程。

多分组操作方式适用于进行大数据量的加解密运算的应用环境，这种操作方式减少了 CPU 在每次分组运算启动和结束时对寄存器的读写操作。CPU 配置完必需的控制和数据寄存器，再配置 **CIPHER_ST** 寄存器之后，硬件自动实现所有的运算和操作，直到数据运算完成。这种方式可释放 CPU 资源。



多分组操作方式一次只能对一个连续地址空间的数据（输入和结果数据存放的地址空间可以是不相同的）进行加解密操作，如果需要将多个不同地址空间的数据使用同一个密钥进行加密或解密操作，则需要注意以下的操作要求：

- 不需要配置 CIPHER_KEY 寄存器。
- 需要重新配置源地址、目的地址和数据长度等信息。
- 对于 CBC、CFB 和 OFB 操作模式，执行完一个分组操作之后，默认操作会自动将本次操作中 CIPHER_IVOUT 的值作为下次的 CIPHER_IVIN，因此，启动下次操作有 2 种方法：
 - 配置 CIPHER_CTRL[ivin_sel]为 0，再配置 CIPHER_DIN 寄存器。
 - 读出本次的 CIPHER_IVOUT 中的值，配置到 CIPHER_IVIN 寄存器中，并配置 CIPHER_CTRL[ivin_sel]为 1，再配置 CIPHER_DIN 寄存器。

时钟门控

当不需要进行加密操作，且状态寄存器 CIPHER_BUSY[cipher_busy]=0b0 时，可以通过配置系统控制器的寄存器，关断 CIPHER 模块时钟以降低功耗：

- 向 SC_PERDIS[cipherclkdis]写 1，禁止 CIPHER 模块时钟。
- 向 SC_PEREN[cipherclken]写 1，使能 CIPHER 模块时钟。

软复位

当判断出某单元出现问题时，可以通过配置系统控制器的寄存器，软复位恢复单元的错误。

- 向 SC_PERCTRL8[cipher_srst]写 1，对 CIPHER 模块软复位。
- 向 SC_PERCTRL8[cipher_srst]写 0，撤消对 CIPHER 模块的软复位，进行正常操作。

3.6.5 寄存器概览

表3-29 CIPHER 寄存器概览（基址是 0x100C_0000）

偏移地址	名称	描述	页码
0x000、0x004、 0x008、0x00C	CIPHER_DIN	CIPHER 模块的 128 位分组输入寄存器	3-76
0x010、0x014、 0x018、0x01C	CIPHER_IVIN	CIPHER 模块的向量分组的输入寄存器（ECB 工作模式下无需配置）	3-77
0x020、0x024、 0x028、0x02C 0x030、0x034、 0x038、0x03C	CIPHER_KEY	CIPHER 模块的密钥输入寄存器	3-78
0x040、0x044、 0x048、0x04C	CIPHER_DOUT	CIPHER 模块 128 位分组输出寄存器	3-79



偏移地址	名称	描述	页码
0x050、0x054、 0x058、0x05C	CIPHER_IVOUT	CIPHER 模块操作完成之后的向量输出寄存器（ECB、CTR 工作模式下无需关注）	3-80
0x060	CIPHER_CTRL	CIPHER 模块控制寄存器	3-81
0x064	INT_CIPHER	CIPHER 模块屏蔽后中断寄存器	3-84
0x068	CIPHER_BUSY	CIPHER 模块运算状态指示寄存器	3-85
0x06C	CIPHER_ST	CIPHER 运算启动/停止控制信号寄存器	3-85
0x070	SRC_START_A DDR	待处理分组数据存储在片外 memory 的起始地址寄存器	3-86
0x074	MEM_LENGTH	待处理分组数据的长度（以 32 位数据宽度衡量的长度）寄存器	3-86
0x078	DEST_START_ ADDR	运算结果分组存储在片外 memory 的起始地址寄存器	3-88
0x07C	INT_MASK	CIPHER 模块中断屏蔽寄存器	3-88
0x080	INT_CIPHER_S TATUS	CIPHER 模块中断状态寄存器	3-89

3.6.6 寄存器描述

CIPHER_DIN

CIPHER_DIN 为 CIPHER 模块的 128 位分组输入寄存器。

配置该寄存器时需要注意：

如果选择进行单分组的处理，即 CIPHER_CTRL[cipher_mode]=0b0 时，需要配置该寄存器：

- 如果选择进行 AES 运算（CIPHER_CTRL[alg_sel]=0b10）
 - 如果选择 1-CFB 操作（CIPHER_CTRL bit[5:1]=0b10010），低 1 位有效，即 CIPHER_DIN bit[0]为有效数据。
 - 如果选择 8-CFB 操作（CIPHER_CTRL bit[5:1]=0b01010），低 8 位有效，即 CIPHER_DIN bit[7:0]为有效数据。
 - 如果选择 128-CFB 操作（CIPHER_CTRL bit[5:1]=0b00010 或 0b11010），128 位数据均有效。
 - 如果选择其他操作模式，128 位数据均有效。
- 如果选择进行 DES 或 3DES 运算（CIPHER_CTRL[alg_sel]=0b00、0b01 或 0b11）



- 如果选择 1-CFB/1-OFB 操作（[CIPHER_CTRL](#) bit[5:1]=0b10010 或 0b10011），低 1 位有效，即 [CIPHER_DIN](#) bit[0]为有效数据。
- 如果选择 8-CFB/8-OFB 操作（[CIPHER_CTRL](#) bit[5:1]=0b01010 或 0b01011），低 8 位有效，即 [CIPHER_DIN](#) bit[7:0]为有效数据。
- 如果选择 64-CFB/64-OFB 操作（[CIPHER_CTRL](#) bit[5:1]=0b00010、0b11010、0b00011 或 0b11011），低 64 位数有效，即 [CIPHER_DIN](#) bit[63:0]为有效数据。
- 如果选择其他操作模式，低 64 位数有效，即 [CIPHER_DIN](#) bit[63:0]为有效数据。

如果选择进行多分组的处理，即 [CIPHER_CTRL](#)[cipher_mode]=0b1，不需要配置该寄存器。

Offset Address		Register Name		Total Reset Value				
0x000~0x00C		CIPHER_DIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	cipher_din							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	cipher_din	CIPHER 模块的 128 位分组输入，每个地址对应一个 32 位宽的数据。 CIPHER_DIN[31:0]: 0x000 地址; CIPHER_DIN[63:32]: 0x004 地址; CIPHER_DIN[95:64]: 0x008 地址; CIPHER_DIN[127:96]: 0x00C 地址。					

CIPHER_IVIN

CIPHER_IVIN 为 CIPHER 模块的向量分组的输入寄存器。

配置该寄存器时需要注意：

- 如果选择进行单分组的处理（[CIPHER_CTRL](#) bit[cipher_mode]=0b0），且执行的不是 ECB 模式（[CIPHER_CTRL](#)[mode]=0b001、0b010、0b011 或 0b100）时：
 - 如果选择不需要进行输入向量配置（[CIPHER_CTRL](#)[ivin_sel]=0b0），则不需要进行该寄存器的配置。
 - 如果选择需要进行输入向量配置（[CIPHER_CTRL](#)[ivin_sel]=0b1），则需要进行该寄存器的配置。如果此时选择进行 AES 运算（即 [CIPHER_CTRL](#) [alg_sel]=0b10），[CIPHER_IVIN](#) bit[127:0]为有效数据；如果选择进行 DES 或 3DES 运算（[CIPHER_CTRL](#)[alg_sel]=0b00、0b01 或 0b11），低 64 位数据有效，即 [CIPHER_IVIN](#) bit[63:0]为有效数据。
- 如果选择进行多分组的处理（[CIPHER_CTRL](#)[cipher_mode]=0b1），且执行的不是 ECB 模式（[CIPHER_CTRL](#)[mode]=0b001、0b010 或 0b011）时：

- 如果选择不需要进行输入向量配置，即 `CIPHER_CTRL[ivin_sel]=0b0`，则不需要进行该寄存器的配置，第一个分组运算的输入向量与该寄存器中的值无关。
- 如果选择需要进行输入向量配置，即 `CIPHER_CTRL[ivin_sel]=0b1`，则需要进行该寄存器的配置，第一个分组运算的输入向量从该寄存器获取。

Offset Address		Register Name		Total Reset Value				
0x010~0x01C		CIPHER_IVIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	cipher_ivin							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	cipher_ivin	CIPHER 模块的 128 位 IV 向量或 Counter 输入的数据，每个地址对应一个 32 位宽的数据。 CIPHER_IVIN[31:0]: 0x010 地址; CIPHER_IVIN[63:32]: 0x014 地址; CIPHER_IVIN[95:64]: 0x018 地址; CIPHER_IVIN[127:96]: 0x01C 地址。					

CIPHER_KEY

CIPHER_KEY 为 CIPHER 模块的密钥输入寄存器。

配置本寄存器时需要注意：

- 选择 DES 运算（`CIPHER_CTRL[alg_sel]=0b00` 或 `0b11`）时，低 64 位数据有效，即 `CIPHER_KEY[63:0]` 为有效数据。
- 选择 3DES 运算（`CIPHER_CTRL[alg_sel]=0b01`）：
选择 3 个密钥运算（`CIPHER_CTRL[key_length]=0b00`、`0b01` 或 `0b10`）时，低 192 位数据有效，此时：
 - `CIPHER_KEY` bit[63:0] 表示第一个密钥。
 - `CIPHER_KEY` bit[127:64] 表示第二个密钥。
 - `CIPHER_KEY` bit[191:128] 表示第三个密钥。
 选择 2 个密钥运算（即 `CIPHER_CTRL[key_length]=0b11`）时，低 128 位数据有效，此时：
 - `CIPHER_KEY` bit[63:0] 表示第一个密钥。
 - `CIPHER_KEY` bit[127:64] 表示第二个密钥。
- 选择 AES 运算（`CIPHER_CTRL[alg_sel]=0b10`）时：
 - 如果选择 128 位密钥操作（`CIPHER_CTRL[key_length]=0b00` 或 `0b11`），低 128 位数据有效，即 `CIPHER_KEY` bit[127:0] 为有效数据。



- 如果选择 192 位密钥操作（[CIPHER_CTRL\[key_length\]=0b01](#)），低 192 位数据有效，即 [CIPHER_KEY bit\[191:0\]](#) 为有效数据。
- 如果选择 256 位密钥操作（[CIPHER_CTRL\[key_length\]=0b10](#)），256 位数据均有效。

Offset Address		Register Name		Total Reset Value				
0x020~0x03C		CIPHER_KEY		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	cipher_key							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	cipher_key	CIPHER 模块的密钥输入，每 1 个地址对应一个 32 位宽的数据。 CIPHER_KEY[31:0]: 0x020 地址; CIPHER_KEY[63:32]: 0x024 地址; CIPHER_KEY[95:64]: 0x028 地址; CIPHER_KEY[127:96]: 0x02C 地址; CIPHER_KEY[159:128]: 0x030 地址; CIPHER_KEY[191:160]: 0x034 地址; CIPHER_KEY[223:192]: 0x038 地址; CIPHER_KEY[255:224]: 0x03C 地址。					

CIPHER_DOUT

CIPHER_DOUT 为 CIPHER 模块 128 位分组输出寄存器。

读取本寄存器时需要注意：

如果选择进行单分组的处理（[CIPHER_CTRL\[cipher_mode\]=0b0](#)），从该寄存器中读取的数据是单分组的运算的结果数据。AES 运算和 DES 或 3DES 运算对应的情况不同：

- 如果选择进行 AES 运算（[CIPHER_CTRL\[alg_sel\]=0b10](#)）
 - 如果选择 1-CFB 模式，即 [CIPHER_CTRL bit\[5:1\]=0b10010](#)，最低位有效，即 [CIPHER_DOUT bit\[0\]](#) 为有效数据。
 - 如果选择 8-CFB 模式，即 [CIPHER_CTRL bit\[5:1\]=0b01010](#)，低 8 位有效，即 [CIPHER_DOUT bit\[7:0\]](#) 为有效数据。
 - 如果选择 128-CFB 操作，即 [CIPHER_CTRL bit\[5:1\]=0b00010](#) 或 [0b11010](#)，128 位数据均有效。
 - 其它模式下 128 位数据均有效。
- 如果选择进行 DES 或 3DES 运算（[CIPHER_CTRL\[alg_sel\]=0b00、0b01 或 0b11](#)）

- 如果选择 1-CFB 或 1-OFB 模式，即 CIPHER_CTRL bit[5:1]=0b10010 或 0b10011，低 1 位有效，即 CIPHER_DOUT bit[0]为有效数据。
- 如果选择 8-CFB 或 8-OFB 模式，即 CIPHER_CTRL bit[5:1]=0b01010 或 0b01011，低 8 位有效，即 CIPHER_DOUT bit[7:0]为有效数据。
- 如果选择 64-CFB 或 64-OFB 模式，即 CIPHER_CTRL bit[5:1]=0b00010、0b00011、0b11010 或 0b11011，低 64 位数据有效，即 CIPHER_DOUT bit[63:0]为有效数据。
- 其它模式下低 64 位数据有效，即 CIPHER_DOUT bit[63:0]为有效数据。

如果选择进行多分组的处理，即 CIPHER_CTRL[ivin_sel]=0b1，从该寄存器中读取的数据是最后一个分组运算的结果数据。

Offset Address	Register Name	Total Reset Value						
0x040~0x04C	CIPHER_DOUT	0x0000_0000						
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0							
Name	cipher_dout							
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0							
Bits	Access	Name	Description					
[31:0]	RO	cipher_dout	CIPHER 模块 128 位分组输出，每 1 个地址对应 32 位的数据。 CIPHER_DOUT[31:0]: 0x040 地址; CIPHER_DOUT[63:32]: 0x044 地址; CIPHER_DOUT[95:64]: 0x048 地址; CIPHER_DOUT[127:96]: 0x04C 地址。					

CIPHER_IVOUT

CIPHER_IVOUT 为 CIPHER 操作完成之后的向量输出寄存器。

读取本寄存器时需要注意：

- 如果执行的是 ECB 或 CTR 工作模式 (CIPHER_CTRL[mode]=0b000、0b100、0b101、0b110 或 0b111)，不需要关注此寄存器。
- 如果选择进行单分组的处理 (CIPHER_CTRL[cipher_mode]=0b0)，该寄存器中的数据是该分组的向量结果输出，可以作为同一数据包的下一个分组运算的向量输入。
 - 如果选择进行 AES 运算 (CIPHER_CTRL[alg_sel]=0b10)，128 位数据均有效。
 - 如果选择进行 DES 或 3DES 运算 (CIPHER_CTRL[cipher_mode]=0b00、0b01 或 0b11)，低 64 位数据有效，即 CIPHER_IVOUT bit[63:0]为有效数据。
- 如果选择进行多分组的处理，即 CIPHER_CTRL[ivin_sel]=0b1，该寄存器中读取的数据是最后一个分组运算的向量结果输出。



- 如果选择进行 AES 运算（`CIPHER_CTRL[cipher_mode]=0b10`），128 位数据均有效。
- 如果选择进行 DES 或 3DES 运算（`CIPHER_CTRL[cipher_mode]=0b00、0b01 或 0b11`），低 64 位数据有效，即 `CIPHER_IVOUT` bit[63:0]为有效数据。

	Offset Address	Register Name	Total Reset Value
	0x050~0x05C	CIPHER_IVOUT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	cipher_ivout		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	cipher_ivout	<p>CIPHER 模块操作完成之后的向量 IV 的输出（ECB、CTR 工作模式下无需关注），每 1 地址对应 32 位的数据。</p> <p>CIPHER_IVOUT[31:0]: 0x050 地址；</p> <p>CIPHER_IVOUT[63:32]: 0x054 地址；</p> <p>CIPHER_IVOUT[95:64]: 0x058 地址；</p> <p>CIPHER_IVOUT[127:96]: 0x05C 地址。</p>

CIPHER_CTRL

CIPHER_CTRL 为控制 CIPHER 操作的寄存器。

配置本寄存器时需要注意：

- 在进行模块的其它寄存器配置之前，必须先配置该寄存器。
- AES 的 CTR 模式不支持多分组工作方式，即在选择 AES 的 CTR 工作模式时，不允许将 `CIPHER_CTRL[cipher_mode]`配置为 1。
- AES 下除了 CFB 模式之外，其它模式不允许将 `CIPHER_CTRL[width]`配置为 01 或 10。
- DES/3DES 下除了 CFB 和 OFB 模式之外，其它模式不允许将 `CIPHER_CTRL[width]`配置为 01 或 10。
- `CIPHER_CTRL[byte_seq_reg]`和 `CIPHER_CTRL[byte_seq_ram]`，分别针对寄存器配置/结果寄存器读取操作和对结果寄存器中的数据进行字节序调整。如果送给该模块的数据（以 DES 操作的数据为例）是字符流的形式，则需要启动字节调整。

例如文本“7654321”对应 0x3736_3534_3332_3120，以字符的方式接收数据之后，存放在存储器中的顺序如下：

- 0x0 地址存放的数据是：0x3435_3637
- 0x4 地址存放的数据是：0x2031_3233

假设数据是从 0x0 的偏移地址开始存放的，这种情况下，则需要把 `byte_seq_ram` 和 `byte_seq_reg` 位均配置为 1。

Offset Address		Register Name		Total Reset Value									
0x060		CIPHER_CTRL		0x0000_0000									
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0					
Name	reserved			byte_seq_ram	byte_seq_reg	dest_addr_set	cipher_mode	ivin_sel	alg_sel	key_length	width	mode	decrypt
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description										
[31:15]	-	reserved	保留。										
[14]	RW	byte_seq_ram	控制对 memory 地址空间的数据字节序调整。 0: 不进行字节序调整; 1: 进行字节序调整。										
[13]	RW	byte_seq_reg	控制对输入数据配置的字节序调整和输出数据寄存器读取的字节序调整。 0: 不进行字节序调整; 1: 进行字节序调整。 输入数据寄存器包括: CIPHER_KEY、CIPHER_IVIN、CIPHER_DIN。 输出数据寄存器包括: CIPHER_IVOUT、CIPHER_DOUT。										
[12]	RW	dest_addr_set	待处理分组数据和运算结果分组数据存储在片外 memory 的起始地址关系控制。 0: 待处理分组数据和运算结果分组数据存放的起始地址相同, 可以不配置 SRC_START_ADDR 和 DEST_START_ADDR; 1: 待处理分组数据和运算结果分组数据存放的起始地址不相同, 需要分别配置 SRC_START_ADDR 和 DEST_START_ADDR。										
[11]	RW	cipher_mode	CIPHER 模块工作方式选择控制。 0: 进行单分组的操作; 1: 进行多分组的操作。										
[10]	RW	ivin_sel	CIPHER_IVIN 的输入选择控制。 0: CIPHER_IVIN 不需要进行配置; 1: CIPHER_IVIN 需要配置。										



[9:8]	RW	alg_sel	<p>算法类型选择控制。</p> <p>00: DES 运算; 01: 3DES 运算; 10: AES 运算; 11: DES 运算。</p>
[7:6]	RW	key_length	<p>密钥长度控制。</p> <p>AES 算法下: 00: 128 位密钥长度; 01: 192 位密钥长度; 10: 256 位密钥长度; 11: 128 位密钥长度。</p> <p>3DES 算法下: 00: 192 位密钥长度 (3 个密钥); 01: 192 位密钥长度 (3 个密钥); 10: 192 位密钥长度 (3 个密钥); 11: 128 位密钥长度 (2 个密钥)。</p>
[5:4]	RW	width	<p>位宽控制。</p> <p>DES/3DES 算法下: 00: 64 位模式; 01: 8 位模式; 10: 1 位模式; 11: 64 位模式。</p> <p>AES 算法下: 00: 128 位模式; 01: 8 位模式; 10: 1 位模式; 11: 128 位模式。</p>

[3:1]	RW	mode	<p>工作模式控制。</p> <p>在 AES 算法下： 000: ECB 模式； 001: CBC 模式； 010: CFB 模式； 011: OFB 模式； 100: CTR 模式； 其它: ECB 模式。</p> <p>在 DES 算法下： 000: ECB 模式； 001: CBC 模式； 010: CFB 模式； 011: OFB 模式； 其它: ECB 模式。</p>
[0]	RW	decrypt	<p>加解密控制。</p> <p>0: 加密； 1: 解密。</p>

INT_CIPHER

INT_CIPHER 为屏蔽后中断寄存器，即中断状态寄存器（INT_CIPHER_STATUS）经过中断屏蔽寄存器（INT_MASK）屏蔽之后的中断寄存器。

Offset Address	Register Name	Total Reset Value						
0x064	INT_CIPHER	0x0000_0000						
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0							
Name	reserved						int_error	int_done
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0							
Bits	Access	Name	Description					
[31:2]	-	reserved	保留。					
[1]	RO	int_error	<p>错误中断状态，CIPHER 模块访问总线出错，或使用非 word 方式访问。</p> <p>0: 无错误； 1: 有错误。</p>					



[0]	RO	int_done	CIPHER 运算完成中断。 0: 未完成操作; 1: 完成操作。
-----	----	----------	---

CIPHER_BUSY

CIPHER_BUSY 为运算状态指示寄存器。

	Offset Address	Register Name	Total Reset Value
	0x068	CIPHER_BUSY	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		cipher_busy
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RO	cipher_busy	CIPHER 运算状态。 0: 没有进行 CIPHER 运算; 1: 正在进行 CIPHER 运算。

CIPHER_ST

CIPHER_ST 为 CIPHER 运算启动/停止控制信号寄存器。

配置本寄存器时需要注意：

- 如果向 [CIPHER_ST](#) 寄存器写 0x0000_0001 之后，CIPHER 模块将启动运算。
- 如果向 [CIPHER_ST](#) 寄存器写 0x0000_0000 之后，CIPHER 模块将停止运行。
- 如果选择进行多分组的处理，即 [CIPHER_CTRL](#) bit[11]=0b1，如果正在进行总线读写，则将完成总线读写之后，使模块停止运行；否则，硬件在读到该值之后，将立刻使模块停止运行。



Offset Address		Register Name		Total Reset Value					
0x06C		CIPHER_ST		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								cipher_st
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	cipher_st	CIPHER 运算启动/停止控制信号。 0: 停止 CIPHER 模块运算; 1: 启动 CIPHER 模块运算。						

SRC_START_ADDR

SRC_START_ADDR 为待处理分组数据存储于片外 memory 的起始地址寄存器。

如果选择进行多个分组的处理，即 CIPHER_CTRL[cipher_mode]=0b1，则需要在启动 CIPHER 模块之前，配置该寄存器。

Offset Address		Register Name		Total Reset Value				
0x070		SRC_START_ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	src_start_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	src_start_addr	待处理分组数据存储于片外 memory 的起始地址。					

MEM_LENGTH

MEM_LENGTH 为待处理分组数据的长度（以 32 位数据宽度衡量的长度）寄存器。

配置本寄存器时需要注意：

- 如果选择进行多分组的处理（CIPHER_CTRL[cipher_mode]=0b1），则需要在启动 CIPHER 模块之前，配置该寄存器。
- 在选择 AES 运算时：



- 如果选择 ECB、CBC、OFB 或 CTR 模式（即 `CIPHER_CTRL[mode]=0b000、0b001、0b011、0b100、0b101、0b110 或 0b111`），要求 `mem_length` 的值可以整除 4。
- 如果选择 128-CFB 模式（`CIPHER_CTRL[mode]=0b010` 且 `CIPHER_CTRL[width]=0b00 或 0b11`），要求 `mem_length` 的值可以整除 4。
- 如果选择执行 8-CFB 模式（`CIPHER_CTRL[mode]=0b010` 且 `CIPHER_CTRL[width]=0b01`）时，对 `mem_length` 的值没有要求。填充数据时，要求将数据填充在高位，有效数据从低字节开始有效。例如，`mem_length=0x0000_0001`，表示仅有一个地址数据有效，该地址的数据用 `data[31:0]` 表示，如果实际数据中，只有一个字节有效，即 `data[7:0]` 是有效的数据，则其它数据是填充数据，对应的结果也是只有低字节有效；如果只有两个字节有效，即 `data[15:0]` 是有效数据，其它数据是填充数据，对应的结果，也是只有低 16 位数据有效；依此类推。
- 选择执行 1-CFB 模式，即 `CIPHER_CTRL[mode]=0b010`，且 `CIPHER_CTRL[width]=0b10` 时，对 `mem_length` 的值没有要求。填充数据时，要求将数据填充在高位，有效数据从低位开始有效。例如，`mem_length=0x0000_0001`，表示仅有一个地址数据有效，该地址的数据用 `data[31:0]` 表示，如果实际数据中，只有一位有效，即 `data[0]` 是有效的数据，其它数据是填充数据，对应的结果，也是只有最低位有效；如果只有两位有效，即 `data[1:0]` 是有效数据，其它数据是填充数据，对应的结果，也是只有低 2 位数据有效；依此类推。
- 在选择 DES/3DES 运算时：
 - 选择执行 ECB 或 CBC 模式（`CIPHER_CTRL[mode]=0b000、0b001、0b100、0b101、0b110 或 0b111`），要求 `mem_length` 的值是偶数。
 - 选择执行 64-CFB 或 64-OFB 模式（`CIPHER_CTRL[mode]=0b010 或 0b011`，且 `CIPHER_CTRL[width]=0b00 或 0b11`），要求 `mem_length` 的值是偶数。
 - 选择执行 8-CFB 或 8-OFB 模式（`CIPHER_CTRL[mode]=0b010 或 0b011`，且 `CIPHER_CTRL[width]=0b01`），对 `mem_length` 的值没有要求，但是，填充数据时，要求将数据填充在高位，有效数据从低字节开始有效。例如，`mem_length=0x0000_0001`，表示仅有一个地址数据有效，该地址的数据用 `data[31:0]` 表示，如果实际数据中，只有一个字节有效，即 `data[7:0]` 是有效的数据，其它数据是填充数据，对应的结果，也是只有低字节有效；如果只有两个字节有效，即 `data[15:0]` 是有效数据，其它数据是填充数据，对应的结果，也是只有低 16 位数据有效；依此类推。
 - 选择执行 1-CFB 或 8-OFB 模式（`CIPHER_CTRL[mode]=0b010 或 0b011`，且 `CIPHER_CTRL[width]=0b10`），对 `mem_length` 的值没有要求，但是，填充数据时，要求将数据填充在高位，有效数据从低位开始有效。例如，`mem_length=0x0000_0001`，表示仅有一个地址数据有效，该地址的数据用 `data[31:0]` 表示，如果实际数据中，只有一位有效，即 `data[0]` 是有效的数据，其它数据是填充数据，对应的结果，也是只有最低位有效；如果只有两位有效，即 `data[1:0]` 是有效数据，其它数据是填充数据，对应的结果，也是只有低 2 位数据有效；依此类推。



	Offset Address	Register Name	Total Reset Value
	0x074	MEM_LENGTH	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	mem_length		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RW	mem_length
	Description		
	待处理分组数据的长度（以 32 位数据宽度衡量的长度）。		

DEST_START_ADDR

DEST_START_ADDR 为运算结果分组存储在片外 memory 的起始地址寄存器。

配置本寄存器时需要注意：

- 如果待处理分组数据和运算结果分组数据存放的起始地址相同，即 `CIPHER_CTRL[dest_addr_set]=0b0` 时，不需要配置该寄存器。
- 如果待处理分组数据和运算结果分组数据存放的起始地址不相同，即 `CIPHER_CTRL[dest_addr_set]=0b1` 时，需要配置该寄存器。

	Offset Address	Register Name	Total Reset Value
	0x078	DEST_START_ADDR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	dest_start_addr		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RW	dest_start_addr
	Description		
	运算结果分组存储在片外 memory 的起始地址。		

INT_MASK

INT_MASK 为中断屏蔽寄存器。该寄存器用来对中断状态寄存器进行屏蔽控制，决定是否产生中断。向 INT_MASK 寄存器写不同的值对应以下情况：

- 向 INT_MASK 寄存器写 0x0000_0003 之后，CIPHER 模块将屏蔽 `INT_CIPHER_STATUS` 寄存器的状态，即不产生中断，即 `INT_CIPHER bit[1:0]=0b00`。
- 向 INT_MASK 寄存器写 0x0000_0001 之后，CIPHER 模块将屏蔽 `INT_CIPHER_STATUS` 寄存器中的 `int_done_status` 状态，该状态不触发中断，即 `INT_CIPHER[int_done]=0b0`。



- 向 `INT_CIPHER_STATUS` 寄存器写 `0x0000_0002` 之后，CIPHER 模块将屏蔽 `INT_CIPHER_STATUS` 寄存器中的 `int_error_status` 状态，该状态不触发中断，即 `INT_CIPHER[int_error]=0b0`。

	Offset Address				Register Name				Total Reset Value																							
	0x07C				INT_MASK				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										<code>int_error_mask</code>	<code>int_done_mask</code>				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:2]	-	reserved		保留。																											
	[1]	RW	<code>int_error_mask</code>		<code>int_error</code> 的屏蔽控制。 0: 产生中断; 1: 不产生中断。																											
	[0]	RW	<code>int_done_mask</code>		<code>int_done</code> 的屏蔽控制。 0: 产生中断; 1: 不产生中断。																											

INT_CIPHER_STATUS

`INT_CIPHER_STATUS` 为中断状态寄存器。该寄存器是写 1 清 0 寄存器，向 `INT_CIPHER_STATUS` 寄存器写不同的值对应以下情况：

- 向 `INT_CIPHER_STATUS` 寄存器写 `0x0000_0003` 之后，CIPHER 模块将 `INT_CIPHER_STATUS` 寄存器清 0。
- 向 `INT_CIPHER_STATUS` 寄存器写 `0x0000_0001` 之后，CIPHER 模块将清除 `INT_DONE_STATUS` 中断信号。
- 向 `INT_CIPHER_STATUS` 寄存器写 `0x0000_0002` 之后，CIPHER 模块将清除 `int_error_status` 中断信号。

Offset Address		Register Name		Total Reset Value					
0x080		INT_CIPHER_STATUS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved							int_error_status	int_done_status
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:2]	-	reserved	保留。						
[1]	WC	int_error_status	错误中断状态，CIPHER 模块的 Master 接口访问 AHB 出错，以及不使用 word 方式访问 slave 接口出错。 0：无错误； 1：有错误。						
[0]	WC	int_done_status	CIPHER 运算完成中断状态。 0：未完成操作； 1：完成操作。						

3.7 Timer

3.7.1 概述

Timer 模块主要实现定时、计数功能，可以供操作系统用作系统时钟，也可以供应用程序用作定时和计数。Timer 模块提供 4 组 Dual-Timer 模块：Dual-Timer0、Dual-Timer1、Dual-Timer2 和 Dual-Timer3。4 组 Dual-Timer 模块自身并无差异，除基地址不同以外，各自在系统中的应用也有所不同。其中：

- Dual-Timer0 包括 Timer0、Timer1，他们共用同一个基地址和同一根中断线。
- Dual-Timer1 包括 Timer2、Timer3，他们共用同一个基地址和同一根中断线。
- Dual-Timer2 包括 Timer4、Timer5，他们共用同一个基地址和同一根中断线。
- Dual-Timer3 包括 Timer6、Timer7，他们共用同一个基地址和同一根中断线。

每组 Dual-Timer 模块包含两个功能完全相同的 Timer。

Dual-Timer0、Dual-Timer1 产生的中断送给 ARM1176；Dual-Timer2、Dual-Timer3 产生的中断送给 ARM926。



3.7.2 特点

Dual-Timer 模块具有以下特点：

- 有 2 个带可编程 8 位预分频器的 32bit/16bit 减法定时器/计数器。
- 计数时钟可配置，可选为系统 APB 总线时钟或晶振分频时钟（3MHz）。
- 支持 3 种计数模式：自由运行模式、周期模式和单次计数模式。
- 有 2 种载入计数初值的方法，分别通过 `TIMERx_LOAD` 和 `TIMERx_BGLOAD` 寄存器实现。
- 当前的计数值可随时读取。
- 当计数值减到 0 时会产生一个中断。

3.7.3 功能描述

典型应用

Hi3520 中的 Timer 主要是供软件使用。基于各种应用，Hi3520 的 4 组 Dual-Timer 提供不同的计时时钟配置。

功能原理

Timer 基于一个 32bit/16bit（可配置）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 3 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。当计数长度为 32bit 时，最大值为 `0xFFFF_FFFF`。当计数长度为 16bit 时，最大值为 `0xFFFF`。在自由模式下，也可以载入计数值，并立即从载入值递减计数，但计到 0 时回转到其最大值。
- 周期模式
定时器持续计数，当计数值减到 0 时从 `TIMERx_BGLOAD` 寄存器中再次载入初值并继续计数。
- 单次计数模式
向定时器中载入计数初值。当定时器的计数值减到 0 时就停止计数，直到重新被载入新值且定时器处于使能状态，才再次开始计数。

每个 Timer 具有一个预分频计数器（prescaler），可将其工作时钟在 Timer 内部再次进行 1 分频、16 分频或 256 分频。进一步提高计数时钟频率的选择灵活性。

对定时器载入计数初值的方法如下：

- 通过写 `TIMERx_LOAD` 寄存器可对定时器载入计数初值。当定时器处于工作状态时，如果向 `TIMERx_LOAD` 寄存器写入值，会导致定时器立刻从新值开始重新计数。适用于所有计数模式。



- 通过写 `TIMERx_BGLOAD` 寄存器可以设定周期计数模式的计数周期。写该寄存器不会立刻影响定时器的当前计数，定时器会继续计数直到计数值减到 0。然后载入 `TIMERx_BGLOAD` 寄存器中的新值开始计数。

3.7.4 工作方式

初始化

系统初始化时应对 Timer 进行初始化。初始化 TimerX（TimerX 中的“X”取值为 0、1、2、3、4、5、6 和 7）时应按以下步骤进行配置：

- 步骤 1 写 `TIMERx_LOAD` 寄存器，为 Timer 载入计数初值。
- 步骤 2 当需要 Timer 工作在周期计数模式下、且计数周期与载入 Timer 的计数初值不同时，写 `TIMERx_BGLOAD` 寄存器，设置 Timer 的计数周期。
- 步骤 3 配置系统控制器的 `SC_CTRL` 寄存器，设置 Timer 的时钟使能信号的参考时钟。
- 步骤 4 配置 `TIMERx_CONTROL` 寄存器，设置 Timer 的计数模式、计数器长度、预分频因子及中断屏蔽，同时启动 Timer 计数。

----结束

中断处理

Timer 主要用于定时产生中断，因此 Timer 的中断处理主要是激活等待定时中断的进程。操作步骤如下：

- 步骤 1 配置 `TIMERx_INTCLR` 寄存器，清除 Timer 中断。
- 步骤 2 激活等待该中断的进程，使其继续执行。
- 步骤 3 当所有等待该中断的进程完成或再次执行到等待中断的休眠状态时，恢复中断现场，继续执行当前被中断的程序。

----结束

时钟选择

4 组 Timer1 均有 2 种计数时钟可选择，下面以 Timer0 为例，时钟选择配置流程如下：

选择 APB 总线时钟进行计数的步骤如下：

- 步骤 1 配置系统控制器的 `SC_CTRL[timeren0ov]=1`。
- 步骤 2 初始化 Timer，开始计数。

----结束

选择 3MHz 时钟进行计数的步骤如下：

- 步骤 1 配置系统控制器的 `SC_CTRL[timeren0sel]=0`。
- 步骤 2 初始化 Timer，开始计数。



----结束

3.7.5 寄存器概览

Timer 模块中的 8 个寄存器除基址和偏移地址各不相同外其他特性都相同。其中：

- Timer0、Timer1 共用一个基址：0x2000_0000。
- Timer2、Timer3 共用一个基址：0x2001_0000。
- Timer4、Timer5 共用一个基址：0x2002_0000。
- Timer6、Timer7 共用一个基址：0x2003_0000。

说明

TIMERx 中的“x”取值为 0、1、2、3、4、5、6、7。

表3-30 Timer 寄存器概览（基址是 0x2000_0000、0x2001_0000、0x2002_0000、0x2003_0000）

Timer0/2/4/6 的偏移地址	Timer1/3/5/7 的偏移地址	名称	描述	页码
0x000	0x020	TIMERx_LOAD	计数初值寄存器	3-93
0x004	0x024	TIMERx_VALUE	当前计数值寄存器	3-94
0x008	0x028	TIMERx_CONTROL	控制寄存器	3-95
0x00C	0x02C	TIMERx_INTCLR	中断清除寄存器	3-98
0x010	0x030	TIMERx_RIS	原始中断寄存器	3-98
0x014	0x034	TIMERx_MIS	屏蔽后中断寄存器	3-99
0x018	0x038	TIMERx_BGLOAD	周期模式计数初值寄存器	3-100

3.7.6 寄存器描述

说明

- TIMER0_XXXX、TIMER2_XXXX、TIMER4_XXXX 和 TIMER6_XXXX 的偏移地址相同，关于这 4 个相似的寄存器描述均以 TIMER0_XXXX 为例进行介绍。
- TIMER1_XXXX、TIMER3_XXXX、TIMER5_XXXX 和 TIMER7_XXXX 的偏移地址相同，关于这 3 个相似的寄存器描述均以 TIMER1_XXXX 为例进行介绍。

3.7.6.1 TIMERx_LOAD

TIMERx_LOAD 为计数初值寄存器。用来配置定时器的计数初值。Timer0~7 各有 1 个计数初值寄存器。

当定时器处于周期模式且计数值递减到 0 时，将 **TIMERx_LOAD** 的值重新载入计数器。当直接写 **TIMERx_LOAD** 寄存器时，定时器当前的计数器将在被 TIMCLKENx 使能的下一个 TIMCLK 的上升沿更新为写入值。



说明

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

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

如果在被 `TIMCLKENx` 使能的 `TIMCLK` 的上升沿到来之前，向 `TIMERx_BGLOAD` 寄存器和 `TIMERx_LOAD` 寄存器都写入了值，则在被 `TIMCLKENx` 使能的 `TIMCLK` 的下一个上升沿定时计数器的值首先更新为 `TIMERx_LOAD` 的写入值。此后，每当计数器递减到 0 时，重新载入 `TIMERx_BGLOAD` 与 `TIMERx_LOAD` 中最晚被写入的寄存器的写入值。

在分别对 `TIMERx_BGLOAD` 寄存器和 `TIMERx_LOAD` 寄存器进行了 2 次写入之后，读 `TIMERx_LOAD` 返回的值为 `TIMERx_BGLOAD` 的写入值。

TIMER0_LOAD

Offset Address	Register Name	Total Reset Value	
0x000	TIMER0_LOAD	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer0_load		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	timer0_load	Timer0 的计数初值。

TIMER1_LOAD

Offset Address	Register Name	Total Reset Value	
0x020	TIMER1_LOAD	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer1_load		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	timer1_load	Timer1 的计数初值。

3.7.6.2 TIMERx_VALUE

`TIMERx_VALUE` 为当前计数值寄存器。用于给出正在递减的计数器的当前值。
Timer0~7 各有 1 个当前计数值寄存器。



当向 **TIMERx_LOAD** 寄存器的写操作发生后，**TIMERx_VALUE** 在 PCLK 时钟域立刻反映出计数器的新载入值，不用等到下一个被 TIMCLKENx 使能的 TIMCLK 时钟沿到来。

说明

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

TIMER0_VALUE

	Offset Address	Register Name	Total Reset Value
	0x004	TIMER0_VALUE	0xFFFF_FFFF
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer0_value		
Reset	1 1		
Bits	Access	Name	Description
[31:0]	RO	timer0_value	正在递减的 Timer0 的当前值。

TIMER1_VALUE

	Offset Address	Register Name	Total Reset Value
	0x024	TIMER1_VALUE	0xFFFF_FFFF
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer1_value		
Reset	1 1		
Bits	Access	Name	Description
[31:0]	RO	timer1_value	正在递减的 Timer1 的当前值。

3.7.6.3 TIMERx_CONTROL

TIMERx_CONTROL 为控制寄存器。Timer0~7 各有 1 个控制寄存器。

说明

当选择用周期模式进行计数时，需要将 **TIMERx_CONTROL[timermode]** 置 1、**TIMERx_CONTROL[oneshot]** 置 0。

TIMER0_CONTROL

Offset Address		Register Name		Total Reset Value																												
0x008		TIMER0_CONTROL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																timeren	timermode	intenable	reserved	timerpre	timersize	oneshot									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:8]	-	reserved	保留。																													
[7]	RW	timeren	定时器使能。 0: 禁止; 1: 使能。																													
[6]	RW	timermode	定时器的计数模式。 0: 自由运行模式; 1: 周期模式。																													
[5]	RW	intenable	原始中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																													
[4]	-	reserved	保留。																													
[3:2]	RW	timerpre	设置 Timer 的预分频因子。 00: 不经过预分频, 时钟频率除以 1; 01: 4 级预分频, 将 Timer 时钟频率除以 16; 10: 8 级预分频, 将 Timer 时钟频率除以 256; 11: 未定义, 若设为该值, 相当于 8 级预分频, 将 Timer 时钟频率除以 256。																													
[1]	RW	timersize	16bit/32bit 计数器操作模式。 0: 16bit 计数器; 1: 32bit 计数器。																													
[0]	RW	oneshot	计数模式为单次计数模式还是周期计数模式。 0: 周期计数模式或自由运行模式; 1: 单次计数模式。																													



TIMER1_CONTROL

	Offset Address								Register Name								Total Reset Value															
	0x028								TIMER1_CONTROL								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								timeren	timermode	intenable	reserved	timerpre	timersize	oneshot	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:8]	-	reserved	保留。																													
[7]	RW	timeren	定时器使能。 0: 禁止; 1: 使能。																													
[6]	RW	timermode	定时器的计数模式。 0: 自由运行模式; 1: 周期模式。																													
[5]	RW	intenable	原始中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																													
[4]	-	reserved	保留。																													
[3:2]	RW	timerpre	设置 Timer 的预分频因子。 00: 不经过预分频, 时钟频率除以 1; 01: 4 级预分频, 将 Timer 时钟频率除以 16; 10: 8 级预分频, 将 Timer 时钟频率除以 256; 11: 未定义, 若设为该值, 相当于预分频因子等于 10。																													
[1]	RW	timersize	16bit/32bit 计数器操作模式。 0: 16bit 计数器; 1: 32bit 计数器。																													
[0]	RW	oneshot	计数模式为单次计数模式还是周期计数模式。 0: 周期计数模式; 1: 单次计数模式。																													



3.7.6.4 TIMERx_INTCLR

TIMERx_INTCLR 为中断清除寄存器。对该寄存器的任何写操作都会清除相应计数器的中断状态。Timer0~7 各有 1 个中断清除寄存器。



注意

本寄存器是只写寄存器，写进去任意值，都会引起 Timer 清中断，内部并不记忆写入的值，无复位值。

TIMER0_INTCLR

	Offset Address	Register Name	Total Reset Value
	0x00C	TIMER0_INTCLR	-
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer0_intclr		
Reset	? ?		
Bits	Access	Name	Description
[31:0]	WO	timer0_intclr	写该寄存器，清除 Timer0 的中断输出。

TIMER1_INTCLR

	Offset Address	Register Name	Total Reset Value
	0x02C	TIMER1_INTCLR	-
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timerx1_intclr		
Reset	? ?		
Bits	Access	Name	Description
[31:0]	WO	timer1_intclr	写该寄存器，清除 Timer1 的中断输出。

3.7.6.5 TIMERx_RIS

TIMERx_RIS 为原始中断寄存器。Timer0~7 各有 1 个原始中断寄存器。



TIMER0_RIS

Offset Address		Register Name		Total Reset Value					
0x010		TIMER0_RIS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								timer0ris
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。写入无效，读时返回 0。						
[0]	RO	timer0ris	Timer0 的原始中断状态。 0: 无中断; 1: 有中断。						

TIMER1_RIS

Offset Address		Register Name		Total Reset Value					
0x030		TIMER1_RIS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								timer1ris
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。写入无效，读时返回 0。						
[0]	RO	timer1ris	Timer1 的原始中断状态。 0: 无中断; 1: 有中断。						

3.7.6.6 TIMERx_MIS

TIMERx_MIS 为屏蔽后中断寄存器。Timer0~7 各有 1 个屏蔽后中断寄存器。



TIMER0_MIS

Offset Address		Register Name		Total Reset Value					
0x014		TIMER0_MIS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								timer0mis
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RO	timer0mis	屏蔽后的 Timer0 的中断状态。 0: 中断无效; 1: 中断有效。						

TIMER1_MIS

Offset Address		Register Name		Total Reset Value					
0x034		TIMER1_MIS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								timer1mis
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RO	timer1mis	屏蔽后的 Timer1 的中断状态。 0: 中断无效; 1: 中断有效。						

3.7.6.7 TIMERx_BGLOAD

TIMERx_BGLOAD 为周期模式计数初值寄存器。Timer0~7 各有 1 个周期模式计数初值寄存器。

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



该寄存器提供了访问 **TIMERx_LOAD** 寄存器的另一种方法。不同之处在于写入值到 **TIMERx_BGLOAD** 寄存器中不会导致定时器立即从新写入值开始计数。

TIMER0_BGLOAD

	Offset Address	Register Name	Total Reset Value
	0x018	TIMER0_BGLOAD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer0bgload		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	timer0bgload	Timer0 的计数初值。 注意：与 TIMERx_LOAD 寄存器有区别，具体请参见 TIMERx_LOAD 寄存器的描述。

TIMER1_BGLOAD

	Offset Address	Register Name	Total Reset Value
	0x038	TIMER1_BGLOAD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	timer1bgload		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	timer1bgload	Timer1 的计数初值。 注意：与 TIMERx_LOAD 寄存器有区别，具体请参见 TIMERx_LOAD 寄存器的描述。

3.8 看门狗

3.8.1 概述

看门狗 WatchDog 用于系统异常情况下，一定时间内发出复位信号，以复位整个系统。

3.8.2 特点

WatchDog 具备以下特点：



- 内部具有一个 32bit 减法计数器，计数时钟源可配置。
- 支持超时时间间隔（即计数初值）可配置。
- 支持寄存器锁定，防止寄存器被误改。
- 支持超时中断产生。
- 支持复位信号产生。
- 支持调试模式。

3.8.3 信号描述

表3-31 WatchDog 接口信号描述

信号名称	方向	描述	对应管脚
WDG_RST	O	WatchDog 输出的复位信号，低电平有效。	WDGRST



注意

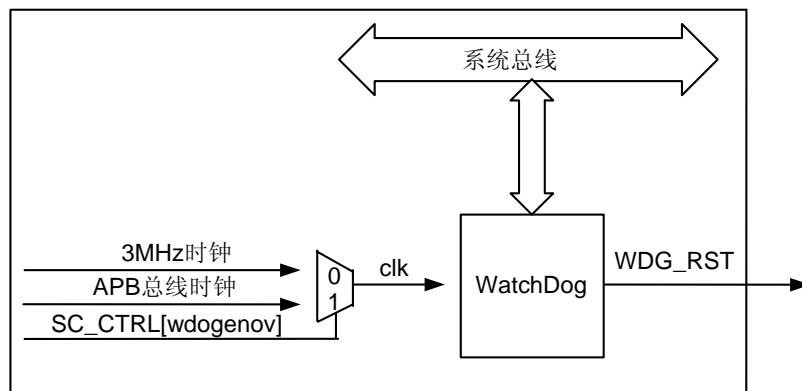
WDGRST 管脚为 OD（Open-Drain）门输出，需要外接上拉电阻。

3.8.4 功能描述

应用框图

WatchDog 应用框图如图 3-25 所示。

图3-25 WatchDog 应用框图





功能原理

WatchDog 的运行基于 1 个 32bit 减法计数器，计数初值由寄存器 `WDG_LOAD` 载入。在 WatchDog 时钟使能情况下，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 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 `WDG_LOAD` 中重新载入计数初值，开始递减计数。

如果计数器第二次计数递减到 0 时，CPU 还没有清除 WatchDog 中断，则 WatchDog 将发出复位信号 `WDG_RST`，计数器停止计数。

根据实际应用需要，可通过配置 `WDG_CONTROL` 使能或者禁止 WatchDog 产生中断和复位信号：

- 当禁止产生中断时，计数器将停止计数。
- 当重新开启中断时，WatchDog 将从 `WDG_LOAD` 的设定值开始计数，而不是从计数器上次停止时的计数值开始计数。在中断到来之前，可以重新载入初值。

WatchDog 的计数时钟可以选择 3MHz 时钟或者 APB 总线时钟，便于选择不同的计数时间范围。

通过配置 `WDG_LOCK` 寄存器，可以禁止对 WatchDog 内部寄存器进行写操作：

- 向 `WDG_LOCK` 写入 `0x1ACC_E551`，可以打开所有 WatchDog 寄存器的写权限。
- 向 `WDG_LOCK` 寄存器写入其他任何值，可以关闭所有 WatchDog 寄存器（`WDG_LOCK` 寄存器除外）的写权限。

该特性保护 WatchDog 的寄存器不被软件错误地修改，从而使得在异常情况下，WatchDog 不致被软件错误地中止操作。

在调试模式下，WatchDog 自动关闭，以防止干扰正常的调试操作。



注意

在系统进入 SLEEP 模式之前，必须关闭 WatchDog。具体操作请参见“3.8.5 工作方式”中的“关闭 WatchDog”。

两个处理器中任何一个进入 debug 模式时，WatchDog 自动关闭，debug 模式撤销，继续进行计数。

3.8.5 工作方式

计数时钟频率配置

系统支持 2 种 WatchDog 计数时钟：3MHz 时钟和 APB 总线时钟，通过 `SC_CTRL[wdogenov]` 进行配置。

WatchDog 计数时间为：

$$T_{\text{WDG}} = \text{Value}_{\text{WDG_LOAD}} \times (1/f_{\text{clk}})$$



说明

T_{WDG} 表示 WatchDog 计数时间， $\text{Value}_{\text{WDG_LOAD}}$ 表示 WatchDog 计数初值， f_{clk} 表示 WatchDog 计数时钟频率。

WatchDog 在不同时钟下的计数时间范围值如下：

- 当选择 3MHz 时钟（3MHz）时，计数时间范围为 0~1432s。
- 当选择 APB 总线时钟（100MHz）时，计数时间范围为 0~43s。

初始化

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 `WDG_LOAD`，设定计数初值。
 - 步骤 2 写寄存器 `WDG_CONTROL`，打开中断屏蔽并启动 WatchDog 计数。
 - 步骤 3 写寄存器 `WDG_LOCK`，给 WatchDog 上锁，防止软件错误修改 WatchDog 的配置。
- 结束

中断处理过程

收到 WatchDog 发出的中断后，应及时清除其中断状态，并使其载入计数初值重新开始计数。WatchDog 中断处理的过程如下所示：

- 步骤 1 向 `WDG_LOCK` 写 `0x1ACC_E551`，为 WatchDog 开锁。
 - 步骤 2 写寄存器 `WDG_INTCLR`，清除 WatchDog 的中断状态，同时也使 WatchDog 自动载入计数初值重新开始计数。
 - 步骤 3 向寄存器 `WDG_LOCK` 写入 `0x1ACC_E551` 以外的任何值，给 WatchDog 上锁。
- 结束

关闭 WatchDog

在系统进入 SLEEP 模式之前，必须关闭 WatchDog。向寄存器 `WDG_CONTROL[inten]` 控制位写入 0，即关闭 WatchDog。向该位写入 1，即打开 WatchDog。

3.8.6 寄存器概览

表3-32 WatchDog 寄存器概览（基址是 `0x2004_0000`）

偏移地址	名称	描述	页码
0x000	<code>WDG_LOAD</code>	计数初值寄存器	3-105
0x004	<code>WDG_VALUE</code>	计数器当前值寄存器	3-105
0x008	<code>WDG_CONTROL</code>	控制寄存器	3-106



偏移地址	名称	描述	页码
0x00C	WDG_INTCLR	中断清除寄存器	3-106
0x010	WDG_RIS	原始中断寄存器	3-107
0x014	WDG_MIS	屏蔽后中断寄存器	3-107
0x018~ 0xBFC	RESERVED	保留	-
0xC00	WDG_LOCK	LOCK 寄存器	3-108

3.8.7 寄存器描述

WDG_LOAD

计数初值寄存器，用来配置 WatchDog 内部计数器的计数初值。

Offset Address	Register Name	Total Reset Value	
0x000	WDG_LOAD	0xFFFF_FFFF	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	wdg_load		
Reset	1 1		
Bits	Access	Name	Description
[31:0]	RW	wdg_load	计数初值。

WDG_VALUE

计数器当前值寄存器，用来读出 WatchDog 内部计数器的当前计数值。

Offset Address	Register Name	Total Reset Value	
0x004	WDG_VALUE	0xFFFF_FFFF	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	wdogvalue		
Reset	1 1		
Bits	Access	Name	Description
[31:0]	RO	wdogvalue	WatchDog 计数器当前值。



WDG_CONTROL

控制寄存器，用来控制 WatchDog 的打开/关闭、中断和复位功能。

Offset Address	Register Name	Total Reset Value	
0x008	WDG_CONTROL	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		
Reset	0 0		
Bits	Access	Name	Description
[31:2]	-	reserved	保留。
[1]	RW	resen	WatchDog 复位信号输出使能。 0: 禁止; 1: 使能。
[0]	RW	inten	WatchDog 中断信号输出使能。 0: 计数器停止计数，计数值保持当前值不变，WatchDog 被关闭; 1: 既启动计数器又使能中断，WatchDog 被启动。 说明：若 inten 表示的中断之前被禁止，则当中断再次被使能时，计数器从 WDG_LOAD 中载入初值，并重新开始计数。

WDG_INTCLR

中断清除寄存器，用来清除 WatchDog 中断，使 WatchDog 重新载入初值进行计数。本寄存器是只写寄存器，写进去任意值，都会引起 WatchDog 清中断，内部并不记忆写入的值，无复位值。

Offset Address	Register Name	Total Reset Value	
0x00C	WDG_INTCLR	-	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	wdg_intclr		
Reset	? ?		
Bits	Access	Name	Description
[31:0]	WO	wdg_intclr	对该寄存器写入任意值可清除 WatchDog 的中断，并使 WatchDog 从寄存器 WDG_LOAD 中重新载入初值重新计数。



WDG_RIS

原始中断寄存器，用来反映 WatchDog 原始中断状态。

	Offset Address	Register Name	Total Reset Value
	0x010	WDG_RIS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		wdogris
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RO	wdogris	WatchDog 原始中断状态，当计数器的计数值递减到 0 时，该位置 1。 0: 未产生中断； 1: 已产生中断。

WDG_MIS

屏蔽后中断寄存器，用来反映屏蔽后的 WatchDog 中断状态。



	Offset Address	Register Name	Total Reset Value
	0x014	WDG_MIS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		wdogmis
Reset	0 0		
	Bits	Access	Name
	[31:1]	-	reserved
	[0]	RO	wdogmis
			Description
			保留。
			WatchDog 的屏蔽后中断状态。 0: 未产生中断或者中断被屏蔽; 1: 已产生中断。

WDG_LOCK

LOCK 寄存器，用来控制 WatchDog 寄存器的读写权限。

	Offset Address	Register Name	Total Reset Value
	0xC00	WDG_LOCK	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	wdg_lock		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RW	wdg_lock
			Description
			向该寄存器写入 0x1ACC_E551，可打开所有寄存器的写权限，写入其他值则关闭写权限。 读该寄存器返回加锁的状态而不是写入该寄存器的值： 0x0000_0000：允许写访问（未加锁）； 0x0000_0001：禁止写访问（已加锁）。

3.9 实时时钟

3.9.1 概述

实时时钟 RTC（Real Time Clock）用于实现时间显示和定时报警功能。



3.9.2 特点

RTC 具备以下特点：

- 内部具有 1 个 32bit 加法计数器
- 计数时钟 1Hz
- 计数初值可配置
- 计数比较值可配置
- 支持超时中断产生
- 支持软复位

3.9.3 功能描述

RTC 的运行基于 1 个 32bit 加法计数器，计数初值由寄存器 `RTC_LR` 载入。计数器的值在每个计数时钟的上升沿加 1。当计数值递加到 `RTC_LR` 寄存器与 `RTC_MR` 寄存器值相等时，RTC 将产生一个中断，然后在下一个计数时钟上升沿，计数器继续递加计数。

RTC 在系统中的中断号为“10”。

根据实际应用需要，可通过配置 `RTC_IMSC` 使能或者禁止 RTC 产生中断信号。此时，存在以下两种情况：

- 当禁止产生中断时，RTC 计数器继续递加计数，将不会对外产生中断，在 `RTC_MIS` 中显示屏蔽后中断的状态，在 `RTC_RIS` 中显示原始中断状态。
- 当重新开启中断时，RTC 计数器仍然继续递加计数，当计数值递加到 `RTC_LR` 寄存器与 `RTC_MR` 寄存器值相等时，RTC 将产生一个中断。


RTC 的计数时钟采用的是 1Hz 时钟，便于通过计数值转换为具体的年、月、日、时、分、秒。

3.9.4 工作方式

计数时钟频率

RTC 采用 1Hz 时钟进行计数，计数最大时间为：

$$T_{\text{RTC}} = (2^{32} - 1) \times (1/f_{\text{rtcclock}}) = 4294967295(\text{秒}) \approx 49710(\text{天})$$

 说明

T_{RTC} 表示 RTC 计数时间， $2^{32} - 1$ 表示 RTC 计数最大值， f_{rtcclock} 表示 RTC 计数时钟频率：1Hz。

软复位

通过配置系统控制器 `SC_PERCTRL8[rtc_srst]`，可以实现对 RTC 的单独软复位。软复位后各个 RTC 配置寄存器的值均恢复为默认值，因此软复位后需要重新对这些寄存器进行初始化配置。



软复位步骤如下：

- 步骤 1 向 SC_PERCTRL8[rtc_srst]写 1，对 RTC 软复位。
 - 步骤 2 向 SC_PERCTRL8[rtc_srst]写 0，撤消对 RTC 的软复位。
- 结束

系统初始化

系统上电复位后，RTC 计数器处于停止计数状态，在系统初始化过程中需要将 RTC 初始化并启动其运行。RTC 的初始化过程如下：

- 步骤 1 配置 RTC_CR[rtc_start]=0b1，启动 RTC 计数器开始计数。
 - 步骤 2 配置 RTC_IMSC[rtc_imsc]=0b0，设置 RTC 中断屏蔽位。
 - 步骤 3 配置 RTC_MR，设置 RTC 比较值。
 - 步骤 4 配置 RTC_LR，设置 RTC 计数初始值。
 - 步骤 5 RTC 按照 1Hz 的计数时钟频率，从 RTC_LR 中的值开始计数，当计数到 RTC_MR 中的值时，将根据 RTC_IMSC 的设置，决定是否产生中断。
- 结束

中断处理

系统收到 RTC 发出的中断后，表示定时时间到，随后转入“定时开机”、“定时关机”等相应操作，RTC 计数器仍然保持递加计数。RTC 中断处理的过程如下：

- 步骤 1 配置 RTC_ICR[rtc_icr]=0b1，清除 RTC 的中断状态。
 - 步骤 2 如果需要继续设置定时时间，则向寄存器 RTC_MR 写入新的比较值。
- 结束

关闭 RTC

一旦配置 RTC_CR，启动 RTC 计数后，RTC 将一直处于计数状态。只有对 RTC 复位后，才能关闭 RTC。对 RTC 的软复位操作请参见“0 软复位”操作。

3.9.5 寄存器概览

表3-33 RTC 寄存器概览（基址是 0x2006_0000）

偏移地址	名称	描述	页码
0x000	RTC_DR	计数器当前值寄存器	3-111
0x004	RTC_MR	RTC 比较寄存器	3-111
0x008	RTC_LR	RTC 加载寄存器	3-112



偏移地址	名称	描述	页码
0x00C	RTC_CR	RTC 使能寄存器	3-112
0x010	RTC_IMSC	中断屏蔽寄存器	3-112
0x014	RTC_RIS	原始中断寄存器	3-113
0x018	RTC_MIS	屏蔽后中断寄存器	3-113
0x01C	RTC_ICR	中断清除寄存器	3-114

3.9.6 寄存器描述

RTC_DR

RTC_DR 为计数器当前值寄存器，用来读取 RTC 内部计数器的当前值。

Offset Address	Register Name	Total Reset Value	
0x000	RTC_DR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rtc_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	rtc_data	当前 RTC 计数值。

RTC_MR

RTC_MR 为 RTC 比较寄存器，用来设置 RTC 的比较值。

Offset Address	Register Name	Total Reset Value	
0x004	RTC_MR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rtc_match		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	rtc_match	设置的 RTC 比较值。



RTC_LR

RTC_LR 为 RTC 加载寄存器，用来设置 RTC 计数初始值。

	Offset Address	Register Name	Total Reset Value
	0x008	RTC_LR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rtc_load		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RW	rtc_load
			Description
			设置 RTC 计数初始值。

RTC_CR

RTC_CR 为控制寄存器，用来使能 RTC。一旦使能，只有系统复位才能清除该寄存器。对该寄存器的任何写操作不起作用。读则返回当前值。

	Offset Address	Register Name	Total Reset Value
	0x00C	RTC_CR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		
Reset	0 0		
	Bits	Access	Name
	[31:1]	-	reserved
			Description
			保留。
	[0]	RW	rtc_start
			Description
			RTC 使能。 0: 禁止。 1: 使能。

RTC_IMSC

RTC_IMSC 为中断屏蔽寄存器，用来反映 RTC 中断屏蔽状态。



	Offset Address	Register Name	Total Reset Value
	0x010	RTC_IMSC	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		rtc_imsc
Reset	0 0		
	Bits	Access	Name
	[31:1]	-	reserved
	[0]	RW	rtc_imsc
			Description
			保留。
			RTC 中断屏蔽控制。 0: 屏蔽。 1: 不屏蔽。

RTC_RIS

RTC_RIS 为原始中断状态寄存器，用来反映 RTC 原始中断状态。

	Offset Address	Register Name	Total Reset Value
	0x014	RTC_RIS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		rtc_ris
Reset	0 0		
	Bits	Access	Name
	[31:1]	-	reserved
	[0]	RO	rtc_ris
			Description
			保留。
			RTC 原始中断状态。 0: 未产生中断。 1: 已产生中断。

RTC_MIS

RTC_MIS 为 RTC 屏蔽后中断寄存器，用来反映屏蔽后的 RTC 中断状态。

Offset Address	Register Name	Total Reset Value	
0x018	RTC_MIS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved rtc_mis		
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RO	rtc_mis	RTC 屏蔽后的中断状态。 0: 未产生中断或者中断被屏蔽。 1: 已产生中断。

RTC_ICR

RTC_ICR 为 RTC 清除中断寄存器，用来清除 RTC 中断。

Offset Address	Register Name	Total Reset Value	
0x01C	RTC_ICR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved rtc_icr		
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	WO	rtc_icr	清除 RTC 中断。 0: 无影响。 1: 清除中断。



3.10 系统控制器

3.10.1 概述

系统控制器模块提供了控制系统运行的手段，它控制系统运行的模式，监控系统运行状态，管理系统中的重要模块（如时钟复位等），完成对外设的某些功能的配置。

3.10.2 特点

系统控制器具有以下特点：

- 控制并监控系统的运行模式
- 提供系统时钟控制和状态查询
- 提供对外设时钟门控的控制、复位控制和查询
- 系统的中断模式的控制
- 提供对系统地址重映射的控制和状态监控
- 提供通用外设控制寄存器对外设进行各种控制
- 提供对部分总线仲裁功能的控制
- 提供对关键寄存器的写保护功能
- 提供芯片的标识寄存器

3.10.3 功能描述

系统运行模式控制

系统有以下 4 种工作模式：

- **NORMAL 模式**
NORMAL 模式是系统正常工作模式。在此模式下，系统由片内 ARMPLL 的输出时钟驱动，所有的模块均能正常工作于此时钟源。
- **SLOW 模式**
SLOW 模式是慢速模式。在此模式下，系统由外接晶振时钟驱动，只有部分片内模块可以工作，如系统控制器、Timer、SMI 等。所有对高速时钟有要求的模块在此时钟下无法工作，如 DDRC、USB 2.0 HOST、由 CRG 提供时钟的 AMBA-PCI 桥、SIO0、SIO1、SIO2、MMC 等。
- **DOZE 模式**
DOZE 模式是低速模式。在此模式下，系统由外接晶振分频的 46.8kHz 低频时钟驱动。大部分片内外设和存储器接口无法在此模式下工作，CPU 和少量片内模块（如系统控制器、Timer 等）可以工作于该模式。
- **SLEEP 模式**
SLEEP 模式是休眠模式。在该模式下，CPU 和大部分模块因时钟关断而停止工作。只有系统控制器、IR 模块可以在晶振分频的 46.8kHz 低频时钟下工作，另外 AMBA-PCI 桥的 AHB 侧也保留低频时钟。



系统控制器提供了一个系统模式切换机制，用于控制系统时钟源的切换。

模式切换由模式控制寄存器 SC_CTRL[modectrl]配置，该域定义了系统当前需要进入的操作模式：

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

说明

X 表示可以为 0 或者 1。

当设置系统工作模式后，由状态机控制模式的自动切换，无需软件的干预。当前系统状态可通过读取 SC_CTRL[modestatus]获得。这几位描述的系统当前状态不仅包括了上述的 4 个主要模式：NORMAL、SLOW、DOZE、SLEEP，还包括 4 个主要模式之间的几个中间态：SW from PLL、SW to PLL、PLLCTL、SW from XTAL、SW to XTAL、XTALCTL。

说明

NORMAL、SLOW、DOZE、SLEEP 这 4 种模式的切换，可配置为直接切换，如系统当前处于 NORMAL 模式，可通过配置 SC_CTRL[modectrl]为“001”进入 DOZE 模式。但实际系统运行过程中，经历了 SW from PLL、SLOW、SW from XTAL 等模式或中间态。

上电复位后，系统控制器默认处于 SLOW 状态。

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

系统控制器和时钟模块配合完成系统时钟和系统模式的切换。当状态机状态发生迁移时，系统控制器发出时钟切换指示信号，时钟模块随之进行时钟切换，并向系统控制器反馈切换完成指示信号，系统控制器检测到切换完成指示信号，完成模式切换。

系统控制器状态机状态和系统时钟之间的关系如表 3-34 所示。

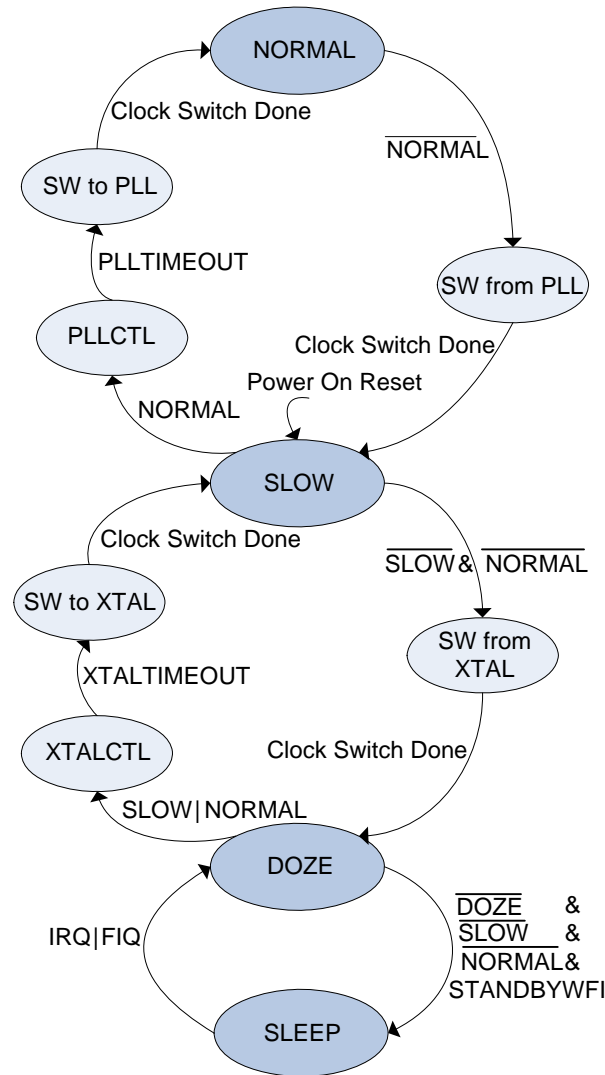
表3-34 系统控制器状态和时钟切换对应关系表

系统控制器状态	24MHz 晶振使能状态	主 PLL 使能状态	系统时钟状态
NORMAL	使能	使能	ARM 子系统的工作时钟来自 PLL 输出。
SLOW	使能	不使能	ARM 子系统的工作时钟来自 24MHz 晶振输入。
DOZE	使能	不使能	ARM 子系统的工作时钟都来自 46.8kHz 晶振分频时钟。
SLEEP	使能	不使能	除系统控制器和红外模块工作在 46.8kHz 外，其它模块的时钟都处于关闭状态。



系统的状态切换过程如图 3-26 所示。

图3-26 系统模式切换图



各种模式之间的切换涉及到的操作如下：

- SLEEP 模式

在 SLEEP 模式下，除系统控制器和 IR 模块时钟由低速 46.8kHz 晶振分频时钟驱动外，其它模块的时钟都被关闭。该模式下当有 FIQ 或者 IRQ 中断发生时，系统迁移到 DOZE 模式，并且 SC_CTRL[modectrl]的值由 SLEEP 自动更新为 DOZE。

说明

从 SLEEP 模式切换到 DOZE 模式，是由中断触发的。因此要求在进入 SLEEP 模式之前保证中断控制器不会对相应的 IRQ 或 FIQ 中断进行屏蔽。当使用 GPIO 的外部中断输入来触发系统切换到 DOZE 时，GPIO 的外部中断输入必须是电平触发，不能使用边沿触发。

- DOZE 模式



在 DOZE 模式下，系统时钟和系统控制器时钟由 46.8kHz 晶振分频时钟驱动。该模式可能发生的状态迁移有：

- 如果 SC_CTRL[modectrl]被设置为 SLOW 模式或者 NORMAL 模式，系统将进入晶振控制状态 SC_XTALCTL，对 24MHz 时钟晶振进行初始化。当晶振稳定后，系统迁移到 SW to XTAL 状态，将系统时钟从 46.8kHz 切换到 24MHz 时钟，进入 SLOW 模式。
- 如果 SC_CTRL[modectrl]被设为 SLEEP 模式，并且 ARM926EJ-S 处于 Wait-for-interrupt 状态，系统进入 SLEEP 模式。

说明

- 系统控制器 SC_XTALCTRL bit[18:3]定义了 24MHz 晶振的稳定时间，当晶振被使能时，超时计数器开始计数。
- 用户可通过设置 ARM1176 系统控制协处理器 CP15，使处理器进入低功耗 (Wait-for-interrupt) 状态。

• SLOW 模式

在 SLOW 模式下，ARM 子系统工作于 24MHz 时钟。该模式下，如果 SC_CTRL[modectrl]被设为 NORMAL 模式，系统将进入 PLL 控制状态 SC_PLLCTL，使能 PLL。当 PLL 稳定后，系统进入 SW to PLL 状态，将系统时钟切换到 PLL 时钟，切换完成后，进入 NORMAL 模式。

说明

系统控制器 SC_PLLCTRL bit[27:3]定义了 PLL 的稳定时间，当使能 PLL 后，超时计数器开始计数，可通过查询 SC_PLLCTRL23 bit[0]判断 PLL 是否已经稳定。

如果 SC_CTRL[modectrl]被设为比 SLOW 更低的模式 (DOZE 或者 SLEEP)，系统迁移到 SW from XTAL 状态，将系统时钟切换到 46.8kHz，切换完成后，进入 DOZE 模式。

• NORMAL 模式

在 NORMAL 模式下，ARM 子系统工作于 PLL 的输出时钟。该模式下，如果 SC_CTRL[modectrl]被设为非 NORMAL 模式，系统迁移到 SW from PLL 状态，将系统时钟切换到 24MHz 时钟，切换完成后，进入 SLOW 模式。

PLL 控制

系统控制器的系统状态机可用于控制片内 PLL 的使能，各种模式下 PLL 的状态如表 3-34 所示。

PLL 频率控制

系统控制器集成了 4 个 PLL 频率控制寄存器，用于定义 PLL 的各种控制系数。具体请参见 SC_PERCTRL0、SC_PERCTRL1、SC_PERCTRL2、SC_PERCTRL3、SC_PERCTRL4、SC_PERCTRL5、SC_PERCTRL6 和 SC_PERCTRL7。

中断响应模式

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

- 中断响应模式是否使能。



- 中断发生后系统的工作模式。
- 触发中断响应模式的中断类型是 FIQ 还是 IRQ。
- 中断模式状态查询和清除机制。

说明

中断响应模式只支持系统运行频率从低速切换到高速，例如从 DOZE 模式切换到 NORMAL；不支持高速模式向低速模式点切换，例如从 NORMAL 模式切换到 SLOW 模式。

软复位

系统控制器支持对芯片全局以及局部模块进行软复位：

- 当配置全局软复位寄存器 SC_SYSSTAT 后，系统控制器将给片内复位模块发起请求，芯片将被复位。
- 当配置系统控制器的相应模块软复位控制位后，系统控制器将控制片内复位模块对这个模块进行复位。

对模块的一次软复位操作包括配置软复位和撤消软复位。例如要对 UART0 模块进行软复位，则需要首先对 SC_PERCTRL8[uart0_srst]写 1 进行软复位，再写 0 撤消软复位。

说明

- 系统软复位通过对寄存器 SC_SYSSTAT 写任意值来实现。系统软复位的撤消自动完成，不需软件干预。
- 各模块的软复位控制请参见 SC_PERCTRL8 和 SC_PERCTRL10。

系统地址重映射控制

系统控制器提供地址重映射控制信号，支持地址译码单元对系统存储地址空间进行重新映射和分配。根据 boot 方式的不同，上电复位后，芯片将 0 地址映射到不同的物理空间。Hi3520 支持以下 3 种 boot 方式：

- 从 NOR Flash 启动
- 从 NAND Flash 启动
- 从 DDR 启动

通过对管脚 EBIADR23 和 EBIADR24 进行设置，选择启动方式，具体情况如表 3-17 所示。

表3-35 启动方式选择

EBIADR23	EBIADR24	Boot 方式
下拉	下拉	从 NOR Flash 启动
下拉	上拉	从 DDR 启动
上拉	下拉	从 NAND Flash 启动
上拉	上拉	从 DDR 启动



说明

地址重映射清除以后，0 地址到 SMI、NAND Flash 或 MDDRC 的存储器空间的映射被清除。此时，建议将 0x0000_0000 ~ 0x0000_0FFF 地址通过配置 CPU 分配给片内的 2KB 的 ITCM。

WatchDog 和 Timer 时钟使能控制

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

- 支持对输入的计数时钟进行采样，生成时钟使能信号，输出给 WatchDog 和 Timer 模块。
- 可通过软件强制将 WatchDog 和 Timer 计数时钟使能拉高，使其内部计数器按总线时钟计数，当系统处于 Debug 模式时，禁止 WatchDog 计数功能。
- 支持对 Timer 的计数时钟源进行选择。

总线仲裁控制

在系统中，128bits interconnect 存在仲裁机制。可以由系统控制器配置该 interconnect 上各个端口的通过 interconnect 的优先级及 timeout 等。

对关键寄存器的写保护

为防止软件对系统控制器的误操作严重影响整个系统，系统控制器提供了对系统控制器的一些关键配置寄存器的写保护功能。包括：

- 模式切换的控制寄存器 SC_CTRL
- 系统全局软复位控制寄存器 SC_SYSSTAT
- 片内 ARMPPLL 的控制寄存器 SC_PERCTRL0、SC_PERCTRL1

对上述关键寄存器进行写操作之前，必须配置寄存器 SC_PERLOCK 取消写保护。操作完成之后配置寄存器 SC_PERLOCK 打开写保护，确保上述关键寄存器不会被软件随意改写。

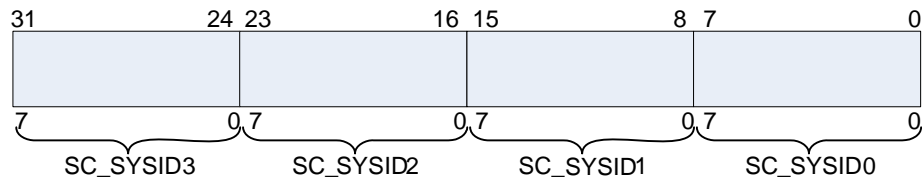
说明

系统默认为复位后不对关键寄存器进行写保护。为启用此功能，建议在系统启动时利用该寄存器 SC_PERLOCK 进行写保护处理。

芯片的标识寄存器

系统控制器提供了芯片标识（ID）寄存器 SC_SYSID。标识寄存器是一个概念上的 32bit 的标识只读寄存器，实际上由 4 个 8bit 标识寄存器组成：SC_SYSID3、SC_SYSID2、SC_SYSID1、SC_SYSID0。读出这 4 个寄存器的值，通过组合得到芯片的 32bit 标识寄存器的值 0x3520_0100，组合的方法如图 3-27 所示。

图3-27 芯片 ID 寄存器位分配图



3.10.4 寄存器概览

系统控制寄存器寄存器概览如表 3-36 所示。

表3-36 系统控制寄存器寄存器概览（基址是 0x2005_0000）

偏移地址	名称	描述	页码
0x0000	SC_CTRL	系统控制寄存器	3-122
0x0004	SC_SYSSTAT	系统状态寄存器	3-125
0x0008	SC_IMCTRL	中断模式控制寄存器	3-126
0x000C	SC_IMSTAT	中断模式状态寄存器	3-127
0x0010	SC_XTALCTRL	晶振控制寄存器	3-128
0x0014	SC_PLLCTRL	PLL 控制寄存器	3-128
0x001C	SC_PERCTRL0	ARM PLL 频率控制寄存器 1	3-129
0x0020	SC_PERCTRL1	ARM PLL 频率控制寄存器 2	3-130
0x0024	SC_PEREN	外设时钟使能寄存器	3-131
0x0028	SC_PERDIS	外设时钟禁止寄存器	3-134
0x002C	SC_PERCLKEN	外设时钟状态寄存器	3-138
0x0034	SC_PERCTRL2	VIDEO PLL0 频率控制寄存器 1	3-141
0x0038	SC_PERCTRL3	VIDEO PLL0 频率控制寄存器 2	3-141
0x003C	SC_PERCTRL4	VIDEO PLL1 频率控制寄存器 1	3-142
0x0040	SC_PERCTRL5	VIDEO PLL1 频率控制寄存器 2	3-143
0x0044	SC_PERLOCK	关键系统控制寄存器的锁定寄存器	3-144
0x0048	SC_PERCTRL6	GMAC PLL 频率控制寄存器 1	3-145
0x004C	SC_PERCTRL7	GMAC PLL 频率控制寄存器 2	3-146
0x0050	SC_PERCTRL8	软复位控制寄存器 1	3-147



偏移地址	名称	描述	页码
0x0054	SC_PERCTRL9	时钟模式控制寄存器 1	3-150
0x0058	SC_PERCTRL10	软复位控制寄存器 2	3-152
0x005C	SC_PERCTRL11	外设工作模式寄存器 1	3-153
0x0060	SC_PERCTRL12	外设工作模式寄存器 2	3-155
0x0064	SC_PERCTRL13	时钟模式控制寄存器 2	3-158
0x0068	SC_PERCTRL14	时钟模式控制寄存器 3	3-159
0x0070	SC_PERCTRL16	时钟模式控制寄存器 4	3-160
0x0074	SC_PERCTRL17	ARM926EJ-S 映射地址寄存器	3-162
0x0078	SC_PERCTRL18	总线仲裁控制寄存器 1	3-162
0x007C	SC_PERCTRL19	总线仲裁控制寄存器 2	3-163
0x0088	SC_PERCTRL22	总线仲裁控制寄存器 5	3-163
0x008C	SC_PERCTRL23	芯片工作模式状态和 PLL 状态寄存器	3-164
0xEE0	SC_SYSID0	芯片 ID 寄存器 0	3-166
0xEE4	SC_SYSID1	芯片 ID 寄存器 1	3-166
0xEE8	SC_SYSID2	芯片 ID 寄存器 2	3-167
0xEEC	SC_SYSID3	芯片 ID 寄存器 3	3-167

3.10.5 寄存器描述

SC_CTRL

SC_CTRL 为系统控制寄存器。用于指定需要系统完成的操作。



注意

该寄存器可被寄存器 SC_PERLOCK 写保护，只有禁用写保护时，对这个寄存器的写操作才有效。



Offset Address		Register Name		Total Reset Value																														
0x0000		SC_CTRL		0x0000_0212																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	timeren7ov	timeren7sel	timeren6ov	timeren6sel	timeren5ov	timeren5sel	timeren4ov	timeren4sel	wdogenov	timeren3ov	timeren3sel	timeren2ov	timeren2sel	timeren1ov	timeren1sel	timeren0ov	timeren0sel	reserved						remapstat	remapclear	reserved	modestatus				modectrl			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0
Bits	Access		Name		Description																													
[31]	RW		timeren7ov		Timer7 计数时钟选择。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 timeren7sel 指定; 1: 由总线时钟进行计数。																													
[30]	RW		timeren7sel		Timer7 计数时钟频率选择 (必须配置为 0)。 0: 使用 3MHz 时钟进行计数; 1: 保留。																													
[29]	RW		timeren6ov		Timer6 计数时钟选择。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 timeren6sel 指定; 1: 由总线时钟进行计数。																													
[28]	RW		timeren6sel		Timer6 计数时钟频率选择 (必须配置为 0)。 0: 使用 3MHz 时钟进行计数; 1: 保留。																													
[27]	RW		timeren5ov		Timer5 计数时钟选择。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 timeren5sel 指定; 1: 由总线时钟进行计数。																													
[26]	RW		timeren5sel		Timer5 计数时钟频率选择 (必须配置为 0)。 0: 使用 3MHz 时钟进行计数; 1: 保留。																													
[25]	RW		timeren4ov		Timer4 计数时钟选择。 0: 使能信号通过采用参考时钟得到, 参考时钟的选择由 timeren4sel 指定; 1: 由总线时钟进行计数。																													



[24]	RW	timeren4sel	Timer4 计数时钟频率选择（必须配置为 0）。 0: 使用 3MHz 时钟进行计数； 1: 保留。
[23]	RW	wdogenov	WDG 计数时钟选择。 0: WDG 使用 3MHz 时钟进行计数； 1: WDG 使用总线时钟进行计数。
[22]	RW	timeren3ov	Timer3 计数时钟选择。 0: 使能信号通过采用参考时钟得到，参考时钟的选择由 timeren3sel 指定； 1: 由总线时钟进行计数。
[21]	RW	timeren3sel	Timer3 计数时钟频率选择（必须配置为 0）。 0: 使用 3MHz 时钟进行计数； 1: 保留。
[20]	RW	timeren2ov	Timer2 计数时钟选择。 0: 使能信号通过采用参考时钟得到，参考时钟的选择由 timeren2sel 指定； 1: 由总线时钟进行计数。
[19]	RW	timeren2sel	Timer2 计数时钟频率选择（必须配置为 0）。 0: 使用 3MHz 时钟进行计数； 1: 保留。
[18]	RW	timeren1ov	Timer1 计数时钟选择。 0: 使能信号通过采用参考时钟得到，参考时钟的选择由 timeren1sel 指定； 1: 由总线时钟进行计数。
[17]	RW	timeren1sel	Timer1 计数时钟频率选择（必须配置为 0）。 0: 使用 3MHz 时钟进行计数； 1: 保留。
[16]	RW	timeren0ov	Timer0 计数时钟选择。 0: 使能信号通过采用参考时钟得到，参考时钟的选择由 timeren0sel 指定； 1: 由总线时钟进行计数。
[15]	RW	timeren0sel	Timer0 计数时钟频率选择（必须配置为 0）。 0: 使用 3MHz 时钟进行计数； 1: 保留。
[14:10]	-	reserved	保留。读时返回 0，写时无影响。



[9]	RO	remapstat	地址重映射的状态。 0: 未进行地址重映射; 1: 进行地址重映射。当加载模式为自加载时, EBICS0N 被 Remap 到地址 0; 当加载模式为从加载时, DDRCSN 被 Remap 到地址 0。
[8]	RW	remapclear	地址重映射清除选择。 0: 保持 Remap 状态; 1: 清除 Remap。 Clear Remap 前后地址映射关系“3.3 3.3 处理器及存储器地址空间映射”。
[7]	-	reserved	保留。读时返回 0, 写时无影响。
[6:3]	RW	modestatus	模式状态位。 这些位返回系统当前的操作模式。这 4 位定义如下: 0x0: SLEEP; 0x1: DOZE; 0x2: SLOW; 0x3: XTAL CTL; 0x4: NORMAL; 0x6: PLL CTL; 0x9: SW from XTAL; 0xA: SW from PLL; 0xB: SW to XTAL; 0xE: SW to PLL; 其他: 保留。
[2:0]	RW	modectrl	模式控制位。这些位定义了要求系统控制器进入的操作模式。这 3 位定义如下: 000: SLEEP; 001: DOZE; 01X: SLOW; 1XX: NORMAL。

SC_SYSSTAT

SC_SYSSTAT 为系统状态寄存器。向该寄存器写入任何值都会使系统控制器向复位模块发出系统软复位请求, 复位模块进行系统软复位。

**注意**

该寄存器可被寄存器 SC_PERLOCK 写保护，只有禁用写保护时，对这个寄存器的写操作才有效。

Offset Address		Register Name		Total Reset Value				
0x0004		SC_SYSSTAT		0x0000_0002				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	softresreq							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0
Bits	Access	Name	Description					
[31:0]	WO	softresreq	对该寄存器的任意写操作都会导致系统软复位。					

SC_IMCTRL

SC_IMCTRL 为中断模式控制寄存器。用于控制中断发生时的系统模式。

Offset Address		Register Name		Total Reset Value						
0x0008		SC_IMCTRL		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved						inmdtype	reserved	itmctrl	itmden
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:8]	-	reserved	保留。读时返回 0，写时无影响。							
[7]	RW	inmdtype	触发系统进入中断模式的中断类型。 0: 仅有 FIQ 中断能使系统进入中断模式； 1: FIQ 中断和 IRQ 中断都能使系统进入中断模式。							
[6:4]	-	reserved	保留。							



[3:1]	RW	itmdctrl	<p>中断模式下系统最低的工作模式，该寄存器的值和 SC_CTRL[modectrl]的值相或后作为中断发生后系统所处的工作模式。定义如下：</p> <p>000: SLEEP; 001: DOZE; 01X: SLOW; 1XX: NORMAL。</p>
[0]	RW	itmden	<p>中断模式使能。</p> <p>0: 禁止; 1: 使能（当中断发生时进入中断模式）。</p>

SC_IMSTAT

SC_IMSTAT 为中断模式状态寄存器。用于监视系统是否处于中断模式，也可以通过配置该寄存器强制系统进入中断模式。



注意

当中断服务程序结束时必须手动清除中断模式。

	Offset Address				Register Name								Total Reset Value																			
	0x000C				SC_IMSTAT								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											itmdstat				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description																												
	[31:1]	-	reserved	保留。读时返回 0，写时无影响。																												



[0]	RW	itmdstat	<p>中断模式状态。该位可用于软件控制直接进入中断模式。</p> <p>读该寄存器时：</p> <p>0：当前未处于中断模式；</p> <p>1：当前处于中断模式。</p> <p>写该寄存器时：</p> <p>0：软件不控制进入中断模式；</p> <p>1：软件控制进入中断模式。</p>
-----	----	----------	--

SC_XTALCTRL

SC_XTALCTRL 为晶振控制寄存器。用于控制初始化时钟模块的稳定等待时间，也就是从 XTAL CTL 中间态跳转到 SW to XTAL 中间态的等待时间。

	Offset Address				Register Name								Total Reset Value																			
	0x0010				SC_XTALCTRL								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								xtaltime								reserved		reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:19]	-		reserved		保留，读时返回 0，写时无影响。																											
[18:3]	RW		xtaltime		晶振切换等待时间。 该字段的值用于指定系统模式切换时从 XTAL CTL 状态跳转到 SW to XTAL 状态的等待时间。等待周期数可通过计算得到（T46.8K 为 46.8KHz 低频时钟周期）： $(65536 - xtaltime) \times T46.8K$ 。																											
[2]	-		reserved		保留。读时返回 0，写时无影响。																											
[1:0]	-		reserved		保留。读时返回写入值。																											

SC_PLLCTRL

SC_PLLCTRL 为 PLL 控制寄存器。用于控制片内 ARM 锁相环（ARMPLL）的使能控制，由软件控制使能，或由系统模式切换来控制使能。此外，该寄存器还用于设置 ARMPLL 锁相环稳定等待时间。



注意

该寄存器可被寄存器 SC_PERLOCK 写保护，只有禁用写保护时，对这个寄存器的写操作才有效。

当处于“由系统模式切换来控制使能 ARMPLL 锁相环”时，在系统处于非 NORMAL 模式下，ARMPLL 被自动关闭。

ARMPLL 的时钟频率由寄存器 SC_PERCTRL0/SC_PERCTRL1 中相应位控制。



说明

系统要求 PLL 在变更频率配置时，至少需要等待 0.5ms 才能输出稳定的时钟。因此该寄存器的 plltime 的配置必须满足此要求。

	Offset Address				Register Name				Total Reset Value																							
	0x0014				SC_PLLCTRL				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				plltime												reserved	reserved	plllover													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:28]	-	reserved	保留。读时返回 0，写时无影响。																													
[27:3]	RW	plltime	ARMPLL 锁相环稳定等待时间。 这段时间用于等待 PLL 启动到 PLL 输出达到稳定的状态。也就是指定从系统模式切换时从 PLL CTL 状态跳转到 SW to PLL 状态的等待时间。超时时间值由下式计算得到（TXIN 为芯片外接晶振的时钟周期）： $(33554432 - plltime) \times TXIN$																													
[2]	-	reserved	保留。读时返回 0，写时无影响。																													
[1]	-	reserved	保留。																													
[0]	RW	plllover	此位必须配置为 0，表示由系统模式切换来控制使能 ARMPLL 锁相环。																													

SC_PERCTRL0

SC_PERCTRL0 为 ARM PLL 频率控制寄存器 1。

Offset Address		Register Name		Total Reset Value					
0x001C		SC_PERCTRL0		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	apll_dsmpd apll_bypass	apll_postdiv2	apll_postdiv1	apll_frac					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31]	RW	apll_dsmpd	APLL 分频模式控制。 0: 小数分频模式; 1: 整数分频模式。						
[30]	RW	apll_bypass	APLL 时钟分频旁路 (bypass) 控制。 0: 非旁路; 1: 旁路。						
[29:27]	RW	apll_postdiv2	APLL 第二级输出分频系数。						
[26:24]	RW	apll_postdiv1	APLL 第一级输出分频系数。						
[23:0]	RW	apll_frac	APLL 小数分频系数。						

SC_PERCTRL1

SC_PERCTRL1 为 ARM PLL 频率控制寄存器 2。

Offset Address		Register Name		Total Reset Value					
0x0020		SC_PERCTRL1		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			apll_reset apll_pd apll_foutvcopd	apll_postdivpd apll_fout4phasepd	apll_refdiv		apll_fbdiv	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:23]	-	reserved	保留。						



[22]	RW	apll_reset	APLL 复位控制。 0: 复位; 1: 不复位。
[21]	RW	apll_pd	APLL Power Down 控制。 0: 关闭; 1: 打开。
[20]	RW	apll_foutvcopd	APLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。
[19]	RW	apll_postdivpd	APLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。
[18]	RW	apll_fout4phasepd	APLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。
[17:12]	RW	apll_refdiv	APLL 参考时钟分频系数。
[11:0]	RW	apll_fbdiv	APLL 整数倍频系数。

SC_PEREN

SC_PEREN 为外设时钟使能寄存器。只写寄存器，用于在外部时钟产生逻辑中产生外设时钟的使能信号。向该寄存器的某位写入 1 可打开对应模块的时钟使能，写入 0 无影响。

	Offset Address 0x0024																Register Name SC_PEREN								Total Reset Value 0xFFFF_FFFF							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clken	vdac0clken	sspelken	arm9clken	nandclken	voadclken	vohdclken	vosdclken	vobuselken	usbelken	ethclken	vi3clken	vi2clken	vi1clken	vi0clken	vibuselken	tdeclken	mmclken	pciclken	ipemclken	irelken	sio2clken	sio1clken	sio0clken	uart3clken	uart2clken	uart1clken	uart0clken	smiclken	ciphercclken	reserved	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bits	[31]																															
Access	WO																															
Name	vdac1clken																															
Description	VIDEO 的 DAC1 时钟使能。 0: 无影响; 1: 使能。																															



[30]	WO	vdac0clken	VIDEO 的 DAC0 时钟使能。 0: 无影响; 1: 使能。
[29]	WO	sspclken	SSP 时钟使能。 0: 无影响; 1: 使能。
[28]	WO	arm9clken	ARM926 时钟使能。 0: 无影响; 1: 使能。
[27]	WO	nandcclken	NANDC 时钟使能。 0: 无影响; 1: 使能。
[26]	WO	voadclken	VO 的 AD 通道时钟使能。 0: 无影响; 1: 使能。
[25]	WO	vohdclken	VO 的 HD 通道时钟使能。 0: 无影响; 1: 使能。
[24]	WO	vosdclken	VO 的 SD 通道时钟使能。 0: 无影响; 1: 使能。
[23]	WO	vobusclken	VO 总线侧时钟使能。 0: 无影响; 1: 使能。
[22]	WO	usbclken	USB 时钟使能。 0: 无影响; 1: 使能。
[21]	WO	ethclken	GMAC 时钟使能。 0: 无影响; 1: 使能。
[20]	WO	vi3clken	VI3 端口时钟使能。 0: 无影响; 1: 使能。



[19]	WO	vi2clken	VI2 端口时钟使能。 0: 无影响; 1: 使能。
[18]	WO	vi1clken	VI1 端口时钟使能。 0: 无影响; 1: 使能。
[17]	WO	vi0clken	VI0 端口时钟使能。 0: 无影响; 1: 使能。
[16]	WO	vibusclken	VI 总线侧时钟使能。 0: 无影响; 1: 使能。
[15]	WO	tdeclken	TDE 时钟使能。 0: 无影响; 1: 使能。
[14]	WO	mmcclken	MMC 时钟使能。 0: 无影响; 1: 使能。
[13]	WO	pciclken	PCI 时钟使能。 0: 无影响; 1: 使能。
[12]	WO	ipcmclken	IPCM 时钟使能。 0: 无影响; 1: 使能。
[11]	WO	ircclken	IR 时钟使能。 0: 无影响; 1: 使能。
[10]	WO	sio2clken	SIO2 时钟使能。 0: 无影响; 1: 使能。
[9]	WO	sio1clken	SIO1 时钟使能。 0: 无影响; 1: 使能。



[8]	WO	sio0clken	SIO0 时钟使能。 0: 无影响; 1: 使能。
[7]	WO	uart3clken	UART3 时钟使能。 0: 无影响; 1: 使能。
[6]	WO	uart2clken	UART2 时钟使能。 0: 无影响; 1: 使能。
[5]	WO	uart1clken	UART1 时钟使能。 0: 无影响; 1: 使能。
[4]	WO	uart0clken	UART0 时钟使能。 0: 无影响; 1: 使能。
[3]	WO	smiclken	SMI 时钟使能。 0: 无影响; 1: 使能。
[2]	WO	cipherclken	CIPHER 时钟使能。 0: 无影响; 1: 使能。
[1:0]	-	reserved	保留。

SC_PERDIS

SC_PERDIS 为外设时钟禁止寄存器。只写寄存器，用于在外部时钟产生逻辑中将外设的时钟使能信号置为无效。向该寄存器的某位写入 1 可关断对应模块的时钟，写入 0 无影响。

寄存器 SC_PEREN 和寄存器 SC_PERDIS 必须配合使用才能实现对某个模块的时钟的打开或者关闭。如 SMI 模块的时钟当前状态为打开，那么对寄存器 SC_PERDIS 写入 0x0000_0004，则关闭 SMI 的时钟；此时如果需要再次打开 SMI 的时钟，就需要对寄存器 SC_PEREN 写入 0x0000_0001。判断对某个模块的时钟打开或关断的操作是否正常，可以通过读寄存器 SC_PERCLKEN 的相关位。如在打开 SMI 的时钟之后，可通过查询 SC_PERCLKEN[3]是否为 1 判断操作是否成功。



Offset Address		Register Name																Total Reset Value														
0x0028		SC_PERDIS																0x0000_0000														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clkdis	vdac0clkdis	sspclkdis	arm9clkdis	nandcclkdis	voadclkdis	vohdclkdis	vosdclkdis	vobuscclkdis	usbclkdis	ethclkdis	vi3clkdis	vi2clkdis	vi1clkdis	vi0clkdis	vibusclkdis	tdecclkdis	mmclkdis	pciclkdis	ipemclkdis	ireclkdis	sio2clkdis	sio1clkdis	sio0clkdis	uart3clkdis	uart2clkdis	uart1clkdis	uart0clkdis	smiclkdis	ciphclkdis	reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31]	WO	vdac1clkdis	VIDEO 的 DAC1 时钟关闭。 0: 无影响; 1: 关闭。																													
[30]	WO	vdac0clkdis	VIDEO 的 DAC0 时钟关闭。 0: 无影响; 1: 关闭。																													
[29]	WO	sspclkdis	SSP 时钟关闭。 0: 无影响; 1: 关闭。																													
[28]	WO	arm9clkdis	ARM926 时钟关闭。 0: 无影响; 1: 关闭。																													
[27]	WO	nandcclkdis	NANDC 时钟关闭。 0: 无影响; 1: 关闭。																													
[26]	WO	voadclkdis	VO 的 AD 通道时钟关闭。 0: 无影响; 1: 关闭。																													
[25]	WO	vohdclkdis	VO 的 HD 通道时钟关闭。 0: 无影响; 1: 关闭。																													
[24]	WO	vosdclkdis	VO 的 SD 通道时钟关闭。 0: 无影响; 1: 关闭。																													



[23]	WO	vobusclkdis	VO 总线侧时钟关闭。 0: 无影响; 1: 关闭。
[22]	WO	usbclkdis	USB 时钟关闭。 0: 无影响; 1: 关闭。
[21]	WO	ethclkdis	GMAC 时钟关闭。 0: 无影响; 1: 关闭。
[20]	WO	vi3clkdis	VI3 端口时钟关闭。 0: 无影响; 1: 关闭。
[19]	WO	vi2clkdis	VI2 端口时钟关闭。 0: 无影响; 1: 关闭。
[18]	WO	vi1clkdis	VI1 端口时钟关闭。 0: 无影响; 1: 关闭。
[17]	WO	vi0clkdis	VI0 端口时钟关闭。 0: 无影响; 1: 关闭。
[16]	WO	vibusclkdis	VI 总线侧时钟关闭。 0: 无影响; 1: 关闭。
[15]	WO	tdeclkdis	TDE 时钟关闭。 0: 无影响; 1: 关闭。
[14]	WO	mmcclkdis	MMC 时钟关闭。 0: 无影响; 1: 关闭。
[13]	WO	pciclkdis	PCI 时钟关闭。 0: 无影响; 1: 关闭。



[12]	WO	ipcmclkdis	IPCM 时钟关闭。 0: 无影响; 1: 关闭。
[11]	WO	ircclkdis	IR 时钟关闭。 0: 无影响; 1: 关闭。
[10]	WO	sio2clkdis	SIO2 时钟关闭。 0: 无影响; 1: 关闭。
[9]	WO	sio1clkdis	SIO1 时钟关闭。 0: 无影响; 1: 关闭。
[8]	WO	sio0clkdis	SIO0 时钟关闭。 0: 无影响; 1: 关闭。
[7]	WO	uart3clkdis	UART3 时钟关闭。 0: 无影响; 1: 关闭。
[6]	WO	uart2clkdis	UART2 时钟关闭。 0: 无影响; 1: 关闭。
[5]	WO	uart1clkdis	UART1 时钟关闭。 0: 无影响; 1: 关闭。
[4]	WO	uart0clkdis	UART0 时钟关闭。 0: 无影响; 1: 关闭。
[3]	WO	smiclkdis	SMI 时钟关闭。 0: 无影响; 1: 关闭。
[2]	WO	cipherclkdis	CIPHER 时钟关闭。 0: 无影响; 1: 关闭。
[1:0]	-	reserved	保留。



SC_PERCLKEN

SC_PERCLKEN 为外设时钟状态寄存器。只读寄存器，用于读出系统控制器内对各模块时钟使能的状态以检验写寄存器 SC_PEREN 和 SC_PERDIS 的操作是否生效。某位读为 0 表示相应的模块时钟关闭；某位读为 1 则表示相应的模块时钟打开。

	Offset Address 0x002C								Register Name SC_PERCLKEN								Total Reset Value 0xFFFF_FFFF															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clkstat	vdac0clkstat	sspclkstat	arm9clkstat	nandclkstat	voadclkstat	vohdclkstat	vosdclkstat	vobuscclkstat	usbclkstat	ethclkstat	vi3clkstat	vi2clkstat	vi1clkstat	vi0clkstat	vibusclkstat	tdeclkstat	mmclkstat	pciclkstat	ipcmclkstat	ireclkstat	sio2clkstat	sio1clkstat	sio0clkstat	uart3clkstat	uart2clkstat	uart1clkstat	uart0clkstat	smicclkstat	ciphercclkstat	reserved	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	Bits	Access	Name	Description																												
	[31]	RO	vdac1clkstat	VIDEO 的 DAC1 时钟状态。 0: 关闭; 1: 打开。																												
	[30]	RO	vdac0clkstat	VIDEO 的 DAC0 时钟状态。 0: 关闭; 1: 打开。																												
	[29]	RO	sspclkstat	SSP 时钟状态。 0: 关闭; 1: 打开。																												
	[28]	RO	arm9clkstat	ARM926 时钟状态。 0: 关闭; 1: 打开。																												
	[27]	RO	nandclkstat	NANDC 时钟状态。 0: 关闭; 1: 打开。																												
	[26]	RO	voadclkstat	VO 的 AD 通道时钟状态。 0: 关闭; 1: 打开。																												



[25]	RO	vohdclkstat	VO 的 HD 通道时钟状态。 0: 关闭; 1: 打开。
[24]	RO	vosdclkstat	VO 的 SD 通道时钟状态。 0: 关闭; 1: 打开。
[23]	RO	vobusclkstat	VO 总线侧时钟状态。 0: 关闭; 1: 打开。
[22]	RO	usbclkstat	USB 时钟状态。 0: 关闭; 1: 打开。
[21]	RO	ethclkstat	GMAC 时钟状态。 0: 关闭; 1: 打开。
[20]	RO	vi3clkstat	VI3 端口时钟状态。 0: 关闭; 1: 打开。
[19]	RO	vi2clkstat	VI2 端口时钟状态。 0: 关闭; 1: 打开。
[18]	RO	vi1clkstat	VI1 端口时钟状态。 0: 关闭; 1: 打开。
[17]	RO	vi0clkstat	VI0 端口时钟状态。 0: 关闭; 1: 打开。
[16]	RO	vibusclkstat	VI 总线侧时钟状态。 0: 关闭; 1: 打开。
[15]	RO	tdeclkstat	TDE 时钟状态。 0: 关闭; 1: 打开。



[14]	RO	mmcclkstat	MMC 时钟状态。 0: 关闭; 1: 打开。
[13]	RO	pciclkstat	PCI 时钟状态。 0: 关闭; 1: 打开。
[12]	RO	ipcmclkstat	IPCM 时钟状态。 0: 关闭; 1: 打开。
[11]	RO	ircclkstat	IR 时钟状态。 0: 关闭; 1: 打开。
[10]	RO	sio2clkstat	SIO2 时钟状态。 0: 关闭; 1: 打开。
[9]	RO	sio1clkstat	SIO1 时钟状态。 0: 关闭; 1: 打开。
[8]	RO	sio0clkstat	SIO0 时钟状态。 0: 关闭; 1: 打开。
[7]	RO	uart3clkstat	UART3 时钟状态。 0: 关闭; 1: 打开。
[6]	RO	uart2clkstat	UART2 时钟状态。 0: 关闭; 1: 打开。
[5]	RO	uart1clkstat	UART1 时钟状态。 0: 关闭; 1: 打开。
[4]	RO	uart0clkstat	UART0 时钟状态。 0: 关闭; 1: 打开。



[3]	RO	smickstat	SMI 时钟状态。 0: 关闭; 1: 打开。
[2]	RO	cipherclkstat	CIPHER 时钟状态。 0: 关闭; 1: 打开。
[1:0]	-	reserved	保留。

SC_PERCTRL2

SC_PERCTRL2 为 VIDEO PLL0 频率控制寄存器 1。

Offset Address		Register Name		Total Reset Value					
0x0034		SC_PERCTRL2		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	vppll0_dsmpd vppll0_bypass	vppll0_postdiv2	vppll0_postdiv1	vppll0_frac					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31]	RW	vppll0_dsmpd	VPLL0 分频模式控制。 0: 整数分频模式; 1: 小数分频模式。						
[30]	RW	vppll0_bypass	VPLL0 时钟分频旁路 (bypass) 控制。 0: 非旁路 (no bypass) ; 1: 旁路 (bypass) 。						
[29:27]	RW	vppll0_postdiv2	VPLL0 第二级输出分频系数。						
[26:24]	RW	vppll0_postdiv1	VPLL0 第一级输出分频系数。						
[23:0]	RW	vppll0_frac	VPLL0 小数分频系数。						

SC_PERCTRL3

SC_PERCTRL3 为 VIDEO PLL0 频率控制寄存器 2。



Offset Address		Register Name		Total Reset Value									
0x0038		SC_PERCTRL3		0x0000_0000									
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0					
Name	reserved				vpll0_reset	vpll0_pd	vpll0_foutvcopd	vpll0_postdivpd	vpll0_fout4phasepd	vpll0_refdiv	vpll0_fbdiv		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0					
Bits	Access	Name	Description										
[31:23]	-	reserved	保留。										
[22]	RW	vpll0_reset	VPLL0 复位控制。 0: 复位; 1: 不复位。										
[21]	RW	vpll0_pd	VPLL0 Power Down 控制。 0: 关闭; 1: 打开。										
[20]	RW	vpll0_foutvcopd	VPLL0 VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。										
[19]	RW	vpll0_postdivpd	VPLL0 POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。										
[18]	RW	vpll0_fout4phasepd	VPLL0 FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。										
[17:12]	RW	vpll0_refdiv	VPLL0 参考时钟分频系数。										
[11:0]	RW	vpll0_fbdiv	VPLL0 整数倍频系数。										

SC_PERCTRL4

SC_PERCTRL4 为 VIDEO PLL1 频率控制寄存器 1。



Offset Address		Register Name		Total Reset Value					
0x003C		SC_PERCTRL4		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	vppll1_dsmpd vppll1_bypass	vppll1_postdiv2	vppll1_postdiv1	vppll1_frac					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31]	RW	vppll1_dsmpd	VPLL1 分频模式控制。 0: 整数分频模式; 1: 小数分频模式。						
[30]	RW	vppll1_bypass	VPLL1 时钟分频旁路 (bypass) 控制。 0: 非旁路 (no bypass) ; 1: 旁路 (bypass) 。						
[29:27]	RW	vppll1_postdiv2	VPLL1 第二级输出分频系数。						
[26:24]	RW	vppll1_postdiv1	VPLL1 第一级输出分频系数。						
[23:0]	RW	vppll1_frac	VPLL1 小数分频系数。						

SC_PERCTRL5

SC_PERCTRL5 为 VIDEO PLL1 频率控制寄存器 2。

Offset Address		Register Name		Total Reset Value					
0x0040		SC_PERCTRL5		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			vppll1_reset vppll1_pd vppll1_foutvcopd vppll1_postdivpd vppll1_fout4phasepd	vppll1_refdiv			vppll1_fbdiv	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:23]	-	reserved	保留。						



[22]	RW	vpll1_reset	VPLL1 复位控制。 0: 复位; 1: 不复位。
[21]	RW	vpll1_pd	VPLL1 Power Down 控制。 0: 关闭; 1: 打开。
[20]	RW	vpll1_foutvcopd	VPLL1 VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。
[19]	RW	vpll1_postdivpd	VPLL1 POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。
[18]	RW	vpll1_fout4phasepd	VPLL1 FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。
[17:12]	RW	vpll1_refdiv	VPLL1 参考时钟分频系数。
[11:0]	RW	vpll1_fbdiv	VPLL1 整数倍频系数。

SC_PERLOCK

SC_PERLOCK 为关键系统控制寄存器的锁定寄存器。



Offset Address		Register Name		Total Reset Value				
0x0044		SC_PERLOCK		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	scper_lockl							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	scper_lockl	<p>关键系统控制寄存器的锁定寄存器，涉及寄存器包括 SC_CTRL、SC_SYSSTAT、SC_PLLCTRL、SC_PERCTRL0、SC_PERCTRL1。</p> <p>向该寄存器写入 0x1ACC_E551，可打开所有寄存器的写权限，写入其他值则关闭写权限。</p> <p>读该寄存器返回加锁的状态而不是写入该寄存器的值。</p> <p>0x0000_0000：允许写访问（未加锁）。</p> <p>0x0000_0001：禁止写访问（已加锁）。</p>					

SC_PERCTRL6

SC_PERCTRL6 为 GMAC PLL 频率控制寄存器 1。

Offset Address		Register Name		Total Reset Value				
0x0048		SC_PERCTRL6		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	epll_dsmpd epll_bypass	epll_postdiv2 epll_postdiv1	epll_frac					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31]	RW	epll_dsmpd	EPLL 分频模式控制。 0：整数分频模式； 1：小数分频模式。					
[30]	RW	epll_bypass	EPLL 时钟分频旁路（bypass）控制。 0：非旁路（no bypass）； 1：旁路（bypass）。					
[29:27]	RW	epll_postdiv2	EPLL 第二级输出分频系数。					
[26:24]	RW	epll_postdiv1	EPLL 第一级输出分频系数。					



[23:0]	RW	epll_frac	EPLL 小数分频系数。
--------	----	-----------	--------------

SC_PERCTRL7

SC_PERCTRL7 为 GMAC PLL 频率控制寄存器 2。

	Offset Address								Register Name								Total Reset Value															
	0x004C								SC_PERCTRL7								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								epll_reset	epll_pd	epll_foutvcopd	epll_postdivpd	epll_fout4phasepd	epll_refdiv				epll_fbdiv														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:23]	-		reserved		保留。																											
[22]	RW		epll_reset		EPLL 复位控制。 0: 复位; 1: 不复位。																											
[21]	RW		epll_pd		EPLL Power Down 控制。 0: 关闭; 1: 打开。																											
[20]	RW		epll_foutvcopd		EPLL VCO 输出 Power Down 控制。 0: 关闭; 1: 打开。																											
[19]	RW		epll_postdivpd		EPLL POSTDIV 输出 Power Down 控制。 0: 关闭; 1: 打开。																											
[18]	RW		epll_fout4phasepd		EPLL FOUT 输出 Power Down 控制。 0: 关闭; 1: 打开。																											
[17:12]	RW		epll_refdiv		EPLL 参考时钟分频系数。																											
[11:0]	RW		epll_fbdiv		EPLL 整数倍频系数。																											



SC_PERCTRL8

SC_PERCTRL8 为软复位控制寄存器 1。

	Offset Address				Register Name								Total Reset Value																			
	0x0050				SC_PERCTRL8								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	usb_srst	usb_hrst	vohd_srst	vosd_srst	voad_srst	vo_srst	vi3_srst	vi2_srst	vi1_srst	vi0_srst	vi_srst	gmac_srst	mdio_srst	eth_srst	sio2_srst	sio1_srst	sio0_srst	mmc_srst	pci_srst	i2c_srst	rtc_srst	ir_srst	uart3_srst	uart2_srst	uart1_srst	uart0_srst	cipher_srst	ssmc_srst	reserved	arm9_srst	arm11_srst	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	usb_srst	USB 模块软复位控制。 0: 软复位; 1: 撤消软复位。
[30]	RW	usb_hrst	USB 总线侧软复位控制。 0: 软复位; 1: 撤消软复位。
[29]	RW	vohd_srst	VO HD 部分软复位控制。 0: 软复位; 1: 撤消软复位。
[28]	RW	vosd_srst	VO SD 部分软复位控制。 0: 软复位; 1: 撤消软复位。
[27]	RW	voad_srst	VO AD 部分软复位控制。 0: 软复位; 1: 撤消软复位。
[26]	RW	vo_srst	VO 总线侧软复位控制。 0: 软复位; 1: 撤消软复位。
[25]	RW	vi3_srst	VI3 接口侧软复位控制。 0: 软复位; 1: 撤消软复位。
[24]	RW	vi2_srst	VI2 接口侧软复位控制。 0: 软复位; 1: 撤消软复位。



[23]	RW	vi1_srst	VI1 接口侧软复位控制。 0: 软复位; 1: 撤消软复位。
[22]	RW	vi0_srst	VI0 接口侧软复位控制。 0: 软复位; 1: 撤消软复位。
[21]	RW	vi_srst	VI 总线侧软复位控制。 0: 软复位; 1: 撤消软复位。
[20]	RW	gmac_srst	GMAC 接口侧软复位控制。 0: 软复位; 1: 撤消软复位。
[19]	RW	mdio_srst	GMAC MDIO 软复位控制。 0: 软复位; 1: 撤消软复位。
[18]	RW	eth_srst	GMAC 总线侧软复位控制。 0: 软复位; 1: 撤消软复位。
[17]	RW	sio2_srst	SIO2 软复位控制。 0: 撤消软复位; 1: 软复位。
[16]	RW	sio1_srst	SIO1 软复位控制。 0: 撤消软复位; 1: 软复位。
[15]	RW	sio0_srst	SIO0 软复位控制。 0: 撤消软复位; 1: 软复位。
[14]	RW	mmc_srst	MMC 软复位控制。 0: 软复位; 1: 撤消软复位。
[13]	RW	pci_srst	PCI 软复位控制。 0: 撤消软复位; 1: 软复位。



[12]	RW	i2c_srst	I ² C 软复位控制。 0: 撤消软复位; 1: 软复位。
[11]	RW	rtc_srst	RTC 软复位控制。 0: 撤消软复位; 1: 软复位。
[10]	RW	ir_srst	IR 软复位控制。 0: 撤消软复位; 1: 软复位。
[9]	RW	uart3_srst	UART3 软复位控制。 0: 撤消软复位; 1: 软复位。
[8]	RW	uart2_srst	UART2 软复位控制。 0: 撤消软复位; 1: 软复位。
[7]	RW	uart1_srst	UART1 软复位控制。 0: 撤消软复位; 1: 软复位。
[6]	RW	uart0_srst	UART0 软复位控制。 0: 撤消软复位; 1: 软复位。
[5]	RW	cipher_srst	CIPHER 软复位控制。 0: 撤消软复位; 1: 软复位。
[4]	RW	ssmc_srst	SSMC 软复位控制。 0: 撤消软复位; 1: 软复位。
[3:2]	-	reserved	保留。
[1]	RW	arm9_srst	ARM926 软复位控制。 0: 软复位; 1: 撤消软复位。
[0]	RW	arm11_srst	ARM1176 软复位控制。 0: 软复位; 1: 撤消软复位。

SC_PERCTRL9

SC_PERCTRL9 为时钟模式控制寄存器 1。

Offset Address		Register Name		Total Reset Value				
0x0054		SC_PERCTRL9		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved ethclk_sel	ethmac_mode aclkout_sel	pciclk_sel mmesap_sel mmclk_sel	volout_sel vo0out_sel vohd_sel	vosd_sel vi3div_sel	vi2div_sel vi1div_sel	vi0div_sel vi3_vi2_sel vi1_vi0_sel	ssmclk_sel arm1freq_sel arm0freq_sel
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。需保持默认值。					
[29]	RW	ethclk_sel	txclk 产生方式。 0: txclk 由内部产生并送给 PHY; 1: txclk 由外部 PHY 提供。					
[28:26]	RW	ethmac_mode	GMAC 时钟模式。 000: MII 10M 模式; 001: MII 100M 模式; 010: GMII 模式; 011: RGMII 10M 模式; 100: RGMII 100M 模式; 101: RGMII 1000M 模式。 其他: 保留。					
[25]	RW	aclkout_sel	ACKOUT 输出时钟。 0: 选择 SIO0 系统时钟; 1: 选择 SIO1 系统时钟。					
[24:23]	RW	pciclk_sel	PCI 模块的工作时钟。 00: 62.5MHz; 01: 31.25MHz; 10: 50MHz; 11: 41.5MHz。					



[22]	RW	mmcsap_sel	MMC 模块采样卡数据的时钟正反相控制。 0: 使用正相时钟; 1: 使用反相时钟。
[21:20]	RW	mmcclk_sel	MMC 模块工作时钟频率控制。 00: 25MHz; 01: 50MHz; 10: 保留; 11: 19.23MHz。
[19]	RW	vo1out_sel	VO1 接口输出时钟正反相控制。 0: VO1 输出反相时钟; 1: VO1 输出正相时钟。
[18]	RW	vo0out_sel	VO0 接口输出时钟正反相控制。 0: VO0 输出反相时钟; 1: VO0 输出正相时钟。
[17:16]	RW	vohd_sel	VO_HD 部分 DATE 时钟与 VO 时钟的分频比控制。 00: DATE : VO = 1:1; 01: DATE : VO = 2:1; 10: DATE : VO = 4:1; 11: 保留。
[15:14]	RW	vosd_sel	VO_SD/AD 部分 DATE 时钟与 VO 时钟的分频比控制。 00: DATE : VO = 1:1; 01: DATE : VO = 2:1; 10: DATE : VO = 4:1; 11: 保留。
[13:12]	RW	vi3div_sel	VI3 分频时钟控制。 00: VI3 端口时钟的 2 分频; 01: VI3 端口时钟的 4 分频; 10: VI3 端口时钟; 11: 保留。
[11:10]	RW	vi2div_sel	VI2 分频时钟控制。 00: VI2 端口时钟的 2 分频; 01: VI2 端口时钟的 4 分频; 10: VI2 端口时钟; 11: 保留。



[9:8]	RW	vi1div_sel	<p>VI1 分频时钟控制。</p> <p>00: VI1 端口时钟的 2 分频;</p> <p>01: VI1 端口时钟的 4 分频;</p> <p>10: VI1 端口时钟;</p> <p>11: 保留。</p>
[7:6]	RW	vi0div_sel	<p>VI0 分频时钟控制。</p> <p>00: VI0 端口时钟的 2 分频;</p> <p>01: VI0 端口时钟的 4 分频;</p> <p>10: VI0 端口时钟;</p> <p>11: 保留。</p>
[5]	RW	vi3_vi2_sel	<p>VI3 接口时钟选择。</p> <p>0: VI3 接口输入的时钟;</p> <p>1: VI2 接口输入的时钟。</p>
[4]	RW	vi1_vi0_sel	<p>VI1 接口时钟选择。</p> <p>0: VI1 接口输入的时钟;</p> <p>1: VI0 接口输入的时钟。</p>
[3]	RW	ssmcclk_sel	<p>SSMC 时钟与总线时钟比例选择。</p> <p>0: SSMC:HCLK=1:1;</p> <p>1: SSMC:HCLK=1:2。</p>
[2]	RW	arm1freq_sel	<p>ARM926 时钟与总线时钟比例选择。</p> <p>0: ARM9:HCLK=2:1;</p> <p>1: ARM9:HCLK=1:1。</p>
[1:0]	RW	arm0freq_sel	<p>ARM1176 时钟与总线时钟比例选择。</p> <p>00: ARM11:HCLK=2:1;</p> <p>01: ARM11:HCLK=3:1;</p> <p>10: ARM11:HCLK=1:1;</p> <p>11: ARM11:HCLK=1:1。</p>

SC_PERCTRL10

SC_PERCTRL10 为软复位控制寄存器 2。



Offset Address		Register Name		Total Reset Value																												
0x0058		SC_PERCTRL10		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														dma_srst	ddrphy_srst	spi_srst	ipcm_srst	nadc_srst	tde_rst	tde_srst											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:7]	-	reserved	保留。																													
[6]	RW	dma_srst	DMA 软复位控制。 0: 撤消软复位; 1: 软复位。																													
[5]	RW	ddrphy_srst	DDRPHY 软复位控制。 0: 撤消软复位; 1: 软复位。																													
[4]	RW	spi_srst	SPI 软复位控制。 0: 撤消软复位; 1: 软复位。																													
[3]	RW	ipcm_srst	IPCM 软复位控制。 0: 撤消软复位; 1: 软复位。																													
[2]	RW	nadc_srst	NANDC 软复位控制。 0: 撤消软复位; 1: 软复位。																													
[1]	RW	tde_rst	TDE 工作时钟软复位控制。 0: 撤消软复位; 1: 软复位。																													
[0]	RW	tde_srst	TDE 总线侧软复位控制。 0: 撤消软复位; 1: 软复位。																													

SC_PERCTRL11

SC_PERCTRL11 为外设工作模式寄存器 1。



		Offset Address 0x005C								Register Name SC_PERCTRL11								Total Reset Value 0x0000_0000															
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		uart1_rtsmode	spi_port	smi_canclewait	ebinandc_time_out								ebismi_time_out								ebi_arb_delay	ebi_mormal_mode	i2c_delay_bypass	pci_sim_fast	pci_type	pci_clkmode	pci_mode						
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																													
[31]	RW	uart1_rtsmode		UART1 RTS 信号模式。 0: 正常模式; 1: 取反。																													
[30]	RW	spi_port		SPI 片选选择。 0: SPI_CS0 有效; 1: SPI_CS1 有效。																													
[29]	RW	smi_canclewait		SMI 模块的外部等待打断信号控制。 0: 正常等待; 1: 强行打断 SMI 接口操作。																													
[28:19]	RW	ebinandc_time_out		NANDC 模块通过 EBI 接口的超时设置。 当在设置的时间周期内未得到响应时, 进入优先调度通道。 当有仲裁争抢时, 建议配置为 0x020。																													
[18:9]	RW	ebismi_time_out		SMI 模块通过 EBI 接口的超时设置。 当在设置的时间周期内未得到响应, 进入优先调度通道。当 有仲裁争抢的时, 建议配置为 0x020。																													
[8:7]	RW	ebi_arb_delay		EBI 在 SMI 和 NANDC 切换间的延时配置。 00: 不延时; 01: 延时 1 个周期; 10: 延时 2 个周期; 11: 延时 3 个周期。																													
[6]	RW	ebi_mormal_mode		EBI 仲裁模式。 0: 正常工作; 1: 停靠模式, 缺省指向 Idle。																													



[5]	RW	i2c_delay_bypass	I2C SDA 相对 SCL 是否增加延时。 0: 无延时; 1: 延时 300ns。
[4]	RW	pci_sim_fast	PCI 仿真模式, 必须配置为 0x0。
[3:2]	RW	pci_type	PCI 模式, 必须配成 00。 00: 标准 PCI; 其他: 保留。
[1]	RW	pci_clkmode	PCI 时钟模式。 0: 外部输入时钟; 1: 内部输出时钟。
[0]	RW	pci_mode	PCI 模式。 0: 从模式; 1: 主模式。

SC_PERCTRL12

SC_PERCTRL12 为外设工作模式寄存器 2。

Offset Address		Register Name		Total Reset Value																												
0x0060		SC_PERCTRL12		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								usb_start_clk	usb_susp_lgcy	ddrck2_dis	ddrck1_dis	ddrck0_dis	ddr_clksel	usb_tune_l	usb_tune0	dac1_powredown	dac0_powredown	you_test_en	sio2_master	sio1_xfs	sio1_xck	sio1_master	sio0_xfs	sio0_xck	sio0_master						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:20]	-		reserved		保留。																											
[19]	RW		usb_start_clk		OHCI 时钟控制信号, 当 OHCI 时钟挂起之后, 需要把该位置 1 而使能 OHCI 模块的 12MHz 和 48MHz 时钟。在 12MHz 和 48MHz 使能之后, 在进入下一次 suspend 之前, 该位要置为 0。 0: OHCI 进入 suspend 状态后, 不使能 12MHz 和 48MHz 时钟; 1: OHCI 进入 suspend 状态后, 使能 12MHz 和 48MHz 时钟。																											



[18]	RW	usb_susp_lgcy	<p>OHCI 时钟的 strap 信号。</p> <p>0: 当 USB 的端口使用 OHCI 控制器时, utmi_suspend_o_n 信号为 0 表示所有的使用 OHCI 的 USB 端口为 suspend 或者 OHCI 控制器处于 global suspend 状态; 当 utmi_suspend_o_n 为 1 时表示使用 OHCI 的 USB 端口没有全部挂起或者说使用 OHCI 没有处于 global suspend 的状态;</p> <p>1: 当 USB 的端口使用 OHCI 控制器时, utmi_suspend_o_n 信号直接反映对应端口的 suspend 状态。</p>
[17]	RW	ddrck2_dis	<p>DDRB 的 CKP0/CKN0 的禁止控制。</p> <p>0: 正常工作;</p> <p>1: 关断时钟。</p>
[16]	RW	ddrck1_dis	<p>DDRA 的 CKP1/CKN1 的禁止控制。</p> <p>0: 正常工作;</p> <p>1: 关断时钟。</p>
[15]	RW	ddrck0_dis	<p>DDRA 的 CKP0/CKN0 的禁止控制。</p> <p>0: 正常工作;</p> <p>1: 关断时钟。</p>
[14]	RW	ddr_clkssel	<p>DDR 时钟选择。</p> <p>0: 正常工作路径;</p> <p>1: 低频工作路径。</p>
[13:12]	RW	usb_tune1	<p>USB 端口 1 的 transmitter 的 high-speed 调节信号, 该静态信号调节 high-speed 下的电流。</p> <p>00: 标准值减少 4.5%;</p> <p>01: 标准设计工作电压;</p> <p>10: 标准值增加 4.5%;</p> <p>11: 标准值增加 9%。</p> <p>注意: 该信号必须在 PHY 的 POR 撤离之前设置, 并且一直保持值不变。</p>
[11:10]	RW	usb_tune0	<p>USB 端口 0 的 transmitter 的 high-speed 调节信号, 该静态信号调节 high-speed 下的电流。</p> <p>00: 标准值减少 4.5%;</p> <p>01: 标准设计工作电压;</p> <p>10: 标准值增加 4.5%;</p> <p>11: 标准值增加 9%。</p> <p>注意: 该信号必须在 PHY 的 POR 撤离之前设置, 并且一直保持值不变。</p>



[9]	RW	dac1_powredown	DAC1 开关控制。 0: 关断; 1: 打开。
[8]	RW	dac0_powredown	DAC0 开关控制。 0: 关断; 1: 打开。
[7]	RW	vou_test_en	图形输出模块的测试使能。 0: 正常工作; 1: 测试模式。
[6]	RW	sio2_master	SIO2 主从模式。 0: 从模式, 从管脚获取时钟和同步; 1: 主模式, 芯片内部产生时钟和同步。
[5]	RW	sio1_xfs	SIO1 发送侧帧同步信号选择。 0: SIO1 发送通道使用 SIO1XFS 管脚或者内部产生的信号; 1: SIO1 发送通道使用 SIO1RFS 管脚或者内部产生的信号。 (SIO1XFS 管脚无效)
[4]	RW	sio1_xck	SIO1 发送侧时钟选择。 0: SIO1 发送通道使用 SIO1XCK 管脚或者内部产生的信号; 1: SIO1 发送通道使用 SIO1RCK 管脚或者内部产生的信号。 (SIO1XCK 管脚无效)
[3]	RW	sio1_master	SIO1 主从模式。 0: 从模式, 从管脚获取时钟和同步; 1: 主模式, 芯片内部产生时钟和同步。
[2]	RW	sio0_xfs	SIO0 发送侧帧同步信号选择。 0: SIO0 发送通道使用 SIO0XFS 管脚或者内部产生的信号; 1: SIO0 发送通道使用 SIO0RFS 管脚或者内部产生的信号。 (SIO0XFS 管脚无效)
[1]	RW	sio0_xck	SIO0 发送侧时钟选择。 0: SIO0 发送通道使用 SIO0XCK 管脚或者内部产生的信号; 1: SIO0 发送通道使用 SIO0RCK 管脚或者内部产生的信号。 (SIO0XCK 管脚无效)
[0]	RW	sio0_master	SIO0 的主从模式。 0: 从模式, 从管脚获取时钟和同步; 1: 主模式, 芯片内部产生时钟和同步。



SC_PERCTRL13

SC_PERCTRL13 为时钟模式控制寄存器 2。

Offset Address		Register Name		Total Reset Value				
0x0064		SC_PERCTRL13		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	sio0_lrclk_sel	sio0_bclk_sel	sio0clk_sel					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	RW	sio0_lrclk_sel	SIO0 采样时钟频率控制，提供 SIO0 位流时钟的 2、4、8、16、32、48、64、128、256 分频选择。 0x0: 2 分频; 0x1: 4 分频; 0x2: 8 分频; 0x3: 16 分频; 0x4: 32 分频; 0x5: 48 分频; 0x6: 64 分频; 0x7: 128 分频; 0x8: 256 分频; 其他: 保留。					
[27:24]	RW	sio0_bclk_sel	SIO0 位流时钟频率控制，提供 SIO0 系统时钟的 1、2、3、4、6、8、12、16、24、48、64、128 分频选择。 0x0: 1 分频; 0x1: 3 分频; 0x2: 2 分频; 0x3: 4 分频; 0x4: 6 分频; 0x5: 8 分频; 0x6: 12 分频; 0x7: 16 分频; 0x8: 24 分频; 0x9: 48 分频; 0x10: 64 分频; 0x11: 128 分频; 其他: 保留。					



[23:0]	RW	sio0clk_sel	SIO0 系统时钟频率控制，提供 EPLL 输出时钟(500MHz)的任意分频选择。 $F_{sio} = (sio0clk_sel \times F_{epll}) / 2^{27}$ 。
--------	----	-------------	---

SC_PERCTRL14

SC_PERCTRL14 为时钟模式控制寄存器 3。

	Offset Address				Register Name				Total Reset Value																											
	0x0068				SC_PERCTRL14				0x0000_0000																											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	sio1_lrlck_sel				sio1_bclk_sel				sio12clk_sel																											
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0											
Bits	Access		Name		Description																															
[31:28]	RW		sio1_lrlck_sel		SIO1 采样时钟频率控制，提供 SIO1 位流时钟的 2、4、8、16、32、48、64、128、256 分频选择。 0x0: 2 分频; 0x1: 4 分频; 0x2: 8 分频; 0x3: 16 分频; 0x4: 32 分频; 0x5: 48 分频; 0x6: 64 分频; 0x7: 128 分频; 0x8: 256 分频; 其他: 保留。																															



[27:24]	RW	sio1_bclk_sel	<p>SIO1 位流时钟频率控制，提供 SIO1 系统时钟的 1、2、3、4、6、8、12、16、24、48、64、128 分频选择。</p> <p>0x0: 1 分频; 0x1: 3 分频; 0x2: 2 分频; 0x3: 4 分频; 0x4: 6 分频; 0x5: 8 分频; 0x6: 12 分频; 0x7: 16 分频; 0x8: 24 分频; 0x9: 48 分频; 0x10: 64 分频; 0x11: 128 分频; 其他: 保留。</p>
[23:0]	RW	sio12clk_sel	<p>SIO1/SIO2 系统时钟频率控制，提供 EPLL 输出时钟(500MHz)的任意分频选择。</p> <p>$F_{sio} = (sio12clk_sel \times F_{epll}) / 2^{27}$。</p>

SC_PERCTRL16

SC_PERCTRL16 为时钟模式控制寄存器 4。

	Offset Address	Register Name	Total Reset Value																																				
	0x0070	SC_PERCTRL16	0x0000_0000																																				
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																						
Name	mddrc1_gate											reserved											sio2_blk_edge			sio1_blk_edge			sio0_blk_edge			sio2_lrcclk_sel				sio2_bclk_sel			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0														
Bits	Access	Name	Description																																				
[31]	RW	mddrc1_gate	MDDRC1 模块的时钟门控控制。 0: 打开; 1: 关闭。																																				
[30:11]	-	reserved	保留。																																				



[10]	RW	sio2_blk_edge	SIO2 位流时钟正反相控制。 0: 正相时钟; 1: 反相时钟。
[9]	RW	sio1_blk_edge	SIO1 位流时钟正反相控制。 0: 正相时钟; 1: 反相时钟。
[8]	RW	sio0_blk_edge	SIO0 位流时钟正反相控制。 0: 正相时钟; 1: 反相时钟。
[7:4]	RW	sio2_lrcclk_sel	SIO2 采样时钟频率控制, 提供 SIO2 位流时钟的 2、4、8、16、32、48、64、128、256 分频选择。 0x0: 2 分频; 0x1: 4 分频; 0x2: 8 分频; 0x3: 16 分频; 0x4: 32 分频; 0x5: 48 分频; 0x6: 64 分频; 0x7: 128 分频; 0x8: 256 分频; 其他: 保留。
[3:0]	RW	sio2_bclk_sel	SIO2 位流时钟频率控制, 提供 SIO2 系统时钟的 1、2、3、4、6、8、12、16、24、48、64、128 分频选择。 0x0: 1 分频; 0x1: 3 分频; 0x2: 2 分频; 0x3: 4 分频; 0x4: 6 分频; 0x5: 8 分频; 0x6: 12 分频; 0x7: 16 分频; 0x8: 24 分频; 0x9: 48 分频; 0x10: 64 分频; 0x11: 128 分频; 其他: 保留。



SC_PERCTRL17

SC_PERCTRL17 为 ARM926 映射地址寄存器。

Offset Address		Register Name		Total Reset Value						
0x0074		SC_PERCTRL17		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved						arm9_bootaddr			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:8]	-	reserved	保留。							
[7:0]	RW	arm9_bootaddr	ARM926 启动的高 8 位地址。例如：该寄存器的值为 0xC0，当 ARM926 访问 0x00xx_xxxx 实际会访问到 0xC0xx_xxxx。							

SC_PERCTRL18

SC_PERCTRL18 为总线仲裁控制寄存器 1。

Offset Address		Register Name		Total Reset Value					
0x0078		SC_PERCTRL18		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	count_en_port1	over_value_port1			count_en_port0	over_value_port0			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31]	RW	count_en_port1	总线上 VI 端口的 timeout 计数使能。 0: 禁止; 1: 使能。						
[30:16]	RW	over_value_port1	总线上 VI 端口的 timeout 计数值。 计数值=over_value_port1×2。						
[15]	RW	count_en_port0	总线上 VO 端口的 timeout 计数使能。 0: 禁止; 1: 使能。						



[14:0]	RW	over_value_port0	总线上 VO 端口的 timeout 计数值。 计数值=over_value_port0×2。
--------	----	------------------	--

SC_PERCTRL19

SC_PERCTRL19 为总线仲裁控制寄存器 2。

Offset Address		Register Name		Total Reset Value					
0x007C		SC_PERCTRL19		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	count_en_port3	over_value_port3			count_en_port2	over_value_port2			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31]	RW	count_en_port3	总线上 TDE1 端口的 timeout 计数使能。 0: 禁止; 1: 使能。						
[30:16]	RW	over_value_port3	总线上 TDE1 端口的 timeout 计数值。 计数值=over_value_port3×2。						
[15]	RW	count_en_port2	总线上 TDE0 端口的 timeout 计数使能。 0: 禁止; 1: 使能。						
[14:0]	RW	over_value_port2	总线上 TDE0 端口的 timeout 计数值。 计数值=over_value_port2×2。						

SC_PERCTRL22

SC_PERCTRL22 为总线仲裁控制寄存器 5。



Offset Address		Register Name		Total Reset Value				
0x0088		SC_PERCTRL22		0x0000_A72E				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved				pri_port3	pri_port2	pri_port1	pri_port0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0	0 1 1 1	0 0 1 0	1 1 1 0
Bits	Access	Name	Description					
[31:12]	-	reserved	保留。					
[11:9]	RW	pri_port3	总线上 TDE1 端口的配置优先级。 0~6 有效, 数值越大优先级越高。各个端口的优先级不可相同。					
[8:6]	RW	pri_port2	总线上 TDE0 端口的配置优先级。 0~6 有效, 数值越大优先级越高。各个端口的优先级不可相同。					
[5:3]	RW	pri_port1	总线上 VI 端口的配置优先级。 0~6 有效, 数值越大优先级越高。各个端口的优先级不可相同。					
[2:0]	RW	pri_port0	总线上 VO 端口的配置优先级。 0~6 有效, 数值越大优先级越高。各个端口的优先级不可相同。					

SC_PERCTRL23

SC_PERCTRL23 为芯片工作模式状态和 PLL 状态寄存器。

Offset Address		Register Name		Total Reset Value										
0x008C		SC_PERCTRL23		0x0000_0000										
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0						
Name	reserved				nf_ecc_type	nf_addr_num	nf_page_size	debug_sel	boot_mode	arm0_pmuirq	epll_lock	vpll_lock	vpll0_lock	apll_lock
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0						
Bits	Access	Name	Description											
[31:15]	-	reserved	保留。											



[14:13]	RO	nf_ecc_type	<p>系统使用 NAND Flash boot 的 ECC 类型。</p> <p>bit[14]和 bit[13]分别对应复用管脚{NFECC0, NFECC1}上电时刻的状态。</p> <p>{NFECC1, NFECC0}的含义如下：</p> <p>00: 禁止；</p> <p>01: 1bit 模式；</p> <p>10: 4bit 模式；</p> <p>11: 8bit 模式。</p>
[12:11]	RO	nf_addr_num	<p>系统使用 NAND Flash boot 的地址数目。</p> <p>bit[12]和 bit[11]分别对应复用管脚{NFNUM0, NFNUM1}上电时刻的状态。</p> <p>{NFNUM1, NFNUM0}的含义如下：</p> <p>00: 3 个地址周期；</p> <p>01: 4 个地址周期；</p> <p>10: 5 个地址周期；</p> <p>11: 6 个地址周期。</p>
[10:9]	RO	nf_page_size	<p>系统使用 NAND Flash boot 的页面设置。</p> <p>bit[10]和 bit[9]分别对应复用管脚{NFPAGE0, NFPAGE1}上电时刻的状态。</p> <p>{NFPAGE1, NFPAGE0}的含义如下：</p> <p>00: 保留；</p> <p>01: 2KB；</p> <p>10: 4KB；</p> <p>11: 保留。</p>
[8:7]	RO	debug_sel	<p>芯片选定的 ARM debug 模式。</p> <p>00: debug ARM1176；</p> <p>10: debug ARM926；</p> <p>01: debug ARM1176+ARM926；</p> <p>11: 保留。</p>
[6:5]	RO	boot_mode	<p>芯片选定的启动模式。</p> <p>00: 从 NOR Flash 启动；</p> <p>10: 从 NAND Flash 启动；</p> <p>X1: 从 DDR 启动。</p>
[4]	RO	arm0_pmuirq	<p>ARM1176 发生 PMU 状态。</p> <p>0: 发生；</p> <p>1: 未发生。</p>



[3]	RO	epll_lock	GMAC PLL 锁定状态。 0: 未锁定; 1: 已经锁定。
[2]	RO	vppll1_lock	Video1 PLL 锁定状态。 0: 未锁定; 1: 已锁定。
[1]	RO	vppll0_lock	Video0 PLL 锁定状态。 0: 未锁定; 1: 已经锁定。
[0]	RO	apll_lock	ARM PLL 锁定状态。 0: 未锁定; 1: 已经锁定。

SC_SYSID0

SC_SYSID0 为芯片 ID 寄存器 0。

	Offset Address			Register Name			Total Reset Value		
	0xEE0			SC_SYSID0			0x00		
Bit	7	6	5	4	3	2	1	0	
Name	sysid0								
Reset	0	0	0	0	0	0	0	0	
Bits	Access	Name	Description						
[7:0]	RO	sysid0	读该寄存器返回 0x00。						

SC_SYSID1

SC_SYSID1 为芯片 ID 寄存器 1。



	Offset Address			Register Name			Total Reset Value		
	0xEE4			SC_SYSID1			0x01		
Bit	7	6	5	4	3	2	1	0	
Name	sysid1								
Reset	0	0	0	0	0	0	0	1	
Bits	Access	Name	Description						
[7:0]	RO	sysid1	读该寄存器返回 0x01。						

SC_SYSID2

SC_SYSID2 为芯片 ID 寄存器 2。

	Offset Address			Register Name			Total Reset Value		
	0xEE8			SC_SYSID2			0x20		
Bit	7	6	5	4	3	2	1	0	
Name	sysid2								
Reset	0	0	1	0	0	0	0	0	
Bits	Access	Name	Description						
[7:0]	RO	sysid2	读该寄存器返回 0x20。						

SC_SYSID3

SC_SYSID3 为芯片 ID 寄存器 3。

	Offset Address			Register Name			Total Reset Value		
	0xEEC			SC_SYSID3			0x35		
Bit	7	6	5	4	3	2	1	0	
Name	sysid3								
Reset	0	0	1	1	0	1	0	1	
Bits	Access	Name	Description						
[7:0]	RO	sysid3	读该寄存器返回 0x35。						



3.11 电源管理与低功耗模式控制

3.11.1 概述

芯片的低功耗模式用来有效的减少芯片的功耗，Hi3520 提供多种低功耗的控制来动态降低芯片的功耗：

- 系统工作模式控制
除了 NORMAL 模式之外，各种模式对功耗都有一定的减小作用，可以根据实际的功耗要求和功能要求选择不同的工作模式。
- 时钟门控和时钟频率调整
提供时钟关断功能，可以关闭不使用模块的时钟，减少芯片的功耗。系统工作的时钟频率可以进行调整，在满足功能的情况下可以调节时钟频率，动态降低芯片功耗。
- DDR 低功耗控制
可以启动 DDR 的自刷新模式，来降低整个产品的功耗。

3.11.2 系统工作模式

系统提供 4 种工作模式，请参见“3.10.3 功能描述”中的“系统运行模式控制”。

3.11.3 时钟门控和时钟频率调整

系统提供以下模块的时钟门控功能，在模块空闲的时候，可以关闭相应的时钟，降低芯片功耗，操作流程可以参见以下模块时钟门控部分。

- VEDU
- PCI
- MMC
- VIU
- VOU
- TDE
- CIPHER
- GMAC
- SIO
- UART
- SMI
- NANDC
- USB 2.0 HOST
- SSP
- IR

Normal 模式下，系统可以通过调整工作频率来降低芯片功耗，步骤如下：

步骤 1 关闭业务模块，使其不访问 DDR。

步骤 2 系统进入 Flash 或者 TCM（Tightly-Coupled Memory）中运行。



- 步骤 3 配置 DDRC_DLL_CONFIG[dll_cali_en]为 1，重新校准 DLL（Delay Locked Loop）。
- 步骤 4 配置 DDRC_CTRL[sr_req]为 1，请求进入自刷新模式。
- 步骤 5 查询 DDRC_STATUS[in_sr]位，直到其值为 1，则执行步骤 6。
- 步骤 6 配置 SC_PLLCTRL[27:3]为 PLL 的稳定时间。
- 步骤 7 配置 SC_PERCTRL0 和 SC_PERCTRL1，进行 PLL 分频比控制。切换 PLL 的时钟，即等待 PLL 时钟稳定，输出给 DDRC。
- 步骤 8 配置 DDRC_CTRL[sr_req]为 0，请求退出自刷新模式。
- 步骤 9 查询 DDRC_STATUS[in_sr]位，直到其值为 0。
- 步骤 10 配置 DDRC_DLL_CONFIG[dll_cali_en]为 0，禁止 DLL 的重校准。
- 步骤 11 程序运行在 DDR 中，并且启用业务模块工作。

----结束

除了调整系统工作频率外，还可单独调整某些模块的工作频率，进一步降低系统的功耗。具体的操作请参见 PCI、SMI 和 MMC 模块的时钟配置部分。

3.11.4 DDR 低功耗控制

提供对芯片中 DDR 控制器的管脚以及对端 DDR 芯片的动态功耗控制：

- 可通过配置 DDRC_CONFIG[pd_en]和 DDRC_CONFIG[pd_prd]，使能 DDR 低功耗自动进入功能，在总线没有访问时，DDRC 控制 DDR 自动进入低功耗模式，节省功耗。
- 可通过配置 DDRC_CTRL[sr_req]为 1，控制 DDR 进入自刷新模式，降低功耗。在请求进入自刷新模式时，需要满足下面的条件：系统在启动 DDR 的自刷新功能时，必须将程序放到片内的程序存储器或 flash 中进行运行，通过 DDR 控制器启动 DDR 的自刷新功能。



4 存储控制器

4.1 DDR 控制器

4.1.1 概述

DDRC (DDR2 SDRAM Controller) 对外提供 DDR2 接口, 用于完成 DDR2 SDRAM 的访问。Hi3520 包含了 2 个 DDRC 接口: DDRC0、DDRC1。

4.1.2 特点

DDRC 有如下特点:

- 提供 1 个 DDR2 SDRAM 片选空间, 兼容 16bit 模式和 32bit 模式 (DDRC1 仅支持 16bit 模式)。
- DDRC 支持最大存储空间: 16bit 模式为 256MB; 32bit 模式为 512MB。
- 支持 DDR2 SDRAM 的 burst 为 4 的传输模式。
- 支持时序参数可配, 以适应不同器件不同频率的需求。
- 控制 DDR SDRAM 以及 DDR2 SDRAM 的 AUTO Refresh、SELF Refresh。
- 支持低功耗模式。
- 支持 DDR2 SDRAM 的工作频率 400MHz。

4.1.3 信号描述

DDRC 的接口信号描述如表 4-1 所示。

表4-1 DDRC 接口信号描述

名称	方向	描述	对应管脚
DDRA_CKP0 /DDR_B_CKP0	O	正向差分时钟 0。	DDRA_CKP0 /DDR_B_CKP0
DDRA_CKN0 /DDR_B_CKN0	O	反向差分时钟 0。	DDRA_CKN0 /DDR_B_CKN0



名称	方向	描述	对应管脚
DDRA_CKP1 /DDR_B_CKP1	O	正向差分时钟 1。	DDRA_CKP1 /DDR_B_CKP1
DDRA_CKN1 /DDR_B_CKN1	O	反向差分时钟 1。	DDRA_CKN1 /DDR_B_CKN1
DDRA_CKE /DDR_B_CKE	O	DDR2 SDRAM 时钟使能。	DDRA_CKE /DDR_B_CKE
DDRA_CSN /DDR_B_CSN	O	DDR2 SDRAM 片选。	DDRA_CSN /DDR_B_CSN
DDRA_RASN /DDR_B_RASN	O	DDR2 SDRAM RAS。	DDRA_RASN /DDR_B_RASN
DDRA_CASN /DDR_B_CASN	O	DDR2 SDRAM CAS。	DDRA_CASN /DDR_B_CASN
DDRA_WEN /DDR_B_WEN	O	DDR2 SDRAM WE。	DDRA_WEN /DDR_B_WEN
DDRA_ODT /DDR_B_ODT	O	DDR2 SDRAM 片上终端使能。	DDRA_ODT /DDR_B_ODT
DDRA_BA0 /DDR_B_BA0	O	DDR2 SDRAM Bank 地址 0。	DDRA_BA0 /DDR_B_BA0
DDRA_BA1 /DDR_B_BA1	O	DDR2 SDRAM Bank 地址 1。	DDRA_BA1 /DDR_B_BA1
DDRA_BA2 /DDR_B_BA2	O	DDR2 SDRAM Bank 地址 2。	DDRA_BA2 /DDR_B_BA2
DDRA_ADR[13:0] /DDR_B_ADR[13:0]	O	DDR SDRAM 地址信号。	DDRA_ADR13~ DDRA_ADR 0 /DDR_B_ADR13~ DDR_B_ADR 0
DDRA_LDM0 /DDR_B_LDM0	O	DDR2 SDRAM0 低数据位屏蔽信号。对应数据总线 DQ0~DQ7。	DDRA_LDM0 /DDR_B_LDM0
DDRA_UDM0 /DDR_B_UDM0	O	DDR2 SDRAM0 高数据位屏蔽信号。对应数据总线 DQ8~DQ15。	DDRA_UDM0 /DDR_B_UDM0
DDRA_LDM1 /DDR_B_LDM1	O	DDR2 SDRAM1 低数据位屏蔽信号。对应数据总线 DQ16~DQ23。	DDRA_LDM1 /DDR_B_LDM1



名称	方向	描述	对应管脚
DDRA_UDM1 /DDR_B_UDM1	O	DDR2 SDRAM1 高数据位屏蔽信号。对应数据总线 DQ24~DQ31。	DDRA_UDM1 /DDR_B_UDM1
DDRA_DQSP0 /DDR_B_DQSP0	I/O	正向 DQS0。对应数据总线 DQ0~DQ7。	DDRA_DQSP0 /DDR_B_DQSP0
DDRA_DQSN0 /DDR_B_DQSN0	I/O	反向 DQS0。对应数据总线 DQ0~DQ7。	DDRA_DQSN0 /DDR_B_DQSN0
DDRA_DQSP1 /DDR_B_DQSP1	I/O	正向 DQS1。对应数据总线 DQ8~DQ15。	DDRA_DQSP1 /DDR_B_DQSP1
DDRA_DQSN1 /DDR_B_DQSN1	I/O	反向 DQS1。对应数据总线 DQ8~DQ15。	DDRA_DQSN1 /DDR_B_DQSN1
DDRA_DQSP2 /DDR_B_DQSP2	I/O	正向 DQS2。对应数据总线 DQ16~DQ23。	DDRA_DQSP2 /DDR_B_DQSP2
DDRA_DQSN2 /DDR_B_DQSN2	I/O	反向 DQS2。对应数据总线 DQ16~DQ23。	DDRA_DQSN2 /DDR_B_DQSN2
DDRA_DQSP3 /DDR_B_DQSP3	I/O	正向 DQS3。对应数据总线 DQ24~DQ31。	DDRA_DQSP3 /DDR_B_DQSP3
DDRA_DQSN3 /DDR_B_DQSN3	I/O	反向 DQS3。对应数据总线 DQ24~DQ31。	DDRA_DQSN3 /DDR_B_DQSN3
DDRA_DQ[31:0] /DDR_B_DQ[31:0]	I/O	数据总线。	DDRA_DQ31~ DDRA_DQ0 /DDR_B_DQ31~ DDR_B_DQ0

信号名称说明：

CS: Chip Select;

RAS: Row Address Select;

CAS: Column Address Select;

WE: Write Enable;

DQS: Data Strobe;

DM: Data Mask。

4.1.4 功能描述

4.1.4.1 应用框图

**注意**

在以下文字和图形描述中，将 DDRC0、DDRC1 统一描述为 DDRC；DDRA、DDRB 则统一描述为 DDR，32bit 模式只针对 DDRC0。

DDRC 实现了 SoC 系统中的 CPU 等主设备对外存 DDR2 SDRAM 的数据访问。通过 CPU 配置 DDRC 的时序参数寄存器，可以支持符合 JEDEC (JESD79) 标准的 DDR2 SDRAM 器件。

DDRC 支持两种 DDR2 SDRAM 的互联模式：

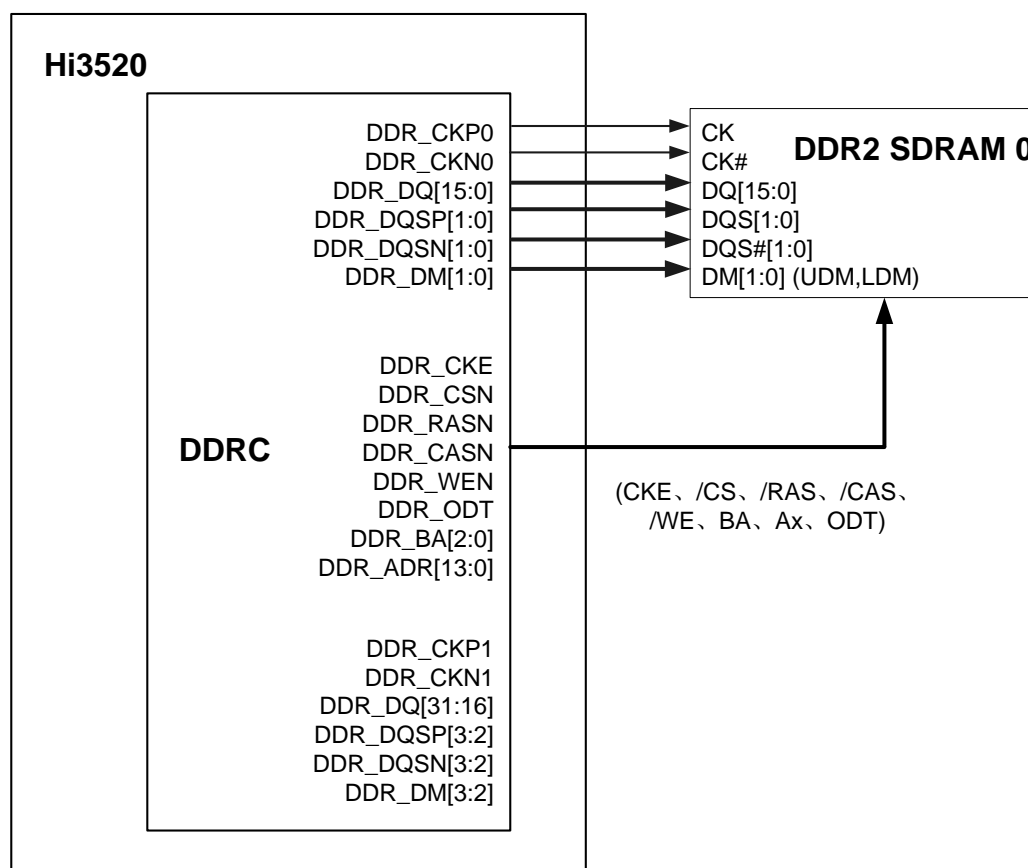
- 16bit 模式

DDRC 只与 1 片数据总线位宽为 16bit 的 DDR2 SDRAM 相连，其连接示意图如图 4-1 所示。

- 32bit 模式

DDRC0 与 2 片数据总线位宽为 16bit 的 DDR2 SDRAM 相连（DDRC1 不支持 32bit 模式），其连接示意图如图 4-2 所示。

图4-1 DDRC 与 1 片 DDR2 SDRAM 的互连示意图



连接说明：

#表示低电平有效。

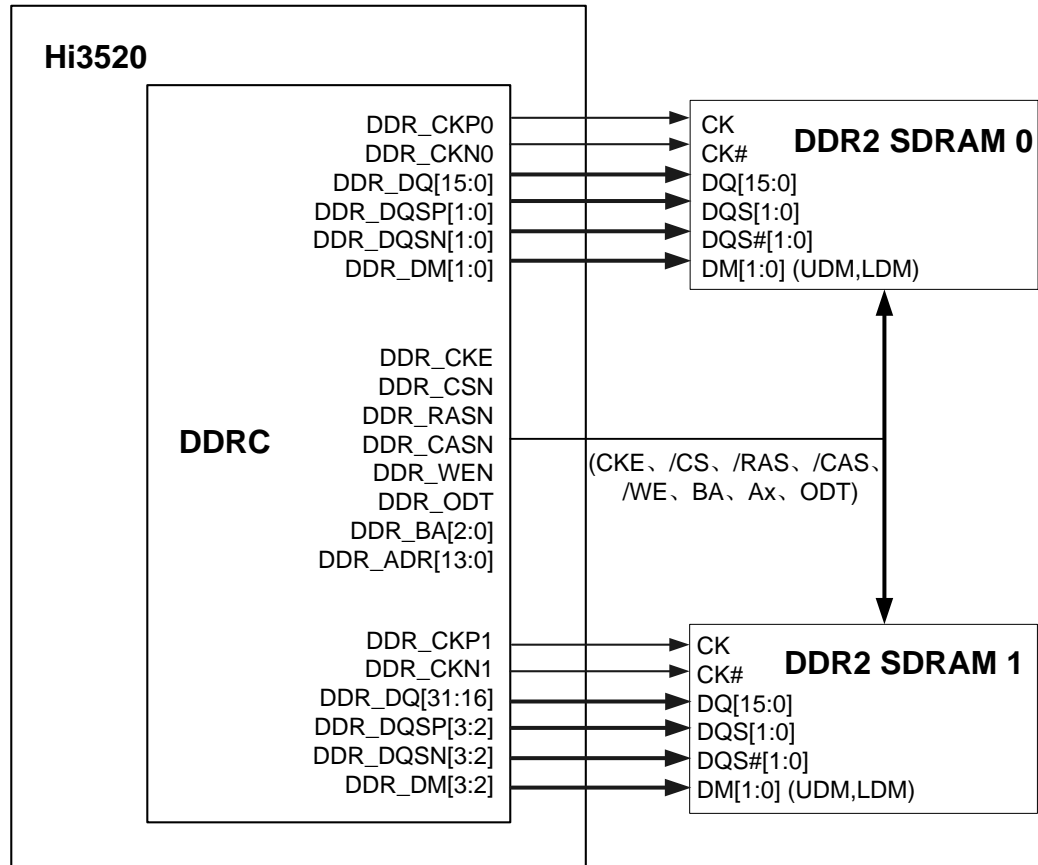
DDR2 SDRAM0 是数据总线位宽为 16bit 的存储器件。图 4-1 中示意了 Hi3520 在 16bit DDR2 模式下 DDRC 与 DDR2 SDRAM 的连接方式。

DDRC 的 DDR_CKE、DDR_CSN、DDR_RASN、DDR_CASN、DDR_WEN、DDR_BA[2:0]、DDR_ADR[13:0]，分别与 DDR2 SDRAM0 命令控制信号相连（图 4-1 中 CKE、/CS、/RAS、/CAS、/WE、BA、Ax）。

DDRC 的 DDR_CKP1、DDR_CKN1、DDR_DQ[31:16]、DDR_DQSP[3:2]、DDR_DQSN[3:2]、DDR_DM[3:2] 输出悬空。

DDR2 SDRAM 容量小于 1Gbit 的场景下，DDRC 的 DDR_BA[2] 信号输出悬空。

图4-2 DDRC 与 2 片 DDR2 SDRAM 的互连示意图



连接说明：

#表示低电平有效。

DDR2 SDRAM0 和 DDR2 SDRAM1 都是数据总线宽度为 16bit 的存储器件。图 4-2 示意在 32bit DDR2 模式下 DDRC0 与 DDR2 SDRAM 的连接方式。

DDRC 的 DDRA_CKE、DDRA_CSN、DDRA_RASN、DDRA_CASN、DDRA_WEN、DDRA_BA[2:0]、DDRA_ADR[13:0]，分别与 DDR2 SDRAM0 命令控制信号相连（图 4-2 中 CKE、/CS、/RAS、/CAS、/WE、BA、Ax）。

DDR2 SDRAM 容量小于 1Gbit 的场景下，DDRC0 的 DDR_BA[2] 信号输出悬空。

4.1.4.2 功能原理

DDRC 支持主流 DRAM 厂商的 DDR2 SDRAM 如表 4-2 所示。（表中针对 DDR2 SDRAM 的工作频率进行描述，没有考虑容量等因素的限制）



表4-2 DDRC 支持的 DDR2 SDRAM 列表

厂商	200MHz	333MHz	400MHz	备注
JESD79 (DDR2 Standard)	DDR2-400 DDR2-533 DDR2-667 DDR2-800	DDR2-667 DDR2-800	DDR2-800	1、2
Micron	-5E DDR2-400 -37E DDR2-533 -3 DDR2-667 -3E DDR2-667 -25 DDR2-800 -25E DDR2-800	-3 DDR2-667 -3E DDR2-667 -25 DDR2-800 -25E DDR2-800	-25 DDR2-800 -25E DDR2-800	3
ELPIDA	-4A DDR2-400 -5C DDR2-533 -6E DDR2-667 -6C DDR2-667 -8E DDR2-800	-6E DDR2-667 -6C DDR2-667 -8E DDR2-800	-8E DDR2-800	3
Hynix	-E3 DDR2-400 -C4 DDR2-533 -Y4 DDR2-667 -Y5 DDR2-667 -S5 DDR2-800 -S6 DDR2-800	-Y4 DDR2-667 -Y5 DDR2-667 -S5 DDR2-800 -S6 DDR2-800	-S5 DDR2-800 -S6 DDR2-800	3
Samsung	-CC DDR2-400 -D5 DDR2-533 -E6 DDR2-667 -E7 DDR2-800	-E6 DDR2-667 -E7 DDR2-800	-E7 DDR2-800	3

备注说明：

备注 1: DDRC 支持满足 JESD79 标准的绝大部分器件，在不同的工作模式下，DDRC 只能支持 DRAM 工作频率高于或等于 DDRC 工作频率的器件。在使用非表 4-2 中厂商的器件时，可以参考该标准来选取器件和等级。

备注 2: 以各个厂商的 DDR2 SDRAM 的工作频率为判断的基本元素。各个厂商的器件在同一工作频率下会有多种容量、位宽等不同版本，DDRC 支持所有表 4-2 中 DDR2 SDRAM 类型。对于 DDR2 SDRAM 容量和位宽的选取，则根据系统芯片的实际应用场景来作出选择。

备注 3: 只支持 DDR2 SDRAM 的 BurstLength 为 4 的模式。

DDRC 按照 JESD79 DDR2 SDRAM 的时序设计实现，通过发送 DDR2 SDRAM 的命令字，完成对 DDR2 SDRAM 的数据访问和状态控制。包括 DDR2 SDRAM 的读写访问、AUTO Refresh、低功耗控制等功能。



命令真值表

DDRC 支持 DDR2 SDRAM 的主要的命令字。DDRC 的命令真值表（Command Truth Table）如表 4-3 所示。

表4-3 DDRC 命令真值表

FUNCTION	DDR CKE	DDR CSN	DDR RASN	DDR CASN	DDR WEN	DDRADR			DDRBA
						11	AP(10)	9:0	
DESELECT	H	H	X	X	X	X	X	X	X
NOP	H	L	H	H	H	X	X	X	X
ACTIVE	H	L	L	H	H	V	V	V	V
READ	H	L	H	L	H	V	V	V	V
WRITE	H	L	H	L	L	V	V	V	V
PRECHARGE	H	L	L	H	L	X	L	X	V
PRECHARGE ALL	H	L	L	H	L	X	H	X	X
AUTO Refresh	H	L	L	L	H	X	X	X	X
SELF Refresh	L	L	L	L	H	X	X	X	X
MODE REGISTER SET	H	L	L	L	L	V	V	V	V

注 1：DDRADR 中的 AP 位用于区分当前的 PRECHARGE 命令为单 BANK 的 PRECHARGE 还是所有 BANK 的 PRECHARGE。对于 AP 的详细定义请参见 DDR 的 JEDEC 标准。

注 2：H 表示高电平；L 表示低电平；V 表示有效；X 表示不关心。

注 3：DDRC 只支持 DDR2 的 Burst Length 为 4 的工作方式，不支持 DDR2 的 Burst Terminate 的操作。

自动刷新

当配置 `DDRC_TIMING2[taref]` 为非零值，DDRC 自动产生周期性 AUTO Refresh 命令，完成对 DDR2 SDRAM 的刷新操作。常温下，DDR2 SDRAM 要求在 64ms 内必须完成 8192 次自动刷新操作，即自动刷新操作的周期为 7.8μs。`DDRC_TIMING2[taref]` 的配置值（Taref）与自动刷新周期（T=7.8μs）的关系为：

$$T \geq \text{Taref} \times 16 \text{DDR 时钟周期}$$

在配置了 `DDRC_TIMING2[taref]`，DDRC 内部的计数器会自动加载 taref 值进行减计数，当计数器值为 0 时，DDRC 发起一次自动刷新操作，同时重新加载 taref 值进行计数。



低功耗管理

DDRC 支持两种模式的低功耗管理：

- 普通低功耗模式
当系统进入空闲状态 `pd_prd` 个时钟周期后，DDRC 控制 DDR2 SDRAM 进入到 PowerDown 的低功耗模式。
配置 `DDRC_CONFIG[pd_en]` 和 `[pr_prd]` 为有效值，DDRC 在系统空闲状态下会自动控制 DDR2 SDRAM 进入低功耗状态。当 `DDRC_CONFIG[pd_en]` 配置为 1，DDRC 在 `DDRC_CONFIG[pr_prd]` 个总线时钟周期内总线没有任何访问操作，则控制 DDR2 SDRAM 进入低功耗模式。
- 自刷新低功耗模式
当系统需要进入到待机模式时，可通过配置 DDRC 的寄存器，控制 DDR2 SDRAM 进入到自刷新低功耗模式。该模式下，可以将 DDR2 SDRAM 的功耗降至最低，同时保持 DDR2 SDRAM 中的数据，此时系统不访问 DDR2 SDRAM。
配置 `DDRC_CTRL[sr_req]`，DDRC 将控制 DDR2 SDRAM 进入自刷新模式。当配置 `DDRC_CTRL[sr_req]` 为 1 时，DDRC 完成正在执行的访问之后，控制 DDR2 SDRAM 进入自刷新模式，并不再响应总线上的请求；当配置 `DDRC_CTRL[sr_req]` 为 0 时，退出自刷新模式。

仲裁机制

DDRC 实现了优先级调度算法。配置 `DDRC_QOS[pri]` 寄存器，DDRC 将给总线的命令添加优先级属性，然后根据命令的优先级属性进行调度，实现对 DDR2 SDRAM 的高效率访问。配置 `DDRC_QOS[qos_en]` 和 `[qos]`，DDRC 将给总线的命令添加响应延迟属性，然后根据响应延迟优先的调度算法，实现总线命令响应延迟的保证。

地址映射方式

DDRC 实现了将系统总线的地址映射为 DDR2 SDRAM 的地址，完成系统对 DDR2 SDRAM 的访问。配置 `DDRC_CONFIG[mem_map]`、`[mem_row]` 和 `[mem_col]` 寄存器，DDRC 根据地址映射算法将系统总线地址转换为 DDR2 SDRAM 地址。

下面示例说明系统总线地址和 DDR2 SDRAM 地址的映射算法。假设系统总线地址低 29 位为 `BUS_ADR[28:0]`，有效地址为 `BUS_ADR[m-1:0]`，Hi3520 的 DDRC 地址为 `DDR_ADR[13:0]`。DDR2 SDRAM 的行地址为 `DDR_ROW[x-1:0]`，列地址为 `DDR_COL[y-1:0]`，BANK 地址为 `DDR_BA[z-1:0]`，以 1Gbit DDR2 SDRAM 为例，其地址为：`[row_addr、col_addr、bank_addr]` 为 `[13、10、3]`，即 `x=13、y=10、z=3`。DDRC 的存储数据总线宽度为 `DW`，此时地址映射关系为：

- `DDRC_CONFIG[mem_map]` 为 0 时，RBC 映射方式：
$$\text{BUS_ADR}[m-1:0] = \{\text{DDR_ROW}[x-1:0], \text{DDR_BA}[z-1:0], \text{DDR_COL}[y-1:0], \text{DW}\{b0\}\}$$
- `DDRC_CONFIG[mem_map]` 为 1 时，BRC 映射方式：
$$\text{BUS_ADR}[m-1:0] = \{\text{DDR_BA}[z-1:0], \text{DDR_ROW}[x-1:0], \text{DDR_COL}[y-1:0], \text{DW}\{b0\}\}$$



说明

在上面的表达式中，参数的关系满足： $m=x+y+z+DW$ ，其中：



- 当 DDRC 为 16bit 应用模式时，DW 为 1。
- 当 DDRC 为 32bit 应用模式时，DW 为 2。

RBC 模式下，A10 作为 DDR 的 AP 功能位，系统总线地址到 DDR2 SDRAM 地址的映射如表 4-4 和表 4-5 所示。

BRC 模式下，按照上面的表达式进行地址映射。

表4-4 DDRC 16bit 模式地址映射表

存储器类型	行地址	列地址	DDRBA			行地址 列地址	DDRADR						
	宽度		宽度	2	1		0	13	12	11	10/AP	9	8
256Mbit 4bank													
16×16	13	9	-	11	10	行地址	-	24	23	22	21	20	[19:12]
						列地址	-	-	-	AP	-	9	[8:1]
512Mbit 4bank													
32×16	13	10	-	12	11	行地址	-	25	24	23	22	21	[20:13]
						列地址	-	-	-	AP	10	9	[8:1]
1Gbit 8bank													
64×16	13	10	13	12	11	行地址	-	26	25	24	23	22	[21:14]
						列地址	-	-	-	AP	10	9	[8:1]
2Gbit 8bank													
128×16	14	10	13	12	11	行地址	27	26	25	24	23	22	[21:14]
						列地址	-	-	-	AP	10	9	[8:1]

表4-5 DDRC 32bit 模式地址映射表

存储器类型	行地址	列地址	DDRBA			行地址 列地址	DDRADR						
	宽度		宽度	2	1		0	13	12	11	10/AP	9	8
256Mbit 4bank													
16×16	13	9	-	12	11	行地址	-	25	24	23	22	21	[20:13]
						列地址	-	-	-	AP	-	10	[9:2]
512Mbit 4bank													
32×16	13	10	-	13	12	行地址	-	26	25	24	23	22	[21:14]
						列地址	-	-	-	AP	11	10	[9:2]



存储器类型 Mbit×bw	行地址 宽度	列地址 宽度	DDRBA			行地址 列地址	DDRADR						
			2	1	0		13	12	11	10/AP	9	8	[7:0]
1Gbit 8bank													
64×16	13	10	14	13	12	行地址	-	27	26	25	24	23	[22:15]
						列地址	-	-	-	AP	11	10	[9:2]
2Gbit 8bank													
128×16	14	10	14	13	12	行地址	28	27	26	25	24	23	[22:15]
						列地址	-	-	-	AP	11	10	[9:2]

4.1.5 工作方式

4.1.5.1 时钟门控

系统进入低功耗模式后，可以关闭 DDRC 的工作时钟。同样，在系统恢复到正常模式之前，需要先打开 DDRC 的工作时钟。

DDRC 时钟关闭流程步骤如下：

- 步骤 1 系统进入 Flash 运行。
- 步骤 2 配置 `DDRC_CONFIG[sr_cc]` 为 1，使能器件时钟控制。
- 步骤 3 配置 `DDRC_CTRL[sr_req]` 为 1，请求进入自刷新模式。
- 步骤 4 查询 `DDRC_STATUS[in_sr]`，直到其值为 1，则执行步骤 5。
- 步骤 5 关断 DDRC 时钟。

----结束

DDRC 时钟打开流程步骤如下：

- 步骤 1 系统进入 NORMAL 模式，打开 DDRC 时钟。
- 步骤 2 等待 DDRC 内部 DLL (Delay Lock Loop) 锁定。
- 步骤 3 配置 `DDRC_CTRL[sr_req]` 为 0，请求退出自刷新模式。
- 步骤 4 查询 `DDRC_STATUS[in_sr]` 位，直到其值为 0，则执行步骤 5。
- 步骤 5 进入正常工作模式。

----结束



4.1.5.2 软复位

DDRC 不能进行单独的复位操作。只能在全局软复位时，才能复位 DDRC。复位之后，DDRC 按照初始化流程对 DDR2 SDRAM 器件进行初始化。

4.1.5.3 初始化

系统上电之后，必须先完成 DDR2 SDRAM 的初始化操作，系统才能访问 DDR2 SDRAM。在进行初始化之前需要注意以下几点：

- 对 DDR2 SDRAM 进行上电操作时，需要遵循 JEDEC 标准。即先提供 VDD，然后提供 VDDQ，最后提供 VREF 和 VTT。
- 该初始化过程需要在系统进入 NORMAL 模式后进行。

DDR2 SDRAM 的初始化步骤如下：

步骤 1 软件等待 200 μ s 以上。

步骤 2 把 DDRC_CTRL 寄存器配置为 0x0000_0000，退出自刷新状态。

步骤 3 软件等待 400ns 以上。

步骤 4 把 DDRC_EMRS01 寄存器设为 0x32，把 DDRC_EMRS23 寄存器设为 0x00，配置 DDR2 SDRAM 器件的模式寄存器和扩展模式寄存器。这里把 DDR 的读延迟（cas latency）设为 6，突发长度（burst length）设定为 8。扩展模式寄存器根据实际需要，可设定 DDR2 SDRAM 器件的模式寄存器值，这里设置为 0。

说明

- Cas Latency 必须和 DDRC_EMRS01[tc]设置成同一个值。
- DDRC_EMRS01 寄存器对应 DDR2 SDRAM 的模式寄存器（MRS）和扩展模式寄存器 1（EMRS1）。配置该寄存器时，参考 DDR2 SDRAM 器件手册的模式寄存器的 A15-A0 的描述，不考虑配置模式寄存器的最高 3 位寄存器选择位，即 bank 地址。
- 配置 DDR2 的扩展模式寄存器 1（EMRS1）时，应将该模式寄存器中的 RDQS 和 \overline{DQS} 都配置为禁止模式。

步骤 5 根据器件的行列地址位宽配置 DDRC_CONFIG 寄存器为 0x7000_7022，表示器件的地址映射模式为 RBC 模式，AP 为 A10，列地址宽度为 10，行地址宽度为 13，初始化进行 7 次自动刷新操作（AUTO Refresh）。DDRC_CONFIG[pd_en]采用上电复位值，即低功耗模式采用复位值。

说明

低功耗配置的上电复位值是关闭的，初始化过程中必须关闭低功耗自动进入功能，在正常使用中，一般建议使能低功耗的控制，可以降低功耗。

步骤 6 配置 DDRC_TIMING0、DDRC_TIMING1、DDRC_TIMING2、DDRC_TIMING3 寄存器的值，其中 cl 值必须跟 DDRC_EMRS01 模式寄存器中的配置一致。

步骤 7 配置 DDRC_ODT_CONFIG 寄存器（建议使用默认值，即在对 DDR2 进行写操作时，odt 有效；读操作时，odt 无效）。

步骤 8 配置 DDRC_QOS。根据系统需求配置不同端口的 DDRC_QOS。建议 DDRC_QOS[qos] 的配置值大于 1，配置范围为 0x002~0x3FF。

步骤 9 将 DDRC_CTRL 寄存器设置为 0x2，启动初始化过程。



步骤 10 等待 `DDRC_STATUS` bit[3]寄存器的值变为 1，初始化完成。

----结束

完成以上步骤以后，DDR2 SDRAM 可以正常工作。

4.1.6 寄存器概览

DDRC0 和 DDRC1 功能上完全相同，基地址分配不同。

DDR0 存储空间基地址为：0xC000_0000；

DDR1 存储空间基地址为：0xE000_0000；

DDRC0 寄存器基地址为：0x2011_0000；

DDRC1 寄存器基地址为：0x2012_0000。

DDRC 寄存器概览如表 4-6 所示。

表4-6 寄存器概览

偏移地址	名称	描述	页码
0x000	DDRC_STATUS	状态寄存器	4-13
0x004	DDRC_CTRL	控制寄存器	4-13
0x008	DDRC_EMRS01	模式配置寄存器 1	4-14
0x00C	DDRC_EMRS23	模式配置寄存器 2	4-16
0x010	DDRC_CONFIG	配置寄存器	4-16
0x020	DDRC_TIMING0	时序寄存器 0	4-18
0x024	DDRC_TIMING1	时序寄存器 1	4-20
0x028	DDRC_TIMING2	时序寄存器 2	4-21
0x02C	DDRC_TIMING3	时序寄存器 3	4-22
0x094	DDRC_ODT_CONFIG	ODT 配置寄存器	4-23
0x104	DDRC_QOS_CONFIG	QOS 配置寄存器	4-23
0x108+n×4	DDRC_QOS	添加优先级属性寄存器	4-24
0x200	DDRC_DLL_CONTROL	DLL 控制寄存器	4-25
0x244	DDRC_DLL_SRST	DLL 复位寄存器	4-25
0x268	DDRC_IO_CONFIG	IO 配置寄存器	4-26

其中：n 为 16ID 的 QOS 变量，取值范围是 0~15。



4.1.7 寄存器描述

DDRC_STATUS

DDRC_STATUS 为状态寄存器。

	Offset Address	Register Name	Total Reset Value															
	0x000	DDRC_STATUS	0x0000_004D															
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																	
Name	reserved														in_init	in_sr	reserved	busy
Reset	0 1 0 0 0 1 1 0 1																	
Bits	Access	Name	Description															
[31:4]	RO	reserved	保留。															
[3]	RO	in_init	初始化（Initialization）状态指示。 0：初始化中； 1：正常工作模式。															
[2]	RO	in_sr	自刷新（SELF Refresh）状态指示。 0：正常状态； 1：自刷新状态。															
[1]	RO	reserved	保留。															
[0]	RO	busy	控制器状态位。 0：空闲状态； 1：忙状态。															

DDRC_CTRL

DDRC_CTRL 为控制寄存器。

Offset Address		Register Name		Total Reset Value						
0x004		DDRC_CTRL		0x0000_0001						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved							clk_ratio	init_req	sr_req
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1		
Bits	Access	Name	Description							
[31:2]	-	reserved	保留。							
[1]	RW	init_req	硬件初始化控制，写 1 有效。 0: 正常工作状态; 1: 启动初始化操作，完成操作后，自动清零。							
[0]	RW	sr_req	自刷新 (SELF Refresh) 请求控制位。 0: 正常工作状态，或退出自刷新状态; 1: 请求进入自刷新状态。							

DDRC_EMRS01

DDRC_EMRS01 为模式配置寄存器 1。

Offset Address		Register Name		Total Reset Value											
0x008		DDRC_EMRS01		0x0000_0000											
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0							
Name	reserved			ddr_rtt1	ddr_al	ddr_rtt0	ddr_drv	ddr_dll_en	reserved	ddr_wr	ddr_rst	ddr_tm	ddr_cas	ddr_bt	ddr_bl
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description												
[31:23]	RW	reserved	保留。必须配置为 0。												



[22]	RW	ddr_rtt1	DDR2 SDRAM 匹配电阻配置。与 ddr_rtt0 共同配置。 [ddr_rtt1、ddr_rtt0]: 00: 禁止匹配; 01: 75Ω; 10: 150Ω; 11: 50Ω。 该值的配置需要参考应用中 DDR2 SDRAM 厂商手册
[21:19]	RW	ddr_al	保留。必须配置为 0。
[18]	RW	ddr_rtt0	DDR2 SDRAM 匹配电阻配置。与 ddr_rtt1 共同配置。 [ddr_rtt1、ddr_rtt0]: 00: 禁止匹配; 01: 75Ω; 10: 150Ω; 11: 50Ω。 该值的配置需要参考应用中 DDR2 SDRAM 厂商手册。
[17]	RW	ddr_drv	DDR2 SDRAM 驱动能力配置。 0: 强驱动; 1: 弱驱动。 强和弱可以由 SI 仿真确定。
[16]	RW	ddr_dll_en	DDR2 SDRAM DLL 使能。 0: 禁止; 1: 使能。
[15:12]	RW	reserved	保留。必须配置为 0。
[11:9]	RW	ddr_wr	DDR2 SDRAM 写恢复时间配置。 010: 2 个时钟周期; 011: 3 个时钟周期; 100: 4 个时钟周期; 101: 5 个时钟周期; 110: 6 个时钟周期; 其他: 保留。
[8]	RW	ddr_rst	DDR2 SDRAM DLL 复位控制。 0: 不复位; 1: 复位。



[7]	RW	ddr_tm	DDR2 SDRAM 测试模式。 0: 普通模式; 1: 测试模式。
[6:4]	RW	ddr_cas	DDR2 SDRAM CAS 配置。 011: 3 个时钟周期; 100: 4 个时钟周期; 101: 5 个时钟周期; 110: 6 个时钟周期; 其他: 保留。
[3]	RW	ddr_bt	DDR2 SDRAM burst 类型配置。 0: 顺序模式; 1: 保留。
[2:0]	RW	ddr_bl	DDR2 SDRAM burst 长度配置。 010: 4; 011: 8; 其他: 保留。

DDRC_EMRS23

DDRC_EMRS23 为模式配置寄存器 2。

	Offset Address				Register Name				Total Reset Value																							
	0x00C				DDRC_EMRS23				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	emrs3								emrs2																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:16]	RW		emrs3		保留。必须配置为 0。																											
[15:0]	RW		emrs2		保留。必须配置为 0。																											

DDRC_CONFIG

DDRC_CONFIG 为配置寄存器。



Offset Address		Register Name		Total Reset Value				
0x010		DDRC_CONFIG		0x0000_6022				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	init_arefnum	pr_prd		reserved sr_cc reserved pd_en	mem_type	reserved ap mem_map mem_bank	mem_row	reserved mem_col
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0	0 0 0 0	0 0 1 0	0 0 1 0
Bits	Access	Name	Description					
[31:28]	RW	init_arefnum	硬件初始化需要发出的自动刷新（AUTO Refresh）次数。 0x0~0x2: 2个; 0x3~0xF: n个时钟周期。 (n表示对应的十进制值)					
[27:20]	RW	pr_prd	自动执行 power down 的等待 CLK 数，当控制器连续没有收到请求就自动关闭器件的 CKE。 0x00: 在 pd_en 有效的情况下一个时钟周期没有收到请求，自动进入低功耗模式； 0x01~0xn: 在 pd_en 有效的情况下 n 个时钟周期没有收到请求，自动进入低功耗模式。 (n表示对应的十进制值，如 0x03: 3个时钟周期没有收到请求，自动进入低功耗模式)					
[19]	RW	reserved	保留。					
[18]	RW	sr_cc	自刷新时钟关闭使能。 0: 禁止; 1: 使能。					
[17]	RW	reserved	保留。					
[16]	RW	pd_en	自动 power down 使能。自动关断 DDR2 SDRAM 器件的 CKE 的使能信号，当连续 PD_PRD 周期没有收到读写请求时，自动关断器件的 CKE 输入。 0: 不自动进入 power down; 1: 自动进入 power down。					
[15:12]	RW	mem_type	外部存储器类型。 0x6: 16bit DDR2; 0x7: 32bit DDR2; 其他: 保留。					
[11:10]	RW	reserved	保留。					



[9]	RW	ap	PRECHARGE ALL 的指示。 0: a10; 1: a8 (在早期器件中有使用, 当前产品很少出现)。
[8]	RW	mem_map	选择 memory 地址译码模式。 0: Row、Bank、Col、Dw; 1: Bank、Row、Col、Dw。
[7]	RW	mem_bank	选择 memory 的 bank 数。 0: 4bank; 1: 8bank。
[6:4]	RW	mem_row	指示单片器件的行地址位数。 000: 11; 001: 12; 010: 13; 011: 14; 100: 15; 101: 16; 其他: 保留。
[3]	RW	reserved	保留。
[2:0]	RW	mem_col	指示单片器件的列地址位数。 000: 8; 001: 9; 010: 10; 011: 11; 100: 12; 其他: 保留。

DDRC_TIMING0

DDRC_TIMING0 为时序寄存器 0。



	Offset Address 0x020								Register Name DDRC_TIMING0								Total Reset Value 0x7FFF_3F1F															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	tmrd			trrd				trp				trcd				reserved	trc							reserved	tras						
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1
	Bits	Access	Name		Description																											
	[31]	RW	reserved		保留。																											
	[30:28]	RW	tmrd		模式寄存器加载（LMR）命令的等待周期。 000~001：1个时钟周期； 010~111：n个时钟周期。 (n表示对应的十进制值，如010：2个时钟周期)																											
	[27:24]	RW	trrd		打开 BANK A 到打开 BANK B 的（ACT bank a to ACT bank b）等待周期。 0x0~0x1：1个时钟周期； 0x2~0xF：n个时钟周期。 (n表示对应的十进制值。如0xF：15个时钟周期)																											
	[23:20]	RW	trp		关闭（PRE period）命令的等待周期。 0x0~0x1：1个时钟周期； 0x2~0xE：n个时钟周期； 0xF：保留。 (n表示对应的十进制值，如0x7：7个时钟周期)																											
	[19:16]	RW	trcd		同 BANK 激活到读或写（ACT to READ or WRITE）命令的等待周期。 0x0~0x3：3时钟周期； 0x4~0xF：n时钟周期。 (n表示对应的十进制值，如0011：3个时钟周期)																											
	[15:14]	RW	reserved		保留。																											
	[13:8]	RW	trc		同 BANK 的激活命令到激活命令（Active a bank to Active a bank）的等待周期。 0x00~0x01：1个时钟周期； 0x02~0x0F：n个时钟周期。 (n表示对应的十进制值，如0x0F：15个时钟周期)																											
	[7:5]	RW	reserved		保留。																											

[4:0]	RW	tras	同 BANK 的激活命令到到关闭命令（ACT to PRE）的等待周期。 0x00~0x01: 1 个时钟周期; 0x00~0x0F: n 个时钟周期。 (n 表示对应的十进制值, 如 0x0F: 15 个时钟周期)
-------	----	------	---

DDRC_TIMING1

DDRC_TIMING1 为时序寄存器 1。

	Offset Address 0x024								Register Name DDRC_TIMING1								Total Reset Value 0xFF01_23FF																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	tsre								trdlat				reserved	trtw				twl				reserved	tcl				trfc									
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1				
	Bits	Access	Name	Description																																
	[31:24]	RW	tsre	退出自刷新（SELF-Refresh）到读命令的等待周期。 0x00~0xFF: n 个时钟周期。 (n 表示对应的十进制值, 如 0xFF: 255 个时钟周期, 一般情况下器件要求 200 个时钟周期)																																
	[23:20]	RW	trdlat	读数据延迟。该参数由板上的 DQS 走线的长度决定。 0x0~0xF: n+1 个周期。 (n 表示对应的十进制值, 如 0x0: 表示 1 个周期的延迟)																																
	[19]	RW	reserved	保留。																																
	[18:16]	RW	trtw	最后一个读数据到第一个写数据延迟。 000~110: n+1 个时钟周期。 (n 表示对应的十进制值, 如 0x0: 表示 1 个周期的延迟)																																
	[15:12]	RW	twl	写命令到写数据的等待周期。 0x0~0xF: n 个时钟周期。 (n 表示对应的十进制值, 如: 0x3: 3 个时钟周期) 注意: 在 DDR2 模式下, twl 配置为 tcl-1, twl 配置时应满足 twl-taond>=1。																																
	[11]	RW	reserved	保留。																																



[10:8]	RW	tcl	DDR 的读命令到读数据的延迟（CAS Latency）。 011: CL=3; 100: CL=4; 101: CL=5; 110: CL=6; 其他: 保留。
[7:0]	RW	trfc	自动刷新命令周期和自动刷新到激活（AREF period or AREF to ACT）命令的等待周期。 0x00~0x01: 1 个时钟周期; 0x02~0xFF: n 个时钟周期。 (n 表示对应的十进制值, 如 0x0F: 15 个时钟周期)

DDRC_TIMING2

DDRC_TIMING2 为时序寄存器 2。

	Offset Address 0x028				Register Name DDRC_TIMING2								Total Reset Value 0x33F3_F7FF																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	tcke	reserved	twtr	twr	reserved	tfaw				reserved	taref																				
Reset	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
Bits	[31:30]		[29:28]		[27:26]		[25:24]																									
Access	RW		RW		RW		RW																									
Name	reserved		tcke		reserved		twtr																									
Description	保留。		DDR 低功耗状态维持的最短时间。 00~01: 1 个时钟周期; 10~11: n 个时钟周期。 (n 表示对应的十进制值)		保留。		写操作最后一个写数据到读命令（Write to Read）的等待周期。 00~01: 1 个时钟周期; 01~11: n 个时钟周期。 (n 表示对应的十进制值, 如 11: 3 个时钟周期)																									

[23:20]	RW	twr	写 (Write Recovery) 恢复的等待周期。 0x0~0x1: 1 个时钟周期; 0x2~0xf: n 个时钟周期。 (n 表示对应的十进制值, 如 0x7: 7 个时钟周期)
[19:18]	RW	reserved	保留。
[17:12]	RW	tfaw	连续 4 个激活命令周期。 0x00~0x3F: n 个时钟周期。 (n 表示对应的十进制值, 如: 0x14: 20 个时钟周期)
[11]	RW	reserved	保留。
[10:0]	RW	taoref	自动刷新周期。 0x000: 自动刷新禁止; 0x001~0x7FF: SDRAM 刷新周期时间为 16×n 时钟周期。 (n 表示对应的十进制值, 如 0x008: 128 个时钟周期, 根据 SDRAM 使用时的运行频率决定)

DDRC_TIMING3

DDRC_TIMING3 为时序寄存器 3。

	Offset Address 0x02C								Register Name DDRC_TIMING3								Total Reset Value 0x0000_0F02																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved								taond				reserved								txard				reserved				trtp							
Reset	0 0 0 0								0 0 0 0				0 0 0 0								0 0 0 0				1 1 1 1				0 0 0 0				0 0 1 0			
Bits																																				
Access	RO								RW				RW								RW				RW				RW							
Name	reserved								taond				reserved								txard				reserved				trtp							
Description	保留。								ODT (On Die Termination) 打开和关闭周期 (Taond/Taofd)。 00: 2/2.5; 01: 3/3.5; 10: 4/4.5; 11: 5/5.5。				保留。								保留。				保留。											



[11:8]	RW	txard	退出 DDR2 SDRAM 低功耗状态等待周期。 0x0~0xF: n 个时钟周期。 (n 代表十进制数, 如 0x7: 为 7 个时钟周期, 取 txp、txard、txards 中的最大值)
[7:3]	RW	reserved	保留。
[2:0]	RW	trtp	读命令到关闭命令的等待延迟。 000~010: 2 个时钟周期; 011~111: n 个时钟周期。 延迟计算公式为: $AL+BL/2+Max(trtp,2)-2$ 。

DDRC_ODT_CONFIG

DDRC_ODT_CONFIG 为 ODT 配置寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x094				DDRC_ODT_CONFIG								0x0000_0001																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										rodt	wodt				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bits	Access		Name		Description																											
[31:2]	RO		reserved		保留。																											
[1]	RW		rodt		读 ODT 使能。 0: 禁止; 1: 使能。																											
[0]	RW		wodt		写 ODT 使能。 0: 禁止; 1: 使能。																											

DDRC_QOS_CONFIG

DDRC_QOS_CONFIG 为 QOS 配置寄存器。

Offset Address		Register Name		Total Reset Value					
0x104		DDRC_QOS_CONFIG		0x0000_3210					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				id_map				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 1 0	0 0 0 1	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	RO	reserved	保留。						
[15:0]	RW	id_map	根据总线上来的 ID 选取 4 位进行 QOS 的配置选择： bit[15:12]: 配置 ID 映射的 bit[3]; bit[11:8]: 配置 ID 映射的 bit[2]; bit[7:4]: 配置 ID 映射的 bit[1]; bit[3:0]: 配置 ID 映射的 bit[0]。 例如: id_map 配置为 0x5320, 这指示总线 ID 的 {ID[5]、ID[3]、ID[2]、ID[0]} 用于 ID 映射, 完成优先级的配置。						

DDRC_QOS

DDRC_QOS 为添加优先级属性寄存器。

Offset Address		Register Name		Total Reset Value						
0x108+n×4		DDRC_QOS		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved				qos_en	reserved	qos		reserved	pri
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:17]	RO	reserved	保留。							
[16]	RW	qos_en	QOS 信号使能。 0: 禁止; 1: 使能。							
[15:14]	RW	reserved	保留。							
[13:4]	RW	qos	通道老化时间配置。推荐该值配置为: 0x002~0x3FF。							
[3]	RW	reserved	保留。							



[2:0]	RW	pri	通道优先级配置。 000: 最高优先级; 001: 次高优先级; 111: 最低优先级。 优先级顺序: 000>001>...>111。
-------	----	-----	---

DDRC_DLL_CONTROL

DDRC_DLL_CONTROL 为 DLL 控制寄存器。

	Offset Address	Register Name	Total Reset Value							
	0x240	DDRC_DLL_CONTROL	0x0000_0000							
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0									
Name	reserved						dll_bypass			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									
Bits	Access	Name	Description							
[31:6]	RO	reserved	保留。							
[5:0]	RW	dll_bypass	DLL_BYPASS 使能。 bit[5]: 命令通道 DLL; bit[4]: ECC 数据通道 DLL; bit[3]: 数据字节 3 通道 DLL; bit[2]: 数据字节 2 通道 DLL; bit[1]: 数据字节 1 通道 DLL; bit[0]: 数据字节 0 通道 DLL。 0: 禁止; 1: 使能。							

DDRC_DLL_SRST

DDRC_DLL_SRST 为 DLL 复位寄存器。

Offset Address		Register Name		Total Reset Value					
0x244		DDRC_DLL_SRST		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								dll_srst_n
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:1]	RO	reserved	保留。						
[0]	RW	dll_srst_n	DLL 软复位。 0: 复位; 1: 不复位。						

DDRC_IO_CONFIG

DDRC_IO_CONFIG 为 IO 配置寄存器。

Offset Address		Register Name		Total Reset Value														
0x268		DDRC_IO_CONFIG		0x0000_0000														
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0										
Name	reserved					ae	ddr_mode	ck_ds	cmd_ds	dqs_ds	dq_ds	rtt	sstl_cmd_en	sstl_ecc_en	sstl_byte3_en	sstl_byte2_en	sstl_byte1_en	sstl_byte0_en
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description															
[31:14]	RO	reserved	保留。															
[13]	RW	ae	模拟信号 IO 输出使能。 0: 禁止; 1: 使能。 用于测试 DLL 的输出。															



[12]	RW	ddr_mode	控制 IO 的输出驱动模式。 0: DDR2 模式; 1: DDR1 模式。
[11]	RW	ck_ds	控制时钟 IO 的驱动模式。 在 DDR2 SDRAM 模式下: 0: SSTL_18 半驱动; 1: SSTL_18 全驱动。 在 DDR SDRAM 模式下: 0: SSTL_2class1; 1: SSTL_2class2。
[10]	RW	cmd_ds	控制命令和地址 IO 的驱动模式。 在 DDR2 SDRAM 模式下: 0: SSTL_18 半驱动; 1: SSTL_18 全驱动。 在 DDR SDRAM 模式下: 0: SSTL_2 class1; 1: SSTL_2 class2。
[9]	RW	dqs_ds	控制 DQ/DM IO 的驱动模式。 在 DDR2 SDRAM 模式下: 0: SSTL_18 半驱动; 1: SSTL_18 全驱动。 在 DDR SDRAM 模式下: 0: SSTL_2 class1; 1: SSTL_2 class2。
[8]	RW	dq_ds	控制 DQS IO 的 DS 驱动模式。 在 DDR2 SDRAM 模式下: 0: SSTL_18 半驱动; 1: SSTL_18 全驱动。 在 DDR SDRAM 模式下: 0: SSTL_2 class1; 1: SSTL_2 class2。
[7:6]	RW	rtt	IO ODT 电阻控制。 00: 禁止; 01: 150Ω; 10: 75Ω; 11: 50Ω。



[5]	RW	sstl_cmd_en	命令 IO 的 IDDQ 控制。 0: SSTL 输入电路使能; 1: SSTL 输入电路禁止; LVCMOS 输入模式使能。 ODT 自动禁止。
[4]	RW	sstl_ecc_en	数据 IO 的 IDDQ 控制。 0: SSTL 输入电路使能。 1: SSTL 输入电路禁止; LVCMOS 输入模式使能。 ODT 自动禁止。
[3]	RW	sstl_byte3_en	数据 IO 的 IDDQ 控制。 0: SSTL 输入电路使能。 1: SSTL 输入电路禁止; LVCMOS 输入模式使能。 ODT 自动禁止。
[2]	RW	sstl_byte2_en	数据 IO 的 IDDQ 控制。 0: SSTL 输入电路使能。 1: SSTL 输入电路禁止, 并且 LVCMOS 输入模式使能。 ODT 自动禁止。
[1]	RW	sstl_byte1_en	数据 IO 的 IDDQ 控制。 0: SSTL 输入电路使能。 1: SSTL 输入电路禁止; LVCMOS 输入模式使能。 ODT 自动禁止。
[0]	RW	sstl_byte0_en	数据 IO 的 IDDQ 控制。 0: SSTL 输入电路使能。 1: SSTL 输入电路禁止; LVCMOS 输入模式使能。 ODT 自动禁止。

4.2 SMI 控制器

4.2.1 概述

SMI (Static Memory Interface) 控制器对外提供异步静态存储器接口, 可以连接 SRAM (Static Random Access Memory)、PSRAM (Pseudo Static Access Memory)、ROM (Read Only Memory)、NOR Flash 等异步静态存储器, 用于实现系统启动、数据存储等功能。同时也可以用于连接带异步静态存储器接口的控制芯片, 实现主控功能。

4.2.2 特点

SMI 控制器具备以下特点:



- 支持异步静态存储器，包括 SRAM、PSRAM、ROM 和 NOR Flash 等。
- 支持异步 page 模式读操作。
- 支持控制器工作时钟可配置。
- 支持 2 个独立的存储器接口：0 通道和 1 通道。
- 支持从 0 通道外接存储器启动。
- 每个独立的存储器接口均可支持 8bit 数据位宽的外部存储器。
- 每个独立的存储器接口当外接 8bit 数据位宽存储器时，最大支持容量为 256Mbit。
- 支持读等待周期 (T_{WSTRD})、写等待周期 (T_{WSTWR}) 和 burst 读等待周期 ($T_{WSTBURSTRD}$) 可配置，最多可配置 31 个 SMI 工作时钟周期。
- 支持读使能等待周期 (T_{WSTOEN}) 和写使能等待周期 (T_{WSTWEN}) 可配置，最多可配为 15 个 SMI 工作时钟周期。
- 支持由读操作转变到写操作的等待周期 ($T_{WSTIDLY}$) 可配置，最多可配为 15 个 SMI 工作时钟周期。
- 支持异步等待信号输入，信号有效极性可配置。
- 支持时钟门控。

4.2.3 信号描述

SMI 控制器的接口信号如表 4-7 所示。

表4-7 SMI 控制器接口信号描述

信号名称	方向	描述	对应管脚
SMI_CS0	O	SMI 控制器输出的 0 通道片选信号，可配置为高电平有效或者低电平有效（默认低电平有效）。	SMICS0N
SMI_CS1	O	SMI 控制器输出的 1 通道片选信号，可配置为高电平有效或者低电平有效（默认低电平有效）。与 GPIO 复用（复用时的配置信息请参见“4.2.5 工作方式”中的“管脚复用配置”）。	SMICS1N
SMI_OEN	O	SMI 控制器输出的读使能信号，低电平有效。	SMIOEN
SMI_WEN	O	SMI 控制器输出的写使能信号，低电平有效。	EBIWEN
SMI_ADDR[24:0]	O	SMI 控制器输出的地址总线。其中 SMI_ADDR[24:15]与 BOOTSEL、FUNSEL、NF 管脚复用（复用时的配置信息请参见“4.2.5 工作方式”中的“管脚复用配置”）。	EBIADR24~ EBIADR0



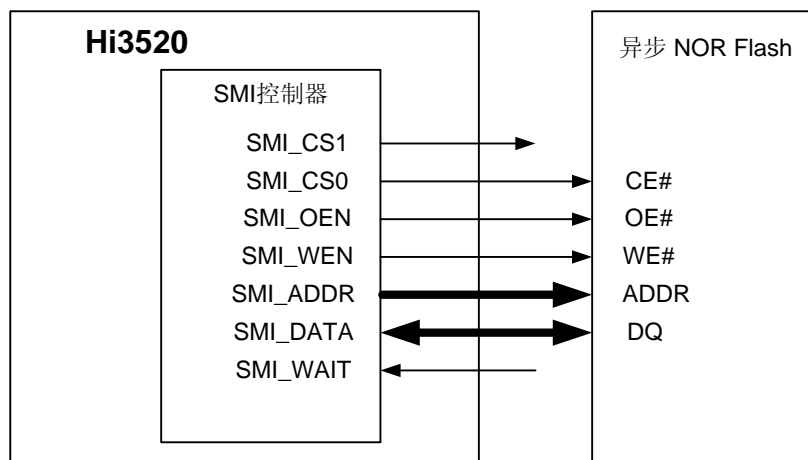
信号名称	方向	描述	对应管脚
SMI_DATA[7:0]	I/O	SMI 控制器双向数据总线。	EBIDQ7~ EBIDQ0
SMI_WAIT	I	外部输入 SMI 控制器的异步等待信号，可配置为高电平有效或者低电平有效（默认低电平有效）。与 IRRCV 管脚复用（复用时的配置信息请参见“4.2.5 工作方式”中的“管脚复用配置”）。	EBIRDYN

4.2.4 功能描述

应用框图

SMI 控制器用于控制内部系统总线与外部异步静态存储器总线之间数据交换，它提供了灵活的时序参数配置，适合连接各种异步静态存储器件，如图 4-3 所示。

图4-3 SMI 控制器与异步静态存储器连接示意图



连接说明：

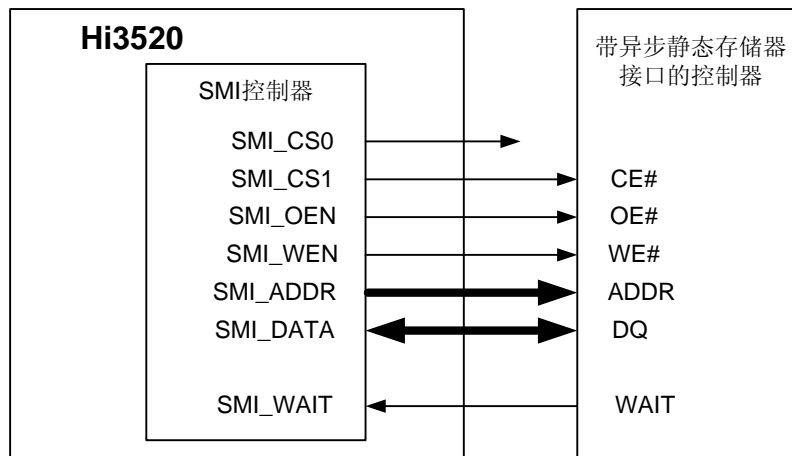
SMI 控制器有两个独立片选信号输出：SMI_CS0 和 SMI_CS1，其中支持从 SMI_CS0 外接 8bit 数据位宽存储器启动。

当 SMI 控制器外接异步静态存储器时，SMI_WAIT 信号未使用。

同时，SMI 控制器提供了异步等待机制，便于连接带有异步静态存储器接口的控制器，如图 4-4 所示。



图4-4 SMI 控制器与带异步静态存储器接口的控制器件连接示意图



连接说明：

SMI 控制器有两个独立片选信号输出：SMI_CS0 和 SMI_CS1，其中支持从 SMI_CS0 外接 8bit 数据位宽存储器启动。

当 SMI 控制器外接带异步静态存储器接口的控制器件时，SMI_WAIT/WAIT 信号用于双方握手协商。

功能原理

SMI 控制器主要根据对接器件的工作时序参数，实现接口时序的转换。SMI 支持通过更改 SMI 控制器工作时钟，来增强 SMI 控制器的兼容性，以尽可能兼容各种静态存储器件。在不同的工作时钟下，相应的可配置的时序参数范围也随之不同（如表 4-8 所示）。

表4-8 SMI 控制器时序参数范围（总线时钟 $f_{\text{BUSCLK}}=200\text{MHz}$ ）

时序参数	$f_{\text{SMICLK}}=200\text{MHz}$	$f_{\text{SMICLK}}=100\text{MHz}$
读等待周期 $T_{\text{WSTRD}}(\text{max})$	155ns	309ns
写等待周期 $T_{\text{WSTWR}}(\text{max})$	155ns	309ns
burst 读等待周期 $T_{\text{WSTBURSTRD}}(\text{max})$	155ns	309ns
读使能等待周期 $T_{\text{WSTOEN}}(\text{max})$	75ns	150ns
写使能等待周期 $T_{\text{WSTWEN}}(\text{max})$	75ns	150ns
读写转向周期 $T_{\text{WSTIDLY}}(\text{max})$	75ns	150ns

注：上表中所得时间为采用去尾法得到的计算结果。

**注意**

SMI 控制器只支持从通道 0 启动。

由于异步静态存储器接口的通用性，所以为了满足对接器件多样性的实际需求，SMI 控制器提供了两种工作模式：

- 时序参数模式
当 SMI 控制器提供的时序配置寄存器的配置范围可以满足对接器件时序要求时，SMI 控制器使用该模式来实现和对接器件的数据传送。SMI 控制器默认为工作在时序参数模式下。
- 异步等待模式
当 SMI 控制器提供的时序配置寄存器的配置范围不能满足对接器件时序要求，或者对接器件的时序不是一个明确的范围时，SMI 控制器使用该模式来实现和对接器件的数据传送。



说明
当使用 SMI 控制器的异步等待模式时，需要对接器件具有相应的异步等待控制信号。

时序参数模式下，典型的时序如图 4-5 所示。图 4-5 给出了在单次读写情况下，SMI 控制器接口上的时序关系。

图4-5 SMI 控制器时序参数模式时序图（读写）

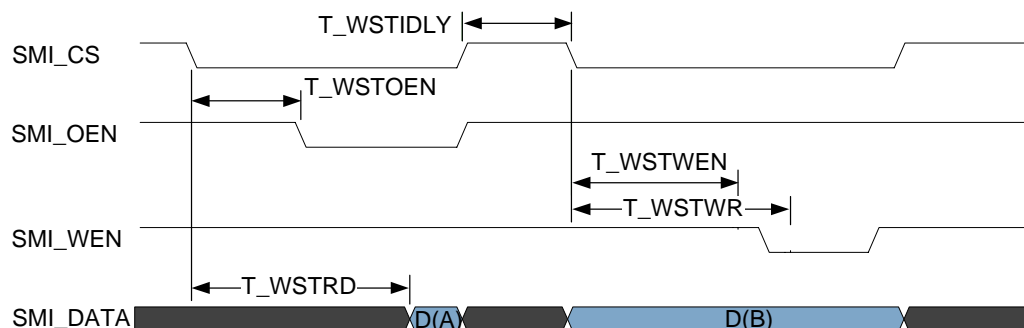


表 4-9 列出了 SMI 控制器读写时序参数。

表4-9 SMI 控制器读写时序参数表

符号	最小值	典型值	最大值	描述
T_WSTOEN	0	-	T _{WSTOEN(max)}	从 SMI 片选信号有效到读使能信号有效的的时间
T_WSTRD	-	-	T _{WSTRD(max)}	从 SMI 片选信号有效到读数据有效的的时间
T_WSTIDLY	0	-	T _{WSTIDLY(max)}	SMI 两次相邻读写操作间隔时间



符号	最小值	典型值	最大值	描述
T_WSTWEN	0	-	T _{WSTWEN(max)}	从 SMI 片选信号有效到写使能信号有效的时间
T_WSTWR	0	-	T _{WSTWR(max)}	从 SMI 片选信号有效到写数据有效的时间

图 4-6 给出了在 page 读情况下，SMI 控制器接口的时序关系。采用 page 读方式，可以大大提高读取数据速度。

图4-6 SMI 控制器时序参数模式时序图（page 读）

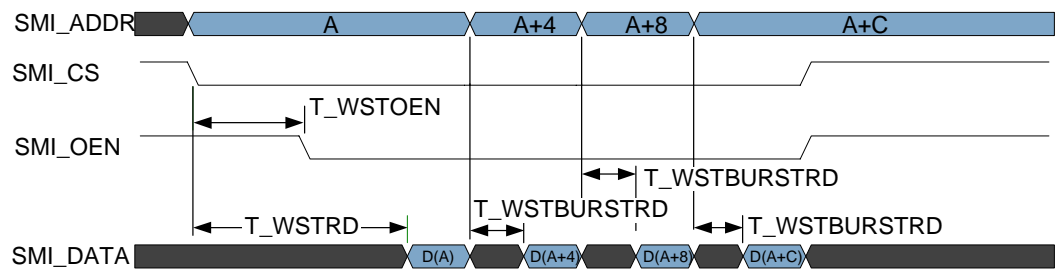


表 4-10 列出了 SMI 控制器 page 读时的时序参数。

表4-10 SMI 控制器 page 读时序参数表

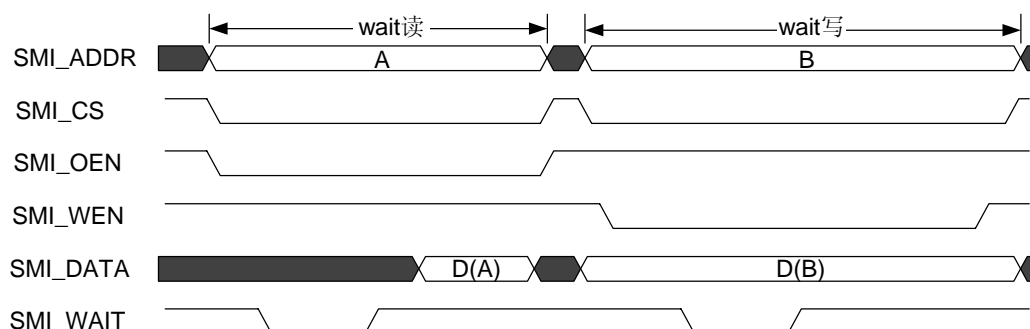
符号	最小值	典型值	最大值	描述
T_WSTOEN	0	-	T _{WSTOEN(max)}	从 SMI 片选信号有效到读使能信号有效的时间
T_WSTRD	-	-	T _{WSTRD(max)}	从 SMI 片选信号有效到读数据有效的时间
T_WSTBURSTRD	-	-	T _{WSTBURSTRD(max)}	SMI page 读过程中两次相邻读操作间隔时间

异步等待模式下时，典型的时序如图 4-7 所示。

图 4-7 给出了当 SMI 控制器外接其他控制芯片，通过 SMI_WAIT 信号进行握手协商时的读写时序关系。



图4-7 SMI 控制器异步等待模式时序图 (wait 读写)



4.2.5 工作方式

管脚复用配置

SMI 控制器管脚与其他管脚复用，使用 SMI 控制器之前，需要配置 IO Config 配置寄存器 reg71 和 reg74 配置说明。

时钟门控

在软件完成当前数据传输，并且未启动新的数据传输情况下，向系统控制器寄存器 SC_PERDIS[smiclkdir]写 1，可以关断 SMI 控制器时钟。

在需要使用 SMI 控制器进行数据传输时，向系统控制器寄存器 SC_PEREN[smiclken]写 1，可以使能 SMI 控制器时钟。

对当前 SMI 控制器时钟状态的查询请参见系统控制器寄存器 SC_PERCLKEN。

时钟配置



注意

寄存器 SMI_CR 中时钟比例的配置，必须与系统控制器 SC_PERCTRL9[ssmclk_sel]时钟比例的配置一致。

在软件完成当前数据传输，并且未启动新的数据传输情况下，可以通过控制 SC_PERCTRL9[ssmclk_sel]来配置 SMI 控制器的工作时钟，可配置为与总线时钟同频或者总线时钟的二分频。默认情况下，SMI 控制器的工作时钟等于总线时钟。

系统启动配置

系统只支持从 SMI 控制器 0 通道外接 8bit 数据位宽存储器启动。



注意

当系统采用从 SMI 控制器 0 通道外接存储器启动时，该外接存储器的片选信号必须为低电平有效。

SMI 控制器支持的存储器地址空间如表 4-11 所示。

表4-11 SMI 控制器支持的存储器地址空间

通道号	地址空间
0 通道	0x8000_0000~0x81FF_FFFF
1 通道	0x8400_0000~0x85FF_FFFF

时序参数模式下的 SMI 控制器初始化配置

需要根据 SMI 控制器外接存储器件的时序参数和特性，对 SMI 控制器进行正确配置，相关时序参数含义请参见“4.2.4 功能描述”中的“功能原理”。



注意

- 当 SMI 控制器外接异步静态存储器时，只需要正确配置 SMI 寄存器时钟比例 `SMI_CR[memclkratio]` 和外接存储器数据位宽 `SMI_BCR[mw]`，就可以对存储器进行正常数据传输。
- 在软件对每一个存储器通道进行配置时，必须满足： $T_{WSTOEN} \leq T_{WSTRD}$ 和 $T_{WSTWEN} \leq T_{WSTWR}$ 。

时序参数模式下，初始化 SMI 控制器步骤如下（以配置 1 通道为例）：

- 步骤 1 配置寄存器 `SMI_CR`，设置 SMI 控制器工作时钟比例。
- 步骤 2 配置寄存器 `SMI_BIDCYR1[idcy]`，设置读写转向周期 $T_{WSTIDLY}$ 。
- 步骤 3 配置寄存器 `SMI_BWSTRDR1[wstrd]`，设置读等待周期 T_{WSTRD} 。
- 步骤 4 配置寄存器 `SMI_BWSTWRR1[wstwr]`，设置写等待周期 T_{WSTWR} 。
- 步骤 5 配置寄存器 `SMI_BWSTOENR1[wstoen]`，设置读使能等待周期 T_{WSTOEN} 。
- 步骤 6 配置寄存器 `SMI_BWSTWENR1[wstwen]`，设置写使能等待周期 T_{WSTWEN} 。
- 步骤 7 配置寄存器 `SMI_BWSTBRDR1[wstbrdr]`，设置 burst 读等待周期 $T_{WSTBURSTRD}$ 。
- 步骤 8 配置寄存器 `SMI_BCR`，设置存储器通道控制参数。

----结束



按照以上步骤进行配置后，CPU 就可以通过配置好的 SMI 控制器访问外接存储器。

异步等待模式下的 SMI 控制器初始化配置

当 SMI 控制器外接带有等待信号输出的存储芯片或者控制芯片时，可以将 SMI 控制器切换到异步等待模式下，通过与外接芯片握手协商来进行数据传输。

异步等待模式下，初始化 SMI 控制器步骤如下（以配置 1 通道为例）：

- 步骤 1 配置 IO Config 寄存器 reg74，将 EBIRDYN 管脚设置为 SMI_WAIT 信号输入。
- 步骤 2 配置寄存器 SMI_CR，设置 SMI 控制器工作时钟比例。
- 步骤 3 配置寄存器 SMI_BCR，设置存储器通道控制参数。
- 步骤 4 根据外接芯片输出的“等待信号”有效极性，配置 SMI_BCR[waitpol]。
- 步骤 5 配置 SMI_BCR[waiten]为 1，使能 SMI 控制器异步等待模式。

----结束

按照以上步骤进行配置后，SMI 控制器即可工作在异步等待模式。

4.2.6 寄存器概览

表4-12 SMI 控制器寄存器概览（基址是 0x1001_0000）

偏移地址	名称	描述	页码
0x000	SMI_BIDCYR1	SMI 控制器 1 通道读写转向周期寄存器	4-37
0x004	SMI_BWSTRDR1	SMI 控制器 1 通道读等待周期寄存器	4-38
0x008	SMI_BWSTWRR1	SMI 控制器 1 通道写等待周期寄存器	4-38
0x00C	SMI_BWSTOENR1	SMI 控制器 1 通道读使能等待周期寄存器	4-39
0x010	SMI_BWSTWENR1	SMI 控制器 1 通道写使能等待周期寄存器	4-40
0x014	SMI_BCR1	SMI 控制器 1 通道控制寄存器	4-40
0x018	SMI_BSR1	SMI 控制器 1 通道状态寄存器	4-42
0x01C	SMI_BWSTBRDR1	SMI 控制器 1 通道突发读等待周期寄存器	4-42
0x020 ~ 0x0DC	RESERVED	保留	-
0x0E0	SMI_BIDCYR0	SMI 控制器 0 通道读写转向周期寄存器	4-43
0x0E4	SMI_BWSTRDR0	SMI 控制器 0 通道读等待周期寄存器	4-44



偏移地址	名称	描述	页码
0x0E8	SMI_BWSTWRR0	SMI 控制器 0 通道写等待周期寄存器	4-44
0x0EC	SMI_BWSTOENR0	SMI 控制器 0 通道读使能等待周期寄存器	4-45
0x0F0	SMI_BWSTWENR0	SMI 控制器 0 通道写使能等待周期寄存器	4-45
0x0F4	SMI_BCR0	SMI 控制器 0 通道控制寄存器	4-46
0x0F8	SMI_BSR0	SMI 控制器 0 通道状态寄存器	4-48
0x0FC	SMI_BWSTBRDR0	SMI 控制器 0 通道突发读等待周期寄存器	4-48
0x100 ~ 0x1FC	RESERVED	保留	-
0x200	SMI_SR	SMI 控制器异步等待状态寄存器	4-49
0x204	SMI_CR	SMI 控制器工作时钟配置寄存器	4-49
0x208 ~ 0xFFC	RESERVED	保留	-

4.2.7 寄存器描述

SMI_BIDCYR1

SMI_BIDCYR1 为 SMI 控制器 1 通道读写转向周期寄存器，用来控制在读写操作之间的等待时间。

	Offset Address	Register Name	Total Reset Value															
	0x000	SMI_BIDCYR1	0x0000_000F															
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																	
Name	reserved														idcy			
Reset	0 1 1 1 1																	
	Bits	Access	Name	Description														
	[31:4]	-	reserved	保留。														



[3:0]	RW	idcy	<p>$T_{WSTIDLY}$，读写转向周期，控制在读写操作之间的等待时间。</p> $T_{WSTIDLY} = idcy \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 idcy 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>
-------	----	------	---

SMI_BWSTRDR1

SMI_BWSTRDR1 为 SMI 控制器 1 通道读等待周期寄存器，用来控制读操作的等待时间。

	Offset Address				Register Name				Total Reset Value																							
	0x004				SMI_BWSTRDR1				0x0000_001F																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								wstrd							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Bits	Access		Name		Description																											
[31:5]	-		reserved		保留。																											
[4:0]	RW		wstrd		<p>T_{WSTRD}，控制读操作的等待时间。</p> <p>对 non burst 读操作而言，控制读操作的等待时间。</p> <p>对 burst 读操作而言，只控制第 1 次读操作的等待时间，后续读操作等待时间由 SMI_BWSTBRDR1 进行配置。</p> $T_{WSTRD} = wstrd \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstrd 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>																											

SMI_BWSTWRR1

SMI_BWSTWRR1 为 SMI 控制器 1 通道写等待周期寄存器，用来控制写操作的等待时间。



Offset Address		Register Name		Total Reset Value				
0x008		SMI_BWSTWRR1		0x0000_001F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved							wstwr
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	1 1 1 1
Bits	Access	Name	Description					
[31:5]	-	reserved	保留					
[4:0]	RW	wstwr	T_{WSTWR} ，控制写操作的等待时间。 $T_{WSTWR} = wstwr \times \left(\frac{1}{f_{SMICLK}} \right)$ 其中 wstwr 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICLK} 表示 SMI 控制器工作时钟频率。					

SMI_BWSTOENR1

SMI_BWSTOENR1 为 SMI 控制器 1 通道读使能等待周期寄存器，用来控制从输出有效片选信号开始到输出有效读使能信号之间的等待时间。

Offset Address		Register Name		Total Reset Value				
0x00C		SMI_BWSTOENR1		0x0000_000F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved							wstoen
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
Bits	Access	Name	Description					
[31:4]	-	reserved	保留。					
[3:0]	RW	wstoen	T_{WSTOEN} ，控制从输出有效片选信号开始到输出有效读使能信号之间的等待时间。 $T_{WSTOEN} = wstoen \times \left(\frac{1}{f_{SMICLK}} \right)$ 其中 wstoen 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICLK} 表示 SMI 控制器工作时钟频率。					

SMI_BWSTWENR1

SMI_BWSTWENR1 为 SMI 控制器 1 通道写使能等待周期寄存器，用来控制从输出有效片选信号开始到输出有效写使能信号之间的等待时间。

Offset Address		Register Name		Total Reset Value				
0x010		SMI_BWSTWENR1		0x0000_000F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved							wstwen
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
Bits	Access	Name	Description					
[31:4]	-	reserved	保留。					
[3:0]	RW	wstwen	<p>T_{WSTWEN}，控制从输出有效片选信号开始到输出有效写使能信号之间的等待时间。</p> $T_{WSTWEN} = wstwen \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstwen 表示系统配置 SMI 控制器工作时钟周期个数；f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>					

SMI_BCR1

SMI_BCR1 为 SMI 控制器 1 通道控制寄存器，用来配置 1 通道的各种传输参数。

Offset Address		Register Name		Total Reset Value															
0x014		SMI_BCR1		0x0030_3020															
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0											
Name	reserved			reserved	reserved	burstlenwrite	reserved	bmwrite	reserved	burstlenread	reserved	bmread	reserved	reserved	mw	wp	waiten	waitpol	reserved
Reset	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 0	0 0 0 0											
Bits	Access	Name	Description																
[31:22]	-	reserved	保留。																
[21]	RW	reserved	保留，此位只能设为 1。																
[20]	RW	reserved	保留，此位只能设为 1。																



[19:18]	RW	burstlenwrite	burst 传输长度，设定 burst 模式下一次 burst 写操作过程中传输数据的个数。 00: burst 4。 01: burst 8。 10、11: 保留。
[17]	RW	reserved	保留，此位只能设为 0。
[16]	RW	bmwrite	写模式配置位。 0: 对外接器件进行 non burst 模式写。 1: 对外接器件进行 burst 模式写。
[15]	-	reserved	保留。
[14]	RW	reserved	保留，此位只能设为 0。
[13]	RW	reserved	保留，此位只能设为 1。
[12]	RW	reserved	保留，此位只能设为 1。
[11:10]	RW	burstlenread	突发传输长度，设定 burst 模式下一次 burst 读操作过程中传输数据的个数。 00: burst 4。 01: burst 8。 10: burst 16。 11: 保留。
[9]	RW	reserved	保留，此位只能设为 0。
[8]	RW	bmread	读模式配置位。 0: 对外接器件进行 non burst 模式读。 1: 对外接器件进行 burst 模式读。
[7]	-	reserved	保留。
[6]	RW	reserved	保留，此位只能设为 0。
[5:4]	RW	mw	SMI 控制器第 1 通道外接器件数据位宽。 00: 8bit。 其他: 保留。
[3]	RW	wp	外接器件写保护位，控制是否对外接器件进行写保护。 0: 外接器件未被设为写保护，例如 SRAM 等可写器件。 1: 外接器件为写保护状态，例如 ROM 等只读器件。



[2]	RW	waiten	SMI 控制器 1 通道“等待信号”使能位。 0: 禁止, 此时 SMI 控制器第 1 通道不受外部输入的“等待信号”控制, 根据内部时序寄存器的配置参数进行数据传输。 1: 使能, 此时 SMI 控制器第 1 通道根据外部输入的“等待信号”进行数据传输。
[1]	RW	waitpol	选择外部输入的“等待信号”有效极性。 0: 低电平有效。 1: 高电平有效。
[0]	RW	reserved	保留, 此位只能设为 0。

SMI_BSR1

SMI_BSR1 为 SMI 控制器 1 通道状态寄存器, 用来显示 1 通道当前状态。

Offset Address		Register Name		Total Reset Value					
0x018		SMI_BSR1		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								waitouterr
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	waitouterr	外部等待超时错误标志位。 读操作时: 0: 没有错误。 1: 外部等待超时错误。 写操作时: 0: 无效。 1: 清除外部等待超时错误状态标志。						

SMI_BWSTBRDR1

SMI_BWSTBRDR1 为 SMI 控制器 1 通道 burst 读等待周期寄存器, 用来控制 burst 读操作过程中非第 1 次读操作的等待时间。



Offset Address		Register Name		Total Reset Value					
0x01C		SMI_BWSTBRDR1		0x0000_001F					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved							wstbrd	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	1 1 1 1	
Bits	Access	Name	Description						
[31:5]	-	reserved	保留。						
[4:0]	RW	wstbrd	<p>$T_{WSTBURSTRD}$, burst 读等待周期。</p> <p>对于 non burst 读操作, 不使用该时序参数。</p> <p>对 burst 读操作, 控制 burst 读操作过程中非第 1 次读操作的等待时间。</p> <p>burst 读操作过程中第 1 次读操作的等待时间由 SMI_BWSTRDR1 进行配置。</p> $T_{WSTBURSTRD} = wstbrd \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstbrd 表示系统配置 SMI 控制器工作时钟周期个数; f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>						

SMI_BIDCYR0

SMI_BIDCYR0 为 SMI 控制器 0 通道读写转向周期寄存器, 用来控制在读写操作之间的等待时间。

Offset Address		Register Name		Total Reset Value					
0x0E0		SMI_BIDCYR0		0x0000_000F					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved							idcy	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	
Bits	Access	Name	Description						
[31:4]	-	reserved	保留。						
[3:0]	RW	idcy	<p>$T_{WSTIDLY}$, 读写转向周期, 控制在读写操作之间的等待时间。</p> $T_{WSTIDLY} = idcy \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 idcy 表示系统配置 SMI 控制器工作时钟周期个数; f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>						



SMI_BWSTRDR0

SMI_BWSTRDR0 为 SMI 控制器 0 通道读等待周期寄存器，用来控制读操作的等待时间。

	Offset Address	Register Name	Total Reset Value													
	0x0E4	SMI_BWSTRDR0	0x0000_001F													
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
Name	reserved												wstrd			
Reset	0 1 1 1 1 1															
Bits	Access	Name	Description													
[31:5]	-	reserved	保留。													
[4:0]	RW	wstrd	<p>T_{WSTRD}，控制读操作的等待时间。</p> <p>对 non burst 读操作而言，控制读操作的等待时间。</p> <p>对 burst 读操作而言，只控制第 1 次读操作的等待时间，后续读操作等待时间由 SMI_BWSTBRDR0 进行配置。</p> $T_{WSTRD} = wstrd \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstrd 表示系统配置 SMI 控制器工作时钟周期个数；f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>													

SMI_BWSTWRR0

SMI_BWSTWRR0 为 SMI 控制器 0 通道写等待周期寄存器，用来控制写操作的等待时间。

	Offset Address	Register Name	Total Reset Value													
	0x0E8	SMI_BWSTWRR0	0x0000_001F													
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
Name	reserved												wstwr			
Reset	0 1 1 1 1 1															
Bits	Access	Name	Description													
[31:5]	-	reserved	保留。													



[4:0]	RW	wstwr	<p>T_{WSTWR}，控制写操作的等待时间。</p> $T_{WSTWR} = wstwr \times \left(\frac{1}{f_{SMICK}} \right)$ <p>其中 wstwr 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICK} 表示 SMI 控制器工作时钟频率。</p>
-------	----	-------	--

SMI_BWSTOENR0

SMI_BWSTOENR0 为 SMI 控制器 0 通道读使能等待周期寄存器，用来控制从输出有效片选信号开始到输出有效读使能信号之间的等待时间。

Offset Address		Register Name		Total Reset Value				
0x0EC		SMI_BWSTOENR0		0x0000_000F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved							wstoen
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
Bits	Access	Name	Description					
[31:4]	-	reserved	保留。					
[3:0]	RW	wstoen	<p>T_{WSTOEN}，控制从输出有效片选信号开始到输出有效读使能信号之间的等待时间。</p> $T_{WSTOEN} = wstoen \times \left(\frac{1}{f_{SMICK}} \right)$ <p>其中 wstoen 表示系统配置 SMI 控制器工作时钟周期个数； f_{SMICK} 表示 SMI 控制器工作时钟频率。</p>					

SMI_BWSTWENR0

SMI_BWSTWENR0 为 SMI 控制器 0 通道写使能等待周期寄存器，用来控制从输出有效片选信号开始到输出有效写使能信号之间的等待时间。

Offset Address		Register Name		Total Reset Value				
0x0F0		SMI_BWSTWENR0		0x0000_000F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved							wstwen
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1
Bits	Access	Name	Description					
[31:4]	-	reserved	保留。					
[3:0]	RW	wstwen	<p>T_{WSTWEN}，控制从输出有效片选信号开始到输出有效写使能信号之间的等待时间。</p> $T_{WSTWEN} = wstwen \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstwen 表示系统配置 SMI 控制器工作时钟周期个数；f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>					

SMI_BCR0

SMI_BCR0 为 SMI 控制器 0 通道控制寄存器，用来配置 0 通道的各种传输参数。

Offset Address		Register Name		Total Reset Value													
0x0F4		SMI_BCR0		0x0030_3000													
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0									
Name	reserved		reserved	reserved	burstlenwrite	reserved	reserved	burstlenread	reserved	reserved	reserved	reserved	mw	wp	waiten	waitpol	reserved
Reset	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description														
[31:22]	-	reserved	保留。														
[21]	RW	reserved	保留，此位只能设为 1。														
[20]	RW	reserved	保留，此位只能设为 1。														
[19:18]	RW	burstlenwrite	<p>burst 传输长度，设定 burst 模式下（请参见图 4-6）一次 burst 写操作过程中传输数据的个数。</p> <p>00: burst 4。 01: burst 8。 10、11: 保留。</p>														



[17]	RW	reserved	保留，此位只能设为 0。
[16]	RW	bmwrite	写模式配置位。 0: 对外接器件进行 non burst 模式写。 1: 对外接器件进行 burst 模式写。
[15]	-	reserved	保留。
[14]	RW	reserved	保留，此位只能设为 0。
[13]	RW	reserved	保留，此位只能设为 1。
[12]	RW	reserved	保留，此位只能设为 1。
[11:10]	RW	burstlenread	burst 传输长度，设定 burst 模式下一次 burst 读操作过程中传输数据的个数。 00: burst 4。 01: burst 8。 10: burst 16。 11: 保留。
[9]	RW	reserved	保留，此位只能设为 0。
[8]	RW	bmread	读模式配置位。 0: 对外接器件进行 non burst 模式读。 1: 对外接器件进行 burst 模式读。
[7]	-	reserved	保留。
[6]	RW	reserved	保留，此位只能设为 0。
[5:4]	RW	mw	SMI 控制器第 0 通道外接器件数据位宽。 00: 8bit。 其他: 保留。
[3]	RW	wp	外接器件写保护位，控制是否对外接器件进行写保护。 0: 外接器件未被设为写保护，例如 SRAM 等可写器件。 1: 外接器件为写保护状态，例如 ROM 等只读器件。
[2]	RW	waiten	SMI 控制器第 0 通道“等待信号”使能位。 0: 禁止，此时 SMI 控制器第 0 通道不受外部输入的“等待信号”控制，根据内部时序寄存器的配置参数进行数据传输。 1: 使能，此时 SMI 控制器第 0 通道根据外部输入的“等待信号”进行数据传输。
[1]	RW	waitpol	选择外部输入的“等待信号”有效极性。 0: 外部输入的“等待信号”低电平有效。 1: 外部输入的“等待信号”高电平有效。



[0]	RW	reserved	保留，此位只能设为 0。
-----	----	----------	--------------

SMI_BSR0

SMI_BSR0 为 SMI 控制器 0 通道状态寄存器，用来显示 0 通道当前状态。

	Offset Address	Register Name	Total Reset Value
	0x0F8	SMI_BSR0	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		waittouterr
Reset	0 0		
	Bits	Access	Name
	[31:1]	-	reserved
	[0]	RW	waittouterr
	Description		
	保留。		
	外部等待超时错误标志位。 读操作时： 0：没有错误。 1：外部等待超时错误。 写操作时： 0：无效。 1：清除外部等待超时错误状态标志。		

SMI_BWSTBRDR0

SMI_BWSTBRDR0 为 SMI 控制器 0 通道 burst 读等待周期寄存器，用来控制 burst 读操作过程中非第一次读操作的等待时间。

	Offset Address	Register Name	Total Reset Value
	0x0FC	SMI_BWSTBRDR0	0x0000_001F
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		wstbrd
Reset	0 1 1 1 1 1		
	Bits	Access	Name
	[31:5]	-	reserved
	Description		
	保留。		



[4:0]	RW	wstbrd	<p>$T_{WSTBURSTRD}$, burst 读等待周期。</p> <p>对于 non burst 读操作, 不使用该时序参数。</p> <p>对 burst 读操作, 控制 burst 读操作过程中非第 1 次读操作的等待时间。</p> <p>burst 读操作过程中第 1 次读操作的等待时间由 SMI_BWSTRDR0 进行配置。</p> $T_{WSTBURSTRD} = wstbrd \times \left(\frac{1}{f_{SMICLK}} \right)$ <p>其中 wstbrd 表示系统配置 SMI 控制器工作时钟周期个数; f_{SMICLK} 表示 SMI 控制器工作时钟频率。</p>
-------	----	--------	---

SMI_SR

SMI_SR 为 SMI 控制器异步等待状态寄存器, 用来表示异步等待模式下的异步等待状态。

	Offset Address				Register Name								Total Reset Value																			
	0x200				SMI_SR								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											waitstatus				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:1]	-	reserved		保留。																											
	[0]	RO	waitstatus		SMI 控制器异步等待状态位。 0: “等待信号” 无效。 1: “等待信号” 有效。																											

注: 在异步等待模式下, 该状态寄存器才表示异步等待状态。

SMI_CR

SMI_CR 为 SMI 控制器工作时钟配置寄存器, 用来配置 SMI 控制器工作时钟与系统总线时钟的关系。



	Offset Address 0x204				Register Name SMI_CR				Total Reset Value 0x0000_0001																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								memclk_ratio		reserved					
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 1							
	Bits	Access	Name		Description																											
	[31:3]	-	reserved		保留。																											
	[2:1]	RW	memclk_ratio		配置 SMI 控制器工作时钟。 00: SMI 控制器工作时钟等于系统总线时钟。 01: SMI 控制器工作时钟等于系统总线时钟的一半。 10、11: 保留。																											
	[0]	RW	reserved		保留, 此位只能设为 1。																											

4.3 NAND Flash 控制器

4.3.1 概述

NANDC (NAND Flash Controller) 提供存储控制器接口连接片外 NAND Flash, 从而完成数据的存取。

4.3.2 特点

NANDC 的主要特点如下:

- 提供 2KB (2048byte+128byte) 片内缓存, 提高读取速度。
- 支持 2 个片选信号和 1 个 ready/busy 信号。
- 支持 8bit data-bus NAND Flash 接口。
- 支持 NAND Boot 功能, 支持 2KB、4KB page size 的 NAND Flash 器件, 支持从片选 0 对应的 NAND Flash 启动。
- 支持 ECC 校验使能和关闭, 支持纠错使能和关闭。
 - 对于 SLC 器件, 支持 512byte Hamming 码 ECC (Error Correcting Code) 校验和 1bit 纠错。
 - 对于 MLC 器件, 支持 512byte 校验和 4/8byte 纠错。
- 支持读写、擦除、编程完成、ECC 校验错误等中断上报。
- 支持不定长数据读写。



- 可以灵活配置控制器发出的命令，从而可以支持各种 NAND Flash 命令操作（包括 cache 读写等）。
- 支持读写 NAND Flash 过程可以被打断，以支持存储器共享总线出让。
- 支持交替操作 2 个 NAND Flash 器件，并行工作以提高效率。
- 支持对 NAND Flash 写保护功能，写保护地址空间可配置。
- 支持 lock 和 lock-down 两种模式。支持 Flashlock 使能/去使能、Flash 整体锁定使能/去使能，默认为 Flashlock 使能，Flash 整体锁定使能。凡是对锁定地址范围内的写操作，NANDC 均上报操作错误中断。
- 支持 EDO（Enhanced Data Out）增强数据输出模式的 NAND Flash 数据读取。

4.3.3 信号描述

NANDC 接口信号描述如表 4-13 所示。

表4-13 NANDC 接口信号描述

信号	I/O	描述	对应管脚
NF_RB	I	NAND Flash 空闲/忙指示信号。 1: 空闲; 0: 忙。	NF_RB
NF_CSN0	O	第 1 个 NAND Flash 片选信号, 低电平有效。	NF_CSN0
NF_CSN1	O	第 2 个 NAND Flash 片选信号, 低电平有效。	NF_CSN1
NF_REN	O	NAND Flash 读使能信号, 低电平有效。	NF_REN
NF_WEN	O	NAND Flash 写使能信号, 低电平有效。	NF_WEN
NF_CLE	O	NAND Flash 命令锁存信号。	NF_CLE
NF_ALE	O	NAND Flash 地址锁存信号。	NF_ALE
NF_ADNUM[1:0]	I	Boot 时, 选择 NANDC 发给 NAND Flash 器件的地址数目。 00: 3 个地址周期; 01: 4 个地址周期; 10: 5 个地址周期; 11: 6 个地址周期。	EBIADR18/EBIADR17



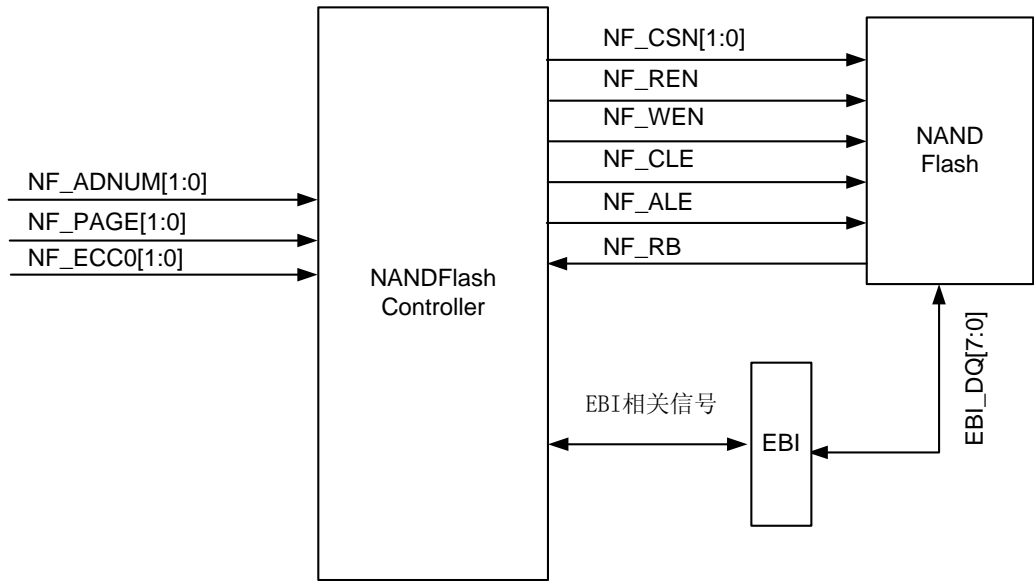
信号	I/O	描述	对应管脚
NF_PAGE[1:0]	I	Boot 时，选择 NAND Flash 器件的 page 容量。 00: 保留; 01: 2KB; 10: 4KB; 11: 保留。	EBIADR21/EBIADR20
NF_ECC0[1:0]	I	Boot 时，选择 ECC 模式。 00: 无 ECC 模式; 01: 1bit 模式; 10: 4bit 模式; 11: 8bit 模式。 默认启用 1bit 纠错的模式	EBIADR16/EBIADR15
EBI_DQ[7:0]	IO	NAND Flash 接口数据地址总线，访问 8bits NAND Flash 时，只有低 8bits 有效。	EBI_DQ0~EBI_DQ7

4.3.4 功能描述

4.3.4.1 接口框图

芯片对外提供 2 个片选和 1 个 ready/busy 信号，可以方便地与 NAND Flash 器件对接。当有两个 ready_busy 信号时，采用线与连接到 NF_RB。

图4-8 NAND Flash Controller 接口框图



4.3.4.2 功能原理

NAND Flash 器件的数据存储结构一般分为 block 和 page，每个 block 包括若干个 page。对 NAND Flash 写入数据前，必须先进行擦除操作，擦除以 block 为单位。然后以 page 为单位进行读写。

不同厂家提供的操作 NAND Flash 的命令会有所不同，应以厂家器件手册为准。

一次典型的读数据操作过程如下：

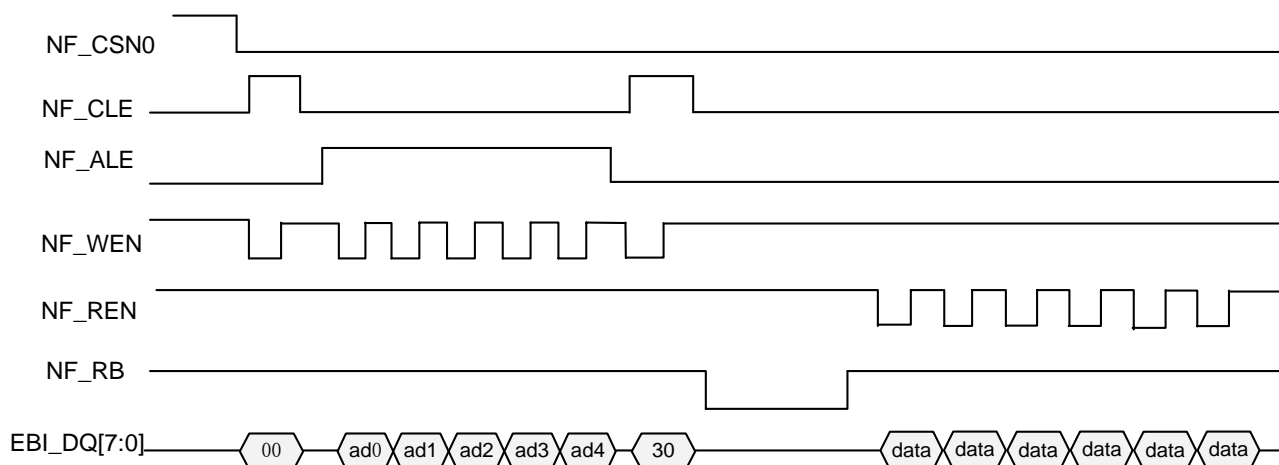
- 步骤 1 向 NAND Flash 发读命令 0x00。
- 步骤 2 发送读取的起始地址（由页内地址和页地址、block 地址共同组成，相关信息，请参见相关厂家 NAND Flash 器件手册）。
- 步骤 3 发送读确认命令 0x30。此时，NAND Flash 器件会拉低 RB 信号，表示 NAND Flash 正在进行内部读操作。过一段时间后（一般约 25 μ s），RB 变高，此时代表 NAND Flash 已经准备好数据。
- 步骤 4 CPU 通过使能 NF_REN 信号，把数据从 NAND Flash 读出来。

----结束

NANDC 读 NAND Flash 一个 page 数据的典型时序如图 4-9 所示。



图4-9 读 NAND Flash 一个 page 数据的典型时序图



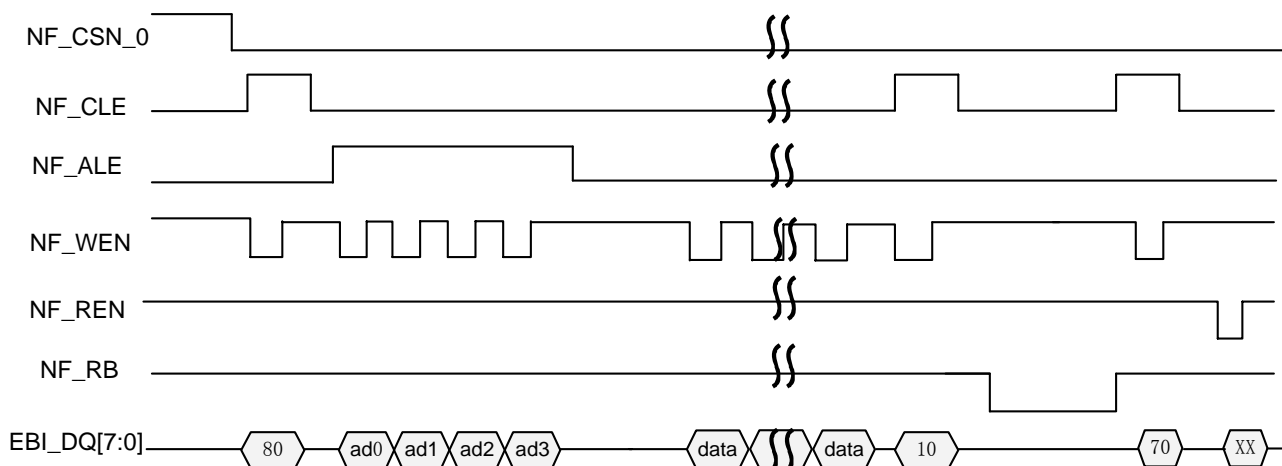
一次典型的编程（写数据）的操作如下：

- 步骤 1 向 NAND Flash 发编程命令 0x80。
- 步骤 2 发送写入数据的起始地址（由页内地址和页地址、block 地址共同组成，需参照对应的 NAND Flash 器件手册）。
- 步骤 3 把数据写入 NAND Flash 的内部缓存。
- 步骤 4 CPU 再发送编程确认命令 0x10。此时，NAND Flash 器件会拉低 RB 信号，表示 NAND Flash 正在进行内部编程操作，维持一段时间后（一般约 200ms），RB 变高，此时，代表 NAND Flash 内部编程已经结束。
- 步骤 5 CPU 通过发送 0x70 读状态，读出本次编程是否成功的状态数据。

----结束

启动 NANDC 进行编程操作时的时序如图 4-10 所示。

图4-10 启动 NANDC 进行编程操作时的时序图





4.3.5 工作方式

管脚复用配置

NF_RB 与 GPIO3_1 进行了复用，当外接器件有两个 ready_busy 信号时，采用线与连接到 NF_RB。

NFCSN1 与 GPIO3_3 进行了复用，当外接器件有 2 个片选信号时，要对 IO Config 配置寄存器 reg72 进行设置，选 NF_CSN1。

时钟门控

当不使用 NAND Flash 时，可以关断 NANDC 的工作时钟，步骤如下：

- 步骤 1 读 NANDC 的 `NFC_STATUS[nfc_ready]`。
- 步骤 2 如果 `NFC_STATUS[nfc_ready]` 为 1，软件确认不再读写 NAND Flash，进入步骤 3；否则，返回步骤 1。
- 步骤 3 向系统寄存器 `SC_PERDIS[nandcclkdis]` 写 1，关闭时钟。

----结束

软复位

写 `NFC_OP` 寄存器启动 NANDC 执行操作后，如果 `NFC_STATUS[nfc_ready]` 变为 0 后，长时间不能变为 1（最大时间长度取决于 NAND Flash 器件，SLC Flash 器件一般小于 4ms、MLC 器件一般小于 11ms），说明 NANDC 出现异常，需要软复位。

通过向 `SC_PERCTRL10[nadc_srst]` 写 1，可实现对 NANDC 的软复位。复位后各配置寄存器的值均复位为默认值，因此复位后需要重新对这些寄存器进行初始化配置。软复位后，为了保证 NAND Flash 的可靠工作，需要对其发出复位命令（如果 NAND Flash 支持复位操作）。

 说明

NANDC 有一个锁定功能。当设置该功能有效后，只有硬件复位才能够取消该功能。

Boot 配置管脚

NANDC 支持 NAND Boot 功能，支持 2KB、4KB page size 的器件，只支持从片选 0 对应的 NAND Flash 启动。

复位后 NANDC Boot 配置管脚的电平决定 Boot 模式。NANDC 在复位结束后，采样一次 Boot 配置管脚的电平值。之后，这些管脚的电平值不再影响 NANDC 的工作状态。Boot 相关的配置管脚如表 4-14 所示。



表4-14 Boot 相关的配置管脚

名称	I/O	描述
NF_NUM[1:0]	I	Boot 时，NANDC 发给 NAND Flash 器件的地址数目。 00: 3 个地址周期; 01: 4 个地址周期; 10: 5 个地址周期; 11: 6 个地址周期。
NF_PAGE[1:0]	I	Boot 时，NAND Flash 器件的 page 容量。 00: 保留; 01: 2KB; 10: 4KB; 11: 保留。
NF_ECC[1:0]	I	Boot 时，选择 ECC 模式。 00: 无 ECC 模式; 01: 1bit 模式; 10: 4bit 模式; 11: 8bit 模式。

Boot 模式

NANDC 默认处于 Boot 模式，只能从片选 0 对应的 NAND Flash 进行 Boot。在 Boot 模式下，主要有以下特点

- CPU 可以直接读取 0x00_0000~0x01_FFFF 地址范围的数据，整个空间大小为 128KB。
- 从 NAND Flash Boot 时，根据 CPU 读取的地址，NANDC 会自动发出读取 NAND Flash 相应 page 的命令，并返回相应的数据。
- 不支持 CPU 写内部 buffer 的操作。
- 需要根据外接 NAND Flash 器件的型号特点，对 Boot 配置管脚信号进行正确的设置。

NORMAL 模式

把寄存器 `NFC_CON[op_mode]` 置 1，切换到 Normal 模式。该模式下，CPU 可以对 NAND Flash 进行擦除、编程、读等各种操作。

NAND Flash 地址设置

NANDC 对地址不作翻译，直接把低位地址寄存器和高位地址寄存器的值按照命令配置寄存器设置的地址数目发送给 NAND Flash 器件。所以软件应把 CPU 的地址翻译为



NAND Flash 的地址，写到地址寄存器。对于各个 Flash 器件的地址设置要求，以 NAND Flash 器件厂商的用户手册为准。

三星 K9F2G08U0M 容量为 256M×8bit，page size 为 2KB 的器件的地址设置要求如表 4-15 所示。A0~A11 为页内地址（列地址），A12~A27 为页地址（行地址）。

表4-15 K9F2G08U0M 的地址表

周期	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
1st cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd cycle	A8	A9	A10	A11	0	0	0	0
3rd cycle	A12	A13	A14	A15	A16	A17	A18	A19
4th cycle	A20	A21	A22	A23	A24	A25	A26	A27

三星 K9GAG08X0M，容量为 2G×8bit，page size 为 4KB 的器件的地址设置要求如表 4-16 所示。A0~A12 为页内地址（列地址），A13~A31 为页地址（行地址）。

表4-16 K9GAG08X0M 的地址表

周期	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
1st cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd cycle	A8	A9	A10	A11	A12	0	0	0
3rd cycle	A13	A14	A15	A16	A17	A18	A19	A20
4th cycle	A21	A22	A23	A24	A25	A26	A27	A28
5th cycle	A29	A30	A31	0	0	0	0	0

地址映射

在 Normal 模式下，NANDC 地址映射如下：

- NANDC 的内部 buffer 的基地址为 0x7000_0000。
- NANDC 的内部寄存器区的基地址为 0x1000_0000。

操作命令

NAND Flash 厂家的器件会提供一些高级命令，NAND Flash 操作的一些基本的命令如表 4-17 所示。



表4-17 NAND Flash 常用命令表

功能	1 st cycle	2 nd cycle	备注
READ	00H	30H	-
PROGRAM	80H	10H	-
BLOCK ERASE	60H	D0H	-
READ ID	90H	-	-
READ STATUS	70H	-	-
RESET	FFH	-	-

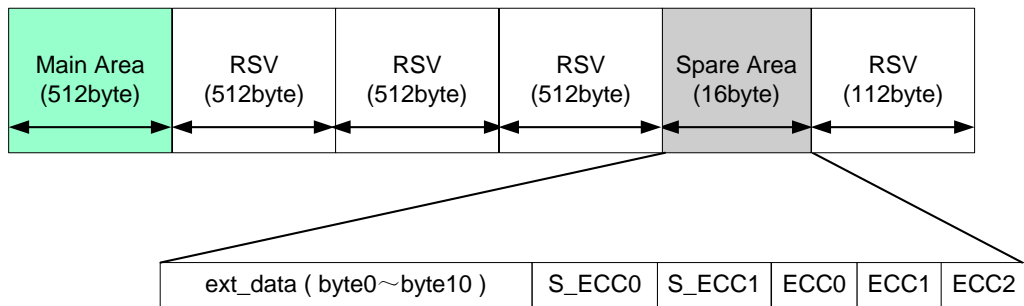
数据存储结构

NANDC 内部 buffer 大小为 (2048+128) byte。读写 NAND Flash 数据在 NANDC buffer 中的数据结构如下：

- 1bit ECC 模式
 - (512+16) byte page size
512byte 有效数据存放在 buffer 的 0x000~0x1FF 地址，Spare area 数据存放地址 0x7FF~0x83F。如图 4-11 所示。
 - 2KB (2048+64) page size
2048byte 有效数据存放在 buffer 的 0x000~0x7FF 地址，64byte spare 数据存放 0x800~0x80F 地址。
byte0~byte511 的 Main Area 区对应地址 0x800~0x80F 的 Spare Area 区。依此类推。
Spare 区的数据结构与 (512+16) byte page size 的数据结构相同。如图 4-12 所示。
当 ECC 使能打开时，写到 NAND Flash 器件内部的数据结构会自动调整为 Main Area 区和对应 Spare Area 区紧邻存放的格式。如图 4-13 所示。
 - 4KB (4096+128) page size
对 4KB page size 的器件操作需要分两次进行，每次读写的数据结构与图 4-12 相同。



图4-11 1bit ECC 模式 (512+16) byte page size NAND Flash 数据存储结构



ECC0、ECC1、ECC2: ECC code for main area data;
S_ECC0、S_ECC1: ECC code for RSV (byte0~byte10) data;
ext_data: 扩展数据区。

图4-12 2KB (2048+64) byte page size NAND Flash 数据存储结构

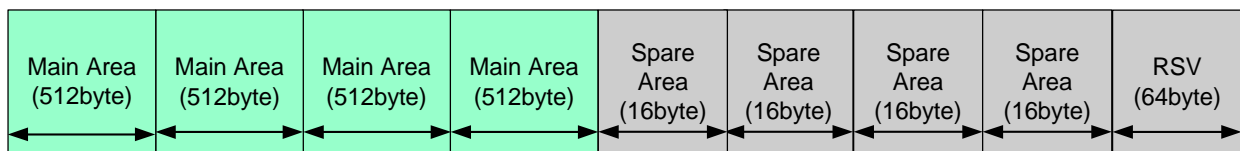
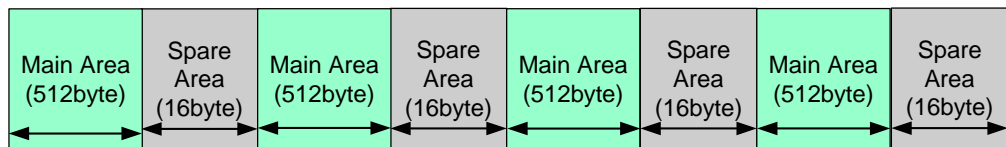


图4-13 自动调整为 Main Area 区后的数据结构



注：在 boot 模式下，当 1bit ECC 功能打开时，要求 NAND Flash 中数据按图 4-12 所示的数据结构存储。

- 4bit ECC 模式

- 2KB (2048+64) page size

2048byte 有效数据存放在 buffer 的 0x000~0x7FF 地址，64byte spare 数据存放 0x800~0x83F 地址。如图 4-14 所示。

byte0~byte511 的 Main Area 区对应 byte2048~byte2063 的 Spare Area 区。依此类推。

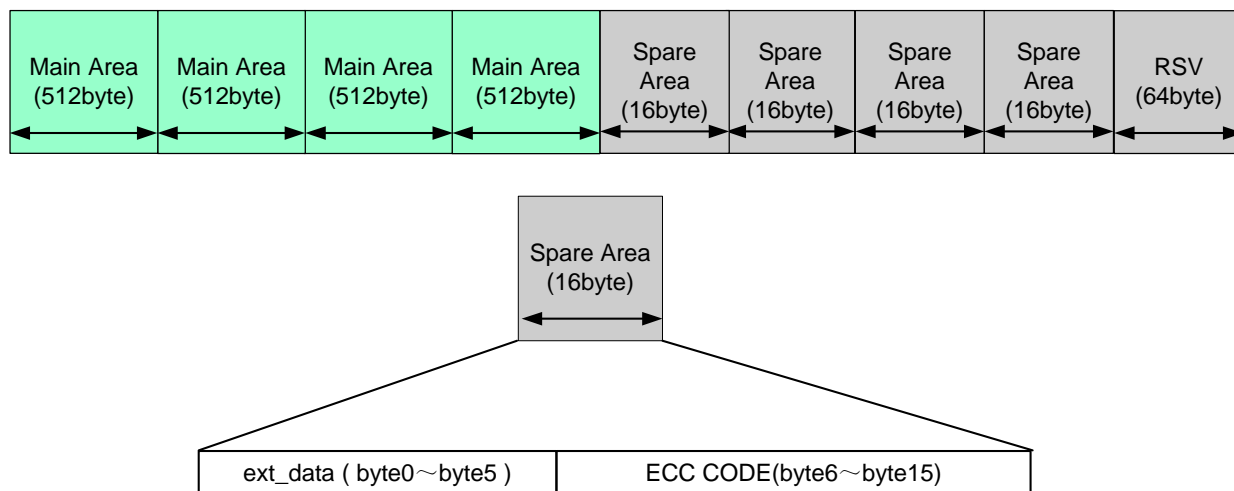
当 ECC 使能打开时，写到 NAND Flash 器件内部的数据结构会自动调整为 Main Area 区和对应 Spare Area 区紧邻存放的格式。如图 4-13 所示。

- 4KB(4096+128) page size

对 4KB page size 的器件操作需要分两次进行，每次读写的数据结构与图 4-14 数据结构相同。



图4-14 4bit ECC 模式 2KB (2048+64) byte page size NAND Flash 数据存储结构



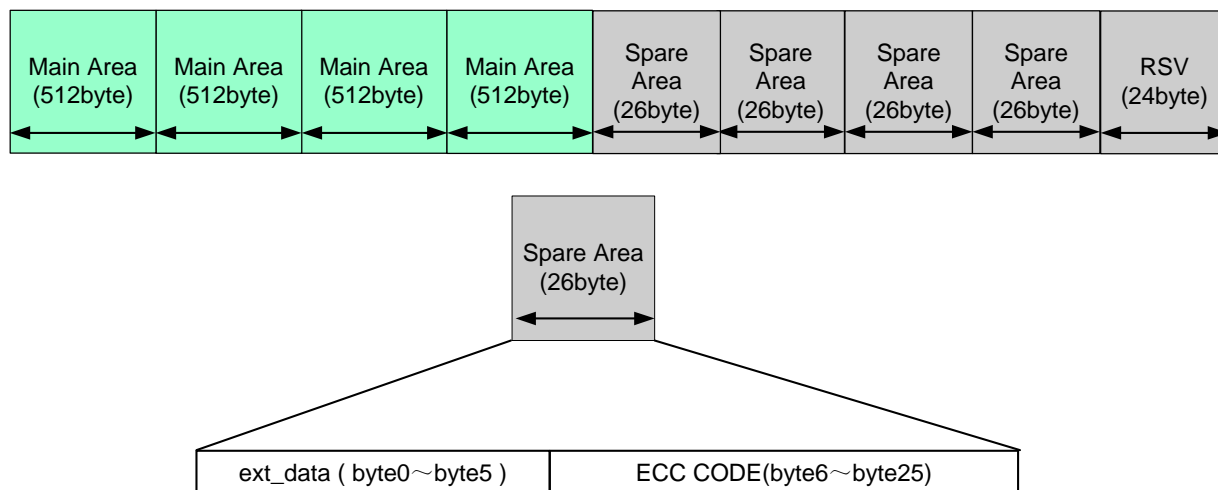
- 8bit ECC 模式

数据在 NANDC 内部 buffer 的存储数据结构如下，是 (2048+26×4) byte。对 4KB page size 的器件操作需要分两次进行。如图 4-15 所示。

当 ECC 使能打开时，写到 NAND Flash 器件内部的数据结构会自动调整为 Main Area 区和对应 Spare Area 区紧邻存放的格式。如图 4-13 所示。

对于东芝器件，其存储区的大小为 (4096+218) byte，因此，还有 (208~218) 共 10 个 byte 的剩余。

图4-15 8bit ECC 模式 (2048+26×4) byte page size NAND Flash 数据存储结构



初始化

初始化步骤如下：



- 步骤 1 向 `NFC_CON[op_mode]` 写 1，进入 Normal 模式。根据对接器件 page size 大小，设置 `NFC_CON[page_size]`。根据外接器件的片选个数和 ready_busy 信号，寄存器 `NFC_CON[ecc_type]`，设定校验和纠错模式。
 - 步骤 2 根据对接器件的时序要求，写寄存器 `NFC_PWIDTH`。
 - 步骤 3 如果是查询方式，写中断使能寄存器 `NFC_INTEN`，屏蔽所有中断。如果是中断方式，只需使能 `op_done` 中断，其余可屏蔽。
- 结束

对 NAND Flash 执行擦除操作

执行擦除操作的步骤如下：

- 步骤 1 向寄存器 `NFC_ADDRL` 和 `NFC_ADDRH` 写入编程 page 地址，向寄存器 `NFC_CMD` 写入擦除命令 `0x0070_D060`。
 - 步骤 2 向寄存器 `NFC_OP` 写入 `0x369`，启动 NANDC 对 NAND Flash 进行擦除操作（假设 NAND Flash 芯片需要 3 个地址，并且是对片选 0 操作）。
 - 步骤 3 查询方式下，检测 `NFC_STATUS[nfc_ready]`，如果为 1，进入步骤 4；否则，继续查询；中断方式下，检测 `NFC_INTS[op_done]`，如果为 1，进入步骤 4。
 - 步骤 4 读寄存器 `NFC_STATUS[nf_status]`，判断擦除是否成功。
- 结束

对 4KB page size NAND Flash 执行编程操作

执行编程操作的步骤如下：

- 步骤 1 CPU 先把 4KB 数据的前半部分（包括 2048byte main 数据区和 104byte 的 spare 数据区）写入 NANDC 的 buffer。（如果 `ecc_type` 为 `2'b00`，写寄存器 `NFC_DATA_NUM`，设置写的数目）
 - 步骤 2 CPU 向寄存器 `NFC_ADDRL` 和 `NFC_ADDRH` 写入 NAND Flash 的 page 地址，向寄存器 `NFC_CMD` 写入编程命令 `0x0070_1080`。
 - 步骤 3 CPU 向寄存器 `NFC_OP` 写入 `0x570`，启动 NANDC 对 NAND Flash 进行写操作（假设 NAND Flash 芯片需要 5 个地址，并且是对片选 0 操作）。
 - 步骤 4 查询方式下，检测 `NFC_STATUS[nfc_ready]`，直到该标志为 1，进入步骤 5；中断方式下，检测 `NFC_INTS[op_done]`，如果是 `op_done` 中断，进入步骤 5。
 - 步骤 5 CPU 把 4K 数据的后半部分（包括 2048byte main 数据区和 104byte 的 spare 数据区）写入 NANDC 的 buffer。向寄存器 `NFC_OP` 写入 `0x0d`，启动 NANDC 对 NAND Flash 进行写并发出编程命令。
 - 步骤 6 查询方式下，检测 `NFC_STATUS[nfc_ready]`，直到该标志为 1，进入步骤 7；中断方式下，检测 `NFC_INTS[op_done]`，如果是 `op_done` 中断，进入步骤 7。
 - 步骤 7 CPU 读寄存器 `NFC_STATUS[nf_status]`，判断编程是否成功。
- 结束



从 4KB page size 的 NAND Flash 读取一个 page 数据

读取数据的过程如下：

- 步骤 1 CPU 向寄存器 `NFC_ADDR_L` 和 `NFC_ADDR_H` 写入 NAND Flash 的 page 地址，向寄存器 `NFC_CMD` 写入读命令 0x3000。（如果 `ecc_type` 为 2'b00，需要写寄存器 `NFC_DATA_NUM`，设置读的数据数目。）
- 步骤 2 CPU 向寄存器 `NFC_OP` 写入 0x56E（假设 NAND Flash 芯片需要 5 个地址，并且是对片选 0 操作），启动 NANDC 读数据。
- 步骤 3 查询方式下，检测 `NFC_STATUS[nfc_ready]`，直到该标志为 1，进入步骤 4；中断方式下，检测 `NFC_INTS[op_done]`，如果是 `op_done` 中断，进入步骤 4。
- 步骤 4 CPU 读寄存器 `NFC_STATUS`，判断是否出现不可纠正的错误，以进行相应的处理。
- 步骤 5 CPU 从 NANDC 的 buffer 读数据写入内存。
- 步骤 6 CPU 向寄存器 `NFC_OP` 写入 0x02，以启动 NANDC 继续读 4K pagesize 的后半部分数据。
- 步骤 7 查询方式下，检测 `NFC_STATUS[nfc_ready]`，直到该标志为 1，进入步骤 8；中断方式下，检测 `NFC_INTS[op_done]`，如果是 `op_done` 中断，进入步骤 8。
- 步骤 8 CPU 读寄存器 `NFC_STATUS`，判断是否出现不可纠正的错误，以进行相应的处理。
- 步骤 9 CPU 从 NANDC 的 buffer 读数据写入内存。

----结束

其它注意事项

其它注意事项如下：

- 各个厂家的 NAND Flash 器件支持的操作命令有所不同，需要根据器件手册合理设置命令寄存器 `NFC_CMD`。同时，不同容量的 NAND Flash 器件需要的地址周期数不同，需要参照器件手册设置 `NFC_OP` 中的 `address_cycles` 域。不同器件支持的时序有所不同，需要根据器件手册合理设置读写脉冲宽度寄存器 `NFC_PWIDTH` 和操作间隔寄存器 `NFC_OPIDLE`。
- 在配置好相关寄存器和 buffer 后，再写 `NFC_OP` 寄存器，以启动 NANDC 发起 Flash 读写操作。之后，不要再写相关的寄存器，否则可能导致 NANDC 或 Flash 工作不正常。
- 写 `NFC_OP` 寄存器启动读写 NAND Flash 之后，在 `NFC_STATUS[nfc_ready]` 标志为 0 期间，不要读写 NANDC 的 buffer。否则，可能返回错误的的数据。

4.3.6 NANDC 寄存器概览

NANDC 寄存器概览如表 4-18 所示。



表4-18 NANDC 寄存器概览（基址是 0x1000_0000）

偏移地址	名称	描述	页码
0x00	NFC_CON	NANDC 配置寄存器	4-64
0x04	NFC_PWIDTH	读写脉冲宽度配置寄存器	4-65
0x08	NFC_OPIDLE	操作间隔配置寄存器	4-66
0x0C	NFC_CMD	命令字配置寄存器	4-66
0x10	NFC_ADDRL	低位地址配置寄存器	4-67
0x14	NFC_ADDRH	高位地址配置寄存器	4-67
0x18	NFC_DATA_NUM	读写数据数目配置寄存器	4-68
0x1C	NFC_OP	操作寄存器	4-69
0x20	NFC_STATUS	状态寄存器	4-70
0x24	NFC_INTEN	中断使能寄存器	4-71
0x28	NFC_INTS	中断状态寄存器	4-73
0x2C	NFC_INTCLR	中断清除寄存器	4-74
0x30	NFC_LOCK	锁地址配置寄存器	4-76
0x34	NFC_LOCK_SA0	锁起始地址 0 配置寄存器	4-77
0x38	NFC_LOCK_SA1	锁起始地址 1 配置寄存器	4-77
0x3C	NFC_LOCK_SA2	锁起始地址 2 配置寄存器	4-78
0x40	NFC_LOCK_SA3	锁起始地址 3 配置寄存器	4-79
0x44	NFC_LOCK_EA0	锁结束地址 0 配置寄存器	4-79
0x48	NFC_LOCK_EA1	锁结束地址 1 配置寄存器	4-80
0x4C	NFC_LOCK_EA2	锁结束地址 2 配置寄存器	4-81
0x50	NFC_LOCK_EA3	锁结束地址 3 配置寄存器	4-81
0x54	NFC_EXPCMD	扩展页命令寄存器	4-82
0x58	NFC_EXBCMD	扩展块命令寄存器	4-82
0x5C	NFC_ECC_TEST	ECC 测试寄存器	4-83

4.3.7 NANDC 寄存器描述

NFC_CON

NFC_CON 为 NANDC 配置寄存器。

Offset Address	Register Name	Total Reset Value												
0x00	NFC_CON	-												
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0													
Name	reserved				edo_en	ecc_type	ext_data_ecc_en	protection_en	rb_sel	es_ctrl	ecc_en	reserved	pagesize	op_mode
Reset	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	0	0 0 0 0	0	1	0	1	?	? 0 0 0	
Bits	Access	Name	Description											
[31:13]	-	reserved	保留。											
[12]	RW	edo_en	以 EDO 模式读取 NAND Flash 数据使能。 0: 正常模式; 1: EDO 模式。 使用该功能时, 需要参照具体器件的要求进行。											
[11:10]	RW	ecc_type	ECC 模式选择。 00: 无 ECC 模式; 01: 1bit ECC 模式; 10: 4bit ECC 模式; 11: 8bit ECC 模式。 复位值由管脚 EBIADR16/EBIADR15 决定。											
[9]	RW	ext_data_ecc_en	扩展数据区进行校验或纠错使能。 0: 禁止; 1: 使能。 注意: 只在 ecc_en 为 1 时有效。											
[8]	RW	protection_en	寄存器读写保护使能寄存器。 0: 禁止; 1: 使能。											
[7]	-	reserved	保留。											



[6]	RW	cs_ctrl	片选控制。 0: 在 NAND Flash 为 busy 时, 保持片选信号为 0; 1: 在 NAND Flash 为 busy 时, 把片选信号置位 1。 该模式对应 NAND Flash 的 cs do not care 模式。
[5]	RW	ecc_en	ECC 使能。 0: 禁止; 1: 使能。
[4:3]	-	reserved	保留。
[2:1]	RW	pagesize	NAND Flash 的页大小。 00: 保留; 01: 2KB; 10: 4KB; 11: 保留。 复位值由管脚 NFC_PAGE_SIZE 决定。
[0]	RW	op_mode	NANDC 所处的工作模式。 0: NANDC 处于 Boot 模式; 1: NANDC 处于 NORMAL 模式。

NFC_PWIDTH

NFC_PWIDTH 为读写脉冲宽度配置寄存器。

	Offset Address	Register Name	Total Reset Value																					
	0x04	NFC_PWIDTH	-																					
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																							
Name	reserved												rw_hcnt				r_lcnt				w_lcnt			
Reset	? ?												0 1 0 1				0 1 0 1				0 1 0 1			
Bits	Access	Name	Description																					
[31:12]	-	reserved	保留。																					
[11:8]	RW	rw_hcnt	NAND Flash 读/写信号高电平宽度。 0x0~0xF: 1~16 个时钟周期。																					
[7:4]	RW	r_lcnt	NAND Flash 读信号低电平宽度。 0x0~0xF: 1~16 个时钟周期。																					
[3:0]	RW	w_lcnt	NAND Flash 写信号低电平宽度。 0x0~0xF: 1~16 个时钟周期。																					



NFC_OPIDLE

NFC_OPIDLE 为操作间隔配置寄存器。

	Offset Address 0x08								Register Name NFC_OPIDLE								Total Reset Value -															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								frb_wait				cmd1_wait				addr_wait				write_data_wait				cmd2_wait				frb_idle			
Reset	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bits	Access		Name		Description																											
[31:24]	-		reserved		保留。																											
[23:20]	RW		frb_wait		发出读写命令后，先延时一段时间，然后再检测 ready 信号是否变为高电平。延时的周期数为：frb_wait×8。																											
[19:16]	RW		cmd1_wait		发送完 Command1 之后的等待周期数。 0x0~0xF：1~16 个时钟周期。																											
[15:12]	RW		addr_wait		发送完 Address 之后的等待周期数。 0x0~0xF：1~16 个时钟周期。																											
[11:8]	RW		write_data_wait		写数据之后的等待周期数。 0x0~0xF：1~16 个时钟周期。																											
[7:4]	RW		cmd2_wait		发送完 Command2 之后的等待周期数。 0x0~0xF：1~16 个时钟周期。																											
[3:0]	RW		frb_idle		NAND Flash 的 ready 信号变高之后，延时一段时间,之后才能发出读信号读数据。 延时的周期数为：frb_idle×8。																											

NFC_CMD

NFC_CMD 为命令字配置寄存器。



Offset Address		Register Name		Total Reset Value												
0x0C		NFC_CMD		-												
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0								
Name	reserved				read_status_cmd		cmd2		cmd1							
Reset	? ? ? ?				0 1 1 1		0 0 0 0		0 0 1 1		0 0 0 0		0 0 0 0		0 0 0 0	
Bits	Access	Name		Description												
[31:24]	-	reserved		保留。												
[23:16]	RW	read_status_cmd		Read status 命令字。												
[15:8]	RW	cmd2		NANDC 发给 NAND Flash 的第 2 个命令。												
[7:0]	RW	cmd1		NANDC 发给 NAND Flash 的第 1 个命令。												

NFC_ADDRL

NFC_ADDRL 为低位地址配置寄存器。

Offset Address		Register Name		Total Reset Value							
0x10		NFC_ADDRL		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	addr_1										
Reset	0 0 0 0		0 0 0 0		0 0 0 0		0 0 0 0		0 0 0 0		
Bits	Access	Name		Description							
[31:0]	RW	addr_1		NAND Flash 低 32bit 地址。							

NFC_ADDRH

NFC_ADDRH 为高位地址配置寄存器。

Offset Address		Register Name		Total Reset Value								
0x14		NFC_ADDRH		-								
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0				
Name	reserved				addr_h							
Reset	? ? ? ?		? ? ? ?		? ? ? ?		0 0 0 0		0 0 0 0		0 0 0 0	
Bits	Access	Name		Description								
[31:16]	-	reserved		保留。								



[15:0]	RW	addr_h	NAND Flash 高 16bit 地址。
--------	----	--------	------------------------

NFC_DATA_NUM

NFC_DATA_NUM 为读写数据数目配置寄存器。

	Offset Address	Register Name	Total Reset Value	
	0x18	NFC_DATA_NUM	-	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	reserved			
Reset	? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 1 0 0 0 0 1 0 0 0 0 0 0			
Bits	Access	Name	Description	
[31:12]	-	reserved	保留。	
[11:0]	RW	nfc_data_num	配置 NANDC 随机读写的数据个数，最大 2152byte (2048+26×4)。 注意：只在 ecc_type 为 00 时有效。	



NFC_OP

NFC_OP 为操作寄存器。

	Offset Address 0x1C				Register Name NFC_OP								Total Reset Value -																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																address_cycles	nf_cs		cmd1_en	addr_en	write_data_en	cmd2_en	wait_ready_en	read_data_en	read_status_en						
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:12]	-	reserved	保留。																													
[11:9]	RW	address_cycles	发给 NAND Flash 的地址周期数。 该字段的复位值由管脚 EBIADR18/EBIADR17 决定，等于 EBIADR18/EBIADR17 加 3																													
[8:7]	RW	nf_cs	选择操作的 NAND Flash 器件。 00: cs0; 01: cs1; 其他: 保留。																													
[6]	RW	cmd1_en	发 Command1 命令使能。 0: 禁止; 1: 使能。																													
[5]	RW	addr_en	向 NAND Flash 写操作地址使能。 0: 禁止; 1: 使能。																													
[4]	RW	write_data_en	向 NAND Flash 写数据使能。 0: 禁止; 1: 使能。 注意: read_data_en 和 write_data_en 不能同时为 1。																													
[3]	RW	cmd2_en	发 Command2 命令使能。 0: 禁止; 1: 使能。																													

[2]	RW	wait_ready_en	等待 ready/busy 信号变高使能。 0: 禁止; 1: 使能。
[1]	RW	read_data_en	启动读状态机, 从 NAND Flash 读数据使能。 0: 禁止; 1: 使能。 注意: read_data_en 和 write_data_en 不能同时为 1。
[0]	RW	read_status_en	该标志为 1 时, 使能向 NAND 发出读 status 的 0x70 命令, 并从 NAND Flash 读取状态数据, 返回的数据写入 NANDC 状态寄存器的 NFC_STATUS 域中 (而不写入内部 buffer)。 在对 NAND 器件进行擦除和编程时, 需要读取擦除和编程的结果, 看是否成功。使能时, CPU 一次操作就可以完成编程擦除等操作, 并从 NAND 返回了是否操作成功的数据, 从而减少了 CPU 的干预。 注意: 在 read_data_en 标志为 1 时, 该标志无效。

NFC_STATUS

NFC_STATUS 为状态寄存器。

	Offset Address	Register Name	Total Reset Value	
	0x20	NFC_STATUS	-	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	<div style="display: flex; justify-content: space-between;"> <div style="width: 40%;">ecc_num</div> <div style="width: 5%; text-align: center; color: blue;">reserved</div> <div style="width: 30%; text-align: center;">nf_status</div> <div style="width: 5%; text-align: center; color: blue;">reserved</div> <div style="width: 5%; text-align: center; color: blue;">ready</div> <div style="width: 5%; text-align: center; color: blue;">nfc_ready</div> </div>			
Reset	0 0			
Bits	Access	Name	Description	
[31:16]	RO	ecc_num	<p>在 ecc_type 为 10 和 11 的情况下, 表示每个 sector (如果 ext_data_ecc_en 为 1, 也包括 6 个 byte 的扩展数据) 的错误 bit 数。</p> <p>bit[3:0]: sector_0 的错误比特数; bit[7:4]: sector_1 的错误比特数; bit[11:8]: sector_2 的错误比特数; bit[15:12]: sector_3 的错误比特数。</p> <p>在 ecc_type=01 的情况下, 表示每个 sector 和 ext_data 的解码结果。</p> <p>bit[1:0]: sector_0 解码结果;</p>	



			<p>bit[3:2]: sector_1 解码结果;</p> <p>bit[5:4]: sector_2 解码结果;</p> <p>bit[7:6]: sector_3 解码结果;</p> <p>bit[9:8]: ext_data_0 解码结果;</p> <p>bit[11:10]: ext_data_1 解码结果;</p> <p>bit[13:12]: ext_data_2 解码结果;</p> <p>bit[15:14]: ext_data_3 解码结果。</p> <p>解码结果含义:</p> <p>00: 无错误;</p> <p>01: 有 1bit 错误;</p> <p>10: 数据无错误, 但存储的检验码有错误;</p> <p>11: 不可纠正的错误 (2bit 和 2bit 以上的错误)。</p>
[15:13]	-	reserved	保留。
[12:5]	RO	nf_status	<p>读回的 NAND Flash 的 status 数据。</p> <p>只在 NFC_OP 寄存器的 read_status 标志为 1, 且该寄存器的 nfc_ready 标志为 1 时有效。</p>
[4:2]	-	reserved	保留。
[1]	RO	ready	NAND Flash 器件的 ready/busy 信号状态。
[0]	RO	nfc_ready	<p>NANDC 的 ready/busy 信号状态。</p> <p>0: NANDC 正在进行操作;</p> <p>1: 操作完成, 可以接收下一次命令。</p> <p>当写 NFC_OP 寄存器启动 NANDC 操作时, 该位自动清零。</p>

NFC_INTEN

NFC_INTEN 为中断使能寄存器。

Offset Address		Register Name		Total Reset Value																				
0x24		NFC_INTEN		-																				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0																
Name	reserved						wr_buf_err_int_en	wr_buf_busy_int_en	err_invalid	err_valid	reserved	cs1_done_en	cs0_done_en	op_done_en										
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																					
[31:9]	-	reserved	保留。																					
[8]	RW	wr_buf_err_int_en	对 lock 地址进行写操作错误中断使能。 0: 禁止; 1: 使能。																					
[7]	RW	wr_buf_busy_int_en	NANDC 读写 NAND Flash 数据期间, CPU 读写 NANDC buffer 错误中断使能。 0: 禁止; 1: 使能。																					
[6]	RW	err_invalid	不可纠正错误, 产生中断。																					
[5]	RW	err_valid	可纠正的错误, 产生中断。																					
[4:3]	-	reserved	保留。																					
[2]	RW	cs1_done_en	ready busy 信号由低变高, 产生中断使能。 0: 禁止; 1: 使能。																					
[1]	RW	cs0_done_en	ready busy 信号由低变高, 产生中断使能。 0: 禁止; 1: 使能。																					
[0]	RW	op_done_en	NANDC 本次操作结束中断使能。 0: 禁止; 1: 使能。																					



NFC_INTS

NFC_INTS 为中断状态寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x28				NFC_INTS				-																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																wr_buf_err_int	wr_buf_busy_int	err_invalid	err_vavid	reserved	cs1_done	cs0_done	op_done								
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:9]	-	reserved	保留。																													
[8]	RO	wr_buf_err_int	对 lock 地址进行写操作产生的中断。 0: 不中断; 1: 中断。																													
[7]	RO	wr_buf_busy_int	NANDC 对 NAND Flash 进行数据操作期间, CPU 读写 NANDC buffer。 0: 不中断; 1: 中断。																													
[6]	RO	err_invalid	不可纠正的错误。 0: 不中断; 1: 中断。 1bit 纠错模式下, 校验 512byte 数据中出现 2bit 以上错误, 产生中断; 4bit 纠错模式下, 校验 512byte 数据中出现 5bit 以上错误, 产生中断; 8bit 纠错模式下, 校验 512byte 数据中出现 8bit 以上错误, 产生中断。																													



[31:9]	-	reserved	保留。
[8]	WO	wr_buf_err_int_clr	清除 wr_buf_err_int 中断。 0: 不清除; 1: 清除。
[7]	WO	wr_buf_busy_int_clr	清除 wr_buf_busy_int 中断。 0: 不清除; 1: 清除。
[6]	WO	r_5bit_err_clr	清除 r_5bit_err 中断。 0: 不清除; 1: 清除。
[5]	WO	r_4bit_err_clr	清除 r_4bit_err 中断。 0: 不清除; 1: 清除。
[4:3]	-	reserved	保留。
[2]	WO	cs1_done_clr	清除 cs1_done 中断。 0: 不清除; 1: 清除。
[1]	WO	cs0_done_clr	清除 cs0_done 中断。 0: 不清除; 1: 清除。
[0]	WO	op_done_clr	清除 op_done 中断。 0: 不清除; 1: 清除。

NFC_LOCK

NFC_LOCK 为锁地址配置寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x30								NFC_LOCK								-															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								lock_excmd_en	lock_en	global_lock_en	lock_down				
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0
Bits	Access		Name		Description																											
[31:4]	-		reserved		保留。																											
[3]	RW		lock_excmd_en		针对扩展写命令（新器件有可能添加的新命令），对保护地址进行写保护使能。 0：禁止； 1：使能。																											
[2]	RW		lock_en		Flash lock 使能。该控制位为 1 时，如果擦除或编程的地址位于锁存首地址与锁存结束地址之间，擦除和编程无效。 0：禁止； 1：使能。																											
[1]	RW		global_lock_en		Flash 全局 lock 使能。为 1 时，将不允许对 NAND Flash 进行擦除或编程操作。 0：禁止； 1：使能。																											
[0]	RW		lock_down		NAND Flash lock 模式。 0：lock 模式； 1：lock_down 模式，写 1 后，再次写无效，只有硬件 reset 才能对该位清 0。																											



NFC_LOCK_SA0

NFC_LOCK_SA0 为锁起始地址 0 配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x34				NFC_LOCK_SA0				-																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_addr0																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											
[20:19]	RW		flash_lock_cs		NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。																											
[18:0]	RW		flash_lock_addr0		锁存首地址 0, 最低位对应 NAND Flash 第 5 个行地址。																											

NFC_LOCK_SA1

NFC_LOCK_SA1 为锁起始地址 1 配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x38				NFC_LOCK_SA1				-																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_addr1																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											

[20:19]	RW	flash_lock_cs	NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。
[18:0]	RW	flash_lock_addr1	锁存首地址 1, 最低位对应 NAND Flash 第 5 个行地址。

NFC_LOCK_SA2

NFC_LOCK_SA2 为锁起始地址 2 配置寄存器。

	Offset Address	Register Name	Total Reset Value
	0x3C	NFC_LOCK_SA2	-
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	flash_lock_cs	flash_lock_addr2
Reset	? ? ? ? ? ? ? ? ? ? ? 0		
Bits	Access	Name	Description
[31:21]	-	reserved	保留。
[20:19]	RW	flash_lock_cs	NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。
[18:0]	RW	flash_lock_addr2	锁存首地址 2, 最低位对应 NAND Flash 第 5 个行地址。



NFC_LOCK_SA3

NFC_LOCK_SA3 为锁起始地址 3 配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x40				NFC_LOCK_SA3				-																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_addr3																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											
[20:19]	RW		flash_lock_cs		NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。																											
[18:0]	RW		flash_lock_addr3		锁存首地址 3, 最低位对应 NAND Flash 第 5 个行地址。																											

NFC_LOCK_EA0

NFC_LOCK_EA0 为锁结束地址 0 配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x44				NFC_LOCK_EA0				-																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_eaddr0																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											

[20:19]	RW	flash_lock_cs	NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。
[18:0]	RW	flash_lock_eaddr0	锁存结束地址 0, 最低位对应 NAND Flash 第 5 个行地址。

NFC_LOCK_EA1

NFC_LOCK_EA1 为锁结束地址 1 配置寄存器。

	Offset Address	Register Name	Total Reset Value
	0x48	NFC_LOCK_EA1	-
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	flash_lock_cs	flash_lock_eaddr1
Reset	? ? ? ? ? ? ? ? ? ? ? 0		
Bits	Access	Name	Description
[31:21]	-	reserved	保留。
[20:19]	RW	flash_lock_cs	NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。
[18:0]	RW	flash_lock_eaddr1	锁存结束地址 1, 最低位对应 NAND Flash 第 5 个行地址。



NFC_LOCK_EA2

NFC_LOCK_EA2 为锁结束地址 2 配置寄存器。

	Offset Address 0x4C				Register Name NFC_LOCK_EA2								Total Reset Value -																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_eaddr2																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											
[20:19]	RW		flash_lock_cs		NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。																											
[18:0]	RW		flash_lock_eaddr2		锁存结束地址 2, 最低位对应 NAND Flash 第 5 个行地址。																											

NFC_LOCK_EA3

NFC_LOCK_EA3 为锁结束地址 3 配置寄存器。

	Offset Address 0x50				Register Name NFC_LOCK_EA3								Total Reset Value -																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								flash_lock_cs	flash_lock_eaddr3																						
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		reserved		保留。																											



[20:19]	RW	flash_lock_cs	NAND Flash 锁存片选。 00: 片选 0; 01: 片选 1; 其他: 保留。
[18:0]	RW	flash_lock_eaddr3	锁存结束地址 3, 最低位对应 NAND Flash 第 5 个行地址。

NFC_EXPCMD

NFC_EXPCMD 为扩展页命令寄存器。

Offset Address		Register Name		Total Reset Value				
0x54		NFC_EXPCMD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	ex_pcmd3		ex_pcmd2		ex_pcmd1		ex_pcmd0	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	ex_pcmd3	NAND Flash 扩展页写操作命令 3。					
[23:16]	RW	ex_pcmd2	NAND Flash 扩展页写操作命令 2。					
[15:8]	RW	ex_pcmd1	NAND Flash 扩展页写操作命令 1。					
[7:0]	RW	ex_pcmd0	NAND Flash 扩展页写操作命令 0。					

NFC_EXBCMD

NFC_EXBCMD 为扩展块命令寄存器。

Offset Address		Register Name		Total Reset Value				
0x58		NFC_EXBCMD		-				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved			ex_bcmd1		ex_bcmd0		
Reset	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	-	reserved	保留。					
[15:8]	RW	ex_bcmd1	NAND Flash 扩展块写操作命令 1。					
[7:0]	RW	ex_bcmd0	NAND Flash 扩展块写操作命令 0。					



NFC_ECC_TEST

NFC_ECC_TEST 为 ECC 测试寄存器。

	Offset Address 0x5C								Register Name NFC_ECC_TEST								Total Reset Value 0x0020_F001															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	version								empty				reserved								ecc_mask	dec_only	enc_only									
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
Bits	Access	Name	Description																													
[31:16]	RO	version	NANDC 版本号。																													
[15:12]	RO	empty	NANDC 内部 FIFO 的 empty 标志。 bit[15]: ECC8 异步 FIFO 的 empty 标志; bit[14]: ECC 同步 FIFO 的 empty 标志; bit[13]: 写 NAND Flash 的 FIFO 的 empty 标志; bit[12]: 读 NAND Flash 的 FIFO 的 empty 标志。																													
[11:3]	-	reserved	保留。																													
[2]	RW	ecc_mask	ECC 功能掩码。 为 0 时, 按 ecc_type 的值决定是否进行校验和纠错。 为 1 时, 不进行 ecc 校验和纠错。但读写 NAND Flash 数据结构仍按照 ecc_type 的格式进行转换。																													
[1]	RW	dec_only	只解码使能。 向该寄存器位写 1 时, 启动 ECC 解码, 但不启动读写 NAND Flash; 读该寄存器位时, 返回 0。																													
[0]	RW	enc_only	只编码使能。 向该寄存器位写 1 时, 启动 ECC 编码, 但不启动读写 NAND Flash; 读该寄存器位时, 返回值为 1 表示 ECC 编解码结束。 为 0 表示正在编解码。																													



5 GMAC 接口

5.1 概述

千兆以太网接口，以下简称 GMAC (Gigabit Media Access Control)，GMAC 模块实现网络接口数据的接收和发送，可以工作在 10Mbit/s、100Mbit/s 或 1000Mbit/s 模式下，支持全双工或者半双工工作模式，提供 MII/RGMII 接口。GMAC 提供使用本机 MAC/广播/组播进行帧过滤的功能，可以通过配置过滤表对网口的帧进行选择性过滤接收，另外还提供流量限制功能。

5.2 特性

GMAC 模块具有以下特性：

- 支持 MII 接口 10Mbit/s 和 100Mbit/s 速率。
- 支持 RGMII 接口 10Mbit/s、100Mbit/s 和 1000Mbit/s 速率。
- 支持 MII/RGMII 全双工工作模式。
- 支持 MII 半双工工作模式。
- 支持帧长有效性检测功能，并丢弃不符合规定长度的数据包。
- 支持对接收帧进行 CRC 校验。
- 支持短帧填充功能，当发送数据不足 64byte，可根据配置填充至 64byte。
- 支持端口全双工工作模式下的内环回、外环回。
- 提供频率为 2.5MHz/18MHz 的 MDIO 接口。
- 支持对端口收发帧进行统计计数。
- 提供接收 8KB 和发送 4KB 的缓冲区。
- 提供接收广播帧、多播帧和单播帧功能。
- 提供对控制报文、广播或多播报文的限速处理功能。
- 提供包过滤功能。
- 提供收包中断和超时中断 2 种中断方式。
- 支持硬件根据配置的地址将接收数据写入到外部存储器。



- 提供按包从外部存储器中搬运包的功能。

5.3 信号描述

外部接口信号描述如表 5-1 所示。

表5-1 外部接口信号

信号名称	方向	描述	对应管脚
PHY_RXC	I	MII: MII 接口接收时钟, 在 100M/10M 模式下分别为 25MHz 和 2.5MHz。 RGMII: RGMII 接口接收时钟, 在 1000M/100M/10M 模式下分别为 125MHz、25MHz 和 2.5MHz。	GRCLK
PHY_RXD[3:0]	I	MII: MII 接口接收数据, 仅使用 PHY_RXD[3:0]。 RGMII: RGMII 接口接收数据, 仅使用 PHY_RXD[3:0]。	GRXD0~ GRXD3
PHY_RXDV	I	MII: MII 接口接收数据有效。 RGMII: RGMII 接口接收数据有效。	GRXDV
PHY_COL	I	MII: MII 接口半双工冲突指示。 RGMII: RGMII 接口半双工冲突指示。 与 GPIO 信号复用。	GCOL
PHY_CRS	I	MII: MII 接口载波有效指示。 RGMII: RGMII 接口载波有效指示。 与 GPIO 信号复用。	GCRS
PHY_TXC	I	MII: MII 接口发送时钟, 在 100M/10M 模式下分别为 25MHz 和 2.5MHz。 RGMII: 不使用此信号。	GTCLK
PHY_GTXC	O	MII: 不使用此信号。 RGMII: RGMII 接口发送时钟, 在 1000M/100M/10M 模式下分别为 125MHz、25MHz 和 2.5MHz。 与 GPIO 信号复用。	GTCLKOUT
PHY_TXEN	O	MII: MII 接口发送数据使能。 RGMII: RGMII 接口发送数据使能。	GTXEN



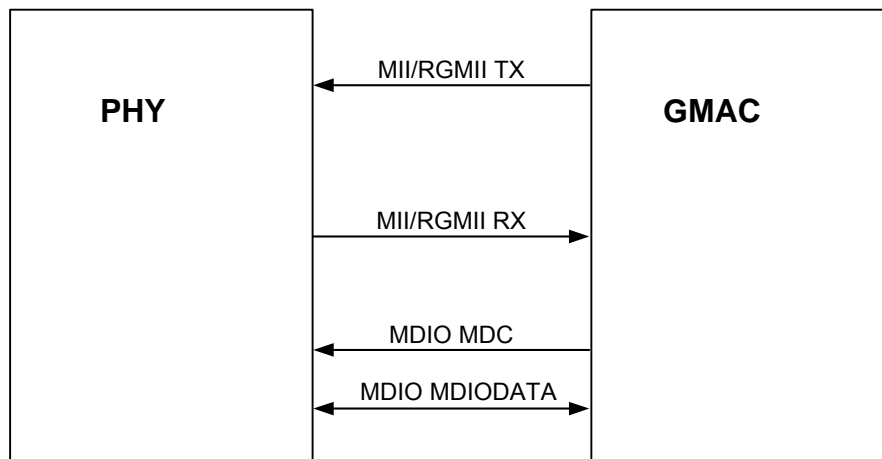
信号名称	方向	描述	对应管脚
PHY_TXD[3:0]	O	MII: MII 接口发送数据, 仅使用 PHY_TXD[3:0]。 RGMII: RGMII 接口发送数据, 仅使用 PHY_TXD[3:0]。	GTXD0~GTXD3
FE_MDC	O	MII: MDIO 接口输出时钟, 2.5MHz 和 18MHz 可配。 RGMII: 同 MII。	MDCK
FE_MDIO	I/O	MII: MDIO 接口双向数据。 RGMII: 同 MII。	MDIO

5.4 功能描述

5.4.1 典型应用

典型应用如图 5-1 所示。

图5-1 典型应用



5.4.2 支持的帧格式

GMAC 支持 Ethernet II 帧格式, Ethernet II 帧格式如图 5-2 所示。

图5-2 Ethernet II 帧格式图





Ethernet II 帧内容描述如表 5-2 所示。

表5-2 Ethernet II 帧描述

域名	位宽 (byte)	描述
DA	6	目的 MAC 地址。
SA	6	源 MAC 地址，指示发送该帧的工作站地址。
Etype	2	Ethernet 类型，大于等于 0x05DD。
Data	46~1500	Ethernet 数据，包括 PAD。
FCS	4	添加的 CRC 校验码。

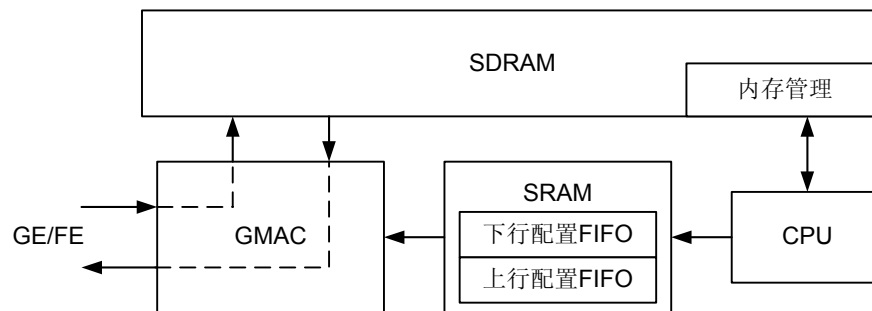
5.4.3 GMAC 上下行帧管理功能

CPU 通过在上下行方向上配置 FIFO 对 GMAC 进行上下行帧地址管理：

- 下行时：GMAC 分辨从外网收到的数据包是否合法，如果是非法包，则丢弃；如果是合法包，则通过总线存放内存中，地址由 CPU 通过下行配置 FIFO 指定。
- 上行时：GMAC 根据 CPU 通过上行配置 FIFO 配置的地址和长度，通过总线搬运指定地址的数据，自行组装成包，然后发送到网络。

总体数据流图如图 5-3 所示。

图5-3 总体数据流图



GE: Gigabit Ethernet
FE: Fast Ethernet, 指百兆网口

上下行配置 FIFO 的空间大小

上行（发送）配置 FIFO 和下行（接收）配置 FIFO 共用 512×43bit 空间的缓存，且两个 FIFO 的空间大小可调。

通过寄存器 CRF_RX_BASE_ADDR bit[8:0]配置接收配置 FIFO 的存储空间。



上行配置 FIFO 的存储空间=512-下行配置 FIFO 的存储空间。



注意

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，且不能配置该值为 0，配置完必须清空 FIFO 一次（配置 CRF_FIFO_CLR 寄存器，偏移地址为 0x0274），才能使配置生效。

下行配置 FIFO 中数据存放格式

下行配置 FIFO 中数据存放格式如图 5-4 所示。

图5-4 下行配置 FIFO 中的数据格式

32bit
Address 1
Address 2
.....
Address N

下行配置 FIFO 中存储 SDRAM 中划分给下行包的缓存地址（32bit），CPU 只写，GMAC 只读。

上行配置 FIFO 的数据存放格式

上行配置 FIFO 的数据存放格式如图 5-5 所示。

图5-5 上行配置 FIFO 中的数据格式

bit[42:11]	bit[10:0]
Address 1	Length 1
Address 2	Length 2
.....
Address N	Length N

上行配置 FIFO 内存储上行包的控制字（43bit），指示 GMAC 搬运包的地址、长度等信息。CPU 分两次写入，必须先配置长度，再配置起始地址。CPU 只写，GMAC 只读。其中：

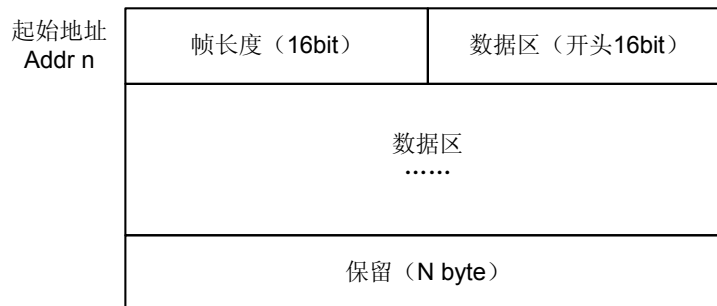
- Address: 表示包或块在 SDRAM 中缓存地址的 bit[31:0]，因为缓存地址为 32bit。

- **Length:** 表示该包或块的数据长度，11 位，范围为 14~2000byte，不可配置为 0。
以最大帧长为 1.6KB 为例，分配给每个帧的存放空间要比 1.6KB 大一些，至少配置为 1600byte+128byte。软件需要保证此大小的地址空间为 GMAC 存放包数据可用的存取空间。

下行 SDRAM 中数据的存放格式

GMAC 成功接收一个包后，会根据下行（接收）配置 FIFO 中的当前地址，将接收包缓存到 SDRAM 中。GMAC 存放到 SDRAM 中的包按照图 5-6 格式存放。

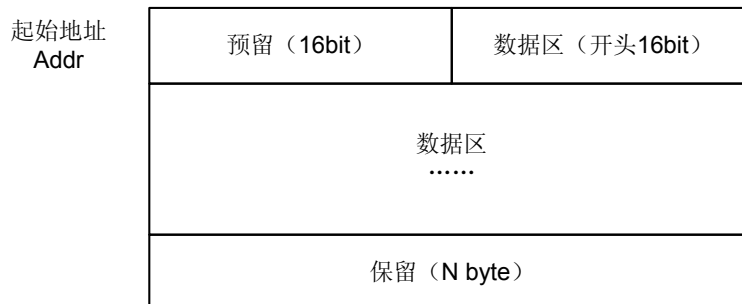
图5-6 下行数据帧在 SDRAM 中存放格式



上行 SDRAM 中数据的存放格式

当 CPU 需要将包发送出去时，需要先将数据按照图 5-7 的格式存放在 SDRAM 中。
上行数据在 SDRAM 中的存放格式如图 5-7 所示。

图5-7 上行数据在 SDRAM 中存放格式



5.4.4 GMAC 限速功能

GMAC 具有对接收报文进行限速的功能，即在某段时间内，当通过的报文数超过设定的最大数量时，后续报文将被丢弃。

GMAC 对报文限速分为 3 种：

- [对控制报文的限速](#)



- 对 IP 报文的限速
- 对广播或多播报文的限速

对控制报文的限速

在一个限速时间 T 内，只允许通过小于等于限速报文个数的控制报文，超过该数的控制报文将被丢弃。该限速时间 T 以 125 μ s 为单位进行计数。

对控制报文进行限速时，需要做如下配置：

- 配置 CONTROL_WORD bit[20]为 1，使能控制报文限速。
- 配置 FLOW_CTRL_PKG_THRSLD bit[15:0]，设定控制报文的包个数上限。
- 配置 CRF_FLOW_TIME_THRSLD bit[7:0]，设定限速时间。

对 IP 报文的限速

对 IP 报文的限速功能和控制报文限速相同。对 IP 报文进行限速时，需要做如下配置：

- 配置 CONTROL_WORD bit[21]为 1，设定对 IP 报文限速。
- 配置 FLOW_CTRL_PKG_THRSLD bit[31:16]，设定 IP 报文的包个数上限。
- 配置 CRF_FLOW_TIME_THRSLD bit[7:0]，设定限速时间。

对广播或多播报文的限速

广播或多播报文的限速时间以 1 μ s 为单位进行计数，在限速时间内，当通过的报文达到限速个数时，后续报文被丢弃。

对广播或者多播报文进行限速时，需要做如下配置：

- 配置 CONTROL_WORD bit[16]为 1，设定对广播或多播报文限速。
- 配置 CRF_BM_PKT_THRSLD，设定广播和多播报文的包数上限。
- 配置 CRF_BM_TIME_THRSLD，设定广播和多播报文的限速时间上限。

5.4.5 GMAC 收包中断管理功能

中断产生

通过配置中断控制寄存器 CRF_INTRPT_THRSLD，产生中断。

当收包中断使能后，下面任何一种情况均会产生收包中断：

- 配置包个数中断使能后，GMAC 搬运到 SDRAM 中的包个数达到配置的中断包个数，则产生一个包个数中断。
- 配置超时收包中断使能后，GMAC 搬运成功至少一个包后，等待时间达到配置的超时时间，则产生一个超时收包中断。

中断清除

**注意**

超时收包中断和包个数中断对应同一个中断状态信号，只需清 0 一次（向 CRF_INTRPT_CLR[rx_crf_intrpt]写 1）。

CPU 收到收包中断时，向中断清零寄存器 CRF_INTRPT_CLR bit[18]写 1，撤消收包中断（超时收包中断和包个数中断均自动清 0），同时 CRF_INTRPT_CLR bit[18]变为 0。

5.4.6 GMAC 非法报文过滤功能

GMAC 具有对非法报文过滤的功能，通过配置过滤列表，指示哪些类型的报文可以通过。该过滤分为 2 级，分别针对 MAC 的帧类型和 IP Protocol 进行过滤。

过滤时，需要首先考虑是否是控制报文：

- 若接收帧的帧类型与配置的 4 个控制报文类型中任一相符（CRF_CTRL_0_TYPE~CRF_CTRL_3_TYPE），则认为是控制报文。控制报文默认为合法报文通过。
- 若接收报文不是控制报文，则需要针对过滤列表进行过滤。针对 MAC 类型和 IP Protocol 进行过滤时，可以配置寄存器 FILTER_LIST0~FILTER_LIST15。其中，只有配置成可以通过 IP 报文时，设置的 IP Protocol 的过滤列表才起作用。

5.5 工作方式

5.5.1 GMAC 配置 PHY 芯片工作状态

GMAC 提供 MDIO 接口实现对 PHY 芯片的管理。MDIO 分为读操作和写操作。

读操作的配置步骤如下：

- 步骤 1 CPU 向 MDIO 单次操作寄存器 MDIO_SINGLE_CMD bit[9:8]写入 PHY 芯片地址，向 MDIO_SINGLE_CMD bit[4:0]写入 PHY 内部寄存器地址，同时配置 MDIO_SINGLE_CMD 的 bit[20]为 1，配置 MDIO_SINGLE_CMD bit[17:16]为 10，启动 MDIO 读操作。
- 步骤 2 MDIO 将从外部 PHY 芯片读回的数据写到 MDIO 读写数据寄存器 MDIO_SINGLE_DATA bit[31:16]，并同时 MDIO_SINGLE_CMD bit[20]配置为 0。
- 步骤 3 CPU 查询 MDIO_SINGLE_DATA bit[31:16]，得到 MDIO 从外部 PHY 芯片读回的数据。

----结束

**说明**

- 当查询到 MDIO_SINGLE_CMD bit[20]为 0 时，表示硬件已经完成对 PHY 芯片的读写操作。具体的寄存器配置内容需要参考相关 PHY 芯片的用户手册。
- 硬件不支持自动获取 PHY 状态的自适应方式。自适应方式需软件定时查询 PHY 状态信息来完成。



写操作的配置步骤如下：

- 步骤 1 CPU 将发送给外部 PHY 芯片的数据写入 MDIO_SINGLE_DATA bit[15:0]。
- 步骤 2 向 MDIO_SINGLE_CMD bit[9:8]写入 PHY 芯片地址，向 MDIO_SINGLE_CMD bit[4:0]写入 PHY 内部寄存器地址，同时配置 MDIO_SINGLE_CMD bit[20]为 1，配置 MDIO_SINGLE_CMD bit[17:16]为 01，启动 MDIO 写操作。
- 步骤 3 MDIO 将 MDIO_SINGLE_DATA bit[15:0]的值写入相应的 PHY 内部寄存器中，完成 MDIO 的写操作，并自动将 MDIO_SINGLE_CMD bit[20]配置为 0。

----结束

5.5.2 GMAC 收发包配置



说明

GMAC 收发包由芯片自动完成，不需要手工配置。

GMAC 收包配置

当 GMAC 接收到合法数据包时，需要在 CPU 的控制下，将报文搬运到 SDRAM 中缓存。

具体步骤如下：

- 步骤 1 配置收包中断寄存器（详细内容请参见“5.4.5 GMAC 收包中断管理功能”）。
- 步骤 2 查询配置 FIFO 数据个数寄存器 CRF_CFF_DATA_NUM：
 - 若未满，则向接收配置 FIFO（下行配置 FIFO）中写收包缓存地址（其入口地址为 0x1009_0480，缓存地址的格式请参见图 5-4），进入步骤 3。
 - 若满，则等待收包中断到来。
- 步骤 3 GMAC 收到数据包后，根据接收地址，将数据包按照图 5-6 所示的格式存放在 SDRAM 中，当达到收包中断条件时，产生收包中断。CPU 收到收包中断后，清中断，并启动收包。

----结束

GMAC 发包配置

GMAC 发包时，由 CPU 写上行配置 FIFO，GMAC 根据 FIFO 中的内容，主动从 SDRAM 中搬运数据，并组装成数据包，发送到网络。

具体步骤如下：

- 步骤 1 CPU 将要发送的数据包存放在上行 SDRAM 中，存放方式请参见图 5-7。



说明

若已配置控制寄存器 CONTROL_WORD bit[22]为 0，则一个完整的包需要按照图 5-7 的格式存放。

- 步骤 2 CPU 写上行配置 FIFO 前，需查询配置 FIFO 数据个数寄存器 CRF_CFF_DATA_NUM：



- 若未滿，則向發送配置 FIFO 中寫上行配置信息（其入口地址為 0x1009_0400 和 0x1009_0580，配置信息的格式請參見圖 5-5），進入步驟 3。
- 若滿，則等待發包中斷到來。

步驟 3 GMAC 模塊自動完成發包處理。

----結束

5.6 GMAC 寄存器概覽

GMAC 寄存器概覽如表 5-3 所示。

表5-3 GMAC 寄存器概覽（基址是 0x1009_0000）

偏移地址	名稱	描述	頁碼
0x0000	STATION_ADDR_LOW	本機 MAC 地址低 32bit 寄存器	5-16
0x0004	STATION_ADDR_HIGH	本機 MAC 地址高 16bit 寄存器	5-16
0x000C	FD_FC_TYPE	流控幀類型寄存器	5-17
0x001C	FC_TX_TIMER	流控時間參數寄存器	5-17
0x0020	FD_FC_ADDR_LOW	流控幀目的地址低 32bit 寄存器	5-18
0x0024	FD_FC_ADDR_HIGH	流控幀目的地址高 16bit 寄存器	5-18
0x0030	IPG_TX_TIMER	發送幀間隙寄存器	5-19
0x0038	PAUSE_THR	發送流控幀間隙寄存器	5-19
0x003C	MAX_FRM_SIZE	最大幀長寄存器	5-20
0x0040	PORT_MODE	端口狀態寄存器	5-21
0x0044	PORT_EN	通道使能寄存器	5-21
0x0048	PAUSE_EN	流控使能寄存器	5-22
0x0050	SHORT_RUNTS_THR	超短幀界限寄存器	5-22
0x0054	DROP_UNK_CTL_FRM	未知控制幀丟棄使能寄存器	5-23
0x0060	TRANSMIT_CONTROL	發送配置寄存器	5-23
0x0064	REC_FILT_CONTROL	接收幀過濾控制寄存器	5-24



偏移地址	名称	描述	页码
0x0068	PORT_MC_ADDR_LOW	多播地址低 32bit 寄存器	5-25
0x006C	PORT_MC_ADDR_HIGH	多播地址高 16bit 寄存器	5-25
0x0080	RX_OCTETS_TOTAL_OK	接收有效帧的字节统计寄存器	5-26
0x0084	RX_OCTETS_BAD	接收错误帧的字节统计寄存器	5-26
0x0088	RX_UC_PKTS	MAC 接收单播帧数统计寄存器	5-26
0x008C	RX_MC_PKTS	接收多播帧数统计寄存器	5-27
0x0090	RX_BC_PKTS	接收的广播帧数统计寄存器	5-27
0x0094	RX_PKTS_64OCTETS	接收帧长为 64byte 的帧数统计寄存器	5-28
0x0098	RX_PKTS_65TO127OCTETS	接收帧长为 65byte~127byte 的帧数统计寄存器	5-28
0x009C	RX_PKTS_128TO255OCTETS	接收帧长为 128byte~255byte 的帧数统计寄存器	5-28
0x00A0	RX_PKTS_256TO511OCTETS	接收到帧长为 256byte~511byte 的帧数统计寄存器	5-29
0x00A4	RX_PKTS_512TO1023OCTETS	接收到帧长为 512byte~1023byte 的帧数统计寄存器	5-29
0x00A8	RX_PKTS_1024TO1518OCTETS	接收帧长为 1024byte~1518byte 的帧数统计寄存器	5-30
0x00AC	RX_PKTS_1519TO_MAXOCTETS	接收到帧长为大于等于 1519byte 的帧数统计寄存器	5-30
0x00B0	RX_FCS_ERRORS	接收 CRC 检验错误的帧数统计寄存器	5-30
0x00B4	RX_TAGGED	接收带 TAG 的帧数统计寄存器	5-31
0x00B8	RX_DATA_ERR	接收数据错误帧数统计寄存器	5-31
0x00BC	RX_ALIGN_ERROR	接收数据非字节对齐错误帧数统计寄存器	5-32
0x00C0	RX_LONG_ERROR	接收帧长大于允许的最大长度的帧数 (CRC 正确) 统计寄存器	5-32
0x00C4	RX_JABBER_ERRORS	接收帧长大于允许的最大长度的帧数 (CRC 错误或 byte 数不是整数) 统计寄存器	5-32



偏移地址	名称	描述	页码
0x00C8	RX_PAUSE_MACCONTROL_FRAMCOUNTER	接收流控帧统计寄存器	5-33
0x00CC	RX_UNKNOWN_MACCONTROL_FRAMCOUNTER	MAC 接收非流控控制帧数统计寄存器	5-33
0x00D0	RX_VERY_LONG_ERR_CNT	接收超长帧数统计寄存器	5-34
0x00D4	RX_RUNT_ERR_CNT	接收帧长小于 64byte 大于等于 12byte 的帧数统计寄存器	5-34
0x00D8	RX_SHORT_ERR_CNT	接收帧长小于 96bit 的帧数统计寄存器	5-35
0x0100	OCTETS_TRANSMITTED_OK	发送成功的好包字节数统计寄存器	5-35
0x0104	OCTETS_TRANSMITTED_BAD	发送成功的坏包字节数统计寄存器	5-35
0x0108	TX_UC_PKTS	发送的单播帧数统计寄存器	5-36
0x010C	TX_MC_PKTS	发送的多播帧数统计寄存器	5-36
0x0110	TX_BC_PKTS	发送的广播帧数统计寄存器	5-37
0x0114	TX_PKTS_64OCTETS	发送帧长为 64byte 的帧数统计寄存器	5-37
0x0118	TX_PKTS_65TO127OCTETS	发送帧长为 65byte~127byte 的帧数统计寄存器	5-37
0x011C	TX_PKTS_128TO255OCTETS	发送帧长为 128byte~255byte 的帧数统计寄存器	5-38
0x0120	TX_PKTS_256TO511OCTETS	发送帧长为 256byte~511byte 的帧数统计寄存器	5-38
0x0124	TX_PKTS_512TO1023OCTETS	发送帧长为 512byte~1023byte 的帧数统计寄存器	5-38
0x0128	TX_PKTS_1024TO1518OCTETS	发送帧长为 1024byte~1518byte 的帧数统计寄存器	5-39
0x012C	TX_PKTS_1519TO_MAXOCTETS	发送帧长大于 1519byte 的帧数统计寄存器	5-39
0x014C	TX_EXCESSIVE_LENGTH_DROP	超过设定的最大帧长导致发送失败次数统计寄存器	5-40
0x0150	TX_UNDERRUN	帧发送过程中发生内部错误而导致发送失败的次数统计寄存器	5-40



偏移地址	名称	描述	页码
0x0158	TX_CRC_ERROR	发送帧长正确 CRC 错误的帧数统计寄存器	5-40
0x015C	TX_PAUSE_FRAME_S	发送 PAUSE 帧数统计寄存器	5-41
0x01A8	LINE_LOOP_BACK	MAC 线路侧环回寄存器	5-41
0x01B0	CF_CRC_STRIP	CRC 剥离使能寄存器	5-42
0x01B4	MODE_CHANGE_EN	端口模式改变使能寄存器	5-43
0x01DC	LOOP_REG	环回配置寄存器	5-43
0x01E0	RECV_CONTROL	接收控制寄存器	5-44
0x01EC	RX_OVERRUN_CNT	FIFO 溢出统计寄存器	5-44
0x01F4	RX_LENGTHFIELD_ERR_CNT	含 PAD 帧统计寄存器	5-45
0x0200	MAC_SA_ADDR_L	源 MAC 地址低 32bit 寄存器	5-45
0x0204	MAC_SA_ADDR_H	源 MAC 地址高 16bit 寄存器	5-46
0x0208	MAC_DA_ADDR_L	目的 MAC 地址低 32bit 寄存器	5-46
0x020C	MAC_DA_ADDR_H	目的 MAC 地址高 16bit 寄存器	5-47
0x0214	CONTROL_WORD	控制寄存器	5-47
0x0218	FLOW_CTRL_PKG_THRSLD	限速包个数寄存器	5-49
0x021C	CRF_FLOW_TIME_THRSLD	限速时间寄存器	5-49
0x0220	FILTER_LIST0	过滤列表寄存器 0	5-50
0x0224	FILTER_LIST1	过滤列表寄存器 1	5-51
0x0228	FILTER_LIST2	过滤列表寄存器 2	5-51
0x022C	FILTER_LIST3	过滤列表寄存器 3	5-52
0x0230	FILTER_LIST4	过滤列表寄存器 4	5-53
0x0234	FILTER_LIST5	过滤列表寄存器 5	5-54
0x0238	FILTER_LIST6	过滤列表寄存器 6	5-55
0x023C	FILTER_LIST7	过滤列表寄存器 7	5-56
0x0240	FILTER_LIST8	过滤列表寄存器 8	5-57



偏移地址	名称	描述	页码
0x0244	FILTER_LIST9	过滤列表寄存器 9	5-58
0x0248	FILTER_LIST10	过滤列表寄存器 10	5-59
0x024C	FILTER_LIST11	过滤列表寄存器 11	5-60
0x0250	FILTER_LIST12	过滤列表寄存器 12	5-61
0x0254	FILTER_LIST13	过滤列表寄存器 13	5-62
0x0258	FILTER_LIST14	过滤列表寄存器 14	5-63
0x025C	FILTER_LIST15	过滤列表寄存器 15	5-64
0x0264	CRF_TX_FIFO_THRSLD	发送 FIFO 水位寄存器	5-65
0x0268	CRF_RX_FIFO_THRSLD	接收 FIFO 水位寄存器	5-66
0x026C	CRF_RX_BASE_ADDR	配置 FIFO 地址寄存器	5-66
0x0270	CRF_INTRPT_THRSLD	中断控制寄存器	5-67
0x0274	CRF_FIFO_CLR	FIFO 清零寄存器	5-68
0x027C	CRF_INTRPT_CLR	中断清除寄存器	5-69
0x0280	ERR_GIVEN_PKG_CNT	因 MAC 标记错误帧尾而被丢弃的包数统计寄存器	5-70
0x0288	ERR_FRM_TYPE_CNT	因帧类型不在过滤列表内而被丢弃的包数统计寄存器	5-70
0x028C	ERR_IP_TYPE_CNT	因 IP 类型不在过滤列表内而被丢弃的包数统计寄存器	5-70
0x0294	OVER_FLOW_CNT	因接收 FIFO 满而丢弃的包数统计寄存器	5-71
0x0298	OVER_LENGTH_CNT	因超过最大长度而丢弃的包数统计寄存器	5-71
0x02A8	CRF_CFF_DATA_NUM	配置 FIFO 数据个数寄存器	5-72
0x02AC	FLOW_OUT_IP_CNT	IP 报文限速丢弃的包数统计寄存器	5-72
0x02B0	FLOW_OUT_CTRL_CNT	控制报文限速丢弃的包数统计寄存器	5-72
0x02B4	IP_CHK_ERR_CNT	IP 头校验错而丢弃的包数统计寄存器	5-73



偏移地址	名称	描述	页码
0x02B8	UDP_CHK_ERR_CNT	UDP 校验错而丢弃的包数统计寄存器	5-73
0x02BC	TX_RUNT_ERR_PKG_CNT	上行因 FIFO 满而丢弃的包数统计寄存器	5-74
0x02C0	RX_ERR_TOTAL_CNT	下行总丢弃的包数统计寄存器	5-74
0x02C8	RX_TRANS_PKG_CNT	下行搬运成功的包数统计寄存器	5-74
0x02CC	TX_TRANS_PKG_CNT	上行搬运成功的包数统计寄存器	5-75
0x02D0	RX_BM_OVERFLOW	广播或多播报文因限速而丢弃的包数统计寄存器	5-75
0x02D4	TX_ADDR_OVERFLOW	CPU 写上行配置 FIFO 溢出寄存器	5-75
0x02D8	RX_ADDR_OVERFLOW	CPU 写下行配置 FIFO 溢出寄存器	5-76
0x0340	CRF_TX_PAUSE	发送流控控制寄存器	5-76
0x0344	CRF_RX_ADDR_NUM	限速下行地址个数寄存器	5-77
0x0348	CRF_CTRL_0_TYPE	控制报文类型 0 寄存器	5-77
0x034C	CRF_CTRL_1_TYPE	控制报文类型 1 寄存器	5-78
0x0350	CRF_CTRL_2_TYPE	控制报文类型 2 寄存器	5-78
0x0354	CRF_CTRL_3_TYPE	控制报文类型 3 寄存器	5-79
0x0358	CRF_BM_PKT_THRSLD	广播和多播报文的限速处理包个数寄存器	5-79
0x035C	CRF_BM_TIME_THRSLD	广播和多播报文的限速时间计数寄存器	5-80
0x03C0	MDIO_SINGLE_CMD	MDIO 单次操作寄存器	5-80
0x03C4	MDIO_SINGLE_DATA	MDIO 读写数据寄存器	5-81
0x03CC	MDIO_CTL	MDIO 控制寄存器	5-82
0x03D0	MDIO_RDATA_STATUS	MDIO 读数据状态寄存器	5-83
0x0400+m ×0x4	TX_CFF_LEN	发包队列包长配置寄存器	5-83



偏移地址	名称	描述	页码
0x0580+m ×0x4	TX_CFF_ADDR	发包队列地址配置寄存器	5-83
0x0480+n ×0x4	RX_CFF_ADDR	收包队列地址配置寄存器	5-84

GMAC 寄存器偏移地址中变量的取值范围和含义如表 5-4 所示。

表5-4 GMAC 寄存器偏移地址变量表

变量名称	取值范围	描述
m	0~31	发送配置 FIFO 的 32 个地址
n	0~31	接收配置 FIFO 的 32 个地址

5.7 GMAC 寄存器描述

STATION_ADDR_LOW

STATION_ADDR_LOW 为本机 MAC 地址低 32bit 寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0000	STATION_ADDR_LOW	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	station_addr_low		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	station_addr_low	MAC_CORE 的 MAC 源地址低 32bit。

STATION_ADDR_HIGH

STATION_ADDR_HIGH 为本机 MAC 地址高 16bit 寄存器。



注意

PMU 使用的源 MAC 地址 (MAC_SA_ADDR_H/MAC_SA_ADDR_L) 与 MAC 侧的源 MAC 地址 (STATION_ADDR_HIGH/STATION_ADDR_LOW) 二者必须同时配置, 保持一致。

	Offset Address								Register Name								Total Reset Value															
	0x0004								STATION_ADDR_HIGH								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																station_addr_high															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:16]	-	forbidden		禁止配置。																											
	[15:0]	RW	station_addr_high		MAC_CORE 的 MAC 源地址高 16bit。																											

FD_FC_TYPE

FD_FC_TYPE 为流控帧类型寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x000C								FD_FC_TYPE								0x0000_8808															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																fd_fc_type															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
	Bits	Access	Name		Description																											
	[31:16]	-	forbidden		禁止配置。																											
	[15:0]	RW	fd_fc_type		全双工模式下流控帧 TYPE 域, 推荐使用默认值。																											

FC_TX_TIMER

FC_TX_TIMER 为流控时间参数寄存器。



Offset Address		Register Name		Total Reset Value				
0x001C		FC_TX_TIMER		0x0000_00FF				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden				fc_tx_timer			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1
Bits	Access	Name	Description					
[31:16]	-	forbidden	禁止配置。					
[15:0]	RW	fc_tx_timer	发送流控帧时间参数，以 512bit 传输时间为时间单位。 百兆模式下，表示 128 个 TXC（25MHz）时钟周期； 千兆模式下，表示 64 个 TXC（125MHz）时钟周期。					

FD_FC_ADDR_LOW

FD_FC_ADDR_LOW 为流控帧目的地址低 32bit 寄存器。

Offset Address		Register Name		Total Reset Value				
0x0020		FD_FC_ADDR_LOW		0xC200_0001				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	fd_fc_addr_low							
Reset	1 1 0 0	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
Bits	Access	Name	Description					
[31:0]	RW	fd_fc_addr_low	流控帧 DA 地址低 32bit。 此处不允许用户修改，使用默认值。					

FD_FC_ADDR_HIGH

FD_FC_ADDR_HIGH 为流控帧目的地址高 16bit 寄存器。



Offset Address		Register Name		Total Reset Value				
0x0024		FD_FC_ADDR_HIGH		0x0000_0180				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden				fd_fc_addr_high			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	1 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	-	forbidden	禁止配置。					
[15:0]	RW	fd_fc_addr_high	流控帧 DA 地址高 16bit。此处不允许用户修改，使用默认值。					

IPG_TX_TIMER

IPG_TX_TIMER 为发送帧间隙寄存器。

Offset Address		Register Name		Total Reset Value				
0x0030		IPG_TX_TIMER		0x0000_0008				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden						ipg_tx_timer	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0
Bits	Access	Name	Description					
[31:8]	-	forbidden	禁止配置。					
[7:0]	RW	ipg_tx_timer	发送帧间隙，单位为 byte。 此处不允许用户修改，使用默认值。					

PAUSE_THR

PAUSE_THR 为发送流控帧间隙寄存器。



Offset Address		Register Name		Total Reset Value					
0x0038		PAUSE_THR		0x0000_002F					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				pause_thr				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	1 1 1 1	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	pause_thr	流控帧间隔时间，若流控时间大于间隔时间，MAC 自动发送流控帧，以 512bit 为时间单位。 百兆模式下，表示 128 个 TXC（25MHz）时钟周期； 千兆模式下，表示 64 个 TXC（125MHz）时钟周期。						

MAX_FRM_SIZE

MAX_FRM_SIZE 为最大帧长寄存器。



注意

max_frm_size 与 PMU 侧的最大帧长配置（CONTROL_WORD[crf_large_packet]）二者必须同时配置，保持一致。

Offset Address		Register Name		Total Reset Value					
0x003C		MAX_FRM_SIZE		0x0000_05EE					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				max_frm_size				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 1	1 1 1 0	1 1 1 0	
Bits	Access	Name	Description						
[31:14]	-	forbidden	禁止配置。						
[13:0]	RW	max_frm_size	MAC 部分允许的最大帧长，单位为 Byte。 当接收到大于设置帧长的帧时，将接收帧作为超长错误帧； 当发送帧大于设置帧长的帧时，将发送帧截断后作为错帧发送。						



PORT_MODE

PORT_MODE 为端口状态寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0040				PORT_MODE				0x0000_0001																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																									port_mode						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Bits	Access	Name		Description																											
	[31:3]	-	forbidden		禁止配置。																											
	[2:0]	RW	port_mode		当前 MAC 端口工作的模式。 000: 10Mbit/s MII; 001: 100Mbit/s MII; 010: 1000Mbit/s GMII; 011: 10Mbit/s RGMII; 100: 100Mbit/s RGMII; 101: 1000Mbit/s RGMII; 其他: 保留。																											

PORT_EN

PORT_EN 为通道使能寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0044				PORT_EN				0x0000_0006																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																									tx_en	rx_en	forbidden				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
	Bits	Access	Name		Description																											
	[31:3]	-	forbidden		禁止配置。																											



[2]	RW	tx_en	发送通道使能。 0: 禁止; 1: 使能。
[1]	RW	rx_en	接收通道使能。 0: 禁止; 1: 使能。
[0]	-	forbidden	禁止配置。

PAUSE_EN

PAUSE_EN 为流控使能寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0048				PAUSE_EN				0x0000_0007																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																								tx_fdfc	rx_fdfc						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Bits	Access		Name		Description																											
[31:2]	-		forbidden		禁止配置。																											
[1]	RW		tx_fdfc		全双工模式发送流控帧使能。 0: 禁止; 1: 使能。																											
[0]	RW		rx_fdfc		全双工模式响应流控帧使能。 0: 禁止; 1: 使能。																											

SHORT_RUNTS_THR

SHORT_RUNTS_THR 为超短帧界限寄存器。



Offset Address		Register Name		Total Reset Value					
0x0050		SHORT_RUNTS_THR		0x0000_000C					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden							short_runs_thr	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0	
Bits	Access	Name	Description						
[31:5]	-	forbidden	禁止配置。						
[4:0]	RW	short_runs_thr	短帧、超短帧界限（只用于统计），小于此配置值的为超短帧，以 byte 为单位。						

DROP_UNK_CTL_FRM

DROP_UNK_CTL_FRM 为未知控制帧丢弃使能寄存器。

Offset Address		Register Name		Total Reset Value				
0x0054		DROP_UNK_CTL_FRM		0x0000_0001				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden							drop_unk_ctl_frm
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
Bits	Access	Name	Description					
[31:1]	-	forbidden	禁止配置。					
[0]	RW	drop_unk_ctl_frm	未知控制帧处理。 0: 正常转发未知控制帧; 1: 丢弃未知控制帧。					

TRANSMIT_CONTROL

TRANSMIT_CONTROL 为发送配置寄存器。

Offset Address		Register Name		Total Reset Value										
0x0060		TRANSMIT_CONTROL		0x0000_00D0										
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0						
Name	forbidden						pad_enable	6	5	4	3	2	1	0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1	1	0	1	0	0	0	0
Bits	Access	Name	Description											
[31:8]	-	forbidden	禁止配置。											
[7]	RW	pad_enable	发送添加 PAD 使能。 0: 禁止; 1: 使能。 说明: 发送添加 PAD 时, 逻辑会自动重新计算 FCS。											
[6]	RW	crc_add	发送添加 FCS 使能。 0: 禁止; 1: 使能。											
[5:0]	-	forbidden	禁止配置。											

REC_FILT_CONTROL

REC_FILT_CONTROL 为接收帧过滤控制寄存器。

Offset Address		Register Name		Total Reset Value								
0x0064		REC_FILT_CONTROL		0x0000_0000								
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0				
Name	forbidden						5	4	3	2	1	0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	0	0	0	0
Bits	Access	Name	Description									
[31:6]	-	forbidden	禁止配置。									
[5]	RW	crc_err_pass	是否过滤 CRC 错误帧。 0: 否; 1: 是。									



[4:3]	-	forbidden	禁止配置。
[2]	RW	bc_drop_en	是否过滤广播帧。 0: 否; 1: 是。
[1]	RW	mc_match_en	是否过滤 DA 不匹配的多播帧。 0: 否; 1: 是。
[0]	RW	uc_match_en	是否过滤 DA 不匹配的单播帧。 0: 否; 1: 是。

PORT_MC_ADDR_LOW

PORT_MC_ADDR_LOW 为多播地址低 32bit 寄存器。

Offset Address		Register Name		Total Reset Value				
0x0068		PORT_MC_ADDR_LOW		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	port_mc_addr_low							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	port_mc_addr_low	多播地址低 32bit, 用于判断多播帧是否匹配。					

PORT_MC_ADDR_HIGH

PORT_MC_ADDR_HIGH 为多播地址高 16bit 寄存器。

Offset Address		Register Name		Total Reset Value				
0x006C		PORT_MC_ADDR_HIGH		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden				port_mc_addr_high			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	-	forbidden	禁止配置。					



[15:0]	RW	port_mc_addr_high	多播地址高 16bit，用于判断多播帧是否匹配。
--------	----	-------------------	--------------------------

RX_OCTETS_TOTAL_OK

RX_OCTETS_TOTAL_OK 为接收有效帧的字节统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0080	RX_OCTETS_TOTAL_OK	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Name	rx_octets_total_ok		
Reset	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0
Bits	Access	Name	Description
[31:0]	RC	rx_octets_total_ok	接收有效帧的字节统计。

RX_OCTETS_BAD

RX_OCTETS_BAD 为接收错误帧的字节统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0084	RX_OCTETS_BAD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Name	rx_octets_bad		
Reset	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0
Bits	Access	Name	Description
[31:0]	RC	rx_octets_bad	接收错帧的字节统计，包括 CRC 错误及对齐错误等。

RX_UC_PKTS

RX_UC_PKTS 为 MAC 接收单播帧数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x0088		RX_UC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_uc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_uc_pkts	接收单播帧的帧数统计，不包括 bad 帧。					

RX_MC_PKTS

RX_MC_PKTS 为接收多播帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x008C		RX_MC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_mc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_mc_pkts	接收多播帧的帧数统计，不包括 bad 帧。					

RX_BC_PKTS

RX_BC_PKTS 为接收的广播帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0090		RX_BC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_bc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_bc_pkts	接收广播帧的帧数统计，不包括 bad 帧。					



RX_PKTS_64OCTETS

RX_PKTS_64OCTETS 为接收帧长为 64byte 的帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0094				RX_PKTS_64OCTETS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_pkts_64octets																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_pkts_64octets		接收帧长为 64byte 的帧数统计，包括 bad 帧。																											

RX_PKTS_65TO127OCTETS

RX_PKTS_65TO127OCTETS 为接收帧长为 65byte~127byte 的帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0098				RX_PKTS_65TO127OCTETS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_pkts_65to127octets																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_pkts_65to127octets		接收帧长为 65byte~127byte 的帧数统计，包括 bad 帧。																											

RX_PKTS_128TO255OCTETS

RX_PKTS_128TO255OCTETS 为接收帧长为 128byte~255byte 的帧数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x009C		RX_PKTS_128TO255OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_pkts_128to255octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_pkts_128to255octets	接收帧长为 128byte~255byte 的帧数统计，包括 bad 帧。					

RX_PKTS_256TO511OCTETS

RX_PKTS_256TO511OCTETS 为接收到帧长为 256byte~511byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x00A0		RX_PKTS_255TO511OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_pkts_256to511octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_pkts_256to511octets	接收帧长为 256byte~511byte 的帧数统计，包括 bad 帧。					

RX_PKTS_512TO1023OCTETS

RX_PKTS_512TO1023OCTETS 为接收到帧长为 512byte~1023byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x00A4		RX_PKTS_512TO1023OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_pkts_512to1023octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_pkts_512to1023octets	接收帧长为 512byte~1023byte 的帧数统计，包括 bad 帧。					



RX_PKTS_1024TO1518OCTETS

RX_PKTS_1024TO1518OCTETS 为接收帧长为 1024byte~1518byte 的帧数统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x00A8	RX_PKTS_1024TO1518OCTETS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_pkts_1024to1518octets		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RC	rx_pkts_1024to1518octets
	Description		
	接收帧长为 1024byte~1518byte 的帧数统计，包括 bad 帧。		

RX_PKTS_1519TOMAXOCTETS

RX_PKTS_1519TOMAXOCTETS 为接收到帧长为大于等于 1519byte 的帧数统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x00AC	RX_PKTS_1519TOMAXOCTETS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_pkts_1519tomaxoctes		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RC	rx_pkts_1519tomaxoctes
	Description		
	接收帧长为 1519byte~最大 byte 的帧数统计，包括 bad 帧。		

RX_FCS_ERRORS

RX_FCS_ERRORS 为接收 CRC 检验错误的帧数统计寄存器。



	Offset Address				Register Name				Total Reset Value																							
	0x00B0				RX_FCS_ERRORS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_fcs_errors																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_fcs_errors		CRC 检测错误的帧数统计，不包括短帧。																											

RX_TAGGED

RX_TAGGED 为接收带 TAG 的帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x00B4				RX_TAGGED				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_tagged																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_tagged		接收带 TAG 的帧数统计。																											

RX_DATA_ERR

RX_DATA_ERR 为接收数据错误帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x00B8				RX_DATA_ERR				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_data_err																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_data_err		接收数据错误帧数统计。																											



RX_ALIGN_ERRORS

RX_ALIGN_ERRORS 为接收数据非字节对齐错误帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x00BC				RX_ALIGN_ERRORS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_align_errors																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_align_errors		接收数据非字节对齐错误帧数统计。																											

RX_LONG_ERRORS

RX_LONG_ERRORS 为接收帧长大于允许的最大长度的帧数（CRC 正确）统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x00C0				RX_LONG_ERRORS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_long_errors																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_long_errors		帧长大于允许的最大长度的帧数统计，CRC 正确。																											

RX_JABBER_ERRORS

RX_JABBER_ERRORS 为接收帧长大于允许的最大长度的帧数（CRC 错误或 byte 数不是整数）统计寄存器。



	Offset Address	Register Name	Total Reset Value
	0x00C4	RX_JABBER_ERRORS	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_jabber_errors		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
	Bits	Access	Name
	[31:0]	RC	rx_jabber_errors
	Description		
	帧长大于允许的最大长度的帧统计，CRC 错误或 byte 数不是整数（奇数个 nibble）。		

RX_PAUSE_MACCONTROL_FRAMCOUNTER

RX_PAUSE_MACCONTROL_FRAMCOUNTER 为接收流控帧统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x00C8	RX_PAUSE_MACCONTROL_FRAMCOUNTER	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_pause_macontrol_framecounter		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
	Bits	Access	Name
	[31:0]	RC	rx_pause_macontrol_framecounter
	Description		
	接收流控帧数统计（当 CRC 检测有效时，需 CRC 正确）。		

RX_UNKNOWN_MACCONTROL_FRAMCOUNTER

RX_UNKNOWN_MACCONTROL_FRAMCOUNTER 为 MAC 接收非流控控制帧数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x00CC		RX_UNKNOWN_MACCONTROL_FRAMECOUNTER		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_unknown_maccontrol_framecounter							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_unknown_maccontrol_framecounter	接收非流控控制帧数统计。					

RX_VERY_LONG_ERR_CNT

RX_VERY_LONG_ERR_CNT 为接收超长帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x00D0		RX_VERY_LONG_ERR_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_very_long_err_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_very_long_err_cnt	接收超长帧（长度大于最大帧长的 2 倍）的帧数统计。					

RX_RUNT_ERR_CNT

RX_RUNT_ERR_CNT 为接收帧长小于 64byte 大于等于 12byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x00D4		RX_RUNT_ERR_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_runt_err_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_runt_err_cnt	帧长小于 64byte，但大于或等于 12byte 的帧数统计。					



RX_SHORT_ERR_CNT

RX_SHORT_ERR_CNT 为接收帧长小于 96bit 的帧数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x00D8				RX_SHORT_ERR_CNT				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_short_err_cnt																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	rx_short_err_cnt		帧长小于 12byte 的帧数统计。																											

OCTETS_TRANSMITTED_OK

OCTETS_TRANSMITTED_OK 为发送成功的好包字节数统计寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0100				OCTETS_TRANSMITTED_OK				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	octets_transmitted_ok																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RC	octets_transmitted_ok		发送成功的好包字节数统计。																											

OCTETS_TRANSMITTED_BAD

OCTETS_TRANSMITTED_BAD 为发送成功的坏包字节数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x0104		OCTETS_TRANSMITTED_BAD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	octets_transmitted_bad							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	octets_transmitted_bad	发送成功的坏包的字节数统计。					

TX_UC_PKTS

TX_UC_PKTS 为发送的单播帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0108		TX_UC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_uc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_uc_pkts	发送的单播帧数统计，不包括坏包。					

TX_MC_PKTS

TX_MC_PKTS 为发送的多播帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x010C		TX_MC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_mc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_mc_pkts	发送的多播帧数统计，不包括坏包。					



TX_BC_PKTS

TX_BC_PKTS 为发送的广播帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0110		TX_BC_PKTS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_bc_pkts							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_bc_pkts	发送的广播帧数统计，不包括坏包。					

TX_PKTS_64OCTETS

TX_PKTS_64OCTETS 为发送帧长为 64byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0114		TX_PKTS_64OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_64octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_64octets	发送的帧长为 64byte 的帧数统计，包括坏包。					

TX_PKTS_65TO127OCTETS

TX_PKTS_65TO127OCTETS 为发送帧长为 65byte~127byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0118		TX_PKTS_65TO127OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_65to127octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_65to127octets	发送帧长为 65byte~127byte 的帧数统计，包括坏包。					



TX_PKTS_128TO255OCTETS

TX_PKTS_128TO255OCTETS 为发送帧长为 128byte~255byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x011C		TX_PKTS_128TO255OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_128to255octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_128to255octets	发送帧长为 128byte~255byte 的帧数统计，包括坏包。					

TX_PKTS_256TO511OCTETS

TX_PKTS_256TO511OCTETS 为发送帧长为 256byte~511byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0120		TX_PKTS_256TO511OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_256to511octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_256to511octets	发送帧长为 256byte~511byte 的帧数统计，包括坏包。					

TX_PKTS_512TO1023OCTETS

TX_PKTS_512TO1023OCTETS 为发送帧长为 512byte~1023byte 的帧数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x0124		TX_PKTS_512TO1023OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_512to1023octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_512to1023octets	发送帧长为 512byte~1023byte 的帧数统计，包括坏包。					

TX_PKTS_1024TO1518OCTETS

TX_PKTS_1024TO1518OCTETS 为发送帧长为 1024byte~1518byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0128		TX_PKTS_1024TO1518OCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_1024to1518octets							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_1024to1518octets	发送帧长为 1024byte~1518byte 的帧数统计，包括坏包。					

TX_PKTS_1519TOMAXOCTETS

TX_PKTS_1519TOMAXOCTETS 为发送帧长大于 1519byte 的帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x012C		TX_PKTS_1519TOMAXOCTETS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pkts_1519tomaxoctes							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pkts_1519tomaxoctes	发送帧长大于等于 1519byte 的帧数统计，包括坏包。					



TX_EXCESSIVE_LENGTH_DROP

TX_EXCESSIVE_LENGTH_DROP 为超过设定的最大帧长导致发送失败次数统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x014C	TX_EXCESSIVE_LENGTH_DROP	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	tx_excessive_length_drop		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
	Bits	Access	Name
	[31:0]	RC	tx_excessive_length_drop
	Description		
	由于超过设定的最大帧长而导致发送失败的次数统计。		

TX_UNDERRUN

TX_UNDERRUN 为帧发送过程中发生内部错误而导致发送失败的次数统计寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0150	TX_UNDERRUN	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	tx_underrun		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
	Bits	Access	Name
	[31:0]	RC	tx_underrun
	Description		
	帧发送过程中发生内部错误而导致发送失败的次数统计。		

TX_CRC_ERROR

TX_CRC_ERROR 为发送帧长正确 CRC 错误的帧数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x0158		TX_CRC_ERROR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_crc_error							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_crc_error	发送帧长正确而 CRC 错误的帧数目统计。					

TX_PAUSE_FRAMES

TX_PAUSE_FRAMES 为发送 PAUSE 帧数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x015C		TX_PAUSE_FRAMES		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_pause_frames							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_pause_frames	发送的 PAUSE 帧数统计。					

LINE_LOOP_BACK

LINE_LOOP_BACK 为 MAC 线路侧环回寄存器。



Offset Address		Register Name		Total Reset Value					
0x01A8		LINE_LOOP_BACK		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden								line_loop_back
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	forbidden	禁止配置。						
[0]	RW	line_loop_back	MAC 线路侧环回使能。 0: 禁止; 1: 使能。						

CF_CRC_STRIP

CF_CRC_STRIP 为 CRC 剥离使能寄存器。

Offset Address		Register Name		Total Reset Value					
0x01B0		CF_CRC_STRIP		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden								cf_crc_strip
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:1]	-	forbidden	禁止配置。						
[0]	RW	cf_crc_strip	MAC 剥离接收方向 CRC 使能。 0: 禁止, 上报包长包括 CRC 的 4byte; 1: 使能, 剥离后上报包长不包括 CRC 的 4byte。						



MODE_CHANGE_EN

MODE_CHANGE_EN 为端口模式改变使能寄存器。

	Offset Address								Register Name								Total Reset Value																			
	0x01B4								MODE_CHANGE_EN								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	forbidden																														mode_change_en					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Bits	Access		Name		Description																															
[31:1]	-		forbidden		禁止配置。																															
[0]	RW		mode_change_en		端口模式改变使能。 0: 禁止; 1: 使能。																															

LOOP_REG

LOOP_REG 为环回配置寄存器。

	Offset Address								Register Name								Total Reset Value																			
	0x01DC								LOOP_REG								0x0000_0002																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	forbidden																														cf2mi_lp_en	cf_ext_drive	lp	forbidden		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0				
Bits	Access		Name		Description																															
[31:3]	-		forbidden		禁止配置。																															
[2]	RW		cf2mi_lp_en		应用侧环回使能。 0: 禁止; 1: 使能。 用于测试模式。																															



[1]	RW	cf_ext_drive_lp	线路侧环回时，MAC 数据的读写使能信号产生方式。 0：由 MAC 内部产生； 1：由 MAC 外部产生。
[0]	-	forbidden	禁止配置。

RECV_CONTROL

RECV_CONTROL 为接收控制寄存器。

	Offset Address	Register Name	Total Reset Value							
	0x01E0	RECV_CONTROL	0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20							
			19 18 17 16							
			15 14 13 12							
			11 10 9 8							
			7 6 5 4							
			3 2 1 0							
Name	forbidden							runt_pkt_en	strip_pad_en	forbidden
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:5]	-	forbidden	禁止配置。							
[4]	RW	runt_pkt_en	接收超短帧透传使能。 0：禁止（丢弃小于 19byte 的帧，不上传给软件）； 1：使能（将小于 19byte 的帧上传给软件）。							
[3]	RW	strip_pad_en	剥离接收帧的 PAD 使能。 0：禁止； 1：使能。							
[2:0]	-	forbidden	禁止配置。							

RX_OVERRUN_CNT

RX_OVERRUN_CNT 为 FIFO 溢出统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x01EC		RX_OVERRUN_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_overnun_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_overnun_cnt	MAC 应用侧接收 FIFO 溢出统计。					

RX_LENGTHFIELD_ERR_CNT

RX_LENGTHFIELD_ERR_CNT 为含 PAD 帧统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x01F4		RX_LENGTHFIELD_ERR_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_lengthfield_err_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_lengthfield_err_cnt	MAC 剥离 PAD 使能有效时，接收到含 PAD 的帧但其帧长不等于 64 的个数统计。					

MAC_SA_ADDR_L

MAC_SA_ADDR_L 为源 MAC 地址低 32bit 寄存器。

Offset Address		Register Name		Total Reset Value				
0x0200		MAC_SA_ADDR_L		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	mac_sa_addr_l							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	mac_sa_addr_l	PMU 使用的配置的本机 MAC 地址的低 32bit，用来填充 SA 域的低 32bit。					



MAC_SA_ADDR_H

MAC_SA_ADDR_H 为源 MAC 地址高 16bit 寄存器。



注意

PMU 使用的源 MAC 地址 (MAC_SA_ADDR_H/MAC_SA_ADDR_L) 与 MAC 侧的源 MAC 地址 (STATION_ADDR_HIGH/STATION_ADDR_LOW) 二者必须同时配置, 保持一致。

	Offset Address		Register Name		Total Reset Value			
	0x0204		MAC_SA_ADDR_H		0x0000_0000			
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden				mac_sa_addr_h			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	Bits	Access	Name	Description				
	[31:16]	-	forbidden	禁止配置。				
	[15:0]	RW	mac_sa_addr_h	PMU 使用的本机 MAC 地址的高 16bit。				

MAC_DA_ADDR_L

MAC_DA_ADDR_L 为目的 MAC 地址低 32bit 寄存器。

	Offset Address		Register Name		Total Reset Value			
	0x0208		MAC_DA_ADDR_L		0x0000_0000			
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	mac_da_addr_l							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	Bits	Access	Name	Description				
	[31:0]	RW	mac_da_addr_l	PMU 使用的配置的目的 MAC 地址的低 32bit, 用来填充 DA 域的低 32bit。				



MAC_DA_ADDR_H

MAC_DA_ADDR_H 为目的 MAC 地址高 16bit 寄存器。

Offset Address		Register Name		Total Reset Value					
0x020C		MAC_DA_ADDR_H		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				mac_da_addr_h				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	mac_da_addr_h	PMU 使用的目的 MAC 地址的高 16bit。						

CONTROL_WORD

CONTROL_WORD 为控制寄存器。



注意

crf_large_packet 与 MAC 侧的最大帧长配置 (MAX_FRM_SIZE 最大帧长寄存器) 二者必须同时配置, 保持一致。

Offset Address		Register Name		Total Reset Value					
0x0214		CONTROL_WORD		0x0C00_0640					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden	crf_ip_head_chk_en crf_udp_sum_chk_e	crf_tx_standard forbidden	crf_insert_0_byte crf_remove_0_byte crf_ip_flow_ctrl crf_ctrl_flow_ctrl	forbidden	crf_filt_unused_pkg crf_bm_flow_ctrl crf_peel_dsa crf_add_da_sa	crf_large_packet		
Reset	0 0 0 0	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0	0 1 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	forbidden	禁止配置。						



[27]	RW	crf_ip_head_chk_en	IP 头校验使能。 0: 禁止; 1: 使能。
[26]	RW	crf_udp_sum_chk_en	UDP 校验和使能。 0: 禁止; 1: 使能。
[25]	RW	crf_tx_standard	发送 FIFO 的发送水线设置标准。 0: 按包或按字节设置。当发送 FIFO 中有一个完整的包, 或者发送 FIFO 中的有效数据个数大于等于 4 倍的发送水线时, 即向 MAC 发送读请求; 1: 按包设置。当发送 FIFO 中有一个完整的包时, 才向 MAC 发送读请求。 推荐配置为 1。
[24]	-	forbidden	禁止配置。
[23]	RW	crf_insert_0_byte	是否在帧头填充 2byte 的 0。 0: 是; 1: 否。 用于将接收到的数据进行字节对齐, 建议分配存储地址时按 4 字节对齐, 此处设置为 0, 可以提高协议栈处理效率。
[22]	RW	crf_remove_0_byte	是否在帧头去除 2byte 的 0。 0: 是; 1: 否。 用于从非 4 字节对齐的位置读取数据发送。建议配置的发包地址按 4 字节对齐, 此处配置为 0, 可以提高协议栈处理效率。
[21]	RW	crf_ip_flow_ctrl	IP 报文是否限速。 0: 否; 1: 是。
[20]	RW	crf_ctrl_flow_ctrl	控制报文限速使能。 0: 禁止; 1: 使能。
[19:18]	-	forbidden	禁止配置。
[17]	RW	crf_filt_unused_pkt	是否过滤非法报文。 0: 否; 1: 是。



[16]	RW	crf_bm_flow_ctrl	对广播或多播报文是否限速。 0: 否; 1: 是。
[15]	RW	crf_peel_dsa	是否剥离 DA/SA。 0: 否; 1: 是。
[14]	RW	crf_add_da_sa	是否添加 DA/SA。 0: 否; 1: 是。
[13:0]	RW	crf_large_packet	配置的最大包长度, 默认为 1600byte (PMU 使用的最大包长)。

FLOW_CTRL_PKG_THRSLD

FLOW_CTRL_PKG_THRSLD 为限速包个数寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0218				FLOW_CTRL_PKG_THRSLD				0xFFFF_FFFF																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	crf_ip_pkg_thrslid								crf_ctrl_pkg_thrslid																							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bits	Access		Name		Description																											
[31:16]	RW		crf_ip_pkg_thrslid		IP 报文的包个数上限, 当在 T 时间内接收的 IP 报文超过设置值, 则限速, 否则不限速。																											
[15:0]	RW		crf_ctrl_pkg_thrslid		控制报文的包个数上限, 当在 T 时间内接收的控制报文超过设置值, 则限速, 否则不限速。																											

CRF_FLOW_TIME_THRSLD

CRF_FLOW_TIME_THRSLD 为限速时间寄存器。



Offset Address		Register Name		Total Reset Value					
0x021C		CRF_FLOW_TIME_THRSLD		0x0000_00FF					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden						crf_flow_time_thrsld		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1	
Bits	Access	Name	Description						
[31:8]	-	forbidden	禁止配置。						
[7:0]	RW	crf_flow_time_thrsld	限速时间，以 125μs 为单位。 限速时间 T= (crf_flow_time_thrsld+1) × 125μs						

FILTER_LIST0

FILTER_LIST0 为过滤列表寄存器 0。

Offset Address		Register Name		Total Reset Value				
0x0220		FILTER_LIST0		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden			crf_filt_cfg0	crf_filt_id0	crf_filt_frm_type0		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:18]	-	forbidden	禁止配置。					
[17]	RW	crf_filt_cfg0	第 0 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析，否则不用考虑本表内容。					
[16]	RW	crf_filt_id0	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。					



[15:0]	RW	crf_filt_frm_type0	<p>过滤类型。</p> <p>当 crf_filt_id0=0 时，配置 16bit 都有效，与 Ethernet Type 域进行比较；</p> <p>当 crf_filt_id0=1 时，配置低 8bit 有效，与 IP PROTOCOL 域进行比较。</p>
--------	----	--------------------	---

FILTER_LIST1

FILTER_LIST1 为过滤列表寄存器 1。

	Offset Address	Register Name	Total Reset Value
	0x0224	FILTER_LIST1	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	forbidden		crf_filt_cfg1 crf_filt_id1 crf_filt_frm_type1
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
Bits	Access	Name	Description
[31:18]	-	forbidden	禁止配置。
[17]	RW	crf_filt_cfg1	<p>第 1 行是否已经配置。</p> <p>0: 否；</p> <p>1: 是。</p> <p>只有配置为 1 时才需要对该表内容进行解析，否则不用考虑本表内容。</p>
[16]	RW	crf_filt_id1	<p>过滤方式选择。</p> <p>0: 针对 Ethernet Type 域进行过滤；</p> <p>1: 针对 IP PROTOCOL 域进行过滤。</p>
[15:0]	RW	crf_filt_frm_type1	<p>过滤类型。</p> <p>当 crf_filt_id1=0 时，配置 16bit 都有效，与 Ethernet Type 域进行比较；</p> <p>当 crf_filt_id1=1 时，配置低 8bit 有效，与 IP PROTOCOL 域进行比较。</p>

FILTER_LIST2

FILTER_LIST2 为过滤列表寄存器 2。



Offset Address		Register Name		Total Reset Value																												
0x0228		FILTER_LIST2		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden												crf_filt_cfg2	crf_filt_id2	crf_filt_frm_type2																	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg2	第 2 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id2	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type2	过滤类型。 当 crf_filt_id2=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id2=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST3

FILTER_LIST3 为过滤列表寄存器 3。



	Offset Address 0x022C								Register Name FILTER_LIST3								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	forbidden																crf_filt_cfg3		crf_filt_id3		crf_filt_frm_type3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																																	
[31:18]	-	forbidden	禁止配置。																																	
[17]	RW	crf_filt_cfg3	第 3 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																																	
[16]	RW	crf_filt_id3	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																																	
[15:0]	RW	crf_filt_frm_type3	过滤类型。 当 crf_filt_id3=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id3=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																																	

FILTER_LIST4

FILTER_LIST4 为过滤列表寄存器 4。



Offset Address		Register Name		Total Reset Value																												
0x0230		FILTER_LIST4		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden												crf_filt_cfg4	crf_filt_id4	crf_filt_frm_type4																	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg4	第 4 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id4	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type4	过滤类型。 当 crf_filt_id4=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id4=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST5

FILTER_LIST5 为过滤列表寄存器 5。



	Offset Address 0x0234								Register Name FILTER_LIST5								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	forbidden																crf_filt_cfg5		crf_filt_id5		crf_filt_frm_type5															
Reset	0 0 0 0								0 0 0 0								0 0 0 0								0 0 0 0											
Bits	Access	Name	Description																																	
[31:18]	-	forbidden	禁止配置。																																	
[17]	RW	crf_filt_cfg5	第 5 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																																	
[16]	RW	crf_filt_id5	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																																	
[15:0]	RW	crf_filt_frm_type5	过滤类型。 当 crf_filt_id5=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id5=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																																	

FILTER_LIST6

FILTER_LIST6 为过滤列表寄存器 6。



Offset Address		Register Name		Total Reset Value																																
0x0238		FILTER_LIST6		0x0000_0000																																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	forbidden																crf_filt_cfg6		crf_filt_id6		crf_filt_frm_type6															
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0											
Bits	Access	Name	Description																																	
[31:18]	-	forbidden	禁止配置。																																	
[17]	RW	crf_filt_cfg6	第 6 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																																	
[16]	RW	crf_filt_id6	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																																	
[15:0]	RW	crf_filt_frm_type6	过滤类型。 当 crf_filt_id6=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id6=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																																	

FILTER_LIST7

FILTER_LIST7 为过滤列表寄存器 7。



Offset Address		Register Name		Total Reset Value																												
0x023C		FILTER_LIST7		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden														crf_filt_cfg7	crf_filt_id7	crf_filt_frm_type7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg7	第 7 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id7	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type7	过滤类型。 当 crf_filt_id7=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id7=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST8

FILTER_LIST8 为过滤列表寄存器 8。



Offset Address		Register Name		Total Reset Value					
0x0240		FILTER_LIST8		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_filt_cfg8	crf_filt_id8	crf_filt_frm_type8		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	forbidden	禁止配置。						
[17]	RW	crf_filt_cfg8	第 8 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。						
[16]	RW	crf_filt_id8	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。						
[15:0]	RW	crf_filt_frm_type8	过滤类型。 当 crf_filt_id8=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id8=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。						

FILTER_LIST9

FILTER_LIST9 为过滤列表寄存器 9。



Offset Address		Register Name		Total Reset Value																												
0x0244		FILTER_LIST9		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden														crf_filt_cfg9	crf_filt_id9	crf_filt_frm_type9															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg9	第 9 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id9	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type9	过滤类型。 当 crf_filt_id9=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id9=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST10

FILTER_LIST10 为过滤列表寄存器 10。



Offset Address		Register Name		Total Reset Value					
0x0248		FILTER_LIST10		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_filt_cfg10	crf_filt_id10	crf_filt_frm_type10		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	forbidden	禁止配置。						
[17]	RW	crf_filt_cfg10	第 10 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。						
[16]	RW	crf_filt_id10	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。						
[15:0]	RW	crf_filt_frm_type10	过滤类型。 当 crf_filt_id10=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id10=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。						

FILTER_LIST11

FILTER_LIST11 为过滤列表寄存器 11。



Offset Address		Register Name		Total Reset Value																												
0x024C		FILTER_LIST11		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden												crf_filt_cfg11	crf_filt_id11	crf_filt_frm_type11																	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg11	第 11 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id11	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type11	过滤类型。 当 crf_filt_id11=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id11=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST12

FILTER_LIST12 为过滤列表寄存器 12。



Offset Address		Register Name		Total Reset Value					
0x0250		FILTER_LIST12		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_filt_cfg12	crf_filt_id12	crf_filt_frm_type12		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	forbidden	禁止配置。						
[17]	RW	crf_filt_cfg12	第 12 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。						
[16]	RW	crf_filt_id12	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。						
[15:0]	RW	crf_filt_frm_type12	过滤类型。 当 crf_filt_id12=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id12=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。						

FILTER_LIST13

FILTER_LIST13 为过滤列表寄存器 13。



Offset Address		Register Name		Total Reset Value																												
0x0254		FILTER_LIST13		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden														crf_filt_cfg13	crf_filt_id13	crf_filt_frm_type13															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																												
[31:18]	-	forbidden		禁止配置。																												
[17]	RW	crf_filt_cfg13		第 13 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																												
[16]	RW	crf_filt_id13		过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																												
[15:0]	RW	crf_filt_frm_type13		过滤类型。 当 crf_filt_id13=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id13=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																												

FILTER_LIST14

FILTER_LIST14 为过滤列表寄存器 14。



Offset Address		Register Name		Total Reset Value																												
0x0258		FILTER_LIST14		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden								crf_filt_cfg14	crf_filt_id14	crf_filt_frm_type14																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:18]	-	forbidden	禁止配置。																													
[17]	RW	crf_filt_cfg14	第 14 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。																													
[16]	RW	crf_filt_id14	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。																													
[15:0]	RW	crf_filt_frm_type14	过滤类型。 当 crf_filt_id14=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id14=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。																													

FILTER_LIST15

FILTER_LIST15 为过滤列表寄存器 15。



Offset Address		Register Name		Total Reset Value					
0x025C		FILTER_LIST15		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden			crf_filt_cfg15	crf_filt_id15	crf_filt_frm_type15			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	forbidden	禁止配置。						
[17]	RW	crf_filt_cfg15	第 15 行是否已经配置。 0: 否; 1: 是。 只有配置为 1 时才需要对该表内容进行解析, 否则不用考虑本表内容。						
[16]	RW	crf_filt_id15	过滤方式选择。 0: 针对 Ethernet Type 域进行过滤; 1: 针对 IP PROTOCOL 域进行过滤。						
[15:0]	RW	crf_filt_frm_type15	过滤类型。 当 crf_filt_id15=0 时, 配置 16bit 都有效, 与 Ethernet Type 域进行比较; 当 crf_filt_id15=1 时, 配置低 8bit 有效, 与 IP PROTOCOL 域进行比较。						

CRF_TX_FIFO_THRSLD

CRF_TX_FIFO_THRSLD 为发送 FIFO 流水线寄存器。

Offset Address		Register Name		Total Reset Value				
0x0264		CRF_TX_FIFO_THRSLD		0x0271_017C				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	forbidden		crf_tx_p_full_th		forbidden		crf_tx_p_empty_th	
Reset	0 0 0 0	0 0 1 0	0 1 1 1	0 0 0 1	0 0 0 0	0 0 0 1	0 1 1 1	1 1 0 0
Bits	Access	Name	Description					
[31:26]	-	forbidden	禁止配置。					



[25:16]	RW	crf_tx_p_full_th	发送 FIFO 的高水位。 此处不允许配置。
[15:10]	-	forbidden	禁止配置。
[9:0]	RW	crf_tx_p_empty_th	发送 FIFO 的低水位。 此处不允许配置。

CRF_RX_FIFO_THRSLD

CRF_RX_FIFO_THRSLD 为接收 FIFO 水位寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0268	CRF_RX_FIFO_THRSLD	0x0640_0190
Bit	31 30 29 28	27 26 25 24	23 22 21 20
	19 18 17 16	15 14 13 12	11 10 9 8
	7 6 5 4	3 2 1 0	
Name	forbidden	crf_rx_p_full_th	forbidden
			crf_rx_p_empty_th
Reset	0 0 0 0	0 1 1 0	0 1 0 0
			0 0 0 0
			0 0 0 0
			0 0 0 1
			1 0 0 1
			0 0 0 0
			0 0 0 0
Bits	Access	Name	Description
[31:26]	-	forbidden	禁止配置。
[25:16]	RW	crf_rx_p_full_th	接收 FIFO 的高水位。 此处不允许配置。
[15:10]	-	forbidden	禁止配置。
[9:0]	RW	crf_rx_p_empty_th	接收 FIFO 的低水位。此处不允许配置。

CRF_RX_BASE_ADDR

CRF_RX_BASE_ADDR 为配置 FIFO 地址寄存器。

	Offset Address	Register Name	Total Reset Value
	0x026C	CRF_RX_BASE_ADDR	0x0000_0100
Bit	31 30 29 28	27 26 25 24	23 22 21 20
	19 18 17 16	15 14 13 12	11 10 9 8
	7 6 5 4	3 2 1 0	
Name	forbidden		
	crf_rx_base_addr		
Reset	0 0 0 0	0 0 0 0	0 0 0 0
			0 0 0 0
			0 0 0 0
			0 0 0 1
			0 0 0 0
			0 0 0 0
Bits	Access	Name	Description
[31:9]	-	forbidden	禁止配置。



[8:0]	RW	crf_rx_base_addr	接收配置 FIFO 的空间大小。 发送配置 FIFO 空间=512-crf_rx_base_addr。
-------	----	------------------	--

CRF_INTRPT_THRSLD

CRF_INTRPT_THRSLD 为中断控制寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0270				CRF_INTRPT_THRSLD				0x001E_000F																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden				crf_rxint_en				crf_txint_en				cfg_fifoemp_intrpt_en				crf_pkg_intrpt_en				crf_intrpt_pkt				crf_time_intrpt_en				crf_intrpt_time			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Bits	Access		Name		Description																											
[31:30]	-		forbidden		禁止配置。																											
[29]	RW		crf_rxint_en		收包中断使能。 0: 禁止; 1: 使能。																											
[28]	RW		crf_txint_en		发包中断使能。 0: 禁止; 1: 使能。																											
[27]	RW		cfg_fifoemp_intrpt_en		发包 FIFO 空中断使能。 0: 禁止; 1: 使能。																											
[26]	RW		crf_pkg_intrpt_en		包个数中断使能。 0: 禁止; 1: 使能。																											
[25:17]	RW		crf_intrpt_pkt		中断包个数。当接收的包个数大于等于配置值时，产生中断信号。																											



[16]	RW	crf_time_intrpt_en	超时中断使能。 0: 禁止; 1: 使能。
[15:0]	RW	crf_intrpt_time	配置的中断时间, 以 125μs 为单位进行计数。当接收时间超过配置值时, 产生中断信号。

CRF_FIFO_CLR

CRF_FIFO_CLR 为 FIFO 清零寄存器。

Offset Address		Register Name		Total Reset Value																												
0x0274		CRF_FIFO_CLR		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																crf_tx_st_rst	crf_cff_clr	crf_rff_clr	crf_tff_clr												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name	Description																													
[31:4]	-	forbidden	禁止配置。																													
[3]	RW	crf_tx_st_rst	发送状态机复位。 0: 不复位; 1: 复位。																													
[2]	RW	crf_cff_clr	配置 FIFO 清零。 0: 不清零; 1: 清零。 注意: 若配置为 1, 配置 FIFO 中的发送和接收地址会被清 0, 需要重新配置接收地址, 建议不要对配置 FIFO 清 0。																													
[1]	RW	crf_rff_clr	接收 FIFO 清零。 0: 不清零; 1: 清零。																													
[0]	RW	crf_tff_clr	发送 FIFO 清零。 0: 不清零; 1: 清零。																													



CRF_INTRPT_CLR

CRF_INTRPT_CLR 为中断清除寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x027C				CRF_INTRPT_CLR				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden								rx_crf_intrpt	tx_crf_intrpt	cfg_fifo_emp_intrpt	forbbiden								rx_intrpt_sta	tx_intrpt_sta	cfg_fifoemp_intrpt_sta										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:19]	-	forbidden	禁止配置。																													
[18]	RW	rx_crf_intrpt	收包原始中断状态，写 1 清中断。 0: 无中断； 1: 有中断。																													
[17]	RW	tx_crf_intrpt	发包原始中断状态，写 1 清中断。 0: 无中断； 1: 有中断。																													
[16]	RW	cfg_fifo_emp_intrpt	发包队列 FIFO 空原始中断状态，写 1 清中断。 0: 无中断； 1: 有中断。																													
[15:3]	-	forbbiden	禁止配置。																													
[2]	RO	rx_intrpt_sta	收包中断状态。 0: 无中断； 1: 有中断。																													
[1]	RO	tx_intrpt_sta	发包中断状态。 0: 无中断； 1: 有中断。																													



[0]	RO	cfg_fifoemp_intrpt_sta	发包队列 FIFO 空中断状态。 0: 无中断; 1: 有中断。
-----	----	------------------------	--

ERR_GIVEN_PKG_CNT

ERR_GIVEN_PKG_CNT 为因 MAC 标记错误帧尾而被丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0280		ERR_GIVEN_PKG_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	err_given_pkg_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	err_given_pkg_cnt	因 MAC 标记错误帧尾而被丢弃的包数统计。					

ERR_FRM_TYPE_CNT

ERR_FRM_TYPE_CNT 为因帧类型不在过滤列表内而被丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0288		ERR_FRM_TYPE_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	err_frm_type_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	err_frm_type_cnt	因帧类型不在过滤列表内而被丢弃的包数统计。					

ERR_IP_TYPE_CNT

ERR_IP_TYPE_CNT 为因 IP 类型不在过滤列表内而被丢弃的包数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x028C		ERR_IP_TYPE_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	err_ip_type_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	err_ip_type_cnt	因 IP 类型不在过滤列表内而被丢弃的包数统计。					

OVER_FLOW_CNT

OVER_FLOW_CNT 为因接收 FIFO 满而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0294		OVER_FLOW_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	over_flow_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	over_flow_cnt	因接收 FIFO 满而丢弃的包数统计。					

OVER_LENGTH_CNT

OVER_LENGTH_CNT 为因超过最大长度而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x0298		OVER_LENGTH_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	over_length_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	over_length_cnt	因超过最大长度而丢弃的包数统计。					



CRF_CFF_DATA_NUM

CRF_CFF_DATA_NUM 为配置 FIFO 数据个数寄存器。

Offset Address		Register Name		Total Reset Value					
0x02A8		CRF_CFF_DATA_NUM		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden		crf_rx_cfg_num		forbidden		crf_tx_cfg_num		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:26]	-	forbidden	禁止配置。						
[25:16]	RO	crf_rx_cfg_num	接收配置 FIFO 中剩余地址个数。						
[15:10]	-	forbidden	禁止配置。						
[9:0]	RO	crf_tx_cfg_num	发送配置 FIFO 中剩余地址个数。						

FLOW_OUT_IP_CNT

FLOW_OUT_IP_CNT 为 IP 报文限速丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02AC		FLOW_OUT_IP_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	flow_out_ip_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	flow_out_ip_cnt	IP 报文限速丢弃的包数统计。					

FLOW_OUT_CTRL_CNT

FLOW_OUT_CTRL_CNT 为控制报文限速丢弃的包数统计寄存器。



Offset Address		Register Name		Total Reset Value				
0x02B0		FLOW_OUT_CTRL_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	flow_out_ctrl_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	flow_out_ctrl_cnt	控制报文限速丢弃的包数统计。					

IP_CHK_ERR_CNT

IP_CHK_ERR_CNT 为 IP 头校验错而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02B4		IP_CHK_ERR_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	ip_chk_err_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	ip_chk_err_cnt	IP 头校验错而丢弃的包数统计。 若接收包的 IP 头部不全或者只有头部而没有净荷，也认为 IP 头部校验错误。					

UDP_CHK_ERR_CNT

UDP_CHK_ERR_CNT 为 UDP 校验错而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02B8		UDP_CHK_ERR_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	udp_chk_err_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	udp_chk_err_cnt	UDP 校验错而丢弃的包数统计。					



TX_RUNT_ERR_PKG_CNT

TX_RUNT_ERR_PKG_CNT 为上行因 FIFO 满而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02BC		TX_RUNT_ERR_PKG_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_runt_err_pkg_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_runt_err_pkg_cnt	上行因 FIFO 满而丢弃的包数统计。					

RX_ERR_TOTAL_CNT

RX_ERR_TOTAL_CNT 为下行总丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02C0		RX_ERR_TOTAL_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_err_total_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_err_total_cnt	下行总丢弃的包数统计。					

RX_TRANS_PKG_CNT

RX_TRANS_PKG_CNT 为下行搬运成功的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02C8		RX_TRANS_PKG_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_trans_pkg_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_trans_pkg_cnt	下行搬运成功的包数统计。					



TX_TRANS_PKG_CNT

TX_TRANS_PKG_CNT 为上行搬运成功的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02CC		TX_TRANS_PKG_CNT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_trans_pkg_cnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_trans_pkg_cnt	上行搬运成功的包数统计。					

RX_BM_OVERFLOW

RX_BM_OVERFLOW 为广播或多播报文因限速而丢弃的包数统计寄存器。

Offset Address		Register Name		Total Reset Value				
0x02D0		RX_BM_OVERFLOW		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_bm_overflow							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_bm_overflow	广播或多播报文因限速而丢弃的包数统计。					

TX_ADDR_OVERFLOW

TX_ADDR_OVERFLOW 为 CPU 写上行配置 FIFO 溢出寄存器。



Offset Address		Register Name		Total Reset Value				
0x02D4		TX_ADDR_OVERFLOW		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tx_addr_overflow							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	tx_addr_overflow	CPU 写上行配置 FIFO 溢出计数器，当 CPU 写上行配置 FIFO 时，由于 FIFO 满而未成功写入的地址个数。					

RX_ADDR_OVERFLOW

RX_ADDR_OVERFLOW 为 CPU 写下行配置 FIFO 溢出寄存器。

Offset Address		Register Name		Total Reset Value				
0x02D8		RX_ADDR_OVERFLOW		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rx_addr_overflow							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RC	rx_addr_overflow	CPU 写下行配置 FIFO 的溢出计数器，当 CPU 写下行配置 FIFO 时，由于 FIFO 满而未成功写入的地址个数。					

CRF_TX_PAUSE

CRF_TX_PAUSE 为发送流控控制寄存器。



Offset Address		Register Name		Total Reset Value																												
0x0340		CRF_TX_PAUSE		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																											crf_tx_pause_auto				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																												
[31:1]	-	forbidden		禁止配置。																												
[0]	RW	crf_tx_pause_auto		发送流控控制。 0: MAC 根据实际流量决定是否发送流控帧，当 PMU 的接收 FIFO 满或将满时，MAC 会自动发送流控帧； 1: 强制 MAC 发送流控帧。																												

CRF_RX_ADDR_NUM

CRF_RX_ADDR_NUM 为限速下行地址个数寄存器。

Offset Address		Register Name		Total Reset Value																												
0x0344		CRF_RX_ADDR_NUM		0x0000_000A																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																							crf_rx_addr_num								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
Bits	Access	Name		Description																												
[31:8]	-	forbidden		禁止配置。																												
[7:0]	RW	crf_rx_addr_num		当下行配置 FIFO 中剩余地址个数小于配置值时，将只通过控制报文，数据报文被全部丢弃。 注意：推荐配置为 0，配置后不要再更改。																												

CRF_CTRL_0_TYPE

CRF_CTRL_0_TYPE 为控制报文类型 0 寄存器。



Offset Address		Register Name		Total Reset Value					
0x0348		CRF_CTRL_0_TYPE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_ctrl_0_type				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	crf_ctrl_0_type	配置的控制报文类型 0，当接收报文的帧类型与配置的类型相同时，认为该报文是控制报文。						

CRF_CTRL_1_TYPE

CRF_CTRL_1_TYPE 为控制报文类型 1 寄存器。

Offset Address		Register Name		Total Reset Value					
0x034C		CRF_CTRL_1_TYPE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_ctrl_1_type				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	crf_ctrl_1_type	配置的控制报文类型 1，当接收报文的帧类型与配置的类型相同时，认为该报文是控制报文。						

CRF_CTRL_2_TYPE

CRF_CTRL_2_TYPE 为控制报文类型 2 寄存器。



Offset Address		Register Name		Total Reset Value					
0x0350		CRF_CTRL_2_TYPE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_ctrl_2_type				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	crf_ctrl_2_type	配置的控制报文类型 2，当接收报文的帧类型与配置的类型相同时，认为该报文是控制报文。						

CRF_CTRL_3_TYPE

CRF_CTRL_3_TYPE 为控制报文类型 3 寄存器。

Offset Address		Register Name		Total Reset Value					
0x0354		CRF_CTRL_3_TYPE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_ctrl_3_type				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	crf_ctrl_3_type	配置的控制报文类型 3，当接收报文的帧类型与配置的类型相同时，认为该报文是控制报文。						

CRF_BM_PKT_THRSLD

CRF_BM_PKT_THRSLD 为广播和多播报文的限速处理包个数寄存器。



Offset Address		Register Name		Total Reset Value					
0x0358		CRF_BM_PKT_THRSLD		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_bm_pkt_thrslid				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:16]	-	forbidden	禁止配置。						
[15:0]	RW	crf_bm_pkt_thrslid	广播和多播报文的包数上限，当在限速单位时间内接收的广播或多播报文超过配置值，则进行限速，否则不限速。						

CRF_BM_TIME_THRSLD

CRF_BM_TIME_THRSLD 为广播和多播报文的限速时间计数寄存器。

Offset Address		Register Name		Total Reset Value					
0x035C		CRF_BM_TIME_THRSLD		0x0000_2710					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	forbidden				crf_bm_time_thrslid				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 1	0 0 0 1	0 0 0 0	
Bits	Access	Name	Description						
[31:20]	-	forbidden	禁止配置。						
[19:0]	RW	crf_bm_time_thrslid	广播和多播报文的限速时间，以 1 μ s 为单位进行计数。						

MDIO_SINGLE_CMD

MDIO_SINGLE_CMD 为 MDIO 单次操作寄存器。



	Offset Address 0x03C0				Register Name MDIO_SINGLE_CMD				Total Reset Value 0x0001_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden				mdio_cmd	forbidden		op_code	forbidden				phy_addr	forbidden		reg_addr																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:21]	-		forbidden		禁止配置。																											
[20]	RW		mdio_cmd		MDIO 操作控制。 读时： 0：MDIO 操作完成； 1：MDIO 操作未完成。 写时： 0：无影响； 1：启动 MDIO 操作。																											
[19:18]	-		forbidden		禁止配置。																											
[17:16]	RW		op_code		操作控制。 00：保留； 01：写操作； 10：读操作； 11：保留。																											
[15:10]	-		forbidden		禁止配置。																											
[9:8]	RW		phy_addr		外部 PHY 地址的低 2bit，其高 3bit 默认为 0。																											
[7:5]	-		forbidden		禁止配置。																											
[4:0]	RW		reg_addr		PHY 器件内部的寄存器地址。																											

MDIO_SINGLE_DATA

MDIO_SINGLE_DATA 为 MDIO 读写数据寄存器。



	Offset Address				Register Name				Total Reset Value																											
	0x03C4				MDIO_SINGLE_DATA				0x0000_0000																											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	mdio_rd_data												mdio_wr_data																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																															
[31:16]	RO		mdio_rd_data		MDIO 从外部 PHY 器件回读的数据。																															
[15:0]	RW		mdio_wr_data		MDIO 写数据。																															

MDIO_CTL

MDIO_CTL 为 MDIO 控制寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x03CC				MDIO_CTL				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden												mdio_in_wor	mdio_in_wor	forbidden	mdc_speed																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:4]	-		forbidden		禁止配置。																											
[3]	RO		mdio_in_work		MDIO 单次操作是否执行。 0: 否; 1: 是。																											
[2]	RW		mdio_in_work_en		MDIO 单次操作使能。 0: 禁止; 1: 使能。																											
[1]	-		forbidden		禁止配置。																											
[0]	RW		mdc_speed		MDIO 接口时钟频率。 0: 2.5MHz; 1: 18MHz。																											



MDIO_RDATA_STATUS

MDIO_RDATA_STATUS 为 MDIO 读数据状态寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x03D0				MDIO_RDATA_STATUS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	forbidden																												mdio_rdata_status			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:1]	-	forbidden		禁止配置。																											
	[0]	RC	mdio_rdata_status		MDIO 读数据是否有效，读清。 0: 有效; 1: 无效。																											

TX_CFF_LEN

TX_CFF_LEN 为发包队列包长配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0400+m×0x4				TX_CFF_LEN				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	tx_cff_len																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RW	tx_cff_len		发包队列包长。																											

TX_CFF_ADDR

TX_CFF_ADDR 为发包队列地址配置寄存器。



	Offset Address				Register Name				Total Reset Value																							
	0x0580+m×0x4				TX_CFF_ADDR				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	tx_cff_addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RW	tx_cff_addr		发包队列地址。																											

RX_CFF_ADDR

RX_CFF_ADDR 为收包队列地址配置寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0480+n×0x4				RX_CFF_ADDR				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rx_cff_addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RW	rx_cff_addr		收包队列地址。																											



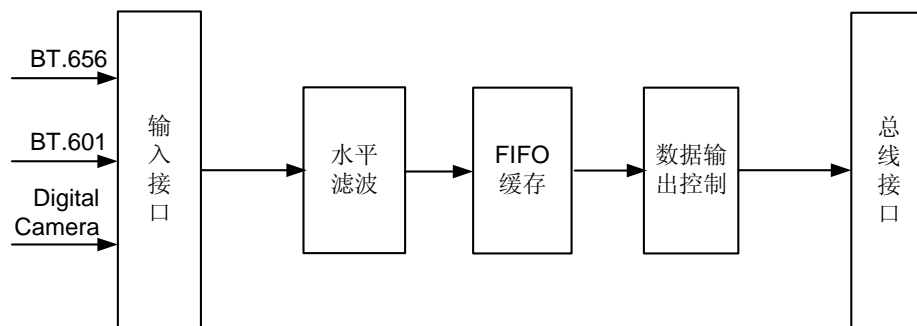
6 视频接口

6.1 视频输入单元

6.1.1 概述

视频输入单元 VIU (Video Input Unit) 可以通过 BT.656/601 接口和 DC (Digital Camera) 接口接收视频数据, 存入指定的内存区域。在此过程中, VIU 可以对视频图像数据进行水平 1/2 的缩小 (Down Scaling) 和色度重采样 (Chrominance Resampling)。VIU 的功能框图如图 6-1 所示。

图6-1 VIU 功能框图



6.1.2 特点

VIU 有以下特点:

- 外部支持 4 个视频端口。
- 内部支持 16 路通道视频处理。
- 每个端口都支持标准 BT.656 视频输入。
- 支持 2 路和 4 路时分复用视频输入。
- 每个端口都支持隔行和逐行输入模式。
- 端口 0 和端口 2 支持 BT.601 视频输入。
- 端口 0 和端口 2 支持直接 Digital Camera 连接 (最大支持 300 万像素)。



- 端口 0 和 1、端口 2 和 3 支持 BT 1120（高清）接口时序。
- 端口 0 和 1、端口 2 和 3 支持 16-bit 同步接口高清模式。
- 支持场接收模式和帧接收模式。
- 输入数据支持 CbYCrY、CrYCbY、YCbYCr 和 YCrYCb 4 种顺序格式。
- 支持 1/2 水平 Down Scaling。
- 支持 Co-Sited 格式到 Interspersed 格式的水平色度重新采样转换。
- 各通道支持 2 级总线优先级可配。
- 支持在一个指定窗口内获取数据。
- 支持图像块遮挡。
- 输出格式支持存储模式：
 - package YCbCr 4:2:2 模式。
 - SPYCbCr 4:2:0 和 SPYCbCr 4:2:2 模式。
- 输出数据格式支持 raw data 存储模式，该模式只用在 Y/C 分开输入的高清模式。
- 支持图象亮度统计及亮度拉伸功能。
- 提供 AXI Master 接口，用于直接把数据写入 DDR 中。
- 提供 AHB Slave 接口，用于软件对 VIU 的寄存器配置和状态信息读取。

6.1.3 信号描述

视频输入接口的外部输入输出管脚信号描述如表 6-1 所示。

表6-1 视频输入接口信号

信号名	方向	描述	对应管脚
VI_DATAIN_P0[7:0]	I	视频端口 0 输入数据。	VI0DAT7~ VI0DAT0
VI_DATAIN_P1[7:0]	I	视频端口 1 输入数据。	VI1DAT7~ VI1DAT0
VI_DATAIN_P2[7:0]	I	视频端口 2 输入数据。	VI2DAT7~ VI2DAT0
VI_DATAIN_P3[7:0]	I	视频端口 3 输入数据。	VI3DAT7~ VI3DAT0
CLK_VI_P0	I	端口 0 视频输入时钟。	VI0CK
CLK_VI_P1	I	端口 1 视频输入时钟。	VI1CK
CLK_VI_P2	I	端口 2 视频输入时钟。	VI2CK
CLK_VI_P3	I	端口 3 视频输入时钟。	VI3CK



信号名	方向	描述	对应管脚
VI_P0_HSYNC_VD	I	<p>端口 0 水平同步脉冲或数据有效信号。寄存器 VIn_PORT_CFG[port_hsync]控制该信号是同步脉冲或数据有效电平。</p> <p>port_hsync=0: 数据有效电平, 高电平或者低电平有效可通过 VIn_PORT_CFG[port_hsync_neg]配置。</p> <p>port_hsync=1: 同步脉冲, 正脉冲或者负脉冲可通过 VIn_PORT_CFG[port_hsync_neg]配置。</p>	VI0HS
VI_P2_HSYNC_VD	I	<p>端口 2 水平同步脉冲或数据有效信号。寄存器 VIn_PORT_CFG[port_hsync]控制该信号是同步脉冲或数据有效电平。</p> <p>port_hsync=0: 数据有效电平, 高电平或者低电平有效可通过 VIn_PORT_CFG[port_hsync_neg]配置。</p> <p>port_hsync=1: 同步脉冲, 正脉冲或者负脉冲可通过 VIn_PORT_CFG[port_hsync_neg]配置。</p>	VI2HS
VI_P0_VSYNC_FIELD	I	<p>端口 0 垂直同步脉冲或场指示信号。寄存器 VIn_PORT_CFG[port_vsync]控制该信号是同步脉冲或场指示信号。</p> <p>port_vsync=0: 场指示信号, 高电平或者低电平配置奇偶场有效可通过 VIn_PORT_CFG[port_vsync_neg]配置。</p> <p>port_vsync=1: 同步脉冲, 正脉冲或者负脉冲可通过 VIn_PORT_CFG[port_vsync_neg]配置。</p>	VI0VS



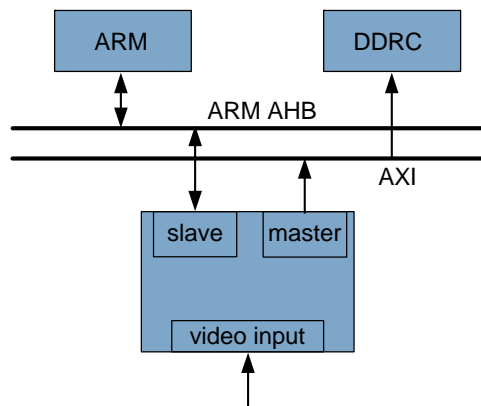
信号名	方向	描述	对应管脚
VI_P2_VSYNC_FIELD	I	<p>端口 2 垂直同步脉冲或场指示信号。寄存器 <code>VIn_PORT_CFG[port_vsync]</code> 控制该信号是同步脉冲或场指示信号。</p> <p><code>port_vsync=0</code>: 场指示信号, 高电平或者低电平配置奇偶场有效可通过 <code>VIn_PORT_CFG[port_vsync_neg]</code> 配置。</p> <p><code>port_vsync=1</code>: 同步脉冲, 正脉冲或者负脉冲可通过 <code>VIn_PORT_CFG[port_vsync_neg]</code> 配置。</p>	VI2VS

6.1.4 功能描述

6.1.4.1 典型应用

VIU 典型应用如图 6-2 所示。

图6-2 VIU 典型应用图



VIU 是一个支持多种时序输入的视频输入采集单元，将采集到的视频数据存储到 DDR 中，系统可以配置不同的功能模式，使之可以灵活的适应不同的外部输入视频接口，支持多种外部输入设备。

6.1.4.2 功能原理

ITU-R BT. 656 YCbCr4:2:2

1. 水平时序



在 ITU-R BT.656 建议中，同步信号集成在数据流中，在数据流中的特殊字节 SAV 和 EAV 分别表示有效行数据的开始和结束。在视频数据流中，由 FF 00 00（FF、00 为图像编码数据的保留值，为非图像数据）构成的定时基准码字的码头来标志紧接着的一个字节为 SAV 或者 EAV，ITU-R BT. 656 的行数据流格式如表 6-2 所示。

表6-2 ITU-R BT.656 YCbCr 4:2:2 行数据格式

定时基准码				行消隐区					定时基准码				720 有效像素 YCbCr 4:2:2							
FF	00	00	EAV	80	10	...	80	10	FF	00	00	SAV	Cb0	Y0	Cr0	Y1	...	Cr718	Y719	

SAV 和 EAV 的相互区别由 SAV/EAV 的特殊比特位“H”区分，SAV/EAV 还包含了垂直消隐位“V”和场指示位“F”。SAV/EAV 的具体描述见表 6-3 所示。

表6-3 SAV/EAV 格式

Bit7	Bit6(F)	Bit5(V)	Bit4(H)	Bit[3:0](P3~P0)
固定值 1	场指示位 1st field: F=0 2nd field: F=1	垂直消隐位 VBI: V=1 Active video: V=0	SAV: H=0 EAV: H=1	校验位。

ITU-R BT.656 建议采用了 8 个有效保留位用来定义有效的 SAV 和 EAV，4 个校验位可纠正一比特出错，检测 2 比特出错。有效的 SAV/EAV 值如表 6-4 所示。

表6-4 有效 SAV/EAV 值

编码	二进制值	场号	垂直消隐期
SAV	10000000	1	-
EAV	10011101	1	-
SAV	10101011	1	是
EAV	10110110	1	是
SAV	11000111	2	-
EAV	11011010	2	-
SAV	11101100	2	是
EAV	11110001	2	是

4 个有效保留位还起到纠错的作用，P0、P1、P2、P3 由 F、V 和 H 位来确定的，如表 6-5 所示。



表6-5 ITU-R BT.656 纠错码表

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

其中：

$$P0=F^{\wedge}V^{\wedge}H$$

$$P1=F^{\wedge}V$$

$$P2=F^{\wedge}H$$

$$P3=V^{\wedge}H$$

2. 垂直时序

垂直时序也是通过定时基准码 SAV/EAV 的“F”和“V”来实现，典型的 525 行和 625 行视频系统垂直时序如图 6-3 和图 6-4 所示。



图6-3 525 行 60 场/秒视频系统垂直时序

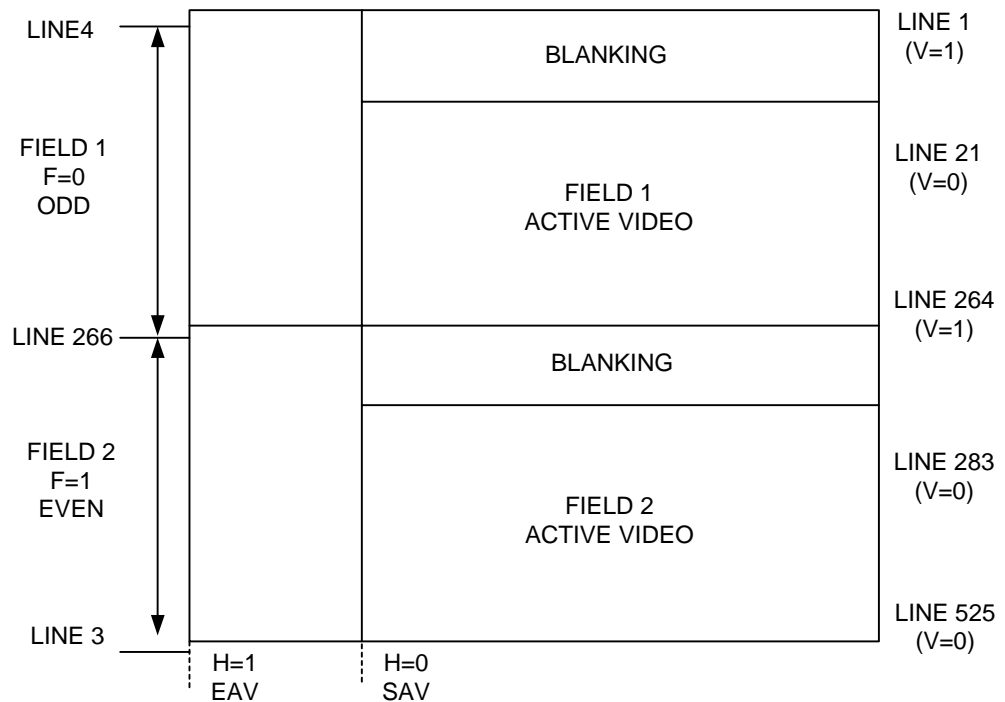
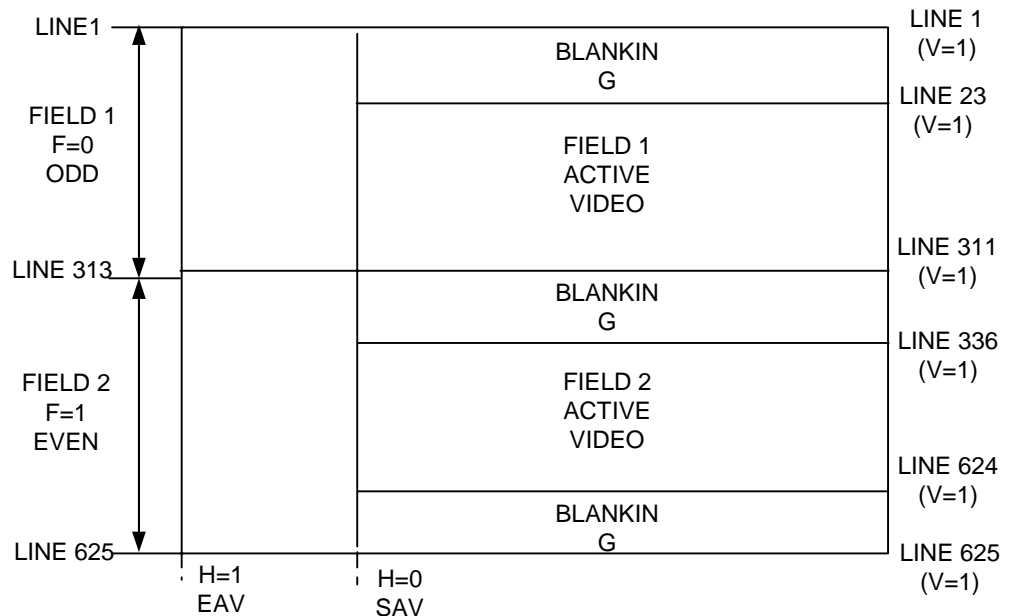


图6-4 625 行 50 场/秒视频系统垂直时序



VIU 内部只根据 SAV/EAV 来识别垂直时序，不会限制具体在哪一行。

BT.656 多路时分复用时序

VIU 除了支持单路标准 BT.656 时序外，还支持 2 路或 4 路 BT.656 通过时分复用，使用一个端口进行传输的时序，具体如图 6-5 和图 6-6 所示。每一路数据都是独立的，各路数据之间任何关系，只是时分复用在一起，通过一个端口输入到 VIU 内部。

图6-5 2 路 BT.656 时分复用时序

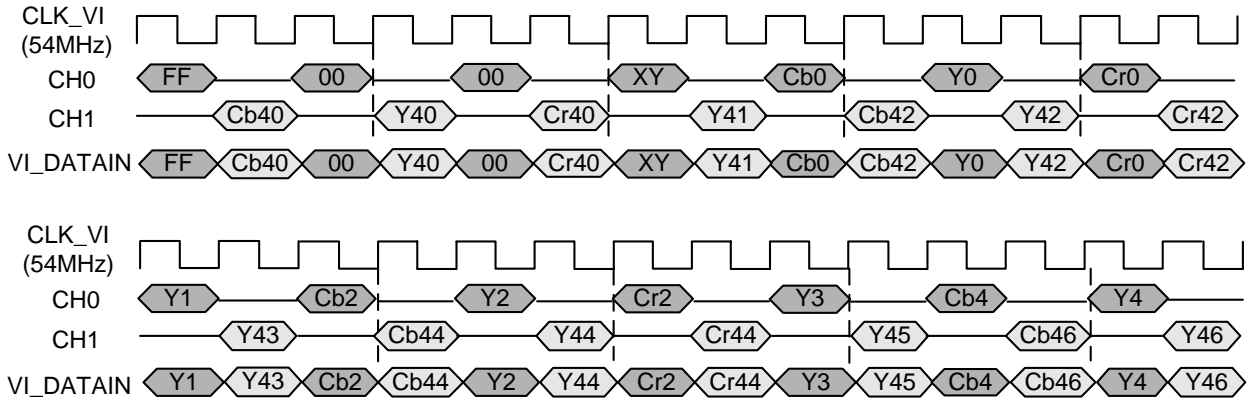
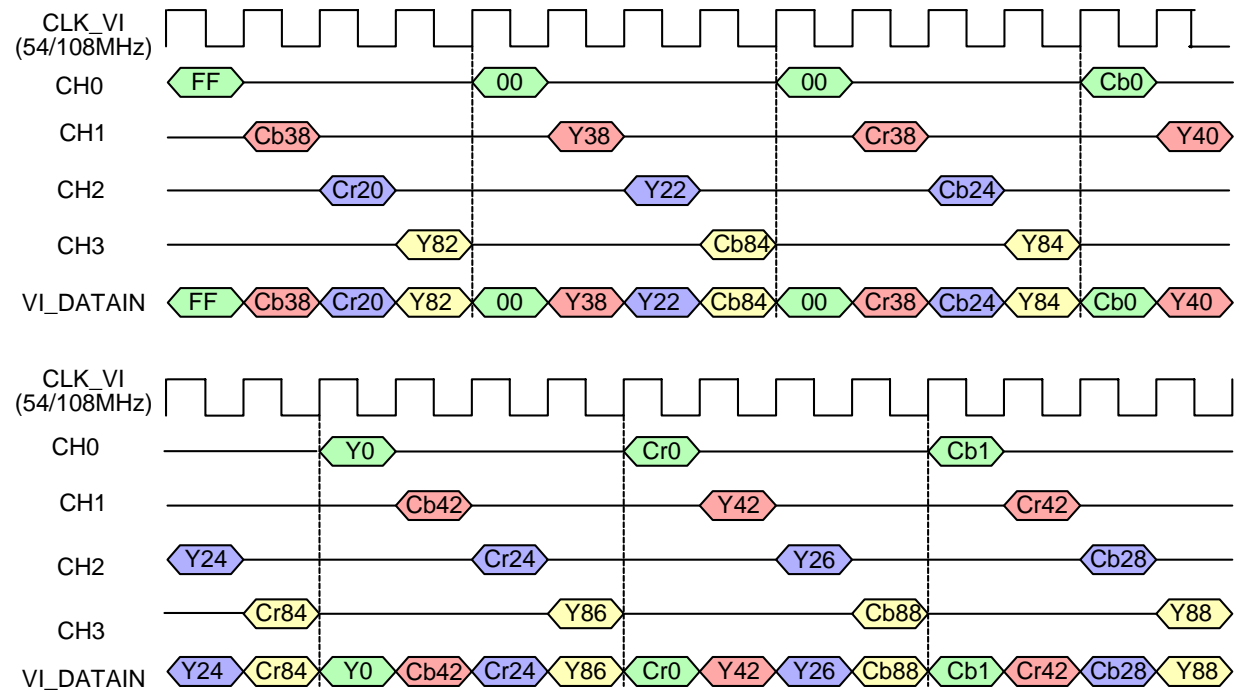


图6-6 4 路 BT.656 时分复用时序



当输入数据为时分复用的格式时，SAV/EAV 中纠错位的位置的低 2 比特用来表示输入数据通道号信息此时需要将 `VIn_CH_CFG[correct_en]` 比特设置为 0，带通道号的 SAV/EAV 格式如表 6-6 所示。



表6-6 带通道号的 SAV/EAV 格式

Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1:0]
1	F	V	H	x	x	通道号

BT 1120（高清）接口时序

VIU 支持 Y/C 分开输入的高清接口时序，此时需要用到 2 个端口，一个端口用来传输亮度，另一个端口用来传输色度，如图 6-7 和图 6-8 所示。

图6-7 高清接口输入时序水平时序

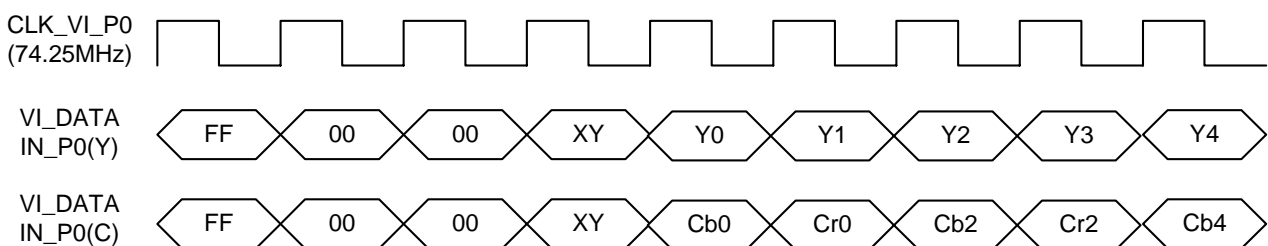
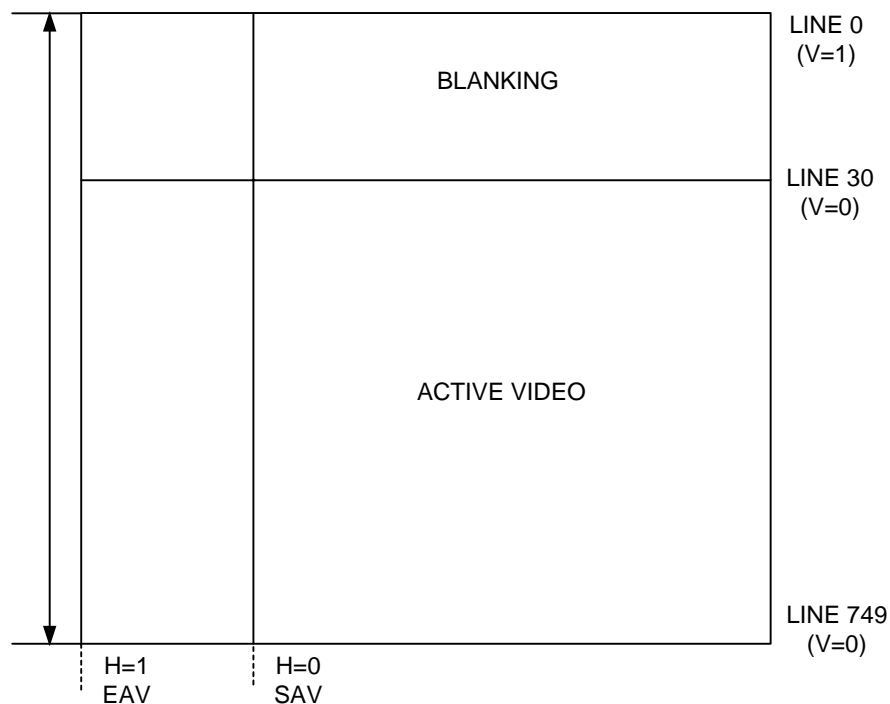


图6-8 高清接口输入时序垂直时序

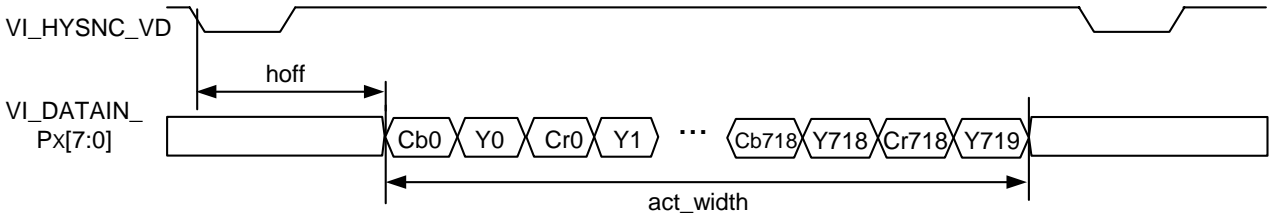


ITU-R BT.601 YCbCr4:2:2

1. 水平时序

水平脉冲表示新的的一行开始如图 6-9 所示。经过 h_{off}（可配，PAL525 行系统该值为 122×2，NTSC625 行系统该值为 132×2）个时钟后，行前消隐区结束，进入行有效数据区，经过 act_{width}（可配，典型值为 720 或 704）个时钟后行有效区域结束，进入行后消隐区。水平同步极性可配。

图6-9 ITU-R BT.601 水平时序图

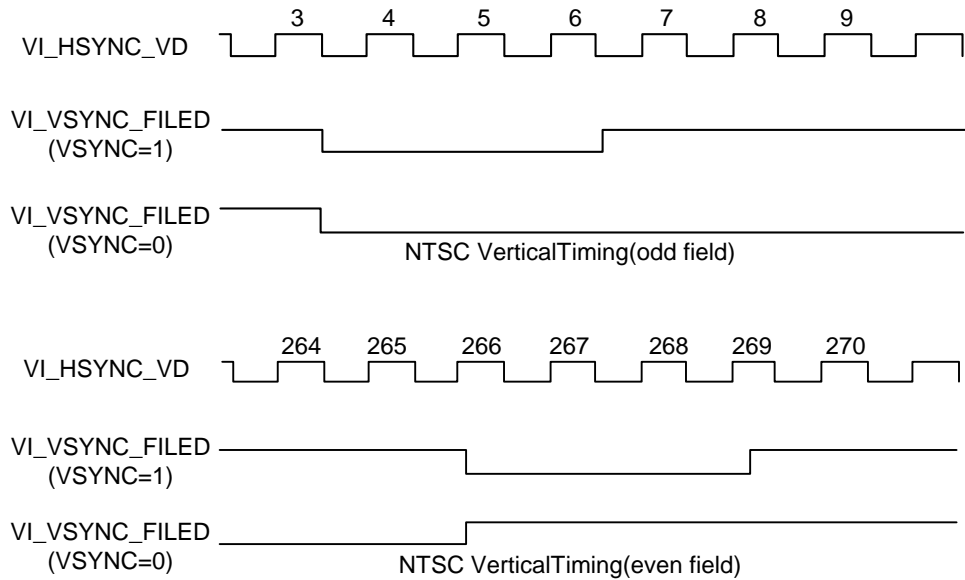


2. 垂直时序

ITU-R BT.601 建议：信号 VSYNC/FIELD 作为垂直同步信号。VSYNC 的脉冲或者 FIELD 的跳变标志奇偶场的开始，VIU 支持以下 2 种垂直同步方法。

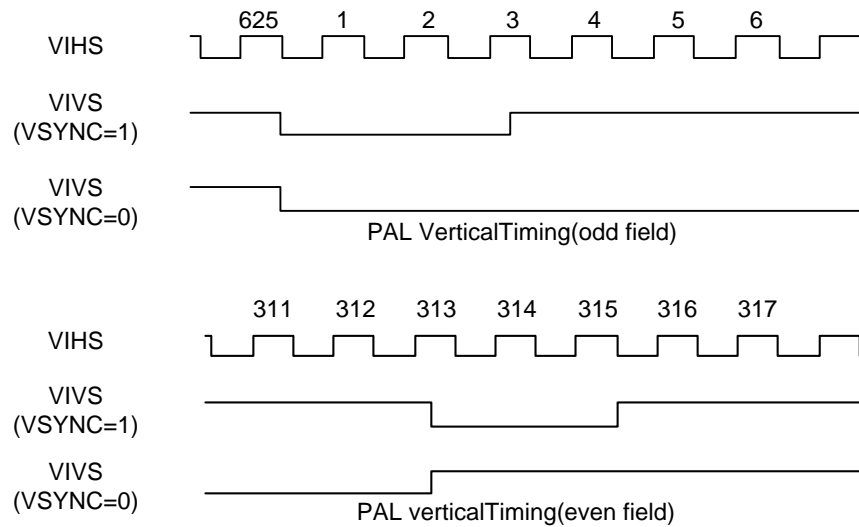
VI 在 NTSC 制式（625 行）和 PAL 制式（525 行）的垂直时序关系图如图 6-10 和图 6-11 所示，其中 VI_HSYNC_VD 为水平同步脉冲，VI_VSYNC_FIELD 在 VSYNC=1 时为垂直同步脉冲；在 VSYNC=0 时为场同步信号。

图6-10 NTSC 制式垂直同步时序图



在 NTSC 隔行扫描制式下，第 1 场的垂直同步信号在第 4 行的起始位置变为低电平，持续 3 行低电平后，在第 7 行的起始位置变为高电平。VIU 接收从第 22 行开始到第 261 行结束的 240 行数据。第 2 场的垂直同步信号在第 266 行的中间位置变为低电平，持续 3 行低电平后，在第 269 行中间位置变为高电平。VIU 接收从第 285 行开始第 524 行结束的 240 行数据。

图6-11 PAL 制式垂直同步时序图



在 PAL 隔行扫描制式下，第 1 场的垂直同步信号在第 1 行起始位置变为低电平，持续 2.5 行低电平后，在第 3 行的中间位置变为高电平。VIU 接收从第 24 行开始到第 310 行结束的 288 行数据。第 2 场的垂直同步信号在第 313 行的中间位置变为低电平，持续 2.5 行低电平后，在第 316 行的起始位置变为高电平。VIU 接收从第 336 行开始到第 623 行结束的 288 行数据。

上面两种为典型的 BT.601 垂直时序，VIU 内部支持场开始到场有效行开始之间的行数可配，场有效行的行数可配，垂直同步的极性可配。

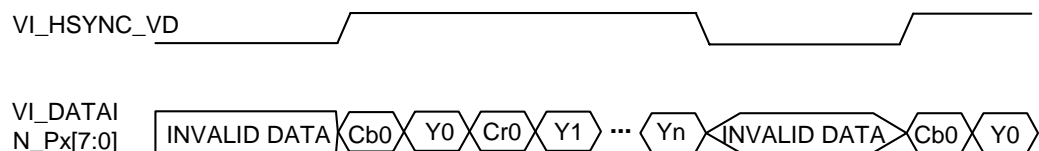
数字摄像头接口时序

VIU 支持分辨率最大为 QXGA (2048x1536) 的摄像头数据传输。

1. 水平时序

VIU 接数字摄像头时，VI_HSYNC_VD 表示数据有效信号，数据有效信号极性可配，水平时序如图 6-12 所示。

图6-12 数字摄像头水平时序



2. 垂直时序

VIU 支持两种垂直时序脉冲方式和行有效方式，如图 6-13 和图 6-14 所示。垂直同步极性可配。



图6-13 数字摄像头垂直时序脉冲方式

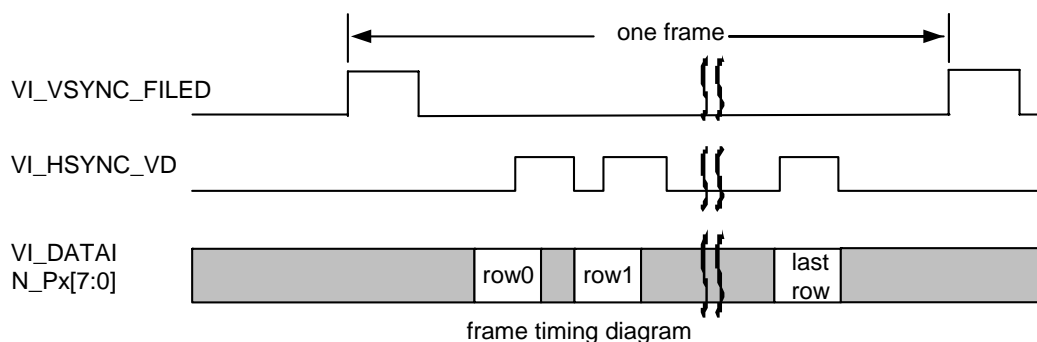
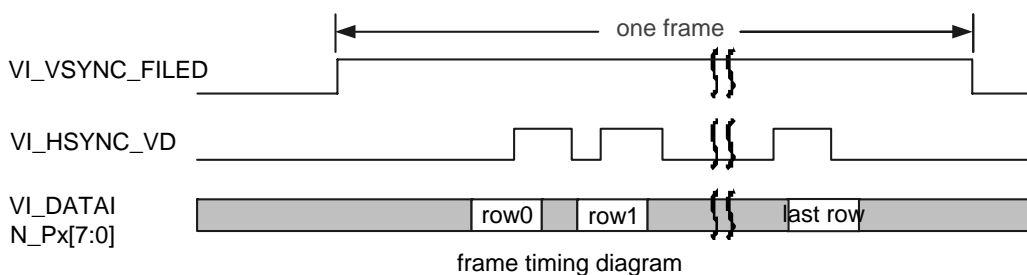


图6-14 数字摄像头垂直时序行有效方式



对于 VIU 内部处理来说，这两种时序相同，VIU 内部检测到一个上升沿或下降沿，这认为是一帧的开始，然后检测数据有效信号，来判断当前数据是否有效。

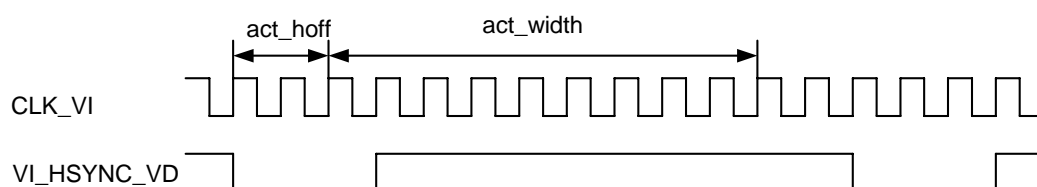
16-bit 同步接口

VIU 支持 Y、C 分开输入且同步信号与数据分离的 16-bit 同步接口。端口 0 和 1 为 1 组，同步信号均来自于端口 0 的同步信号；端口 2 和 3 为 1 组，同步信号均来自于端口 2 的同步信号。

1. 水平时序

16bit 同步并行接口水平同步时序如图 6-15 所示。

图6-15 16bit 同步并行接口水平同步时序图

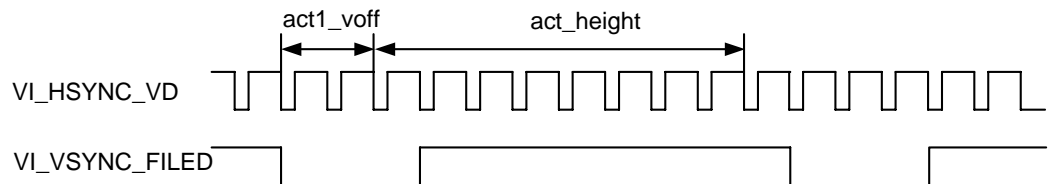


act_hoff 和 act_width 可配，请参见寄存器 VI_Px_HSYNC；vi_px_hsync_vd 极性可配，请参见 [VIn_PORT_CFG](#)。

2. 垂直时序

16bit 同步并行接口垂直同步时序如图 6-16 所示。

图6-16 16-bit 同步并行接口垂直时序图

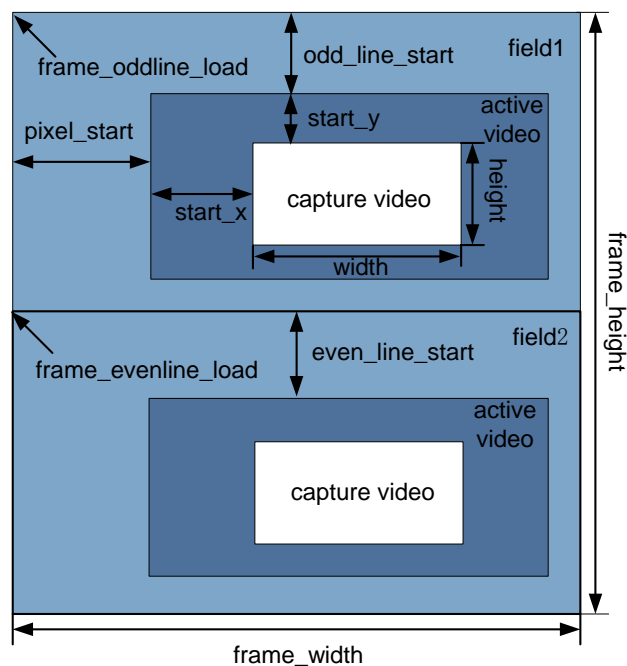


act1_voff 和 act1_height 可配，请参见寄存器 VI_Px_VSYNC1；vi_px_vsync_field 极性可配，请参见 VIn_PORT_CFG。

视频有效范围

有效视频范围如图 6-17 所示。开始于水平消隐和垂直消隐之后。而实际显示的视图区域常常包含在有效视频范围之内，相对有效视频的边界有一点点缩小，其目的是避免边缘效应。

图6-17 有效图像区域与水平垂直消隐关系图



注：相关参数详细定义请参见寄存器“6.1.7 VIn_CAP_START”的相关描述内容。



图像数据采集

VIU 模块的主要任务就是把视频数据流采集并存储到 DDR 中。VIU 内部，在水平方向上支持色度空间的 co-sited 格式到 interspersed 格式的转换，同时也支持对亮度以及色度的 1/2 缩小。

1. 两种 YCbCr 4:2:2 的色度采样方式

在水平方向上 co-sited 和 interspersed 格式色度关系如图 6-18 和图 6-19 所示。

图6-18 YCbCr 4:2:2 co-sited 采样格式

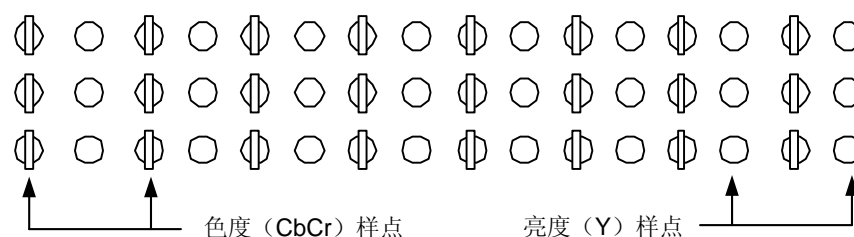


图6-19 YCbCr 4:2:2 interspersed 采样格式



2. co-sited 格式到 interspersed 格式的转换

BT.656/601、YcbCr 4:2:2 标准接口输入的是 co-sited 格式数据。VIU 支持 co-sited 方式直接采集存储和转换成 interspersed 方式存储。

- co-sited 采样

直接将输入图像的亮度和色度分量写入内存，对亮度和色度分量的采样位置不作任何修改，因此，缓存中的图像采样的亮度和色度分量和图 6-18 相同，仍然是 YCbCr 4:2:2 co-sited 格式。

- interspersed 采样

色度信号被重新采样，使得色度分量采样在空间上位于亮度分量的中间，如图 6-19 所示 interspersed 格式。

3. 水平方向 1/2 缩小

水平方向 1/2 缩小时，VIU 将输入图像的水平分辨率（Y、Cb、Cr）降低一半：

- 对于亮度（Y）分量，用 8 阶 FIR 滤波器进行插值滤波。
- 对于色度分量（Cb，Cr），用 4 阶 FIR 滤波器进行插值滤波。

水平方向 1/2 缩小时，可同时进行 co-sited 格式到 interspersed 格式的转换。

图像存储模式

图像存储模式包括：

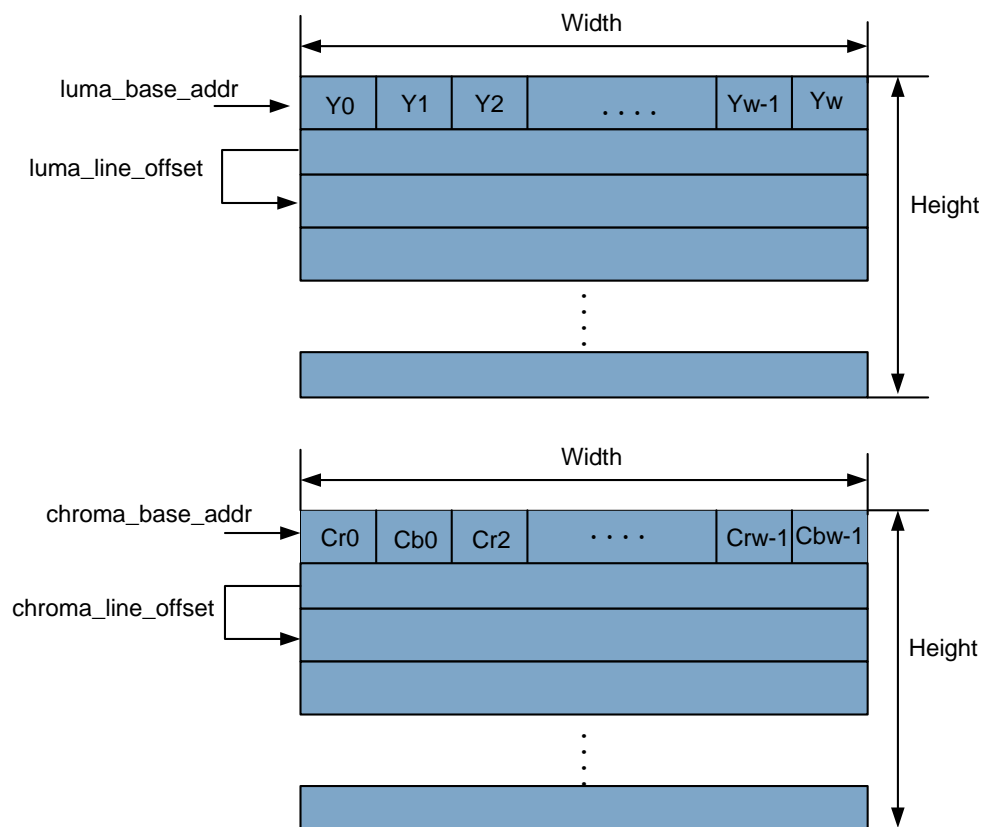
- semi-planar YcbCr 存储
- package 存储
- raw data 存储

1. semi-planar YcbCr 存储

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

- 在 1 行内，亮度、色度分量各自连续存储。
- 连续 2 行之间的存储，可以通过系统定义的行首与行首之间的存储间隔参数 offset 定义。亮度和色度分量在 DDR 中的存储位置由起始地址 base_addr 来指示。VI 捕获的如图 6-20 所示。

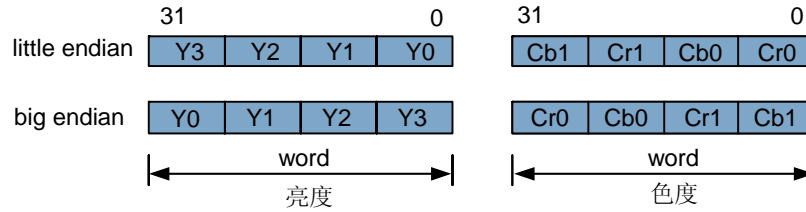
图6-20 YCbCr4:2:2 的存储模式





在 DDR，数据的存储是以 word（32bit）为单位。由 4 个 8bit 像素组成一个 32bit 的 word，在 4 个字节构成一个 word 时有 2 种方式：big endian 和 little endian。图 6-21 是以亮度和色度分量为例来说明 big endian 和 little endian 的存储方式。

图6-21 big endian 和 little endian 图像存储模式

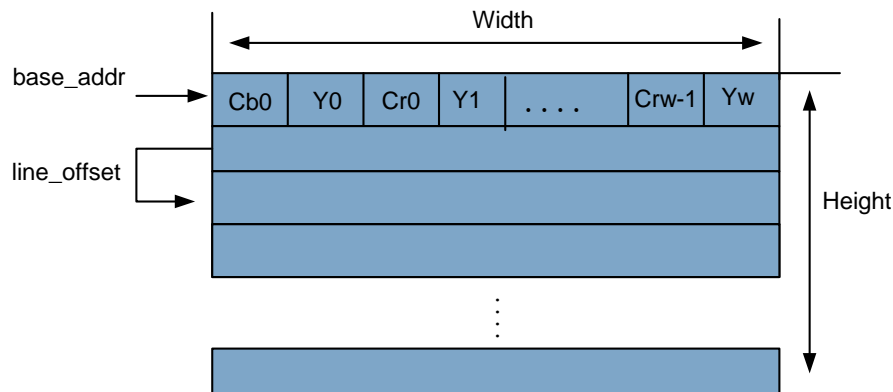


VIU 只支持采用 little endian 方式存储数据的 DDR。

2. package 存储

package 方式存储是将 4:2:2 Y/Cb/Cr 数据按图 6-22 的方式存储。

图6-22 图像存储 package 模式

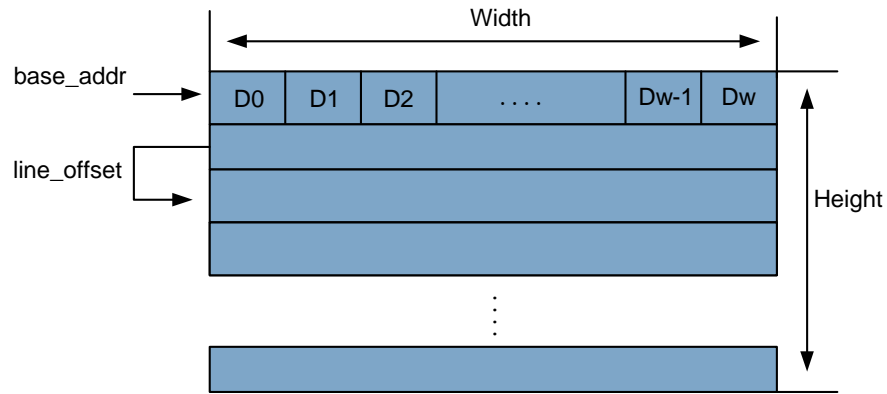


3. raw data 存储

raw data 方式存储是将数据按数据顺序依次存放到一个 word 中。由于在 DDR 中，1 个 word 由 4 个 8bit 组成，当数据为 8bit 时，其存储方式如图 6-23 所示。



图6-23 raw data 8bit 存储模式



亮度统计

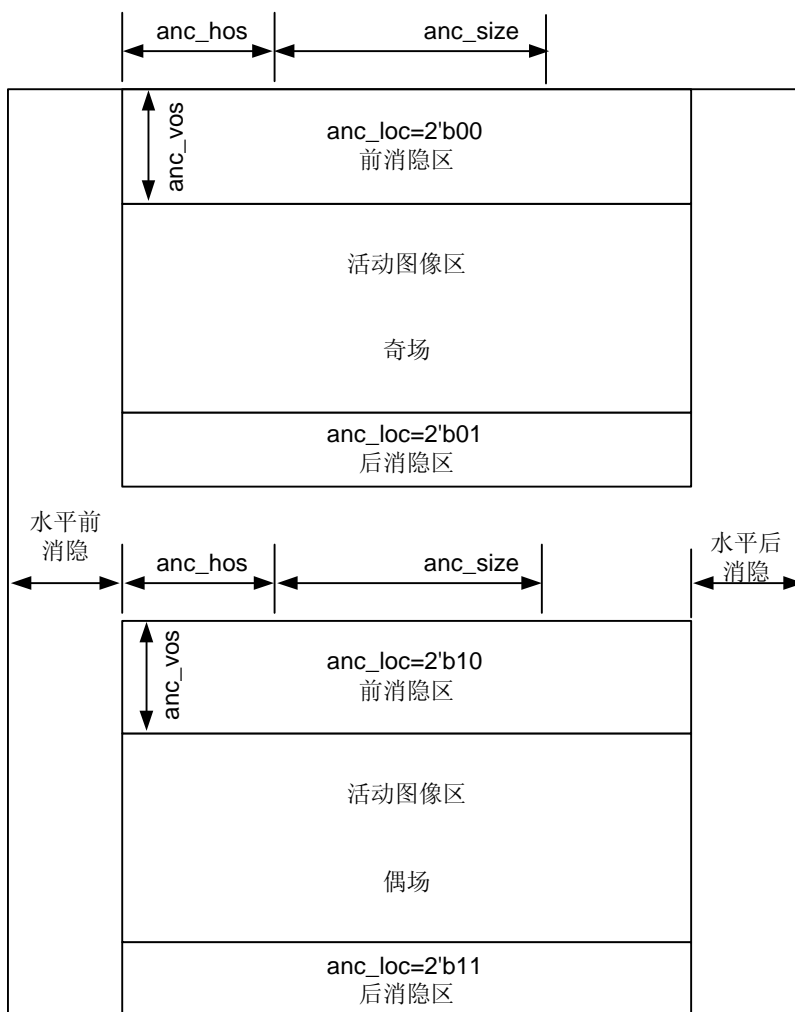
在 VIU 模块中，亮度统计是对当前图象平均亮度的估计，用于判断摄像头是否被遮蔽。其算法结构为：用一个 32bit 的加法器来对输入亮度分量累加，当接收完一幅图象数据后，送出亮度的累加值。

消隐区数据捕获

在许多情况下，通常通过消隐区传输 1 行到 2 行数据，提供其他的数据增值服务，VIU 提供消隐区数据捕获功能，用户需要配置消隐区数据所在位置，如图 6-24 所示。



图6-24 消隐区数据位置示意图

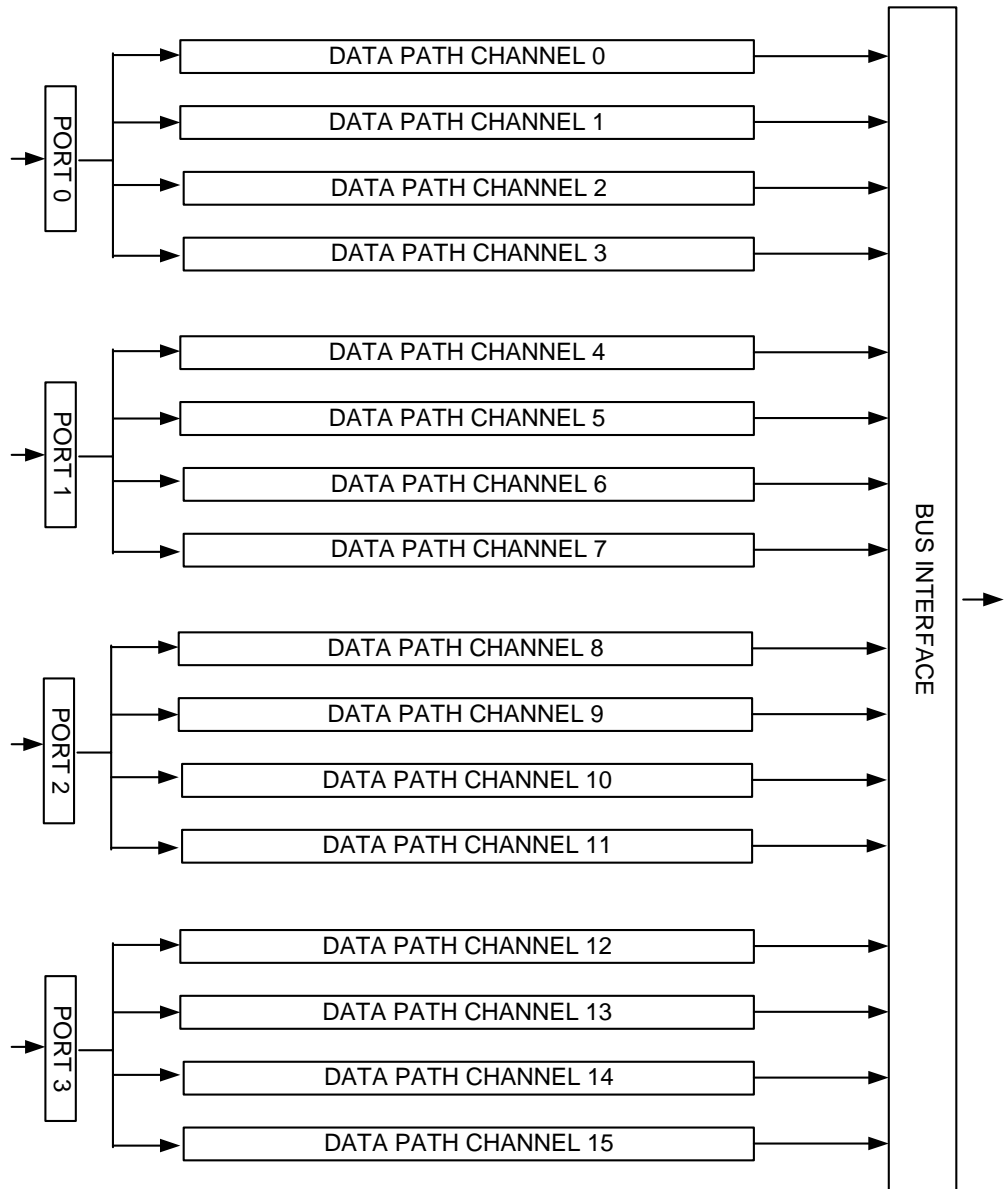


外部端口与内部通道对应关系

VIU 外部有 4 个端口用于接收数据，内部有 16 路数据处理通道，用于对端口输入的数据进行处理，外部端口与内部数据通道对应关系如图 6-25 所示。



图6-25 VIU 端口与通道对应关系示意图



当端口输入的视频数据为非时分复用的情况下，端口输入的数据使用最上面的数据通道进行处理，如端口 0 使用通道 0，端口 1 使用通道 4；当端口输入的视频数据为时分复用的情况下，端口输入的数据使用那个数据通道进行处理，需要根据各个数据通道所配置的通道号来决定。

6.1.5 工作方式

6.1.5.1 管脚复用配置

VI 管脚与 VO、GPIO 管脚复用，可以通过配置 IO Config 配置寄存器 reg0~reg35 选择配置为 VI 功能管脚。



6.1.5.2 软复位

通过 SC_PERCTRL8 bit[25:23]控制 VIU 模块端口 0 到 3 的软复位；

通过 SC_PERCTRL8 bit[21]控制 VIU 模块总线的软复位，具体内容请参见系统控制寄存器 SC_PERCTRL8 相关描述内容。

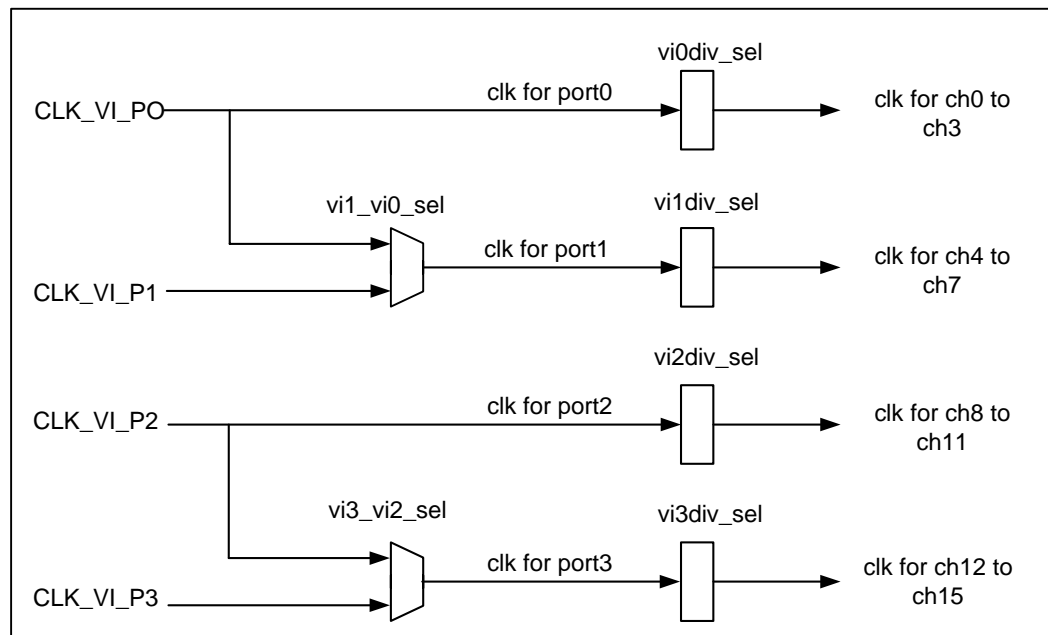
6.1.5.3 时钟配置

VIU 模块的时钟配置主要在系统控制寄存器 SC_PERCTRL9 bit[13:4]中完成，SC_PERCTRL9 的相关比特位的含义及配置方法如下：

- vi1_vi0_sel、vi3_vi2_sel 为端口时钟选择位，用来选择 1 端口和 3 端口的时钟来源。当端口 0 和 1 工作在亮度和色度分离输入的模式下，需要将端口 0 的时钟选到端口 1 上去，即配置 vi1_vi0_sel 为 1；当端口 2 和 3 工作在亮度和色度分离输入的模式下，需要将端口 2 的时钟选到端口 3 上去，即配置 vi3_vi2_sel 为 1；
- vi0div_sel、vi1div_sel、vi2div_sel、vi3div_sel 为分频时钟控制位，当相应端口为时分复用的输入格式时，用来控制分频倍数给内部各通道使用，4 路时分复用设置 4 分频，2 路时分复用设置 2 分频。

VIU 的时钟配置框图如图 6-26 所示。

图6-26 VIU 模块的时钟配置框图



6.1.5.4 中断

VIU 在系统中的中断号为“17”，在 VIU 中，每个内部通道有 8 个中断源，在 VIU 的寄存器中，每个通道有一个表示中断使能的寄存器 `VIn_INT_EN` 和经过使能控制后产生的中断状态寄存器 `VIn_INT_STATUS`。



6.1.5.5 使用指南

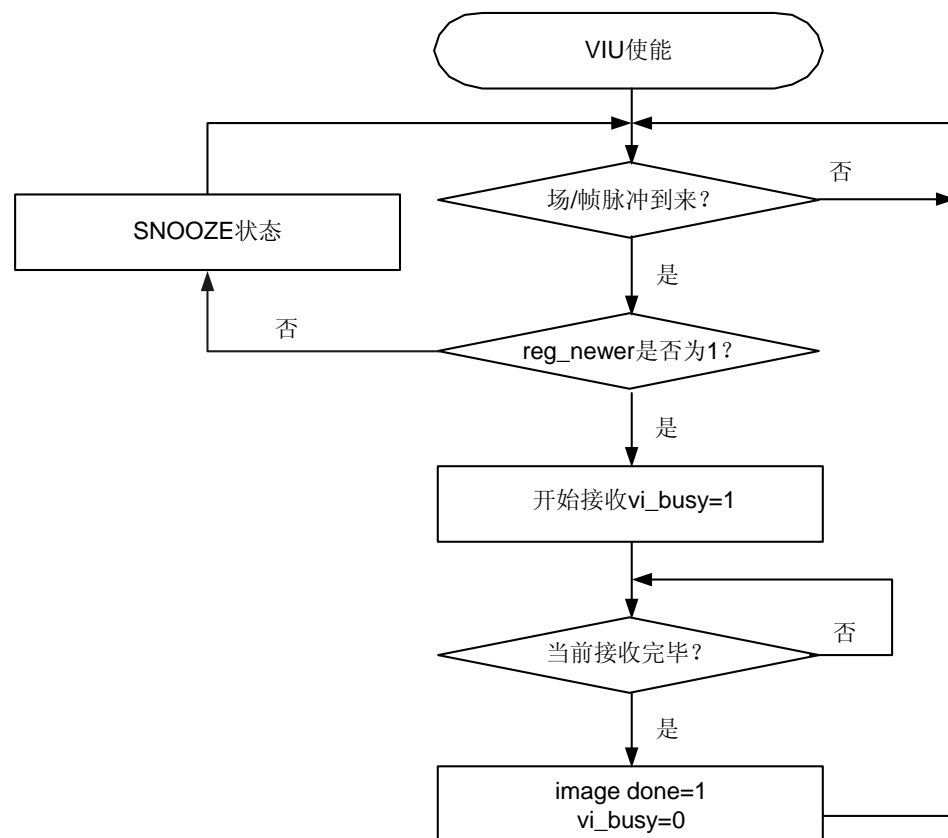
VIU 的 reg_newer 功能

- 在软件使能 VIU 某个通道之前，软件应该完成以下操作：
 - 完成对 VIU 的图像寄存器 `VIn_REG_NEWER` 的写操作。
 - 写 `reg_newer` 位为“1”，通知 VIU 模块当前的寄存器已经准备就绪。
- 使能 VIU 后，VIU 逻辑开始工作，当一场/帧到来的时候，则有：
 - 如果 `reg_newer` 为 0，则 VIU 将不会接收数据，置硬件状态为 SNOOZE，等待下一场/帧的数据的到来。
 - 如果 `reg_newer` 为 1，则开始接收数据，同时给出寄存器更新中断 (`reg_update_int`)，并设置硬件状态为 busy。
- 当接收完毕当前数据后，清除硬件 busy 状态。等到下一场/帧到来的时候，则：
 - 如果 `reg_newer` 为 0，则放弃下一场/帧数据的接收。
 - 如果 `reg_newer` 为 1，则可以紧接着前一次数据继续接收下一场/帧的数据。

硬件工作流程

VIU 的硬件工作流程如图 6-27 所示。

图6-27 VIU 的硬件工作流程



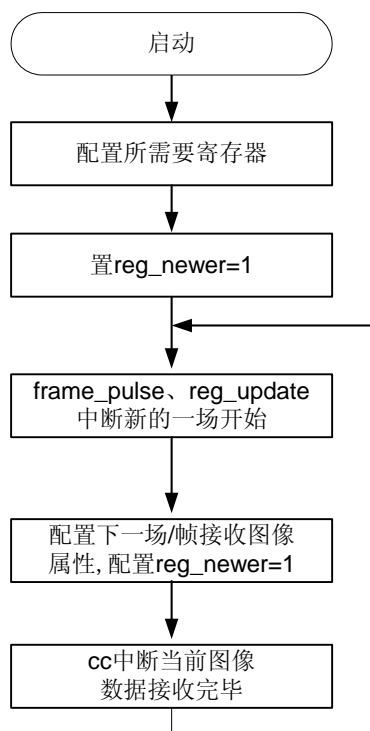


在 BT.656/601 和 digital camera 模式下，每接收完一场/帧规定数据，在下一场的到来时，VIU 将检测 reg_newer 位。如果 reg_newer 位为 1（表示软件已经更新或者确认 VIU 的寄存器），VIU 将自动 load 软件所配置的寄存器值到工作寄存器（工作寄存器软件不可访问），然后将 reg_newer 位清 0，并开始接收下一场/帧数据。否则，只有等到 reg_newer 为 1 且新的一场/帧到来时开始接收数据。

软件配置流程

在中断方式下，软件的操作流程如图 6-28 所示。

图6-28 软件操作流程



在使用 BT.656 模式和 digital camera 接口情况下，不需要配置时序寄存器，需要配置以下寄存器：VIn_CAP_SIZE、VIn_LINE_OFFSET、VIn_YBASE_ADDR、VIn_UBASE_ADDR、VIn_VBASE_ADDR、VIn_CH_CTRL、VIn_CAP_START。

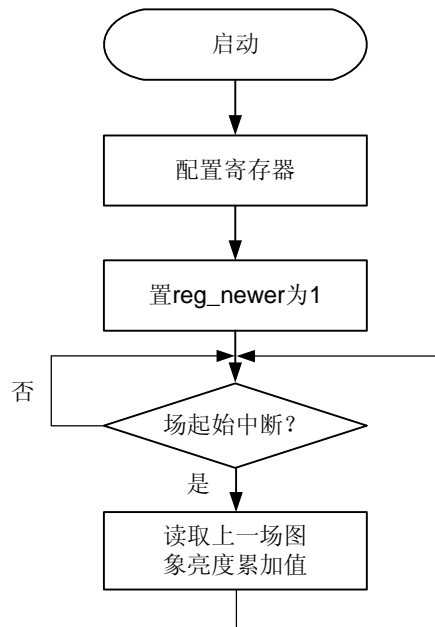
在 BT.601 模式下，还需要配置时序寄存器 VI_Px_VSYNC1、VI_Px_VSYNC2 和 VI_Px_HSYNC1。以 PAL 制为例，时序寄存器的值为寄存器默认值。

时序寄存器包括垂直同步寄存器和水平同步寄存器。

亮度统计值软件读取操作流程

亮度统计值软件读取操作流程如图 6-29 所示。

图6-29 亮度统计值软件读取流程



6.1.6 寄存器概览

VIU 寄存器概览如表 6-7 所示。

表6-7 VIU 寄存器概览（基址是 0x1010_0000）

偏移地址	名称	描述	页码
0x0000+n×0x1000	VIn_PORT_CFG	端口配置寄存器	6-25
0x0004+n×0x1000	VIn_CH_CFG	通道配置寄存器	6-27
0x0008+n×0x1000	VIn_CH_CTRL	通道控制寄存器	6-30
0x000C+n×0x1000	VIn_REG_NEWER	通道寄存器配置完毕寄存器	6-32
0x0010+n×0x1000	VIn_CAP_START	图像获取起始位置寄存器	6-32
0x0014+n×0x1000	VIn_CAP_SIZE	图像获取大小寄存器	6-34
0x0018+n×0x1000	VIn_Y_STORESIZE	Y 分量数据存储大小寄存器	6-34
0x001C+n×0x1000	VIn_U_STORESIZE	Cb 分量数据存储大小寄存器	6-35
0x0020+n×0x1000	VIn_V_STORESIZE	Cr 分量数据存储大小寄存器	6-35
0x0024+n×0x1000	VIn_LINE_OFFSET	图像存储行间距寄存器	6-38
0x0028+n×0x1000	VIn_YBASE_ADDR	Y 分量基地址寄存器	6-38
0x002C+n×0x1000	VIn_UBASE_ADDR	Cb 分量基地址寄存器	6-39



偏移地址	名称	描述	页码
0x0030+n×0x1000	VIn_VBASE_ADDR	Cr 分量基地址寄存器	6-39
0x0034	VI_INT_DLY_CNT	中断延时计数器寄存器	6-40
0x0038+n×0x1000	VIn_INT_EN	中断使能寄存器	6-40
0x003C+n×0x1000	VIn_INT_STATUS	中断状态寄存器	6-42
0x0040+n×0x1000	VIn_RAW_INT	原始中断状态寄存器	6-43
0x0044	VI_INT_INDICATOR	中断指示寄存器	6-45
0x0048+n×0x1000	VI_RAW_INT_INDICATOR	原始中断指示寄存器	6-47
0x004C+n×0x1000	VIn_STATUS	状态寄存器	6-49
0x0050+n×0x1000	VIn_LUM_ADDER	亮度统计寄存器	6-50
0x0054+n×0x1000	VIn_LUM_STRH	亮度拉伸寄存器	6-50
0x0058+n×0x1000	VIn_LUM_DIFF_ADDER	亮度差值累加和寄存器	6-51
0x005C+n×0x1000	VIn_BLOCK0_START	遮挡块 0 起始位置寄存器	6-52
0x0060+n×0x1000	VIn_BLOCK1_START	遮挡块 1 起始位置寄存器	6-52
0x0064+n×0x1000	VIn_BLOCK2_START	遮挡块 2 起始位置寄存器	6-52
0x0068+n×0x1000	VIn_BLOCK3_START	遮挡块 3 起始位置寄存器	6-53
0x006C+n×0x1000	VIn_BLOCK0_SIZE	遮挡块 0 大小寄存器	6-53
0x0070+n×0x1000	VIn_BLOCK1_SIZE	遮挡块 1 大小寄存器	6-54
0x0074+n×0x1000	VIn_BLOCK2_SIZE	遮挡块 2 大小寄存器	6-54
0x0078+n×0x1000	VIn_BLOCK3_SIZE	遮挡块 3 大小寄存器	6-55
0x007C+n×0x1000	VIn_BLOCK0_COLOR	遮挡块 0 填充颜色寄存器	6-55
0x0080+n×0x1000	VIn_BLOCK1_COLOR	遮挡块 1 填充颜色寄存器	6-56
0x0084+n×0x1000	VIn_BLOCK2_COLOR	遮挡块 2 填充颜色寄存器	6-56
0x0088+n×0x1000	VIn_BLOCK3_COLOR	遮挡块 3 填充颜色寄存器	6-57



偏移地址	名称	描述	页码
0x009C+n×0x1000	VIn_ANC0_START	消隐区数据块 0 起始位置寄存器	6-57
0x00A0+n×0x1000	VIn_ANC0_SIZE	消隐区数据块 0 大小寄存器	6-58
0x00A4+n×0x1000	VIn_ANC1_START	消隐区数据块 1 起始位置寄存器	6-59
0x00A8+n×0x1000	VIn_ANC1_SIZE	消隐区数据块 1 大小寄存器	6-59
0x00AC+n×0x1000	VIn_ANC_WORD	消隐区数据寄存器	6-60
0x00CC+n×0x1000	VIn_ANC_WORD	消隐区数据寄存器	6-60
0x00EC	VI_P0_VSYNC1	端口 0 奇场垂直同步寄存器	6-61
0x80EC	VI_P2_VSYNC1	端口 2 奇场垂直同步寄存器	6-62
0x00F0	VI_P0_VSYNC2	端口 0 偶场垂直同步寄存器	6-62
0x80F0	VI_P2_VSYNC2	端口 2 偶场垂直同步寄存器	6-63
0x00F4	VI_P0_HSYNC	端口 0 水平同步寄存器	6-64
0x80F4	VI_P2_HSYNC	端口 2 水平同步寄存器	6-64
0x00F8	VI_PRIO_CONFIG	总线优先级控制寄存器	6-64
0x00FC+n×0x1000	VIn_LUM_COEF0	亮度滤波系数寄存器 0	6-67
0x0100+n×0x1000	VIn_LUM_COEF1	亮度滤波系数寄存器 1	6-67
0x0104+n×0x1000	VIn_LUM_COEF2	亮度滤波系数寄存器 2	6-68
0x0108+n×0x1000	VIn_LUM_COEF3	亮度滤波系数寄存器 3	6-68
0x010C+n×0x1000	VIn_CHROMA_COEF0	色度滤波系数寄存器 0	6-69
0x0110+n×0x1000	VIn_CHROMA_COEF1	色度滤波系数寄存器 1	6-70

注：寄存器 VIn_ABC 中涉及的“n”表示 VI 的通道号，满足 $0 \leq n < 15$ ，且 n 为整数。

6.1.7 寄存器描述

VIn_PORT_CFG

VIn_PORT_CFG 为端口配置寄存器，用于配置外部各个端口的参数。

VIn_PORT_CFG 只有在 n=0、4、8、12 的时有效。



Offset Address	Register Name	Total Reset Value	
0x0000+n×0x1000	VIn_PORT_CFG	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	port_scan_mode port_cap_mode port_mux_mode reserved port_vsync port_vsync_neg port_hsync port_hsync_neg port_en	
Reset	0 0		
Bits	Access	Name	Description
[31:12]	-	reserved	保留。
[11:10]	RW	port_scan_mode	端口数据输入模式。 bit[11] 0: 亮度色度复合模式; 1: 亮度色度分离模式。 bit[10] 0: 隔行输入模式; 1: 逐行输入模式。
[9:8]	RW	port_cap_mode	端口数据接收模式。 00: BT.656 模式; 01: BT.601 模式; 10: 数字摄像头模式; 其他: 保留。 说明: 端口 1 和端口 3 在非高清模式下, 数据接收模式只能是 BT.656 模式。
[7:6]	RW	port_mux_mode	端口多路时分复用输入工作模式。 00: 1 路 D1; 01: 2 路 D1 时分复用数据; 10: 4 路 D1 或 half D1 时分复用数据; 其他: 保留。
[5]	-	reserved	保留。
[4]	RW	port_vsync	管脚 VI_P_VSYNC_FIELD 配置信号。 0: 场号 (奇场或偶场) 或行有效信号; • BT.601 模式下表示场号; • camera 接口情况下表示行有效信号。 1: 垂直同步脉冲。



			说明：16-bit 同步接口不需要配置该 bit。
[3]	RW	port_vsync_neg	管脚 VI_P_VSYNC_FIELD 极性配置。 0: 高电平有效。 在脉冲模式下 (port_vsync=1)，正脉冲表示同步脉冲； 在场号模式下 (port_vsync=0)，高电平表示偶数场，低电平表示奇数场； 行有效模式下，高电平表示行有效。 1: 低电平有效。 在脉冲模式下 (port_vsync=1)，负脉冲表示同步脉冲； 在场号模式下 (port_vsync=0)，低电平表示偶数场，高电平表示奇数场； 行有效模式下，低电平表示行有效。
[2]	RW	port_hsync	管脚 VI_P_HSYNC_VD 配置信号。 0: VI_P_HSYNC_VD 为数据有效信号； 1: VI_P_HSYNC_VD 为水平同步脉冲信号。 说明：16-bit 同步接口该比特配置为 1。
[1]	RW	port_hsync_neg	管脚 VI_P_HSYNC_VD 极性配置。 0: 高电平有效。 在脉冲模式下 (port_hsync=1)，正脉冲表示同步脉冲； 在数据有效模式下 (port_hsync=0)，高电平表示数据有效。 1: 低电平有效。 在脉冲模式下 (port_hsync=1)，负脉冲表示同步脉冲； 在数据有效模式下 (port_hsync=0)，低电平表示数据有效。
[0]	RW	port_en	端口使能。 0: 禁止； 1: 使能。

VIn_CH_CFG

VIn_CH_CFG 为通道配置寄存器，用于配置内部各个通道的参数。



Offset Address	Register Name	Total Reset Value	
0x0004+n×0x1000	VIn_CH_CFG	0x0000_1480	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved fix_code yc_channel seav_f_neg chrom_swap ch_id_en ch_id reserved even_line_sel odd_line_sel correct_en down_scaling chroma_resample store_method cap_sel cap_seq little_endian store_mode data_width		
Reset	0 1 0 1 0 0 0 1 0 0 0 0 0 0 0		
Bits	Access	Name	Description
[31:27]	-	reserved	保留。
[26]	RW	fix_code	BT.656 定时基准码最高 bit 配置。 0: 固定为 1; 1: 固定为 0。
[25]	RW	yc_channel	当前通道 Y/C 配置。 0: 当前通道为亮度 Y 通道; 1: 当前通道为色度 C 通道。 说明: 只在 Y/C 分离模式下, 该寄存器才有效。
[24]	RW	seav_f_neg	BT.656 定时基准码场指示位 (F) 极性。 0: 1st field:F=0, 2nd field:F=1 (标准); 1: 1st field:F=1, 2nd field:F=0 (非标准)。
[23]	RW	chrom_swap	通道数据 swap 功能使能。 0: 按照正常方式输出; 1: halfword 内字节倒序。 说明: 该功能只用于 Y/C 分开输入的高清模式, 通道 4 和通道 12 用于传输色度, 此时色度的存储顺序可配置调整为: 0: 色度存储顺序为 Cr1Cb1Cr0Cb0; 1: 色度存储顺序为 Cb1Cr1Cb0Cr0。
[22]	RW	ch_id_en	通道号检测使能。 0: 禁止; 1: 使能。 说明: 该功能只用于对应端口工作在 2 路或 4 路时分复用模式下。



[21:20]	RW	ch_id	<p>通道号。</p> <p>00: 选择通道号为 00 的数据; 01: 选择通道号为 01 的数据; 10: 选择通道号为 10 的数据; 11: 选择通道号为 11 的数据。</p> <p>说明: 当通道号检测使能时, 只有等于该通道号的数据才能够进入该通道。</p>
[19:17]	-	reserved	保留。
[16:15]	RW	even_line_sel	<p>偶场图像色度数据获取行选择。</p> <p>00: 仅对奇数行采集; 01: 仅对偶数行采集; 10: 对奇数行和偶数行都采集; 其他: 保留。</p> <p>说明: 行号是从 0 开始计数的; 此隔行获取功能只对色度分量有效, 亮度奇偶行都采集, 用于获得 YcbCr 4:2:0 的数据; 输入为帧模式 (DC 模式和逐行高清模式) 时由 even_line_sel 控制色度丢行; 存储模式为帧模式时, 不允许配置成奇场只采集奇行, 偶场只采集偶行, 推荐配置为奇场只采集偶行, 偶场只采集奇行; 高清模式下只需配置通道 4 或通道 12。</p>
[14:13]	RW	odd_line_sel	<p>奇场图像色度数据获取行选择。</p> <p>00: 仅对奇数行采集; 01: 仅对偶数行采集; 10: 对奇数行和偶数行都采集; 其他: 保留。</p> <p>说明: 行号是从 0 开始计数的; 此隔行获取功能只对色度分量有效, 亮度奇偶行都采集, 用于获得 YcbCr 4:2:0 的数据; 输入数据为帧模式 (DC 模式和逐行高清模式) 时 odd_line_sel 无效; 存储模式为帧模式时, 不允许配置成奇场只采集奇行, 偶场只采集偶行, 推荐配置为奇场只采集偶行, 偶场只采集奇行; 高清模式下只需配置通道 4 或通道 12。</p>
[12]	RW	correct_en	<p>SAV (Start of Active Video) /EAV (End of Activevideo) 数据校验使能。</p> <p>0: 禁止; 1: 使能。</p>
[11]	RW	down_scaling	<p>水平 1/2 缩放。</p> <p>0: 禁止; 1: 使能。</p> <p>说明: 在 Y/C 分开输入的高清模式下, 不支持此功能。</p>



[10]	RW	chroma_resample	色度重新采样。 0: 禁止; 1: 使能。co-sited 到 interspersed 转换。 说明: 色度重新采样只支持 co-sited 到 interspersed 转换, 在 Y/C 分开输入的高清模式下, 不支持此功能。
[9:8]	RW	store_method	存储方式 (Store Method)。 00: planar Y/Cb/Cr; 01: semi-planar YCbCr; 10: package YCbCr4:2:2; 11: raw data, 只用于 Y/C 分开输入的高清模式。
[7:6]	RW	cap_sel	图像数据获取场选择。 00: 仅对奇数场 (顶场) 采集; 01: 仅对偶数场 (底场) 采集; 10: 对奇数场和偶数场都采集; 其他: 保留。
[5:4]	RW	cap_seq	YCbCr 输入顺序支持位。 00: CbYCrY; 01: CrYCbY; 10: YCbYCr; 11: YCrYCb。
[3]	-	reserved	保留。
[2]	RW	store_mode	存储模式 (Store Mode)。 0: 场存储模式; 1: 帧存储模式。 说明: 若端口输入为逐行模式时, 该寄存器需要配置成场存储模式。
[1:0]	RW	data_width	数据位宽 (Data Width)。 00: 8bit; 10: 10bit。 其他: 保留。

VIIn_CH_CTRL

VIIn_CH_CTRL 为控制寄存器, 控制数据获取的开始和结束。VIIn_CH_CTRL 可实现 VIU 的使能和 reg_newer 中断控制。



Offset Address	Register Name	Total Reset Value	
0x0008+n×0x1000	VIn_CH_CTRL	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved debug_en lum_strh_en anc1_en anc0_en reserved block3_en block2_en block1_en block0_en reserved ch_en		
Reset	0 0		
Bits	Access	Name	Description
[31:16]	-	reserved	保留。
[15]	RW	debug_en	测试模式使能。 0: 禁止; 1: 使能。 测试使能未打开时测试寄存器回读值为 0。
[14]	RW	lum_strh_en	亮度拉伸使能信号。 0: 不进行亮度拉伸; 1: 进行亮度拉伸。 说明: 亮度和色度分离输入模式下不能进行亮度拉伸。
[13]	RW	anc1_en	消隐区数据块 1 捕获使能。 0: 禁止; 1: 使能。
[12]	RW	anc0_en	消隐区数据块 0 捕获使能。 0: 禁止; 1: 使能。
[11:8]	-	reserved	保留。
[7]	RW	block3_en	块屏蔽 3 使能。 0: 禁止; 1: 使能。
[6]	RW	block2_en	块屏蔽 2 使能。 0: 禁止; 1: 使能。
[5]	RW	block1_en	块屏蔽 1 使能。 0: 禁止; 1: 使能。



[4]	RW	blcok0_en	块屏蔽 0 使能。 0: 禁止; 1: 使能。
[3:1]	-	reserved	保留。
[0]	RW	ch_en	VIU 通道使能。 0: 禁止; 1: 使能。 说明: 任何模式切换 (多路切换或者 BT.656 模式到 BT.601 模式切换等) 时, 要先将该 bit 置 0, 切换完毕再置 1。

VIn_REG_NEWER

VIn_REG_NEWER 为通道寄存器配置完毕寄存器, 用来指示通道下一场/帧的寄存器是否配置完毕。

Offset Address	Register Name	Total Reset Value	
0x000C+n×0x1000	VIn_REG_NEWER	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	reg_newer	
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RW	reg_newer	下一场/帧寄存器准备完毕。 0: 下一需要接收的场/帧寄存器未准备好, 硬件将放弃下一场/帧的接收; 1: 下一需要接收的场/帧寄存器已准备好, 在检测到下一场/帧的场/帧开始时, 硬件开始接收下一场数据。 说明: VIU 硬件在自动更新内部工作寄存器后, 将自动清零该位。

VIn_CAP_START

VIn_CAP_START 为图像获取起始位置寄存器, 配置图像获取的起始位置。

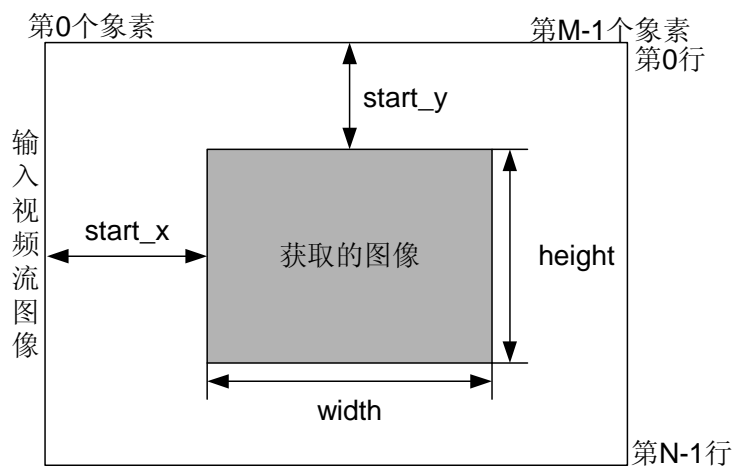
VIn_CAP_SIZE 为图像获取大小寄存器，配置图像获取的大小。

VIn_CAP_START 和 VIn_CAP_SIZE 描述了从输入视频图像中获取矩形图像，如图 6-30 所示。

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

width 和 start_x 以输入像素为单位（即亮度像素为单位）；在 YCbCr 4:2:2 图像数据获取时，height 和 start_y 以行为单位。

图6-30 图像获取参数示意图



注：start_y 和 start_x 在 VIn_CAP_START 寄存器中分别占用 12bit 宽度。
width 和 height 在 VIn_CAP_SIZE 寄存器中分别占用 12 位宽度。

	Offset Address	Register Name	Total Reset Value
	0x0010+n×0x1000	VIn_CAP_START	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	start_y	start_x
Reset	0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	start_y	开始获取图像的行号。
[11:0]	RW	start_x	开始获取图像的像素号。



VIn_CAP_SIZE

VIn_CAP_SIZE 为图像获取大小寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0014+n×0x1000	VIn_CAP_SIZE	0x0012_02d0
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved height width		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	height	获取图像的高度（以行为单位）。 帧存储模式时，获取图像高度为帧图像实际高度的 1/2，VI 捕获两场组成一帧图像。 VI 支持的最小捕获高度为 1 行。
[11:0]	RW	width	获取图像一行的宽度（以像素为单位）。 VI 支持的最小捕获宽度为 4 个像素。

VIn_Y_STORESIZE

VI_Y_STORESIZE 为 Y 分量数据存储大小寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0018+n×1000	VIn_Y_STORESIZE	0x000F_00A0
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved y_height y_width		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	y_height	Y 分量数据存储高度（以行为单位）。 说明：工作在亮度,色度分离模式下,如果色度通道输出 YCbCr4:2:0 数据的时候,需要将色度的高度配置为亮度的一半。
[11:0]	RW	y_width	Y 分量数据存储宽度减 1（以 128bit 为单位，行尾不够一个 128 比特，以一个 128 比特算）。



VIn_U_STORESIZE

VIn_U_STORESIZE 为 Cb 分量数据存储大小寄存器。在 package 存储模式时该寄存器不适用；semi-planar YCbCr 数据格式时，该寄存器表示色度存储大小；planar YCbCr 数据格式时，该寄存器表示 Cb 分量存储大小。

Offset Address	Register Name	Total Reset Value	
0x001C+n×0x1000	VIn_U_STORESIZE	0x000F_00A0	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	c_height	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	c_height	semi-planar YCbCr 模式下，色度分量数据存储高度（以行为单位）。 planar YCbCr 模式下，Cb 分量数据存储高度（以行为单位）。 在 package 存储模式和 raw data 接收模式下无效。
[11:0]	RW	c_width	semi-planar YCbCr 模式下，色度分量数据存储宽度减 1（以 128bit 为单位，行尾不够一个 128bit，以一个 128bit 算）。 planar YCbCr 模式下，Cb 分量数据存储宽度（以 word 为单位，不够一个 word，以一个 word 算）。 在 package 存储模式和 raw data 接收模式下无效。

VIn_V_STORESIZE

VIn_V_STORESIZE 为 Cr 分量数据存储大小寄存器。

Offset Address	Register Name	Total Reset Value	
0x0020+n×0x1000	VIn_V_STORESIZE	0x000F00A0	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	v_height	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	v_height	Cr 分量数据存储高度（以行为单位）。



[11:0]	RW	v_width	Cr 分量数据存储宽度减 1（以 128bit 为单位，行尾不够一个 128bit，以一个 128bit 算）。 在 semi-planar YCbCr 存储模式、package YCbCr 存储模式和 raw data 接收模式下无效。
--------	----	---------	--

当接收 YCbCr4:2:2 数据时，数据存储大小计算步骤如下：

步骤 1 判断 cap_width 的奇偶。若偶数不变，奇数则减 1。

```
cap_width=cap_width-cap_width%2
```

步骤 2 判断是否 1/2 缩放。

```
down_scaling?
yes:cap_width=cap_width/2
no:cap_width=cap_width
```

步骤 3 不同存储模式下，数据存储大小的计算。

1. 存储模式为 Y/Cb/Cr。

数据位宽为 8bit:

```
y_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
c_width = if(cap_width%32==0)
    cap_width/32
else
    cap_width/32+1
v_width=if(cap_width%32==0)
    cap_width/32
else
    cap_width/32+1
```

数据位宽为 10bit:

```
y_width=if(cap_width%8==0)
    cap_width/8
else
    cap_width/8+1
c_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
v_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
```



2. 存储模式为 **planar Y/C**。

数据位宽为 **8bit**:

```
y_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
c_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
```

数据位宽为 **10bit**:

```
y_width=if(cap_width%8==0)
    cap_width/8
else
    cap_width/8+1
c_width=if(cap_width%8==0)
    cap_width/8
else
    cap_width/8+1
```

3. 存储模式为 **package** 模式。

数据位宽为 **8bit**:

```
y_width=if(cap_width%8==0)
    cap_width/8
else
    cap_width/8+1
```

数据位宽为 **10bit**:

```
y_width=if(cap_width%4==0)
    cap_width/4
else
    cap_width/4+1
```

4. 存储模式为 **raw data**。

数据位宽为 **8bit**:

```
y_width=if(cap_width%16==0)
    cap_width/16
else
    cap_width/16+1
```

数据位宽为 **10bit**:

```
y_width=if(cap_width%8==0)
    cap_width/8
else
    cap_width/8+1
```

步骤 4 减一处理。

```
y_width=y_width-1
```



```
c_width=c_width-1
```

```
v_width=v_width-1
```

----结束

VIn_LINE_OFFSET

VIn_LINE_OFFSET 为图像存储行间距寄存器，配置获取数据存储行偏移的大小。以 word 为单位，各路分别控制。

	Offset Address	Register Name	Total Reset Value			
	0x0024+n×0x1000	VIn_LINE_OFFSET	0x0000_0000			
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	<table border="1" style="width:100%; text-align:center;"> <tr> <td style="width:33%;">yline_offset</td> <td style="width:33%;">uline_offset</td> <td style="width:33%;">vline_offset</td> </tr> </table>			yline_offset	uline_offset	vline_offset
yline_offset	uline_offset	vline_offset				
Reset	0 0					
	Bits	Access	Name			
	[31:20]	RW	yline_offset			
	[19:10]	RW	uline_offset			
	[9:0]	RW	vline_offset			
	Description					
	以 128bit 为单位（真正的偏移为 yline_offset 左移 4bit）。raw data 存储模式下，仅该 stride 有效。package YCbCr4:2:2 存储模式下，仅该 stride 有效。planar Y/C 或者 Y/Cb/Cr 存储模式下，表示 Y 的行 stride。					
	以 128bit 为单位（真正的偏移为 uline_offset 左移 4bit）。raw data 存储模式下，无效。package YCbCr4:2:2 存储模式下，无效。planar Y/C 存储模式下，表示 C 分量的行 stride。planar Y/Cb/Cr 存储模式下，表示 Cb 分量行 stride。					
	以 128bit 为单位（真正的偏移为 vline_offset 左移 4bit）。raw data 存储模式下，无效。package YCbCr4:2:2 存储模式下，无效。planar Y/C 存储模式下，无效。planar Y/Cb/Cr 存储模式下，表示 Cr 分量行 stride。					

VIn_YBASE_ADDR

VIn_YBASE_ADDR0 为 Y 分量基地址 0 寄存器，配置 Y 分量的存放开始地址。最后 3bits 为 0。



Offset Address	Register Name	Total Reset Value	
0x0028+n×0x1000	VIn_YBASE_ADDR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	vi_ybase_addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	vi_ybase_addr	Y 通道基地址。 最低 4 位为 0，128bit 对齐。 raw data 存储模式下，仅该地址有效。 package YCbCr4:2:2 存储模式下，仅该地址有效。 planar Y/C 或者 planar Y/Cb/Cr 存储模式下，表示 Y 的首地址。

VIn_UBASE_ADDR

VIn_UBASE_ADDR 为 Cb 分量基地址 0 寄存器，配置 Cb 分量的存放开始地址。最后 3bits 为 0。

Offset Address	Register Name	Total Reset Value	
0x002C+n×0x1000	VIn_UBASE_ADDR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	vi_ubase_addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	vi_ubase_addr	Cb 通道基地址。 最低 4 位为 0，128 比特对齐。 raw data 存储模式下，无效。 package YCbCr4:2:2 存储模式下，无效。 planar Y/C 存储模式下，表示 C 分量的首地址。 planar Y/Cb/Cr 或者 Y/Cb/Cr 存储模式下，表示 Cb 分量的首地址。

VIn_VBASE_ADDR

VIn_VBASE_ADDR0 为 Cr 分量基地址 0 寄存器，配置 Cr 分量的存放开始地址。最后 3bits 为 0。



Offset Address	Register Name	Total Reset Value	
0x0030+n×0x1000	VIn_VBASE_ADDR	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	vi_vbase_addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	vi_vbase_addr	Cr 通道基地址。 最低 4 位为 0，128 比特对齐。 raw data 存储模式下，无效。 package YCbCr4:2:2 存储模式下，无效。 planar Y/C 存储模式下，无效。 planar Y/Cb/Cr 存储模式下，表示 Cr 分量的首地址。

VI_INT_DLY_CNT

VI_INT_DLY_CNT 为中断延時計数器寄存器。

Offset Address	Register Name	Total Reset Value	
0x0034	VI_INT_DLY_CNT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	int_dly_cnt		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	int_dly_cnt	中断延時計数器。 计数器以总线时钟为单位，当计数到该寄存器值时，若 16 个通道有中断发生，则 VIU 会上报一个中断。

VIn_INT_EN

VIn_INT_EN 为中断使能寄存器。



Offset Address		Register Name		Total Reset Value																																								
0x0038+n×0x1000		VIn_INT_EN		0x0000_0000																																								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Name	reserved																							chdiv_err_int_en	ntsc_pal_trans_int_en	frame_pulse_int_en	reg_update_int_en	proc_err_int_en	err_int_en	field_throw_int_en	buf_ovf_int_en	cc_int_en												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Bits	Access	Name	Description																																									
[31:9]	-	reserved	保留。																																									
[8]	RW	chdiv_err_int_en	通道分配错误指示中断使能。 0: 屏蔽; 1: 使能。																																									
[7]	RW	ntsc_pal_trans_int_en	制式转化中断使能。 0: 屏蔽; 1: 使能。																																									
[6]	RW	frame_pulse_int_en	场起始中断使能。 0: 屏蔽; 1: 使能。																																									
[5]	RW	reg_update_int_en	寄存器更新中断使能。 0: 屏蔽; 1: 使能。																																									
[4]	RW	proc_err_int_en	BT.656 模式下, 保护比特位错误中断使能。 0: 屏蔽; 1: 使能。																																									
[3]	RW	err_int_en	总线错误中断使能。 0: 屏蔽; 1: 使能。																																									
[2]	RW	field_throw_int_en	场/帧丢失中断使能。 0: 屏蔽; 1: 使能。																																									



[1]	RW	buf_ovf_int_en	内部 FIFO 溢出错误中断使能。 0: 屏蔽; 1: 使能。
[0]	RW	cc_int_en	数据获取完毕中断使能。 0: 一场图像获取完毕中断; 1: 一帧图像获取完毕中断。

VIn_INT_STATUS

VIn_INT_STATUS 为中断状态寄存器，对中断状态寄存器的相应位写 1，则清除该中断位。各通道分别控制，用于软硬交互，可屏蔽。

Offset Address	Register Name	Total Reset Value	
0x003C+n×0x1000	VIn_INT_STATUS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	chdiv_err_int ntsc_pal_trans_int frame_pulse_int reg_update_int proc_err_int error_int field_throw_int buf_ovf_int cc_int	
Reset	0 0		
Bits	Access	Name	Description
[31:9]	-	reserved	保留。
[8]	WC	chdiv_err_int	通道分配错误指示中断状态。 0: 无中断; 1: 有中断。
[7]	WC	ntsc_pal_trans_int	制式变换中断状态。 0: 无中断; 1: 有中断。 说明：制式变换中断，制式发生变化就会上报，不限于 N 制和 P 制，只要输入数据有效行数发生变换即认为制式变换。
[6]	WC	frame_pulse_int	场起始中断状态。 0: 无中断; 1: 有中断。



[5]	WC	reg_update_int	工作寄存器更新中断状态。 0: 无中断; 1: 有中断。 当 VIn_CH_CFG [store_mode]为 1 时, 表示一帧图像获取完毕中断。 当 VIn_CH_CFG [store_mode]为 0 时, 表示一场图像获取完毕中断。
[4]	WC	proc_err_int	保护位错误中断状态 (BT.656 模式)。 0: 无中断; 1: 有中断。
[3]	WC	error_int	AHB 总线错误中断状态。 0: 无中断; 1: 有中断。
[2]	WC	field_throw_int	场数据丢失中断状态。 0: 无中断; 1: 有中断。 说明: 配置为只获取奇场或者只获取偶场时, 硬件不会上报场数据丢失。
[1]	WC	buf_ovf_int	内部缓冲 FIFO 溢出中断状态。 0: 无中断; 1: 有中断。
[0]	WC	cc_int	当前图像数据获取完毕中断状态。 0: 无中断; 1: 有中断。 当 VIn_CH_CFG [store_mode]为 0 时, 表示一场图像获取完毕中断。 当 VIn_CH_CFG [store_mode]为 1 时, 表示一帧图像获取完毕中断。

VIn_RAW_INT

VIn_RAW_INT 为原始中断状态寄存器。清除为对中断状态寄存器的相应 bit 写 1, 则清除该中断比特位, 各通道分别控制, 用于 DEBUG, 不可屏蔽。



Offset Address	Register Name	Total Reset Value	
0x0040+n×0x1000	VIn_RAW_INT	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	chdiv_err_raw_int ntsc_pal_trans_raw_int frame_pulse_raw_int reg_update_raw_int proc_err_raw_int error_raw_int field_throw_raw_int buf_ovf_raw_int cc_raw_int	
Reset	0 0		
Bits	Access	Name	Description
[31:9]	-	reserved	保留。
[8]	RO	chdiv_err_raw_int	通道分配错误指示原始中断状态。 0: 无中断; 1: 有中断。
[7]	RO	ntsc_pal_trans_raw_int	制式变换原始中断状态。 0: 无原始中断; 1: 有原始中断。 说明: 制式变换中断, 制式发生变化就会上报, 不限于 N 制和 P 制, 只要输入数据有效行数发生变换即认为制式变换。
[6]	RO	frame_pulse_raw_int	场起始原始中断状态。 0: 无中断; 1: 有中断。
[5]	RO	reg_update_raw_int	工作寄存器更新原始中断状态。 0: 无中断; 1: 有中断。
[4]	RO	proc_err_raw_int	保护位错误原始中断状态 (BT.656 模式)。 0: 无中断; 1: 有中断。
[3]	RO	error_raw_int	AHB 总线错误原始中断状态。 0: 无中断; 1: 有中断。
[2]	RO	field_throw_raw_int	场数据丢失原始中断状态。 0: 无中断; 1: 有中断。



[1]	RO	buf_ovf_raw_int	内部缓冲 FIFO 溢出原始中断状态。 0: 无中断; 1: 有中断。
[0]	RO	cc_raw_int	当前图像数据获取完毕原始中断状态 (cc:capture completion)。 0: 一场图像获取完毕原始中断; 1: 一帧图像获取完毕原始中断。 当 store_mode 为 0 时, 表示一场图像获取完毕中断。 当 store_mode 为 1 时, 表示一帧图像获取完毕中断。

VI_INT_INDICATOR

VI_INT_INDICATOR 为中断指示寄存器, 用于指示 16 个通道中, 哪个或哪些通道发生中断。

	Offset Address	Register Name	Total Reset Value																														
	0x0044+n×0x1000	VI_INT_INDICATOR	0x0000_0000																														
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																
Name	reserved																ch15_int_indicator	ch14_int_indicator	ch13_int_indicator	ch12_int_indicator	ch11_int_indicator	ch10_int_indicator	ch9_int_indicator	ch8_int_indicator	ch7_int_indicator	ch6_int_indicator	ch5_int_indicator	ch4_int_indicator	ch3_int_indicator	ch2_int_indicator	ch1_int_indicator	ch0_int_indicator	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																														
[31:16]	-	reserved	保留。																														
[15]	RO	ch15_int_indicator	通道 15 中断指示。 0: 无中断; 1: 有中断。																														
[14]	RO	ch14_int_indicator	通道 14 中断指示。 0: 无中断; 1: 有中断。																														
[13]	RO	ch13_int_indicator	通道 13 中断指示。 0: 无中断; 1: 有中断。																														



[12]	RO	ch12_int_indicator	通道 12 中断指示。 0: 无中断; 1: 有中断。
[11]	RO	ch11_int_indicator	通道 11 中断指示。 0: 无中断; 1: 有中断。
[10]	RO	ch10_int_indicator	通道 10 中断指示。 0: 无中断; 1: 有中断。
[9]	RO	ch9_int_indicator	通道 9 中断指示。 0: 无中断; 1: 有中断。
[8]	RO	ch8_int_indicator	通道 8 中断指示。 0: 无中断; 1: 有中断。
[7]	RO	ch7_int_indicator	通道 7 中断指示。 0: 无中断; 1: 有中断。
[6]	RO	ch6_int_indicator	通道 6 中断指示。 0: 无中断; 1: 有中断。
[5]	RO	ch5_int_indicator	通道 5 中断指示。 0: 无中断; 1: 有中断。
[4]	RO	ch4_int_indicator	通道 4 中断指示。 0: 无中断; 1: 有中断。
[3]	RO	ch3_int_indicator	通道 3 中断指示。 0: 无中断; 1: 有中断。
[2]	RO	ch2_int_indicator	通道 2 中断指示。 0: 无中断; 1: 有中断。



[1]	RO	ch1_int_indicator	通道 1 中断指示。 0: 无中断; 1: 有中断。
[0]	RO	ch0_int_indicator	通道 0 中断指示。 0: 无中断; 1: 有中断。

VI_RAW_INT_INDICATOR

VI_RAW_INT_INDICATOR 为原始中断指示寄存器，用于指示 16 个通道中，哪些通道发生原始中断。

	Offset Address	Register Name	Total Reset Value																													
	0x0048+n×0x1000	VI_RAW_INT_INDICATOR	0x0000_0000																													
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
Name	reserved																ch15_raw_int_indicator	ch14_raw_int_indicator	ch13_raw_int_indicator	ch12_raw_int_indicator	ch11_raw_int_indicator	ch10_raw_int_indicator	ch9_raw_int_indicator	ch8_raw_int_indicator	ch7_raw_int_indicator	ch6_raw_int_indicator	ch5_raw_int_indicator	ch4_raw_int_indicator	ch3_raw_int_indicator	ch2_raw_int_indicator	ch1_raw_int_indicator	ch0_raw_int_indicator
Reset	0 0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name	Description																													
[31:16]	-	reserved	保留。																													
[15]	RO	ch15_raw_int_indicator	通道 15 原始中断指示。 0: 无中断; 1: 有中断。																													
[14]	RO	ch14_raw_int_indicator	通道 14 原始中断指示。 0: 无中断; 1: 有中断。																													
[13]	RO	ch13_raw_int_indicator	通道 13 原始中断指示。 0: 无中断; 1: 有中断。																													
[12]	RO	ch12_raw_int_indicator	通道 12 原始中断指示。 0: 无中断; 1: 有中断。																													



[11]	RO	ch11_raw_int_indicator	通道 11 原始中断指示。 0: 无中断; 1: 有中断。
[10]	RO	ch10_raw_int_indicator	通道 10 原始中断指示。 0: 无中断; 1: 有中断。
[9]	RO	ch9_raw_int_indicator	通道 9 原始中断指示。 0: 无中断; 1: 有中断。
[8]	RO	ch8_raw_int_indicator	通道 8 原始中断指示。 0: 无中断; 1: 有中断。
[7]	RO	ch7_raw_int_indicator	通道 7 原始中断指示。 0: 无中断; 1: 有中断。
[6]	RO	ch6_raw_int_indicator	通道 6 原始中断指示。 0: 无中断; 1: 有中断。
[5]	RO	ch5_raw_int_indicator	通道 5 原始中断指示。 0: 无中断; 1: 有中断。
[4]	RO	ch4_raw_int_indicator	通道 4 原始中断指示。 0: 无中断; 1: 有中断。
[3]	RO	ch3_raw_int_indicator	通道 3 原始中断指示。 0: 无中断; 1: 有中断。
[2]	RO	ch2_raw_int_indicator	通道 2 原始中断指示。 0: 无中断; 1: 有中断。
[1]	RO	ch1_raw_int_indicator	通道 1 原始中断指示。 0: 无中断; 1: 有中断。



[0]	RO	ch0_raw_int_indicator	通道 0 原始中断指示。 0: 无中断; 1: 有中断。
-----	----	-----------------------	------------------------------------

VIn_STATUS

VIn_STATUS 为状态寄存器。

	Offset Address 0x004C+n×0x1000	Register Name VIn_STATUS	Total Reset Value 0x0000_0020
Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Name	reserved	act_height	vi_busy field2 snooze proc_err bus_err frame_loss buf_ovf image_done
Reset	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0 0
Bits	Access	Name	Description
[31:20]	-	reserved	保留。
[19:8]	RO	act_height	检测到的一场中有效像素数据的行数。
[7]	RO	vi_busy	VIU 当前工作状态。 0: 空闲; 1: 忙。
[6]	RO	field2	当前接收为偶数场。 0: 奇数场; 1: 偶数场。
[5]	RO	snooze	当前 VIU 处于睡眠状态。 0: 非睡眠; 1: 睡眠。
[4]	RO	proc_err	保护位错误状态。 0: 正确; 1: 错误。
[3]	RO	bus_err	总线错误状态。 0: 正确; 1: 错误。



[2]	RO	frame_loss	VIU 丢失一场数据。 0: 未丢失。 1: 丢失。
[1]	RO	buf_ovf	VIU 内部 buffer 溢出。 0: 未溢出; 1: 溢出。
[0]	RO	image_done	VIU 接收完毕当前场数据。 0: 未接受完毕; 1: 接受完毕。

VIIn_LUM_ADDER

VIIn_LUM_ADDER 为亮度统计寄存器，统计整个活动图像的亮度信息，而不是获取图像的区域亮度。

	Offset Address	Register Name	Total Reset Value
	0x0050+n×0x1000	VIIn_LUM_ADDER	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	lum_adder		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	vi_lum_adder
			亮度数值累加。

当打开图像块遮挡功能后，亮度统计寄存器对被遮挡的图像块亮度不敏感，即统计被遮挡区域时，使用的是被遮挡图像的亮度，而不是遮挡块的亮度。

VIIn_LUM_STRH

亮度拉伸寄存器用于调整输入图像亮度值，调节公式如下：

$$luma_out = (sign(luma_in - m0) * (k * |luma_in - m0| + 64) >> 7) + m0$$

其中 $m0$ 为上一帧或场的平均亮度，可以用亮度累加值（寄存器 VIIn_LUM_ADDER）除以输入图像像素个数得到（输入图像像素个数为接口接收到的像素个数，不是捕获图像像素个数）。

k 为拉伸系数，正常范围为 64~196，64 是缩小相邻像素亮度差，196 是放大相邻像素亮度差，128 是保持原值不变。

该寄存器中的 $m1$ 建议配置成与 $m0$ 相同的值。



	Offset Address	Register Name	Total Reset Value
	0x0054+n×0x1000	VIn_LUM_STRH	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	m1	reserved
Reset	0 0	m0	k
	Bits	Access	Name
	[31:30]	-	reserved
	[29:20]	RW	m1
	[19:18]	-	reserved
	[17:8]	RW	m0
	[7:0]	RW	k
			Description
			保留。
			亮度均值 1。
			保留。
			亮度均值 0。
			亮度拉伸系数。

VIn_LUM_DIFF_ADDER

亮度差值累加寄存器，输入图像亮度与亮度拉伸寄存器的亮度均值 m1 之差的绝对值之和。亮度差值累加和统计整个活动图像的亮度信息，而不只是获取图像的区域亮度。帧存储模式时，寄存器按帧进行统计；场存储模式时，寄存器按场进行统计。输入图像亮度为缩放之前的亮度；可参考该寄存器来决定亮度拉伸功能是否打开：

- 当该寄存器值较大时，说明上一帧或场各个像素之间亮度差较大，若要平滑各个像素之间的亮度差，可将亮度拉伸寄存器（VIn_LUM_STRH）中的 k 值配置成小于 128 的值。
- 当该寄存器值较小时，说明上一帧或场各个像素之间亮度差较小，若要增加各个像素之间的亮度差，可将亮度拉伸寄存器（VIn_LUM_STRH）中的 k 值配置成大于 128 的值。

	Offset Address	Register Name	Total Reset Value
	0x0058+n×0x1000	VIn_LUM_DIFF_ADDER	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	lum_diff_adder		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	lum_diff_adder
			Description
			亮度差值累加（高清 720p 模式时，通道 0 亮度差值累加为输入图像的亮度差值累加，通道 4 的亮度差值累加无效）。亮度差值累加和寄存器值要在下一帧/场开始时才写入寄存器。



VIn_BLOCK0_START

VIn_BLOCK0_START 为遮挡块 0 起始位置寄存器。

	Offset Address	Register Name	Total Reset Value
	0x005C+n×0x1000	VIn_BLOCK0_START	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block0_starty	block0_stone_startx
Reset	0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	block0_starty	开始图像块 0 填充的行号。
[11:0]	RW	block0_startx	开始图像块 0 填充的像素号。

VIn_BLOCK1_START

VIn_BLOCK1_START 为遮挡块 1 起始位置寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0060+n×0x1000	VIn_BLOCK1_START	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block1_starty	block1_stone_startx
Reset	0 0		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	block1_starty	开始图像块 1 填充的行号。
[11:0]	RW	block1_startx	开始图像块 1 填充的像素号。

VIn_BLOCK2_START

VIn_BLOCK2_START 为遮挡块 2 起始位置寄存器。



	Offset Address	Register Name	Total Reset Value
	0x0064+n×0x1000	VIn_BLOCK2_START	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block2_stone_starty	block2_stone_startx
Reset	0 0		
Bits	Access	name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	block2_starty	开始图像块 2 填充的行号。
[11:0]	RW	block2_startx	开始图像块 2 填充的像素号。

VIn_BLOCK3_START

VIn_BLOCK3_START 为遮挡块 3 起始位置寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0068+n×0x1000	VIn_BLOCK3_START	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block3_stone_starty	block3_stone_startx
Reset	0 0		
Bits	Access	name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	block3_starty	开始图像块 3 填充的行号。
[11:0]	RW	block3_startx	开始图像块 3 填充的像素号。

VIn_BLOCK0_SIZE

VIn_BLOCK0_SIZE 为遮挡块 0 大小寄存器。



	Offset Address	Register Name	Total Reset Value
	0x006C+n×0x1000	VIn_BLOCK0_SIZE	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block0_stone_height	block0_stone_width
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:12]	RW	block0_height
	[11:0]	RW	block0_width
			Description
			保留。
			图像块 0 填充的高度（以行为单位）。
			图像块 0 填充的宽度（以像素为单位）。

VIn_BLOCK1_SIZE

VIn_BLOCK1_SIZE 为遮挡块 1 大小寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0070+n×0x1000	VIn_BLOCK1_SIZE	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block1_stone_height	block1_stone_width
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:12]	RW	block1_height
	[11:0]	RW	block1_width
			Description
			保留。
			图像块 1 填充的高度（以行为单位）。
			图像块 1 填充的宽度（以像素为单位）。

VIn_BLOCK2_SIZE

VIn_BLOCK2_SIZE 为遮挡块 2 大小寄存器。



	Offset Address	Register Name	Total Reset Value
	0x0074+n×0x1000	VIn_BLOCK2_SIZE	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block2_stone_height	block2_stone_width
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:12]	RW	block2_height
	[11:0]	RW	block2_width
			Description
			保留。
			图像块 2 填充的高度（以行为单位）。
			图像块 2 填充的宽度（以像素为单位）。

VIn_BLOCK3_SIZE

VIn_BLOCK3_SIZE 为遮挡块 3 大小寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0078+n×0x1000	VIn_BLOCK3_SIZE	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block3_stone_height	block3_stone_width
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:12]	RW	block3_height
	[11:0]	RW	block3_width
			Description
			保留。
			图像块 3 填充的高度（以行为单位）。
			图像块 3 填充的宽度（以像素为单位）。

VIn_BLOCK0_COLOR

VIn_BLOCK0_COLOR 为遮挡块 0 填充颜色寄存器。

填充颜色时需要注意以下两点：

- 填充图像颜色必须为 YCbCr 格式数据。
- 当 VIU 配置为多个块遮挡，各个块之间有叠加部分时，输出叠加部分的颜色按照 0、1、2、3 的优先级覆盖。即填充块 0 的优先级最高，填充块 3 的优先级最低。

	Offset Address	Register Name	Total Reset Value
	0x007C+n×0x1000	VIn_BLOCK0_COLOR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block0_stone_y	block0_stone_u
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:16]	RW	block0_y
	[15:8]	RW	block0_u
	[7:0]	RW	block0_v
	Description		
	保留。		
	图像块 0 填充色的 Y 分量。		
	图像块 0 填充色的 Cb 分量。		
	图像块 0 填充色的 Cr 分量。		

VIn_BLOCK1_COLOR

VIn_BLOCK1_COLOR 为图像块 1 填充颜色寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0080+n×0x1000	VIn_BLOCK1_COLOR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block1_stone_y	block1_stone_u
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:16]	RW	block1_y
	[15:8]	RW	block1_u
	[7:0]	RW	block1_v
	Description		
	保留。		
	图像块 1 填充色的 Y 分量。		
	图像块 1 填充色的 Cb 分量。		
	图像块 1 填充色的 Cr 分量。		

VIn_BLOCK2_COLOR

VIn_BLOCK2_COLOR 为图像块 2 填充颜色寄存器。



	Offset Address	Register Name	Total Reset Value
	0x0084+n×0x1000	VIn_BLOCK2_COLOR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block2_stone_y	block2_stone_u
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:16]	RW	block2_y
	[15:8]	RW	block2_u
	[7:0]	RW	block2_v
			Description
			保留。
			图像块 2 填充色的 Y 分量。
			图像块 2 填充色的 Cb 分量。
			图像块 2 填充色的 Cr 分量。

VIn_BLOCK3_COLOR

VIn_BLOCK3_COLOR 为图像块 3 填充颜色寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0088+n×0x1000	VIn_BLOCK3_COLOR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	block3_stone_y	block3_stone_u
Reset	0 0		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:16]	RW	block3_y
	[15:8]	RW	block3_u
	[7:0]	RW	block3_v
			Description
			保留。
			图像块 3 填充色的 Y 分量。
			图像块 3 填充色的 Cb 分量。
			图像块 3 填充色的 Cr 分量。

VIn_ANC0_START

VIn_ANC0_START 为消隐区数据块 0 起始位置寄存器。



Offset Address		Register Name		Total Reset Value		
0x009C+n×0x1000		VI_ANC0_START		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved	Vbi_location1	Vbi-vos1	Vbi-hos1		
Reset	0 0					0
Bits	Access	Name	Description			
[31:26]	-	reserved	保留。			
[25:24]	RW	anc0_loc	消隐区数据所在的消隐区。 00: 奇场前消隐区; 01: 奇场后消隐区; 10: 偶场前消隐区; 11: 偶场后消隐区。 说明: 在逐行模式下, 只有奇场前消隐, DC 模式下无法接收消隐区数据。			
[23:12]	RW	anc0_vos	消隐区开始到消隐区数据所在行的距离 (以行为单位)。			
[11:0]	RW	anc0_hos	行有效数据开始到消隐区数据开始的水平偏移 (以时钟为单位)。			

VIIn_ANC0_SIZE

VIIn_ANC0_SIZE 为消隐区数据块 0 大小寄存器。

Offset Address		Register Name		Total Reset Value		
0x00A0+n×0x1000		VIIn_ANC0_SIZE		0x0000_0020		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved			vbi-size1		
Reset	0 0					0
Bits	Access	Name	Description			
[31:12]	-	reserved	保留。			
[11:0]	RW	anc0_size	消隐区数据块的大小 (以时钟为单位)。			



VIn_ANC1_START

VIn_ANC1_START 为消隐区数据块 1 起始位置寄存器。

	Offset Address	Register Name	Total Reset Value	
	0x00A4+n×0x1000	VIn_ANC1_START	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	Reserved	Vbi_location2	Vbi-vos2	
Reset	0 0			
	Bits	Access	Name	Description
	[31:26]	-	Reserved	保留。
	[25:24]	RW	anc1_loc	消隐区数据所在的消隐区。 00: 奇场前消隐区; 01: 奇场后消隐区; 10: 偶场前消隐区; 11: 偶场后消隐区。 说明: 在逐行模式下, 只有奇场前消隐, DC 模式下无法接收消隐区数据。
	[23:12]	RW	anc1_vos	消隐区开始到消隐区数据所在行的距离 (以行为单位)。
	[11:0]	RW	anc1_hos	行有效数据开始到消隐区数据开始的水平偏移 (以时钟为单位)。

VIn_ANC1_SIZE

VIn_ANC1_SIZE 为消隐区数据块 1 大小寄存器。

	Offset Address	Register Name	Total Reset Value	
	0x00A8+n×0x1000	VIn_ANC1_SIZE	0x0000_0020	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	reserved	vbi-size2		
Reset	0 1 0 0 0 0 0			
	Bits	Access	Name	Description
	[31:12]	-	reserved	保留。
	[11:0]	RW	anc1_size	消隐区数据块的大小 (以时钟为单位)。



VIn_ANC0_WORD

VIn_ANC0_WORD 为获取的 16 个 word 消隐区数据存放的寄存器组 1。寄存器组 1 包含 8 个寄存器，每个寄存器为 32bit，从 VIn_ANC0_WORD0 开始存放。每个寄存器对应的偏移地址如下：

- 0x00AC: VIn_ANC0_WORD0;
- 0x00B0: VIn_ANC0_WORD1;
- 0x00B4: VIn_ANC0_WORD2;
- 0x00B8: VIn_ANC0_WORD3;
- 0x00BC: VIn_ANC0_WORD4;
- 0x00C0: VIn_ANC0_WORD5;
- 0x00C4: VIn_ANC0_WORD6;
- 0x00C8: VIn_ANC0_WORD7。

下面以 VIn_ANC0_WORD0 为例，其他依次类推。

	Offset Address	Register Name	Total Reset Value
	0x00AC+n×0x1000	VIn_ANC0_WORD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	VIn_ANC0_WORD		
Reset	0 1 0 0 0 0 0		
Bits	Access	Name	Description
[31:0]	RO	VIn_ANC0_WORD	消隐区数据 1 组的第 1 个 word 数据存放寄存器。

VIn_ANC1_WORD

VIn_ANC1_WORD 为获取的 16 个 word 消隐区数据存放的寄存器组 2。寄存器组 2 包含 8 个寄存器，每个寄存器为 32bit，从 VIn_ANC1_WORD0 开始存放。每个寄存器对应的偏移地址如下：

- 0x00CC: VIn_ANC1_WORD0;
- 0x00D0: VIn_ANC1_WORD1;
- 0x00D4: VIn_ANC1_WORD2;
- 0x00D8: VIn_ANC1_WORD3;
- 0x00DC: VIn_ANC1_WORD4;
- 0x00E0: VIn_ANC1_WORD5;
- 0x00E4: VIn_ANC1_WORD6;
- 0x00E8: VIn_ANC1_WORD7。

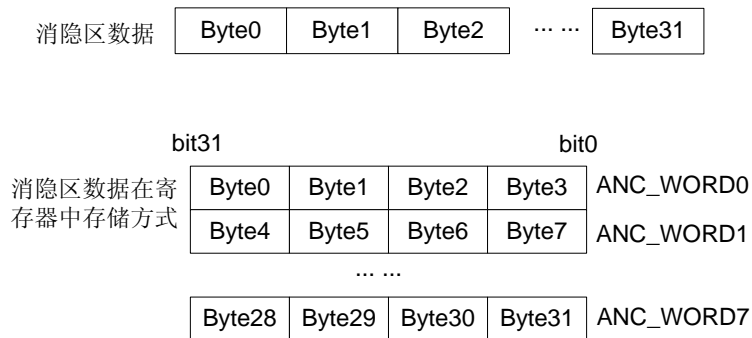


下面以 VIn_ANC1_WORD0 为例，其他依次类推。

	Offset Address	Register Name	Total Reset Value
	0x00CC+n×0x1000	VIn_ANC1_WORD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	VIn_ANC1_WORD		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	VIn_ANC1_WORD
	Description		
	消隐区数据 2 组的第 1 个 word 数据存放寄存器。		

消隐区数据在寄存器中存放的顺序如图 6-31 所示。

图6-31 消隐区数据在寄存器中的存放顺序示意图



VI_P0_VSYNC1

当端口 0 工作在 BT.601 模式下，VI_P0_VSYNC1 为第 1 场垂直同步配置寄存器，该寄存器仅第 0 路通道有效。当端口 0 工作在 16-bit 同步接口模式下，且水平同步为脉冲方式，VI_P0_VSYNC1 为垂直同步配置寄存器，该寄存器对通道 0 和 4 有效。

	Offset Address	Register Name	Total Reset Value
	0x00EC	VI_P0_VSYNC1	0x0001_511F
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	act1_voff	act1_height
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 1 1 1 1		
	Bits	Access	Name
	[31:24]	-	reserved
	[23:12]	RW	act1_voff
	Description		
	保留。		
	BT.601 模式下为第 1 场开始到活动图像垂直距离；16-bit 同步接口模式下，帧开始到活动图像垂直距离。以行为单位，		



			配置值为实际行数减 1，默认为 21。
[11:0]	RW	act1_height	BT.601 模式下为第 1 场活动图像的高度；16-bit 同步接口模式下为活动图像的高度。以行为单位，配置值为实际行数减 1，默认为 287。

注：在使用 16-bit 并行接口的时候，亮度通道和色度通道的时序寄存器，均需要配置（亮度和色度配置一致）。

VI_P2_VSYNC1

当端口 2 工作在 BT.601 模式下，VI_P2_VSYNC1 为第 1 场垂直同步配置寄存器，该寄存器仅第 8 路通道有效。当端口 2 工作在 16-bit 同步接口模式下，且水平同步为脉冲方式，VI_P2_VSYNC1 为垂直同步配置寄存器，该寄存器对通道 8 和 12 有效。

	Offset Address	Register Name	Total Reset Value
	0x80EC	VI_P0_VSYNC1	0x0001_511F
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	act1_voff	act1_height
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 1 1 1 1		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	act1_voff	BT.601 模式下为第 1 场开始到活动图像垂直距离；16-bit 同步接口模式下，帧开始到活动图像垂直距离。以行为单位，配置值为实际行数减 1，默认为 21。
[11:0]	RW	act1_height	BT.601 模式下为第 1 场活动图像的高度；16-bit 同步接口模式下为活动图像的高度。以行为单位，配置值为实际行数减 1，默认为 287。

注：在使用 16-bit 并行接口的时候，亮度通道和色度通道的时序寄存器，均需要配置（亮度和色度配置一致）。

VI_P0_VSYNC2

当端口 0 工作在 BT.601 模式下，VI_P0_VSYNC2 为第 2 场垂直同步配置寄存器，仅第 0 路通道有效。



Offset Address		Register Name		Total Reset Value	
0x00F0		VI_P0_VSYNC2		0x0001_611F	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Name	reserved	act2_voff	act2_height		
Reset	0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1				
Bits	Access	Name	Description		
[31:24]	-	reserved	保留。		
[23:12]	RW	act2_voff	第 2 场开始到活动图像垂直距离（以行为单位）。 配置值为实际行数减 1，默认为 22。		
[11:0]	RW	act2_height	第 2 场活动图像的高度（以行为单位）。 配置值为实际行数减 1，默认为 287。		

注：在使用 16-bit 并行接口的时候，亮度通道和色度通道的时序寄存器，均需要配置（亮度和色度配置一致）。

VI_P2_VSYNC2

当端口 2 工作在 BT.601 模式下，VI_P2_VSYNC2 为第 2 场垂直同步配置寄存器，仅第 8 路通道有效。

Offset Address		Register Name		Total Reset Value	
0x80F0		VI_P0_VSYNC2		0x0001_611F	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Name	reserved	act2_voff	act2_height		
Reset	0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1				
Bits	Access	Name	Description		
[31:24]	-	reserved	保留。		
[23:12]	RW	act2_voff	第 2 场开始到活动图像垂直距离（以行为单位）。 配置值为实际行数减 1，默认为 22。		
[11:0]	RW	act2_height	第 2 场活动图像的高度（以行为单位）。 配置值为实际行数减 1，默认为 287。		

注：在使用 16-bit 并行接口的时候，亮度通道和色度通道的时序寄存器，均需要配置（亮度和色度配置一致）。



VI_P0_HSYNC

VI_P0_HSYNC 为水平同步配置寄存器，当端口 0 工作在 BT.601 模式下，该寄存器仅第 0 路通道有效，当端口 0 工作在 16-bit 同步接口模式下，且水平同步为脉冲方式，该寄存器对通道 0 和 4 有效。

	Offset Address	Register Name	Total Reset Value
	0x00F4	VI_P0_HSYNC	0x0004_159F
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	act_hoff	act_width
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1		
Bits	Access	Name	Description
[31:24]	-	reserved	保留。
[23:12]	RW	act_hoff	上一行结束到本行活动数据区域的距离（以时钟为单位）。 BT.601 模式下，配置值为像素的 2 倍减 1，默认为 263。 16-bit 同步接口模式下，配置为像素个数减 1。
[11:0]	RW	act_width	活动图像宽度（以时钟为单位）。 BT.601 模式下，配置为像素的 2 倍减 1，默认为 1439（720 个像素）。 16-bit 同步接口模式下，配置为像素个数减 1。

注：在使用 16-bit 并行接口的时候，亮度通道和色度通道的时序寄存器，均需要配置（亮度和色度配置一致）。

VI_P2_HSYNC

VI_P2_HSYNC 为水平同步配置寄存器。当端口 2 工作在 BT.601 模式下，该寄存器仅第 8 路通道有效，当端口 0 工作在 16-bit 同步接口模式下，且水平同步为脉冲方式，该寄存器对通道 8 和 12 有效。



Offset Address		Register Name		Total Reset Value								
0x80F4		VI_P2_HSYNC		0x0004_159F								
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0											
Name	reserved			act_hoff				act_width				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 1											
Bits	Access	Name	Description									
[31:24]	-	reserved	保留。									
[23:12]	RW	act_hoff	上一行结束到本行活动数据区域的距离（以时钟为单位）。 BT.601 模式下，配置值为像素的 2 倍减 1，默认为 263。 16-bit 同步接口模式下，配置为象素个数减 1。									
[11:0]	RW	act_width	活动图像宽度（以时钟为单位）。 BT.601 模式下，配置为像素的 2 倍减 1，默认为 1439（720 个像素）。 16-bit 同步接口模式下，配置为象素个数减 1。									

注：在使用 16bit BT.601 输入的时候,亮度通道和色度通道的时序寄存器,均需要配置（亮度和色度配置一致）

VI_PRIO_CFG

VI_PRIO_CFG 为 VIU 内部 16 个通道的优先级别配置寄存器，每个通道都支持 2 级优先级可配。

Offset Address		Register Name		Total Reset Value																
0x00F8		VI_PRIO_CFG		0x0004_0000																
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																			
Name	reserved			outstanding_max	vi15_prio_ctrl	vi14_prio_ctrl	vi13_prio_ctrl	vi12_prio_ctrl	vi11_prio_ctrl	vi10_prio_ctrl	vi9_prio_ctrl	vi8_prio_ctrl	vi7_prio_ctrl	vi6_prio_ctrl	vi5_prio_ctrl	vi4_prio_ctrl	vi3_prio_ctrl	vi2_prio_ctrl	vi1_prio_ctrl	vi0_prio_ctrl
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																			
Bits	Access	Name	Description																	
[31:20]	-	reserved	保留。																	
[19:16]	RW	outstanding_max	AXI 总线 Outstanding 最大个数。																	
[15]	RW	vi15_prio_ctrl	通道 15 优先级控制。 0: 正常优先级; 1: 高优先级。																	



[14]	RW	vi14_prio_ctrl	通道 14 优先级控制。 0: 正常优先级; 1: 高优先级。
[13]	RW	vi13_prio_ctrl	通道 13 优先级控制。 0: 正常优先级; 1: 高优先级。
[12]	RW	vi12_prio_ctrl	通道 12 优先级控制。 0: 正常优先级; 1: 高优先级。
[11]	RW	vi11_prio_ctrl	通道 11 优先级控制。 0: 正常优先级; 1: 高优先级。
[10]	RW	vi10_prio_ctrl	通道 10 优先级控制。 0: 正常优先级; 1: 高优先级。
[9]	RW	vi9_prio_ctrl	通道 9 优先级控制。 0: 正常优先级; 1: 高优先级。
[8]	RW	vi8_prio_ctrl	通道 8 优先级控制。 0: 正常优先级; 1: 高优先级。
[7]	RW	vi7_prio_ctrl	通道 7 优先级控制。 0: 正常优先级; 1: 高优先级。
[6]	RW	vi6_prio_ctrl	通道 6 优先级控制。 0: 正常优先级; 1: 高优先级
[5]	RW	vi5_prio_ctrl	通道 5 优先级控制。 0: 正常优先级; 1: 高优先级。
[4]	RW	vi4_prio_ctrl	通道 4 优先级控制。 0: 正常优先级; 1: 高优先级。
[3]	RW	vi3_prio_ctrl	通道 3 优先级控制。



			0: 正常优先级; 1: 高优先级。
[2]	RW	vi2_prio_ctrl	通道 2 优先级控制。 0: 正常优先级; 1: 高优先级。
[1]	RW	vi1_prio_ctrl	通道 1 优先级控制。 0: 正常优先级; 1: 高优先级。
[0]	RW	vi0_prio_ctrl	通道 0 优先级控制。 0: 正常优先级; 1: 高优先级。

VIn_LUM_COEF0

VIn_LUM_COEF0 为亮度滤波系数寄存器 0。

滤波系数格式为：最高位为符号位，低 9 比特为系数绝对值。

lum_coef0~lum_coef7 之和为 512。

1/2 缩放时亮度滤波系数推荐值为 (lum_coef0~lum_coef7)：-16、0、145、254、145、0、-16、0。

	Offset Address	Register Name	Total Reset Value
	0x00FC+n×0x1000	VIn_LUM_COEF0	0x0000_0210
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved lum_coef1 reserved lum_coef0		
Reset	0 1 0 0 0 0 1 0 0 0 0		
Bits	Access	Name	Description
[31:26]	-	reserved	保留。
[25:16]	RW	lum_coef1	亮度滤波系数 1，默认值为 0。
[15:10]	-	reserved	保留。
[9:0]	RW	lum_coef0	亮度滤波系数 0，默认值为-16。

VIn_LUM_COEF1

VIn_LUM_COEF1 为亮度滤波系数寄存器 1。



滤波系数格式为补码形式，最高位为符号位。

	Offset Address	Register Name	Total Reset Value
	0x0100+n×0x1000	VIn_LUM_COEF1	0x00FE_0091
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	lum_coef3	reserved
Reset	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1		
	Bits	Access	Name
	[31:26]	-	reserved
	[25:16]	RW	lum_coef3
	[15:10]	-	reserved
	[9:0]	RW	lum_coef2
			Description
			保留。
			亮度滤波系数 3，默认值为 254。
			保留。
			亮度滤波系数 2，默认值为 145。

VIn_LUM_COEF2

VIn_LUM_COEF2 为亮度滤波系数寄存器 2。

滤波系数格式为补码形式，最高位为符号位。

	Offset Address	Register Name	Total Reset Value
	0x0104+n×0x1000	VIn_LUM_COEF2	0x0000_0091
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	lum_coef5	reserved
Reset	0 1 0 0 1 0 0 0 1		
	Bits	Access	Name
	[31:26]	-	reserved
	[25:16]	RW	lum_coef5
	[15:10]	-	reserved
	[9:0]	RW	lum_coef4
			Description
			保留。
			亮度滤波系数 5，默认值为 0。
			保留。
			亮度滤波系数 4，默认值为 145。

VIn_LUM_COEF3

VIn_LUM_COEF3 为亮度滤波系数寄存器 3。

滤波系数格式为补码形式，最高位为符号位。



	Offset Address 0x0108+n×0x1000	Register Name VIn_LUM_COEF3	Total Reset Value 0x0000_0210	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Name	reserved	lum_coef7	reserved lum_coef6	
Reset	0 0		1 0 0 0 0 0 1 0 0 0 0	
	Bits	Access	Name	Description
	[31:26]	-	reserved	保留。
	[25:16]	RW	lum_coef7	亮度滤波系数 7，默认值为 0。
	[15:10]	-	reserved	保留。
	[9:0]	RW	lum_coef6	亮度滤波系数 6，默认值为-16。

VIn_CHROMA_COEF0

VIn_CHROMA_COEF0 为色度滤波系数寄存器 0。

滤波系数格式为：最高位为符号位，低 9 比特为系数绝对值。

chroma_coef0~chroma_coef3 之和为 512。

1/2 缩放时色度滤波系数推荐值为 (chroma_coef0~chroma_coef3)：148、171、148、45。

色度重采样时色度滤波系数推荐值为 (chroma_coef0~chroma_coef3)：-32、416、160、-32。



	Offset Address	Register Name	Total Reset Value
	0x010C+n×0x1000	VIn_CHROMA_COEF0	0x00AB_0094
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	chroma_coef1	reserved chroma_coef0
Reset	0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0		
	Bits	Access	Name
	[31:26]	-	reserved
	[25:16]	RW	chroma_coef1
	[15:10]	-	reserved
	[9:0]	RW	chroma_coef0
	Description		
	保留。		
	色度滤波系数 1，默认值为 171。		
	保留。		
	色度滤波系数 0，默认值为 148。		

VIn_CHROMA_COEF1

VIn_CHROMA_COEF1 为色度滤波系数寄存器 1。

滤波系数格式为补码形式，最高位为符号位。

	Offset Address	Register Name	Total Reset Value
	0x0110+n×0x1000	VIn_CHROMA_COEF1	0x002D_0094
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	chroma_coef3	reserved chroma_coef2
Reset	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0		
	Bits	Access	Name
	[31:26]	-	reserved
	[25:16]	RW	chroma_coef3
	[15:10]	-	reserved
	[9:0]	RW	chroma_coef2
	Description		
	保留。		
	色度滤波系数 3，默认值为 45。		
	保留。		
	色度滤波系数 2，默认值为 148。		



6.2 VOU

6.2.1 概述

VOU (Video Output Unit) 模块主动从内存相应位置读取视频和图形数据，并通过 3 个显示通道 (HD 高清、AD 辅助和 SD 标清) 输出。VOU 可以同时处理 8 层图像数据，包括 3 个视频层、4 个图形层和 1 个硬件鼠标层。VOU 通过 3 个显示通道分别支持不同规格的输出接口，包括 BT.1120、BT.656、数字 RGB 等数字接口和 CVBS、S-Video、YPbPr、RGB 等模拟输出接口。

6.2.2 特点

VOU 的特点如下：

- 支持 2 种规格的输出接口

数字接口：

- 支持 10bit ITU-R BT.1120/YCbCr 4:2:2 标准输出接口
- 支持 8bit ITU-R BT.656/YCbCr 4:2:2 标准输出接口 (PAL/NTSC 制式 @27MHz)
- 支持 30bit 数字 RGB 输出接口

模拟接口：

- 支持 M-NTSC、NTSC-J 输出
- 支持 (B、D、G、H、I) PAL、(N) PAL、(Nc) PAL、(M) PAL 输出
- 支持多制式 S-video 输出
- 支持 YPbPr 输出
- 支持 VGA 输出

- 支持同时处理 8 层数据

高清视频层：

- 支持软件可配置的隔行、逐行和只读底场 3 种数据读取方式
- 支持输入像素格式：Semi-Planar YCbCr4:2:2、Semi-Planar YCbCr4:2:0
- 支持全局 alpha 值
- 支持水平垂直缩放 (缩小限制在 2 倍以内；放大限制在 4 倍以内)
- 支持水平亮度 8 阶滤波和水平色度 4 阶滤波 (YCbCr4:2:2 和 YCbCr4:2:0 数据格式)，2 种滤波方式各有 32 组滤波系数，且滤波系数可配置
- 支持垂直亮度 4 阶滤波和垂直色度 4 阶滤波 (YCbCr4:2:2 数据格式)，2 种滤波方式各有 32 组滤波系数，且滤波系数可配置
- 支持垂直亮度 4 阶滤波和垂直色度 2 阶滤波 (YCbCr4:2:0 数据格式)，2 种滤波方式各有 32 组滤波系数，且滤波系数可配置
- 支持 2 种模式的 de-interlace：2 场中值滤波和 4 场模式
- 支持输入最大分辨率为 1440×1152、最小分辨率为 32×32

辅助视频层：与高清视频层规格相同。

标清视频层：



- 支持软件可配置的隔行、逐行和只读底场 3 种数据读取方式
- 支持输入像素格式：Semi-Planar YCbCr4:2:2、Semi-Planar YCbCr4:2:0
- 支持全局 alpha 值
- 支持输入最大分辨率为 720×576、最小分辨率为 32×32

图形层 0:

- 支持的 16bit 像素格式有：ARGB1555
- 支持关键色处理
- 支持全局 alpha 和像素 alpha
- 支持 BT.601 和 BT.709 可选的颜色空间转换

图形层 1: 与图形层 0 规格相同

图形层 2: 与图形层 0 规格相同

图形层 3: 与图形层 0 规格相同

硬件鼠标层: 与图形层 0 规格相同

- 支持叠加特性

- 支持混合输入数据格式 AYCbCr4444
- 支持 3 个混合器：Mixer1 用于高清通道；Mixer2 用于辅助通道；Mixer3 用于标清通道
- Mixer1 支持的叠加层有 24 位背景色层、高清视频层、图形层 0、图形层 1、硬件鼠标层，其中背景色层优先级最低，鼠标层优先级最高，其他层优先级可配
- Mixer2 支持的叠加层有 24 位背景色层、辅助视频层、图形层 1、图形层 2、硬件鼠标层，其中背景色层优先级最低，鼠标层优先级最高，其他层优先级可配
- Mixer3 支持的叠加层有 24 位背景色层、标清视频层、图形层 3，其中背景色层优先级最低，其他层优先级可配。

- 支持 3 个显示通道

高清显示通道

- 支持 BT.1120 数字接口输出
- 支持数字 RGB 接口输出
- 支持 YPbPr 模拟接口（1080i 和 720p）
- 支持 VGA 模拟接口，最大支持 1280×1024@60Hz
- 支持 YCbCr to RGB 转换，转换系数可配置
- 支持对比度/亮度/饱和度可调节
- 支持动态 Gamma 调节
- 支持主模式，芯片自己产生接口输出时序
- 支持从模式，处于从模式时，可以根据主模式输入的时序进行主从图片的叠加，然后输出
- 支持多片级联

辅助显示通道

- 支持 VGA 输出，最大支持 800×600@60Hz
- 支持 CVBS 输出



- 支持 YCbCr to RGB 转换，转换系数可配置
- 支持对比度/亮度/饱和度可调节
- 动态 Gamma 调节

标清显示通道

- 支持 CVBS 模拟输出 (NTSC/PAL)
- 支持 YCbCr to RGB 转换，转换系数可配置
- 支持对比度/亮度/饱和度可调节

6.2.3 信号描述

VOU 数字输出接口信号如表 6-8 所示。

表6-8 VOU 数字输出接口信号

信号名	方向	描述	对应管脚
clk_vo0out	O	BT.656 随路时钟输出。支持正相输出和反相输出可配置。	VO0CK
clk_vo1out	O	BT.1120 随路时钟输出。支持正相输出和反相输出可配置。	VO1CK
vo_vga0_hsync	O	高清通道水平同步信号输出。	VO1DAT0
vo_vga0_vsync	O	高清通道垂直同步信号输出。	VO1DAT1
vo_vga0_dv	O	高清通道数据有效信号。	VORGBDV
vo_vga1_hsync	O	辅助通道水平同步信号输出。	VO1DAT13
vo_vga1_vsync	O	辅助通道垂直同步信号输出。	VO1DAT14
vo_656[0]	O	BT.656 数据输出。	VO0DAT0
vo_656[1]			VO0DAT1
vo_656[2]			VO0DAT2
vo_656[3]			VO0DAT3
vo_656[4]			VO0DAT4
vo_656[5]			VO0DAT5
vo_656[6]			VO0DAT6
vo_656[7]			VO0DAT7
vo_1120[0]	O	BT.1120 数据输出。	VO1DAT0
vo_1120[1]			VO1DAT1
vo_1120[2]			VO1DAT2
vo_1120[3]			VO1DAT3



信号名	方向	描述	对应管脚
vo_1120[4]			VO1DAT4
vo_1120[5]			VO1DAT5
vo_1120[6]			VO1DAT6
vo_1120[7]			VO1DAT7
vo_1120[8]			VO1DAT8
vo_1120[9]			VO1DAT9
vo_1120[10]			VO1DAT10
vo_1120[11]			VO1DAT11
vo_1120[12]			VO1DAT12
vo_1120[13]			VO1DAT13
vo_1120[14]			VO1DAT14
vo_1120[15]			VO1DAT15
vo_dig0_g_y[0]	O	数字 RGB 接口 G 分量输出。	VO0DAT7
vo_dig0_g_y[1]			VO1DAT2
vo_dig0_g_y[2]			VO1DAT3
vo_dig0_g_y[3]			VO1DAT4
vo_dig0_g_y[4]			VO1DAT5
vo_dig0_g_y[5]			VO1DAT6
vo_dig0_g_y[6]			VO1DAT7
vo_dig0_g_y[7]			VO1DAT8
vo_dig0_b_Pb[0]	O	数字 RGB 接口 B 分量输出。	VO1DAT9
vo_dig0_b_Pb[1]			VO1DAT10
vo_dig0_b_Pb[2]			VO1DAT11
vo_dig0_b_Pb[3]			VO1DAT12
vo_dig0_b_Pb[4]			VO1DAT13
vo_dig0_b_Pb[5]			VO1DAT14
vo_dig0_b_Pb[6]			VO1DAT15
vo_dig0_b_Pb[7]			GPIO3_0
vo_dig0_r_Pr[0]	O	数字 RGB 接口 R 分量输出。	VO0CK
vo_dig0_r_Pr[1]			VO0DAT0
vo_dig0_r_Pr[2]			VO0DAT1



信号名	方向	描述	对应管脚
vo_dig0_r_Pr[3]			VO0DAT2
vo_dig0_r_Pr[4]			VO0DAT3
vo_dig0_r_Pr[5]			VO0DAT4
vo_dig0_r_Pr[6]			VO0DAT5
vo_dig0_r_Pr[7]			VO0DAT6

VOU 模拟输出接口信号如表 6-9 所示。

表6-9 VOU 模拟输出接口信号

信号名	方向	描述	对应管脚
vo_dac0	O	分量模拟视频输出。	DACVGA1R
vo_dac1	O	分量模拟视频输出。	DACVGA1G
vo_dac2	O	分量模拟视频输出。	DACVGA1B
vo_dac3	O	分量模拟视频输出/复合视频模拟输出。	DACVGA0R
vo_dac4	O	分量模拟视频输出/复合视频模拟输出。	DACVGA0G
vo_dac5	O	分量模拟视频输出。	DACVGA0B

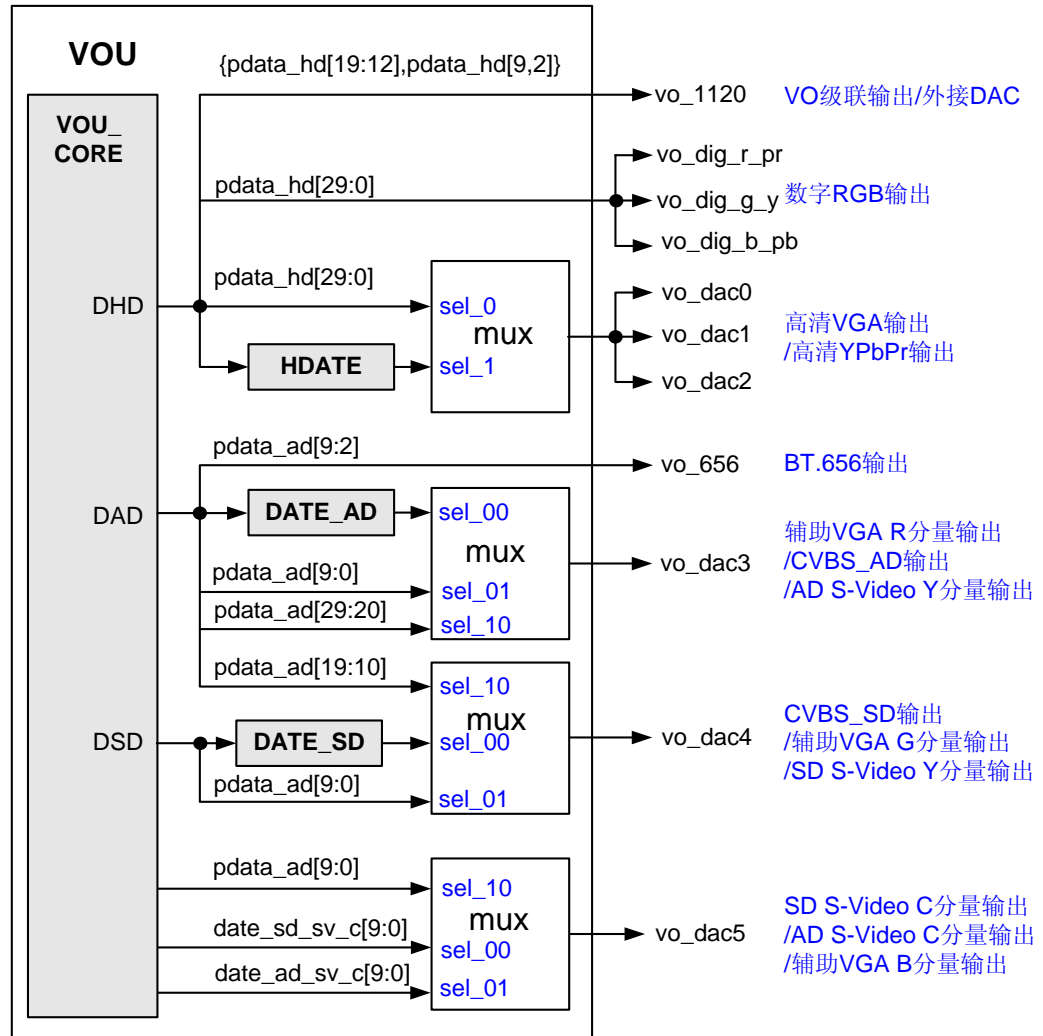
6.2.4 功能描述

VOU 输出接口说明

VOU 接口复用关系如图 6-32 所示。通过 VOMUX 寄存器的配置完成接口复用的选择控制。



图6-32 VOU 接口复用关系图



视频层功能

1. De-interlace

根据不同的图像质量要求，De-interlace 算法分为如下 2 种：

- 2 场中值滤波
- 4 场模式（运动估计、时域滤波、时空加权、边缘检测）

De-interlace 运算 4 场模式分全带宽和半带宽 2 种模式，在全带宽模式下才能进行时域滤波，半带宽模式下不能。

由于处理所需数据量不同，系统带宽消耗也不同，当输出一帧时，2 种算法需要读入的数据量分别是 2 场和 4 场。

2. 缩放



VOU 的缩放单元采用 8 阶、4 阶和 2 阶 32 相位精度的 FIR 滤波器，所有滤波器系数可配置：

- 水平方向亮度的缩放，采用 8 阶 32 相位 FIR 滤波器实现。
- 水平方向色度、垂直方向亮度的缩放，采用 4 阶 32 相位 FIR 滤波器。
- YCbCr4:2:2 格式下的垂直色度缩放采用 4 阶 32 相位 FIR 滤波器。
- YCbCr4:2:0 格式下的垂直色度缩放采用 2 阶 32 相位 FIR 滤波器。

图形层功能

1. Alpha 处理

VOU 有 5 个图形层，分别是图形层 0、图形层 1、图形层 2、图形层 3 和鼠标层。各层之间按照优先级叠加，叠加 alpha 值为 129 级，alpha 值的来源与层相关：

- 视频层的 alpha 值只能来源于寄存器配置的全局 alpha 值。
- 图形层的 alpha 值可以有 2 个来源：
 - 像素 alpha 值：某一个像素的叠加属性。
 - 全局 alpha 值。某一层的叠加属性。

当 n ($n \geq 2$) 层叠加时，2 种属性将被同时考虑， n 个 alpha 相乘得到 n 层的叠加 alpha 值。全局 alpha 根据配置得出；像素格式中没有 alpha 值时，像素 alpha 值默认为最大值 1。

像素 alpha 值有一种特殊情况，在 RGB1555 格式时，alpha 值仅有 1bit，该 bit 不是真实的 alpha 值，仅是 alpha 的索引，真实的 alpha 值是根据该索引值选择 alpha 寄存器中的值，其 alpha 值可以由 alpha0 和 alpha1 的值来选择。



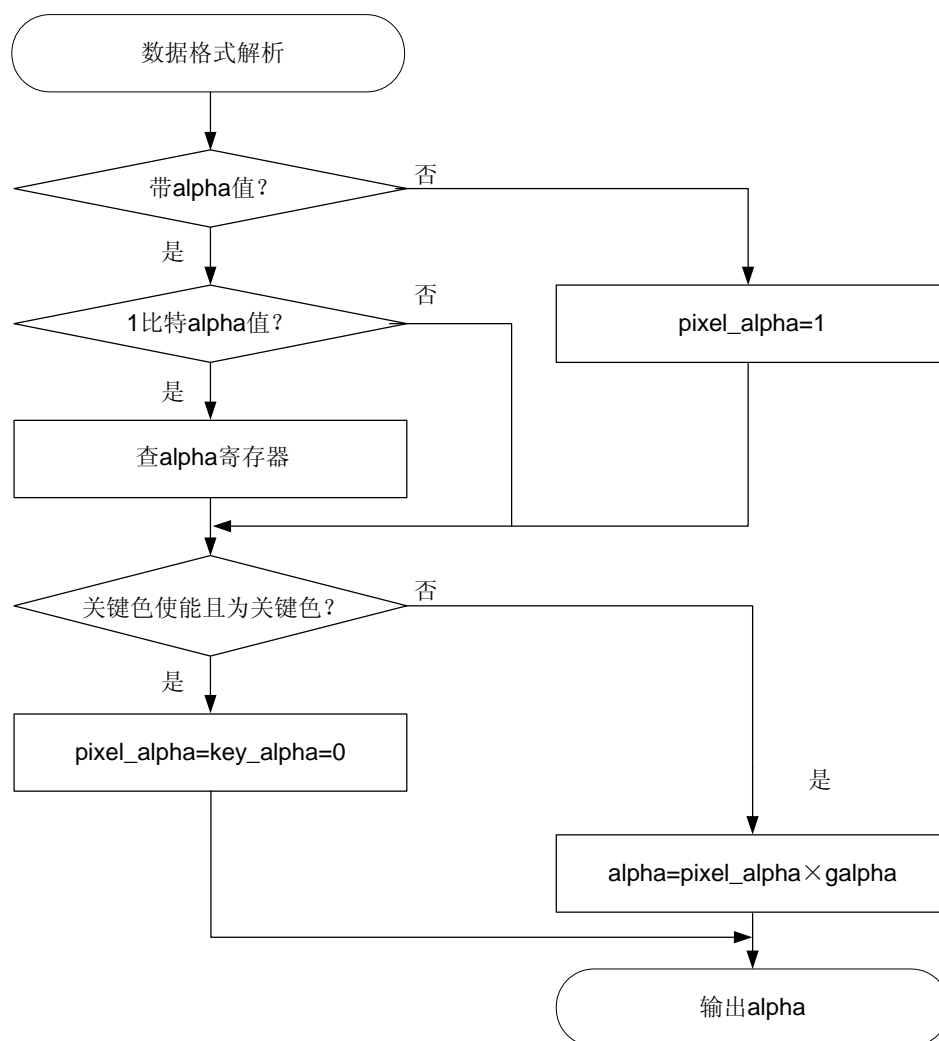
注意

当判断像素为关键色时，相当于该像素的叠加不使能。

图形层像素 alpha 处理流程图如图 6-33 所示：



图6-33 图形层 alpha 格式数据处理流程



pixel_alpha: 像素 alpha 值。
galpha: 全局 alpha 值。
key_alpha: 关键色 alpha 值为 0。

2. 关键色处理

在图形层中，与选定的颜色相同的颜色称为关键色。关键色由寄存器配置，使能时，硬件在相应的图层中进行匹配，当找到关键色时，将该颜色的 alpha 值替换成 0 值。关键色部分将做透明处理。

视频层的坐标说明

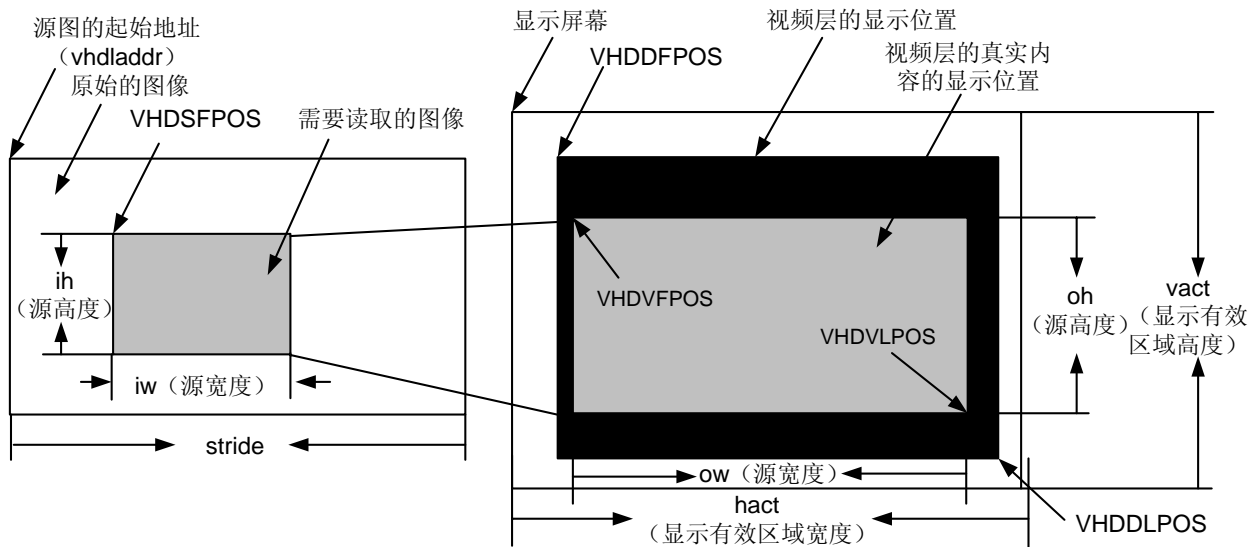
视频层有 3 套坐标，如图 6-34 所示，具体如下：

- 需要读取数据的源起始坐标
- 需要在屏幕上显示的起始结束坐标
- 视频内容在屏幕上显示的起始结束坐标

**注意**

- 源起始座标必须配成 (0, 0)。
- 显示内容的起始结束座标和视频内容的起始结束座标要配成一致。

图6-34 3套座标的示意图

**色彩空间转换**

VOU 支持色彩空间转换和颜色增强处理，支持从 YCbCr 到 RGB、RGB 到 YCbCr 的转换，具体如下：

- YCbCr 到 RGB 转换
进行叠加时的数据格式均为 YCbCr，如果是 VGA 显示时，需要从 YCbCr 转变到 RGB。
- RGB 到 YCbCr 转换
数据格式为 RGB 的图层，如果 TV 显示时，需要转换为 YCbCr 4:2:2 数据格式。

YCbCr 到 RGB 转换的转换矩阵公式如下：

$$\begin{bmatrix} R'255 \\ G'255 \\ B'255 \end{bmatrix} = \begin{bmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{bmatrix} \bullet \begin{bmatrix} Y'-in_dc0 \\ Cb-in_dc1 \\ Cr-in_dc2 \end{bmatrix}$$

RGB 到 YCbCr 转换的转换矩阵公式如下：



$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} out_dc0 \\ out_dc1 \\ out_dc2 \end{bmatrix} + \begin{bmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{bmatrix} \bullet \begin{bmatrix} R'255 \\ G'255 \\ B'255 \end{bmatrix}$$

中断

VOU 中有 17 个中断源，有一个中断状态寄存器和一个中断屏蔽寄存器，当中断源被屏蔽时，中断状态寄存器仍会记录中断发生的状态，但是不上报中断。

VOU 支持以下中断：

- 高清视频层、辅助视频层、标清视频层、图形层 0、图形层 1、图形层 2、图形层 3 和硬件鼠标层寄存器更新中断
- 高清视频层、辅助视频层写总线低带宽中断
- 总线错误中断
- 高清显示通道、辅助显示通道和标清显示通道低带宽告警中断
- 高清显示通道、辅助显示通道和标清显示通道垂直时序中断

6.2.5 工作方式

Surface 寄存器更新

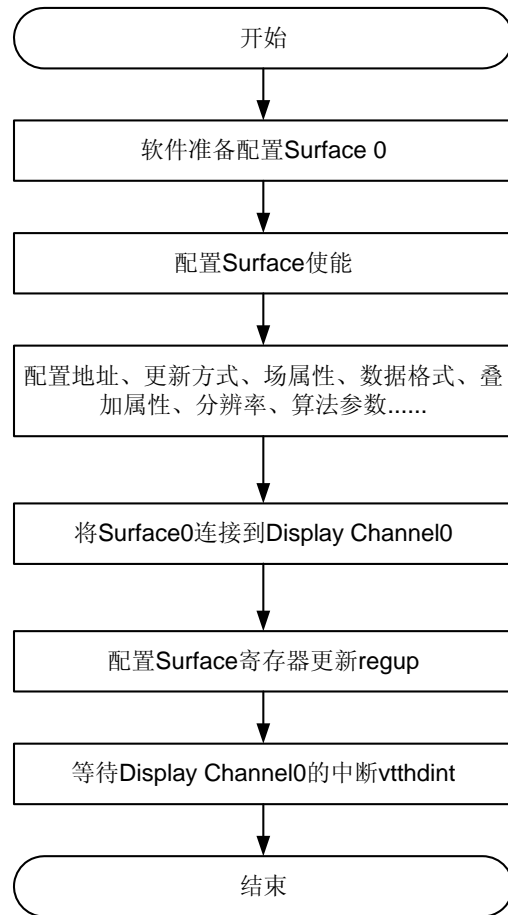
Surface 寄存器（非即时寄存器）更新如图 6-35 所示。



注意

- 每个 surface 必须连接到一个显示通道；
- 配置所有 surface 参数后必须配置寄存器更新 regup。

图6-35 Surface 寄存器配置流程（建议方案）



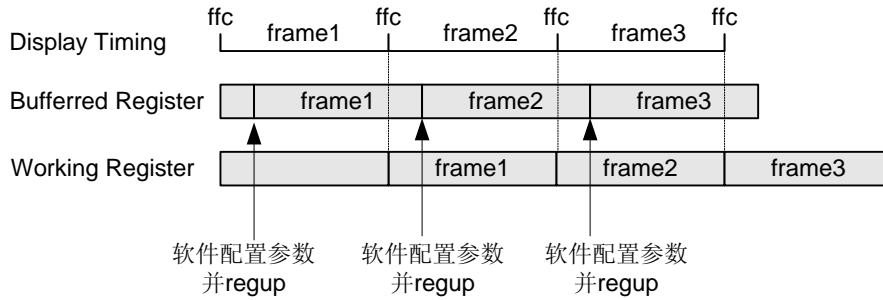
当 surface0 连接到 display channel0 后，surface0 就根据 display channel0 的时序中断来更新寄存器，因此软件需根据 display channel0 的中断 vtthdint 来判断是否到寄存器更新时间。

硬件中有 2 套 surface 寄存器：

- working register: 指当前数据通路中正在使用的寄存器配置
- buffered register

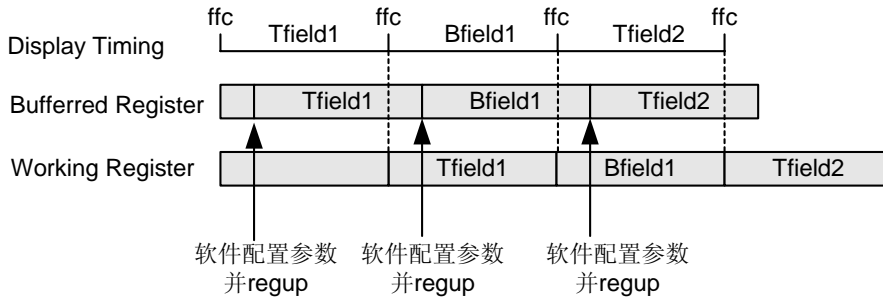
软件配置的寄存器首先缓存在 buffered register 中，到了寄存器更新时间才将 buffered register 中的值导入 working register。Surface 寄存器更新如图 6-36 所示。

图6-36 Surface 寄存器更新方式（帧更新模式）



对于视频层，除了支持按照帧更新，当隔行输出时还支持按照场更新，更新方式如图 6-37 所示。此时需要配置视频层的通道控制寄存器（regup_rate）为场更新，图 6-37 中的 ffc 是指帧/场切换点。

图6-37 Surface 寄存器更新方式（场更新模式）



Display channel 寄存器更新

Display channel 寄存器（即时寄存器）分为以下 2 类：

- 运算相关，包括 CSC、CLIP、背景色、Gamma 相关寄存器。这类寄存器在任何时候可以根据需要直接配置，但为了避免个别像素点的显示异常，最好在消隐区配置。
- 其他的都是时序控制相关，当需要更改，必须先关闭显示通道，配置完成后再打开。

叠加（CBM）寄存器配置方式

能够和 Mixer1 和 Mixer2 连接的 surface 共有 5 个，同一个 surface 不能同时驱动 2 个 Mixer。Surface 所驱动的 Mixer 可以配置，不能实时变更，当需要变更，必须先关闭相关的视频输出接口和相关 surface，再变更 surface 所驱动的 Mixer，具体配置过程如下：

步骤 1 关闭相关的视频输出接口。

步骤 2 配置 CBCFG bit[29:24]，确定各个 surface 与 Mixer 的对应关系。



步骤 3 配置各个 surface 相关寄存器。

步骤 4 配置各个 surface 优先级。

步骤 5 打开各个视频输出接口。

----结束

Mixer1 和 Mixer2 的 surface 优先级的配置：通过配置 CBCFG bit[14:0]可以实现优先级的配置，建议将驱动 Mixer1 的 surface 放在低优先级，驱动 Mixer2 的 surface 放在高优先级，一个 surface 不能驱动两个 Mixer。

例如，Mixer1 的驱动为 VDC_HD、GDC_G0 和 GDC_G2，优先级从低到高，Mixer2 的驱动为 VDC_AD 和 GDC_G1，优先级从低到高。则配置如下：

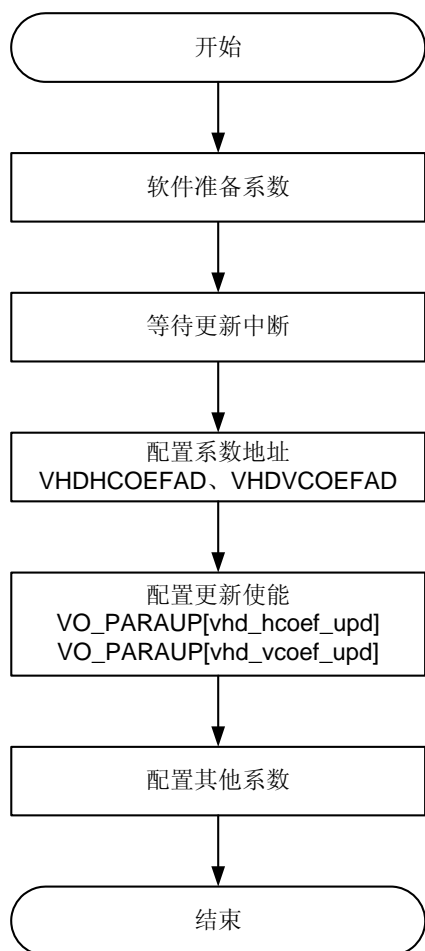
- VDC_HD,GDC_G0 和 GDC_G2 驱动 Mixer1。
`sur_attr0 = 0; sur_attr2 = 0; sur_attr4 = 0;`
- VDC_AD 和 GDC_G1 驱动 Mixer2。
`sur_attr1 = 1; sur_attr3 = 1;`
- 配置优先级。
`mixer_prio0 = 1; mixer_prio1 = 3; mixer_prio2 =5; mixer_prio3 = 2;mixer_prio4 =4;`

片内系数存储与更新

片内系数的数据量较大，为避免增加 CPU 负担，软件不直接通过 APB Slave 配置。软件需要配置系数更新使能和系数存储地址，由硬件自动更新（通过 AXI Master 读取）。软件更新后，软件将地址和更新命令放入系数更新队列，等待 AXI Master 进行更新。因此软件可以连续配置多个系数更新命令。更新片内系数更新的过程如图 6-38 所示。片内系数可以通过 APB Slave 读取。



图6-38 片内系数更新



时钟关断

当关闭 VOU 总线时钟时，必须先将所有接口禁止，再关闭总线时钟。这样可以避免时钟关断后无法清除中断状态的问题。

6.2.6 寄存器概览

VOU 寄存器概览如表 6-10 所示。

表6-10 VOU 寄存器概览（基址是 0x2013_0000）

偏移地址	名称	描述	页码
0x0000	VO_CTRL	VO 控制寄存器	6-98
0x0004	VO_INTSTA	VO 中断状态寄存器	6-98
0x0008	VO_INTMSK	VO 中断屏蔽寄存器	6-100
0x000C	VO_VERSION1	VO 版本寄存器 1	6-102



偏移地址	名称	描述	页码
0x0010	VO_VERSION2	VO 版本寄存器 2	6-103
0x001C	VO_MUX	VO 输出接口复选寄存器	6-103
0x002C	VO_PARAUP	缩放、Gamma 系数更新使能寄存器	6-104
0x0030	VHDHCOEFAD	VHD 水平亮度和水平色度滤波系数地址寄存器	6-106
0x0034	VHDVCOEFAD	VHD 垂直亮度和水平色度滤波系数地址寄存器	6-106
0x0038	VADHCOEFAD	VAD 水平亮度和水平色度滤波系数地址寄存器	6-106
0x003C	VADVCOEFAD	VAD 垂直亮度和水平色度滤波系数地址寄存器	6-107
0x0040	DHDACCAD	DHD 通道 ACC 系数查找表地址寄存器	6-107
0x0044	DADACCAD	DAD 通道 ACC 系数查找表地址寄存器	6-107
0x0100	VHDCTRL	VHD 通道控制寄存器	6-108
0x0104	VHDUPD	VHD 通道更新使能寄存器	6-109
0x0108	VHDLADDR	VHD 通道 De-interlace 上一帧地址寄存器	6-110
0x010C	VHDLCAADDR	VHD 通道 De-interlace 上一帧色度地址寄存器	6-110
0x0110	VHDCADDR	VHD 通道 De-interlace 当前帧地址寄存器	6-110
0x0114	VHDCCAADDR	VHD 通道 De-interlace 当前帧色度地址寄存器	6-111
0x0118	VHDNADDR	VHD 通道 De-interlace 下一帧地址寄存器	6-111
0x011C	VHDNCAADDR	VHD 通道 De-interlace 下一帧色度地址寄存器	6-112
0x0120	VHDSTRIDE	VHD 通道行偏移地址寄存器	6-112
0x0124	VHDCBMPARA	VHD 通道叠加相关参数寄存器，为非即时寄存器	6-112
0x0128	VHDORESO	VHD 通道输出分辨率寄存器，为非即时寄存器	6-113
0x012C	VHDIRESO	VHD 通道输入分辨率寄存器，为非即时寄存器	6-113



偏移地址	名称	描述	页码
0x0130	VHDSFPOS	VHD 通道源位图的起始位置寄存器，为非即时寄存器	6-114
0x0134	VHDDFPOS	VHD 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-115
0x0138	VHDDLPOS	VHD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-115
0x013C	VHDVFPOS	VHD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器	6-116
0x0140	VHDVLPOS	VHD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-116
0x0144	VHDBK	VHD 通道视频层的背景色寄存器	6-117
0x0150	VHDLMSP	VHD 通道亮度缩放参数配置寄存器	6-117
0x0154	VHDCHMSP	VHD 通道色度缩放参数配置寄存器	6-118
0x0158	VHDLMHSP	VHD 通道亮度水平缩放参数配置寄存器，为非即时寄存器	6-119
0x015C	VHDLMVSP	VHD 通道亮度垂直缩放参数配置寄存器，为非即时寄存器	6-119
0x0160	VHDCHMHSP	VHD 通道色度水平缩放参数配置寄存器	6-120
0x0164	VHDCHMVSP	VHD 通道色度垂直缩放参数配置寄存器	6-120
0x0170	VHDDIECTRL	VHD 通道 De-interlace 运算控制寄存器，为非即时寄存器	6-121
0x0174	VHDDIETHD	VHD 通道 De-interlace 运算水线寄存器，为非即时寄存器	6-123
0x0178	VHDDIEADDR	VHD 通道 De-interlace 历史信息 buffer 地址寄存器，为非即时寄存器	6-123
0x0180+n 1×0x4	VHDDIETSMIX	VHD 通道 De-interlace 时空加权系数寄存器，为即时寄存器	6-124
0x0190+n 2×0x4	VHDDIETFLT	VHD 通道 De-interlace 时域滤波系数寄存器，为即时寄存器	6-124
0x01A0+n 3×0x4	VHDDIEVFLT	VHD 通道 De-interlace 垂直滤波加权系数寄存器，为即时寄存器	6-125
0x01F0	VHDSTATUS	VHD 通道视频层状态寄存器	6-126



偏移地址	名称	描述	页码
0x0200	VADCTRL	VAD 控制寄存器，为非即时寄存器	6-126
0x0204	VADUPD	VAD 通道更新使能寄存器	6-128
0x0208	VADLADDR	VAD 通道 De-interlace 上一帧地址寄存器	6-128
0x020C	VADLCADDR	VAD 通道 De-interlace 上一帧色度地址寄存器	6-128
0x0210	VADCADDR	VAD 通道 De-interlace 当前帧地址寄存器	6-129
0x0214	VADCCADDR	VAD 通道 De-interlace 当前帧色度地址寄存器	6-129
0x0218	VADNADDR	VAD 通道 De-interlace 下一帧地址寄存器	6-130
0x021C	VADNCADDR	VAD 通道 De-interlace 下一帧色度地址寄存器	6-130
0x0220	VADSTRIDE	VAD 通道行偏移地址寄存器	6-130
0x0224	VADCBMPARA	VAD 通道叠加相关参数寄存器，为非即时寄存器	6-131
0x0228	VADORESO	VAD 通道输出分辨率寄存器，为非即时寄存器	6-131
0x022C	VADIRESO	VAD 通道输入分辨率寄存器，为非即时寄存器	6-132
0x0230	VADSFPOS	VAD 通道源位图的起始位置寄存器，为非即时寄存器	6-133
0x0234	VADDFPOS	VAD 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-133
0x0238	VADDLPOS	VAD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-134
0x023C	VADVFPPOS	VAD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器	6-134
0x0240	VADVLPOS	VAD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-135
0x0244	VADBK	VAD 通道视频层的背景色寄存器	6-135
0x0250	VADLMSP	VAD 通道亮度缩放参数配置寄存器	6-136



偏移地址	名称	描述	页码
0x0254	VADCHMSP	VAD 通道色度缩放参数配置寄存器	6-137
0x0258	VADLMHSP	VAD 通道亮度水平缩放参数配置寄存器，为非即时寄存器	6-138
0x025C	VADLMVSP	VAD 通道亮度垂直缩放参数配置寄存器，为非即时寄存器	6-138
0x0260	VADCHMHSP	VAD 通道色度水平缩放参数配置寄存器	6-139
0x0264	VADCHMVSP	VAD 通道色度垂直缩放参数配置寄存器	6-139
0x0270	VADDIECTRL	VAD 通道 De-interlace 运算控制寄存器，为非即时寄存器	6-140
0x0274	VADDIETHD	VAD 通道 De-interlace 运算水线寄存器，为非即时寄存器	6-142
0x0278	VADDIEADDR	VAD 通道 De-interlace 历史信息 buffer 地址寄存器，为非即时寄存器	6-142
0x0280+n 1×0x4	VADDIETSMIX	VAD 通道 De-interlace 时空加权系数寄存器，为即时寄存器	6-143
0x0290+n 2×0x4	VADDIETFLT	VAD 通道 De-interlace 时域滤波系数寄存器，为即时寄存器	6-143
0x02A0+n 3×0x4	VADDIEVFLT	VAD 通道 De-interlace 垂直滤波加权系数寄存器，为即时寄存器	6-144
0x02F0	VADSTATUS	VAD 通道视频层状态寄存器	6-145
0x0300	VSDCTRL	VSD 通道控制寄存器	6-145
0x0304	VSDUPD	VSD 通道更新使能寄存器	6-147
0x0310	VSDADDR	VSD 通道当前帧地址寄存器	6-147
0x0314	VSDCADDR	VSD 通道当前帧色度地址寄存器	6-147
0x0320	VSDSTRIDE	VSD 通道行偏移地址寄存器	6-148
0x0324	VSDCBMPARA	VSD 通道叠加相关参数寄存器，为非即时寄存器	6-148
0x0328	VSDORESO	VSD 通道输出分辨率寄存器，为非即时寄存器	6-149
0x032C	VSDIRESO	VSD 通道输入分辨率寄存器，为非即时寄存器	6-149
0x0330	VSDSFPOS	VSD 通道源位图的起始位置寄存器，为非即时寄存器	6-150



偏移地址	名称	描述	页码
0x0334	VSDDFPOS	VSD 通道显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器	6-150
0x0338	VDDLPOS	VSD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-151
0x033C	VSDVFPOS	VSD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器	6-151
0x0340	VSDLPOS	VSD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-152
0x0344	VSDBK	VSD 通道视频层的背景色寄存器	6-152
0x0400	G0CTRL	G0 通道控制寄存器，为非即时寄存器	6-153
0x0404	G0UPD	G0 通道更新使能寄存器	6-154
0x0408	G0ADDR	G0 通道帧地址寄存器	6-154
0x040C	G0STRIDE	G0 通道帧偏移地址寄存器	6-155
0x0410	G0CBMPARA	G0 通道叠加相关参数寄存器，为非即时寄存器	6-155
0x0414	G0CKEYMAX	G0 通道 color key 最大值寄存器，为非即时寄存器	6-156
0x0418	G0CKEYMIN	G0 通道 color key 最小值寄存器，为非即时寄存器	6-156
0x041C	G0IRESO	G0 通道输入分辨率寄存器，为非即时寄存器	6-157
0x0420	G0ORESO	G0 通道输出分辨率寄存器，为非即时寄存器	6-158
0x0424	G0SFPOS	G0 通道源位图的起始位置寄存器，为非即时寄存器	6-158
0x0428	G0DFPOS	G0 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-159
0x042C	G0DLPOS	G0 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-159
0x0500	G1CTRL	G1 通道控制寄存器，为非即时寄存器	6-160
0x0504	G1UPD	G1 通道更新使能寄存器	6-160
0x0508	G1ADDR	G1 通道帧地址寄存器	6-161



偏移地址	名称	描述	页码
0x050C	G1STRIDE	G1 通道帧偏移地址寄存器	6-161
0x0510	G1CBMPARA	G1 通道叠加相关参数寄存器, 为非即时寄存器	6-162
0x0514	G1CKEYMAX	G1 通道 color key 最大值寄存器, 为非即时寄存器	6-163
0x0518	G1CKEYMIN	G1 通道 color key 最小值寄存器, 为非即时寄存器	6-163
0x051C	G1IRESO	G1 通道输入分辨率寄存器, 为非即时寄存器	6-164
0x0520	G1ORESO	G1 通道输出分辨率寄存器, 为非即时寄存器	6-165
0x0524	G1SFPOS	G1 通道源位图的起始位置寄存器, 为非即时寄存器	6-165
0x0528	G1DFPOS	G1 通道显示窗口的起始位置寄存器, 以像素为单位, 非即时寄存器	6-166
0x052C	G1DLPOS	G1 通道显示窗口的结束位置寄存器, 以像素为单位, 为非即时寄存器	6-166
0x0600	G2CTRL	G2 通道控制寄存器, 为非即时寄存器	6-167
0x0604	G2UPD	G2 通道更新使能寄存器	6-167
0x0608	G2ADDR	G2 通道帧地址寄存器	6-168
0x060C	G2STRIDE	G2 通道帧偏移地址寄存器	6-168
0x0610	G2CBMPARA	G2 通道叠加相关参数寄存器, 为非即时寄存器	6-169
0x0614	G2CKEYMAX	G2 通道 color key 最大值寄存器, 为非即时寄存器	6-170
0x0618	G2CKEYMIN	G2 通道 color key 最小值寄存器, 为非即时寄存器	6-170
0x061C	G2IRESO	G2 通道输入分辨率寄存器, 为非即时寄存器	6-171
0x0620	G2ORESO	G2 通道输出分辨率寄存器, 为非即时寄存器	6-172
0x0624	G2SFPOS	G2 通道源位图的起始位置寄存器, 为非即时寄存器	6-172



偏移地址	名称	描述	页码
0x0628	G2DFPOS	G2 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-173
0x062C	G2DLPOS	G2 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-173
0x0700	G3CTRL	G3 通道控制寄存器，为非即时寄存器	6-174
0x0704	G3UPD	G3 通道更新使能寄存器	6-174
0x0708	G3ADDR	G3 通道帧地址寄存器	6-175
0x070C	G3STRIDE	G3 通道帧偏移地址寄存器	6-175
0x0710	G3CBMPARA	G3 通道叠加相关参数寄存器，为非即时寄存器	6-176
0x0714	G3CKEYMAX	G3 通道 color key 最大值寄存器，为非即时寄存器	6-177
0x0718	G3CKEYMIN	G3 通道 color key 最小值寄存器，为非即时寄存器	6-177
0x071C	G3IRESO	G3 通道输入分辨率寄存器，为非即时寄存器	6-178
0x0720	G3ORESO	G3 通道输出分辨率寄存器，为非即时寄存器	6-179
0x0724	G3SFPOS	G3 通道源位图的起始位置寄存器，为非即时寄存器	6-179
0x0728	G3DFPOS	G3 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-180
0x072C	G3DLPOS	G3 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-180
0x0800	HCCTRL	HC 通道控制寄存器，为非即时寄存器	6-181
0x0804	HCUPD	HC 通道更新使能寄存器	6-181
0x0808	HCADDR	HC 通道帧地址寄存器	6-182
0x080C	HCSTRIDE	HC 通道帧偏移地址寄存器	6-182
0x0810	HCCBMPARA	HC 通道叠加相关参数寄存器，为非即时寄存器	6-183
0x0814	HCCKEYMAX	HC 通道 color key 最大值寄存器，为非即时寄存器	6-184
0x0818	HCCKEYMIN	HC 通道 color key 最小值寄存器，为非即时寄存器	6-184



偏移地址	名称	描述	页码
0x081C	HCIRESO	HC 通道输入分辨率寄存器，为非即时寄存器	6-185
0x0820	HCORESO	HC 通道输出分辨率寄存器，为非即时寄存器	6-186
0x0824	HCSFPOS	HC 通道源位图的起始位置寄存器，为非即时寄存器	6-186
0x0828	HCFPOS	HC 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器	6-187
0x082C	HCDLPOS	HC 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器	6-187
0x0B00	CBMBKG1	Mixer1 叠加背景色寄存器，为即时寄存器	6-188
0x0B04	CBMBKG2	Mixer2 叠加背景色寄存器，为即时寄存器	6-188
0x0B0C	CBCFG	Colorbar 配置寄存器，为即时寄存器	6-188
0x0C00	DHDCTRL	HD 显示通道控制寄存器，为即时寄存器	6-191
0x0C04	DHDVSYNC	HD 显示通道垂直同步时序寄存器，为即时寄存器	6-193
0x0C08	DHDHSYNC1	HD 显示通道水平同步时序寄存器 1，为即时寄存器	6-193
0x0C0C	DHDHSYNC2	HD 显示通道水平同步时序寄存器 2，为即时寄存器	6-194
0x0C10	DHDVPLUS	HD 显示通道垂直同步补充时序寄存器，为即时寄存器	6-194
0x0C14	DHDVPLUS	HD 显示通道同步脉冲宽度寄存器，为即时寄存器	6-195
0x0C18	DHDFIFOTHD	HD 显示通道异步 FIFO 门限配置寄存器，为即时寄存器	6-195
0x0C1C	DHDVTTHD	HD 显示通道垂直时序门限寄存器，为即时寄存器	6-196
0x0C20	DHDCSCIDC	HD 显示通道色彩空间转换输入直流分量寄存器，为即时寄存器	6-196
0x0C24	DHDCSCODC	HD 显示通道色彩空间转换输出直流分量寄存器，为即时寄存器	6-197



偏移地址	名称	描述	页码
0x0C28	DHDCSCP0	HD 显示通道色彩空间转换参数 0 寄存器，为即时寄存器	6-197
0x0C2C	DHDCSCP1	HD 显示通道色彩空间转换参数 1 寄存器，为即时寄存器	6-198
0x0C30	DHDCSCP2	HD 显示通道色彩空间转换参数 2 寄存器，为即时寄存器	6-198
0x0C34	DHDCSCP3	HD 显示通道色彩空间转换参数 3 寄存器，为即时寄存器	6-199
0x0C38	DHDCSCP4	HD 显示通道色彩空间转换参数 4 寄存器，为即时寄存器	6-199
0x0C3C	DHDCLIPL	HD 显示通道嵌位处理最低门限值寄存器，为即时寄存器	6-200
0x0C40	DHDCLIPH	HD 显示通道嵌位处理最高门限值寄存器，为即时寄存器	6-200
0x0C44	DHDGMMTHD1	HD 显示通道 Gamma 运算门限 1 寄存器，为即时寄存器	6-201
0x0C48	DHDGMMTHD2	HD 显示通道 Gamma 运算门限 2 寄存器，为即时寄存器	6-201
0x0C50+t 1×0x4	DHDGMMLOWt	HD 显示通道 Gamma 查找表（亮度等级为 low）寄存器，为即时寄存器	6-201
0x0C60+t 2×0x4	DHDGMMMEDt	HD 显示通道 Gamma 查找表（亮度等级为 med）寄存器，为即时寄存器	6-202
0x0C70+t 3×0x4	DHDGMMHIGHT	HD 显示通道 Gamma 查找表（亮度等级为 high）寄存器，为即时寄存器	6-202
0x0C80+t 4×0x4	DHDGMMMLt	HD 显示通道 Gamma 查找表（亮度等级为 middle_low）寄存器，为即时寄存器	6-203
0x0CA0+t 5×0x4	DHDGMMMHt	HD 显示通道 Gamma 查找表（亮度等级为 middle_high）寄存器，为即时寄存器	6-203
0x0CB0	DHDGMM3LOW	HD 显示通道 Gamma 三区域的低门限亮度统计值寄存器	6-204
0x0CB4	DHDGMM3MED	HD 显示通道 Gamma 三区域的中间门限亮度统计值寄存器	6-204
0x0CB8	DHDGMM3HIGH	HD 显示通道 Gamma 三区域的高门限亮度统计值寄存器	6-205
0x0CC0	DHDGMM8MLOW	HD 显示通道 Gamma 中间区域分成 8 段后，低门限亮度统计值寄存器	6-205



偏移地址	名称	描述	页码
0x0CC4	DHDGMM8MHIGH	HD 显示通道 Gamma 中间区域分成 8 段后, 高门限亮度统计值寄存器	6-206
0x0CE0	CCDIMGMOD	级联多画面模式配置寄存器	6-206
0x0CF0	DHDSTATE	HD 显示通道状态寄存器	6-207
0x0D00	DADCTRL	AD 显示通道控制寄存器, 为即时寄存器	6-207
0x0D04	DADVSYNC	AD 显示通道垂直同步时序寄存器, 为即时寄存器	6-209
0x0D08	DADHSYNC1	AD 显示通道水平同步时序寄存器 1, 为即时寄存器	6-210
0x0D0C	DADHSYNC2	AD 显示通道水平同步时序寄存器 2, 为即时寄存器	6-210
0x0D10	DADVPLUS	AD 显示通道垂直同步补充时序寄存器, 为即时寄存器	6-211
0x0D14	DADPWR	AD 显示通道同步脉冲宽度寄存器, 为即时寄存器	6-211
0x0D18	DADFIFOTHD	AD 显示通道异步 FIFO 门限配置寄存器, 为即时寄存器	6-211
0x0D1C	DADVTTHD	AD 显示通道垂直时序门限寄存器, 为即时寄存器	6-212
0x0D20	DADCSCIDC	AD 显示通道色彩空间转换输入直流分量寄存器, 为即时寄存器	6-212
0x0D24	DADCSCODC	AD 显示通道色彩空间转换输出直流分量寄存器, 为即时寄存器	6-213
0x0D28	DADCSCP0	AD 显示通道色彩空间转换参数 0 寄存器, 为即时寄存器	6-213
0x0D2C	DADCSCP1	AD 显示通道色彩空间转换参数 1 寄存器, 为即时寄存器	6-214
0x0D30	DADCSCP2	AD 显示通道色彩空间转换参数 2 寄存器, 为即时寄存器	6-215
0x0D34	DADCSCP3	AD 显示通道色彩空间转换参数 3 寄存器, 为即时寄存器	6-215
0x0D38	DADCSCP4	AD 显示通道色彩空间转换参数 4 寄存器, 为即时寄存器	6-216



偏移地址	名称	描述	页码
0x0D3C	DADCLIPL	AD 显示通道嵌位处理最低门限值寄存器，为即时寄存器	6-216
0x0D40	DADCLIPH	AD 显示通道嵌位处理最高门限值寄存器，为即时寄存器	6-216
0x0D44	DADGMMTHD1	AD 显示通道 Gamma 运算门限 1 寄存器，为即时寄存器	6-217
0x0D48	DADGMMTHD2	AD 显示通道 Gamma 运算门限 2 寄存器，为即时寄存器	6-217
0x0D50+t 1×0x4	DADGMMLOWt	AD 显示通道 Gamma 查找表，亮度等级为 low 的寄存器，为即时寄存器	6-218
0x0D60+t 2×0x4	DADGMMMEDt	AD 显示通道 Gamma 查找表，亮度等级为 med 的寄存器，为即时寄存器	6-218
0x0D70+t 3×0x4	DADGMMHIGHT	AD 显示通道 Gamma 查找表，亮度等级为 high 的寄存器，为即时寄存器	6-219
0x0D80+t 4×0x4	DADGMMMLt	AD 显示通道 Gamma 查找表，亮度等级为 middle_low 的寄存器，为即时寄存器	6-219
0x0DA0+t 5×0x4	DADGMMMHt	AD 显示通道 Gamma 查找表，亮度等级为 middle_high 的寄存器，为即时寄存器	6-220
0x0DB0	DADGMM3LOW	AD 显示通道 Gamma 三区域的低门限亮度统计值寄存器	6-220
0x0DB4	DADGMM3MED	AD 显示通道 Gamma 三区域的中间门限亮度统计值寄存器	6-221
0x0DB8	DADGMM3HIGH	AD 显示通道 Gamma 三区域的高门限亮度统计值寄存器	6-221
0x0DC0	DADGMM8MLOW	AD 显示通道 Gamma 中间区域分成 8 段后，低门限亮度统计值寄存器	6-222
0x0DC4	DADGMM8MHIGH	AD 显示通道 Gamma 中间区域分成 8 段后，高门限亮度统计值寄存器	6-222
0x0DF0	DADSTATE	AD 显示通道状态寄存器	6-222
0x0E00	DSDCTRL	SD 显示通道控制寄存器，为即时寄存器	6-223
0x0E04	DSDVSYNC	SD 显示通道垂直同步时序寄存器，为即时寄存器	6-225
0x0E08	DSDHSYNC1	SD 显示通道水平同步时序寄存器 1，为即时寄存器	6-225



偏移地址	名称	描述	页码
0x0E0C	DSDHSYNC2	SD 显示通道水平同步时序寄存器 2, 为即时寄存器	6-226
0x0E10	DSDVPLUS	SD 显示通道垂直同步补充时序寄存器, 为即时寄存器	6-226
0x0E14	DSDPWR	SD 显示通道同步脉冲宽度寄存器, 为即时寄存器	6-227
0x0E18	DSDFIFOTHD	SD 显示通道异步 FIFO 门限配置寄存器, 为即时寄存器	6-227
0x0E1C	DSDVTTHD	SD 显示通道垂直时序门限寄存器, 为即时寄存器	6-228
0x0E20	DSDCSCIDC	SD 显示通道色彩空间转换输入直流分量寄存器, 为即时寄存器	6-228
0x0E24	DSDCSCODC	SD 显示通道色彩空间转换输出直流分量寄存器, 为即时寄存器	6-229
0x0E28	DSDCSCP0	SD 显示通道色彩空间转换参数 0 寄存器, 为即时寄存器	6-229
0x0E2C	DSDCSCP1	SD 显示通道色彩空间转换参数 1 寄存器, 为即时寄存器	6-230
0x0E30	DSDCSCP2	SD 显示通道色彩空间转换参数 2 寄存器, 为即时寄存器	6-230
0x0E34	DSDCSCP3	SD 显示通道色彩空间转换参数 3 寄存器, 为即时寄存器	6-231
0x0E38	DSDCSCP4	SD 显示通道色彩空间转换参数 4 寄存器, 为即时寄存器	6-231
0x0E3C	DSDCLIPL	SD 显示通道位处理最低门限值寄存器, 为即时寄存器	6-232
0x0E40	DSDCLIPH	SD 显示通道嵌位处理最高门限值寄存器, 为即时寄存器	6-232
0x0EF0	DSDSTATE	SD 显示通道状态寄存器	6-233
0x0F00+ m×0x4	CCDSPOSm	级联模式子图像 m 的起始位置和子画面有效标识寄存器	6-233
0x0F80+ m×0x4	CCDFPOSm	级联模式子图像 m 的结束位置寄存器	6-234
0x1000~ 0x111C	VHDHLCOEf	VHD 通道水平亮度缩放滤波系数寄存器	6-235



偏移地址	名称	描述	页码
0x1200~ 0x128C	VHDHCCOEF	VHD 通道水平色度缩放滤波系数寄存器	6-236
0x1300~ 0x138C	VHDVLCOEF	VHD 通道垂直亮度缩放滤波系数寄存器	6-236
0x1400~ 0x148C	VHDVCCOEF	VHD 通道垂直色度缩放滤波系数寄存器	6-237
0x1500~ 0x161C	VADHLCOEF	VAD 通道水平亮度缩放滤波系数寄存器	6-238
0x1700~ 0x178C	VADHCCOEF	VAD 通道水平色度缩放滤波系数寄存器	6-238
0x1800~ 0x188C	VADVLCOEF	VAD 通道垂直亮度缩放滤波系数寄存器	6-239
0x1900~ 0x198C	VADVCCOEF	VAD 通道垂直色度缩放滤波系数寄存器	6-240
0x2300~ 0x237C	VHDMIMGSPoS _p	VHD 通道 De-interlace 分区, 子图像 p 的起始位置和子画面有效标识寄存器	6-240
0x2380~ 0x23FC	VHDMIMGFPoS _p	VHD 通道 De-interlace 分区, 子图像 p 的结束位置寄存器	6-241
0x2400~ 0x247C	VADMIMGSPoS _p	VAD 通道 De-interlace 分区, 子图像 p 的起始位置和子画面有效标识寄存器	6-242
0x2480~ 0x24FC	VADMIMGFPoS _p	VAD 通道 De-interlace 分区, 子图像 p 的结束位置寄存器	6-242

VOU 寄存器偏移地址中变量的取值范围和含义如表 6-11 所示。

表6-11 VOU 寄存器偏移地址变量表

变量名称	取值范围	描述
n1	0~3	De-interlace 时空加权系数的组数
n2	0~3	De-interlace 时域滤波系数的组数
n3	0~3	De-interlace 运算参数的组数
t1	0~2	亮度等级为 low 的查找表数据的个数
t2	0~2	亮度等级为 med 的查找表数据的个数
t3	0~2	亮度等级为 high 的查找表数据的个数



变量名称	取值范围	描述
t4	0~2	亮度等级为 middle_low 的查找表数据的个数
t5	0~2	亮度等级为 middle_high 的查找表数据的个数
m	0~31	级联模式下子画面的个数

6.2.7 寄存器描述

VO_CTRL

VO_CTRL 为 VO 控制寄存器。

	Offset Address 0x0000	Register Name VO_CTRL	Total Reset Value 0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0	
Name	reserved			
Reset	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	
	Bits	Access	Name	Description
	[31:20]	-	reserved	保留。
	[19:16]	RW	outstd_wid0	AXI 总线写 ID0 的 outstanding。
	[15:12]	-	reserved	保留。
	[11:8]	RW	outstd_rid0	AXI 总线读 ID0 的 outstanding。
	[7:4]	-	reserved	保留。
	[3:0]	-	arb_mode	VO 内部各 Surface 总线数据申请的仲裁模式。 0x0: 轮询; 0x1: 图形层优先; 其他: 保留。

VO_INTSTA

VO_INTSTA 为 VO 中断状态寄存器。对中断状态位写 1 时，清除相应的中断。



Offset Address		Register Name		Total Reset Value																																			
0x0004		VO_INTSTA		0x0000_0000																																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name	be_int	reserved	vhd_st_wr_int	vad_st_wr_int	hcerr_int	g3rr_int	g2rr_int	g1rr_int	g0rr_int	vhdr_int	vaddr_int	vsdr_int	reserved										dhduf_int	dhdvtrhd_int	daduf_int	dadvtrhd_int	dsduf_int	dsdvtrhd_int											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
Bits	Access	Name	Description																																				
[31]	WC	be_int	总线（AXI_Master）错误中断状态。 0：无中断； 1：有中断。																																				
[30]	-	reserved	保留。																																				
[29]	WC	vhd_st_wr_int	VHD 写总线低带宽中断状态。 0：无中断； 1：有中断。																																				
[28]	WC	vad_st_wr_int	VAD 写总线低带宽中断状态。 0：无中断； 1：有中断。																																				
[27]	WC	hcerr_int	硬件鼠标寄存器更新完毕中断状态。 0：无中断； 1：有中断。																																				
[26]	WC	g3rr_int	G3（图形层 3）寄存器更新完毕中断状态。 0：无中断； 1：有中断。																																				
[25]	WC	g2rr_int	G2（图形层 2）寄存器更新完毕中断状态。 0：无中断； 1：有中断。																																				
[24]	WC	g1rr_int	G1（图形层 1）寄存器更新完毕中断状态。 0：无中断； 1：有中断。																																				
[23]	WC	g0rr_int	G0（图形层 0）寄存器更新完毕中断状态。 0：无中断； 1：有中断。																																				



[22]	WC	vhdrr_int	VDC_HD 寄存器更新完毕中断状态。 0: 无中断; 1: 有中断。
[21]	WC	vadrr_int	VDC_AD 寄存器更新完毕中断状态。 0: 无中断; 1: 有中断。
[20]	WC	vsdrr_int	VDC_SD 寄存器更新完毕中断状态。 0: 无中断; 1: 有中断。
[19:6]	-	reserved	保留。
[5]	WC	dhduf_int	HD 通道低带宽告警中断状态。 0: 无中断; 1: 有中断。
[4]	WC	dhdvtthd_int	HD 通道垂直时序中断状态。 0: 无中断; 1: 有中断。
[3]	WC	daduf_int	AD 通道低带宽告警中断状态。 0: 无中断; 1: 有中断。
[2]	WC	dadvttthd_int	AD 通道垂直时序中断状态。 0: 无中断; 1: 有中断。
[1]	WC	dsduf_int	SD 通道低带宽告警中断状态。 0: 无中断; 1: 有中断。
[0]	WC	dsdvtthd_int	SD 通道垂直时序中断状态。 0: 无中断; 1: 有中断。

VO_INTMSK

VO_INTMSK 为 VO 中断屏蔽寄存器。该寄存器与 VO_INTSTA 对应。



Offset Address		Register Name		Total Reset Value																																		
0x0008		VO_INTMSK		0x0000_0000																																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Name	be_intmsk	reserved	vhd_st_wr_intmask	vad_st_wr_intmask	hcrr_intmsk	g3rr_intmsk	g2rr_intmsk	g1rr_intmsk	g0rr_intmsk	vhdr_intmsk	vaddr_intmsk	vsdr_intmsk	reserved										dhduf_intmsk	dhdvthd_intmsk	daduf_intmsk	dadvthd_intmsk	dsduf_intmsk	dsdvthd_intmsk										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Bits	Access		Name		Description																																	
[31]	RW		be_intmsk		总线 (AXI_Master) 错误中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[30]	-		reserved		保留。																																	
[29]	RW		vhd_st_wr_intmask		VHD 写总线低带宽中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[28]	RW		vad_st_wr_intmask		VAD 写总线低带宽中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[27]	RW		hcrr_intmsk		硬件鼠标寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[26]	RW		g3rr_intmsk		G3 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[25]	RW		g2rr_intmsk		G2 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	
[24]	RW		g1rr_intmsk		G1 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																																	



[23]	RW	g0rr_intmsk	G0 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[22]	RW	vhdrr_intmsk	VDC_HD 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[21]	RW	vadrr_intmsk	VDC_AD 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[20]	RW	vsdrr_intmsk	VDC_SD 寄存器更新完毕中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[19:6]	-	reserved	保留。
[5]	RW	dhduf_intmsk	HD 通道低带宽告警中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[4]	RW	dhdvttthd_intmsk	HD 通道时序门限中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[3]	RW	daduf_intmsk	AD 通道低带宽告警中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[2]	RW	dadvttthdg_intmsk	AD 通道时序门限中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[1]	RW	dsduf_intmsk	SD 通道低带宽告警中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[0]	RW	dsdvtthd_intmsk	SD 通道时序门限中断屏蔽。 0: 屏蔽; 1: 不屏蔽。

VO_VERSION1

VO_VERSION1 为 VO 版本寄存器 1。



Offset Address		Register Name		Total Reset Value				
0x000C		VO_VERSION1		0x7675_6F76				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	voversion0							
Reset	0 1 1 1	0 1 1 0	0 1 1 1	0 1 0 1	0 1 1 0	1 1 1 1	0 1 1 1	0 1 1 0
Bits	Access	Name	Description					
[31:0]	RO	voversion0	VO 版本寄存器。					

VO_VERSION2

VO_VERSION2 为 VO 版本寄存器 2。

Offset Address		Register Name		Total Reset Value				
0x0010		VO_VERSION2		0x3030_3034				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	voversion1							
Reset	0 0 1 1	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1	0 1 0 0
Bits	Access	Name	Description					
[31:0]	RO	voversion1	VO 版本寄存器。					

VO_MUX

VO_MUX 为 VO 输出接口复选寄存器。

Offset Address		Register Name		Total Reset Value								
0x001C		VO_MUX		0x0000_0000								
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0				
Name	reserved				dac5_mux	dac4_mux	dac3_mux	reserved bit120_mu	reserved	dac2_mux	dac1_mux	dac0_mux
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description									
[31:14]	-	reserved	保留。									



[13:12]	RW	dac5_mux	VODAC5 输出选择。 00: SD 通路 S-Video 输出 (C 分量) ; 01: AD 通路 S-Video 输出 (C 分量) ; 10: AD 通路 VGA 输出 (B 分量) ; 11: 保留。
[11:10]	RW	dac4_mux	VODAC4 输出选择。 00: SD 通路 CVBS 输出, 或者 SD 通路 S-Video 输出 (Y 分量) ; 01: SD 通路 BT.656 输出; 10: AD 通路 VGA 输出 (G 分量) ; 11: 保留。
[9:8]	RW	dac3_mux	VODAC3 输出选择。 00: AD 通路 CVBS 输出, 或者 AD 通路 S-Video 输出 (Y 分量) ; 01: AD 通路 BT.656 输出; 10: AD 通路 VGA 输出 (R 分量) ; 11: 保留。
[7]	-	reserved	保留。
[6]	RW	bt1120_mux	BT.1120 输出端口选择。 0: BT.1120 端口输出 INTF (主时序模块) 的数据; 1: BT.1120 端口输出 IPI (从时序模块) 的数据。
[5:3]	-	reserved	保留。
[2]	RW	dac2_mux	VODAC2 输出选择。 0: HD 通路 VGA 输出 (B 分量) ; 1: HD 通路 YPbPr 输出 (Pb 分量) 。
[1]	RW	dac1_mux	VODAC1 输出选择。 0: HD 通路 VGA 输出 (G 分量) ; 1: HD 通路 YPbPr 输出 (Y 分量) 。
[0]	RW	dac0_mux	VODAC0 输出选择。 0: HD 通路 VGA 输出 (R 分量) ; 1: HD 通路 YPbPr 输出 (Pr 分量) 。

VO_PARAUP

VO_PARAUP 为缩放、Gamma 系数更新使能寄存器。VO 的缩放系数是通过 AXI Master 配置, 软件通过 APB Slave 配置起始地址和参数需要更新的标志。



Offset Address		Register Name		Total Reset Value																												
0x002C		VO_PARAUP		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								dad_acc_upd	dhd_acc_upd	vad_vcoef_upd	vad_hcoef_upd	vhd_vcoef_upd	vhd_hcoef_upd		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:6]	-	reserved	保留。																													
[5]	RW	dad_acc_upd	DAD ACC 运算查找表是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													
[4]	RW	dhd_acc_upd	DHD ACC 运算查找表是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													
[3]	RW	vad_vcoef_upd	VAD 垂直亮度、垂直色度滤波系数是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													
[2]	RW	vad_hcoef_upd	VAD 水平亮度、水平色度滤波系数是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													
[1]	RW	vhd_vcoef_upd	VHD 垂直亮度、垂直色度滤波系数是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													
[0]	RW	vhd_hcoef_upd	VHD 水平亮度、水平色度滤波系数是否需要更新。硬件更新后自动清零。 0: 不更新; 1: 更新。																													



VHDHCOEFAD

VHDHCOEFAD 为 VHD 水平亮度和水平色度滤波系数地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0030		VHDHCOEFAD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	coef_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	coef_addr	系数放在 local memory 的起始地址。					

VHDVCOEFAD

VHDVCOEFAD 为 VHD 垂直亮度和水平色度滤波系数地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0034		VHDVCOEFAD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	coef_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	coef_addr	系数放在 local memory 的起始地址。					

VADHCOEFAD

VADHCOEFAD 为 VAD 水平亮度和水平色度滤波系数地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0038		VADHCOEFAD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	coef_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	coef_addr	系数放在 local memory 的起始地址。					



VADVCOEFAD

VADVCOEFAD 为 VAD 垂直亮度和水平色度滤波系数地址寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x003C				VADVCOEFAD				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	coef_addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RW	coef_addr		系数放在 local memory 的起始地址。																											

DHDACCAD

DHDACCAD 为 DHD 通道 ACC 系数查找表地址寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0040				DHDACCAD				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	coef_addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RW	coef_addr		系数放在 local memory 的起始地址。																											

DADACCAD

DADACCAD 为 DAD 通道 ACC 系数查找表地址寄存器。



Offset Address		Register Name		Total Reset Value				
0x0044		DADACCAD		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	coef_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	coef_addr	系数放在 local memory 的起始地址。					

VHDCTRL

VHDCTRL 为 VHD 通道控制寄存器。

Offset Address		Register Name		Total Reset Value						
0x0100		VHDCTRL		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	vhd_en	reserved			regup_rate	bfield_first	lm_rmode	chm_rmode	reserved	ifmt
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31]	RW	vhd_en	Surface 使能。 0: 禁止; 1: 使能。							
[30:18]	-	reserved	保留。							
[17]	RW	regup_rate	隔行输出时, Surface 的寄存器更新频率。 0: 一帧更新一次; 1: 一场更新一次。							
[16]	RW	bfield_first	底场优先 (T: Top Field; B: Bottom Field)。 0: 顶场优先, T0B0T1B1 (T0B0 为一帧); 1: 底场优先, B0T0B1T1 (B0T0 为一帧)。							



[15:14]	RW	lm_rmode	亮度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。
[13:12]	RW	chm_rmode	色度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。
[11:4]	-	reserved	保留。
[3:0]	RW	ifmt	输入数据格式。 0x3: SPYCbCr4:2:0; 0x4: SPYCbCr4:2:2(1x2 格式); 其他: 保留。

VHDUPD

VHDUPD 为 VHD 通道更新使能寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x0104				VHDUPD								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											regup				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:1]	-		reserved		保留。																											
[0]	RW		regup		Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新, 更新完成后, 硬件自动清零。																											



VHDLADDR

VHDLADDR 为 VHD 通道 De-interlace 上一帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0108		VHDLADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdldaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdldaddr	De-interlace 上一帧的地址。					

VHDLCAADDR

VHDLCAADDR 为 VHD 通道 De-interlace 上一帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x010C		VHDLCAADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdldcaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdldcaddr	De-interlace 上一帧的色度地址。					

VHDCADDR

VHDCADDR 为 VHD 通道 De-interlace 当前帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。



Offset Address		Register Name		Total Reset Value				
0x0110		VHDCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdccaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdccaddr	De-interlace 当前帧的地址。					

VHDCADDR

VHDCADDR 为 VHD 通道 De-interlace 当前帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0114		VHDCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdccaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdccaddr	De-interlace 当前帧的色度地址。					

VHDNADDR

VHDNADDR 为 VHD 通道 De-interlace 下一帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0118		VHDNADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdnaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdnaddr	De-interlace 下一帧的地址。					



VHDNCADDR

VHDNCADDR 为 VHD 通道 De-interlace 下一帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x011C		VHDNCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdncaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vhdncaddr	De-interlace 下一帧的色度地址。					

VHDSTRIDE

VHDSTRIDE 为 VHD 通道行偏移地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0120		VHDSTRIDE		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vhdstride				vhdstride			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	RW	vhdstride	色度帧 buffer 的 stride（针对 semi-planar 格式），以 word 为单位。					
[15:0]	RW	vhdstride	亮度帧 buffer 的 stride（针对 semi-planar 格式），以 word 为单位。					

VHDCBMPARA

VHDCBMPARA 为 VHD 通道叠加相关参数寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value						
0x0124		VHDCBMPARA		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved						galpha			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:8]	-	reserved	保留。							
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。							

VHDORESO

VHDORESO 为 VHD 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0128		VHDORESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	oh			reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	oh	输出的视频图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，视频层的实际高度必须是偶数。逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	ow	输出的视频图像输出宽度，以像素为单位。 配置值为实际宽度减 1。 注意：视频层的实际宽度必须是偶数。						

VHDIRESO

VHDIRESO 为 VHD 通道输入分辨率寄存器，为非即时寄存器。

**注意**

- 隔行输出时，实际高度只能配置为偶数，当输入数据格式为 YCbCr4:2:0 时，高度必须是 4 的整数倍。
- 逐行输出时，当输入数据格式为 YCbCr4:2:0 时，高度必须是 2 的整数倍。视频层的实际宽度必须是偶数。

Offset Address		Register Name		Total Reset Value				
0x012C		VHDIRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	输入的视频图像高度，以行为单位。 配置值为实际高度减 1。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	输入的视频图像宽度，以像素为单位。 配置值为实际宽度减 1。					

VHDSFPOS

VHDSFPOS 为 VHD 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0130		VHDSFPOS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	src_yfpos		reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。					
[15:12]	-	reserved	保留。					



[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。
--------	----	-----------	---------------------------

VHDDFPOS

VHDDFPOS 为 VHD 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0134				VHDDFPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				disp_yfpos				reserved				disp_xfpos																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		disp_yfpos		显示列起始坐标。																											
[15:12]	RW		reserved		保留。																											
[11:0]	RW		disp_xfpos		显示行起始坐标。																											

VHDDLPOS

VHDDLPOS 为 VHD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0138				VHDDLPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				disp_ylpos				reserved				disp_xlpos																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		disp_ylpos		显示列结束坐标。																											
[15:12]	-		reserved		保留。																											
[11:0]	RW		disp_xlpos		显示行结束坐标。																											



VHDFVPOS

VHDFVPOS 为 VHD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x013C				VHDFVPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				video_yfpos				reserved				video_xfpos																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	video_yfpos		视频内容列起始坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	video_xfpos		视频内容行起始坐标。																											

VHDLVPOS

VHDLVPOS 为 VHD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0140				VHDLVPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				video_ylpos				reserved				video_xlpos																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	video_ylpos		视频内容列结束坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	video_xlpos		视频内容行结束坐标。																											



VHDBK

VHDBK 为 VHD 通道视频层的背景色寄存器。

Offset Address		Register Name		Total Reset Value												
0x0144		VHDBK		0x0000_0000												
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0								
Name	vbk_alpha				vbk_y				vbk_cb				vbk_cr			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0			
Bits	Access	Name	Description													
[31:24]	RW	vbk_alpha	视频层的背景填充色。 取值范围为 0~128。													
[23:16]	RW	vbk_y	Y 分量。													
[15:8]	RW	vbk_cb	Cb 分量。													
[7:0]	RW	vbk_cr	Cr 分量。													

VHDLMSF

VHDLMSF 为 VHD 通道亮度缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value												
0x0150		VHDLMSF		0x0000_0000												
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0								
Name	hlmsc_en	vlmsc_en	hlmid_en	vlmid_en	reserved				shift_field	fld_offset						
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0			
Bits	Access	Name	Description													
[31]	RW	hlmsc_en	水平亮度缩放使能。 0: 禁止; 1: 使能。													
[30]	RW	vlmsc_en	垂直亮度缩放使能。 0: 禁止; 1: 使能。													



[29]	RW	hlmid_en	水平亮度缩放中值滤波使能。 0: 禁止; 1: 使能。
[28]	RW	vlmid_en	垂直亮度缩放中值滤波使能。 0: 禁止; 1: 使能。
[27:17]	-	reserved	保留。
[16]	RW	shift_field	位置偏移所影响的场。 0: 底场显示时偏移; 1: 顶场显示时偏移。
[15:0]	RW	fld_offset	场位置偏移。 补码表示, 最高位为符号位, 4.12 格式。 注意: 该值的配置与缩放比例有关。

VHDCHMSP

VHDCHMSP 为 VHD 通道色度缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value																																
0x0154		VHDCHMSP		0x0000_0000																																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	hchmsc_en	vchmsc_en	hlmid_en	vlmid_en	reserved												shift_field	fld_offset																		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Bits	Access	Name	Description																																	
[31]	RW	hchmsc_en	水平色度缩放使能。 0: 禁止; 1: 使能。																																	
[30]	RW	vchmsc_en	垂直色度缩放使能。 0: 禁止; 1: 使能。																																	



[29]	RW	hlmid_en	水平色度缩放中值滤波使能。 0: 禁止; 1: 使能。
[28]	RW	vlmid_en	垂直色度缩放中值滤波使能。 0: 禁止; 1: 使能。
[27:17]	-	reserved	保留。
[16]	RW	shift_field	位置偏移所影响的场。 0: 底场显示时偏移; 1: 顶场显示时偏移。
[15:0]	RW	fld_offset	场位置偏移。 补码表示, 最高位为符号位, 4.12 格式。 注意: 该值的配置与缩放比例有关。

VHDLMHSP

VHDLMHSP 为 VHD 通道亮度水平缩放参数配置寄存器, 为非即时寄存器。

亮度水平缩放比率 = 输入宽度/输出宽度。

	Offset Address				Register Name								Total Reset Value																			
	0x0158				VHDLMHSP								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				lm_hphase								lm_hratio																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		lm_hphase		亮度水平缩放初始相位, 0.12 格式。																											
[15:0]	RW		lm_hratio		亮度水平缩放倍数, 4.12 格式。																											

VHDLMVSP

VHDLMVSP 为 VHD 通道亮度垂直缩放参数配置寄存器, 为非即时寄存器。

亮度垂直缩放比率 = 输入高度/输出高度。



Offset Address		Register Name		Total Reset Value					
0x015C		VHDLMVSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	lm_vphase			lm_vratio				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	lm_vphase	亮度垂直缩放初始相位，0.12 格式。						
[15:0]	RW	lm_vratio	亮度垂直缩放倍数，4.12 格式。						

VHDCHMHSP

VHDCHMHSP 为 VHD 通道色度水平缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value					
0x0160		VHDCHMHSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	chm_hphase			chm_hratio				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	chm_hphase	色度水平缩放初始相位，0.12 格式。						
[15:0]	RW	chm_hratio	色度水平缩放倍数，4.12 格式。						

VHDCHMVSP

VHDCHMVSP 为 VHD 通道色度垂直缩放参数配置寄存器。



Offset Address		Register Name		Total Reset Value																												
0x0164		VHDCHMVSP		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				chm_hphase								chm_vratio																			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0							
Bits	Access	Name		Description																												
[31:29]	-	reserved		保留。																												
[28:16]	RW	chm_hphase		色度垂直缩放初始相位，最高位为符号位，1.12 格式。																												
[15:0]	RW	chm_vratio		色度垂直缩放倍数，4.12 格式。																												

VHDDIECTRL

VHDDIECTRL 为 VHD 通道 De-interlace 运算控制寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value																												
0x0170		VHDDIECTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	die_luma_en	die_chroma_en	reserved	die_ftt	reserved	die_lmmode	reserved	die_chmode	die_rf_mode	reserved	lm_mov_tsmix_en	lm_st_tsmix_en	lm_tfft_en	chm_tsmix_en	stinfo_rst	stinfo_stop	reserved										die_reff_cfg_en	die_reff_cfg				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																												
[31]	RW	die_luma_en		De-interlace 亮度使能。 0: 禁止; 1: 使能。																												
[30]	RW	die_chroma_en		De-interlace 色度使能。 0: 禁止; 1: 使能。																												
[29]	-	reserved		保留。																												



[28]	RW	die_frt	De-interlace 输出帧率比。 0: 输入场率=输出帧率; 1: 输入场率=2×输出帧率。
[27]	-	reserved	保留。
[26]	RW	die_lmmode	De-interlace 亮度运算模式。 0: 4 场模式; 1: 2 场中值方式。
[25]	-	reserved	保留。
[24]	RW	die_chmmode	De-interlace 色度运算模式。 0: 4 场模式; 1: 2 场中值方式。
[23]	RW	die_rf_mode	De-interlace4 场模式下参考场的省带宽控制。 0: 参考场读入整场数据; 1: 参考场只读入 1/2 场数据。
[22]	-	reserved	保留。
[21]	RW	lm_mov_tsmix_en	亮度时空加权使能（运动部分）。 0: 禁止; 1: 使能。
[20]	RW	lm_st_tsmix_en	亮度时空加权使能（静止部分）。 0: 禁止; 1: 使能。
[19]	RW	lm_tflt_en	亮度时域滤波使能。 0: 禁止; 1: 使能。
[18]	RW	chm_tsmix_en	色度时空加权使能。 0: 禁止; 1: 使能。
[17]	RW	stinfo_rst	静止次数复位控制。 0: 静止次数正常更新; 1: 静止次数清零复位。
[16]	RW	stinfo_stop	静止次数更新暂停控制。 0: 静止次数正常更新; 1: 静止次数暂停更新。
[15:2]	-	reserved	保留。



[1]	RW	die_reff_cfg_en	<p>参考场配置控制。</p> <p>0: De-interlace 使用硬件产生的参考场；</p> <p>1: De-interlace 使用软件配置的参考场，即 die_reff_cfg 配置有效。</p> <p>注意：De-interlace 模式下，VHDCTRL[regup_date]配置为 1 时，该比特有效。</p>
[0]	RW	die_reff_cfg	<p>De-interlace 软件配置的参考场（保留场）。</p> <p>0: 保留顶场；</p> <p>1: 保留底场。</p> <p>注意：当 die_reff_cfg_en 配置为 1 时，该比特有效。</p>

VHDDIETHD

VHDDIETHD 为 VHD 通道 De-interlace 运算流水线寄存器，为非即时寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x0174								VHDDIETHD								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								fld_diff_thd				med_thd				reserved	st_thd	reserved	md_thd												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:22]	-		reserved		保留。																											
[21:16]	RW		fld_diff_thd		场差流水线，用于时空加权的系数选择。																											
[15:12]	RW		med_thd		中值检测流水线。																											
[11]	-		reserved		保留。																											
[10:8]	RW		st_thd		静止次数流水线。																											
[7:5]	-		reserved		保留。																											
[4:0]	RW		md_thd		动静判决的运动流水线。																											

VHDDIEADDR

VHDDIEADDR 为 VHD 通道 De-interlace 历史信息 buffer 地址寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0178		VHDDIEADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	dieaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	dieaddr	De-interlace 历史信息 buffer 地址。					

VHDDIETSMIX

VHDDIETSMIX 为 VHD 通道 De-interlace 时空加权系数寄存器，为即时寄存器。时空加权系数共 4 组，每组 8 个系数。

Offset Address		Register Name		Total Reset Value				
0x0180+n1×0x4		VHDDIETSMIX		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tsmix_n7	tsmix_n6	tsmix_n5	tsmix_n4	tsmix_n3	tsmix_n2	tsmix_n1	tsmix_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	RW	tsmix_n7	n1 组时空加权系数 7 (n1=0~3)。					
[27:24]	RW	tsmix_n6	n1 组时空加权系数 6 (n1=0~3)。					
[23:20]	RW	tsmix_n5	n1 组时空加权系数 5 (n1=0~3)。					
[19:16]	RW	tsmix_n4	n1 组时空加权系数 4 (n1=0~3)。					
[15:12]	RW	tsmix_n3	n1 组时空加权系数 3 (n1=0~3)。					
[11:8]	RW	tsmix_n2	n1 组时空加权系数 2 (n1=0~3)。					
[7:4]	RW	tsmix_n1	n1 组时空加权系数 1 (n1=0~3)。					
[3:0]	RW	tsmix_n0	n1 组时空加权系数 0 (n1=0~3)。					

VHDDIETFLT

VHDDIETFLT 为 VHD 通道 De-interlace 时域滤波系数寄存器，为即时寄存器。时域滤波系数共 4 组，每组 8 个系数。



Offset Address		Register Name		Total Reset Value				
0x0190+n2×0x4		VHDDIETFLT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tflt_n7	tflt_n6	tflt_n5	tflt_n4	tflt_n3	tflt_n2	tflt_n1	tflt_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	RW	tflt_n7	n2 组时域滤波系数 7 (n2=0~3)。					
[27:24]	RW	tflt_n6	n2 组时域滤波系数 6 (n2=0~3)。					
[23:20]	RW	tflt_n5	n2 组时域滤波系数 5 (n2=0~3)。					
[19:16]	RW	tflt_n4	n2 组时域滤波系数 4 (n2=0~3)。					
[15:12]	RW	tflt_n3	n2 组时域滤波系数 3 (n2=0~3)。					
[11:8]	RW	tflt_n2	n2 组时域滤波系数 2 (n2=0~3)。					
[7:4]	RW	tflt_n1	n2 组时域滤波系数 1 (n2=0~3)。					
[3:0]	RW	tflt_n0	n2 组时域滤波系数 0 (n2=0~3)。					

VHDDIEVFLT

VHDDIEVFLT 为 VHD 通道 De-interlace 垂直滤波加权系数寄存器，为即时寄存器。运算参数共 4 组，每组 8 个系数。

Offset Address		Register Name		Total Reset Value				
0x01A0+n3×0x4		VHDDIEVFLT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vflt_n7	vflt_n6	vflt_n5	vflt_n4	vflt_n3	vflt_n2	vflt_n1	vflt_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	RW	vflt_n7	n3 组 De-interlace 垂直滤波加权系数 7 (n3=0~3)。					
[27:24]	RW	vflt_n6	n3 组 De-interlace 垂直滤波加权系数 6 (n3=0~3)。					
[23:20]	RW	vflt_n5	n3 组 De-interlace 垂直滤波加权系数 5 (n3=0~3)。					
[19:16]	RW	vflt_n4	n3 组 De-interlace 垂直滤波加权系数 4 (n3=0~3)。					
[15:12]	RW	vflt_n3	n3 组 De-interlace 垂直滤波加权系数 3 (n3=0~3)。					
[11:8]	RW	vflt_n2	n3 组 De-interlace 垂直滤波加权系数 2 (n3=0~3)。					



[7:4]	RW	vflt_n1	n3 组 De-interlace 垂直滤波加权系数 1 (n3=0~3)。
[3:0]	RW	vflt_n0	n3 组 De-interlace 垂直滤波加权系数 0 (n3=0~3)。

VHDSTATUS

VHDSTATUS 为 VHD 通道视频层状态寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x01F0				VHDSTATUS				0x0800_0001																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										die_ref_field					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bits	Access		Name		Description																											
[31:1]	-		reserved		保留。																											
[0]	RO		die_ref_field		De-interlace 运算的参考场。当 de-interlace 不使能时，该寄存器为 0。 0: 参考顶场; 1: 参考底场。																											

VADCTRL

VADCTRL 为 VAD 控制寄存器，为非即时寄存器。该寄存器可以配置视频层的相关信息。



Offset Address		Register Name		Total Reset Value																												
0x0200		VADCTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vad_en	reserved												regup_rate	bfield_first	lm_rmode	chm_rmode	reserved								ifmt						
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0							
Bits	Access	Name	Description																													
[31]	RW	vad_en	Surface 使能。 0: 禁止; 1: 使能。																													
[30:18]	-	reserved	保留。																													
[17]	RW	regup_rate	隔行输出时 Surface 的寄存器更新频率。 0: 一帧更新一次; 1: 一场更新一次。																													
[16]	RW	bfield_first	底场优先 (T: Top Field; B: Bottom Field)。 0: 顶场优先, T0B0T1B1 (T0B0 为一帧); 1: 底场优先, B0T0B1T1 (B0T0 为一帧)。																													
[15:14]	RW	lm_rmode	亮度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。																													
[13:12]	RW	chm_rmode	色度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。																													
[11:4]	-	reserved	保留。																													
[3:0]	RW	ifmt	输入数据格式。 0x3: SPYCbCr4:2:0; 0x4: SPYCbCr4:2:2(1x2 格式); 其他: 保留。																													



VADUPD

VADUPD 为 VAD 通道更新使能寄存器。

Offset Address		Register Name		Total Reset Value					
0x0204		VADUPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

VADLADDR

VADLADDR 为 VAD 通道 De-interlace 上一帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0208		VADLADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadladdr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadladdr	De-interlace 上一帧的地址。					

VADLCADDR

VADLCADDR 为 VAD 通道 De-interlace 上一帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。



Offset Address		Register Name		Total Reset Value				
0x020C		VADLCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadlcaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadlcaddr	De-interlace 上一帧的色度地址。					

VADCADDR

VADCADDR 为 VAD 通道 De-interlace 当前帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0210		VADCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadccaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadccaddr	De-interlace 当前帧的地址。					

VADCCADDR

VADCCADDR 为 VAD 通道 De-interlace 当前帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0214		VADCCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadccaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadccaddr	De-interlace 当前帧的色度地址。					



VADNADDR

VADNADDR 为 VAD 通道 De-interlace 下一帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0218		VADNADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadnaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadnaddr	De-interlace 下一帧的地址。					

VADNCADDR

VADNCADDR 为 VAD 通道 De-interlace 下一帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x021C		VADNCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadncaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vadncaddr	De-interlace 下一帧的色度地址。					

VADSTRIDE

VADSTRIDE 为 VAD 通道行偏移地址寄存器。



Offset Address		Register Name		Total Reset Value				
0x0220		VADSTRIDE		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vadcstride				vadstride			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	RW	vadcstride	色度帧 buffer 的 stride（针对 semi-planar 格式），以 word 为单位。					
[15:0]	RW	vadstride	帧 buffer 的 stride（针对 semi-planar 格式，指亮度的 stride），以 word 为单位。					

VADCBMPARA

VADCBMPARA 为 VAD 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0224		VADCBMPARA		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved						galpha	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:8]	-	reserved	保留。					
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127，0 表示全透明，127 表示完全不透明。					

VADORESO

VADORESO 为 VAD 通道输出分辨率寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0228		VADORESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	oh			reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	oh	输出视频图像的高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，视频层的实际高度必须是偶数。逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	ow	输出视频图像的宽度，以像素为单位。 配置值为实际宽度减 1。 注意：视频层的实际宽度必须是偶数。						

VADIRESO

VADIRESO 为 VAD 通道输入分辨率寄存器，为非即时寄存器。



注意

- 隔行输出时，实际高度只能配置为偶数，当输入数据格式为 YCbCr4:2:0 时，高度必须是 4 的整数倍。
- 逐行输出时，当输入数据格式为 YCbCr4:2:0 时，高度必须是 2 的整数倍；视频层的实际宽度必须是偶数。

Offset Address		Register Name		Total Reset Value					
0x022C		VADIRESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	ih			reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						



[27:16]	RW	ih	输入视频图像的高度，以行为单位。 配置值为实际高度减 1。
[15:12]	RW	reserved	保留。
[11:0]	RW	iw	输入视频图像的宽度，以像素为单位。 配置值为实际宽度减 1。

VADSFPOS

VADSFPOS 为 VAD 通道源位图的起始位置寄存器，为非即时寄存器。

	Offset Address 0x0230				Register Name VADSFPOS				Total Reset Value 0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				src_yfpos				reserved				src_xfpos																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		src_yfpos		源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。																											
[15:12]	-		reserved		保留。																											
[11:0]	RW		src_xfpos		源起始 X 坐标。 0 为第一行第一个像素。																											

VADDFPOS

VADDFPOS 为 VAD 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0234		VADDFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	disp_yfpos			reserved	disp_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	disp_yfpos	显示列起始坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	disp_xfpos	显示行起始坐标。						

VADDLPOS

VADDLPOS 为 VAD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0238		VADDLPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	disp_ylpos			reserved	disp_xlpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	disp_ylpos	显示列结束坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	disp_xlpos	显示行结束坐标。						

VADVFPOS

VADVFPOS 为 VAD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x023C		VADVFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	video_yfpos			reserved	video_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	video_yfpos	视频内容列起始坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	video_xfpos	视频内容行起始坐标。						

VADVLPOS

VADVLPOS 为 VAD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0240		VADVLPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	video_ylpos			reserved	video_xlpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	video_ylpos	视频内容列结束坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	video_xlpos	视频内容行结束坐标。						

VADBK

VADBK 为 VAD 通道视频层的背景色寄存器。



Offset Address		Register Name		Total Reset Value				
0x0244		VADBK		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vbk_alpha		vbk_y		vbk_cb		vbk_cr	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	vbk_alpha	视频层的背景填充色。 取值范围为 0~128。					
[23:16]	RW	vbk_y	Y 分量。					
[15:8]	RW	vbk_cb	Cb 分量。					
[7:0]	RW	vbk_cr	Cr 分量。					

VADLMSP

VADLMSP 为 VAD 通道亮度缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value				
0x0250		VADLMSP		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	hlmsc_en	vlmsc_en	hlmid_en	vlmid_en	reserved		shift_field	fld_offset
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31]	RW	hlmsc_en	水平亮度缩放使能。 0: 禁止; 1: 使能。					
[30]	RW	vlmsc_en	垂直亮度缩放使能。 0: 禁止; 1: 使能。					
[29]	RW	hlmid_en	水平亮度缩放中值滤波使能。 0: 禁止; 1: 使能。					



[28]	RW	vlmid_en	垂直亮度缩放中值滤波使能。 0: 禁止; 1: 使能。
[27:17]	-	reserved	保留。
[16]	RW	shift_field	位置偏移所影响的场。 0: 底场显示时偏移; 1: 顶场显示时偏移。
[15:0]	RW	fld_offset	场位置偏移。 补码表示, 最高位为符号位, 4.12 格式。 注意: 该值的配置与缩放比例有关。

VADCHMSP

VADCHMSP 为 VAD 通道色度缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value																												
0x0254		VADCHMSP		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	hchmsc_en	vchmsc_en	hlmid_en	vlmid_en	reserved								shift_field	fld_offset																		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31]	RW		hchmsc_en		水平色度缩放使能。 0: 禁止; 1: 使能。																											
[30]	RW		vchmsc_en		垂直色度缩放使能。 0: 禁止; 1: 使能。																											
[29]	RW		hlmid_en		水平色度缩放中值滤波使能。 0: 禁止; 1: 使能。																											



[28]	RW	vlmid_en	垂直色度缩放中值滤波使能。 0: 禁止; 1: 使能。
[27:17]	-	reserved	保留。
[16]	RW	shift_field	位置偏移所影响的场。 0: 底场显示时偏移; 1: 顶场显示时偏移。
[15:0]	RW	fld_offset	场位置偏移。 补码表示, 最高位为符号位, 4.12 格式。 该值的配置与缩放比例有关。

VADLMHSP

VADLMHSP 为 VAD 通道亮度水平缩放参数配置寄存器, 为非即时寄存器。

亮度水平缩放比率 = 输入宽度/输出宽度。

Offset Address		Register Name		Total Reset Value					
0x0258		VADLMHSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	lm_hphase			lm_hratio				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name		Description					
[31:28]	-	reserved		保留。					
[27:16]	RW	lm_hphase		亮度水平缩放初始相位, 0.12 格式。					
[15:0]	RW	lm_hratio		亮度水平缩放倍数, 4.12 格式。					

VADLMVSP

VADLMVSP 为 VAD 通道亮度垂直缩放参数配置寄存器, 为非即时寄存器。

亮度垂直缩放比率 = 输入高度/输出高度。



Offset Address		Register Name		Total Reset Value					
0x025C		VADLMVSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	lm_vphase			lm_vratio				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	lm_vphase	亮度垂直缩放初始相位，0.12 格式。						
[15:0]	RW	lm_vratio	亮度垂直缩放倍数，4.12 格式。						

VADCHMHSP

VADCHMHSP 为 VAD 通道色度水平缩放参数配置寄存器。

Offset Address		Register Name		Total Reset Value					
0x0260		VADCHMHSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	chm_hphase			chm_hratio				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	chm_hphase	色度水平缩放初始相位，0.12 格式。						
[15:0]	RW	chm_hratio	色度水平缩放倍数，4.12 格式。						

VADCHMVSP

VADCHMVSP 为 VAD 通道色度垂直缩放参数配置寄存器。



Offset Address		Register Name		Total Reset Value					
0x0264		VADCHMVSP		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	chm_hphase				chm_vratio			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	chm_hphase	色度垂直缩放初始相位，最高位为符号位，1.12 格式。						
[15:0]	RW	chm_vratio	色度垂直缩放倍数，4.12 格式。						

VADDIECTRL

VADDIECTRL 为 VAD 通道 De-interlace 运算控制寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value						
0x0270		VADDIECTRL		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	die_luma_en die_chroma_en reserved die_ftt reserved die_lmmode reserved die_chmode die_rf_mode reserved lm_mov_tsmix_en lm_st_tsmix_en lm_tft_en chm_tsmix_en stinfo_rst stinfo_stop	reserved				reserved				die_refc_cfg_en die_refc_cfg
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31]	RW	die_luma_en	De-interlace 亮度使能。 0: 禁止; 1: 使能。							
[30]	RW	die_chroma_en	De-interlace 色度使能。 0: 禁止; 1: 使能。							
[29]	-	reserved	保留。							



[28]	RW	die_frt	De-interlace 输出帧率比。 0: 输入场率=输出帧率; 1: 输入场率=2×输出帧率。
[27]	-	reserved	保留。
[26]	RW	die_lmmode	De-interlace 亮度运算模式。 0: 4 场模式; 1: 2 场中值方式。
[25]	-	reserved	保留。
[24]	RW	die_chmmode	De-interlace 色度运算模式。 0: 4 场模式; 1: 2 场中值方式。
[23]	RW	die_rf_mode	De-interlace4 场模式下参考场的省带宽控制。 0: 参考场读入整场数据; 1: 参考场只读入 1/2 场数据。
[22]	-	reserved	保留。
[21]	RW	lm_mov_tsmix_en	亮度时空加权使能（运动部分）。 0: 禁止; 1: 使能。
[20]	RW	lm_st_tsmix_en	亮度时空加权使能（静止部分）。 0: 禁止; 1: 使能。
[19]	RW	lm_tflt_en	亮度时域滤波使能。 0: 禁止; 1: 使能。
[18]	RW	chm_tsmix_en	色度时空加权使能。 0: 禁止; 1: 使能。
[17]	RW	stinfo_rst	静止次数复位控制。 0: 静止次数正常更新; 1: 静止次数清零复位。
[16]	RW	stinfo_stop	静止次数更新暂停控制。 0: 静止次数正常更新; 1: 静止次数暂停更新。
[15:2]	-	reserved	保留。



[1]	RW	die_reff_cfg_en	<p>参考场配置控制。</p> <p>0: De-interlace 使用硬件产生的参考场；</p> <p>1: De-interlace 使用软件配置的参考场，即 die_reff_cfg 配置有效。</p> <p>注意：De-interlace 模式下，VADCTRL[regup_date]配置为 1 时，该比特有效。</p>
[0]	RW	die_reff_cfg	<p>De-interlace 软件配置的参考场（保留场）。</p> <p>0: 保留顶场；</p> <p>1: 保留底场。</p> <p>注意：当 die_reff_cfg_en 配置为 1 时，该比特有效。</p>

VADDIETHD

VADDIETHD 为 VAD 通道 De-interlace 运算流水线寄存器，为非即时寄存器。

	Offset Address	Register Name	Total Reset Value										
	0x0274	VADDIETHD	0x0000_0000										
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0					
Name	reserved				fld_diff_thd		med_thd	reserved	st_thd	reserved		md_thd	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
	Bits	Access	Name	Description									
	[31:22]	-	reserved	保留。									
	[21:16]	RW	fld_diff_thd	场差流水线，用于时空加权的系数选择。									
	[15:12]	RW	med_thd	中值检测流水线。									
	[11]	-	reserved	保留。									
	[10:8]	RW	st_thd	静止次数流水线									
	[7:5]	-	reserved	保留。									
	[4:0]	RW	md_thd	动静判决的运动流水线。									

VADDIEADDR

VADDIEADDR 为 VAD 通道 De-interlace 历史信息 buffer 地址寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0278		VADDIEADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	dieaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	dieaddr	De-interlace 历史信息 buffer 地址。					

VADDIETSMIX

VADDIETSMIX 为 VAD 通道 De-interlace 时空加权系数寄存器，为即时寄存器。时空加权系数共 4 组，每组 8 个系数。

Offset Address		Register Name		Total Reset Value				
0x0280+n1×0x4		VADDIETSMIX		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tsmix_n7	tsmix_n6	tsmix_n5	tsmix_n4	tsmix_n3	tsmix_n2	tsmix_n1	tsmix_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	RW	tsmix_n7	n1 组时空加权系数 7 (n1=0~3)。					
[27:24]	RW	tsmix_n6	n1 组时空加权系数 6 (n1=0~3)。					
[23:20]	RW	tsmix_n5	n1 组时空加权系数 5 (n1=0~3)。					
[19:16]	RW	tsmix_n4	n1 组时空加权系数 4 (n1=0~3)。					
[15:12]	RW	tsmix_n3	n1 组时空加权系数 3 (n1=0~3)。					
[11:8]	RW	tsmix_n2	n1 组时空加权系数 2 (n1=0~3)。					
[7:4]	RW	tsmix_n1	n1 组时空加权系数 1 (n1=0~3)。					
[3:0]	RW	tsmix_n0	n1 组时空加权系数 0 (n1=0~3)。					

VADDIETFLT

VADDIETFLT 为 VAD 通道 De-interlace 时域滤波系数寄存器，为即时寄存器。时域滤波系数共 4 组，每组 8 个系数。



	Offset Address 0x0290+n2×0x4								Register Name VADDIETFLT								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	tflt_n7				tflt_n6				tflt_n5				tflt_n4				tflt_n3				tflt_n2				tflt_n1				tflt_n0			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0							
Bits	[31:28]				[27:24]				[23:20]				[19:16]				[15:12]				[11:8]				[7:4]				[3:0]			
Access	RW				RW				RW				RW				RW				RW				RW							
Name	tflt_n7				tflt_n6				tflt_n5				tflt_n4				tflt_n3				tflt_n2				tflt_n1				tflt_n0			
Description	n2 组时域滤波系数 7 (n2=0~3)。																															
n2 组时域滤波系数 6 (n2=0~3)。																																
n2 组时域滤波系数 5 (n2=0~3)。																																
n2 组时域滤波系数 4 (n2=0~3)。																																
n2 组时域滤波系数 3 (n2=0~3)。																																
n2 组时域滤波系数 2 (n2=0~3)。																																
n2 组时域滤波系数 1 (n2=0~3)。																																
n2 组时域加权系数 0 (n2=0~3)。																																

VADDIEVFLT

VADDIEVFLT 为 VAD 通道 De-interlace 垂直滤波加权系数寄存器，为即时寄存器。运算参数共 4 组，每组 8 个系数。

	Offset Address 0x02A0+n3×0x4								Register Name VADDIEVFLT								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vflt_n7				vflt_n6				vflt_n5				vflt_n4				vflt_n3				vflt_n2				vflt_n1				vflt_n0			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0							
Bits	[31:28]				[27:24]				[23:20]				[19:16]				[15:12]				[11:8]											
Access	RW				RW				RW				RW				RW				RW											
Name	vflt_n7				vflt_n6				vflt_n5				vflt_n4				vflt_n3				vflt_n2											
Description	n3 组 De-interlace 垂直滤波加权系数 7 (n3=0~3)。																															
n3 组 De-interlace 垂直滤波加权系数 6 (n3=0~3)。																																
n3 组 De-interlace 垂直滤波加权系数 5 (n3=0~3)。																																
n3 组 De-interlace 垂直滤波加权系数 4 (n3=0~3)。																																
n3 组 De-interlace 垂直滤波加权系数 3 (n3=0~3)。																																
n3 组 De-interlace 垂直滤波加权系数 2 (n3=0~3)。																																



[7:4]	RW	vflt_n1	n3 组 De-interlace 垂直滤波加权系数 1 (n3=0~3)。
[3:0]	RW	vflt_n0	n3 组 De-interlace 垂直滤波加权系数 0 (n3=0~3)。

VADSTATUS

VADSTATUS 为 VAD 通道视频层状态寄存器。

	Offset Address	Register Name	Total Reset Value
	0x02F0	VADSTATUS	0x0800_0001
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		die_ref_field
Reset	0 0 0 0 1 0 1		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RO	die_ref_field	De-interlace 运算的参考场。当 de-interlace 禁止时，该域为 0。 0: 参考顶场; 1: 参考底场。

VSDCTRL

VSDCTRL 为 VSD 通道控制寄存器。



Offset Address		Register Name		Total Reset Value																												
0x0300		VSDCTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vsd_en	reserved																regup_rate	bfield_first	lm_rmode	chm_rmode	reserved								ifmt		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31]	RW	vsd_en	Surface 使能。 0: 禁止; 1: 使能。																													
[30:18]	-	reserved	保留。																													
[17]	RW	regup_rate	隔行输出时 surface 的寄存器更新频率。 0: 一帧更新一次; 1: 一场更新一次。																													
[16]	RW	bfield_first	底场优先 (T: Top Field; B: Bottom Field)。 0: 顶场优先, T0B0T1B1 (T0B0 为一帧); 1: 底场优先, B0T0B1T1 (B0T0 为一帧)。																													
[15:14]	RW	lm_rmode	亮度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。																													
[13:12]	RW	chm_rmode	色度逐隔行读取模式。 00: 隔行读取帧 buffer 数据, 并根据时序顺序读入顶底场; 01: 逐行读取帧 buffer 数据; 10: 隔行显示一帧时仅读入顶场; 11: 隔行显示一帧时仅读入底场。																													
[11:4]	-	reserved	保留。																													
[3:0]	RW	ifmt	输入数据格式。 0x3: SPYCbCr4:2:0; 0x4: SPYCbCr4:2:2(1x2 格式); 其他: 保留。																													



VSDUPD

VSDUPD 为 VSD 通道更新使能寄存器。

Offset Address		Register Name		Total Reset Value					
0x0304		VSDUPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

VSDADDR

VSDADDR 为 VSD 通道当前帧地址寄存器。对于 package 像素格式是帧 buffer 地址；对于 semi-planar 像素格式是亮度帧 buffer 地址。

Offset Address		Register Name		Total Reset Value				
0x0310		VSDADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vsdaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vsdaddr	Surface 帧 buffer 地址。					

VSDCADDR

VSDCADDR 为 VSD 通道当前帧色度地址寄存器。对于 package 像素格式该地址无效；对于 semi-planar 像素格式是色度帧 buffer 地址。



Offset Address		Register Name		Total Reset Value				
0x0314		VSDCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vsdcaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	vsdcaddr	Surface 色度帧 buffer 地址，以 word 为单位。					

VSDSTRIDE

VSDSTRIDE 为 VSD 通道行偏移地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0320		VSDSTRIDE		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vsdcstride				vsdstride			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	RW	vsdcstride	色度帧 buffer 的 stride（针对 semi-planar 格式），以 word 为单位。					
[15:0]	RW	vsdstride	帧 buffer 的 stride（针对 semi-planar 格式，指亮度的 stride），以 word 为单位。					

VSDCBMPARA

VSDCBMPARA 为 VSD 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0324		VSDCBMPARA		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved						alpha	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:8]	-	reserved	保留。					



[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127，0 为全透明，127 为完全不透明。
-------	----	--------	---

VSDORES0

VSDORES0 为 VSD 通道输出分辨率寄存器，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0328				VSDORES0				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				oh				reserved				ow																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:28]	-	reserved	保留。																													
[27:16]	RW	oh	输出的视频图像高度，以行为单位。 配置值实际高度减 1。 注意：对于隔行输出，视频层的实际高度必须是偶数。逐行输出无此限制。																													
[15:12]	RW	reserved	保留。																													
[11:0]	RW	ow	输出的视频图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：视频层的实际宽度必须是偶数。																													

VSDIRES0

VSDIRES0 为 VSD 通道输入分辨率寄存器，为非即时寄存器。



- 隔行输出时，实际高度只能配置为偶数，当输入数据格式为 YCbCr4:2:0 时，高度必须是 4 的整数倍。
- 逐行输出时，当输入数据格式为 YCbCr4:2:0 时，高度必须是 2 的整数倍；视频层的实际宽度必须是偶数。



Offset Address		Register Name		Total Reset Value					
0x032C		VSDIRESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	ih			reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	ih	输入的视频图像高度，以行为单位。 配置值为实际高度减 1。						
[15:12]	-	reserved	保留。						
[11:0]	RW	iw	输入的视频图像宽度，以像素单位。 配置值为实际宽度减 1。						

VSDSFPOS

VSDSFPOS 为 VSD 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0330		VSDSFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	src_yfpos			reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。						
[15:12]	RW	reserved	保留。						
[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。						

VSDDFPOS

VSDDFPOS 为 VSD 通道显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0334		VSDDFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	disp_yfpos			reserved	disp_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	disp_yfpos	显示列起始坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	disp_xfpos	显示行起始坐标。						

VSDDLPOS

VSDDLPOS 为 VSD 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0338		VSDDLPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	disp_ylpos			reserved	disp_xlpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	disp_ylpos	显示列结束坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	disp_xlpos	显示行结束坐标。						

VSDVFPOS

VSDVFPOS 为 VSD 通道视频内容在显示窗口的起始位置寄存器，以像素为单位，为非即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x033C		VSDVFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	video_yfpos			reserved	video_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	video_yfpos	视频内容列起始坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	video_xfpos	视频内容行起始坐标。						

VSDVLPOS

VSDVLPOS 为 VSD 通道视频内容在显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0340		VSDVLPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	video_ylpos			reserved	video_xlpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	video_ylpos	视频内容列结束坐标。						
[15:12]	-	reserved	保留。						
[11:0]	RW	video_xlpos	视频内容行结束坐标。						

VSDBK

VSDBK 为 VSD 通道视频层的背景色寄存器。



Offset Address		Register Name		Total Reset Value				
0x0344		VSDBK		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vbk_alpha		vbk_y		vbk_cb		vbk_cr	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	vbk_alpha	视频层的背景填充色。 取值范围为 0~128。					
[23:16]	RW	vbk_y	Y 分量。					
[15:8]	RW	vbk_cb	Cb 分量。					
[7:0]	RW	vbk_cr	Cr 分量。					

G0CTRL

G0CTRL 为 G0 通道控制寄存器，为非即时寄存器。该寄存器可以配置图形层的相关信息。

Offset Address		Register Name		Total Reset Value				
0x0400		G0CTRL		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	g0_en	csc_en	csc_mode	reserved			bitext	ifmt
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31]	RW	g0_en	Surface 使能。 0: 禁止; 1: 使能。					
[30]	RW	csc_en	颜色空间转换使能。 0: 禁止; 1: 使能。					
[29]	RW	csc_mode	颜色空间转换标准。 0: BT.601; 1: BT.709。					
[28:10]	-	reserved	保留。					



[9:8]	RW	bitext	Surface 输入位图 bit 位扩展模式。 0X: 低位扩展 0; 10: 低位扩展最高 bit 位; 11: 低位扩展最高若干 bit 位。
[7:0]	RW	ifmt	输入数据格式。 0x49: aRGB1555; 其他: 保留。

G0UPD

G0UPD 为 G0 通道更新使能寄存器。

Offset Address		Register Name		Total Reset Value					
0x0404		G0UPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

G0ADDR

G0ADDR 为 G0 通道帧地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0408		G0ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	g0addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	g0addr	Surface 帧 buffer 地址。					



G0STRIDE

G0STRIDE 为 G0 通道帧偏移地址寄存器。

Offset Address		Register Name		Total Reset Value					
0x040C		G0STRIDE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				g0stride				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	g0stride	帧 buffer 的行偏移。						

G0CBMPARA

G0CBMPARA 为 G0 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value							
0x0410		G0CBMPARA		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved				key_mode	key_en	reserved	palpha_en	reserved	palpha_range	alpha
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description								
[31:16]	-	reserved	保留。								
[15]	RW	key_mode	Color key 模式。 0: 当 $Keymin \leq Pixel \leq Keymax$ 时，处理为关键色； 1: 当 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时，处理为关键色。								
[14]	RW	key_en	Color key 使能。 0: 禁止； 1: 使能。								



[13]	-	reserved	保留。
[12]	RW	palpha_en	像素 alpha 使能。 0: 禁止; 1: 使能。
[11:9]	-	reserved	保留。
[8]	RW	palpha_range	像素 alpha 的范围。 0: 0~128; 1: 0~255。
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。

G0CKEYMAX

G0CKEYMAX 为 G0 通道 color key 最大值寄存器，为非即时寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0414	G0CKEYMAX	0x0000_0000
Bit	31 30 29 28	27 26 25 24	23 22 21 20
	19 18 17 16	15 14 13 12	11 10 9 8
	7 6 5 4	3 2 1 0	
Name	va0	keyr_max	keyg_max
	keyb_max		
Reset	0 0 0 0	0 0 0 0	0 0 0 0
	0 0 0 0	0 0 0 0	0 0 0 0
	0 0 0 0	0 0 0 0	0 0 0 0
	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description
[31:24]	RW	va0	Alpha0 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时， alpha 值用该值替换。
[23:16]	RW	keyr_max	Color key R 分量最大值。
[15:8]	RW	keyg_max	Color key G 分量最大值。
[7:0]	RW	keyb_max	Color key B 分量最大值。

G0CKEYMIN

G0CKEYMIN 为 G0 通道 color key 最小值寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0418		G0CKEYMIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	val		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	val	Alpha1 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。					
[23:16]	RW	keyr_min	Color key R 分量最小值。					
[15:8]	RW	keyg_min	Color key G 分量最小值。					
[7:0]	RW	keyb_min	Color key B 分量最小值。					

G0IRESO

G0IRESO 为 G0 通道输入分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x041C		G0IRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	图形层输入图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	图形层输入图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					



GOORESO

GOORESO 为 G0 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0420		GOORESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	oh		reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	oh	图形层输出图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	ow	图形层输出图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					

G0SFPOS

G0SFPOS 为 G0 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0424		G0SFPOS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	src_yfpos		reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。					
[15:12]	-	reserved	保留。					



[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。
--------	----	-----------	---------------------------

G0DFPOS

G0DFPOS 为 G0 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0428		G0DFPOS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	disp_yfpos		reserved	disp_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	disp_yfpos	列起始坐标。					
[15:12]	-	reserved	保留。					
[11:0]	RW	disp_xfpos	行起始坐标。					

G0DLPOS

G0DLPOS 为 G0 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x042C		G0DLPOS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	disp_ylpos		reserved	disp_xlpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	disp_ylpos	列结束坐标。					
[15:12]	-	reserved	保留。					
[11:0]	RW	disp_xlpos	行结束坐标。					

G1CTRL

G1CTRL 为 G1 通道控制寄存器，为非即时寄存器。该寄存器可以配置图形层的相关信息。

Offset Address		Register Name		Total Reset Value																												
0x0500		G1CTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	gl_en	csc_en	csc_mode	reserved																		bitext	ifmt									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name		Description																												
[31]	RW	gl_en		Surface 使能。 0: 禁止; 1: 使能。																												
[30]	RW	csc_en		颜色空间转换使能。 0: 禁止; 1: 使能。																												
[29]	RW	csc_mode		颜色空间转换标准。 0: BT.601; 1: BT.709。																												
[28:10]	-	reserved		保留。																												
[9:8]	RW	bitext		Surface 输入位图 bit 位扩展模式。 0X: 低位扩展 0; 10: 低位扩展最高 bit 位; 11: 低位扩展最高若干 bit 位。																												
[7:0]	RW	ifmt		输入数据格式。 0x49: aRGB1555; 其他: 保留。																												

G1UPD

G1UPD 为 G1 通道更新使能寄存器。



Offset Address		Register Name		Total Reset Value					
0x0504		G1UPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

G1ADDR

G1ADDR 为 G1 通道帧地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0508		G1ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	gladdr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	gladdr	Surface 帧 buffer 地址。					

G1STRIDE

G1STRIDE 为 G1 通道帧偏移地址寄存器。

Offset Address		Register Name		Total Reset Value					
0x050C		G1STRIDE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				g1stride				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	g1stride	帧 buffer 的行偏移。						

G1CBMPARA

G1CBMPARA 为 G1 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value							
0x0510		G1CBMPARA		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved				key_mode	key_en	reserved	palpha_en	reserved	palpha_range	alpha
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description								
[31:16]	-	reserved	保留。								
[15]	RW	key_mode	Color key 模式。 0: 当 $Keymin \leq Pixel \leq Keymax$ 时，处理为关键色； 1: 当 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时，处理为关键色。								
[14]	RW	key_en	Color key 使能。 0: 禁止； 1: 使能。								
[13]	-	reserved	保留。								
[12]	RW	palpha_en	像素 alpha 使能。 0: 禁止； 1: 使能。								



[11:9]	RW	reserved	保留。
[8]	RW	palpha_range	像素 alpha 的范围。 0: 0~128; 1: 0~255。
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。

G1CKEYMAX

G1CKEYMAX 为 G1 通道 color key 最大值寄存器，为非即时寄存器。

	Offset Address 0x0514								Register Name G1CKEYMAX								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0								keyr_max								keyg_max								keyb_max							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description																												
	[31:24]	RW	va0	Alpha0 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。																												
	[23:16]	RW	keyr_max	Color key R 分量最大值。																												
	[15:8]	RW	keyg_max	Color key G 分量最大值。																												
	[7:0]	RW	keyb_max	Color key B 分量最大值。																												

G1CKEYMIN

G1CKEYMIN 为 G1 通道 color key 最小值寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0518		G1KEYMIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	val		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	val	Alpha1 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。					
[23:16]	RW	keyr_min	Color key R 分量最小值。					
[15:8]	RW	keyg_min	Color key G 分量最小值。					
[7:0]	RW	keyb_min	Color key B 分量最小值。					

G1IRESO

G1IRESO 为 G1 通道输入分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x051C		G1IRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	图形层输入图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	图形层输入图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					



G1ORESO

G1ORESO 为 G1 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0520		G1ORESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	oh		reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	oh	图形层输出图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	ow	图形层输出图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					

G1SFPOS

G1SFPOS 为 G1 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0524		G1SFPOS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	src_yfpos		reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。					
[15:12]	-	reserved	保留。					



[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。
--------	----	-----------	---------------------------

G1DFPOS

G1DFPOS 为 G1 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x0528				G1DFPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				yfpos								reserved				xfpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	yfpos		列起始坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	xfpos		行起始坐标。																											

G1DLPOS

G1DLPOS 为 G1 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x052C				G1DLPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				ylpos								reserved				xlpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	ylpos		列结束坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	xlpos		行结束坐标。																											



G2CTRL

G2CTRL 为 G2 通道控制寄存器，为非即时寄存器。该寄存器可以配置图形层的相关信息。

Offset Address		Register Name		Total Reset Value																												
0x0600		G2CTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g2_en	csc_en	csc_mode	reserved												bitext	ifmt															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																												
[31]	RW	g2_en		Surface 使能。 0: 禁止; 1: 使能。																												
[30]	RW	csc_en		颜色空间转换使能。 0: 禁止; 1: 使能。																												
[29]	RW	csc_mode		颜色空间转换标准。 0: BT.601; 1: BT.709。																												
[28:10]	-	reserved		保留。																												
[9:8]	RW	bitext		Surface 输入位图 bit 位扩展模式。 0X: 低位扩展 0; 10: 低位扩展最高 bit 位; 11: 低位扩展最高若干 bit 位。																												
[7:0]	RW	ifmt		输入数据格式。 0x49: aRGB1555; 其他: 保留。																												

G2UPD

G2UPD 为 G2 通道更新使能寄存器。



Offset Address		Register Name		Total Reset Value					
0x0604		G2UPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

G2ADDR

G2ADDR 为 G2 通道帧地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0608		G2ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	g2addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	g2addr	Surface 帧 buffer 地址。					

G2STRIDE

G2STRIDE 为 G2 通道帧偏移地址寄存器。



Offset Address		Register Name		Total Reset Value					
0x060C		G2STRIDE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				g2stride				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	g2stride	帧 buffer 的行偏移。						

G2CBMPARA

G2CBMPARA 为 G2 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value							
0x0610		G2CBMPARA		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved				key_mode	key_en	reserved	palpha_en	reserved	palpha_range	alpha
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description								
[31:16]	-	reserved	保留。								
[15]	RW	key_mode	Color key 模式。 0: 当 $Keymin \leq Pixel \leq Keymax$ 时，处理为关键色； 1: 当 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时，处理为关键色。								
[14]	RW	key_en	Color key 使能。 0: 禁止； 1: 使能。								
[13]	-	reserved	保留。								
[12]	RW	palpha_en	像素 alpha 使能。 0: 禁止； 1: 使能。								



[11:9]	-	reserved	保留。
[8]	RW	palpha_range	像素 alpha 的范围。 0: 0~128; 1: 0~255。
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。

G2CKEYMAX

G2CKEYMAX 为 G2 通道 color key 最大值寄存器，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0614				G2CKEYMAX				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0				keyr_max				keyg_max				keyb_max																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:24]	RW		va0		Alpha0 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。																											
[23:16]	RW		keyr_max		Color key R 分量最大值。																											
[15:8]	RW		keyg_max		Color key G 分量最大值。																											
[7:0]	RW		keyb_max		Color key B 分量最大值。																											

G2CKEYMIN

G2CKEYMIN 为 G2 通道 color key 最小值寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0618		G2CKEYMIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	val		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	val	Alpha1 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。					
[23:16]	RW	keyr_min	Color key R 分量最小值。					
[15:8]	RW	keyg_min	Color key G 分量最小值。					
[7:0]	RW	keyb_min	Color key B 分量最小值。					

G2IRESO

G2IRESO 为 G2 通道输入分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x061C		G2IRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	图形层输入图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	图形层输入图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					



G2ORESO

G2ORESO 为 G2 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0620		G2ORESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	oh			reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	oh	图形层输出图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	ow	图形层输出图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。						

G2SFPOS

G2SFPOS 为 G2 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0624		G2SFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	src_yfpos			reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	src_yfpos	源起始 Y 坐标值。 0 为第一行，隔行输出时，配置值必须是偶数。						
[15:12]	-	reserved	保留。						



[11:0]	RW	src_xfpos	源起始 X 坐标值。 0 为第一行第一个像素。
--------	----	-----------	----------------------------

G2DFPOS

G2DFPOS 为 G2 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x0628				G2DFPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				yfpos								reserved				xfpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		yfpos		列起始坐标。																											
[15:12]	-		reserved		保留。																											
[11:0]	RW		xfpos		行起始坐标。																											

G2DLPOS

G2DLPOS 为 G2 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x062C				G2DLPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				ylpos								reserved				xlpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:16]	RW		ylpos		列结束坐标。																											
[15:12]	-		reserved		保留。																											
[11:0]	RW		xlpos		行结束坐标。																											

G3CTRL

G3CTRL 为 G3 通道控制寄存器，为非即时寄存器。该寄存器可以配置图形层的相关信息。

Offset Address		Register Name		Total Reset Value																												
0x0700		G3CTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g3_en	csc_en	csc_mode	reserved												bitext	ifmt															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31]	RW		g3_en		Surface 使能。 0: 禁止; 1: 使能。																											
[30]	RW		csc_en		颜色空间转换使能。 0: 禁止; 1: 使能。																											
[29]	RW		csc_mode		颜色空间转换标准。 0: BT.601; 1: BT.709。																											
[28:10]	-		reserved		保留。																											
[9:8]	RW		bitext		Surface 输入位图 bit 位扩展模式。 0X: 低位扩展 0; 10: 低位扩展最高 bit 位; 11: 低位扩展最高若干 bit 位。																											
[7:0]	RW		ifmt		输入数据格式。 0x49: aRGB1555; 其他: 保留。																											

G3UPD

G3UPD 为 G3 通道更新使能寄存器。



Offset Address		Register Name		Total Reset Value					
0x0704		G3UPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

G3ADDR

G3ADDR 为 G3 通道帧地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0708		G3ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	g3addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	g3addr	Surface 帧 buffer 地址。					

G3STRIDE

G3STRIDE 为 G3 通道帧偏移地址寄存器。



Offset Address		Register Name		Total Reset Value					
0x070C		G3STRIDE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				g3stride				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	g3stride	帧 buffer 的行偏移。						

G3CBMPARA

G3CBMPARA 为 G3 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value							
0x0710		G3CBMPARA		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved				key_mode	key_en	premult_en	palpha_en	reserved	palpha_range	alpha
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description								
[31:16]	-	reserved	保留。								
[15]	RW	key_mode	Color key 模式。 0: 当 $Keymin \leq Pixel \leq Keymax$ 时，处理为关键色； 1: 当 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时，处理为关键色。								
[14]	RW	key_en	Color key 使能。 0: 禁止； 1: 使能。								
[13]	-	premult_en	保留。								
[12]	RW	palpha_en	像素 alpha 使能。 0: 禁止； 1: 使能。								



[11:9]	-	reserved	保留。
[8]	RW	palpha_range	像素 alpha 的范围。 0: 0~128; 1: 0~255。
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。

G3CKEYMAX

G3CKEYMAX 为 G3 通道 color key 最大值寄存器，为非即时寄存器。

	Offset Address 0x0714								Register Name G3CKEYMAX								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0								keyr_max								keyg_max								keyb_max							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:24]	RW	va0		Alpha0 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。																											
	[23:16]	RW	keyr_max		Color key R 分量最大值。																											
	[15:8]	RW	keyg_max		Color key G 分量最大值。																											
	[7:0]	RW	keyb_max		Color key B 分量最大值。																											

G3CKEYMIN

G3CKEYMIN 为 G3 通道 color key 最小值寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0718		G3CKEYMIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	val		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	val	Alpha1 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。					
[23:16]	RW	keyr_min	Color key R 分量最小值。					
[15:8]	RW	keyg_min	Color key G 分量最小值。					
[7:0]	RW	keyb_min	Color key B 分量最小值。					

G3IRESO

G3IRESO 为 G3 通道输入分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x071C		G3IRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	图形层输入图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	图形层输入图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。					



G3ORESO

G3ORESO 为 G3 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0720		G3ORESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	oh			reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	oh	图形层输出图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，图形层的实际高度必须是偶数；逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	ow	图形层输出图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：图形层的实际宽度必须是偶数。						

G3SFPOS

G3SFPOS 为 G3 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0724		G3SFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	src_yfpos			reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。						
[15:12]	RW	reserved	保留。						



[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。
--------	----	-----------	---------------------------

G3DFPOS

G3DFPOS 为 G3 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x0728				G3DFPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				yfpos								reserved				xfpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	yfpos		列起始坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	xfpos		行起始坐标。																											

G3DLPOS

G3DLPOS 为 G3 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x072C				G3DLPOS								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				ylpos								reserved				xlpos															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											
	[27:16]	RW	ylpos		列结束坐标。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	xlpos		行结束坐标。																											



HCCTRL

HCCTRL 为 HC 通道控制寄存器，为非即时寄存器。该寄存器可以配置硬件鼠标层的相关信息。

Offset Address		Register Name		Total Reset Value																												
0x0800		HCCTRL		0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	hc_en	csc_en	csc_mo		reserved												bitext	ifmt														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31]	RW		hc_en		Surface 使能。 0: 禁止; 1: 使能。																											
[30]	RW		csc_en		颜色空间转换使能。 0: 禁止; 1: 使能。																											
[29]	RW		csc_mode		颜色空间转换标准。 0: BT.601; 1: BT.709。																											
[28:10]	-		reserved		保留。																											
[9:8]	RW		bitext		Surface 输入位图 bit 位扩展模式。 0X: 低位扩展 0; 10: 低位扩展最高 bit 位; 11: 低位扩展最高若干 bit 位。																											
[7:0]	RW		ifmt		输入数据格式。 0x49: aRGB1555; 其他: 保留。																											

HCUPD

HCUPD 为 HC 通道更新使能寄存器。



Offset Address		Register Name		Total Reset Value					
0x0804		HCUPD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								regup
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RW	regup	Surface 的寄存器更新。 当软件配置完该层寄存器后写 1 更新，更新完成后，硬件自动清零。						

HCADDR

HCADDR 为 HC 通道帧地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x0808		HCADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	hcaddr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	hcaddr	Surface 帧 buffer 地址。					

HCSTRIDE

HCSTRIDE 为 HC 通道帧偏移地址寄存器。



Offset Address		Register Name		Total Reset Value					
0x080C		HCSTRIDE		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				hcstride				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	hcstride	帧 buffer 的行偏移。						

HCCBMPARA

HCCBMPARA 为 HC 通道叠加相关参数寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value							
0x0810		HCCBMPARA		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved				key_mode	key_en	reserved	palpha_en	reserved	palpha_range	galpha
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description								
[31:16]	-	reserved	保留。								
[15]	RW	key_mode	Color key 模式。 0: 当 $Keymin \leq Pixel \leq Keymax$ 时，处理为关键色； 1: 当 $Pixel \leq Keymin$ 或者 $Pixel \geq Keymax$ 时，处理为关键色。								
[14]	RW	key_en	Color key 使能。 0: 禁止； 1: 使能。								
[13]	-	reserved	保留。								
[12]	RW	palpha_en	像素 alpha 使能。 0: 禁止； 1: 使能。								



[11:9]	-	reserved	保留。
[8]	RW	palpha_range	像素 alpha 的范围。 0: 0~128; 1: 0~255。
[7:0]	RW	galpha	叠加全局 alpha 值。 取值范围为 0~127, 0 表示全透明, 127 表示完全不透明。

HCCKEYMAX

HCCKEYMAX 为 HC 通道 color key 最大值寄存器，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0814				HCCKEYMAX				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0				keyr_max				keyg_max				keyb_max																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	[31:24]				[23:16]				[15:8]				[7:0]																			
Access	RW				RW				RW				RW																			
Name	va0				keyr_max				keyg_max				keyb_max																			
Description	Alpha0 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。				Color key R 分量最大值。				Color key G 分量最大值。				Color key B 分量最大值。																			

HCCKEYMIN

HCCKEYMIN 为 HC 通道 color key 最小值寄存器，为非即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0818		HCKEYMIN		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	val		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RW	val	Alpha1 值。 当数据格式为 alpha RGB1555 并且数据最高 bit 位值为 0 时，alpha 值用该值替换。					
[23:16]	RW	keyr_min	Color key R 分量最小值。					
[15:8]	RW	keyg_min	Color key G 分量最小值。					
[7:0]	RW	keyb_min	Color key B 分量最小值。					

HCIRESO

HCIRESO 为 HC 通道输入分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x081C		HCIRESO		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	ih		reserved	iw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					
[27:16]	RW	ih	硬件鼠标层输入图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，硬件鼠标层的实际高度必须是偶数；逐行输出无此限制。					
[15:12]	-	reserved	保留。					
[11:0]	RW	iw	硬件鼠标层输入图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：硬件鼠标层的实际宽度必须是偶数。					



HSCORESO

HSCORESO 为 HC 通道输出分辨率寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0820		HSCORESO		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	oh			reserved	ow			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	oh	硬件鼠标层输出图像高度，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，硬件鼠标层的实际高度必须是偶数；逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	ow	硬件鼠标层输出图像宽度，以像素为单位。 配置值为实际宽度减 1。 注意：硬件鼠标层的实际宽度必须是偶数。						

HCSFPOS

HCSFPOS 为 HC 通道源位图的起始位置寄存器，为非即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0824		HCSFPOS		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	src_yfpos			reserved	src_xfpos			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:16]	RW	src_yfpos	源起始 Y 坐标。 0 为第一行，隔行输出时，配置值必须是偶数。						
[15:12]	-	reserved	保留。						



[11:0]	RW	src_xfpos	源起始 X 坐标。 0 为第一行第一个像素。
--------	----	-----------	---------------------------

HCFPOS

HCFPOS 为 HC 通道显示窗口的起始位置寄存器，以像素为单位，非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0828				HCFPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				disp_yfpos				reserved				disp_xfpos																			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0															
Bits	Access			Name			Description																									
[31:28]	-			reserved			保留。																									
[27:16]	RW			disp_yfpos			列起始坐标。																									
[15:12]	-			reserved			保留。																									
[11:0]	RW			disp_xfpos			行起始坐标。																									

HCDLPOS

HCDLPOS 为 HC 通道显示窗口的结束位置寄存器，以像素为单位，为非即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x082C				HCDLPOS				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				disp_ylpos				reserved				disp_xlpos																			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0																			
Bits	Access			Name			Description																									
[31:28]	-			reserved			保留。																									
[27:16]	RW			disp_ylpos			列结束坐标。																									
[15:12]	-			reserved			保留。																									
[11:0]	RW			disp_xlpos			行结束坐标。																									



CBMBKG1

CBMBKG1 为 Mixer1 叠加背景色寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0B00		CBMBKG1		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		cbm_bkgcr1	cbm_bkgcb1		cbm_bkgy1		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	-	reserved	保留。					
[23:16]	RW	cbm_bkgcr1	Mixer1 叠加背景色，Y 分量。					
[15:8]	RW	cbm_bkgcb1	Mixer1 叠加背景色，Cb 分量。					
[7:0]	RW	cbm_bkgy1	Mixer1 叠加背景色，Cr 分量。					

CBMBKG2

CBMBKG2 为 Mixer2 叠加背景色寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0B04		CBMBKG2		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		cbm_bkgcr2	cbm_bkgcb2		cbm_bkgy2		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	-	reserved	保留。					
[23:16]	RW	cbm_bkgcr2	Mixer2 叠加背景色，Y 分量。					
[15:8]	RW	cbm_bkgcb2	Mixer2 叠加背景色，Cb 分量。					
[7:0]	RW	cbm_bkgy2	Mixer2 叠加背景色，Cr 分量。					

CBCFG

CBCFG 为 Colorbar 配置寄存器，为即时寄存器。



		Offset Address 0x0B0C								Register Name CBCFG								Total Reset Value 0x0000_0000																			
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name		reserved								reserved								mixer3_prio		mixer_prio4		mixer_prio3		mixer_prio2		mixer_prio1		mixer_prio0									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																																	
[31:30]	-	reserved		保留。																																	
[29]	RW	sur_attr5		GDC_HC 链接。 0: Mixer1; 1: Mixer2。																																	
[28]	RW	sur_attr4		GDC_G2 链接。 0: Mixer1; 1: Mixer2。																																	
[27]	RW	sur_attr3		GDC_G1 链接。 0: Mixer1; 1: Mixer2。																																	
[26]	RW	sur_attr2		GDC_G0 链接。 0: Mixer1; 1: 保留。																																	
[25]	RW	sur_attr1		VDC_AD 链接。 0: Mixer1; 1: Mixer2。																																	
[24]	RW	sur_attr0		VDC_HD 链接。 0: Mixer1; 1: 保留。																																	
[23:16]	-	reserved		保留。																																	
[15]	RW	mixer3_prio		Mixer3 叠加层优先级。 0: G3 优先级高于 VSD; 1: VSD 优先级高于 G3。																																	



[14:12]	RW	mixer_prio4	<p>Mixer1 和 Mixer2 的叠加层优先级配置，表示优先级 4（优先级最高）的驱动层。</p> <p>000: 没有层驱动；</p> <p>001: vdc_hd；</p> <p>010: vdc_ad；</p> <p>011: gdc_g0；</p> <p>100: gdc_g1；</p> <p>101: gdc_g2；</p> <p>其他: 保留。</p>
[11:9]	RW	mixer_prio3	<p>Mixer1 和 Mixer2 的叠加层优先级配置，表示优先级 3 的驱动层。</p> <p>000: 没有层驱动；</p> <p>001: vdc_hd；</p> <p>010: vdc_ad；</p> <p>011: gdc_g0；</p> <p>100: gdc_g1；</p> <p>101: gdc_g2；</p> <p>其他: 保留。</p>
[8:6]	RW	mixer_prio2	<p>Mixer1 和 Mixer2 的叠加层优先级配置，表示优先级 2 的驱动层。</p> <p>000: 没有层驱动；</p> <p>001: vdc_hd；</p> <p>010: vdc_ad；</p> <p>011: gdc_g0；</p> <p>100: gdc_g1；</p> <p>101: gdc_g2；</p> <p>其他: 保留。</p>
[5:3]	RW	mixer_prio1	<p>Mixer1 和 Mixer2 的叠加层优先级配置，表示优先级 1 的驱动层。</p> <p>000: 没有层驱动；</p> <p>001: vdc_hd；</p> <p>010: vdc_ad；</p> <p>011: gdc_g0；</p> <p>100: gdc_g1；</p> <p>101: gdc_g2；</p> <p>其他: 保留。</p>



[2:0]	RW	mixer_prio0	<p>Mixer1 和 Mixer2 的叠加层优先级配置，表示优先级 0（优先级最低）的驱动层。</p> <p>000：没有层驱动；</p> <p>001：vdc_hd；</p> <p>010：vdc_ad；</p> <p>011：gdc_g0；</p> <p>100：gdc_g1；</p> <p>101：gdc_g2；</p> <p>其他：保留。</p>
-------	----	-------------	---

DHDCTRL

DHDCTRL 为 HD 显示通道控制寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C00		DHDCTRL		0x0000_00EC					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	intf_en slave_mode cbar_sel	reserved			clipen cscen gmmen gmmmod	reserved	idy ibs ivs	iop synm intfb	intfdm
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0	1 1 0 0	
Bits	Access	Name	Description						
[31]	RW	intf_en	<p>显示接口使能。</p> <p>0：禁止；</p> <p>1：使能。</p> <p>注意：使能时，接口才会输出。</p>						
[30]	RW	slave_mode	<p>显示接口从模式使能。</p> <p>0：禁止（主模式）；</p> <p>1：使能（此时由输入的 BT.1120 Capture 时序驱动）。</p>						
[29:28]	RW	cbar_sel	<p>Colorbar 色域空间控制。</p> <p>00、01：RGB 空间；</p> <p>10：YUV 空间；</p> <p>11：RGB 空间。</p>						
[27:17]	-	reserved	保留。						



[16]	RW	clipen	输出嵌位使能。 0: 禁止; 1: 使能。
[15]	RW	cscen	颜色空间转换使能。 0: 禁止; 1: 使能。
[14]	RW	gmmen	输出 Gamma 校正使能。 0: 禁止; 1: 使能。
[13]	RW	gmmmode	输出 Gamma 校正模式。 0: Gamma 表由硬件产生; 1: Gamma 表由软件配置。
[12:11]	-	reserved	保留。
[10]	RW	idv	数据有效信号输出反相使能。 0: 禁止; 1: 使能。
[9]	RW	ihs	水平同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[8]	RW	ivs	垂直同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[7]	RW	iop	逐行或者隔行显示。 0: 隔行显示; 1: 逐行显示。
[6]	RW	synm	同步模式。 0: 时序标签模式 (例如 BT.656); 1: 同步信号模式 (例如 LCD 显示)。
[5:4]	RW	intfb	输出接口位宽模式。 00: 单分量模式 (每个时钟输出 1 个分量); 01: 2 分量模式 (每个时钟输出 2 个分量); 10: 3 分量模式 (每个时钟输出 3 个分量); 11: 保留。



[3:0]	RW	intfdm	接口数据格式。 0x0: YCbCr4:2:2 数据格式; 0xC: RGB888/YCbCr444 输出; 其他: 保留。
-------	----	--------	---

DHDVSYNC

DHDVSYNC 为 HD 显示通道垂直同步时序寄存器，为即时寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x0C04				DHDVSYNC				0x0010_A257																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				vfb				vbb				vact																			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	1	1	1
Bits	Access	Name	Description																													
[31:28]	-	reserved	保留。																													
[27:20]	RW	vfb	隔行输出时，表示顶场垂直消隐前肩； 逐行输出时，表示垂直消隐前肩。																													
[19:12]	RW	vbb	隔行输出时，表示顶场垂直消隐后肩； 逐行输出时，表示垂直消隐后肩加垂直脉冲的宽度。																													
[11:0]	RW	vact	隔行输出时，表示顶场的活动图象的高度； 逐行输出时，表示一帧的活动图象的高度。 配置值为实际值减 1。																													

DHDHSYNC1

DHDHSYNC1 为 HD 显示通道水平同步时序寄存器 1，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0C08		DHDHSYNC1		0x00D7_031F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	hbb				hact			
Reset	0 0 0 0	0 0 0 0	1 1 0 1	0 1 1 1	0 0 0 0	0 0 1 1	0 0 0 1	1 1 1 1
Bits	Access	Name	Description					
[31:16]	RW	hbb	水平消隐后肩，以像素为单位。					
[15:0]	RW	hact	活动区域水平像素数。					

DHDHSYNC2

DHDHSYNC2 为 HD 显示通道水平同步时序寄存器 2，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0C0C		DHDHSYNC2		0x0000_0027				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved				hfb			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 1
Bits	Access	Name	Description					
[31:16]	-	reserved	保留。					
[15:0]	RW	hfb	水平消隐前肩，以像素为单位。					

DHDVPLUS

DHDVPLUS 为 HD 显示通道垂直同步补充时序寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0C10		DHDVPLUS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	bvfb		bvbb		bvact		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					



[27:20]	RW	bvfb	隔行输出时，表示底场垂直消隐前肩。
[19:12]	RW	bvbb	隔行输出时，表示底场垂直消隐后肩垂直脉冲的宽度。
[11:0]	RW	bvact	隔行输出时，表示底场的活动图象的高度。 配置值为实际值减 1。

DHDPWR

DHDPWR 为 HD 显示通道同步脉冲宽度寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C14		DHDPWR		0x0003_007F					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			vpw		hpw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 1 1 1	1 1 1 1	
Bits	Access	Name	Description						
[31:24]	-	reserved	保留。						
[23:16]	RW	vpw	配置值为垂直脉冲宽度减 1。						
[15:0]	RW	hpw	配置值为水平脉冲宽度减 1。						

DHDFIFOTHD

DHDFIFOTHD 为 HD 显示通道异步 FIFO 门限配置寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C18		DHDFIFOTHD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved					aalmthd			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:12]	-	reserved	保留。						
[11:0]	RW	aalmthd	数据告警门限。 当异步 FIFO 中数据少于该值时，给出告警信息。						



DHDVTTHD

DHDVTTHD 为 HD 显示通道垂直时序门限寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C1C		DHDVTTHD		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				thd_mode	reserved	vtmgthd		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:17]	-	reserved	保留。						
[16]	RW	thd_mode	门限中断产生的模式。 0: 帧模式，以帧为单位做门限计数； 1: 场模式，在隔行显示时以场为单位做门限计数。						
[15:13]	-	reserved	保留。						
[12:0]	RW	vtmgthd	垂直时序门限值，当垂直时序计数器到达该门限时触发 HD 通道垂直时序中断。						

DHDCSCIDC

DHDCSCIDC 为 HD 显示通道色彩空间转换输入直流分量寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C20		DHDCSCIDC		0x07C3_0180					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscidc2		cscidc1		cscidc0			
Reset	0 0 0 0	0 1 1 1	1 1 0 0	0 0 1 1	0 0 0 0	0 0 0 1	1 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:27]	-	reserved	保留。						
[26:18]	RW	cscidc2	输入分量 2 直流参数，MSB 为符号位，补码表示。						
[17:9]	RW	cscidc1	输入分量 1 直流参数，MSB 为符号位，补码表示。						
[8:0]	RW	cscidc0	输入分量 0 直流参数，MSB 为符号位，补码表示。						



DHDCSCODC

DHDCSCODC 为 HD 显示通道色彩空间转换输出直流分量寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C24		DHDCSCODC		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved		cscodc2		cscodc1		cscodc0		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:27]	-	reserved	保留。						
[26:18]	RW	cscodc2	输出分量 2 直流参数，MSB 为符号位，补码表示。						
[17:9]	RW	cscodc1	输出分量 1 直流参数，MSB 为符号位，补码表示。						
[8:0]	RW	cscodc0	输出分量 0 直流参数，MSB 为符号位，补码表示。						

DHDCSCP0

DHDCSCP0 为 HD 显示通道色彩空间转换参数 0 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0C28		DHDCSCP0		0x0000_012A				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	cscp01		reserved	cscp00			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0
Bits	Access	Name	Description					
[31:29]	-	reserved	保留。					
[28:16]	RW	cscp01	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。					
[15:13]	-	reserved	保留。					
[12:0]	RW	cscp00	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。					



DHDCSCP1

DHDCSCP1 为 HD 显示通道色彩空间转换参数 1 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value																													
0x0C2C		DHDCSCP1		0x012A_01CB																													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved				cscp10								reserved				cscp02																
Reset	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1
Bits	Access	Name	Description																														
[31:29]	-	reserved	保留。																														
[28:16]	RW	cscp10	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。																														
[15:13]	-	reserved	保留。																														
[12:0]	RW	cscp02	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。																														

DHDCSCP2

DHDCSCP2 为 HD 显示通道色彩空间转换参数 2 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value																														
0x0C30		DHDCSCP2		0x1F77_1FC9																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved				cscp12								reserved				cscp11																	
Reset	0	0	0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0	1
Bits	Access	Name	Description																															
[31:29]	-	reserved	保留。																															
[28:16]	RW	cscp12	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。																															
[15:13]	-	reserved	保留。																															
[12:0]	RW	cscp11	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。																															



DHDCSCP3

DHDCSCP3 为 HD 显示通道色彩空间转换参数 3 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C34		DHDCSCP3		0x021D_012A					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp21			reserved	cscp20			
Reset	0 0 0 0	0 0 1 0	0 0 0 1	1 1 0 1	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp21	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp20	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						

DHDCSCP4

DHDCSCP4 为 HD 显示通道色彩空间转换参数 4 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C38		DHDCSCP4		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved					cscp22			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:13]	-	reserved	保留。						
[12:0]	RW	cscp22	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						



DHDCLIPL

DHDCLIPL 为 HD 显示通道嵌位处理最低门限值寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C3C		DHDCLIPL		0x0010_1010					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	clipcl2		clipcl1		clipcl0			
Reset	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 0 0	
Bits	Access	Name	Description						
[31:30]	-	reserved	保留。						
[29:20]	RW	clipcl2	分量 2 最低门限值 Y/R，无符号整数。						
[19:10]	RW	clipcl1	分量 1 最低门限值 Cb/G，无符号整数。						
[9:0]	RW	clipcl0	分量 0 最低门限值 Cr/B，无符号整数。						

DHDCLIPH

DHDCLIPH 为 HD 显示通道嵌位处理最高门限值寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C40		DHDCLIPH		0x00EB_F0F0					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	clipch2		clipch1		clipch0			
Reset	0 0 0 0	0 0 0 0	1 1 1 0	1 0 1 1	1 1 1 1	0 0 0 0	1 1 1 1	0 0 0 0	
Bits	Access	Name	Description						
[31:30]	-	reserved	保留。						
[29:20]	RW	clipch2	分量 2 最高门限值 Y/R，无符号整数。						
[19:10]	RW	clipch1	分量 1 最高门限值 Cb/G，无符号整数。						
[9:0]	RW	clipch0	分量 0 最高门限值 Cr/B，无符号整数。						



DHDGMMTHD1

DHDGMMTHD1 为 HD 显示通道 Gamma 运算门限 1 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C44		DHDGMMTHD1		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	thd_med_low		thd_high		thd_low			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:30]	-	reserved	保留。						
[29:20]	RW	thd_med_low	亮度等级 med_low 门限值。						
[19:10]	RW	thd_high	亮度等级 high 门限值。						
[9:0]	RW	thd_low	亮度等级 low 门限值。						

DHDGMMTHD2

DHDGMMTHD2 为 HD 显示通道 Gamma 运算门限 2 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0C48		DHDGMMTHD2		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			gmm_multiple		thd_med_high			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	reserved	保留。						
[17:10]	RW	gmm_multiple	Gamma 运算乘数。						
[9:0]	RW	thd_med_high	亮度等级 med_high 门限值。						

DHDGMMLOWt

DHDGMMLOWt 为 HD 显示通道 Gamma 查找表（亮度等级为 low）寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0C50+t1×0x4		DHDGMMLOWt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datat						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datat	Gamma 查找表数据 t1。					

DHDGMMMEdt

DHDGMMMEdt 为 HD 显示通道 Gamma 查找表（亮度等级为 med）寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0C60+t2×0x4		DHDGMMMEdt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datat						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datat	Gamma 查找表数据 t2。					

DHDGMMHIGht

DHDGMMHIGht 为 HD 显示通道 Gamma 查找表（亮度等级为 high）寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0C70+t3×0x4		DHDGMMHIGHt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datat						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datat	Gamma 查找表数据 t3。					

DHDGMMMLt

DHDGMMMLt 为 HD 显示通道 Gamma 查找表（亮度等级为 middle_low）寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0C80+t4×0x4		DHDGMMMLt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datat						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datat	Gamma 查找表数据 t4。					

DHDGMMMHT

DHDGMMMHT 为 HD 显示通道 Gamma 查找表（亮度等级为 middle_high）寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0CA0+t5×0x4		DHDGMMMht		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datat						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datat	Gamma 查找表数据 t5。					

DHDGMM3LOW

DHDGMM3LOW 为 HD 显示通道 Gamma 三区域的低门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value				
0x0CB0		DHDGMM3LOW		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved			cnt3_low				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:21]	-	reserved	保留。					
[20:0]	RO	cnt3_low	三区域的低门限亮度统计值。					

DHDGMM3MED

DHDGMM3MED 为 HD 显示通道 Gamma 三区域的中间门限亮度统计值寄存器。



Offset Address		Register Name		Total Reset Value					
0x0CB4		DHDGMM3MED		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt3_med					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt3_med	三区域的中间门限亮度统计值。						

DHDGMM3HIGH

DHDGMM3HIGH 为 HD 显示通道 Gamma 三区域的高门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0CB8		DHDGMM3HIGH		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt3_high					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt3_high	三区域的高门限亮度统计值。						

DHDGMM8MLOW

DHDGMM8MLOW 为 HD 显示通道 Gamma 中间区域分成 8 段后，低门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0CC0		DHDGMM8MLOW		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt8_med_low					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						



[20:0]	RO	cnt8_med_low	中间区域的低门限亮度统计值。
--------	----	--------------	----------------

DHDGMM8MHIGH

DHDGMM8MHIGH 为 HD 显示通道 Gamma 中间区域分成 8 段后，高门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0CC4		DHDGMM8MHIGH		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				cnt8_med_high				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt8_med_high	中间区域的高门限亮度统计值。						

CCDIMGMOD

CCDIMGMOD 为级联多画面模式配置寄存器。

Offset Address		Register Name		Total Reset Value					
0x0CE0		CCDIMGMOD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved						ccd_en	img_right	img_mode
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:9]	-	reserved	保留。						
[8]	RW	ccd_en	级联模式使能。 0: 使能; 1: 禁止。						
[7]	RW	img_right	图像错误标识。 0: 错误; 1: 正确。						



[6:0]	RW	img_mode	多画面分布指示标识，用于区分多画面的模式。软件可以自定义多画面的情况与寄存器值的匹配。 例如： 0x00：标清 1 画面； 0x01：标清 4 画面； 0x02：标清 6 画面； 其他：xxxx（由软件自己定义）。
-------	----	----------	--

DHDSTATE

DHDSTATE 为 HD 显示通道状态寄存器。

Offset Address		Register Name		Total Reset Value						
0x0CF0		DHDSTATE		0x0000_0006						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved							bottom_field	vblank	vback_blank
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0		
Bits	Access	Name	Description							
[31:3]	-	reserved	保留。							
[2]	RO	bottom_field	DHD 显示顶底场标识。 0：顶场； 1：底场。							
[1]	RO	vblank	DHD 显示消隐区标识。 0：有效区； 1：消隐区。							
[0]	RO	vback_blank	DHD 显示后消隐标识。 0：不是后消隐区； 1：是后消隐区。							

DADCTRL

DADCTRL 为 AD 显示通道控制寄存器，为即时寄存器。



		Offset Address 0x0D00								Register Name DADCTRL								Total Reset Value 0x0000_0000																			
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name		intf_en	reserved	cbar_sel		reserved								clipen	cscen	gmmen	gmmmode	reserved	idv	ihs	ivs	iop	synm	intfb	intfdm												
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description																																	
[31]	RW	intf_en		显示接口使能。 0: 禁止; 1: 使能。 使能时, 接口才会输出。																																	
[30]	-	reserved		保留。																																	
[29:28]	RW	cbar_sel		Colorbar 色域空间控制。 00、01: YUV 空间; 10: YUV 空间; 11: RGB 空间。																																	
[27:17]	-	reserved		保留。																																	
[16]	RW	clipen		输出嵌位使能。 0: 禁止; 1: 使能。																																	
[15]	RW	cscen		颜色空间转换使能。 0: 禁止; 1: 使能。																																	
[14]	RW	gmmen		输出 Gamma 校正使能。 0: 禁止; 1: 使能。																																	
[13]	RW	gmmmode		输出 Gamma 校正模式。 0: Gamma 表由硬件产生; 1: Gamma 表由软件配置。																																	
[12:11]	-	reserved		保留。																																	
[10]	RW	idv		数据有效信号输出反相使能。 0: 禁止; 1: 使能。																																	



[9]	RW	ihs	水平同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[8]	RW	ivs	垂直同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[7]	RW	iop	逐行或者隔行显示。 0: 隔行显示; 1: 逐行显示。
[6]	RW	synm	同步模式。 0: 时序标签模式 (例如 BT.656); 1: 同步信号模式 (例如 LCD 显示)。
[5:4]	RW	intfb	输出接口位宽模式。 00: 单分量模式 (每个时钟输出 1 个分量); 01: 2 分量模式 (每个时钟输出 2 个分量); 10: 3 分量模式 (每个时钟输出 3 个分量); 11: 保留。
[3:0]	RW	intfdm	接口数据格式。 0x0: YCbCr4:2:2 数据格式; 0xC: RGB888/YCbCr444 数据格式; 其他: 保留。

DADVSYNC

DADVSYNC 为 AD 显示通道垂直同步时序寄存器，为即时寄存器。

	Offset Address 0x0D04								Register Name DADVSYNC								Total Reset Value 0x0011_511F															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				vfb				vbb				vact																			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	1
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:20]	RW		vfb		隔行输出时，表示顶场垂直消隐前肩； 逐行输出时，表示垂直消隐前肩。																											



[19:12]	RW	vbb	隔行输出时，表示顶场垂直消隐后肩； 逐行输出时，表示垂直消隐后肩加垂直脉冲的宽度。
[11:0]	RW	vact	隔行输出时，表示顶场的活动图象的高度； 逐行输出时，表示一帧的活动图象的高度。 配置值为实际值减 1。

DADHSYNC1

DADHSYNC1 为 AD 显示通道水平同步时序寄存器 1，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D08		DADHSYNC1		0x0107_02CF					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	hbb				hact				
Reset	0 0 0 0	0 0 0 1	0 0 0 0	0 1 1 1	0 0 0 0	0 0 1 0	1 1 0 0	1 1 1 1	
Bits	Access	Name	Description						
[31:16]	RW	hbb	水平消隐后肩，以像素为单位。						
[15:0]	RW	hact	活动区域水平像素数。						

DADHSYNC2

DADHSYNC2 为 AD 显示通道水平同步时序寄存器 2，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D0C		DADHSYNC2		0x0000_0017					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				hfb				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 1	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RW	hfb	水平消隐前肩，以像素为单位。						



DADVPLUS

DADVPLUS 为 AD 显示通道垂直同步补充时序寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D10		DADVPLUS		0x0011_611F					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	bvfb		bvbb		bvact			
Reset	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 1	0 1 1 0	0 0 0 1	0 0 0 1	1 1 1 1	
Bits	Access	Name	Description						
[31:28]	-	reserved	保留。						
[27:20]	RW	bvfb	隔行输出时，表示底场垂直消隐前肩。						
[19:12]	RW	bvbb	隔行输出时，表示底场垂直消隐后肩垂直脉冲的宽度。						
[11:0]	RW	bvact	隔行输出时，表示底场的活动图象的高度。 配置值为实际值减 1。						

DADPWR

DADPWR 为 AD 显示通道同步脉冲宽度寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D14		DADPWR		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved		vpw		hpw				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:24]	-	reserved	保留。						
[23:16]	RW	vpw	配置值为垂直脉冲宽度减 1。						
[15:0]	RW	hpw	配置值为水平脉冲宽度减 1。						

DADFIFOTHD

DADFIFOTHD 为 AD 显示通道异步 FIFO 门限配置寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0D18		DADFIFOTHD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved					aalmthd			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:12]	-	reserved	保留。						
[11:0]	RW	aalmthd	数据告警门限。 当异步 FIFO 中数据小于该值时，给出告警信息。						

DADVTTHD

DADVTTHD 为 AD 显示通道垂直时序门限寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0D1C		DADVTTHD		0x0000_0001				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved			thd_mode	reserved	vtmgthd		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
Bits	Access	Name	Description					
[31:17]	-	reserved	保留。					
[16]	RW	thd_mode	门限中断产生的模式。 0：帧模式，以帧为单位做门限计数； 1：场模式，在隔行显示时以场为单位做门限计数。					
[15:13]	-	reserved	保留。					
[12:0]	RW	vtmgthd	垂直时序门限值，当垂直时序计数器到达该门限时触发 AD 通道垂直时序中断。					

DADCSCIDC

DADCSCIDC 为 AD 显示通道色彩空间转换输入直流分量寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0D20		DADCSCIDC		0x07C3_0180					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscidc2		cscidc1		cscidc0			
Reset	0 0 0 0	0 1 1 1	1 1 0 0	0 0 1 1	0 0 0 0	0 0 0 1	1 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:27]	-	reserved	保留。						
[26:18]	RW	cscidc2	输入分量 2 直流参数，MSB 为符号位，补码表示。						
[17:9]	RW	cscidc1	输入分量 1 直流参数，MSB 为符号位，补码表示。						
[8:0]	RW	cscidc0	输入分量 0 直流参数，MSB 为符号位，补码表示。						

DADCSCODC

DADCSCODC 为 AD 显示通道色彩空间转换输出直流分量寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D24		DADCSCODC		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscodc2		cscodc1		cscodc0			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:27]	-	reserved	保留。						
[26:18]	RW	cscodc2	输出分量 2 直流参数，MSB 为符号位，补码表示。						
[17:9]	RW	cscodc1	输出分量 1 直流参数，MSB 为符号位，补码表示。						
[8:0]	RW	cscodc0	输出分量 0 直流参数，MSB 为符号位，补码表示。						

DADCSCP0

DADCSCP0 为 AD 显示通道色彩空间转换参数 0 寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0D28		DADCSCP0		0x0000_012A					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp01			reserved	cscp00			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp01	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp00	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						

DADCSCP1

DADCSCP1 为 AD 显示通道色彩空间转换参数 1 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0D2C		DADCSCP1		0x012A_01CB					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp10			reserved	cscp02			
Reset	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0	0 0 0 0	0 0 0 1	1 1 0 0	1 0 1 1	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp10	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp02	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						



DADCSCP2

DADCSCP2 为 AD 显示通道色彩空间转换参数 2 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value						
0x0D30		DADCSCP2		0x1F77_1FC9						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved	cscp12				reserved	cscp11			
Reset	0 0 0 1	1 1 1 1	0 1 1 1	0 1 1 1	0 0 0 1	1 1 1 1	1 1 0 0	1 0 0 1		
Bits	Access	Name	Description							
[31:29]	-	reserved	保留。							
[28:16]	RW	cscp12	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。							
[15:13]	-	reserved	保留。							
[12:0]	RW	cscp11	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。							

DADCSCP3

DADCSCP3 为 AD 显示通道色彩空间转换参数 3 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value						
0x0D34		DADCSCP3		0x021D_012A						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved	cscp21				reserved	cscp20			
Reset	0 0 0 0	0 0 1 0	0 0 0 1	1 1 0 1	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0		
Bits	Access	Name	Description							
[31:29]	-	reserved	保留。							
[28:16]	RW	cscp21	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。							
[15:13]	-	reserved	保留。							
[12:0]	RW	cscp20	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。							



DADCSCP4

DADCSCP4 为 AD 显示通道色彩空间转换参数 4 寄存器，为即时寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x0D38								DADCSCP4								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																cscp22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:13]	-		reserved		保留。																											
[12:0]	RW		cscp22		5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。																											

DADCLIPL

DADCLIPL 为 AD 显示通道嵌位处理最低门限值寄存器，为即时寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x0D3C								DADCLIPL								0x0010_1010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	clipcl2								clipcl1								clipcl0														
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
Bits	Access		Name		Description																											
[31:30]	-		reserved		保留。																											
[29:20]	RW		clipcl2		分量 2 最低门限值 Y/R，无符号整数。																											
[19:10]	RW		clipcl1		分量 1 最低门限值 Cb/G，无符号整数。																											
[9:0]	RW		clipcl0		分量 0 最低门限值 Cr/B，无符号整数。																											

DADCLIPH

DADCLIPH 为 AD 显示通道嵌位处理最高门限值寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0D40		DADCLIPH		0x00EB_F0F0				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	clipch2		clipch1		clipch0		
Reset	0 0 0 0	0 0 0 0	1 1 1 0	1 0 1 1	1 1 1 1	0 0 0 0	1 1 1 1	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:20]	RW	clipch2	分量 2 最高门限值 Y/R, 无符号整数。					
[19:10]	RW	clipch1	分量 1 最高门限值 Cb/G, 无符号整数。					
[9:0]	RW	clipch0	分量 0 最高门限值 Cr/B, 无符号整数。					

DADGMMTHD1

DADGMMTHD1 为 AD 显示通道 Gamma 运算门限 1 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0D44		DADGMMTHD1		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	thd_med_low		thd_high		thd_low		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:20]	RW	thd_med_low	亮度等级 med_low 门限值。					
[19:10]	RW	thd_high	亮度等级 high 门限值。					
[9:0]	RW	thd_low	亮度等级 low 门限值。					

DADGMMTHD2

DADGMMTHD2 为 AD 显示通道 Gamma 运算门限 2 寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value					
0x0D48		DADGMMTHD2		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			gmm_multiple			thd_med_high		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:18]	-	reserved	保留。						
[17:10]	RW	gmm_multiple	Gamma 运算乘数。						
[9:0]	RW	thd_med_high	亮度等级 med_high 门限值。						

DADGMMLOWt

DADGMMLOWt 为 AD 显示通道 Gamma 查找表，亮度等级为 low 的寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0D50+t1×0x4		DADGMMLOWt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datan						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datan	Gamma 查找表数据 t1。					

DADGMMMEDt

DADGMMMEDt 为 AD 显示通道 Gamma 查找表，亮度等级为 med 的寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0D60+t2×0x4		DADGMMMEDt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datan						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datan	Gamma 查找表数据 t2。					

DADGMMHIGHt

DADGMMHIGHt 为 AD 显示通道 Gamma 查找表，亮度等级为 high 的寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0D70+t3×0x4		DADGMMHIGHt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datan						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datan	Gamma 查找表数据 t3。					

DADGMMMLt

DADGMMMLt 为 AD 显示通道 Gamma 查找表，亮度等级为 middle_low 的寄存器，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0D80+t4×0x4		DADGMMMLt		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datan						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datan	Gamma 查找表数据 t4。					

DADGMMMHT

DADGMMMHT 为 AD 显示通道 Gamma 查找表，亮度等级为 middle_high 的寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0DA0+t5×0x4		DADGMMMHT		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	table_datan						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:0]	RW	table_datan	Gamma 查找表数据 t5。					

DADGMM3LOW

DADGMM3LOW 为 AD 显示通道 Gamma 三区域的低门限亮度统计值寄存器。



Offset Address		Register Name		Total Reset Value					
0x0DB0		DADGMM3LOW		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt3_low					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt3_low	三区域的低门限亮度统计值。						

DADGMM3MED

DADGMM3MED 为 AD 显示通道 Gamma 三区域的中间门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0DB4		DADGMM3MED		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt3_med					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt3_med	三区域的中间门限亮度统计值。						

DADGMM3HIGH

DADGMM3HIGH 为 AD 显示通道 Gamma 三区域的高门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0DB8		DADGMM3HIGH		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt3_high					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						



[20:0]	RO	cnt3_high	三区域的高门限亮度统计值。
--------	----	-----------	---------------

DADGMM8MLOW

DADGMM8MLOW 为 AD 显示通道 GMMA 中间区域分成 8 段后，低门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0DC0		DADGMM8MLOW		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt8_med_low					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt8_med_low	中间区域的低门限亮度统计值。						

DADGMM8MHIGH

DADGMM8MHIGH 为 AD 显示通道 Gamma 中间区域分成 8 段后，高门限亮度统计值寄存器。

Offset Address		Register Name		Total Reset Value					
0x0DC4		DADGMM8MHIGH		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			cnt8_med_high					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:21]	-	reserved	保留。						
[20:0]	RO	cnt8_med_high	中间区域的高门限亮度统计值。						

DADSTATE

DADSTATE 为 AD 显示通道状态寄存器。



Offset Address		Register Name		Total Reset Value						
0x0DF0		DADSTATE		0x0000_0006						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved							bottom_field	vblank	vback_blank
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0		
Bits	Access	Name	Description							
[31:3]	-	reserved	保留。							
[2]	RW	bottom_field	DAD 显示顶底场标识。 0: 顶场; 1: 底场。							
[1]	RW	vblank	DAD 显示消隐区标识。 0: 有效区; 1: 消隐区。							
[0]	RW	vback_blank	DAD 显示后消隐标识。 0: 非后消隐区; 1: 后消隐区。							

DSDCTRL

DSDCTRL 为 SD 显示通道控制寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value						
0x0E00		DSDCTRL		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	intf_en reserved cbar_sel	reserved			clipen cscen	reserved		idv ihs ivs	iop synm intfb	intfdm
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31]	RW	intf_en	显示接口使能。 0: 禁止; 1: 使能。 使能时，接口才会输出。							



[30]	-	reserved	保留。
[29:28]	RW	cbar_sel	Colorbar 色域空间控制。 00、01: YUV 空间; 10: YUV 空间; 11: RGB 空间。
[27:17]	-	reserved	保留。
[16]	RW	clipen	输出嵌位使能。 0: 禁止; 1: 使能。
[15]	RW	cscen	颜色空间转换使能。 0: 禁止; 1: 使能。
[14:11]	-	reserved	保留。
[10]	RW	idv	数据有效信号输出反相使能。 0: 禁止; 1: 使能。
[9]	RW	ihs	水平同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[8]	RW	ivs	垂直同步脉冲输出反相使能。 0: 禁止; 1: 使能。
[7]	RW	iop	逐行或者隔行显示。 0: 隔行显示; 1: 逐行显示。
[6]	RW	synm	同步模式。 0: 时序标签模式 (例如 BT.656); 1: 同步信号模式 (例如 LCD 显示)。
[5:4]	RW	intfb	输出接口位宽模式。 00: 单分量模式 (每个时钟输出 1 个分量); 01: 2 分量模式 (每个时钟输出 2 个分量); 10: 3 分量模式 (每个时钟输出 3 个分量); 11: 保留。



[3:0]	RW	intfdm	接口数据格式。 0x0: YCbCr4:2:2 数据格式; 0xC: RGB888/YCbCr444 输出; 其他: 保留。
-------	----	--------	---

DSDVSYNC

DSDVSYNC 为 SD 显示通道垂直同步时序寄存器，为即时寄存器。

	Offset Address 0x0E04								Register Name DSDVSYNC								Total Reset Value 0x0011_511F															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				vfb				vbb				vact																			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	1
Bits	Access		Name		Description																											
[31:28]	-		reserved		保留。																											
[27:20]	RW		vfb		隔行输出时，表示顶场垂直消隐前肩； 逐行输出时，表示垂直消隐前肩。																											
[19:12]	RW		vbb		隔行输出时，表示顶场垂直消隐后肩； 逐行输出时，表示垂直消隐后肩加垂直脉冲的宽度。																											
[11:0]	RW		vact		隔行输出时，表示顶场的活动图象的高度； 逐行输出时，表示一帧的活动图象的高度。 配置值为实际高度减 1。																											

DSDHSYNC1

DSDHSYNC1 为 SD 显示通道水平同步时序寄存器 1，为即时寄存器。



Offset Address		Register Name		Total Reset Value				
0x0E08		DSDHSYNC1		0x0107_02CF				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	hbb				hact			
Reset	0 0 0 0	0 0 0 1	0 0 0 0	0 1 1 1	0 0 0 0	0 0 1 0	1 1 0 0	1 1 1 1
Bits	Access	Name	Description					
[31:16]	RW	hbb	水平消隐后肩，以像素为单位。					
[15:0]	RW	hact	活动区域水平像素数。					

DSDHSYNC2

DSDHSYNC2 为 SD 显示通道水平同步时序寄存器 2，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E0C		DSDHSYNC2		0x0000_0017				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved				hfb			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 1 1 1
Bits	Access	Name	Description					
[31:16]	-	reserved	保留。					
[15:0]	RW	hfb	水平消隐前肩，以像素为单位。					

DSDVPLUS

DSDVPLUS 为 SD 显示通道垂直同步补充时序寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E10		DSDVPLUS		0x0011_611F				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	bvfb		bvbb		bvact		
Reset	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 1	0 1 1 0	0 0 0 1	0 0 0 1	1 1 1 1
Bits	Access	Name	Description					
[31:28]	-	reserved	保留。					



[27:20]	RW	bvfb	隔行输出时，表示底场垂直消隐前肩。
[19:12]	RW	bvbb	隔行输出时，表示底场垂直消隐后肩垂直脉冲的宽度。
[11:0]	RW	bvact	隔行输出时，表示底场的活动图象的高度。 配置值为实际高度减 1。

DSDPWR

DSDPWR 为 SD 显示通道同步脉冲宽度寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value						
0x0E14		DSDPWR		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved				vpw		hpw			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:24]	-	reserved	保留。							
[23:16]	RW	vpw	配置值为垂直脉冲宽度减 1。							
[15:0]	RW	hpw	配置值为水平脉冲宽度减 1。							

DSDFIFOTHD

DSDFIFOTHD 为 SD 显示通道异步 FIFO 门限配置寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E18		DSDFIFOTHD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved						aalmthd		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:12]	-	reserved	保留。						
[11:0]	RW	aalmthd	数据告警门限。 当异步 FIFO 中数据少于该值时，给出告警信息。						



DSDVTTHD

DSDVTTHD 为 SD 显示通道垂直时序门限寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E1C		DSDVTTHD		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				thd_mode	reserved	vtmgthd		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:17]	-	reserved	保留。						
[16]	RW	thd_mode	门限中断产生的模式。 0: 帧模式，以帧为单位做门限计数； 1: 场模式，在隔行显示时以场为单位做门限计数。						
[15:13]	-	reserved	保留。						
[12:0]	RW	vtmgthd	垂直时序门限值，当垂直时序计数器到达该门限时触发 SD 通道垂直时序中断。						

DSDCSCIDC

DSDCSCIDC 为 SD 显示通道色彩空间转换输入直流分量寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E20		DSDCSCIDC		0x07C3_0180				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	cscidc2		cscidc1		cscidc0		
Reset	0 0 0 0	0 1 1 1	1 1 0 0	0 0 1 1	0 0 0 0	0 0 0 1	1 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:27]	-	reserved	保留。					
[26:18]	RW	cscidc2	输入分量 2 直流参数，MSB 为符号位。补码表示。					
[17:9]	RW	cscidc1	输入分量 1 直流参数，MSB 为符号位。补码表示。					
[8:0]	RW	cscidc0	输入分量 0 直流参数，MSB 为符号位。补码表示。					



DSDCSCODC

DSDCSCODC 为 SD 显示通道色彩空间转换输出直流分量寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E24		DSDCSCODC		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved		cscodc2		cscodc1		cscodc0		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:27]	-	reserved	保留。						
[26:18]	RW	cscodc2	输出分量 2 直流参数，MSB 为符号位。补码表示。						
[17:9]	RW	cscodc1	输出分量 1 直流参数，MSB 为符号位。补码表示。						
[8:0]	RW	cscodc0	输出分量 0 直流参数，MSB 为符号位。补码表示。						

DSDCSCP0

DSDCSCP0 为 SD 显示通道色彩空间转换参数 0 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E28		DSDCSCP0		0x0000_012A				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	cscp01		reserved	cscp00			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0
Bits	Access	Name	Description					
[31:29]	-	reserved	保留。					
[28:16]	RW	cscp01	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。					
[15:13]	-	reserved	保留。					
[12:0]	RW	cscp00	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。					

DSDCSCP1

DSDCSCP1 为 SD 显示通道色彩空间转换参数 1 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E2C		DSDCSCP1		0x012A_01CB					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp10			reserved	cscp02			
Reset	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0	0 0 0 0	0 0 0 1	1 1 0 0	1 0 1 1	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp10	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp02	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						

DSDCSCP2

DSDCSCP2 为 SD 显示通道色彩空间转换参数 2 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E30		DSDCSCP2		0x1F77_1FC9					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp12			reserved	cscp11			
Reset	0 0 0 1	1 1 1 1	0 1 1 1	0 1 1 1	0 0 0 1	1 1 1 1	1 1 0 0	1 0 0 1	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp12	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp11	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						



DSDCSCP3

DSDCSCP3 为 SD 显示通道色彩空间转换参数 3 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E34		DSDCSCP3		0x021D_012A					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	cscp21			reserved	cscp20			
Reset	0 0 0 0	0 0 1 0	0 0 0 1	1 1 0 1	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0	
Bits	Access	Name	Description						
[31:29]	-	reserved	保留。						
[28:16]	RW	cscp21	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						
[15:13]	-	reserved	保留。						
[12:0]	RW	cscp20	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						

DSDCSCP4

DSDCSCP4 为 SD 显示通道色彩空间转换参数 4 寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value					
0x0E38		DSDCSCP4		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved					cscp22			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:13]	-	reserved	保留。						
[12:0]	RW	cscp22	5.8 数据格式：1bit 符号位、4bit 整数位、8bit 小数位。补码表示。						



DSDCLIPL

DSDCLIPL 为 SD 显示通道位处理最低门限值寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E3C		DSDCLIPL		0x0010_1010				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	clipcl2		clipcl1		clipcl0		
Reset	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:20]	RW	clipcl2	分量 2 最低门限值 Y/R，无符号整数。					
[19:10]	RW	clipcl1	分量 1 最低门限值 Cb/G，无符号整数。					
[9:0]	RW	clipcl0	分量 0 最低门限值 Cr/B，无符号整数。					

DSDCLIPH

DSDCLIPH 为 SD 显示通道嵌位处理最高门限值寄存器，为即时寄存器。

Offset Address		Register Name		Total Reset Value				
0x0E40		DSDCLIPH		0x00EB_F0F0				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	clipch2		clipch1		clipch0		
Reset	0 0 0 0	0 0 0 0	1 1 1 0	1 0 1 1	1 1 1 1	0 0 0 0	1 1 1 1	0 0 0 0
Bits	Access	Name	Description					
[31:30]	-	reserved	保留。					
[29:20]	RW	clipch2	分量 2 最高门限值 Y/R，无符号整数。					
[19:10]	RW	clipch1	分量 1 最高门限值 Cb/G，无符号整数。					
[9:0]	RW	clipch0	分量 0 最高门限值 Cr/B，无符号整数。					



DSDSTATE

DSDSTATE 为 SD 显示通道状态寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0EF0	DSDSTATE	0x0000_0006
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0		
Bits	Access	Name	Description
[31:3]	-	reserved	保留。
[2]	RW	bottom_field	DSD 显示顶底场标识。 0: 顶场; 1: 底场。
[1]	RW	vblank	DSD 显示消隐区标识。 0: 有效区; 1: 消隐区。
[0]	RW	vback_blank	DSD 显示后消隐标识。 0: 不是后消隐区; 1: 是后消隐区。

CCDSPOSm

CCDSPOSm 为级联模式子图像 m 的起始位置和子画面有效标识寄存器。



	Offset Address 0x0F00+m×0x4				Register Name CCDSPOSm				Total Reset Value 0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	img_valid		reserved		spos_y								reserved		spos_x																	
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0											
	Bits	Access	Name		Description																											
	[31]	RW	img_valid		子画面 m 有效标识。 0: 无效; 1: 有效。																											
	[30:28]	-	reserved		保留。																											
	[27:16]	RW	spos_y		垂直起始坐标，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，实际坐标必须是偶数；逐行输出无此限制。																											
	[15:12]	-	reserved		保留。																											
	[11:0]	RW	spos_x		水平起始坐标，以像素单位。 配置值为实际坐标减 1。 注意：实际坐标必须是偶数。																											

CCDFPOSm

CCDFPOSm 为级联模式子图像 m 的结束位置寄存器。

	Offset Address 0x0F80+m×0x4				Register Name CCDFPOSm				Total Reset Value 0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				fpos_y								reserved		fpos_x																	
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0											
	Bits	Access	Name		Description																											
	[31:28]	-	reserved		保留。																											



[27:16]	RW	fpos_y	垂直起始坐标，以行为单位。 配置值为实际高度减 1。 注意：对于隔行输出，实际坐标必须是偶数；逐行输出无此限制。
[15:12]	-	reserved	保留。
[11:0]	RW	fpos_x	水平起始坐标，以像素单位。 配置值为实际坐标减 1。 注意：实际坐标必须是偶数。

VHDHLCOEFG

VHDHLCOEFG 为 VHD 通道水平亮度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4、0x8 和 0xC 的 4 个寄存器组成，表示 8 阶系数。

	Offset Address 0x1000~0x111C				Register Name VHDHLCOEFG				Total Reset Value 0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				hlcoefn2				reserved				hlcoefn1																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31:26]	-		reserved		保留。																											
[25:16]	RW		hlcoefn2		水平亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 时，表示第 4 阶系数； 当寄存器地址的 bit[3:0]为 0x8 时，表示第 6 阶系数； 当寄存器地址的 bit[3:0]为 0xC 时，表示第 8 阶系数。																											
[15:10]	-		reserved		保留。																											
[9:0]	RW		hlcoefn1		水平亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 时，表示第 3 阶系数； 当寄存器地址的 bit[3:0]为 0x8 时，表示第 5 阶系数； 当寄存器地址的 bit[3:0]为 0xC 时，表示第 7 阶系数。																											



VHDHCCOEF

VHDHCCOEF 为 VHD 通道水平色度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

Offset Address		Register Name		Total Reset Value					
0x1200~0x128C		VHDHCCOEF		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved		hccoeffn2		reserved		hccoeffn1		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:26]	-	reserved	保留。						
[25:16]	RW	hccoeffn2	水平色度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。						
[15:10]	-	reserved	保留。						
[9:0]	RW	hccoeffn1	水平色度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。						

VHDVLCOEF

VHDVLCOEF 为 VHD 通道垂直亮度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

Offset Address		Register Name		Total Reset Value					
0x1300~0x138C		VHDVLCOEF		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved		vlcoeffn2		reserved		vlcoeffn1		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:26]	-	reserved	保留。						



[25:16]	RW	vlcoefn2	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。
[15:10]	-	reserved	保留。
[9:0]	RW	vlcoefn1	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。

VHDVCCOEF

VHDVCCOEF 为 VHD 通道垂直色度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

Offset Address		Register Name		Total Reset Value				
0x1400~0x148C		VHDVCCOEF		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		vccoefn2		reserved		vccoefn1	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:26]	-	reserved	保留。					
[25:16]	RW	vccoefn2	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。					
[15:10]	-	reserved	保留。					
[9:0]	RW	vccoefn1	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。					



VADHLCOEf

VADHLCOEf 为 VAD 通道水平亮度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4、0x8 和 0xC 的 4 个寄存器组成，表示 8 阶系数。

		Offset Address 0x1500~0x161C				Register Name VADHLCOEf				Total Reset Value 0x0000_0000																							
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved				hlcoefn2				reserved				hlcoefn1																			
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access		Name		Description																											
	[31:26]	-		reserved		保留。																											
	[25:16]	RW		hlcoefn2		水平亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 时，表示第 4 阶系数； 当寄存器地址的 bit[3:0]为 0x8 时，表示第 6 阶系数； 当寄存器地址的 bit[3:0]为 0xC 时，表示第 8 阶系数。																											
	[15:10]	-		reserved		保留。																											
	[9:0]	RW		hlcoefn1		水平亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 时，表示第 3 阶系数； 当寄存器地址的 bit[3:0]为 0x8 时，表示第 5 阶系数； 当寄存器地址的 bit[3:0]为 0xC 时，表示第 7 阶系数。																											

VADHCCOEf

VADHCCOEf 为 VAD 通道水平色度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

		Offset Address 0x1700~0x178C				Register Name VADHCCOEf				Total Reset Value 0x0000_0000																							
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved				hccoefn2				reserved				hccoefn1																			
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access		Name		Description																											
	[31:26]	-		reserved		保留。																											



[25:16]	RW	hccoefn2	水平色度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。
[15:10]	-	reserved	保留。
[9:0]	RW	hccoefn1	水平色度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。

VADVLCOEFL

VADVLCOEFL 为 VAD 通道垂直亮度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

Offset Address		Register Name		Total Reset Value				
0x1800~0x188C		VADVLCOEFL		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		vlcoefn2		reserved		vlcoefn1	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:26]	-	reserved	保留。					
[25:16]	RW	vlcoefn2	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。					
[15:10]	-	reserved	保留。					
[9:0]	RW	vlcoefn1	垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。					



VADVCCOEF

VADVCCOEF 为 VAD 通道垂直色度缩放滤波系数寄存器。共有 18 组寄存器，每组寄存器由低 4 位地址为 0x0、0x4 或 0x8、0xC 的 2 个寄存器组成，表示 4 阶系数。

		Offset Address 0x1900~0x198C				Register Name VADVCCOEF				Total Reset Value 0x0000_0000																							
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved				vccoefn2				reserved				vccoefn1																			
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access		Name		Description																											
	[31:26]	-		reserved		保留。																											
	[25:16]	RW		vccoefn2		垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 2 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 4 阶系数。																											
	[15:10]	-		reserved		保留。																											
	[9:0]	RW		vccoefn1		垂直亮度缩放滤波系数。 当寄存器地址的 bit[3:0]为 0x0 或者 0x8 时，表示第 1 阶系数； 当寄存器地址的 bit[3:0]为 0x4 或者 0xC 时，表示第 3 阶系数。																											

VHDMIMGSPoS_p

VHDMIMGSPoS_p 为 VHD 通道 De-interlace 分区，子图像 p 的起始位置和子画面有效标识寄存器。

		Offset Address 0x2300~0x237C				Register Name VHDMIMGSPoS _p				Total Reset Value 0x0000_0000																							
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved	die_valid	die_mode	reserved	spos_y				reserved				spos_x																			
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access		Name		Description																											
	[31]	-		reserved		保留。																											



[30]	RW	die_valid	子画面 p 的 de-interlace 有效标识。 0: 无效; 1: 有效。
[29]	RW	die_mode	子画面 p 的 de-interlace 模式 0: 根据 VHDDIECTRL 寄存器配置工作; 1: 采用两场中值模式。
[28]	-	reserved	保留。
[27:16]	RW	spos_y	垂直起始坐标, 以行为单位。 注意: 对于隔行输出, 配置实际坐标必须是偶数。逐行输出无此限制。
[15:12]	-	reserved	保留。
[11:0]	RW	spos_x	水平起始坐标, 以像素单位。 注意: 配置实际坐标必须是偶数。

VHDMIMGFPOSp

VHDMIMGFPOSp 为 VHD 通道 De-interlace 分区, 子图像 p 的结束位置寄存器。

Offset Address		Register Name		Total Reset Value					
0x2380~0x23FC		VHDMIMGFPOSp		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved	fpos_y			reserved	fpos_x			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:28]	RW	reserved	保留。						
[27:16]	RW	fpos_y	垂直结束坐标, 以行为单位。 注意: 对于隔行输出, 配置实际坐标必须是奇数。逐行输出无此限制。						
[15:12]	-	reserved	保留。						
[11:0]	RW	fpos_x	水平结束坐标, 以像素单位。 配置实际坐标必须是奇数。						



VADMIMGSPoS_p

VADMIMGSPoS_p 为 VAD 通道 De-interlace 分区，子图像 p 的起始位置和子画面有效标识寄存器。

	Offset Address 0x2400~0x247C				Register Name VADMIMGSPoS _p								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	die_valid	die_mode	reserved	spos_y								reserved	spos_x																		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																											
[31]	-		reserved		保留。																											
[30]	RW		die_valid		子画面 p 的 de-interlace 有效标识。 0: 无效; 1: 有效。																											
[29]	RW		die_mode		子画面 p 的 de-interlace 模式。 0: 根据 VADDIECTR 寄存器配置工作; 1: 采用 2 场中值模式。																											
[28]	-		reserved		保留。																											
[27:16]	RW		spos_y		垂直起始坐标，以行为单位。 注意：对于隔行输出，配置实际坐标必须是偶数。逐行输出无此限制。																											
[15:12]	-		reserved		保留。																											
[11:0]	RW		spos_x		水平起始坐标，以像素单位。 配置实际坐标必须是偶数。																											

VADMIMGFPOS_p

VADMIMGFPOS_p 为 VAD 通道 De-interlace 分区，子图像 p 的结束位置寄存器。



	Offset Address 0x2480~0x24FC				Register Name VADMIMGFPOSp								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				fpos_y								reserved				fpos_x															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																													
[31:28]	-	reserved	保留。																													
[27:16]	RW	fpos_y	垂直结束坐标，以行为单位。 注意：对于隔行输出，配置实际坐标必须是奇数。逐行输出无此限制。																													
[15:12]	-	reserved	保留。																													
[11:0]	RW	fpos_x	水平结束坐标，以像素单位。 配置实际坐标必须是奇数。																													



7 音频接口

7.1 概述

音频输入输出接口 SIO (Sonic Input/Output)，用于和片外 Audio CODEC 芯片连接，完成音乐（语音）的播放及录制。提供 3 组音频接口 SIO0、SIO1 和 SIO2，SIO0 完成语音对讲的输入输出，SIO1 和 SIO2 完成 8/16 路音频输入。

7.2 特点

SIO 接口支持 PCM (Pulse Code Modulation) 和 I²S 两种模式。其中，PCM 接口主要用于语音通道，比如 VOIP 电话；I²S 接口主要用于配合 AUDIO CODEC 完成音乐播放和录音。SIO 接口还支持 DMA 操作。

7.2.1 PCM 接口特点

PCM 接口有如下特点：

- 支持主模式和从模式。
- 支持单声道 8bit 或 16bit 线性 PCM 编码的发送和接收。
- 支持 2/4/8/16 路 8/16 bit 数据的多路接收。
- 支持由芯片内部产生位时钟以及帧同步信号，也可支持外接时钟及同步信号。
- PCM 接口帧同步信号仅支持短脉冲同步信号（同步信号的持续时间为 1 个时钟周期），支持标准和自定义 2 种模式。
- 支持发送和接收通道单独使能。
- 接收通道和发送通道具有独立的 FIFO（深度为 16）。

7.2.2 I²S 接口特点

I²S 接口有如下特点：

- 支持主模式和从模式。
- 支持左右声道 16/18/20/24/32bit 数据位宽的发送和接收。
- 支持 2/4/8/16 路 8/16bit 数据的多路接收。



- 支持 8K~192K 采样率。
- I²S 接收通道和发送通道具有独立的 FIFO，并且，每个通道的左声道和右声道均有独立的 FIFO，其 FIFO 深度为 16，FIFO 水线可调。
- I²S 支持发送和接收通道单独使能。
- 对于 I²S 接口 16bit 数据宽度的传输模式，支持左右声道接收数据合并成一个 32bit 数据在接收 FIFO 中存储，支持左右声道发送数据合并成一个 32bit 数据写入发送 FIFO，从而提高 FIFO 的缓冲能力。多路（2/4/8/16 路 8/16bit 数据）接收时不支持该合并功能。

7.3 信号描述

SIO 模块提供 3 组 SIO 接口：SIO0、SIO1 和 SIO2。这 3 组接口的信号描述如表 7-1 所示。

表7-1 SIO 接口信号描述

信号名称	方向	描述	对应管脚
SIO0_DI	I	数据输入。	SIO0DI
SIO0_DO	O	数据输出。	SIO0DO
SIO0_XFS	I/O	I ² S 发送左右声道选择信号（与 DAC 接口），或 PCM 帧同步信号。与 GPIO 复用。（复用时的配置信息请参见“ 管脚复用配置 ”）	SIO0XFS
SIO0_RFS	I/O	I ² S 接收左右声道选择信号（与 ADC 接口），或 PCM 帧同步信号。	SIO0RFS
SIO0_XCK	I/O	I ² S/PCM 发送位流时钟。与 GPIO 复用。（复用时的配置信息请参见“ 管脚复用配置 ”）	SIO0XCK
SIO0_RCK	I/O	I ² S/PCM 接收位流时钟。	SIO0RCK
SIO_MCLK	O	I ² S 或 PCM 接口主时钟，可作为 CODEC 的工作时钟。 与 GPIO 复用。（复用时的配置信息请参见“ 管脚复用配置 ”）	ACKOUT
SIO1_DI	I	数据输入。	SIO1DI
SIO1_RFS	I/O	I ² S 接收左右声道选择信号（与 ADC 接口），或 PCM 帧同步信号。	SIO1RFS
SIO1_RCK	I/O	I ² S/PCM 接收位流时钟。	SIO1RCK
SIO2_DI	I	数据输入。 与 SIO1_DO 或 GPIO 复用。（复用时的配置信息请参见“ 管脚复用配置 ”）	SIO2DI



信号名称	方向	描述	对应管脚
SIO2_RFS	I/O	I ² S 接收左右声道选择信号（与 ADC 接口），或 PCM 帧同步信号。 与 SIO1_XFS 或 GPIO 复用。（复用时的配置信息请参见“管脚复用配置”）	SIO2RFS
SIO2_RCK	I/O	I ² S/PCM 接收位流时钟。 与 SIO1_XCK 或 GPIO 复用。（复用时的配置信息请参见“管脚复用配置”）	SIO2RCK

有一些 AUDIO CODEC 的发送左右声道选择信号和接收左右声道选择信号是同一个信号，与这些 CODEC 对接时，只需连接 SIO0_RFS 即可，SIO0 内部 XFS 信号通过配置系统控制寄存器 SC_PERCTRL12[sio0_xfs]=0b1 由 RFS 获得，这样 SIO0_XFS 可被用作 GPIO。位流时钟可通过 SC_PERCTRL12[sio0_xck]配置；SIO1 通过配置 SC_PERCTRL12[sio1_xfs]和 SC_PERCTRL12[sio1_xck]。

7.4 功能描述

典型应用

SIO0 用于语音对讲（输入输出），以下对 I²S 接口的典型连接进行说明。



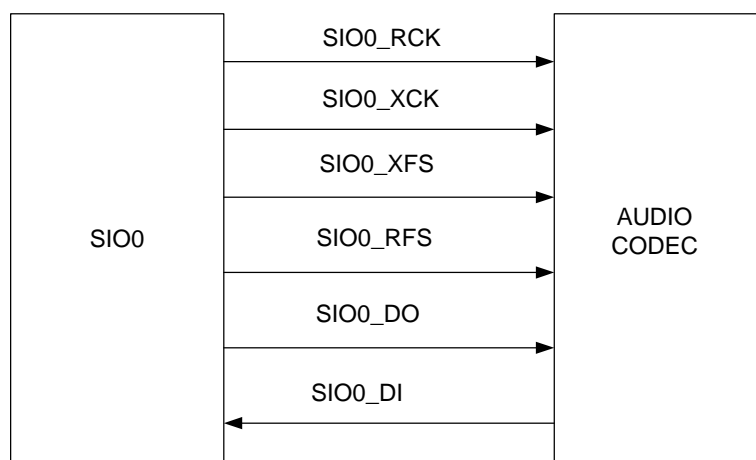
注意

在主模式下，AUDIO CODEC 的工作时钟要使用芯片提供的主时钟（SIO_MCLK）信号，而不能使用外接的晶振，否则，可能导致声音失真。



主模式下，I²S 接口的典型连接如图 7-1 所示。

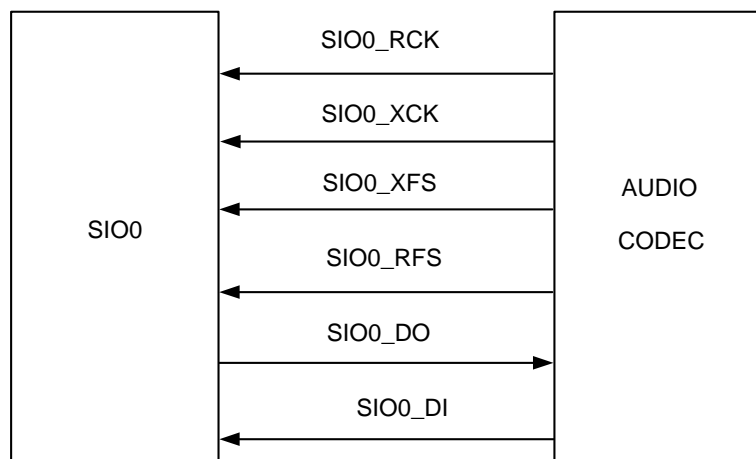
图7-1 I²S 接口主模式连接示意图



在主模式下，位流时钟和左右声道选择信号由 SIO 送给 AUDIO CODEC。而在从模式下，位流时钟和左右声道选择信号由 AUDIO CODEC 送给 SIO。

从模式下，I²S 接口的典型连接如图 7-2 所示。

图7-2 I²S 接口从模式连接示意图



在从模式下，AUDIO CODEC 的主工作时钟可以使用芯片提供的主时钟（SIO_MCLK）信号，也可以使用外接的晶振。

由 SIO 提供时钟和同步信号时，PCM 接口的典型连接图 7-3 所示。AUDIO CODEC 的工作时钟要用芯片提供的 SIO_MCLK 信号，而不能用外接的晶振，否则，可能导致声音失真。

由 SIO 提供时钟和同步信号时（主模式），PCM 接口的连接同 I²S 模式一样，如图 7-1 所示。

由 AUDIO CODEC 提供时钟和同步信号时（从模式），PCM 接口的连接同 I²S 模式一样，如图 7-2 所示。

由 AUDIO CODEC 提供时钟和同步信号时，AUDIO CODEC 的主工作时钟可以使用芯片提供的主时钟（SIO_MCLK）信号，也可以使用外接的晶振。

SIO1 和 SIO2 用于典型 8/16 路 16bits 音频数据的录音，以 SIO1 示意，对其连接进行说明。I²S/PCM 模式连接如图 7-3 和图 7-4 所示。

图7-3 录音 I²S/PCM 主模式

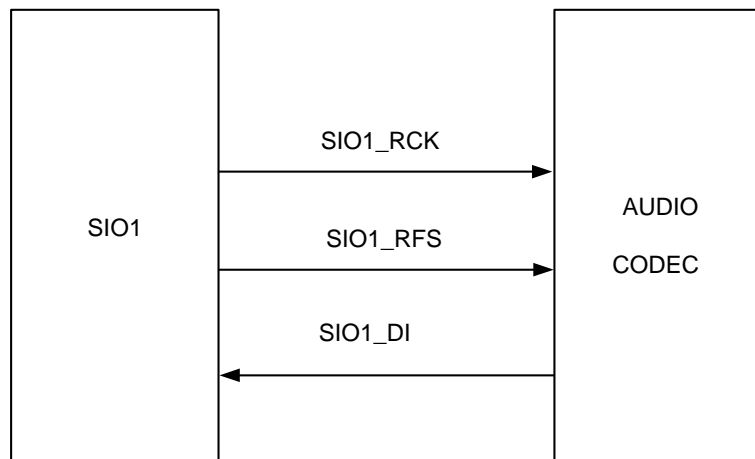
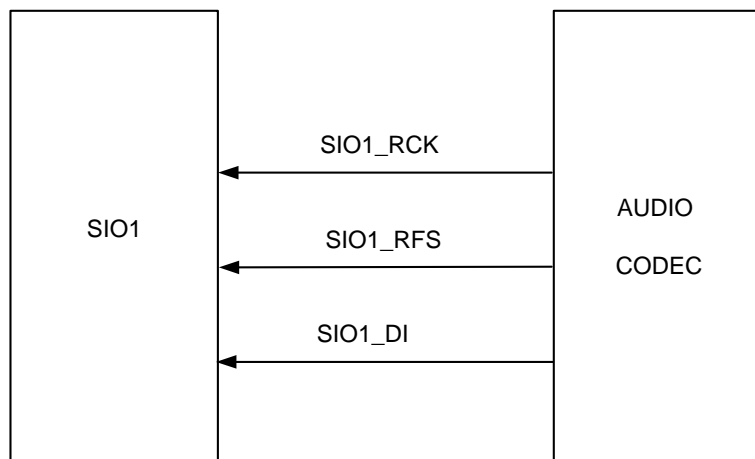


图7-4 录音 I²S/PCM 从模式



录音采用多路接收功能，即支持接收输入 2/4/8/16 路，数据位宽为 8/16bits。SIO2 连接与上描述相同，两者在应用上的不同是，SIO1 除了有 SIO1_RCK、SIO1_RFS、SIO1_DI 管脚，还可以将 SIO2_RCK、SIO2_RFS、SIO2_DI 管脚复用为 SIO1_XCK、SIO1_XFS、SIO1_DO，增加应用灵活性。



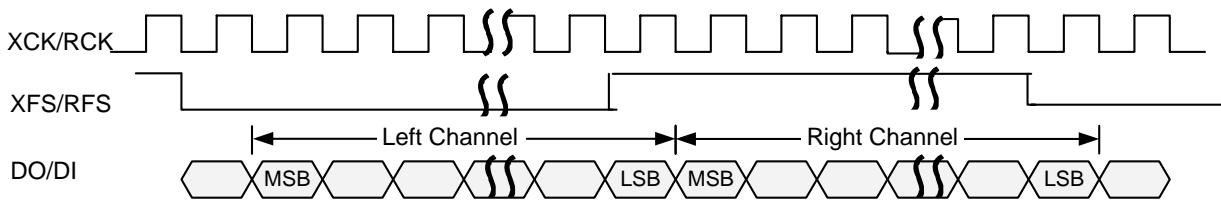
功能原理

SIO 接收通过内部总线传送过来的音频数据，然后按照设定的采样率，把音频数据通过 I²S 或 PCM 接口传送给对接的 AUDIO CODEC，AUDIO CODEC 进行 DA (Digital-to-Analog) 转换后进行声音播放。同时，通过 I²S 或 PCM 接口，SIO 接收对接 AUDIO CODEC 进行 AD (Analog-to-Digital) 转换后的音频数据，先存入内部 FIFO，然后由 CPU 取走并存储，从而完成录音功能。

I²S 接口传输数据分为左右两个声道，根据 XFS (RFS) 信号的高低电平区分，如图 7-5 所示。按照协议，用 XCK/RCK 时钟的上升沿进行数据采样，MSB 在 XFS/RFS 变化的下一个时钟周期有效。总是先传送 MSB，后传送 LSB。

I²S 接口的时序如图 7-5 所示。

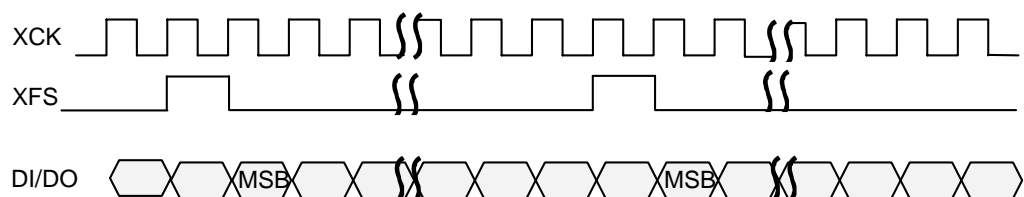
图7-5 I²S 接口时序



PCM 接口传输的数据是单声道数据，XFS 标识数据的起始位置，先发送（接收）最高有效位 MSB，使用时钟的下降沿采样数据。标准模式时序中，MSB 数据在 XFS 高电平脉冲之后一个周期有效；自定义模式时序中，MSB 的位置是与 XFS 的高电平脉冲对齐的。

PCM 接口标准模式下的时序如图 7-6 所示。

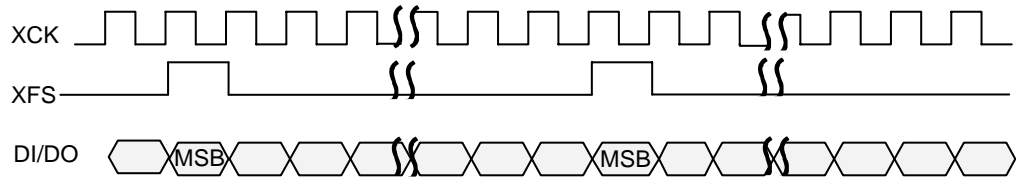
图7-6 PCM 接口标准模式时序



PCM 接口自定义模式下的时序如图 7-7 所示。

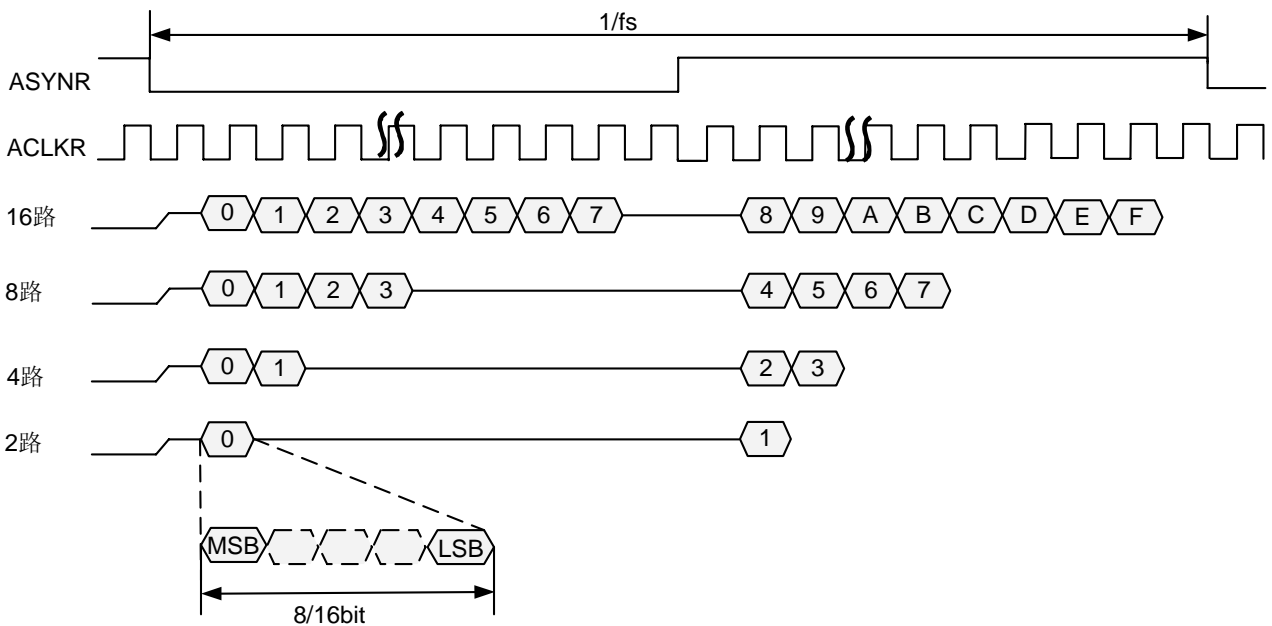


图7-7 PCM 接口自定义模式时序



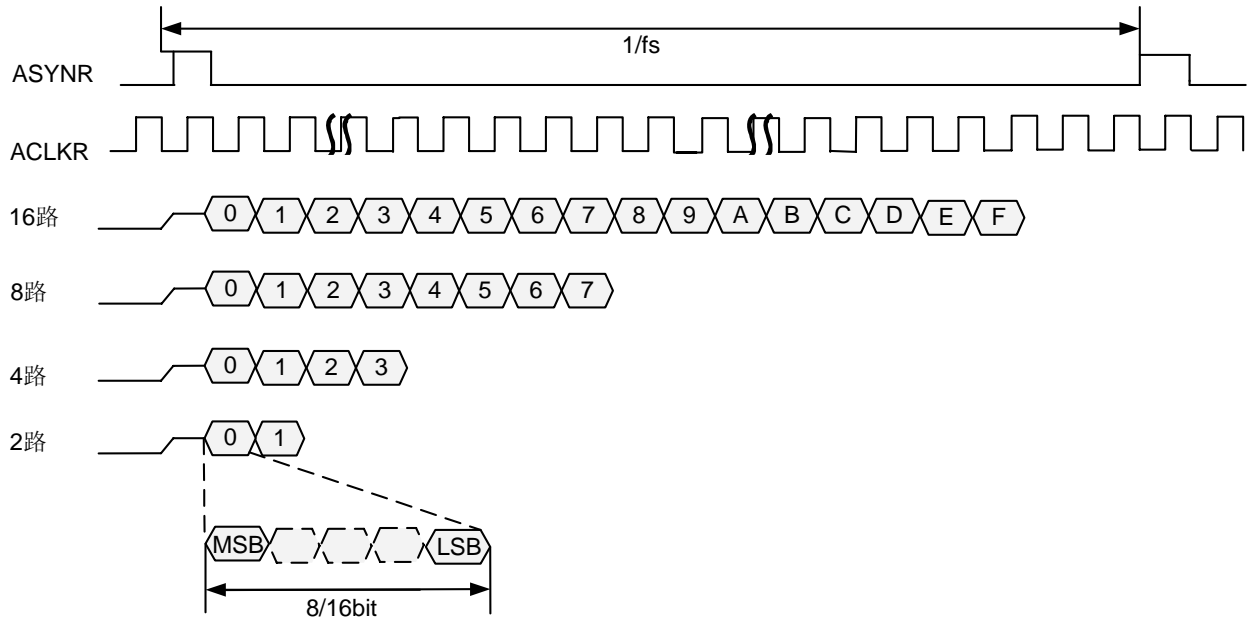
I²S 进行多路（2/4/8/16 路 8/16bit）接收时，数据分别放于 I²S 时序的左右声道，如图 7-8 所示。

图7-8 I²S 2/4/8/16 路接收



PCM 模式下的多路接收，如图 7-9 所示。支持 PCM 标准和自定义两种模式，与单声道不同的是，SIO 可以选择数据采样时刻（上升沿或下降沿）接收。以上升沿为例。

图7-9 PCM 2/4/8/16 路接收



7.5 工作方式

管脚复用配置

通过对 IO Config 寄存器 reg65~reg70 进行配置，可以实现 SIO 的管脚复用，管脚复用关系如表 7-2 所示。

表7-2 管脚复用关系

信号名称	复用信号
SIO0XFS	GPIO0_2
SIO0XCK	GPIO0_3
ACKOUT	GPIO0_4
SIO2DI	SIO1DO
SIO2RFS	SIO1XFS
SIO2RCK	SIO1XCK



时钟门控

当不进行录音或者音频播放时，设置 SIO0（SIO1、SIO2）的 `SIO_CT_SET[rx_enable]` 和 `SIO_CT_SET[tx_enable]` 为 0 后，可以通过向只写寄存器 `SC_PERDIS` 相应位写入 1 关断 SIO 时钟，写入 0 无影响：

- 向 `SC_PERDIS[sio0_clkdis]` 写 1，关断 SIO0 的时钟。
- 向 `SC_PERDIS[sio1_clkdis]` 写 1，关断 SIO1 的时钟。
- 向 `SC_PERDIS[sio2_clkdis]` 写 1，关断 SIO2 的时钟。

如果要恢复时钟产生，可以通过向只写寄存器 `SC_PEREN` 相应位写入 1 打开时钟使能，写入 0 无影响：

- 向 `SC_PEREN[sio0clken]` 写 1，打开 SIO0 的时钟使能。
- 向 `SC_PEREN[sio1clken]` 写 1，打开 SIO1 的时钟使能。
- 向 `SC_PEREN[sio2clken]` 写 1，打开 SIO2 的时钟使能。

SIO 的时钟使能状态可以通过只读寄存器 `SC_PERCLKEN` 相应位查看，为 0 表示相应的模块时钟关闭；为 1 则表示相应的模块时钟打开。

时钟配置

3 个 SIO 相互独立，时钟配置方式相同。

如果 SIO0、SIO1 或 SIO2 工作在主模式下，需配置 `SC_PERIPHCTRL12[sio0_master]`、`SC_PERIPHCTRL12[sio1_master]` 或 `SC_PERIPHCTRL12[sio2_master]` 为 1，即选取主模式。然后通过配置 `SC_PERIPHCTRL16`、`SC_PERIPHCTRL13`、`SC_PERIPHCTRL14` 选择位流时钟和同步时钟的相应分频比。

软复位

通过配置系统控制器 `SC_PERIPHCTRL8[sio0_srst]`、`SC_PERIPHCTRL8[sio1_srst]` 和 `SC_PERIPHCTRL8[sio2_srst]` 为 1，可实现对 SIO0、SIO1 和 SIO2 的单独软复位。复位后各配置寄存器的值均复位为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

中断或查询方式下的播放和录音

1. 初始化

初始化步骤如下：

- 步骤 1 设置 `SIO_CT_SET/SIO_CT_CLR[rx_enable]` 和 `SIO_CT_SET/SIO_CT_CLR[tx_enable]` 为 0，使 SIO 处于禁止状态；
- 步骤 2 设置 `SIO_MODE[sio_mode]`，选择 I²S 或 PCM 模式；如果是 PCM 模式，设置 `SIO_MODE[pcm_mode]`，选择时序类型。多路录音则要设置 `SIO_MODE[ext_rec_en]`、`SIO_MODE[chn_num]` 和 `SIO_MODE[clk_edge]`。
- 步骤 3 如果 SIO 为主模式，对时钟频率进行配置（如果是从模式，不需配置）。



- 步骤 4 配置 `SIO_DATA_WIDTH_SET` 寄存器、配置 `SIO_SIGNED_EXT` 寄存器，以设定正确的位宽。
- 步骤 5 配置 `SIO_CT_SET[rx_fifo_threshold]`和 `SIO_CT_SET[tx_fifo_threshold]`，即：设置接收 FIFO 和发送 FIFO 的水线。
- 步骤 6 如果是 I²S 模式，根据实际读写 FIFO 数据的方式，配置 `SIO_I2S_POS_MERGE_EN` 和 `SIO_I2S_START_POS`。同时，设置 `SIO_CT_SET[tx_data_merge_en]`和 `SIO_CT_SET[rx_data_merge_en]`。多路录音则不用进行该步骤设置。
- 步骤 7 根据中断屏蔽的需要，设置 SIO 的中断屏蔽寄存器 `SIO_INTMASK`，设置 `SIO_CT_SET[intr_en]`。
- 步骤 8 对外接的 AUDIO CODEC 进行设置。

----结束

2. 音频播放

音频播放步骤如下：

- 步骤 1 设置 `SIO_CT_SET[tx_fifo_disable]`为 1，然后再设置为 0，以清除发送 FIFO 的残留数据。
- 步骤 2 将发送数据写入发送 FIFO，向 `SIO_CT_SET[tx_enable]`写 1，启动数据发送。
- 步骤 3 查询方式下，通过读取 `SIO_TX_STA` 判断 TX_FIFO 状态；中断方式下，则根据中断状态 `SIO_INTSTATUS[tx_intr]`上报判断。当检测到发送 FIFO 中数据深度低于水线时，向发送 FIFO 写入数据，如此反复。如果数据发送全部完成，进入步骤 4。在传送完成之前，要保证 TX_FIFO 中的数据没有溢出，否则会造成声音不连续。
- 步骤 4 把 `SIO_CT_SET[tx_enable]`设置为 0。

----结束

3. 录音

录音步骤如下：

- 步骤 1 清除接收 FIFO 的残留数据，方法是：把 `SIO_CT_SET[rx_fifo_disable]`设置为 1，然后再设置为 0。
- 步骤 2 向 `SIO_CT_SET[rx_enable]`写 1，启动数据接收。

查询方式下，通过读取
- 步骤 3 `SIO_RX_STA` 检测 RX_FIFO 状态；中断方式下，则根据相应中断状态位检测。当检测到接收 FIFO 中数据深度高于水线时，从接收 FIFO 读出数据，如此反复。如果数据接收完成，进入步骤 4。在接收完成之前，要保证 RX_FIFO 中的数据没有溢出，否则会造数据丢失。
- 步骤 4 把 `SIO_CT_SET[rx_enable]`设置为 0，并把接收 FIFO 中剩余的数据全部读出。

----结束



DMA 方式下的播放和录音

1. 初始化

与查询和中断方式下的播放和录音方式相同。

2. 音频播放

步骤如下：

- 步骤 1 配置中断屏蔽寄存器 `SIO_INTMASK[tx_intr]` 为 1，屏蔽发送中断。
- 步骤 2 配置 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，具体请参见 DMA 配置章节。
- 步骤 3 设置 `SIO_CT_SET[tx_fifo_disable]` 为 1，然后再设置为 0，以清除发送 FIFO 的残留数据。
- 步骤 4 向发送 FIFO 写入初始数据，深度超过 FIFO 水线。（可写入全 0 的数据，代表静音。目的是：当启动播放时，因为 DMA 还没有向 FIFO 中写入数据，此时 SIO 会报发送 FIFO 下溢。如果先写入初始数据，就可以防止播放刚启动时误报 FIFO 溢出。）
- 步骤 5 设置 `SIO_CT_SET[tx_enable]` 为 1，启动播放。
- 步骤 6 通过 DMA 中断上报，判断数据是否发送完成，如果完成，则设置 `SIO_CT_SET[tx_enable]` 为 0。

----结束

3. 录音

步骤如下：

- 步骤 1 配置 DMA 数据通道，包括数据传输源地址、目的地址、数据传输个数、传输类型等参数，具体请参见 DMA 配置章节。
- 步骤 2 清除接收 FIFO 的残留数据，方法是：把 `SIO_CT_SET[rx_fifo_disable]` 设置为 1，然后再设置为 0。
- 步骤 3 设置 `SIO_CT_SET[rx_enable]` 为 1，启动数据接收。
- 步骤 4 如果停止录音，设置 `SIO_CT_SET[rx_enable]` 为 0。

----结束

7.6 寄存器概览

3 组 SIO 的寄存器基址如下：

- SIO0: 0x1004_0000。
- SIO1: 0x1005_0000。
- SIO2: 0x1006_0000。

寄存器概览如表 7-3 所示。



表7-3 SIO 寄存器概览

偏移地址	名称	描述	页码
0x03C	SIO_VERSION	SIO 版本寄存器	7-13
0x040	SIO_MODE	SIO 模式寄存器	7-13
0x044	SIO_INTSTATUS	SIO 中断状态寄存器	7-15
0x048	SIO_INTCLR	SIO 中断清除寄存器	7-17
0x04C	SIO_I2S_LEFT_XD	I ² S 左通道数据发送寄存器	7-18
0x050	SIO_I2S_RIGHT_XD	I ² S 右通道数据发送寄存器	7-18
0x050	SIO_PCM_XD	PCM 数据发送寄存器	7-18
0x054	SIO_I2S_LEFT_RD	I ² S 左通道数据接收寄存器	7-19
0x058	SIO_I2S_RIGHT_RD	I ² S 右通道数据接收寄存器	7-19
0x058	SIO_PCM_RD	PCM 数据接收寄存器	7-20
0x05C	SIO_CT_SET	I ² S/PCM 控制设置寄存器	7-20
0x060	SIO_CT_CLR	I ² S/PCM 控制清除寄存器	7-22
0x064	RESERVED	保留	-
0x068	SIO_RX_STA	SIO 接收状态寄存器	7-24
0x06C	SIO_TX_STA	SIO 发送状态寄存器	7-25
0x070~0x074	RESERVED	保留	-
0x078	SIO_DATA_WIDTH_SET	I ² S/PCM 数据宽度设置寄存器	7-25
0x07C	SIO_I2S_START_POS	I ² S 左右通道起始位置控制寄存器	7-26
0x080	I2S_POS_FLAG	I ² S 左右声道操作当前位置状态寄存器	7-27
0x084	SIO_SIGNED_EXT	高位数据符号扩展使能寄存器	7-28
0x088	SIO_I2S_POS_MERGE_EN	I ² S 左右声道合并使能寄存器	7-28
0x08C	SIO_INTMASK	SIO 中断屏蔽寄存器	7-29
0x090~0x09C	RESERVED	保留	-
0x0A0	SIO_I2S_DUAL_RX_CHN	I ² S 左右通道合并后数据接收寄存器	7-30



偏移地址	名称	描述	页码
0x0C0	SIO_I2S_DUAL_TX_CHN	I ² S 左右通道合并后数据发送寄存器	7-31

7.7 寄存器描述

SIO_VERSION

版本寄存器，用来记录 SIO 的版本号和进行 SIO 的自测试。

offset Address	Register Name	Total Reset Value
0x03C	SIO_VERSION	0x0000_0013

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							sio_loop	version							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
Bits	[31:9]		[8]	[7:0]																												
Access	-		RW	RO																												
Name	reserved		sio_loop	vesion																												
Description	保留。		SIO 循环和正常模式选择。 0: 正常模式; 1: SIO 发送和接收数据环回模式。用于 SIO 的自测试。在该模式下, 在 SIO 的对外接口处, SIO 接收串行数据线与 SIO 发送串行数据线直接相连。	SIO 的版本号。																												

SIO_MODE

模式寄存器，用来对 SIO 工作的基本模式进行选择：

- 主模式下，CRG 送时钟和同步信号给 CODEC，同时给 SIO。
- 从模式下，时钟和同步信号由外部 CODEC 送给 SIO。

I²S 或 PCM 的主从模式选择由系统控制寄存器 SC_PERCTRL12 设定，具体请参见“系统寄存器描述”中的“SC_PERCTRL12”。



offset Address	Register Name	Total Reset Value	
0x040	SIO_MODE	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	clk_edge, chn_num, ext_rec_en, pcm_mode, sio_mode	
Reset	0 0		
Bits	Access	Name	Description
[31:7]	-	reserved	保留。
[6]	RW	clk_edge	PCM 多路接收模式下，采样数据时钟边沿选择。 0: 下降沿有效； 1: 上升沿有效。
[5:4]	RW	chn_num	多路接收的通道数选择。 00: 2 chn; 01: 4 chn; 10: 8 chn; 11: 16 chn。 .
[3]	RW	ext_rec_en	标准模式下，I ² S 接收两个声道的数据，即左声道和右声道。 PCM 只接收一个声道的数据。 多路接收模式下，对于 I ² S 或 PCM，接收通道数可以配置。 该模式下，通道的数据宽度必须为 8bit 或 16bit。 0: 标准的 I ² S 或 PCM 接收模式； 1: 扩展的 I ² S 或 PCM 多路接收模式。
[2]	-	reserved	保留，必须配置为 0。
[1]	RW	pcm_mode	PCM 时序模式。 0: 标准模式； 1: 自定义模式。
[0]	RW	sio_mode	PCM/I ² S 模式选择。 0: I ² S 模式； 1: PCM 模式。



SIO_INTSTATUS

SIO 的中断状态指示寄存器。

对于接收中断，当接收 FIFO 的数据深度大于 FIFO 阈值时，会一直把高电平锁存到中断状态寄存器中，一直产生中断（即使 CPU 清一次中断，但中断状态寄存器会在下一个时钟周期再次置位）。因此，建议 CPU 的处理步骤为：

- 步骤 1 向 `SIO_CT_CLR[intr_en]` 写 1，关闭全局中断使能。
- 步骤 2 读中断状态寄存器 `SIO_INTSTATUS`。
- 步骤 3 根据中断源进行相应处理。
- 步骤 4 向 `SIO_INTCLR` 的相应位写 1，清除中断。
- 步骤 5 向写寄存器 `SIO_CT_SET[intr_en]` 写 1，打开全局中断使能。

----结束

发送中断的产生方式与接收中断产生方式相同，因此对于发送中断的处理，建议也采用相同的方式。

本寄存器是原始中断状态寄存器。在相应中断位屏蔽的情况下，中断条件满足时，对应中断状态位仍然会置位，但不会触发中断。



Offset Address	Register Name	Total Reset Value	
0x044	SIO_INTSTATUS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	tx_left_fifo_under tx_right_fifo_under rx_left_fifo_over rx_right_fifo_over tx_intr rx_intr	
Reset	0 0		
Bits	Access	Name	Description
[31:6]	-	reserved	保留。
[5]	RO	tx_left_fifo_under	I ² S 模式下，左声道发送 FIFO 下溢中断状态。PCM 模式下无效。 0: 未产生中断； 1: 已产生中断。
[4]	RO	tx_right_fifo_under	I ² S 模式下，右声道发送 FIFO 下溢中断状态。PCM 模式下，PCM 发送 FIFO 下溢标志。 0: 未产生中断； 1: 已产生中断。
[3]	RO	rx_left_fifo_over	I ² S 模式下，左声道接收 FIFO 上溢中断状态。PCM 模式下无效。 0: 未产生中断； 1: 已产生中断。
[2]	RO	rx_right_fifo_over	I ² S 模式下，右声道接收 FIFO 上溢中断状态。PCM 模式下，PCM 接收 FIFO 下溢标志。 0: 未产生中断； 1: 已产生中断。
[1]	RO	tx_intr	发送 FIFO 低于阈值中断状态。 0: 未产生中断； 1: 已产生中断。
[0]	RO	rx_intr	接收 FIFO 高于阈值中断状态。 0: 未产生中断； 1: 已产生中断。



SIO_INTCLR

中断清除寄存器，可以按位清除。

	Offset Address	Register Name	Total Reset Value
	0x048	SIO_INTCLR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		tx_left_fifo_under tx_right_fifo_under rx_left_fifo_over rx_right_fifo_over tx_intr rx_intr
Reset	0 0		
Bits	Access	Name	Description
[31:6]	-	reserved	保留。
[5]	WO	tx_left_fifo_under	I ² S 模式下，左声道发送 FIFO 下溢中断清除。PCM 模式下无效。 0: 不清除; 1: 清除。
[4]	WO	tx_right_fifo_under	I ² S 模式下，右声道发送 FIFO 下溢中断清除。PCM 模式下，PCM 发送 FIFO 下溢中断清除。 0: 不清除; 1: 清除。
[3]	WO	rx_left_fifo_over	I ² S 模式下，左声道接收 FIFO 上溢中断清除。PCM 模式下无效。 0: 不清除; 1: 清除。
[2]	WO	rx_right_fifo_over	I ² S 模式下，右声道接收 FIFO 上溢中断清除。PCM 模式下，PCM 接收 FIFO 下溢中断清除。 0: 不清除; 1: 清除。
[1]	WO	tx_intr	发送 FIFO 低于阈值中断清除。 0: 不清除; 1: 清除。
[0]	WO	rx_intr	接收 FIFO 高于阈值中断清除。 0: 不清除; 1: 清除。



SIO_I2S_LEFT_XD

I²S 模式下的左声道数据发送寄存器。

向寄存器写数据时，有效数据需放在寄存器的低 bit 区域。例如，数据宽度为 8bit 时，bit[7:0]为有效数据，bit[31:8]为无效数据；数据宽度为 16bit 时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。

offset Address	Register Name	Total Reset Value	
0x04C	SIO_I2S_LEFT_XD	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	tx_left_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	WO	tx_left_data	左声道发送数据。

SIO_I2S_RIGHT_XD

I²S 模式下的右声道数据发送寄存器。PCM 数据发送寄存器与 I²S 右声道数据发送寄存器是复用的。

向寄存器写数据时，有效数据需放在寄存器的低 bit 区域。例如，数据宽度为 8bit 时，bit[7:0]为有效数据，bit[31:8]为无效数据；数据宽度为 16bit 时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。

Offset Address	Register Name	Total Reset Value	
0x050	SIO_I2S_RIGHT_XD	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	tx_right_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	WO	tx_right_data	右声道发送数据。

SIO_PCM_XD

PCM 模式下的数据发送寄存器。PCM 数据发送寄存器与 I²S 右声道数据发送寄存器是复用的。



向寄存器写有效数据时，有效数据需放在寄存器的低 bit 区域。例如，8bit 宽度时，bit[7:0]为有效数据，bit[31:8]为无效数据。16bit 宽度时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。

	Offset Address	Register Name	Total Reset Value
	0x050	SIO_PCM_XD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved tx_data		
Reset	0 0		
Bits	Access	Name	Description
[31:16]	-	reserved	保留。
[15:0]	WO	tx_data	PCM 发送数据。

SIO_I2S_LEFT_RD

I²S 左声道数据接收寄存器。

SIO 模块把接收到的有效数据放在寄存器的低 bit 区域。例如，数据宽度为 8bit 时，bit[7:0]为有效数据，bit[31:8]为无效数据；数据宽度为 16bit 时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。

	Offset Address	Register Name	Total Reset Value
	0x054	SIO_I2S_LEFT_RD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_left_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	rx_left_data	I ² S 左声道接收数据。

注：I²S 模式下，在接收不使能时，右声道数据可能还没有写入 FIFO，从而导致左声道 FIFO 数据数目比右声道 FIFO 数据数目多一个。因此在 CPU 启动下一次接收前，应该把左右声道 FIFO 中数据全部读空。

SIO_I2S_RIGHT_RD

I²S 右声道数据接收寄存器。PCM 数据接收寄存器与 I²S 右声道数据接收寄存器是复用的。

SIO 模块把接收到的有效数据放在寄存器的低 bit 区域。例如，数据宽度为 8bit 时，bit[7:0]为有效数据，bit[31:8]为无效数据；数据宽度为 16bit 时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。



	Offset Address	Register Name	Total Reset Value
	0x058	SIO_I2S_RIGHT_RD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_right_data		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	rx_right_data
			I ² S 右声道接收数据。

注：I²S 模式下，在接收不使能时，右声道数据可能还没有写入 FIFO，从而导致左声道 FIFO 数据数目比右声道 FIFO 数据数目多一个。因此在 CPU 启动下一次接收前，应该把左右声道 FIFO 中数据全部读空。

SIO_PCM_RD

PCM 接收数据寄存器，它与 I²S 右声道接收寄存器是复用的。

SIO 模块把接收到的有效数据放在寄存器的低 bit 区域。例如，数据宽度为 8bit 时，bit[7:0]为有效数据，bit[31:8]为无效数据；数据宽度为 16bit 时，bit[15:0]为有效数据，bit[31:16]为无效数据。超出有效数据宽度的 bit 位由 SIO 模块自动置 0。

	Offset Address	Register Name	Total Reset Value
	0x058	SIO_PCM_RD	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		rx_data
Reset	0 0		
	Bits	Access	Name
	[31:16]	-	reserved
			保留。
	[15:0]	RO	rx_data
			PCM 接收数据。

SIO_CT_SET

为了能够方便的对 SIO 控制寄存器进行位操作，在 SIO 中，0x05C 为设置寄存器地址，当向 0x05C 寄存器中相应位写入 1 时，对应位被设为 1，写 0 无效；该寄存器属性为读写。



Offset Address		Register Name		Total Reset Value																														
0x05C/0x060		SIO_CT_SET		0x0000_8000																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved																rst_n	intr_en	rx_enable	tx_enable	rx_fifo_disable	tx_fifo_disable	rx_data_merge_en	tx_data_merge_en	rx_fifo_threshold	tx_fifo_threshold								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																															
[31:16]	-	reserved	保留。																															
[15]	RW	rst_n	I ² S/PCM 通道复位，低电平有效。 它对 I ² S/PCM 接收和发送模块（包括 FIFO）进行复位，因此发送和接收 FIFO 状态寄存器会变为 0。但不会复位 CPU 接口寄存器模块。																															
[14]	RW	intr_en	中断全局使能。 0：禁止； 1：使能。																															
[13]	RW	rx_enable	接收通道使能。 0：禁止； 1：使能。																															
[12]	RW	tx_enable	发送通道使能。 0：禁止； 1：使能。																															
[11]	RW	rx_fifo_disable	接收 FIFO 禁止。 0：使能； 1：禁止。																															
[10]	RW	tx_fifo_disable	发送 FIFO 禁止。 0：使能； 1：禁止。																															



[9]	RW	rx_data_merge_en	<p>数据接收合并使能，只在 I²S 模式下且数据位宽为 16bit 时有效。</p> <p>0: 禁止; 1: 使能。</p> <p>该位为 1 时，左右声道数据拼成一个 32bit 数据在 FIFO 中进行存储，左声道 16bit 数据占据高 16 位，右声道 16bit 数据占据低 16 位。这样可以提高 FIFO 的利用率，以提供更长时间的缓冲能力。</p> <p>CPU 从接收 FIFO 中读数据时，按照如下顺序：从左声道 FIFO 读出一个 32bit 数据（左右声道各 16bit 合并成的一个 32bit 数据），再从右声道 FIFO 读出一个 32bit 数据，如此反复。</p>
[8]	RW	tx_data_merge_en	<p>数据发送合并使能，只在 I²S 模式下且数据位宽为 16bit 时有效。</p> <p>0: 禁止; 1: 使能。</p> <p>该位为 1 时，左右声道数据拼成一个 32bit 数据在 FIFO 中进行存储，左声道 16bit 数据占据高 16 位，右声道 16bit 数据占据低 16 位。这样可以提高 FIFO 的利用率，以提供更长时间的缓冲能力。</p> <p>在这种情况下，CPU 向发送 FIFO 中写入数据时，按照如下顺序：向左声道 FIFO 写入一个 32bit 数据（左右声道各 16bit 合并成的一个 32bit 数据），再向右声道 FIFO 写入一个 32bit 数据，如此反复。</p>
[7:4]	RW	rx_fifo_threshold	<p>接收 FIFO 阈值。</p> <p>当 rx_right_depth ≥ (rx_fifo_threshold + 1) 时，报接收中断和 DMA 请求。</p>
[3:0]	RW	tx_fifo_threshold	<p>发送 FIFO 阈值。</p> <p>当 tx_right_depth < (tx_fifo_threshold + 1) 时，报发送中断和 DMA 请求。</p>

SIO_CT_CLR

为了能够方便的对 SIO 控制寄存器进行位操作，在 SIO 中，0x060 为清除寄存器地址，当向寄存器相应位写入 1 时，对应位被清除，写 0 无效。该寄存器属性为只写。



Offset Address		Register Name		Total Reset Value																														
0x05C/0x060		SIO_CT_SET		0x0000_8000																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved																rst_n	intr_en	rx_enable	tx_enable	rx_fifo_disable	tx_fifo_disable	rx_data_merge_en	tx_data_merge_en	rx_fifo_threshold	tx_fifo_threshold								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description																													
[31:16]	-		reserved		保留。																													
[15]	RW		rst_n		I ² S/PCM 通道复位，低电平有效。 它对 I ² S/PCM 接收和发送模块（包括 FIFO）进行复位，因此发送和接收 FIFO 状态寄存器会变为 0。但不会复位 CPU 接口寄存器模块。																													
[14]	RW		intr_en		中断全局使能。 0：禁止； 1：使能。																													
[13]	RW		rx_enable		接收通道使能。 0：禁止； 1：使能。																													
[12]	RW		tx_enable		发送通道使能。 0：禁止； 1：使能。																													
[11]	RW		rx_fifo_disable		接收 FIFO 禁止。 0：使能； 1：禁止。																													
[10]	RW		tx_fifo_disable		发送 FIFO 禁止。 0：使能； 1：禁止。																													



[9]	RW	rx_data_merge_en	<p>数据接收合并使能，只在 I²S 模式下且数据位宽为 16bit 时有效。</p> <p>0: 禁止; 1: 使能。</p> <p>该位为 1 时，左右声道数据拼成一个 32bit 数据在 FIFO 中进行存储，左声道 16bit 数据占据高 16 位，右声道 16bit 数据占据低 16 位。这样可以提高 FIFO 的利用率，以提供更长时间的缓冲能力。</p> <p>CPU 从接收 FIFO 中读数据时，按照如下顺序：从左声道 FIFO 读出一个 32bit 数据（左右声道各 16bit 合并成的一个 32bit 数据），再从右声道 FIFO 读出一个 32bit 数据，如此反复。</p>
[8]	RW	tx_data_merge_en	<p>数据发送合并使能，只在 I²S 模式下且数据位宽为 16bit 时有效。</p> <p>0: 禁止; 1: 使能。</p> <p>该位为 1 时，左右声道数据拼成一个 32bit 数据在 FIFO 中进行存储，左声道 16bit 数据占据高 16 位，右声道 16bit 数据占据低 16 位。这样可以提高 FIFO 的利用率，以提供更长时间的缓冲能力。</p> <p>在这种情况下，CPU 向发送 FIFO 中写入数据时，按照如下顺序：向左声道 FIFO 写入一个 32bit 数据（左右声道各 16bit 合并成的一个 32bit 数据），再向右声道 FIFO 写入一个 32bit 数据，如此反复。</p>
[7:4]	RW	rx_fifo_threshold	<p>接收 FIFO 阈值。</p> <p>当 rx_right_depth\geq(rx_fifo_threshold+1)时，报接收中断和 DMA 请求。</p>
[3:0]	RW	tx_fifo_threshold	<p>发送 FIFO 阈值。</p> <p>当 tx_right_depth$<$(tx_fifo_threshold+1)时，报发送中断和 DMA 请求。</p>

SIO_RX_STA

SIO 接收状态寄存器。



Offset Address		Register Name		Total Reset Value		
0x068		SIO_RX_STA		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	<div style="display: flex; justify-content: space-between;"> <div style="width: 65%; background-color: #cccccc; text-align: center; color: blue;">reserved</div> <div style="width: 15%; text-align: center; color: blue;">rx_left_depth</div> <div style="width: 15%; text-align: center; color: blue;">rx_right_depth</div> </div>					
Reset	0 0					
Bits	Access	Name	Description			
[31:10]	-	reserved	保留。			
[9:5]	RO	rx_left_depth	左声道接收 FIFO 深度指示。 只在 I ² S 模式下有效。			
[4:0]	RO	rx_right_depth	I ² S 模式下，为右声道接收 FIFO 深度指示。 PCM 模式下，为 PCM 接收 FIFO 深度指示。			

SIO_TX_STA

SIO 发送状态寄存器。

Offset Address		Register Name		Total Reset Value		
0x06C		SIO_TX_STA		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	<div style="display: flex; justify-content: space-between;"> <div style="width: 65%; background-color: #cccccc; text-align: center; color: blue;">reserved</div> <div style="width: 15%; text-align: center; color: blue;">tx_left_depth</div> <div style="width: 15%; text-align: center; color: blue;">tx_right_depth</div> </div>					
Reset	0 0					
Bits	Access	Name	Description			
[31:10]	RO	reserved	保留。			
[9:5]	RO	tx_left_depth	左声道发送 FIFO 深度指示。 只在 I ² S 模式下有效。			
[4:0]	RO	tx_right_depth	I ² S 模式下，为右声道发送 FIFO 深度指示。 PCM 模式下，为 PCM 发送 FIFO 深度指示。			

SIO_DATA_WIDTH_SET

该寄存器配置了在 I²S/PCM 模式下的数据宽度。



Offset Address	Register Name	Total Reset Value	
0x078	SIO_DATA_WIDTH_SET	0x0000_0009	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	reserved rx_mode tx_mode	
Reset	0 1 0 0 1		
Bits	Access	Name	Description
[31:6]	-	reserved	保留。
[5:3]	RW	rx_mode	接收数据长度配置位。 000: 8bit; 001: 16bit; 010: 18bit; 011: 20bit; 100: 24bit; 101: 32bit; 110~111: 保留。 对于 I ² S 模式, 支持 16/18/20/24/32bit; 对于 PCM 模式, 支持 8/16bit; 对于多路接收, 两种模式只支持 8/16bit。
[2:0]	RW	tx_mode	发送数据长度配置位。 000: 8 bit; 001: 16bit; 010: 18bit; 011: 20bit; 100: 24bit; 101: 32bit; 110~111: 保留。 对于 I ² S 模式, 支持 16/18/20/24/32bit; 对于 PCM 模式, 支持 8/16bit。

SIO_I2S_START_POS

I²S 左右声道起始位置配置控制寄存器。

在 I²S 模式下, 左右声道数据操作地址合并使能后, 控制起始访问是从左声道开始还是从右声道开始。



Offset Address		Register Name		Total Reset Value		
0x07C		SIO_I2S_START_POS		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved				start_pos_write	start_pos_read
Reset	0 0					
Bits	Access	Name	Description			
[31:2]	-	reserved	保留。			
[1]	RW	start_pos_write	写发送 FIFO 时： 0：从左声道开始访问； 1：从右声道开始访问。			
[0]	RW	start_pos_read	读接收 FIFO 时： 0：从左声道开始访问； 1：从右声道开始访问。			

I2S_POS_FLAG

I²S 左右声道操作当前位置状态寄存器。

在 I²S 模式下，左右声道数据操作地址合并使能后，指示下一次访问寄存器是从左声道开始还是从右声道开始。

Offset Address		Register Name		Total Reset Value		
0x080		I2S_POS_FLAG		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	reserved				start_pos_write	start_pos_read
Reset	0 0					
Bits	Access	Name	Description			
[31:2]	-	reserved	保留。			
[1]	RO	start_pos_write	写发送 FIFO 时：			



			0: 下一次从左声道开始访问; 1: 下一次从右声道开始访问。
[0]	RO	start_pos_read	读接收 FIFO 时: 0: 下一次从左声道开始访问; 1: 下一次从右声道开始访问。

SIO_SIGNED_EXT

高位数据符号扩展使能寄存器。该标志只对接收数据有效，对发送数据无效。PCM 模式和 I²S 模式下接收到的数据都支持符号扩展。

在接收有效数据位宽为 8/16/18/20/24 bit 时，如果该标志使能，把接收到的数据转换为 32bit 数据时，把 32bit 数据的高位无效比特设置为接收数据最高有效 bit 对应的值，然后再写入接收 FIFO。

以 16bit 位宽为例：

```
if (data_rx[15]==1)
    data_rx[31:16]=0xffff;
else
    data_rx[31:16]=0x0000;
```

	Offset Address	Register Name	Total Reset Value
	0x084	SIO_SIGNED_EXT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		signed_ext_en
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RW	signed_ext_en	高位数据符号扩展使能。 0: 禁止; 1: 使能。

SIO_I2S_POS_MERGE_EN

在 I²S 模式下，对左右声道数据的操作地址合并使能。



在 I²S 模式下，用 DMA 方式读写 SIO 的 FIFO 数据时，因为左右声道数据地址不同，需要 CPU 不断配置 DMA 操作的地址，导致 CPU 效率低。为了提高 CPU 的效率，提供左右声道数据的读写的统一地址使能控制。

使能情况下，读左右声道数据统一使用 SIO_I2S_DUAL_RX_CHN 寄存器，写左右声道数据统一使用 SIO_I2S_DUAL_TX_CHN 寄存器。

	Offset Address	Register Name	Total Reset Value
	0x088	SIO_I2S_POS_MERGE_EN	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		merge_en
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。
[0]	RW	merge_en	在 I ² S 模式下，对左右声道数据的操作地址合并使能。 0: 禁止; 1: 使能。

SIO_INTMASK

该寄存器是中断屏蔽寄存器。

	Offset Address	Register Name	Total Reset Value
	0x08C	SIO_INTMASK	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		tx_left_fifo_under tx_right_fifo_under rx_left_fifo_over rx_right_fifo_over tx_intr rx_intr
Reset	0 1 1 1 1 1 1		
Bits	Access	Name	Description
[31:6]	-	reserved	保留。



[5]	RW	tx_left_fifo_under	I ² S 模式下，左声道发送 FIFO 下溢中断屏蔽。PCM 模式下无效。 0: 不屏蔽; 1: 屏蔽。
[4]	RW	tx_right_fifo_under	I ² S 模式下，右声道发送 FIFO 下溢中断屏蔽。PCM 模式下，PCM 发送 FIFO 下溢中断屏蔽。 0: 不屏蔽; 1: 屏蔽。
[3]	RW	rx_left_fifo_over	I ² S 模式下，左声道接收 FIFO 上溢中断屏蔽。PCM 模式下无效。 0: 不屏蔽; 1: 屏蔽。
[2]	RW	rx_right_fifo_over	I ² S 模式下，右声道接收 FIFO 上溢中断屏蔽。PCM 模式下，PCM 接收 FIFO 下溢中断屏蔽。 0: 不屏蔽; 1: 屏蔽。
[1]	RW	tx_intr	发送 FIFO 低于阈值中断屏蔽。 0: 不屏蔽; 1: 屏蔽。
[0]	RW	rx_intr	接收 FIFO 高于阈值中断屏蔽。 0: 不屏蔽; 1: 屏蔽。

SIO_I2S_DUAL_RX_CHN

该寄存器为 I²S 左右声道操作地址合并使能以后，读取接收数据的寄存器。

Offset Address	Register Name	Total Reset Value	
0x0A0	SIO_I2S_DUAL_RX_CHN	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rx_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	rx_data	接收到的数据。



SIO_I2S_DUAL_TX_CHN

该寄存器为 I²S 左右声道操作地址合并使能后，写发送数据的寄存器。

	Offset Address	Register Name	Total Reset Value
	0x0C0	SIO_I2S_DUAL_TX_CHN	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	tx_data		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	WO	tx_data	发送的数据。



8 MMC/SD/SDIO 控制器

8.1 概述

MMC/SD/SDIO 控制器（以下简称 MMC）用来处理对 SD/MMC 存储卡的读写等操作，并可以通过 SDIO（Secure Digital Input/Output）协议实现对扩展外设（如 Blue Tooth、WiFi 等）的支持。MMC 可以控制符合以下协议的设备（并对低版本的协议向下兼容）：

- SD mem-version 2.00
- SDIO-version 1.10
- MMC-version 4.2

8.2 特点

MMC 控制器具有以下特点：

- 支持 DMA 数据传输方式。
- 包含数据发送方向与接收方向 2 个 FIFO，大小为 16×32bit。
- FIFO 阈值可配置，DMA 传输时 burst 大小可配置。
- FIFO 上溢出与下溢出中断告警，防止数据传输错误。
- 命令与数据的 CRC 生成与校验。
- 支持接口时钟频率可编程。
- 支持低功耗模式关断 MMC 控制器时钟。
- 支持低功耗模式关断接口时钟。
- 支持接口数据位宽为 1bit 和 4bit 模式下的数据传输。
- 支持 1byte~65535byte 的块数据读写操作。
- 支持 MMC 卡流数据读写方式。
- 支持接口数据位宽为 1bit 和 4bit 数据传输模式下的 SDIO 中断检测。
- 支持 SDIO suspend 和 resume 操作。
- 支持 SDIO read wait 操作。



8.3 信号描述

MMC 接口信号如表 8-1 所示。

表8-1 MMC 接口信号描述

信号名称	方向	描述	对应管脚
SDIOCK	O	输出接口时钟信号，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1CK
SDIOCMD	I/O	双向指令信号，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1DAT2
SDIODAT0	I/O	双向数据信号 0，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1DAT3
SDIODAT1	I/O	双向数据信号 1，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1DAT4
SDIODAT2	I/O	双向数据信号 2，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1DAT5
SDIODAT3	I/O	双向数据信号 3，与 VO 管脚复用。（复用时的配置信息请参见“8.5.1 管脚复用配置”）	VO1DAT6

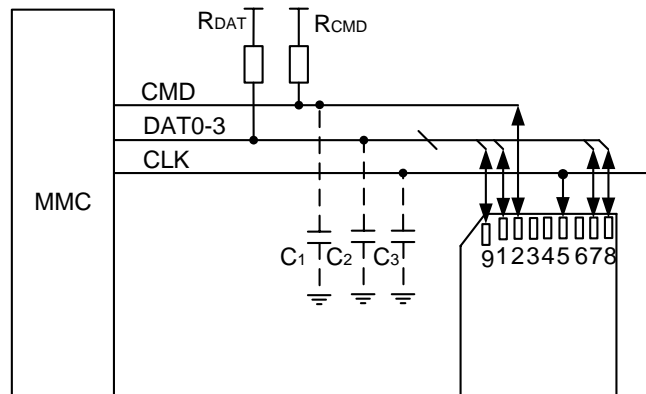
注：MMC_CCLK、MMC_CMD、MMC_DAT0~MMC_DAT3 分别对应后续说明中 CLK、CMD 及 DAT0~3 信号。

8.4 功能描述

典型应用

MMC 控制器的典型应用电路如图 8-1 所示。

图8-1 MMC 控制器典型应用电路图



说明

除图 8-1 中信号线外，卡槽一般还提供机械写保护信号、卡检测信号，系统可借助 GPIO 检测这 2 根信号线上的电平来完成机械写保护、热插拔时的卡检测功能。

MMC 控制器通过 1 根时钟信号线、1 根双向指令信号线和 4 根双向数据信号线与卡设备对接来完成命令与数据的交互。指令信号、数据信号均工作在上拉模式，上拉电阻参数及各信号线负载电容限制如表 8-2 所示。

表8-2 信号线负载参数

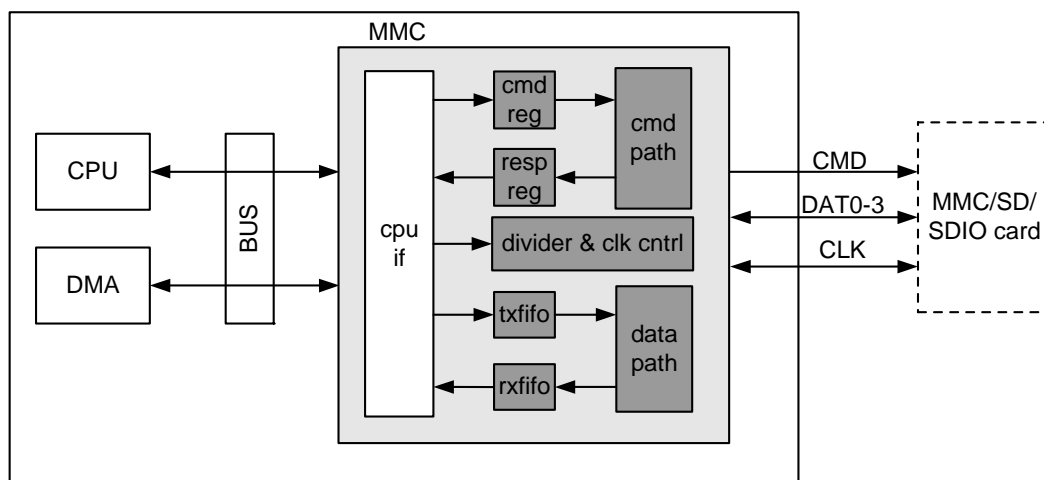
参数	最小值	最大值	描述
R _{DAT} 、R _{CMD}	10k Ω	100k Ω	上拉电阻。
负载电容 C _x	-	40pF	负载电容 C _x = C _{mmchost} + C _{bus} + C _{card} 。每张卡最大负载电容 C _{card} 为 10pF，所以 C _{mmchost} + C _{bus} ≤ 30pF。
信号线电感	-	16nH	F _{pp} ≤ 20MHz。

功能原理

MMC 控制器的功能框图如图 8-2 所示。



图8-2 MMC 控制器功能框图



MMC 控制器通过内部总线与系统连接，主要由命令通道、数据通道、接口时钟控制单元构成，各单元功能如下：

- 命令通道
完成指令的发送与响应的接收。
- 数据通道
配合命令通道完成数据读写操作。
- 接口时钟控制单元
根据应用需求改变接口时钟频率，控制接口时钟的关断与开启。

指令与响应

MMC 控制器与卡设备之间的所有交互操作均是通过指令的形式来完成的，包括卡初始化序列、寄存器读写、状态查询、数据传输等。

指令根据是否有数据传输可分为数据传输指令和非数据传输指令：

- 非数据传输指令
MMC 控制器基于指令信号线 CMD，进行串行方式指令的发送与响应的接收。
- 数据传输指令
除指令线上的交互外还伴随着数据线 DAT0-3 上的数据传输。

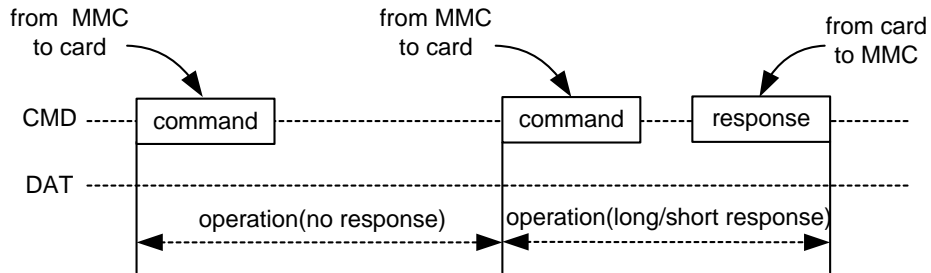
指令根据响应的类型又可分为：

- 无响应指令
如卡复位指令。
- 短响应指令
数据传输指令、卡状态查询等均属于这类指令。
- 长响应指令
仅用于读取卡的 CID、CSD 寄存器信息。



MMC 控制器与卡设备之间的非数据传输指令操作如图 8-3 所示。数据传输指令请参见图 8-6、图 8-7 所示。

图8-3 MMC 控制器非数据指令操作



指令（包括非数据传输指令和数据传输指令）为 48bit 的串行数据，由起始位、传输位、指令序号、指令参数、CRC 校验位和终止位组成。卡收到指令后，会根据指令类型返回 48bit 或 136bit 的响应。MMC 控制器的指令格式及响应格式如图 8-4 和图 8-5 所示。

图8-4 MMC 控制器指令格式

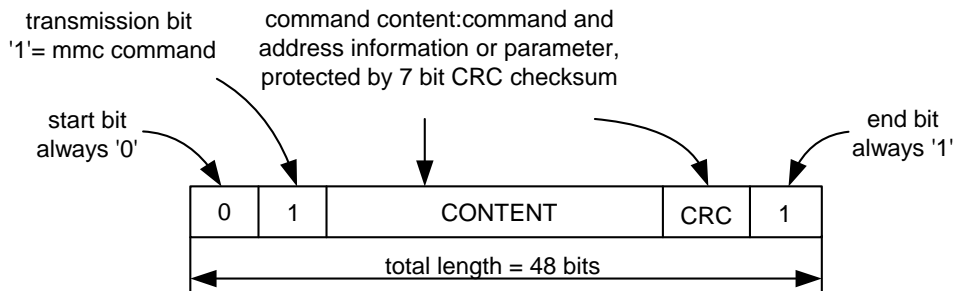
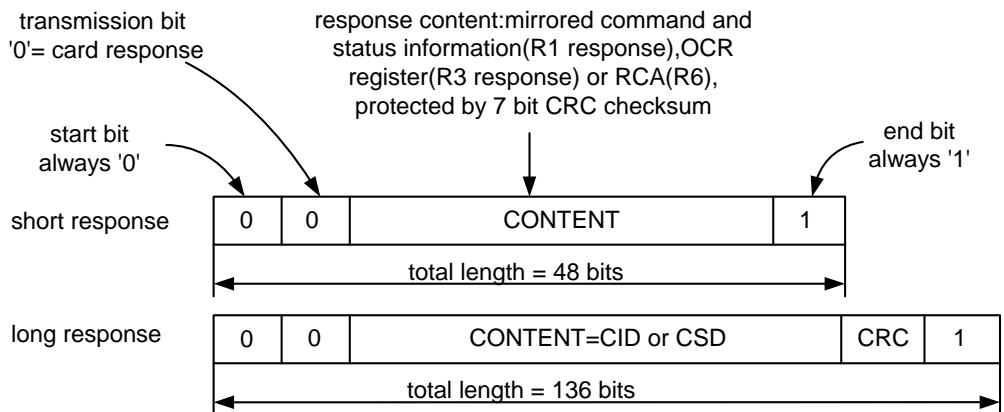


图8-5 卡设备响应格式





数据传输指令

MMC/SD/SDIO 卡支持数据读写方式如下：

- 流数据读写指令（stream）
仅 MMC 卡支持，其只使用 1 根数据线即 DAT0 进行数据传输，无 CRC 校验。
- 单块读写指令（single block）
一次传输完成一个块大小的数据读写，不需要使用停止命令结束一次数据传输。
- 多块读写指令（multiple block）
 - predefined block count 方式
在多块读写指令前，发送块数量指令指定待传输的数据量。
 - open ended 方式
发送读写指令后，在数据传输末尾，需使用停止指令来结束一次数据传输。

两种方式的差别在于控制器通知卡结束一次传输的方式不同。SD 卡仅支持 open ended 方式，而 MMC 卡两种方式均支持。

SDIO 设备的多块读写指令，不同于上述两种方式，在发送读写指令时，指令参数中包含待传输的数据量。

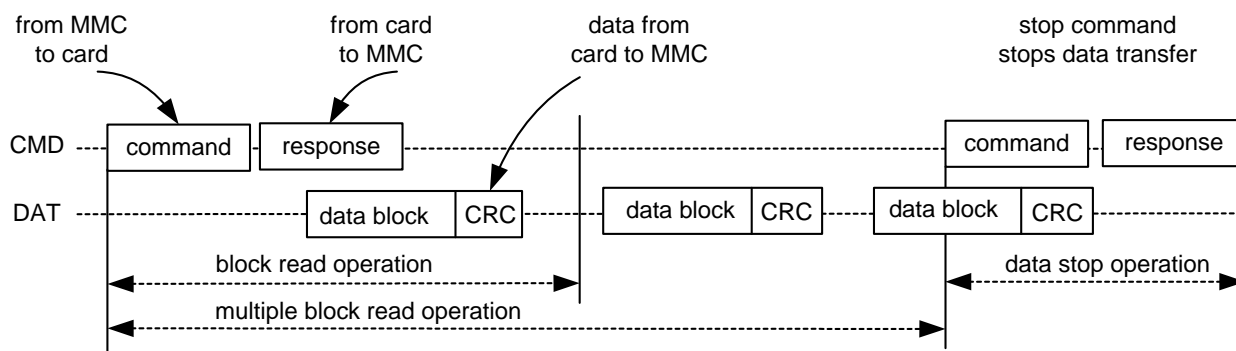
其中单块和多块读写指令为较常用的数据传输方式。通常 SD/MMC 卡数据传输的一个块大小为 512 字节，而 SDIO 设备可根据应用自定义。

说明

以块读写指令方式进行数据传输时，传输数据总量必须为块大小的整数倍。

数据传输指令均为短响应指令，并伴随着数据线上的数据传输。指令、响应及数据线上的时序配合关系如图 8-6、图 8-7 所示。

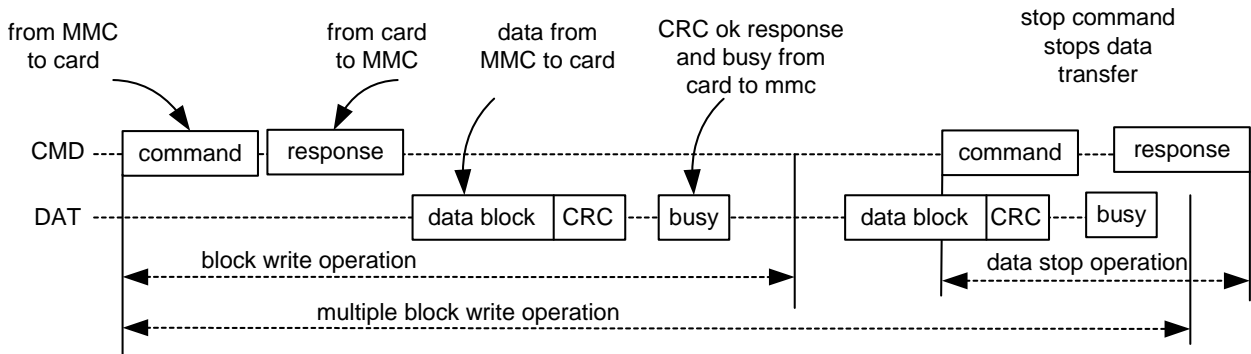
图8-6 单块与多块读操作



MMC 控制器向卡发送单块或多块读指令，在接收响应的过程中，并开始接收以块为单位的数据，其中每块数据中均包含有 CRC 校验位，以保证数据传输的完整性。在单块读指令操作时，控制器在接收一块数据后完成一次数据传输；在 open ended 方式的多块读操作中，控制器在接收多块数据后，需发送一条停止指令结束本次数据传输。



图8-7 单块与多块写操作



MMC 控制器向卡发送单块或多块写指令，在接收到响应后，开始往卡发送以块为单位的数据，其中每块数据中均包含有 CRC 校验位，卡会对每块数据进行 CRC 校验，并反馈 CRC 状态以确认数据传输的正确性。在单块写指令操作时，控制器在发送一块数据后完成一次数据传输；在 open ended 方式的多块写操作中，控制器在发送多块数据后，需发送一条停止指令结束本次数据传输。写操作结束后，卡可能会因为编程 flash 而处于繁忙状态，控制器需查询 DAT0 状态，以确认卡脱离繁忙状态后才能对卡进行下一步操作。

块方式读写中控制器与卡之间可采用 1bit 或 4bit 数据线方式进行数据传输。

在进行数据传输指令之前，应分别设置控制器与卡的数据传输位宽模式（1bit 或 4bit），使其保持一致。控制器的数据位宽可通过 MMC_CTYPE 寄存器来进行设置，卡的数据位宽则通过发送相应的指令进行设置。1bit 和 4bit 模式下的数据传输格式分别如图 8-8 和图 8-9 所示。

图8-8 1bit 数据线传输模式下的块数据格式

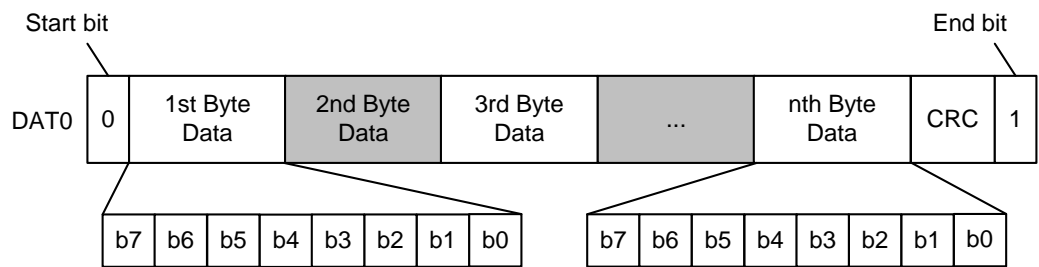
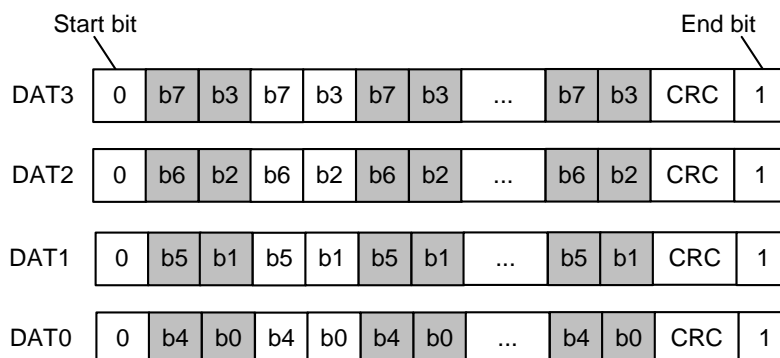




图8-9 4bit 数据线传输模式下的块数据格式



8.5 工作方式

8.5.1 管脚复用配置

MMC 控制器对外管脚与 GPIO 复用，使用 MMC 前，需要配置 IO Config 寄存器使能相应管脚的 MMC 功能，具体内容请参见寄存器 reg48~reg52 配置说明。

8.5.2 时钟门控

在软件完成当前命令或数据传输且未启动新的传输的情况下，可关断 MMC 时钟，但需要确保控制器已处于空闲状态。

步骤如下：

- 步骤 1 读取控制器状态寄存器 `MMC_STATUS`。
- 步骤 2 若 `MMC_STATUS[7:4]`、`MMC_STATUS[10]` 均为 0，则向 `MMC_CTRL` 写入 0，屏蔽控制器中断、DMA 请求使能等，进入步骤 3；若其中有一个为非 0，则延时等待，返回步骤 1。
- 步骤 3 将寄存器 `SC_PERDIS[mmc_clkdis]` 配置为 1，关闭控制器时钟。
- 步骤 4 如果需重新开启控制器工作时钟，则向 `SC_PEREN[mmc_clken]` 写 1。

----结束

8.5.3 软复位

在数据传输出现异常而导致控制器无法回到空闲状态时，可以配置 `SC_PERCTRL8[mmc_srst]` 寄存器进行软复位。默认复位，配置为 1 撤销软复位。可通过查询寄存器 `MMC_STATUS[data_fsm_busy]` 确认控制器是否处于空闲状态。建议在带电热插拔应用场景中在插卡后软复位控制器。



8.5.4 时钟配置

工作时钟配置

在使用 MMC 控制器前，需为其配置合适的工作时钟频率。MMC 控制器的工作时钟使用系统 PLL 的分频时钟，控制器工作时钟与 PLL 输出时钟频率关系为：

$$F_{\text{MMCCLK}} = F_{\text{PLL}} / [(2 \times \text{mmcclk_sel}) + 8]$$

其中分频因子 `mmcclk_sel` 为寄存器 `SC_PERCTRL9[mmcclk_sel]` 配置值，MMCCLK 频率 F_{MMCCLK} 应低于 50MHz。

接口时钟配置

遵从不同协议版本的卡处于不同的状态时，均使用不同的时钟频率。因此 MMC 控制器内部提供一个偶数分频器以便于将工作时钟分频至合适的接口时钟。控制器工作时钟 MMCCLK 与接口时钟 MMC_CCLK 的频率关系为：

$$F_{\text{MMC_CCLK}} = F_{\text{MMCCLK}} / (2 \times \text{clk_divider})$$

其中分频因子 `clk_divider` 为 `MMC_CLKDIV[clk_divider]` 的配置值。



注意

- 只有当 `MMC_CMD[start_cmd]` 和 `MMC_CMD[update_clk_only]` 置 1，时钟配置寄存器的 `MMC_CLKDIV`、`MMC_CLKENA` 的值才会被载入。当载入成功以后，控制器会自动清除 `MMC_CMD[start_cmd]`。如果此时有其它指令正在执行，则会产生 HLE(Hardware Locked Error) 中断。若产生 HLE 中断，重新执行操作即可。
- 当有指令执行和数据传输时，不能变更卡的时钟参数。

在改变卡的时钟频率之前，程序必须保证没有数据或指令正在传输。为了避免输出到多媒体卡的时钟产生毛刺，当改变卡的时钟频率的时候应该遵照以下步骤：

- 步骤 1 关闭接口时钟。将 `MMC_CLKENA` 寄存器配置为 `0x0000_0000`，并将 `MMC_CMD[start_cmd]`、`MMC_CMD[update_clk_regs_only]` 和 `MMC_CMD[wait_prvdata_complete]` 置 1，等待直到 `MMC_CMD[start_cmd]` 被自动清除。
- 步骤 2 设置分频因子。根据所需要的时钟频率设置 `MMC_CLKDIV`，并将 `MMC_CMD[start_cmd]` 和 `MMC_CMD[update_clk_regs_only]` 置 1，等待直到 `MMC_CMD[start_cmd]` 被自动清除。
- 步骤 3 重新使能接口时钟。将 `MMC_CLKENA` 寄存器配置为 `0x0000_0001`，并将 `MMC_CMD[start_cmd]` 和 `MMC_CMD[update_clk_regs_only]` 置 1，等待直到 `MMC_CMD[start_cmd]` 被自动清除。

----结束



8.5.5 初始化

在使用 MMC 控制器与卡进行命令和数据交互前，要进行初始化。步骤如下：

- 步骤 1 配置 IO Config 寄存器，使能相应管脚的 MMC 功能。请参见“8.5.1 管脚复用配置”。
- 步骤 2 配置 MMC 控制器工作时钟频率。请参见“8.5.4 工作时钟配置”。
- 步骤 3 当卡上电、指令和数据信号线上拉稳定后，软复位 MMC 控制器。请参见“8.5.3 软复位”。
- 步骤 4 清中断。将 MMC_RINTSTS 所有位置 1，清除原始中断状态位。
- 步骤 5 设置中断屏蔽寄存器。将 MMC_INTMASK 所有位置 1，使能各中断源。若使用 DMA 方式进行数据传输，应屏蔽接收/发送 FIFO 数据请求中断，将 MMC_INTMASK[txdr_int_mask]、MMC_INTMASK[rxdr_int_mask]置 0。
- 步骤 6 将 MMC_CTRL[int_enable]置 1，使能控制器中断功能。若使用 DMA 方式进行数据传输，应将 MMC_CTRL[dma_enable]置 1，使能控制器 DMA 请求功能。
- 步骤 7 配置超时参数寄存器 MMC_TMOUT。
- 步骤 8 配置 FIFO 参数寄存器 MMC_FIFOTH。其中包括 DMA 传输时的 burstsize 大小及接收方向与发送方向的阈值 rx_wmark、tx_wmark。

----结束

完成以上步骤后，就可以配置接口时钟，往卡发送指令了。

8.5.6 非数据传输指令

MMC 控制器在指令发送后，一旦收到任何响应（包括错误响应、有效响应和产生响应超时），控制器都会将 MMC_RINTSTS[cmd_done]置位。短响应被保存到 MMC_RESP0 寄存器中，长响应被保存到 MMC_RESP0~MMC_RESP3 寄存器中，MMC_RESP3[31] 为最高有效位，MMC_RESP0[0]为最低有效位。当指令发出以后，其错误是由指令响应以及 MMC_RINTSTS 相应的错误位来反映的。



注意

只有当 MMC_CMD[start_cmd]置 1、MMC_CMD[Update_clk_only]置 0 时，指令相关寄存器 MMC_BYTCNT、MMC_BLKSIZE、MMC_CMDARG、MMC_CMD 的值才会被载入。当载入成功以后，控制器会自动清除 MMC_CMD[start_cmd]，除非有其它指令正在执行，在这种情况下，就会产生 HLE 中断。若产生 HLE 中断，重新执行操作即可。在非数据传输指令执行时，MMC_BYTCNT、MMC_BLKSIZE 值被忽略。

发送非数据传输指令的步骤如下：

- 步骤 1 在 MMC_CMDARG 中设置相应的指令参数。



- 步骤 2 设置指令寄存器 `MMC_CMD`，其设置如表 8-3 所示。
- 步骤 3 等待指令被 MMC 控制器执行。如果指令已执行，控制器自动对 `MMC_CMD[start_cmd]` 清零。
- 步骤 4 通过查询 `MMC_RINTSTS[hle_int_status]`，检查是否产生 HLE 中断。
- 步骤 5 等待指令执行完毕。不管是收到响应或者响应时间超时，MMC 控制器都会将 `MMC_RINTSTS[cmd_done]` 置为 1。
- 步骤 6 检查是否有响应异常，有必要可读取响应值。可读取 `MMC_RINTSTS[rto_int_status]`、`MMC_RINTSTS[rerc_int_status]`、`MMC_RINTSTS[re_int_status]` 寄存器来进行检查响应超时、响应 CRC 错误或响应错误等。

----结束

表8-3 非数据指令 MMC_CMD 参考配置

参数	取值	描述
start_cmd	1	指令发送启动位。
update_clk_regs_only	0	非时钟参数更新指令。
data_expected	0	非数据传输指令。
cmd_index	Cmd index	命令序号。
send_initialization	0	当指令为卡复位时置 1，如 CMD0。
stop_abort_cmd	0	当指令为停止数据传输时置 1，如 CMD12。
rresponse_length	0	当响应为长响应类型时置 1。
rresponse_expect	1	当指令无响应时置 0，如：CMD0、CMD4、CMD15。
wait_prvdata_complete	1 或 0	在发送指令之前，控制器必须等待直到正在处理的数据传输指令结束。建议此位总置 1，除非该指令是为了在数据传输时查询卡状态或停止当前数据的传输。
check_response_crc	1 或 0	控制器是否会检查响应的 CRC 校验位。

8.5.7 单块或多块读数据

读取单块或多块数据的步骤如下：

- 步骤 1 向 `MMC_CTRL[fifo_reset]` 写 1，复位 FIFO 指针，查询等待直至该位自动清 0。
- 步骤 2 在 `MMC_BYTCNT` 寄存器写入待传输数据的字节数。
- 步骤 3 向 `MMC_BLKSIZE` 寄存器写入块的大小。



- 步骤 4 向 `MMC_CMDARG` 写入读取数据的起始地址。
- 步骤 5 根据表 8-4 的参数来设置指令寄存器 `MMC_CMD`，对于 SD/MMC 卡，分别使用 `CMD17/CMD18` 来进行单块/多块操作；对于 SDIO 卡，使用 `CMD53` 来进行单块/多块的读操作。一旦 `MMC_CMD` 寄存器配置完成，MMC 控制器就开始执行指令；而当指令被送到总线上以后，就会产生 `cmd_done` 中断。
- 步骤 6 检测 `MMC_RINTSTS[rxdr_int_status]`和 `MMC_RINTSTS[hto_int_status]`，如果其中之一为 1 或都为 1，应该从 `MMC_DATA` 寄存器读取 FIFO 中的数据，以便 MMC 控制器接收后面的数据；同时程序应检查数据错误中断，即寄存器 `MMC_RINTSTS[7]`、`MMC_RINTSTS[9]`、`MMC_RINTSTS[13]`和 `MMC_RINTSTS[15]`，如果有需要，程序可以发送一个停止指令中止数据的传输。
- 步骤 7 当 `MMC_RINTSTS[dto_int_status]`为 1 时，数据传输完成，从 `MMC_DATA` 寄存器中读取残留在 FIFO 中的数据。
- 步骤 8 若执行指令时已将 `MMC_CMD[send_auto_stop]`置为 1，控制器会自动发送停止指令结束一次数据传输，请参见“8.5.11 Auto-stop 使用配置”。

----结束

表8-4 单块或多块读数据 `MMC_CMD` 参考配置

参数	取值	描述
默认		
<code>start_cmd</code>	1	指令发送启动位。
<code>update_clk_regs_only</code>	0	非时钟参数更新指令。
<code>card_number</code>	0	-
<code>send_initialization</code>	0	当指令为卡复位时置 1，如 <code>CMD0</code> 。
<code>stop_abort_cmd</code>	0	当指令为停止数据传输时置 1，如 <code>CMD12</code> 。
<code>send_auto_stop</code>	0 or 1	请参见“8.5.11 Auto-stop 使用配置”。
<code>transfer_mode</code>	0	块传输。
<code>read_write</code>	0	从卡中读取数据。
<code>rspnse_length</code>	0	数据指令均为短响应。
<code>data_expected</code>	1	数据传输指令。
<code>rspnse_expect</code>	1	当指令无响应时置 0，如： <code>CMD0</code> 、 <code>CMD4</code> 、 <code>CMD15</code> 。
<code>cmd_index</code>	Cmd index	命令序号。



参数	取值	描述
wait_prvdata_complete	1 或 0	在发送指令之前，主设备必须等待正在处理的数据传输指令结束，建议此位总置 1，除非该指令是为了查询卡状态或停止当前数据的传输。
check_response_crc	1 或 0	控制器是否会检查响应的 CRC 校验位。

8.5.8 单块与多块写数据

写入单块或多块数据的步骤如下：

- 步骤 1 向 `MMC_CTRL[fifo_reset]` 写 1，复位 FIFO 指针，查询等待直至该位自动清 0。
- 步骤 2 向 `MMC_BYTCNT` 寄存器写入待传输数据的大小。
- 步骤 3 向 `MMC_BLKSIZE` 寄存器写入块的大小。
- 步骤 4 向 `MMC_CMDARG` 写入数据的起始地址。
- 步骤 5 将数据写入 FIFO（写 `MMC_DATA` 寄存器），通常在最开始的时候应写满 FIFO。
- 步骤 6 根据表 8-5 的参数来设置 `MMC_CMD`，对于 SD/MMC 卡，分别使用 CMD24/CMD25 来进行单块/多块操作；对于 SDIO 卡，使用 CMD53 来进行单块/多块的写操作。
- 步骤 7 检测 `MMC_RINTSTS[txdr_int_status]` 和 `MMC_RINTSTS[hto_int_status]`，如果其中之一为 1 或两者都为 1，写寄存器 `MMC_DATA` 往 FIFO 填充数据；同时应检测数据错误中断，即检测 `MMC_RINTSTS[7]`、`MMC_RINTSTS[9]`、`MMC_RINTSTS[13]` 和 `MMC_RINTSTS[15]`，如果有需要，程序可以发送一个停止指令以中止数据的传输。当 `MMC_RINTSTS[dto_int_status]` 为 1，数据传输结束。
- 步骤 8 若执行指令时已将 `MMC_CMD[send_auto_stop]` 置 1，控制器会自动发送停止指令结束一次数据传输。请参见“8.5.11 Auto-stop 使用配置”。
- 步骤 9 查询并等待 `MMC_STATUS[data_busy]` 由 1 变为 0。

----结束

表8-5 单块或多块写数据 MMC_CMD 参考配置

参数	取值	描述
Default		
start_cmd	1	指令发送启动位。
update_clk_regs_only	0	非时钟参数更新指令。
card_number	0	-
send_initialization	0	当指令为卡复位时置 1，如 CMD0。



参数	取值	描述
stop_abort_cmd	0	当指令为停止数据传输时置 1，如 CMD12。
send_auto_stop	0 or 1	请参见“8.5.11 Auto-stop 使用配置”。
transfer_mode	0	块传输。
read_write	1	往卡写入数据。
rspnse_length	0	数据指令均为短响应。
data_expected	1	数据传输指令。
rspnse_expect	1	当指令无响应时置 0，如：CMD0，CMD4，CMD15。
cmd_index	Cmd index	-
wait_prvdata_complete	1 或 0	在发送指令之前，主设备必须等待直到正在处理的数据传输指令结束，建议此位总置 1，除非该指令是为了查询卡状态或停止当前数据的传输。
check_response_crc	1 或 0	控制器是否会检查响应的 CRC 校验位。

8.5.9 流数据读写

流数据的读写方式，除了将 `MMC_CMD[transfer_mode]` 置 1 外，其它与块数据的读写方式一致。对于流数据的传输，通常需要使用控制器 auto-stop 功能。

8.5.10 DMA 方式数据传输

使用 DMA 方式进行数据传输前，需将 `MMC_CMD[dma_enable]` 置 1 来使能控制器 DMA 功能，并将寄存器 `MMC_INTMASK[txdr_int_mask]` 和 `MMC_INTMASK[txdr_int_mask]` 清 0，屏蔽控制器的 Receive FIFO data request 和 Transmit FIFO data request 中断。

使用 DMA 方式进行数据传输步骤如下：

- 步骤 1 复位 FIFO 指针。
- 步骤 2 配置 `MMC_BYTCNT`、`MMC_BLKSIZE`、`MMC_CMDARG`、`MMC_CMD` 寄存器启动数据传输指令。
- 步骤 3 分配 DMA 通道，并配置相应通道寄存器 `DMAC_CX_SRC_ADDR`、`DMAC_CX_DEST_ADDR`、`DMAC_CX_CONTROL`、`DMAC_CX_CONFIG`；若使用 DMA 链表方式传输，还需配置通道的 `DMAC_CX_LLI` 寄存器，并启动该通道进行传输。
- 步骤 4 查询等待至 DMA 中断上报，表明数据传输结束。



步骤 5 检测控制器数据传输错误中断，数据传输完成中断。

步骤 6 执行停止指令，并查询 `MMC_STATUS[data_busy]`。

----结束

进行数据传输时，需要注意如下事项：

- 以非 DMA 方式读写卡时，只有当数据传送完成时，DTO（Data Transfer Over）中断才会产生。而当卡的数据全部送出的时候，有可能还会有数据残留在 FIFO 中，而 `Rx_wmark` 中断的产生与否是由残留在 FIFO 中数据的字节数来决定的。软件判断出 DTO 中断产生后，应读出 FIFO 中所有的残留数据。
- 以 DMA 方式读写卡时，只有当所有 FIFO 中的数据都从 DMA 接口单元写入到内存中以后，DTO 中断才会产生。
- 数据传输时，数据量大小应该为 FIFO 数据宽度的整数倍。如果需要写入到卡的数据为 15 个字节，那么应该传送 16 个字节的数据到 FIFO，或当 DMA 方式使能时对 DMA 进行编程以执行 16 个字节数据的传输。程序仍然可以把 `MMC_BYTCNT` 设为 15，这时，只有 15 个字节的数据被传输到卡中。类似的，当从卡中读取 15 个比特的数据的时候，主设备将会从 FIFO 中读出 16 个字节的数据。

8.5.11 Auto-stop 使用配置

在多块读写指令操作中，需使用停止指令完成一次数据传输。停止指令的发送可以通过非数据传输指令的方式发送，也可以使用控制器的 auto-stop 功能发送。auto-stop 使用配置如下：

在执行块数据传输指令操作中，将 `MMC_CMD[send_auto_stop]` 置 1，在所有数据传输完成后，控制器会自动发送一次停止指令，以便卡能返回相应的状态；该停止指令的完成由 `MMC_RINTSTS[auto_cmd_done]` 中断位来反映，其响应被保存在 `MMC_RESP1` 中。

Auto-stop 功能应用场合：

- SD 卡
多块读写操作，如 `CMD18` 和 `CMD25`。
- MMC 卡
 - 流数据读写操作。
 - open-ended 方式的多块读写操作，如 `CMD18` 和 `CMD25`。
 - predefined block count 方式的多块读写操作不需要使用 auto-stop 功能，在 `CMD18` 和 `CMD25` 指令之前发送 `CMD23` 指令指定此次待传输的块数量。
- SDIO 设备
不需要使用 auto-stop 的功能。

8.5.12 停止或中止数据传输

停止指令用于打断 MMC 控制器与卡之间的数据传输，中止指令用于打断 I/O 数据的传输（仅用于 `SDIO_IOONLY` 或 `SDIO_COMBO`）。这两种指令的用法如下：

- 停止指令



该指令可以在数据传送的任何阶段进行发送。因为该指令是为了停止数据的传输，所以需要将指令寄存器 `MMC_CMD[5:0]` 设为 `CMD12`、将 `MMC_CMD[14]` 设置为 1、将 `MMC_CMD[13]` 设置为 0。

- 发送中止指令

仅用于 `SDIO_IOONLY` 或 `SDIO_COMBO`。为了中止数据的传输，需要通过 `CMD52` 设置 `SDIO` 卡的寄存器 `CCCR[ASx]` 位。

8.5.13 Suspend 和 Resume 操作

对于 `SDIO` 卡（内部最多可容纳 7 个功能设备），控制器可通过 `suspend` 操作暂停某一功能设备的数据传输，将 `SD` 接口总线让出给另一个有着更高优先级的功能设备。高优先级的功能设备完成数据传输后，控制器可以通过 `resume` 操作恢复前一功能设备未完成的传输。

`Suspend` 与 `Resume` 操作通过设置 `SDIO` 卡的 `CCCR` 寄存器相应比特来实现。读写 `CCCR` 寄存器，使用指令 `CMD52`。

`Suspend` 操作步骤如下：

- 步骤 1 通过查询 `CCCR` 寄存器的 `SBS` 位，判断 `SDIO` 卡是否支持 `suspend/resume` 操作。
- 步骤 2 通过查询 `CCCR` 寄存器的 `FSx` 位和 `BS` 位，判断待暂停的功能设备是否正在进行数据传输。**注意：如果 `BS` 位为 1，那么 `FSx` 位所指定的功能设备正在进行数据传输。**
- 步骤 3 为了暂停当前数据传输，需要将 `CCCR` 寄存器的 `BR` 位置 1。
- 步骤 4 检测 `CCCR` 寄存器的 `BS`（`Bus` 状态）位和 `BR`（`Bus` 释放）位状态是否清零。`BS` 位在数据总线正被使用的时候保持为 1；`BR` 位在总线完全释放之前都保持为 1。当 `BR` 与 `BS` 位都为 0 的时候，所选功能设备的数据传输就被成功暂停。
- 步骤 5 如果暂停正在进行的读操作，那么在 `suspend` 操作成功完成以后，必须置 `MMC_CTRL[abort_read_data]` 位来复位控制器数据传输功能，复位完成后，`MMC_CTRL[abort_read_data]` 位自动清零。
- 步骤 6 读 `MMC_TCBCNT` 寄存器获取已传输数据字节数。

----结束

`Resume` 操作步骤如下：

- 步骤 1 检查卡是否处于非传输状态，以确认总线处于空闲状态。
- 步骤 2 如果卡处于 `disconnect` 状态，使用 `CMD7` 将它选中。卡的状态可以通过 `CMD52/CMD53` 指令获取。
- 步骤 3 检查待恢复的功能设备是否准备好进行数据传输。可以通过查询 `CCCR` 寄存器的 `RFx` 标志位来进行确认，如果 `RF=1`，那么该功能设备就已准备好进行数据传输。
- 步骤 4 为了恢复传输，使用指令 `CMD52` 将功能设备号写入 `CCCR` 寄存器的 `FSx` 位。发送 `CMD52` 指令的同时应启动控制器进入数据传输状态：即往 `MMC_BLKSIZE` 寄存器写入块的大小，`MMC_BYTCNT` 寄存器写入剩余待传输数据量，`MMC_CMDARG` 寄存器的配置如表 8-6 所示，`MMC_CMD` 寄存器的配置与块传输类似。



步骤 5 当 CMD52 指令成功发送以后，数据传输被恢复了。读取 DF (Resume Data Flag) 标志位，如果为 1，那么在功能被恢复的同时，数据就开始传输了；如果标志是 0，那么就无数据需要传输。

步骤 6 如果 DF 标志位为 0，那么在读数据的情况下，控制器会等待一段时间后产生数据超时错误中断。

----结束

表8-6 Resume 操作 MMC_CMDARG 参考配置

MMC_CMDARG	取值	描述
bit[31]	0x1	读写标志
bit[30:28]	0x0, 访问 CCCR	功能设备号
bit[27]	0x1, 先写后读	实时标志
bit[26]	-	-
bit[25:9]	0x0D	寄存器地址
bit[8]	-	-
bit[7:0]	被恢复的功能号	写数据

8.5.14 Read wait 操作

Read wait 操作用于对 SDIO 卡暂停当前功能设备的数据传输，给任一功能设备发送指令。控制器可以根据应用需要随意决定暂停数据传输的时间长度。其步骤如下：

步骤 1 检查卡是否支持 read wait 操作；读取 CCCR 寄存器的 SRW 位。如果这一位为 1，那么卡的所有功能设备都支持 read wait 操作。使用 CMD52 读取这一位。

步骤 2 如果卡支持 read wait 等操作，将 MMC_CTRL[read_wait]位置 1。

步骤 3 如需恢复数据传输，清零 MMC_CTRL[read_wait]。

----结束

8.6 寄存器概览

MMC 寄存器概览如表 8-7 所示。

表8-7 MMC 寄存器概览（基址是 0x1003_0000）

偏移地址	名称	描述	页码
0x000	MMC_CTRL	控制寄存器	8-19



偏移地址	名称	描述	页码
0x004	RESERVED	保留	-
0x008	MMC_CLKDIV	时钟分频因子寄存器	8-20
0x00C	RESERVED	保留	-
0x010	MMC_CLKENA	时钟使能寄存器	8-21
0x014	MMC_TMOU	超时参数寄存器	8-21
0x018	MMC_CTYPE	接口位宽寄存器	8-22
0x01C	MMC_BLKSI	块大小寄存器	8-22
0x020	MMC_BYTCNT	传输数据大小寄存器	8-23
0x024	MMC_INTMASK	中断屏蔽寄存器	8-23
0x028	MMC_CMDARG	指令参数寄存器	8-25
0x02C	MMC_CMD	指令寄存器	8-26
0x030	MMC_RESP0	指令响应寄存器 0	8-28
0x034	MMC_RESP1	指令响应寄存器 1	8-28
0x038	MMC_RESP2	指令响应寄存器 2	8-29
0x03C	MMC_RESP3	指令响应寄存器 3	8-29
0x040	MMC_MINTSTS	中断状态寄存器	8-29
0x044	MMC_RINTSTS	原始中断状态寄存器	8-31
0x048	MMC_STATUS	状态寄存器	8-34
0x04C	MMC_FIFOTH	FIFO 参数寄存器	8-35
0x050~0x058	RESERVED	保留	-
0x05C	MMC_TCBCNT	接口传输计数寄存器	8-37
0x060	MMC_TBBCNT	FIFO 传输计数寄存器	8-37
0x064~0xFF	RESERVED	保留	-
0x100	MMC_DATA	数据寄存器	8-38



8.7 寄存器描述

MMC_CTRL

MMC_CTRL 为 MMC 控制寄存器，用于完成控制器的全局功能控制，包括中断全局使能控制、DMA 使能控制等。

Offset Address	Register Name	Total Reset Value	
0x000	MMC_CTRL	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved	abort_read_data send_irq_response read_wait dma_enable int_enable reserved dma_reset fifo_reset reserved	
Reset	0 0		
Bits	Access	Name	Description
[31:9]	-	reserved	保留。默认为 0，不可写 1。
[8]	RW	abort_read_data	读状态机复位控制位。 0: 默认保持值； 1: 如果使用 suspend 操作暂停正在进行的数据读传输，在 suspend 操作完成后，软件将该位使能，使控制器从数据传输状态（等待下一个 block 数据）恢复到 IDLE 状态。当控制器回到 IDLE 状态后，该比特会自动清 0。
[7]	RW	send_irq_response	0: 默认保持值； 1: 发送自动 IRQ 响应。 该比特在发送响应后自动清 0。 为了等待 MMC 产生中断，控制器发送 CMD40 并等待来自 MMC 中断响应。同时，如果控制器希望不再停留在中断等待状态，可以使能该比特，送出 CMD40 响应并回到 IDLE 状态。
[6]	RW	read_wait	0: 清除读等待； 1: 使能读等待。 发送读等待到 SDIO 设备。

[5]	RW	dma_enable	DMA 传输模式使能。 0: 禁止; 1: 使能。 即使 DMA 模式已经使能, CPU 仍然可以对 FIFO 进行读写, 但在实际操作中应该避免。如果 DMA 和 CPU 同时对 FIFO 进行读写, 控制器无法仲裁其优先级。
[4]	RW	int_enable	全局中断使能位。 0: 禁止; 1: 使能。 只有该比特有效且有中断源被使能的情况下, 中断输出才有效。
[3]	-	reserved	保留。
[2]	RW	dma_reset	DMA 接口功能复位控制。 0: 不复位; 1: 复位。 写 1 使能复位, 该比特在完成复位后自动清 0。
[1]	RW	fifo_reset	FIFO 复位控制位。 0: 不复位 FIFO 读写指针; 1: 复位 FIFO 读写指针。 写 1 使能复位, 该比特在完成复位后自动清 0。
[0]	-	reserved	保留。默认为 0, 不可写 1。

MMC_CLKDIV

MMC_CLKDIV 为接口时钟分频因子寄存器, 用于控制接口时钟频率。控制器工作时钟 MMCCLK 与接口时钟 MMC_CCLK 的频率关系为:

$F_{MMC_CCLK} = F_{MMCCLK} / 2 \times clk_divider$ 。只有当 **MMC_CMD[start_cmd]** 和 **MMC_CMD[Update_clk_only]** 置 1, 该寄存器的值才会被载入。

	Offset Address	Register Name	Total Reset Value
	0x008	MMC_CLKDIV	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		clk_divider
Reset	0 0		
Bits	Access	Name	Description
[31:8]	-	reserved	保留。



[7:0]	RW	clk_divider	接口时钟分频系数，偶数分频。 如 0x0 为无分频、0x1 为 2 分频、0xFF 为 510 分频。
-------	----	-------------	--

MMC_CLKENA

MMC_CLKENA 为 MMC 接口时钟使能寄存器。只有当 MMC_CMD[start_cmd]和 MMC_CMD[Update_clk_only]置 1，该寄存器的值才会被载入。

	Offset Address	Register Name	Total Reset Value																													
	0x010	MMC_CLKENA	0x0000_0000																													
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
Name	reserved																clk_low_power	reserved										clk_enable				
Reset	0 0																															
Bits	Access	Name	Description																													
[31:17]	-	reserved	保留。																													
[16]	RW	clk_low_power	卡的低功耗控制。 0: 非低功耗模式； 1: 低功耗模式。当卡处于 IDLE 状态，控制器自动关闭接口时钟。该功能一般只使用在 MMC、SD 存储卡，对于 SDIO 设备，为了能检测到 SDIO 中断，不可关闭接口时钟。																													
[15:1]	-	reserved	保留。																													
[0]	RW	clk_enable	卡的时钟使能。 0: 禁止； 1: 使能。																													

MMC_TMOUT

MMC_TMOUT 为超时参数寄存器。用于数据读操作、命令响应中超时参数配置。

	Offset Address	Register Name	Total Reset Value
	0x014	MMC_TMOUT	0xFFFF_FF40
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	data_timeout		response_timeout
Reset	1 0 1 0 0 0 0 0 0		
Bits	Access	Name	Description
[31:8]	RW	data_timeout	数据读操作超时参数，该值也做 data starvation 中断超时参数。单位为接口时钟周期，建议设置为 0xFF_FFFF。
[7:0]	RW	response_timeout	响应超时参数。单位为接口时钟周期，建议值为 0xFF。

MMC_CTYPE

MMC_CTYPE 为接口位宽配置寄存器，用于配置控制器工作在 1bit 数据位宽或 4bit 数据位宽。控制器与卡的位宽应该保持一致。

	Offset Address	Register Name	Total Reset Value
	0x018	MMC_CTYPE	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		card_width
Reset	0 0		
Bits	Access	Name	Description
[31:1]	-	reserved	保留。其中第 16 位不可写 1。
[0]	RW	card_width	配置卡接口的总线宽度。 0: 1bit 模式; 1: 4bit 模式。

MMC_BLKSIZE

MMC_BLKSIZE 为块大小寄存器，SD/MMC 卡一般为 512 字节，SDIO 设备可根据应用需求自定义。



	Offset Address	Register Name	Total Reset Value																							
	0x01C	MMC_BLKSI _Z	0x0000_0200																							
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																									
Name	reserved																block_size									
Reset	0 0																0 0									
	Bits	Access	Name	Description																						
	[31:16]	-	reserved	保留。																						
	[15:0]	RW	block_size	块大小配置，如配置为 0x0200，表示块大小为 512 字节。																						

MMC_BYTCNT

MMC_BYTCNT 为传输数据大小寄存器。若传输数据大小等于块大小，为单块数据传输；若传输数据大小等于块大小的整数倍，为多块数据传输。传输数据大小必须为块大小的整数倍。

	Offset Address	Register Name	Total Reset Value																													
	0x020	MMC_BYTCNT	0x0000_0200																													
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
Name	byte_count																															
Reset	0 0																0 0															
	Bits	Access	Name	Description																												
	[31:0]	RW	byte_count	待传输数据字节数，如配置为 0x0200，表示待传输数据为 512 字节。块传输方式时应该设置为 block size 的整数倍。																												

MMC_INTMASK

MMC_INTMASK 为中断屏蔽寄存器，用于屏蔽 MMC_RINTSTS 对应位中断请求。



Offset Address		Register Name		Total Reset Value																													
0x024		MMC_INTMASK		0x0000_0000																													
Bit																																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved															sdio_int_mask	ebe_int_mask	acd_int_mask	sbe_int_mask	hle_int_mask	frun_int_mask	hto_int_mask	drto_int_mask	rto_int_mask	dere_int_mask	rrec_int_mask	rxdr_int_mask	txdr_int_mask	dto_int_mask	cd_int_mask	re_int_mask	reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																														
[31:17]	-	reserved	保留。																														
[16]	RW	sdio_int_mask	SDIO 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[15]	RW	ebe_int_mask	EBE (End-bit error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[14]	RW	acd_int_mask	ACD (Auto Command Done) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[13]	RW	sbe_int_mask	SBE (Start-bit Error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[12]	RW	hle_int_mask	HLE (Hardware Locked Write Error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[11]	RW	frun_int_mask	FRUN (FIFO underrun/overrun error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[10]	RW	hto_int_mask	HTO (Data starvation-by-host timeout) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														
[9]	RW	drto_int_mask	DRTO (Data Read Timeout) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。																														



[8]	RW	rto_int_mask	RTO (Response Timeout) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[7]	RW	dcrc_int_mask	DCRC (Data CRC error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[6]	RW	rcrc_int_mask	RCRC (Response CRC error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[5]	RW	rxdr_int_mask	RXDR (Receive FIFO data request) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[4]	RW	txdr_int_mask	TXDR (Transmit FIFO data request) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[3]	RW	dto_int_mask	DTO (Data Transfer Over) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[2]	RW	cd_int_mask	CD (Command done) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[1]	RW	re_int_mask	RE (Response error) 中断屏蔽。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[0]	-	reserved	保留。

MMC_CMDARG

MMC_CMDARG 为指令参数寄存器。寄存器 `MMC_CMD[5:0]` 指定指令序号所对应的指令参数。如 `cmd17` 单块数据读操作, `MMC_CMD[5:0]=17`, 指令参数寄存器应配置为卡内空间地址。各指令序号对应指令参数请参考 SD/MMC/SDIO 协议。

Offset Address		Register Name		Total Reset Value		
0x028		MMC_CMDARG		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	cmd_arg					
Reset	0 0					
Bits	Access	Name	Description			
31:0	RW	cmd_arg	指令参数配置。			

MMC_CMD

MMC_CMD 为指令寄存器，用于指定指令特征，如序号、响应类型、是否进行数据传输、数据传输方向、传输模式等。

Offset Address		Register Name		Total Reset Value		
0x02C		MMC_CMD		0x0000_0000		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Name	start_cmd	reserved	update_clk_reg_only	card_number	send_initialization stop_abort_cmd wait_prvdata_complete send_auto_stop transfer_mode read/write data_transfer_expected check_reponse_crc response_length response_expect	cmd_index
Reset	0 0					
Bits	Access	Name	Description			
[31]	RW	start_cmd	指令执行或加载接口时钟参数启动位。 与 update_clk_reg_only 位配合使用，在 update_clk_reg_only 为 0 时将该位置 1，用于启动指令执行；在 update_clk_reg_only 为 1 时将该位置 1，用于加载接口时钟参数。 当指令被执行或时钟参数已载入，该比特自动清 0。当该比特为 1 时，CPU 不允许修改时钟和指令相关的寄存器。如果修改，hardware lock error 中断就会产生。			
[30:22]	-	reserved	保留。不可写 1。			
[21]	RW	update_clk_reg_only	0：正常指令顺序； 1：不发送命令，重新载入接口时钟控制寄存器（MMC_CLKDIV，MMC_CLKENA）的值。在不需要发指令给卡的前提下，用来调整接口时钟的频率和控制接口时钟的开关。			



			<p>在正常命令顺序中，即该位置为 0，下面的寄存器值会被控制器载入：MMC_CMD，MMC_CMDARG，MMC_TMOUT，MMC_CTYPE，MMC_BLKSIZE，MMC_BYTCNT。控制器将新的寄存器值使用到新的指令中去。</p> <p>如果该比特设为 1，指令不会传到卡中，不会产生 Command Done 中断。</p>
[20:16]	RW	card_number	正在使用的卡的序号，应设置为 0。
[15]	RW	send_initialization	<p>0：在发送该指令前不要送出初始序列；</p> <p>1：在发送该指令前送出初始序列（80 个时钟周期）。</p> <p>在上电以后，卡需要 80 个时钟周期进行初始化；因此需要在往卡发送第一条指令时需设置该位。</p>
[14]	RW	stop_abort_cmd	<p>数据读写状态机复位控制位。</p> <p>0：无影响；</p> <p>1：在数据传输过程中发送停止指令异常中止本次传输时，将该位置 1 用于将控制器从数据传输状态恢复到 IDLE 状态。</p>
[13]	RW	wait_prvdata_complete	<p>0：立即发送指令（即使前一个数据传输还没完成）；</p> <p>1：等到前一个数据传输完成后才开始发送指令。</p> <p>在指令发送时将该位配置为 0 可用来在数据传输时读取卡状态或中止本次传输。</p>
[12]	RW	send_auto_stop	<p>0：数据传完以后不会发停止指令；</p> <p>1：数据传完以后发送停止指令。</p> <p>当被使能以后，控制器在每次数据传输完以后就会自动送出停止指令。</p> <p>predefined block count 方式、cmd53 等不要停止指令。</p> <p>open-ended 传输方式需要停止指令。</p> <p>在非数据传输时，该位被忽略。</p>
[11]	RW	transfer_mode	<p>0：用于块读写指令；</p> <p>1：用于流数据读写指令。</p> <p>在非数据传输时，该位被忽略。</p>
[10]	RW	read/write	<p>0：从卡读取数据；</p> <p>1：往卡写数据。</p> <p>在非数据传输时，该位被忽略。</p>
[9]	RW	data_transfer_expected	<p>0：非数据指令；</p> <p>1：数据指令。</p>
[8]	RW	check_response_crc	<p>0：不检查指令响应 CRC；</p> <p>1：检查指令响应 CRC。</p> <p>一些指令回复没有返回有效的 CRC。为了禁止控制器对 CRC</p>



			进行检查，软件需要针对这些指令禁止该功能。
[7]	RW	response_length	0: 短响应指令; 1: 长响应指令。
[6]	RW	response_expect	0: 无响应指令; 1: 有响应指令。
[5:0]	RW	cmd_index	指令序号。

MMC_RESP0

MMC_RESP0 为指令响应寄存器 0。

	Offset Address	Register Name	Total Reset Value
	0x030	MMC_RESP0	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	response0		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	response0	48bit 短响应的 bit[39:8]或 136 比特长响应的 bit[31:0]。

MMC_RESP1

MMC_RESP1 为指令响应寄存器 1。

	Offset Address	Register Name	Total Reset Value
	0x034	MMC_RESP1	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	response1		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RO	response1	指令长响应 bit[63:32]或控制器 auto-stop 指令的响应。 当控制器发出 auto-stop 指令，响应的 bit[39:8]就会被保存在该寄存器。上一个指令的响应仍然会被保存在 MMC_RESP0 寄存器内。Auto-stop 只供数据传输使用，而且回复的类型总是短响应。



MMC_RESP2

MMC_RESP2 为指令响应寄存器 2。

	Offset Address	Register Name	Total Reset Value
	0x038	MMC_RESP2	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	response2		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	response2
			Description
			指令长响应的 bit[95:64]。

MMC_RESP3

MMC_RESP3 为指令响应寄存器 3。

	Offset Address	Register Name	Total Reset Value
	0x03C	MMC_RESP3	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	response3		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	response3
			Description
			指令长响应 bit[127:96]。

MMC_MINTSTS

MMC_MINTSTS 为屏蔽后的中断状态寄存器。MMC_MINTSTS = MMC_RINTSTS & MMC_INTMASK。只有当 MMC_RINTSTS 和 MMC_INTMASK 对应位均为 1，且 MMC_CTRL[int_enable]为 1 时，中断才会上报。



Offset Address		Register Name		Total Reset Value																																										
0x040		MMC_MINTSTS		0x0000_0000																																										
Bit																																														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
Name	reserved																sdio_int	ebe_int	acd_int	sbe_int	hle_int	frun_int	hto_int	drto_int	rto_int	dere_int	rerc_int	rxdr_int	txdr_int	dto_int	cd_int	re_int	reserved													
Reset	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description																																											
[31:17]	-	reserved	保留。																																											
[16]	RO	sdio_int	屏蔽后的 SDIO 中断。 0: 未产生中断; 1: 已产生中断。																																											
[15]	RO	ebe_int	屏蔽后的 EBE (End-bit error (read)/Write no CRC) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[14]	RO	acd_int	屏蔽后的 ACD (Auto Command Done) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[13]	RO	sbe_int	屏蔽后的 SBE (Start-bit Error) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[12]	RO	hle_int	屏蔽后的 HLE (Hardware Locked Write Error) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[11]	RO	frun_int	屏蔽后的 FRUN (FIFO underrun/overflow error) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[10]	RO	hto_int	屏蔽后的 HTO (Data starvation-by-host timeout) 中断。 0: 未产生中断; 1: 已产生中断。																																											
[9]	RO	drto_int	屏蔽后的 DRTO (Data Read Timeout) 中断。 0: 未产生中断; 1: 已产生中断。																																											



[8]	RO	rto_int	屏蔽后 RTO (Response Timeout) 中断。 0: 未产生中断; 1: 已产生中断。
[7]	RO	dcrc_int	屏蔽后的 DCRC (Data CRC error) 中断。 0: 未产生中断; 1: 已产生中断。
[6]	RO	rcrc_int	屏蔽后的 RCRC (Response CRC error) 中断。 0: 未产生中断; 1: 已产生中断。
[5]	RO	rxdr_int	屏蔽后的 RXDR (Receive FIFO data request) 中断。 0: 未产生中断; 1: 已产生中断。
[4]	RO	txdr_int	屏蔽后的 TXDR (Transmit FIFO data request) 中断。 0: 未产生中断; 1: 已产生中断。
[3]	RO	dto_int	屏蔽后的 DTO (Data Transfer Over) 中断。 0: 未产生中断; 1: 已产生中断。
[2]	RO	cd_int	屏蔽后的 CD (Command done) 中断。 0: 未产生中断; 1: 已产生中断。
[1]	RO	re_int	屏蔽后的 RE (Response error) 中断。 0: 未产生中断; 1: 已产生中断。
[0]	-	reserved	保留。

MMC_RINTSTS

MMC_RINTSTS 为原始中断状态寄存器。对应位写 1 清 0，写 0 无效，即写 1 表示清中断。



Offset Address	Register Name	Total Reset Value	
0x044	MMC_RINTSTS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved sdio_int_status ebe_int_status acd_int_status sbe_int_status hle_int_status frun_int_status hto_int_status drto_int_status rto_int_status dorc_int_status rerc_int_status rxdr_int_status txdr_int_status dto_int_status cd_int_status re_int_status reserved		
Reset	0 0		
Bits	Access	Name	Description
[31:17]	-	reserved	保留。
[16]	RW	sdio_int_status	屏蔽前的 SDIO 中断。 0: 未产生中断; 1: 已产生中断。
[15]	RW	ebe_int_status	屏蔽前的 EBE (End-bit error (read)/Write no CRC) 中断。 0: 未产生中断; 1: 读操作时出现 end bit 错误, 写操作时卡未返回 CRC 状态或返回负的 CRC 状态时, 该中断产生。
[14]	RW	acd_int_status	屏蔽前的 ACD (Auto Command Done) 中断。 0: 未产生中断; 1: 控制器自动发送的 auto-stop 指令执行中断。
[13]	RW	sbe_int_status	屏蔽前的 SBE (Start-bit Error) 中断。 0: 未产生中断; 1: 当从卡读取数据时, 数据的 start bit 错误中断。在 4bit 模式下, 如果数据都没有 start bit, 该中断产生。
[12]	RW	hle_int_status	屏蔽前的 HLE (Hardware Locked Write Error) 中断。 0: 未产生中断; 1: 在硬件锁定寄存器值时, 仍试图对这些寄存器进行写操作。
[11]	RW	frun_int_status	屏蔽前的 FRUN (FIFO underrun/overflow error) 中断。 0: 未产生中断; 1: 当 FIFO 满时仍往 FIFO 写数据, 或 FIFO 空时仍从 FIFO 读数据时该中断产生。
[10]	RW	hto_int_status	屏蔽前的 Data starvation-by-host timeout (HTO) 中断。 0: 未产生中断; 1: 为了防止数据丢失, 如果在发送数据给卡时 FIFO 是空的, 或从卡接收数据时 FIFO 是满的, 控制器输出时钟



			(MMC_CCLK) 会被停止。当时钟被停止后, data-starvation 计数器就会启动。如果计数器数满溢出, 而此时系统仍然没有往空的 FIFO 写数据或从满的 FIFO 读数据, 该中断产生。这时候, 需要系统对 FIFO 的进行读写操作, 输出时钟才会重新启动。
[9]	RW	drto_int_status	屏蔽前的 DRTO (Data Read Timeout) 中断。 0: 未产生中断; 1: 接收数据超时中断。
[8]	RW	rto_int_status	屏蔽前 RTO (Response Timeout) 中断。 0: 未产生中断; 1: 指令响应超时中断 (未收到响应)。
[7]	RW	dcrc_int_status	屏蔽前的 DCRC (Data CRC error) 中断。 0: 未产生中断; 1: 接收数据 CRC 校验错误中断。
[6]	RW	rcrc_int_status	屏蔽前的 RCRC (Response CRC error) 中断。 0: 未产生中断; 1: 指令响应 CRC 校验错误中断。
[5]	RW	rxdr_int_status	屏蔽前的 RXDR (Receive FIFO data request) 中断。 0: 未产生中断; 1: 当从卡读数据时, FIFO 中数据多于 FIFO 读阈值 rx_wmark 时产生中断。使用 DMA 进行数据传输时, 应屏蔽该中断。
[4]	RW	txdr_int_status	屏蔽前的 TXDR (Transmit FIFO data request) 中断。 0: 未产生中断; 1: 当往卡写数据时, FIFO 中数据少于或等于 FIFO 写阈值 tx_wmark 时产生中断。使用 DMA 进行数据传输时, 应屏蔽该中断。
[3]	RW	dto_int_status	屏蔽后的 DTO (Data Transfer Over) 中断。 0: 未产生中断; 1: 数据传输完毕中断。即使出现 Start Bit Error、CRC error 或 Read Data Timeout, 该中断仍然产生。
[2]	RW	cd_int_status	屏蔽前的 CD (Command done) 中断。 0: 未产生中断; 1: 命令执行完毕并收到响应产生中断。即使出现 response error、response CRC error 或 response timeout, 该中断仍然产生。

[1]	RW	re_int_status	屏蔽前的 RE (Response error) 中断。 0: 未产生中断; 1: 收到的指令响应有错误时该中断产生。
[0]	-	reserved	保留。

MMC_STATUS

MMC_STATUS 为 MMC 状态寄存器，反映控制器工作状态。

Offset Address	Register Name	Total Reset Value	
0x048	MMC_STATUS	0x0000_0000	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	<div style="display: flex; justify-content: space-between;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">dma_req dma_ack</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">fifo_count</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">resp_index</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">data_fsm_busy data_busy data_3_status</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">cmd_fsm_state</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">fifo_full fifo_empty fifo_tx_watermark fifo_rx_watermark</div> </div>		
Reset	0 0		
Bits	Access	Name	Description
[31]	RO	dma_req	DMA 请求状态位。 0: 控制器未请求 DMA 传输; 1: 控制器正请求 DMA 传输。
[30]	RO	dma_ack	DMA 确认状态位。 0: DMAC 未清除 MMC 控制器请求; 1: DMAC 清除 MMC 控制器请求。
[29:17]	RO	fifo_count	FIFO 已有数据数量，以 word 为单位。
[16:11]	RO	resp_index	上一个指令响应的序号。包括 auto-stop 指令的响应。
[10]	RO	data_fsm_busy	数据发送/接收状态机的状态。 0: 数据发送/接收状态机处于闲置状态; 1: 数据发送/接收状态机处于繁忙状态。
[9]	RO	data_busy	0: 卡处于闲置状态; 1: 执行写或擦除操作后卡处于繁忙状态。 该状态位直接反映卡数据线 data[0]信号的取反。执行写或擦除等操作后需由软件查询该位，直至该位由 1 变为 0 后，才能对卡进行下一步的操作。
[8]	RO	data_3_status	该位直接反映卡数据线 data[3]信号状态。



[7:4]	RO	cmd_fsm_state	<p>控制器命令状态机状态。</p> <p>0x0: Idle;</p> <p>0x1: Send init sequence;</p> <p>0x2: Tx cmd start bit;</p> <p>0x3: Tx cmd tx bit;</p> <p>0x4: Tx cmd index +arg;</p> <p>0x5: Tx cmd crc7;</p> <p>0x6: Tx cmd end bit;</p> <p>0x7: Rx resp start bit;</p> <p>0x8: Rx resp IRQ response;</p> <p>0x9: Rx resp tx bit;</p> <p>0xA: Rx resp cmd idx;</p> <p>0xB: Rx resp data;</p> <p>0xC: Rx resp crc7;</p> <p>0xD: Rx resp end bit;</p> <p>0xE: Cmd path wait NCC;</p> <p>0xF: Wait, CMD-to-response turnaround。</p>
[3]	RO	fifo_full	<p>FIFO 满状态标志。</p> <p>0: FIFO 不满。</p> <p>1: FIFO 满。</p>
[2]	RO	fifo_empty	<p>FIFO 空状态标志。</p> <p>0: FIFO 非空。</p> <p>1: FIFO 空。</p>
[1]	RO	fifo_tx_watermark	FIFO 内数据个数小于写阈值 tx_wmark。
[0]	RO	fifo_rx_watermark	FIFO 内数据大于或等于读阈值 rx_wmark。

MMC_FIFOTH

MMC_FIFOTH 为 MMC FIFO 参数寄存器，建议配置值为 0x2007_0008。用 DMA 进行数据传输时，应将 DMA 与控制器的 burst 设置为同样大小，当数据正以 DMA 模式传输时不要改变该寄存器值。

Offset Address		Register Name	Total Reset Value
0x04C		MMC_FIFOTH	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved burst_size rx_wmark reserved tx_wmark		
Reset	0 0		
Bits	Access	Name	Description
[31]	-	reserved	保留。
[30:28]	RW	burst_size	<p>DMA 一次 burst 传输数据量，以 word 为单位。应该被设置为跟 DMAC 相同的大小。</p> <p>000: 1; 001: 4; 010: 8。 其它: 保留。</p> <p>允许使用的 burst_size 和 tx_wmark 组合为:</p> <p>Burst_size = 1, Tx_wmark = 1~15; Burst_size = 4, Tx_wmark = 4; Burst_size = 4, Tx_wmark = 4; Burst_size = 4, Tx_wmark = 12; Burst_size = 8, Tx_wmark = 8; Burst_size = 8, Tx_wmark = 4。</p> <p>允许使用的 burst_size 和 rx_wmark 组合为:</p> <p>Burst_size = 1, Rx_wmark = 0~14; Burst_size = 4, Rx_wmark = 3; Burst_size = 4, Rx_wmark = 7; Burst_size = 4, Rx_wmark = 11; Burst_size = 8, Rx_wmark = 7; Burst_size = 8, Rx_wmark = 11。</p>
[27:16]	RW	rx_wmark	<p>读数据时的 FIFO 阈值。当 FIFO 已有数据个数大于该值时，产生 DMA 请求；若中断使能，则产生中断请求。</p> <p>非 DMA 模式下 receive FIFO data request (RXDR) 中断会被使能并产生中断请求。在数据传输末尾，如果 FIFO 计数没有大于该值，不会产生中断。在 Data Transfer Over 中断产生后来由软件完成剩余数据的读取。</p> <p>DMA 模式下，在数据传输末尾，当剩余的数据比阈值低，DMA 仍会在 Data Transfer Over 中断产生之前通过 burst 方式把数据读取。</p>



			限制: $Tx_wmark \leq FIFO_DEPTH - 2$ 建议值: $(FIFO_DEPTH/2) - 1$; 即多于 $(FIFO_DEPTH / 2) - 1$ 时发出请求。
[15:12]	-	resevered	保留。
[11:0]	RW	tx_wmark	发送数据时的 FIFO 阈值。当 FIFO 已有数据个数小于或等于该值时, 产生 DMA 请求; 若中断使能, 则产生中断请求。 非 DMA 模式下 transmit FIFO data request (TXDR) 中断会被使能并产生中断请求。在数据传输末尾, 中断产生后由软件完成剩余字节的传送。 DMA 模式下, 在数据传输末尾, 若剩余的数据小于 burst size, DMA 仍通过 burst 方式传送完成剩余数据的传输。 限制: $Tx_wmark \geq 1$ 建议值: $FIFO_DEPTH/2$, 即小于或等于 $FIFO_DEPTH/2$ 时发出请求。

MMC_TCBCNT

MMC_TCBCNT 为接口传输计数寄存器, 用于统计完成一次数据传输指令后接口传输的数据字节数。在数据传输过程中, 该寄存器返回值为 0; 数据传输结束后该寄存器反映控制器与卡之间数据传输字节数。该寄存器在启动一次新的数据传输指令时清 0。

	Offset Address	Register Name	Total Reset Value
	0x05c	MMC_TCBCNT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	trans_card_byte_coun		
Reset	0 0		
Bits	Access	Name	Description
[31: 0]	RO	trans_card_byte_coun	控制器与卡之间数据传输字节数。

MMC_TBBCNT

MMC_TBBCNT 为 FIFO 传输计数寄存器。用于实时统计执行数据传输指令时 CPU/DMA 与控制器 FIFO 之间传输的数据字节数, 在数据传输过程中动态变化。该寄存器在启动一次新的数据传输指令时清 0。



	Offset Address	Register Name	Total Reset Value
	0x060	MMC_TBBCNT	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	trans_fifo_byte_count		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RO	trans_fifo_byte_count
	Description		
	CPU/DMA 与控制器 FIFO 之间传送数据的字节数。		

MMC_DATA

MMC_DATA 为数据寄存器，为 FIFO 入口地址。在读写 FIFO 时，应先读取 [MMC_STATUS\[fifo_count\]](#) 得到 FIFO 剩余空间，以此确定读写的数据量，以免造成 FIFO 溢出。

	Offset Address	Register Name	Total Reset Value
	0x100	MMC_DATA	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	fifo_entrance		
Reset	0 0		
	Bits	Access	Name
	[31:0]	RW	fifo_entrance
	Description		
	FIFO 入口地址。		



9 PCI

9.1 概述

PCI (Peripheral Component Interconnect) 总线是一种通用的本地总线 (Local Bus)。PCI 接口以其灵活的可扩展性用于实现产品的不同应用形态和应用场合, 实现多设备级联和扩展符合 PCI/miniPCI 接口的外设, 如 SATA、WiFi、PCI-to-PCI Bridge 等。

Hi3520 的 PCI 接口用来实现片外 PCI 总线到片内 AHB (Advanced High-performance Bus) 总线的转换, 符合 PCI2.3 总线协议, 并兼容 miniPCI 接口协议。Hi3520 的 PCI 接口有两种应用模式, 可分别实现 PCI Host 和 PCI Device 功能:

- Host 模式
当配置为 Host 模式时, 可用作整个 PCI 总线的主控设备, 对整个 PCI 总线进行配置管理和仲裁。
- Device 模式
当配置为 Device 模式时, 可用于实现 PCI Device 功能, 通过 PCI 总线和 PCI Host, 以及 PCI 总线上其它 PCI 设备进行通信。

9.2 特点

Hi3520 的 PCI 接口有以下特点:

- 支持 PCI2.3 协议。
- 总线位宽 32bit, 总线频率最高可达 66MHz。
- 支持 PCI Host 和 PCI Device 功能:
 - PCI Host 模式下
 - 支持 PCI_INTA、PCI_INTB 中断。
 - 支持 memory read/write, configuration read/write 命令, 配置访问时可支持 Type0 和 Type1 配置命令。
 - 内建 PCI Bus Arbiter, 且最多支持 5 个 PCI 设备的总线仲裁。
 - PCI Device 模式下:
 - 支持 PCI_INTA 中断。



- 支持 memory read/write, I/O read/write, configuration read/write 命令。
- 支持 PCI 侧到 AHB 侧的地址翻译。
- 支持用户可编程的 doorbell 中断。
- 支持 PCI 总线奇偶校验。
- 支持对 memory 的 prefetchable 访问和 non-prefetchable 访问。

9.3 信号描述

PCI 接口信号如表 9-1 所示。

表9-1 PCI 接口信号描述

信号名称	方向	描述	对应管脚
PCI_CLK	I/O	PCI 总线时钟信号。 • PCI Host 模式下, PCI 时钟信号可由芯片内部产生, PCI_CLK 为输出信号, 将片内产生的 PCI 时钟输出给总线上的其它设备使用; 在 PCI Host 模式下 PCI 时钟也可选择为由外部时钟源输入, 片内不产生 PCI 时钟, PCI_CLK 为输入信号。 • PCI Device 模式下, PCI_CLK 则固定为输入信号。	PCICLK
PCI_RST	I/O	PCI 总线复位信号。 • PCI Host 模式下, PCI_RST 由片内产生, 并输出到总线上供其它设备使用, PCI_RST 固定为输出信号。 • PCI Device 模式下, PCI_RST 固定为输入信号。	PCIRSTN
PCI_AD[31:0]	I/O	PCI 总线地址/数据信号。	PCIAD31~PCIAD0
PCI_CBE[3:0]	I/O	PCI 总线命令/字节使能信号。	PCICBE3~PCICBE0
PCI_FRAME	I/O	PCI 总线 frame 信号。	PCIFRAMEN
PCI_IRDY	I/O	PCI 总线 initiator ready 信号。	PCIIRDYN
PCI_TRDY	I/O	PCI 总线 target ready 信号。	PCITRDYN
PCI_DEVSEL	I/O	PCI 总线 device select 信号。	PCIDEVSELN
PCI_STOP	I/O	PCI 总线 stop 信号。	PCISTOPN



信号名称	方向	描述	对应管脚
PCI_IDSEL	I	PCI 总线 initial device select 信号。 • PCI Device 模式下为 PCI_IDSEL。 • PCI Host 模式下为 PCI_INTB。	PCIIDSEL
PCI_INTB	I	PCI 总线 INTB 信号	PCIIDSEL
PCI_PAR	I/O	PCI 总线 parity 信号。	PCIPAR
PCI_SERR	I/O	PCI 总线 system error 信号。	PCISERRN
PCI_PERR	I/O	PCI 总线 parity error 信号。	PCIPERRN
PCI_INTA	I/O	PCI 总线 INTA 信号。 • PCI Host 模式下，PCI_INTA 为输入信号。 • PCI Device 模式下，PCI_INTA 为输出信号。	PCIINTAN
PCI_REQ[4:0]	I	PCI 总线 bus request 信号。 • PCI Host 模式下，PCI_REQ[4:0]为输入信号。 • PCI Device 模式下，PCI_REQ[0]被复用作 PCI 的总线仲裁信号 PCI_GNT，PCI_REQ[4]被复用作 GPIO3_7。 说明 当 Hi3520 为 PCI Device 模式时，不会使用 PCI_REQ[3:1]，此时需要对管脚进行上拉处理，将其固定接为高电平。	PCIREQ4N~ PCIREQ0N
PCI_GNT[4:0]	O	PCI 总线 bus grant 信号。 • PCI Host 模式下，PCI_GNT[4:0]为输出信号。 • PCI Device 模式下，PCI_GNT[0]被复用作 PCI 的总线申请信号 PCI_REQ；PCI_GNT[4]被复用作 GPIO7_1。 说明 当 Hi3520 为 PCI Device 模式时，不会使用 PCI_REQ [3:1]，此时需要对管脚悬空。	PCIGRANT4N~ PCIGRANT0N

注：PCI 接口信号中 PCI_FRAME、PCI_IRDY、PCI_TRDY、PCI_STOP、PCI_DEVSEL、PCI_SERR、PCI_PERR、PCI_INTA 和 PCI_INTB 在实际应用需做上拉处理。

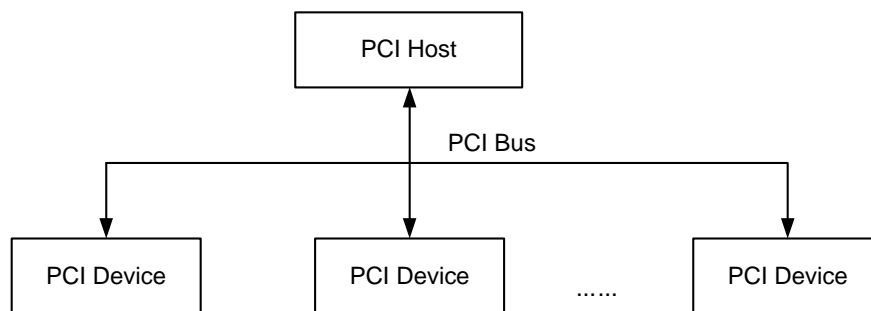


9.4 功能描述

典型应用

PCI 总线典型应用模式如图 9-1 所示。

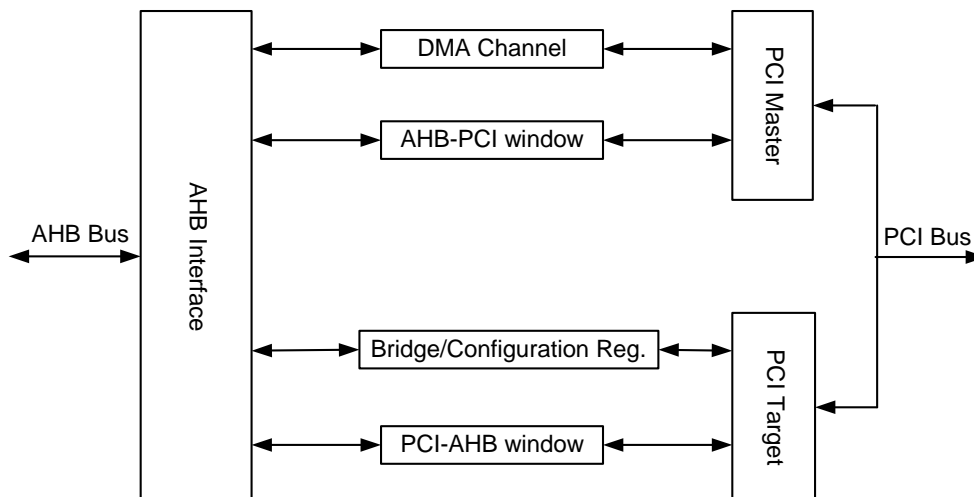
图9-1 PCI 总线典型应用



PCI 总线是一种并行的本地总线，具有总线带宽高，扩展性强的特点。因此，基于 PCI 接口可以非常灵活地实现芯片间的级联扩展以及各种其它外设的扩展应用。

Hi3520 PCI 模块架构如图 9-2 所示。

图9-2 Hi3520 PCI 模块架构示意图



Hi3520 PCI 模块内建 DMAC，DMA 通道可用于以 DMA 方式进行数据的搬运。除此之外，ARM core 也可以通过 AHB-PCI window 访问 PCI 总线上的其它 PCI 设备。PCI 总线上其它 PCI 设备通过 PCI-AHB window 来访问 Hi3520。



功能原理

PCI 总线的主要信号包括地址/数据信号，命令/字节使能信号，以及接口控制信号等。基于不同的应用模式，PCI 总线操作主要有以下几种：memory 操作、I/O 操作和配置操作。各种操作的时序图如图 9-3~图 9-5 所示。

图9-3 PCI 总线 memory (I/O) 读操作时序图

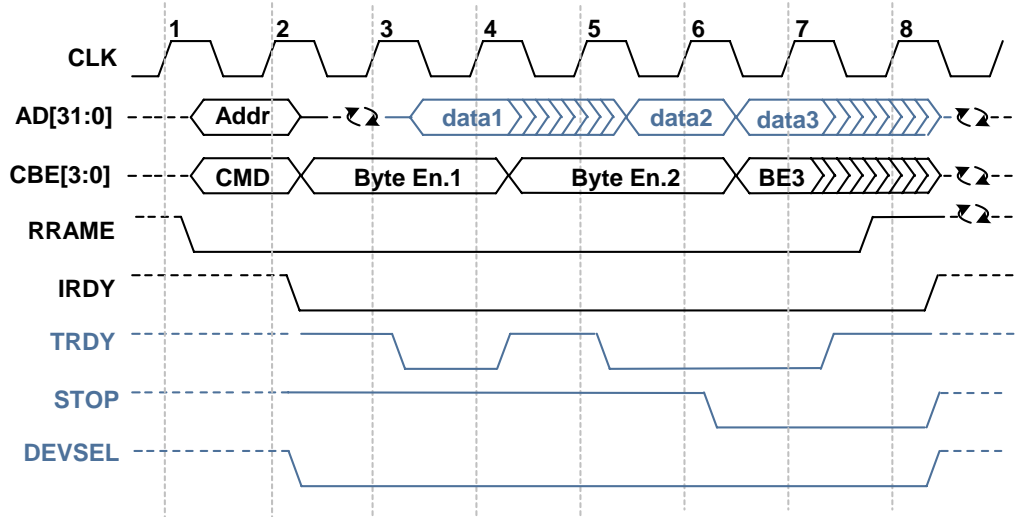


图9-4 PCI 总线 memory (I/O) 写操作时序图

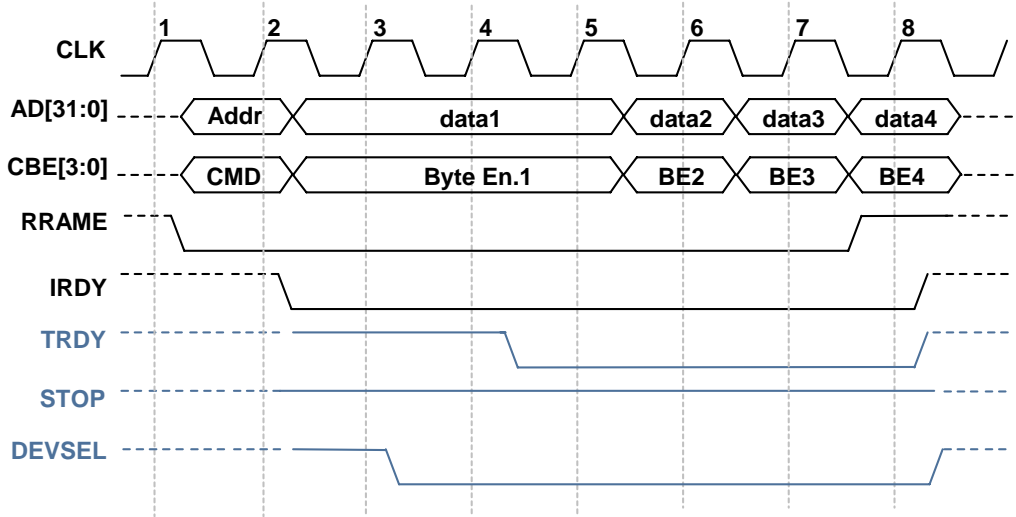
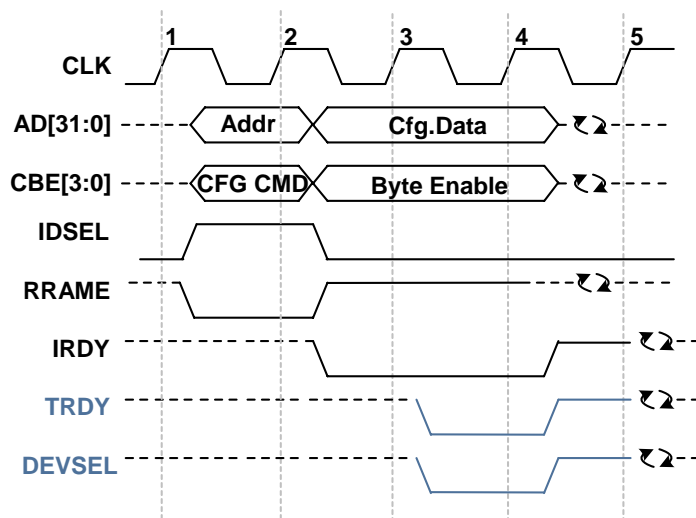




图9-5 PCI 总线配置访问时序图



9.5 工作方式

9.5.1 管脚复用配置

PCI 接口管脚复用关系如下：

- 当 Hi3520 为 PCI Device 模式时，管脚复用关系如下：
 - PCIREQ0N 复用为 PCI 总线仲裁信号 PCIGRANT_SLAVEN。
 - PCIGRANT0N 复用为 PCI 总线申请信号 PCIREQ_SLAVEN。
 - 管脚 PCIREQ3N、PCIREQ4N、PCIGRANT3N 和 PCIGRANT4N 分别与 GPIO3_6、GPIO3_7、GPIO7_0 和 GPIO7_1 复用。
- 当 Hi3520 为 PCI Host 模式时，PCIIDSEL 管脚为 PCI_INTB 功能。
- 管脚 PCIREQ3N、PCIREQ4N、PCIGRANT3N 和 PCIGRANT4N 与 GPIO 的复用关系可以通过 IO_Config 的 reg78~reg81 寄存器进行配置。上电缺省为选择的是 GPIO，当 PCI 需要工作在 PCI HOST 模式下时，先配置 IO_Config 寄存器将管脚配置成 PCI 的相应功能，并对 PCIREQ3N 和 PCIREQ4N 上拉，再配置 SC_PERCTRL11 bit[0]为 1，将 PCI 配置成 Host 模式。

说明

- 当 Hi3520 为 PCI Device 模式时，PCIIDSEL、PCIREQ0N 和 PCIGRANT0N 的复用是自动的，不需要配置 IO Config 寄存器。
- 当 Hi3520 为 PCI Host 模式时，PCIIDSEL 的复用是自动的，不需要配置 IO Config 寄存器。

9.5.2 时钟门控

在当前 PCI 总线处于空闲状态时，可通过配置系统控制寄存器来关断 PCI 时钟。

- 当 Hi3520 为 PCI Host 模式时，PCI 时钟可由如下两种方式产生：



- 内部产生：PCI 时钟是由 Hi3520 内部时钟模块产生，除了 Hi3520 PCI 模块使用之外还从芯片管脚输出供其它 PCI 设备使用。此时如果 Hi3520 关断 PCI 时钟，将只关断自己 PCI 模块的时钟，不会影响送往片外的时钟。
- 外部产生：Hi3520 的时钟是由外部时钟源输入，Hi3520 将只关断 PCI 模块的时钟。
- 当 Hi3520 为 PCI Device 模式时，PCI 时钟固定为 PCI 总线时钟输入，此时 Hi3520 将只关断自己 PCI 模块的时钟，不会影响到其它 PCI 设备。

关于 Hi3520 PCI 时钟门控的配置，详见 SC_PEREN bit[14]和 SC_PERDIS bit[14]配置说明。

9.5.3 时钟配置



说明

系统控制寄存器 SC_PERCTRL11 bit[1]只有在 PCI Host 模式下才可配置。

通过配置系统控制寄存器 SC_PERCTRL11 bit[1]，可控制芯片 PCI 时钟是由片内产生还是由外部时钟源输入：

- 当 SC_PERCTRL11 bit[1]为 1 时，PCICLK 为输出，此时芯片内部产生 PCI 时钟通过 PCICLK 管脚送出到片外供整个 PCI 总线使用。
- 当 SC_PERCTRL11 bit[1]为 0 时，PCICLK 为输入，此时整个 PCI 总线的时钟都由外部时钟源提供。

在 PCI Device 模式下，SC_PERCTRL11 bit[1]固定配置为 0，即固定为外部输入 PCI 时钟。

当 Hi3520 为 PCI Host 模式且 PCI 时钟是由 Hi3520 片内产生时，可通过配置系统控制寄存器 SC_PERCTRL9 bit[24:23]来选择所产生 PCI 时钟的频率。具体的配置请参见 SC_PERCTRL9 bit[24:23]的配置说明。

9.5.4 软复位

通过配置系统控制器 SC_PERCTRL8 bit[13]为 1，可实现对整个 PCI 模块的单独软复位。特别需要注意的是，当作为 PCI Host 模式时，此时对 PCI Host 的软复位会通过 PCIRST 管脚将整个 PCI 总线上的所有设备都复位。因此，在对 PCI Host 进行软复位后，要对所有 PCI Device 重新进行初始化。

当作为 PCI Device 模式时，配置系统控制器 SC_PERCTRL8 bit[13]为 1 只是对自己的 PCI 模块进行复位，不会对 PCI 总线的其它 PCI 设备产生影响。但是在软复位结束后，同样需要 PCI Host 对复位过的 PCI Device 进行初始化。

对 Hi3520 PCI 模块的一次软复位操作包括配置软复位和撤销软复位。当对系统控制器 SC_PERCTRL8 bit[13] 写入 1 后，PCI 模块进入复位状态，在不小于 4ms 的时间间隔后，对系统控制器 SC_PERCTRL8 bit[13]写入 0，撤销软复位。

9.5.5 工作模式配置



注意

当需要使用复用的 PCIREQ 和 PCIGRANT 时，必须先配置 IO_CONFIG 将管脚配置为 PCI 相应功能，然后再将 Hi3520 切换为 PCI Host 模式。

通过配置系统控制器 SC_PERCTRL11 bit[0]，可实现将 Hi3520 配置为 PCI Host 模式或者 PCI Device 模式：

- 当 SC_PERCTRL11 bit[0] 为 1 时，Hi3520 为 PCI Host 模式。
- 当 SC_PERCTRL11 bit[0] 为 0 时，Hi3520 为 PCI Device 模式。

9.5.6 通过 window 进行数据传输



注意

只有当 Hi3520 作为 PCI Host 模式时才能通过 AHB-PCI window 对 PCI 总线上的其它设备进行访问。

通过 window 进行数据传输时，可相应选择访问 prefetchable 空间还是 non-prefetchable 空间。

Hi3520 可通过 AHB-PCI window 实现 ARM core 对 PCI 总线上其它 PCI 设备的访问，PCI 总线上的其它 PCI 设备也通过 PCI-AHB window 实现对 Hi3520 的访问。

Hi3520 通过 AHB-PCI window 访问 PCI 设备

Hi3520 通过 AHB-PCI window 访问 PCI 设备的步骤如下：

- 步骤 1 Hi3520 作为 PCI Host，初始化 PCI 总线上其它 PCI 设备。
- 步骤 2 ARM core 发起 AHB 总线操作，此时的 AHB 地址就是所要访问的 PCI 设备的目标地址。即 AHB-PCI window 两侧的地址是透传的，不需要地址翻译。

----结束

Hi3520 作为 PCI Master，支持发出 Memory Write、I/O Write、I/O Read 和 Memory Read Multiple 命令。

PCI 设备通过 PCI-AHB window 访问 Hi3520

PCI 设备通过 PCI-AHB window 访问 Hi3520 的步骤如下：

- 步骤 1 PCI Host 设备初始化整个 PCI 总线。
- 步骤 2 PCI 设备发起对 Hi3520 的访问。当 Hi3520 为 PCI Host 模式时，此时的 PCI 地址就是所要访问的 Hi3520 的目标地址，直接透传到 AHB 侧，不需要地址翻译。当 Hi3520 为 PCI Device 模式，PCI 地址需要通过地址翻译才能到达 AHB 侧。



----结束

Hi3520 作为 PCI Target 时，支持如下 PCI 访问命令：Memory Write、Memory Read、Configuration Read with Type0/Type1、Configuration Writer with Type0/Type1 和 Memory Read Multiple。

9.5.7 通过 DMA 通道进行数据传输

Hi3520 PCI 模块内建 DMAC，可直接由 PCI 接口发起 DMA 操作，此时不需要 ARM core 的干预，可获得更好的系统性能。进行数据传输时存在以下两种情况：

- 当 Hi3520 为 PCI Host 模式时，通过 DMA 通道进行数据传输不会引起 PCI 总线中断。
- 当 Hi3520 为 PCI Device 模式时，此时通过 DMA 通道进行数据传输会涉及到 PCI 总线中断。

当 Hi3520 工作在 PCI HOST 模式时，对 PCI 总线的初始化是通过配置访问来实现的。需要注意的是，此时的配置访问都是通过 DMA 方式来完成。

Hi3520 PCI 接口的 DMA 操作命令如表 9-2 所示。

表9-2 PCI 接口信号描述

操作	命令	描述
Read DMA	0010	I/O Read
	0110	Memory Read
	1010	Configuration Read
	1100	Memory Read Multiple
Write DMA	0011	I/O Write
	0111	Memory Write
	1011	Configuration Write

注：表 9-2 中未列出的命令均做保留处理。

当 Hi3520 为 PCI Host 模式时，通过 DMA 通道进行数据传输的步骤如下：

- 步骤 1 配置 CPU_IMASK 寄存器，使能 DMA Read/Write 中断。
- 步骤 2 将数据传输的目标地址写入 RDMA_PCI_ADDR/WDMA_PCI_ADDR。
- 步骤 3 将数据传输的源地址写入 RDMA_AHB_ADDR/WDMA_AHB_ADDR。
- 步骤 4 将 DMA 控制命令写入 RDMA_CONTROL/WDMA_CONTROL，其中：
 1. RDMA_CONTROL/WDMA_CONTROL[31:8]为 transfer size
 2. RDMA_CONTROL/WDMA_CONTROL[7:4]为 command
 3. RDMA_CONTROL/WDMA_CONTROL[3]为 pci_interrupt bit



4. `RDMA_CONTROL/WDMA_CONTROL`[1]为 stop bit
5. `RDMA_CONTROL/WDMA_CONTROL`[0]为 start bit

----结束

当 Hi3520 为 PCI Device 模式时，通过 DMA 通道进行数据传输的步骤如下：

- 步骤 1 配置 CPU_IMASK 寄存器，使能 DMA Read/Write 中断。
- 步骤 2 将数据传输的目标地址写入 `RDMA_PCI_ADDR/WDMA_PCI_ADDR`。
- 步骤 3 将数据传输的源地址写入 `RDMA_AHB_ADDR/WDMA_AHB_ADDR`。
- 步骤 4 将 DMA 控制命令写入 `RDMA_CONTROL/WDMA_CONTROL`，其中：
 1. `RDMA_CONTROL/WDMA_CONTROL`[31:8]为 transfer size
 2. `RDMA_CONTROL/WDMA_CONTROL`[7:4]为 command
 3. `RDMA_CONTROL/WDMA_CONTROL`[3]为 pci_interrupt bit
 4. `RDMA_CONTROL/WDMA_CONTROL`[1]为 stop bit
 5. `RDMA_CONTROL/WDMA_CONTROL`[0]为 start bit

----结束

说明

- 无论 Hi3520 为 PCI Host 模式还是 PCI Device 模式，当 DMA 操作结束后不需要以中断方式通知 CPU 时，此时可不执行步骤 1，或者配置 CPU_IMASK 寄存器将对应的中断屏蔽位关闭。
- 当 Hi3520 为 PCI Host 模式时，`RDMA_CONTROL/WDMA_CONTROL`[3]固定为 0b0。

9.6 寄存器概览

Hi3520 PCI 接口实现了一个标准的符合 PCI V2.3 规范的 PCI 配置空间。此外，在 AHB 侧也实现了一些其它的配置寄存器，以方便 CPU 对 PCI 模块的各种工作方式的配置。

9.6.1 AHB 侧寄存器

AHB 侧寄存器概览如表 9-3 所示。

表9-3 AHB 侧寄存器概览（基址是 0xB000_0000）

偏移地址	名称	描述	页码
0x000	WDMA_PCI_ADDR	DMA 写操作目的地址寄存器	9-13
0x004	WDMA_AHB_ADDR	DMA 写操作源地址寄存器	9-13
0x008	WDMA_CONTROL	DMA 写操作 transfer size 和控制命令寄存器	9-14



偏移地址	名称	描述	页码
0x00C ~ 0x01C	RESERVED	保留	-
0x020	RDMA_PCI_ADDR	DMA 读操作目的地址寄存器	9-15
0x024	RDMA_AHB_ADDR	DMA 读操作源地址寄存器	9-15
0x028	RDMA_CONTROL	DMA 读操作 transfer size 和控制命令寄存器	9-15
0x02C ~ 0x03C	RESERVED	保留	-
0x040	CPU_IMASK	中断屏蔽寄存器	9-16
0x044	CPU_ISTATUS	中断状态寄存器	9-19
0x048	CPU_ICMD	中断命令寄存器	9-22
0x04C	CPU_VERSION	设备版本寄存器	9-22
0x050 ~ 0x06C	RESERVED	保留	-
0x070	PCIAHB_ADDR_NP	PCI-AHB 窗口非预取范围地址控制寄存器	9-23
0x074	PCIAHB_ADDR_PF	PCI-AHB 窗口预取范围地址控制寄存器	9-24
0x078	PCIAHB_TIMER	PCI-AHB 读操作超时寄存器	9-24
0x07C	AHBPCI_TIMER	AHB-PCI 读操作超时寄存器	9-24
0x080	PCI_CONTROL	PCI 控制寄存器	9-25
0x084	PCI_DV	PCI Vendor 和 Vendor ID 寄存器	9-26
0x088	PCI_SUB	PCI Subsystem 设备和 Subsystem Vendor ID 寄存器	9-26
0x08C	PCI_CREV	PCI Class Code 和 Revision ID 寄存器	9-27
0x090	PCI_BROKEN	PCI 仲裁 Master 死锁状态寄存器	9-27
0x094	PCIAHB_SIZE_NP	PCI-AHB 窗口非预取空间范围寄存器	9-28
0x098	PCIAHB_SIZE_PF	PCI-AHB 窗口预取空间范围寄存器	9-28
0x09C ~ 0x3FC	RESERVED	保留	-



9.6.2 PCI 配置空间头标区寄存器

表9-4 PCI 配置空间寄存器概览

偏移地址	名称	描述	页码
0x000	Vendor ID Device ID	供应商识别字段和设备识别字段寄存器	9-29
0x004	Command and Status	设备命令和状态寄存器	9-29
0x008	Revision ID and Class Code	版本识别字段和分类代码字段寄存器	9-30
0x00C	Cacheline Size Master Latency Timer Head Typer	Cache 行容量、PCI Master Latency Time、头标类型寄存器	9-31
0x010	Base Address 0 (BAR0)	基地址寄存器 0 (保留)	-
0x014	Base Address 1 (BAR1)	基地址寄存器 1 (保留)	-
0x018	Base Address 2 (BAR2)	基地址寄存器 2 (保留)	-
0x01C	Base Address 3 (BAR3)	基地址寄存器 3	9-31
0x020	Base Address 4 (BAR4)	基地址寄存器 4	9-32
0x024	Base Address 5 (BAR5)	基地址寄存器 5	9-32
0x028	Cardbus CIS 指针	Cardbus CIS 指针寄存器	9-32
0x02C	Subsystem Vendor ID Subsystem ID	子系统厂商 ID 和子系统 ID 寄存器	9-33
0x030	RESERVED	保留	-
0x034	Capabilities Pointer	能力指针寄存器	9-33
0x038	RESERVED	保留	-
0x03C	Interrupt Line Interrupt Pin Min_Gnt Max_Lat	中断寄存器	9-34



偏移地址	名称	描述	页码
0x040 ~ 0x07C	RESERVED	保留	-
0x080	PCI_IMASK	中断屏蔽寄存器 (PCI Device Only)	9-34
0x084	PCI_ISTATUS	中断状态寄存器 (PCI Device Only)	9-35
0x088	PCI_ICMD	中断命令寄存器 (PCI Device Only)	9-36
0x08C	PCI_VERSION	PCI 模块版本寄存器	9-37
0x090 ~ 0x0FC	RESERVED	保留	-

注：PCI_IMASK、PCI_ISTATUS、PCI_ICMD 寄存器只有当 Hi3520 为 PCI Device 模式时才会使用。

9.7 寄存器描述

9.7.1 AHB 侧寄存器描述

WDMA_PCI_ADDR

WDMA_PCI_ADDR 为 DMA 写操作目的地址寄存器。

	Offset Address				Register Name								Total Reset Value																							
	0x000				WDMA_PCI_ADDR								0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	wdma pci addr																																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	Bits	Access	Name	Description																																
	[31:0]	RW	wdma pci addr	DMA 写操作目的地址。																																

WDMA_AHB_ADDR

WDMA_AHB_ADDR 为 DMA 写操作源地址寄存器。

Offset Address		Register Name		Total Reset Value				
0x004		WDMA_AHB_ADDR		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	wdma ahb addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RW	wdma ahb addr	DMA 写操作源地址。					

WDMA_CONTROL

WDMA_CONTROL 为 DMA 写操作 transfer size 和控制命令寄存器。

Offset Address		Register Name		Total Reset Value								
0x008		WDMA_CONTROL		0x0000_0000								
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0				
Name	write DMA transfer size						dma command		pci interrupt enable	reserved	stop dma	start dma
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description									
[31:8]	RW	write DMA transfer size	DMA 写操作 transfer size, 以 byte 为单位, 最大值为最大为 (16M - 1) Byte。									
[7:4]	RW	dma command	DMA 写操作命令, 具体描述请参考表 9-2。									
[3]	RW	pci interrupt enable	PCI 中断使能。 0: 禁止; 1: 使能。 注意: 在发起 DMA 操作时若将此位置 1, 在 DMA 操作结束后将会产生 PCI 中断通过 PCI_INTA 上报给 PCI Host。因此, 只有当 Hi3520 为 PCI Device 模式时才会使用这一比特, 当 Hi3520 为 PCI Host 时此位固定为 0。									
[2]	-	reserved	保留。									



[1]	RW	stop dma	DMA 写操作停止命令。 0: DMA 操作继续; 1: DMA 操作停止。
[0]	RW	start dma	DMA 写操作开始命令。 0: DMA 操作已经结束; 1: 开始 DMA 写操作。

RDMA_PCI_ADDR

RDMA_PCI_ADDR 为 DMA 读操作目的地址寄存器。

	Offset Address	Register Name	Total Reset Value
	0x020	RDMA_PCI_ADDR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rdma pci addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	rdma pci addr	DMA 读操作目的地址。

RDMA_AHB_ADDR

RDMA_AHB_ADDR 为 DMA 读操作源地址寄存器。

	Offset Address	Register Name	Total Reset Value
	0x024	RDMA_AHB_ADDR	0x0000_0000
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	rdma ahb addr		
Reset	0 0		
Bits	Access	Name	Description
[31:0]	RW	rdma ahb addr	DMA 读操作源地址。

RDMA_CONTROL

RDMA_CONTROL 为 DMA 读操作 transfer size 和控制命令寄存器。



	Offset Address				Register Name				Total Reset Value																							
	0x028				RDMA_CONTROL				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	read DMA transfer size																dma command				pci int.	reserved	stop dma	start dma								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description																												
	[31:8]	RW	read dma transfer size	DMA 读操作 transfer size, 以 byte 为单位, 最大值为 16MB。																												
	[7:4]	RW	dma command	DMA 读操作命令, 具体描述请参考表 9-2。																												
	[3]	RW	pci int	PCI 中断使能。 0: 禁止; 1: 使能。 注意: 在发起 DMA 操作时若将此位置为 1, 在 DMA 操作结束后将会产生 PCI 中断通过 PCI_INTA 上报给 PCI Host。因此, 只有当 Hi3520 为 PCI Device 模式时才会使用这一比特, 当 Hi3520 为 PCI Host 时此位固定为 0。																												
	[2]	-	reserved	保留。																												
	[1]	RW	stop dma	DMA 读操作停止命令。 0: DMA 操作继续; 1: DMA 操作停止。																												
	[0]	RW	sart dma	DMA 读操作开始命令。 0: DMA 操作已经结束; 1: 开始 DMA 读操作。																												

CPU_IMASK

CPU_IMASK 为中断屏蔽寄存器。



Offset Address		Register Name		Total Reset Value																	
0x040		CPU_IMASK		0x0000_0000																	
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0													
Name	reserved	imask_serr	imask_pci_int.	reserved	imask_pci_doorbell	reserved	post_error	fetch_error	discard	ahb_error	parity_error	dma_abort	dma_end	reserved	post_error	fetch_error	discard	ahb_error	parity_error	dma_abort	dma_end
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description																		
[31:29]	-	reserved	保留。																		
[28]	RW	imask_serr	SERR 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[27:24]	RW	imask_pci_int.	PCI 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[23:20]	-	reserved	保留。																		
[19:16]	RW	imask_pci_doorbell	PCI doorbell 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[15]	-	reserved	保留。																		
[14]	RW	imask_ahbpci_post	AHB-PCI 窗口 post 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[13]	RW	imask_ahbpci_fetch	AHB-PCI 窗口 fetch 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[12]	RW	imask_ahbpci_discard	AHB-PCI 窗口 discard 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		
[11]	RW	imask_rdma_ahb_error	DMA 读操作 ahb_error 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																		



[10]	RW	imask_rdma_parity	DMA 读操作 parity 中断屏蔽。 0: 屏蔽; 1: 不屏蔽; 。
[9]	RW	imask_rdma_abort	DMA 读操作 abort 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[8]	RW	imask_rdma_end	DMA 读操作 end 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[7]	-	reserved	保留。
[6]	RW	imask_pciahb_post	PCI-AHB 窗口 post 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[5]	RW	imask_pciahb_fetch	PCI-AHB 窗口 fetch 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[4]	RW	imask_pciahb_discard	PCI-AHB 窗口 discard 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[3]	RW	imask_wdma_ahb_error	DMA 写操作 ahb_error 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[2]	RW	imask_wdma_parity	DMA 写操作 parity 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[1]	RW	imask_wdma_abort	DMA 读操作 abort 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[0]	RW	imask_wdma_end	DMA 写操作 end 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。



CPU_ISTATUS

CPU_ISTATUS 为中断状态寄存器。当向此寄存器对应的比特位写入 1 时，清除相应的中断。

	Offset Address				Register Name				Total Reset Value																								
	0x044				CPU_ISTATUS				0x0000_0000																								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved				istatus_serr	istatus_pci_int.				reserved				istatus_pci_doorbell				reserved	post_error	fetch_error	discard	ahb_error	parity_error	dma_abort	dma_end	reserved	post_error	fetch_error	discard	ahb_error	parity_error	dma_abort	dma_end
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	[31:29]				[28]				[27:24]				[23:20]				[19:16]																
Access	-				WC				-				WC																				
Name	reserved				istatus_serr				istatus_pci_int				reserved				istatus_pci_doorbell																
Description	保留。				SERR 中断状态， 当设备被配置为 PCI Host 模式时，istatus_serr 表示 PCI Host 检测到 PCI 总线上有奇偶校验错误。 当设备被配置为 PCI Device 模式时，istatus_serr 保留。 0: 无中断； 1: 有中断。				PCI 中断状态。 0: 无中断； 1: 有中断。 当设备被配置为 PCI Host 模式时，istatus_pci_int 分别对应 PCI_INTD、PCI_INTC、PCI_INTB 和 PCI_INTA 的中断状态。 当设备被配置为 PCI Device 模式时，istatus_pci_int 保留。				保留。				PCI doorbell 中断状态。 0: 无中断； 1: 有中断。 PCI doorbell 中断是提供给用户使用的自定义中断资源。 bit[19:16]分别对应 PCI_ICMD [19:16]这 4bit 中断命令控制字的中断状态。 当设备被配置为 PCI Device 模式时，istatus_pci_doorbell 有效。 当设备被配置为 PCI Host 时，istatus_pci_doorbell 保留。																



[15]	-	reserved	保留。
[14]	WC	istatus_ahbpci_post	AHB-PCI 窗口 post 中断状态，此中断表示芯片内部 AHB Master 进行写操作时访问到了未知的地址，而导致 AHB Slave 没有响应。一般情况下，当 PCI 总线发生 Master Abort 或者 Target Abort 时，此中断状态位会有效。 0: 无中断； 1: 有中断。
[13]	WC	istatus_ahbpci_fetch	AHB-PCI 窗口 fetch 中断状态，此中断表示芯片内部 AHB Master 进行读操作时访问到了未知的地址，而导致 AHB Slave 没有响应。一般情况下，当 PCI 总线发生 Master Abort 或者 Target Abort 时，此中断状态位会有效。 0: 无中断； 1: 有中断。
[12]	WC	istatus_ahbpci_discard	AHB-PCI 窗口 discard 中断状态，此中断表示 AHB 总线上的 AHB Master 发起了对片外 PCI Target 的读操作，当读数据返回时，AHB Master 在系统配置的时限内没有发起 Retry 操作把数据取走，导致返回的数据被丢弃。因此，此中断表示芯片 AHB Master 发生错误。 0: 无中断； 1: 有中断。
[11]	WC	istatus_rdma_ahb_error	DMA 读操作 ahb_error 中断状态，此中断表示在 DMA 读操作过程中 AHB Master 收到 AHB Slave 的 Error 响应。需要注意的是，当发生 Read DMA AHB error 时，此时 DMA 操作也将中止（RDMA_CONTROL 寄存器第 0bit 置 0）。 0: 无中断； 1: 有中断。
[10]	WC	istatus_rdma_parity	DMA 读操作 parity 中断状态，此中断表示在 DMA 读操作过程中 PCI 总线上发生奇偶校验错误。 0: 无中断； 1: 有中断。
[9]	WC	istatus_rdma_abort	DMA 读操作 abort 中断状态，此中断表示在 DMA 读操作时 PCI Master 发出了一个未知的 PCI Address 而导致没有得到任何 PCI Device 的响应，或者由于其他原因发生 PCI Abort。 0: 无中断； 1: 有中断。



[8]	WC	istatus_rdma_end	<p>DMA 读操作 end 中断状态，此中断表示 DMA 读操作结束。</p> <p>需要特别注意的是，当在 DMA 读操作过程中发生了 DMA Abort，此时除了向 CPU 上报 DMA Abort 中断之外也会上报读 DMA End 中断（即如果发生了 DMA Abort，那么 RDMA_CONTROL（0x28）寄存器的第 0 bit 也会同时被清零），软件在此处做特殊处理。</p> <p>0：无中断； 1：有中断。</p>
[7]	-	reserved	保留。
[6]	WC	istatus_pciahb_post	<p>PCI-AHB 窗口 post 中断状态，此中断表示 PCI Master 通过 PCI-AHB 窗口进行写操作时发出了未知的 AHB Slave 地址，导致 AHB Slave 发出了 Error 响应，或者由于其他原因导致 AHB Slave 发出 Error 响应。</p> <p>0：无中断； 1：有中断。</p>
[5]	WC	istatus_pciahb_fetch	<p>PCI-AHB 窗口 fetch 中断状态，此中断表示 PCI Master 通过 PCI-AHB 窗口进行读操作时发出了未知的 AHB Slave 地址，导致 AHB Slave 发出了 Error 响应，或者由于其他原因导致 AHB Slave 发出 Error 响应。</p> <p>0：无中断； 1：有中断。</p>
[4]	WC	istatus_pciahb_discard	<p>PCI-AHB 窗口 discard 中断状态。此中断表示 PCI 总线上的 PCI Master 发起了对片外其他 PCI Target 的读操作，当读数据返回时，PCI Master 在系统配置的时限内没有发起 Retry 操作把数据取走，导致返回的数据被丢弃。因此，此中断表示芯片 PCI Master 发生错误。</p> <p>0：无中断； 1：有中断。</p>
[3]	WC	istatus_wdma_ahb_error	<p>DMA 写操作 ahb_error 中断状态。此中断表示在 DMA 写操作过程中 AHB Master 收到 AHB Slave 的 Error 响应。需要注意的是，当发生 Write DMA AHB error 时，此时 DMA 操作也将中止（WDMA_CONTROL 寄存器第 0bit 置 0）。</p> <p>0：无中断； 1：有中断。</p>
[2]	WC	istatus_wdma_parity	<p>DMA 写操作 parity 中断状态。此中断表示在 DMA 写操作过程中 PCI 总线上发生奇偶校验错误。</p> <p>0：无中断； 1：有中断。</p>

[1]	WC	istatus_wdma_abort	DMA 写操作 abort 中断状态。此中断表示在 DMA 写操作时 PCI Master 发出了一个未知的 PCI Address 而导致没有得到任何 PCI Device 的响应，或者由于其他原因发生了 PCI Abort。 0: 无中断； 1: 有中断。
[0]	WC	istatus_wdma_end	DMA 写操作 end 中断状态，此中断表示 DMA 写操作结束。需要特别注意的是，如果在 DMA 写操作过程中发生了 DMA Abort，此时除了向 CPU 报 DMA Abort 中断之外也会上报 DMA End 中断（即如果发生了 DMA Abort，那么 WDMA_CONTROL (0x08) 寄存器的第 0 bit 也会同时被清零），软件在此处做特殊处理。 0: 无中断； 1: 有中断。

CPU_ICMD

CPU_ICMD 为中断命令寄存器。

	Offset Address 0x048								Register Name CPU_ICMD								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cpu_doorbell				reserved																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access		Name				Description																								
	[31:24]	-		reserved				保留。																								
	[23:20]	WO		cpu_doorbell				CPU doorbell 中断命令。 CPU doorbell 中断是提供给用户自定义的中断资源，用户通过配置 bit[23:20]的任何一比特或者几比特，可发出中断，通过 PCI_INTA 管脚送达 PCI Host 设备。CPU doorbell 中断仅用于 Hi3520 为 PCI Device 模式下，且此中断命令高电平有效。																								
	[19:0]	-		reserved				保留。																								

CPU_VERSION

CPU_VERSION 为设备版本寄存器。



说明

- PCI Device 模式时，复位值为 0x0000_1380。
- PCI Host 模式时，复位值为 0x0000_2380。



Offset Address		Register Name		Total Reset Value					
0x04C		CPU_VERSION		-					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				device type	device version			
Reset	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:12]	RO	device type	设备类型。 0x1: PCI Device; 0x2: PCI Host.						
[11:0]	RO	device version	设备版本。						

PCIAHB_ADDR_NP

PCIAHB_ADDR_NP 为 PCI-AHB 窗口非预取范围地址控制寄存器。

当 Hi3520 为 PCI Host 模式时，PCIAHB_ADDR_NP[pciahb_addr_np]和 PCIAHB_ADDR_PF[pciahb_addr_pf]的值分别与 PCI 配置空间中 Base Address 3 和 Base Address 4 的值相同，此时 PCI Device 通过 PCI 总线访问 Hi3520 时就是依据这两个寄存器的值来发出目标地址。

Offset Address		Register Name		Total Reset Value				
0x070		PCIAHB_ADDR_NP		0xB000_0001				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	pciahb_addr_np						reserved	window_en
Reset	1 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1
Bits	Access	Name	Description					
[31:8]	RW	pciahb_addr_np	PCI-AHB 窗口非预取地址 AHB 侧基地址。					
[7:1]	-	reserved	保留。					
[0]	RW	window_en	PCI-AHB 窗口非预取使能。 0: 禁止; 1: 使能。					



PCIAHB_ADDR_PF

PCIAHB_ADDR_PF 为 PCI-AHB 窗口预取范围地址控制寄存器。

	Offset Address				Register Name								Total Reset Value																							
	0x074				PCIAHB_ADDR_PF								0xB000_0001																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	pciahb_addr_pf																reserved								window_en											
Reset	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
Bits	Access		Name		Description																															
[31:8]	RW		pciahb_addr_pf		PCI-AHB window 预取窗口地址 AHB 侧基地址。																															
[7:1]	-		reserved		保留。																															
[0]	RW		window_en		PCI-AHB 窗口预取使能。 0: 禁止; 1: 使能。																															

PCIAHB_TIMER

PCIAHB_TIMER 为 PCI-AHB 读操作超时寄存器。

	Offset Address				Register Name								Total Reset Value																							
	0x078				PCIAHB_TIMER								0x0000_0100																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved																pciahb_discard																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0				
Bits	Access		Name		Description																															
[31:12]	-		reserved		保留。																															
[11:0]	RW		pciahb_discard		PCI-AHB window 读操作超时时计数值。																															

AHBPCI_TIMER

AHBPCI_TIMER 为 AHB-PCI 读操作超时寄存器。



Offset Address		Register Name		Total Reset Value					
0x07C		AHBPCI_TIMER		0x0000_0100					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved					ahbpci_discard			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:12]	-	reserved	保留。						
[11:0]	RW	ahbpci_discard	AHB-PCI window 读操作超时计数值。						

PCI_CONTROL

PCI_CONTROL 为 PCI 控制寄存器。



- 说明
- PCI Device 模式时，复位值为 0x0000_0001。
 - PCI Host 模式时，复位值为 0x0000_0007。

Offset Address		Register Name		Total Reset Value					
0x080		PCI_CONTROL		-					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved						master_en	memory_en	pci_ready
Reset	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	
Bits	Access	Name	Description						
[31:3]	-	reserved	保留。						
[2]	RW	master_en	PCI Master 使能。 0: 禁止; 1: 使能。						
[1]	RW	memory_en	Memory 空间访问使能。 0: 禁止; 1: 使能。						
[0]	RO	pci_ready	PCI 准备好。 该 bit 在 PCI 复位的过程被清 0，然后再经过 2 ²⁵ 个 PCI 时钟周期置 1。CPU 必须等到这个比特被置位才能开始对 PCI 总线进行配置访问。						



说明

- 当 Hi3520 为 PCI Host 模式时, PCI_CONTROL[master_en]、PCI_CONTROL[memory_en]和 PCI_CONTROL[pci_ready]在 PCI 复位结束后自动置 1。
- 当 Hi3520 为 PCI Device 模式时, PCI_CONTROL[master_en]和 PCI_CONTROL[memory_en]分别与 PCI 配置空间中命令状态寄存器 COMMAND_STATUS[1]和 COMMAND_STATUS[0]的作用相同, 此时 PCI_CONTROL[master_en]和 PCI_CONTROL[memory_en]的配置是由 PCI Host 来完成。

PCI_DV

PCI_DV 为 PCI Vendor 和 Vendor ID 寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x084				PCI_DV				0x3520_19E5																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	device ID								vendor ID																							
Reset	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	1
	Bits	Access	Name		Description																											
	[31:16]	RO	device ID		设备 ID。																											
	[16:15]	RO	vendor ID		厂商 ID。																											

PCI_SUB

PCI_SUB 为 PCI Subsystem 设备和 Subsystem Vendor ID 寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x088				PCI_SUB				0x3520_19E5																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	subsystem ID								subsystem vendor ID																							
Reset	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	1
	Bits	Access	Name		Description																											
	[31:16]	RO	subsystem ID		子系统 ID。																											
	[16:15]	RO	subsystem vendor ID		子系统生产商 ID。																											



PCI_CREV

PCI_CREV 为 PCI 设备分类类别码和 Revision ID 寄存器。

	Offset Address 0x8C								Register Name PCI_CREV								Total Reset Value 0x0400_0010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	class_code																revision															
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	Bits	Access	Name		Description																											
	[31:8]	RO	class_code		设备分类类别码。																											
	[7:0]	RO	revision		Revision ID。																											

PCI_BROKEN

PCI_BROKEN 为 PCI 仲裁 Master 死锁状态寄存器。可通过对相应的比特位写入 1 清除 PCI Master broken 标志。

	Offset Address 0x090								Register Name PCI_BROKEN								Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																broken_status															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:8]	-	reserved		保留。																											
	[7]	WC	broken_status		PCI Master broken 标志，该 bit 位对应于 PCI Host 本身的 PCI Master 的仲裁状态。 0: PCI Master OK; 1: PCI Master broken。																											
	[6]	WC	broken_status		PCI Master broken 标志，该 bit 位对应于 PCI_REQ[0]和 PCI_GNT[0]所连接 PCI Device 的 Master 的仲裁状态。 0: PCI Master OK; 1: PCI Master broken。																											
	[5:4]	-	reserved		保留。																											



[3:0]	WC	broken_status	PCI Master broken 标志，PCI_BROKEN[3:0]分别对应于 bridge_req[7:0]和 bridge_gnt[7:0]中每一对点对点信号所连接 PCI Device 的 Master 的仲裁状态。 0: PCI Master OK; 1: PCI Master broken。
-------	----	---------------	---

注：PCI 总线上的 PCI Master 在申请到 PCI 总线后 16 个 PCI 时钟周期内未发出任何总线操作，此时 PCI Host 会把 PCI_BROKEN 寄存器中该 PCI Master 对应比特置 1，此后，PCI Master 如果再申请 PCI 总线，PCI Host 的仲裁器将屏蔽该 Master 的申请，直到 CPU 将其对应的 broken status 标志清除为止。

PCIAHB_SIZ_NP

PCIAHB_SIZ_NP 为 PCI-AHB 窗口非预取空间范围寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x094				PCIAHB_SIZ_NP				0xFFC0_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	pciahb_siz_np																reserved															
Reset	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:8]	RW	pciahb_siz_np		PCI-AHB 窗口非预取空间范围。																											
	[7:0]	-	reserved		保留。																											

PCIAHB_SIZ_PF

PCIAHB_SIZ_PF 为 PCI-AHB 窗口预取空间范围寄存器。

	Offset Address				Register Name				Total Reset Value																							
	0x98				PCIAHB_SIZ_PF				0xFF80_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	pciahb_siz_pf																reserved															
Reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:8]	RW	pciahb_siz_pf		PCI-AHB 窗口预取空间范围。																											
	[7:0]	-	reserved		保留。																											



9.7.2 PCI 配置空间寄存器描述

Vendor ID and Device ID

Vendor ID and Device ID 为供应商识别字段和设备识别字段寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x000				Vendor ID and Device ID								0x3520_19E5																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	device ID												vendor ID																			
Reset	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	1
	Bits	Access	Name		Description																											
	[31:16]	RO	device ID		设备 ID。																											
	[16:15]	RO	vendor ID		供应商 ID。																											

Command and Status

Command and Status 为设备命令寄存器和状态寄存器。

	Offset Address				Register Name								Total Reset Value																							
	0x004				Command &Status								0x04B0_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	parity_error	serr	master_abort	target_abort	signal_abort	devsel_timing	master_perr	faste_btob	reserved	66M_cap	cap_list	int_ststus	reserved								int.len	fast_btb_en	serr_en	reserved	perr_resp	VGA_en	mem_w_i	special_cyc	master_en	memory_en	I/O_en					
Reset	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																															
	[31]	RW	parity_error		Device 检测到奇偶校验错误。																															
	[30]	RW	serr		地址周期奇偶校验错误。																															
	[29]	RW	master_abort		PCI master abort。																															
	[28]	RW	target_abort		PCI target abort。																															
	[27]	RW	signal_abort		目标设备发出 abort 信号标志。																															
	[26:25]	RO	devsel_timing		目标设备地址译码响应速度状态。																															
	[24]	RW	master_perr		Master 检测到奇偶校验错误。																															



[23]	RO	fast_btob	快速背对背能力。
[22]	-	reserved	保留。
[21]	RO	66M_cap	66MHz 能力。
[20]	RO	cap_list	能力列表。
[19]	RO	int_status	中断状态。
[18:11]	-	reserved	保留。
[10]	RW	int_en	中断使能。
[9]	RO	fast_btob	高速背对背传输允许。
[8]	RW	serr_en	SERR 使能。
[7]	-	reserved	保留。
[6]	RW	perr_resp	奇偶错误响应。
[5]	RO	VGA_en	VGA 调色板控制。
[4]	RO	mem_w_i_en	存储器写并无效使能。
[3]	RO	special_cyc	特殊周期控制。
[2]	RW	master_en	总线主设备使能。
[1]	RW	memory_en	存储器空间使能。
[0]	RW	I/O_en	I/O 空间使能。

Revision ID and Class Code

Revision ID and Class Code 为版本识别字段和分类代码字段寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x008				Revision ID and Class Code								0x0400_0010																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	class code																revision ID															
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bits	Access		Name		Description																											
[31:8]	RO		class code		设备分类类别码。																											
[7:0]	RO		revision		revision 版本号。																											



Cacheline Size and Master Latency Timer

Cacheline Size and Master Latency Timer 为 Cache 行容量和 PCI Master Latency Time 寄存器。

Offset Address		Register Name		Total Reset Value					
0x00C		Cacheline Size and Latency Timer		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	bist		head type		master latency timer		reserved	cacheline size	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:24]	-	bist	保留。						
[23:16]	RO	head type	头标类型。						
[15:8]	RW	master latency timer	PCI master 延时寄存器，其中 bit[9:8]为只读位，值为 0b00，表明 latency timer 的时钟颗粒度为 8 个 PCI 时钟周期。						
[7:0]	RO	cacheline size	cacheline 大小。 注意：Hi3520 PCI 接口并未实现对 Memory Cacheline 的操作，因此，此处的 cacheline size 固定为 0x00。						

Base Address 3

Base Address 3 为基地址寄存器 3。

Offset Address		Register Name		Total Reset Value					
0x01C		Base Address 3		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	base_address3						reserved		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:8]	RW	base_address3	基地址寄存器 3。						
[7:0]	-	reserved	保留。						



Base Address 4

Base Address 4 为基地址寄存器 4。

	Offset Address 0x020				Register Name Base Address 4								Total Reset Value 0x0000_0008																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	base address 4																				reserved				prefet_flag	reserved						
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				1	0 0 0						
Bits	Access		Name		Description																											
[31:0]	RW		base address 4		基地址寄存器 4。																											
[7:4]	-		reserved		保留。																											
[3]	RO		prefet_flag		预取空间标志。																											
[2:0]	-		reserved		保留。																											

Base Address 5

Base Address 5 为基地址寄存器 5。

	Offset Address 0x024				Register Name Base Address 5								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	base address 5																				prefet_flag	reserved										
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				1	0 0 0						
Bits	Access		Name		Description																											
[31:4]	RW		base address 5		基地址寄存器 5。																											
[3]	RO		prefet_flag		预取空间标志。																											
[2:0]	-		reserved		保留。																											

Cardbus CIS Pointer

Cardbus CIS Pointer 为 Cardbus CIS 指针寄存器。



	Offset Address				Register Name								Total Reset Value																			
	0x028				Cardbus CIS Pointer								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cardbus_cis_point																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:0]	RO	cardbus_cis_point		Cardbus CIS 指针寄存器。																											

Subsystem Vendor ID and Subsystem ID

Subsystem Vendor ID and Subsystem ID 为子系统厂商 ID 和子系统供应商 ID 寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x02C				Subsystem Vendor ID and Subsystem ID								0x3511_19E5																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	subsystem ID												subsystem vendor ID																			
Reset	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	1
	Bits	Access	Name		Description																											
	[31:16]	RO	subsystem ID		子系统 ID。																											
	[16:15]	RO	subsystem vendor ID		子系统供应商 ID。																											

Capabilities Pointer

Capabilities Pointer 为能力指针寄存器。



说明

Hi3520 PCI 接口部分并未实现 PCI 的扩展能力。



Offset Address		Register Name		Total Reset Value				
0x034		Capabilities Pointer		0x0000_0078				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved				capabilities pointer			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 1	1 0 0 0
Bits	Access	Name	Description					
[31:16]	-	reserved	保留。					
[15:0]	RO	capabilities pointer	能力指针寄存器。					

Interrupt Line Interrupt Pin MIN_GNT MAX_LAT

Interrupt Line Interrupt Pin MIN_GNT MAX_LAT 为中断寄存器。

Offset Address		Register Name		Total Reset Value				
0x03C		Interrupt Line & Interrupt Pin & MIN_GNT & MAX_LAT		0x0000_0100				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	max_lat		min_lat		interrupt pin		interrupt line	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:24]	RO	max_lat	MAX latency 寄存器。					
[23:16]	RO	min_gnt	MIN grant 寄存器。					
[15:8]	RO	interrupt in	中断引脚寄存器。					
[7:0]	RW	interrupt line	中断线寄存器。					

PCI_IMASK

PCI_IMASK 为中断屏蔽寄存器 (PCI Device Only)。

说明

PCI_IMASK、[PCL_ISTATUS](#)、[PCL_ICMD](#) 这三个寄存器只有当 Hi3520 为 PCI Device 模式时才会使用。此时如果 Hi3520 触发 PCI 中断，那么中断就会通过 PCI_INTA 发送到 PCI Host 设备。



	Offset Address 0x080								Register Name PCI_IMASK								Total Reset Value 0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved								imask_cpu_doorbell				reserved								rdma_end				reserved								wdma_end			
Reset	0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0				0 0 0 0			
Bits	Access		Name		Description																															
[31:24]	-		reserved		保留。																															
[23:20]	RW		imask_cpu_doorbell		CPU Doorbell 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																															
[19:9]	-		reserved		保留。																															
[8]	RW		imask_rdma_end		DMA 读操作结束中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																															
[7:1]	-		reserved		保留。																															
[0]	RW		imask_wdma_end		DMA 写操作结束中断屏蔽。 0: 屏蔽; 1: 不屏蔽。																															

PCI_ISTATUS

PCI_ISTATUS 为中断状态寄存器（PCI Device Only）。当向此寄存器的相应比特位写 1 时，清除相应的中断。



Offset Address		Register Name		Total Reset Value						
0x084		PCI_ISTATUS		0x0000_0000						
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0		
Name	reserved		istatus_cpu_doorbell	reserved			rdma_end	reserved		wdma_end
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0		
Bits	Access	Name	Description							
[31:24]	-	reserved	保留。							
[23:20]	WC	cpu_doorbell	CPU Doorbell 中断状态。 0: 无中断; 1: 有中断。 当用户配置了 CPU_ICMD bit[23:20]的 CPU doebell 中断后, Hi3520 会通过 PCI_INTA 发出 doorbell 中断, 同时所配置的中断状态会反映在 bit[23:20]的这 4bit 上。							
[19:9]	-	reserved	保留。							
[8]	WC	rdma_end	DMA 读操作结束中断状态。 0: 无中断; 1: 有中断。							
[7:1]	-	reserved	保留							
[0]	WC	wdma_end	DMA 写操作结束中断状态。 0: 无中断; 1: 有中断。							

PCI_ICMD

PCI_ICMD 为中断命令寄存器 (PCI Device Only)。



Offset Address		Register Name		Total Reset Value					
0x088		PCI_ICMD		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved			icmd_cpu_doorbell	reserved				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:20]	-	reserved	保留。						
[19:16]	RW	icmd_pci_doorbell	PCI Doorbell Interrupt 命令。PCI doorbell 中断是提供给用户自定义的可通过 PCI 总线发起的中断资源。此中断由 PCI Host 设备通过配置 bit[19:16]的任何 bit 位来发起中断，当配置完这 4bit 后，Hi3520 会产生上报 ARM CPU 的中断。PCI Doorbell Interrupt 命令为高电平有效。 PCI Doorbell Interrupt 仅用于 Hi3520 作为 PCI Device 模式时。						
[15:0]	-	reserved	保留。						

PCI_VERSION

PCI_VERSION 为 PCI 模块版本寄存器。



说明

- PCI Device 模式时，复位值为 0x0000_0138。
- PCI Host 模式时，复位值为 0x0000_0238。

Offset Address		Register Name		Total Reset Value				
0x04C		CPU_VERSION		-				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved			device type	device version			
Reset	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?	? ? ? ?
Bits	Access	Name	Description					
[31:16]	-	reserved	保留。					
[15:12]	RO	device type	设备类型。					
[11:0]	RO	device version	设备版本。					



10 USB 2.0 Host

10.1 概述

USB 2.0 Host 控制器支持 High-speed (480Mbit/s)、Full-speed (12Mbit/s) 和 Low-speed (1.5Mbit/s) 三种数据传输。USB 2.0 Host 控制器完全支持 USB 2.0、OHCI Rev 1.0a 和 EHCI Rev 1.0 协议。USB 2.0 Host 控制器中包含一个 Root Hub (USB 系统中的一部分, 通过 Hub 可以扩展 USB 接口)。USB 2.0 Host 控制器的大部分硬件逻辑可以:

- 完成对传输的控制和处理
- 对数据包的解析和打包
- 对 USB 传输信号的编码和解码
- 为驱动程序提供了中断向量等接口

10.2 功能描述

功能特点

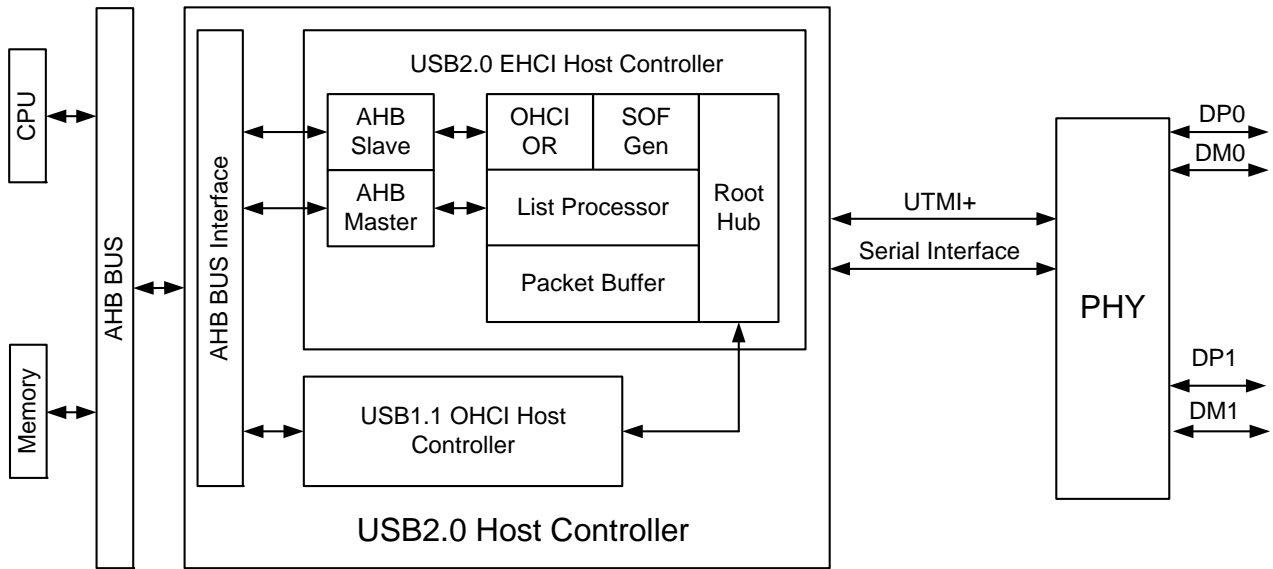
USB 2.0 Host 控制器具有以下功能特点:

- 支持 USB 2.0 协议
- 支持 OHCI Rev 1.0a、EHCI Rev 1.0 协议
- 支持 High-speed、Full-speed、Low-speed 三种数据传输模式
- 支持低功耗的解决方案
- 支持 Control Transfer、Bulk Transfer、Isochronous Transfer、Interrupt Transfer 四种基本数据传输类型
- 支持通过连接 USB Hub, 最多连接 127 个设备

USB 模块逻辑框图如[图 10-1](#) 所示。



图10-1 USB 模块逻辑框图



UTMI: USB2.0 Transceiver Macrocell Interface

EHCI: Enhanced Host Controller Interface

OHCI: Open Host Controller Interface

工作原理

USB 2.0 Host 支持以下 4 种标准的传输方式:

- Control Transfer (控制传输)**
 主要用于 USB Host 与 USB Device 端点 0 之间的传输, 某些特定型号的 USB 设备的控制传输可能用到其他的端点。控制传输是双向传输, 数据量通常较小, 可以传输 8byte、16byte、32byte 或 64byte 的数据, 依赖于设备和传输速度。
- Bulk Transfer (批量传输)**
 主要用于没有带宽和间隔时间要求的情况下发送和接收大量的数据, 这种类型的设备适合于传输非常慢和大量被延迟的传输, 可以等到所有其他类型的数据的传送完成之后再传送和接收数据。它的特点是以错误检测和重传的方式保证 USB Host 与 USB Device 的数据被无差错地发送。
- Isynchronous Transfer (同步传输)**
 主要用于间隔时间要求严格并具有较强容错性的流数据传输, 或者用于数据传输速率恒定的即时应用中。同步传输提供了确定的带宽和间隔时间。
- Interrupt Transfer (中断传输)**
 中断传输类型用于支持那些不需要经常发送或者接收数据、但是服务周期有限的设备。中断管道是一个流管道, 因此一般都是单向的。端点描述符能识别出给定中断管道的通信流是流入还是流出主机。



10.3 工作方式

10.3.1 接口信号

集成 USB PHY 的 USB 2.0 Host 接口信号描述如表 10-1 所示。

表10-1 USB 2.0 Host 的接口信号描述

信号名	方向	功能描述	对应管脚
USBDP0	I/O	USB 端口 0 的 D+数据线	USBDP0
USBDM0	I/O	USB 端口 0 的 D-数据线	USBDM0
USBDP1	I/O	USB 端口 1 的 D+数据线	USBDP1
USBDM1	I/O	USB 端口 1 的 D-数据线	USBDM1
USBREXT	I/O	外接电阻连接（外接参考电阻 3.4KΩ±1%，尽量靠近管脚）	USBREXT
USBVDDA33T0_0	I/O	3.3V 模拟电源，接 3.3V（±5%）	USBVDDA33T0_0
USBVDDA33T0_1	I/O	3.3V 模拟电源，接 3.3V（±5%）	USBVDDA33T0_1
USBVSSA33T0_0	I/O	模拟地	USBVSSA33T0_0
USBVSSA33T0_1	I/O	模拟地	USBVSSA33T0_1
USBVSSA33T0_2	I/O	模拟地	USBVSSA33T0_2
USBVDDA33C	I/O	3.3V 模拟电源，接 3.3V（±5%）	USBVDDA33C
USBVSSA33C	I/O	模拟地	USBVSSA33C
USBVDDA33T1_0	I/O	3.3V 模拟电源，接 3.3V（±5%）	USBVDDA33T1_0
USBVDDA33T1_1	I/O	3.3V 模拟电源，接 3.3V（±5%）	USBVDDA33T1_1
USBVSSA33T1_0	I/O	模拟地	USBVSSA33T1_0
USBVSSA33T1_1	I/O	模拟地	USBVSSA33T1_1
USBVSSA33T1_2	I/O	模拟地	USBVSSA33T1_2
USBVDD0	I/O	1.0V 数字电源，接 1.0V（±5%）	USBVDD0
USBVSS0	I/O	数字地	USBVSS0
USBVDD1	I/O	1.0V 数字电源，接 1.0V（±5%）	USBVDD1
USBVSS1	I/O	数字地	USBVSS1



10.3.2 典型应用

内部集成 PHY 的参考设计如图 10-2 所示。

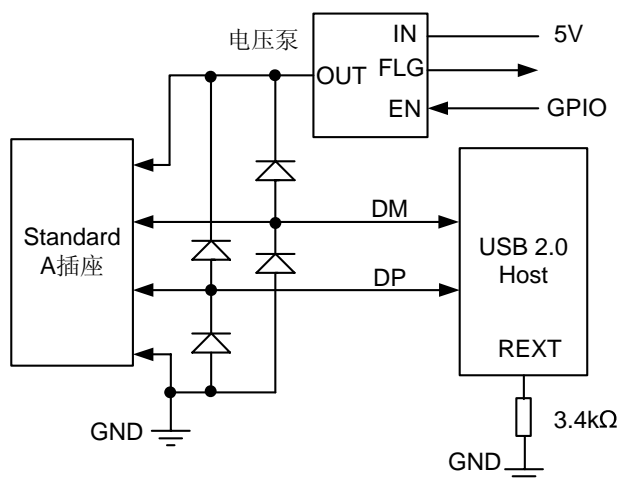


注意

为避免影响信号质量，PCB 设计需要注意以下几点：

- DP/DM 已经单端 45Ω 匹配好了，不用外接匹配
- 外接的 ESD 器件电容值在 1pF 左右
- 电压泵的 EN 信号控制以及过流保护都通过 GPIO 来控制

图10-2 USB 2.0 Host 参考设计



10.3.3 时钟门控

如果需开启控制器工作时钟，则向 `SC_PEREN[usb_clken]` 写 1。如果关闭工作时钟，则向 `SC_PERDIS[usb_clkdis]` 写 1。时钟复位后默认是开启状态。

10.3.4 软复位

USB 2.0 Host 默认一直处于复位状态，要启动 USB 2.0 Host 进入工作状态，必须配置系统控制器的 `SC_PERCTRL8[usb_srst]` 为 1 撤销 USB PHY 侧复位；然后将 `SC_PERCTRL8[usb_hrst]` 置 1 撤销 USB 总线侧的复位，并注意调整系统控制器中 `SC_PERCTRL12[usb_tune0]` / `SC_PERCTRL12[usb_tune1]` 值（推荐配置为标准电平）。



10.4 寄存器概览



说明

由于该 USB 模块是一个标准的 USB 2.0 Host，内部寄存器也是标准的 EHCI 及 OHCI 寄存器，在 EHCI 协议及 OHCI 协议中有详细的描述，请参照协议。下面仅对几个厂家特别定义的寄存器进行描述。

USB 寄存器概览如表 10-2 所示。

表10-2 USB 寄存器概览（基地址是 0x100B_0000）

偏移地址	名称	描述	页码
0x90	INTNREG00	配置微帧长度寄存器	10-5
0x94	INTNREG01	配置 FIFO OUT/IN 水线寄存器	10-6
0x98	INTNREG02	配置 FIFO 深度寄存器	10-6
0x9C	INTNREG03	中断 Memory 传输使能寄存器	10-7
0xA0	INTNREG04	DEBUG 寄存器	10-7
0xA4	INTNREG05	控制及状态寄存器	10-8
0xA8	INTNREG06	AHB 错误状态寄存器	10-9
0xAC	INTNREG07	AHB 错误地址寄存器	10-10

注：EHCI 寄存器基地址是 0x100B_0000，OHCI 寄存器基地址是 0x100A_0000，表 10-2 中寄存器基地址是 EHCI 寄存器基地址。

10.5 寄存器描述

INTNREG00

INTNREG00 为配置微帧长度寄存器。

	Offset Address	Register Name	Total Reset Value																						
	0x90	INTNREG00	0x0000_0000																						
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																								
Name	reserved												val												en
Reset	0 0																								
	Bits	Access	Name	Description																					
	[31:14]	-	reserved	保留。																					



[13:1]	RW	val	微帧计数器值。该寄存器仅用于仿真。正常工作时，微帧长度是协议规定的 125μs；仿真时，为了缩短仿真时间，根据需要可以配置该寄存器改变微帧长度。
[0]	RW	en	使能该寄存器。 0：禁止； 1：使能。

INTNREG01

INTNREG01 为配置 FIFO OUT/IN 流水线寄存器。

Offset Address		Register Name		Total Reset Value				
0x94		INTNREG01		0x0020_0020				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	out_threshold				in_threshold			
Reset	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0
Bits	Access	Name	Description					
[31:16]	RW	out_threshold	发送流水线。发送数据时，当 FIFO 中数据超过发送流水线，开始发送数据。					
[15:0]	RW	in_threshold	接收流水线。接收数据时，当 FIFO 中的数据超过接收流水线，将数据从 FIFO 中读走。					

INTNREG02

INTNREG02 为配置 FIFO 深度寄存器。

Offset Address		Register Name		Total Reset Value				
0x98		INTNREG02		0x0000_0080				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved					fifo_depth		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:12]	-	reserved	保留。					
[11:0]	RW	fifo_depth	FIFO 深度（这里所定义的值是 32bit）。					



INTNREG03

INTNREG03 为中断 Memory 传输使能寄存器。

Offset Address		Register Name		Total Reset Value					
0x9C		INTNREG03		0x0000_0001					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved								brk_en
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	
Bits	Access	Name	Description						
[31:1]	-	reserved	保留。						
[0]	RO	brk_en	中断 Memory 传输使能。 0: 禁止; 1: 使能。						

INTNREG04

INTNREG04 为 DEBUG 寄存器。

Offset Address		Register Name		Total Reset Value								
0xA0		INTNREG04		0x0000_0000								
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0				
Name	reserved						auto_en	nak_reldfx_en	reserved	scaledwn_enum_time	hccparam_en	hesparam_en
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0				
Bits	Access	Name	Description									
[31:6]	-	reserved	保留。									
[5]	RW	auto_en	自动规格使能。 0: 使能（当软件复位 run/stop，但 hchalted 位还未设置时挂起信号无效）； 1: 禁止（即软件清除 run/stop 位时端口不挂起）。 默认为 0。									



[4]	RW	nak_reldfix_en	NAK 重载使能。 0: 使能; 1: 禁止。
[3]	-	reserved	保留。
[2]	RW	scaledwn_enum_time	减少端口枚举时间。 0: 禁止; 1: 使能。
[1]	RW	hccparam_en	HCCPARAMS 寄存器可写使能。 0: 禁止; 1: 使能。
[0]	RW	hcsparam_en	HCSPARAMS 寄存器可写使能。 0: 禁止; 1: 使能。

INTNREG05

INTNREG05 为控制及状态寄存器。用于读写 PHY 寄存器。

USB Host 接口可以配置成 UTMI 接口或者 ULPI 接口，接口类型不同，该寄存器的描述也不同。

接口为 UTMI 时，该寄存器描述如下所示。

	Offset Address				Register Name								Total Reset Value																			
	0xA4				INTNREG05								0x0000_1000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								vbusy	vport			vcontrol_loadm	vcontrol			vstatus															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Bits	[31:18]			[17]			[16:13]																									
Access	-			RO			RW																									
Name	reserved			vbusy			vport																									
Description	保留。			“1”表明硬件现在正在执行数据写操作，只有当进程结束时，该位清零。			端口号，不能超过支持的端口数。																									



[12]	RW	vcontrol_loadm	装载使能。 0: 使能; 1: 禁止。
[11:8]	RW	vcontrol	端口控制信号。
[7:0]	RO	vstatus	端口状态信号。

接口为 ULPI 时，该寄存器描述如下所示。

		Offset Address	Register Name	Total Reset Value						
		0xA4	INTNREG05	0x0000_0000						
Bit		31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name		vendor control	reserved	vport	access	immediat address	extend address	value		
Reset		0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name		Description						
[31]	R/W	vendor control		向该位写“1”，进行寄存器读写操作。操作完成，该位自动清零。						
[30:28]	-	reserved		保留。						
[27:24]	RW	vport		端口号，不能超过支持的端口数。						
[23:22]	RW	access		寄存器读写操作。 00、01：保留； 10：写寄存器； 11：读寄存器。						
[21:16]	RW	immediat address		寄存器地址。						
[15:8]	RW	extend address		扩展寄存器地址。						
[7:0]	RW	value		寄存器值。 进行写寄存器操作时，为写入寄存器的值； 进行读寄存器操作时，vendor control 清零后，为读寄存器的值。						

INTNREG06

INTNREG06 为 AHB 错误状态寄存器。

Offset Address		Register Name		Total Reset Value							
0xA8		INTNREG06		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	err_capture				reserved				hburst_err	num_beat_err	num_beat_ok
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0			
Bits	Access	Name	Description								
[31]	RW	err_capture	发生了 AHB 错误。								
[30:12]	-	reserved	保留。								
[11:9]	RO	hburst_err	发生 AHB 错误时控制传输阶段 hburst 值。								
[8:4]	RO	num_beat_err	发生 AHB 错误时该次 burst 传输的拍数。最大拍数为 16。 0x00~0x10: 有效; 0x11~0x1F: 保留。								
[3:0]	RO	num_beat_ok	发生 AHB 错误时该次 burst 传输中已经成功完成的拍数。								

INTNREG07

INTNREG07 为 AHB 错误地址寄存器。

Offset Address		Register Name		Total Reset Value				
0xAC		INTNREG07		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	err_addr							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RO	err_addr	发生 AHB 错误时控制传输阶段的地址。					



11 其它外设接口

11.1 I²C 接口

11.1.1 概述

I²C (The Inter-Integrated Circuit) 接口实现标准 I²C Master 和 Slave 功能，兼容 Philips I²C 总线协议，可作为 Master 完成对 I²C 总线上 Slave 数据的发送和接收，也可作为 Slave 对 Master 的发送数据和接收数据请求做出相应的响应。主要用于控制音视频 A/D、D/A 等外部 I²C 器件。

11.1.2 特点

I²C 接口具备以下特性：

- 支持标准 I²C 总线协议
- 支持 Master 和 Slave 操作
- 支持 7bit 和 10bit 设备地址
- 支持可编程时钟，可实现通讯速率控制
- 支持 DMA (Directory Memory Access) 接口
- 支持中断和查询两种工作方式

11.1.3 信号描述

I²C 接口信号描述如表 11-1 所示。



注意

I²C 管脚 SDA 和 SCL 使用 OD 门连接方式以符合 I²C 规范。

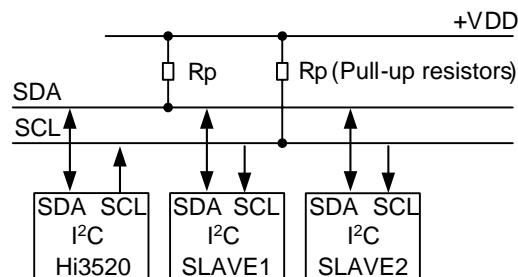
表11-1 I²C 接口信号描述

信号名称	方向	描述	对应管脚
I2C_SDA	I/O	I ² C 双向数据信号。与 GPIO 复用（复用时的配置信息请参见“11.1.5 管脚复用配置”）。	SDA
I2C_SCL	I/O	I ² C 时钟信号。与 GPIO 复用（复用时的配置信息请参见“11.1.5 管脚复用配置”）。	SDL

11.1.4 功能描述

典型应用

I²C 总线典型应用电路如图 1-1 所示。

图11-1 I²C 总线典型应用电路原理图

I²C 总线是一种双线、双向串行总线，它提供了一种简单有效的数据传输方式，可以很大的简化设备间的连接，非常适合多设备间的短距离少量数据传输。I²C 总线的灵活性使得系统开发和设备扩展容易实现。

功能原理

一次典型的 I²C 数据传输主要包括起始信号、从地址发送、数据传送、结束信号等部分。其时序如图 11-2 所示。数据帧格式如图 11-3 所示。

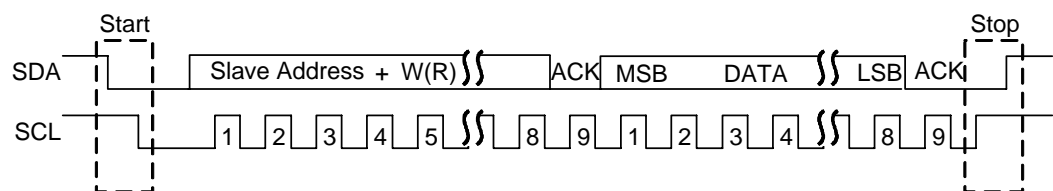
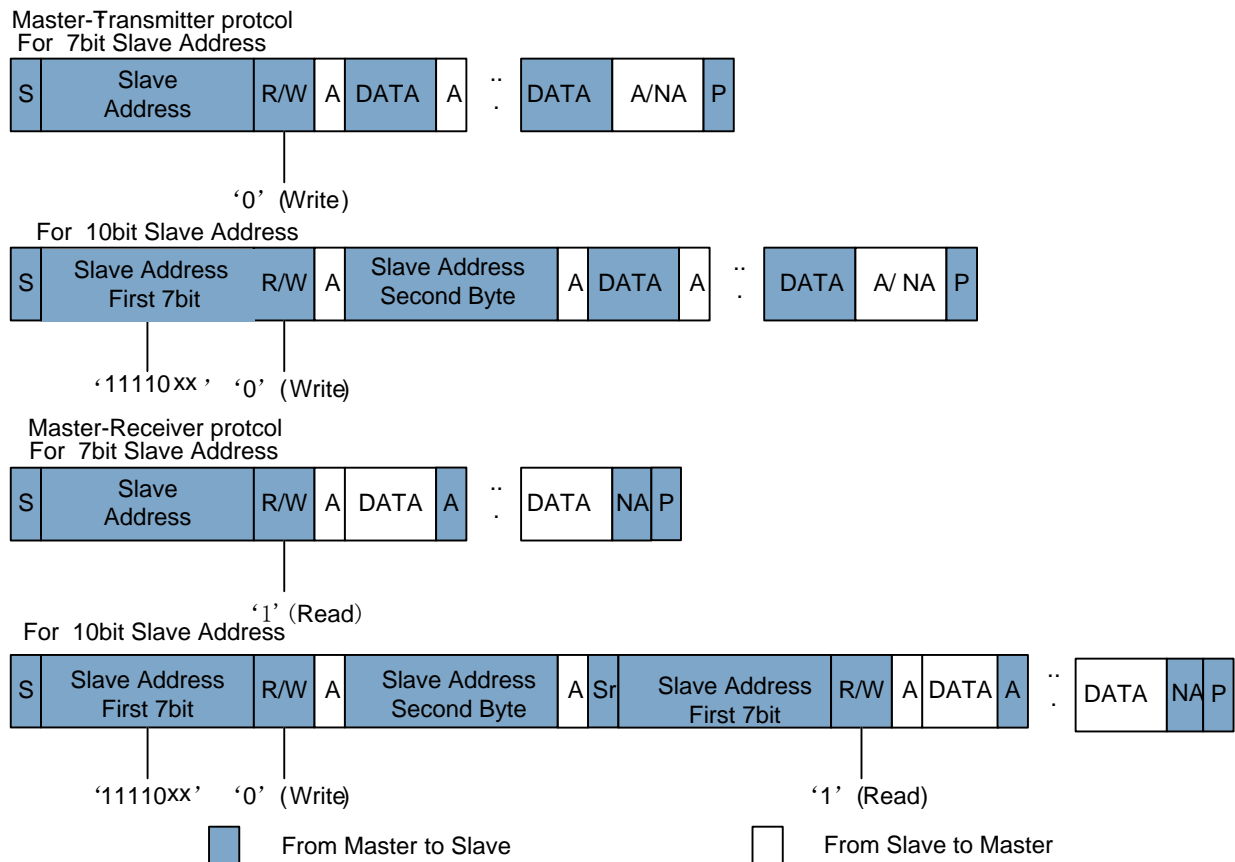
图11-2 I²C 数据传输时序图

图11-3 I²C 数据传输帧格式图

A: Acknowledge(SDA low)

NA: No Acknowledge(SDA high)

S: Start Condition

Sr: repeated Start Condition

P: Stop Condition

起始信号（Start）是 I²C 标准协议规定，由总线上的 Master 发出，用来唤醒所有 Slave 并指示数据传送开始的特殊信号。

从地址发送（Slave Address+W/R）是指 Master 在发出起始信号后所发出的第一个数据包，包括 7 个从地址位（如果是 10bit 从地址，则需发送两个数据包，除 10bit 地址外，剩余位根据协议规定为保留位）和 1 个读/写数据位。各 Slave 依靠该地址信息来确认是否需要传送或接收数据。当从地址被成功接收后，相应 Slave 会在第 9 个时钟周期（SCL）将 SDA（Serial Data）拉低，从而反馈给 Master 一个应答信号（ACK），然后就可以开始进行后续的数据传输。

数据传送（DATA）是指 Master 在成功接收到从地址响应信号后按照读、写命令进行相应的数据接收或发送。在数据传输过程中 SDA 只能在 SCL 为低电平时变化，而在 SCL 高电平时保持；同时接收设备在每接收到 1 个字节后需要给发送设备发出应答信号（ACK）。如果发送设备没有成功接收到应答信号，就会终止数据传输或者重新开始发送。



结束信号（Stop）是 Master 在当前数据传输完成并且没有新的数据传输需要发起时，所发出的 I²C 标准协议规定的表示数据传输结束的特殊信号。当 Master 发出结束信号后 Slave 必须释放总线。

11.1.5 工作方式

管脚复用配置

I²C 管脚 SDA 与 GPIO0_0 进行了复用，可以通过配置 IO Config 寄存器 reg63 bit[0] 选择为 SDA 管脚；

I²C 管脚 SCL 与 GPIO0_1 进行了复用，可以通过配置 IO Config 寄存器 reg64 bit[0] 选择为 SCL 管脚。

时钟配置

I²C Master 模式下需要通过配置 I2C_SS_SCL_HCNT、I2C_SS_SCL_LCNT、I2C_FS_SCL_HCNT、I2C_FS_SCL_LCNT 等寄存器可以设置标准模式和快速模式下 I²C 总线 SCL 信号的高低电平宽度，即相对于 I²C 工作时钟的周期数，其配置值的计算公式分别为： $*_HCNT = T_{scl_h} \times F_{I2C} - 8$ 和 $*_LCNT = T_{scl_l} \times F_{I2C} - 1$ 。

说明

- T_{scl_h} 为 SCL 高电平宽度，单位 cycle。
- T_{scl_l} 为 SCL 低电平宽度，单位 μs 。
- F_{I2C} 为 I²C 工作时钟，单位 MHz。
- I²C Slave 模式下无需配置 I2C_SS_SCL_HCNT、I2C_SS_SCL_LCNT、I2C_FS_SCL_HCNT 和 I2C_FS_SCL_LCNT 寄存器。

I2C_SS_SCL_HCNT、I2C_SS_SCL_LCNT、I2C_FS_SCL_HCNT 和 I2C_FS_SCL_LCNT 典型配置值如表 1-2~表 1-5 所示。

表11-2 I2C_SS_SCL_HCNT 典型配置值

I ² C 总线速率 (kbit/s)	I ² C 工作时钟 (MHz)	SCL 高电平宽度 (μs)	I2C_SS_SCL_HCNT (cycle)
100	100	4	400

表11-3 I2C_SS_SCL_LCNT 典型配置值

I ² C 总线速率 (kbit/s)	I ² C 工作时钟 (MHz)	SCL 低电平宽度 (μs)	I2C_SS_SCL_LCNT (cycle)
100	100	4.7	470



表11-4 I2C_FS_SCL_HCNT 典型配置值

I ² C 总线速率 (kbit/s)	I ² C 工作时钟 (MHz)	SCL 高电平宽度 (μ s)	I2C_FS_SCL_HCNT (cycle)
400	100	0.6	60

表11-5 I2C_FS_SCL_LCNT 典型配置值

I ² C 总线速率 (kbit/s)	I ² C 工作时钟 (MHz)	SCL 低电平宽度 (μ s)	I2C_FS_SCL_LCNT (cycle)
400	100	1.3	130

软复位

通过配置系统控制器 SC_PERCTRL8[i2c_srst]，可实现对 I²C 控制器的单独软复位。复位后各配置寄存器的值均为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

中断或查询方式下的数据传输

Master 模式初始化步骤如下：

- 步骤 1 设定超时标志，检测 I2C_STATUS[activity]（判断 I²C 控制器是否空闲），如果是 1，则延时等待，继续检测，直到 I2C_STATUS[activity]变为 0 或者超时，然后向 I2C_ENABLE[enable]写 0，使 I²C 处于关闭状态。
- 步骤 2 写相应的配置值到 I2C_CON，配置主模式以及传输速率模式等参数。
- 步骤 3 将对接器件的 Slave 地址写入 I2C_TAR[i2c_tar]。
- 步骤 4 配置 I2C_SS_SCL_HCNT、I2C_SS_SCL_LCNT，设定相应的 I²C 总线时钟周期。
- 步骤 5 配置 I2C_RX_TL、I2C_TX_TL，设定相应的 TX_FIFO 及 RX_FIFO 水准值。
- 步骤 6 如果驱动程序采用中断方式，则需设定 I2C_INTR_MASK，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。
- 步骤 7 向 I2C_ENABLE[enable]写 1，使能 I²C，完成初始化配置。

----结束

Slave 模式初始化步骤如下：

- 步骤 1 设定超时标志，检测 I2C_STATUS[activity]（判断 I²C 控制器是否空闲），如果是 1，则延时等待，继续检测，直到 I2C_STATUS[activity]变为 0 或者超时，然后向 I2C_ENABLE[enable]写 0，使 I²C 处于关闭状态。
- 步骤 2 写相应的配置值到 I2C_CON，配置从模式以及传输模式等参数。



- 步骤 3 将作为 Slave 模式时响应的从地址写入 `I2C_SAR[i2c_sar]`。
- 步骤 4 配置 `I2C_RX_TL`、`I2C_TX_TL`，设定相应的 TX_FIFO 及 RX_FIFO 水位值。
- 步骤 5 如果驱动程序采用中断方式则需设定 `I2C_INTR_MASK`，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。
- 步骤 6 向 `I2C_ENABLE[enable]`写 1，使能 I²C，完成初始化配置。

----结束

Master 模式数据发送步骤如下：

- 步骤 1 将发送数据写入 `I2C_DATA_CMD`，启动数据发送。
- 步骤 2 查询方式下，进行连续数据发送时通过读取 `I2C_STATUS` 和 `I2C_TXFLR` 检测 TX_FIFO 状态；中断方式下，则根据相应中断状态位检测。在数据传送完成之前，既要保证 TX_FIFO 中的数据没有溢出（否则会造成数据丢失），同时也要保证 TX_FIFO 中始终有数据（否则 I²C 会认为该次传输已全部完成而发出结束信号）。
- 步骤 3 检测 `I2C_STATUS[tfe]`是否为 1。如果为 1，则表示 I²C 完成全部数据发送；如果为 0，则表示 I²C 未完成全部数据发送，此时会继续进行检测，直到检测到 1 为止。

----结束

Slave 模式数据发送步骤如下：

- 步骤 1 I²C 总线上的其它 Master 发起数据接收操作，并且该操作的地址与 `I2C_SAR` 寄存器中的值相匹配，I²C 使能 `I2C_RAW_INTR_STAT[r_rd_req]`，在中断模式下使能中断。
- 步骤 2 查询方式下，软件检测到 `I2C_RAW_INTR_STAT[r_rd_req]`有效，将需要发送的数据写入 `I2C_DATA_CMD` 寄存器；中断方式下，则根据相应中断状态位检测，发现是 RD_REQ 原始中断，即将需要发送的数据写入 `I2C_DATA_CMD` 寄存器。
- 步骤 3 检测 `I2C_STATUS[tfe]`是否为 1。如果为 1，则表示 I²C 完成数据发送；如果为 0，则表示 I²C 未完成数据发送，此时会继续进行检测，直到检测到 1 为止。

----结束

Master 模式数据接收步骤如下：

- 步骤 1 将读命令（0x0100）写入 `I2C_DATA_CMD`，启动数据接收。
- 步骤 2 进行连续数据接收时，首先要发送次数与接收数据数目相同的读命令（0x0100）到 `I2C_DATA_CMD`（例如要接收 3 个数据，则需发送 3 次读命令到 `I2C_DATA_CMD` 寄存器）。在数据接收完成之前，既要保证 TX_FIFO 中的数据（即读命令 0x100）没有溢出，又保证 TX_FIFO 非空（否则 I²C 会认为没有新的数据需要接收而结束），另外在连续数据接收过程中还需要检测 RX_FIFO 的状态（如果采用中断方式，则根据相应中断状态位检测），避免 RX_FIFO 溢出。
- 步骤 3 检测 `I2C_STATUS[activity]`是否为 0。如果为 0，则表示 I²C 完成全部数据接收；如果为 1，则表示 I²C 未完成全部数据接收，此时会继续进行检测，直到检测到 0 为止。

----结束



Slave 模式数据接收步骤如下：

- 步骤 1 I²C 总线上的其它 Master 发起数据发送操作，并且该操作的地址与 I2C_SAR 寄存器中的值相匹配。
- 步骤 2 I²C 接收数据，将其存放在 RX_FIFO 中。在连续数据接收过程中需要检测 RX_FIFO 的状态（如果采用中断方式，则根据相应中断状态位检测），避免 RX_FIFO 溢出。
- 步骤 3 检测 I2C_RAW_INTR_STAT[r_rx_done]是否为 1。如果为 1，则表示 I²C 完成全部数据接收；如果为 0，则表示 I²C 未完成全部数据接收，此时会继续进行检测，直到检测到 1 为止。

----结束

DMA 方式下的数据传输

初始化步骤如下：

- 步骤 1 设定超时标志，检测 I2C_STATUS[activity]（判断 I²C 控制器是否空闲），如果是 1，则延时等待，继续检测，直到 I2C_STATUS[activity]变为 0 或者超时，然后向 I2C_ENABLE[enable]写 0，使 I²C 处于关闭状态。
- 步骤 2 写相应的配置值到 I2C_CON，配置主模式以及传输速率模式等参数。
- 步骤 3 Master 模式下将对接器件的 Slave 地址写入 I2C_TAR[i2c_tar]。
- 步骤 4 配置 I2C_SS_SCL_HCNT、I2C_SS_SCL_LCNT，设定相应的 I²C 总线时钟周期。
- 步骤 5 配置 I2C_DMA_TDLR、I2C_DMA_RDLR，设定 DMA TX_FIFO 及 RX_FIFO 水线值。
- 步骤 6 向 I2C_ENABLE[enable]写 1，使能 I²C，完成初始化配置。

----结束

数据发送步骤如下：

- 步骤 1 配置 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。
- 步骤 2 配置 I2C_DMA_CR 为 0x2，使能 I²C 的 DMA 发送功能。
- 步骤 3 查询 DMA 中断状态，判断数据是否发送完成，如果完成则关闭 I²C 的 DMA 发送功能。

----结束

数据接收包括数据发送和数据接收两步，首先将读命令发送给 I²C 控制器，然后从 RX_FIFO 中读取接收数据，步骤如下：

- 步骤 4 配置 DMA 数据通道，包括数据传输源（传输源内容包括读命令 0x100 等）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。
- 步骤 5 配置 I2C_DMA_CR 为 0x0003，使能 I²C 的 DMA 发送和接收功能（因数据接收时首先要发送 0x100 到 I2C_DATA_CMD 寄存器，所以要使能 DMA 发送功能）。



步骤 6 通过 DMA 中断状态查询，判断数据是否接收完成，如果完成则关闭 I²C 的 DMA 接收功能。

----结束

11.1.6 寄存器概览

表11-6 I²C 寄存器概览（基地址是 0x200D_0000）

偏移地址	名称	描述	页码
0x0000	I2C_CON	I ² C 控制寄存器	11-9
0x0004	I2C_TAR	I ² C 访问 Slave 地址寄存器	11-10
0x0008	I2C_SAR	I ² C Slave 模式下自身地址寄存器	11-11
0x0010	I2C_DATA_CMD	I ² C 数据通道寄存器	11-12
0x0014	I2C_SS_SCL_HCNT	标准速度下的 SCL 时钟高电平时间配置寄存器	11-12
0x0018	I2C_SS_SCL_LCNT	标准速度下的 SCL 时钟低电平时间配置寄存器	11-13
0x001C	I2C_FS_SCL_HCNT	快速速度下的 SCL 时钟高电平时间配置寄存器	11-13
0x0020	I2C_FS_SCL_LCNT	快速速度下的 SCL 时钟低电平时间配置寄存器	11-14
0x002C	I2C_INTR_STAT	中断状态寄存器	11-14
0x0030	I2C_INTR_MASK	中断屏蔽寄存器	11-16
0x0034	I2C_RAW_INTR_STAT	原始中断状态寄存器	11-18
0x0038	I2C_RX_TL	RX_FIFO 的水线配置寄存器	11-19
0x003C	I2C_TX_TL	TX_FIFO 的水线配置寄存器	11-20
0x0040	I2C_CLR_INTR	组合及独立中断清除寄存器	11-20
0x0044	I2C_CLR_RX_UNDER	RX_UNDER 中断清除寄存器	11-20
0x0048	I2C_CLR_RX_OVER	RX_OVER 中断清除寄存器	11-21
0x004C	I2C_CLR_TX_OVER	TX_OVER 中断清除寄存器	11-21
0x0054	I2C_CLR_TX_ABRT	ABRT 中断清除寄存器	11-22
0x005C	I2C_CLR_ACTIVITY	ACTIVITY 状态寄存器	11-22
0x0060	I2C_CLR_STOP_DET	STOP_DET 中断清除寄存器	11-23
0x0064	I2C_CLR_START_DET	START_DET 中断清除寄存器	11-23



偏移地址	名称	描述	页码
0x0068	I2C_CLR_GEN_CALL	GEN_CALL 中断清除寄存器	11-24
0x006C	I2C_ENABLE	I ² C 工作模式使能寄存器	11-24
0x0070	I2C_STATUS	I ² C 状态寄存器	11-25
0x0074	I2C_TXFLR	TX_FIFO 中的数据个数指示寄存器	11-26
0x0078	I2C_RXFLR	RX_FIFO 中的数据个数指示寄存器	11-26
0x0080	I2C_TX_ABRT_SOURCE	TX_ABRT 的源头中断寄存器	11-27
0x0088	I2C_DMA_CR	I ² C DMA 通道开启控制寄存器	11-28
0x008C	I2C_DMA_TDLR	TX_FIFO 进行 DMA 操作时的水线配置寄存器	11-29
0x0090	I2C_DMA_RDLR	RX_FIFO 进行 DMA 操作时的水线配置寄存器	11-29

11.1.7 寄存器描述

I2C_CON

I2C_CON 为 I²C 控制寄存器。



注意

I2C_CON 只在 I²C 被禁止（即 I2C_ENABLE[enable]为 0）时才可配置。

	Offset Address				Register Name				Total Reset Value							
	0x0000				I2C_CON				0x0075							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								reserved	i2c_restart_en	i2c_10bitaddr_master	reserved	speed		master_mode	
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1
Bits	Access		Name		Description											
[15:7]	-		reserved		保留。											



[6]	RW	reserved	保留。此位只能设置为 1。
[5]	RW	i2c_restart_en	Master 产生“重新开始”条件使能。 0: 禁止, 此时无法实现“重新开始”条件的功能; 1: 使能。
[4]	RW	i2c_10bitaddr_master	发送 7bit 地址/10bit 地址选择。 0: 7bit 地址; 1: 10bit 地址。
[3]	-	reserved	保留。
[2:1]	RW	speed	I ² C 操作速度选择。 00: 非法; 01: 标准速度, 100kbit/s; 10: 快速速度, 400kbit/s; 11: 保留。 注意: 如配置时写入 00 或 11, 会被认为是写入 10。
[0]	RW	master_mode	Master 功能使能。 0: 禁止; 1: 使能。

I2C_TAR

I2C_TAR 为 I²C 访问 Slave 地址寄存器。



注意

I2C_TAR 只在 I²C 被禁止 (即 I2C_ENABLE[enable]为 0) 时才可配置。



	Offset Address				Register Name								Total Reset Value			
	0x0004				I2C_TAR								0x009C			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				special	gc_or_start	i2c_tar									
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0
	Bits	Access	Name		Description											
	[15:12]	-	reserved		保留。											
	[11]	RW	special		general call 和 start byte 功能使能。 0: 禁止; 1: 使能。											
	[10]	RW	gc_or_start		如果 bit[11]为 1, 决定执行功能是 general call 还是 start byte。 0: general call; 1: start byte。											
	[9:0]	RW	i2c_tar		I ² C 作为 Master 时要访问的 Slave 的地址。											

I2C_SAR

I2C_SAR 为 I²C 作为 Slave 时自身的地址寄存器。



注意

I2C_SAR 只在 I²C 被禁止 (即 I2C_ENABLE[enable]为 0) 时才可配置。

	Offset Address				Register Name								Total Reset Value			
	0x0008				I2C_SAR								0x0055			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				i2c_sar											
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
	Bits	Access	Name		Description											
	[15:10]	-	reserved		保留。											
	[9:0]	RW	i2c_sar		I ² C 作为 Slave 时自身的地址。											



I2C_DATA_CMD

I2C_DATA_CMD 为 I²C 数据通道寄存器。

	Offset Address				Register Name								Total Reset Value				
	0x0010				I2C_DATA_CMD								0x0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved								cmd	dat							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access		Name		Description												
[15:9]	-		reserved		保留。												
[8]	RW		cmd		读/写控制位。 0: 写操作, 表示 I ² C 控制器将要向 I ² C 总线发送数据, 此时低 8 位 (DAT) 是 I ² C 要向 I ² C 总线发送的数据。 1: 读操作, 表示 I ² C 控制器将要从 I ² C 总线读回数据。												
[7:0]	RW		dat		将要在 I ² C 总线上发送/接收的数据。 读此 8bit 会读出在 I ² C 总线上接收的数据; 写此 8bit 会把写入的数据发送到 I ² C 总线上。												

I2C_SS_SCL_HCNT

I2C_SS_SCL_HCNT 为标准速度下的 SCL 时钟高电平时间配置寄存器。



注意

I2C_SS_SCL_HCNT 只在 I²C 被禁止 (即 I2C_ENABLE[enable]为 0) 时才可配置。



Offset Address				Register Name								Total Reset Value				
0x0014				I2C_SS_SCL_HCNT								0x007A				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_ss_scl_hcnt															
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0
Bits	Access		Name			Description										
[15:0]	RW		i2c_ss_scl_hcnt			标准速度下的 SCL 时钟高电平时间（计算公式请参见“11.1.5 时钟配置”）。 注意：配置的最小值为 6，写入小于 6 的值时会被认为是 6。										

I2C_SS_SCL_LCNT

I2C_SS_SCL_LCNT 为标准速度下的 SCL 时钟低电平时间配置寄存器。



注意

I2C_SS_SCL_LCNT 只在 I²C 被禁止（即 I2C_ENABLE[enable]为 0）时才可配置。

Offset Address				Register Name								Total Reset Value				
0x0018				I2C_SS_SCL_LCNT								0x008F				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_ss_scl_lcnt															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
Bits	Access		Name			Description										
[15:0]	RW		i2c_ss_scl_lcnt			标准速度下的 SCL 时钟低电平时间（计算公式请参见“11.1.5 时钟配置”）。 注意：配置的最小值为 8，写入小于 8 的值时会被认为是 8。										

I2C_FS_SCL_HCNT

I2C_FS_SCL_HCNT 为快速速度下的 SCL 时钟高电平时间配置寄存器。

**注意**

I2C_FS_SCL_HCNT 只在 I²C 被禁止（即 I2C_ENABLE[enable]为 0）时才可配置。

	Offset Address				Register Name								Total Reset Value			
	0x001C				I2C_FS_SCL_HCNT								0x0013			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_fs_scl_hcnt															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
Bits	Access	Name		Description												
[15:0]	RW	i2c_fs_scl_hcnt		快速速度下的 SCL 时钟高电平时间（计算公式请参见“11.1.5 时钟配置”）。 注意：配置的最小值为 6，写入小于 6 的值时会被认为是 6。												

I2C_FS_SCL_LCNT

I2C_FS_SCL_LCNT 为快速速度下的 SCL 时钟低电平时间配置寄存器。

**注意**

I2C_FS_SCL_LCNT 只在 I²C 被禁止（即 I2C_ENABLE[enable]为 0）时才可配置。

	Offset Address				Register Name								Total Reset Value			
	0x0020				I2C_FS_SCL_LCNT								0x0028			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_fs_scl_lcnt															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Bits	Access	Name		Description												
[15:0]	RW	i2c_fs_scl_lcnt		快速速度下的 SCL 时钟低电平时间（计算公式请参见“11.1.5 时钟配置”）。 注意：配置的最小值为 8，写入小于 8 的值时会被认为是 8。												

I2C_INTR_STAT

I2C_INTR_STAT 为中断状态寄存器。



	Offset Address				Register Name								Total Reset Value			
	0x002C				I2C_INTR_STAT								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				gen_call	start_det	stop_det	activity	rx_done	tx_abrt	rd_req	tx_empty	tx_over	rx_full	rx_over	rx_under
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15:12]	-	reserved		保留。											
	[11]	RO	gen_call		GEN_CALL 中断，一次 general call 请求被接收状态。 0: 未被接收； 1: 已被接收。 I ² C 将接收到的数据放在 RX Buffer 中。											
	[10]	RO	start_det		START_DET 中断，指示在 I ² C 总线上是否发生了开始条件。 0: 未发生； 1: 已发生。											
	[9]	RO	stop_det		STOP_DET 中断，指示在 I ² C 总线上是否发生了停止条件。 0: 未发生； 1: 已发生。											
	[8]	RO	activity		ACTIVITY 中断，指示 I ² C 的 Activity 状态。 0: 空闲； 1: 忙。											
	[7]	RO	rx_done		RX_DONE 中断，当 I ² C 作为 Slave 时，该数据位指示数据接收是否完成。 0: 未完成； 1: 完成。											
	[6]	RO	tx_abrt		TX_ABRT 中断，有多种情况可以触发此位（详细描述请参见“ I2C_TX_ABRT_SOURCE ”）。											
	[5]	RO	rd_req		RD_REQ 中断，当 I ² C 作为 Slave 时，该数据位指示是否有 Master 设备发起读数据请求。 0: 无请求； 1: 有请求。											



[4]	RO	tx_empty	TX_EMPTY 中断, 指示 TX_FIFO 中数据是否到达或低于水位值。 0: 大于水位; 1: 等于或低于水位。
[3]	RO	tx_over	TX_OVER 中断, TX_FIFO 溢出标志。 0: 未溢出; 1: 溢出。
[2]	RO	rx_full	RX_FULL 中断, 指示 RX_FIFO 中数据是否到达或大于水位值。 0: 小于水位; 1: 等于或大于水位。
[1]	RO	rx_over	RX_OVER 中断, RX_FIFO 溢出标志。 0: 未溢出; 1: 溢出。
[0]	RO	rx_under	RX_UNDER 中断, 当接收 FIFO 为空时, 内部总线接口发起了读 I2C_DATA_CMD 的请求。 0: 无意义; 1: RX_FIFO 为空时, CPU 读 I2C_DATA_CMD。

I2C_INTR_MASK

I2C_INTR_MASK 为中断屏蔽寄存器。

	Offset Address 0x0030				Register Name I2C_INTR_MASK								Total Reset Value 0x08FF			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				m_gen_call	m_start_det	m_stop_det	m_activity	m_rx_done	m_tx_abrt	m_rd_req	m_tx_empty	m_tx_over	m_rx_full	m_rx_over	m_rx_under
Reset	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
Bits	Access		Name		Description											
[15:12]	-		reserved		保留。											



[11]	RW	m_gen_call	GEN_CALL 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[10]	RW	m_start_det	START_DET 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[9]	RW	m_stop_det	STOP_DET 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[8]	RW	m_activity	ACTIVITY 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[7]	RW	m_rx_done	RX_DONE 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[6]	RW	m_tx_abrt	TX_ABRT 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[5]	RW	m_rd_req	RD_REQ 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[4]	RW	m_tx_empty	TX_EMPTY 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[3]	RW	m_tx_over	TX_OVER 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[2]	RW	m_rx_full	RX_FULL 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
[1]	RW	m_rx_over	RX_OVER 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。



[0]	RW	m_rx_under	RX_UNDER 中断屏蔽。 0: 屏蔽; 1: 不屏蔽。
-----	----	------------	-------------------------------------

I2C_RAW_INTR_STAT

I2C_RAW_INTR_STAT 为原始中断状态寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x0034				I2C_RAW_INTR_STAT								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				r_gen_call	r_start_det	r_stop_det	r_activity	r_rx_done	r_tx_abrt	r_rd_req	r_tx_empty	r_tx_over	r_rx_full	r_rx_over	r_rx_under
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:12]	-	reserved	保留。													
[11]	RO	r_gen_call	GEN_CALL 原始中断状态。 0: 无中断; 1: 有中断。													
[10]	RO	r_start_det	START_DET 原始中断状态。 0: 无中断; 1: 有中断。													
[9]	RO	r_stop_det	STOP_DET 原始中断状态。 0: 无中断; 1: 有中断。													
[8]	RO	r_activity	ACTIVITY 原始中断状态。 0: 无中断; 1: 有中断。													
[7]	RO	r_rx_done	RX_DONE 原始中断状态。 0: 无中断; 1: 有中断。													
[6]	RO	r_tx_abrt	原始中断状态。 0: 无中断; 1: 有中断。													



[5]	RO	r_rd_req	RD_REQ 原始中断状态。 0: 无中断; 1: 有中断。
[4]	RO	r_tx_empty	R_TX_EMPTY 原始中断状态。 0: 无中断; 1: 有中断。
[3]	RO	r_tx_over	TX_OVER 原始中断状态。 0: 无中断; 1: 有中断。
[2]	RO	r_rx_full	RX_FULL 原始中断状态。 0: 无中断; 1: 有中断。
[1]	RO	r_rx_over	RX_OVER 原始中断状态。 0: 无中断; 1: 有中断。
[0]	RO	r_rx_under	RX_UNDER 原始中断状态。 0: 无中断; 1: 有中断。

I2C_RX_TL

I2C_RX_TL 为 RX_FIFO 的水线配置寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x0038				I2C_RX_TL								0x0003			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								rx_tl							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bits	Access	Name	Description													
[15:8]	-	reserved	保留。													
[7:0]	RW	rx_tl	RX_FIFO 的水线值，实际值等于配置值加 1。 注意：当配置值超过 FIFO 深度（8）时被认为是 8。													



I2C_TX_TL

I2C_TX_TL 为 TX_FIFO 的水线配置寄存器。

		Offset Address				Register Name				Total Reset Value							
		0x003C				I2C_TX_TL				0x0003							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved								tx_tl							
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bits	Access	Name				Description											
[15:8]	RW	reserved				保留。											
[7:0]	RW	tx_tl				TX_FIFO 的水线值。 注意：当配置值超过 FIFO 深度（8）时被认为是 8。											

I2C_CLR_INTR

I2C_CLR_INTR 为组合及独立中断清除寄存器。

		Offset Address				Register Name				Total Reset Value							
		0x0040				I2C_CLR_INTR				0x0000							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved															clr_intr
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name				Description											
[15:1]	-	reserved				保留。											
[0]	RO	clr_intr				读此寄存器，清所有组合中断和独立中断、以及 I2C_TX_ABRT_SOURCE 寄存器。 注意： I2C_TX_ABRT_SOURCE[abrt_sbyte_norstrt] 及其引发的组合中断无法被清除。											

I2C_CLR_RX_UNDER

I2C_CLR_RX_UNDER 为 RX_UNDER 中断清除寄存器。



	Offset Address				Register Name								Total Reset Value			
	0x0044				I2C_CLR_RX_UNDER								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														clr_rx_under	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15:1]	-	reserved		保留。											
	[0]	RO	clr_rx_under		读此寄存器，清 RX_UNDER 中断。											

I2C_CLR_RX_OVER

I2C_CLR_RX_OVER 为 RX_OVER 中断清除寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x0048				I2C_CLR_RX_OVER								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														clr_rx_over	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15:1]	-	reserved		保留。											
	[0]	RO	clr_rx_over		读此寄存器，清 RX_OVER 中断。											

I2C_CLR_TX_OVER

I2C_CLR_TX_OVER 为 TX_OVER 中断清除寄存器。



Offset Address		Register Name		Total Reset Value	
0x004C		I2C_CLR_TX_OVER		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_tx_over
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_tx_over	读此寄存器，清 TX_OVER 中断。		

I2C_CLR_TX_ABRT

I2C_CLR_TX_ABRT 为 ABRT 中断清除寄存器。

Offset Address		Register Name		Total Reset Value	
0x0054		I2C_CLR_TX_ABRT		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_tx_abrt
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_tx_abrt	读此寄存器，清 TX_ABRT 中断和 I2C_TX_ABRT_SOURCE 寄存器。		

I2C_CLR_ACTIVITY

I2C_CLR_ACTIVITY 为 ACTIVITY 状态寄存器。



Offset Address		Register Name		Total Reset Value	
0x005C		I2C_CLR_ACTIVITY		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_activity
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_activity	读此寄存器，可获得 ACTIVITY 中断状态，硬件自动清 0。		

I2C_CLR_STOP_DET

I2C_CLR_STOP_DET 为 STOP_DET 中断清除寄存器。

Offset Address		Register Name		Total Reset Value	
0x0060		I2C_CLR_STOP_DET		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_stop_det
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_stop_det	读此寄存器，清 STOP_DET 中断。		

I2C_CLR_START_DET

I2C_CLR_START_DET 为 START_DET 中断清除寄存器。



Offset Address		Register Name		Total Reset Value	
0x0064		I2C_CLR_START_DET		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_start_det
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_start_det	读此寄存器，清 START_DET 中断。		

I2C_CLR_GEN_CALL

I2C_CLR_GEN_CALL 为 GEN_CALL 中断清除寄存器。

Offset Address		Register Name		Total Reset Value	
0x0068		I2C_CLR_GEN_CALL		0x0000	
Bit	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				clr_gen_call
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description		
[15:1]	-	reserved	保留。		
[0]	RO	clr_gen_call	读此寄存器，清 GEN_CALL 中断。		

I2C_ENABLE

I2C_ENABLE 为 I²C 使能寄存器。用于关闭或使能 I²C。

当 I²C 处于数据传输状态时（即 I2C_STATUS[activity]为 1），I²C 可以被关闭，但需要注意：

- 如果 I²C 处于发送状态时关闭，则 I²C 在完成当前字节发送后停止继续传输，同时清除 TX_FIFO 的数据。
- 如果 I²C 处于接收状态时关闭，则 I²C 在接收完当前字节后不响应这次传输，即发出 NACK。



		Offset Address				Register Name								Total Reset Value			
		0x006C				I2C_ENABLE								0x0000			
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved														enable	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description													
[15:1]	-	reserved		保留。													
[0]	RW	enable		I ² C 的使能寄存器。 0: 禁止; 1: 使能。													

I2C_STATUS

I2C_STATUS 为 I²C 状态寄存器。

		Offset Address				Register Name								Total Reset Value			
		0x0070				I2C_STATUS								0x0006			
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		reserved											rff	rfne	tfe	tfnf	activity
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Bits	Access	Name		Description													
[15:5]	-	reserved		保留。													
[4]	RO	rff		指示 RX_FIFO 是否已满。 0: 未滿; 1: 已滿。													
[3]	RO	rfne		指示 RX_FIFO 是否已空。 0: 已空; 1: 未空。													
[2]	RO	tfe		指示 TX_FIFO 是否已空。 0: 未空; 1: 已空。													



[1]	RO	tfnf	指示 TX_FIFO 是否已满。 0: 已满; 1: 未滿。
[0]	RO	activity	I ² C 总线状态。 0: 空闲; 1: 忙。

I2C_TXFLR

I2C_TXFLR 为 TX_FIFO 中的数据个数指示寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x0074				I2C_TXFLR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											txflr				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access			Name			Description									
[15:4]	-			reserved			保留。									
[3:0]	RO			txflr			指示 TX_FIFO 中的数据个数。									

I2C_RXFLR

I2C_RXFLR 为 RX_FIFO 中的数据个数指示寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x0078				I2C_RXFLR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											rxflr				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access			Name			Description									
[15:4]	-			reserved			保留。									
[3:0]	RO			rxflr			指示 RX_FIFO 中的数据个数。									



I2C_TX_ABRT_SOURCE

I2C_TX_ABRT_SOURCE 为 TX_ABRT 的源头中断寄存器。

	Offset Address 0x0080				Register Name I2C_TX_ABRT_SOURCE								Total Reset Value 0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				arb_master_dis	abrt_10b_rd_norstrt	abrt_sbyte_norstrt	abrt_hs_norstrt	abrt_sbyte_ackdet	abrt_hs_ackdet	abrt_gcall_read	abrt_gcall_noack	abrt_txdata_noack	abrt_10addr2_noack	abrt_10addr1_noack	abrt_7b_addr_noack
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:12]	-		reserved		保留。											
[11]	RO		arb_master_dis		是否由 Master 功能禁用的情况下，尝试发起 Master 操作导致错误。 0: 无此错误； 1: 有此错误。											
[10]	RO		abrt_10b_rd_norstrt		是否由不支持 restart 功能时，作为 Master 时对 10bit 地址的 Slave 发出读命令导致错误。 0: 无此错误； 1: 有此错误。											
[9]	RO		abrt_sbyte_norstrt		是否由不支持 restart 功能时，作为 Master 时尝试发送 start byte 导致错误。 0: 无此错误； 1: 有此错误。											
[8]	RO		abrt_hs_norstrt		是否由不支持 restart 功能时，作为 Master 时尝试高速操作导致错误。 0: 无此错误； 1: 有此错误。											
[7]	RO		abrt_sbyte_ackdet		是否由作为 Master 时发出 start byte 而被响应导致错误。 0: 无此错误； 1: 有此错误。											



[6]	RO	abrt_hs_ackdet	是否由作为 Master 要进行高速传输时，高速主机码被响应导致错误。 0: 无此错误; 1: 有此错误。
[5]	RO	abrt_gcall_read	是否由作为 Master 时发出 general call，而 CPU 向 I ² C 发出读命令导致错误。 0: 无此错误; 1: 有此错误。
[4]	RO	abrt_gcall_noack	是否由作为 Master 时发出 general call，但没被响应导致错误。 0: 无此错误; 1: 有此错误。
[3]	RO	abrt_txdata_noack	是否由作为 Master 发送器，发送的地址被 Slave 响应，而发送的数据没被响应导致错误。 0: 无此错误; 1: 有此错误。
[2]	RO	abrt_10addr2_noack	是否由作为 Master 时，发送的 10bit 地址的第 2 字节没被响应导致错误。 0: 无此错误; 1: 有此错误。
[1]	RO	abrt_10addr1_noack	是否由作为 Master 时，发送的 10bit 地址的第 1 字节没被响应导致错误。 0: 无此错误; 1: 有此错误。
[0]	RO	abrt_7b_addr_noack	是否由作为 Master 时，发送的 7bit 地址没被响应导致错误。 0: 无此错误; 1: 有此错误。

I2C_DMA_CR

I2C_DMA_CR 为 I²C DMA 通道开启控制寄存器。



	Offset Address				Register Name								Total Reset Value			
	0x0088				I2C_DMA_CR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													tdmae	rdmae	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:2]	-		reserved		保留。											
[1]	RW		tdmae		是否打开 TX_FIFO 的 DMA 通道。 0: 不打开; 1: 打开。											
[0]	RW		rdmae		是否打开 RX_FIFO 的 DMA 通道。 0: 不打开; 1: 打开。											

I2C_DMA_TDLR

I2C_DMA_TDLR 为 TX_FIFO 进行 DMA 操作时的流水线配置寄存器。

	Offset Address				Register Name								Total Reset Value			
	0x008C				I2C_DMA_TDLR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													dmatdl		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:3]	-		reserved		保留。											
[2:0]	RW		dmatdl		TX_FIFO DMA 操作时的流水线值。											

I2C_DMA_RDLR

I2C_DMA_RDLR 为 RX_FIFO 进行 DMA 操作时的流水线配置寄存器。



	Offset Address				Register Name								Total Reset Value			
	0x0090				I2C_DMA_RDLR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												dmardl			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name			Description										
[15:3]	-		reserved			保留。										
[2:0]	RW		dmardl			RX_FIFO DMA 操作时的水线值，实际值等于配置值加 1。										

11.2 UART

11.2.1 概述

通用异步收发器 UART (Universal Asynchronous Receiver Transmitter) 是一个异步串行的通信接口，主要功能是将来自外围设备的数据进行串并转换之后传入内部总线，以及将数据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。

Hi3520 提供了以下 4 个 UART 单元：

- UART0
主要用于调试。
- UART1
主要用于接 RS-485 总线和控制云台。
- UART2、UART3
主要用于扩展接口，如外部 MCU (Micro Controller Unit)。

11.2.2 特点

UART 模块有以下特点：

- 支持 $16 \times 8\text{bit}$ 的发送 FIFO 和 $16 \times 12\text{bit}$ 的接收 FIFO。
- 支持数据位和停止位位宽可编程。数据位可通过编程设定为 5/6/7/8 比特；停止位可通过编程设定为 1 比特或 2 比特。
- 支持奇、偶校验方式或者无校验。
- 支持传输速率可编程。
- 支持接收 FIFO 中断、发送 FIFO 中断、接收超时中断、错误中断。
- 支持初始中断状态查询和屏蔽后中断状态查询。



- 支持通过编程禁止 UART 模块或者 UART 发送/接收功能以降低功耗。
- 支持关断 UART 时钟以节省功耗。
- UART0、UART1 和 UART2 支持 DMA 操作，UART3 不支持。

11.2.3 信号描述

UART0 接口信号如表 11-7 所示。

表11-7 UART0 接口信号描述

信号名称	方向	描述	对应管脚
UART0_RXD	I	UART0 接收数据信号。	URXD0
UART0_TXD	O	UART0 发送数据信号。	UTXD0

UART1 接口信号如表 11-8 所示。

表11-8 UART1 接口信号描述

信号名称	方向	描述	对应管脚
UART1_RXD	I	UART1 接收数据信号。	URXD1
UART1_TXD	O	UART1 发送数据信号。	UTXD1
UART1_RTS	O	UART1 请求发送信号。	URTSN1
UART1_CTS	I	UART1 清除发送信号。	UCTSN1

UART2 接口信号如表 11-9 所示。

表11-9 UART2 接口信号描述

信号名称	方向	描述	对应管脚
UART2_RXD	I	UART2 接收数据信号。与 VI 和 GPIO 的部分信号复用。（复用时的配置信息请参见“11.2.5 管脚复用配置”）	VI0HS
UART2_TXD	O	UART2 发送数据信号。与 VI 和 GPIO 接口复用。（复用时的配置信息请参见“11.2.5 管脚复用配置”）	VI0VS

UART3 接口信号如表 11-9 所示。



表11-10 UART3 接口信号描述

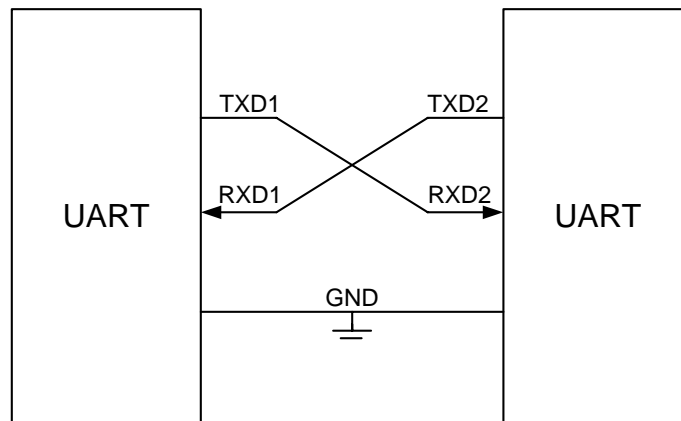
信号名称	方向	描述	对应管脚
UART3_RXD	I	UART3 接收数据信号。与 VI 和 GPIO 接口复用。（复用时的配置信息请参见“11.2.5 管脚复用配置”）	VI2HS
UART3_TXD	O	UART3 发送数据信号。与 VI 和 GPIO 接口复用。（复用时的配置信息请参见“11.2.5 管脚复用配置”）	VI2VS

11.2.4 功能描述

应用框图

UART 的典型应用框图如图 11-4 和图 11-5 所示。

图11-4 UART 的典型应用框图一



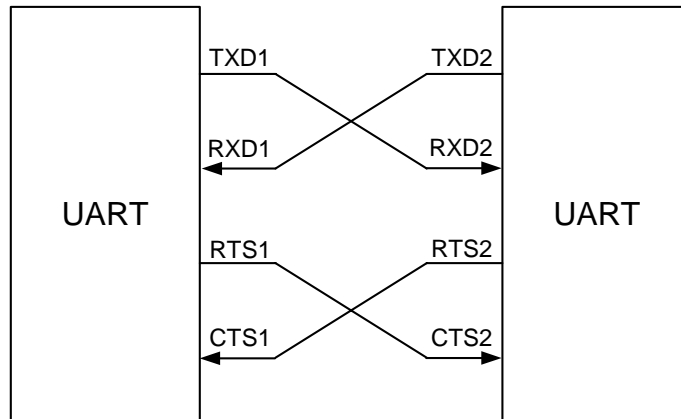
UART 是一种异步双向串行总线，它提供了一种简单有效的数据传输方式，只需要两根数据线互相对接。UART0、UART1、UART2、UART3 供了这种方式。



注意

在配置成 RTS 流控时，不能通过配置 `UART_CR` 寄存器来控制 RTS 的输出值。

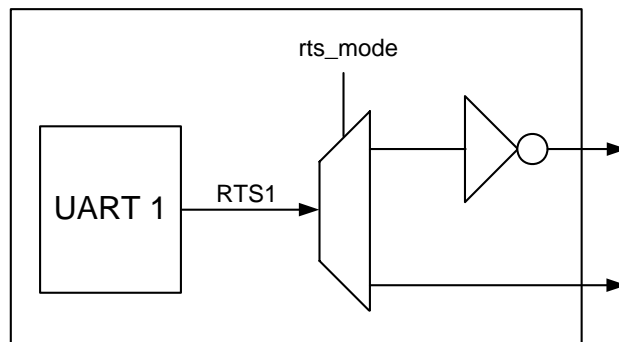
图11-5 UART 的典型应用框图二



如果对接芯片需要 RTS 或者 CTS 流控来控制数据流，也可以通过对接 RTS 和 CTS 管脚达到握手以此控制收发。UART1 对接 RS-485 总线芯片时需要连接 RTS 和 CTS 信号。

RTS 流控信号输出模式如图 11-6 所示。

图11-6 RTS 流控信号输出模式框图



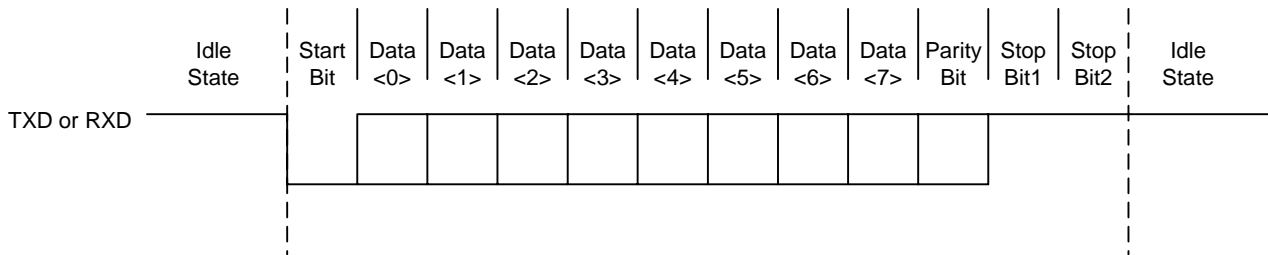
在 RTS 流控下，不能通过 UART_CR 寄存器来配置 RTS 的输出值，但是可以通过配置系统控制器中的 SC_PERCTRL11[uart1_rtsmode]来控制 RTS 输出的电平模式。

功能原理

UART 的一次帧传输主要包括起始信号、数据、校验位和结束信号，如图 11-7 所示。数据帧从某一 UART 的 TXD 端输出，从另一个 UART 的 RXD 端输入。



图11-7 UART 帧格式



起始信号、数据、校验位和结束信号的含义如下：

- 起始信号 (start bit)
一个数据帧开始的标志，UART 协议规定 TXD 信号出现一个低电平就表示一个数据帧的开始。在 UART 不传输数据时，应该保持高电平。
- 数据信号 (data bit)
数据位宽可以根据不同的应用要求进行调整，可以配置成 5/6/7/8 比特数据位宽。
- 校验位 (parity bit)
校验位是 1 比特纠错信号，UART 的校验位有奇校验、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 [UART_LCR_H](#) 寄存器。
- 结束信号 (stop bit)
结束信号即数据帧的停止位，支持 1 比特和 2 比特停止位两种配置。数据帧的结束信号就是把 TXD 拉成高电平。

11.2.5 工作方式

管脚复用配置

UART2 管脚与 VI 和 GPIO 的部分管脚复用。使用 UART2 前，可通过配置寄存器 reg0~reg1 来实现。

UART3 管脚与 VI 和 GPIO 的部分管脚复用，使用 UART3 前，可通过配置寄存器 reg18~reg19 来实现。

时钟门控

在软件完成当前数据传输且未启动新的数据传输的情况下可关断 UART 时钟，但需要确保硬件已处于空闲状态（即 [UART_FR](#) bit[3]为 0 时）。关断 UART 时钟的步骤如下：

- 步骤 1 读 [UART_FR](#)。
- 步骤 2 若 [UART_FR](#) bit[3]为 0，则配置 [UART_CR](#)[uarten]=0，进入步骤 3；若 [UART_FR](#) bit[3]为 1，则延时等待，返回步骤 1。
- 步骤 3 配置系统控制器 [SC_PERDIS](#)[7:4]=0xF，关闭 UART 时钟。

----结束



波特率配置

通过配置寄存器 `UART_IBRD` 和 `UART_FBRD` 可以设置 UART 工作的波特率，波特率计算公式为：当前波特率=UART 参考时钟频率/（16×分频系数）。

分频系数有整数和小数两部分组成，分别对应寄存器 `UART_IBRD` 和 `UART_FBRD`。

例如：内部总线时钟为 60MHz，如果配置 `UART_IBRD` 为 0x1E，`UART_FBRD` 为 0x00，按照波特率计算公式，则当前的波特率为 $60/(16 \times 30) = 0.125\text{Mbit/s}$ 。

UART 波特率配置的典型值为：9,600bit/s、14,400bit/s、19,200bit/s、38,400bit/s、57,600bit/s、76,800bit/s、115,200bit/s、230,400bit/s、460,800bit/s。

分频系数值的计算以及分频系数寄存器的配置举例如下：

如果要求波特率为 230400bit/s，并且 `UARTCLK` 为 100MHz，那么分频系数为 $(100 \times 10^6)/(16 \times 230400) = 27.1267$ ，因此 `IBRD`（整数部分）为 27，`FBRD`（小数部分）为 0.1267。

计算 6bit `UART_FBRD` 寄存器中的数值：根据 $m = \text{integer}(\text{FBRD} \times 2^n + 0.5)$ （ $n = \text{UART_FBRD}$ 寄存器的宽度），计算出 $m = \text{integer}(0.1267 \times 2^6 + 0.5) = 8$ ，在 `UART_IBRD` 寄存器中配置 0x001B，`UART_FBRD` 寄存器中配置 0x08。

当分频系数小数部分配置成 8 时，波特率除数的实际数值为 $27 + 8/64 = 27.125$ ，产生的波特率为 $(100 \times 10^6) / (16 \times 27.125) = 230414.75$ ，误差率为 $(230414.75 - 230400) / 230400 \times 100 = 0.006\%$ 。

使用 6bit `UART_FBRD` 寄存器最大的误差率为 $1/64 \times 100 = 1.56\%$ ，当 $m=1$ 时会出现，误差率累计超过 64 个时钟周期。

软复位

通过配置系统控制器可实现对 UART 控制器的单独软复位。

- 配置系统控制器 `SC_PERCTRL8` bit[6]为 1，可实现对 UART0 控制器的单独软复位。
- 配置系统控制器 `SC_PERCTRL8` bit[7]为 1，可实现对 UART1 控制器的单独软复位。
- 配置系统控制器 `SC_PERCTRL8` bit[8]为 1，可实现对 UART2 控制器的单独软复位。
- 配置系统控制器 `SC_PERCTRL8` bit[9]为 1，可实现对 UART3 控制器的单独软复位。

复位后各配置寄存器的值均为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

中断或查询方式下的数据传输

初始化步骤如下：

步骤 1 向 `UART_CR` bit[0]写 0，使 UART 处于禁止状态。

步骤 2 写相应的配置值到 `UART_IBRD`、`UART_FBRD` 寄存器，配置传输速率。



- 步骤 3 配置 `UART_CR`、`UART_LCR_H`，设定相应的 UART 工作模式。
- 步骤 4 配置 `UART_IFLS` 设定相应的发送及接收 FIFO 阈值。
- 步骤 5 如果驱动程序采用中断方式则需设定 `UART_IMSC`，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。
- 步骤 6 向 `UART_CR` bit[0]写 1，使能 UART，完成初始化配置。

----结束

数据发送步骤如下：

- 步骤 7 将发送数据写入 `UART_DR`，启动数据发送。
- 步骤 8 查询方式下，进行连续数据发送时通过读取 `UART_FR` bit[5]检测 TX_FIFO 状态，根据 TX_FIFO 的状态决定是否向 TX_FIFO 中发送数据；中断方式下，则根据相应中断状态位检测；决定是否向 TX_FIFO 中发送数据。
- 步骤 9 通过检测 `UART_FR` bit[7]是否为 1，判断 UART 是否完成全部数据发送。

----结束

数据接收的处理方式如下：

- 查询方式下，进行数据接收时通过读取 `UART_FR` bit[4]检测 RX_FIFO 状态，根据 RX_FIFO 的状态决定是否读取 RX_FIFO 中的数据。
- 中断方式下，则根据相应中断状态位检测决定是否读取 RX_FIFO 中的数据。

DMA 方式下的数据传输



注意

UART0、UART1 和 UART2 支持 DMA 传输方式，而 UART3 不支持 DMA 传输方式。

初始化步骤如下：

- 步骤 1 向 `UART_CR` bit[0]写 0，使 UART 处于禁止状态。
- 步骤 2 写相应的配置值到 `UART_IBRD`、`UART_FBRD` 寄存器，配置传输速率。
- 步骤 3 配置 `UART_CR`、`UART_LCR_H`，设定相应的 UART 工作模式。
- 步骤 4 配置 `UART_IFLS` 设定相应的发送及接收 FIFO 阈值。
- 步骤 5 如果驱动程序采用中断方式则需设定 `UART_IMSC`，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。
- 步骤 6 向 `UART_CR` bit[0]写 1，使能 UART，完成初始化配置。

----结束



数据发送步骤如下：

- 步骤 1 配置 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“3.5 直接存储器存取控制器”的相关描述。
- 步骤 2 配置 `UART_DMCCR` 为 0x2，使能 UART 的 DMA 发送功能。
- 步骤 3 通过 DMA 中断上报，判断数据是否发送完成，如果完成则关闭 UART 的 DMA 发送功能。

----结束

数据接收步骤如下：

- 步骤 1 配置 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。
- 步骤 2 配置 `UART_DMCCR` 为 0x1，使能 UART 的 DMA 接收功能。
- 步骤 3 通过 DMA 状态查询，判断数据是否接收完成，如果完成则关闭 UART 的 DMA 接收功能。

----结束

11.2.6 寄存器概览

Hi3520 提供 4 个 UART 单元，UART0、UART1、UART2 和 UART3 的基址分别如下：

- UART0 寄存器基址是 0x2009_0000。
- UART1 寄存器基址是 0x200A_0000。
- UART2 寄存器基址是 0x200B_0000。
- UART3 寄存器基址是 0x200C_0000。

UART 寄存器概览如表 11-11 所示。

表11-11 UART 寄存器概览

偏移地址	名称	描述	页码
0x000	UART_DR	数据寄存器	11-38
0x004	UART_RSR	接收状态寄存器/错误清除寄存器	11-39
0x008~0x014	RESERVED	保留	-
0x018	UART_FR	标志寄存器	11-40
0x01C~0x020	RESERVED	保留	-
0x024	UART_IBRD	整数波特率寄存器	11-41
0x028	UART_FBRD	小数波特率寄存器	11-41



偏移地址	名称	描述	页码
0x02C	UART_LCR_H	线控寄存器	11-42
0x030	UART_CR	控制寄存器	11-44
0x034	UART_IFLS	中断 FIFO 阈值选择寄存器	11-45
0x038	UART_IMSC	中断屏蔽寄存器	11-46
0x03C	UART_RIS	原始中断状态寄存器	11-47
0x040	UART_MIS	屏蔽后中断状态寄存器	11-48
0x044	UART_ICR	中断清除寄存器	11-49
0x048	UART_DMACR	DMA 控制寄存器	11-50

11.2.7 寄存器描述

UART_DR

UART_DR 为 UART 数据寄存器，存放接收数据和发送数据，同时可以从该寄存器中读出接收状态。

	Offset Address					Register Name					Total Reset Value					
	0x000					UART_DR					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oe	be	pe	fe	data						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	[15:12]		[11]		[10]		[9]									
Access	-		RO		RO		RO									
Name	reserved		oe		be		pe									
Description	保留。		溢出错误。 0: 无溢出错误; 1: 有溢出错误, 接收 FIFO 满且接收了一个数据。		Break 错误。 0: 无 break 错误; 1: 有 break 错误, 即接收数据的输入保持低的时间比一个全字传输 (包括 start、data、parity、stop bit) 还要长。		校验错误。									



			0: 无校验错误; 1: 有校验错误。
[8]	RO	fe	帧错误。 0: 无帧错误; 1: 有帧错误 (错误的停止位)。
[7:0]	RW	data	接收数据和发送数据。

UART_RSR

UART_RSR 为接收状态寄存器/错误清除寄存器。

- 寄存器读时作为接收状态寄存器。
- 寄存器写时作为错误清除寄存器。

接收状态也可以从 [UART_DR](#) 中读出。从 [UART_DR](#) 中读出的 break、frame、parity 的状态信息要比从 UART_RSR 读出的信息优先级高 (即 UART_DR 中的状态变化比 UART_RSR 更快)。

对 UART_RSR 寄存器的任何写操作都会对 UART_RSR 寄存器进行复位。

	Offset Address				Register Name		Total Reset Value	
	0x004				UART_RSR		0x00	
Bit	7	6	5	4	3	2	1	0
Name	reserved				oe	be	pe	fe
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:4]	-	reserved	保留。					
[3]	RW	oe	溢出错误。 0: 无溢出错误; 1: 溢出错误。 当 FIFO 满时, FIFO 中的内容保持有效, 因为不会有下一个数据写到 FIFO 中, 只是移位寄存器会溢出。CPU 必须立刻读数据以腾空 FIFO。					



[2]	RW	be	Break 错误。 0: 无 break 错误; 1: break 错误。 Break 的条件: 接收数据的输入保持低的时间比一个全字传输 (定义了 start、data、parity、stop bit) 还要长。
[1]	RW	pe	校验错误。 0: 无校验错误; 1: 接收数据的校验错误。 FIFO 模式下, 该错误与 FIFO 顶部的数据相关联。
[0]	RW	fe	帧错误。 0: 无帧错误; 1: 接收到的数据的停止位错误 (有效的停止位为 1)。

UART_FR

UART_FR 为 UART 标志寄存器。

	Offset Address	Register Name	Total Reset Value												
	0x018	UART_FR	0x0012												
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
Name	reserved							txfe	rxff	txff	rxfe	busy	reserved		
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Bits	Access	Name	Description												
[15:8]	-	reserved	保留。												
[7]	RO	txfe	该位的含义由 UART_LCR_H[fen] 的状态决定。 如果 UART_LCR_H[fen] 为 0, 则当发送 holding register 空时该位置 1; 如果 UART_LCR_H[fen] 为 1, 则当发送 FIFO 为空时该位置 1。												
[6]	RO	rxff	该位的含义由 UART_LCR_H[FEN] 的状态决定。 如果 UART_LCR_H[fen] 为 0, 则当接收 holding register 满时该位置 1; 如果 UART_LCR_H[fen] 为 1, 则当接收 FIFO 为满时该位置 1。												



[5]	RO	txff	该位的含义由 UART_LCR_H[FEN] 的状态决定。 如果 UART_LCR_H[fen] 为 0，则当发送 holding register 满时该位置 1； 如果 UART_LCR_H[fen] 为 1，当发送 FIFO 为满时该位置 1。
[4]	RO	rxfe	该位的含义由 UART_LCR_H[FEN] 的状态决定。 如果 UART_LCR_H[fen] 为 0，则当接收 holding register 空时该 bit 置 1； 如果 UART_LCR_H[fen] 为 1，则当接收 FIFO 为空时该位就置 1。
[3]	RO	busy	UART 忙闲状态位。 0: UART 空闲或者完成发送数据； 1: UART 正忙于发送数据。 该位一旦置位，该状态一直保持到整个字节（包括所有的停止位）完全从移位寄存器中发送出去。 一旦发送 FIFO 非空该位就置位，不管 UART 使能与否。
[2:0]	-	reserved	保留。

UART_IBRD

UART_IBRD 为整数波特率寄存器。

	Offset Address					Register Name					Total Reset Value					
	0x024					UART_IBRD					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	baud divint															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15:0]	RW	baud divint		整数波特率分频值。复位时全部清 0。											

UART_FBRD

UART_FBRD 为小数波特率寄存器。

**注意**

- 整数波特率寄存器和小数波特率寄存器的值必须等到当前数据发送和接收完毕才能更新。
- 最小的分频值为 1，最大的分频值为 65535 ($2^{16}-1$)。即 `UART_IBRD=0` 是无效的，而此时 `UART_FBRD` 将被忽略。同样，如果 `UART_IBRD=65535` (`0xFFFF`)，`UART_FBRD` 就只能是 0，如果比 0 大，则会导致发送和接收的失败。
- 假设 `UART_FBRD=0x1E`、`UART_IBRD=0x01`，这就表示分频系数的整数部分为 30，小数部分为 0.015625，整个分频系数为 30.015625。
- $\text{UART 的波特率} = \text{内部总线频率} / (16 \times \text{分频系数}) = \text{内部总线频率} / (16 \times 30.015625)$ 。

	Offset Address		Register Name		Total Reset Value			
	0x028		UART_FBRD		0x00			
Bit	7	6	5	4	3	2	1	0
Name	reserved		baud divfrac					
Reset	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description				
	[7:6]	-	reserved	保留。				
	[5:0]	RW	band divfrac	小数波特率分频值。复位时全部清 0。				

UART_LCR_H

UART_LCR_H 为传输模式控制寄存器，`UART_LCR_H`、`UART_IBRD`、`UART_FBRD` 组成一个 30bit 宽的寄存器。如果更新 `UART_IBRD` 和 `UART_FBRD` 的内容，必须同时更新 `UART_LCR_H`。

	Offset Address		Register Name		Total Reset Value											
	0x02C		UART_LCR_H		0x0000											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								sps	wlen	fen	stp2	eps	pen	brk	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description												
	[15:8]	-	reserved	保留。												



[7]	RW	sps	<p>校验选择。</p> <p>当本寄存器的 bit[1]、bit[2]、bit[7]被置位时，校验位就会作为 0 发送和检测；</p> <p>当本寄存器的 bit[1]、bit[7]被置位，bit[2]为 0 时，校验位就会作为 1 发送和检测。</p> <p>当 bit[1]、bit[2]、bit[7]都清 0，则 stick parity 禁止。</p>
[6:5]	RW	wlen	<p>指示发送和接收一个帧里数据比特的数目。</p> <p>00: 5bit; 01: 6bit; 10: 7bit; 11: 8bit。</p>
[4]	RW	fen	<p>发送和接收 FIFO 使能控制。</p> <p>0: 发送和接收 FIFO 禁止； 1: 发送和接收 FIFO 使能。</p>
[3]	RW	stp2	<p>发送帧尾 2bit 停止位判断。</p> <p>0: 发送的帧尾没有 2bit 停止位； 1: 发送的帧尾有 2bit 停止位。</p> <p>接收逻辑在接收时不检查 2bit 的停止位。</p>
[2]	RW	eps	<p>发送和接收过程中的奇偶校验选择。</p> <p>0: 在发送和接收过程中生成奇校验或检查奇校验； 1: 在发送和接收过程中生成偶校验或检查偶校验。</p> <p>当 <code>UART_LCR_H[fen]</code> 为 0 时，该位不起作用。</p>
[1]	RW	pen	<p>校验选择位。</p> <p>0: 不作校验； 1: 发送方向产生校验，接收方向作校验检查。</p>
[0]	RW	brk	<p>发送 break。</p> <p>0: 无效； 1: 在完成当前数据的发送后，UTXD 连续输出低电平。</p> <p>注意：要正确的执行 break 命令，软件将该位置 1 的时间必须超过 2 个完整帧；在正常使用中，该位必须清 0。</p>



UART_CR

UART_CR 为 UART 控制寄存器。

配置 UART_CR 遵循以下步骤：

- 步骤 1 向 UART_CR[0]写 0，禁止 UART。
- 步骤 2 等待当前数据发送或接收结束。
- 步骤 3 将 UART_LCR_H[fen]清 0。
- 步骤 4 配置 UART_CR。
- 步骤 5 向 UART_CR[0]写 1，使能 UART。

----结束

	Offset Address 0x030				Register Name UART_CR								Total Reset Value 0x0300			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ctsen	rtsen	reserved		rts	dtr	rx	txe	lbe							uarten
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15]	RW	ctsen		CTS 硬件流控使能。 0：不使能 CTS 硬件流控； 1：使能 CTS 硬件流控，只有当 nUARTCTS 信号有效时才发送数据。											
	[14]	RW	rtsen		RTS 硬件流控使能。 0：不使能 RTS 硬件流控； 1：使能 RTS 硬件流控，只有当接收 FIFO 有空间时才请求接收数据。											
	[13:12]	-	reserved		保留。											
	[11]	RW	rts		请求发送。 该 bit 为 UART modem 状态输出信号 nUARTRTS 的取反。 0：输出信号不变； 1：即该 bit 配置为 1，则输出信号为 0。											



[10]	RW	dtr	<p>数据发送准备。</p> <p>该 bit 为 UART modem 状态输出信号 nUARTDTR 的取反。</p> <p>0: 输出信号不变;</p> <p>1: 即该 bit 配置为 1, 则输出信号为 0。</p>
[9]	RW	rxen	<p>UART 接收使能。</p> <p>0: 禁止;</p> <p>1: 使能。</p> <p>在接收的过程中如果 UART 被禁止, 则当前数据的接收就会在正常停止之前结束。</p>
[8]	RW	txen	<p>UART 发送使能。</p> <p>0: 禁止;</p> <p>1: 使能。</p> <p>在发送的过程中如果 UART 被禁止, 则当前数据的发送就会在正常停止之前结束。</p>
[7]	RW	lbe	<p>环回使能。</p> <p>0: 禁止;</p> <p>1: UARTTXD 输出环回到 UARTRXD。</p>
[6:1]	-	reserved	保留。
[0]	RW	uarten	<p>UART 使能。</p> <p>0: 禁止;</p> <p>1: 使能。</p> <p>如果在发送和接收过程中将 UART 禁止, 则会在正常停止之前结束当前数据的传送。</p>

UART_IFLS

UART_IFLS 为中断 FIFO 阈值选择寄存器, 用于设置 FIFO 的中断 (UART_TXINTR 或 UART_RXINTR) 触发线。

Offset Address		Register Name		Total Reset Value												
0x034		UART_IFLS		0x0012												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										rxifsel		txifsel			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Bits	Access	Name	Description													
[15:6]	-	reserved	保留。													
[5:3]	RW	rxifsel	接收中断 FIFO 的阈值选择，接收中断的触发点如下。 000: 接收 FIFO \geq 1/8full; 001: 接收 FIFO \geq 1/4full; 010: 接收 FIFO \geq 1/2full; 011: 接收 FIFO \geq 3/4full; 100: 接收 FIFO \geq 7/8full; 101~111: 保留。													
[2:0]	RW	txifsel	发送中断 FIFO 的阈值选择，发送中断的触发点如下。 000: 发送 FIFO \leq 1/8full; 001: 发送 FIFO \leq 1/4full; 011: 发送 FIFO \leq 3/4full; 010: 发送 FIFO \leq 1/2full; 100: 发送 FIFO \leq 7/8full; 101~111: 保留。													

UART_IMSC

UART_IMSC 为中断屏蔽寄存器，用于屏蔽中断。

Offset Address		Register Name		Total Reset Value												
0x038		UART_IMSC		0x0000												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oeim	beim	peim	feim	rtim	txim	rxim	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:11]	-	reserved	保留。													



[10]	RW	oeim	溢出错误中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[9]	RW	beim	break 错误中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[8]	RW	peim	校验中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[7]	RW	feim	帧错误中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[6]	RW	rtim	接收超时中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[5]	RW	txim	发送中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[4]	RW	rxim	接收中断的屏蔽状态。 0: 屏蔽该中断; 1: 不屏蔽该中断。
[3:0]	-	reserved	保留。

UART_RIS

UART_RIS 为原始中断状态寄存器，其内容不受中断屏蔽寄存器的影响。

	Offset Address				Register Name				Total Reset Value							
	0x03C				UART_RIS				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				oeris	beris	peris	feris	rtris	txris	rxris	reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:11]	-		reserved		保留。											



[10]	RO	oeris	原始的溢出错误中断状态。 0: 未产生中断; 1: 已产生中断。
[9]	RO	beris	原始的 break 错误中断状态。 0: 未产生中断; 1: 已产生中断。
[8]	RO	peris	原始的校验中断状态。 0: 未产生中断; 1: 已产生中断。
[7]	RO	feris	原始的错误中断状态。 0: 未产生中断; 1: 已产生中断。
[6]	RO	rtris	原始的接收超时中断状态。 0: 未产生中断; 1: 已产生中断。
[5]	RO	txris	原始的发送中断状态。 0: 未产生中断; 1: 已产生中断。
[4]	RO	rxris	原始的接收中断状态。 0: 未产生中断; 1: 已产生中断。
[3:0]	-	reserved	保留。

UART_MIS

UART_MIS 为屏蔽后中断状态寄存器，其内容为原始中断状态和中断屏蔽进行“与”操作后的结果。

	Offset Address				Register Name				Total Reset Value							
	0x040				UART_MIS				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				oemis	bemis	pemis	femis	rtmis	txmis	rxmis	reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:11]	-		reserved		保留。											



[10]	RO	oemis	屏蔽后的溢出错误中断状态。 0: 未产生中断; 1: 已产生中断。
[9]	RO	bemis	屏蔽后的 break 错误中断状态。 0: 未产生中断; 1: 已产生中断。
[8]	RO	pemis	屏蔽后的校验中断状态。 0: 未产生中断; 1: 已产生中断。
[7]	RO	femis	屏蔽后的错误中断状态。 0: 未产生中断; 1: 已产生中断。
[6]	RO	rtmis	屏蔽后的接收超时中断状态。 0: 未产生中断; 1: 已产生中断。
[5]	RO	txmis	屏蔽后的发送中断状态。 0: 未产生中断; 1: 已产生中断。
[4]	RO	rxmis	屏蔽后的接收中断状态。 0: 未产生中断; 1: 已产生中断。
[3:0]	-	reserved	保留。

UART_ICR

UART_ICR 为中断清除寄存器，写 1 时相应的中断被清除，写 0 则不起作用。

	Offset Address				Register Name				Total Reset Value							
	0x044				UART_ICR				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				oeic	beic	peic	feic	rtic	txic	rxic	reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access		Name		Description											
[15:11]	-		reserved		保留。											



[10]	WO	oeic	清除溢出错误中断。 0: 无效; 1: 清除中断。
[9]	WO	beic	清除 break 错误中断。 0: 无效; 1: 清除中断。
[8]	WO	peic	清除校验中断。 0: 无效; 1: 清除中断。
[7]	WO	feic	清除错误中断。 0: 无效; 1: 清除中断。
[6]	WO	rtic	清除接收超时中断。 0: 无效; 1: 清除中断。
[5]	WO	txic	清除发送中断。 0: 无效; 1: 清除中断。
[4]	WO	rxic	清除接收中断。 0: 无效; 1: 清除中断。
[3:0]	-	reserved	保留。

UART_DMACR

UART_DMACR 为 DMA 控制寄存器，用于配置发送 FIFO 和接收 FIFO 的 DMA 使能。



Offset Address		Register Name		Total Reset Value												
0x048		UART_DMACR		0x0000												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													dmaonerr	txdmae	rxdmae
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:3]	-	reserved	保留。													
[2]	RW	dmaonerr	UART 错误中断 (UARTEINTR) 出现时的接收通道 DMA 使能控制。 0: 当 UART 错误中断 (UARTEINTR) 有效时, 接收通道 DMA 的请求输出 (UARTRXDMSREQ 或 UARRTXDMABREQ) 有效; 1: 当 UART 错误中断 (UARTEINTR) 有效时, 接收通道 DMA 的请求输出 (UARTRXDMSREQ 或 UARRTXDMABREQ) 无效。													
[1]	RW	txdmae	发送 FIFO 的 DMA 使能控制。 0: 禁止; 1: 使能。													
[0]	RW	rxdmae	接收 FIFO 的 DMA 使能控制。 0: 禁止; 1: 使能。													

11.3 SPI

11.3.1 概述

SPI (Synchronous Peripheral Interface) 控制器, 可以作为一个 master 或 slave 与外部的设备来进行同步串行通信。支持以下接口协议:

- A Motorola SPI (Synchronous Peripheral Interface) -compatible interface
- A Texas Instruments synchronous serial interface
- A National Semiconductor Microwire interface

SPI 主要应用于外接触摸屏、SD 卡、WiFi 等。



11.3.2 特点

SPI 模块有以下特点：

- 支持主操作，最多支持 2 个 slave。
- 支持从操作。
- 支持接口时钟频率可编程。
- 收/发分开的 16bit 宽、深度为 8 的 FIFO（发送 FIFO 和接收 FIFO 各一个）。
- 支持三种帧格式：SPI、Microwire、TI synchronous serial。
- 数据帧大小可编程：4bit~16bit。
- 内部提供环回测试模式。
- 支持 DMA 操作。

11.3.3 信号描述

SPI 接口信号如表 11-12 所示。

表11-12 SPI 接口信号描述

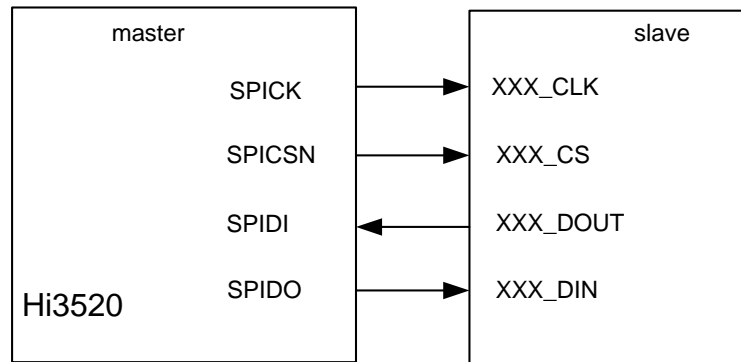
信号名称	方向	描述	对应管脚
SPICK	I/O	SPI 时钟输出，只与 VO 管脚复用。（复用时的配置信息请参见“管脚复用配置”）	VO1DAT8
SPIDI	I	SPI 的数据输入，只与 VO 管脚复用。（复用时的配置信息请参见“管脚复用配置”）	VO1DAT9
SPIDO	O	SPI 的数据输出，只与 VO 管脚复用。（复用时的配置信息请参见“管脚复用配置”）	VO1DAT10
SPICSN0 /SPICSN1	I/O	当 SPI 配置为 SPI 和 Microwire 帧格式时，该信号作片选信号，当 SPI 配置为 TI 帧格式时，该信号做帧同步信号使用。（复用时的配置信息请参见“管脚复用配置”）	VO1DAT11 /VO1DAT12

11.3.4 功能描述

应用框图

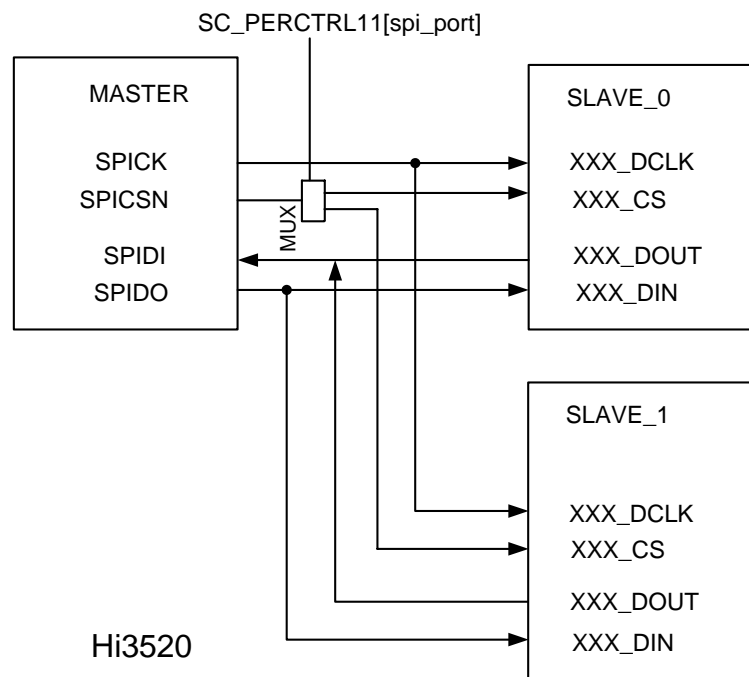
SPI 接单 slave 时的应用框图如图 11-8 所示。SPI 接单 slave 时，使用的是 SPI 默认的片选管脚 SPICSN0。

图11-8 当 SPI 接单 slave 时的应用



SPI 接单 slave 时的应用框图如图 11-9 所示。当 SPI 外接 2 个 slave 器件时，SPI 自带的片选信号已不能满足要求，此时通过系统控制器 SC_PERCTRL11[spi_port]配置，来选择将片选信号送给 SPICSN0 和 SPICSN1 其中一个管脚。两个管脚同时只能有一个有效，SC_PERCTRL11[spi_port]配置为 0 时，SPICSN0 有效；SC_PERCTRL11[spi_port]配置为 1 时，SPICSN1 有效。

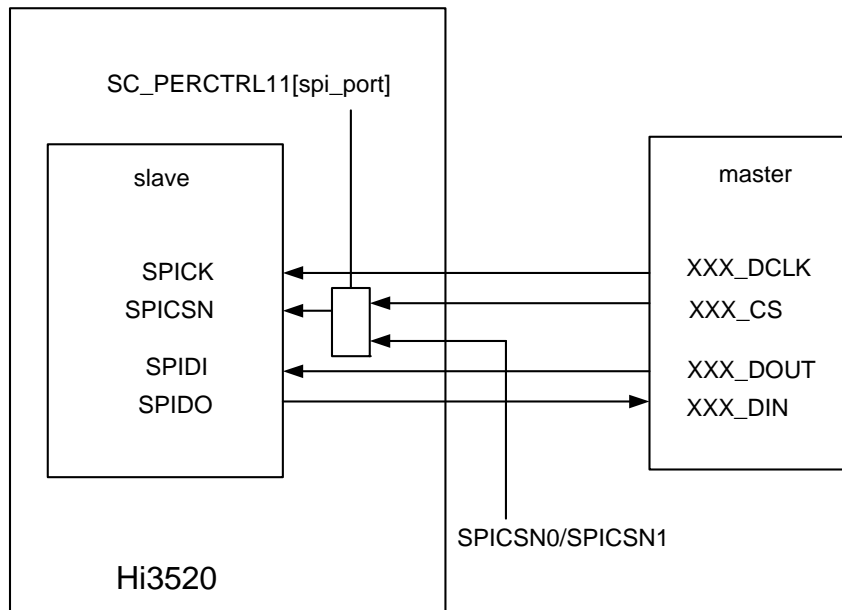
图11-9 当 SPI 接单 slave 时的应用



当 SPI 做为 slave 时的应用框图如图 11-10 所示。外部的 Master 设备将选择 SPICSN0 和 SPICSN1 中的一个片选管脚做为输入连接，注意此时确认系统控制器 SC_PERCTRL11[spi_port]配置，外部的 Master 设备的片选信号才能正确输入 SPI 接口。推荐使用 SPI 的默认片选接口 SPICSN0。



图11-10 当 SPI 作 slave 时的应用



功能原理

说明

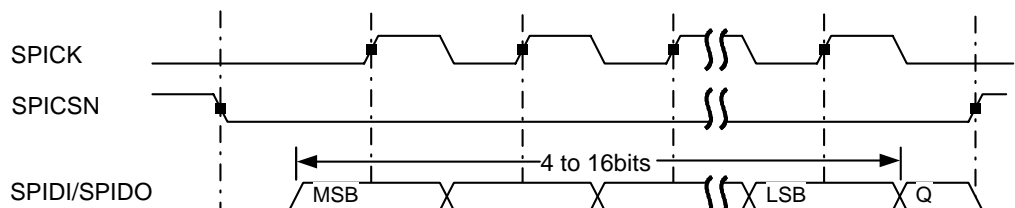
spo 表示 SPICLKOUT 极性，sph 表示 SPICLKOUT 相位。

图 11-11 ~ 图 11-22 中，以下缩略语或字母意义不变：

- MSB: Most Significant Bit
- LSB: Least Significant Bit
- Q: An undefined signal

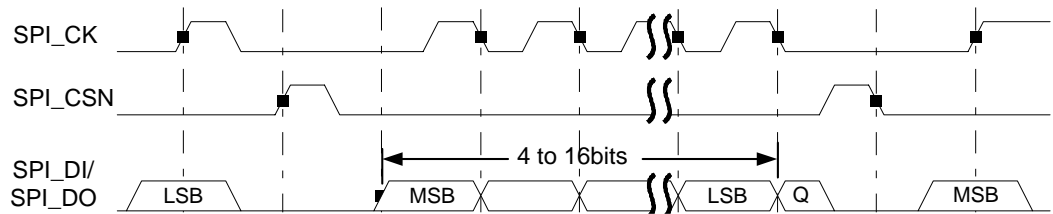
spo=0、sph=0 时 Motorola SPI 单帧格式如图 11-11 所示。

图11-11 Motorola SPI 单帧格式 (spo=0、sph=0)



spo=0、sph=0 时 Motorola SPI 连续帧格式如图 11-12 所示。

图11-12 Motorola SPI 连续帧格式 (spo=0、sph=0)



在该模式下，当 SPI 处于空闲状态时：

- SPICK 信号设置为低电平。
- SPICSN 信号设置为高电平。
- 发送数据线 SPIDO 强制为低电平。

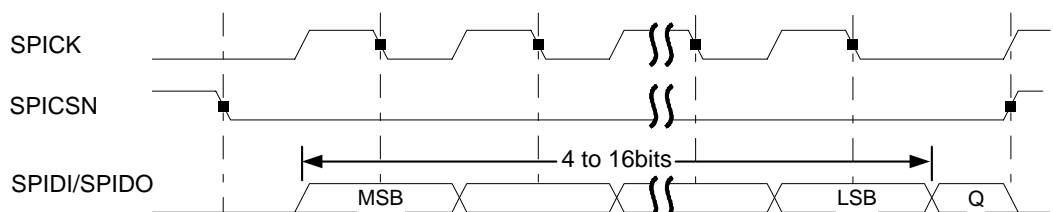
当 SPI 处于使能状态，而且发送 FIFO 内部存在有效数据时，设置 SPICSN 信号为低表示开始传输数据。这样就会使能 slave 的数据放在 master 的输入 SPIDI 线上。半个 SPICK 时钟周期之后，有效的 master 数据传输到 SPIDO 管脚。此时 master 和 slave 数据都已经有效，SPICK 主时钟管脚会在接下来的半个 SPICK 时钟周期之后变化为高电平。数据就会在 SPICK 时钟的上升沿被捕获，在时钟的下降沿被传送。

如果传输单个 word，当捕捉到传输的最后 1 个 bit，SPICSN 会在接下来的 1 个 SPICK 时钟之后恢复为高电平。

如果是连续的传输，SPICSN 信号在 2 个 word 传输之间必须将 SPICK 时钟拉高一个时钟周期。这是因为 sph 为 0 时，salve 选择管脚会固定其内部串行设备寄存器的数据，使它不会变化。因此在连续传输时，主设备必须在每 2 个 word 传输之间将 SPICSN 信号拉高。连续传输结束时，SPICSN 会在捕捉到最后 1 个 bit 之后的 1 个 SPICK 时钟周期之后恢复为高电平。

spo=0、sph=1 时 Motorola SPI 单帧帧格式如图 11-13 所示。

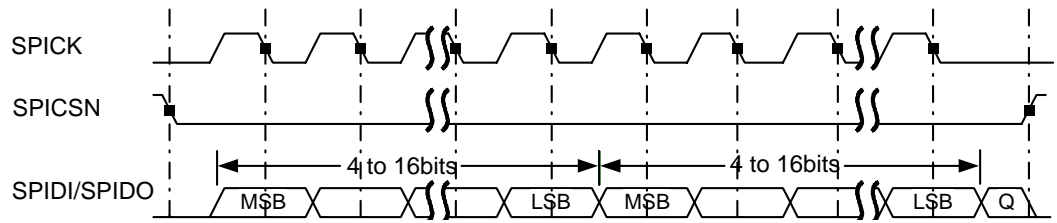
图11-13 Motorola SPI 单帧帧格式 (spo=0、sph=1)



spo=0、sph=1 时 Motorola SPI 连续帧帧格式如图 11-14 所示。



图11-14 Motorola SPI 连续帧格式 (spo=0、sph=1)



在该模式下，当 SPI 处于空闲状态时：

- SPICK 信号设置为低
- SPICSN 设置为高
- 发送数据线 SPIDO 强制为低

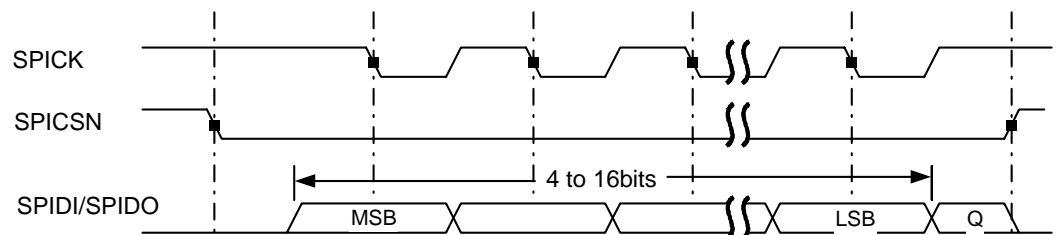
当 SPI 为使能状态，而且在发送 FIFO 内部有有效数据时，设置 SPICSN 信号为低表示开始传输数据。半个 SPICK 时钟周期之后，master 和 slave 的有效数据就分别在各自的传输线上有效。同时，SPICK 从一个上升沿开始有效。数据就会在 SPICK 时钟的上升沿被捕获，在时钟的下降沿被传送。

如果传输单个 word，当捕捉到传输的最后 1 个 bit，SPICSN 会在接下来的 1 个 SPICK 时钟之后恢复为高电平。

当连续传输时，在传输数据 word 之间 SPICSN 保持为低。连续传输结束时，SPICSN 会在最后 1 个 bit 捕获之后的 1 个 SPICK 时钟之后恢复为高电平。

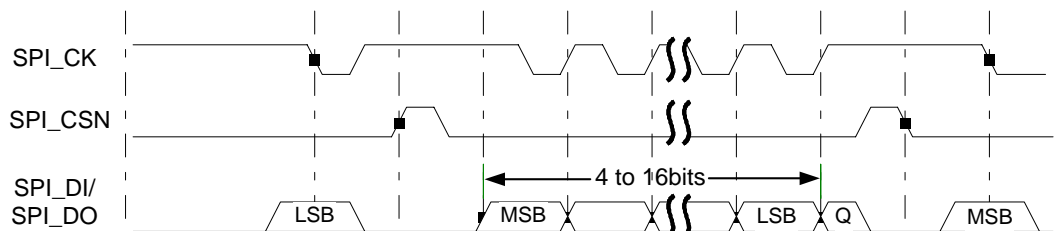
spo=1、sph=0 时 Motorola SPI 单帧格式如图 11-15 所示。

图11-15 Motorola SPI 单帧格式 (spo=1、sph=0)



spo=1、sph=0 时 Motorola SPI 连续帧格式如图 11-16 所示。

图11-16 Motorola SPI 连续帧格式 (spo=1、sph=0)



在该配置下，当 SPI 处于空闲状态时：

- SPICK 信号设置为高电平。
- SPICSN 信号设置为高电平。
- 发送数据线 SPIDO 强制为低电平。

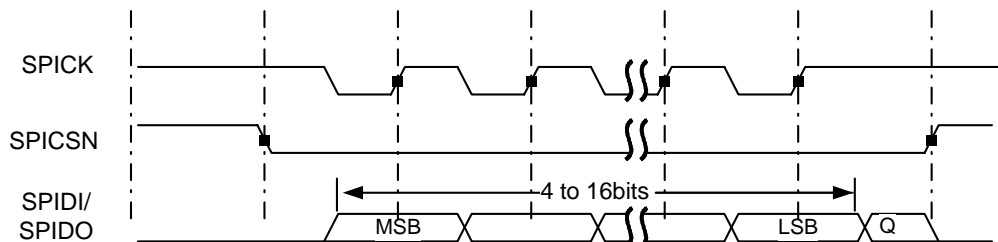
当 SPI 为使能状态，而且在发送 FIFO 内部有有效数据时，设置 SPICSN 信号为低表示开始传输数据，此时 slave 的数据就会立刻发送到 master 的接收数据线 SPIDI 上。半个 SPICK 周期之后，master 的有效数据就传送到 SPIDO 线上。再过半个 SPI_CK 时钟周期之后，SPICK master pin 设置为低。这意味着数据是在 SPICK 时钟的下降沿被捕获，在 SPICK 时钟的上升沿被传送。

如果传输单个 word，当捕捉到传输的最后 1 个 bit，SPICSN 会在接下来的 1 个 SPICK 时钟之后恢复为高电平。

如果是连续的传输，SPICSN 信号在 2 个 word 传输之间必须拉高。这是因为，当 sph 为 0 时，salve 选择管脚固定其内部串行设备寄存器的数据，使它不会变化。SPICSN 就会在捕获到最后 1 个 bit 之后的 1 个 SPICK 时钟周期之后恢复为高电平。

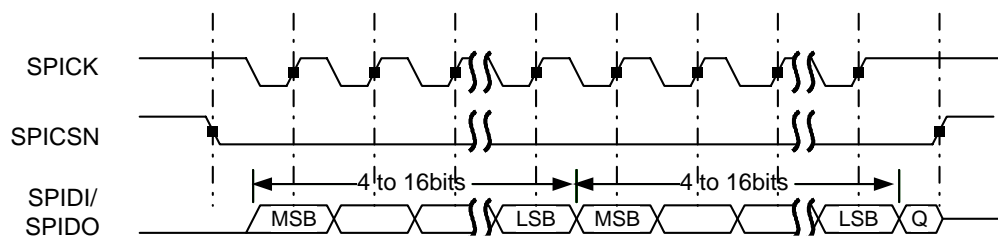
spo=1、sph=1 时 Motorola SPI 单帧格式如图 11-17 所示。

图11-17 Motorola SPI 单帧格式 (spo=1、sph=1)



spo=1、sph=1 时 Motorola SPI 连续帧格式如图 11-18 所示。

图11-18 Motorola SPI 连续帧格式 (spo=1、sph=1)



在该模式下，当 SPI 处于空闲状态时：

- SPICK 信号设置为高电平。
- SPICSN 信号设置为高电平。
- 发送数据线 SPIDO 强制为低电平。



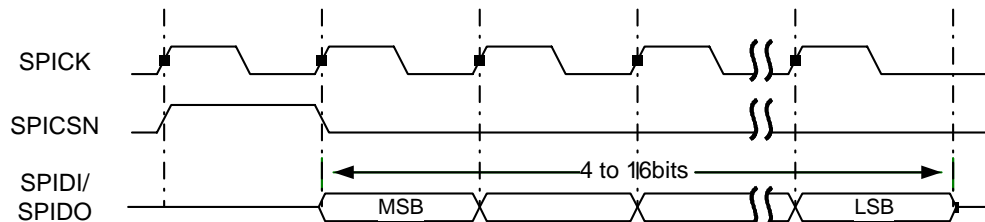
当 SPI 为使能状态，而且在发送 FIFO 内部有有效数据时，设置 SPICSN master 信号为低表示开始传输数据。半个 SPICK 时钟之后，master 和 slave 数据都会在各自己的传输线上有效。同时，时钟 SPICK 从一个下降沿开始有效。数据是在 SPICK 时钟的下降沿被捕获，在时钟的上升沿被传送。

当传输单个 word 时，SPICSN 会在传输的最后 1 个 bit 捕获之后的一个 SPICK 时钟周期之后恢复为高电平。

如果是连续传输，SPICSN 信号始终保持为低，直到传输结束时，SPICSN 就会在捕获到最后一 bit 之后的一个 SPICK 时钟周期之后恢复到空闲为高的状态。对于连续的传输来说，信号 SPICSN 在传输的过程中一直保持为低，结束的方式与单个传输的方式相同。

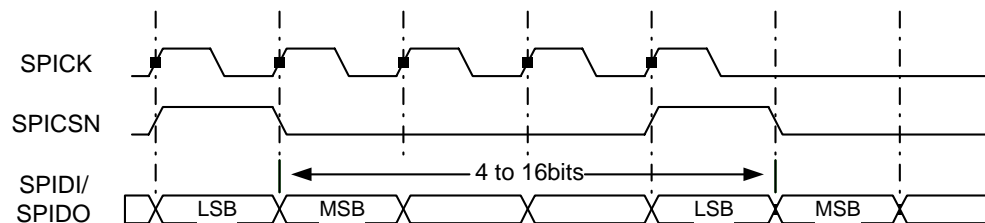
TI 同步串行单帧格式如图 11-19 所示。

图11-19 TI 同步串行单帧格式



TI 同步串行连续帧格式如图 11-20 所示。

图11-20 TI 同步串行连续帧格式



在该模式下，当 SPI 处于空闲状态时：

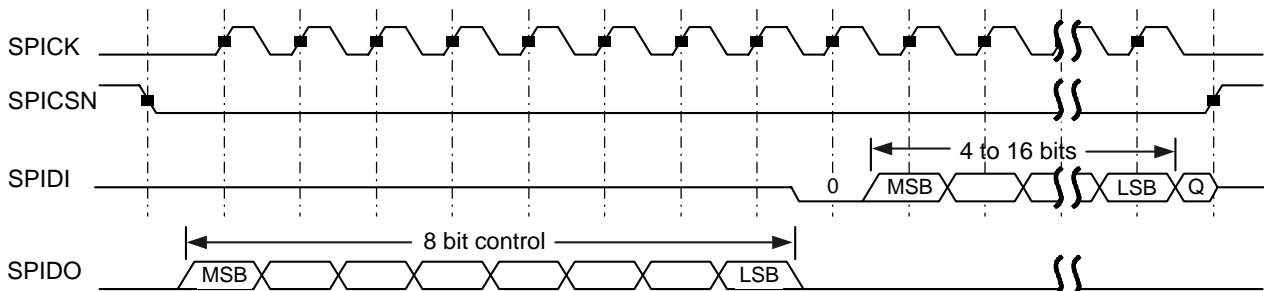
- SPICK 为低电平。
- SPICSN 为低电平。
- 传输数据线 SPIDO 保持为高阻。

一旦发送 FIFO 有数据，SPICSN 就会产生一个 SPICK 时钟周期的高电平脉冲，将被发送的数据就会从发送 FIFO 传送到发送逻辑串行移位寄存器。在 SPICK 时钟的下一个上升沿，4bit~16bit 数据帧的 MSB 就会从 SPIDO 移位输出。同样，从外部串行 slave 设备接收数据的 MSB 会从 SPIDI 管脚移位输入。

SPI 和片外串行设备在 SPICK 时钟的下降沿将数据存入串行移位寄存器。接收串行寄存器在接收到 LSB 之后的第一个 SPICK 时钟上升沿将数据送给接收 FIFO。

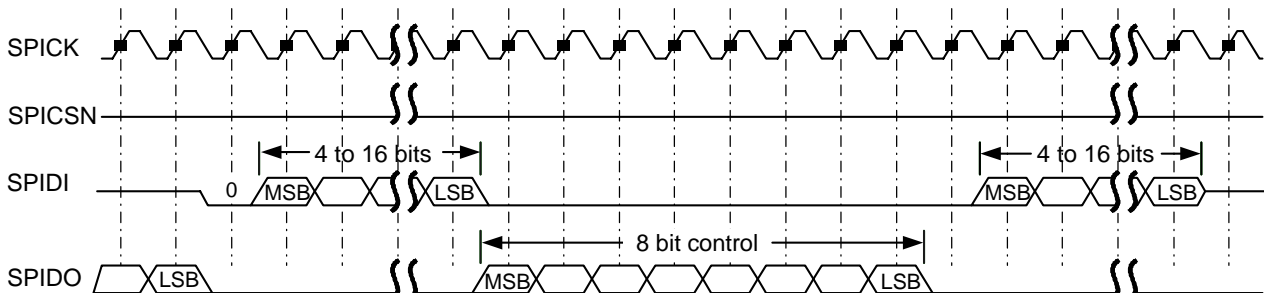
National Semiconductor Microwire 单帧格式如图 11-21 所示。

图11-21 National Semiconductor Microwire 单帧格式



National Semiconductor Microwire 连续帧格式如图 11-22 所示。

图11-22 National Semiconductor Microwire 连续帧格式



Microwire 的格式与 SPI 的格式非常相近，使用 master-slave 信息的传输技术，只不过 SPI 是全双工通信，而 Microwire 半双工通信。在 SPI 向外部芯片发送串行数据的时候，都要先加 8bit 控制字。在这个过程中，SPI 没有接收到任何数据。传输完毕之后，片外芯片对接收到的数据进行解码，在与 8bit 控制信息间隔一个时钟周期之后，slave 开始响应所需求的数据。返回的数据长度为 4bit~16bit，使得整个帧的长度为 13bit~25bit。

在该模式下，当 SPI 处于空闲状态时：

- SPICK 信号设置为低电平。
- SPICSN 设置为高电平。
- 发送数据线 SPIDO 强制为低电平。

向发送 FIFO 内部写进一个控制字节开始一次传送。SPICSN 的下降沿引发数据的传输，发送 FIFO 的数据被发送到串行移位寄存器，8bit 控制帧的 MSB 被发送到发送管脚 SPIDO。在帧的传送过程中，SPICSN 保持为低。SPIDI 在这个传送过程中保持为高阻。

片外的串行从设备在 SPICK 时钟的每一个上升沿将数据锁存到串行移位寄存器中。当从设备锁存完最后 1 个 bit 的数据之后，在接下来的 1 个时钟周期的等待时间里，对接收到的数据开始解码，然后从设备反馈给 SPI 所要求的数据。每 1 个 bit 都是在 SPICK 时钟的下降沿写到 SPIDI 的。对单个数据传送来说，在帧的结尾，SPICSN 在最后 1 个



bit 写到接收串行寄存器之后的 1 个时钟周期后拉高，这样就使接收到的数据传送到接收 FIFO。

对于连续的传送来说，数据传送的开始和结束都和单个数据的传送方式相同。在这个传送过程中，信号 SPICSN 时一直保持为低的，传送的数据也是连续的。下一帧的控制字直接和上一帧的 LSB 相邻。当帧的 LSB 锁存到 SPI 之后，接收到的每一个数值都是在 SPICK 时钟的下降沿取自接收移位寄存器。

11.3.5 工作方式

管脚复用配置

SPI 管脚与 VO 复用，使用 SPI 前，需要配置系统控制器使能相应管脚的 SPI 功能，请参见管脚复用模块 IO Config 寄存器 reg54~reg58 的配置说明。

时钟门控

在软件完成当前数据传输且未启动新的数据传输的情况下可关断 SPI 时钟，但需要确保硬件已处于空闲状态（即 SPI_SR[bsy]为 0 时）。

步骤如下：

步骤 1 读 SPI_SR。

步骤 2 若 SPI_SR[bsy]为 0，则写 SPI_CR1[sse]=0，进入步骤 3；若 SPI_SR[bsy]为 1，则延时等待，返回步骤 1。

步骤 3 写 SC_PERDIS[spi_clkdis]，关闭 SPI 工作时钟。

----结束

时钟配置

- SPI 的工作时钟的最小频率由此公式确定：
SPI 的工作时钟频率(min) $\geq 2 \times F_{SPICK}(\max)$
当 SPI 工作时钟确定，在 2 分频时 SPI 的输出时钟频率最大。
- SPI 工作时钟的最大频率由此公式确定：
SPI 的工作时钟频率(max) $\leq 254 \times 256 \times F_{SPICK}(\min)$
当 SPI 工作时钟确定，在 254×256 分频时，SPI 输出时钟频率最小。
- 输出的时钟频率 SPICK 为：
 $F_{SPICK} = \text{SPI 的工作时钟频率} / (\text{CPSDVSR} \times (1 + \text{SCR}))$

配置不同的分频数 CPSDVSR 和 SCR，可以得到不同的 SPI 输出时钟频率，请参见 SPI_CR0、SPI_CPSR 寄存器配置描述。

表 11-13 给出了 SPI 时钟分频的典型配置值。



表11-13 时钟分频的典型配置

SPI 工作时钟频率(MHz)	CPSDVSR	SCR	SPICK(MHz)
100	2	1	25
100	2	4	10
24	2	1	6
24	2	4	2.4

软复位

对接器件容许的情况下，可以任意时间对 SPI 进行软复位，不会对系统造成影响。通过配置系统控制器 SC_PERCTRL10[spi_srst]为 1，可实现对 SPI 的单独软复位。复位后各配置寄存器的值均复位为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

中断或查询方式下的数据传输

初始化步骤如下：

- 步骤 1 向 SPI_CR1[sse]写 0，使 SPI 处于禁止状态。
- 步骤 2 写相应的配置值到 SPI_CR0，配置帧格式及传输数据位宽等参数。
- 步骤 3 配置 SPI_CPSR 寄存器，设定需要的时钟分频因子。
- 步骤 4 如果驱动程序采用中断方式则需设定 SPI_INTMASK，使能相应中断信号；采用查询方式时应禁止产生相应中断信号。
- 步骤 5 向 SPI_CR1[sse]写 1，使能 SPI，完成初始化配置。

----结束

查询发送定长（以 4 个为例）数据的步骤如下：

- 步骤 1 配置 SPI_CPSR 寄存器，设置预分频因子 cpsdvsr。
- 步骤 2 配置 SPI_CR0，设置串行时钟率、帧格式（SPI、TI、MW）和帧长 dss。
- 步骤 3 向 SPI_CR1[see]写 1，使能 SPI。
- 步骤 4 查询状态寄存器 SPI_SR，判断 FIFO 空满状态。如果 SPI_SR[tnf]为 1，说明发送 FIFO 不满，则写 SPI_DR 寄存器，共写 4 个数据。
- 步骤 5 查询状态寄存器 SPI_SR，判断 FIFO 空满状态，直至完全发送。
- 步骤 6 向 SPI_CR1[see]写 0，禁止 SPI。

----结束



查询接收定长（以 4 个为例）数据的步骤如下：

- 步骤 1 配置 `SPI_CPSR`[cpsdvsr]，设置预分频因子。
- 步骤 2 配置 `SPI_CR0` 寄存器，设置串行时钟率、帧格式（SPI、TI、MW）和帧长。
- 步骤 3 向 `SPI_CR1`[see]写 1，使能 SPI。
- 步骤 4 查询状态寄存器 `SPI_SR`，判断接收 FIFO 状态；如果接收 FIFO 不空，则读 `SPI_DR` 寄存器，接收一个 half word，共读 4 个数据。
- 步骤 5 查询状态寄存器 `SPI_SR`，直至不忙。
- 步骤 6 向 `SPI_CR1`[see]写 0，禁止 SPI。

----结束

DMA 方式下的数据传输

初始化步骤如下：

- 步骤 1 向 `SPI_CR1`[sse]写 0，使 SPI 处于禁止状态。
- 步骤 2 写相应的配置值到 `SPI_CR0`，配置帧格式及传输数据位宽等参数。
- 步骤 3 配置 `SPI_CPSR` 寄存器，设定需要的时钟分频因子。
- 步骤 4 配置 `SPI_INTMASK` 寄存器，禁止产生相应中断信号。
- 步骤 5 配置 `SPI_DMOCR` 寄存器，使能 SPI 的 DMA 功能。

----结束

数据发送步骤如下：

- 步骤 1 获取一个 DMAC 的通道。
- 步骤 2 配置这个 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
- 步骤 3 启动 DMAC，响应 SPI 发送 FIFO 的 DMA 请求进行数据传输。
- 步骤 4 向 `SPI_CR1`[sse]写 1，使能 SPI，完成初始化配置。
- 步骤 5 如果配置的 DMAC 传输的数据长度为奇数个，DMAC 会先使用数据长度除以步骤 2 中配置的 burst 长度，确定需要响应的 burst 个数。这样虽然 SPI 会在所有发出 DMA burst 请求的同时产生 single 请求，但是 DMAC 只会先响应 burst 请求，直到最后不够一个 burst 长度时，才响应 SPI 的 single 请求完成全部的数据传输。
- 步骤 6 通过 DMA 中断上报，判断数据是否发送完成，如果完成则关闭 SPI 的 DMA 发送功能。

----结束

数据接收步骤如下：

- 步骤 1 获取一个 DMAC 通道。



- 步骤 2 配置该 DMAC 通道的配置寄存器和控制寄存器中的相关参数。
- 步骤 3 启动 DMAC，响应 SPI 接收 FIFO 的 DMA 请求进行数据传输。
- 步骤 4 向 `SPI_CR1[sse]`写 1，使能 SPI，完成初始化配置。
- 步骤 5 如果配置的 DMAC 传输的数据长度为奇数个，DMAC 会先使用数据长度除以步骤 2 中配置的 burst 长度，确定需要响应的 burst 个数。这样虽然 SPI 会在所有发出 DMA burst 请求的同时产生 single 请求，但是 DMAC 只会先响应 burst 请求，直到最后不够一个 burst 长度时响应 SPI 的 single 请求完成全部的数据传输。
- 步骤 6 通过 DMA 中断上报，判断数据是否接收完成，如果完成则关闭 SPI 的 DMA 接收功能。
- 结束

11.3.6 寄存器概览

SPI 寄存器概览如表 11-14 所示。

表11-14 SPI 寄存器概览（基地址是 0x200E_0000）

偏移地址	名称	描述	页码
0x000	SPI_CR0	控制寄存器 0	11-63
0x004	SPI_CR1	控制寄存器 1	11-65
0x008	SPI_DR	接收 FIFO 和发送 FIFO 的数据寄存器	11-66
0x00C	SPI_SR	状态寄存器	11-66
0x010	SPI_CPSR	时钟预分频寄存器	11-67
0x014	SPI_INTMASK	中断屏蔽设置或清除寄存器	11-68
0x018	SPI_RINTSTATUS	原始中断状态寄存器	11-69
0x01C	SPI_MINTSTATUS	中断屏蔽状态寄存器	11-70
0x020	SPI_INTCLR	中断清除寄存器	11-70
0x024	SPI_DMACR	DMA 控制寄存器	11-71

11.3.7 寄存器描述

SPI_CR0

SPI_CR0 为控制寄存器 0，用来控制 SPI 的各种功能。



	Offset Address				Register Name				Total Reset Value							
	0x000				SPI_CR0				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	scr							sph	spo	fff	dss					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description												
	[15:8]	RW	scr	串行时钟率。SCR (System Clock Reference) 的值是用来产生 SPI 发送和接收的比特率的。比特率由此公式计算： $F_{SSPCLK} = \frac{CPSDVSR \times (1 + SCR)}{2}$ 如果 SPICPSR[CPSDVSR] 的值为一个 2~254 之间的偶数时，SCR 为一个 0~255 之间的值。												
	[7]	RW	sph	SPICK 相位控制（只对 Motorola SPI 帧格式适用）。 0: 在传输的第 1 个时钟沿捕获数据； 1: 在传输的第 2 个时钟沿捕获数据。												
	[6]	RW	spo	SPICK 电平（只对 Motorola SPI 帧格式适用）。 0: 在 SPICK 管脚产生一个稳定为低的值； 1: 如果没有数据传输，在 SPICK 管脚产生一个稳定为高的值。												



[5:4]	RW	frf	帧格式选择。 00: Motorola SPI 帧格式; 01: TI 同步串行帧格式; 10: National Microwire 帧格式; 11: 保留。
[3:0]	RW	dss	数据大小选择。 0000~0010: 保留。 0011: 4bit; 0100: 5bit; 0101: 6bit; 0110: 7bit; 0111: 8bit; 1000: 9bit; 1001: 10bit; 1010: 11bit; 1011: 12bit; 1100: 13bit; 1101: 14bit; 1110: 15bit; 1111: 16bit。

SPI_CR1

SPI_CR1 为控制寄存器 1，用来控制 SPI 的各种功能。

	Offset Address				Register Name				Total Reset Value							
	0x004				SPI_CR1				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												sod	ms	sse	lbm
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:4]	-	reserved	保留。													
[3]	RW	sod	slave 模式输出不使能。这一 bit 只与 slave 模式有关(ms=1)。在多 slave 的系统中，可能 SPI master 采用广播的方式发信息给所有的 slave，而确保只有一个 slave 将数据驱动到自己的串行输出线上，这样从 slave 输出到													



			<p>SPI_SPIDI 是连接在一起的。对于这样的系统操作，如果 slave 不准备驱动 SPI_SPIDO 的话，就设置 sod bit。</p> <p>0: SPI 可以在 slave 模式驱动 SPITXD 输出；</p> <p>1: SPI 不能在 slave 模式驱动 SPITXD 输出。</p>
[2]	RW	ms	<p>Master 或者 slave 模式选择，这一比特只能在 SPI 处于非使能状态的时候改变。</p> <p>0: 设备被配置成 master（默认）；</p> <p>1: 设备被配置成 slave。</p>
[1]	RW	sse	<p>同步串行接口使能。</p> <p>0: 禁止；</p> <p>1: 使能。</p>
[0]	RW	lbm	<p>环回模式。</p> <p>0: 正常的串行接口操作使能；</p> <p>1: 发送串行移位寄存器输出在内部连接在接收串行移位寄存器输入上。</p>

SPI_DR

SPI_DR 为数据寄存器，当该寄存器被读的时候就是 SPI 接收 FIFO 的出口。当该寄存器被写的时候就是 SPI 的发送 FIFO 的入口。在 Microwire 帧格式下，发送 FIFO 内数据位宽固定为 8bit，接收时无限制。当 SPI_CR1[sse]设置为 0 时，接收和发送 FIFO 并不被清 0，这样在我们启动 SPI 之前就可以将数据放在发送 FIFO 里。

	Offset Address					Register Name					Total Reset Value					
	0x008					SPI_DR					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description											
	[15:0]	RW	data		读时为接收 FIFO，写时为发送 FIFO。											

SPI_SR

SPI_SR 为 FIFO 状态寄存器，是一个只读状态寄存器。



	Offset Address					Register Name					Total Reset Value					
	0x00C					SPI_SR					0x0003					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										bsy	rff	rne	tnf	tfe	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bits	Access	Name	Description													
[15:5]	-	reserved	保留。													
[4]	RO	bsy	SPI 繁忙标记。 0: 空闲状态; 1: 繁忙状态。													
[3]	RO	rff	接收 FIFO 满状态。 0: 未滿; 1: 滿。													
[2]	RO	rne	接收 FIFO 空状态。 0: 空; 1: 非空。													
[1]	RO	tnf	发送 FIFO 不滿意状态。 0: 滿; 1: 未滿。													
[0]	RO	tfe	发送 FIFO 空状态。 0: 非空; 1: 空。													

SPI_CPSR

SPI_CPSR 为时钟分频系数寄存器，它指定了一个分频因子，对输入的 SPI 工作时钟进行分频，产生 SPICK。这个因子必须是 2~254 之间的偶数，最低的 1bit 位必须是零。如果向这个寄存器写一个奇数，那么读回来的数据的最低的 1bit 位则必然为零。



Offset Address		Register Name		Total Reset Value												
0x010		SPI_CPSR		0x0000												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved							cpsdvsr								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:8]	-	reserved	保留。													
[7:0]	RW	cpsdvsr	时钟预分频系数。该数值必须是 2~254 之间的偶数，与 SPI_CRO[scr]一起对输入时钟 SPICLK 的频率进行分频。CPSDVSr 最低位读作 0。													

SPI_INTMASK

SPI_INTMASK 为中断屏蔽寄存器，写相应的比特位用于屏蔽或清除一个中断。

Offset Address		Register Name		Total Reset Value												
0x014		SPI_INTMASK		0x0000												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											txim	rxim	rtim	rorim	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:4]	-	reserved	保留。													
[3]	RW	txim	发送 FIFO 半空或更少时产生的中断。 0: 屏蔽该中断; 1: 不被屏蔽该中断。													
[2]	RW	rxim	接收 FIFO 半空或更少时产生的中断。 0: 屏蔽该中断; 1: 不屏蔽该中断。													
[1]	RW	rtim	接收 FIFO 不空并且在超时周期之前不读接收 FIFO 产生的超时中断。 0: 屏蔽该中断; 1: 不被屏蔽该中断;													



[0]	RW	rorm	接收 FIFO 在满情况下被写产生的溢出中断。 0: 屏蔽该中断; 1: 不被屏蔽该中断。
-----	----	------	---

SPI_RINTSTATUS

SPI_RINTSTATUS 为原始中断状态寄存器，读该寄存器时，得到没有屏蔽的当前原始中断状态。写该寄存器时无效。

	Offset Address				Register Name				Total Reset Value							
	0x018				SPI_RINTSTATUS				0x0008							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txris	rxris	ptris	rorris
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Bits	Access	Name	Description													
[15:4]	-	reserved	保留。													
[3]	RO	txris	给出 SPITXINTR 中断的原始（屏蔽之前的）中断状态。 0: 中断无效; 1: 中断有效。													
[2]	RO	rxris	给出 SPIRXINTR 中断的原始（屏蔽之前的）中断状态。 0: 中断无效; 1: 中断有效。													
[1]	RO	ptris	给出 SPIRTXINTR 中断的原始（屏蔽之前的）中断状态。 0: 中断无效; 1: 中断有效。													
[0]	RO	rorris	给出 SPIRORXINTR 中断的原始（屏蔽之前的）中断状态。 0: 中断无效; 1: 中断有效。													



SPI_MINTSTATUS

SPI_MINTSTATUS 为屏蔽后的中断状态寄存器，是只读状态寄存器，写无效。

	Offset Address				Register Name				Total Reset Value							
	0x01C				SPI_MINTSTATUS				0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txmis	rxmis	rtmis	rormis
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description													
[15:4]	-	reserved	保留。													
[3]	RO	txmis	给出发送 FIFO 屏蔽中断状态（屏蔽之后）SPITXINTR 中断。 0：中断无效； 1：中断有效。													
[2]	RO	rxmis	给出接收 FIFO 屏蔽中断状态（屏蔽之后）SPIRXINTR 中断。 0：中断无效； 1：中断有效。													
[1]	RO	rtmis	给出接收超时屏蔽中断状态（屏蔽之后）SPIRTINTR 中断。 0：中断无效； 1：中断有效。													
[0]	RO	rormis	给出接收溢出屏蔽中断状态（屏蔽之后）SPIRORINTR 中断。 0：中断无效； 1：中断有效。													

SPI_INTCLR

SPI_INTCLR 为中断清除寄存器，写 1 时清除相应的中断，写 0 无效。



Offset Address		Register Name	Total Reset Value												
0x020		SPI_INTCLR	0x0000												
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
Name	reserved											rtic	roric		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0														
Bits	Access	Name	Description												
[15:2]	-	reserved	保留。												
[1]	WO	rtic	SPIRTINTR 中断清除。 0: 不清除该中断; 1: 清除该中断。												
[0]	WO	roric	SPIRORINTR 中断清除。 0: 不清除该中断; 1: 清除该中断。												

SPI_DMCCR

SPI_DMCCR 为 DMA 功能使能寄存器，用于 SPI 中 DMA 请求功能的使能。

Offset Address		Register Name	Total Reset Value												
0x024		SPI_DMCCR	0x0000												
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
Name	reserved											txdmae	rxdmae		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0														
Bits	Access	Name	Description												
[15:2]	-	reserved	保留。												
[1]	RW	txdmae	发送 FIFO 的 DMA 操作使能。 0: 禁止; 1: 使能。												
[0]	RW	rxdmae	接收 FIFO 的 DMA 操作使能。 0: 禁止; 1: 使能。												



11.4 IR

11.4.1 概述

红外遥控接收单元 IR (Infrared Remoter) 通过红外接口接收红外数据，并支持 NEC with simple repeat code、NEC with full repeat code、SONY 和 TC9012 四种数据格式解码，及接收数据错误检测和红外遥控唤醒等功能。

11.4.2 特点

IR 模块具有以下特点：

- 软件可配置关闭红外遥控接收模块。
- 支持接收数据溢出中断、接收数据帧格式错误中断、接收数据帧中断、按键释放的中断、各种中断构成的组合中断。
- 支持初始中断状态查询和屏蔽后中断状态查询。
- 支持中断清除和屏蔽（写清）。
- 支持红外遥控唤醒。
- 支持参考时钟频率 1MHz~128MHz 可选，软件可编程控制分频因子使工作时钟预分频到 1MHz。

11.4.3 信号描述

IR 接口信号如表 11-15 所示。

表11-15 IR 接口信号表

信号名称	方向	描述	对应管脚
IR_RCV	I	从管脚接收到的串行红外数据。 ^a	IRRCV

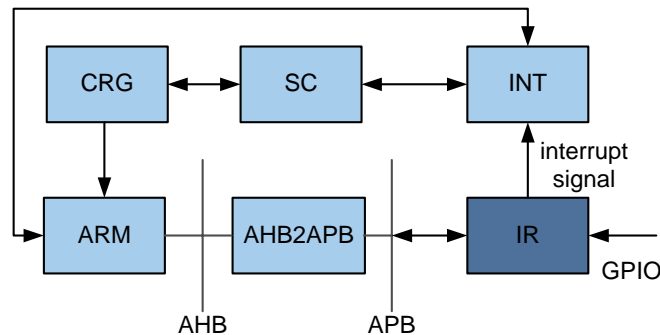
a: 作为红外接收功能时，IRRCV 管脚在单板上要求上拉。

11.4.4 功能描述

当 IR 模块接收到红外遥控器发射的红外信号时，便对其进行解码，然后传送给 ARM 系统。ARM 系统再根据接收到的码的不同进行相应的操作，实现期望的功能。IR 模块连接在 ARM 子系统内的 APB 总线上，当芯片处于低功耗状态时（CPU 停止工作），IR 模块会在接收一个完整的帧数据后，产生中断信号送给 INT 模块，再由 INT 送出控制信号控制系统控制器，进而控制 CRG 唤醒 ARM 子系统，实现红外遥控唤醒功能。

IR 模块功能框图如图 11-23 所示。

图11-23 IR 模块功能框图



通过对多种红外遥控器发出的信号进行分析，发现在不同的遥控器发出的红外指令中，引导码各不相同，而且后面的控制指令也有较大差别，甚至指令码的位数也不相同，这是因为这些红外遥控器的设计没有遵循统一的红外遥控标准。尽管遵循的标准不同，但是基本的编码思想是相同的，都是采用不同的周期和不同占空比的脉冲分别表示 0 和 1。不同遥控器占空比可能不同，且脉冲周期也不相同。根据这些不同，对一些码型类似的红外数据进行分类：NEC with simple repeat code 的数据格式、NEC with full repeat code 的数据格式、TC9012 的数据格式和 SONY 的数据格式。

红外接收数据码型统计情况如表 11-16~表 11-18 所示。

表11-16 红外接收数据码型的统计表（NEC with simple repeat code）

数据格式		NEC with simple repeat code			
		uPD6121G	D6121/BU5777/D1913	LC7461M-C13	AEHA
引导码 (10μs)	LEAD_S	900	900	900	337.6
	LEAD_E	450	450	450	168.8
bit0 (10μs)	B0_L	56	56	56	42.2
	B0_H	56	56	56	42.2
bit1 (10μs)	B1_L	56	56	56	42.2
	B1_H	169	169	169	126.6
simple repeat code (10μs)	SLEAD_S	900	900	900	337.6
	SLEAD_E	225	225	225	337.6
burst (10μs)		55	55	55	42.2
帧长 (10μs)		10800	10800	10800	8777.6~12828.8
有效数据位		32	32	42	48



表11-17 红外接收数据码型的统计表（NEC with full repeat code）

数据格式		NEC with full repeat code						
		uPD6121G	LC7461 M-C13	MN602 4-C5D6	MN6014 -C6D6	MATNEW	MN6030	PANA SONIC
引导码 (10μs)	LEAD_S	900	900	337.6	349.2	348.8	349	352
	LEAD_E	450	450	337.6	349.2	374.4	349	352
bit0 (10μs)	B0_L	56	56	84.4	87.3	43.6	87.3	88
	B0_H	56	56	84.4	87.3	43.6	87.3	88
bit1 (10μs)	B1_L	56	56	84.4	87.3	43.6	87.3	88
	B1_H	169	169	253.2	174.6	130.8	261.9	264
simple repeat code (10μs)	SLEAD_S	无	无	无	无	无	无	无
	SLEAD_E							
burst (10μs)		55	55	84.4	87.3	43.6	87.3	88
帧长 (10μs)		10800	10800	10130	10470	12413.6~ 16594.4	10500	10400
有效数据位		32	42	22	24	48	22	22

表11-18 红外接收数据码型的统计表（TC9012 和 SONY 码）

数据格式		TC9012	SONY			
		TC9012F/9243	SONY-D7C5	SONY-D7C6	SONY-D7C8	SONY-D7C13
引导码 (10μs)	LEAD_S	450	240	240	240	240
	LEAD_E	450	60	60	60	60
bit0 (10μs)	B0_L	56	60	60	60	60
	B0_H	56	60	60	60	60
bit1 (10μs)	B1_L	56	120	120	120	120
	B1_H	169	60	60	60	60
simple repeat code (10μs)	SLEAD_S	无	无	无	无	无
	SLEAD_E					
burst (10μs)		56	无	无	无	无



数据格式	TC9012	SONY			
	TC9012F/9243	SONY-D7C5	SONY-D7C6	SONY-D7C8	SONY-D7C13
帧长 (10μs)	10800	4500	4500	4500	4500
有效数据位	32	12	13	15	20

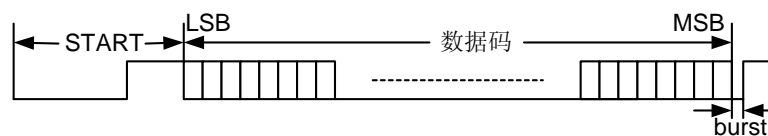
11.4.4.2 NEC with simple repeat code 数据格式

帧格式

NEC with simple repeat code 数据格式是由 START（引导码）、数据码和 burst 三部分组成，其中 START 是由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，它是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。

发送单个 NEC with simple repeat code 的帧格式如图 11-24 所示。

图11-24 发送单个 NEC with simple repeat code 码的帧格式



如果按键时间持续超过一帧的时间，则在收到完整数据帧后，接下来收到的数据帧仅由简化的引导码和 burst 信号组成。引导码也是由起始码（低电平）和结束码（高电平）组成，持续按键连续发送 NEC with simple repeat code 码的帧格式如图 11-25 所示。

图11-25 持续按键连续发送 NEC with simple repeat code 码的帧格式

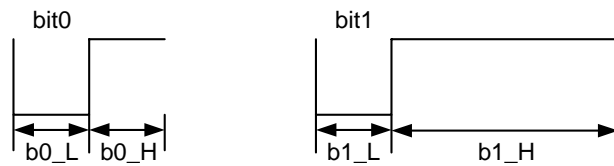


码格式

NEC with simple repeat code 的 bit0 或 bit1 定义如图 11-26 所示。



图11-26 NEC with simple repeat code 码 bit0 和 bit1 定义



NEC simple repeat code 单发代码格式和连发代码格式分别如图 11-27 和图 11-28 所示。

图11-27 NEC with simple repeat code 码单发代码格式

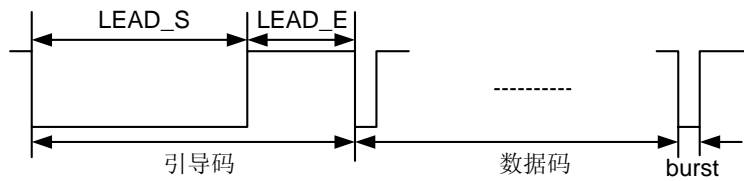
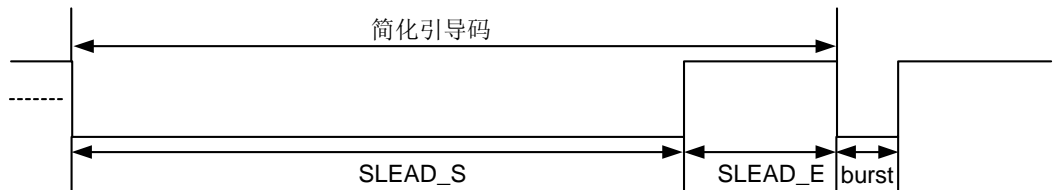


图11-28 NEC with simple repeat code 码连发代码格式



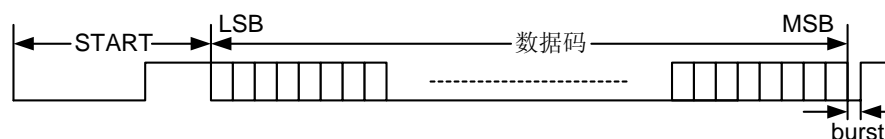
注 1：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定，请参见表 11-16~表 11-18。
注 2：帧长不能大于 160ms，否则无法识别简化引导码。

11.4.4.3 NEC with full repeat code 数据格式

帧格式

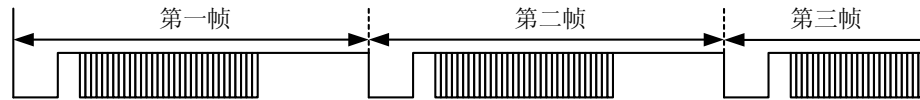
NEC with full repeat code 的数据格式是由 START（引导码）、数据码和 burst 三部分组成。START 是由一个起始码（低电平）和一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，它是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。发送单个 NEC with full repeat code 帧格式如图 11-29 所示。

图11-29 发送单个 NEC with full repeat code 码的帧格式



如果按键时间持续超过一帧的时间，则在收到完整数据帧（第一帧）后，接下来收到的数据帧还是一个完整的数据帧格式（即按照帧间隔重复发送第一帧数据），持续按键连续发送 NEC with full repeat code 码的帧格式如图 11-30 所示。

图11-30 持续按键连续发送 NEC with full repeat code 码的帧格式

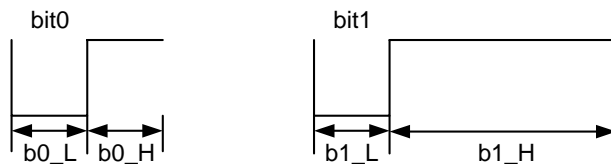


通过图 11-28 和图 11-30 可以看出：NEC with simple repeat code 与 NEC with full repeat code 唯一不同之处就是重复帧的格式，NEC with simple repeat code 发送的是简化的引导码，而 NEC with full repeat code 发送的是完整帧格式，第一帧和重复帧完全相同。

码格式

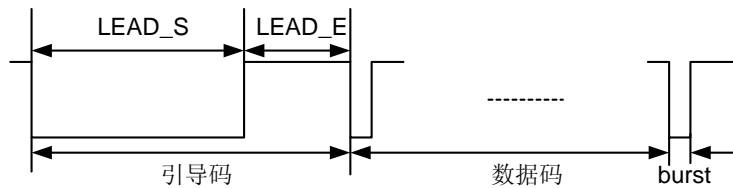
NEC with full repeat code 码 bit0 或 bit1 定义如图 11-31 所示。

图11-31 NEC with full repeat code 码 bit0 和 bit1 定义



NEC with full repeat code 码单发代码格式如图 11-32 所示。

图11-32 NEC with full repeat code 码单发代码格式



注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定，请参见表 11-16~表 11-18。

11.4.4.4 TC9012 数据格式

帧格式

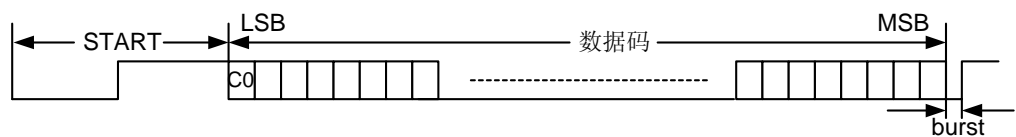


注意

根据 TC9012 码的数据格式特点，所有按键编码的第一位都必须全是 1 或者全是 0，否则会产生不需要的持续按键帧。

TC9012 的数据格式是由 START（引导码）、数据码和 burst 三部分组成，其中 START 是由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，其是按照 LSB first 的顺序接收的；burst 信号用于接收最后一个数据码位。发送单个 TC9012 码的帧格式如图 11-33 所示。

图11-33 发送单个 TC9012 码的帧格式



如果按键时间持续超过一帧的时间，则在收到完整数据帧后，接下来收到的数据帧由引导码、一个数据位和 burst 信号三部分组成。引导码也是由起始码（低电平）和结束码（高电平）组成；该数据位是上一帧接收的第一个数据位（C0）的反码。发送连续 TC9012 码的帧格式如图 11-34 所示。

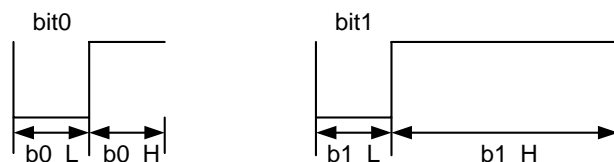
图11-34 持续按键连续发送 TC9012 码的帧格式



码格式

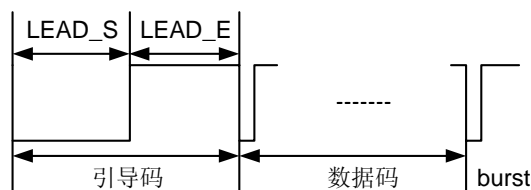
TC9012 码 bit0 或 bit1 定义如图 11-35 所示。

图11-35 TC9012 码 bit0 和 bit1 定义



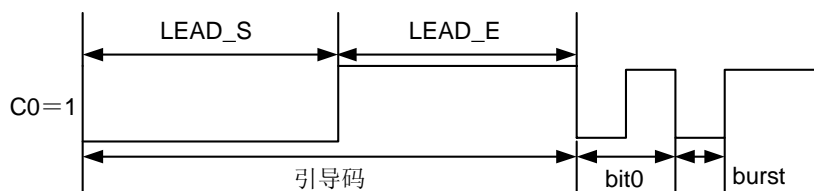
TC9012 码单发代码格式如图 11-36 所示。

图11-36 TC9012 码单发代码格式



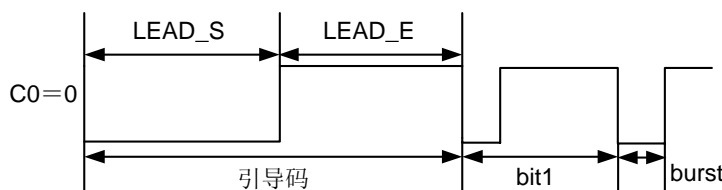
C0=1 时，TC9012 码连发代码格式如图 11-37 所示。

图11-37 TC9012 码连发代码格式 (C0=1)



C0=0 时，TC9012 码连发代码格式如图 11-38 所示。

图11-38 TC9012 码连发代码格式 (C0=0)



注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定。请参见表 11-16~表 11-18。另外值得注意的是帧长不能大于 160ms，否则无法识别重复帧。

11.4.4.5 SONY 的数据格式

帧格式

SONY 码数据格式是由 START（引导码）和数据码两部分组成。其中 START 由一个起始码（低电平），一个结束码（高电平）组成；数据码的有效位数以及某位表示的含义由具体的码型而定，其是按照 LSB first 的顺序接收的。发送单个 SONY 码帧格式如图 11-39 所示。

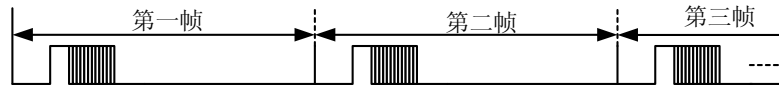
图11-39 发送单个 SONY 帧格式





如果按键时间持续超过一帧的时间，则再收到完整数据帧后，接下来收到的数据帧还是一个完整的数据帧格式。持续按键连续发送 SONY 码帧格式如图 11-40 所示。

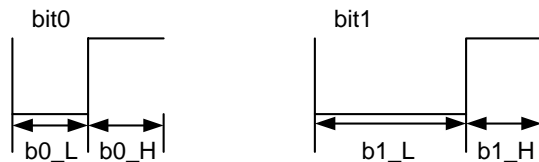
图11-40 持续按键连续发送 SONY 码帧格式



码格式

SONY 码 bit0 或 bit1 定义如图 11-41 所示。

图11-41 bit0 和 bit1 定义



注：图中高低电平的脉宽宽度以及帧长均有各个具体码型决定。请参见表 11-16~表 11-18。

11.4.5 工作方式

管脚复用配置

IR 管脚与 EBI 复用，使用 IR 前，需要配置 IO Config 寄存器，使能相应管脚的 IR 功能。详细内容请参见 reg74 配置说明。

时钟门控

配置系统控制器 SC_PEREN bit[11]为 1，使能 IR 模块的时钟 pclk；配置系统控制器 SC_PERDIS bit[11]为 1，关闭 IR 模块的时钟 pclk。

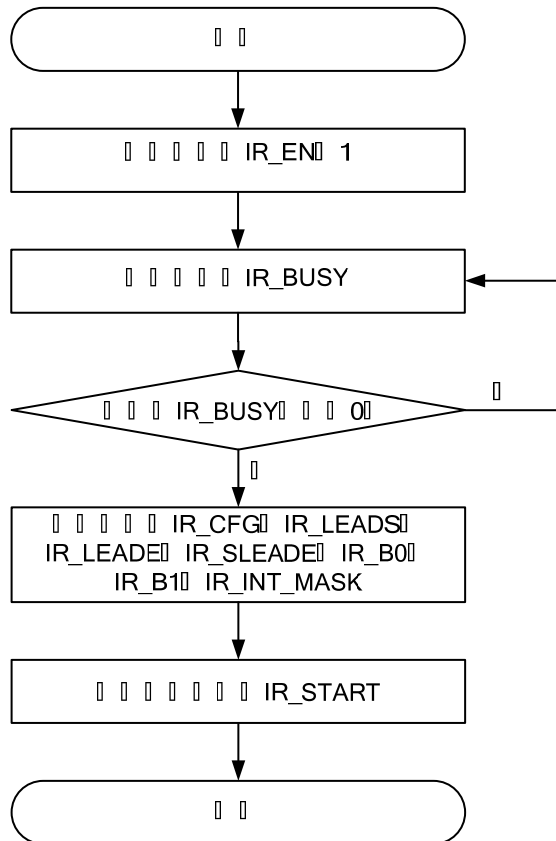
软复位

配置系统控制器 SC_PERCTRL8 bit[10]为 1，对 IR 模块单独软复位。复位后各配置寄存器的值均复位为默认值，因此复位后需要重新对这些寄存器进行初始化配置。

寄存器配置实例

IR 模块初始化操作流程如图 11-42 所示。

图11-42 IR 模块初始化操作流程



IR 模块初始化操作流程如下：

步骤 2 选中 IR 模块地址空间，开始 IR 初始化配置操作。

步骤 3 配置 `IR_EN[0]` 为 1，打开 IR 接收模块。

步骤 4 读 `IR_BUSY`，判断 IR 模块配置的当前状态。

- 若读取的值为 1，表明 IR 模块处于配置忙状态，则继续查询 `IR_BUSY`（注意：此时软件不要对 IR 模块的其他控制寄存器进行配置，否则配置无效）。
- 若读取的值为 0，表明 IR 模块处于配置空闲状态，则执行步骤 5。

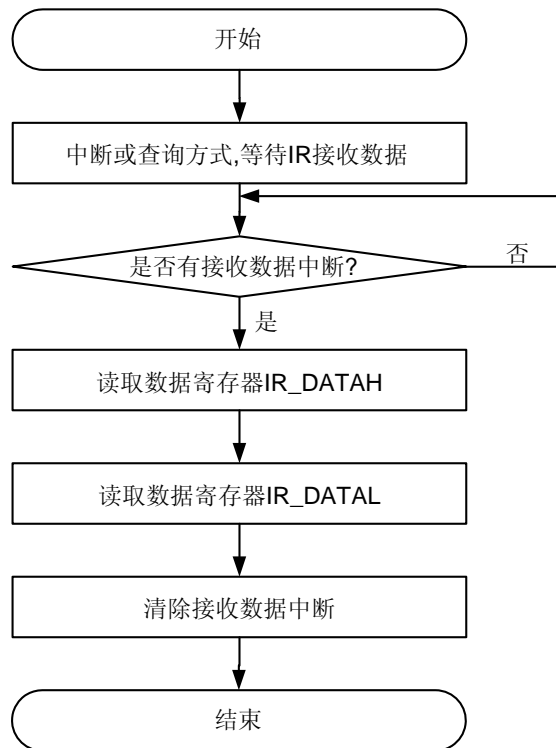
步骤 5 配置 `IR_CFG`、`IR_LEADS`、`IR_LEADE`、`IR_SLEADE`、`IR_B0`、`IR_B1`、`IR_INT_MASK`。注意：用户可以根据需要更新相应寄存器，如果不更新，则寄存器保持原值。

步骤 6 配置 `IR_START`。必须要等所有的 IR 控制寄存器都配置完成后，才能配置 `IR_START`，因为它被用来产生启动信号，只要对其进行配置，IR 模块就会根据控制寄存器的值进行红外数据接收。

----结束



图11-43 读取解码数据的操作流程



读取解码数据的操作流程如下：

步骤 1 选中 IR 模块地址空间。

步骤 2 中断或查询方式等待接收数据帧。

- 中断方式下，当 CPU 接收到 IR 模块的中断请求信号时，查询 `IR_INT_STATUS[intms_rcv]` 的值。若读取的值为 1 表明 IR 模块接收到一个数据帧，执行步骤 3；若读取的值为 0，重新执行步骤 2，继续等待中断。
- 查询方式下，软件不停（或每间隔一定时间）读取 `IR_INT_STATUS[intrs_rcv]` 的值，若读取的值为 1 表明 IR 模块接收到一个数据帧，执行步骤 3；若读取的值为 0 时，表明 IR 模块尚未接收到数据帧，重新执行步骤 2，继续查询。

步骤 3 读取数据寄存器 `IR_DATAH`。（如果一帧内的数据位数不大于 32 位，可以省略此步骤。）

步骤 4 读取数据寄存器 `IR_DATA L`。

步骤 5 清除接收数据中断。

----结束

11.4.6 IR 寄存器概览

IR 寄存器概览如表 11-19 所示。



表11-19 IR 寄存器概览（基址是 0x2007_0000）

偏移地址	名称	描述	页码
0x000	IR_EN	IR 接收使能控制寄存器	11-83
0x004	IR_CFG	IR 配置寄存器	11-84
0x008	IR_LEADS	引导码起始位裕量配置寄存器	11-85
0x00C	IR_LEADE	引导码结束位裕量配置寄存器	11-86
0x010	IR_SLEADE	简化引导码结束位裕量配置寄存器	11-87
0x014	IR_B0	数据 0 的判断电平裕量配置寄存器	11-88
0x018	IR_B1	数据 1 的判断电平裕量配置寄存器	11-89
0x01C	IR_BUSY	配置忙标志寄存器	11-90
0x020	IR_DATAH	IR 接收解码数据的高 16 位寄存器	11-91
0x024	IR_DATAH	IR 接收解码数据的低 32 位寄存器	11-91
0x028	IR_INT_MASK	IR 中断屏蔽寄存器	11-91
0x02C	IR_INT_STATUS	IR 中断状态寄存器	11-93
0x030	IR_INT_CLR	IR 中断清除寄存器	11-94
0x034	IR_START	IR 启动配置寄存器	11-95

11.4.7 IR 寄存器描述

IR_EN

IR_EN 为 IR 接收使能控制寄存器。



注意

软件必须先配置寄存器 IR_EN[0]=0b1，才能配置其他寄存器，否则配置无效。当寄存器 IR_EN[0]=0b0 时，其他寄存器只可读不可写，且读出值为寄存器的复位值。



	Offset Address				Register Name				Total Reset Value																							
	0x000				IR_EN				0x0000_0000																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										ir_en					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:1]	-	reserved		保留。																											
	[0]	RW	ir_en		IR 接收模块的使能。 0: 关闭 IR 接收模块; 1: 打开 IR 接收模块。																											

IR_CFG

IR_CFG 为 IR 配置寄存器。



注意

必须在确保 `IR_BUSY[0]=0b0` 并且 `IR_EN[0]=0b1` 时，才能配置此寄存器，否则配置无效，寄存器保持原值。

IR 支持的参考时钟频率为 1MHz~128MHz，其与分频因子 `ir_freq` 的对应关系是：

- 当参考时钟频率为 1MHz 时，分频因子 `ir_freq` 需配置为 0x00。
- 当参考时钟频率为 128MHz 时，分频因子 `ir_freq` 需配置为 0x7F。

当 IR 的参考时钟为 1MHz~128MHz 内的非整数倍频率时，选用四舍五入的方法选择相应的分频因子。举例：参考时钟为 12.1MHz，选用分频因子为 0x0B；参考时钟为 12.8MHz，选用分频因子为 0x0C。

对于频偏和计数偏差的关系：基频 f ，频率变化 Df ，则频偏率 $ratio=Df/f$ ；计数器计数偏差 $Dcnt$ ；判断电平宽度 s (μs 为单位)，则计数偏差： $Dcnt = \lceil 0.1 \times s \times ratio \rceil$ 。所以在时钟有频偏的情况下，参数值的有效范围要移位，如果频率上升，相应的裕量值应改为： $[min+Dcnt, max+Dcnt]$ ，其中 min 和 max 为无偏移时的裕量值；如果频率下降，相应的裕量值应改为： $[min-Dcnt, max-Dcnt]$ 。以引导码的起始位裕量举例来说：假如基频为 100MHz，频率上漂 0.1MHz，那么 $ratio=0.1/100=0.001$ ，设 $s=9000 \mu s$ ，则 $Dcnt = \lceil 0.1 \times 9000 \times 0.001 \rceil = 1$ ，则 `ir_leads` 的裕量值应改为 $[0x033D, 0x3CD]$ 。



Offset Address		Register Name		Total Reset Value					
0x004		IR_CFG		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				ir_format	ir_bits		reserved	ir_freq
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:14]	RW	ir_format	数据码型。 00: NEC with simple repeat code 的数据格式； 01: TC9012 的数据格式； 10: NEC with full repeat code 的数据格式； 11: SONY 的数据格式。 关于具体码型属于哪类码族，请参见表 11-16~表 11-18。						
[13:8]	RW	ir_bits	一帧内的数据位。 0x00~0x2F: 分别对应一帧内包含 1~48 个数据位； 0x30~0x3F: 保留。 如果软件对该域配置 0x30~0x3F 范围内的值，则配置无效，ir_bits 保持原值不变。						
[7]	-	reserved	保留。						
[6:0]	RW	ir_freq	工作时钟分频因子。 0x00~0x7F: 分别对应工作时钟分频因子 1~128。						

IR_LEADS

IR_LEADS 为引导码起始位裕量配置寄存器。



注意

必须在确保 `IR_BUSY[0]=0b0` 并且 `IR_EN[0]=0b1` 时，才能配置此寄存器，否则配置无效，寄存器保持原值。

为了准确判断引导码的起始位，需要在具体码型的典型值左右考虑一定的裕量，具体码型的典型值请参见表 11-16~表 11-18 中 LEAD_S 的值。



- 对于典型值不小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 LEAD_S 的典型值为 900，那么相应的 cnt_leads_min=900×92%=828=0x33C，cnt_leads_max=900×108%=972=0x3CC。
- 对于典型值小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 20%。举例说明：SONY-D7C5 码型，其 LEAD_S 的典型值为 240，那么相应的 cnt_leads_min=240×80%=192=0xC0，cnt_leads_max=240×120%=288=0x120。

基本的配置原则：cnt_leads_max 不小于 cnt_leads_min，并且 cnt_leads_min 大于 cnt0_b_max 和 cnt1_b_max。

Offset Address		Register Name		Total Reset Value				
0x008		IR_LEADS		0x033C_03CC				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		cnt_leads_min		reserved		cnt_leads_max	
Reset	0 0 0 0	0 0 1 1	0 0 1 1	1 1 0 0	0 0 0 0	0 0 1 1	1 1 0 0	1 1 0 0
Bits	Access	Name	Description					
[31:26]	-	reserved	保留。					
[25:16]	RW	cnt_leads_min	引导码起始位的最小脉宽。 0x000~0x007：保留。					
[15:10]	-	reserved	保留。					
[9:0]	RW	cnt_leads_max	引导码起始位的最大脉宽。 0x000~0x007：保留。					

IR_LEADE

IR_LEADE 为引导码结束位裕量配置寄存器。



注意

- 必须在确保 IR_BUSY[0]=0b0 并且 IR_EN[0]=0b1 时，才能配置此寄存器，否则配置无效，寄存器保持原值。
- 对于 NEC with simple repeat code 的码族，其 cnt_sleade 的裕量范围和 cnt_leade 的裕量范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别简化引导码，会导致帧格式错误。

为了准确判断引导码的结束位，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 8%。具体码型的典型值请参见表 11-16~表 11-18 中 LEAD_E 的值。



- 对于典型值不小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 LEAD_E 的典型值为 450，那么相应的 cnt_leade_min=450×92%=414=0x19E，cnt_leade_max=450×108%=486=0x1E6。
- 对于典型值小于 400（其精度为 10 μs）的脉宽，建议裕量范围设为典型值的 20%。举例说明：SONY-D7C5 码型，其 LEAD_E 的典型值为 60，那么相应的 cnt_leade_min=60×80%=48=0x030，cnt_leade_max=60×120%=72=0x048。

基本的配置原则是：cnt_leade_max 不小于 cnt_leade_min 的值。

	Offset Address				Register Name								Total Reset Value																			
	0x00C				IR_LEADE								0x019E_01E6																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cnt_leade_min								reserved				cnt_leade_max															
Reset	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0
	Bits	Access	Name	Description																												
	[31:25]	-	reserved	保留。																												
	[24:16]	RW	cnt_leade_min	引导码结束位的最小脉宽。 0x000~0x007：保留。																												
	[15:9]	-	reserved	保留。																												
	[8:0]	RW	cnt_leade_max	引导码结束位的最大脉宽。 0x000~0x007：保留。																												

IR_SLEADE

IR_SLEADE 为简化引导码结束位裕量配置寄存器。



注意

- 必须在确保 IR_BUSY[0]=0b0 并且 IR_EN[0]=0b1 时，才能配置此寄存器，否则配置无效，寄存器保持原值。
- 对于 NEC with simple repeat code 的码族，cnt_sleade 的裕量范围和 cnt_leade 的裕量范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别简化引导码，会导致帧格式错误。
- 对于 NEC with simple repeat code 的数据格式，才需配置此寄存器；对于其他格式，无需配置此寄存器。

为了准确判断简化引导码的结束位，需要在具体码型的典型值左右考虑一定的裕量。具体码型的典型值请参见表 11-16~表 11-18 中 SLEAD_E 的值。



- 对于典型值不小于 225（其精度为 10 μ s）的脉宽，建议裕量范围设为典型值的 8%。举例说明：D6121 码型，其 SLEAD_E 的典型值为 225，那么相应的 cnt_sleade_min=225 \times 92%=207=0xCF，cnt_sleade_max=225 \times 108%=243=0xF3。
- 对于典型值小于 225（其精度为 10 μ s）的脉宽，建议裕量范围设为典型值的 20%。举例说明：比如某种码型其 SLEAD_E 的典型值为 60，那么相应的 cnt_sleade_min=60 \times 80%=48=0x30，cnt_sleade_max=60 \times 120%=72=0x48。

基本的配置原则是：cnt_sleade_max 不小于 cnt_sleade_min 的值。

Offset Address		Register Name		Total Reset Value				
0x010		IR_SLEADE		0x00CF_00F3				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		cnt_sleade_min		reserved		cnt_sleade_max	
Reset	0 0 0 0	0 0 0 0	1 1 0 0	1 1 1 1	0 0 0 0	0 0 0 0	1 1 1 1	0 0 1 1
Bits	Access	Name	Description					
[31:25]	-	reserved	保留。					
[24:16]	RW	cnt_sleade_min	简化引导码结束位的最小脉宽。 0x000~0x007：保留。					
[15:9]	-	reserved	保留。					
[8:0]	RW	cnt_sleade_max	简化引导码起始位的最大脉宽。 0x000~0x007：保留。					

IR_B0

IR_B0 为数据 0 的判断电平裕量配置寄存器。



注意

- 必须在确保 IR_BUSY[0]=0b0 并且 IR_EN[0]=0b1 时，才能配置此寄存器，否则配置无效，寄存器保持原值。
- 对于四类码型，bit0 判断电平裕量范围和 bit1 判断电平范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别 bit1，只能被误解接收到 bit0。

为了准确判断 bit0，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 20%。

- 对于 NEC with simple repeat code、NEC with simple repeat code 和 TC9012 这三类码型，其包含的具体码型的典型值请参见表 11-16~表 11-18 中 B0_H 的值。举例说明：D6121 码型，其 B0_H 的典型值为 56（其精度为 10 μ s），那么相应的 cnt0_b_min=56 \times 80%=45=0x2D，cnt0_b_max=56 \times 120%=67=0x43。



- 对于 SONY 的数据格式，其包含的具体码型的典型值请参见表 11-16~表 11-18 中 B0_L 的值。举例说明：SONY-D7C5 码型，其 B0_L 的典型值为 60（其精度为 10μs），那么相应的 $\text{cnt0_b_min}=60 \times 80\% = 48 = 0x30$ ， $\text{cnt0_b_max}=60 \times 120\% = 72 = 0x48$ 。

基本的配置原则是：cnt0_b_max 不小于 cnt0_b_min 的值。

	Offset Address 0x014								Register Name IR_B0								Total Reset Value 0x002D_0043															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cnt0_b_min				reserved				cnt0_b_max															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
Bits	Access		Name		Description																											
[31:23]	-		reserved		保留。																											
[22:16]	RW		cnt0_b_min		bit0 判断电平的最小脉宽。 0x00~0x07：保留。																											
[15:7]	-		reserved		保留。																											
[6:0]	RW		cnt0_b_max		bit0 判断电平的最大脉宽。 0x00~0x07：保留。																											

IR_B1

IR_B1 为数据 1 的判断电平裕量配置寄存器。



- 必须在确保 IR_BUSY[0]=0b0 并且 IR_EN[0]=0b1 时，才能配置此寄存器，否则配置无效，寄存器保持原值。
- 对于四类码型，bit0 判断电平裕量范围和 bit1 判断电平范围设置不能重合，否则当实际计数值落入重合范围内时，无法识别 bit1，只能被误解接收到 bit0。

为了准确判断 bit1，需要在具体码型的典型值左右考虑一定的裕量，裕量范围约为典型值的 20%。

- 对于 NEC with simple repeat code、NEC with simple repeat code 和 TC9012 这三类码族，其包含的具体码型的典型值请参见表 11-16~表 11-18 中 B1_H 的值。举例说明：D6121 码型，其 B1_H 的典型值为 169（其精度为 10μs），那么相应的 $\text{cnt1_b_min}=169 \times 80\% = 135 = 0x87$ ， $\text{cnt1_b_max}=169 \times 120\% = 203 = 0xCB$ 。
- 对于 SONY 的数据格式，其包含的具体码型的典型值请参见表 11-16~表 11-18 中 B1_L 的值。举例说明：SONY-D7C5 码型，其 B1_L 的典型值为 120（其精度为



10 μ s), 那么相应的 $cnt1_b_min=120\times 80\%=96=0x60$,
 $cnt1_b_max=120\times 120\%=144=0x90$ 。

基本的配置原则是: $cnt1_b_max$ 不小于 $cnt1_b_min$ 的值。

	Offset Address				Register Name								Total Reset Value																			
	0x018				IR_B1								0x0087_00CB																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cnt1_b_min								reserved				cnt1_b_max															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1
	Bits	Access	Name		Description																											
	[31:25]	-	reserved		保留。																											
	[24:16]	RW	cnt1_b_min		bit1 判断电平的最小脉宽。 0x000~0x007: 保留。																											
	[15:9]	-	reserved		保留。																											
	[8:0]	RW	cnt1_b_max		bit1 判断电平的最大脉宽。 0x000~0x007: 保留。																											

IR_BUSY

IR_BUSY 为配置忙标志寄存器。

	Offset Address				Register Name								Total Reset Value																			
	0x01C				IR_BUSY								0x0000_0000																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															ir_busy																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:1]	-	reserved		保留。																											
	[0]	RO	ir_busy		忙状态标志。 0: 空闲状态, 软件可以配置数据; 1: 忙状态, 软件不可以配置数据。																											



IR_DATAH

IR_DATAH 为 IR 接收解码数据的高 16 位寄存器。

IR_DATAH 是接收到的解码数据的高 16 位，IR_DATAH 是接收到的解码数据的低 32 位。具体哪些数据位有效取决于具体码型一帧内包含的有效数据位数，请参见表 11-16~表 11-18 的有效数据位。

数据存储原则：按照由高到低的顺序存储在 IR_DATAH 和 IR_DATAH 中（MSB……LSB），先存满 IR_DATAH，然后再存放 IR_DATAH，未用到的高位作为保留位。软件读取数据的顺序必须是：先读 IR_DATAH，然后再读 IR_DATAH。

对于具体每个数据位表示的含义，硬件不做判断，仅负责接收所有数据位，最终由软件统一处理。

Offset Address		Register Name		Total Reset Value					
0x020		IR_DATAH		0x0000_0000					
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0	
Name	reserved				ir_datah				
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
Bits	Access	Name	Description						
[31:16]	-	reserved	保留。						
[15:0]	RO	ir_datah	接收到的解码数据的高 16 位数据。						

IR_DATAH

IR_DATAH 为 IR 接收解码数据的低 32 位寄存器。

Offset Address		Register Name		Total Reset Value				
0x024		IR_DATAH		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	ir_datah							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:0]	RO	ir_datah	接收到的解码数据的低 32 位数据。					

IR_INT_MASK

IR_INT_MASK 为 IR 中断屏蔽寄存器。



注意

必须在确保 `IR_EN[0]=0b1` 时，才能配置此寄存器，否则配置无效，寄存器保持原值。如果中断全部屏蔽后，无法支持红外遥控唤醒功能。

涉及到的中断定义如下：

- 接收数据溢出中断
如果出现当前帧的数据 CPU 没有及时响应取走，而下一帧数据也已经收到的情况，下一帧数据将会覆盖当前帧数据，同时上报屏蔽前接收数据溢出错中断请求。
- 接收数据帧格式错误中断
如果接收到的数据帧不完整以及数据脉宽不满足裕量范围，则会上报屏蔽前的接收帧格式错误中断请求。
- 接收数据帧中断
当接收到一个完整的帧数据后，则会上报屏蔽前接收到数据帧中断请求。
- 支持按键释放的检测中断
对于 NEC with simple repeat code 和 TC9012 码族的数据格式，在检测到一个有效起始同步码之后的 160ms 内，如果没有再次检测到起始同步码，或者检测到非简化引导码而是有效数据帧时，则会上报屏蔽前遥控器按键释放中断。对于 NEC with full repeat code 和 SONY 两种码制不支持按键释放中断。

硬件没有中断优先级仲裁，任何一个屏蔽后的中断源有效，都会产生中断。

Offset Address		Register Name		Total Reset Value							
0x028		IR_INT_MASK		0x0000_0000							
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0			
Name	reserved							intm_release	intm_overflow	intm_framerr	intm_rev
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0			
Bits	Access	Name	Description								
[31:4]	-	reserved	保留。								
[3]	RW	intm_release	禁止或允许按键释放的中断请求。 0：允许中断请求； 1：禁止中断请求。								



[2]	RW	intm_overflow	禁止或允许接收数据溢出错的中断请求。 0: 允许中断请求; 1: 禁止中断请求。
[1]	RW	intm_framerr	禁止或允许接收帧格式错误的中断请求。 0: 允许中断请求; 1: 禁止中断请求。
[0]	RW	intm_rcv	禁止或允许接收到数据帧的中断请求。 0: 允许中断请求; 1: 禁止中断请求。

IR_INT_STATUS

IR_INT_STATUS 为 IR 中断状态寄存器。

Offset Address		Register Name		Total Reset Value				
0x02C		IR_INT_STATUS		0x0000_0000				
Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved			intms_release intms_overflow intms_framerr intms_rcv	reserved			intrs_release intrs_overflow intrs_framerr intrs_rcv
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Bits	Access	Name	Description					
[31:20]	-	reserved	保留。					
[19]	RO	intms_release	屏蔽后的按键释放的中断状态。 0: 未产生中断; 1: 已产生中断。					
[18]	RO	intms_overflow	屏蔽后的接收数据溢出错中断状态。 0: 未产生中断; 1: 已产生中断。					
[17]	RO	intms_framerr	屏蔽后的接收帧格式错误中断状态。 0: 未产生中断; 1: 已产生中断。					



[16]	RO	intms_rcv	屏蔽后的接收到数据帧中断状态。 0: 未产生中断; 1: 已产生中断。
[15:4]	-	reserved	保留。
[3]	RO	intrs_release	屏蔽前的按键释放的中断状态。 0: 未产生中断; 1: 已产生中断。
[2]	RO	intrs_overflow	屏蔽前的接收数据溢出错误中断状态。 0: 未产生中断; 1: 已产生中断。
[1]	RO	intrs_framerr	屏蔽前的接收帧格式错误中中断状态。 0: 未产生中断; 1: 已产生中断。
[0]	RO	intrs_rcv	屏蔽前的接收到数据帧中断状态。 0: 未产生中断; 1: 已产生中断。

IR_INT_CLR

IR_INT_CLR 为 IR 中断清除寄存器。

	Offset Address	Register Name	Total Reset Value							
	0x030	IR_INT_CLR	0x0000_0000							
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0									
Name	reserved						intc_release	intc_overflow	intc_framerr	intc_rcv
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									
Bits	Access	Name	Description							
[31:4]	-	reserved	保留。							
[3]	WO	intc_release	清除遥控器按键释放中断请求。 0: 不清除; 1: 清除。							



[2]	WO	intc_overflow	清除接收数据溢出错中断请求。 0: 不清除; 1: 清除。
[1]	WO	intc_framerr	清除接收帧格式错误中断请求。 0: 不清除; 1: 清除。
[0]	WO	intc_rcv	清除接收到数据帧中断请求。 0: 不清除, 1: 清除。 如果接收数据帧中断请求产生后, 软件未读走 IR_DATAL 中的数据就直接对本位进行写 1 操作, 无法清除该中断请求。

IR_START

IR_START 为 IR 启动配置寄存器。

在其他寄存器的值配置完成后, 启动 IR 模块时, 只要往该地址进行一次写操作 (写操作数可以为任意值), 就可以启动配置寄存器。

	Offset Address								Register Name								Total Reset Value															
	0x034								IR_START								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																ir_start															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bits	Access	Name		Description																											
	[31:1]	-	reserved		保留。																											
	[0]	WO	ir_start		IR 启动配置寄存器。																											

11.5 GPIO

11.5.1 概述

Hi3520 支持 8 组 GPIO (General Purpose Input/Output), 每组 GPIO 提供 8 个可编程的输入输出管脚。每个管脚可以配置为输入或者输出。这些管脚用于生成特定应用的输出信号或采集特定应用的输入信号。作为输入管脚时, GPIO 可作为中断源; 作为输出



管脚时，每个 GPIO 都可以独立地清 0 或置 1。部分 GPIO 管脚与其他管脚复用，管脚复用说明请参见“2.2 软件管脚复用”，相关的控制请参见“2.5 复用寄存器描述”。

11.5.2 特点

GPIO 模块具有以下特点：

- 每个 GPIO 管脚均可配置为输入或输出。
 - 作为输入管脚时，可作为中断源，每个 GPIO 管脚都具有独立的中断控制。
 - 作为输出管脚时，每个 GPIO 管脚都可以独立地清 0 或置 1。
- GPIO 的中断通过 [GPIO_IS](#) 等 7 个寄存器进行控制。通过这些寄存器可以选择中断源、极性以及边沿特性。GPIO 对应的中断寄存器请参见“3.4 INT”。
 - 当有多个中断同时发生时，将会统一汇集成一个中断进行上报，GPIO 的中断映射关系请参见表 3-4 VIC0 中断映射表和表 3-5 VIC1 中断映射表。
 - [GPIO_IS](#)、[GPIO_IBE](#)、[GPIO_IEV](#) 三个寄存器共同决定了中断源的特性和中断触发类别。

通过 [GPIO_RIS](#) 和 [GPIO_MIS](#) 分别读取中断的原始状态和屏蔽后的状态。通过 [GPIO_IE](#) 可以控制中断的最终上报情况。此外还提供了单独的 [GPIO_IC](#) 用于对中断状态进行清除控制。

11.5.3 信号描述

GPIO 接口信号如表 11-20 所示。复用时的配置信息请参见“11.5.5 工作方式”中的“管脚复用配置”。

表11-20 GPIO 接口信号描述

信号名称	方向	描述	对应管脚
GPIO0_0	I/O	GPIO 双向数据信号。与 I ² C 的数据信号复用。	SDA
GPIO0_1	I/O	GPIO 双向数据信号。与 I ² C 的时钟信号复用。	SCL
GPIO0_2	I/O	GPIO 双向数据信号。与 SIO0 的 I ² S 发送左右声道选择信号复用。	SIO0XFS
GPIO0_3	I/O	GPIO 双向数据信号。与 SIO0 发送位流时钟信号复用。	SIO0XCK
GPIO0_4	I/O	GPIO 双向数据信号。与 SIO0 的主时钟信号复用。	ACKOUT
GPIO0_5	I/O	GPIO 双向数据信号。与 SIO2 数据输入信号复用。	SIO2DI
GPIO0_6	I/O	GPIO 双向数据信号。与 SIO2 的 I ² S 发送左右声道选择信号复用。	SIO2RFS
GPIO0_7	I/O	GPIO 双向数据信号。与 SIO2 发送位流时钟信号复用。	SIO2RCK



信号名称	方向	描述	对应管脚
GPIO1_0	I/O	GPIO 双向数据信号。与 GMAC 发送侧时钟输出信号复用。	GTCLKOUT
GPIO1_1	I/O	GPIO 双向数据信号。与 GMAC 碰撞指示信号复用。	GCOL
GPIO1_2	I/O	GPIO 双向数据信号。与 GMAC 载波侦听信号复用。	GCRS
GPIO1_3	I/O	GPIO 双向数据信号。与 VI0 接口的水平同步信号复用。	VI0HS
GPIO1_4	I/O	GPIO 双向数据信号。与 VI0 接口的垂直同步信号复用。	VI0VS
GPIO1_5	I/O	GPIO 双向数据信号。与 VO0 标清图像输出时钟信号复用。	VO0CK
GPIO1_6	I/O	GPIO 双向数据信号。与 VO1 高清图像或者级联输出 DAT7 信号复用。	VO1DAT7
GPIO1_7	I/O	GPIO 双向数据信号。与 VO1 高清图像或者级联输出 DAT15 信号复用。	VO1DAT15
GPIO2_0	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT0 信号复用。	VO0DAT0
GPIO2_1	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT1 信号复用。	VO0DAT1
GPIO2_2	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT2 信号复用。	VO0DAT2
GPIO2_3	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT3 信号复用。	VO0DAT3
GPIO2_4	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT4 信号复用。	VO0DAT4
GPIO2_5	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT5 信号复用。	VO0DAT5
GPIO2_6	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT6 信号复用。	VO0DAT6
GPIO2_7	I/O	GPIO 双向数据信号。与 VO0 标清图像输出数据 DAT7 信号复用。	VO0DAT7
GPIO3_0	I/O	GPIO 双向数据信号。与 VOBDAT7 复用，表示为 VO RGB 输出的 blue 数据 DAT7。	GPIO3_0
GPIO3_1	I/O	GPIO 双向数据信号。与 Nand Flash 的空闲指示信号复用。	NFRB



信号名称	方向	描述	对应管脚
GPIO3_2	I/O	GPIO 双向数据信号。与 SMI 片选信号 1 复用。	SMICS1N
GPIO3_3	I/O	GPIO 双向数据信号。与 Nand Flash 片选信号 1 复用。	NFCS1N
GPIO3_4	I/O	GPIO 双向数据信号。与 VI2 接口的水平同步信号复用。	VI2HS
GPIO3_5	I/O	GPIO 双向数据信号。与 VI2 接口的垂直同步信号复用。	VI2VS
GPIO3_6	I/O	GPIO 双向数据信号。与 PCI 总线申请信号 3 复用。	PCIREQ3N
GPIO3_7	I/O	GPIO 双向数据信号。与 PCI 总线申请信号 4 复用。	PCIREQ4N
GPIO4_0	I/O	GPIO 双向数据信号。与 VI1 数据 0 信号复用。	VI1DAT0
GPIO4_1	I/O	GPIO 双向数据信号。与 VI1 数据 1 信号复用。	VI1DAT1
GPIO4_2	I/O	GPIO 双向数据信号。与 VI1 数据 2 信号复用。	VI1DAT2
GPIO4_3	I/O	GPIO 双向数据信号。与 VI1 数据 3 信号复用。	VI1DAT3
GPIO4_4	I/O	GPIO 双向数据信号。与 VI1 数据 4 信号复用。	VI1DAT4
GPIO4_5	I/O	GPIO 双向数据信号。与 VI1 数据 5 信号复用。	VI1DAT5
GPIO4_6	I/O	GPIO 双向数据信号。与 VI1 数据 6 信号复用。	VI1DAT6
GPIO4_7	I/O	GPIO 双向数据信号。与 VI1 数据 7 信号复用。	VI1DAT7
GPIO5_0	I/O	GPIO 双向数据信号。与 VI2 数据 0 信号复用。	VI2DAT0
GPIO5_1	I/O	GPIO 双向数据信号。与 VI2 数据 1 信号复用。	VI2DAT1
GPIO5_2	I/O	GPIO 双向数据信号。与 VI2 数据 2 信号复用。	VI2DAT2
GPIO5_3	I/O	GPIO 双向数据信号。与 VI2 数据 3 信号复用。	VI2DAT3
GPIO5_4	I/O	GPIO 双向数据信号。与 VI2 数据 4 信号复用。	VI2DAT4
GPIO5_5	I/O	GPIO 双向数据信号。与 VI2 数据 5 信号复用。	VI2DAT5
GPIO5_6	I/O	GPIO 双向数据信号。与 VI2 数据 6 信号复用。	VI2DAT6
GPIO5_7	I/O	GPIO 双向数据信号。与 VI2 数据 7 信号复用。	VI2DAT7
GPIO6_0	I/O	GPIO 双向数据信号。与 VI3 数据 0 信号复用。	VI3DAT0
GPIO6_1	I/O	GPIO 双向数据信号。与 VI3 数据 1 信号复用。	VI3DAT1
GPIO6_2	I/O	GPIO 双向数据信号。与 VI3 数据 2 信号复用。	VI3DAT2



信号名称	方向	描述	对应管脚
GPIO6_3	I/O	GPIO 双向数据信号。与 VI3 数据 3 信号复用。	VI3DAT3
GPIO6_4	I/O	GPIO 双向数据信号。与 VI3 数据 4 信号复用。	VI3DAT4
GPIO6_5	I/O	GPIO 双向数据信号。与 VI3 数据 5 信号复用。	VI3DAT5
GPIO6_6	I/O	GPIO 双向数据信号。与 VI3 数据 6 信号复用。	VI3DAT6
GPIO6_7	I/O	GPIO 双向数据信号。与 VI3 数据 7 信号复用。	VI3DAT7
GPIO7_0	I/O	GPIO 双向数据信号。与 PCI 总线仲裁信号 3 复用。	PCIGRANT 3N
GPIO7_1	I/O	GPIO 双向数据信号。与 PCI 总线仲裁信号 4 复用。	PCIGRANT 4N
GPIO7_2	I/O	GPIO 双向数据信号。与 V01 高清图像或者级联输出时钟信号复用。	VO1CK
GPIO7_3	I/O	GPIO 双向数据信号。与 VO RGB 输出的数据有效指示信号复用。	VORGBDV
GPIO7_4	I/O	GPIO 双向数据信号。与 VIO1 高清图像或者级联输出 DAT13 复用。	VO1DAT13
GPIO7_5	I/O	GPIO 双向数据信号。与 VIO1 高清图像或者级联输出 DAT14 复用。	VO1DAT14
GPIO7_6	I/O	不使用。	-
GPIO7_7	I/O	不使用。	-

11.5.4 功能描述

每组 GPIO 提供 8 个可编程的输入输出管脚。每个管脚可以配置为输入或者输出。这些管脚用于生成或采集特定应用的输出或输入信号。

GPIO 可以根据电平或跳变值产生可屏蔽的中断。GPIOINTR (General Purpose Input Output Interrupt) 信号给中断控制器一个指示，表示有中断发生。

11.5.5 工作方式

管脚复用配置

GPIO 的管脚均与其他管脚复用，在使用相应的 GPIO 管脚前，需要先配置 IO Config 复用管脚寄存器使能相应管脚的 GPIO 功能，具体情况请参见表 11-21。



表11-21 管脚复用配置

信号名称	使能配置	复用情况
GPIO0_0	reg63	与 I ² C 的数据信号 SDA 复用。
GPIO0_1	reg64	与 I ² C 的时钟信号 SCL 复用。
GPIO0_2	reg65	与 SIO0 的 I ² S 发送左右声道选择信号复用。
GPIO0_3	reg66	与 SIO0 发送位流时钟信号复用。
GPIO0_4	reg67	与 SIO0 的主时钟信号复用。
GPIO0_5	reg68	与 SIO2 数据输入信号、SIO1DO 信号复用。
GPIO0_6	reg69	与 SIO2 的 I ² S 发送左右声道选择信号、SIO1XFS 信号复用。
GPIO0_7	reg70	与 SIO2 发送位流时钟信号、SIO1XCK 信号复用。
GPIO1_0	reg75	与 GMAC 发送侧时钟输出信号复用。
GPIO1_1	reg76	与 GMAC 碰撞指示信号复用。
GPIO1_2	reg77	与 GMAC 载波侦听信号复用。
GPIO1_3	reg0	与 VI0 接口的水平同步信号、URXD2 信号复用。
GPIO1_4	reg1	与 VI0 接口的垂直同步信号、UTXD2 信号复用。
GPIO1_5	reg36	与 VO0 标清图像输出时钟信号、VORDAT0 信号复用。
GPIO1_6	reg53	与 VO1 高清图像输出 DAT7 信号、VOGDAT6 信号复用。
GPIO1_7	reg61	与 VO1 高清图像输出 DAT15 信号复用。
GPIO2_0	reg37	与 VO0 标清图像输出数据 DAT0 信号、VORDAT1 信号复用。
GPIO2_1	reg38	与 VO0 标清图像输出数据 DAT1 信号、VORDAT2 信号复用。
GPIO2_2	reg39	与 VO0 标清图像输出数据 DAT2 信号、VORDAT3 信号复用。
GPIO2_3	reg40	与 VO0 标清图像输出数据 DAT3 信号、VORDAT4 信号复用。
GPIO2_4	reg41	与 VO0 标清图像输出数据 DAT4 信号、VORDAT5 信号复用。



信号名称	使能配置	复用情况
GPIO2_5	reg42	与 VO0 标清图像输出数据 DAT5 信号、VORDAT6 信号复用。
GPIO2_6	reg43	与 VO0 标清图像输出数据 DAT6 信号、VORDAT7 信号复用。
GPIO2_7	reg44	与 VO0 标清图像输出数据 DAT7 信号、VOGDAT0 信号复用。
GPIO3_0	reg62	与 VOBDAT7 复用，表示为 VO RGB 输出的 blue 数据 DAT7。
GPIO3_1	reg73	与 Nand Flash 的空闲指示信号复用。
GPIO3_2	reg71	与 SMI 片选信号 1 复用。
GPIO3_3	reg72	与 Nand Flash 片选信号 1 复用。
GPIO3_4	reg18	与 VI2 接口的水平同步信号、URXD3 信号复用。
GPIO3_5	reg19	与 VI2 接口的垂直同步信号、URXD3 信号复用。
GPIO3_6	reg78	与 PCI 总线申请信号 3 复用。
GPIO3_7	reg79	与 PCI 总线申请信号 4 复用。
GPIO4_0	reg10	与 VI1 数据 0 信号、VOUDAT8 信号复用。
GPIO4_1	reg11	与 VI1 数据 1 信号、VOUDAT9 信号复用。
GPIO4_2	reg12	与 VI1 数据 2 信号、VOUDAT10 信号复用。
GPIO4_3	reg13	与 VI1 数据 3 信号、VOUDAT11 信号复用。
GPIO4_4	reg14	与 VI1 数据 4 信号、VOUDAT12 信号复用。
GPIO4_5	reg15	与 VI1 数据 5 信号、VOUDAT13 信号复用。
GPIO4_6	reg16	与 VI1 数据 6 信号、VOUDAT14 信号复用。
GPIO4_7	reg17	与 VI1 数据 7 信号、VOUDAT15 信号复用。
GPIO5_0	reg20	与 VI2 数据 0 信号复用。
GPIO5_1	reg21	与 VI2 数据 1 信号复用。
GPIO5_2	reg22	与 VI2 数据 2 信号复用。
GPIO5_3	reg23	与 VI2 数据 3 信号复用。
GPIO5_4	reg24	与 VI2 数据 4 信号复用。
GPIO5_5	reg25	与 VI2 数据 5 信号复用。



信号名称	使能配置	复用情况
GPIO5_6	reg26	与 VI2 数据 6 信号复用。
GPIO5_7	reg27	与 VI2 数据 7 信号复用。
GPIO6_0	reg28	与 VI3 数据 0 信号复用。
GPIO6_1	reg29	与 VI3 数据 1 信号复用。
GPIO6_2	reg30	与 VI3 数据 2 信号复用。
GPIO6_3	reg31	与 VI3 数据 3 信号复用。
GPIO6_4	reg32	与 VI3 数据 4 信号复用。
GPIO6_5	reg33	与 VI3 数据 5 信号复用。
GPIO6_6	reg34	与 VI3 数据 6 信号复用。
GPIO6_7	reg35	与 VI3 数据 7 信号复用。
GPIO7_0	reg80	与 PCI 总线仲裁信号 3 复用。
GPIO7_1	reg81	与 PCI 总线仲裁信号 4 复用。
GPIO7_2	reg45	与 V01 高清图像或者级联输出时钟信号、SDIOCK 复用。
GPIO7_3	reg82	与 VO RGB 输出的数据有效指示信号复用。
GPIO7_4	reg59	与 VIO1 高清图像或者级联输出 DAT13 信号、VOBDAT4、VGA0HS 信号复用。
GPIO7_5	reg60	与 VIO1 高清图像或者级联输出 DAT14 信号、VOBDAT5、VGA0VS 信号复用。
GPIO7_6	-	不使用。
GPIO7_7	-	不使用。

接口复位

上电复位时，所有的寄存器都被清 0，因此管脚默认为输入。

复位信号有效时，GPIO 有如下状态：

- 通过清除 GPIO_IE 中相应的比特位使中断无效。
- 所有的寄存器被清 0。
- 所有的管脚都被配置为输入。
- 原始中断寄存器都被清 0。
- 中断被设为边沿触发的中断。



通用输入输出

每个管脚可以配置为输入或者输出，具体步骤如下：

步骤 1 参考表 11-21 配置 IO Config 复用管脚寄存器的相应位，使能相应的 GPIO 管脚功能。

步骤 2 配置寄存器 GPIO_DIR，选择 GPIO 是作为输入还是输出。

- GPIO 用于输入：外部信号通过 GPIO 管脚送进来，此时可通过 GPIO_DATA 寄存器查看输入信号值。需要注意的是，输入的信号会同时送到和 GPIO 复用的管脚上。
- GPIO 用于输出：先向 GPIO_DATA 寄存器写入相应值后，再通过 GPIO 输出写入值。需要注意的是，此时若已使能 GPIO 中断功能，则当输出信号满足触发条件时，也会产生中断。

----结束

中断操作

如果要产生中断，且避免假中断，则必须按照下面的初始化顺序：

步骤 1 配置 GPIO_IS，选择边沿触发或电平触发。

步骤 2 配置 GPIO_IEV，选择下降沿/上升沿触发和高电平/低电平触发。

步骤 3 如果选择边沿触发，需配置 GPIO_IBE，选择单沿或双沿触发方式。

步骤 4 保证 GPIO 数据线在以上操作过程中保持稳定。

步骤 5 向寄存器 GPIO_IC 写 0xFF，清中断。

步骤 6 配置 GPIO_IE 为 1，使能中断。

----结束

GPIO 的中断设置由 7 个寄存器控制。当有一个或多个 GPIO 管脚产生中断，一个组合中断输出会送到中断控制器。边沿触发和电平触发有以下不同：

- 边沿触发的中断：软件必需清除该中断以使能更深的中断。
- 电平触发的中断：外部中断源应该保持该电平直到处理器识别到该中断。

11.5.6 寄存器概览

8 组 GPIO 寄存器的基地址如表 11-22 所示。

表11-22 8 组 GPIO 寄存器对应的基地址

寄存器	基地址
GPIO0	0x2015_0000
GPIO1	0x2016_0000
GPIO2	0x2017_0000



寄存器	基地址
GPIO3	0x2018_0000
GPIO4	0x2019_0000
GPIO5	0x201A_0000
GPIO6	0x201B_0000
GPIO7	0x201C_0000

表 11-23 是单组 GPIO 内部寄存器的偏移地址以及定义，GPIO0~GPIO7 具有相同的寄存器组。GPIO_n 对应的寄存器地址为：GPIO_n 基地址+该寄存器偏移地址。

表11-23 GPIO 寄存器概览

偏移地址	名称	描述	页码
0x000~0x3FC	GPIO_DATA	GPIO 数据寄存器	11-104
0x400	GPIO_DIR	GPIO 方向控制寄存器	11-105
0x404	GPIO_IS	GPIO 中断触发寄存器	11-106
0x408	GPIO_IBE	GPIO 双沿触发中断寄存器	11-106
0x40C	GPIO_IEV	GPIO 触发中断条件寄存器	11-106
0x410	GPIO_IE	GPIO 中断屏蔽寄存器	11-107
0x414	GPIO_RIS	GPIO 原始中断状态寄存器	11-107
0x418	GPIO_MIS	GPIO 屏蔽状态中断寄存器	11-108
0x41C	GPIO_IC	GPIO 中断清除寄存器	11-108
0x420	GPIO_RESERVED	GPIO 保留寄存器	11-109

11.5.7 寄存器描述

GPIO_DATA

GPIO_DATA 为 GPIO 数据寄存器。用来对输入或输出数据进行缓存。

当配置 GPIO_DIR 中对应位为输出时，写入 GPIO_DATA 寄存器的值将会输出到相应的管脚（注意需要配置正确的管脚复用）；如果配置为输入时，将会读取相应输入管脚的值。



注意

当 [GPIO_DIR](#) 相应的比特配置为输入时，有效读取的结果将返回管脚的值；当配置为输出的时候，有效读取的结果将返回写入的值。

GPIO_DATA 寄存器利用 PADDR[9:2]实现了读写寄存器比特的屏蔽操作。该寄存器对应 256 个地址空间。PADDR[9:2]分别对应 GPIO_DATA[7:0]，当相应的 bit 为高时，则可以对相应的位进行读写操作；反之，若对应 bit 为低则不能进行操作。例如：

- 若地址为 0x3FC (0b11_1111_1100)，则对 GPIO_DATA[7:0]这 8bit 操作全部有效。
- 若地址为 0x200 (0b10_0000_0000)，则仅对 GPIO_DATA[7]的操作有效。

	Offset Address			Register Name			Total Reset Value	
	0x000~0x3FC			GPIO_DATA			0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_data							
Reset	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description				
	[7:0]	RW	gpio_data	当 GPIO 配置为输入模式时，为 GPIO 输入数据；当 GPIO 配置为输出模式时，为输出数据。各比特均可独立控制。与 GPIO_DIR 配合使用。				

GPIO_DIR

GPIO_DIR 为 GPIO 方向控制寄存器。用来配置 GPIO 管脚方向。

	Offset Address			Register Name			Total Reset Value	
	0x400			GPIO_DIR			0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_dir							
Reset	0	0	0	0	0	0	0	0
	Bits	Access	Name	Description				
	[7:0]	RW	gpio_dir	GPIO 方向控制寄存器。bit[7:0]分别对应 GPIO_DATA [7:0]，各比特可独立控制。 0: 输入； 1: 输出。				



GPIO_IS

GPIO_IS 为 GPIO 中断触发寄存器。用来配置 GPIO 管脚触发电平方式。

Offset Address		Register Name					Total Reset Value	
0x404		GPIO_IS					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_is							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RW	gpio_is	GPIO 中断触发控制寄存器，bit[7:0]分别对应 GPIO_DATA[7:0]，各比特独立控制。 0: 边沿触发中断； 1: 电平触发中断。					

GPIO_IBE

GPIO_IBE 为 GPIO 双沿触发中断寄存器。用来配置 GPIO 管脚沿触发方式。

Offset Address		Register Name					Total Reset Value	
0x408		GPIO_IBE					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_ibe							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RW	gpio_ibe	GPIO 中断沿触发控制寄存器，bit[7:0]分别对应 GPIO_DATA[7:0]，各比特独立控制。 0: 单边沿触发中断，具体是上升沿还是下降沿触发由 GPIO_IEV 控制； 1: 双边触发中断。					

GPIO_IEV

GPIO_IEV 为 GPIO 触发中断条件寄存器。用来配置 GPIO 管脚触发中断条件。



Offset Address		Register Name					Total Reset Value	
0x40C		GPIO_IEV					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_iev							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RW	gpio_iev	GPIO 触发中断条件寄存器，bit[7:0]分别对应 GPIO_DATA[7:0] ，各比特独立控制。 0: 下降沿或低电平触发中断； 1: 上升沿或高电平触发中断。					

GPIO_IE

GPIO_IE 为 GPIO 中断屏蔽寄存器。用来屏蔽 GPIO 管脚中断。

Offset Address		Register Name					Total Reset Value	
0x410		GPIO_IE					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_ie							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RW	gpio_ie	GPIO 中断屏蔽寄存器，bit[7:0]分别对应 GPIO_DATA[7:0] ，各比特独立控制。 0: 屏蔽中断； 1: 不屏蔽中断。					

GPIO_RIS

GPIO_RIS 为 GPIO 原始中断状态寄存器。用来查询 GPIO 管脚原始中断状态。



Offset Address		Register Name					Total Reset Value	
0x414		GPIO_RIS					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_ris							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RO	gpio_ris	GPIO 原始中断寄存器，bit[7:0]分别对应 GPIO_DATA[7:0] ，指示未屏蔽的中断状态。该状态不受 GPIO_IE 寄存器屏蔽控制。 0: 已发生中断； 1: 未发生中断。					

GPIO_MIS

GPIO_MIS 为 GPIO 屏蔽状态中断寄存器。用来查询 GPIO 管脚屏蔽后的中断状态。

Offset Address		Register Name					Total Reset Value	
0x418		GPIO_MIS					0x00	
Bit	7	6	5	4	3	2	1	0
Name	gpio_mis							
Reset	0	0	0	0	0	0	0	0
Bits	Access	Name	Description					
[7:0]	RO	gpio_mis	GPIO 屏蔽后中断寄存器，bit[7:0]分别对应 GPIO_DATA[7:0] ，指示经屏蔽后的中断状态。该状态受 GPIO_IE 寄存器屏蔽控制。 0: 中断无效； 1: 中断有效。					

GPIO_IC

GPIO_IC 为 GPIO 中断清除寄存器。用来清除 GPIO 管脚产生的中断，同时清除 [GPIO_RIS](#) 寄存器和 [GPIO_MIS](#) 寄存器。



	Offset Address			Register Name			Total Reset Value		
	0x41C			GPIO_IC			0x00		
Bit	7	6	5	4	3	2	1	0	
Name	gpio_ic								
Reset	0	0	0	0	0	0	0	0	
	Bits	Access	Name	Description					
	[7:0]	WC	gpio_ic	GPIO 中断清除寄存器，bit[7:0]分别对应 GPIO_DATA[7:0]，各比特可独立控制。 0: 无影响； 1: 清除中断。					

GPIO_RESERVED

GPIO_RESERVED 为 GPIO 保留寄存器。必须按要求配置。

	Offset Address			Register Name			Total Reset Value		
	0x420			GPIO_RESERVED			0x00		
Bit	7	6	5	4	3	2	1	0	
Name	reserved								
Reset	0	0	0	0	0	0	0	0	
	Bits	Access	Name	Description					
	[7:0]	RW	reserved	必须设为 0x00。					



12 测试接口

12.1 概述

Hi3520 的测试接口 JTAG 符合 IEEE (Institute of Electrical and Electronics Engineers) 1149.1 标准, 可用于 ARM 软件调试和板级测试。

12.2 工作模式

Hi3520 有 2 种工作模式, 通过芯片管脚 TESTMODE 的配置可以实现这 2 种模式之间的切换。Hi3520 工作模式说明如表 12-1 所示。

表12-1 Hi3520 工作模式说明表

TESTMODE	模式说明
0	Hi3520 正常工作, 此时可通过 JTAG 对 ARM 软件进行调试。
1	Hi3520 处于测试模式, 此时可以进行芯片 DFT (Design For Test) 测试和板级互连测试。

12.3 JTAG 调试

12.3.1 接口信号

Hi3520 的 JTAG 接口信号与标准的 JTAG 接口信号完全一致, Hi3520 的 JTAG 接口信号的描述如表 12-2 所示。

表12-2 Hi3520 的 JTAG 接口信号表

信号名	信号描述
TCK	JTAG 时钟输入, 芯片内部下拉。建议单板下拉。



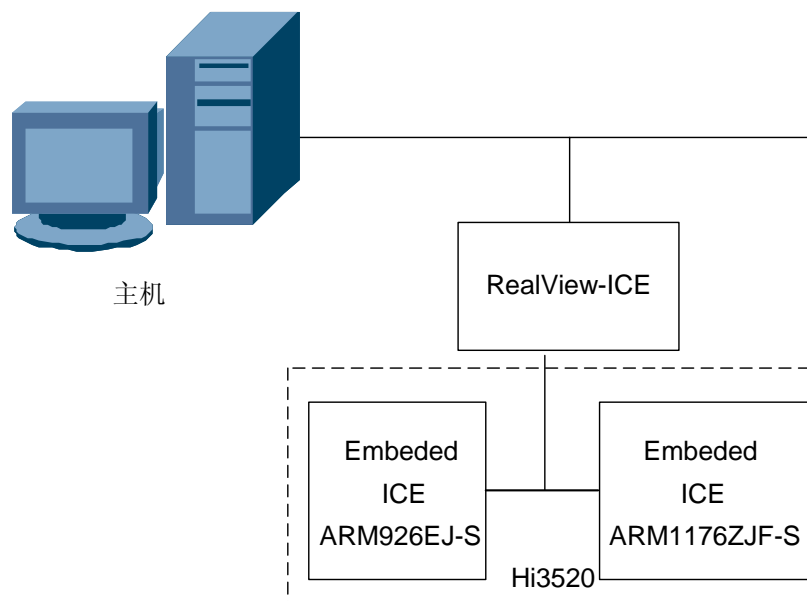
信号名	信号描述
TDI	JTAG 数据输入，芯片内部上拉。建议单板上拉。
TMS	JTAG 模式选择输入，芯片内部上拉。建议单板上拉。
TRSTN	JTAG 复位输入，芯片内部下拉。正常工作建议单板下拉。如果通过 JTAG 口连接 Realview-ICE 等调试器，建议单板上拉。
TDO	JTAG 数据输出。建议单板上拉。

12.3.2 调试模式

对 ARM 进行调试

用一个能够支持 JTAG 接口的 ICE 设备（如 RealView-ICE）和主机相连，并采用相应的调试软件，就可以通过 JTAG 接口对 2 个 ARM 进行调试，如图 12-1 所示。

图12-1 对 ARM 进行调试的系统示例图



通过 JTAG 接口单元（如 ARM RealView-ICE）进行初始设置时，需要将管脚 TESTMODE 设为 0。

系统可以通过对 EBIADR21 和 EBIADR22 进行上下拉处理，选择单独连接 ARM926、单独连接 ARM1176 或同时连接 ARM926 和 ARM1176，具体配置如表 12-3 所示。



注意

- 如果不使用 ARM 的 JTAG，推荐将 EBIADR21 和 EBIADR22 下拉；
- 对 EBIADR21 和 EBIADR22 上下拉操作更改后，需要对芯片进行管脚复位后才可生效。

表12-3 ARM9 和 ARM11 的连接设置

EBIADR21	EBIADR22	连接的具体情况
下拉	下拉	单独连接 ARM1176ZJF-S。
上拉	下拉	单独连接 ARM926EJ-S。
下拉	上拉	菊花链连接 ARM1176ZJF-S 和 ARM926EJ-S。
上拉	上拉	保留。

板级测试模式

Hi3520 除了支持通过 JTAG 接口进行软件调试外，还支持在单板上的一些互连测试，如 Hi3520 与其他芯片的连接测试等。板级互连测试通过标准的 JTAG 控制器实现。在进行板级互连测试时，需要将 TESTMODE 设置为 1。



13 视频处理

13.1 视频编解码器

13.1.1 概述

视频编解码器是一个支持 H.264、JPEG/MJPEG 协议的视频编码和视频解码的处理单元，它由运行于 ARM 处理器上的 Video Codec Firmware 和内嵌的视频编解码硬件加速引擎组成，支持单独进行视频编码或单独进行视频解码，也支持同时进行视频编码和视频解码、即同编同解。

13.1.2 特点

视频编解码器有以下特点：

- 支持 H.264 Main Profile@Level 4.0 及以下层次的编解码
 - 单独编码或者单独解码时最大支持 8 路 D1 实时编/解码（PAL：D1@200fps；NTSC：D1@240fps）或 2 路 720p30 实时编/解码
 - 同编同解时最大支持 4 路（D1+CIF）双码流编码+4 路 D1 解码 或 720p30 同编同解
- 支持 JPEG/MJPEG Baseline 编解码
 - 单独进行 JPEG 编码或解码时，在 300 万像素的最大分辨率下，帧率可达到 20fps
 - 支持 JPEG 同编同解
- 支持大小码流同时编码
 - 大小码流支持 H.264/H.264、H.264/JPEG、JPEG/H.264、JPEG/JPEG 四种协议组合
 - 大小码流使用相同的源图像，大码流直接由源图像编码得到，小码流由源图像缩小后编码得到
 - 大小码流编码图像尺寸的水平、垂直比例均可设置为 1:1、2:1、4:1
 - 小码流编码图像支持的最大尺寸为 CIF
- 支持编码前对隔行图像进行 de-interlace 处理
 - de-interlace 功能可使能/禁止
- 支持编码前时域滤波



- 时域滤波功能可开关
- 支持编码前 OSD 叠加处理
 - 支持最多 4 个区域的编码前 OSD 叠加
 - 支持任意大小，任意位置（不超出图像大小和位置）OSD 叠加
 - 支持 129 级的 alpha 叠加
 - OSD 叠加功能可开关
- 支持运动检测
 - 支持 SAD（Sum Of Absolute Difference）值输出
 - 支持运动矢量 MV 输出
- H264 支持 CBR/VBR/ABR 码率控制，16kbit/s~20Mbit/s
- 编码帧率可配置
 - 支持低帧率编码
 - 支持分数帧率编码
- 支持数字水印插入

13.2 TDE

13.2.1 概述

2D 图形加速引擎 TDE（Two Dimensional Engine）利用硬件进行图形绘制，可以大大减少对 CPU 的占用，同时提高了内存带宽的资源利用率。TDE 通过 AHB Master 总线接口读写位图数据、滤波缩放系数、链表节点参数信息以及一些链表信息等；通过 AHB Slave 总线接口获得 CPU 的寄存器配置信息。

图形数据接口包括源 1 和源 2 两条通路，其功能如下：

- 源 1 在单源操作时完成直接拷贝与直接填充的功能。
- 源 2 在单源操作时可完成各种复杂的操作，如图像缩放、抗闪烁等。

源 1 与源 2 协同工作时可以完成颜色混合、ROP 等操作，并且用来支持处理宏块格式的图像。

13.2.2 特点

TDE 模块具有以下功能特点：

- 源位图 1 支持 RGB444、RGB555、RGB565、RGB888、ARGB4444、ARGB1555、ARGB8565、ARGB8888、CLUT1、CLUT2、CLUT4、CLUT8、ACLUT44、ACLUT88、A1、A8、YCbCr888、AYCbCr8888、YCbCr422、byte、halfword、YCbCr400MB、YCbCr422MBH、YCbCr422MBV、YCbCr420MB、YCbCr444MB 的格式。源位图 2 支持 RGB444、RGB555、RGB565、RGB888、ARGB4444、ARGB1555、ARGB8565、ARGB8888、CLUT1、CLUT2、CLUT4、CLUT8、ACLUT44、ACLUT88、A1、A8、YCbCr888、AYCbCr8888、YCbCr422、YCbCr400MB、YCbCr422MBH、YCbCr422MBV、YCbCr420MB、YCbCr444MB 的格式。



- 输出位图支持 RGB444、RGB555、RGB565、RGB888、ARGB4444、ARGB1555、ARGB8565、ARGB8888、CLUT1、CLUT2、CLUT4、CLUT8、ACLUT44、ACLUT88、A1、A8、YCbCr888、AYCbCr8888、YCbCr422、byte、halfword、YCbCr400MB、YCbCr422MBH、YCbCr422MBV、YCbCr420MB、YCbCr444MB 的格式。
- 只支持小端系统。
- 支持源位图 1、源位图 2 和输出位图格式分别可配。
- 支持 Gamma 校正、亮度对比度的调节。
- 支持 CLUT 查找表。
- 支持 RGB 与 YCbCr 的转换。
- 支持直接拷贝。
- 支持直接填充。
- 支持 2D-resize 操作。
- 支持抗闪烁操作。
- 支持 clip 操作。
- 支持 alpha blending 操作。
- 支持 ROP 操作。
- 支持 colorkey 操作。
- 支持可编程扫描方式。
- 支持 clip mask 功能。
- 提供同步链表/异步链表方式的软件接口。
- 提供状态中断。



A 缩略语

A

ACD	Auto Command Done	自动停止指令完成
AES	Advanced Encryption Standard	先进的加密标准
AHB	Advanced High-performance Bus	-
AMBA	Advanced Microcontroller Bus Architecture	先进的微处理器总线结构
ARM	ARM	ARM 公司的 RISC Core

B

BVACT	Bottom Vertical Active Area	底场垂直活动有效区域
BVBB	Bottom Vertical Back Blank	底场垂直后消隐
BVFB	Bottom Vertical Front Blank	底场垂直前消隐

C

CBC	Cipher Block Chaining	密码分组链接
CD	Command Done	指令完成
CFB	Cipher Feedback	密码反馈
CL	CAS Latency	读延迟
CPU	Central Processing Unit	中央处理单元
CRC	Cyclic Redundancy Check	循环冗余校验
CRG	Clock Reset Generation	时钟复位产生模块
CTR	Counter	计数器

**D**

DCRC	Data CRC Error	数据 CRC 错误
DDR	Double Data-Rate	双数据速率
DES	Data Encryption Standard	数据加密标准
DFT	Design For Test	可测试性设计
DLL	Delay Locked Loop	延迟锁相环
DMA	Direct Memory Access	直接存储器存取
DMAC	Direct Memory Access Controller	直接存储器存取控制器
DQS	Data Strobe	数据选通信号
DRTO	Data Read Timeout	数据读超时
DTO	Data Transfer Over	数据传输完成
DVR	Digital Video Recorder	硬盘录像机

E

EBE	End-bit error	结束位错误
EBI	External Bus Interface	外部总线接口
ECB	Electronic Codebook	电子密码书
EOF	End Of Frame	帧结束
EOP	End Of Packet	包结束
ETH	Ethernet MAC	以太网接口

F

FIFO	First In First Out	先入先出
FIQ	Fast Interrupt Request	快速中断请求
FRUN	FIFO Underrun/Overrun Error	FIFO 溢出错误

G

GPIO	General Purpose Input/Output	通用目的输入/输出
-------------	------------------------------	-----------

H

HACT	Horizontal Active Area	水平活动有效区域
-------------	------------------------	----------



HCCA	Host Controller Communication Area	主机控制器通信区域
HFB	Horizontal Front Blank	水平前消隐
HLE	Hardware Locked Error	硬件锁定错误
HPW	Horizontal Pulse Width	水平脉冲宽度
HTO	Data starvation-by-host timeout	控制器读写数据超时
HBB	Horizontal Back Blank	水平后消隐
I		
I²C	Inter-Integrated Circuit	一种串行总线协议标准
IEEE	Institute of Electrical and Electronics Engineers	美国电气和电子工程师协会
I²S	Inter-IC Sound	I ² S 音频输入输出接口
IR	Infrared Remoter	红外遥控接口
IRQ	Interrupt Request	中断请求
ISR	Interrupt Service Routine	中断服务程序
ITCM	Instruction TCM	指令紧耦合存储器
IV	Initialization Vector	初始向量
J		
JTAG	Joint Test Action Group	联合测试行动小组
L		
LSB	Least Significant Bit	结尾 bit 位
M		
MAC	Media Access Control	媒体访问控制
MCU	Micro Controller Unit	微型控制单元
MDIO	Management Data Input/Output	控制数据输入输出接口
MII	Media Independent Interface	介质无关接口
MMC	Multi-media Card	多媒体卡
MSB	Most Significant Bit	起始 bit 位

**N**

NTSC National Television Systems Committee 国家电视系统委员会制式

O

OFB Output Feedback 输出反馈

OHCI Open Host Controller Interface 公开主机控制器接口

OSD On Screen Display 屏幕显示图形层

OTG On-The-Go -

P

PAL Phase Alternating Line 逐行倒向制式

PCB Printed Circuit Board 印刷电路板

PCI Peripheral Component Interconnect 一种通用的本地总线

PCM Pulse Code Modulation 脉冲编码调制

PID Packet ID 包标识

PSRAM Pseudo Static Random Access Memory 伪静态随机存储器

Q

QXGA Quantum Extended Graphics Array 昆腾扩展图像序列是一种计算机图形显示标准格式

R

RAM Random-Access Memory 随机存取存储器

RCRC Response CRC error 响应 CRC 错误

RE Response error 响应错误

ROM Read Only Memory 只读存储器

ROP Raster Operation 光栅操作

RTO Response Timeout 响应超时

RXDR Receive FIFO data request 接收 FIFO 数据请求

S

SAD Sum Of Absolute Difference 图象绝对误差和



SBE	Start-bit Error	起始位错误
SCL	Serial Clock	串行时钟
SCR	System Clock Reference	系统时钟参考
SD	Secure Digital	安全数字
SDA	Serial Data	串行数据
SDIO	secure digital Input/Output	安全数字输入输出接口
SDRAM	Synchronous Dynamic Random Access Memory	同步动态随机存取存储器
SFD	Start of Frame Delimiter	帧前导码
SI	Specific Information	特定信息
SIO	Sonic Input/Output	音频输入输出接口
SMI	Static Memory Interface	静态存储器接口
SOF	Start Of Frame	起始帧
SPI	Synchronous Peripheral Interface	同步外设接口
SRAM	Static Random Access Memory	静态随机存储器
SSP	Synchronous Serial Port	同步串口
T		
TCM	Tightly-Coupled Memory	紧耦合存储器
TDE	Two Dimension Engine	2D 引擎
TVACT	Top Vertical Active Area	顶场垂直活动有效区域
TVBB	Top Vertical Back Blank	顶场垂直后消隐
TVFB	Top Vertical Front Blank	顶场垂直前消隐
TXDR	Transmit FIFO Data Request	发送 FIFO 数据请求
U		
UART	Universal Asynchronous Receiver Transmitter	通用异步收发器
USB	Universal Serial Bus	通用串行总线
V		
VACT	Vertical Active Area	垂直活动有效区域
VBB	Vertical Back Blank	垂直后消隐



VBI	Vertical Blanking Interval	垂直消隐间隔
VEDU	Video Encoding Decoding Unit	视频编解码器
VFB	Vertical Front Blank	垂直前消隐
VIU	Video Input Unit	视频输入单元
VOU	Video Output Unit	视频输出单元
VPW	Vertical Pulse Width	垂直脉冲宽度