

FR30xx Specification

Bluetooth Low Energy SOC with SIG Mesh

版本号: v1.0

日期: 2023.07

FREQCHIP
富芮坤

修订版本

版本号	日期	更新内容
V1.0	2023.07.28	首版

Freqchip Confidential

目录

目录	I
表目录	XV
图目录	XVII
1. 系统概述	1
1.1. 概述	1
1.2. Arm CM1 (CM33) 处理器	2
1.3. 存储器地址映像	2
1.4. 内核中断映像	4
2. 系统控制单元 (System Controller)	7
2.1. 概述	7
2.2. 时钟控制	7
2.3. 系统复位	11
2.4. 系统低功耗	11
2.5. 系统控制寄存器	12
2.5.1. 系统寄存器映像	12
2.5.2. SOC Clock Source configure (Offset 0x00)	14
2.5.3. CPU BUS Clock Divider (Offset 0x04)	15
2.5.4. BLOCK Clock Select (Offset 0x08)	15
2.5.5. AUDIO Clock Select (Offset 0x0C)	18
2.5.6. BLOCK Clock Divider0 (Offset 0x10)	19
2.5.7. BLOCK Clock Divider1 (Offset 0x14)	20
2.5.8. BLOCK Clock Divider2 (Offset 0x18)	20
2.5.9. AUDIO Clock Divider (Offset 0x1C)	21
2.5.10. BUS_MEM Clock Enable (Offset 0x20)	21
2.5.11. CACHE_DMAC Clock Enable (Offset 0x24)	22
2.5.12. SDIO_CAN Clock Enable (Offset 0x28)	22
2.5.13. TIMER_WDT Clock Enable (Offset 0x2C)	23
2.5.14. GPIO_PWM Clock Enable (Offset 0x30)	23
2.5.15. UART_I2C Clock Enable (Offset 0x34)	23
2.5.16. SPI Clock Enable (Offset 0x38)	24
2.5.17. Algorithm Clock Enable (Offset 0x3C)	24
2.5.18. AUDIO Device Clock Enable (Offset 0x40)	25
2.5.19. MISC Clock Enable (Offset 0x44)	25
2.5.20. CPU_BUS Soft Reset (Offset 0x48)	26
2.5.21. CACHE_DMAC Soft Reset (Offset 0x4C)	26
2.5.22. SDIO_CAN Soft Reset (Offset 0x50)	27
2.5.23. TIMER_WDT Soft Reset (Offset 0x54)	27

2.5.24. GPIO_PWM Soft Reset (Offset 0x58)	28
2.5.25. UART_I2C Soft Reset (Offset 0x5C)	28
2.5.26. SPI Soft Reset (Offset 0x60)	29
2.5.27. Algorithm Soft Reset (Offset 0x64)	30
2.5.28. AUDIO Device Soft Reset (Offset 0x68)	31
2.5.29. MISC Soft Reset (Offset 0x6C)	32
2.5.30. PortA Input Enable (Offset 0x80)	32
2.5.31. PortB Input Enable (Offset 0x84)	33
2.5.32. PortC Input Enable (Offset 0x88)	33
2.5.33. PortD Input Enable (Offset 0x8C)	33
2.5.34. PortA Pull Enable (Offset 0xA0)	33
2.5.35. PortB Pull Enable (Offset 0xA4)	34
2.5.36. PortC Pull Enable (Offset 0xA8)	34
2.5.37. PortD Pull Enable (Offset 0xAC)	34
2.5.38. PortA Pull Select (Offset 0xC0)	34
2.5.39. PortB Pull Select (Offset 0xC4)	35
2.5.40. PortC Pull Select (Offset 0xC8)	35
2.5.41. PortD Pull Select (Offset 0xCC)	35
2.5.42. PORTA Pull Resistance Configure (Offset 0xE0)	35
2.5.43. PORTB Pull Resistance Configure (Offset 0xE4)	36
2.5.44. PORTC Pull Resistance Configure (Offset 0xE8)	36
2.5.45. PORTD Pull Resistance Configure (Offset 0xEC)	36
2.5.46. PORTA Drive configure (Offset 0x100)	37
2.5.47. PORTB Drive configure (Offset 0x104)	37
2.5.48. PORTC Drive configure (Offset 0x108)	37
2.5.49. PORTD Drive configure (Offset 0x10C)	37
2.5.50. PortA Function MUX_L (Offset 0x120)	38
2.5.51. PortA Function MUX_H (Offset 0x124)	38
2.5.52. PortB Function MUX_L (Offset 0x128)	39
2.5.53. PortB Function MUX_H (Offset 0x12C)	39
2.5.54. PortC Function MUX_L (Offset 0x130)	40
2.5.55. PortC Function MUX_H (Offset 0x134)	40
2.5.56. PortD Function MUX_L (Offset 0x138)	40
2.5.57. PortD Function MUX_H (Offset 0x13C)	41
2.5.58. PQSPI Input Enable (Offset 0x180)	41
2.5.59. PQSPI Pull Enable (Offset 0x184)	41
2.5.60. PQSPI Pull Select (Offset 0x188)	42
2.5.61. PQSPI Pull Resistance Configure (Offset 0x18C)	42
2.5.62. PQSPI Drive configure (Offset 0x190)	42
2.5.63. PQSPI Function MUX (Offset 0x194)	42
2.5.64. POSPI Input Enable (Offset 0x198)	43

2.5.65. POSPI Pull Enable (Offset 0x19C)	43
2.5.66. POSPI Pull Select (Offset 0x1A0)	43
2.5.67. POSPI Pull Resistance Configure (Offset 0x1A4)	43
2.5.68. POSPI Drive configure (Offset 0x1A8)	44
2.5.69. POSPI Function MUX (Offset 0x1AC)	44
2.5.70. PQSPI Data (Offset 0x1B0)	44
2.5.71. PQSPI Data Output Enable (Offset 0x1B4)	44
2.5.72. POSPI Data (Offset 0x1B8)	45
2.5.73. POSPI Data Output Enable (Offset 0x1BC)	45
2.5.74. SDIOH_MISC (Offset 0x1D0)	45
2.5.75. MSPI DATA SWAP (Offset 0x1D4)	46
2.5.76. USB MISC (Offset 0x1DC)	46
2.5.77. DMA_REQ_MUX (Offset 0x2C0~0x2D8)	46
3. 通用输入输出接口 (GPIO)	53
3.1. 概述	53
3.2. 主要特性	53
3.3. 结构框图	53
3.4. GPIO 寄存器	54
3.4.1. GPIO 寄存器地址映像	54
3.4.2. GPIO_OutputEN (Offset 0x00)	54
3.4.3. GPIO_IN_DATA (Offset 0x04)	55
3.4.4. GPIO_OUT_DATA (Offset 0x08)	55
3.4.5. GPIO_BIT_SET_DATA (Offset 0x0C)	55
3.4.6. GPIO_BIT_CLEAR_DATA (Offset 0x10)	56
3.4.7. EXTI_EN (Offset 0x14)	56
4. 外部中断控制器 (EXTI)	57
4.1. 概述	57
4.2. 主要特性	57
4.3. EXTI 系统框图	57
4.4. EXTI 寄存器	58
4.4.1. EXTI 寄存器映像	58
4.4.2. EXTI_INT_EN (Offset 0x00)	58
4.4.3. EXTI_INT_STATUS (Offset 0x04)	59
4.4.4. EXTI_TYPE (Offset 0x08)	59
4.4.5. EXTI_CNT0 ~ EXTI_CNT15 (Offset 0x0C ~ 0x44)	60
5. 通用异步收发器 (UART)	61
5.1. 概述	61
5.2. 主要特性	61
5.3. 功能描述	61
5.3.1. 波特率计算	61
5.3.2. IrDA SIR 1.0 协议	62

5.4. Uart 寄存器	63
5.4.1. Uart 寄存器映像	63
5.4.2. DATA/DLL (Offset 0x00)	64
5.4.3. IER/DLH (Offset 0x04)	65
5.4.4. FCR/IID (Offset 0x08)	66
5.4.5. LCR (Offset 0x0C)	67
5.4.6. MCR (Offset 0x10)	68
5.4.7. LSR (Offset 0x14)	69
5.4.8. MSR (Offset 0x18)	70
5.4.9. USR (Offset 0x7C)	71
5.4.10. DLF (Offset 0xC0)	71
5.5. 使用流程	73
5.5.1. 发送数据流程	73
5.5.2. 接收数据流程	74
6. 串行外设接口主设备 (SPI_Master)	75
6.1. 概述	75
6.2. 主要特性	75
6.3. 功能描述	76
6.3.1. 发送和接收 FIFO 及中断	76
6.3.2. 传输模式	76
6.3.3. DMA 接口	77
6.4. SPI 寄存器	77
6.4.1. SPI 寄存器映像	77
6.4.2. CTRLR0 (Offset 0x00)	78
6.4.3. CTRLR1 (Offset 0x04)	80
6.4.4. SSIENR (Offset 0x08)	81
6.4.5. MWCR (Offset 0x0C)	81
6.4.6. SER (Offset 0x10)	81
6.4.7. BAUDR (Offset 0x14)	82
6.4.8. TXFTLR (Offset 0x18)	82
6.4.9. RXFTLR (Offset 0x1C)	82
6.4.10. TXFLR (Offset 0x20)	82
6.4.11. RXFLR (Offset 0x24)	83
6.4.12. SR (Offset 0x28)	83
6.4.13. IMR (Offset 0x2C)	83
6.4.14. ISR (Offset 0x30)	84
6.4.15. RISR (Offset 0x34)	84
6.4.16. TXOICR (Offset 0x38)	84
6.4.17. RXOICR (Offset 0x3C)	85
6.4.18. RXUICR (Offset 0x40)	85
6.4.19. ICR (Offset 0x48)	85

6.4.20. DMACR (Offset 0x4C)	85
6.4.21. DMATDLR (Offset 0x50)	86
6.4.22. DMARDLR (Offset 0x54)	86
6.4.23. DR(offset 0x60)	86
6.4.24. RX_SAMPLE_DLY (Offset 0xF0)	86
6.4.25. CTRLR2 (Offset 0xF4)	87
6.5. 使用流程	88
6.5.1. SPI 标准模式发送	88
6.5.2. SPI 标准模式接收	89
6.5.3. Dual (二线)、Quad (四线)、Octal (八线) 写	89
6.5.4. Dual (二线)、Quad (四线)、Octal (八线) 读	90
6.6. MOTOROLA SPI 时序	93
7. 串行外设接口从设备 (SPI_Slave)	94
7.1. 概述	94
7.2. 主要特性	94
7.3. 功能描述	95
7.3.1. 发送和接收 FIFO 及中断	95
7.3.2. 传输模式	95
7.3.3. DMA 接口	96
7.4. SPI 寄存器	96
7.4.1. SPI 寄存器映像	96
7.4.2. CTRLR0 (Offset 0x00)	97
7.4.3. SSIENR (Offset 0x08)	99
7.4.4. MWCR (Offset 0x0C)	99
7.4.5. TXFTLR (Offset 0x18)	99
7.4.6. RXFTLR (Offset 0x1C)	100
7.4.7. TXFLR (Offset 0x20)	100
7.4.8. RXFLR (Offset 0x24)	100
7.4.9. SR (Offset 0x28)	100
7.4.10. IMR (Offset 0x2C)	101
7.4.11. ISR (Offset 0x30)	101
7.4.12. RISR (Offset 0x34)	102
7.4.13. TXOICR (Offset 0x38)	102
7.4.14. RXOICR (Offset 0x3C)	102
7.4.15. RXUICR (Offset 0x40)	103
7.4.16. ICR (Offset 0x48)	103
7.4.17. DMACR (Offset 0x4C)	103
7.4.18. DMATDLR (Offset 0x50)	103
7.4.19. DMARDLR (Offset 0x54)	104
7.4.20. DR(offset 0x60)	104
7.4.21. CTRLR2 (Offset 0xF4)	104

7.5. 使用流程	104
7.6. MOTOROLA SPI 时序	106
8. 内部集成电路总线接口 (I2C)	107
8.1. 概述	107
8.2. 主要特性	107
8.3. 功能描述	107
8.3.1. I2C 时序控制	107
8.3.2. I2C_SDA_HOLD 功能	109
8.3.3. SCL、SDA 拉低超时功能	109
8.3.4. 7bit 地址格式、10bit 地址格式	111
8.3.5. 突刺抑制功能	111
8.4. I2C 寄存器	112
8.4.1. I2C 寄存器映像	112
8.4.2. CONTROL (Offset 0x00)	114
8.4.3. TAR (Offset 0x04)	115
8.4.4. SAR (Offset 0x08)	116
8.4.5. DATA_CMD (Offset 0x10)	116
8.4.6. SCL_HCNT (Offset 0x1C)	117
8.4.7. SCL_LCNT (Offset 0x20)	117
8.4.8. INTR_STAT (Offset 0x2C)	117
8.4.9. INTR_MASK (Offset 0x30)	120
8.4.10. RAW_INTR_STAT (Offset 0x34)	121
8.4.11. RX_TL (Offset 0x38)	121
8.4.12. TX_TL (Offset 0x3C)	122
8.4.13. CLR_INTR (Offset 0x40)	122
8.4.14. CLR_RX_UNDER (Offset 0x44)	122
8.4.15. CLR_RX_OVER (Offset 0x48)	123
8.4.16. CLR_TX_OVER (Offset 0x4C)	123
8.4.17. CLR_RD_REQ (Offset 0x50)	123
8.4.18. CLR_TX_ABRT (Offset 0x54)	123
8.4.19. CLR_RX_DONE (Offset 0x58)	124
8.4.20. CLR_ACTIVITY (Offset 0x5C)	124
8.4.21. CLR_STOP_DET (Offset 0x60)	124
8.4.22. CLR_START_DET (Offset 0x64)	125
8.4.23. ENABLE (Offset 0x6C)	125
8.4.24. STATUS (Offset 0x70)	126
8.4.25. TXFLR (Offset 0x74)	127
8.4.26. RXFLR (Offset 0x78)	127
8.4.27. SDA_HOLD (Offset 0x7C)	127
8.4.28. TX_ABRT_SOURCE (Offset 0x80)	128
8.4.29. SLV_DATA_NACK_ONLY (Offset 0x84)	129

8.4.30. DMA_CR (Offset 0x88)	130
8.4.31. DMA_TDLR (Offset 0x8C)	130
8.4.32. DMA_RDLR (Offset 0x90)	130
8.4.33. SDA_SETUP (Offset 0x94)	131
8.4.34. SPKLEN (Offset 0xA0)	131
8.4.35. CLR_RESTART_DET (Offset 0xA8)	131
8.4.36. SCL_STUCK_AT_LOW_TIMEOUT (Offset 0xAC)	132
8.4.37. SDA_STUCK_AT_LOW_TIMEOUT (Offset 0xB0)	132
8.4.38. CLR_SCL_STUCK_DET (Offset 0xB4)	132
8.5. 使用流程	132
8.5.1. 主机使用流程	132
8.5.2. 从机使用流程	133
8.5.3. 工作时序图	134
9. 音频接口 (I2S)	137
9.1. 概述	137
9.2. 主要特性	137
9.3. 结构框图	137
9.4. 功能描述	138
9.4.1. BCLK、FRMCLK 分频	138
9.4.2. I2S_NORMAL, I2S_ADJUST, I2S_FRMINV 配置效果	138
9.5. I2S 寄存器	139
9.5.1. I2S 寄存器映像	139
9.5.2. CONTROL0 (Offset 0x00)	140
9.5.3. FRM_DIV (Offset 0x04)	141
9.5.4. CONTROL1 (Offset 0x08)	141
9.5.5. DATA_L (Offset 0x0C)	142
9.5.6. DATA_R (Offset 0x10)	142
9.5.7. STATUS (Offset 0x14)	143
9.5.8. INTE (Offset 0x18)	144
9.5.9. FIFO_CFG_L (Offset 0x1C)	144
9.5.10. FIFO_CFG_R (Offset 0x20)	145
9.5.11. DMA_CFG (Offset 0x24)	145
10. 真随机数发生器 (TRNG)	146
10.1. 概述	146
10.2. 主要特性	146
10.3. TRNG 结构框图	146
10.4. TRNG 寄存器	146
10.4.1. TRNG 寄存器映像	146
10.4.2. TRNG_IMR (Offset 0x100)	147
10.4.3. TRNG_ISR (Offset 0x104)	148
10.4.4. TRNG_ICR (Offset 0x108)	148

10.4.5. TRNG_CONFIG (Offset 0x10C)	149
10.4.6. TRNG_VALID (Offset 0x110)	149
10.4.7. TRNG_DATA0 ~ 5 (Offset 0x114 ~ 0x128)	149
10.4.8. RND_SOURCE_ENABLE (Offset 0x12C)	150
10.4.9. SAMPLE_CNT1 (Offset 0x130)	150
10.4.10. TRNG_RESET (Offset 0x140)	150
10.4.11. TRNG_BUSY (Offset 0x1B8)	150
10.4.12. RST_BITS_COUNTER (Offset 0x1BC)	151
11. 数字麦克风 (PDM)	152
11.1. 概述	152
11.2. 主要特性	152
11.3. PDM 系统框图	152
11.4. PDM 寄存器	153
11.4.1. PDM 寄存器映像	153
11.4.2. PDM_Control (Offset 0x00)	153
11.4.3. RXFF_FTLR (Offset 0x04)	155
11.4.4. PDM_FIFO_RESET (Offset 0x08)	155
11.4.5. PDM_INTEN (Offset 0x0C)	155
11.4.6. PDM_INT_RWST (Offset 0x10)	156
11.4.7. PDM_INT_STATUS (Offset 0x14)	156
11.4.8. PDM_VOLUME_L_CONFIG (Offset 0x18)	156
11.4.9. PDM_VOLUME_R_CONFIG (Offset 0x0x1C)	157
11.4.10. PDM_FIFO (Offset 0x24)	157
11.4.11. DMA_CFG (Offset 0x28)	157
12. DMA 控制器 (DMA)	158
12.1. 概述	158
12.2. 主要特性	158
12.3. 功能描述	158
12.3.1. DMA 优先级	159
12.3.2. 传输总长度、数据位宽、Burst length 三者关系	159
12.3.3. 链表传输功能	160
12.3.4. 离散与聚合功能	160
12.4. DMA 寄存器	162
12.4.1. DMA 寄存器映像	162
12.4.2. SARx (Offset 0x00, 0x58, 0xB0, 0x108, 0x160, 0x1B8, 0x210, 0x268)	163
12.4.3. DARx (Offset 0x08, 0x60, 0xB8, 0x110, 0x168, 0x1C0, 0x218, 0x270)	164
12.4.4. LLPx (Offset 0x10, 0x68, 0xC0, 0x118, 0x170, 0x1C8, 0x220, 0x278)	164
12.4.5. CTLx (Offset 0x18, 0x70, 0xC8, 0x120, 0x178, 0x1D0, 0x228, 0x280)	164
12.4.6. CFGx (Offset 0x40, 0x98, 0xF0, 0x148, 0x1A0, 0x1F8, 0x250, 0x2A8)	166
12.4.7. SGRx(Offset 0x48, 0xA0, 0xF8, 0x150, 0x1A8, 0x200, 0x258, 0x2B0)	168
12.4.8. DSRx(Offset 0x50, 0xA8, 0x100, 0x158, 0x1B0, 0x208, 0x260, 0x2B8)	169

12.4.9. RawTfr (Offset 0x2C0)	169
12.4.10. RawErr (Offset 0x2E0)	169
12.4.11. StatusTfr (Offset 0x2E8)	170
12.4.12. StatusErr (Offset 0x308)	170
12.4.13. MaskTfr (Offset 0x310)	170
12.4.14. MaskErr (Offset 0x330)	171
12.4.15. ClearTfr (Offset 0x338)	172
12.4.16. ClearErr (Offset 0x358)	172
12.4.17. StatusInt (Offset 0x360)	173
12.4.18. DmaCfgReg (Offset 0x398)	173
12.4.19. ChEnReg (Offset 0x3A0)	174
12.5. 注意事项	174
13. 脉冲宽度调制 (PWM)	175
13.1. 概述	175
13.2. 主要特性	175
13.3. PWM 系统框图	175
13.4. PWM 寄存器	176
13.4.1. PWM 寄存器映像	176
13.4.2. PWM_Output_EN (Offset 0x04)	179
13.4.3. PWM_ENABLE (Offset 0x08)	179
13.4.4. PWM0~15_Posedge (Offset 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48.....)	179
13.4.5. PWM0~15_Negedge (Offset 0x14, 0x1C, 0x24, 0x2C, 0x34, 0x3C, 0x44, 0x4C.....)	179
13.4.6. PWM_Output_Select (Offset 0x90)	180
13.4.7. Capture_Prescale (Offset 0x94)	180
13.4.8. Capture_Status (Offset 0x98)	181
13.4.9. Capture_INT_EN (Offset 0x9C)	181
13.4.10. Capture_CTL (Offset 0xA0)	181
13.4.11. Capture_Value0~7 (Offset 0xA4 ~ 0xC0)	182
13.4.12. PWM_Update (Offset 0xE4)	182
13.4.13. PWM_Output_Value (Offset 0xF0)	182
13.4.14. PWM_Inverter_EN (Offset 0xF8)	183
13.4.15. PWM_CNT_EN (Offset 0xFC)	183
13.4.16. PWM0~15_Prescale	183
13.4.17. PWM0~15_Period	184
13.5. 使用流程	184
13.5.1. PWM 模式	184
13.5.2. Capture 输入捕获功能	185
14. 低功耗电源管理 (PMU)	186
14.1. 概述	186
14.2. 访问控制接口	186
14.3. PMU_CONTROL	187

14.3.1. PMU_CONTROL 寄存器映像	187
14.3.2. PMU_CLK_SEL (0x00)	187
14.3.3. FSM_CLKDIV (0x01)	187
14.3.4. CLK_EN (0x04)	188
14.3.5. RESET_CTRL (0x05)	188
14.4. PMU_IWDG	189
14.4.1. PMU_IWDG 功能描述	189
14.4.2. PMU_IWDG 系统框图	189
14.4.3. PMU_IWDG 寄存器映像	190
14.4.4. IWDG_CTRL (0x10)	190
14.4.5. IWDG_TOUT_LEN (0x11, 0x12)	190
14.4.6. IWDG_LEN (0x13, 0x14, 0x15, 0x16)	191
14.5. PMU_RTC	191
14.5.1. PMU_RTC 功能描述	191
14.5.2. PMU_RTC 寄存器映像	191
14.5.3. RTC_CTRL (0x17)	192
14.5.4. RTC_CNT (0x18, 0x19, 0x1A, 0x1B)	192
14.5.5. ALARM_A (0x1C, 0x1D, 0x1E, 0x1F)	193
14.5.6. ALARM_B (0x20, 0x21, 0x22, 0x23)	193
14.6. PMU_GPIO	194
14.6.1. PMU_GPIO 功能描述	194
14.6.2. PMU_GPIO 寄存器映像	194
14.6.3. PMU_IO_PE (0x44)	194
14.6.4. PMU_IO_PS (0x46)	195
14.6.5. PMU_IO_DAT (0x48)	195
14.6.6. PMU_IO_OEN (0x4A)	195
14.6.7. PMU_IO_Monitor_EN (0x4C)	196
14.6.8. PMU_IO_Reference_Status (0x4E)	196
14.6.9. PMU_IO_Monitor_Status (0x50)	196
14.6.10. PMU_IO_Monitor_Clear (0x52)	197
14.6.11. PMU_IO_Monitor_INT_EN (0x54)	197
14.7. PMU_INT_CTRL	198
14.7.1. PMU_INT_CTRL 功能描述	198
14.7.2. PMU_INT_CTRL 寄存器映像	198
14.7.3. INT_EN0 (0x32)	199
14.7.4. INT_EN1 (0x33)	199
14.7.5. INT_RAW0 (0x34)	200
14.7.6. INT_RAW1 (0x35)	201
14.7.7. INT_STS0 (0x36)	201
14.7.8. INT_STS1 (0x37)	201
14.7.9. Analog_Status (0x38)	202

14.7.10. Analog_Monitor_Level (0x39)	202
14.7.11. ADKey_Filter (0x3A)	203
14.7.12. LVD_Filter (0x3B)	203
14.7.13. Charge_ACOK_Filter (0x3C)	203
14.7.14. Charge_FULL_Filter (0x3D)	204
14.7.15. OTD_Filter (0x3F)	204
15. 液晶屏并行驱动接口 (PARALLEL_INTERFACE)	205
15.1. 概述	205
15.2. 主要特性	205
15.3. PARALLEL 结构框图	205
15.4. PARALLEL 寄存器	206
15.4.1. PARALLEL 寄存器地址映像	206
15.4.2. INTF_CFG (Offset 0x00)	206
15.4.3. CSX (Offset 0x04)	207
15.4.4. CRM (Offset 0x08)	207
15.4.5. BUS_STATUS (Offset 0x0C)	208
15.4.6. CFG (Offset 0x10)	208
15.4.7. DATA_WR_LEN (Offset 0x14)	208
15.4.8. DATA_CFG (Offset 0x18)	209
15.4.9. TX_FIFO (Offset 0x1C)	210
15.4.10. RD_REQ (Offset 0x20)	210
15.4.11. DAT_RD (Offset 0x24)	210
15.4.12. TXFF_AEMP_LV (Offset 0x28)	210
15.4.13. TXFF_CLR (Offset 0x2C)	210
15.4.14. INT_CONTROL (Offset 0x30)	211
15.4.15. INT_STATUS (Offset 0x34)	211
15.4.16. DMA (Offset 0x38)	212
15.5. 使用流程	212
15.5.1. 写命令或写参数	212
15.5.2. 写数据	213
15.5.3. 读数据	214
16. 通用定时器 (Timer)	216
16.1. 概述	216
16.2. 主要特性	216
16.3. Timer 寄存器	216
16.3.1. Timer 寄存器地址映像	216
16.3.2. LOAD_VALUE (Offset 0x00)	217
16.3.3. COUNT_VALUE (Offset 0x04)	217
16.3.4. CONTROL (Offset 0x08)	217
16.3.5. INT_CLR (Offset 0x0C)	217
16.3.6. INT_STATUS (Offset 0x10)	218

16.4. 使用流程	218
17. 数模转换器 (ADC)	219
17.1. 概述	219
17.2. 主要特性	219
17.3. 结构框图	219
17.4. 功能描述	220
17.5. ADC 寄存器	220
17.5.1. ADC 寄存器映像	220
17.5.2. Control (Offset 0x00)	221
17.5.3. Config (Offset 0x04)	222
17.5.4. Timing (Offset 0x08)	222
17.5.5. Channel_Status (Offset 0x18)	223
17.5.6. ADC_INT_Enable (Offset 0x20).....	223
17.5.7. ADC_INT_Status (Offset 0x24).....	223
17.5.8. ADC_INT_Raws (Offset 0x28).....	223
17.5.9. ACT0(Offset 0x2C)	224
17.5.10. ACT1(Offset 0x30)	225
17.5.11. ChannelMap[0 ~ 7] (Offset 0x34 ~ 0x50)	225
17.5.12. ChannelData[0 ~ 7] (Offset 0x74 ~ 0x90)	226
18. USB 全速设备接口 (USB)	227
18.1.1. 概述	227
18.1.2. 主要特性	227
18.2. USB 系统框图.....	227
18.3. USB 寄存器	228
18.3.1. USB 寄存器地址映像	228
18.3.2. USB 公共寄存器 (Offset 0x00~0x0F)	230
18.3.3. USB 端点 0 控制状态寄存器 (Offset 0x10~0x1F)	236
18.3.4. USB 端点 1~5 控制状态寄存器 (Offset 0x10~0x1F)	239
18.3.5. USB 端点 FIFO 寄存器 (Offset 0x20~0x34)	247
18.4. 使用流程	248
18.4.1. 端点选择	248
18.4.2. 端点 FIFO 配置	248
18.4.3. Device 模式发送	248
18.4.4. Device 模式接收	249
18.4.5. Device 模式注意事项	249
18.4.6. Device 模式暂停	250
18.4.7. Device 模式 SOF 包.....	250
18.4.8. Host 模式接收	250
18.4.9. Host 模式发送	251
18.4.10. Host 模式传输调度	252
18.4.11. Host 模式 Reset	252

18.4.12. Host 模式 Suspend	252
19. 循环冗余校验 (CRC)	253
19.1. 概述	253
19.2. 主要特性	253
19.3. 结构框图	253
19.4. 功能描述	254
19.5. CRC 寄存器	254
19.5.1. CRC 寄存器映像	254
19.5.2. CRC_CTRL(Offset 0x00)	255
19.5.3. CRC_STATUS(Offset 0x04)	255
19.5.4. CRC_DATA(Offset 0x08)	255
19.5.5. CRC_RESULT(Offset 0x0C)	256
20. AES 硬件加速器 (AES)	257
20.1. 概述	257
20.2. 主要特性	257
20.3. 功能描述	257
20.4. AES 寄存器	258
20.4.1. AES 寄存器映像	258
20.4.2. DATA_IN0~3 (Offset 0x00, 0x04, 0x08, 0x0C)	259
20.4.3. KEY0~7 (Offset 0x10, 0x14, 0x18, 0x1C, 0x20, 0x24, 0x28, 0x2C)	259
20.4.4. IV0~3 (Offset 0x30, 0x34, 0x38, 0x3C)	260
20.4.5. AES_CTRL (Offset 0x40)	260
20.4.6. AES_STATE (Offset 0x44)	261
20.4.7. DATA_OUT0~3 (Offset 0x48, 0x4C, 0x50, 0x54)	261
20.4.8. AES_CTRL (Offset 0x40)	262
20.4.9. AES_STATE (Offset 0x44)	262
20.4.10. DATA_OUT0~3 (Offset 0x48, 0x4C, 0x50, 0x54)	263
21. 散列处理器 (SHA)	264
21.1. 概述	264
21.2. 主要特性	264
21.3. SHA 寄存器	264
21.3.1. SHA 寄存器映像	264
21.3.2. SHA_CTRL (Offset 0x00)	265
21.3.3. SHA_INT_STATE (Offset 0x04)	266
21.3.4. HASH_VAL (Offset 0x08 ~ 0x24)	266
21.3.5. DATA_1 (Offset 0x48 ~ 0x84)	267
21.3.6. DATA_2 (Offset 0x88 ~ 0xB4)	267
22. 联系信息	268
23. 附录 I	269
24. 附录 II	277
25. 附录 III	278

Freqchip Confidential

表目录

表格 1-1 存储器映像.....	2
表格 1-2 中断映像表.....	4
表格 2-1 系统复位源.....	11
表格 2-2 低功耗模式.....	11
表格 2-3 系统寄存器映像.....	12
表格 3-1 GPIO 寄存器地址映像.....	54
表格 4-1 EXTI 寄存器映像.....	58
表格 5-1 Uart 寄存器映像.....	64
表格 6-1 SPIM 寄存器.....	77
表格 7-1 SPIS 寄存器.....	96
表格 7-2 RXFLR 寄存器.....	100
表格 8-1 I2C 时序参数与寄存器关系.....	107
表格 8-2 I2C 寄存器映像.....	113
表格 9-1 I2S 寄存器映像.....	139
表格 10-1 TRNG 寄存器映像.....	146
表格 11-1 PDM 寄存器映像.....	153
表格 12-1 DMA 寄存器映像.....	162
表格 13-1 PWM 寄存器映像.....	176
表格 14-1 PMU_CONTROL 寄存器映像.....	187
表格 14-2 PMU_IWDG 寄存器映像.....	190
表格 14-3 PMU_RTC 寄存器映像.....	191
表格 14-4 PMU_GPIO 寄存器映像.....	194
表格 14-5 PMU_INT_CTRL 寄存器映像.....	198
表格 15-1 LCD 寄存器地址映像.....	206
表格 16-1 Timer 寄存器地址映像.....	216
表格 17-1 ADC 寄存器映像.....	220
表格 18-1 USB 寄存器地址映像.....	228
表格 19-1 CRC 模式对比.....	254
表格 19-2 寄存器地址映像.....	254
表格 20-1 AES 不同密钥长度对比.....	257
表格 20-2 寄存器地址映像.....	258
表格 21-1 SHA 寄存器映像.....	264

表格 23-1 PortA_L 功能复用	269
表格 23-2 PortA_H 功能复用	270
表格 23-3 PortB_L 复用功能	271
表格 23-4 PortB_H 复用功能	272
表格 23-5 PortC_L 复用功能	273
表格 23-6 PortC_H 复用功能	274
表格 23-7 PortD_L 复用功能	275
表格 23-8 PortD_H 复用功能	276
表格 24-1 PMU PortA 复用功能	277

Freqchip Confidential

图目录

图 1-1 系统框图.....	1
图 2-1 外设时钟配置.....	10
图 3-1 GPIO 结构框图.....	53
图 4-1 EXTI 系统框图.....	57
图 5-1 IrDA 框图.....	62
图 5-2 IrDA 时序示意图.....	63
图 6-1 SPI Master 接线示意图.....	75
图 6-2 指令和地址使用标准模式、数据使用多线模式.....	90
图 6-3 指令使用标准模式、地址和数据使用多线模式。.....	90
图 6-4 指令、地址和数据使用多线模式.....	90
图 6-5 只有指令使用多线模式.....	90
图 6-6 读数据、指令和地址使用标准模式、数据使用多线模式.....	91
图 6-7 读数据、指令使用标准模式、地址和数据使用多线模式.....	91
图 6-8 读数据，指令、地址和数据使用多线模式.....	92
图 6-9 读数据，没有指令，没有地址，仅数据传输.....	92
图 6-10 MOTOROLA SPI 时序图.....	93
图 7-1 SPI Slave 接线示意图.....	94
图 7-2 MOTOROLA SPI 时序图.....	106
图 8-1 I2C 时序参数图.....	107
图 8-2 一个时钟周期时序图.....	108
图 8-3 SDA_TX_HOLD 使用示意图.....	109
图 8-4 SDA_RX_HOLD 使用示意图.....	109
图 8-5 主机发送 9 个 CLK 试图恢复 SDA.....	110
图 8-6 7bit 地址示意图.....	111
图 8-7 10bit 地址示意图.....	111
图 8-8 突刺抑制示意图.....	112
图 8-9 DATA_CMD 寄存器示意图.....	134
图 8-10 主机发送并产生 STOP.....	135
图 8-11 主机接收并产生 STOP 时序.....	135
图 8-12 主机发送并产生 Restart 时序.....	135
图 8-13 主机接收并产生 Restart 时序.....	136
图 8-14 主机发送，产生 STOP，Tx FIFO 非空.....	136

图 8-15 主机接收，产生 STOP，TxFIFO 非空.....	136
图 9-1 I2S 结构框图	137
图 9-2 飞利浦标准格式时序图.....	138
图 9-3 I2S 时序图（左对齐，FRME 反相）.....	139
图 9-4 I2S 时序图（右对齐，FRME 反相）.....	139
图 10-1 TRNG 结构框图	146
图 11-1 PDM 系统框图.....	152
图 12-1 离散传输示例图.....	161
图 12-2 源聚合传输示例图.....	162
图 13-1 PWM 系统框图.....	175
图 13-2 PWM 时序图.....	184
图 13-3 输入捕获时序图.....	185
图 14-1 PMU_IWDG 系统框图	189
图 15-1 PARALLEL 结构框图.....	205
图 15-2 8080 写命令时序.....	212
图 15-3 6800 写命令时序.....	213
图 15-4 8080 写数据时序.....	213
图 15-5 6800 写数据时序.....	214
图 15-6 8080 读数据时序.....	214
图 15-7 6800 读数据时序.....	215
图 17-1 ADC 结构框图.....	219
图 18-1 USB 系统框图	227
图 19-1 CRC 结构框图.....	253

1. 系统概述

1.1. 概述

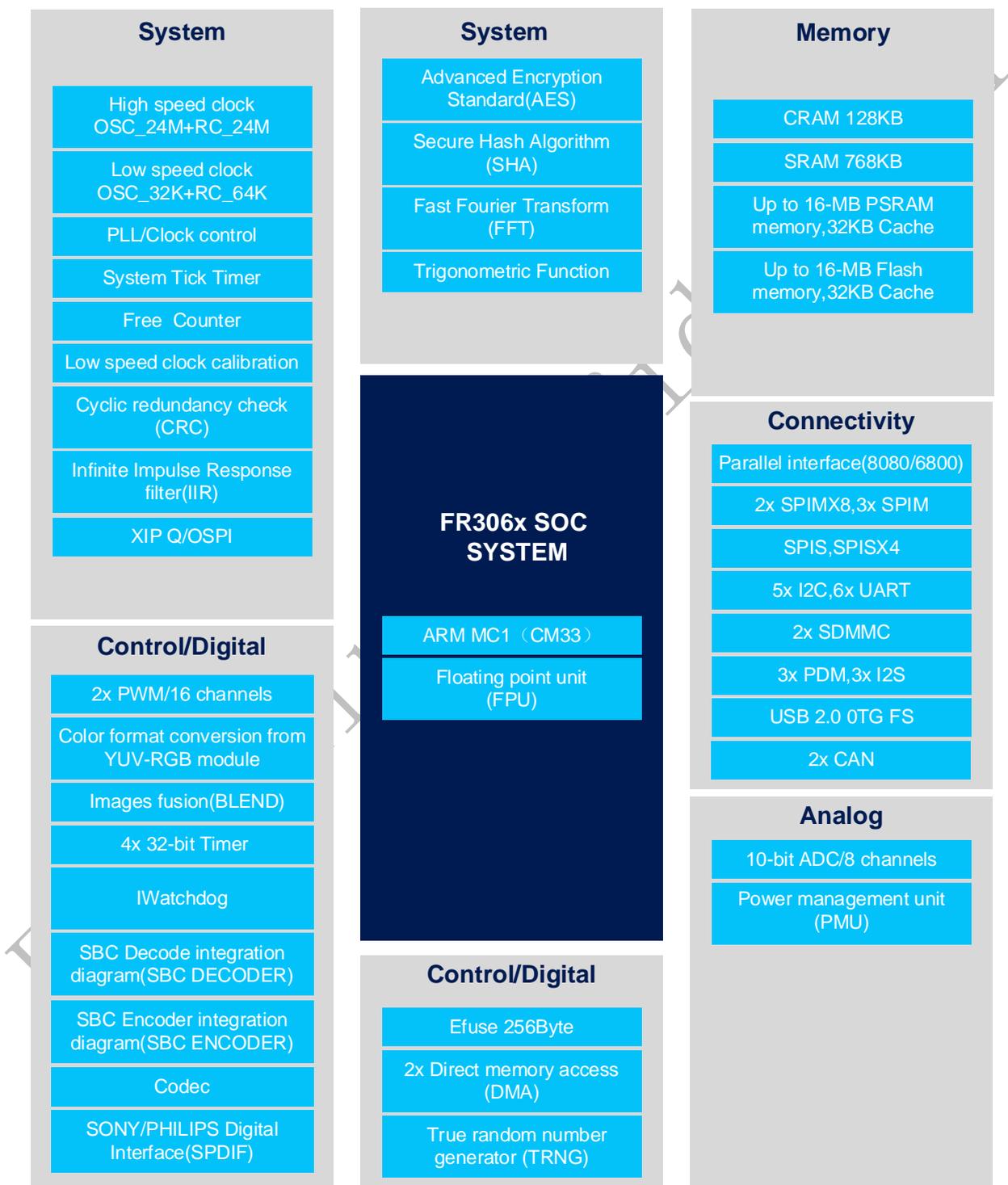


图 1-1 系统框图

1.2. Arm CM1 (CM33) 处理器

CM1 (CM33) 处理器是一个 32 位处理器，具有低中断延迟和低调试成本的特点。处理器基于 ARMv8 架构，支持硬件浮点运算 (FPU)。适用于需要高性能和低功耗的微控制器市场产品。

1.3. 存储器地址映像

表格 1-1 存储器映像

起始地址	描述	总线
0x0000_0000 ~ 0x0005_FFFF	BOOT ROM (128KB)	AHB
0x0800_0000 ~ 0x08FF_FFFF	Flash 最大支持 16M byte。 Flash cache 32K Byte。 (Fr30xx 系列芯片不同型号 Flash 大小有所差异)	
0x1FFE_0000 ~ 0x1FFF_FFFF	RAM 896K Byte。 系统 RAM 由 CRAM (128K), SRAM0 (256K), SRAM1 (256K), SRAM2 (256K), 组合构成。 (Fr30xx 系列芯片不同型号 RAM 大小有所差异)	
0x2000_0000 ~ 0x2000_3FFF		
0x2000_4000 ~ 0x2000_7FFF		
0x2000_8000 ~ 0x2000_BFFF		
0x3000_0000 ~ 0x30FF_FFFF	PSRAM 最大支持 16M byte。 通过 OSPI 扩展 RAM, 增加扩展后可通过同地址直接访问。	
0x1000_0000 ~ 0x1000_FFFF	DMAC0	
0x1001_0000 ~ 0x1001_FFFF	USBOTG	
0x4000_0000 ~ 0x4000_FFFF	SDIOH0	
0x4001_0000 ~ 0x4001_FFFF	SDIOH1	
0x4002_0000 ~ 0x4002_FFFF	DMAC1	
0x4003_0000 ~ 0x4003_FFFF	BlendIn	
0x4004_0000 ~ 0x4004_FFFF	BlendOut	
0x1010_0000 ~ 0x1010_FFFF	SBC_DEC	
0x1011_0000 ~ 0x1011_FFFF	SBC_ENC	
0x1013_0000 ~ 0x1013_FFFF	CRC	
0x1015_0000 ~ 0x1015_FFFF	PIPO efuse	
0x1016_0000 ~ 0x1016_FFFF	System Tick Timer	

0x1017_0000 ~ 0x1017_FFFF	Free Counter	
0x1018_0000 ~ 0x1018_FFFF	MCAN0	
0x1019_0000 ~ 0x1019_FFFF	MCAN1	
0x5000_0000	GPIOA	APB1
0x5000_8000	GPIOB	
0x5001_0000	UART0	
0x5001_8000	UART1	
0x5002_0000	I2C0	
0x5002_8000	I2C1	
0x5003_0000	SPIM0	
0x5004_0000	SPIS0	
0x5005_0000	PWM0	
0x5006_0000	I2S0	
0x5007_0000	PDM0	
0x5008_0000	IIR	
0x500A_0000	FFT	
0x500C_0000	SPIMX8_0	
0x500D_0000	PARALLEL_INTERFACE	
0x5010_0000	GPIOC	
0x5010_8000	GPIOD	
0x5011_0000	UART2	
0x5011_8000	UART3	
0x5012_0000	I2C2	
0x5012_8000	I2C3	
0x5013_0000	SPIM1	
0x5014_0000	SPIS1	
0x5015_0000	PWM1	
0x5016_0000	I2S1	
0x5017_0000	PDM1	
0x5018_0000	SPDIF	
0x5019_0000	CODEC	
0x501C_0000	SPIMX8_1	
0x5021_0000	UART4	APB3
0x5021_8000	UART5	
0x5022_0000	I2C4	

0x5022_8000	I2C5	
0x5023_0000	SPIM2	
0x5026_0000	I2S2	
0x5027_0000	PDM2	
0x502A_0000	YUV2RGB	
0xE005_0000	System Controller	EPPB
0xE006_0000	Timer0	
0xE006_0014	Timer1	
0xE006_8000	Timer2	
0xE006_8014	Timer3	
0xE009_0000	IPC	
0xE00A0000	CALI	
0xE00D0000	TRNG	
0xE00E0000	ADC	

1.4. 内核中断映像

表格 1-2 中断映像表

中断号	中断源
0	Timer0
1	Timer1
2	Timer2
3	Timer3
4	DMAC0
5	DMAC1
6	SDIOH0
7	SDIOH1
8	IPC
9	USBOTG
10	IIR
11	Blend
12	FFT
13	AES

中断号	中断源
14~15	
16	GPIOA
17	GPIOB
18	GPIOC
19	GPIOD
20	UART0
21	UART1
22	UART2
23	UART3
24	UART4
25	UART5
26	I2C0
27	I2C1
28	I2C2
29	I2C3
30	I2C4
31	I2C5
32	SPIM0
33	SPIM1
34	SPIM2
35	SPIS0
36	SPIS1
37	SPIMX8_0
38	SPIMX8_1
39	I2S0
40	I2S1
41	I2S2
42	PDM0
43	PDM1
44	PDM2
45	ADC
46	Codec
47	SPDIF

中断号	中断源
48	SBC_DEC
49	SBC_ENC
50	
51	PARALLEL
52	APP_WDT
53	CALI_IRQ
54	TRNG_IRQ
55	Tick
56~63	
64	YUV2RGB
65	PMU

2. 系统控制单元 (System Controller)

2.1. 概述

系统控制单元。提供对系统时钟、内核时钟、外设时钟的管理；提供各外设的复位控制；提供 I/O 端口控制。

2.2. 时钟控制

SOC 时钟源：

- HSE_24M (外部高速时钟) - 由外部晶振/陶瓷谐振器产生。
- HIS_24M (内部高速时钟) - 由内部高速 RC 振荡器产生。
- LSE (外部低速时钟) - 由外部晶振/陶瓷谐振器产生。
- LSI_32K (内部低速时钟) - 由内部低速 RC 振荡器产生。

SOC 主要时钟体系：

- CORE_HSCLK - 通过选择 HSE_24M/HIS_24M 时钟之一获得的时钟。
- SPLCLK、AUPLLCLK - 源自 HSE_24M 时钟通过 SPLL 锁相环倍频而来。

SOC 内核时钟：

- SYSCLK - 通过选择 CORE_HSCLK、SPLCLK (可以通过 1 到 15 时钟范围内的可编程系数进行分频的时钟) 两者之一获得的时钟

SYSCLK 作为系统时钟源可以通过 1 到 15 时钟范围内的可编程系数进行分频，其最大允许频率为 192MHz。可以为 APB、AHB、CORE、RAM、DMA0...提供时钟。

外设时钟源:

- OSPI、SDIO_x、USB、SPIMX8_x、SPIS_x、QSPI 有以下时钟源可供选择，可以通过 1 到 15 时钟范围内的可编程系数进行分频

- CORE_HSCLK
- SPLCLK
- AUPLLCLK

SDIO_x: SDIO0/1, SPIMX8_x: SPIMX8_0/1, SPIS_x: SPI0/1

- GPIO_x、I2C_x、UART_x、IIRFLT、CAN_x、FFT 有以下时钟源可供选择，可以通过 1 到 15 时钟范围内的可编程系数进行分频

- CORE_HSCLK
- 2 分频的 SPLCLK

GPIO_x: GPIOA/B/C/D, I2C_x: I2C0/1/2/3/4/5, UART_x: UART0/1/2/3/4/5,

CAN_x: CAN0/1

- SPIM_x、SBC、TRIFFUNC 有以下时钟源可供选择，可以通过 1 到 15 时钟范围内的可编程系数进行分频

- CORE_HSCLK
- SPLCLK

SPIM_x: SPIM0/1

- TIMEx、PWM_x、FRSPIM 有以下时钟源可供选择，可以通过 1 到 15 时钟范围内的可编程系数进行分频

- CORE_HSCLK

TIMEx: TIMER0/1/2/3, TIMER0/1 与 TIMER2/3 都有单独的时钟源选择与分频,

PWM: PWM0/1

- EFUSE_x、APP_TICK、SYS_TICK 有以下时钟源可供选择
 - CORE_HSCLK

EFUSE_x: EFUSE0/1
- Free Counter 有以下时钟源可供选择
 - CORE_HSCLK, 固定 24 分频 1MHZ。
- I2S_x、CDC 有以下时钟源可供选择, 可以通过 1 到 15 时钟范围内的可编程系数进行分频
 - CORE_HSCLK
 - SPLLCCLK
 - AUPLLCLK

I2C_x: I2S0/1/2
- PDM_x 有以下时钟源可供选择, 可以通过 1 到 15 时钟范围内的可编程系数进行分频
 - CORE_HSCLK
 - AUPLLCLK

PDM_x: PDM0/1/2, 每个 PDM 外设都有单独的时钟源选择与分频
- PARALLEL、SPDIF 有以下时钟源可供选择, 可以通过 1 到 15 时钟范围内的可编程系数进行分频
 - CORE_HSCLK
 - 2 分频的 SPLLCCLK
 - 2 分频的 AUPLLCLK

详细时钟树如下, 未缩放原图见[附录 III](#):

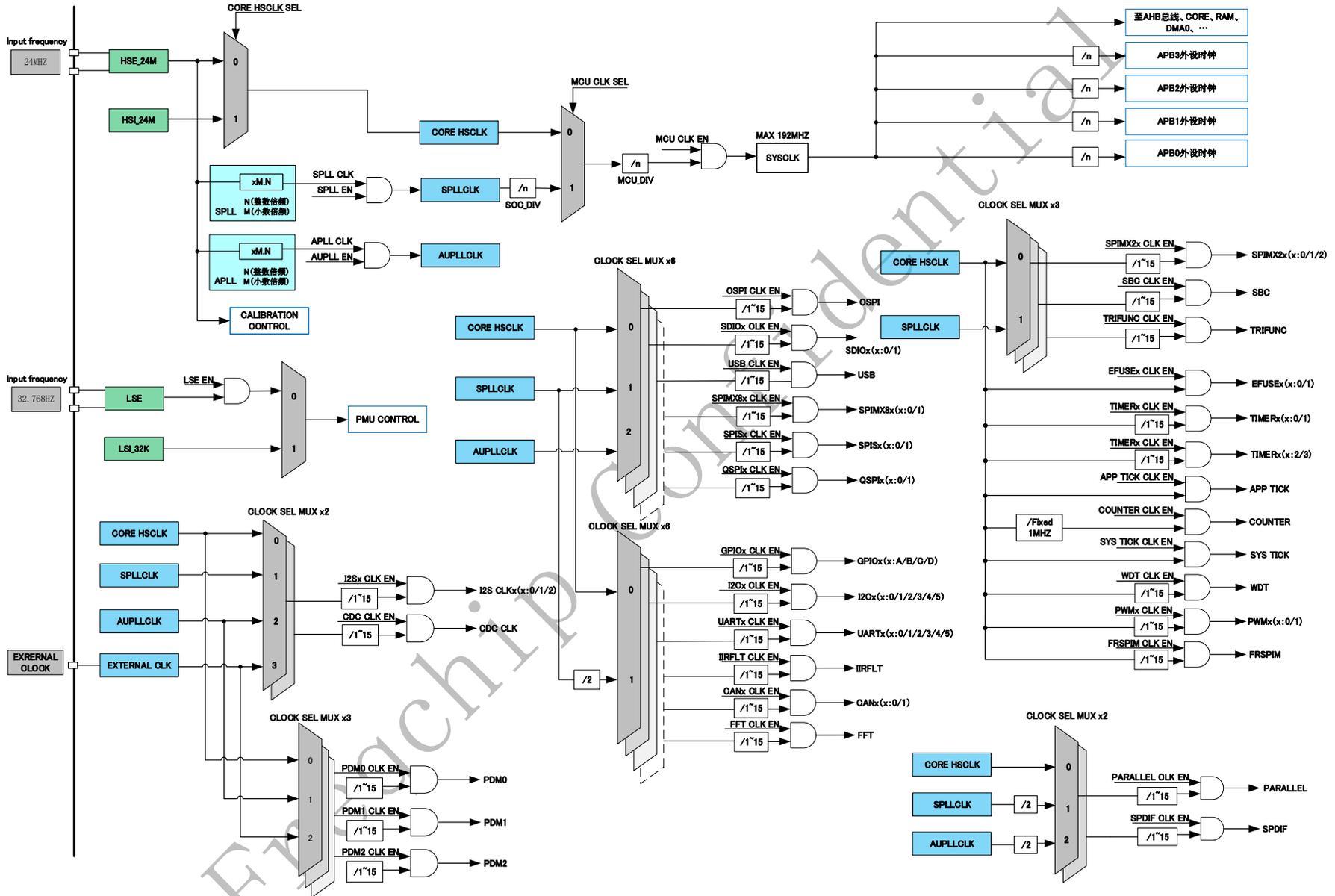


图 2-1 外设时钟配置

2.3. 系统复位

表格 2-1 系统复位源

复位源	描述
芯片上电 POR 复位	CPU 复位，所有外设复位，PMU 域复位
外部 IO Reset 复位	
PMU IWDT 复位	
软件调试复位	CPU 复位，所有外设复位

2.4. 系统低功耗

表格 2-2 低功耗模式

模式	描述	进入条件	退出条件
STOP	CPU 停止运行，除 CPU 外的其他模块仍可以工作，。	使用 WFI 指令可以进入 STOP 模式。	任意系统中断唤醒
SLEEP	CPI 停止运行，所有外设停止运行，部分 Memory 处于 retention 模式。PMU 域保持工作。	通过 PMU 域控制进入 SLEEP 模式	任意 PMU 域中断唤醒。

2.5. 系统控制寄存器

2.5.1. 系统寄存器映像

系统寄存器基地址：0xE005_00000

表格 2-3 系统寄存器映像

地址	寄存器	寄存器描述
0xE005_0000	SOC Clock Source configure	SOC 时钟源配置
0xE005_0004	CPU BUS Clock Divider	CPU 总线时钟分频
0xE005_0008	BLOCK Clock Select	Block 时钟源选择
0xE005_000C	AUDIO Clock Select	Audio 时钟源选择
0xE005_0010	BLOCK Clock Divider0	Block0 时钟分频
0xE005_0014	BLOCK Clock Divider1	Block1 时钟分频
0xE005_0018	BLOCK Clock Divider2	Block2 时钟分频
0xE005_001C	AUDIO Clock Divider	Audio 时钟分频
0xE005_0020	BUS_MEM Clock Enable	CPU 总线时钟使能
0xE005_0024	CACHE_DMACH Clock Enable	CACHE 和 DMACH 时钟使能
0xE005_0028	SDIO_CAN Clock Enable	SDIO 和 CAN 时钟使能
0xE005_002C	TIMER_WDT Clock Enable	TIMER 和 WDT 时钟使能
0xE005_0030	GPIO_PWM Clock Enable	GPIO 和 PWM 时钟使能
0xE005_0034	UART_I2C Clock Enable	UART 和 I2C 时钟使能
0xE005_0038	SPI Clock Enable	SPI 时钟使能
0xE005_003C	Algorithm Clock Enable	数字算法器时钟使能
0xE005_0040	AUDIO Device Clock Enable	Audio 时钟使能
0xE005_0044	MISC Clock Enable	MISC 时钟使能
0xE005_0048	CPU_BUS Soft Reset	CPU 总线软件复位
0xE005_004C	CACHE_DMACH Soft Reset	Cache 和 dma 软件复位
0xE005_0050	SDIO_CAN Soft Reset	SDIO 和 CAN 软件复位
0xE005_0054	TIMER_WDT Soft Reset	TIMER 和 WDT 软件复位
0xE005_0058	GPIO_PWM Soft Reset	GPIO 和 PWM 软件复位
0xE005_005C	UART_I2C Soft Reset	UART 和 I2C 软件复位
0xE005_0060	SPI Soft Reset	SPI 软件复位
0xE005_0064	HW_ENGINE Soft Reset	Engine 软件复位

0xE005_0068	AUDIO Device Soft Reset	Audio 软件复位
0xE005_006C	MISC Soft Reset	MISC 软件复位
0xE005_0080	PortA Input Enable	PortA 输入使能
0xE005_0084	PortB Input Enable	PortB 输入使能
0xE005_0088	PortC Input Enable	PortC 输入使能
0xE005_008C	PortD Input Enable	PortD 输入使能
0xE005_00A0	PortA Pull Enable	PortA 上下拉使能
0xE005_00A4	PortB Pull Enable	PortB 上下拉使能
0xE005_00A8	PortC Pull Enable	PortC 上下拉使能
0xE005_00AC	PortD Pull Enable	PortD 上下拉使能
0xE005_00C0	PortA Pull Select	PortA 上下拉选择
0xE005_00C4	PortB Pull Select	PortB 上下拉选择
0xE005_00C8	PortC Pull Select	PortC 上下拉选择
0xE005_00CC	PortD Pull Select	PortD 上下拉选择
0xE005_00E0	PORTA Pull Resistance Configure	PortA 上下拉电阻选择
0xE005_00E4	PORTB Pull Resistance Configure	PortB 上下拉电阻选择
0xE005_00E8	PORTC Pull Resistance Configure	PortC 上下拉电阻选择
0xE005_00EC	PORTD Pull Resistance Configure	PortD 上下拉电阻选择
0xE005_0100	PORTA Drive configure	PortA 驱动能力配置
0xE005_0104	PORTB Drive configure	PortB 驱动能力配置
0xE005_0108	PORTC Drive configure	PortC 驱动能力配置
0xE005_010C	PORTD Drive configure	PortD 驱动能力配置
0xE005_0120	PortA Function MUX_L	PortA_L 端口复用配置
0xE005_0124	PortA Function MUX_H	PortA_H 端口复用配置
0xE005_0128	PortB Function MUX_L	PortB_L 端口复用配置
0xE005_012C	PortB Function MUX_H	PortB_H 端口复用配置
0xE005_0130	PortC Function MUX_L	PortC_L 端口复用配置
0xE005_0134	PortC Function MUX_H	PortC_H 端口复用配置
0xE005_0138	PortD Function MUX_L	PortD_L 端口复用配置
0xE005_013C	PortD Function MUX_H	PortD_H 端口复用配置
0xE005_0180	PQSPI Input Enable	Port QSPI 输入使能
0xE005_0184	PQSPI Pull Enable	Port QSPI 上下拉使能
0xE005_0188	PQSPI Pull Select	Port QSPI 上下拉
0xE005_018C	PQSPI Pull Resistance Configure	Port QSPI 上下拉电阻选择

0xE005_0190	PQSPI Drive configure	Port QSPI 驱动能力配置
0xE005_0194	PQSPI Function MUX	Port QSPI 端口复用配置
0xE005_0198	POSPI Input Enable	Port OSPI 输入使能
0xE005_019C	POSPI Pull Enable	Port OSPI 上下拉使能
0xE005_01A0	POSPI Pull Select	Port OSPI 上下拉选择
0xE005_01A4	POSPI Pull Resistance Configure	Port OSPI 上下拉电阻选择
0xE005_01A8	POSPI Drive configure	Port OSPI 驱动能力配置
0xE005_01AC	POSPI Function MUX	PORT OSPI 端口复用配置
0xE005_01B0	PQSPI Data	PORT QSPI 输入输出数据寄存器
0xE005_01B4	PQSPI Data Output Enable	PORT QSPI 输出使能
0xE005_01B8	POSPI Data	PORT OSPI 输入输出数据寄存器
0xE005_01BC	POSPI Data Output Enable	PORT OSPI 输出使能
0xE005_01D0	SDIOH_MISC	SDIOH MISC 寄存器
0xE005_01D4	MSPI DATA SWAP	MSPI 数据线 SWAP 寄存器
0xE005_01DC	USB MISC	USB MISC 寄存器
0xE005_02C0~0xE005_02FC	DMA_REQ_MUX	DMAC 外设请求 ID 配置

2.5.2. SOC Clock Source configure (Offset 0x00)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9	MCU_CLK_SEL	R/W	1'b0	MCU 时钟选择 0: CORE_HSCLK (OSC24M/RC24M) 1: SPLL CLOCK
8	CORE_HSCLK_SEL	R/W	1'b0	CORE 高速时钟源选择 0: OSC_24M C 1: RC24M
7:4	SOC_CLK_DIV	R/W	4'b0	SOC 时钟分频 (MCU_CLK_SEL = 1) SYSTEM PLL CLK=SPLL_CLK/SOC_CLK_DIV+1
3	SPLL_CKDIG_DIV2_EN	R/W	1'b0	2 分频的 SPLL 时钟源外设时钟使能 0: Disable 1: Enable
2:0	Reserved	R/W	'd0	Reserved

2.5.3. CPU BUS Clock Divider (Offset 0x04)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:14	APB2_RATIO	R/W	2'b00	APB2 时钟分频 APB2_PCLK=SYSCLK/AHB_CLK/(Ratio+1)
13:12	APB1_RATIO	R/W	2'b00	APB1 时钟分频 APB1_PCLK= SYSCLK /AHB_CLK/(Ratio+1)
11:10	APB0_RATIO	R/W	2'b00	APB0 时钟分频 APB0_PCLK= SYSCLK /AHB_CLK/(Ratio+1)
9:8	APB_RATIO	R/W	2'b00	APB 时钟分频 APB_PCLK= SYSCLK /AHB_CLK/(Ratio+1)
7:4	MCU_ST_CLK_DIV	R/W	4'hB	MCU 内部 system tick 参考时钟分频 REF_CLK = CORE_HSCLK/(DIV+1)
3:0	MCU_CLK_DIV	R/W	4'h0	MCU clock 分频 SYSCLK = MCU_CLK/(DIV+1)

2.5.4. BLOCK Clock Select (Offset 0x08)

比特	名称	属性	复位值	描述
31:29	Reserved	R/W	'd0	Reserved
28	MCAN_CLK_SEL	R/W	1'b0	MCAN 时钟源选择 0: CORE_HSCLK 1: SPLL
27	BLEND_CLK_SEL	R/W	1'b0	Blend 时钟源选择 0: CORE_HSCLK 1: SPLL
26	SSIM_CLK_SEL	R/W	1'b0	SPIM 时钟源选择 0: CORE_HSCLK 1: SPLL
25	FFT_CLK_SEL	R/W	1'b0	FFT 时钟源选择 0: CORE_HSCLK 1: SPLL
24	TRIFUNC_CLK_SEL	R/W	1'b0	TRIFUNC 时钟源选择

				0: CORE_HSCLK 1: SPLL
23	IIRFLT_CLK_SEL	R/W	1'b0	IIRFLT 时钟源选择 0: CORE_HSCLK 1: SPLL
22	I2C_CLK_SEL	R/W	1'b0	I2C 时钟源选择 0: CORE_HSCLK 1: SPLL
21	UART_CLK_SEL	R/W	1'b0	UART 时钟源选择 0: CORE_HSCLK 1: SPLL
20	GPIO_CLK_SEL	R/W	1'b0	GPIO 时钟源选择 0: CORE_HSCLK 1: SPLL
19:18	PARALLEL_CLK_SEL	R/W	2'b00	PARALLEL 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
17:16	USB_CLK_SEL	R/W	2'b00	USB 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
15:14	MSPI1_CLK_SEL	R/W	2'b00	SPIMX8_1 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
13:12	MSPI0_CLK_SEL	R/W	2'b00	SPIMX8_0 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved

11:10	SSIS_CLK_SEL	R/W	2'b00	SPIS 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
9:8	OSPI_CLK_SEL	R/W	2'b00	OSPI 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
7:6	QSPI1_CLK_SEL	R/W	2'b00	QSPI1 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
5:4	QSPI0_CLK_SEL	R/W	2'b00	QSPI0 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
3:2	SDIOH1_CLK_SEL	R/W	2'b00	SDIOH1 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved
1:0	SDIOH0_CLK_SEL	R/W	2'b00	SDIOH0 时钟源选择 00: CORE_HSCLK 01: SPLL 10: Audio PLL 11: reserved

2.5.5. AUDIO Clock Select (Offset 0x0C)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	'd0	Reserved
10	SBC_CLK_SEL	R/W	1'b0	SBC 时钟源选择: 0: CORE_HSCLK 1: SPLL
9:8	PDM2_CLK_SEL	R/W	2'b00	PDM2 时钟源选择: 00: CORE_HSCLK 01: Audio PLL 10: External clock 11: reserved
7:6	PDM1_CLK_SEL	R/W	2'b00	PDM1 时钟源选择: 00: CORE_HSCLK 01: Audio PLL 10: External clock 11: reserved
5:4	PDM0_CLK_SEL	R/W	2'b00	PDM0 时钟源选择: 00: CORE_HSCLK 01: Audio PLL 10: External clock 11: reserved
3:2	I2S_CLK_SEL	R/W	2'b00	I2S 时钟源选择: 00: CORE_HSCLK 01: Audio PLL 10: External clock 11: reserved
1:0	SPDIF_CLK_SEL	R/W	2'b00	SPDIF 时钟源选择: 00: CORE_HSCLK 01: Audio PLL 10: External clock 11: reserved

2.5.6. BLOCK Clock Divider0 (Offset 0x10)

比特	名称	属性	复位值	描述
31:28	MSPI1_CLK_DIV	R/W	4'b0	SPIMX8_1 参考时钟分频 $spimx8_1_clk = source_clk / (DIV + 1)$
27:24	MSPI0_CLK_DIV	R/W	4'b0	SPIMX8_0 参考时钟分频 $spimx8_0_clk = source_clk / (DIV + 1)$
23:20	SSIS_CLK_DIV	R/W	4'b0	SPIS 参考时钟分频 $spis_clk = source_clk / (DIV + 1)$
19:16	OSPI_CLK_DIV	R/W	4'b0	OSPI 参考时钟分频 $ospi_clk = source_clk / (DIV + 1)$
15:12	QSPI1_CLK_DIV	R/W	4'b0	QSPI1 参考时钟分频 $qspi1_clk = source_clk / (DIV + 1)$
11:8	QSPI0_CLK_DIV	R/W	4'b0	QSPI0 参考时钟分频 $qspi0_clk = source_clk / (DIV + 1)$
7:4	SDIOH1_CLK_DIV	R/W	4'b0	SDIOH1 参考时钟分频 $sdioh_clk = source_clk / (DIV + 1)$
3:0	SDIOH0_CLK_DIV	R/W	4'b0	SDIOH0 参考时钟分频 $sdioh_clk = source_clk / (DIV + 1)$

2.5.7. BLOCK Clock Divider1 (Offset 0x14)

比特	名称	属性	复位值	描述
31:28	SSIM_CLK_DIV	R/W	4'b0	SPIM 参考时钟分频 spim_clk=source_clk/(DIV+1)
27:24	FFT_CLK_DIV	R/W	4'b0	FFT 参考时钟分频 fft_clk=source_clk/(DIV+1)
23:20	TRIFUNC_CLK_DIV	R/W	'd0	TRIFUNC 参考时钟分频 trifunc_clk=source_clk/(DIV+1)
19:16	IIRFLT_CLK_DIV	R/W	4'b0	IIRFLT 参考时钟分频 iirflt_clk=source_clk/(DIV+1)
15:12	I2C_CLK_DIV	R/W	4'b0	I2C 参考时钟分频 i2c_clk=source_clk/(DIV+1)
11:8	UART_CLK_DIV	R/W	4'b0	UART 参考时钟分频 uart_clk=source_clk/(DIV+1)
7:4	GPIO_CLK_DIV	R/W	4'b0	GPIO 参考时钟分频 gpio_clk=source_clk/(DIV+1)
3:0	PARALLEL_CLK_DIV	R/W	4'b0	PARALLEL 参考时钟分频 parallel_clk=source_clk/(DIV+1)

2.5.8. BLOCK Clock Divider2 (Offset 0x18)

比特	名称	属性	复位值	描述
31:24	Reserved	R/W	'd0	Reserved
27:24	TIMER2/3_CLK_DIV	R/W	4'b0	TIMER2/3 参考时钟分频 timer1_clk=source_clk/(DIV+1)
23:20	TIMER0/1_CLK_DIV	R/W	4'b0	TIMER0/1 参考时钟分频 timer0_clk=source_clk/(DIV+1)
19:16	USB_CLK_DIV	R/W	4'b0	USB 参考时钟分频 usb_clk=source_clk/(DIV+1)
15:12	FRSPIM_CLK_DIV	R/W	4'b0	FRSPIM 参考时钟分频 prspim_clk=source_clk/(DIV+1)
11:8	PWM_CLK_DIV	R/W	4'b0	PWM 参考时钟分频

				pwm_clk=source_clk/(DIV+1)
7:4	MCAN_CLK_DIV	R/W	4'b0	MACN 参考时钟分频 mcan_clk=source_clk/(DIV+1)
3:0	BLEND_CLK_DIV	R/W	4'b0	BLEND 参考时钟 blend_clk=source_clk/(DIV+1)

2.5.9. AUDIO Clock Divider (Offset 0x1C)

比特	名称	属性	复位值	描述
31:24	Reserved	R/W	'd0	Reserved
23:20	SBC_CLK_DIV	R/W	4'b0	SBC Master 时钟分频 sbc_clk=source_clk/(DIV+1)
19:16	PDM2_CLK_DIV	R/W	4'b0	PDM2 Master 时钟分频 pdm2_clk=source_clk/(DIV+1)
15:12	PDM1_CLK_DIV	R/W	4'b0	PDM1 Master 时钟分频 pdm1_clk=source_clk/(DIV+1)
11:8	PDM0_CLK_DIV	R/W	4'b0	PDM0 Master 时钟分频 pdm0_clk=source_clk/(DIV+1)
7:4	I2S_CLK_DIV	R/W	4'b0	I2S Master 时钟分频 i2s_clk=source_clk/(DIV+1)
3:0	SPDIF_CLK_DIV	R/W	4'b0	SPDIF Master 时钟分频 spdif_clk=source_clk/(DIV+1)

2.5.10. BUS_MEM Clock Enable (Offset 0x20)

比特	名称	属性	复位值	描述
31:17	Reserved	R/W	'd0	Reserved
16	OSPI_P_CLKEN	R/W	1'b0	OSPI_P 时钟使能
15	QSPI1_P_CLKEN	R/W	1'b0	QSPI1_P 时钟使能
14	QSPI0_P_CLKEN	R/W	1'b0	QSPI0_P 时钟使能
13	LIB_ROM_CLKEN	R/W	1'b1	LIB_ROM 时钟使能
12	BOOT_ROM_CLKEN	R/W	1'b1	BOOT_ROM 时钟使能
11	SRAM2_CLKEN	R/W	1'b1	SRAM2 时钟使能
10	SRAM1_CLKEN	R/W	1'b1	SRAM1 时钟使能

9	SRAM0_CLKEN	R/W	1'b1	SRAM0 时钟使能
8	CRAM1_CLKEN	R/W	1'b1	CRAM1 时钟使能
7	CRAM0_CLKEN	R/W	1'b1	CRAM0 时钟使能
6	APB2_CLKEN	R/W	1'b0	APB2 时钟使能
5	APB1_CLKEN	R/W	1'b0	APB1 时钟使能
4	APB0_CLKEN	R/W	1'b0	APB0 时钟使能
3	APB_CLKEN	R/W	1'b0	APB 时钟使能
2	SYS_BUS_GCLKEN	R/W	1'b0	SYS_BUS 时钟使能
1	CODE_BUS_CLKEN	R/W	1'b0	CODE_BUS 时钟使能
0	CM33_MAS_CLKEN	R/W	1'b0	CM33_MAS 时钟使能

2.5.11. CACHE_DMACHEN Clock Enable (Offset 0x24)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	'd0	Reserved
3	DMAC1_CLKEN	R/W	1'b0	DMAC1 时钟使能
2	DMAC0_CLKEN	R/W	1'b0	DMAC0 时钟使能
1	AHBC_CLKEN	R/W	1'b0	AHBC 时钟使能
0	PFC_CLKEN	R/W	1'b0	PFC 时钟使能

2.5.12. SDIO_CAN Clock Enable (Offset 0x28)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	'd0	Reserved
3	MCAN1_GCLKEN	R/W	1'b0	MCAN1 时钟使能
2	MCAN0_GCLKEN	R/W	1'b0	MCAN0 时钟使能
1	SDIOH1_GCLKEN	R/W	1'b0	SDIOH1 时钟使能
0	SDIOH0_GCLKEN	R/W	1'b0	SDIOH0 时钟使能

2.5.13. TIMER_WDT Clock Enable (Offset 0x2C)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	'd0	Reserved
4	WDT0_CLKEN	R/W	1'b0	APP_WDT 时钟使能
3	TIMER11_CLKEN	R/W	1'b0	APP_TIMER3 时钟使能
2	TIMER10_CLKEN	R/W	1'b0	APP_TIMER2 时钟使能
1	TIMER01_CLKEN	R/W	1'b0	APP_TIMER1 时钟使能
0	TIMER00_CLKEN	R/W	1'b0	APP_TIMER0 时钟使能

2.5.14. GPIO_PWM Clock Enable (Offset 0x30)

比特	名称	属性	复位值	描述
31:6	Reserved	R/W	'd0	Reserved
5	PWM1_CLKEN	R/W	1'b0	PWM1 时钟使能
4	PWM0_CLKEN	R/W	1'b0	PWM0 时钟使能
3	GPIOD_CLKEN	R/W	1'b0	GPIOD 时钟使能
2	GPIOC_CLKEN	R/W	1'b0	GPIOC 时钟使能
1	GPIOB_CLKEN	R/W	1'b0	GPIOB 时钟使能
0	GPIOA_CLKEN	R/W	1'b0	GPIOA 时钟使能

2.5.15. UART_I2C Clock Enable (Offset 0x34)

比特	名称	属性	复位值	描述
31:12	Reserved	R/W	'd0	Reserved
11	I2C5_CLKEN	R/W	1'b0	I2C5 时钟使能
10	I2C4_CLKEN	R/W	1'b0	I2C4 时钟使能
9	I2C3_CLKEN	R/W	1'b0	I2C3 时钟使能
8	I2C2_CLKEN	R/W	1'b0	I2C2 时钟使能
7	I2C1_CLKEN	R/W	1'b0	I2C1 时钟使能
6	I2C0_CLKEN	R/W	1'b0	I2C0 时钟使能
5	UART5_CLKEN	R/W	1'b0	UART5 时钟使能
4	UART4_CLKEN	R/W	1'b0	UART4 时钟使能
3	UART3_CLKEN	R/W	1'b0	UART3 时钟使能

2	UART2_CLKEN	R/W	1'b0	UART2 时钟使能
1	UART1_CLKEN	R/W	1'b0	UART1 时钟使能
0	UART0_CLKEN	R/W	1'b0	UART0 时钟使能

2.5.16. SPI Clock Enable (Offset 0x38)

比特	名称	属性	复位值	描述
31:12	Reserved	R/W	'd0	Reserved
12	OSPI_H_CLKEN	R/W	1'b0	OSPI_H 时钟使能
11	OSPI_CLKEN	R/W	1'b0	OSPI 时钟使能
10	QSPI1_H_CLKEN	R/W	1'b0	QSPI1_H 时钟使能
9	QSPI1_CLKEN	R/W	1'b0	QSPI1 时钟使能
8	QSPI0_H_CLKEN	R/W	1'b0	QSPI0_H 时钟使能
7	QSPI0_CLKEN	R/W	1'b0	QSPI0 时钟使能
6	MSPI1_CLKEN	R/W	1'b0	SPIMX8_1 时钟使能
5	MSPI0_CLKEN	R/W	1'b0	SPIMX8_0 时钟使能
4	SSIS1_CLKEN	R/W	1'b0	SPIS1 时钟使能
3	SSIS0_CLKEN	R/W	1'b0	SPIS0 时钟使能
2	SSIM2_CLKEN	R/W	1'b0	SPIM2 时钟使能
1	SSIM1_CLKEN	R/W	1'b0	SPIM1 时钟使能
0	SSIM0_CLKEN	R/W	1'b0	SPIM0 时钟使能

2.5.17. Algorithm Clock Enable (Offset 0x3C)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	'd0	Reserved
10	YUV2RGB_CLKEN	R/W	1'b0	YUV2RGB 时钟使能
9	BLEND_CLKEN	R/W	1'b0	BLEND 时钟使能
8	CRC_CLKEN	R/W	1'b0	CRC 时钟使能
7	TRNG_CLKEN	R/W	1'b0	TRNG 时钟使能
6	FFT_CLKEN	R/W	1'b0	FFT 时钟使能
5	TRIFUNC_CLKEN	R/W	1'b0	TRIFUNC 时钟使能
4	IIRFLT_CLKEN	R/W	1'b0	IIRFLT 时钟使能
3	AES_CLKEN	R/W	1'b0	AES 时钟使能

2	RSA_CLKEN	R/W	1'b0	RSA 时钟使能
1	SHA_CLKEN	R/W	1'b0	SHA 时钟使能
0	SEC_CLKEN	R/W	1'b0	SEC 时钟使能

2.5.18. AUDIO Device Clock Enable (Offset 0x40)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	'd0	Reserved
10	MP3D_CLKEN	R/W	1'b0	MP3D 时钟使能
9	SBCENC_HCLKEN	R/W	1'b0	SBCENC 时钟使能
8	SBCDEC_CLKEN	R/W	1'b0	SBCENC 时钟使能
7	CODEC_CLKEN	R/W	1'b0	CODEC 时钟使能
6	PDM2_CLKEN	R/W	1'b0	PDM2 时钟使能
5	PDM1_CLKEN	R/W	1'b0	PDM1 时钟使能
4	PDM0_CLKEN	R/W	1'b0	PDM0 时钟使能
3	I2S2_CLKEN	R/W	1'b0	I2S2 时钟使能
2	I2S1_CLKEN	R/W	1'b0	I2S1 时钟使能
1	I2S0_CLKEN	R/W	1'b0	I2S0 时钟使能
0	SPDIF_CLKEN	R/W	1'b0	SPDIF 时钟使能

2.5.19. MISC Clock Enable (Offset 0x44)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9	Free_Counter_CLKEN	R/W	1'b0	Free_Counter 时钟使能
8	ADC_CLKEN	R/W	1'b0	ADC 时钟使能
7	EFUSE1_CLKEN	R/W	1'b0	EFUSE1 时钟使能
6	EFUSE0_CLKEN	R/W	1'b0	EFUSE0 时钟使能
5	CALI_CLKEN	R/W	1'b0	CALI 时钟使能
4	FRSPIM_CLKEN	R/W	1'b0	FRSPIM 时钟使能
3	APP_IPC_CLKEN	R/W	1'b0	APP_IPC 时钟使能
2	TICK_CLKEN	R/W	1'b0	TICK 时钟使能
1	PARALLEL_CLKEN	R/W	1'b0	PARALLEL 时钟使能
0	USB_CLKEN	R/W	1'b0	USB 时钟使能

注：时钟使能寄存器所有 bit。写 ‘0’：Disable，写 ‘1’：Enable。

2.5.20. CPU_BUS Soft Reset (Offset 0x48)

比特	名称	属性	复位值	描述
31:17	Reserved	R/W	'd0	Reserved
16	OSPI_P_SFT_RST	R/W	1'b0	OSPI_P 复位 写 ‘1’ 复位，硬件自动清零。
15	QSPI1_P_SFT_RST	R/W	1'b0	QSPI1_P 复位 写 ‘1’ 复位，硬件自动清零。
14	QSPI0_P_SFT_RST	R/W	1'b0	QSPI0_P 复位 写 ‘1’ 复位，硬件自动清零。
13:7	Reserved	R/W	'd0	Reserved
6	APB2_SFT_RST	R/W	1'b0	APB2 复位 写 ‘1’ 复位，硬件自动清零。
5	APB1_SFT_RST	R/W	1'b0	APB1 复位 写 ‘1’ 复位，硬件自动清零。
4	APB0_SFT_RST	R/W	1'b0	APB0 复位 写 ‘1’ 复位，硬件自动清零。
3	APB_SFT_RST	R/W	1'b0	APB 复位 写 ‘1’ 复位，硬件自动清零。
2:1	Reserved	R/W	'd0	Reserved
0	CM33_MAS_SFT_RST	R/W	1'b0	CM33_MAS 复位 写 ‘1’ 复位，硬件自动清零。

2.5.21. CACHE_DMACH Soft Reset (Offset 0x4C)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	'd0	Reserved
3	DMAC1_SFT_RST	R/W	1'b0	DMAC1 复位 写 ‘1’ 复位，硬件自动清零。
2	DMAC0_SFT_RST	R/W	1'b0	DMAC0 复位 写 ‘1’ 复位，硬件自动清零。
1	AHBC_SFT_RST	R/W	1'b0	AHBC 复位

				写 ‘1’ 复位，硬件自动清零。
0	PFC_SFT_RST	R/W	1'b0	PFC 复位 写 ‘1’ 复位，硬件自动清零。

2.5.22. SDIO_CAN Soft Reset (Offset 0x50)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	'd0	Reserved
3	MCAN1_SFT_RST	R/W	1'b0	DMAC1 复位 写 ‘1’ 复位，硬件自动清零。
2	MCAN0_SFT_RST	R/W	1'b0	DMAC0 复位 写 ‘1’ 复位，硬件自动清零。
1	SDIOH1_SFT_RST	R/W	1'b0	SDIOH1 复位 写 ‘1’ 复位，硬件自动清零。
0	SDIOH0_SFT_RST	R/W	1'b0	SDIOH0 复位 写 ‘1’ 复位，硬件自动清零。

2.5.23. TIMER_WDT Soft Reset (Offset 0x54)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	'd0	Reserved
3	TIMER3_SFT_RST	R/W	1'b0	APP_TIMER3 复位 写 ‘1’ 复位，硬件自动清零。
2	TIMER2_SFT_RST	R/W	1'b0	APP_TIMER2 复位 写 ‘1’ 复位，硬件自动清零。
1	TIMER1_SFT_RST	R/W	1'b0	APP_TIMER1 复位 写 ‘1’ 复位，硬件自动清零。
0	TIMER0_SFT_RST		1'b0	APP_TIMER0 复位 写 ‘1’ 复位，硬件自动清零。

2.5.24. GPIO_PWM Soft Reset (Offset 0x58)

比特	名称	属性	复位值	描述
31:6	Reserved	R/W	'd0	Reserved
5	PWM1_SFT_RST	R/W	1'b0	PWM1 复位 写 '1' 复位, 硬件自动清零。
4	PWM0_SFT_RST	R/W	1'b0	PWM0 复位 写 '1' 复位, 硬件自动清零。
3	GPIOD_SFT_RST	R/W	1'b0	GPIOD 复位 写 '1' 复位, 硬件自动清零。
2	GPIOC_SFT_RST	R/W	1'b0	GPIOC 复位 写 '1' 复位, 硬件自动清零。
1	GPIOB_SFT_RST	R/W	1'b0	GPIOB 复位 写 '1' 复位, 硬件自动清零。
0	GPIOA_SFT_RST	R/W	1'b0	GPIOA 复位 写 '1' 复位, 硬件自动清零。

2.5.25. UART_I2C Soft Reset (Offset 0x5C)

比特	名称	属性	复位值	描述
31:12	Reserved	R/W	'd0	Reserved
11	I2C5_SFT_RST	R/W	1'b0	I2C5 复位 写 '1' 复位, 硬件自动清零。
10	I2C4_SFT_RST	R/W	1'b0	I2C4 复位 写 '1' 复位, 硬件自动清零。
9	I2C3_SFT_RST	R/W	1'b0	I2C3 复位 写 '1' 复位, 硬件自动清零。
8	I2C2_SFT_RST	R/W	1'b0	I2C2 复位 写 '1' 复位, 硬件自动清零。
7	I2C1_SFT_RST	R/W	1'b0	I2C1 复位 写 '1' 复位, 硬件自动清零。
6	I2C0_SFT_RST	R/W	1'b0	I2C0 复位 写 '1' 复位, 硬件自动清零。
5	UART5_SFT_RST	R/W	1'b0	UART5 复位

				写 ‘1’ 复位，硬件自动清零。
4	UART4_SFT_RST	R/W	1'b0	UART4 复位 写 ‘1’ 复位，硬件自动清零。
3	UART3_SFT_RST	R/W	1'b0	UART3 复位 写 ‘1’ 复位，硬件自动清零。
2	UART2_SFT_RST	R/W	1'b0	UART2 复位 写 ‘1’ 复位，硬件自动清零。
1	UART1_SFT_RST	R/W	1'b0	UART1 复位 写 ‘1’ 复位，硬件自动清零。
0	UART0_SFT_RST	R/W	1'b0	UART0 复位 写 ‘1’ 复位，硬件自动清零。

2.5.26. SPI Soft Reset (Offset 0x60)

比特	名称	属性	复位值	描述
31:13	Reserved	R/W	'd0	Reserved
12	OSPI_REF_SFT_RST	R/W	1'b0	OSPI_REF 复位 写 ‘1’ 复位，硬件自动清零。
11	OSPI_SFT_RST	R/W	1'b0	OSPI 复位 写 ‘1’ 复位，硬件自动清零。
10	QSPI1_REF_SFT_RST	R/W	1'b0	QSPI1_REF 复位 写 ‘1’ 复位，硬件自动清零。
9	QSPI1_SFT_RST	R/W	1'b0	QSPI1 复位 写 ‘1’ 复位，硬件自动清零。
8	QSPI0_REF_SFT_RST	R/W	1'b0	QSPI0_REF 复位 写 ‘1’ 复位，硬件自动清零。
7	QSPI0_SFT_RST	R/W	1'b0	QSPI0 复位 写 ‘1’ 复位，硬件自动清零。
6	MSPI1_SFT_RST	R/W	1'b0	SPIMX8_1 复位 写 ‘1’ 复位，硬件自动清零。
5	MSPI0_SFT_RST	R/W	1'b0	SPIMX8_0 复位 写 ‘1’ 复位，硬件自动清零。
4	SSIS1_SFT_RST	R/W	1'b0	SPIS1 复位 写 ‘1’ 复位，硬件自动清零。

3	SSIS0_SFT_RST	R/W	1'b0	SPIS0 复位 写 '1' 复位，硬件自动清零。
2	SSIM2_SFT_RST	R/W	1'b0	SPIM2 复位 写 '1' 复位，硬件自动清零。
1	SSIM1_SFT_RST	R/W	1'b0	SPIM1 复位 写 '1' 复位，硬件自动清零。
0	SSIM0_SFT_RST	R/W	1'b0	SPIM0 复位 写 '1' 复位，硬件自动清零。

2.5.27. Algorithm Soft Reset (Offset 0x64)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	'd0	Reserved
10	YUV2RGB_SFT_RST	R/W	1'b0	YUV2RGB 复位 写 '1' 复位，硬件自动清零。
9	BLEND_SFT_RST	R/W	1'b0	BLEND 复位 写 '1' 复位，硬件自动清零。
8	CRC_SFT_RST	R/W	1'b0	CRC 复位 写 '1' 复位，硬件自动清零。
7	TRNG_SFT_RST	R/W	1'b0	GPIOD 复位 写 '1' 复位，硬件自动清零。
6	FFT_SFT_RST	R/W	1'b0	FFT 复位 写 '1' 复位，硬件自动清零。
5	TRIFUNC_SFT_RST	R/W	1'b0	TRIFUNC 复位 写 '1' 复位，硬件自动清零。
4	IIRFLT_SFT_RST	R/W	1'b0	IIRFLT 复位 写 '1' 复位，硬件自动清零。
3	AES_SFT_RST	R/W	1'b0	AES 复位 写 '1' 复位，硬件自动清零。
2	RSA_SFT_RST	R/W	1'b0	RSA 复位 写 '1' 复位，硬件自动清零。
1	SHA_SFT_RST	R/W	1'b0	SHA 复位 写 '1' 复位，硬件自动清零。
0	SEC_SFT_RST	R/W	1'b0	SEC 复位

				写 '1' 复位，硬件自动清零。
--	--	--	--	------------------

2.5.28. AUDIO Device Soft Reset (Offset 0x68)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9	SBCENC_SFT_RST	R/W	1'b0	SBCENC 复位 写 '1' 复位，硬件自动清零。
8	SBCDEC_SFT_RST	R/W	1'b0	SBCDEC 复位 写 '1' 复位，硬件自动清零。
7	CODEC_SFT_RST	R/W	1'b0	CODEC 复位 写 '1' 复位，硬件自动清零。
6	PDM2_SFT_RST	R/W	1'b0	PDM2 复位 写 '1' 复位，硬件自动清零。
5	PDM1_SFT_RST	R/W	1'b0	PDM1 复位 写 '1' 复位，硬件自动清零。
4	PDM0_SFT_RST	R/W	1'b0	PDM0 复位 写 '1' 复位，硬件自动清零。
3	I2S2_SFT_RST	R/W	1'b0	I2S2 复位 写 '1' 复位，硬件自动清零。
2	I2S1_SFT_RST	R/W	1'b0	I2S1 复位 写 '1' 复位，硬件自动清零。
1	I2S0_SFT_RST	R/W	1'b0	I2S0 复位 写 '1' 复位，硬件自动清零。
0	SPDIF_SFT_RST	R/W	1'b0	SPDIF 复位 写 '1' 复位，硬件自动清零。

2.5.29. MISC Soft Reset (Offset 0x6C)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9	Free_Counter_SFT_RST	R/W	1'b0	Free_Counter 复位 写 '1' 复位, 硬件自动清零。
8	ADC_SFT_RST	R/W	1'b0	ADC 复位 写 '1' 复位, 硬件自动清零。
7	EFUSE1_SFT_RST	R/W	1'b0	EFUSE1 复位 写 '1' 复位, 硬件自动清零。
6	EFUSE0_SFT_RST	R/W	1'b0	EFUSE0 复位 写 '1' 复位, 硬件自动清零。
5	CALI_SFT_RST	R/W	1'b0	CALI 复位 写 '1' 复位, 硬件自动清零。
4	FRSPIM_SFT_RST	R/W	1'b0	FRSPIM 复位 写 '1' 复位, 硬件自动清零。
3	APP_IPC_SFT_RST	R/W	1'b0	APP_IPC 复位 写 '1' 复位, 硬件自动清零。
2	TICK_SFT_RST	R/W	1'b0	TICK 复位 写 '1' 复位, 硬件自动清零。
1	PARALLEL_SFT_RST	R/W	1'b0	PARALLEL 复位 写 '1' 复位, 硬件自动清零。
0	USB_SFT_RST	R/W	1'b0	USB 复位 写 '1' 复位, 硬件自动清零。

2.5.30. PortA Input Enable (Offset 0x80)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortA_IE	R/W	16'b0	PortA 端口输入使能 (高电平有效) Bit[0]对应 PortA0, 1bit 对应 1 引脚

2.5.31. PortB Input Enable (Offset 0x84)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortB_IE	R/W	16'b0	PortB 端口输入使能（高电平有效） Bit[0]对应 PortB0, 1bit 对应 1 引脚

2.5.32. PortC Input Enable (Offset 0x88)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortC_IE	R/W	16'b0	PortC 端口输入使能（高电平有效） Bit[0]对应 PortC0, 1bit 对应 1 引脚

2.5.33. PortD Input Enable (Offset 0x8C)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortD_IE	R/W	16'b0	PortD 端口输入使能（高电平有效） Bit[0]对应 PortD0, 1bit 对应 1 引脚

2.5.34. PortA Pull Enable (Offset 0xA0)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortA_PE	R/W	16'b0	PortA 上下拉使能（高电平有效） Bit[0]对应 PortA0, 1bit 对应 1 引脚

2.5.35. PortB Pull Enable (Offset 0xA4)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortB_PE	R/W	16'b0	PortB 上下拉使能（高电平有效） Bit[0]对应 PortB0，1bit 对应 1 引脚

2.5.36. PortC Pull Enable (Offset 0xA8)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortC_PE	R/W	16'b0	PortC 上下拉使能（高电平有效） Bit[0]对应 PortC0，1bit 对应 1 引脚

2.5.37. PortD Pull Enable (Offset 0xAC)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortD_PE	R/W	16'b0	PortD 上下拉使能（高电平有效） Bit[0]对应 PortD0，1bit 对应 1 引脚

2.5.38. PortA Pull Select (Offset 0xC0)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortA_PS	R/W	16'b0	PortA 上下拉选择 0: 下拉 1: 上拉 Bit[0]对应 PortA0，1bit 对应 1 引脚

2.5.39. PortB Pull Select (Offset 0xC4)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortB_PS	R/W	16'b0	PortB 上下拉选择 0: 下拉 1: 上拉 Bit[0]对应 PortB0, 1bit 对应 1 引脚

2.5.40. PortC Pull Select (Offset 0xC8)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortC_PS	R/W	16'b0	PortC 上下拉选择 0: 下拉 1: 上拉 Bit[0]对应 PortC0, 1bit 对应 1 引脚

2.5.41. PortD Pull Select (Offset 0xCC)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortD_PS	R/W	16'b0	PortD 上下拉选择 0: 下拉 1: 上拉 Bit[0]对应 PortD0, 1bit 对应 1 引脚

2.5.42. PORTA Pull Resistance Configure (Offset 0xE0)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortA_PR	R/W	16'b0	PortA0~ PortA15 上下拉电阻选择 0: 30K 1: 60K

				Bit[0]对应 PortA0, 1bit 对应 1 引脚
--	--	--	--	-------------------------------

2.5.43. PORTB Pull Resistance Configure (Offset 0xE4)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:12	PortB_PR_1	R/W	4'b0	PortB12~ PortB15 上下拉电阻选择 0: 50K 1: 100K Bit[12]对应 PortB12, 1bit 对应 1 引脚
11:0	PortB_PR_0	R/W	12'b0	PortB0~ PortB11 上下拉电阻选择 1bit 对应 1Pin 0: 30K 1: 60K Bit[0]对应 PortB0, 1bit 对应 1 引脚

2.5.44. PORTC Pull Resistance Configure (Offset 0xE8)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortC_PR	R/W	16'b0	PortC0~ PortC15 上下拉电阻选择 0: 50K 1: 100K Bit[0]对应 PortC0, 1bit 对应 1 引脚

2.5.45. PORTD Pull Resistance Configure (Offset 0xEC)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	PortD_PR	R/W	16'b0	PortD0~ PortD15 上下拉电阻选择 0: 30K 1: 60K Bit[0]对应 PortD0, 1bit 对应 1 引脚

2.5.46. PORTA Drive configure (Offset 0x100)

比特	名称	属性	复位值	描述
31:0	PortA_DS	R/W	2'b00	PA0~PA15 上拉电阻选择 00: 8mA 01: 12mA 10: 16mA 11: 24mA Bit[1]和 Bit[0]对应 PortA0, 2bit 对应 1 引脚

2.5.47. PORTB Drive configure (Offset 0x104)

比特	名称	属性	复位值	描述
31:0	PortB_DS	R/W	2'b00	PB0~PB15 驱动能力选择 00: 8mA 01: 12mA 10: 16mA 11: 24mA Bit[1]和 Bit[0]对应 PortB0, 2bit 对应 1 引脚

2.5.48. PORTC Drive configure (Offset 0x108)

比特	名称	属性	复位值	描述
31:0	PortC_DS	R/W	2'b00	PC0~PC15 驱动能力选择 00: 8mA 01: 12mA 10: 16mA 11: 24mA Bit[1]和 Bit[0]对应 PortC0, 2bit 对应 1 引脚

2.5.49. PORTD Drive configure (Offset 0x10C)

比特	名称	属性	复位值	描述
31:0	PortD_DS	R/W	2'b00	PD0~PD15 驱动能力选择

				00: 8mA 01: 12mA 10: 16mA 11: 24mA Bit[1]和 Bit[0]对应 PortD0, 2bit 对应 1 引脚
--	--	--	--	--

2.5.50. PortA Function MUX_L (Offset 0x120)

比特	名称	属性	复位值	描述
31:28	PA7_MUX	R/W	4'h0	PA7 复用功能配置。配置值: 0x0 ~ 0xF
27:24	PA6_MUX	R/W	4'h0	PA6 复用功能配置。配置值: 0x0 ~ 0xF
23:20	PA5_MUX	R/W	4'h0	PA5 复用功能配置。配置值: 0x0 ~ 0xF
19:16	PA4_MUX	R/W	4'h0	PA4 复用功能配置。配置值: 0x0 ~ 0xF
15:12	PA3_MUX	R/W	4'h0	PA3 复用功能配置。配置值: 0x0 ~ 0xF
11:8	PA2_MUX	R/W	4'h0	PA2 复用功能配置。配置值: 0x0 ~ 0xF
7:4	PA1_MUX	R/W	4'h0	PA1 复用功能配置。配置值: 0x0 ~ 0xF
3:0	PA0_MUX	R/W	4'h0	PA0 复用功能配置。配置值: 0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.51. PortA Function MUX_H (Offset 0x124)

比特	名称	属性	复位值	描述
31:28	PA15_MUX	R/W	4'h0	PA15 复用功能配置。配置值: 0x0 ~ 0xF
27:24	PA14_MUX	R/W	4'h0	PA14 复用功能配置。配置值: 0x0 ~ 0xF
23:20	PA13_MUX	R/W	4'h0	PA13 复用功能配置。配置值: 0x0 ~ 0xF
19:16	PA12_MUX	R/W	4'h0	PA12 复用功能配置。配置值: 0x0 ~ 0xF
15:12	PA11_MUX	R/W	4'h0	PA11 复用功能配置。配置值: 0x0 ~ 0xF
11:8	PA10_MUX	R/W	4'h0	PA10 复用功能配置。配置值: 0x0 ~ 0xF
7:4	PA9_MUX	R/W	4'h0	PA9 复用功能配置。配置值: 0x0 ~ 0xF
3:0	PA8_MUX	R/W	4'h0	PA8 复用功能配置。配置值: 0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.52. PortB Function MUX_L (Offset 0x128)

比特	名称	属性	复位值	描述
31:28	PB7_MUX	R/W	4'h0	PB7 复用功能配置。配置值：0x0 ~ 0xF
27:24	PB6_MUX	R/W	4'h0	PB6 复用功能配置。配置值：0x0 ~ 0xF
23:20	PB5_MUX	R/W	4'h0	PB5 复用功能配置。配置值：0x0 ~ 0xF
19:16	PB4_MUX	R/W	4'h0	PB4 复用功能配置。配置值：0x0 ~ 0xF
15:12	PB3_MUX	R/W	4'h0	PB3 复用功能配置。配置值：0x0 ~ 0xF
11:8	PB2_MUX	R/W	4'h0	PB2 复用功能配置。配置值：0x0 ~ 0xF
7:4	PB1_MUX	R/W	4'h0	PB1 复用功能配置。配置值：0x0 ~ 0xF
3:0	PB0_MUX	R/W	4'h0	PB0 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.53. PortB Function MUX_H (Offset 0x12C)

比特	名称	属性	复位值	描述
31:28	PB15_MUX	R/W	4'h0	PB15 复用功能配置。配置值：0x0 ~ 0xF
27:24	PB14_MUX	R/W	4'h0	PB14 复用功能配置。配置值：0x0 ~ 0xF
23:20	PB13_MUX	R/W	4'h0	PB13 复用功能配置。配置值：0x0 ~ 0xF
19:16	PB12_MUX	R/W	4'h0	PB12 复用功能配置。配置值：0x0 ~ 0xF
15:12	PB11_MUX	R/W	4'h0	PB11 复用功能配置。配置值：0x0 ~ 0xF
11:8	PB10_MUX	R/W	4'h0	PB10 复用功能配置。配置值：0x0 ~ 0xF
7:4	PB9_MUX	R/W	4'h0	PB9 复用功能配置。配置值：0x0 ~ 0xF
3:0	PB8_MUX	R/W	4'h0	PB8 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.54. PortC Function MUX_L (Offset 0x130)

比特	名称	属性	复位值	描述
31:28	PC7_MUX	R/W	4'h0	PC7 复用功能配置。配置值：0x0 ~ 0xF
27:24	PC6_MUX	R/W	4'h0	PC6 复用功能配置。配置值：0x0 ~ 0xF
23:20	PC5_MUX	R/W	4'h0	PC5 复用功能配置。配置值：0x0 ~ 0xF
19:16	PC4_MUX	R/W	4'h0	PC4 复用功能配置。配置值：0x0 ~ 0xF
15:12	PC3_MUX	R/W	4'h0	PC3 复用功能配置。配置值：0x0 ~ 0xF
11:8	PC2_MUX	R/W	4'h0	PC2 复用功能配置。配置值：0x0 ~ 0xF
7:4	PC1_MUX	R/W	4'h0	PC1 复用功能配置。配置值：0x0 ~ 0xF
3:0	PC0_MUX	R/W	4'h0	PC0 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.55. PortC Function MUX_H (Offset 0x134)

比特	名称	属性	复位值	描述
31:28	PC15_MUX	R/W	4'h0	PC15 复用功能配置。配置值：0x0 ~ 0xF
27:24	PC14_MUX	R/W	4'h0	PC14 复用功能配置。配置值：0x0 ~ 0xF
23:20	PC13_MUX	R/W	4'h0	PC13 复用功能配置。配置值：0x0 ~ 0xF
19:16	PC12_MUX	R/W	4'h0	PC12 复用功能配置。配置值：0x0 ~ 0xF
15:12	PC11_MUX	R/W	4'h0	PC11 复用功能配置。配置值：0x0 ~ 0xF
11:8	PC10_MUX	R/W	4'h0	PC10 复用功能配置。配置值：0x0 ~ 0xF
7:4	PC9_MUX	R/W	4'h0	PC9 复用功能配置。配置值：0x0 ~ 0xF
3:0	PC8_MUX	R/W	4'h0	PC8 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.56. PortD Function MUX_L (Offset 0x138)

比特	名称	属性	复位值	描述
31:28	PD7_MUX	R/W	4'h0	PD7 复用功能配置。配置值：0x0 ~ 0xF
27:24	PD6_MUX	R/W	4'h0	PD6 复用功能配置。配置值：0x0 ~ 0xF
23:20	PD5_MUX	R/W	4'h0	PD5 复用功能配置。配置值：0x0 ~ 0xF
19:16	PD4_MUX	R/W	4'h0	PD4 复用功能配置。配置值：0x0 ~ 0xF
15:12	PD3_MUX	R/W	4'h0	PD3 复用功能配置。配置值：0x0 ~ 0xF

11:8	PD2_MUX	R/W	4'h0	PD2 复用功能配置。配置值：0x0 ~ 0xF
7:4	PD1_MUX	R/W	4'h0	PD1 复用功能配置。配置值：0x0 ~ 0xF
3:0	PD0_MUX	R/W	4'h0	PD0 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.57. PortD Function MUX_H (Offset 0x13C)

比特	名称	属性	复位值	描述
31:28	PD15_MUX	R/W	4'h0	PD15 复用功能配置。配置值：0x0 ~ 0xF
27:24	PD14_MUX	R/W	4'h0	PD14 复用功能配置。配置值：0x0 ~ 0xF
23:20	PD13_MUX	R/W	4'h0	PD13 复用功能配置。配置值：0x0 ~ 0xF
19:16	PD12_MUX	R/W	4'h0	PD12 复用功能配置。配置值：0x0 ~ 0xF
15:12	PD11_MUX	R/W	4'h0	PD11 复用功能配置。配置值：0x0 ~ 0xF
11:8	PD10_MUX	R/W	4'h0	PD10 复用功能配置。配置值：0x0 ~ 0xF
7:4	PD9_MUX	R/W	4'h0	PD9 复用功能配置。配置值：0x0 ~ 0xF
3:0	PD8_MUX	R/W	4'h0	PD8 复用功能配置。配置值：0x0 ~ 0xF

具体功能详见[附录 I](#)。

2.5.58. PQSPI Input Enable (Offset 0x180)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_IE	R/W	10'b0	Port QSPI 端口输入使能（高电平有效）

2.5.59. PQSPI Pull Enable (Offset 0x184)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_PE	R/W	10'b0	Port QSPI 上下拉使能（高电平有效）

2.5.60. PQSPI Pull Select (Offset 0x188)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_PE	R/W	10'b0	Port QSPI 上下拉选择 (高电平有效) 0: 下拉 1: 上拉 1bit 对应 1 引脚

2.5.61. PQSPI Pull Resistance Configure (Offset 0x18C)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_PR	R/W	10'b0	Port QSPI 上下拉电阻选择 0: 30K 1: 60K 1bit 对应 1 引脚

2.5.62. PQSPI Drive configure (Offset 0x190)

比特	名称	属性	复位值	描述
31:20	Reserved	R/W	'd0	Reserved
19:0	PQSPI_PR	R/W	10'b0	Port QSPI IO 驱动能力选择 00: 8mA 01: 12mA 10: 16mA 11: 24mA 2bit 对应 1 引脚

2.5.63. PQSPI Function MUX (Offset 0x194)

比特	名称	属性	复位值	描述
31:20	Reserved	R/W	'd0	Reserved
19:0	PQSPI_MUX	R/W	20'b0	

2.5.64. POSPI Input Enable (Offset 0x198)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	POSPI_IE	R/W	10'b0	Port OSPI 端口输入使能 (高电平有效) 1bit 对应 1 引脚

2.5.65. POSPI Pull Enable (Offset 0x19C)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	POSPI_PE	R/W	10'b0	Port OSPI 上下拉使能 (高电平有效) 1bit 对应 1 引脚

2.5.66. POSPI Pull Select (Offset 0x1A0)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_PE	R/W	10'b0	Port QSPI 上下拉选择 (高电平有效) 0: 下拉 1: 上拉 1bit 对应 1 引脚

2.5.67. POSPI Pull Resistance Configure (Offset 0x1A4)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	POSPI_PR	R/W	10'b0	Port OSPI 上下拉电阻选择 0: 30K 1: 60K 1bit 对应 1 引脚

2.5.68. POSPI Drive configure (Offset 0x1A8)

比特	名称	属性	复位值	描述
31:20	Reserved	R/W	'd0	Reserved
19:0	POSPI_PR	R/W	10'b0	Port OSPI IO 驱动能力选择 00: 8mA 01: 12mA 10: 16mA 11: 24mA 2bit 对应 1 引脚

2.5.69. POSPI Function MUX (Offset 0x1AC)

比特	名称	属性	复位值	描述
31:20	Reserved	R/W	'd0	Reserved
19:0	POSPI_MUX	R/W	20'b0	

2.5.70. PQSPI Data (Offset 0x1B0)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_DATA	R/W	10'b0	PQSPI 输入输出数据寄存器 写 '1' 输出高, 写 '0' 输出低, 读为当前引脚状态

2.5.71. PQSPI Data Output Enable (Offset 0x1B4)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	PQSPI_DATA	R/W	10'h3FF	PQSPI 端口输出使能 (低电平有效)

2.5.72. POSPI Data (Offset 0x1B8)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	POSPI_DATA	R/W	10'b0	POSPI 输入输出数据寄存器 写 '1' 输出高, 写 '0' 输出低, 读为当前引脚状态

2.5.73. POSPI Data Output Enable (Offset 0x1BC)

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	'd0	Reserved
9:0	POSPI_DATA	R/W	10'h3FF	POSPI 端口输出使能 (低电平有效)

2.5.74. SDIOH_MISC (Offset 0x1D0)

比特	名称	属性	复位值	描述
31:7	Reserved	R/W	'd0	Reserved
6	SDIOH1_DAT_SWAP	R/W	1'b0	SDIOH1 数据线 SWAP 0: Disable 1: Enable 详情见 附录 I 如果此位为 1 例如 PortD12~PortD15 则对应 SDIOH1_DAT A0/1/2/3。
5:3	Reserved	R/W	'd0	Reserved
2	SDIOH0_DAT_SWAP	R/W	1'b0	SDIOH0 数据线 SWAP 0: Disable 1: Enable 详情见 附录 I 如果此位为 1 例如 PortB2~PortB5 则对应 SDIOH0_DATA 4/5/6/7。
1:0	Reserved	R/W	'd0	Reserved

2.5.75. MSPI DATA SWAP (Offset 0x1D4)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	'd0	Reserved
0	MSPIO_DAT_SWAP	R/W	1'b0	SPIMX8_0 数据线 SWAP 0: Disable 1: Enable 详情见 附录 I 如果此位为 1, PortA8~PortA15 对应 SPIMX8_0_IO_4/5/6/7/0/1/2/3。

2.5.76. USB MISC (Offset 0x1DC)

比特	名称	属性	复位值	描述
31:12	Reserved	R/W	'd0	Reserved
11	OTG_CID	R/W	1'b0	CID 配置
10	OTG_VBUSLO	R/W	1'b0	VBUSLO 配置
9	OTG_VBUSSES	R/W	1'b0	VBUS_SES 配置
8	OTG_VBUSVAL	R/W	1'b0	VBUS_VAL 配置
7:0	USB_PHY_ADAP_CFG	R/W	8'b0	USB PHY adapter 配置

2.5.77. DMA_REQ_MUX (Offset 0x2C0~0x2D8)

比特	名称	属性	复位值	描述
31:24	SPIMX8_0_RX	R/W	8'b0	SPIMX8_0_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	SPIMX8_0_TX	R/W	8'b0	SPIMX8_0_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	SPIMX8_1_RX	R/W	8'b0	SPIMX8_1_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	SPIMX8_1_TX	R/W	8'b0	SPIMX8_1_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。

Offset 0x2C4				
31:24	SPIM0_RX	R/W	8'b0	SPIM0_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	SPIM0_TX	R/W	8'b0	SPIM0_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	SPIM1_RX	R/W	8'b0	SPIM1_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	SPIM1_TX	R/W	8'b0	SPIM1_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2C8				
31:24	SPIM2_RX	R/W	8'b0	SPIM2_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	SPIM2_TX	R/W	8'b0	SPIM2_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	SPIS_RX	R/W	8'b0	SPIS_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	SPIS_TX	R/W	8'b0	SPIS_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2CC				
31:24	SPISX4_RX	R/W	8'b0	SPISX4_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	SPISX4_TX	R/W	8'b0	SPISX4_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	UART0_RX	R/W	8'b0	UART0_RX 请求分配。

				DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	UART0_TX	R/W	8'b0	UART0_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2B0				
31:24	UART1_RX	R/W	8'b0	UART1_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	UART1_TX	R/W	8'b0	UART1_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	UART2_RX	R/W	8'b0	UART2_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	UART2_TX	R/W	8'b0	UART2_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2B4				
31:24	UART3_RX	R/W	8'b0	UART3_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	UART3_TX	R/W	8'b0	UART3_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	UART4_RX	R/W	8'b0	UART4_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	UART4_TX	R/W	8'b0	UART4_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2B8				
31:24	UART5_RX	R/W	8'b0	UART5_RX 请求分配。

				DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	UART5_TX	R/W	8'b0	UART5_TX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	BLEND_RGB0	R/W	8'b0	BLEND_RGB0 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	BLEND_RGB1	R/W	8'b0	BLEND_RGB1 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2BC				
31:24	BLEND_MASK	R/W	8'b0	BLEND_MASK 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	BLEND_ORGB	R/W	8'b0	BLEND_ORGB 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	SPDIF_TRANS	R/W	8'b0	SPDIF_TRANS 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	PARALLEL_INTERFACE	R/W	8'b0	PARALLEL_INTERFACE 请求分 配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2C0				
31:24	I2S0_RX_LEFT	R/W	8'b0	I2S0_RX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	I2S0_TX_LEFT	R/W	8'b0	I2S0_TX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	I2S0_RX_RIGTH	R/W	8'b0	I2S0_RX_RIGTH 请求分配。

				DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	I2S0_TX_RIGTH	R/W	8'b0	I2S0_TX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2C4				
31:24	I2S1_RX_LEFT	R/W	8'b0	I2S1_RX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	I2S1_TX_LEFT	R/W	8'b0	I2S1_TX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	I2S1_RX_RIGTH	R/W	8'b0	I2S1_RX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	I2S1_TX_RIGTH	R/W	8'b0	I2S1_TX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2C8				
31:24	I2S2_RX_LEFT	R/W	8'b0	I2S2_RX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	I2S2_TX_LEFT	R/W	8'b0	I2S2_TX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	I2S2_RX_RIGTH	R/W	8'b0	I2S2_RX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	I2S2_TX_RIGTH	R/W	8'b0	I2S2_TX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2CC				
31:24	PDM0_RX	R/W	8'b0	PDM0_RX 请求分配。

				DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	PDM1_RX	R/W	8'b0	PDM1_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	PDM2_RX	R/W	8'b0	PDM2_RX 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	CODEC_RX_LEFT	R/W	8'b0	CODEC_RX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2D0				
31:24	CODEC_TX_LEFT	R/W	8'b0	CODEC_TX_LEFT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	CODEC_RX_RIGTH	R/W	8'b0	CODEC_RX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	CODEC_TX_RIGTH	R/W	8'b0	CODEC_TX_RIGTH 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	CODEC_RX_0	R/W	8'b0	CODEC_RX_0 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2D4				
31:24	CODEC_RX_1	R/W	8'b0	CODEC_RX_1 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
23:16	SBC_DEC_IN	R/W	8'b0	SBC_DEC_IN 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	SBC_DEC_OUT	R/W	8'b0	SBC_DEC_OUT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。

7:0	SBC_ENC_IN	R/W	8'b0	SBC_ENC_IN 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
Offset 0x2D8				
31:24	Reserved	R/W	'd0	Reserved
23:16	SBC_ENC_OUT	R/W	8'b0	SBC_ENC_OUT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
15:8	YUV2RGB_IN	R/W	8'b0	YUV2RGB_IN 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。
7:0	YUV2RGB_OUT	R/W	8'b0	YUV2RGB_OUT 请求分配。 DMA0 请求号 1~15 对应 1~ 15, DMA1 请求号 1~15 对应 17~31。

3. 通用输入输出接口（GPIO）

3.1. 概述

集成了通用数字 IO 的输入输出功能，同时在输入模式下，还支持外部中断检测处理（支持硬件防抖）。

GPIO 共有 4 组控制器，PortA、PortB、PortC、PortD。每个 GPIO 控制器都是完全独立的，没有互相共享任何资源。

3.2. 主要特性

- 4 组 GPIO 控制器
- 支持输入、输出模式，支持内部防抖

3.3. 结构框图

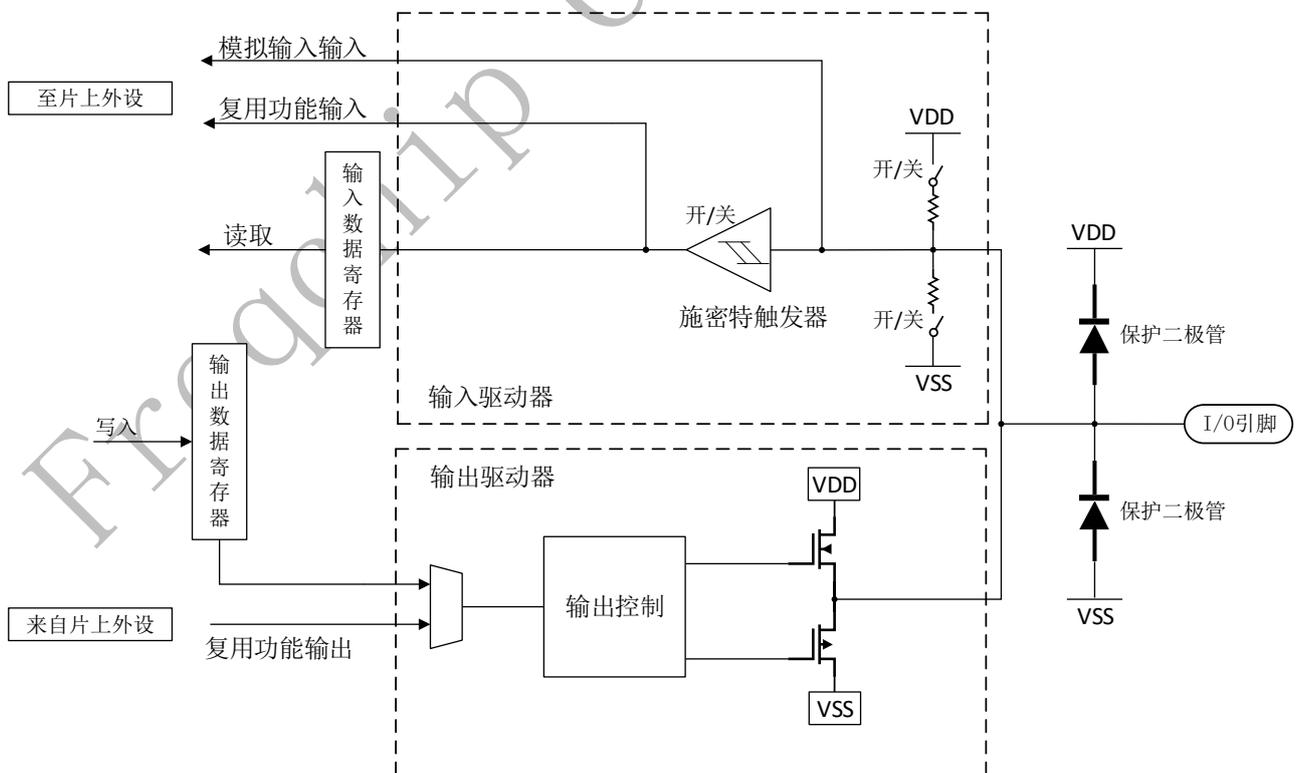


图 3-1 GPIO 结构框图

3.4. GPIO 寄存器

3.4.1. GPIO 寄存器地址映像

GPIOA 寄存器基地址: 0x5000_0000

GPIOB 寄存器基地址: 0x5000_8000

GPIOC 寄存器基地址: 0x5010_0000

GPIOD 寄存器基地址: 0x5010_8000

表格 3-1 GPIO 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	GPIO_OutputEN	GPIO 端口输出使能寄存器
0x04	GPIO_IN_DATA	GPIO 端口输入数据寄存器
0x08	GPIO_OUT_DATA	GPIO 端口输出数据寄存器
0xC	GPIO_BIT_SET_DATA	GPIO 端口输出位操作
0x10	GPIO_BIT_CLEAR_DATA	GPIO 端口清除位操作
0x14	EXTI_EN	EXTI 模式使能

3.4.2. GPIO_OutputEN (Offset 0x00)

比特	名称	属性	复位值	描述
15:0	GPIO_OutputEN	R/W	16'h00	GPIO 输出使能控制寄存器 1bit 对应 1 引脚 1: 输出禁止 0: 输出使能

3.4.3. GPIO_IN_DATA (Offset 0x04)

比特	名称	属性	复位值	描述
15:0	GPIO_IN_DATA	R	16'h00	端口输入数据寄存器 1bit 对应 1 引脚，读出当前引脚状态。

3.4.4. GPIO_OUT_DATA (Offset 0x08)

比特	名称	属性	复位值	描述
15:0	GPIO_OUT_DATA	W	16'h00	端口输出数据寄存器 1bit 对应 1 引脚，写 '1' 输出高，写 '0' 输出低。

3.4.5. GPIO_BIT_SET_DATA (Offset 0x0C)

比特	名称	属性	复位值	描述
31:16	BITS_RESET	W	16'h00	1bit 对应 1 引脚 0: 无效 1: 引脚输出低
15:0	BITS_SET	W	16'h00	1bit 对应 1 引脚 0: 无效 1: 引脚输出高

3.4.6. GPIO_BIT_CLEAR_DATA (Offset 0x10)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	'd0	Reserved
15:0	BITS_RESET	W	16'h00	1bit 对应 1 引脚 0: 无效 1: 引脚输出低

3.4.7. EXTI_EN (Offset 0x14)

比特	名称	属性	复位值	描述
15:0	Port_EXTI_EN	R/W	16'h00	端口 EXTI 模式使能 1bit 对应 1 引脚 0: Disable 0: Enable

4. 外部中断控制器（EXTI）

4.1. 概述

EXTI 支持 64 路相互独立的检测电路并且可以向处理器产生中断请求。每个外部中断控制器都是完全独立的，没有互相共享任何资源。

EXTI 提供 4 种触发类型：高电平触发、低电平触发、上升沿触发、下降沿触发。每个检测电路都可以分别配置。

4.2. 主要特性

- PortA~PortD 每组 16 路，共计 64 路独立外部中断控制
- 电平触发信号硬件防抖

4.3. EXTI 系统框图

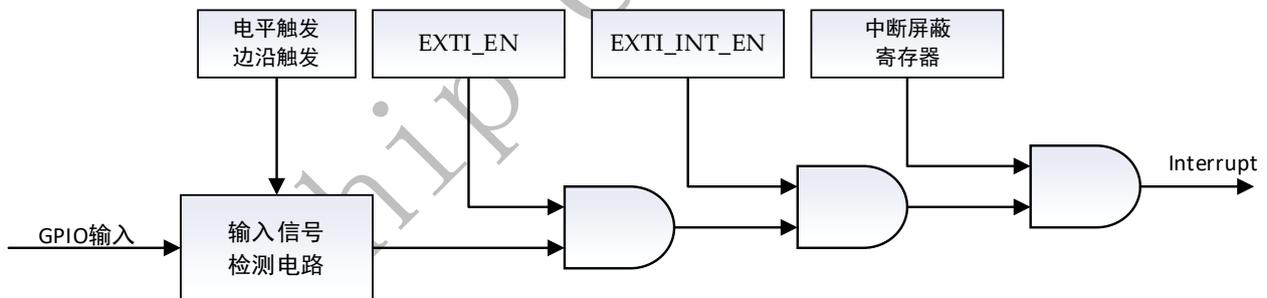


图 4-1 EXTI 系统框图

4.4. EXTI 寄存器

4.4.1. EXTI 寄存器映像

PortA_EXTI 寄存器基地址: 0x5000_0018

PortB_EXTI 寄存器基地址: 0x5000_8018

PortC_EXTI 寄存器基地址: 0x5010_0018

PortD_EXTI 寄存器基地址: 0x5010_8018

表格 4-1 EXTI 寄存器映像

偏移	寄存器	寄存器描述
0x00	EXTI_INT_EN	外部中断线使能寄存器
0x04	EXTI_INT_STATUS	中断状态寄存器
0x08	EXTI_TYPE	触发类型选择
0x0C~0x44	EXTI_CNT0~ EXTI_CNT15	中断线防抖计数值

4.4.2. EXTI_INT_EN (Offset 0x00)

比特	名称	属性	复位值	描述
15:0	EXTI_INT_EN (EXTI 请求线 0 ~ 15)	R/W	15'h00	EXTI (0~15) 中断使能。 EXTI_INT_EN[7:0] 对应引脚 Pin7 ~ Pin0。 EXTI_INT_EN[15:8] 对应引脚 Pin15 ~ Pin8。 1bit 对应 1 条请求线。 0: Disable 1: Enable

4.4.3. EXTI_INT_STATUS (Offset 0x04)

比特	名称	属性	复位值	描述
15:0	EXTI_STS	R/W	15'h00	EXTI (0~15) 中断状态。 EXTI_STS[7:0] 对应引脚 Pin7 ~ Pin0。 EXTI_STS[15:8] 对应引脚 Pin15 ~ Pin8。 1bit 对应 1 条请求线。 写操作：写 '1' 清除状态。 读操作：读出当前中断状态。

4.4.4. EXTI_TYPE (Offset 0x08)

比特	名称	属性	复位值	描述
31:0	EXTI_TYPE	R/W	32'h00	Port 管脚 EXTI 触发类型。 2bit 对应 1 引脚。 EXTI_Port_TYPE[1:0]对应 Pin0 EXTI_Port_TYPE[3:2]对应 Pin1 配置类型： 00：低电平触发。 01：高电平触发。 10：上升沿触发。 11：下降沿触发。

4.4.5. EXTI_CNT0 ~ EXTI_CNT15 (Offset 0x0C ~ 0x44)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:12	PRE_VAL	R/W	4'h0	EXTI (0 ~ 15) 时钟预分频值。分频公式： $CLK = GPIO_CLK / (1 + PRE_VAL)$ 。
11:0	DES_CNT	R/W	12'h0	防抖滤波计数值。 当 EXTI 线上检测到触发信号，信号的持续时间超过了 DES_CNT 设置时间，中断信号置位

5. 通用异步收发器 (UART)

5.1. 概述

UART(通用异步收发器 Universal Asynchronous Receiver/Tranimitter)提供了一种灵活的方法与其他支持标准 NRZ 异步串行数据格式的外部设备进行全双工数据交换。

5.2. 主要特性

- 支持深度 64，宽度 8bit 的 FIFO
- 支持 IrDA SIR 1.0
- 支持小数分频

5.3. 功能描述

5.3.1. 波特率计算

UART 精准波特率计算，可带有小数分频。时钟源可选 CORE_HSCLK 或 SPLL，当选择 SPLL 时钟源时 UART 时钟可实现灵活分频。

整数分频使用两个寄存器共 16bit，其中 DLL 设置低 8 位，DLH 设置高 8 位。

波特率分频计算公式：
$$\text{Baud Rate Divior} = \frac{\text{Serial Clock Frequency}}{16 \times \text{Required Baud Rate}} = \text{BRD}_I + \text{BRD}_F$$

BRD_I ：整数分频部分（DLH 寄存器、DLL 寄存器）

BRD_F ：小数分频部分（DLF 寄存器）

5.3.2. IrDA SIR 1.0 协议

Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode 支持与以红外辐射为传输介质的远程设备进行双向数据通信。IrDA 1.0 规定最大的波特率为 115.2Kbaud。

IrDA SIR 物理层规定使用反相归零调制方案(RZI)，该方案用一个红外光脉冲代表逻辑‘0’。SIR 发送编码器对从 UART 输出的 NRZ(非归零)比特流进行调制。输出脉冲流被传送到一个外部输出驱动器和红外 LED。最高只支持到 115.2Kbps 速率。在正常模式里，脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 UART。在空闲状态里，解码器输入通常是高。发送编码器输出的极性和解码器的输入相反。当解码器输入低时，检测到一个起始位。

- SIR 发送逻辑把‘0’作为高脉冲发送，把‘1’作为低电平发送。
- SIR 接收逻辑把高电平状态解释为‘1’，把低脉冲解释为‘0’。
- 发送编码器输出与解码器输入有着相反的极性。当空闲时，SIR 输出处于低状态
- 在 IrDA 模式里，STOP 位必须配置成 1 个停止位

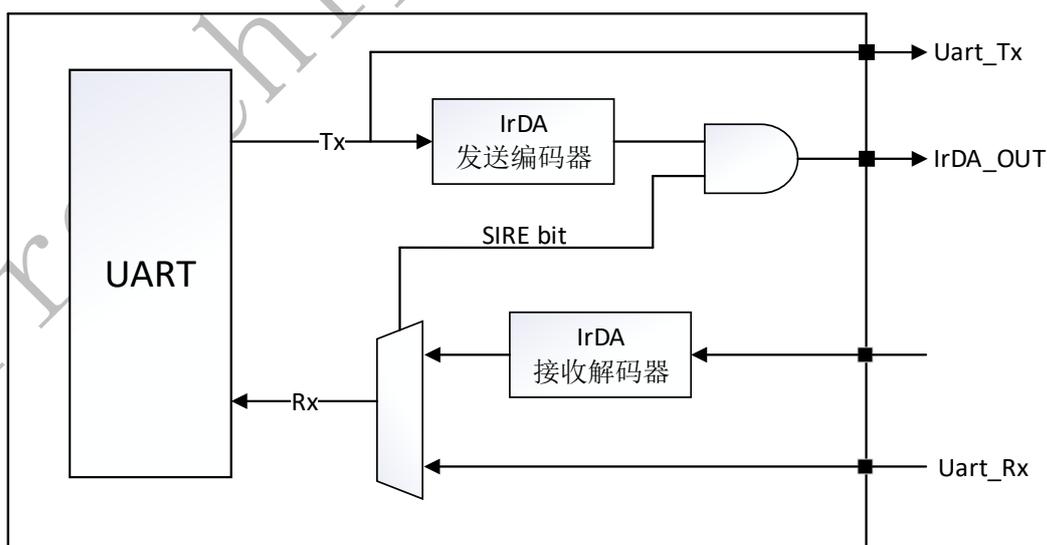


图 5-1 IrDA 框图

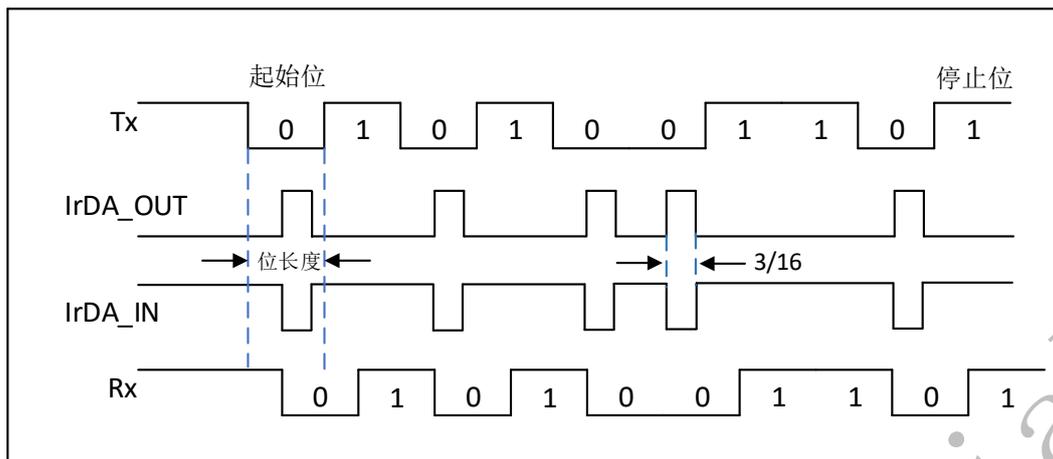


图 5-2 IrDA 时序示意图

5.4. Uart 寄存器

5.4.1. Uart 寄存器映像

Uart0 寄存器基地址: 0x5001_0000

Uart1 寄存器基地址: 0x5001_8000

Uart2 寄存器基地址: 0x5011_0000

Uart3 寄存器基地址: 0x5011_8000

Uart4 寄存器基地址: 0x5021_0000

Uart5 寄存器基地址: 0x5021_8000

表格 5-1 Uart 寄存器映像

偏移	寄存器	寄存器描述
0x00	DATA/DLL	数据寄存器/波特率分频整数部分低 8 位
0x04	IER/DLH	中断使能寄存器/波特率分频整数部分高 8 位
0x08	FCR/IID	FIFO 配置寄存器/中断号查询寄存器
0x0C	LCR	线控制寄存器
0x10	MCR	调制解调器控制寄存器
0x14	LSR	线状态寄存器
0x18	MSR	调制解调器状态寄存器
0x7C	USR	串口状态寄存器
0xC0	DLF	小数分频寄存器

5.4.2. DATA/DLL (Offset 0x00)

比特	名称	属性	复位值	描述
31:9	Reserved	R/W	0x0	Reserved
8:0	DATA	R/W	9'h00	<p>数据接收发送。</p> <p>读操作：</p> <p>当启用 FIFO 时，读数据会将数据从 RxFIFO 中读出。当 RxFIFO 已满时，再次接收到新的数据，则新的数据会被丢失。</p> <p>当禁用 FIFO 时，若寄存器中存在未被读取的数据，再次接收到新的数据，则新的数据会被丢失。</p> <p>写操作：</p> <p>当启用 FIFO 时，写数据会将数据写入 TxFIFO 当 TxFIFO 已满时，再次写入新数据无效。</p> <p>当禁用 FIFO 时，若寄存器中存在未被发送的数据，再次写入新数据无效。</p>

DLL 寄存器，具体如下表所示：

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7:0	DLL	R/W	8'h00	计算波特率分频整数部分，低 8 位。 使用详情参照 波特率计算 章节 注：此寄存器只有在 LCR 寄存器的 DLAB 置 '1' 时可以设置，否则将映射为 DATA 寄存器

注：LCR 寄存器的 DLAB 置 '1' 时是访问 DLL 寄存器描述内容，否则访问的是 DATA 寄存器内容。

5.4.3. IER/DLH (Offset 0x04)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7	PTIME	R/W	1'b0	发送阈值中断。 0: Disable 1: Enable 注：此中断要与 ETBEI 位配合使用
6:4	Reserved	R/W	0x0	Reserved
3	EDSSI	R/W	1'b0	调制解调器状态中断。中断第四优先级。 MSR 寄存器中的流控状态触发中断。 0: Disable 1: Enable
2	ELSI	R/W	1'b0	接收线路状态中断。中断第一优先级。 LSR 状态寄存器中 Overrun/parity/ framing errors, break 信号都将触发中断。 0: Disable 1: Enable
1	ETBEI	R/W	1'b0	发送寄存器空（不使用 FIFO）。 发送阈值中断（使用 FIFO 时，Tx FIFO 中数据等于或小于 FCR 寄存器中 TE 所设阈值，此模式需要设置 PTIME = 1）。 中断第三优先级。 0: Disable 1: Enable 注：阈值设置参考 FCR 寄存器的 TET
0	ERBFI	R/W	1'b0	接收数据中断（不使用 FIFO）。

				接收阈值中断或接收字符超时中断（使用 FIFO 时，接收数据未达到阈值，并持续 4 字节时间没有收到新的数据）。 中断第二优先级 0: Disable 1: Enable 注：阈值设置参考 FCR 寄存器的 RT
--	--	--	--	--

DLH 寄存器，具体如下表所示：

比特	名称	属性	复位值	描述
31:9	Reserved	R/W	0x0	Reserved
8:0	DLH	R/W	8'h00	计算波特率分频整数部分，高 8 位。 使用详情参照 波特率计算 章节 注：此寄存器只有在 LCR 寄存器的 DLAB 置 '1' 时可以设置，否则将映射为 IER 寄存器

注：LCR 寄存器的 DLAB 置 '1' 时可以访问此寄存器描述内容，否则访问的是 IER 寄存器内容

5.4.4. FCR/IID (Offset 0x08)

‘写’操作时将使用 FCR 寄存器描述，具体如下表所示：

比特	名称	属性	复位值	描述
31:8	Reserved	W	0x0	Reserved
7:6	RT	W	2'h0	接收中断触发阈值。 用于接收中断状态和 DMA 请求。 0x0: FIFO 中有 1 字节数据 0x1: FIFO 1/4 满 0x2: FIFO 1/2 满 0x3: FIFO 差 2 字节满
5:4	TET	W	2'h0	发送中断触发阈值。 用于发送中断状态和 DMA 请求。 0x0: FIFO 空 0x1: FIFO 中有 2 字节数据 0x2: FIFO 1/4 满 0x3: FIFO 1/2 满
3	DMAM	W	1'b0	DMA 模式选择。 0: 禁用 FIFO 时 DMA 模式选择

				1: 启用 FIFO 时 DMA 模式选择
2	XFIFOR	W	1'b0	TxFIFO 复位。 写 '1' 复位 TxFIFO, 写 '0' 无效。
1	RFIFOR	W	1'b0	RxFIFO 复位。 写 '1' 复位 TxFIFO, 写 '0' 无效。
0	FIFOE	W	1'b0	FIFO 使能。 0: FIFO Disable 1: FIFO Enable

‘读’操作时将使用 IID 寄存器描述，具体如下表所示：

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3:0	IID	R	4'h1	中断号查询。读出当前处于活跃状态的最高优先级中断。 0x0: 调制解调器状态中断 0x1: 没有中断响应 0x2: 发送空中断 0x4: 接收阈值中断 0x6: 接收线状态中断 (读 LSR 寄存器清除) 0xC: 字符超时中断 (读 RxFIFO 清除)

5.4.5. LCR (Offset 0x0C)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7	DLAB	R/W	1'b0	波特率整数寄存器访问使能。 0: 禁用访问 1: 可以访问 DLL、DLH 注： 在初始波特率设置之后，必须清除此位，以便访问其他寄存器
6	BC	R/W	1'b0	Break 条件控制。 持续拉低 Tx 线，创造 Break 条件。 0: 释放 Tx 线

				1: 持续拉低 Tx 线
5	SP	R/W	1'b0	0/1 校验。 当 SP = 1, EPS = 1, PEN = 1 时, 选择 0 校验。 当 SP = 1, EPS = 0, PEN = 1 时, 选择 1 校验。 0: 0/1 校验禁用 1: 0/1 校验使能
4	EPS	R/W	1'b0	奇偶校验选择。 0: 奇校验 1: 偶校验
3	PEN	R/W	1'b0	校验使能。 0: 禁用校验 1: 使能校验
2	STOP	R/W	1'b0	停止位选择。 0: 1 停止位 1: 1.5 或 2 停止位 注: 当 DLS 设置为 '0' 时为 1.5 个停止位, 选择其他配置时为 2 个停止位。
1:0	DLS	R/W	2'h0	数据长度。 0x0: 数据长度为 5bit 0x1: 数据长度为 6bit 0x2: 数据长度为 7bit 0x3: 数据长度为 8bit

5.4.6. MCR (Offset 0x10)

比特	名称	属性	复位值	描述
31:7	Reserved	R/W	0x0	Reserved
6	SIRE	R/W	1'b0	IrDA 功能。 0: IrDA 功能禁用。 1: IrDA 功能使能。
5	AFCE	R/W	1'b0	自动流控功能。 0: 自动流控功能禁用。 1: 自动流控功能使能。

4:2	Reserved	R/W	0x0	Reserved
1	RTS	R/W	1'b0	<p>RTS (Request to send) 信号控制。 不使用自动流控时为手动控制。</p> <p>0: RTS Inactive 1: RTS Active</p> <p>注: 使用自动流控时, 若此位置'0' RTS 信号将无效。</p>
0	Reserved	R/W	0x0	Reserved

5.4.7. LSR (Offset 0x14)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7	RFE	R	1'b0	<p>RxFIFO 错误状态。 用来指示在 FIFO 中是否至少有一个奇偶校验错误、帧错误或中断指示。当 LSR 被读取, 且错误字符在接收器 FIFO 的顶部, 并且在 FIFO 中没有后续错误时, 该位将被清除</p> <p>0: RxFIFO 正常 1: RxFIFO 中存在错误</p>
6	TEMT	R	1'b1	<p>发送器空。 当 FIFO 启用时, 表示发送移位寄存器和 TxFIFO 空。 当 FIFO 禁用时, 表示发送移位寄存器和发送数据寄存器空。</p> <p>0: 发送器非空 1: 发送器空</p>
5	THRE	R	1'b1	<p>若 IER 寄存器的 PTIME = 0, 则表示发送寄存器或 TxFIFO 空。 若 IER 寄存器的 PTIME = 1, 并且 FCR 寄存器的 FIFOE = 1, 则等于或小于 FCR 寄存器的 TET 设置的发送阈值时, 表示发送阈值中断。</p>
4	BI	R	1'b0	<p>Break 错误。 接收数据时检测到连续的低电平信号产生 Break 信号。</p>

				读 LSR 寄存器清除 BI 位。
3	FE	R	1'b0	帧错误。 接收数据时没有检测到 STOP 位，产生 framing error 信号。 读 LSR 寄存器清除 FE 位。
2	PE	R	1'b0	校验错误。 奇偶，0/1 校验错误。产生 parity error 信号 读 LSR 寄存器清除 PE 位。
1	OE	R	1'b0	溢出错误。 当 FIFO 启用时，RxFIFO 满后继续接收到新的数据产生 overrun error 信号。 当 FIFO 禁用时，DATA 寄存器中数据未被读走，又收到了新的数据时，产生 overrun error 信号。 读 LSR 寄存器清除 OE 位。
0	DR	R	1'b0	数据就绪。 在接收寄存器或者 RxFIFO 中存在至少一个字符时，产生 data ready 信号。 读走数据后，DR 位清除。

5.4.8. MSR (Offset 0x18)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	0x0	Reserved
4	CTS	R	1'b0	CTS 当前状态。 0: CTS Inactive 1: CTS Active
3:0	Reserved	R/W	0x0	Reserved

5.4.9. USR (Offset 0x7C)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	0x0	Reserved
4	RFF	R	1'b0	RxFIFO 满。 0: RxFIFO 非满 1: RxFIFO 满
3	RFNE	R	1'b0	RxFIFO 非空。 0: RxFIFO 空 1: RxFIFO 非空
2	TFE	R	1'b1	TxFIFO 空。 0: TxFIFO 非空 1: TxFIFO 空
1	TFNF	R	1'b1	TxFIFO 非满。 0: TxFIFO 满 1: TxFIFO 非满
0	Reserved	R/W	0x0	Reserved

5.4.10. DLF (Offset 0xC0)

比特	名称	属性	复位值	描述
31:6	Reserved	R/W	0x0	Reserved
5:0	DLF	R/W	4'h0	在计算波特率分频中加入小数。 0x0: $0/64 = 0.0000$ 0x1: $1/64 = 0.015625$ 0x2: $2/64 = 0.03125$ 0x3: $3/64 = 0.046875$ 0x4: $4/64 = 0.0625$ 0x5: $5/64 = 0.078125$ 0x6: $6/64 = 0.09375$ 0x7: $7/64 = 0.109375$ 0x8: $8/64 = 0.125$ 0x9: $9/64 = 0.140625$ 0xA: $10/64 = 0.15625$

			<p>0xB: $11/64 = 0.171875$</p> <p>0xC: $12/64 = 0.1875$</p> <p>0xD: $13/64 = 0.203125$</p> <p>0xE: $14/64 = 0.21875$</p> <p>0xF: $15/64 = 0.234375$</p> <p>...</p> <p>0x3F: $63/64 = 0.984375$</p> <p>使用详情参照波特率计算章节</p>
--	--	--	---

Freqchip Confidential

5.5. 使用流程

5.5.1. 发送数据流程

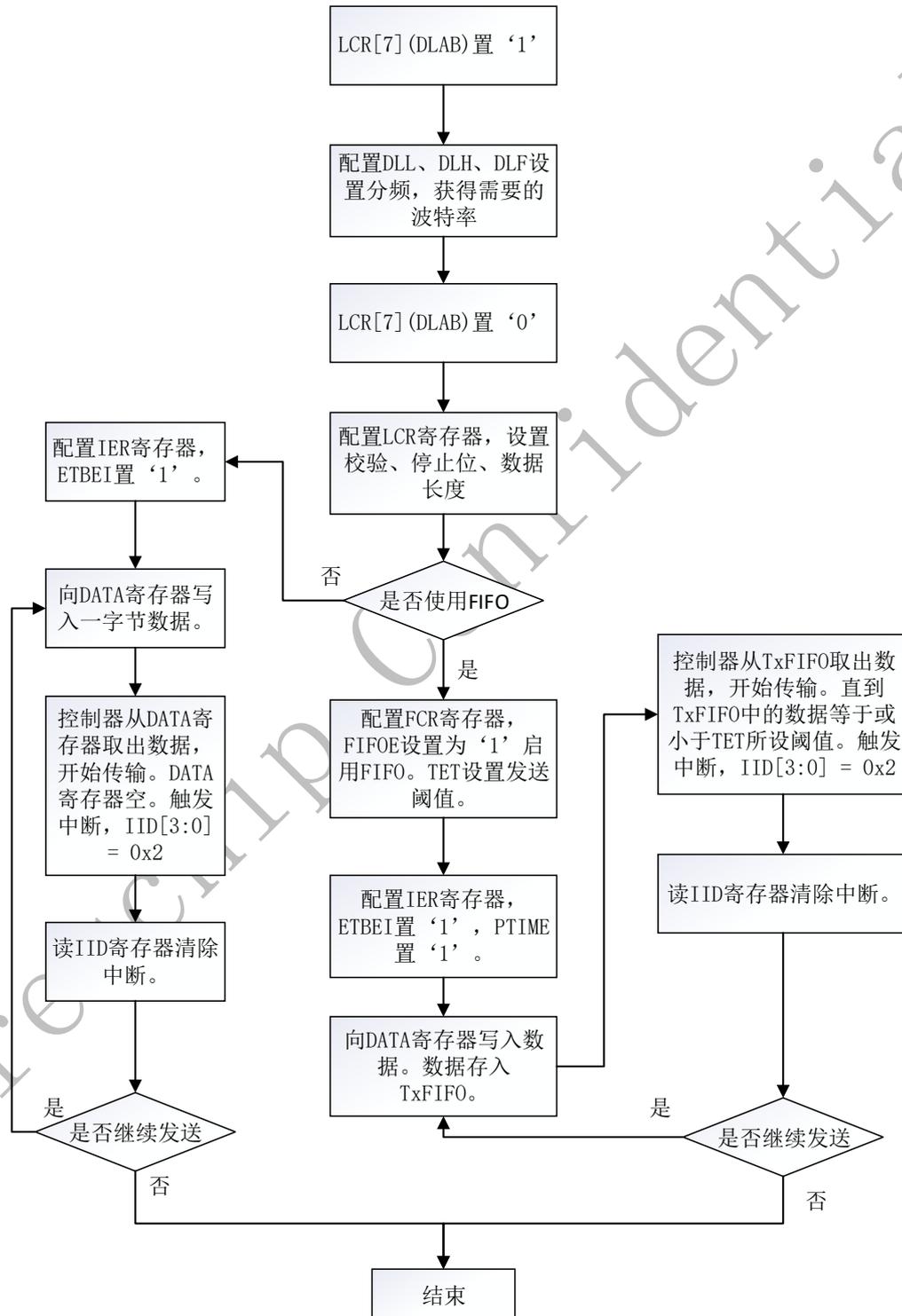


图 5-3 发送数据流程

5.5.2. 接收数据流程

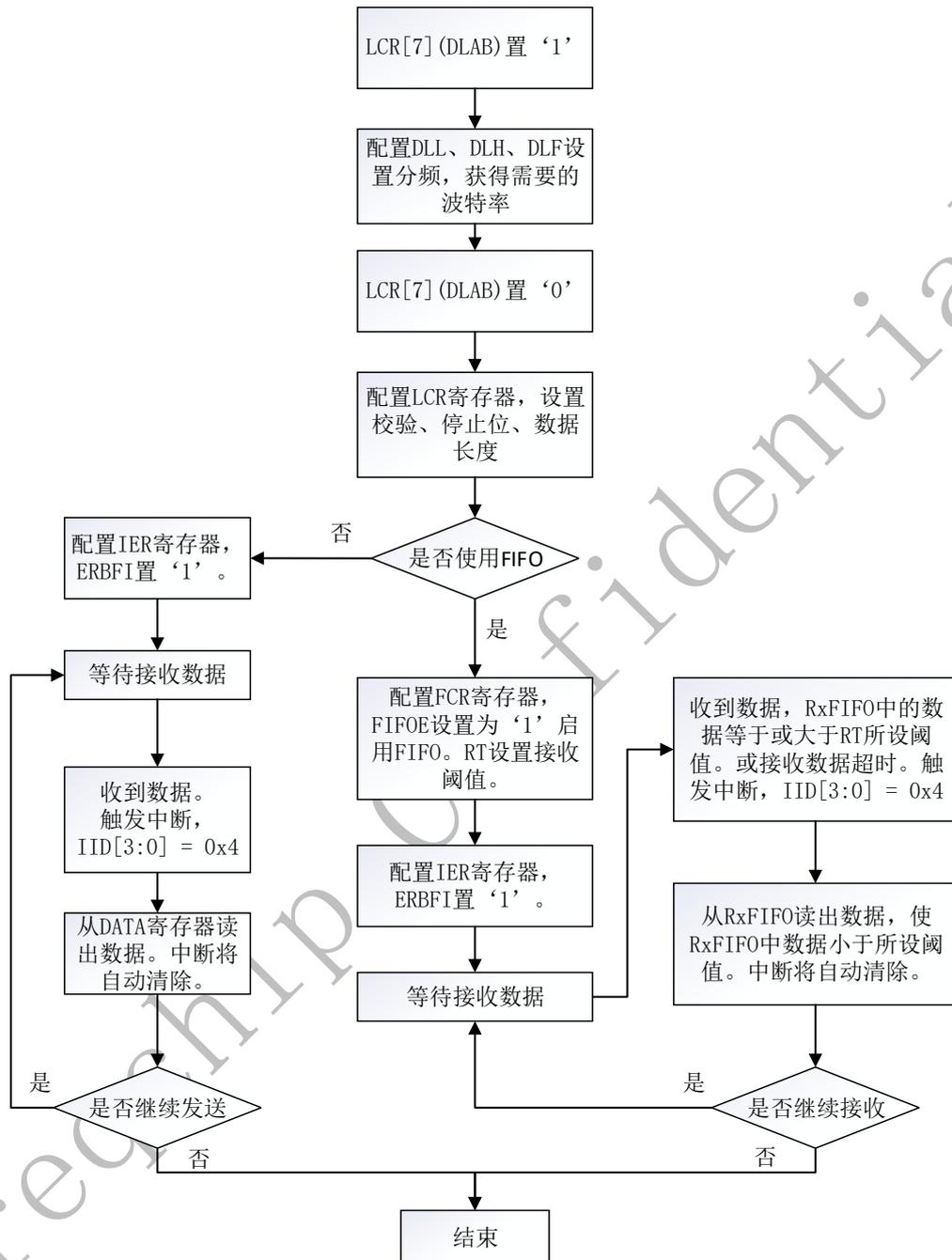


图 5-4 接收数据流程

6. 串行外设接口主设备（SPI_Master）

6.1. 概述

串行外设接口主设备提供了两个符合行业标准的多线主设备 SPI 接口，（2 线 SPIM 以及 8 线 SPIMX8）支持 Motorola, TI, National Semiconductor Microwire 接口，允许与任何 SPI 从设备通信。其中 SPIMX8 还支持高速 DDR 模式。除了标准的 8 位字长之外，SPI 主设备还支持可配置 4 至 32 位字长，用于与非标准的 SPI 字长进行通信。

SPI_Master 信号包括标准串行时钟（SCLK）、从设备选择（CS）、数据发送（TXD）、数据接收（RXD）和数据输出使能（SSOE）。

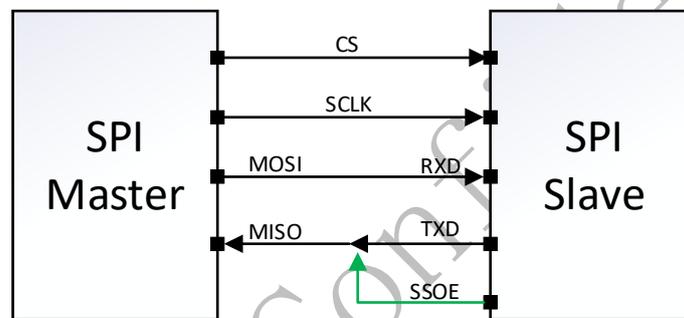


图 6-1 SPI Master 接线示意图

6.2. 主要特性

- SPIM 提供 4 至 16 位数据宽度，SPIMX8 提供 4 至 32 位数据宽度
- SPIM 支持 Standard（一线）Dual（二线），SPIMX8 支持 Standard（一线）Dual（二线）Quad（四线）Octal（八线）接入方式
- SPIM 提供深度 64 位宽 16bit 的 FIFO，SPIMX8 提供深度 64 位宽 32bit 的 FIFO
- 支持高速 DDR，仅 SPIMX8 可用
- 比特率高达 48Mbps
- 支持 DMA

6.3. 功能描述

6.3.1. 发送和接收 FIFO 及中断

SPIM 提供了深度 32，位宽 16bit 的 FIFO 缓存，SPIMX8 提供深度 64，位宽 32bit 的 FIFO。FIFO 缓存中的每个数据实体只能包含一个数据帧，不能将两个相同位的数据帧放在同一个 FIFO 位置。如果只需要取 8 位的数据帧，FIFO 实体中的其他高位需要忽略。

通过写数据寄存器（DR），可以将数据存入发送 FIFO。当发送 FIFO 中的缓存实体个数小于等于发送阈值（TXFTLR）时，会产生 FIFO 空中断（TXE_INTR）。如果尝试继续向已满的发送 FIFO 中填数据，会产生发送 FIFO 溢出中断（TXO_INTR）。

通过读数据寄存器（DR），可以从接收 FIFO 里获取数据。当接收 FIFO 中的缓存实体个数大于等于接收阈值（RXFTLR + 1）时，会产生 FIFO 满中断（RXF_INTR）。如果接收 FIFO 满了后，继续收到数据，新收到的数据会被丢弃，并且产生接收 FIFO 溢出中断（RXO_INTR）。如果尝试从空的接收 FIFO 中读取数据，则会产出接收 FIFO 下溢中断（RXU_INTR），并且读出的数据无效。

6.3.2. 传输模式

传输模式（TMOD）可以通过控制寄存器 0（CTRLR0）设置。注意，此配置对于 Microwire 传输方式无效。

当 **TMOD = 0（收发有效）** 时，发送和接收逻辑都是有效的，从发送 FIFO 里取出数据，并通过 TXD 发送出去，同时从 RXD 上接收数据，存放到接收移动寄存器。在数据帧结束位置，接收到的数据会被存放到接收 FIFO。

当 **TMOD = 1（仅发送）** 时，接收数据是无效的，并且不会存放在接收 FIFO 里，发送数据依然从发送 FIFO 中获取，并通过 TXD 发送出去，但是在数据帧结束后，不会将 RXD 上的数据存放到接收 FIFO 里。使用该模式时，需要屏蔽相关接收中断。

当 **TMOD = 2（仅接收）** 时，发送数据是无效的。不会从发送 FIFO 中取数据，TXD 输出的是一个常量。从 RXD 接收到的数据，会在数据帧结束后，存放到接收 FIFO 里。使用该模式时，需要屏蔽相关发送中断。

当 $\text{TMOD} = 3$ (Flash Read) 时, 发送数据用于向 Flash 设备发送操作码或地址。通常, 这需要四个数据帧(8 位操作码后跟 24 位地址)。在传输操作码和地址期间, 接收逻辑不会捕获任何数据(只要其 Tx 上传输数据, Rx 上的数据就会被忽略)。直到传输 FIFO 为空。因此, 在发送 FIFO 中应该只有足够的帧来为 EEPROM 提供操作码和地址。

当发送 FIFO 变为空(所有控制信息已发送), 接收线(Rx)上的数据是有效的, 并存储在 RxFIFO 中。传输继续进行, 直到收到的数据帧数量与 CTRLR1 寄存器中 NDF 字段的值+1 匹配为止。

6.3.3. DMA 接口

SPIM 与 SPIMX8 各自使用了两个 DMA 信道, 分别对应发送和接收数据。与 DMA 操作相关的寄存器如下所示:

- DMACR – 用来使能 DMA 操作的控制寄存器
- DMATDLR – 用来设置发送 FIFO 的 DMA 请求阈值, 小于等于该阈值产生 DMA 请求
- DMARDLR – 用来设置接收 FIFO 的 DMA 请求阈值, 大于该阈值产生 DMA 请求

6.4. SPI 寄存器

6.4.1. SPI 寄存器映像

SPIM0 基地址: 0x5003_0000

SPIM1 基地址: 0x5013_0000

SPIM2 基地址: 0x5023_0000

SPIMX8_0 基地址: 0x500C_0000

SPIMX8_1 基地址: 0x501C_0000

表格 6-1 SPI 寄存器

偏移	寄存器	寄存器描述
0x00	CTRLR0	控制寄存器 0
0x04	CTRLR1	控制寄存器 1
0x08	SSIENR	使能寄存器
0x0C	MWCR	Microwire 协议传输控制寄存器

0x10	SER	主节点从选择使能寄存器
0x14	BAUDR	波特率设置寄存器
0x18	TXFTLR	发送 FIFO 空中断阈值寄存器
0x1C	RXFTLR	接收 FIFO 满中断阈值寄存器
0x20	TXFLR	发送 FIFO 数量寄存器
0x24	RXFLR	接收 FIFO 数量寄存器
0x28	SR	传输状态寄存器
0x2C	IMR	中断屏蔽寄存器
0x30	ISR	中断状态寄存器
0x34	RISR	原始中断状态寄存器
0x38	TXOICR	发送 FIFO 溢出中断清除寄存器
0x3C	RXOICR	接收 FIFO 溢出中断清除寄存器
0x40	RXUICR	接收 FIFO 下溢中断清除寄存器
0x48	ICR	中断清除寄存器
0x4C	DMACR	DMA 控制寄存器
0x50	DMATDLR	DMA 传输数据寄存器
0x54	DMARDLR	DMA 接收数据寄存器
0x60	DR	数据寄存器
0xF0	RX_SAMPLE_DLY	接收延迟寄存器
0xF4	CTRLR2	控制寄存器 2 (仅多线模式下有效)

6.4.2. CTRLR0 (Offset 0x00)

比特	名称	属性	复位值	描述
31:25	Reserved	R	0x0	Reserved
24	SSTE	R/W	1'b0	连续数据包传输时 CS 线翻转使能： 0: CS 不翻转，保持低电平 1: CS 翻转 注： 只在 SCPH 为 0 时有效。
23	Reserved	R	0x0	Reserved
22:21	SPI_FRF	R/W	2'b00	SPI 模式选择： 00: Standard (一线) 01: Dual (二线) 10: Quad (四线)

				11: Octal (八线)
20:16	DFS_32	R/W	5'h0	<p>数据帧宽度</p> <p>0x3: 4 位数据位宽</p> <p>0x4: 5 位数据位宽</p> <p>0x5: 6 位数据位宽</p> <p>0x6: 7 位数据位宽</p> <p>0x7: 8 位数据位宽</p> <p>0x8: 9 位数据位宽</p> <p>0x9: 10 位数据位宽</p> <p>0xA: 11 位数据位宽</p> <p>0xD: 12 位数据位宽</p> <p>0xC: 13 位数据位宽</p> <p>0xD: 14 位数据位宽</p> <p>0xE: 15 位数据位宽</p> <p>0xF: 16 位数据位宽</p> <p>.....</p> <p>0x1F: 32 位数据位宽</p>
15:12	CFS	R/W	4'h0	<p>控制帧宽度，针对 Microwire 帧格式。</p> <p>0x0: 1 位控制位宽</p> <p>0x1: 2 位控制位宽</p> <p>0x2: 3 位控制位宽</p> <p>0x3: 4 位控制位宽</p> <p>0x4: 5 位控制位宽</p> <p>0x5: 6 位控制位宽</p> <p>0x6: 7 位控制位宽</p> <p>0x7: 8 位控制位宽</p> <p>0x8: 9 位控制位宽</p> <p>0x9: 10 位控制位宽</p> <p>0xA: 11 位控制位宽</p> <p>0xB: 12 位控制位宽</p> <p>0xC: 13 位控制位宽</p> <p>0xD: 14 位控制位宽</p> <p>0xE: 15 位控制位宽</p> <p>0xF: 16 位控制位宽</p>
11	SRL	R/W	1'b0	<p>移位寄存器循环，发送移位寄存器与接收移位寄存器对接。</p>

				1: 测试模式, : TX 和 RX 对接 0: 正常模式
10	Reserved	R	0x0	Reserved
9:8	TMOD	R/W	2'b00	传输模式选择。 00: 发送和接收 01: 仅发送 (Standard 模式不可用) 10: 仅接收 (Standard 模式不可用) 11: Flash Read 注: 当选择为 Dual (二线) 或 Quad (四线) 时, TMOD 只能配置为仅发送 或 仅接收。
7	SCPOL	R/W	1'b0	时钟极性, 仅适用于 Motorola SPI 0: 非活动状态下, 时钟保持低电平 1: 非活动状态下, 时钟保持高电平
6	SCPH	R/W	1'b0	时钟相位, 仅适用于 Motorola SPI 0: 数据在 SCLK 的第一个沿有效 1: 数据在 SCLK 的第二个沿有效
5:4	FRF	R/W	2'b00	帧格式选择 0: MOTOROLA SPI: Motorola SPI Frame Format 1: TEXAS SSP: Texas Instruments SSP Frame Format 2: NS_MICROWIRE: National Microwire Frame Format 3: RESERVED
3:0	Reserved	R	0x0	Reserved

6.4.3. CTRLR1 (Offset 0x04)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	NDF	R/W	16'h00	接收数据长度设置寄存器: TMOD = 10 或 TMOD = 11, 设置该寄存器字可配置连续接收数据长度。可持续接收数据等此寄存器值加 1。最多可达 64K 数据。

6.4.4. SSIENR (Offset 0x08)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	SSI_EN	R/W	1'b0	是否使能 SPIM，当不使能时，会清空接收和发送 FIFO 1: 使能 0: 不使能

6.4.5. MWCR (Offset 0x0C)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	MHS	R/W	1'b0	Microwire 协议的 busy/ready 握手接口。 使能后转移最后一个数据/控制位之后，在清除 SR 寄存器中的 BUSY 状态之前，检查目标从机的 ready 状态 0: Disable 1: Enable
1	MDD	R/W	1'b0	在 Microwire 协议里，定义数据传输方向 0: 接收数据 1: 发送数据
0	MWMOD	R/W	1'b0	Microwire 协议数据传输是否是顺序的 0: 非顺序的数据传输 1: 顺序的数据传输

6.4.6. SER (Offset 0x10)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	SER	R/W	1'b0	从选择使能。 0: CS 片选信号不使能 1: CS 片选信号使能

6.4.7. BAUDR (Offset 0x14)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	SCKDV	R/W	16'h00	<p>SPI 输出时钟分频。</p> <p>SCKDV 的设置值必须为偶数。</p> <p>$SPI_CLK_OUT = APB_SPI_CLK / SCKDV$</p> <p>SCKDV 取值范围：2 ~ 65534</p> <p>注：APB_SPI_CLK 可通过 Clock Control 寄存器的 SPI0/1_Master_CLK_SEL 选择 48MHz 或 96MHz。</p>

6.4.8. TXFTLR (Offset 0x18)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	TFT	R/W	5'h0	发送 FIFO 阈值，如果发送 FIFO 中的实体个数小于等于该阈值时，产生发送 FIFO 空中断

6.4.9. RXFTLR (Offset 0x1C)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	RFT	R/W	5'h0	接收 FIFO 阈值，如果接收 FIFO 中的实体个数大于等于该阈值 + 1 时，产生接收 FIFO 满中断

6.4.10. TXFLR (Offset 0x20)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	TXTFL	R	6'h0	发送 FIFO 中有效数据实体个数

6.4.11. RXFLR (Offset 0x24)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	RXTFL	R	6'h0	接收 FIFO 中有效数据实体个数

6.4.12. SR (Offset 0x28)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RFF	R	1'b0	0: 接收 FIFO 非满 1: 接收 FIFO 满,
3	RFNE	R	1'b0	0: 接收 FIFO 空 1: 接收 FIFO 非空
2	TFE	R	1'b1	0: 发送 FIFO 非空 1: 发送 FIFO 空
1	TFNF	R	1'b1	0: 发送 FIFO 满 1: 发送 FIFO 非满
0	BUSY	R	1'b0	SPI 忙标志 0: SPI 空闲或者没有使能 1: SPI 正在传输数据

6.4.13. IMR (Offset 0x2C)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIM	R/W	1'b1	接收 FIFO 满中断屏蔽。
3	RFOIM	R/W	1'b1	接收 FIFO 溢出中断屏蔽。
2	RXUIM	R/W	1'b1	接收 FIFO 下溢中断屏蔽。
1	TXOIM	R/W	1'b1	发送 FIFO 溢出中断屏蔽。
0	TXEIM	R/W	1'b1	发送 FIFO 空中断屏蔽。 0: 屏蔽 1: 不屏蔽

6.4.14. ISR (Offset 0x30)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIS	R	1'b0	接收 FIFO 满中断状态。(IMR 屏蔽后)
3	RXOIS	R	1'b0	接收 FIFO 溢出中断状态。(IMR 屏蔽后)
2	RXUIS	R	1'b0	接收 FIFO 下溢中断状态。(IMR 屏蔽后)
1	TXOIS	R	1'b0	发送 FIFO 溢出中断状态。(IMR 屏蔽后)
0	TXEIS	R	1'b0	发送 FIFO 空中断状态。(IMR 屏蔽后) 0: Inactive 1: Active

6.4.15. RISR (Offset 0x34)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIR	R	1'b0	原始接收 FIFO 满中断状态。
3	RXOIR	R	1'b0	原始接收 FIFO 溢出中断状态。
2	RXUIR	R	1'b0	原始接收 FIFO 下溢中断状态。
1	TXOIR	R	1'b0	原始发送 FIFO 溢出中断状态。
0	TXEIR	R	1'b0	原始发送 FIFO 空中断状态。 0: Inactive 1: Active

6.4.16. TXOICR (Offset 0x38)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	TXOICR	R	1'b0	发送 FIFO 溢出中断清除。 读取该寄存器清除发送 FIFO 溢出中断状态。

6.4.17. RXOICR (Offset 0x3C)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	RXOICR	R	1'b0	接收 FIFO 溢出中断清除。 读取该寄存器清除接收 FIFO 溢出中断状态。

6.4.18. RXUICR (Offset 0x40)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	RXUICR	R	1'b0	接收 FIFO 下溢中断清除。 读取该寄存器清除接收 FIFO 下溢中断状态。

6.4.19. ICR (Offset 0x48)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	ICR	R	1'b0	清中断状态 读取该寄存器会清除发送 FIFO 溢出，接收 FIFO 溢出和接收 FIFO 下溢中断状态。

6.4.20. DMACR (Offset 0x4C)

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	TDMAE	R/W	1'b0	0: 发送 DMA 不使能 1: 发送 DMA 使能
0	RDMAE	R/W	1'b0	0: 接收 DMA 不使能 1: 接收 DMA 使能

6.4.21. DMATDLR (Offset 0x50)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
4:0	DMATDL	R/W	5'h0	DMA 发送数据阈值，当发送 FIFO 中的有效数据实体个数小于等于该阈值，并且 TDMAE = 1 时，产生 DMA_Tx 请求

6.4.22. DMARDLR (Offset 0x54)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
4:0	DMARDL	R/W	5'h0	DMA 接收数据阈值，当接收 FIFO 中的有效数据实体个数大于等于该阈值 + 1 时，并且 RDMAE = 1 时，产生 DMA_Rx 请求

6.4.23. DR(offset 0x60)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	DR	R/W	16'h00	数据寄存器，读取该寄存器时，从接收 FIFO 取值，写入该寄存器时，会写入到发送 FIFO。

6.4.24. RX_SAMPLE_DLY (Offset 0xF0)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
7:0	RSD	R/W	8'h00	接收延迟采样。 接收数据时在原采样点延迟 RSD 个时钟信号在进行采样。（时钟信号为 APB_SPI_CLK）

6.4.25. CTRLR2 (Offset 0xF4)

比特	名称	属性	复位值	描述
31:30	Reserved	R	0x0	Reserved
29	SPI_RX_NO_INST_ADD	R/W	1'b0	无地址指令接收模式 0: Disable 1: Eneable
28:24	Reserved	R	0x0	Reserved
23:22	RX_ENDIAN	R/W	2'b00	接收采样延时 单位为 SPI_CLK
21:20	TX_ENDIAN	R/W	2'b00	发送采样延时 单位为 SPI_CLK
19:17	Reserved	R	0x0	Reserved
16	DDR_EN	R/W	1'b0	DDR 模式使能 0: Disable 1: Enable
15:11	WAIT_CYCLES	R/W	5'h0	等待周期。 在二线。四线模式下，控制帧发送和数据接收之间的等待周期数。
10	Reserved	R	0x0	Reserved
9:8	INST_L	R/W	2'h0	指令长度。 在二线。四线模式下，指令长度。 0x0: 没有指令 0x1: 4bit 指令 0x2: 8bit 指令 0x3: 16bit 指令
7:6	Reserved	R	0x0	Reserved
5:2	ADDR_L	R/W	4'h0	地址长度。 只有将设置的长度写入 FIFO 后，传输开始。 0x0: 0bit 地址位宽 0x1: 4bit 地址位宽 0x2: 8bit 地址位宽 0x3: 12bit 地址位宽 0x4: 16bit 地址位宽

				0x5: 20bit 地址位宽 0x6: 24bit 地址位宽 0x7: 28bit 地址位宽 0x8: 32bit 地址位宽 0x9: 36bit 地址位宽 0xA: 40bit 地址位宽 0xB: 44bit 地址位宽 0xC: 48bit 地址位宽 0xD: 52bit 地址位宽 0xE: 56bit 地址位宽 0xF: 60bit 地址位宽
1:0	TRANS_TYPE	R/W	2'b00	地址和指令传输格式设置寄存器： 00: 指令、地址都使用一线模式。 01: 指令使用一线模式发送，地址使用多线模式 10: 指令和地址都使用多线模式。 11: 保留 注：多线模式使用二线或四线由 SPI_FRF 决定。

注：CTRLR2 寄存器仅在 SPIM 的 Dual（二线）与 SPIMX8 的 Dual（二线）、Quad（四线）、Octal（八线）模式下有效。

6.5. 使用流程

6.5.1. SPI 标准模式发送

1. 配置 SPI 控制寄存器 CTRLR0。通过 FRF 选择通讯协议，DFS_32 选择数据位宽，SPI_FRF 选择一线二线四线八线，SCPH、SCPOL 选择时钟相位、极性。
2. 配置 BAUDR 寄存器设置时钟分频。
3. 配置 SER 寄存器，设置 SSI_EN 位，拉低 CS 片选信号。
4. 向 DR 寄存器写入数据，数据将存在 FIFO 中。
5. SPI 发送数据。
6. 等 FIFO 数据未滿或者为空时，重复步骤 4，直到数据发送完成。
7. 配置 SER 寄存器，清 SSI_EN 位，释放 CS 片选信号。

6.5.2. SPI 标准模式接收

1. 配置 SPI 控制寄存器 CTRLR0。通过 FRF 选择通讯协议，DFS_32 选择数据位宽，SPI_FRF 选择一线二线四线八线，SCPH、SCPOL 选择时钟相位、极性。
2. 配置 BAUDR 寄存器设置时钟分频。
3. 配置 SER 寄存器，设置 SSI_EN 位，拉低 CS 片选信号。
4. 向 DR 寄存器写入要接收的数据长度的数据（用来产生 Clock）。不能超过 FIFO 深度。
5. 判断 TxFIFO 非空时，读取 TxFIFO 中的数据。

6.5.3. Dual（二线）、Quad（四线）、Octal（八线）写

当使用多线方式通讯时（SPI_FRF = 1、SPI_FRF = 2、SPI_FRF = 3）。写操作可以分为三个部分。

- 指令部分
- 地址部分
- 数据部分

与标准模式相比较，需要多配置的相关寄存器：

- CTRLR0.SPI_FRF：选择二线、四线、八线
- CTRLR0.DFS：数据长度
- CTRLR2.INST_L：指令长度
- CTRLR2.ADDR_L：地址长度
- CTRLR2.TRANS_TYPE：传输类型

其中指令占一个 FIFO 深度，地址可以占多个 FIFO 深度。将指令和地址全部写入 TxFIFO，并使得写入的数据与配置的长度相匹配时，传输开始。

用两线模式举例说明：

Case 1: 指令和地址，使用标准模式。数据使用多线模式。CTRLR2.TRANS_TYPE = 0。

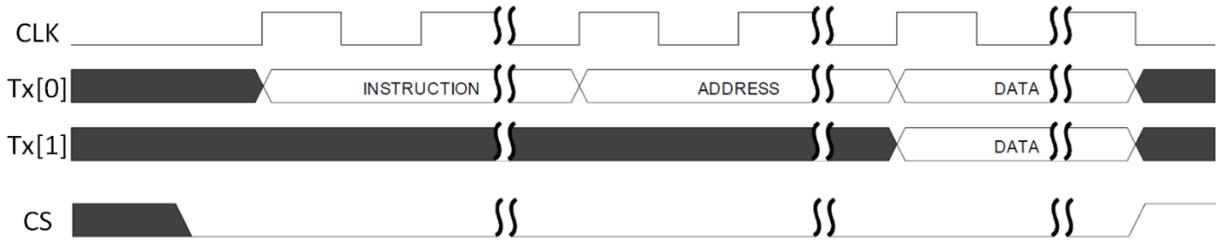


图 6-2 指令和地址使用标准模式、数据使用多线模式

Case 2: 指令使用标准模式。地址和数据使用多线模式。CTRLR2.TRANS_TYPE = 1。

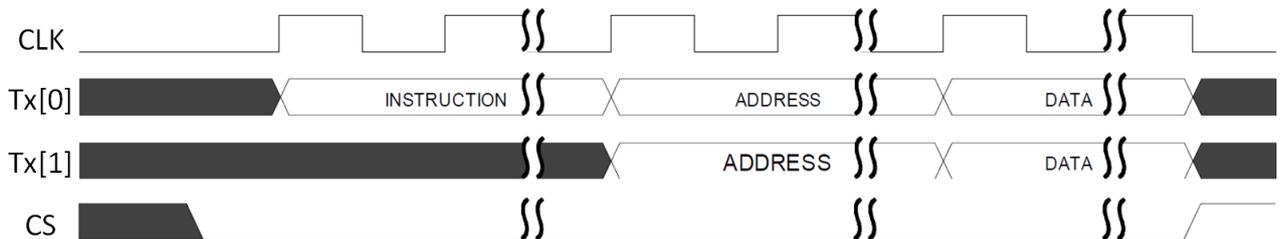


图 6-3 指令使用标准模式、地址和数据使用多线模式。

Case 3: 指令、地址和数据使用多线模式。CTRLR2.TRANS_TYPE = 2。

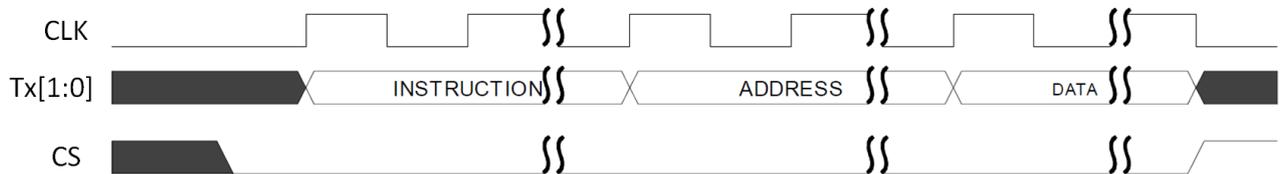


图 6-4 指令、地址和数据使用多线模式

Case 4: 只有指令使用多线模式。CTRLR2.TRANS_TYPE = 2。

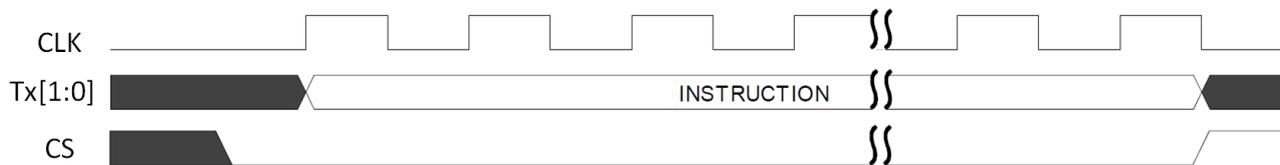


图 6-5 只有指令使用多线模式

6.5.4. Dual（二线）、Quad（四线）、Octal（八线）读

当使用多线方式通讯时（SPI_FRF = 1、SPI_FRF = 2、SPI_FRF = 2）。写操作可以分为四个部分。

■ 指令部分

- 地址部分
- 等待周期
- 数据部分

与标准模式相比较，需要多配置的相关寄存器：

- CTRLR0.SPI_FRF：选择二线、四线、八线
- CTRLR0.DFS：数据长度
- CTRLR2.INST_L：指令长度
- CTRLR2.ADDR_L：地址长度
- CTRLR2.TRANS_TYPE：传输类型
- CTRLR2.WAIT_CYCLES：等待周期

其中指令占一个 FIFO 深度，地址可以占多个 FIFO 深度。将指令和地址全部写入 Tx FIFO，并使得写入的数据与配置的长度相匹配时，传输开始。

用两线模式举例说明：



图 6-6 读数据、指令和地址使用标准模式、数据使用多线模式

Case 2: 指令使用标准模式。地址和数据使用多线模式。CTRLR2.TRANS_TYPE = 1。

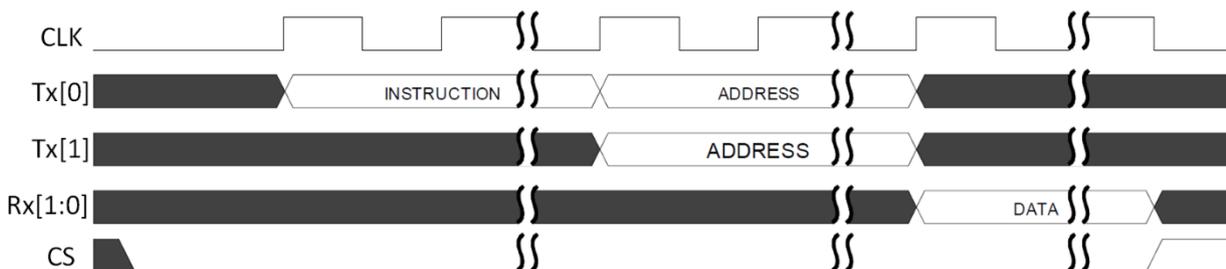


图 6-7 读数据、指令使用标准模式、地址和数据使用多线模式

Case 3: 指令、地址和数据使用多线模式。CTRLR2.TRANS_TYPE = 2。

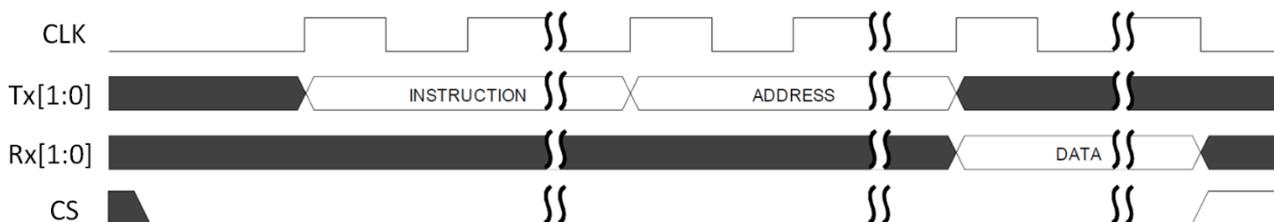


图 6-8 读数据，指令、地址和数据使用多线模式

Case 4: 没有指令，没有地址，仅数据传输。

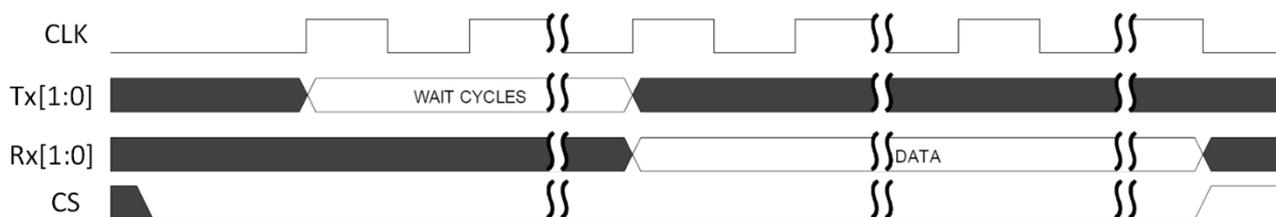


图 6-9 读数据，没有指令，没有地址，仅数据传输

Freqchip Confidential

6.6. MOTOROLA SPI 时序

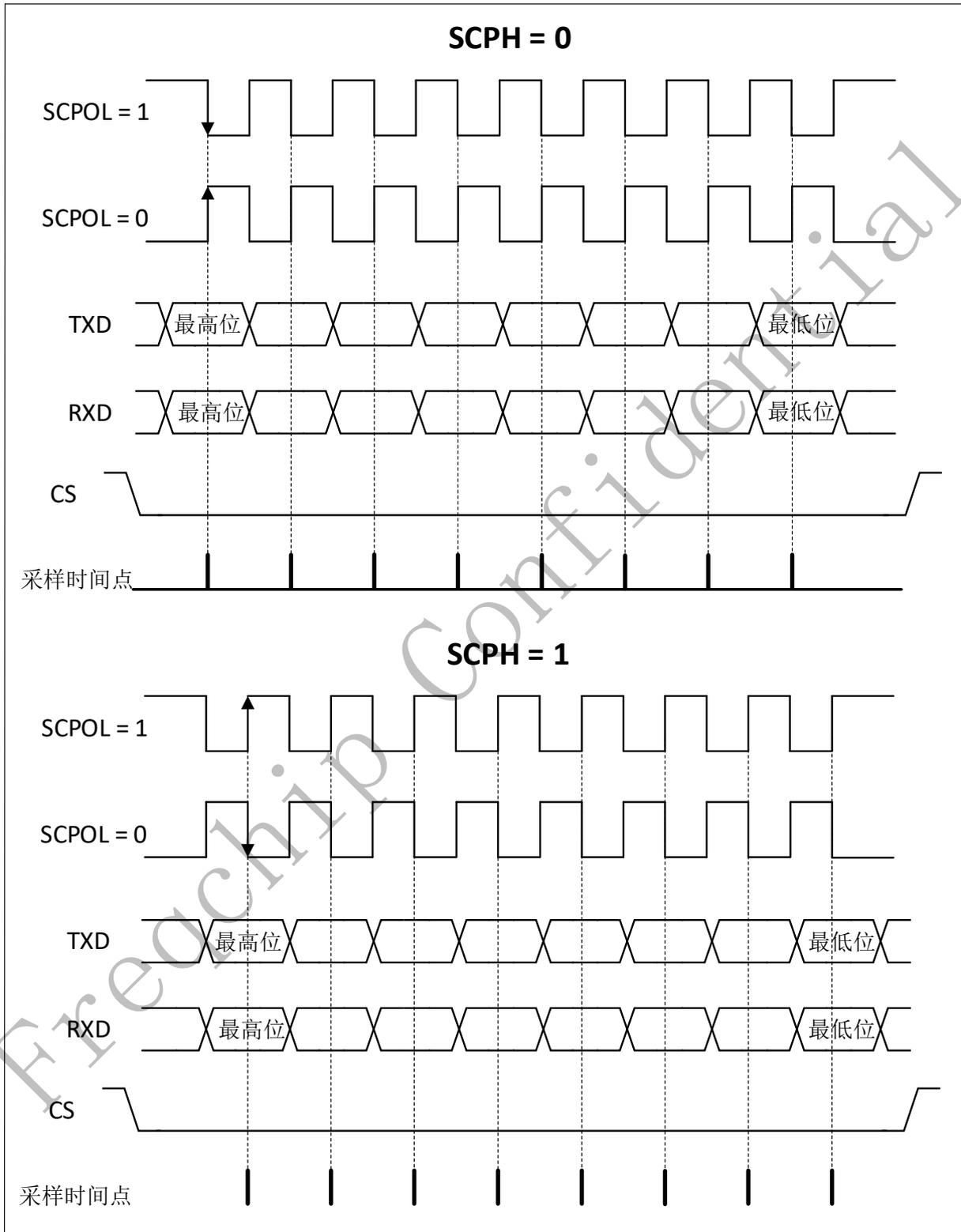


图 6-10 MOTOROLA SPI 时序图

7. 串行外设接口从设备（SPI_Slave）

7.1. 概述

串行外设接口从设备（SPIS）提供了两个符合行业标准的从设备 SPI 接口（单线 SPIS0、4 线 SPIS1）。支持 Motorola, TI, National Semiconductor Microwire 接口，允许与任何 SPI 主设备通信。除了标准的 8 位字长之外，SPIS0 支持可配置 4 至 16 位字长，SPIS1 还支持 4 至 32 位字长，用于与非标准的 SPI 字长进行通信。

SPIS 信号包括标准串行时钟（SCLK）、从设备选择（CS）、数据发送（TXD）、数据接收（RXD）和数据输出使能（SSOE）。

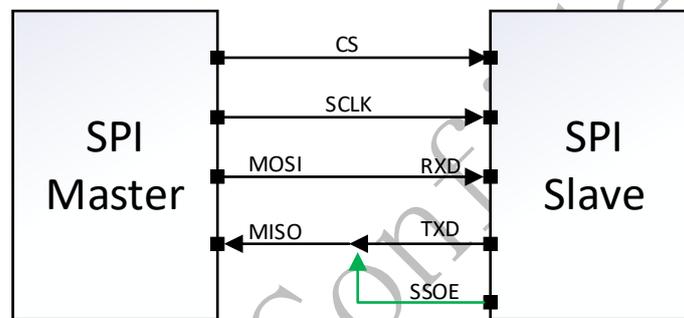


图 7-1 SPI Slave 接线示意图

7.2. 主要特性

- SPIS0 支持可配置 4 至 16 位数据位宽，SPIS1 支持可配置 4 至 32 位数据位宽
- 比特率高达 48 Mbps
- 支持 DMA
- SPIS0 支持 Standard（一线），SPIS1 支持 Dual（二线）、Quad（四线）接入方式
- SPIS0 提供深度 16，位宽 16bit 的 FIFO，SPIS1 支持深度 64，位宽 32bit 的 FIFO

7.3. 功能描述

7.3.1. 发送和接收 FIFO 及中断

SPIS0 提供了深度 32，位宽 16bit 的 FIFO 缓存，SPIS1 提供深度 64，位宽 32bit 的 FIFO。FIFO 缓存中的每个数据实体只能包含一个数据帧，不能将两个相同位的数据帧放在同一个 FIFO 位置。如果只需要取 8 位的数据帧，FIFO 实体中的其他高位需要忽略。

通过写数据寄存器（DR），可以将数据存入发送 FIFO。当发送 FIFO 中的缓存实体个数小于等于发送阈值（TXFTLR）时，会产生 FIFO 空中断（TXE_INTR）。如果尝试继续向已满的发送 FIFO 中填数据，会产生发送 FIFO 溢出中断（TXO_INTR）。

通过读数据寄存器（DR），可以从接收 FIFO 里获取数据。当接收 FIFO 中的缓存实体个数大于等于接收阈值（RXFTLR + 1）时，会产生 FIFO 满中断（RXF_INTR）。如果接收 FIFO 满了后，继续收到数据，新收到的数据会被丢弃，并且产生接收 FIFO 溢出中断（RXO_INTR）。如果尝试从空的接收 FIFO 中读取数据，则会产出接收 FIFO 下溢中断（RXU_INTR），并且读出的数据无效。

7.3.2. 传输模式

传输模式（TMOD）可以通过控制寄存器 0（CTRLR0）设置。注意，此配置对于 Microwire 传输方式无效。

当 **TMOD = 0（收发有效）** 时，发送和接收逻辑都是有效的，从发送 FIFO 里取出数据，并通过 TXD 发送出去，同时从 RXD 上接收数据，存放到接收移动寄存器。在数据帧结束位置，接收到的数据会被存放到接收 FIFO。

当 **TMOD = 1（仅发送）** 时，接收数据是无效的，并且不会存放在接收 FIFO 里，发送数据依然从发送 FIFO 中获取，并通过 TXD 发送出去，但是在数据帧结束后，不会将 RXD 上的数据存放到接收 FIFO 里。使用该模式时，需要屏蔽相关接收中断。

当 **TMOD = 2（仅接收）** 时，发送数据是无效的。不会从发送 FIFO 中取数据，TXD 输出的是一个常量。从 RXD 接收到的数据，会在数据帧结束后，存放到接收 FIFO 里。使用该模式时，需要屏蔽相关发送中断。

7.3.3. DMA 接口

SPIS 使用了两个 DMA 信道，分别对应发送和接收数据。SPIS 中与 DMA 操作相关的寄存器如下所示：

- DMACR – 用来使能 DMA 操作的控制寄存器
- DMATDLR – 用来设置发送 FIFO 的 DMA 请求阈值，小于等于该阈值产生 DMA 请求
- DMARDLR – 用来设置接收 FIFO 的 DMA 请求阈值，大于该阈值产生 DMA 请求

7.4. SPI 寄存器

7.4.1. SPI 寄存器映像

SPIS0 寄存器基地址：0x5004_0000

SPIS0 寄存器基地址：0x5014_0000

表格 7-1 SPIS 寄存器

偏移	寄存器	寄存器描述
0x00	CTRLR0	控制寄存器
0x08	SSIENR	使能寄存器
0x0C	MWCR	Microwire 协议传输控制寄存器
0x18	TXFTLR	发送 FIFO 阈值配置寄存器
0x1C	RXFTLR	接收 FIFO 阈值配置寄存器
0x20	TXFLR	发送 FIFO 数量寄存器
0x24	RXFLR	接收 FIFO 数量寄存器
0x28	SR	传输状态寄存器
0x2C	IMR	中断屏蔽寄存器
0x30	ISR	中断状态寄存器
0x34	RISR	原始中断状态寄存器
0x38	TXOICR	发送 FIFO 溢出中断清除寄存器
0x3C	RXOICR	接收 FIFO 溢出中断清除寄存器
0x40	RXUICR	接收 FIFO 下溢中断清除寄存器
0x48	ICR	中断清除寄存器

0x4C	DMACR	DMA 控制寄存器
0x50	DMATDLR	DMA 发送 FIFO 阈值设置
0x54	DMARDLR	DMA 接收 FIFO 阈值设置
0x60	DR	数据寄存器
0xF4	CTRLR2	控制寄存器 2 (仅多线模式下有效)

7.4.2. CTRLR0 (Offset 0x00)

比特	名称	属性	复位值	描述
31:25	Reserved	R	0x0	Reserved
24	SSTE	R/W	1'b0	连续数据包传输时 CS 线翻转使能： 0: CS 不翻转，保持低电平 1: CS 翻转 注：只在 SCPH 为 0 时有效。
23:21	Reserved	R	0x0	Reserved
20:16	DFS_32	R/W	5'h0	数据帧宽度 0x3: 4 位数据位宽 0x4: 5 位数据位宽 0x5: 6 位数据位宽 0x6: 7 位数据位宽 0x7: 8 位数据位宽 0x8: 9 位数据位宽 0x9: 10 位数据位宽 0xA: 11 位数据位宽 0xD: 12 位数据位宽 0xC: 13 位数据位宽 0xD: 14 位数据位宽 0xE: 15 位数据位宽 0xF: 16 位数据位宽 0x1F: 32 位数据位宽
15:12	CFS	R/W	4'h0	控制帧宽度，针对 Microwire 帧格式。 0x0: 1 位控制位宽 0x1: 2 位控制位宽 0x2: 3 位控制位宽

				<p>0x3: 4 位控制位宽 0x4: 5 位控制位宽 0x5: 6 位控制位宽 0x6: 7 位控制位宽 0x7: 8 位控制位宽 0x8: 9 位控制位宽 0x9: 10 位控制位宽 0xA: 11 位控制位宽 0xB: 12 位控制位宽 0xC: 13 位控制位宽 0xD: 14 位控制位宽 0xE: 15 位控制位宽 0xF: 16 位控制位宽</p> <p>注: 选择 Microwire 格式时有效。</p>
11	SRL	R/W	1'b0	<p>移位寄存器循环, 发送移位寄存器与接收移位寄存器对接。</p> <p>1: 测试模式, TX 和 RX 对接 0: 正常模式</p>
10	SLV_OE	R/W	1'b0	<p>从设备输出使能。</p> <p>1: 输出不使能 0: 输出使能</p>
9:8	TMOD	R/W	2'b00	<p>传输模式选择。</p> <p>00: 发送和接收 01: 仅发送 (Standard 模式不可用) 10: 仅接收 (Standard 模式不可用) 11: 保留</p> <p>注: 当选择为 Dual (二线) 或 Quad (四线) 时, TMOD 只能配置为仅发送 或 仅接收。</p>
7	SCPOL	R/W	1'b0	<p>时钟极性, 仅适用于 Motorola SPI</p> <p>0: 非活动状态下, 时钟保持低电平 1: 非活动状态下, 时钟保持高电平</p>
6	SCPH	R/W	1'b0	<p>时钟相位, 仅适用于 Motorola SPI</p> <p>0: 数据在 SCLK 的第一个沿有效 1: 数据在 SCLK 的第二个沿有效</p>
5:4	FRF	R/W	2'b00	<p>帧格式选择</p>

				0: MOTOROLA SPI: Motorola SPI Frame Format 1: TEXAS SSP: Texas Instruments SSP Frame Format 2: NS_MICROWIRE: National Microwire Frame Format 3: RESERVED
3:0	Reserved	R	0x0	Reserved

7.4.3. SSIENR (Offset 0x08)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	SSI_EN	R/W	1'b0	是否使能 SPIS, 当不使能时, 会清空接收和发送 FIFO 1: 使能 0: 不使能

7.4.4. MWCR (Offset 0x0C)

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	MDD	R/W	1'b0	在 Microwire 协议里, 定义数据传输方向 0: 接收数据 1: 发送数据
0	MWMOD	R/W	1'b0	Microwire 协议数据传输是否是顺序的 0: 非顺序的数据传输 1: 顺序的数据传输

7.4.5. TXFTLR (Offset 0x18)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3:0	TFT	R/W	4'h0	发送 FIFO 阈值, 如果发送 FIFO 中的实体个数小于等于该阈值时, 产生发送 FIFO 空中

				断
--	--	--	--	---

7.4.6. RXFTLR (Offset 0x1C)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3:0	RFT	R/W	4'h0	接收 FIFO 阈值，如果接收 FIFO 中的实体个数大于该阈值时，产生接收 FIFO 满中断

7.4.7. TXFLR (Offset 0x20)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	TXTFL	R	5'h0	发送 FIFO 中有效数据实体个数

7.4.8. RXFLR (Offset 0x24)

表格 7-2 RXFLR 寄存器

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	RXTFL	R	5'h0	接收 FIFO 中有效数据实体个数

7.4.9. SR (Offset 0x28)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5	TXE	R	1'b0	发送错误。 0: 无错误 1: 发送错误，当发送开始时，发送 FIFO 为空
4	RFF	R	1'b0	0: 接收 FIFO 非满 1: 接收 FIFO 满，
3	RFNE	R	1'b0	0: 接收 FIFO 空

				1: 接收 FIFO 非空
2	TFE	R	1'b1	0: 发送 FIFO 非空 1: 发送 FIFO 空
1	TFNF	R	1'b1	0: 发送 FIFO 满 1: 发送 FIFO 非满
0	BUSY	R	1'b0	SPI 忙标志 0: SPI 空闲或者没有使能 1: SPI 正在传输数据

7.4.10. IMR (Offset 0x2C)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIM	R/W	1'b1	接收 FIFO 满中断屏蔽。
3	RFOIM	R/W	1'b1	接收 FIFO 溢出中断屏蔽。
2	RXUIM	R/W	1'b1	接收 FIFO 下溢中断屏蔽。
1	TXOIM	R/W	1'b1	发送 FIFO 溢出中断屏蔽。
0	TXEIM	R/W	1'b1	发送 FIFO 空中断屏蔽。 0: 屏蔽 1: 不屏蔽

7.4.11. ISR (Offset 0x30)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIS	R	1'b0	接收 FIFO 满中断状态。(IMR 屏蔽后)
3	RXOIS	R	1'b0	接收 FIFO 溢出中断状态。(IMR 屏蔽后)
2	RXUIS	R	1'b0	接收 FIFO 下溢中断状态。(IMR 屏蔽后)
1	TXOIS	R	1'b0	发送 FIFO 溢出中断状态。(IMR 屏蔽后)
0	TXEIS	R	1'b0	发送 FIFO 空中断状态。(IMR 屏蔽后) 0: Inactive 1: Active

7.4.12. RISR (Offset 0x34)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	RXFIR	R	1'b0	原始接收 FIFO 满中断状态。
3	RXOIR	R	1'b0	原始接收 FIFO 溢出中断状态。
2	RXUIR	R	1'b0	原始接收 FIFO 下溢中断状态。
1	TXOIR	R	1'b0	原始发送 FIFO 溢出中断状态。
0	TXEIR	R	1'b0	原始发送 FIFO 空中断状态。 0: Inactive 1: Active

7.4.13. TXOICR (Offset 0x38)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	TXOICR	R	1'b0	发送 FIFO 溢出中断清除。 读取该寄存器清除发送 FIFO 溢出中断状态。

7.4.14. RXOICR (Offset 0x3C)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	RXOICR	R	1'b0	接收 FIFO 溢出中断清除。 读取该寄存器清除接收 FIFO 溢出中断状态。

7.4.15. RXUICR (Offset 0x40)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	RXUICR	R	1'b0	接收 FIFO 下溢中断清除。 读取该寄存器清除接收 FIFO 下溢中断状态。

7.4.16. ICR (Offset 0x48)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	ICR	R	1'b0	清中断状态 读取该寄存器会清除发送 FIFO 溢出，接收 FIFO 溢出和接收 FIFO 下溢中断状态。

7.4.17. DMACR (Offset 0x4C)

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	TDMAE	R/W	1'b0	0: 发送 DMA 不使能 1: 发送 DMA 使能
0	RDMAE	R/W	1'b0	0: 接收 DMA 不使能 1: 接收 DMA 使能

7.4.18. DMATDLR (Offset 0x50)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3:0	DMATDL	R/W	4'h0	DMA 发送数据阈值，当发送 FIFO 中的有效数据实体个数小于等于该阈值，并且 TDMAE = 1 时，产生 DMA_Tx 请求

7.4.19. DMARDLR (Offset 0x54)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3:0	DMARDL	R/W	4'h0	DMA 接收数据阈值，当接收 FIFO 中的有效数据实体个数大于该阈值，并且 RDMAE=1 时，产生 DMA_Rx 请求

7.4.20. DR(offset 0x60)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	DR	R/W	16'h00	数据寄存器，读取该寄存器时，从接收 FIFO 取值，写入该寄存器时，会写入到发送 FIFO。

7.4.21. CTRLR2 (Offset 0xF4)

比特	名称	属性	复位值	描述
31	SLV_QUAD_MODE	R/W	1'b0	SPIS1 Quad (四线) 模式使能 0: Disable 1: Enable
30	SLV_DUAL_MODE	R/W	1'b0	SPIS1 Dual (二线) 模式使能 0: Disable 1: Enable
29:0	Reserved	R	0x0	Reserved

7.5. 使用流程

SPIS 的数据传输由主设备的产生的 SCLK_IN 控制。当 CS 信号没有拉低时，SPIS 保持在空闲状态，并且需要将 TXD 线保持在高电平，避免干扰其他 SPI 从设备的数据传输。

SSOE 信号线，可以控制 TXD 的输出缓存，选择是否输出 TXD 信号。

SPI 从设备连续数据传输的软件配置流程：

1. 配置 [CTRLR0](#) (SCPH 和 SCPOL 需和主设备一致), 设置 FIFO 中断阈值 ([TXFTLR](#) 和 [RXFTLR](#)), 使能相关中断 ([IMR](#))。注: 在 `SSI_EN = 0` 时配置。
2. 使能 SPIS (`SSI_EN = 1`)
3. 如果传输模式是发送接收模式 (`TMOD = 0`) 或者仅发送模式 (`TMOD = 1`), 将数据写入发送 FIFO (写 DR); 如果传输模式是仅接收模式 (`TMOD = 2`), 不需要向发送 FIFO 填写任何数据。
4. 当传输正在进行, 可以通过查询 `BUSY` 状态位确认传输是否完成。如果产生了发送 FIFO 空中断, 就去写发送 FIFO (写 DR), 如果产生了接收 FIFO 满中断, 则从接收 FIFO 读取数据 (读 DR)。
5. 当主设备拉高 CS 信号, 或 FIFO 数据全部发送完毕, `BUSY` 状态位会被复位。

7.6. MOTOROLA SPI 时序

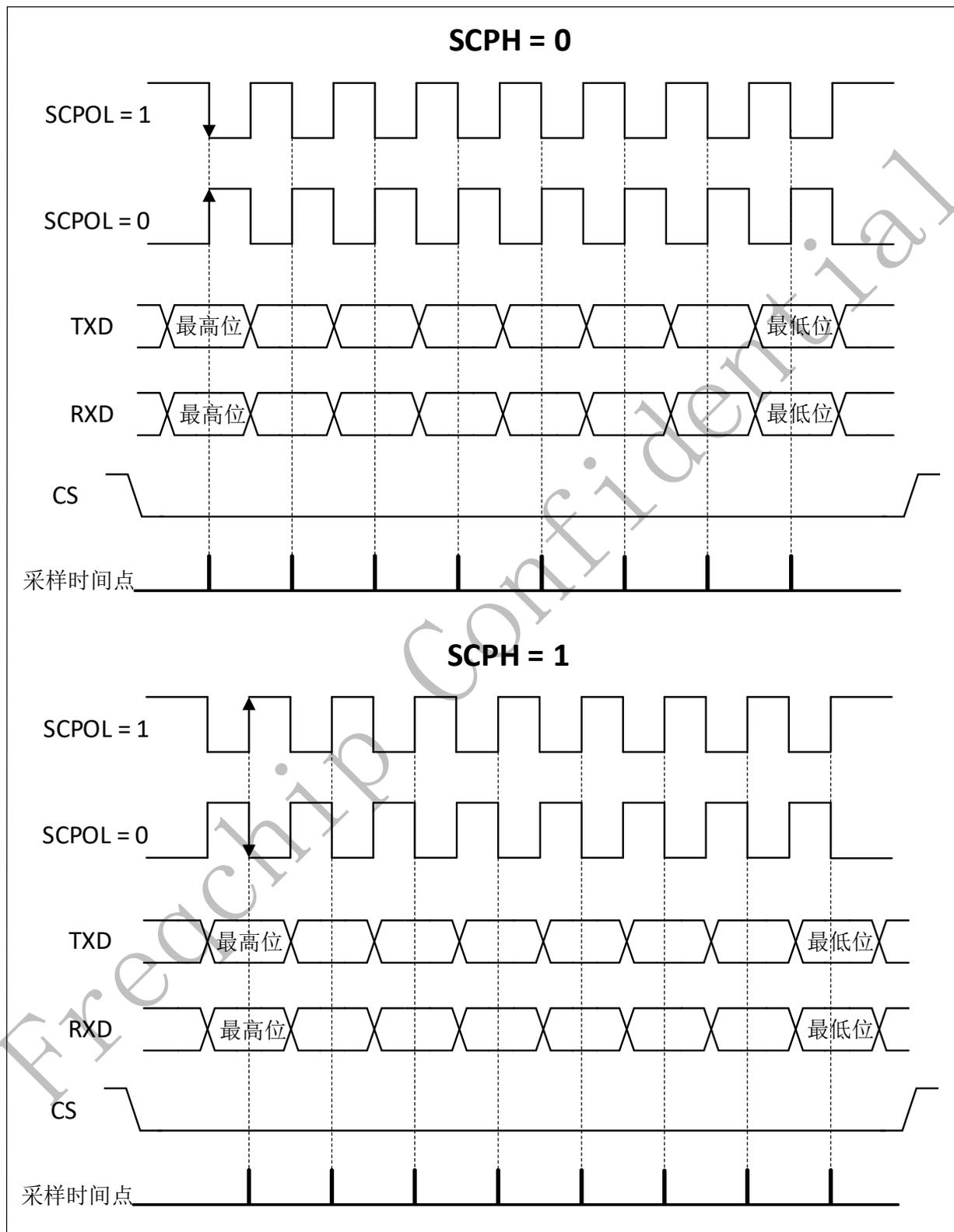


图 7-2 MOTOROLA SPI 时序图

8. 内部集成电路总线接口（I2C）

8.1. 概述

I2C 总线接口连接微控制器和串行 I2C 总线，它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁。

8.2. 主要特性

- Tx FIFO 深度 32，位宽 8bit
- Rx FIFO 深度 32，位宽 8bit
- 主机、从机拉时钟功能
- 7bit、10bit 地址可选
- SCL、SDA 低电平超时检测
- 突刺抑制功能

8.3. 功能描述

8.3.1. I2C 时序控制

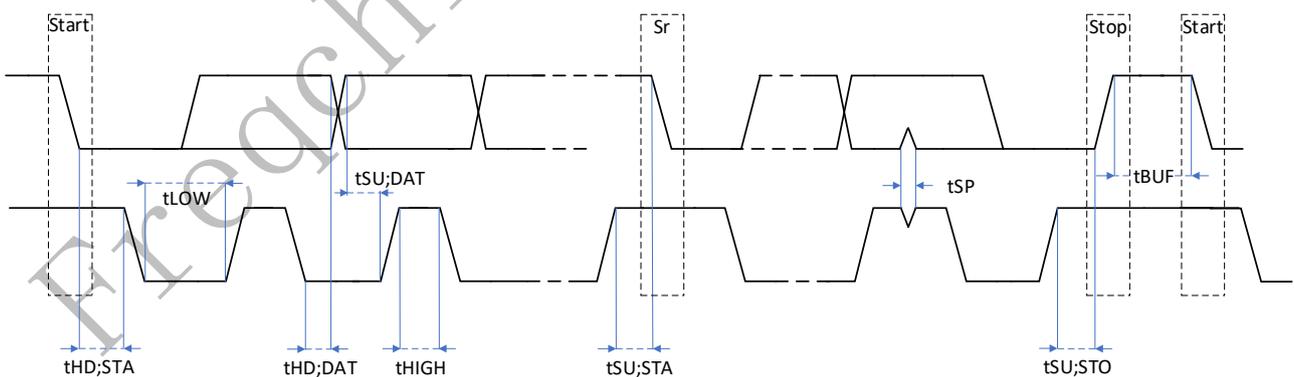


图 8-1 I2C 时序参数图

表格 8-1 I2C 时序参数与寄存器关系

参数	符号	寄存器
SCL 时钟低电平周期	tLOW	SCL_LCNT
SCL 时钟高电平周期	tHIGH	SCL_HCNT

重复起始条件的建立时间	tSU;STA	SCL_HCNT
(重复) 起始条件的保持时间	tHD;STA	SCL_HCNT
停止条件建立时间	tSU;STO	SCL_HCNT
停止和启动条件之间的总线空闲时间	tBUF	SCL_LCNT
突刺长度	tSP	SPKLEN
数据保持时间	tHD;DAT	SDA_HOLD
数据建立时间	tSU;DAT	SDA_SETUP

I2C 提供了较为灵活的时序修改方法，所涉及的寄存器如 [错误!未找到引用源。](#) 所示，所调整的时序图可参照 [错误!未找到引用源。](#)。上述寄存器调整的单位均以 APB_I2C_CLK 为单位。

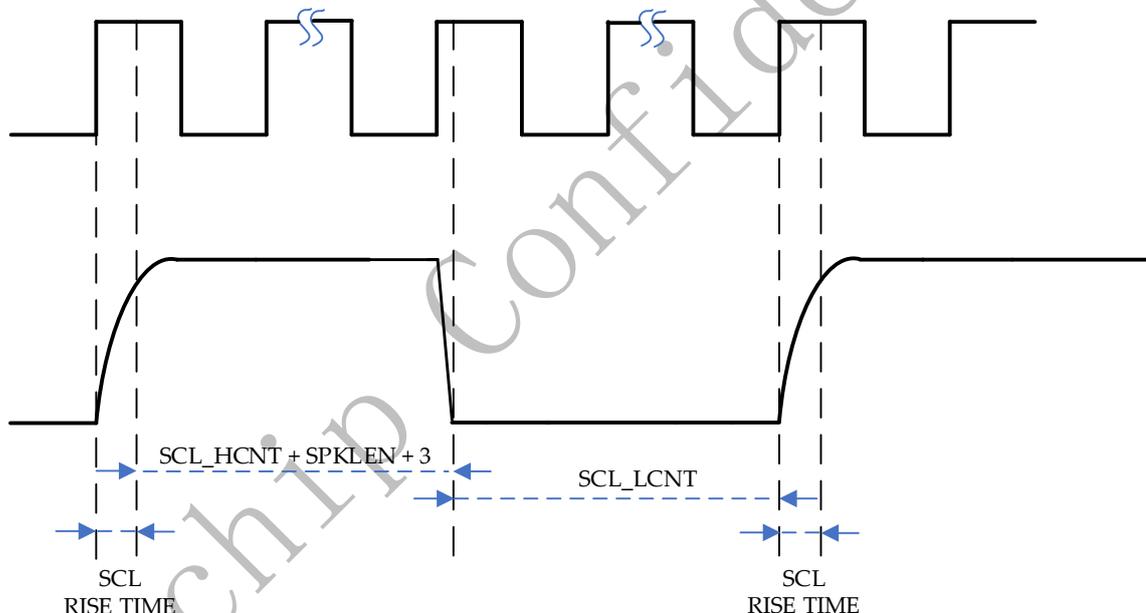


图 8-2 一个时钟周期时序图

● 一个时钟周期中：

$$\text{时钟信号高电平持续时间} = (\text{SCL_HCNT} + \text{SPKLEN} + 3) * \text{APB_I2C_CLK}$$

$$\text{时钟信号低电平持续时间} = (\text{SCL_LCNT}) * \text{APB_I2C_CLK}$$

$$\text{时钟周期} = (\text{SCL_HCNT} + \text{SPKLEN} + \text{SCL_LCNT} + 3) * \text{APB_I2C_CLK} + \text{SCL_RISE_TIME}$$

8.3.2. I2C_SDA_HOLD 功能

[SDA_HOLD](#) 寄存器中提供了 SDA_RX_HOLD[23:16]、SDA_TX_HOLD[15:0]两部分可以配置。

- 对于 SDA_TX_HOLD，将作用于 I2C 时序的 tHD，DAT 部分。

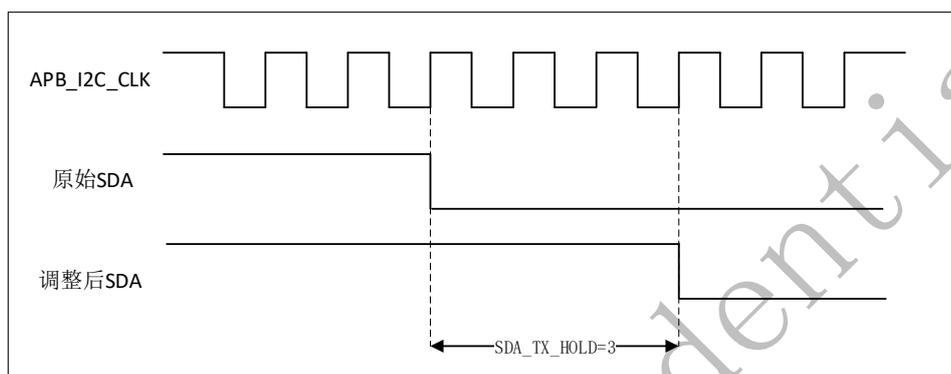


图 8-3 SDA_TX_HOLD 使用示意图

- 对于 SDA_RX_HOLD，将作用于 I2C 内部采样时间点。

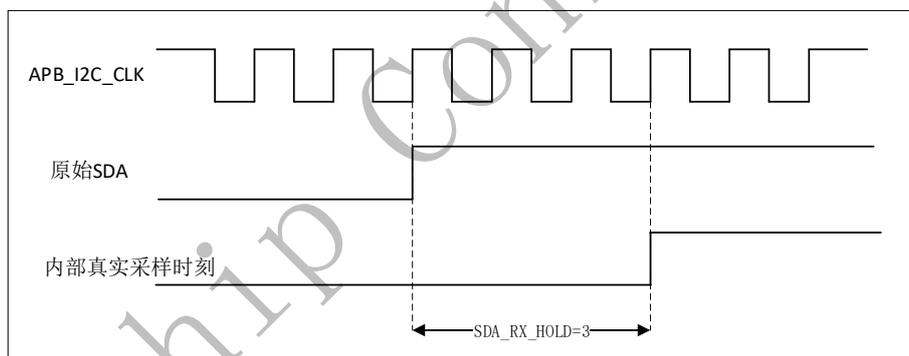


图 8-4 SDA_RX_HOLD 使用示意图

8.3.3. SCL、SDA 拉低超时功能

- SCL 线拉低超时检测功能

在通讯中，若从机因为数据未准备好，或者从机故障等原因，持续拉低 SCL 线，主机可以通过配置相关寄存器进行 SCL 拉低超时检测。

使能 [CONTROL](#) 寄存器的 BUS_DETEC_FEATURE_CTRL 位，开启检测功能。设置 [SCL STUCK AT LOW TIMEOUT](#) 寄存器配置超时时间。当检测到 SCL 被拉低超过配置时

间后，状态寄存器 [RAW_INTR_STAT](#) 的 SCL_STUCK_AT_LOW 置位，可以产生中断。此时可通过系统寄存器复位 I2C，或检测从机持续拉低原因。

■ SDA 线拉低超时检测功能

在通讯中，若从机因为故障等原因，持续拉低 SCL 线，主机可以通过配置相关寄存器进行 SDA 拉低超时检测。

使能 [CONTROL](#) 寄存器的 BUS_DETEC_FEATURE_CTRL 位，开启检测功能。设置 [SDA_STUCK_AT_LOW_TIMEOUT](#) 寄存器配置超时时间。当检测到 SDA 被拉低超过配置时间后，[TX_ABRT_SOURCE](#) 寄存器的 ABRT_SDA_STUCK_AT_LOW 置位，同时导致状态寄存器 [RAW_INTR_STAT](#) 的 TX_ABRT 置位，可以产生中断。此时若 [ENABLE](#) 寄存器的 SDA_STUCK_RECOVERY 设置为 ‘1’，则发起 SDA 恢复机制。

SDA 恢复机制：

主机发送 9 个 I2C 时钟周期试图恢复总线，如果 SDA 没有恢复，它将再次继续发送最多 9 个 SCL 时钟试图恢复总线，如果 SDA 线路在 9 个时钟脉冲内被恢复，那么主机将发送 STOP 来释放总线。

如果还没有恢复，此时可以查询 [STATUS](#) 寄存器的 SDA_STUCK_NOT_RECOVERED 位，查看 SDA 是否恢复成功。若不成功，可通过系统寄存器复位 I2C，或检测从机持续拉低原因。

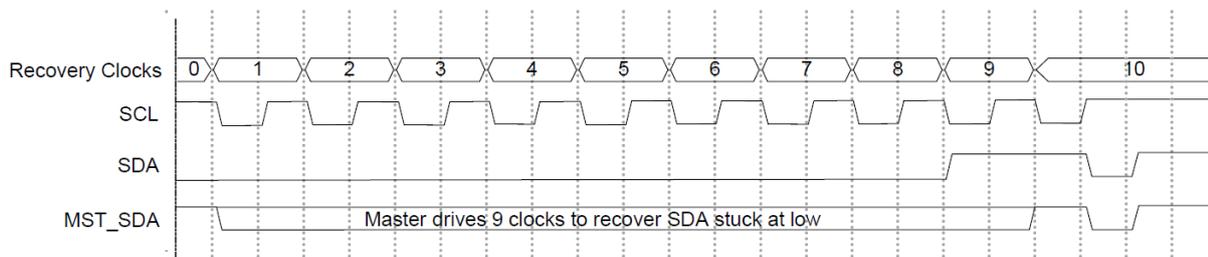


图 8-5 主机发送 9 个 CLK 试图恢复 SDA

8.3.4. 7bit 地址格式、10bit 地址格式

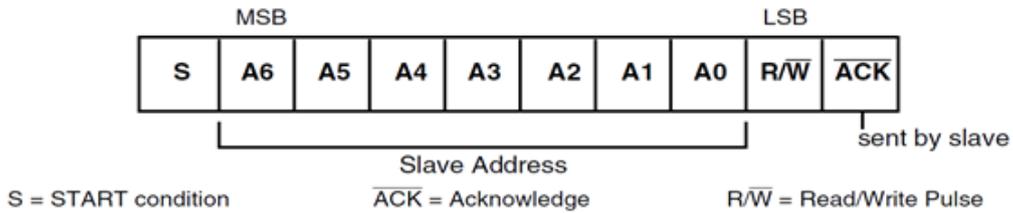


图 8-6 7bit 地址示意图

在 7 位地址格式中，如上图所示，第一个字节的前 7 位(位 7:1)为从地址，LSB 位(位 0)为 R/W 位。当位 0 (R/W)设置为 ‘0’ 时，为主机写操作。当位 0 (R/W)设置为 ‘1’ 时，为主机读操作。

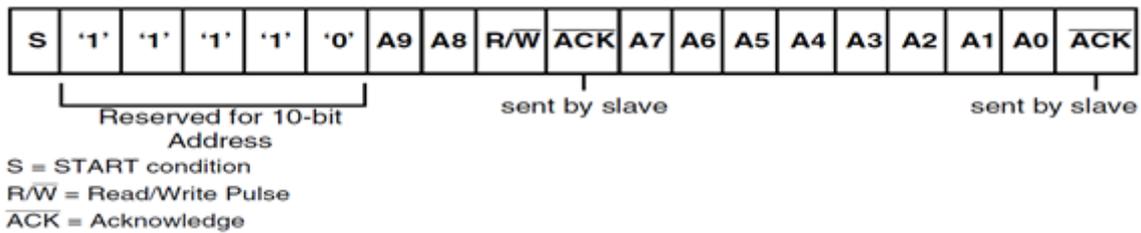


图 8-7 10bit 地址示意图

在 10 位寻址期间，传输两个字节来设置 10 位地址。第一个字节的传输包含以下位定义。前 5 位(位 7:3)通知从服务器这是一个 10 位的传输，接下来的 2 位(位 2:1)是从机地址的 9:8 位，LSB 位(位 0)是 R/W 位。传输的第二个字节是从机地址的 7:0 位。

8.3.5. 突刺抑制功能

Spike Suppression（突刺抑制功能），作用于 I2C 时序的 tSP 部分，详情参照 [I2C 时序控制](#) 章节。于此功能相关的寄存器为 [SPKLEN](#)。

该功能基于监控输入信号（SCL 和 SDA）的计数器，在内部采样之前检查它们是否在预定数量的 APB_I2C_CLK 周期内保持稳定。每个信号（SCL 和 SDA）都有一个单独的计数器。APB_I2C_CLK 周期的数量可以由用户编程。

每当其输入信号改变时，就会启动计数器。根据输入信号的行为，会出现下列情形之：

- 输入信号保持不变，直到计数器达到其计数设定值。当这种情况发生时，认为信号有效被内部采样，计数器被重置并停止。在检测到输入信号上的新变化之前，计数器不会重新启动。

- 在计数器达到其计数设定值之前，输入信号再次发生变化。当这种情况发生时，计数器被重置并停止，信号被认为无效不被内部采样。计数器保持停止状态，直到检测到输入信号的新变化

如下图所示，将 FS_SPKLEN 或 HS_SPKLEN 设置为 5 时，突刺抑制工作过程。第一次检测到 SCL 变化，开始计数，计数未满 5 次，SCL 发生变化，内部的 SCL 没有采样，并保持原状态。SCL 再次发生变化，重新计数，计数满 5 次，内部认为 SCL 有效进行采样，内部 SCL 变化。

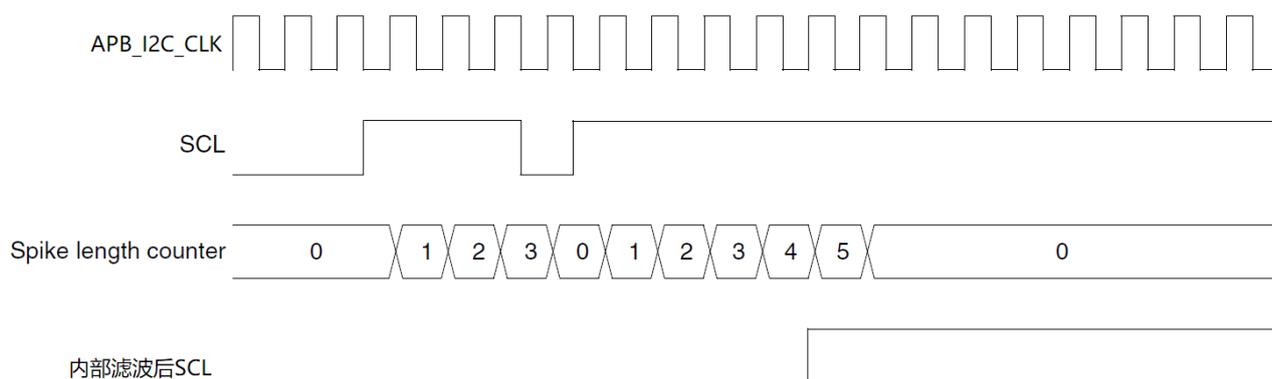


图 8-8 突刺抑制示意图

8.4. I2C 寄存器

8.4.1. I2C 寄存器映像

注：I2C 章节寄存器虽多，但并不复杂，如寄存器映像表所示，其中绿色背景的是与时序控制相关的寄存器，蓝色背景的是与状态、中断状态、状态清除相关的寄存器。其他寄存器控制功能相关。

I2C0 寄存器基地址：0x5002_0000

I2C1 寄存器基地址：0x5002_8000

I2C2 寄存器基地址：0x5012_0000

I2C3 寄存器基地址：0x5012_8000

I2C4 寄存器基地址：0x5022_0000

I2C5 寄存器基地址：0x5022_8000

表格 8-2 I2C 寄存器映像

偏移	寄存器	寄存器描述
0x00	CONTROL	I2C 控制寄存器
0x04	TAR	目标地址寄存器
0x08	SAR	从设备地址寄存器
0x10	DATA_CMD	数据、命令寄存器
0x1C	SCL_HCNT	时钟信号高电平计数
0x20	SCL_LCNT	时钟信号低电平计数
0x2C	INTR_STAT	中断状态寄存器
0x30	INTR_MASK	中断屏蔽寄存器
0x34	RAW_INTR_STAT	原始中断寄存器
0x38	RX_TL	Rx FIFO 阈值
0x3C	TX_TL	Tx FIFO 阈值
0x40	CLR_INTR	中断清除寄存器
0x44	CLR_RX_UNDER	RX_UNDER 状态清除
0x48	CLR_RX_OVER	RX_OVER 状态清除
0x4C	CLR_TX_OVER	TX_OVER 状态清除
0x50	CLR_RD_REQ	RD_REQ 状态清除
0x54	CLR_TX_ABRT	TX_ABRT 状态清除
0x58	CLR_RX_DONE	RX_DONE 状态清除
0x5C	CLR_ACTIVITY	ACTIVITY 状态清除
0x60	CLR_STOP_DET	STOP_DET 状态清除
0x64	CLR_START_DET	START_DET 状态清除
0x6C	ENABLE	I2C 使能寄存器
0x70	STATUS	传输状态寄存器
0x74	TXFLR	Tx FIFO 有效数据个数
0x78	RXFLR	Rx FIFO 有效数据个数
0x7C	SDA_HOLD	SDA 状态保持寄存器
0x80	TX_ABRT_SOURCE	Tx 中止传输原因寄存器
0x84	SLV_DATA_NACK_ONLY	从模式接收时，NAK 控制
0x88	DMA_CR	DMA 控制寄存器
0x8C	DMA_TDLR	DMA Tx 阈值寄存器
0x90	DMA_RDLR	DMA Rx 阈值寄存器
0x94	SDA_SETUP	SDA 配置寄存器

0xA0	SPKLEN	峰值抑制寄存器
0xA8	CLR_RESTART_DET	RESTART_DET 状态清除
0xAC	SCL_STUCK_AT_LOW_TIMEOUT	SCL 拉低超时时间寄存器
0xB0	SDA_STUCK_AT_LOW_TIMEOUT	SDA 拉低超时时间寄存器
0xB4	CLR_SCL_STUCK_DET	SCL_STUCK_AT_LOW 状态清除

8.4.2. CONTROL (Offset 0x00)

比特	名称	属性	复位值	描述
31:12	Reserved	R	0x0	Reserved
11	BUS_DETEC_FEATURE_CTRL	R/W	1'b0	主模式下： 0: 总线检测功能 Disable 1: 总线检测功能 Enable 检测 SCL 或 SDA 是否被长时间拉低。
10	STOP_DET_IF_MASTER_ACTIVE	R/W	1'b0	主模式下： 0: 不论总线空闲或忙碌，Master 都可以发出 STOP_DET 中断。 1: 只有总线忙碌时，Master 才发出 STOP_DET 中断。
9	RX_FIFO_FULL_HLD_CTRL	R/W	1'b0	Rx FIFO 满时是否保持总线状态。 0: Rx FIFO 满时不保持总线状态 1: Rx FIFO 满时保持总线状态
8	TX_EMPTY_CTRL	R/W	1'b0	TX_EMPTY 中断是否产生。 0: 不产生 TX_EMPTY 中断 1: 产生 TX_EMPTY 中断
7	STOP_DET_IFADDRESSED	R/W	1'b0	从模式下： 0: 从机总是能发出 STOP_DET 中断 1: 从机只有在地址匹配后才能发出 STOP_DET 中断
6	SLAVE_DISABLE	R/W	1'b1	从模式禁用。 0: 从模式不禁用 1: 从模式禁用 注: SLAVE_DISABLE = 0 时，

				MASTER_MODE 应设置为 0。
5	RESTART_EN	R/W	1'b1	主模式下 RESTART 信号是否使用。 0: 主机不可以发送 RESTART 信号 1: 主机可以发送 RESTART 信号
4	Reserved	R/W	1'b0	Reserved
3	10BITADDR_SLAVE	R/W	1'b1	从模式地址选择。 0: 7bit 地址 1: 10bit 地址
2:1	SPEED	R/W	2'b10	10: Fast (400 kbit/s <= Speed <= 1000 kbit/s)
0	MASTER_MODE	R/W	1'b1	主模式使能。 0: Disable 1: Enable 注: MASTER_MODE = 1 时, SLAVE_DISABLE 应设置为 1。

8.4.3. TAR (Offset 0x04)

比特	名称	属性	复位值	描述
31:14	Reserved	R	0x0	Reserved
13	DEVICE_ID	R/W	1'b0	0: 执行 TAR[9:0]目标地址。 1: 不执行 TAR[9:0]目标地址。
12	10BITADDR_MASTER	R/W	1'b1	主模式寻址方式。 0: 7bit 地址寻址 1: 10bit 地址寻址
11	SPECIAL	R/W	1'b0	允许发送目标地址。 0: 允许发送目标地址 1: 不允许发送目标地址
10	Reserved	R	0x0	Reserved
9:0	TAR	R/W	10'h55	目标地址

8.4.4. SAR (Offset 0x08)

比特	名称	属性	复位值	描述
31:10	Reserved	R	0x0	Reserved
9:0	SAR	R/W	10'h00	从设备地址。

8.4.5. DATA_CMD (Offset 0x10)

比特	名称	属性	复位值	描述
31:12	Reserved	R	0x0	Reserved
11	FIRST_DATA_BYTE	R	1'b0	指示主模式接收发送或从模式接收到地址后接收的第一个字节。 0: 接收到地址后的第一个字节 1: 未接收到地址后的第一个字节
10	RESTART	W	1'b0	Restart 信号控制。 0: 执行本次操作前不发 Restart 信号 1: 执行本次操作前发 Restart 信号
9	STOP	W	1'b0	Stop 信号控制。 0: 执行本次操作后不发 Stop 信号 1: 执行本次操作后发 Stop 信号
8	CMD	W	1'b0	写操作, 读操作选择。 0: 写操作 1: 读操作
7:0	DAT	R/W	8'h00	收到的数据从此处读出。 发送的数据从此处写入。

8.4.6. SCL_HCNT (Offset 0x1C)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	SCL_HCNT	R/W	16'h3C	时钟信号高电平计数。最小值不应低于 6，单位为 APB_I2C_CLK。 注：在 I2C 未使能时配置 (ENABLE = 0)，否则无效。

使用详情参照 [I2C 时序控制](#) 章节。

8.4.7. SCL_LCNT (Offset 0x20)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	SCL_LCNT	R/W	16'h82	时钟信号低电平计数。最小值不应低于 8，单位为 APB_I2C_CLK。 注：在 I2C 未使能时配置 (ENABLE = 0)，否则无效。

使用详情参照 [I2C 时序控制](#) 章节。

8.4.8. INTR_STAT (Offset 0x2C)

比特	名称	属性	复位值	描述
31:15	Reserved	R	0x0	Reserved
14	SCL_STUCK_AT_LOW	R	1'b0	SCL 信号线的低电平时间超过 SCL_STUCK_LOW_TIMEOUT 寄存器设置值时，SCL_STUCK_AT_LOW 状态置位。 0: Status inactive 1: Status active
13	MASTER_ON_HOLD	R	1'b0	主机模式下。 主机发送时，TX FIFO 为空，主机拉住总线，MASTER_ON_HOLD 状态置位 0: Status inactive 1: Status active

12	RESTART_DET	R	1'b0	Slave 模式下，当地址匹配后，总线上检测到了 RESTART 信号。 0: Status inactive 1: Status active
11	Reserved	R	1'b0	Reserved
10	START_DET	R	1'b0	检测总线上的 START 或 RESTART 信号。 0: Status inactive 1: Status active
9	STOP_DET	R	1'b0	检测到 STOP 信号。 Slave 模式下： 若 STOP_DET_IFADDRESSED = 1，则只有在地址匹配后才能检查到 STOP 信号。 若 STOP_DET_IFADDRESSED = 0，则不需检测地址匹配就能检查到 STOP 信号。 Master 模式下： STOP_DET_IF_MASTER_ACTIVE = 1 时，只有总线忙碌时，Master 才发出 STOP_DET 中断。 STOP_DET_IF_MASTER_ACTIVE = 0 时，不论总线空闲或忙碌，Master 都可以发出 STOP_DET 中断。 0: Status inactive 1: Status active
8	ACTIVITY	R	1'b0	总线忙碌时 ACTIVITY 置位（主机或从机在收发数据） 0: Status inactive 1: Status active ----- 三种清除方式： 1、读 CLR_ACTIVITY 寄存器 2、读 CLR_INTR 寄存器 3、I2C Disable 注：状态位将一直保持置位，除非清除。

7	RX_DONE		1'b0	<p>从机发送时，在从机发送完数据后，主机回应 NAK，RX_DONE 置位。</p> <p>一般用在一次传输的最后一个字节，表示传输结束。</p> <p>0: Status inactive 1: Status active</p>
6	TX_ABRT	R	1'b0	<p>主机或从机发送时，没有按照预期设置发送完成，发送中止，TX_ABRT 置位。</p> <p>TX_ABRT_SOURCE 寄存器中可以查询发送中止的原因。</p> <p>0: Status inactive 1: Status active</p>
5	RD_REQ	R	1'b0	<p>作从机时，当收到主机的读请求时 RD_REQ 置位，并拉低 SCL。当检测到此状态位置位后，应将需要发送的数据写入 DATA_CMD 寄存器，并清除此位后，控制器释放 SCL，并发送数据。</p> <p>0: Status inactive 1: Status active</p>
4	TX_EMPTY	R	1'b0	<p>CONTROL.TX_EMPTY_CTRL = 0 时： 当 Tx FIFO 中的数据，小于 TX_TL 寄存器设置的阈值时，TX_EMPTY 置位。</p> <p>CONTROL.TX_EMPTY_CTRL = 1 时： 当 Tx FIFO 中的数据，小于 TX_TL 寄存器设置的阈值，并且从移位寄存器中传输完当前字节时 TX_EMPTY 置位。</p> <p>0: Status inactive 1: Status active</p> <p>注：此位会在 Tx FIFO 中数据超过阈值时自动清除。</p>
3	TX_OVER	R	1'b0	<p>当 Tx FIFO 中已经存在数据时，CPU 尝试通过写入 DATA_CMD 寄存器发出另一个 I2C 命令，则 TX_OVER 置位。</p> <p>0: Status inactive 1: Status active</p>

2	RX_FULL	R	1'b0	<p>当 Rx FIFO 中的数据，大于或等于 RX_TL 寄存器设置的阈值时，TX_EMPTY 置位。</p> <p>注：此位会在 Rx FIFO 中数据小于阈值时自动清除。</p> <p>0: Status inactive 1: Status active</p>
1	RX_OVER	R	1'b0	<p>当 Rx FIFO 已满，并接收到了新的数据，RX_OVER 置位。</p> <p>0: Status inactive 1: Status active</p> <p>注：若 RX_FIFO_FULL_HLD_CTRL = 1，则在 Rx FIFO 满后，拉住 SCL 信号线，不在会在收到新的数据。</p> <p>若 RX_FIFO_FULL_HLD_CTRL = 0，RxFIFO 满后,将会继续接收数据，接收到的数据都将丢失。</p>
0	RX_UNDER	R	1'b0	<p>Rx FIFO 空时，继续读取 Rx FIFO，RX_UNDER 状态置位。</p>

8.4.9. INTR_MASK (Offset 0x30)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
14	SCL_STUCK_AT_LOW	R/W	1'b1	<p>中断状态屏蔽。</p> <p>0: 相应位中断状态屏蔽。 1: 相应位中断状态不屏蔽。</p> <p>注：屏蔽后，INTR_STAT 寄存器中相应位，将会被屏蔽，不会置位。</p>
13	MASTER_ON_HOLD	R/W	1'b0	
12	RESTART_DET	R/W	1'b0	
11	Reserved	R/W	1'b0	
10	START_DET	R/W	1'b0	
9	STOP_DET	R/W	1'b0	
8	ACTIVITY	R/W	1'b0	
7	RX_DONE	R/W	1'b1	
6	TX_ABRT	R/W	1'b1	
5	RD_REQ	R/W	1'b1	
4	TX_EMPTY	R/W	1'b1	

3	TX_OVER	R/W	1'b1	
2	RX_FULL	R/W	1'b1	
1	RX_OVER	R/W	1'b1	
0	RX_UNDER	R/W	1'b1	

8.4.10. RAW_INTR_STAT (Offset 0x34)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
14	SCL_STUCK_AT_LOW	R	1'b0	中断原始状态寄存器。 0: 相应状态位未置位。 1: 相应状态位置位。 注: 原始状态寄存器, 不受 INTR_STAT 寄存器影响。
13	MASTER_ON_HOLD	R	1'b0	
12	RESTART_DET	R	1'b0	
11	Reserved	R	1'b0	
10	START_DET	R	1'b0	
9	STOP_DET	R	1'b0	
8	ACTIVITY	R	1'b0	
7	RX_DONE	R	1'b0	
6	TX_ABRT	R	1'b0	
5	RD_REQ	R	1'b0	
4	TX_EMPTY	R	1'b0	
3	TX_OVER	R	1'b0	
2	RX_FULL	R	1'b0	
1	RX_OVER	R	1'b0	
0	RX_UNDER	R	1'b0	

8.4.11. RX_TL (Offset 0x38)

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:0	RX_TL	R/W	8'h00	Rx FIFO 阈值

8.4.12. TX_TL (Offset 0x3C)

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:0	TX_TL	R/W	8'h00	Tx FIFO 阈值

8.4.13. CLR_INTR (Offset 0x40)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_INTR	R	1'b0	读取该寄存器以清除组合中断、所有单独中断和 TX_ABRT_SOURCE 寄存器。但不清除硬件可清除中断，只清除软件可清除中断。

8.4.14. CLR_RX_UNDER (Offset 0x44)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_RX_UNDER	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 RX_UNDER 中断。

8.4.15. CLR_RX_OVER (Offset 0x48)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_RX_OVER	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 RX_OVER 中断。

8.4.16. CLR_TX_OVER (Offset 0x4C)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_TX_OVER	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 TX_OVER 中断。

8.4.17. CLR_RD_REQ (Offset 0x50)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_RD_REQ	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 RD_REQ 中断。

8.4.18. CLR_TX_ABRT (Offset 0x54)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_TX_ABRT	R	1'b0	读取这个寄存器以清除 RAW_INTR_STAT 寄存器的 TX_ABRT 中断和 TX_ABRT_SOURCE 寄存器。

8.4.19. CLR_RX_DONE (Offset 0x58)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_RX_DONE	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 RX_DONE 中断。

8.4.20. CLR_ACTIVITY (Offset 0x5C)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_ACTIVITY	R	1'b0	如果 I2C 不再处于活动状态，读取这个寄存器将清除 ACTIVITY 中断。如果总线上的 I2C 模块仍然处于活动状态，则继续设置 ACTIVITY 中断位。如果模块被禁用，并且总线上没有进一步的活动，它将被硬件自动清除。

8.4.21. CLR_STOP_DET (Offset 0x60)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_STOP_DET	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 STOP_DET 中断。

8.4.22. CLR_START_DET (Offset 0x64)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_START_DET	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 START_DET 中断。

8.4.23. ENABLE (Offset 0x6C)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3	SDA_STUCK_RECOVERY	R/W	1'b0	当检测到 SDA 线拉低超时， (TX_ABRT_SOURCE 寄存器中的 ABRT_SDA_STUCK_AT_LOW 置位。) 0: Master 禁用 SDA 恢复机制 1: Master 发起 SDA 恢复机制
2	TX_CMD_BLOCK	R/W	1'b0	主机模式下： 0: 一旦 Tx FIFO 中有数据存在，I2C 总线就开始传输开始 1: 阻塞数据在 I2C 总线上的传输，即使 Tx FIFO 有数据要传输。 注：只有在 FIFO 空时，设置有效。
1	ABORT	R/W	1'b0	主模式下中止传输。 设置此位后，主机发出 STOP 并在完成当前传输后刷新 Tx FIFO，并产生 TX_ABORT 中断状态。完成中止操作后硬件自己清除此位。 0: 正常传输 1: 中止传输 注：只有设置 ENABLE = 1 时，设置此位有效。
0	ENABLE	R/W	1'b0	I2C 使能。 0: I2C Disable 1: I2C Enable

8.4.24. STATUS (Offset 0x70)

比特	名称	属性	复位值	描述
31:12	Reserved	R	0x0	Reserved
11	SDA_STUCK_NOT_RECOVERED	R	1'b0	主模式下，恢复机制启动后，SDA 状态 0: 恢复机制启动后，SDA 未恢复 1: 恢复机制启动后，SDA 恢复
10	SLV_HOLD_RX_FIFO_FULL	R	1'b0	从机 Bus Hold 状态。 0: 从机未拉住总线 1: 从机 Rx FIFO 满，拉住总线
9	SLV_HOLD_TX_FIFO_EMPTY	R	1'b0	从机 Bus Hold 状态。 0: 从机未拉住总线 1: 从机 Tx FIFO 空，拉住总线
8	MST_HOLD_RX_FIFO_FULL	R	1'b0	主机 Bus Hold 状态。 0: 主机未拉住总线 1: 主机 Rx FIFO 满，拉住总线
7	MST_HOLD_TX_FIFO_EMPTY	R	1'b0	主机 Bus Hold 状态。 0: 主机未拉住总线 1: 主机 Tx FIFO 空，拉住总线
6	SLV_ACTIVITY	R	1'b0	从机状态。 0: 从机空闲 1: 从机正忙
5	MST_ACTIVITY	R	1'b0	主机状态。 0: 主机空闲 1: 主机正忙
4	RFF	R	1'b0	Rx FIFO 满状态。 0: Rx FIFO 不满 1: Rx FIFO 满
3	RFNE	R	1'b0	Rx FIFO 空状态。 0: Rx FIFO 空 1: Rx FIFO 非空
2	TFE	R	1'b0	Tx FIFO 空状态。

比特	名称	属性	复位值	描述
				0: Tx FIFO 非空 1: Tx FIFO 空
1	TFNF	R	1'b0	Tx FIFO 满状态。 0: Tx FIFO 满 1: Tx FIFO 不满
0	ACTIVITY	R	1'b0	I2C 状态。 0: I2C 空闲 1: I2C 正忙

8.4.25. TXFLR (Offset 0x74)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	TXFLR	R	6'h0	Tx FIFO 中有效数据个数。

8.4.26. RXFLR (Offset 0x78)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	RXFLR	R	6'h0	Rx FIFO 中有效数据个数。

8.4.27. SDA_HOLD (Offset 0x7C)

比特	名称	属性	复位值	描述
31:24	Reserved	R	0x0	Reserved
23:16	SDA_RX_HOLD	R/W	8'h0	接收时, SDA 的保持时间
15:0	SDA_TX_HOLD	R/W	16'h1	发送时, SDA 的保持时间

使用详情参照 [I2C SDA_HOLD 功能](#) 章节。

8.4.28. TX_ABRT_SOURCE (Offset 0x80)

比特	名称	属性	复位值	描述
31:23	TX_FLUSH_CNT	R	9'h0	发送被中止后, Tx FIFO 中残留的数据量。
22:21	Reserved	R	2'b00	Reserved
20	ABRT_DEVICE_WRITE	R	1'b0	主模式下。 0: 未中止 1: 主机发送过程中, 修改了 DATA_CMD 的 CMD 位, 导致方向变化, 产生错误, 中止传输。
19	ABRT_DEVICE_SLVADDR_NOACK	R	1'b0	主模式下, 地址未匹配导致中止。 0: 未中止 1: 主机发送地址, 收到 NAK, 中止传输
18	Reserved	R	1'b0	Reserved
17	ABRT_SDA_STUCK_AT_LOW	R	1'b0	主模式下, 检测到 SDA 持续拉低的时间超过 SDA_STUCK_AT_LOW_TIMEOUT 寄存器配置的时间。 0: 未中止 1: SDA 拉低超时
16	ABRT_USER_ABRT	R	1'b0	主模式下, 检测到中止传输 0: 未检测到中止传输 1: 检测到中止传输
15	ABRT_SLVRD_INTX	R	1'b0	0: 未中止 1: 从机发送时, 向 DATA_CMD 寄存器的 CMD 位写了 '1'。
14	ABRT_SLV_ARBLOST	R	1'b0	0: 未中止 1: 从机发送时, 无法控制总线。
13	ABRT_SLVFLUSH_TXFIFO	R	1'b0	0: 未中止 1: 从机收到读请求时, TxFIFO 中有数据残留。
12	ARB_LOST	R	1'b0	0: 未中止 1: 主机发送或从机发送时仲裁丢失

11	ABRT_MASTER_DIS	R	1'b0	0: 未中止 1: 主模式被禁用的情况下, 启动主模式操作。
10	ABRT_10B_RD_NORSTRT	R	1'b0	0: 未中止 1: RESTART_EN = 0 时, 主机使用 10bit 地址进行读操作。
9	Reserved	R	1'b0	Reserved
8	ABRT_HS_NORSTRT	R	1'b0	0: 未中止 1: RESTART_EN = 0 时, 主机试图切换 HS 高速模式。
7	Reserved	R	1'b0	Reserved
6	ABRT_SBYTE_ACKDET	R	1'b0	0: 未中止 1: HS 模式下, 主机码被响应 ACK
5:4	Reserved	R	2'b00	Reserved
3	ABRT_TXDATA_NOACK	R	1'b0	0: 未中止 1: 主模式下, 数据响应 NAK
2	ABRT_10ADDR2_NOACK	R	1'b0	0: 未中止 1: 主模式下, 使用 10bit 地址模式, 地址的第二个字节响应 NAK
1	ABRT_10ADDR1_NOACK	R	1'b0	0: 未中止 1: 主模式下, 使用 10bit 地址模式, 地址的第一个字节响应 NAK
0	ABRT_7ADDR_NOACK	R	1'b0	0: 未中止 1: 主模式下, 使用 7bit 地址模式, 地址响应 NAK

8.4.29. SLV_DATA_NACK_ONLY (Offset 0x84)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	NACK	R	1'b0	从模式接收时, NAK 控制。 0: 产生正常的 ACK/NAK 1: 在接收时产生 NAK, 收到的数据将不会存入 Rx FIFO

8.4.30. DMA_CR (Offset 0x88)

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	TDMAE	R/W	1'b0	DMA 发送使能 0: DMA Transmit Enable 1: DMA Transmit Disable
0	RDMAE	R/W	1'b0	DMA 接收使能 0: DMA Receive Enable 1: DMA Receive Disable

8.4.31. DMA_TDLR (Offset 0x8C)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	DMATDL	R/W	6'h00	DMA 请求阈值。 当 FIFO 中的数据等于或低于 DMATDL 时产生 DMA Tx 请求

8.4.32. DMA_RDLR (Offset 0x90)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	DMARDL	R/W	6'h00	DMA 请求阈值。 当 Rx FIFO 中的数据等于或大于 DMATDL + 1 时产生 DMA Rx 请求

8.4.33. SDA_SETUP (Offset 0x94)

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:0	SDA_SETUP	R/W	8'h64	<p>最小值不应低于 8，单位为 APB_I2C_CLK。参照 I2C 时序的 tSU:DAT 部分，详情参照 I2C 时序控制 章节。</p> <p>注：在 I2C 未使能时配置 (ENABLE = 0)，否则无效。</p>

8.4.34. SPKLEN (Offset 0xA0)

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:0	SPKLEN	R/W	8'h5	<p>突刺抑制逻辑过滤掉的最长峰值。最小值应不小于 1。</p> <p>参照 I2C 时序的 tSP 部分，详情参照 I2C 时序控制 章节。</p> <p>注：在 I2C 未使能时配置 (ENABLE = 0)，否则无效。</p>

8.4.35. CLR_RESTART_DET (Offset 0xA8)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_RESTART_DET	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 RESTART_DET 中断。

8.4.36. SCL_STUCK_AT_LOW_TIMEOUT (Offset 0xAC)

比特	名称	属性	复位值	描述
31:0	SCL_STUCK_LOW_TIMEOUT	R/W	0xFFFFFFFF	SCL 拉低超时时间配置，超过配置时间， SCL STUCK AT LOW 状态置位。

8.4.37. SDA_STUCK_AT_LOW_TIMEOUT (Offset 0xB0)

比特	名称	属性	复位值	描述
31:0	SDA_STUCK_LOW_TIMEOUT	R/W	0xFFFFFFFF	SDA 拉低超时时间配置，超过配置时间。 ABRT SDA STUCK AT LOW 状态置位。

8.4.38. CLR_SCL_STUCK_DET (Offset 0xB4)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CLR_SCL_STUCK_DET	R	1'b0	读取这个寄存器来清除 RAW_INTR_STAT 寄存器的 SCL_STUCK_AT_LOW 中断。

8.5. 使用流程

8.5.1. 主机使用流程

1. 通过将 ENABLE 寄存器的[0]位写 '0' 来禁用 I2C。
2. 写入 CONTROL 寄存器的 SPEED 配置速度模式，设置 10BITADDR_SLAVE，选择从机 7bit 或 10bit 寻址模式。
3. 向 TAR 寄存器写入要寻址的 I2C 从设备地址。设置 10BITADDR_MASTER，选择主机 7bit 或 10bit 寻址模式。

4. ENABLE 寄存器的[0]位写 ‘1’ 使能 I2C。
5. 写 DATA_CMD 寄存器的 CMD、DAT。此时 I2C 控制器会生成 START 条件和地址字节。若 TX FIFO 中还存在数据，I2C 将继续传输。

注意事项：

- 主机写操作时。向 DATA_CMD 的 DAT 写入需要发送的数据，并同时写入 CMD、RESTART、STOP 需要的状态一并写入。在传输过程中要保持 CMD 为 ‘0’ 不变；需要 STOP 或 RESTART 条件时，在写入相应字节时，同时对 STOP 或 RESTART 写 ‘1’。
(注：每个字节的发送，都是写 DATA_CMD 整个寄存器，而不是只写 DAT)
- 主机读操作时。向 DATA_CMD 写入时，I2C 控制器将会自动忽略 DAT 部分，只保留了 CMD、RESTART、STOP（统称为 command），需要接收多少数据，就要向 Tx FIFO 中写入多少个 command。
(注：每个字节的接收，都是向 DATA_CM 写入 command，然后控制器自动存入 Tx FIFO，需要接收多少数据，就要写多少个 command)

注：若中断未开启，可使用查询 RAW_INTR_STAT 状态寄存器的方法。

8.5.2. 从机使用流程

1. 通过将 ENABLE 寄存器的[0]位写 ‘0’ 来禁用 I2C。
2. 向 SAR 寄存器写入从设备地址。
3. 写入 CONTROL 寄存器的 10BITADDR_SLAVE，选择从机 7bit 或 10bit 寻址模式。向 MASTER_MODE 写 ‘0’ 禁用主机，向 SLAVE_DISABLE 写 ‘0’ 使能从机，
4. ENABLE 寄存器的[0]位写 ‘1’ 使能 I2C。

注：若中断未开启，可使用查询 RAW_INTR_STAT 状态寄存器的方法。

8.5.2.1. 从机发送数据流程

1. 主设备向从机发出地址查询。
2. 从机匹配地址，并判断数据方向，判断为主机 Read 请求时，RD_REQ 中断状态，并拉低 SCL 线。直到准备好数据发送。

3. 如果在 RD_REQ 中断置位后，TxFIFO 中还有残留的数据，那么将会触发 TX_ABRT 中断，并自动清除 TxFIFO 中残留的数据。
4. 向 DATA_CMD 寄存器中写入需要发送的数据，并将 CMD 置 ‘0’。
5. 清除 RD_REQ、TX_ABRT 中断。
6. 从机释放 SCL，并发送数据。

注：若中断未开启，可使用查询 RAW_INTR_STAT 状态寄存器的方法。

8.5.2.2. 从机接收数据流程

1. 主设备向从机发出地址查询。
2. 从机匹配地址，并判断数据方向，判断为主机 Write 请求，从机开始接收数据，将数据存入 RxFIFO。
3. 接收数据到达阈值，从机 Rx_FULL 中断置位。
4. 从 DATA_CMD 寄存器读出数据。

注：若中断未开启，可使用查询 RAW_INTR_STAT 状态寄存器的方法。

8.5.3. 工作时序图

Start、Stop、Restart 通过 [DATA_CMD](#) 寄存器控制。



图 8-9 DATA_CMD 寄存器示意图

- DATA: 收到的数据从该字段读取，将要发送的数据写入该字段
- CMD: 这个位决定了要执行的传输是读 (CMD=1) 还是写 (CMD=0)
- Stop: 该位决定在发送或接收数据字节后是否生成 STOP
- Restart: 该位决定了在发送或接收数据字节之前是否生成 RESTART(或者在重启功能未使能的情况下，STOP 后跟 START)

1. 主机发送，并产生 STOP 时序

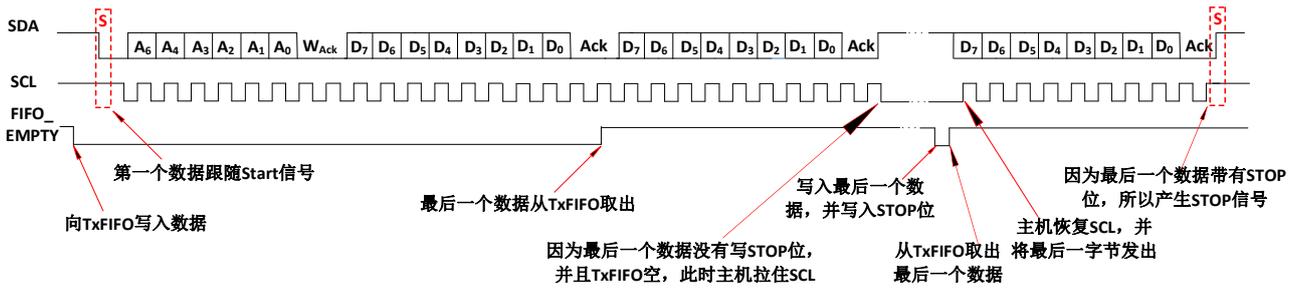


图 8-10 主机发送并产生 STOP

2. 主机接收，并产生 STOP 时序

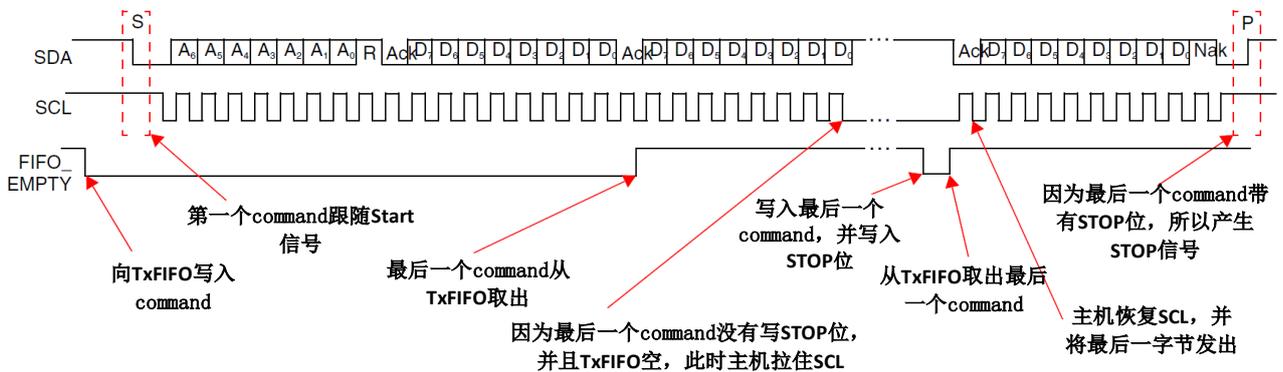


图 8-11 主机接收并产生 STOP 时序

如下两图所示，用户可以控制在 I2C 总线上产生 RESTART 条件。如果 DATA_CMD 寄存器的位 10（Restart）被设置，并且 RESTART_EN=1，一个 Restart 将在下一次操作前生成。如果 RESTART_EN=0，则会生成一个 STOP 后跟着一个 START 来代替 restart。

3. 主机发送时，并产生 Restart 时序

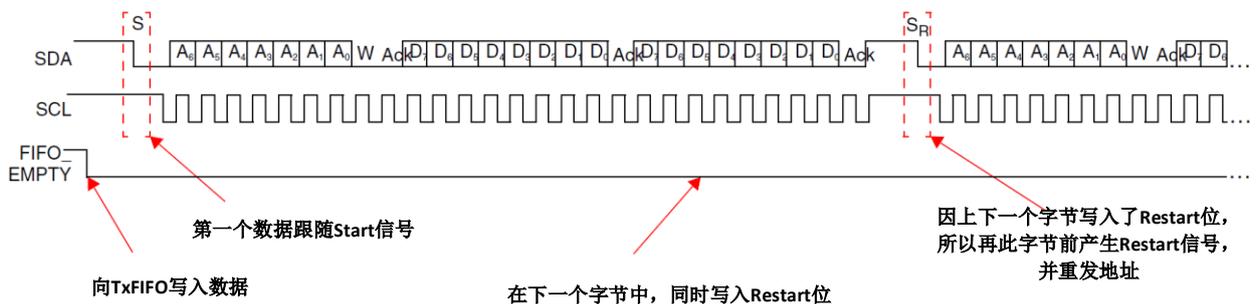


图 8-12 主机发送并产生 Restart 时序

4. 主机接收时，并产生 Restart 时序

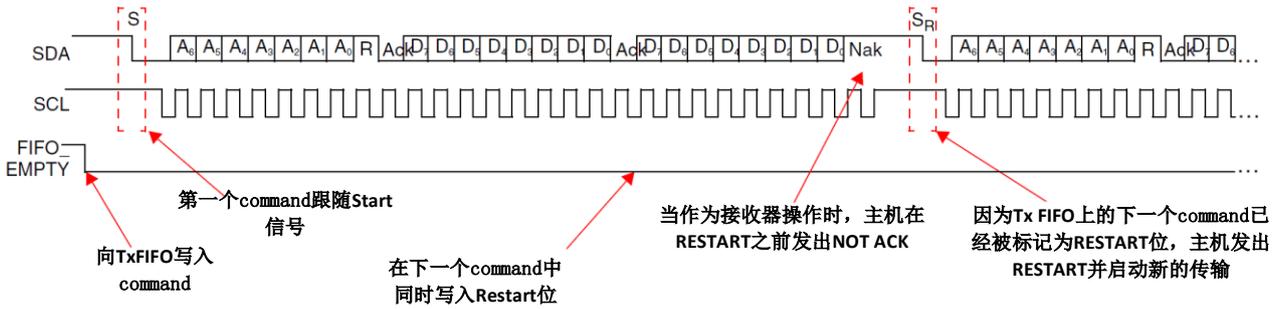


图 8-13 主机接收并产生 Restart 时序

5. 主机发送时, 产生 STOP, 但 Tx FIFO 非空

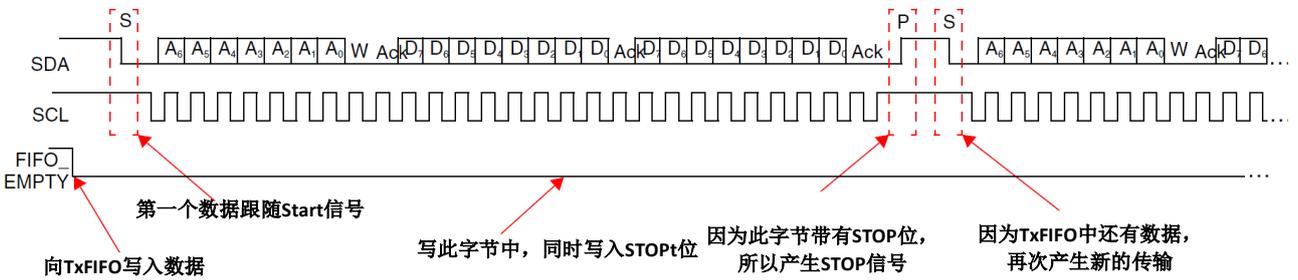


图 8-14 主机发送, 产生 STOP, Tx FIFO 非空

6. 主机接收时, 产生 STOP, 但 Tx FIFO 非空

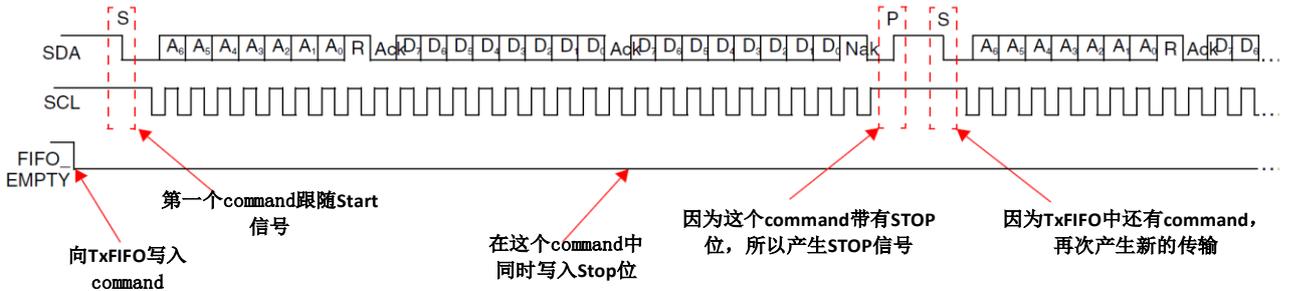


图 8-15 主机接收, 产生 STOP, Tx FIFO 非空

9. 音频接口（I2S）

9.1. 概述

I2S(Inter-IC Sound)总线, 又称集成电路内置音频总线, 可以通过 I2S 音频协议与外部设备进行通信。

9.2. 主要特性

- 支持 MSB（左对齐），LSB（右对齐），飞利浦标准格式。
- 支持主从模式下的 16bit、24bit 有效数据传输。
- Tx FIFO、Rx FIFO 各有深度 32，位宽 32bit 的 FIFO 缓存空间。
- 全双工通讯

9.3. 结构框图

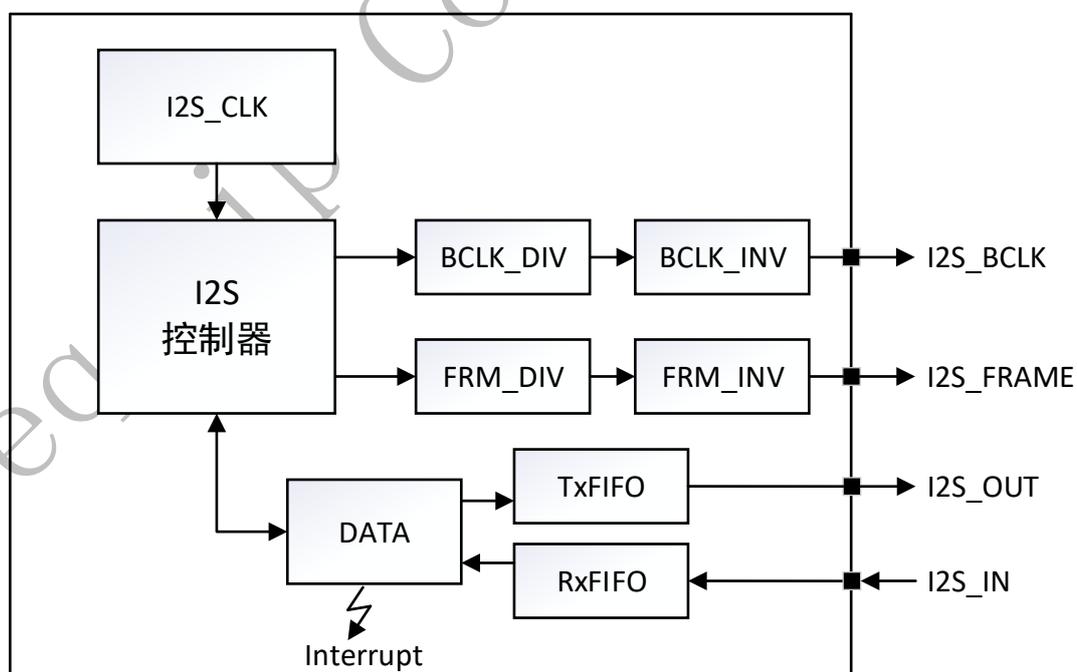


图 9-1 I2S 结构框图

9.4. 功能描述

9.4.1. BCLK、FRMCLK 分频

时钟信号输出分频，与声道信号输出分频。

$$\text{BCLK 分频计算公式: } \text{BCLK} = \frac{\text{I2S_CLK}}{(\text{BCLK_DIV} + 1)}$$

$$\text{FRAMECLK 分频计算公式: } \text{FRAMECLK} = \frac{\text{BCLK}}{2 \times (\text{FRMCLK_DIV} + 1)}$$

其中 BCLK 为 I2S 外设时钟，FRAMECLK 为声道时钟。

9.4.2. I2S_NORMAL, I2S_ADJUST, I2S_FRMINV 配置效果

I2S_NORMAL = 1, I2S_ADJUST = 0, I2S_FRMINV = 0

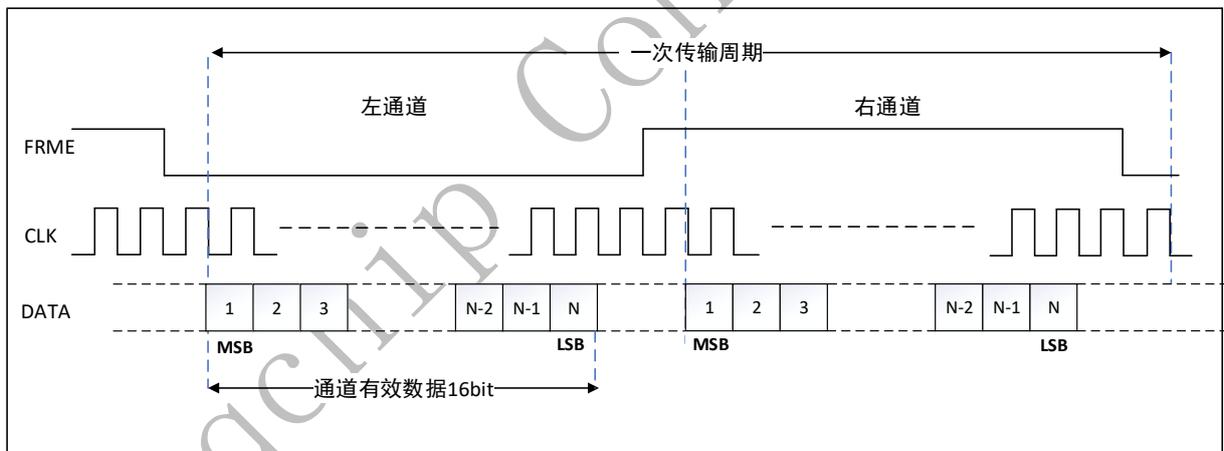


图 9-2 飞利浦标准格式时序图

I2S_NORMAL = 0, I2S_ADJUST = 0, I2S_FRMINV = 1

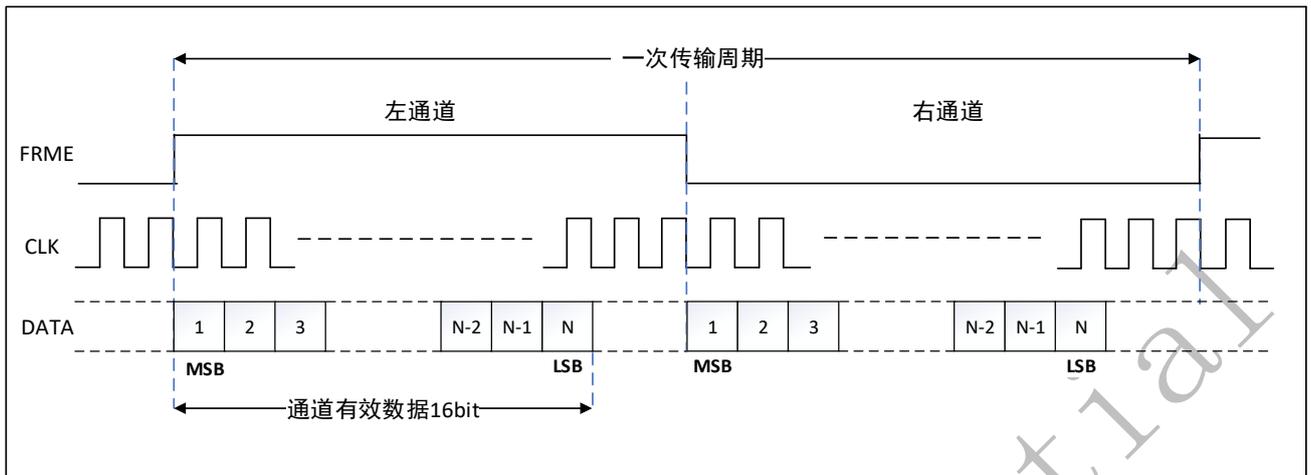


图 9-3 I2S 时序图（左对齐，FRME 反相）

I2S_NORMAL = 0, I2S_ADJUST = 1, I2S_FRMINV = 1, I2SFBOFF = (FRMDIV + 1) - 16

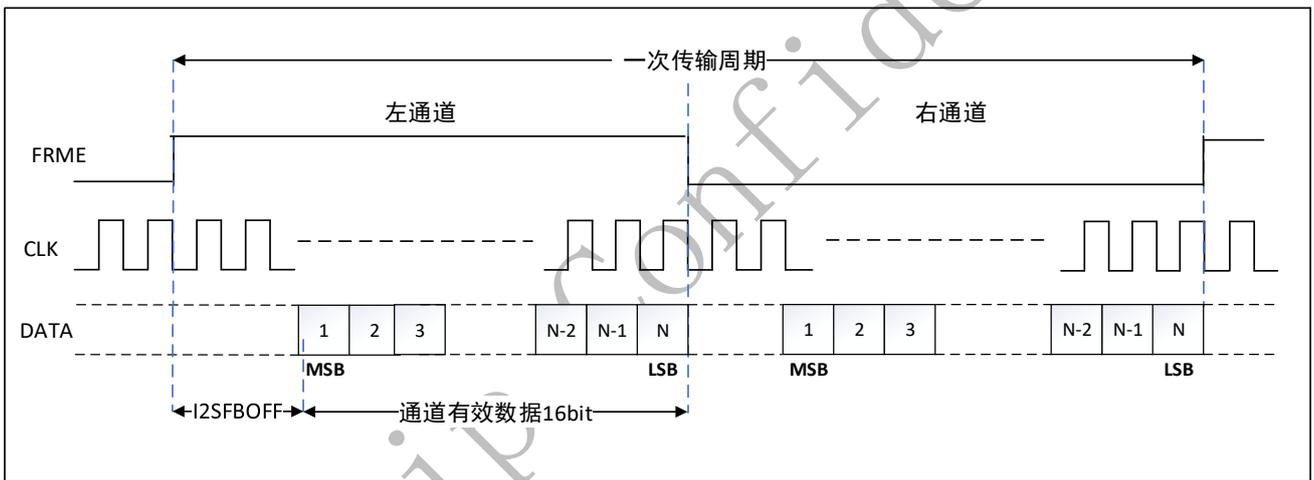


图 9-4 I2S 时序图（右对齐，FRME 反相）

9.5. I2S 寄存器

9.5.1. I2S 寄存器映像

I2S0 寄存器基地址: 0x5006_0000

I2S1 寄存器基地址: 0x5016_0000

I2S2 寄存器基地址: 0x5026_0000

表格 9-1 I2S 寄存器映像

偏移	寄存器	寄存器描述
0x00	CONTROL0	控制寄存器 0

0x04	FRM_DIV	时钟与声道分频寄存器
0x08	CONTROL1	控制寄存器 1
0x0C	DATA_L	左声道数据寄存器
0x10	DATA_R	右声道数据寄存器
0x14	STATUS	状态寄存器
0x18	INTE	中断使能寄存器
0x1C	FIFO_CFG_L	左声道 FIFO 配置寄存器
0x20	FIFO_CFG_R	右声道 FIFO 配置寄存器
0x24	DMA_CFG	DMA 配置寄存器

9.5.2. CONTROL0 (Offset 0x00)

比特	名称	属性	复位值	描述
31:17	Reserved	R/W	0x0	Reserved
16	SBC_ACCESS	R/W	1'b0	I2S TxFIFO 数据帧来源 0: TxFIFO data from CPU 1: TxFIFO data from sbc decode
15:12	Reserved	R/W	0x0	Reserved
11	TXFF_L_CLR	R/W	1'b0	左声道 TxFIFO 复位
10	TXFF_R_CLR	R/W	1'b0	右声道 TxFIFO 复位
9	RXFF_L_CLR	R/W	1'b0	左声道 RxFIFO 复位
8	RXFF_R_CLR	R/W	1'b0	右声道 RxFIFO 复位
7	TXFF_L_EN	R/W	1'b0	左声道 TxFIFO 使能
6	TXFF_R_EN	R/W	1'b0	右声道 TxFIFO 使能
5	RXFF_L_EN	R/W	1'b0	左声道 RxFIFO 使能
4	RXFF_R_EN	R/W	1'b0	右声道 RxFIFO 使能
3	LR_WD_SWAP	R/W	1'b0	左右声道同时写入数据格式配置 仅当 LR_WD_EN=1 时有效，可写入 DATA_L 或 DATA_R 0: 读写数据寄存器时，高 16bit 为左声道，低 16bit 为右声道。 1: 读写数据寄存器时，高 16bit 为右声道，低 16bit 为左声道。
2	INTEN	R/W	1'b0	I2S 中断使能

				0: Disable 1: Enable
1	LR_WD_EN	R/W	1'b0	左右声道同时写入数据模式，与 bit3 配合使用 0: Disable 1: Enable
0	MSTSLV	R/W	1'b0	主从模式选择 0: 主机模式 1: 从机模式

9.5.3. FRM_DIV (Offset 0x04)

比特	名称	属性	复位值	描述
31:16	FRMDIV	R/W	16'hF	声道分频，FRMDIV 必须为奇数。 $FRMCLK = BCLK / (FRM_DIV + 1) \times 2$
15:0	BCLKDIV	R/W	16'h7	外设时钟分频。 $BCLK = I2S_CLK / (BCLKDIV + 1)$

9.5.4. CONTROL1 (Offset 0x08)

比特	名称	属性	复位值	描述
31:16	I2SFBOFF	R/W	00h	与 bit7 和 bit8 配合使用实现右对齐。 设置此值以调整 I2S 第一个串行数据位置以适合正确调整，必须大于 1。例如 I2S 数据帧长度为 16 bit 时，I2SFBOFF 值为 $(FRMDIV + 1) - 16$ 。
15	Reserved	R/W	0x0	Reserved
14:12	I2S_DATA_LEN	R/W	3'b000	I2S 数据帧长度 000: 8 bit 001: 16 bit 010: 20 bit 011: 24 bit 100: 32 bit
11:10	Reserved	R/W	0x0	Reserved
9	I2S_LSB1ST	R/W	1'b0	I2S 帧内的位排序

				0: MSB 在前 1: LSB 在前
8	I2S_ADJUST	R/W	1'b0	I2S 左右对齐选择, bit7 必须为 0 才能使用。 0: 左对齐 1: 右对齐, 需要配合 bit31:16 使用
7	I2S_NORMAL	R/W	1'b0	I2S 飞利浦标准模式使能 0: Disable 1: Enable
6:4	Reserved	R/W	0x0	Reserved
3	I2S_LP	R/W	1'b0	移位寄存器循环, 发送移位寄存器与接收移位寄存器对接。 1: 测试模式, TX 和 RX 对接 0: 正常模式
2	I2S_BCLKINV	R/W	1'b0	BCLK 线翻转 0: Disable 1: Enable
1	I2S_FRMINV	R/W	1'b0	FRMCLK 线翻转 0: Disable 1: Enable
0	I2S_EN	R/W	1'b0	I2S 使能 0: Disable 1: Enable

9.5.5. DATA_L (Offset 0x0C)

比特	名称	属性	复位值	描述
31:0	DATA_L	R/W	32'h00	写操作: 向 TxFIFO 写入数据。 读操作: 从 RxFIFO 读出数据

9.5.6. DATA_R (Offset 0x10)

比特	名称	属性	复位值	描述
31:0	DATA_R	R/W	32'h00	写操作: 向 TxFIFO 写入数据。 读操作: 从 RxFIFO 读出数据

9.5.7. STATUS (Offset 0x14)

比特	名称	属性	复位值	描述
31: 18	Reserved	R/W	0x0	Reserved
17	R_TXFFEMPTY	R	1'b0	右声道 TxFIFO 空
16	R_TXFFAEMPTY	R	1'b0	右声道 TxFIFO 阈值空
15	R_TXFFFULL	R	1'b0	右声道 TxFIFO 满
14	L_TXFFEMPTY	R	1'b0	左声道 TxFIFO 空
13	L_TXFFAEMPTY	R	1'b0	左声道 TxFIFO 阈值空
12	L_TXFFFULL	R	1'b0	左声道 TxFIFO 满
11	TXFFEMPTY	R	1'b0	左右声道 TxFIFO 空
10	TXFFAEMPTY	R	1'b0	左右声道 TxFIFO 阈值空
9	TXFFFULL	R	1'b0	左右声道 TxFIFO 满
8	R_RXFFEMPTY	R	1'b0	右声道 RxFIFO 空
7	R_RXFFAFULL	R	1'b0	右声道 RxFIFO 阈值满
6	R_RXFFFULL	R	1'b0	右声道 RxFIFO 满
5	L_RXFFEMPTY	R	1'b0	左声道 RxFIFO 空
4	L_RXFFAFULL	R	1'b0	左声道 RxFIFO 阈值满
3	L_RXFFFULL	R	1'b0	左声道 RxFIFO 满
2	RXFFEMPTY	R	1'b0	左右声道 RxFIFO 空
1	RXFFAFULL	R	1'b0	左右声道 RxFIFO 阈值满
0	RXFFFULL	R	1'b0	左右声道 RxFIFO 满

9.5.8. INTE (Offset 0x18)

比特	名称	属性	复位值	描述
31:18	Reserved	R/W	0x0	Reserved
17	R_TXFFEMPTY_EN	R/W	1'b0	FIFO 状态使能 0: Disable 1: Enable
16	R_TXFFAEMPTY_EN	R/W	1'b0	
15	R_TXFFFULL_EN	R/W	1'b0	
14	L_TXFFEMPTY_EN	R/W	1'b0	
13	L_TXFFAEMPTY_EN	R/W	1'b0	
12	L_TXFFFULL_EN	R/W	1'b0	
11	TXFFEMPTY_EN	R/W	1'b0	
10	TXFFAEMPTY_EN	R/W	1'b0	
9	TXFFFULL_EN	R/W	1'b0	
8	R_RXFFEMPTY_EN	R/W	1'b0	
7	R_RXFFAFULL_EN	R/W	1'b0	
6	R_RXFFFULL_EN	R/W	1'b0	
5	L_RXFFEMPTY_EN	R/W	1'b0	
4	L_RXFFAFULL_EN	R/W	1'b0	
3	L_RXFFFULL_EN	R/W	1'b0	
2	RXFFEMPTY_EN	R/W	1'b0	
1	RXFFAFULL_EN	R/W	1'b0	
0	RXFFFULL_EN	R/W	1'b0	

9.5.9. FIFO_CFG_L (Offset 0x1C)

比特	名称	属性	复位值	描述
31:24	L_RXFF_AFULL	R/W	8'h1A	左声道 RxFIFO Amost 满阈值
23:16	L_RXFF_EMPTY	R/W	8'h6	左声道 RxFIFO 空阈值
15:8	L_TXFF_FULL	R/W	8'h1A	左声道 TxFIFO 满阈值
7:0	L_TXFF_AEMPTY	R/W	8'h6	左声道 TxFIFO Amost 空阈值

9.5.10. FIFO_CFG_R (Offset 0x20)

比特	名称	属性	复位值	描述
31:24	R_RXFF_AFULL	R/W	8'h1A	右声道 RxFIFO Amost 满阈值
23:16	R_RXFF_EMPTY	R/W	8'h6	右声道 RxFIFO 空阈值
15:8	R_TXFF_FULL	R/W	8'h1A	右声道 TxFIFO 满阈值
7:0	R_TXFF_AEMPTY	R/W	8'h6	右声道 TxFIFO Amost 空阈值

9.5.11. DMA_CFG (Offset 0x24)

比特	名称	属性	复位值	描述
31:21	Reserved	R/W	0x0	Reserved
20:16	DMATDLR	R/W	5'h10	DMA Tx 阈值。 Rx FIFO 数量等于或小于 DMA_TX_Level 时，产生 DMA Tx 请求
15:13	Reserved	R/W	0x0	Reserved
12:8	DMARDLR	R/W	5'h10	DMA Rx 阈值。 Rx FIFO 数量等于或大于 DMA_RX_Level 时，产生 DMA Rx 请求
7:4	Reserved	R/W	0x0	Reserved
3	DMACR_R_RX	R/W	1'b0	右声道 RxFIFO 清除 DMA 搬运请求，高电平有效
2	DMACR_L_RX	R/W	1'b0	左声道 RxFIFO 清除 DMA 搬运请求，高电平有效
1	DMACR_R_TX	R/W	1'b0	右声道 TxFIFO 清除 DMA 搬运请求，高电平有效
0	DMACR_L_TX	R/W	1'b0	左声道 TxFIFO 清除 DMA 搬运请求，高电平有效

10. 真随机数发生器 (TRNG)

10.1. 概述

True Random Number Generator (TRNG) 真随机数生成器允许从数字逻辑中生成和收集真正的随机比特流。

10.2. 主要特性

- 192bit 真随机数发生器

10.3. TRNG 结构框图

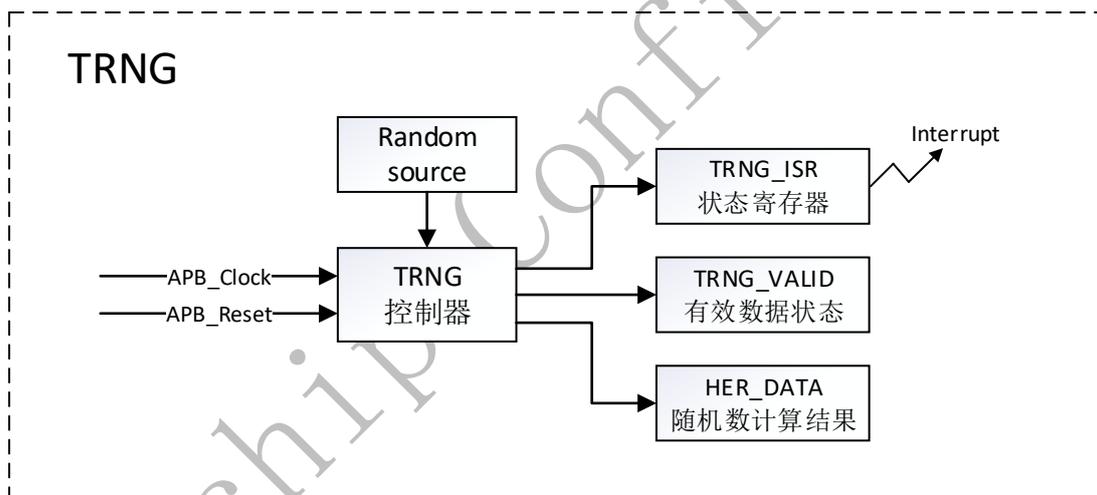


图 10-1 TRNG 结构框图

10.4. TRNG 寄存器

10.4.1. TRNG 寄存器映像

TRNG 寄存器基地址: 0xE00D_0000

表格 10-1 TRNG 寄存器映像

偏移	寄存器	寄存器描述
0x100	TRNG_IMR	中断屏蔽寄存器

0x104	TRNG_ISR	中断状态寄存器
0x108	TRNG_ICR	中断清除寄存器
0x10C	TRNG_CONFIG	TRNG 配置寄存器
0x110	TRNG_VALID	有效数据位状态寄存器
0x114	TRNG_DATA0	随机数结果寄存器
0x118	TRNG_DATA1	
0x11C	TRNG_DATA2	
0x120	TRNG_DATA3	
0x124	TRNG_DATA4	
0x128	TRNG_DATA5	
0x12C	RND_SOURCE_ENABLE	随机数源使能寄存器
0x130	SAMPLE_CNT1	采样计数值寄存器
0x140	TRNG_RESET	TRNG 复位寄存器
0x1B8	TRNG_BUSY	TRNG 忙碌状态寄存器
0x1BC	RST_BITS_COUNTER	计数复位寄存器

10.4.2. TRNG_IMR (Offset 0x100)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h00	Reserved
3	VN_ERR_INT_MASK	R/W	1'b1	Von Neumann errors 错误中断。 写 '1' 屏蔽
2	CRNGT_ERR_INT_MASK	R/W	1'b1	CRNGT errors 错误中断。 写 '1' 屏蔽
1	AUTOCORR_ERR_INT_MASK	R/W	1'b1	Autocorrelation errors 错误中断。 写 '1' 屏蔽
0	EHR_VALID_INT_MASK	R/W	1'b1	TRNG 已经收集 192bit 中断 写 '1' 屏蔽

10.4.3. TRNG_ISR (Offset 0x104)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h00	Reserved
3	VN_ERR	R	1'b0	Von Neumann errors 错误状态标志。 如果 32 个连续的收集位是相同的，即 32 个 0 或 32 个 1，就会发生 Von Neumann 错误。
2	CRNGT_ERR	R	1'b0	CRNGT errors 错误状态标志。 连续随机数生成测试失败，当 16 个采集位的两个连续块相等时，故障发生。
1	AUTOCORR_ERR	R	1'b0	Autocorrelation errors 错误状态标志。 表示连续 4 次自相关测试失败。 注： 设置后，TRNG 将停止工作，直到下一次重置。
0	EHR_VALID	R	1'b0	TRNG 已经收集 192bit 状态标志。 置 '1' 时，HER_DATA[0,1,2,3,4,5] 寄存器准备好被读取。

10.4.4. TRNG_ICR (Offset 0x108)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h00	Reserved
3	VN_ERR	W	1'b0	写 '1' 清除 Von Neumann errors 错误状态标志。
2	CRNGT_ERR	W	1'b0	写 '1' 清除 CRNGT errors 错误状态标志。
1	AUTOCORR_ERR	W	1'b0	软件无法操作此位。 TRNG 复位后自动清除。
0	EHR_VALID	W	1'b0	读取完 HER_DATA[0~5] 寄存器后，写 '1' 清除状态标志。

10.4.5. TRNG_CONFIG (Offset 0x10C)

比特	名称	属性	复位值	描述
31:2	Reserved	R/W	30'h00	Reserved
1:0	ENG_SRC_SEL	R/W	2'b0	配置环形振荡器中的反相器链长度。 00: 选择最短的反相器链长度 01: 选择较短的反相器链长度 10: 选择较长的反相器链长度 11: 选择最长的反相器链长度

10.4.6. TRNG_VALID (Offset 0x110)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	EHR_VALID	R	1'b0	当置 '1' 时，表示 TRNG 中的位收集完成，可以从 EHR_DATA[0~5] 寄存器中读取数据。

10.4.7. TRNG_DATA0 ~ 5 (Offset 0x114 ~ 0x128)

比特	名称	属性	复位值	描述
31:0	TRNG_DATA5 (0x124)	R	32'h00	随机数[191:160]
31:0	TRNG_DATA4 (0x124)	R	32'h00	随机数[159:128]
31:0	TRNG_DATA3 (0x120)	R	32'h00	随机数[127:96]
31:0	TRNG_DATA2 (0x11C)	R	32'h00	随机数[95:64]
31:0	TRNG_DATA1 (0x118)	R	32'h00	随机数[63:32]
31:0	TRNG_DATA0 (0x114)	R	32'h00	随机数[31:0]

10.4.8. RND_SOURCE_ENABLE (Offset 0x12C)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	RND_SRC_EN	R/W	1'b0	随机数源使能。 0: Disable 1: Enable

10.4.9. SAMPLE_CNT1 (Offset 0x130)

比特	名称	属性	复位值	描述
31:0	SAMPLE_CNT1	R/W	32'hFFFF	控制 TRNG 采样环形振荡器的单个输出位的频率。

10.4.10. TRNG_RESET (Offset 0x140)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	TRNG_RESET	R	1'b0	写 '1' 将导致内部 TRNG 复位。

10.4.11. TRNG_BUSY (Offset 0x1B8)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	TRNG_BUSY	R	1'b0	反映 TRNG busy 信号的状态

10.4.12. RST_BITS_COUNTER (Offset 0x1BC)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	RST_BITS_COUNTER	W	1'b0	向此位写入任何值将重置位计数器和 TRNG VALID 寄存器。 注：只有在 RND_SRC_EN == 0 时有效

11. 数字麦克风 (PDM)

11.1. 概述

Pulse Density Modulation (PDM)模块可以从外部音频(例如数字麦克风)输入脉冲密度调制信号，支持单通道、双通道数据输入。

11.2. 主要特性

- 支持深度 32，位宽 16bit 的缓存 FIFO
- 硬件高通滤波器
- 支持单通道、双通道数据输入

11.3. PDM 系统框图

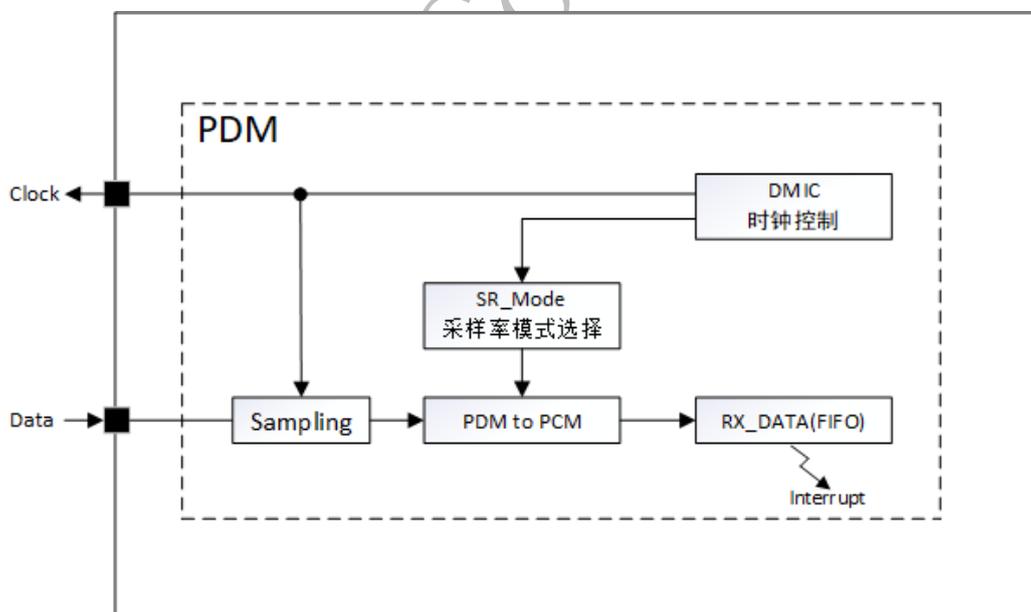


图 11-1 PDM 系统框图

11.4. PDM 寄存器

11.4.1. PDM 寄存器映像

PDM0 寄存器基地址: 0x5007_0000

PDM1 寄存器基地址: 0x5017_0000

PDM2 寄存器基地址: 0x5027_0000

表格 11-1 PDM 寄存器映像

偏移	寄存器	寄存器描述
0x00	PDM_Control	PDM 配置寄存器
0x04	RXFF_FTLR	RX FIFO 阈值寄存器
0x08	PDM_FIFO_RESET	PDM FIFO 软件复位
0x0C	PDM_INTEN	PDM 中断使能
0x10	PDM_INT_RWST	PDM 原始中断状态寄存器
0x14	PDM_INT_STATUS	PDM 中断状态寄存器
0x18	PDM_VOLUME_L_CONFIG	PDM 左声道音量配置
0x1C	PDM_VOLUME_R_CONFIG	PDM 右声道音量配置
0x24	PDM_FIFO	PDM 数据寄存器
0x28	DMA_CFG	DMA 控制寄存器

11.4.2. PDM_Control (Offset 0x00)

比特	名称	属性	复位值	描述
31:15	Reserved	R	0x0	Reserved
15	HPF_EN	R/W	1'b0	高通滤波器使能。 0: High-Pass Filter Disable 1: High-Pass Filter Enable
14	LR_SWAP	R/W	1'b0	左右声道复制使能 0: Disable

				1: Enable
13	MONO_RIGHT	R/W	1'b0	单声道模式时，信道选择 0: 左声道 1: 右声道
12	MONO_STEREO	R/W	1'b0	单声道与双声道选择 0: Stereo 1: Mono
11:10	OSR_MODE	R/W	2'b10	过采样率模式选择： 00: 32/31.25/34 支持所有采样率 01: 64/62.5/68 支持 96K/88.2K 采样率 10: 128/125/136 支持 44K/44.1K 采样率 注：normal mode 模式下，OSR = 32/68/136 在 usb mode 下如果采样率为 44.1K，OSR = 34/68/136，其他采样率 OSR= 3125/625/125
9:6	PDM_SMP_RATE	R/W	0x0	采样率选择 0000: 8K 0001: 12K 0010: 16K 0011: 24K 0100: 32K 0101: 48K 0110: 96K 0111: 192K 1000: 8.0214K (不支持 normal mode) 1001: 11.025K 1010: 22.05K 1011: 44.1K 1100: 88.2K 1101: 176.4K
5	USB_MODE	R/W	1'b1	USB 模式选择 0: normal mode 时钟来源 AUPLL 1: usb mode 时钟来源 CORE_HS
4	PDM_CLK_INV	R/W	1'b0	PDM 时钟线反向

				0: Disable 1: Enable
3	PDM_DAT_INV	R/W	1'b0	PDM 数据线反向 0: Disable 1: Enable
2	PDM_SOFT_RESET	R/W	1'b0	PDM 软件复位 0: Disable 1: Enable
1	PDM_CLK_EN	R/W	1'b0	PDM 时钟使能 0: 下降沿采样 1: 上升沿采样
0	PDM_EN	R/W	1'b0	PDM 使能。 0: Disable 1: Enable

11.4.3. RXFF_FTLR (Offset 0x04)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5:0	RXFF_FTLR	R/W	0x20	RXFIFO almost 满阈值

11.4.4. PDM_FIFO_RESET (Offset 0x08)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	RXFF_RST	R/W	1'b0	RX FIFO 复位
1	RXFF_POP_RST	R/W	1'b0	RX FIFO POP 复位
0	RXFF_PUSH_RST	R/W	1'b0	RX FIFO PUSH 复位

11.4.5. PDM_INTEN (Offset 0x0C)

比特	名称	属性	复位值	描述
----	----	----	-----	----

31:3	Reserved	R	0x0	Reserved
2	RXFF_EMPTY_INTEN	R/W	1'b0	RX FIFO 空中断使能
1	RXFF_AFULL_INTEN	R/W	1'b0	RX FIFO almost 阈值满中断使能
0	RXFF_FULL_INTEN	R/W	1'b0	RX FIFO 满中断使能

11.4.6. PDM_INT_RWST (Offset 0x10)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	RXFF_EMPTY_INT_RAW	R/W	1'b1	RX FIFO 空原始中断状态
1	RXFF_AFULL_INT_RAW	R/W	1'b0	RX FIFO almost 阈值满原始中断状态
0	RXFF_FULL_INT_RAW	R/W	1'b0	RX FIFO 满原始中断状态

11.4.7. PDM_INT_STATUS (Offset 0x14)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	RXFF_EMPTY_INT_STATUS	R/W	1'b0	RX FIFO 空中断状态
1	RXFF_AFULL_INT_STATUS	R/W	1'b0	RX FIFO almost 阈值满中断状态
0	RXFF_FULL_INT_STATUS	R/W	1'b0	RX FIFO 满中断状态

11.4.8. PDM_VOLUME_L_CONFIG (Offset 0x18)

比特	名称	属性	复位值	描述
31:17	Reserved	R	0x0	Reserved
16:0	VOLUME_L	R/W	0x2000	PDM 左声道音量 (unsigned) dB of volume = $20 * \lg(\text{VOLUME_LEFT} / 2^{13})$

11.4.9. PDM_VOLUME_R_CONFIG (Offset 0x0x1C)

比特	名称	属性	复位值	描述
31:17	Reserved	R	0x0	Reserved
16:0	VOLUME_R	R/W	0x2000	PDM 右声道音量 (unsigned) dB of volume = $20 * \lg(\text{VOLUME_LEFT} / 2^{13})$

11.4.10. PDM_FIFO (Offset 0x24)

比特	名称	属性	复位值	描述
31:16	R_DAT	R	0x0	右声道数据 FIFO
15:0	L_DAT	R	0x0	左声道数据 FIFO

11.4.11. DMA_CFG (Offset 0x28)

比特	名称	属性	复位值	描述
31:7	Reserved	R	0x0	Reserved
6	DMA_EN	R/W	1'b0	DMA 使能 0: Disable 1: Enable
5:0	DMA_RDLR	R/W	6'b0	DMA 接收数据阈值

12. DMA 控制器（DMA）

12.1. 概述

直接存储器存取（direct memory access **DMA**）使用硬件传输数据的方法，用来提供存储器与存储器之间和外设与存储器之间的高速数据传输。无需 CPU 干预，节省了 CPU 资源。每个通道相互独立，专门管理独自通道的访问请求。

12.2. 主要特性

- DMA0 拥有 8 个独立的可配置通道（请求），DMA1 拥有 2 个。
- 支持独立总线读写，DMA0（M1~M4）DMA1（M1~M3）。
- 支持外设与存储器之间的传输
- 支持存储器与存储器之间的传输
- 支持链表循环传输功能
- 可配置数据源和目标数据区的传输宽度(字节、半字、全字)
- 支持硬件优先级（优先级可配。同优先级下，通道号越低，优先级越高）
- DMA0 通道 0 和 1 支持 64 Byte FIFO 深度，2~7 支持 32Byte FIFO 深度
- DMA1 通道 0 和 1 支持 64 Byte FIFO 深度
- 通道 1、通道 2 支持 32 Byte FIFO 深度

12.3. 功能描述

DMA 控制器和 CPU 核心共享系统数据总线，执行直接存储器数据传输。当 CPU 和 DMA 同时访问相同的目标（RAM 或外设）时，DMA 请求会暂停 CPU 访问系统总线若干的周期，总线仲裁器会执行循环调度，以保证 CPU 在获得访问资源后的正常执行。

12.3.1. DMA 优先级

优先级可配，可通过 CFGx.CH_PROOR 配置通道优先级，7 为最高优先级，0 为最低优先级。优先级相同的通道，通道号越低优先级越高。

注：

当 DMA 控制器在同一时间接收到多个请求时，仲裁器将根据硬件优先级来决定响应哪一个外设请求。

如果 DMA 正在为一个低优先级的通道传输数据时，一个高优先级的通道请求产生了，那么 DMA 将会先把低优先级通道的数据传输完毕，然后再来处理高优先级的通道要求。

12.3.2. 传输总长度、数据位宽、Burst length 三者关系

CTLx.寄存器中，BLOCK_TS 为传输的总长度，单位为源数据位宽(SRC_TR_WIDTH)。SRC_TR_WIDTH、DST_TR_WIDTH 分别为源数据位宽，目标数据位宽。SRC_MSIZ、DEST_MSIZ 为 Burst length。

例如：

BLOCK_TS = 5.

SRC_TR_WIDTH = 0x2. (4 byte)

DST_TR_WIDTH = 0x2. (4 byte)

SRC_MSIZ = 0x1. (待传输的数据项为 4)

DEST_MSIZ = 0x1. (待传输的数据项为 4)

DMA 传输的总长度位 $5(\text{BLOCK_TS}) * 4(\text{SRC_TR_WIDTH}) = 20\text{byte}$ 。

DMA 收到一次传输请求后，将待传输的数据项 $4 * 4\text{ byte} = 16\text{ byte}$ 的数据读取到 DMA FIFO 中。再由 FIFO 搬往目标地址。

DMA 再次收到一次传输请求，将剩余的 $20 - 16 = 4\text{ byte}$ 的数据搬往目标地址。

注： SRC_TR_WIDTH * SRC_MSIZ 的数据量不能超过 DMA 通道 FIFO 深度。如果是内存与外设之间的传输，SRC_TR_WIDTH * SRC_MSIZ 需要搬运的数据量也不能超过外设 FIFO 的深度。

12.3.3. 链表传输功能

可通过使用链表实现循环传输。每个链表中包含着每次传输的 **1.目标地址**，**2.源地址**，**3.下一次传输的链表基地址**（如本次传输是最后一次，则赋值为 NULL），**4.控制寄存器**。

在第一个链表传输完成后，将查询第一个链表的 *Nxet 值，如果不为 NULL，则进行下一个链表的传输。

12.3.4. 离散与聚合功能

直接存储器访问(DMA)是一种不需要 CPU 参与的传输，在存储器之间，存储器与外设之间进行传输数据块的技术。DMA 不仅减轻系统处理单元的工作，而且以比处理器的读取和写入速率高得多的速率传输数据。

传统的 block DMA 一次只能传输物理上连续的一个块的数据，完成传输后发起中断。而离散(Scatter)-聚合(Gather) DMA 提供从一个非连续的存储器到另一个存储器，采用通过一系列较小的连续数据块传输的方法进行数据传输。允许一次传输多个物理上不连续的块，完成传输后只发起一次中断。这样做的好处是直观的，大大减少了中断的次数，提高了数据传输的效率。

12.3.4.1. 离散(Scatter)

离散与目的地传输相关，当到达离散边界时，目标地址按编程数量递增或递减—离散递增。图 12-1 显示了一个目的地离散传输的例子。当达到离散边界时，地址由存储在 DSRx 寄存器中的 DSI 字段中的值递增或递减，乘以 DST_TR_WIDTH/8。其中 DST_TR_WIDTH 在 CTLx 寄存器中描述。连续离散边界直接的目的地传输数量记录在 DSRx 寄存器的 DSC 字段。

通过向 CTLx 寄存器的 DST_SCATTER_EN 字段写 1 来使能 scatter。CTLx 的 DINC 字段确定当到达离散边界时，地址是增加、减少还是保持不变。如果在整个 DMA 传输过程中 CTLx 的 DINC 字段表示固定地址控制，则 CTLx 寄存器的 DST_SCATTER_EN 字段会被忽略，并自动禁用 scatter 特性。

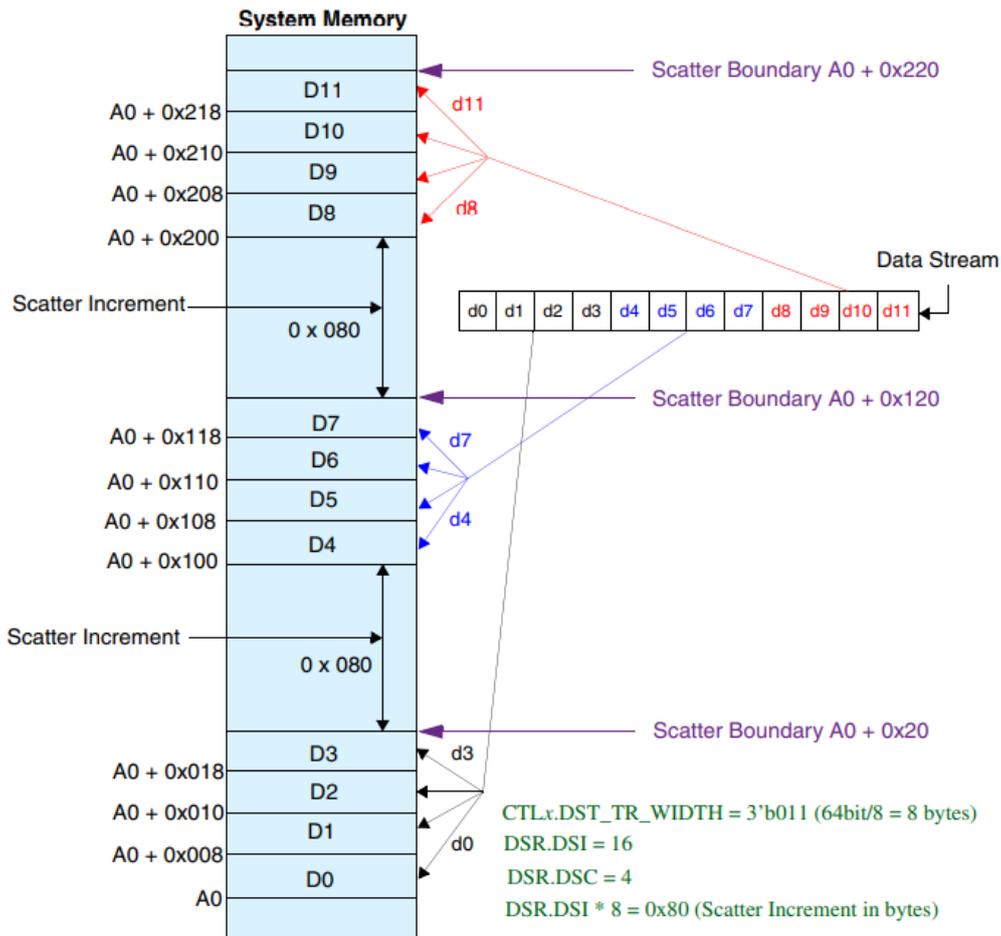


图 12-1 离散传输示例图

12.3.4.2. 聚合(Gather)

聚合与源传输相关。当到达聚合边界时，源地址按编程数量递增或递减。连续聚合边界之间的源传输数量被编程到源聚集计数 SGRx 寄存器的 SGC 字段中。当到达聚合边界时，源地址由存储在 SGRx 寄存器 SGI 字段的值递增或递减，乘以 CTLx 寄存器的 SRC_TR_WIDTH/8。图 12-2 显示了一个源聚合传输的例子。

通过向 CTLx 寄存器的 SRC_GATHER_EN 字段写 1 来使能聚合功能。CTLx 寄存器的 SINC 字段确定当到达聚合边界时，地址是递增、递减还是保持固定。如果 CTLx 寄存器的 SINC 字段表示在整个 DMA 传输过程中由固定地址控制，则忽略 CTLx 寄存器的 SRC_GATHER_EN 字段，并自动禁用聚合特性。

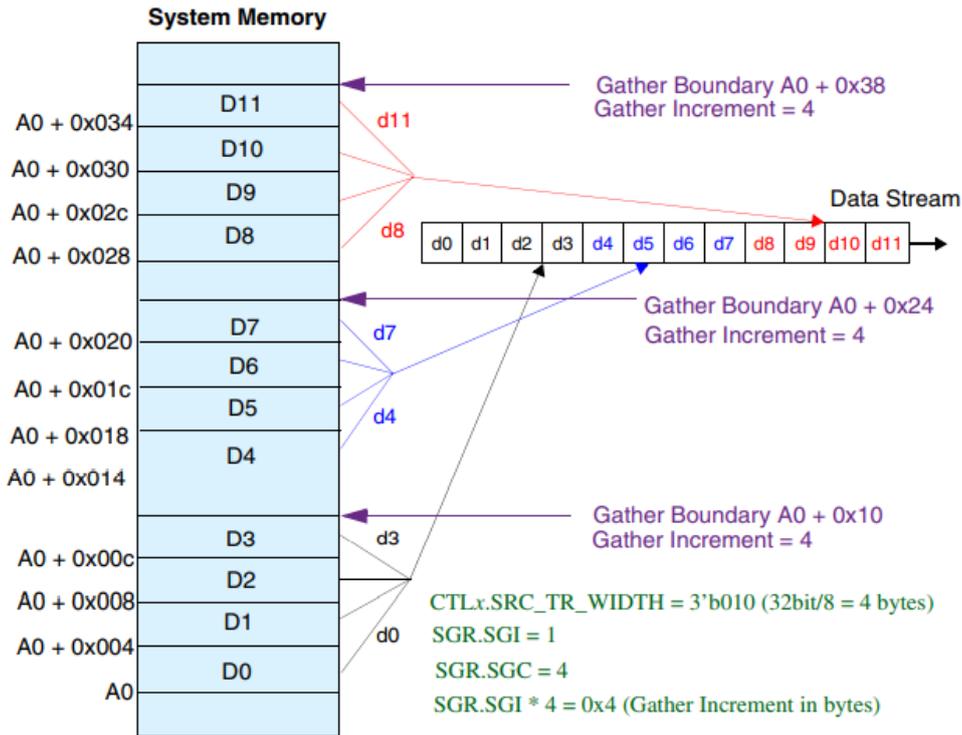


图 12-2 源聚合传输示例图

12.4. DMA 寄存器

12.4.1. DMA 寄存器映像

DMA0 寄存器基地址: 0x1000_0000

DMA1 寄存器基地址: 0x4002_0000

表格 12-1 DMA 寄存器映像

偏移	寄存器	寄存器描述
0x00, 0x58, 0xB0, 0x108, 0x160, 0x1B8, 0x210, 0x268	SARx	通道 0~8 源地址寄存器
0x08, 0x60, 0xB8, 0x110, 0x168, 0x1C0, 0x218, 0x270	DARx	通道 0~8 目标地址寄存器
0x10, 0x68, 0xC0, 0x118, 0x170, 0x1C8,	LLPx	通道 0~8 链表地址寄存器

0x220, 0x278		
0x18, 0x70, 0xC8, 0x120, 0x178, 0x1D0, 0x228, 0x280	CTLx	通道 0~8 控制寄存器
0x40, 0x98, 0xF0, 0x148, 0x1A0, 0x1F8, 0x250, 0x2A8	CFGx	通道 0~8 配置寄存器
0x48, 0xA0, 0xF8, 0x150, 0x1A8, 0x200, 0x258, 0x2B0	SGRx	通道 0~8 源聚合寄存器
0x50, 0xA8, 0x100, 0x158, 0x1B0, 0x208, 0x260, 0x2B8	DSRx	通道 0~8 目的离散寄存器
0x2C0	RawTfr	传输完成原始状态寄存器
0x2E0	RawErr	传输错误原始状态寄存器
0x2E8	StatusTfr	传输完成状态寄存器
0x308	StatusErr	传输错误状态寄存器
0x310	MaskTfr	传输完成掩码寄存器
0x330	MaskErr	传输错误掩码寄存器
0x338	ClearTfr	传输完成状态清除寄存器
0x358	ClearErr	传输错误状态清除寄存器
0x360	StatusInt	中断状态寄存器
0x398	DmaCfgReg	DMA 配置寄存器
0x3A0	ChEnReg	DMA 通道使能寄存器

12.4.2. SARx (Offset 0x00, 0x58, 0xB0, 0x108, 0x160, 0x1B8, 0x210, 0x268)

比特	名称	属性	复位值	描述
63:32	Reserved	R/W	0x0	Reserved
31:0	SAR	R/W	32'h00	DMA 传输源地址

12.4.3. DARx (Offset 0x08, 0x60, 0xB8, 0x110, 0x168, 0x1C0, 0x218, 0x270)

比特	名称	属性	复位值	描述
63:32	Reserved	R/W	0x0	Reserved
31:0	DAR	R/W	32'h00	DMA 传输目标地址

12.4.4. LLPx (Offset 0x10, 0x68, 0xC0, 0x118, 0x170, 0x1C8, 0x220, 0x278)

比特	名称	属性	复位值	描述
63:32	Reserved	R/W	0x0	Reserved
31:2	LOC	R/W	30'h00	如果启用链表功能，则下一个链表的内存起始地址
1:0	LMS	R/W	2'b00	下一个链表选用的 Master 0: AHB Master1 1: AHB Master2 2: AHB Master3 3: AHB Master4

12.4.5. CTLx (Offset 0x18, 0x70, 0xC8, 0x120, 0x178, 0x1D0, 0x228, 0x280)

比特	名称	属性	复位值	描述
63:44	Reserved	R/W	0x0	Reserved
43:32	BLOCK_TS	R/W	12'h000	DMA 搬运长度。
31:29	Reserved	R/W	0x0	Reserved
28	LLP_SRC_EN	R/W	1'b0	源地址链表使能
27	LLP_DST_EN	R/W	1'b0	目标地址链表使能
26:25	SMS	R/W	2'b00	源 Master 选择 0: AHB Master1 1: AHB Master2 2: AHB Master3 3: AHB Master4

24:23	DMS	R/W	2'b00	<p>目标 Master 选择</p> <p>0: AHB Master1</p> <p>1: AHB Master2</p> <p>2: AHB Master3</p> <p>3: AHB Master4</p>
22:20	TT_FC	R/W	3'b000	<p>传输类型和流控制器选择</p> <p>0: 内存 到 内存, DMA 控制</p> <p>1: 内存 到 外设, DMA 控制</p> <p>2: 外设 到 内存, DMA 控制</p> <p>3: 外设 到 外设, DMA 控制</p>
19	Reserved	R/W	0x0	Reserved
18	DST_SCATTER_EN	R/W	0x0	<p>目的地离散使能。</p> <p>0x0: 未使能。</p> <p>0x1: 使能。</p>
17	SRC_GATHER_EN	R/W	0x0	<p>源聚合使能。</p> <p>0x0: 未使能。</p> <p>0x1: 使能。</p>
16:14	SRC_MSIZ	R/W	3'b000	<p>源 Burst 传输长度选择。</p> <p>当收到 DMA 请求时, 待搬运的数据项, 一个数据项的长度为 SRC_TR_WIDTH (源位宽)。</p> <p>0x0: 待传输的数据项为 1。</p> <p>0x1: 待传输的数据项为 4。</p> <p>0x2: 待传输的数据项为 8。</p> <p>0x3: 待传输的数据项为 16。</p> <p>0x4: 待传输的数据项为 32。</p> <p>0x5: 待传输的数据项为 64。</p> <p>注: SRC_MSIZ 应与 DEST_MSIZ 配置同。</p>
13:11	DEST_MSIZ	R/W	3'b000	<p>目标 Burst 传输长度选择。</p> <p>当收到 DMA 请求时, 待搬运的数据项, 一个数据项的长度为 DST_TR_WIDTH (源位宽)。</p> <p>0x0: 待传输的数据项为 1。</p> <p>0x1: 待传输的数据项为 4。</p> <p>0x2: 待传输的数据项为 8。</p> <p>0x3: 待传输的数据项为 16。</p> <p>0x4: 待传输的数据项为 32。</p>

				0x5: 待传输的数据项为 64。
10:9	SINC	R/W	2'b0	源地址递增，递减选择。 0x0: 源地址递增。 0x1: 源地址递减。 0x2: 源地址不变。
8:7	DINC	R/W	2'b0	目标地址递增，递减选择。 0x0: 目标地址递增。 0x1: 目标地址递减。 0x2: 目标地址不变。
6:4	SRC_TR_WIDTH	R/W	3'b000	源传输位宽。 0x0: 1 byte 0x1: 2 byte 0x2: 4 byte 0x3: 8 byte 0x4: 16 byte 0x5: 32 byte
3:1	DST_TR_WIDTH	R/W	3'b000	目标传输位宽。 0x0: 1 byte 0x1: 2 byte 0x2: 4 byte 0x3: 8 byte 0x4: 16 byte 0x5: 32 byte
0	INT_EN	R/W	1'b0	中断使能。 0x0: 中断关闭 0x1: 中断开启

12.4.6. CFGx (Offset 0x40, 0x98, 0xF0, 0x148, 0x1A0, 0x1F8, 0x250, 0x2A8)

比特	名称	属性	复位值	描述
63:47	Reserved	R/W	0x0	Reserved
46:43	DEST_PER	R/W	4'h0	目标外设 ID。

				<p>当目标地址是外设时，设置此位。ID 值由用户自定义。</p> <p>与系统寄存器 DMA_REQ_MUX0， DMA_REQ_MUX1， DMA_REQ_MUX2，设置的 ID 相同即可。</p>
42:39	SRC_PER	R/W	4'h0	<p>源外设 ID。</p> <p>当源地址是外设时，设置此位。ID 值由用户自定义。</p> <p>与系统寄存器 DMA_REQ_MUX0， DMA_REQ_MUX1， DMA_REQ_MUX2，设置的 ID 相同即可。</p>
38:34	Reserved	R/W	0x0	Reserved
33	FIFO_MODE	R/W	1'b0	<p>0x0: FIFO_MODE_0，可用于指定传输宽度的单个 AHB 传输的空间/数据。</p> <p>0x1: FIFO_MODE_1，对于目的地传输，可用数据大于或等于一半的 FIFO 深度，而对于源传输，可用空间大于一半的 FIFO 深度。异常发生在突发事务请求的末尾</p>
32:20	Reserved	R/W	0x0	Reserved
19	SRC_HS_POL	R/W	1'b0	<p>源外设握手接口极性。</p> <p>0x0: 源握手信号高有效 0x1: 源握手信号低有效</p> <p>当源地址为外设时要设置此位。</p>
18	DST_HS_POL	R/W	1'b0	<p>目标外设握手接口极性。</p> <p>0x0: 目标握手信号高有效 0x1: 目标握手信号低有效</p> <p>当目标地址为外设时要设置此位。</p>
17:12	Reserved	R/W	0x0	Reserved
11	HS_SEL_SRC	R/W	1'b0	<p>源握手信号选择。</p> <p>0x0: 硬件握手 0x1: 软件握手</p> <p>当源地址为寄存器时，忽略此位。</p>
10	HS_SEL_DST	R/W	1'b0	<p>目标握手信号选择。</p> <p>0x0: 硬件握手</p>

				0x1: 软件握手 当目标地址为寄存器时, 忽略此位。
9	FIFO_EMPTY	R	1'b0	通道 FIFO 状态标志。 0x0: 通道 FIFO 非空 0x1: 通道 FIFO 空
8	CH_SUSP	R/W	1'b0	通道暂停。 0x0: 通道正常 0x1: 通道暂停
7:5	CH_PROOR	R/W	3'b000	通道优先级。 7 为最高优先级, 0 为最低优先级。优先级相同的通道, 通道号越低优先级越高。 0x0: 通道优先级 0 (最低优先级) 0x1: 通道优先级 1 0x2: 通道优先级 2 0x3: 通道优先级 3 0x4: 通道优先级 4 0x5: 通道优先级 5 0x6: 通道优先级 6 0x7: 通道优先级 7 (最高优先级)
4:0	Reserved	R/W	0x0	Reserved

12.4.7. SGRx(Offset 0x48, 0xA0, 0xF8, 0x150, 0x1A8, 0x200, 0x258, 0x2B0)

比特	名称	属性	复位值	描述
31:20	SGC	R/W	0x0	源聚合计数。连续聚合边界之间的源连接传输计数。
19:0	SGI	R/W	3'b000	源聚合间隔。

12.4.8. DSRx(Offset 0x50, 0xA8, 0x100, 0x158, 0x1B0, 0x208, 0x260, 0x2B8)

比特	名称	属性	复位值	描述
31:20	DSC	R/W	0x0	目的地离散计数。连续离散边界之间的目的地连续传输计数。
19:0	RAW	R/W	3'b000	目的地离散间隔。

12.4.9. RawTfr (Offset 0x2C0)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	0x0	Reserved
2:0	RAW	R/W	3'b000	对应通道传输完成中断原始状态。 每 bit 对应一个通道，bit0 对应通道 0。 0x0: 传输未完成 0x1: 传输完成

12.4.10. RawErr (Offset 0x2E0)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	0x0	Reserved
2:0	Raw	R	3'b000	对应通道传输错误中断原始状态。 每 bit 对应一个通道，bit0 对应通道 0。 0x0: 传输正常 0x1: 传输错误

12.4.11. StatusTfr (Offset 0x2E8)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	0x0	Reserved
2:0	Status	R	3'b000	对应通道传输完成中断状态。 每 bit 对应一个通道，bit0 对应通道 0。 此寄存器状态将请求 DMA 中断。 0x0: 传输未完成 0x1: 传输完成

12.4.12. StatusErr (Offset 0x308)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	0x0	Reserved
2:0	Status	R	3'b000	对应通道传输错误中断状态。 每 bit 对应一个通道，bit0 对应通道 0。 此寄存器状态将请求 DMA 中断。 0x0: 传输正常 0x1: 传输错误

12.4.13. MaskTfr (Offset 0x310)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	0x0	Reserved
10:8	INT_MASK_WE	W	3'b000	掩码寄存器写使能。 每 bit 对应一个通道，bit8 对应通道 0。 写 MaskTfr 寄存器时需要将通道写使能与对 应通道一起写入。 0x0: 写使能关闭 0x1: 写使能打开
7:3	Reserved	R/W	0x0	Reserved

2:0	INT_MASK	R/W	3'b000	<p>对应通道传输完成中断状态掩码。</p> <p>每 bit 对应一个通道，bit0 对应通道 0。</p> <p>StatusTfr 寄存器的内容被 MaskTfr 寄存器的内容屏蔽。</p> <p>0x0: 屏蔽中断状态</p> <p>0x1: 不屏蔽中断状态</p>
-----	----------	-----	--------	---

12.4.14. MaskErr (Offset 0x330)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	0x0	Reserved
10:8	INT_MASK_WE	W	3'b000	<p>掩码寄存器写使能。</p> <p>每 bit 对应一个通道，bit8 对应通道 0。</p> <p>写 MaskErr 寄存器时需要将通道写使能与对应通道一起写入。</p> <p>0x0: 写使能关闭</p> <p>0x1: 写使能打开</p>
7:3	Reserved	R/W	0x0	Reserved
2:0	INT_MASK	R/W	3'b000	<p>对应通道传输错误中断状态掩码。</p> <p>每 bit 对应一个通道，bit0 对应通道 0。</p> <p>StatusErr 寄存器的内容被 MaskErr 寄存器的内容屏蔽。</p> <p>0x0: 屏蔽中断状态</p> <p>0x1: 不屏蔽中断状态</p>

12.4.15. ClearTfr (Offset 0x338)

比特	名称	属性	复位值	描述
31:3	Reserved	W	0x0	Reserved
2:0	CLEAR	W	3'b000	清除对应通道中断状态。 每 bit 对应一个通道，bit0 对应通道 0。 0x0: 无操作 0x1: 清除中断传输完成状态

12.4.16. ClearErr (Offset 0x358)

比特	名称	属性	复位值	描述
31:3	Reserved	W	0x0	Reserved
2:0	CLEAR	W	3'b000	清除对应通道中断状态。 每 bit 对应一个通道，bit0 对应通道 0。 0x0: 无操作 0x1: 清除中断传输错误状态

12.4.17. StatusInt (Offset 0x360)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4	ERR	R	0x0	StatusErr 的 OR 值 0x0: OR 值为 0 0x1: OR 值为 1
3	DSTT	R	0x0	StatusDstTran 的 OR 值 0x0: OR 值为 0 0x1: OR 值为 1
2	SRCT	R	0x0	StatusSrcTran 的 OR 值 0x0: OR 值为 0 0x1: OR 值为 1
1	BLOCK	R	0x0	StatusBlock 的 OR 值 0x0: OR 值为 0 0x1: OR 值为 1
0	TFR	R	0x0	StatusTfr 的 OR 值 0x0: OR 值为 0 0x1: OR 值为 1

12.4.18. DmaCfgReg (Offset 0x398)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	DAM_EN	R/W	1'b0	DMA 使能。 0x0: DMA 关闭。 0x1: DMA 开启。

12.4.19. ChEnReg (Offset 0x3A0)

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	0x0	Reserved
10:8	CH_EN_WE	W	3'b000	通道写使能。 每 bit 对应一个通道，bit8 对应通道 0。 写 ChEnReg 寄存器时需要将通道写使能与对应通道一起写入。 0x0: 写使能关闭 0x1: 写使能打开
7:3	Reserved	R/W	0x0	Reserved
2:0	CH_EN	R/W	3'b000	对应通道使能。 每 bit 对应一个通道，bit0 对应通道 0。 通道使能后开始传输。 0x0: 通道关闭 0x1: 通道使能

12.5. 注意事项

1. 在通道使能后，不能再对通道寄存器有任何写操作。如果一定要写通道寄存器，只能在把使能关闭后再写。
2. 使用储存器与外设之间传输时，需要设置请求 ID。需配置 CFGx 寄存器的 DEST_PER 与 SRC_PER。设置值要与系统寄存器中 [DMA REQ MUX](#) 选用的外设配置一样的 ID 号。

13. 脉冲宽度调制 (PWM)

13.1. 概述

PWM 是一个通用的多通道脉宽调制器模块，用于电机控制，音调产生，电池充电，加热元件等。

在通用 PWM 模式下，基于参数选择，占空比更新可以异步执行或同步执行。在同步模式下，所有通道更新在 PWM 周期的开始，这是有用的电机控制，并可用于保持通道波形之间的固定死区空间。异步模式与 LED 控制等应用程序相关，这些应用程序不需要同步更新。

13.2. 主要特性

- 16 路 PWM 通道输出或 16 路输入捕获
- 两种 PWM 输出模式可选 (PWM 模式)
- 各 PWM 通道有独立的分频、占空比、周期配置
- PWM 反相输出可选
- 输入捕获中断响应

13.3. PWM 系统框图

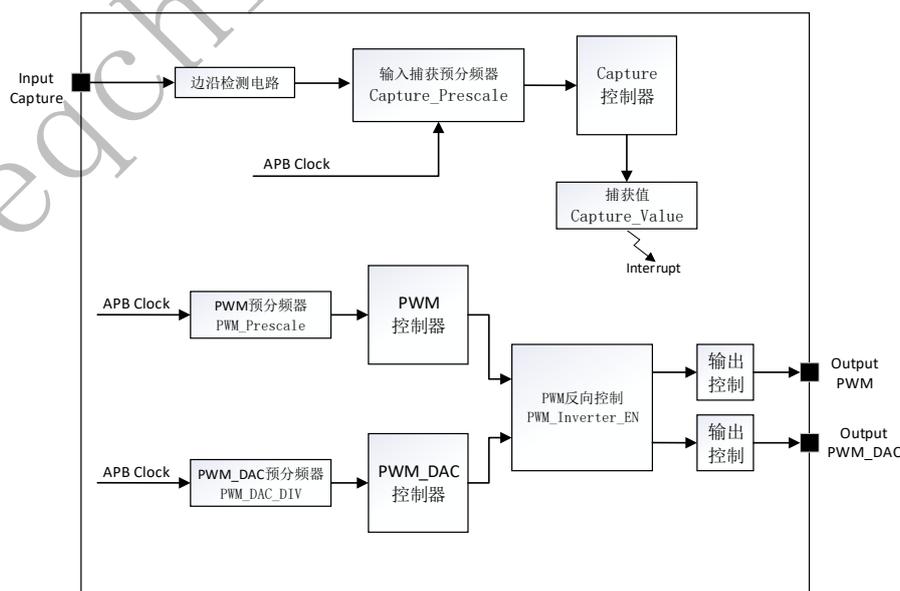


图 13-1 PWM 系统框图

13.4. PWM 寄存器

13.4.1. PWM 寄存器映像

PWM0 寄存器基地址：0x5005_0000

PWM1 寄存器基地址：0x5015_0000

表格 13-1 PWM 寄存器映像

偏移	寄存器	寄存器描述
0x04	PWM_Output_EN	PWM 输出使能
0x08	PWM_ENABLE	通道使能寄存器
0x10	PWM0_Posedge	PWM 通道 0 ~ 15 上升沿位置寄存器 PWM 通道 0 ~ 15 下降沿位置寄存器
0x14	PWM0_Negedge	
0x18	PWM1_Posedge	
0x1C	PWM1_Negedge	
0x20	PWM2_Posedge	
0x24	PWM2_Negedge	
0x28	PWM3_Posedge	
0x2C	PWM3_Negedge	
0x30	PWM4_Posedge	
0x34	PWM4_Negedge	
0x38	PWM5_Posedge	
0x3C	PWM5_Negedge	
0x40	PWM6_Posedge	
0x44	PWM6_Negedge	
0x48	PWM7_Posedge	
0x4C	PWM7_Negedge	
0x50	PWM8_Posedge	
0x54	PWM8_Negedge	
0x58	PWM9_Posedge	
0x5C	PWM9_Negedge	
0x60	PWM10_Posedge	
0x64	PWM10_Negedge	

0x68	PWM11_Posedge	
0x6C	PWM11_Negedge	
0x70	PWM12_Posedge	
0x74	PWM12_Negedge	
0x78	PWM13_Posedge	
0x7C	PWM13_Negedge	
0x80	PWM14_Posedge	
0x84	PWM14_Negedge	
0x88	PWM15_Posedge	
0x8C	PWM15_Negedge	
0x90	PWM_Output_Select	PWM 输出控制选择
0x94	Capture_Prescale	输入捕获预分频寄存器
0x98	Capture_Status	输入捕获状态寄存器
0x9C	Capture_INT_Mask	输入捕获中断掩码寄存器
0xA0	Capture_CTL	输入捕获控制寄存器
0xA4	Capture_Value0	通道 0~7 输入捕获值
0xA8	Capture_Value1	
0xAC	Capture_Value2	
0xB0	Capture_Value3	
0xB4	Capture_Value4	
0xB8	Capture_Value5	
0xBC	Capture_Value6	
0xC0	Capture_Value7	
0xE4	PWM_Update	PWM 输出同步更新
0xE8	PWM_DAC_Mode	DAC 模式选择
0xF0	PWM_Output_Value	PWM 手动输出值
0xF8	PWM_Inverter_EN	PWM 反相输出使能
0xFC	PWM_CNT_EN	PWM 计数器、分频器使能寄存器
0x100	PWM0_Prescale	PWM 通道 0~15 预分频寄存器 PWM 通道 0~15 周期寄存器
0x104	PWM0_Period	
0x108	PWM1_Prescale	
0x10C	PWM1_Period	
0x110	PWM2_Prescale	
0x114	PWM2_Period	

0x118	PWM3_Prescale
0x11C	PWM3_Period
0x120	PWM4_Prescale
0x124	PWM4_Period
0x128	PWM5_Prescale
0x12C	PWM5_Period
0x130	PWM6_Prescale
0x134	PWM6_Period
0x138	PWM7_Prescale
0x13C	PWM7_Period
0x140	PWM8_Prescale
0x144	PWM8_Period
0x148	PWM9_Prescale
0x14C	PWM9_Period
0x150	PWM10_Prescale
0x154	PWM10_Period
0x158	PWM11_Prescale
0x15C	PWM11_Period
0x160	PWM12_Prescale
0x164	PWM12_Period
0x168	PWM13_Prescale
0x16C	PWM13_Period
0x170	PWM14_Prescale
0x174	PWM14_Period
0x178	PWM15_Prescale
0x17C	PWM15_Period

13.4.2. PWM_Output_EN (Offset 0x04)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_Output_EN	R/W	16'hFF	PWM 输出使能： 1bit 对应 1 通道。 0: Enable 1: Disable

13.4.3. PWM_ENABLE (Offset 0x08)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_Enable	R/W	16'h00	通道使能。 1bit 对应 1 通道。 0: Disable 1: Enable

13.4.4. PWM0~15_Posedge (Offset 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48.....)

比特	名称	属性	复位值	描述
31:0	PWM_Posedge	R/W	32'h00	PWM 模式：周期中波形的上升沿位置。

注：PWM0_Posedge ~ PWM15_Posedge 寄存器描述相同。

13.4.5. PWM0~15_Negedge (Offset 0x14, 0x1C, 0x24, 0x2C, 0x34, 0x3C, 0x44, 0x4C.....)

比特	名称	属性	复位值	描述
31:0	PWM_Negedge	R/W	32'h00	PWM 模式：周期中波形的下降沿位置。

注：PWM0_Negedge ~ PWM15_Negedge 寄存器描述相同。

13.4.6. PWM_Output_Select (Offset 0x90)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_Output_Select	R/W	16'h00	PWM 输出控制选择。 1bit 对应 1 通道。 0: 使用 PWM_Posedge 和 PWM_Negedge 控制输出波形 1: 使用 PWM Output Value 控制输出波形。

13.4.7. Capture_Prescale (Offset 0x94)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h00	Reserved
3:0	Capture_Prescale	R/W	4'h0	输入捕获时钟预分频。基于 APB_Clock 时钟源。 0000 : 1 (default) 0001 : 2 0010 : 4 0011 : 8 0100 : 16 0101 : 32 0110 : 64 0111 : 128 1000 : 256 1001 : 512 1010 : 1024 1011 : 2048 Others: 2048

13.4.8. Capture_Status (Offset 0x98)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	Capture_Status	R/W	8'h00	输入捕获状态。 1bit 对应 1 通道。 写操作：写 '1' 清除状态位。 读操作：当前通道是否更新了捕获值。

13.4.9. Capture_INT_EN (Offset 0x9C)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	Capture_INT_EN	R/W	8'h00	输入捕获通道中断使能。 1bit 对应 1 通道。 0: Disable 1: Enable

13.4.10. Capture_CTL (Offset 0xA0)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:8	Capture_EN	R/W	8'h00	输入捕获通道使能。 1bit 对应 1 通道。 0: Disable 1: Enable
7:0	Capture_Mode	R/W	8'h00	输入捕获模式选择。 1bit 对应 1 通道。 0: 将捕获到的信号连续存入各通道的数据寄存器中。 1: 当清除 Capture_Status 对应通道位进行一次捕获，Capture_Status 对应通道置位后停止捕获。

13.4.11. Capture_Value0~7 (Offset 0xA4 ~ 0xC0)

比特	名称	属性	复位值	描述
31:0	Capture_Value	R	32'h00	输入捕获结果值。 捕获到 连续两次上升沿 之间的计数值。如果计数值超过了 32 位，则储存值为 0。

13.4.12. PWM_Update (Offset 0xE4)

比特	名称	属性	复位值	描述
31:16	PWM_Status	R	16'h00	PWM 工作指示。 1bit 对应 1 通道。 0: 对应通道停止工作 1: 对应通道正在工作
15:0	PWM_Update	R/W	16'h00	PWM 配置更新。 1bit 对应 1 通道。 写 '0' : 对应通道的 Period, Prescale, Posedge, Negedge 寄存器不更新。 写 '1' : 对应通道的 Period, Prescale, Posedge, Negedge 寄存器同步更新。

13.4.13. PWM_Output_Value (Offset 0xF0)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_Output_Value	R/W	16'h00	PWM 手动输出控制。 (当 PWM_Output_Select 选择 1 时有效) 1bit 对应 1 通道。 0: 对应通道输出高电平 1: 对应通道输出低电平

13.4.14. PWM_Inverter_EN (Offset 0xF8)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_Inverter_EN	R/W	16'h00	PWM 反相输出使能。 1bit 对应 1 通道。 0: PWM 正常输出 1: PWM 反相输出

13.4.15. PWM_CNT_EN (Offset 0xFC)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_CNT_EN	R/W	16'hFFFF	PWM 周期、预分频计数器使能。 1bit 对应 1 通道。 0: Disable 1: Enable

13.4.16. PWM0~15_Prescale

Offset: 0x100, 0x108, 0x110, 0x118, 0x120, 0x128, 0x130, 0x138.....

比特	名称	属性	复位值	描述
31:0	PWM_Prescale	R/W	32'h08	PWM 对应通道预分频。 $PWMx_Clock = APB_Clock / PWM_Prescale + 1$

注：PWM0_Prescale ~ PWM15_Prescale 寄存器描述相同。

13.4.17. PWM0~15_Period

Offset 0x104, 0x10C, 0x114, 0x11C, 0x124, 0x12C, 0x134, 0x13C.....

比特	名称	属性	复位值	描述
31:0	PWM_Period	R/W	32'h08	PWM 对应通道周期。 $PWMx_Clock = APB_Clock / PWM_Prescale + 1$ $Period = PWMx_Clock * (PWM_Period + 1)$

PWM0_Period ~ PWM15_Period 寄存器描述相同。

13.5. 使用流程

13.5.1. PWM 模式

使用普通 PWM 模式，主要配置 Prescale（预分频寄存器），Period（周期寄存器）、PWM_Posedge（上升沿位置寄存器）、PWM_Negedge（下降沿位置寄存器）。使用关系如

图所示：

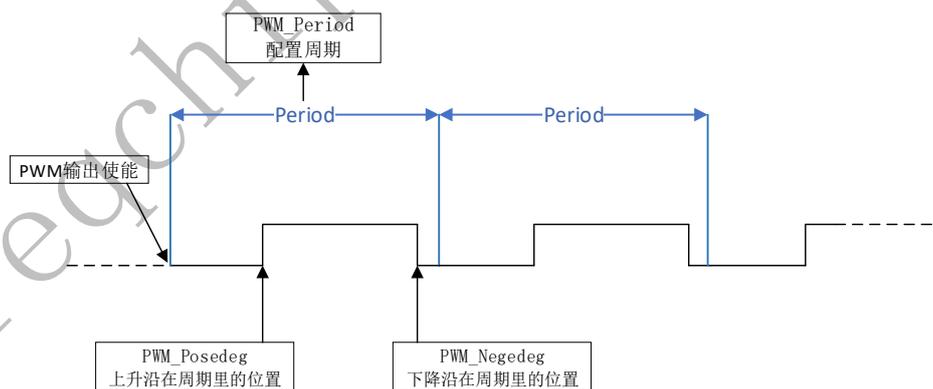


图 13-2 PWM 时序图

同步更新： 由于在配置 Prescale、Period、PWM_Posedge、PWM_Negedge 寄存器时存在时间先后的问题，导致电机等应用出现问题。可引用 PWM_Update 寄存器进行同步更新，将所需要修改的寄存器全部设置完成后，再设置 PWM_Update 进行同步更新。

异步更新：在某些简单应用中并不需要同步更新，此时可以直接修改相应寄存器完成 PWM 频率或占空比的修改。

13.5.2. Capture 输入捕获功能

芯片输入捕获默认捕捉上升沿，当捕获到两次上升沿后会自动计算出两次上升沿之间的计数值。计数值时钟由 $APB_Clock/Capture_Prescale$ 决定。

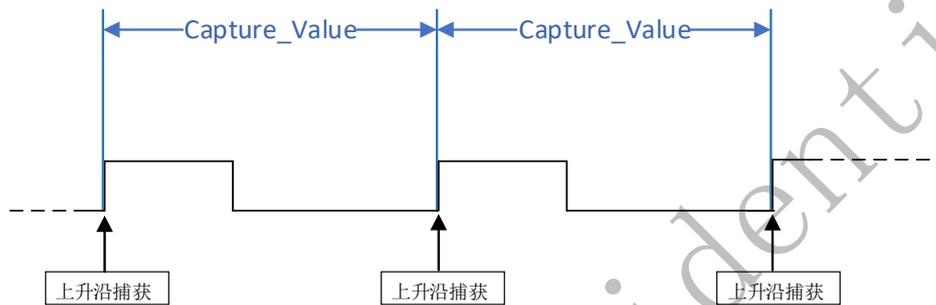


图 13-3 输入捕获时序图

14. 低功耗电源管理（PMU）

14.1. 概述

PMU 集成了电源管理，逻辑控制等功能，具体如下所述。

- 电池充电管理
- 低电压检测
- 独立看门狗
- RTC
- I/O 检测唤醒

14.2. 访问控制接口

PMU 相关寄存器，SOC 通过 frspim 接口进行访问。

14.3. PMU_CONTROL

14.3.1. PMU_CONTROL 寄存器映像

表格 14-1 PMU_CONTROL 寄存器映像

地址	寄存器	寄存器描述
0x00	PMU_CLK_SEL	PMU 域时钟源选择
0x01	FSM_CLKDIV	时钟分频配置
0x04	CLK_EN	PMU 域时钟使能
0x05	RESET_CTRL	PMU 域复位控制

14.3.2. PMU_CLK_SEL (0x00)

比特	名称	属性	复位值	描述
7:1	Reserved	R/W	7'h0	Reserved
0	PMU_CLK_SEL	R/W	1'b0	PMU 时钟源选择 0: 内部低速时钟 1: 外部低速时钟

14.3.3. FSM_CLKDIV (0x01)

比特	名称	属性	复位值	描述
7:5	FSM_OFF_DIV	R/W	3'h0	PMU power-off 时钟源分频 $PMU_CLK / ((FSM_OFF_DIV + 1) * 2)$
4:0	FSM_ON_DIV	R/W	5'b0	PMU power-on 时钟源分频 $PMU_CLK / ((FSM_ON_DIV + 1) * 2)$

14.3.4. CLK_EN (0x04)

比特	名称	属性	复位值	描述
7	Efuse_CLK_EN	R/W	1'b0	Efuse 时钟使能
6	Filter_CLK_EN	R/W	1'b0	Filter 时钟使能
5:3	Reserved	R/W	3'b0	Reserved
2	RTC_CLK_EN	R/W	1'b0	RTC 时钟使能
1	IWDT_CLK_EN	R/W	1'b0	IWDT 时钟使能
0	TickLP_CLK_EN	R/W	1'b0	睡眠 Tick 时钟使能。 0: Disable 1: Enable

14.3.5. RESET_CTRL (0x05)

比特	名称	属性	复位值	描述
7	Efuse_Reset	R/W	1'b0	Efuse 时钟使能
6	Filter_Reset	R/W	1'b0	Filter 时钟使能
5:3	Reserved	R/W	3'b0	Reserved
2	RTC_Reset	R/W	1'b0	RTC 时钟使能
1	IWDT_Reset	R/W	1'b0	IWDT 时钟使能
0	TickLP_Reset	R/W	1'b0	睡眠 Tick 时钟使能。 0: Disable 1: Enable

14.4. PMU_IWDG

14.4.1. PMU_IWDG 功能描述

IWDG（独立看门狗）使用内部低速时钟（LowPowerRC）。采用 32 位计数器，向下计数。当计数值从 iwdt_Count 配置值计数到 0 时，产生中断信号。继续从 iwdt_Timeout 配置值向下计数，若计数到 0 时，产生系统复位。

注：不配置 iwdt_Timeout 寄存器将只能产生中断，不产生系统复位。

14.4.2. PMU_IWDG 系统框图

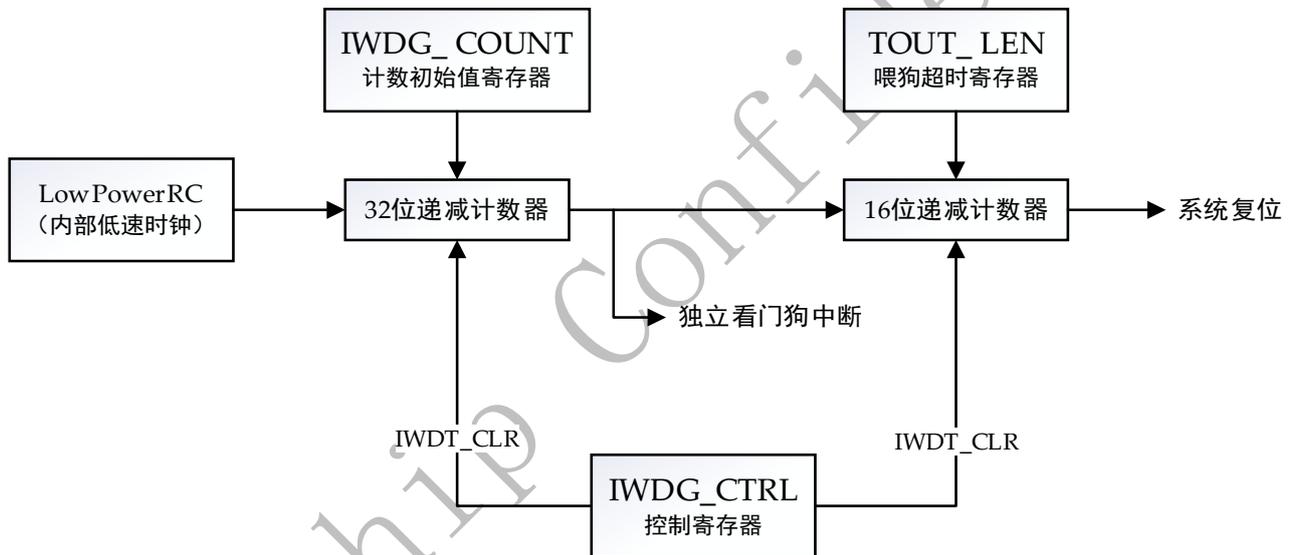


图 14-1 PMU_IWDG 系统框图

14.4.3. PMU_IWDG 寄存器映像

表格 14-2 PMU_IWDG 寄存器映像

地址	寄存器	寄存器描述
0x10	IWDG_CTRL	独立看门狗控制寄存器
0x11	IWDG_TOUT_0	独立看门狗超时寄存器 0 (LSB[7:0])
0x12	IWDG_TOUT_1	独立看门狗超时寄存器 1 (MSB[15:8])
0x13	IWDG_COUNT_0	独立看门狗计数初始值寄存器 (LSB[7:0])
0x14	IWDG_COUNT_1	独立看门狗计数初始值寄存器 ([15:8])
0x15	IWDG_COUNT_2	独立看门狗计数初始值寄存器 ([23:6])
0x16	IWDG_COUNT_3	独立看门狗计数初始值寄存器 (MSB[31:24])

14.4.4. IWDG_CTRL (0x10)

比特	名称	属性	复位值	描述
7:5	Reserved	R/W	6'h0	Reserved
4	IWDT_CLR	R/W	1'b0	喂狗清计数，从配置值重新递减计数
3	IWDT_INT_EN	R/W	1'b0	独立看门狗中断使能
2	Reserved	R/W	1'b0	Reserved
1	IWDT_RST_CHIP	R/W	1'b0	独立看门狗复位芯片使能
0	IWDT_EN	R/W	1'b0	独立看门狗使能

14.4.5. IWDG_TOUT_LEN (0x11, 0x12)

比特	名称	属性	复位值	描述
7:0	TOUT_LEN_1	R/W	8'h00	喂狗超时配置，IWDG_LEN 计数为 0 时，产生中断，TOUT_LEN 开始向下递减计数，计数为 0 时系统复位。 注：若此值为 0，将不会复位系统，只会产生中断
7:0	TOUT_LEN_0	R/W	8'h00	

14.4.6. IWDG_LEN (0x13, 0x14, 0x15, 0x16)

比特	名称	属性	复位值	描述
7:0	IWDG_LEN_3 (0x16)	R/W	8'h00	初始计数值 MSB[31:24]
7:0	IWDG_LEN_2 (0x15)	R/W	8'h00	初始计数值[23:16]
7:0	IWDG_LEN_1 (0x14)	R/W	8'h00	初始计数值[15:8]
7:0	IWDG_LEN_0 (0x13)	R/W	8'h00	初始计数值 LSB[7:0]

14.5. PMU_RTC

14.5.1. PMU_RTC 功能描述

使用内部低速时钟 (LowPowerRC)。采用 32 位计数器，向上计数。带闹钟功能，可配置两个闹钟时间。

注：使用闹钟功能时，只需使能 PMU 中断控制寄存器的闹钟 A/B 中断使能位，当计数值到达闹钟设定值时，就可以触发中断。

14.5.2. PMU_RTC 寄存器映像

表格 14-3 PMU_RTC 寄存器映像

地址	寄存器	寄存器描述
0x17	RTC_CTRL	RTC 控制寄存器
0x18	RTC_CNT	RTC 计数值[7:0]
0x19	RTC_CNT	RTC 计数值[15:8]
0x1A	RTC_CNT	RTC 计数值[23:16]
0x1B	RTC_CNT	RTC 计数值[31:24]
0x1C	ALARM_A	闹钟 A[7:0]
0x1D	ALARM_A	闹钟 A[15:8]
0x1E	ALARM_A	闹钟 A[23:16]
0x1F	ALARM_A	闹钟 A[31:24]
0x20	ALARM_B	闹钟 B[7:0]

0x21	ALARM_B	闹钟 B[15:8]
0x22	ALARM_B	闹钟 B[23:16]
0x23	ALARM_B	闹钟 B[31:24]

14.5.3. RTC_CTRL (0x17)

比特	名称	属性	复位值	描述
7	UPD_VAL_RD	R/W	1'b0	RTC 更新值读取 写 1 更新 RTC_CNT 值
7:6	Reserved	R/W	2'b00	Reserved
5	Alarm_B_EN	R/W	1'b0	闹钟 B 使能。 0: Disable 1: Enable
4	Alarm_A_EN	R/W	1'b0	闹钟 A 使能。 0: Disable 1: Enable
3	Alarm_B_CLR	R/W	1'b0	闹钟 B 复位。 写 1 硬件复位
2	Alarm_A_CLR	R/W	1'b0	闹钟 A 复位。 写 1 硬件复位
1	Reserved	R/W	1'b0	Reserved
0	RTC_EN	R/W	1'b0	RTC 使能。 0: 停止 RTC 计数 1: 开始 RTC 计数

14.5.4. RTC_CNT (0x18, 0x19, 0x1A, 0x1B)

比特	名称	属性	复位值	描述
7:0	RTC_CNT (0x1B)	R/W	8'h00	RTC 计数值[31:24]
7:0	RTC_CNT (0x1A)	R/W	8'h00	RTC 计数值[23:16]
7:0	RTC_CNT (0x19)	R/W	8'h00	RTC 计数值[15:8]
7:0	RTC_CNT (0x18)	R/W	8'h00	RTC 计数值[7:0] 写: 更新计数值 读: 当前计数值

14.5.5. ALARM_A (0x1C,0x1D,0x1E,0x1F)

比特	名称	属性	复位值	描述
7:0	Alarm_A (0x1F)	R/W	8'h00	闹钟 A 设定值[31:24]
7:0	Alarm_A (0x1E)	R/W	8'h00	闹钟 A 设定值[23:16]
7:0	Alarm_A (0x1D)	R/W	8'h00	闹钟 A 设定值[15:8]
7:0	Alarm_A (0x1C)	R/W	8'h00	闹钟 A 设定值[7:0] 写：更新闹钟值 读：闹钟值

14.5.6. ALARM_B (0x20,0x21,0x22,0x23)

比特	名称	属性	复位值	描述
7:0	Alarm_B (0x23)	R/W	8'h00	闹钟 B 设定值[31:24]
7:0	Alarm_B (0x22)	R/W	8'h00	闹钟 B 设定值[23:16]
7:0	Alarm_B (0x21)	R/W	8'h00	闹钟 B 设定值[15:8]
7:0	Alarm_B (0x20)	R/W	8'h00	闹钟 B 设定值[7:0] 写：更新闹钟值 读：闹钟值

14.6. PMU_GPIO

14.6.1. PMU_GPIO 功能描述

PMU 控制域提供了 8 个 PMU_I/O，由 PMU_IO0 ~ PMU_IO7 组成。在 SOC 进入低功耗模式时，PMU_IO 会持续工作，可以用来输出电平信号，或用来检测电平变化触发 PMU 中断，以唤醒 SOC。

14.6.2. PMU_GPIO 寄存器映像

表格 14-4 PMU_GPIO 寄存器映像

地址	寄存器	寄存器描述
0x44	PMU_IO_PE	PMU_IO 上下拉使能
0x46	PMU_IO_PS	PMU_IO 上下拉选择
0x48	PMU_IO_DAT	PMU_IO 输出状态
0x4A	PMU_IO_OEN	PMU_IO 输出使能
0x4C	PMU_IO_Monitor_EN	PMU_IO 引脚监控状态使能
0x4E	PMU_IO_Reference_Status	PMU_IO 引脚监控参考状态
0x50	PMU_IO_Monitor_Status	PMU_IO 引脚监控状态
0x52	PMU_IO_Monitor_Clear	PMU_IO 引脚监控状态清除
0x54	PMU_IO_Monitor_INT_EN	PMU_IO 引脚监控中断使能

14.6.3. PMU_IO_PE (0x44)

比特	名称	属性	复位值	描述
7:0	PMU_IO_PE	R/W	8'hFF	PMU_IO 上下拉使能： 0：禁用上下拉 1：使能上下拉 Bit[0]对应 PMU_IO0，1bit 对应 1 引脚

14.6.4. PMU_IO_PS (0x46)

比特	名称	属性	复位值	描述
7:0	PMU_IO_PS	R/W	8'h00	PMU_IO 上下拉选择： 0: 下拉 1: 上拉 Bit[0]对应 PMU_IO0, 1bit 对应 1 引脚

14.6.5. PMU_IO_DAT (0x48)

比特	名称	属性	复位值	描述
7:0	PMU_IO_Data	R/W	8'h00	PMU_IO 状态： IO 配置为输出时写操作控制 IO 状态，配置为输入时读为引脚状态。 0: 低电平 1: 高电平 Bit[0]对应 PMU_IO0, 1bit 对应 1 引脚

14.6.6. PMU_IO_OEN (0x4A)

比特	名称	属性	复位值	描述
7:0	PMU_IO_OEN	R/W	8'hFF	PortA 输入/输出选择： 0: Output 1: Input Bit[0]对应 PMU_IO0, 1bit 对应 1 引脚

14.6.7. PMU_IO_Monitor_EN (0x4C)

比特	名称	属性	复位值	描述
7:0	PMU_IO_Monitor_EN	R/W	8'h00	PMU_IO 引脚监控状态使能： 0: Enable 1: Disable Bit[0]对应 PMU_IO0，1bit 对应 1 引脚

14.6.8. PMU_IO_Reference_Status (0x4E)

比特	名称	属性	复位值	描述
7:0	PMU_IO_Reference_Status	R/W	8'h00	PMU_IO 引脚监控状态参考值。 例：写入 0x01。并使能 PMU_IO0 引脚监控状态使能寄存器 ，则当检测到 PortA0 由高变低时触发中断。

14.6.9. PMU_IO_Monitor_Status (0x50)

比特	名称	属性	复位值	描述
7:0	PMU_IO_Monitor_Status	R	8'h00	PMU_IO 引脚监控状态值。 指示引脚是否发生变化。 例：读 0x01，表示引脚 PMU_IO0 被监控到变化。

14.6.10. PMU_IO_Monitor_Clear (0x52)

比特	名称	属性	复位值	描述
7:0	PMU_IO_Monitor_Clear	W	8'h00	PMU_IO 引脚监控状态清除。 写 '1' 清除 PMU_IO_Monitor_Status 对应 bit。

14.6.11. PMU_IO_Monitor_INT_EN (0x54)

比特	名称	属性	复位值	描述
7:1	Reserved	R/W	7'b00	Reserved
0	PMU_IO_Monitor_INT_EN	R/W	1'b0	引脚监控中断使能

14.7. PMU_INT_CTRL

14.7.1. PMU_INT_CTRL 功能描述

包含了 PMU 域的中断管理，中断开启/关闭，中断清除，中断查询；

中断产生：首先使能 PMU 子模块的中断请求，再使能 PMU 中断管理 INT_EN0 (0x32)，INT_EN1 (0x33) 寄存器中的对应位，即可触发 PMU 中断。

中断清除：清除中断状态时，使用 PMU 子模块中的清除功能。

14.7.2. PMU_INT_CTRL 寄存器映像

表格 14-5 PMU_INT_CTRL 寄存器映像

地址	寄存器	寄存器描述
0x32	INT_EN0	PMU 中断使能寄存器 0
0x33	INT_EN1	PMU 中断使能寄存器 1
0x34	INT_RAW0	PMU 原始中断状态寄存器 0
0x35	INT_RAW1	PMU 原始中断状态寄存器 1
0x36	INT_STS0	PMU 中断状态寄存器 0
0x37	INT_STS1	PMU 中断状态寄存器 1
0x38	Analog_Status	PMU 模拟信号状态寄存器
0x39	Analog_Monitor_Level	PMU 模拟信号监控参考电平
0x3A	ADKey_Filter	AD Key 滤波配置
0x3B	LVD_Filter	低压检测滤波配置
0x3C	Charge_ACOK_Filter	充电插/拔检测滤波配置
0x3D	Charge_FULL_Filter	充电满检测滤波配置
0x3F	OTD_Filter	过温检测滤波配置

14.7.3. INT_EN0 (0x32)

比特	名称	属性	复位值	描述
7	PMU_IO_Monitor_INT_EN	R/W	1'b0	PMU_IO_Monitor 中断使能 0: Disable 1: Enable
6	GPIO_Wakeup1_INT_EN	R/W	1'b0	GPIO_Wakeup1 中断使能 0: Disable 1: Enable
5	GPIO_Wakeup0_INT_EN	R/W	1'b0	GPIO_Wakeup0 中断使能 0: Disable 1: Enable
4	Reserved	R/W	1'b0	Reserved
3	Keyscan_INT_EN	R/W	1'b0	Keyscan 中断使能 0: Disable 1: Enable
2	AlarmB_INT_EN	R/W	1'b0	RTC AlarmB 中断使能 0: Disable 1: Enable
1	AlarmA_INT_EN	R/W	1'b0	RTC AlarmA 中断使能 0: Disable 1: Enable
0	IWDT_INT_EN	R/W	1'b0	IWDT 中断使能 0: Disable 1: Enable

14.7.4. INT_EN1 (0x33)

比特	名称	属性	复位值	描述
7	OTD	R/W	1'b0	OTD 中断使能 0: Disable 1: Enable

6	Charge_4P35_INT_EN	R/W	1'b0	Charge 4P35 中断使能 0: Disable 1: Enable
5	Reserved	R/W	1'b0	Reserved
4	Charge_FULL_INT_EN	R/W	1'b0	Charge FULL 中断使能 0: Disable 1: Enable
3	Charge_ACOK_INT_EN	R/W	1'b0	Charge ACOK 中断使能 0: Disable 1: Enable
2	LVD_INT_EN	R/W	1'b0	LVD 中断使能 0: Disable 1: Enable
1	ADKey1_INT_EN	R/W	1'b0	ADKey1 中断使能 0: Disable 1: Enable
0	ADKey0_INT_EN	R/W	1'b0	ADKey0 中断使能。 0: Disable 1: Enable

14.7.5. INT_RAW0 (0x34)

比特	名称	属性	复位值	描述
7	PMU_IO_Monitor_INT_RAW	R	1'b0	PMU_IO_Monitor 中断原始状态
6	GPIO_Wakeup1_INT_RAW	R	1'b0	GPIO_Wakeup1 中断原始状态
5	GPIO_Wakeup0_INT_RAW	R	1'b0	GPIO_Wakeup0 中断原始状态
4	Reserved	R	1'b0	Reserved
3	Keyscan_INT_RAW	R	1'b0	Keyscan 中断原始状态
2	AlarmB_INT_RAW	R	1'b0	RTC AlarmB 中断原始状态
1	AlarmA_INT_RAW	R	1'b0	RTC AlarmA 中断原始状态
0	IWDT_INT_RAW	R	1'b0	IWDT 中断原始状态

14.7.6. INT_RAW1 (0x35)

比特	名称	属性	复位值	描述
7	OTD_INT_RAW	R	1'b0	OTD 中断原始状态
6	Charge_4P35_INT_RAW	R	1'b0	Charge 4P35 中断原始状态
5	Reserved	R	1'b0	Reserved
4	Charge_FULL_INT_RAW	R	1'b0	Charge FULL 中断原始状态
3	Charge_ACOK_INT_RAW	R	1'b0	Charge ACOK 中断原始状态
2	LVD_INT_RAW	R	1'b0	LVD 中断原始状态
1	ADKey1_INT_RAW	R	1'b0	ADKey1 中断原始状态
0	ADKey0_INT_RAW	R	1'b0	ADKey0 中断原始状态

14.7.7. INT_STS0 (0x36)

比特	名称	属性	复位值	描述
7	PMU_IO_Monitor_INT_STS	R	1'b0	PMU_IO_Monitor 中断状态
6	GPIO_Wakeup1_INT_STS	R	1'b0	GPIO_Wakeup1 中断状态
5	GPIO_Wakeup0_INT_STS	R	1'b0	GPIO_Wakeup0 中断状态
4	Reserved	R	1'b0	Reserved
3	Keyscan_INT_STS	R	1'b0	Keyscan 中断状态
2	AlarmB_INT_STS	R	1'b0	RTC AlarmB 中断状态
1	AlarmA_INT_STS	R	1'b0	RTC AlarmA 中断状态
0	IWDT_INT_STS	R	1'b0	IWDT 中断状态

14.7.8. INT_STS1 (0x37)

比特	名称	属性	复位值	描述
7	OTD_INT_STS	R	1'b0	OTD 中断状态
6	Charge_4P35_INT_STS	R	1'b0	Charge 4P35 中断状态

5	Reserved	R	1'b0	Reserved
4	Charge_FULL_INT_STS	R	1'b0	Charge FULL 中断状态
3	Charge_ACOK_INT_STS	R	1'b0	Charge ACOK 中断状态
2	LVD_INT_STS	R	1'b0	LVD 中断状态
1	ADKey1_INT_STS	R	1'b0	ADKey1 中断状态
0	ADKey0_INT_STS	R	1'b0	ADKey0 中断状态

14.7.9. Analog_Status (0x38)

比特	名称	属性	复位值	描述
7	Analog_OTD_STS	R	1'b0	过温检测，模拟信号状态
6	Analog_Charge_4P35_STS	R	1'b0	充电满 4.35V，模拟信号状态
5	Reserved	R	1'b0	Reserved
4	Analog_Charge_FULL_STS	R	1'b0	充电满到充电阈值，模拟信号状态
3	Analog_Charge_ACOK_STS	R	1'b0	充电插/拔，模拟信号状态
2	Analog_LVD_STS	R	1'b0	低电压检测，模拟信号状态
1	Analog_ADKey1_STS	R	1'b0	ADKey1 模拟信号状态
0	Analog_ADKey0_STS	R	1'b0	ADKey0 模拟信号状态

14.7.10. Analog_Monitor_Level (0x39)

比特	名称	属性	复位值	描述
7	OTD_Monitor_level	R	1'b0	过温检测，中断触发状态选择 0: 芯片温度超过检测阈值 1: 芯片温度未超过检测阈值
6	Charge_4P35_Monitor_level	R	1'b0	充电满 4.35V，中断触发状态选择 0: 充电已满 4.35V 触发中断 1: 充电未满 4.35V 触发中断
5	Reserved	R	1'b0	Reserved
4	Charge_FULL_Monitor_level	R	1'b0	充电满，中断触发状态选择

				0: 充电满触发中断 1: 充电不满触发中断
3	Charge_ACOK_Monitor_level	R	1'b0	充电插/拔, 中断触发状态选择 0: 充电电源插入 1: 充电电源拔出
2	LVD_Monitor_level	R	1'b0	低压检测, 中断触发状态选择 0: VBAT 电压降低至检测阈值 1: VBAT 电压未家底至检测阈值
1	ADKey1_Monitor_level	R	1'b0	ADKey1, 中断触发状态选择。 0: ADKey 状态为 '1' 时触发 1: ADKey 状态为 '0' 时触发
0	ADKey0_Monitor_level	R	1'b0	ADKey0, 中断触发状态选择。 0: ADKey 状态为 '1' 时触发 1: ADKey 状态为 '0' 时触发

14.7.11. ADKey_Filter (0x3A)

比特	名称	属性	复位值	描述
7:0	ADKey_Filter	R/W	8'h00	ADKey 滤波设置

14.7.12. LVD_Filter (0x3B)

比特	名称	属性	复位值	描述
7:0	LVD_Filter	R/W	8'h00	低电压检测滤波设置

14.7.13. Charge_ACOK_Filter (0x3C)

比特	名称	属性	复位值	描述
7:0	Charge_ACOK_Filter	R/W	8'h00	充电拔/插检测滤波设置

14.7.14. Charge_FULL_Filter (0x3D)

比特	名称	属性	复位值	描述
7:0	Charge_ACOK_Filter	R/W	8'h00	充电拔/插检测滤波设置

14.7.15. OTD_Filter (0x3F)

比特	名称	属性	复位值	描述
7:0	OTD_Filter	R/W	8'h00	过温检测 滤波设置

15. 液晶屏并行驱动接口 (PARALLEL_INTERFACE)

15.1. 概述

PARALLEL_INTERFACE 接口符合标准的 8080, 6800 总线时序, 可用来驱动符合 8080 时序或者符合 6800 时序的液晶显示屏, 或其他外设器件。

15.2. 主要特性

- 支持 8080, 6800 时序。
- Tx FIFO 高达 128 字节。
- 总线 8BIT、16BIT 可选。

15.3. PARALLEL 结构框图

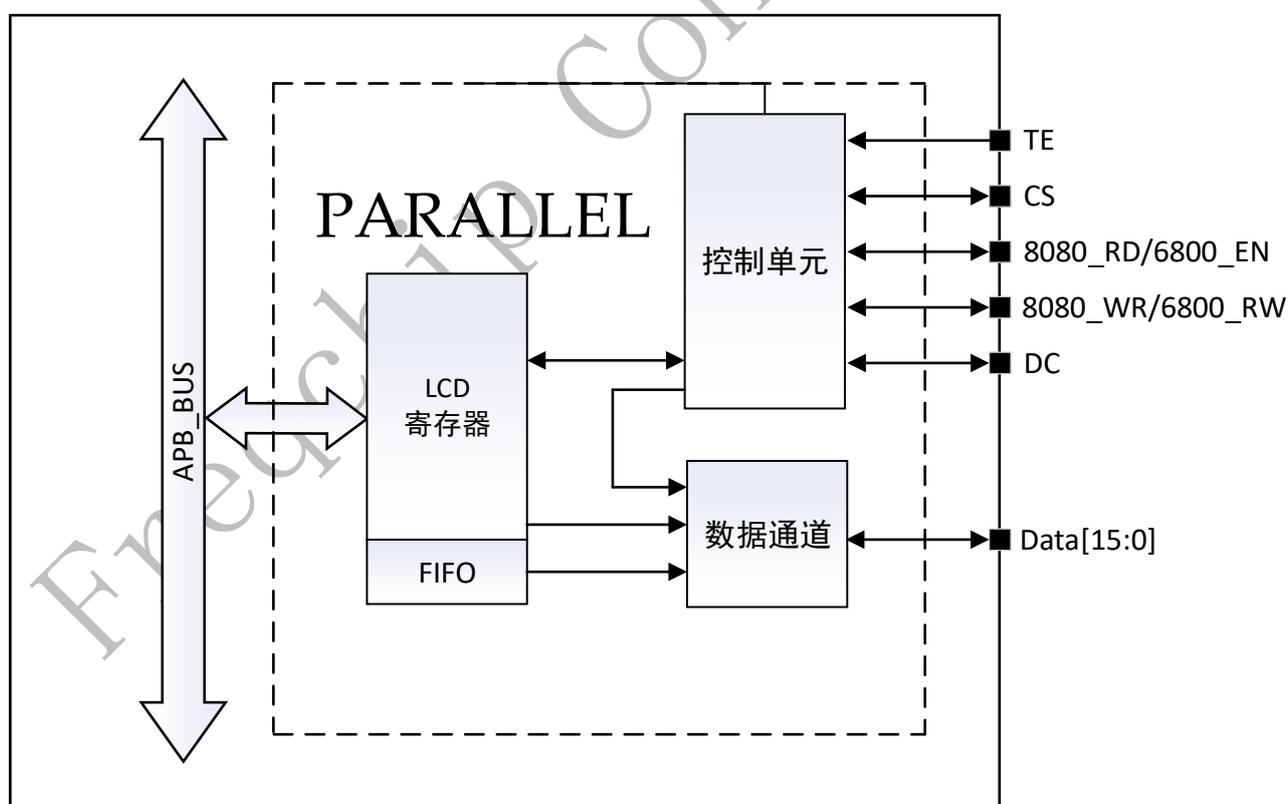


图 15-1 PARALLEL 结构框图

15.4. PARALLEL 寄存器

15.4.1. PARALLEL 寄存器地址映像

PARALLEL 寄存器基地址：0x500D_0000

表格 15-1 LCD 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	INTF_CFG	接口配置寄存器
0x04	CSX	CS 线控制寄存器
0x08	CRM	读写时钟配置
0x0C	BUS_STATUS	状态寄存器
0x10	CFG	控制寄存器
0x14	DATA_WR_LEN	数据类型寄存器
0x18	DATA_CFG	数据配置寄存器
0x1C	TX_FIFO	Tx FIFO 寄存器
0x20	RD_REQ	读请求寄存器
0x24	DAT_RD	读数据寄存器
0x28	TXFF_AEMP_LV	Tx FIFO 剩余数据深度
0x2C	TXFF_CLR	Tx FIFO 清除寄存器
0x30	INT_CONTROL	中断使能寄存器
0x34	INT_STATUS	中断查询、清除寄存器
0x38	DMA	DMA 配置寄存器

15.4.2. INTF_CFG (Offset 0x00)

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3	PARA_WIDTH	R/W	1'b0	总线位宽选择 0: 8bit 1: 16bit
2	P68_MODE	R/W	1'b0	模式选择 0: 8080 时序

				1: 6800 时序
1	CS_POLARITY	R/W	1'b0	CS 极性选择 0: low active 1: high active
0	DC_POLARITY	R/W	1'b0	DC 极性选择 0: low active 1: high active

15.4.3. CSX (Offset 0x04)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	CSX	R/W	1'b1	总线 CS 信号控制 1: 拉高总线。 0: 拉低总线。

15.4.4. CRM (Offset 0x08)

比特	名称	属性	复位值	描述
31:15	Reserved	R	0x0	Reserved
14:12	WR_H_LEN	R/W	3'b000	DAT WR 操作时, 总线 WRCLK 高电平持续的个数
11	Reserved	R	0x0	Reserved
10:8	WR_L_LEN	R/W	3'b000	DAT WR 操作时, 总线 WRCLK 低电平持续的个数
7:4	RDCLK_CFG	R/W	4'h0	以 m_clk 为时钟源, 对 rdclk 进行分频: 3: 4 分频 4: 6 分频 5: 8 分频 6: 16 分频 7: 32 分频 8: 64 分频
3	Reserved	R	0x0	Reserved
2:0	WRCLK_CFG	R/W	3'b000	以 m_clk 为时钟源, , 对 wrclk 进行分频:

				0: 1分频 1: 2分频 2: 3分频 3: 4分频 4: 6分频 5: 8分频
--	--	--	--	--

15.4.5. BUS_STATUS (Offset 0x0C)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	BUS_BUSY	R	1'b0	总线忙信号： 写操作前，软件需查询 BUS_BUSY 是否为 0，当为 0 时方可进行写操作， 读操作前，当查询到 BUS_BUSY 为 0 时，可发起读请求；读请求发起后，当再次查询到 BUS_BUSY 为 0 时，可读取数据（读 LCD_DAT_RD 寄存器）。

15.4.6. CFG (Offset 0x10)

比特	名称	属性	复位值	描述
31:17	Reserved	R	0x0	Reserved
16	WR_CDX	R/W	1'b0	写类型： 1:参数 0:命令
15:0	WR_DATA	R/W	16'h00	写命令或写参数。8 线模式下传输低 8bit

15.4.7. DATA_WR_LEN (Offset 0x14)

比特	名称	属性	复位值	描述
31:24	Reserved	R	0x0	Reserved
23:0	BURST_WR_LEN	R/W	24'h00	BURST 写数据个数

15.4.8. DATA_CFG (Offset 0x18)

比特	名称	属性	复位值	描述
31:8	Reserved	R	8'h00	Reserved
7:6	DATA_TRANS_SEQ_3	R/W	2'b11	8bit bus 模式下，第四次传输数据选择，或 16bit bus 模式下，第二次传输数据 MSB 选择： 00: FIFO_DATA [7:0] 01: FIFO_DATA [15:8] 10: FIFO_DATA [23:16] 11: FIFO_DATA [31:24]
5:4	DATA_TRANS_SEQ_2	R/W	2'b10	8bit bus 模式下，第三次传输数据选择，或 16bit bus 模式下，第二次传输数据 LSB 选择： 00: FIFO_DATA [7:0] 01: FIFO_DATA [15:8] 10: FIFO_DATA [23:16] 11: FIFO_DATA [31:24]
3:2	DATA_TRANS_SEQ_1	R/W	2'b01	8bit bus 模式下，第二次传输数据选择，或 16bit bus 模式下，第一次传输数据 MSB 选择： 00: FIFO_DATA [7:0] 01: FIFO_DATA [15:8] 10: FIFO_DATA [23:16] 11: FIFO_DATA [31:24]
1:0	DATA_TRANS_SEQ_0	R/W	2'b00	8bit bus 模式下，第一次传输数据选择，或 16bit bus 模式下，第一次传输数据 LSB 选择： 00: FIFO_DATA [7:0] 01: FIFO_DATA [15:8] 10: FIFO_DATA [23:16] 11: FIFO_DATA [31:24]

15.4.9. TX_FIFO (Offset 0x1C)

比特	名称	属性	复位值	描述
31:0	DAT_WR	W	0x0	往 TX FIFO 中 PUSH 数据。 完成一次 BURST 发送后, 当 FIFO 为空, 并查询到 BUS_BUSY 为 0 后, 可发起下一次 BURST 发送。

15.4.10. RD_REQ (Offset 0x20)

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved
0	RD_REQ	R/W	1'b0	读请求。往该 bit 写 1, 则在总线上发起一次读操作。

15.4.11. DAT_RD (Offset 0x24)

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	DAT_RD	R/W	15'h00	发起读请求后, 当查询到 BUS_BUSY 为 0, 可获取总线上读到的数据。

15.4.12. TXFF_AEMP_LV (Offset 0x28)

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	TXFF_AEMP_LV	R/W	4'h00	TX FIFO almost 空阈值

15.4.13. TXFF_CLR (Offset 0x2C)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	TXFF_RST	R/W	1'b0	复位 TX FIFO 1: reset fifo internal logic

				0: release
1	TXFF_PUSH_CLR	R/W	1'b0	复位写 FIFO 指针 1: reset push pointer 0: release
0	TXFF_POP_CLR	R/W	1'b0	复位读 FIFO 指针 1: reset pop pointer 0: release

15.4.14. INT_CONTROL (Offset 0x30)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	TXFIFO_AEMPT_INEN	R/W	1'b0	TX FIFO almost 空中断使能 0: Disable 1: Enable
1	TXFIFO_EMPTY_INEN	R/W	1'b0	TX FIFO 空中断使能 0: Disable 1: Enable
0	TXFIFO_FULL_INEN	R/W	1'b0	TX FIFO 满中断使能 0: Disable 1: Enable

15.4.15. INT_STATUS (Offset 0x34)

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	TXFIFO_AEMPT	R	1'b0	TX FIFO almost 空状态
1	TXFIFO_EMPTY	R	1'b0	TX FIFO 空状态
0	TXFIFO_FULL	R	1'b0	TX FIFO 满状态

15.4.16. DMA (Offset 0x38)

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5	DMA_CR	R/W	1'b0	DMA 使能 0: Disable 1: Enable
4:0	DMA_TDLR	R/W	5'h16	dma tx data level register: 当 txff 中数据个数小于等于该阈值时, dma_tx_req 信号置起

15.5. 使用流程

15.5.1. 写命令或写参数

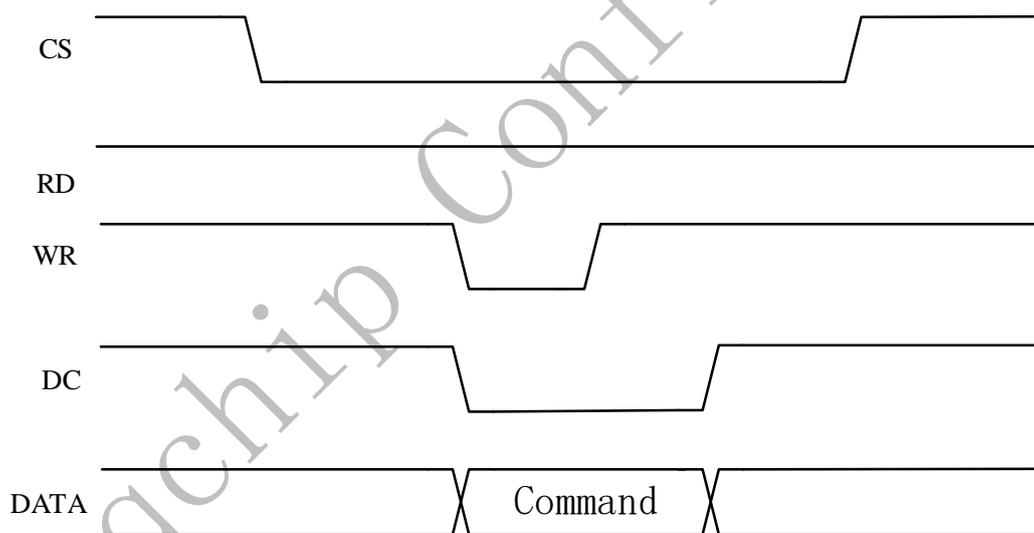


图 15-2 8080 写命令时序

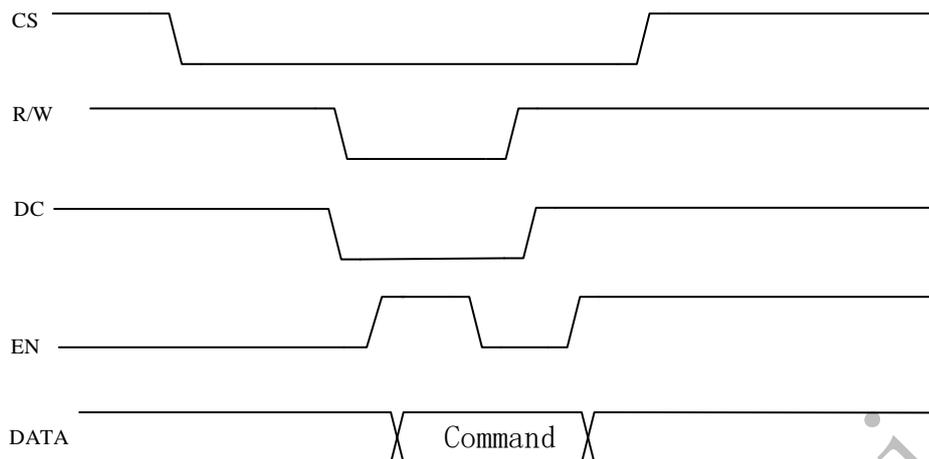


图 15-3 6800 写命令时序

1. 首先拉低 CSX 信号。
2. 配置 CFG 寄存器。如果需要发送 CMD，则 WR_CDX 应配置为 0。如果是发送参数，则 WR_CDX 应配置为 1。将命令或参数写入 WR_DATA。
3. 读取 BUS_BUSY 状态，确保上次传输已完成，若再次传输重复步骤 2、3 即可。
4. 最后拉高 CSX 信号，结束。

15.5.2. 写数据

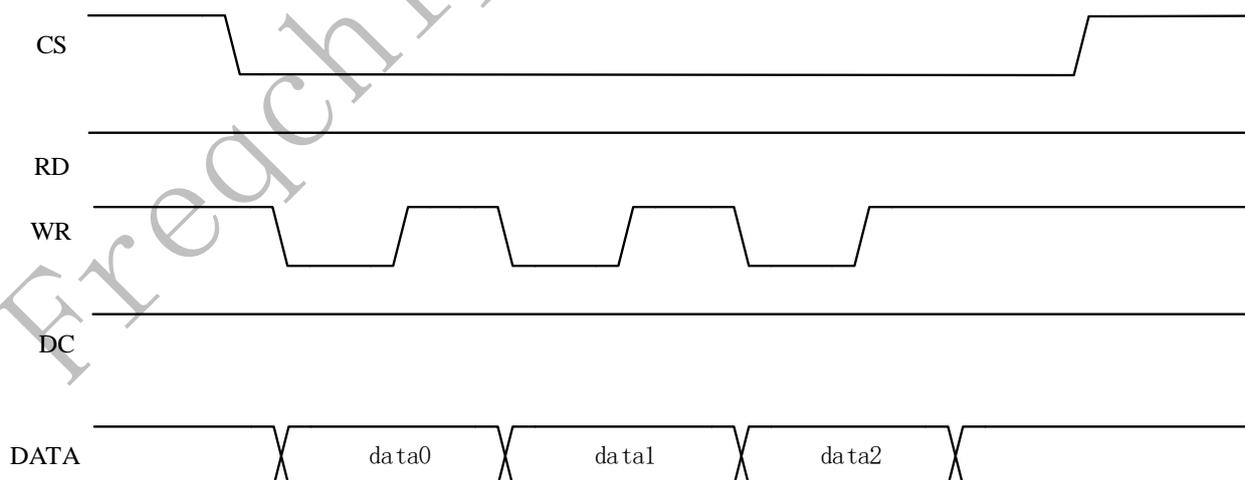


图 15-4 8080 写数据时序

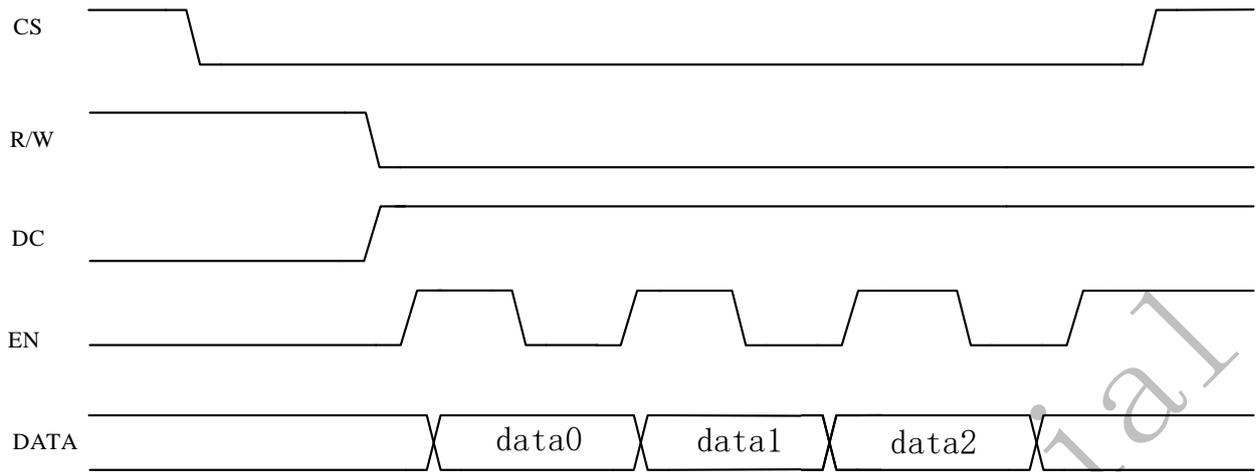


图 15-5 6800 写数据时序

1. 首先拉低 CSX 信号。
2. 根据所需传输数据的个数，配置 DAT_WR_LEN。
3. 向 FIFO 中写入数据，读取 BUS_BUSY 状态，确保传输已完成。
4. 最后拉高 CSX 信号，结束。

15.5.3. 读数据

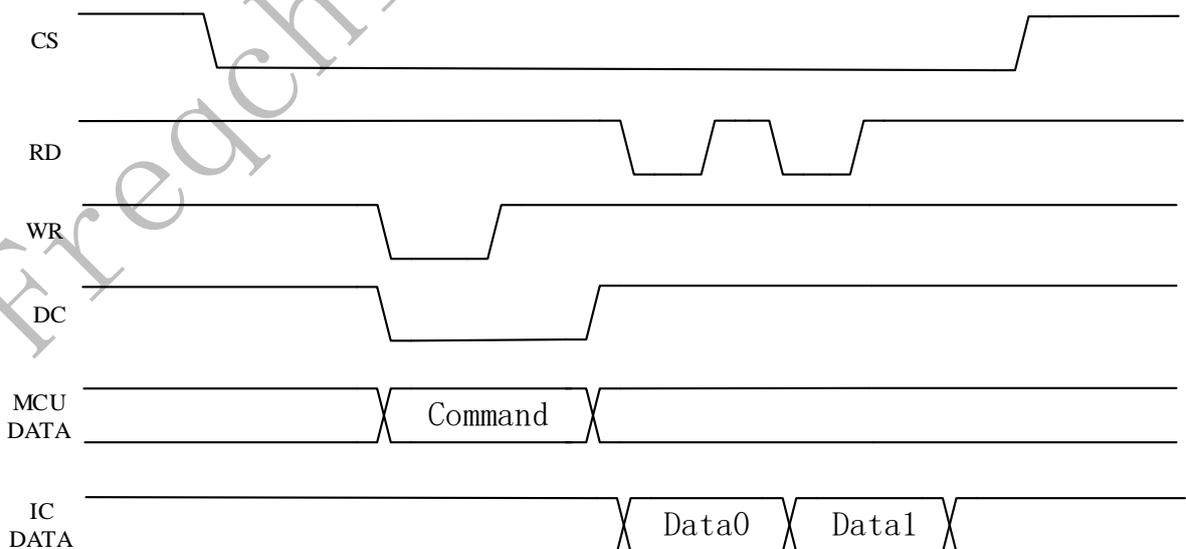


图 15-6 8080 读数据时序

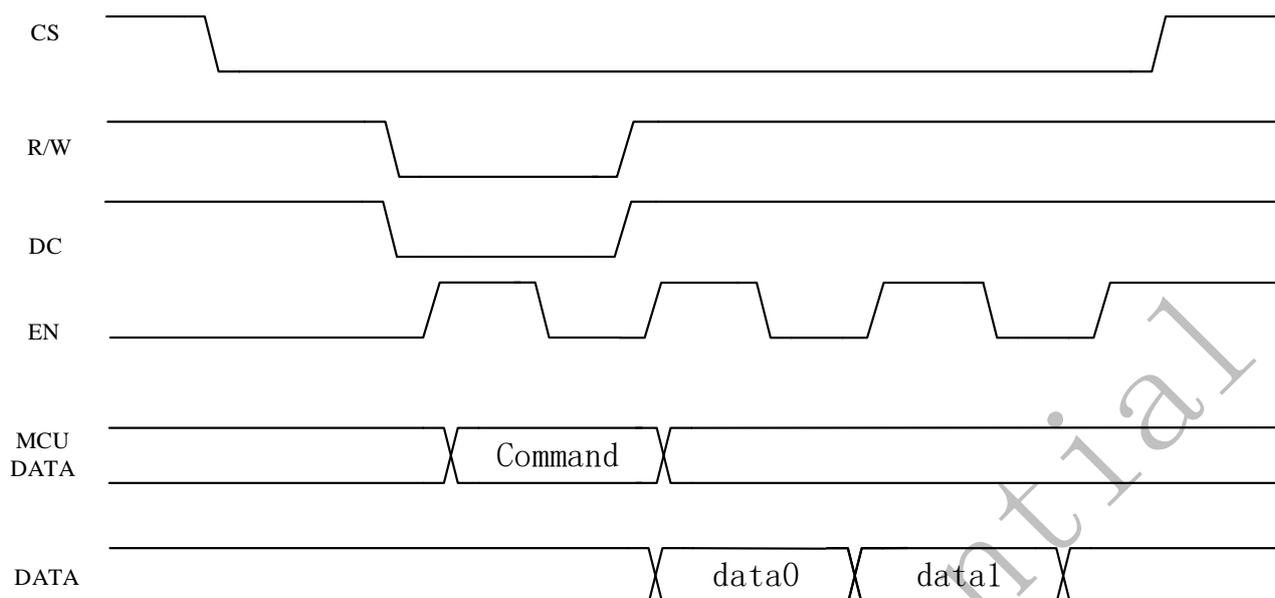


图 15-7 6800 读数据时序

1. 首先拉低 CSX 信号。
2. 配置 RD_REQ，每请求一次在总线上读取一次数据。
3. 读取 BUS_BUSY 状态，确保数据已准备好，读取 DAT_RD 获取一次数据，重复 2 和 3 步骤直到读完所有数据。
4. 最后拉高 CSX 信号，结束。

16. 通用定时器（Timer）

16.1. 概述

提供了四路 32bit 定时器，向下计数，支持循环计数，支持中断触发。每个定时器都是完全独立的，没有互相共享任何资源。

16.2. 主要特性

- 32bit 计数
- 支持循环计数

16.3. Timer 寄存器

16.3.1. Timer 寄存器地址映像

Timer0 寄存器基地址：0xE006_0000

Timer1 寄存器基地址：0xE006_0014

Timer2 寄存器基地址：0xE006_8000

Timer3 寄存器基地址：0xE006_8014

表格 16-1 Timer 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	LOAD_VALUE	计数值装载寄存器
0x04	COUNT_VALUE	当前计数寄存器
0x08	CONTROL	控制寄存器
0x0C	INT_CLR	中断清除寄存器
0x10	INT_STATUS	中断查询寄存器

16.3.2. LOAD_VALUE (Offset 0x00)

比特	名称	属性	复位值	描述
31:0	LOAD	R/W	32'h00	定时器初始值

16.3.3. COUNT_VALUE (Offset 0x04)

比特	名称	属性	复位值	描述
31:0	CNT	R/W	32'hFFFFFFFF	当前计数值

16.3.4. CONTROL (Offset 0x08)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	0x0	Reserved
2	INT_MASK	R/W	1'b1	中断使能 0: Enable 1: Disable
1	CNT_MODE	R/W	1'b0	计数模式选择: 0: 单次计数 1: 循环计数
0	CNT_EN	R/W	1'b0	计数使能: 1: 开始计数 0: 停止计数

16.3.5. INT_CLR (Offset 0x0C)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
15:0	INT_CLR	R	16'h00	读清除中断状态。

16.3.6. INT_STATUS (Offset 0x10)

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
15:0	INT_STATUS	R	16'h00	读出中断状态。

16.4. 使用流程

1. 写 LOAD_VALUE 寄存器，配置定时器初值。
2. 写 CONTROL 寄存器，配置定时器分频，选择循环计数，使能计数（定时器会自动使能定时器中断）。
3. 定时器开始从初始值向下计数，直到计数为 0 触发中断。
4. 写 INT_CLR 寄存器清除中断。

17. 数模转换器（ADC）

17.1. 概述

10 位 ADC（模拟数字转换器）。提供多达 8 个通道，可测量外部输入模拟信号或内部模拟信号。

17.2. 主要特性

- 10 位分辨率
- 8 个独立的采样通道
- 连续或单次转换模式
- 支持电池电压检测
- 核心温度检测

17.3. 结构框图

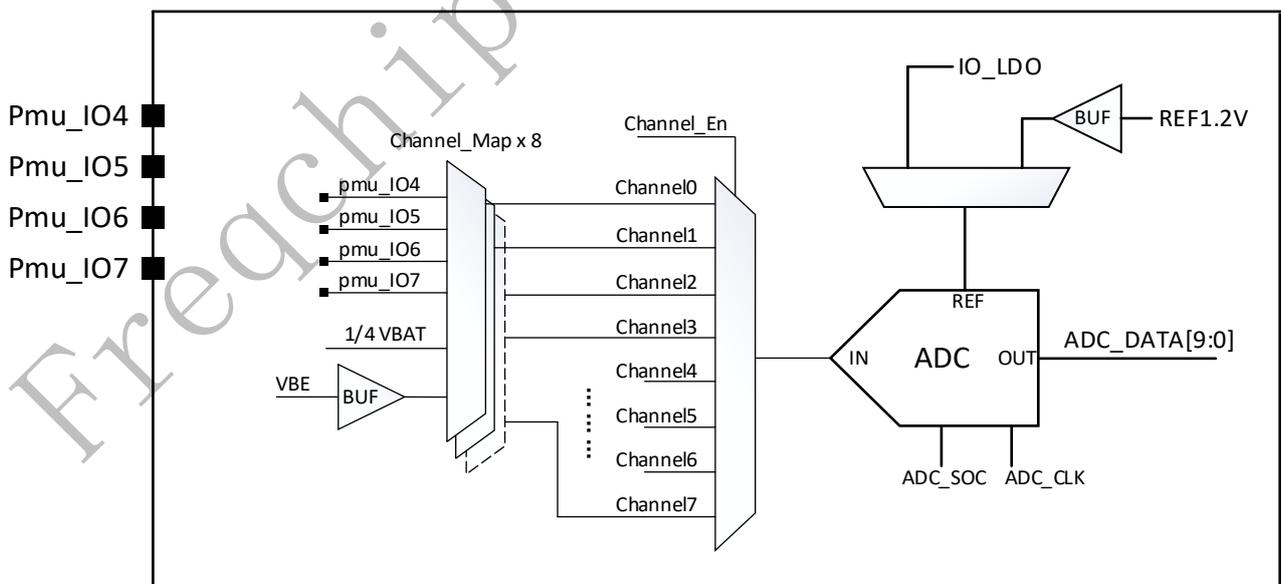


图 17-1 ADC 结构框图

17.4. 功能描述

ADC 控制器共有 8 路通道，每路通道的测量源可以选择外部输入电压（pmu_io4, pmu_io5, pmu_io6, pmu_io7）、电池电量（1/4VBAT）或核心温度（VBE）。

参考源可选择内部 1.2V 基准源，或 IOLDO。

转换模式可以选择软件单次触发，或硬件自动循环触发。

ADC 采样时钟使用固定时钟。 $ADC_{CLK} = \frac{24MHz}{(2+1) \times 2} = 4MHz$

ADC 完成一次转换时间约为 3us。

17.5. ADC 寄存器

17.5.1. ADC 寄存器映像

ADC 寄存器基地址：0xE00E_0000

表格 17-1 ADC 寄存器映像

偏移	寄存器	寄存器描述
0x00	Control	控制寄存器
0x04	Config	配置寄存器
0x08	Timing	时序控制寄存器
0x18	Channel_Status	通道状态寄存器
0x20	ADC_INT_Enable	中断使能寄存器
0x24	ADC_INT_Status	中断状态寄存器
0x28	ADC_INT_Raws	中断原始状态寄存器
0x2C	ACT0	模拟控制寄存器 0
0x30	ACT1	模拟控制寄存器 1
0x34	ChannelMap[0]	通道选择寄存器

偏移	寄存器	寄存器描述
0x38	ChannelMap[1]	
0x3C	ChannelMap[2]	
0x40	ChannelMap[3]	
0x44	ChannelMap[4]	
0x48	ChannelMap[5]	
0x4C	ChannelMap[6]	
0x50	ChannelMap[7]	
0x74	ChannelData[0]	通道数据寄存器
0x78	ChannelData[1]	
0x7C	ChannelData[2]	
0x80	ChannelData[3]	
0x84	ChannelData[4]	
0x88	ChannelData[5]	
0x8C	ChannelData[6]	
0x90	ChannelData[7]	

17.5.2. Control (Offset 0x00)

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h0	Reserved
3	ADC_SoftTrigger	R/W	1'b0	软件手动触发转换。 写 '1' 开始一次转换，写 '0' 无效。
2	Reserved	R/W	1'b0	Reserved
1	ADC_Reset	R/W	1'b0	ADC 复位。 写 '1' 复位，硬件自动清 0。
0	ADC_ConvertEN	R/W	1'b0	ADC 转换使能。 0: 停止转换 1: 开启转换

17.5.3. Config (Offset 0x04)

比特	名称	属性	复位值	描述
31:13	Reserved	R/W	19'h0	Reserved
12:9	SoftTriggerChannel	R/W	4'b0	软件触发选中通道。 仅在触发模式选择软件手动触发时有效。 选择转换通道 0 ~ 7。
8:7	Reserved	R/W	2'b0	Reserved
6	ADC_SoftTrigger	R/W	1'b0	手动触发转换。 仅在触发模式选择软件手动触发时有效。 写 '1' 开始一次转换, 写 '0' 无效。
5	TriggerMode	R/W	1'b0	触发模式。 0: 软件手动触发 1: 硬件自动触发
4	ConvertMode	R/W	1'b0	转换模式。 0: 单次转换 1: 循环多次转换
3:0	ChannelMax	R/W	1'b0	使用硬件循环转换时最大通道数量。 最大通道数量 = ChannelMax + 1

17.5.4. Timing (Offset 0x08)

比特	名称	属性	复位值	描述
31:28	TimeoutCycle	R/W	5'b0	转换超时周期。
27:22	SampleCycle	R/W	5'b0	采样周期。
21:16	SetupCycle	R/W	6'b0	采样建立周期。
15:8	Reserved	R/W	8'b0	Reserved
7:0	Clock_DIV	R/W	8'b0	ADC 采样时钟分频。 $48\text{MHz} / ((\text{CLK_DIV} + 1) * 2)$ Clock_DIV 固定配置 2。

17.5.5. Channel_Status (Offset 0x18)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'b0	Reserved
7:0	Channel_Data_Valid	R/W	8'b0	通道数据有效标志。 通道转换完成后对应 bit 置位，读取通道数据后自动清 '0'。

17.5.6. ADC_INT_Enable (Offset 0x20)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	27'b0	Reserved
4	Error_INT_EN	R/W	1'b0	转换发生错误中断使能。
3	Channel_Valid_INT_EN	R/W	1'b0	通道转换完成中断使能。 注：任意通道完成转换都会触发中断。
2:0	Reserved	R/W	2'b0	Reserved

17.5.7. ADC_INT_Status (Offset 0x24)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	27'b0	Reserved
4	Error_INT_Status	R/W	1'b0	转换发生错误中断状态。
3	Channel_Valid_INT_Status	R/W	1'b0	通道转换完成中断状态。
2:0	Reserved	R/W	2'b0	Reserved

17.5.8. ADC_INT_Raws (Offset 0x28)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	27'b0	Reserved

比特	名称	属性	复位值	描述
4	Error_INT_Raw	R/W	1'b0	转换发生错误原始中断状态。
3	Channel_Valid_INT_Raw	R/W	1'b0	通道转换完成原始中断状态。
2:0	Reserved	R/W	2'b0	Reserved

17.5.9. ACT0(Offset 0x2C)

比特	名称	属性	复位值	描述
31:20	Reserved	R/W	12'b0	Reserved
19:18	REF_mode	R/W	2'b0	参考源选择: 0: 内部 1.2V 1: 无效 2: 无效 3: IOLDO
17	Reserved	R/W	1'b0	Reserved
16	REF_en	R/W	1'b0	参考源使能
15:6	Reserved	R/W	10'h0	Reserved
5	Input_BUF_en	R/W	1'b0	内部信号输入使能。 0: Disable 1: Enable
4	Input_channel_en	R/W	1'b0	外部输入通道使能。 0: Disable 1: Enable
3	Reserved	R/W	1'b0	Reserved
2	Analog_Init	R/W	1'b0	Analog Init。
1	ADC_EN	R/W	1'b0	ADC analog enable。 0: analog disable 1: analog enable
0	Reserved	R/W	1'b0	Reserved

17.5.10. ACT1(Offset 0x30)

比特	名称	属性	复位值	描述
31:3	Reserved	R/W	29'b0	Reserved
2	VBAT_EN	R/W	1'b0	1/4VBAT 电池电量测量使能。
1	VBE_EN	R/W	1'b0	VBE 内部核心温度测量使能。
0	Reserved	R/W	1'b0	Reserved

17.5.11. ChannelMap[0 ~ 7] (Offset 0x34 ~ 0x50)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7:0	ChannelMap[0] (Offset 0x34)	R/W	0x0	通道[0]转换源选择。 0: PMU_IO4 1: PMU_IO4 2: PMU_IO5 3: PMU_IO6 4: VBE 5: 1/4VBAT
7:0	ChannelMap[1] (Offset 0x38)	R/W	0x0	同上
7:0	ChannelMap[2] (Offset 0x3C)	R/W	0x0	同上
7:0	ChannelMap[3] (Offset 0x40)	R/W	0x0	同上
7:0	ChannelMap[4] (Offset 0x44)	R/W	0x0	同上
7:0	ChannelMap[5] (Offset 0x48)	R/W	0x0	同上
7:0	ChannelMap[6] (Offset 0x4C)	R/W	0x0	同上
7:0	ChannelMap[7] (Offset 0x50)	R/W	0x0	同上

17.5.12. ChannelData[0 ~ 7] (Offset 0x74 ~ 0x90)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7:0	ChannelData[0] (Offset 0x74)	R/W	0x0	通道[0]数据寄存器。
7:0	ChannelData[1] (Offset 0x78)	R/W	0x0	通道[1]数据寄存器。
7:0	ChannelData[2] (Offset 0x7C)	R/W	0x0	通道[2]数据寄存器。
7:0	ChannelData[3] (Offset 0x80)	R/W	0x0	通道[3]数据寄存器。
7:0	ChannelData[4] (Offset 0x84)	R/W	0x0	通道[4]数据寄存器。
7:0	ChannelData[5] (Offset 0x88)	R/W	0x0	通道[5]数据寄存器。
7:0	ChannelData[6] (Offset 0x8C)	R/W	0x0	通道[6]数据寄存器。
7:0	ChannelData[7] (Offset 0x90)	R/W	0x0	通道[7]数据寄存器。

18. USB 全速设备接口 (USB)

18.1.1. 概述

USB 外设实现了 USB2.0 全速总线和 AHB 总线间的接口。为微控制器提供了与其他符合 USB 规范的设备进行通讯连接。

18.1.2. 主要特性

- 符合 USB2.0 全速设备的技术规范
- 可配置端点 8 对(端点 0~7)USB 端点 (in out 双向传输)
- 硬件 CRC (循环冗余校验) 生成/校验
- 硬件 NRZI 编码/解码和位填充
- 硬件 ACK/NACK 相应
- 所有端点共享 1024 byte FIFO, 每端点 FIFO 大小可配, 所有端点 FIFO 总和不超过 1024 byte

18.2. USB 系统框图

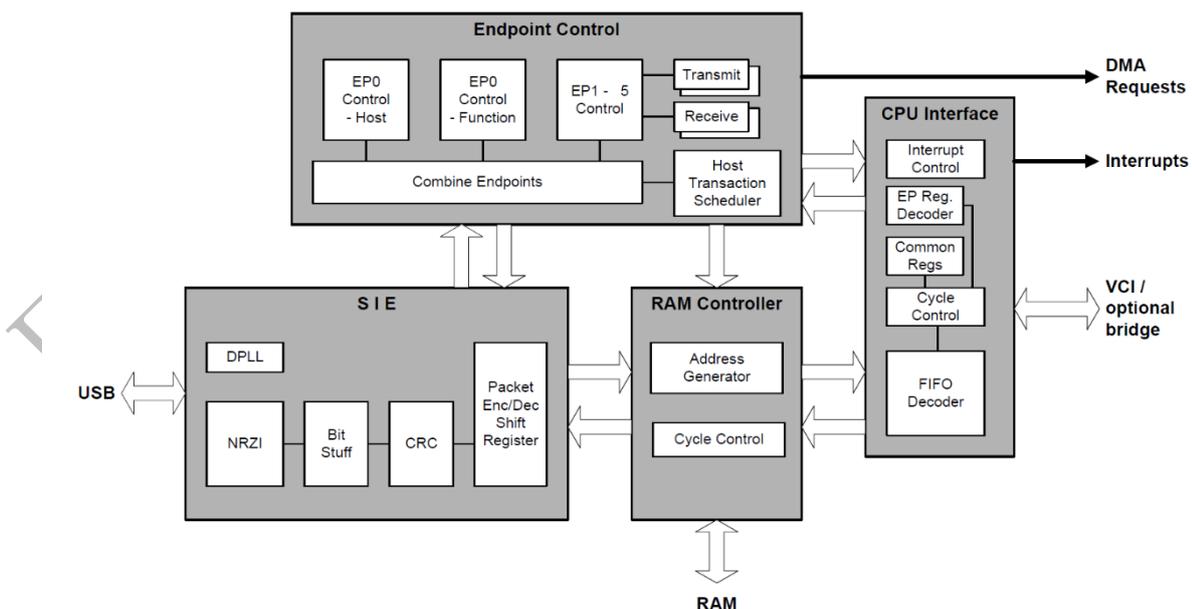


图 18-1 USB 系统框图

18.3. USB 寄存器

18.3.1. USB 寄存器地址映像

USB 寄存器可分为三部分：

1. USB 公共寄存器（Offset 0x00~0x0F）：这些寄存器为整个内核提供控制和状态。
2. 端点控制/状态寄存器（Offset 0x10~0x1F）：这些寄存器为端点提供控制和状态。

注：端点 0 与其他端点寄存器内容略有不同；当选择为外设模式（Device）或选择主模式（HOST）时，寄存器内容有所不同。

3. 端点 FIFOs（0x20~0x34）：此地址范围提供对端点 FIFO 的访问。

USB 寄存器基地址：0x1001_0000

表格 18-1 USB 寄存器地址映像

偏移	寄存器	寄存器描述
1.USB 公共寄存器		
0x00	Faddr	地址寄存器
0x01	Power	USB 电源管理寄存器
0x02	IntrTx1	发送完成中断状态寄存器。端点 0 Rx/Tx，端点 1~5 Tx
0x03	IntrTx2	Reserved
0x04	IntrRx1	接收中断状态寄存器。端点 1~5 Rx
0x05	IntrRx2	Reserved
0x06	IntrUSB	USB 中断状态寄存器
0x07	IntrTx1E	发送完成中断使能寄存器
0x08	IntrTx2E	发送完成中断使能寄存器
0x09	IntrRx1E	接收中断使能寄存器
0x0A	IntrRx2E	接收中断使能寄存器
0x0B	IntrUSB E	USB 中断使能寄存器
0x0C	Frame1	帧计数值，低 8 位
0x0D	Frame2	帧计数值，高 2 位
0x0E	Index	端点选择
0x0F	DevCtl	USB 设备控制寄存器
2.USB 端点寄存器，外设模式（Device）		
0x10	TxMaxP	Tx 端点最大数据包长度（只有端点 1~5 可以设置）

0x11	CSR0	端点 0 控制/状态寄存器 0
	TxCSR1	Tx 端点控制/状态寄存器 1（只有端点 1~5 可以设置）
0x12	CSR2	端点 0 控制/状态寄存器 1
	TxCSR2	Tx 端点控制/状态寄存器 2（只有端点 1~5 可以设置）
0x13	RxMaxP	Rx 端点最大数据包长度
0x14	RxCSR1	Rx 端点控制/状态寄存器 1（只有端点 1~5 可以设置）
0x15	RxCSR2	Rx 端点控制/状态寄存器 2（只有端点 1~5 可以设置）
0x16	Count0	端点 0 接收计数
	RxCount1	端点 1~5 接收计数，低字节（只有端点 1~5 可以设置）
0x17	RxCount2	端点 1~5 接收计数，高字节（只有端点 1~5 可以设置）
0x18~0x1B	Reserve	Reserve
0x1C	TxFIFO1	Tx 端点 FIFO 配置（只有端点 1~5 可以设置）
0x1B	TxFIFO2	Tx 端点 FIFO 配置（只有端点 1~5 可以设置）
0x1D	RxFIFO1	Rx 端点 FIFO 配置（只有端点 1~5 可以设置）
0x1F	RxFIFO2	Rx 端点 FIFO 配置（只有端点 1~5 可以设置）
2.USB 端点寄存器，主机模式（Host）。		
0x10	TxMaxP	Tx 端点最大数据包长度（只有端点 1~5 可以设置）
0x11	CSR0	端点 0 控制/状态寄存器 0
	TxCSR1	Tx 端点控制/状态寄存器 1（只有端点 1~5 可以设置）
0x12	CSR2	端点 0 控制/状态寄存器 1
	TxCSR2	Tx 端点控制/状态寄存器 2（只有端点 1~5 可以设置）
0x13	RxMaxP	Rx 端点最大数据包长度
0x14	RxCSR1	Rx 端点控制/状态寄存器 1（只有端点 1~5 可以设置）
0x15	RxCSR2	Rx 端点控制/状态寄存器 2（只有端点 1~5 可以设置）
0x16	Count0	端点 0 接收计数
	RxCount1	端点 1~5 接收计数，低字节（只有端点 1~5 可以设置）
0x17	RxCount2	端点 1~5 接收计数，高字节（只有端点 1~5 可以设置）
0x18	TxType	设置 Tx 端点的事务协议和外围端点号（只有端点 1~5 可以设置）
0x19	NAKLimit0	设置端点 0 上的 NAK 响应超时
	TxInterval	设置 Tx 中断端点的轮询间隔，单位 ms（只有端点 1~5 可以设置）
0x1A	RxType	设置 Rx 端点的事务协议和外围端点号（只有端点 1~5 可以设置）
0x1B	RxInterval	设置 Rx 中断端点的轮询间隔，单位 ms（只有端点 1~5 可以设置）
0x1C	TxFIFO1	Tx 端点 FIFO 配置（只有端点 1~5 可以设置）
0x1B	TxFIFO2	Tx 端点 FIFO 配置（只有端点 1~5 可以设置）

0x1D	RxFIFO1	Rx 端点 FIFO 配置（只有端点 1~5 可以设置）
0x1F	RxFIFO2	Rx 端点 FIFO 配置（只有端点 1~5 可以设置）
3.端点 FIFO 寄存器		
0x20	END_0_FIFO	端点 0 FIFO 地址，读写都访问此地址
0x24	END_1_FIFO	端点 1 FIFO 地址，读写都访问此地址
0x28	END_2_FIFO	端点 2 FIFO 地址，读写都访问此地址
0x2C	END_3_FIFO	端点 3 FIFO 地址，读写都访问此地址
0x30	END_4_FIFO	端点 4 FIFO 地址，读写都访问此地址
0x34	END_5_FIFO	端点 5 FIFO 地址，读写都访问此地址

18.3.2. USB 公共寄存器（Offset 0x00~0x0F）

18.3.2.1. Faddr（Offset 0x00）

比特	名称	属性	复位值	描述
7	Reserved	R	1'b0	Reserved
6:0	Func Addr	R/W	7'h0	设备模式下（DevCtl.HostMode = 0），当收到主机设置地址命令时，将收到的地址写入此寄存器。 主机模式下（DevCtl.HostMode = 1），当发送设置地址命令时，将地址写入此寄存器。

18.3.2.2. Power（Offset 0x01）

比特	名称	Device Mode	Host Mode	复位值	描述
7	ISO Update	R/W	R	1'b0	USB 控制器在发送前等待一个 SOF 令牌包，如果在一个 SOF 令牌之前接收到了 IN 令牌，将发送一个零长度的数据包。 （只在 Device 模式的同步传输时有效）
6	Vbus Val	R	R	1'b0	USB 控制器 VbusVal 输入信号
5	Vbus Sess	R	R	1'b0	USB 控制器 VbusSess 输入信号

比特	名称	Device Mode	Host Mode	复位值	描述
4	Vbus Lo	R	R	1'b0	USB 控制器 VbusLo 输入信号
3	Reset	R	R/W	1'b0	Device 模式：当总线上存在 Reset 信号时，可从此位读到总线 Reset 状态。 Host 模式：写 '1' 在总线上产生 Reset 信号，写 '0' 恢复。
2	Resume	R/W	R/W	1'b0	当设备处于 Suspend 模式时生成 Resume 信令。在 Device 模式下，CPU 应该在 10ms(最多 15ms)后清除此位，以结束恢复信令。在 Host 模式下，CPU 应该在 20ms 后清除此位。
1	Suspend Mode	R	W	1'b0	Device 模式：读该位表示进入挂起模式。 Host 模式：该位由 CPU 在进入 Suspend 模式时设置。 当 CPU 读取中断寄存器或 Set Resume bit 或离开主机模式时，该位将被清除。
0	Enable Suspend	R/W	R	1'b0	当总线上接收到 Suspend 信号时启用进入 Suspend 模式。 (只在 Device 模式有效)

18.3.2.3. IntrTx1 (Offset 0x02)

比特	名称	属性	复位值	描述
7:6	Reserved	R	0x0	Reserved
5	EP5	R	1'b0	端点 5 发送完成中断标志
4	EP4	R	1'b0	端点 4 发送完成中断标志
3	EP3	R	1'b0	端点 3 发送完成中断标志
2	EP2	R	1'b0	端点 2 发送完成中断标志
1	EP1	R	1'b0	端点 1 发送完成中断标志
0	EP0	R	1'b0	端点 0 发送完成 或 接收数据中断标志

注：读寄存器将清除所有端点标志

18.3.2.4. IntrTx2 (Offset 0x03)

比特	名称	属性	复位值	描述
7:0	Reserved	R	8'h00	Reserved

18.3.2.5. IntrRx1 (Offset 0x04)

比特	名称	属性	复位值	描述
7:6	Reserved	R	0x0	Reserved
5	EP5	R	1'b0	端点 5 接收数据中断标志
4	EP4	R	1'b0	端点 4 接收数据中断标志
3	EP3	R	1'b0	端点 3 接收数据中断标志
2	EP2	R	1'b0	端点 2 接收数据中断标志
1	EP1	R	1'b0	端点 1 接收数据中断标志
0	Reserved	R	1'b0	Reserved

注：读寄存器将清除所有端点标志

18.3.2.6. IntrRx2 (Offset 0x05)

比特	名称	属性	复位值	描述
7:0	Reserved	R	0x0	Reserved

18.3.2.7. IntrUSB (Offset 0x06)

比特	名称	属性	复位值	描述
7	VBus Error	R	1'b0	VBus 电压低于阈值
6	Sess Req	R	1'b0	在检测到会话请求信令时产生
5	Discon	R	1'b0	Device 模式：在通讯结束时产生 Host 模式：检测到设备断开时产生
4	Conn	R	1'b0	检测到设备连接 (只在 Host 有效)
3	SOF	R	1'b0	有新的 SOF 令牌包
2	Reset	R	1'b0	Device 模式：检测到总线上 Reset 信号
1	Resume	R	1'b0	USB 控制器处于 Suspend 模式时检测到 Resume 信令
0	Suspend	R	1'b0	Device 模式：检测到总线上 Suspend 信号

注：读寄存器将清除所有端点标志

18.3.2.8. IntrTx1E (Offset 0x07)

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	0x0	Reserved
5	EP5	R/W	1'b1	端点 5 发送完成中断使能
4	EP4	R/W	1'b1	端点 4 发送完成中断使能
3	EP3	R/W	1'b1	端点 3 发送完成中断使能
2	EP2	R/W	1'b1	端点 2 发送完成中断使能
1	EP1	R/W	1'b1	端点 1 发送完成中断使能
0	EP0	R/W	1'b1	端点 0 发送完成，接收中断使能

18.3.2.9. IntrTx2E (Offset 0x08)

比特	名称	属性	复位值	描述
7:0	Reserved	R/W	0x0	Reserved

18.3.2.10. IntrRx1E (Offset 0x09)

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	0x0	Reserved
5	EP5	R/W	1'b1	端点 5 接收中断使能
4	EP4	R/W	1'b1	端点 4 接收中断使能
3	EP3	R/W	1'b1	端点 3 接收中断使能
2	EP2	R/W	1'b1	端点 2 接收中断使能
1	EP1	R/W	1'b1	端点 1 接收中断使能
0	Reserved	R/W	0x0	Reserved

18.3.2.11. IntrRx2E (Offset 0x0A)

比特	名称	属性	复位值	描述
7:0	Reserved	R/W	0x0	Reserved

18.3.2.12. IntrUSBE (Offset 0x0B)

比特	名称	属性	复位值	描述
7	VBus Error	R/W	1'b0	VBus Error 中断使能
6	Sess Req	R/W	1'b0	Sess Req 中断使能
5	Discon	R/W	1'b0	Discon 中断使能
4	Conn	R/W	1'b0	Conn 中断使能
3	SOF	R/W	1'b0	SOF 中断使能
2	Reset	R/W	1'b1	Reset 中断使能
1	Resume	R/W	1'b1	Resume 中断使能
0	Suspend	R/W	1'b0	Suspend 中断使能

18.3.2.13. Frame1 (Offset 0x0C)

比特	名称	属性	复位值	描述
7:0	Lower 8 bits of Frame Number	R	8'h00	Frame1 是一个 8 位的只读寄存器，在外设模式中保存最后接收帧号的较低 8 位，在主机模式中保存当前帧号的较低 8 位。

18.3.2.14. Frame2 (Offset 0x0D)

比特	名称	属性	复位值	描述
7:3	Reserved	R	5'h0	Reserved
2:0	Upper 3 bits of Frame Number	R	3'b000	Frame2 是一个 3 位的只读寄存器，在外设模式中保存最后接收到的帧号的高 3 位，在主机模式中保存当前帧号的高 3 位。

18.3.2.15. Index (Offset 0x0E)

比特	名称	属性	复位值	描述
7:4	Reserved	R	4'h0	Reserved
3:0	Selected Endpoint	R/W	4'h0	端点选择。详见章节 端点选择

18.3.2.16. DevCtl (Offset 0x0F)

比特	名称	属性	复位值	描述
7	CID	R	1'b1	CID 号 0: Host 模式 1: Device 模式
6	FSDev	R	1'b0	检测到全速设备连接。
5	LSDev	R	1'b0	检测到低速设备连接。
4	PUCON	R	1'b0	当 USB D+线上需要一个上拉电阻作为外设操作
3	PDCON	R	1'b1	当 USB D+线上需要一个下拉电阻作为主机操作
2	HostMode	R	1'b0	当作为主机时
1	HostReq	R/W	1'b0	写 '1' 后, 当已进入挂起模式 (总线上 3ms 没有活动), USB 控制器将启动主机初始化。当主机初始化完成后硬件自动清除。
0	Session	R/W	1'b0	当作为 HOST 时, 该位由软件设置或清除, 以开始或结束一个会话。 当作为 Device 时, 该位在会话开始/结束时被 USB 控制器设置/清除, 软件无需干预。软件也可以设置它来发起会话请求协议, 并清除它以执行与主机的软断开连接(然后结束会话)。

18.3.3. USB 端点 0 控制状态寄存器 (Offset 0x10~0x1F)

18.3.3.1. CSR0 (Offset 0x11)

IN Device Mode:

比特	名称	属性	复位值	描述
7	ServicedSetupEnd	W	1'b0	写 '1' 清除 SetupEnd 位 (硬件自动清除)
6	ServicedRxPktRdy	W	1'b0	写 '1' 清除 RxPktRdy 位 (硬件自动清除)
5	SendStall	W	1'b0	写 '1' 以终止当前事务。发送一次 STALL。 (硬件自动清除)
4	SetupEnd	R	1'b0	当控制传输在设置 DataEnd 位之前结束时， 将设置此位。此时将产生一个中断并刷新 FIFO。
3	DataEnd	W	1'b0	以下三种情况时设置此位。 1. 为最后一个数据包设置 TxPktRdy 时。 2. 卸载最后一个数据包后清除 RxPktRdy 时。 3. 为零长度的数据包设置 TxPktRdy 时。 (硬件自动清除)
2	SentStall	R	1'b0	当 STALL 发送后置位 (写 '0' 清除)
1	TxPktRdy	R/W	1'b0	将数据包装入 FIFO 后设置此位。当数据包发 送成功后，自动清除。当该位被清除时将产生 一个发送完成中断。(硬件自动清除)
0	RxPktRdy	R	1'b0	收到数据包时置 '1'。该位被设置时产生一 个中断。CPU 通过设置 ServicedRxPktRdy 位 来清除此位。

IN Host Mode:

比特	名称	属性	复位值	描述
7	NAK Timeout	R	1'b0	一次传输中，端点连续接收 NAK 包时间超过 配置的超时时间 (NAKLimit0 寄存器配置超 时) 置位。端点 0 停止工作，写 '0' 清除此 位，端点 0 继续工作。
6	StatusPkt	R/W	1'b0	设置 TxPktRdy 或 ReqPkt 位的同时设置这个 位，以执行状态阶段事务。设置这个位确保数 据切换设置为 1，以便 DATA1 包用于状态阶 段事务。
5	ReqPkt	R/W	1'b0	设置这个位来请求一个 IN 事务。RxPktRdy 置位时自动清除。
4	Error	R	1'b0	当三次尝试执行一个事务而没有从外设得到响 应时，将设置此位，并产生端点 0 中断。

3	SetupPkt	R/W	1'b0	CPU 在设置 TxPktRdy 位的同时设置这个位，为事务发送 SETUP 令牌而不是 OUT 令牌
2	RxStall	R	1'b0	收到 STALL 时置位。 (写 '0' 清除)
1	TxPktRdy	R/W	1'b0	写操作：将数据包装入 FIFO 后设置此位。当数据包发送成功后，自动清除。当该位被清除时将产生一个发送完成中断。(硬件自动清除) 读操作：数据包没有发送完成读为 '1'，发送完成读为 '0'。
0	RxPktRdy	R	1'b0	收到数据包时置 '1'。该位被设置时产生一个中断。 (写 '0' 清除)

18.3.3.2. CSR2 (Offset 0x12)

比特	名称	属性	复位值	描述
7:6	Reserved	R	0x0	Reserved
0	FlushFIFO	W	1'b0	写 '1' 刷新端点 0 FIFO, FIFO 指针复位, 清除 TxPktRdy/RxPktRdy 位。 注： 除非 TxPktRdy/RxPktRdy 有效, 否则 FlushFIFO 无效。

18.3.3.3. Count0 (Offset 0x16)

比特	名称	属性	复位值	描述
7	Reserved	R	0x0	Reserved
6:0	Endpoint 0 Count	R	7'h0	它表示端点 0 FIFO 中接收的数据字节数。当设置 RxPktRdy (CSR0.D0)时, 返回的值是有效的。

18.3.3.4. NAKLimit0 (HostMode Only. Offset 0x19)

比特	名称	属性	复位值	描述
7:0	Endpoint 0 NAK Limit	R/W	8'h00	端点 0 连续接收 NAK 包帧数 (2~255)。连续接收 NAK 包超过设置值, 将会置位 NAK Timeout 标志。

18.3.4. USB 端点 1~5 控制状态寄存器 (Offset 0x10~0x1F)

18.3.4.1. TxMaxP (Offset 0x10)

比特	名称	属性	复位值	描述
7:0	Maximum Packet Size	R/W	8'h00	端点发送最大数据包。以 8 字节为单位, 设置 128 则最大数据包为 1023 个字节。最大数据包不应超过 USB 规范要求。 端点 FIFO 配置章节

18.3.4.2. TxCSR1 (Offset 0x11)

IN Device Mode:

比特	名称	属性	复位值	描述
7	Reserved	R	1'b0	Reserved
6	ClrDataTog	W	1'b0	复位端点数据包, 从 Data0 开始
5	SentStall	R	1'b0	当 STALL 发送后置位。 (写 '0' 清除)
4	SendStall	R/W	1'b0	写 '1' 以终止当前事务。发送一次 STALL。 (同步传输时无效) 写 '0' 不再发送 STALL。
3	FlushFIFO	W	1'b0	写 '1' 刷新端点 0 FIFO, FIFO 指针复位, 清除 TxPktRdy 位。

				注：除非 TxPktRdy 有效，否则 FlushFIFO 无效。
2	UnderRun	R	1'b0	当 TxPktRdy 位没有置位时，如果接收到 IN 令牌。 (写 '0' 清除)
1	FIFONotEmpty	R	1'b0	当 Tx FIFO 存在数据包时置位。 (写 '0' 清除)
0	TxPktRdy	R/W	1'b0	写操作：将数据包装入 FIFO 后设置此位。当数据包发送成功后，自动清除。当该位被清除时将产生一个发送完成中断。(硬件自动清除) 读操作：数据包没有发送完成读为 '1'，发送完成读为 '0'。

IN Host Mode:

比特	名称	属性	复位值	描述
7	NAK Timeout	R	1'b0	一次传输中，端点连续接收 NAK 包时间超过配置的超时时间 (TxInterval 寄存器配置超时) 置位。 (写 '0' 清除，只在批量传输中有效)
6	ClrDataTog	W	1'b0	复位端点数据包，从 Data0 开始
5	RxStall	R	1'b0	收到 STALL 时设置此位。TxPktRdy 置位并刷新 FIFO 时会被清除。 (写 '0' 清除)
4	Reserved	R	1'b0	Reserved
3	FlushFIFO	W	1'b0	写 '1' 刷新端点 0 FIFO，FIFO 指针复位，清除 TxPktRdy 位。 注：除非 TxPktRdy 有效，否则 FlushFIFO 无效。
2	Error	R	1'b0	当三次尝试执行一个事务而没有从外设得到响应时，将设置此位，并产生端点中断。 (写 '0' 清除，只在批量传输，中断传输中有效)
1	FIFO NotEmpty	R	1'b0	当 Tx FIFO 存在数据包时置位 (写 '0' 清除)
0	TxPktRdy	R/W	1'b0	写操作：将数据包装入 FIFO 后设置此位。当数据包发送成功后，自动清除。当该位被清除

				<p>时将产生一个发送完成中断。（硬件自动清除）</p> <p>读操作：数据包没有发送完成读为‘1’，发送完成读为‘0’。</p>
--	--	--	--	---

18.3.4.3. TxCSR2 (Offset 0x12)

比特	名称	属性	复位值	描述
7	AutoSet	R/W	1'b0	写‘1’，TxPktRdy 将自动设置数据的最大包大小(TxMaxP 中的值)载入 Tx FIFO。如果数据包的大小小于最大数据包的大小加载，TxPktRdy 将必须手动设置。
6	ISO	R/W	1'b0	写‘1’，IN 端点启用同步传输， 写‘0’，IN 端点启用批量传输或中断传输
5	Mode	R/W	1'b1	写‘1’，配置为 IN 端点。 写‘0’，配置为 OUT 端点。 注： 此位只有在 Tx Rx 共用一块 FIFO 地址时有用。当 FIFO 足够时，建议 Rx Tx 使用不同地址的 FIFO。
4	DMAEnable	R/W	1'b0	IN 端点使能 DMA
3	FrcDataTog	R/W	1'b0	强制数据包为 Data0 或 Data1，并将数据包从 FIFO 清除
2	DMAMode	R	1'b0	0：模式 0，为所有数据包生成一个 DMA 请求和一个中断。 1：模式 1，为 TxMaxP 字节大小的数据包生成一个 DMA 请求(但没有中断)。
1:0	Reserved	R	1'b0	Reserved

18.3.4.4. RxMaxP (Offset 0x13)

比特	名称	属性	复位值	描述
7:0	Maximum Packet Size	R/W	8'h00	<p>端点接收最大数据包。以 8 字节为单位，设置 128 则最大数据包为 1023 个字节。</p> <p>最大数据包不应超过 USB 规范要求。</p> <p>端点 FIFO 配置章节</p>

18.3.4.5. RxCSR1 (Offset 0x14)

IN Device Mode:

比特	名称	属性	复位值	描述
7	ClrDataTog	W	1'b0	复位端点数据包，从 Data0 开始
6	SentStall	R	1'b0	<p>当 STALL 发送后置位。</p> <p>(写 '0' 清除)</p>
5	SendStall	R/W	1'b0	<p>写 '1' 以终止当前事务。发送一次 STALL。</p> <p>(同步传输时无效)</p> <p>写 '0' 不再发送 STALL。</p>
4	FlushFIFO	W	1'b0	<p>写 '1' 刷新端点 0 FIFO, FIFO 指针复位, 清除 RxPktRdy 位。</p> <p>注: 除非 RxPktRdy 有效, 否则 FlushFIFO 无效。</p>
3	DataError	R	1'b0	<p>当 RxPktRdy 置位并且数据包发生 CRC 错误或 bit 错误时置位。</p> <p>清除 RxPktRdy 时, 会将此位清除。</p> <p>注: 只在同步传输时有效</p>
2	OverRun	R	1'b0	<p>当 Rx FIFO 无法装下接收到的数据包时置位。</p> <p>(写 '0' 清除)</p> <p>注: 只在同步传输时有效</p>
1	FIFOFull	R	1'b0	接收 FIFO 满
0	RxPktRdy	R	1'b0	<p>收到数据包时置 '1'。该位被设置时产生一个中断。</p> <p>(写 '0' 清除)</p>

IN Host Mode:

比特	名称	属性	复位值	描述
7	ClrDataTog	W	1'b0	复位端点数据包，从 Data0 开始
6	RxStall	R	1'b0	收到 STALL 时置位，并产生端点中断。 (写 '0' 清除)
5	ReqPkt	R/W	1'b0	设置这个位来请求一个 IN 事务。RxPktRdy 置位时自动清除。
4	FlushFIFO	W	1'b0	写 '1' 刷新端点 0 FIFO，FIFO 指针复位，清除 RxPktRdy 位。 注： 除非 RxPktRdy 有效，否则 FlushFIFO 无效。
3	DataError/NAK Timeout	R	1'b0	同步模式下，当 RxPktRdy 置位，接收到的数据包存在 CRC 错误时置位。 批量传输下，端点 0 接收 NAK 回应时间超过配置时间（NAKLimit0 寄存器配置超时时间）时置位。 (写 '0' 清除)
2	Error	R	1'b0	当三次尝试执行一个事务而没有从外设得到响应时，将设置此位，并产生端点中断。 (写 '0' 清除)
1	FIFOFull	R	1'b0	接收 FIFO 满
0	RxPktRdy	R	1'b0	收到数据包时置 '1'。该位被设置时产生一个中断。 (写 '0' 清除)

18.3.4.6. RxCSR2 (Offset 0x15)

比特	名称	属性	复位值	描述
7	AutoClear	R/W	1'b0	写 '1'，RxPktRdy 位将在 RxMaxP 字节的报文从 Rx FIFO 中卸载后自动清除。当小于最大报文大小的报文被卸载时，必须手动清除 RxPktRdy。
6	Device 模式: IOS Host 模式: AutoReq	R/W	1'b0	Device 模式下，CPU 设置这个位使 OUT 端点能够进行同步传输，清除这个位使 OUT 端点能够进行 Bulk 或 Interrupt 传输。

				Host 模式下，设置了 ReqPkt 位，清除 RxPktRdy 位后，ReqPkt 位将自动设置
5	DMAEnable	R/W	1'b0	OUT 端点使能 DMA
4	DMAMode	R/W	1'b0	0: 模式 0，为所有接收到的数据包生成一个 DMA 请求，以及一个中断(如果启用)。 1: 模式 1，对大小为 RxMaxP 字节的数据包生成一个 DMA 请求(但没有中断)，对于其他大小的数据包生成一个中断(但没有 DMA 请求)。
3:0	Reserved	R	0x0	Reserved

18.3.4.7. RxCount1 (Offset 0x16)

比特	名称	属性	复位值	描述
7:0	Endpoint Rx Count lower 8 bits	R	8'h00	它表示 Rx 端点 FIFO 中接收的数据字节数的低 8 位。当设置 RxPktRdy (RxCSR1.D0)时，返回的值是有效的。

18.3.4.8. RxCount2 (Offset 0x17)

比特	名称	属性	复位值	描述
7:3	Reserved	R	0x0	Reserved
2:0	Endpoint Rx Count upper 3 bits	R	3'b000	它表示 Rx 端点 FIFO 中接收的数据字节数的高 3 位。当设置 RxPktRdy (RxCSR1.D0)时，返回的值是有效的。

18.3.4.9. TxType (HostMode Only. Offset 0x18)

比特	名称	属性	复位值	描述
7:6	Reserved	R	0x0	Reserved
5:4	Protocol	R/W	2'b00	端点传输类型：

				00: 无 01: 同步传输（等时传输） 10: 批量传输 11: 中断传输
3:0	Target Endpoint Number	R/W	4'h0	设备接收数据的端点号。

18.3.4.10. TxInterval (HostMode Only. Offset 0x19)

比特	名称	属性	复位值	描述
7:0	Tx Polling Interval/NAK Limit (n)	R/W	8'h00	中断或同步传输：轮询间隔（1~255ms） 批量传输：端点连续接收 NAK 包帧数，（2~255）。连续接收 NAK 包超过设置值，将会置位 NAK Timeout 标志。

18.3.4.11. RxType (HostMode Only. Offset 0x1A)

比特	名称	属性	复位值	描述
7:6	Reserved	R	0x0	Reserved
5:4	Protocol	R/W	3'b00	端点传输类型： 00: 无 01: 同步传输（等时传输） 10: 批量传输 11: 中断传输
3:0	Target Endpoint Number	R/W	4'h0	设备发送数据的端点号。

18.3.4.12. RxInterval (HostMode Only. Offset 0x1B)

比特	名称	属性	复位值	描述
7:0	Rx Polling Interval/NAK Limit (n)	R/W	8'h00	中断或同步传输：轮询间隔（1~255 ms） 批量传输：端点连续接收 NAK 包帧数，（2~255）。连续接收 NAK 包超过设置值，将会置位 NAK Timeout 标志。

18.3.4.13. TxFIFO1、TxFIFO2 (Offset 0x1C、0x1D)

比特	名称	描述																		
TxFIFO1		地址以 8byte 为单位。FIFO 起始地址 0x000，结束地址 0x3FF。																		
7:0	AD[7:0]																			
TxFIFO2																				
3:0	AD[11:8]																			
		<table border="1"> <thead> <tr> <th>AD[11 : 0]</th> <th>Start Address</th> </tr> </thead> <tbody> <tr> <td>0x000</td> <td>0x000</td> </tr> <tr> <td>0x001</td> <td>0x008</td> </tr> <tr> <td>0x002</td> <td>0x010</td> </tr> <tr> <td>0x080</td> <td>0x400</td> </tr> </tbody> </table>	AD[11 : 0]	Start Address	0x000	0x000	0x001	0x008	0x002	0x010	0x080	0x400								
AD[11 : 0]	Start Address																			
0x000	0x000																			
0x001	0x008																			
0x002	0x010																			
0x080	0x400																			
4	DPB	写 '1' 支持双包模式。（不推荐） 写 '0' 支持单包模式。默认使用单包模式。																		
7:5	SZ[2:0]	端点 FIFO 大小。 <table border="1"> <thead> <tr> <th>SZ[2 : 0]</th> <th>端点 FIFO 大小 (byte)</th> </tr> </thead> <tbody> <tr> <td>3'b000</td> <td>8</td> </tr> <tr> <td>3'b001</td> <td>16</td> </tr> <tr> <td>3'b010</td> <td>32</td> </tr> <tr> <td>3'b011</td> <td>64</td> </tr> <tr> <td>3'b100</td> <td>128</td> </tr> <tr> <td>3'b101</td> <td>256</td> </tr> <tr> <td>3'b110</td> <td>512</td> </tr> <tr> <td>3'b111</td> <td>1024</td> </tr> </tbody> </table> 如果 DPB=0，FIFO 大小为设置大小。DPB=1，FIFO 大小是设置的两倍。	SZ[2 : 0]	端点 FIFO 大小 (byte)	3'b000	8	3'b001	16	3'b010	32	3'b011	64	3'b100	128	3'b101	256	3'b110	512	3'b111	1024
SZ[2 : 0]	端点 FIFO 大小 (byte)																			
3'b000	8																			
3'b001	16																			
3'b010	32																			
3'b011	64																			
3'b100	128																			
3'b101	256																			
3'b110	512																			
3'b111	1024																			

18.3.4.14. RxFIFO1、RxFIFO2 (Offset 0x1E、0x1F)

比特	名称	描述																		
RxFIFO1		地址以 8byte 为单位。FIFO 起始地址 0x000，结束地址 0x3FF。																		
7:0	AD[7:0]																			
RxFIFO2																				
3:0	AD[11:8]																			
		<table border="1"> <thead> <tr> <th>AD[11 : 0]</th> <th>Start Address</th> </tr> </thead> <tbody> <tr> <td>0x000</td> <td>0x000</td> </tr> <tr> <td>0x001</td> <td>0x008</td> </tr> <tr> <td>0x002</td> <td>0x010</td> </tr> <tr> <td>0x080</td> <td>0x400</td> </tr> </tbody> </table>	AD[11 : 0]	Start Address	0x000	0x000	0x001	0x008	0x002	0x010	0x080	0x400								
AD[11 : 0]	Start Address																			
0x000	0x000																			
0x001	0x008																			
0x002	0x010																			
0x080	0x400																			
4	DPB	写 ‘1’ 支持双包模式。（不推荐） 写 ‘0’ 支持单包模式。默认使用单包模式。																		
7:5	SZ[2:0]	<p>端点 FIFO 大小。</p> <table border="1"> <thead> <tr> <th>SZ[2 : 0]</th> <th>端点 FIFO 大小 (byte)</th> </tr> </thead> <tbody> <tr> <td>3'b000</td> <td>8</td> </tr> <tr> <td>3'b001</td> <td>16</td> </tr> <tr> <td>3'b010</td> <td>32</td> </tr> <tr> <td>3'b011</td> <td>64</td> </tr> <tr> <td>3'b100</td> <td>128</td> </tr> <tr> <td>3'b101</td> <td>256</td> </tr> <tr> <td>3'b110</td> <td>512</td> </tr> <tr> <td>3'b111</td> <td>1024</td> </tr> </tbody> </table> <p>如果 DPB=0，FIFO 大小为设置大小。DPB=1，FIFO 大小是设置的两倍。</p>	SZ[2 : 0]	端点 FIFO 大小 (byte)	3'b000	8	3'b001	16	3'b010	32	3'b011	64	3'b100	128	3'b101	256	3'b110	512	3'b111	1024
SZ[2 : 0]	端点 FIFO 大小 (byte)																			
3'b000	8																			
3'b001	16																			
3'b010	32																			
3'b011	64																			
3'b100	128																			
3'b101	256																			
3'b110	512																			
3'b111	1024																			

18.3.5. USB 端点 FIFO 寄存器 (Offset 0x20~0x34)

地址	名称	属性	复位值	描述
0x20	END_0_FIFO	R/W	0x0	端点 0 FIFO 地址，读写都访问此地址
0x24	END_1_FIFO	R/W	0x0	端点 1 FIFO 地址，读写都访问此地址
0x28	END_2_FIFO	R/W	0x0	端点 2 FIFO 地址，读写都访问此地址
0x2C	END_3_FIFO	R/W	0x0	端点 3 FIFO 地址，读写都访问此地址
0x30	END_4_FIFO	R/W	0x0	端点 4 FIFO 地址，读写都访问此地址
0x34	END_5_FIFO	R/W	0x0	端点 5 FIFO 地址，读写都访问此地址

18.4. 使用流程

18.4.1. 端点选择

每个端点都有自己的一组控制/状态寄存器。在任何一个时间，只有一组 Tx 控制/状态寄存器和一组 Rx 控制/状态寄存器出现在内存映射中。在访问端点的控制/状态寄存器之前，应该将端点编号写入 Index 寄存器，以确保在内存映射中出现正确的控制/状态寄存器。

18.4.2. 端点 FIFO 配置

USB 提供了 1024 byte 的 FIFO。其中端点 0 固定使用 64byte。其他端点需要通过 TxFIFO1、TxFIFO2 或 RxFIFO1、RxFIFO2 配置。

端点 FIFO 的配置大小，建议符合 USB 规范：

- **控制传输**：高速模式的最大包长固定为 64 个字节；全速模式可在 8、16、32、64 字节中选择；低速模式的最大包长固定为 8 个字节。
- **批量传输**：高速模式的最大包长固定为 512 个字节；全速模式最大包长可在 8、16、32、64 字节中选择；低速模式不支持批量传输。
- **同步传输**：高速模式的最大包长上限为 1024 个字节；全速模式最大包长上限为 1023 个字节；低速模式不支持同步传输。
- **中断传输**：高速模式的最大包长上限为 1024 个字节；全速模式最大包长上限为 64 个字节；低速模式最大包长上限为 8 个字节。

注意：为任何端点设置的最大数据包大小（TxMaxP）不能超过 FIFO 大小。当 FIFO 中有数据时，不应该写入 TxMaxP 寄存器，因为可能会发生意外的结果

18.4.3. Device 模式发送

当主机请求 IN 传输时，需要设备发送数据。

将要发送的数据包都加载到 Tx FIFO 中，设置 TXCSR1 中的 TxPktRdy 位。如果设置了 TxCSR2 中的 AutoSet 位，TxPktRdy 位将在加载一个最大大小的数据包到 FIFO 时自动设置。对于小于最大数据包大小的数据包，TxPktRdy 将总是必须手动设置。

当 TxPktRdy 位被手动或自动设置时，TXCSR1 中的 FIFONotEmpty 位也被设置，数据包准备发送。

当数据包成功发送后，TxPktRdy 和 FIFONotEmpty 都将被清除，并生成适当的 Tx 端点中断(如果启用)。然后，下一个包可以装入 FIFO。

18.4.4. Device 模式接收

当主机请求 OUT 传输时，需要设备接收数据。

当一个数据包收到并放置在 Rx FIFO，RXCS1 中 RxPktRdy 位(D0)和 FIFOFull 位(D1)置位，并生成适当的 Rx 端点中断(如果启用)，数据包现在可以从 FIFO 卸载，卸载的数据量可读取 RxCount1、RxCount2 确定。

当数据包被卸载后，RxPktRdy 位需要被清除，以允许进一步的报文被接收。如果设置了 RxCSR2 (D7)中的 AutoClear 位，并且从 FIFO 中卸载一个最大报文，则 RxPktRdy 位自动清除。FIFOFull 位也被清除。对于小于最大数据包大小的数据包，RxPktRdy 总是必须被清除手动。

18.4.5. Device 模式注意事项

- 控制传输中的 0 长度 data

一个零长度的 OUT 数据包用于表示控制传输的结束。在正常操作中，只有在设备请求的整个长度被转移之后(即 CPU 设置了 DataEnd 之后)才会收到这样的数据包。但是，如果主机在整个设备请求传输之前发送了一个零长度的 OUT 数据包，这意味着传输的提前结束。在这种情况下，USB 控制器将自动刷新状态机，并设置 SetupEnd。

- 控制传输中的 STALL

在以下条件下，USB 控制器将自动向 Control 传输发出一个 STALL。

1. 主机在控制传输的 OUT 数据阶段发送的数据比在 SETUP 阶段的设备请求中指定的要多。
2. 主机在控制传输的 IN 数据阶段请求的数据比在 SETUP 阶段的设备请求中指定的多。
3. 主机使用 OUT 数据令牌发送超过 MaxP 的数据。
4. 主机发送错误的 PID 用于控制传输的 OUT 状态阶段。

5. 主机在 OUT 状态阶段发送不止一个 0 长度的数据包。

18.4.6. Device 模式暂停

如果当 USB 控制器运行在 Device 模式，USB 上没有活动 3ms，并且在 Power 寄存器中设置了 Enable Suspend 位，USB 控制器将进入 Suspend 模式。如果暂停中断已启用，则此时将生成一个中断。

当检测到 Resume 信号时，USB 控制器将退出 Suspend 模式并重新启用系统时钟。如果 Resume 中断被启用，将会产生一个中断

CPU 也可以通过在 Power 寄存器中设置 Resume 位来强制 USB 控制器离开 Suspend 模式。当设置此位时，USB 控制器将退出 Suspend 模式并将 Resume 信号驱动到总线上。CPU 应该在 10ms(最多 15ms)后清除此位以结束恢复信令。注意:在这种情况下不会产生 Resume 中断。

18.4.7. Device 模式 SOF 包

当 USB 控制器运行在 Device 模式时，它应该每毫秒从主机收到一个 Start-Of-Frame 包。当接收到 SOF 包时，将包中包含的 11 位帧号写入到 Frame1 和 Frame2 两个寄存器中，并在 SOF_PULSE 上产生一个持续一个 USB 位周期的输出脉冲。一个 SOF 中断也会产生(如果启用)。

一旦 USB 控制器开始接收 SOF 包，它预计每毫秒接收一个。如果在 1.00358 ms 后没有收到 SOF 报文，则认为该报文丢失，虽然帧寄存器没有更新，但是产生了 SOF_PULSE 和 SOF 中断。USB 控制器将继续每毫秒产生一个 SOF_PULSE 和 SOF 中断，直到这些脉冲能够重新同步到接收到的 SOF 包，当这些包再次被成功接收。

18.4.8. Host 模式接收

当 USB 控制器作为主机运行时，IN 事务的处理方式类似于当 USB 控制器作为 Device 运行时，OUT 事务的处理方式，除了事务需要首先通过在 RXCS1 中设置 ReqPkt 位来启

动。这向事务调度程序表明在这个端点上有一个活动事务。然后，事务调度程序向目标发送一个 IN 令牌。

当一个数据包收到并放置在 Rx FIFO，RXCS1 中 RxPktRdy 位(D0)和 FIFOFull 位(D1)置位，并生成适当的 Rx 端点中断(如果启用)，数据包现在可以从 FIFO 卸载，卸载的数据量可读取 RxCount1、RxCount2 确定。

当报文被卸载时，RxPktRdy 应该被清除。RxCSR2 寄存器中的 AutoClear 位可以用来在从 FIFO 中卸载最大大小的数据包时自动清除 RxPktRdy。RxCSR2 中也有一个 AutoReq 位，当 RxPktRdy 位被清除时，会自动设置 ReqPkt 位。AutoClear 和 AutoReq 位可以与外部 DMA 控制器一起使用，在没有 CPU 干预的情况下执行完整的 Bulk 传输。

如果目标用 NAK 响应 IN 令牌，USB 控制器将继续重试该事务，直到达到已设置的 NAK 数量限制为止。如果目标回应了 STALL，USB 控制器将不会重试事务，将 RXCS1 寄存器 RxStall 位置位并产生端点中断。如果目标在要求的时间内没有响应 IN 令牌，USB 控制器将重试该事务。如果在三次尝试后目标函数仍然没有响应，USB 控制器将清除 ReqPkt 位，将 RXCS1 寄存器 Error 位置位并产生端点中断。

18.4.9. Host 模式发送

将数据包加载至 Tx FIFO 后设置 TxCSR1 寄存器的 TxPktRdy 位。AutoSet 在 TxCSR2 可以导致 TxPktRdy 位时自动设置一个最大大小的包已经加载到 FIFO。同样，AutoSet 位可以与外部 DMA 控制器一起使用，在没有 CPU 干预的情况下执行完整的 Bulk 传输。

如果目标用 NAK 响应 OUT 令牌，则 USB 控制器将继续重试该事务，直到达到已设置的 NAK 数量限制为止。如果目标回应了 STALL，但是，USB 控制器将不会重试事务，将 TXCS1 寄存器 RxStall 位置位并产生端点中断。如果目标在要求的时间内没有响应 OUT 令牌，USB 控制器将重试该事务。如果在三次尝试后目标函数仍然没有响应，USB 控制器将刷新 FIFO，并设置 TXCS1 中的 Error 位并产生端点中断。

18.4.10. Host 模式传输调度

当作为 Host 运行时，USB 控制器维护一个 1 毫秒的帧计数器。如果目标函数是全速设备，USB 控制器将在每帧的开始自动发送一个 SOF 包。如果目标函数是低速设备，总线上将传输一个“K”状态作为“保持活动”以停止低速设备进入 Suspend 模式。

在传输了 SOF 包之后，USB 控制器将遍历所有配置的端点，寻找活动的事务。活动事务定义为 ReqPkt 位设置的 Rx 端点或 TxPktRdy 位设置的 Tx 端点。

一个活动的同步或中断事务只会在一个帧的第一个事务调度程序周期上被发现，并且如果该端点的间隔计数器已经计数到零。这确保每 n 帧每个端点只发生一个中断/同步事务(其中 n 是该端点在 TxInterval/RxInterval 寄存器中设置的间隔)。注意:USB 2.0 规范允许同步事务安排的间隔高达 65 秒。上面描述的机制可以满足高达 255ms 的间隔。需要在软件中实现大于 255ms 的间隔。

一个活动的批量传输事务将立即启动，只要在有足够的时间在下一个 SOF 包到来之前完成事务。如果事务需要重试(例如，因为收到了 NAK 或目标没有响应)，那么事务将先不会重试，直到事务调度程序首先检查了其他有活动事务的端点。这确保了发送大量 NAK 的端点不会阻塞总线上的其他事务。

18.4.11. Host 模式 Reset

如果 USB 控制器在 Host 模式下设置了 Power 寄存器中的 Reset 位，则在总线上产生 Reset 信号。当 CPU 清除该位后，USB 控制器将启动它的帧计数器和事务调度程序。

18.4.12. Host 模式 Suspend

如果设置了 Power 寄存器中的 SuspendMode 位，USB 控制器将完成当前事务，然后停止事务调度程序和帧计数器。不会再启动任何事务，也不会生成任何 SOF 包。

要退出暂停模式，应该清除 Power 寄存器中的 SuspendMode 位。直到 Resume 位为高时，USB 控制器会在总线上生成 Resume 信号。20 毫秒后，应清除 Resume 位，此时帧计数器和事务调度程序将启动。

19. 循环冗余校验（CRC）

19.1. 概述

循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然差错。计算单元使用硬件多项式发生器，将原始数据计算产生 CRC 码。

19.2. 主要特性

- 支持四种多项式。
- CRC8，多项式：0x07， X^8+X^2+X+1 。
- CRC16_CCITT_FALSE，多项式：0x1021， $X^{16}+X^{12}+X^5+1$ 。
- CRC16_XMODEM，多项式：0x1021， $X^{16}+X^{12}+X^5+1$ 。
- CRC32_MPEG2，多项式：04C11DB7， $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ 。

19.3. 结构框图

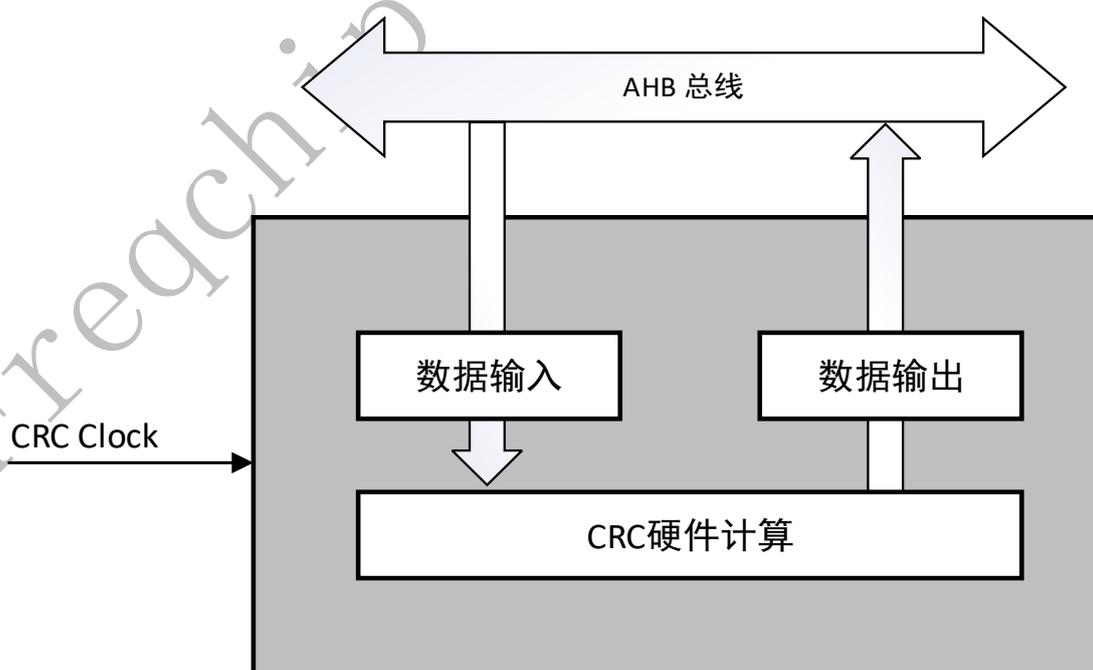


图 19-1 CRC 结构框图

19.4. 功能描述

表格 19-1 CRC 模式对比

模式	位宽	多项式	初始值	结果异或值	输入反转	输出反转
CRC8	8 Bit	0x07	0x00	0x00	False	False
CRC16_CCITT_FALSE	16 Bit	0x1021	0xFFFF	0xFFFF	False	False
CRC16_XMODEM	16 Bit	0x1021	0x0000	0x0000	False	False
CRC32_MPEG2	32 Bit	0x04C11DB7	0xFFFFFFFF	0x00000000	False	False

19.5. CRC 寄存器

CRC 基地址: 0x1013_0000

19.5.1. CRC 寄存器映像

表格 19-2 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	CRC_CTRL	CRC 控制寄存器
0x04	CRC_STATUS	CRC 状态寄存器
0x08	CRC_DATA	CRC 数据寄存器
0x0C	CRC_RESULT	CRC 结果寄存器

19.5.2. CRC_CTRL(Offset 0x00)

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	27'h0	Reserved
4	INIT_VAL	R/W	1'b0	CRC 初始值。 0: CRC 初始值设置为 0x00000000。 1: CRC 初始值设置为 0xFFFFFFFF。 CRC_RST = 1 时生效。
3	CRC_RST	R/W	1'b0	复位清除 CRC
2:1	CRC_MODE	R/W	2'b0	CRC 计算模式选择。 00: CRC8 01: CRC16_CCITT_FALSE 10: CRC16_XMODEM 11: CRC32_MPEG2
0	CRC_Start	R/W	1'b0	启动 CRC 计算

19.5.3. CRC_STATUS(Offset 0x04)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'b0	Reserved
0	CRC_DONE	R	1'b0	CRC 计算完成

19.5.4. CRC_DATA(Offset 0x08)

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'b0	Reserved
7:0	DATA	R/W	8'b0	CRC 数据输入

19.5.5. CRC_RESULT(Offset 0x0C)

比特	名称	属性	复位值	描述
31:0	RESULT	R	32'b0	CRC 计算结果

Frequchip Confidential

20. AES 硬件加速器（AES）

20.1. 概述

AES 硬件加速器可以实现 AES (Advanced Encryption Standard) 编码算法，即 Rijndael 算法。支持三种不同长度的密钥，分别是 AES-128、AES-192、AES-256。支持 ECB(电子密码本)模式和 CBC(加密分组链接)模式。

20.2. 主要特性

- 支持 128 位、192 位或 256 位密钥。
- 支持电子密码本（ECB）模式、加密分组链接（CBC）模式。

20.3. 功能描述

表格 20-1 AES 不同密钥长度对比

	密钥长度	加密块长度	执行轮数
AES-128	128 bit	128bit	10
AES-192	192 bit	128bit	12
AES-256	256 bit	128bit	14

AES 硬件加速器，支持三种密钥长度，加密以 128bit 一块为单位进行。

20.4. AES 寄存器

AES 基地址: 0x1100_0000

20.4.1. AES 寄存器映像

表格 20-2 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	DATA_IN0	原始数据输入寄存器 0
0x04	DATA_IN1	原始数据输入寄存器 1
0x08	DATA_IN2	原始数据输入寄存器 2
0x0C	DATA_IN3	原始数据输入寄存器 3
0x10	KEY_0	密钥 0
0x14	KEY_1	密钥 1
0x18	KEY_2	密钥 2
0x1C	KEY_3	密钥 3
0x20	KEY_4	密钥 4
0x24	KEY_5	密钥 5
0x28	KEY_6	密钥 6
0x2C	KEY_7	密钥 7
0x30	IV_0	初始化向量 0
0x34	IV_1	初始化向量 1
0x38	IV_2	初始化向量 2
0x3C	IV_3	初始化向量 3
0x40	AES_CTRL	AES 控制寄存器
0x44	AES_STATE	AES 状态寄存器
0x48	DATA_OUT0	加密数据输出寄存器 0
0x4C	DATA_OUT1	加密数据输出寄存器 1
0x50	DATA_OUT2	加密数据输出寄存器 2
0x54	DATA_OUT3	加密数据输出寄存器 3

20.4.2. DATA_IN0~3 (Offset 0x00, 0x04, 0x08, 0x0C)

比特	名称	属性	复位值	描述
31:0	DATA_IN0 ~ DATA_IN3	R/W	32'h00	IN0: 输入明文 0 ~ 31 IN1: 输入明文 32 ~ 63 IN2: 输入明文 64 ~ 95 IN3: 输入明文 96 ~ 127

注:

Big_Endian in AES_CTRL register is 0:

Datain[127:0] = {Datain_3[31:0], Datain_2[31:0], Datain_1[31:0], Datain_0[31:0]};

Big_Endian in AES_CTRL register is 1:

Datain[127:0] = { Datain_0[31:0], Datain_1[31:0], Datain_2[31:0], Datain_3[31:0]}.

20.4.3. KEY0~7 (Offset 0x10, 0x14, 0x18, 0x1C, 0x20, 0x24, 0x28, 0x2C)

比特	名称	属性	复位值	描述
31:0	KEY_0 ~ KEY_7	R/W	32'h00	KEY_0: 密钥 0 ~ 31 KEY_1: 密钥 32 ~ 63 KEY_2: 密钥 64 ~ 95 KEY_3: 密钥 96 ~ 127 KEY_4: 密钥 128 ~ 159 KEY_5: 密钥 160 ~ 191 KEY_6: 密钥 192 ~ 223 KEY_7: 密钥 223 ~ 255

注:

Big_Endian in AES_CTRL register is 0:

Key-128[127:0] = { Key_3[31:0], Key_2[31:0], Key_1[31:0], Key_0[31:0]};

Key-192[191:0] = { Key_5[31:0], Key_4[31:0], Key_3[31:0], Key_2[31:0], Key_1[31:0], Key_0[31:0]};

Key-256[255:0] = { Key_7[31:0], Key_6[31:0], Key_5[31:0], Key_4[31:0], Key_3[31:0], Key_2[31:0],
Key_1[31:0], Key_0[31:0]};

Big_Endian in AES_CTRL register is 1:

Key-128[127:0] = { Key_0[31:0], Key_1[31:0], Key_2[31:0], Key_3[31:0];

Key-192[191:0] = { Key_0[31:0], Key_1[31:0], Key_2[31:0], Key_3[31:0], Key_4[31:0], Key_5[31:0];

Key-256[255:0] = { Key_0[31:0], Key_1[31:0], Key_2[31:0], Key_3[31:0], Key_4[31:0], Key_5[31:0],
Key_6[31:0], Key_7[31:0];

20.4.4. IV0~3 (Offset 0x30, 0x34, 0x38, 0x3C)

比特	名称	属性	复位值	描述
31:0	IV0 ~ IV3	R/W	32'h00	IV0: initial vector 0 ~ 31 IV1: initial vector 32 ~ 63 IV2: initial vector 64 ~ 95 IV3: initial vector 96 ~ 127

注:

Big_Endian in AES_CTRL register is 0:

IV[127:0] = { IV_3[31:0], IV_2[31:0], IV_1[31:0], IV_0[31:0];

Big_Endian in AES_CTRL register is 1:

IV[127:0] = { IV_0[31:0], IV_1[31:0], IV_2[31:0], IV_3[31:0].

20.4.5. AES_CTRL (Offset 0x40)

比特	名称	属性	复位值	描述
31:9	Reserved	R/W	23'h00	Reserved
8	Big_Endian	R/W	1'b0	0:小端 1:大端
7:6	Opcode	R/W	1'b0	b00: encrypt b01: decrypt b10: key expand b11: reserved
5:4	Key_len	R/W	1'b0	b00: 128 bit key b01: 192 bit key b10: 256 bit key b11: reserved

3	Mode	R/W	1'b0	0: ECB mode 1: CBC mode
1:2	Reserved	R/W	2'b0	Reserved
0	Start	R/W	1'b0	AES 工作使能。硬件自动清除

20.4.6. AES_STATE (Offset 0x44)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	Busy	R	0'b0	AES 忙。 0: AES 空闲。 1: AES 忙。

20.4.7. DATA_OUT0~3 (Offset 0x48, 0x4C, 0x50, 0x54)

比特	名称	属性	复位值	描述
31:0	DATA_OUT0 ~ DATA_OUT3	R/W	32'h00	OUT0: 输出密文 0 ~ 31 OUT1: 输出密文 32 ~ 63 OUT2: 输出密文 64 ~ 95 OUT3: 输出密文 96 ~ 127

注:

Big_Endian in AES_CTRL register is 0:

Dataout[127:0]={Dataout_3[31:0],Dataout_2[31:0],Dataout_1[31:0],Dataout_0[31:0]};

Big_Endian in AES_CTRL register is 1:

Dataout[127:0]={Dataout_0[31:0],Dataout_1[31:0], Dataout_2[31:0],Dataout_3[31:0]}.

20.4.8. AES_CTRL (Offset 0x40)

比特	名称	属性	复位值	描述
31:9	Reserved	R/W	23'h00	Reserved
8	Big_Endian	R/W	1'b0	0:小端 1:大端
7:6	Opcode	R/W	1'b0	b00: encrypt b01: decrypt b10: key expand b11: reserved
5:4	Key_len	R/W	1'b0	b00: 128 bit key b01: 192 bit key b10: 256 bit key b11: reserved
3	Mode	R/W	1'b0	0: ECB mode 1: CBC mode
1:2	Reserved	R/W	2'b0	Reserved
0	Start	R/W	1'b0	AES 工作使能。硬件自动清除

20.4.9. AES_STATE (Offset 0x44)

比特	名称	属性	复位值	描述
31:1	Reserved	R/W	31'h00	Reserved
0	Busy	R	0'b0	AES 忙。 0: AES 空闲。 1: AES 忙。

20.4.10. DATA_OUT0~3 (Offset 0x48, 0x4C, 0x50, 0x54)

比特	名称	属性	复位值	描述
31:0	DATA_OUT0 ~ DATA_OUT3	R/W	32'h00	OUT0: 输出密文 0 ~ 31 OUT1: 输出密文 32 ~ 63 OUT2: 输出密文 64 ~ 95 OUT3: 输出密文 96 ~ 127

注:

Big_Endian in AES_CTRL register is 0:

Dataout[127:0]={Dataout_3[31:0],Dataout_2[31:0],Dataout_1[31:0],Dataout_0[31:0]};

Big_Endian in AES_CTRL register is 1:

Dataout[127:0]={Dataout_0[31:0],Dataout_1[31:0], Dataout_2[31:0],Dataout_3[31:0]}.

21. 散列处理器（SHA）

21.1. 概述

消息摘要的消息传递中有着广泛的应用。为了从消息中获得固定长度的消息摘要，哈希算法被提出，SHA 家族，SHA-1, SHA-224, SHA-256, SHA-384, SHA512, SHA-512/224 和 SHA-512/256。这些算法能够确定消息的完整性，对消息的任何更改都很可能产生不同的消息摘要。此属性在生成和验证数字签名和消息身份验证码以及生成随机数时非常有用。

每个算法实际上都是哈希计算。哈希计算从填充的消息生成一个消息调度，并使用该调度以及函数、常量操作来迭代地生成一系列哈希值。

21.2. 主要特性

- 支持 7 种散列算法：SHA-1, SHA-224, SHA-256, SHA-384, SHA512, SHA512/256

21.3. SHA 寄存器

21.3.1. SHA 寄存器映像

表格 21-1 SHA 寄存器映像

偏移	寄存器	寄存器描述
0x00	SHA_CTRL	控制寄存器
0x04	SHA_INT_STATE	状态寄存器
0x08	HASH_VAL_L[0]	哈希计算结果低 256bit
0x0C	HASH_VAL_L[1]	
0x10	HASH_VAL_L[2]	
0x14	HASH_VAL_L[3]	
0x18	HASH_VAL_L[4]	

0x1C	HASH_VAL_L[5]	哈希计算结果高 256bit
0x20	HASH_VAL_L[6]	
0x24	HASH_VAL_L[7]	
0x28	HASH_VAL_H[0]	
0x2C	HASH_VAL_H[1]	
0x30	HASH_VAL_H[2]	
0x34	HASH_VAL_H[3]	
0x38	HASH_VAL_H[4]	
0x3C	HASH_VAL_H[5]	
0x40	HASH_VAL_H[6]	
0x44	HASH_VAL_H[7]	
0x48~0x84	DATA_1[0] ~ DATA_1[15]	
0x88~0xB4	DATA_2[0] ~ DATA_2[15]	DATA2 512bit

21.3.2. SHA_CTRL (Offset 0x00)

比特	名称	属性	复位值	描述
31:7	Reserved	R/W	25'h00	Reserved
6	Calculate	R/W	1'b0	开启哈希计算。 计算完成后硬件自动清除。
5	Endian	R/W	1'b0	大小端模式： 0: 大端 1: 小端
4	Interrupt enable	R/W	1'b0	中断使能： 0: disable 1: enable
3	Initial enable	R/W	1'b0	哈希初值初始化。
2:0	Mode	R/W	3'b0	Mode: 001: SHA-1 010: SHA-256 011: SHA-224 100: SHA-512

				101: SHA-384 110: SHA-512/256 111: SHA-512/224
--	--	--	--	--

21.3.3. SHA_INT_STATE (Offset 0x04)

比特	名称	属性	复位值	描述
31:2	Reserved	R/W	30'h00	Reserved
1	Int_Error	R/W	1'b0	计算错误标志。 硬件置 '1' 后需软件清零。
0	Int_Done	R/W	1'b0	计算完成标志。 硬件置 '1' 后需软件清零。

21.3.4. HASH_VAL (Offset 0x08 ~ 0x24)

比特	名称	属性	复位值	描述
31:0	HASH_VAL_L[0] HASH_VAL_L[1] HASH_VAL_L[2] HASH_VAL_L[3] HASH_VAL_L[4] HASH_VAL_L[5] HASH_VAL_L[6] HASH_VAL_L[7]	R/W	32'h00	哈希计算结果。 选择 SHA-1, SHA-256, SHA-224 时, 哈希计算结果 512bit 储存在 HASH_VAL_L[0] ~ HASH_VAL_L[7]
31:0	HASH_VAL_H[0] HASH_VAL_H[1] HASH_VAL_H[2] HASH_VAL_H[3] HASH_VAL_H[4] HASH_VAL_H[5] HASH_VAL_H[6]	R/W	32'h00	哈希计算结果。 选择 SHA-384, SHA-512, SHA-5612/224, SHA-512/256 时, 哈希计算结果的高 512bit 储存在 HASH_VAL_H[0] ~ HASH_VAL_H[7]

	HASH_VAL_H[7]			
--	---------------	--	--	--

21.3.5. DATA_1 (Offset 0x48 ~ 0x84)

比特	名称	属性	复位值	描述
31:0	DATA_1[0] ~ DATA_1[15]	R/W	32'h00	数据输入。 选择 SHA-1, SHA-256, SHA-224 时, 数据按照 512bit/块写入。

21.3.6. DATA_2 (Offset 0x88 ~ 0xB4)

比特	名称	属性	复位值	描述
31:0	DATA_2[0] ~ DATA_2[15]	R/W	32'h00	数据输入。 选择 SHA-384, SHA-512 时, 数据按照 1024bit/块写入。低 512bit 写入 DATA_1 高 512bit 写入 DATA_2。

22. 联系信息

公司：上海富芮坤微电子有限公司

地址：中国(上海)自由贸易试验区碧波路 912 弄 8 号 501-A 室

电话：+86-21-5027-0080

Website: www.freqchip.com

Sales Email: sales@freqchip.com

本档的所有部分，其著作权归上海富芮坤微电子有限公司（简称富芮坤）所有，未经富芮坤授权许可，任何个人及组织不得复制、转载、仿制本档的全部或部分。富芮坤保留在不另行通知的情况下随时对产品或本档进行更改、修正、增强的权利。购买者应在订购前获得富芮坤产品的最新相关资料。

23. 附录 I

表格 23-1 PortA_L 功能复用

引脚 功能	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
0x0	GPIO_A0	GPIO_A1	GPIO_A2	GPIO_A3	GPIO_A4	GPIO_A5	GPIO_A6	GPIO_A7
0x1	UART0_RXD	UART0_TXD	UART0_CTS	UART0_RTS	UART1_RXD	UART1_TXD	UART1_CTS	UART1_RTS
0x2	SPIM0_SCLK	SPIM0_CSN	SPIM0_MOSI	SPIM0_MISO	SPIS0_SCLK	SPIS0_CSN	SPIS0_MOSI	SPIS0_MISO
0x3	I2C0_SCK	I2C0_SDA	I2C1_SCK	I2C1_SDA	I2C2_SCK	I2C2_SDA	I2C0_SCK	I2C0_SDA
0x4	PWM0_0	PWM0_1	PWM0_2	PWM0_3	PWM0_4	PWM0_5	PWM0_6	PWM0_7
0x5					MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX
0x6								
0x7						SPIMX8_0_DQS	SPIMX8_0_CLK	SPIMX8_0_CSN
0x8								
0x9					USB_DP	USB_DM	USB_DP	USB_DM
0xA			PDM0_SCK	PDM0_SDA			PDM0_SCK	PDM0_SDA
0xB					I2S0_SCLK	I2S0_FRM	I2S0_MOSI	I2S0_MISO
0xC							SPDIF_OUT	SPDIF_IN
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-2 PortA_H 功能复用

引脚 功能	PA8	PA9	PA10	PA11	PA12	PA13	PA14	PA15
0x0	GPIO_A8	GPIO_A9	GPIO_A10	GPIO_A11	GPIO_A12	GPIO_A13	GPIO_A14	GPIO_A15
0x1	UART2_RXD	UART2_TXD	UART2_CTS	UART2_RTS	UART1_RXD	UART1_TXD	UART1_CTS	UART1_RTS
0x2	SPIM0_SCLK	SPIM0_CSN	SPIM0_MOSI	SPIM0_MISO	SPIS0_SCLK	SPIS0_CSN	SPIS0_MOSI	SPIS0_MISO
0x3	I2C1_SCK	I2C1_SDA	I2C2_SCK	I2C2_SDA	I2C0_SCK	I2C0_SDA	I2C1_SCK	I2C1_SDA
0x4	PWM0_8	PWM0_9	PWM0_10	PWM0_11	PWM0_12	PWM0_13	PWM0_14	PWM0_15
0x5					MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX
0x6								
0x7	SPIMX8_0_IO[0/4]	SPIMX8_0_IO[1/5]	SPIMX8_0_IO[2/6]	SPIMX8_0_IO[3/7]	SPIMX8_0_IO[4/0]	SPIMX8_0_IO[5/1]	SPIMX8_0_IO[6/2]	SPIMX8_0_IO[7/3]
0x8	PARAL_CSX	PARAL8080_WRX PARAL6800_R/W	PARAL8080_RDX PARAL6800_EN	PARAL_DCX	PARAL_D0	PARAL_D1	PARAL_D2	PARAL_D3
0x9								
0xA			PDM0_SCK	PDM0_SDA			PDM0_SCK	PDM0_SDA
0xB					I2S0_SCLK	I2S0_FRM	I2S0_MOSI	I2S0_MISO
0xC							SPDIF_OUT	SPDIF_IN
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-3 PortB_L 复用功能

引脚 功能	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
0x0	GPIO_B0	GPIO_B1	GPIO_B2	GPIO_B3	GPIO_B4	GPIO_B5	GPIO_B6	GPIO_B7
0x1	UART2_RXD	UART2_TXD	UART2_CTS	UART2_RTS	UART3_RXD	UART3_TXD	UART3_CTS	UART3_RTS
0x2	SPIM1_SCLK	SPIM1_CSN	SPIM1_MOSI	SPIM1_MISO	SPIS0_SCLK	SPIS0_CSN	SPIS0_MOSI	SPIS0_MISO
0x3	I2C2_SCK	I2C2_SDA	I2C0_SCK	I2C0_SDA	I2C1_SCK	I2C1_SDA	I2C2_SCK	I2C2_SDA
0x4	PWM0_0	PWM0_1	PWM0_2	PWM0_3	PWM0_4	PWM0_5	PWM0_6	PWM0_7
0x5					MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX
0x6	SDIOH0_CLK	SDIOH0_CMD	SDIOH0_DAT[0/4]	SDIOH0_DAT[1/5]	SDIOH0_DAT[2/6]	SDIOH0_DAT[3/7]	SDIOH0_CLK	SDIOH0_CMD
0x7	SPIMX8_0_SCLK	SPIMX8_0_CSN	SPIMX8_0_IO0	SPIMX8_0_IO1	SPIMX8_0_IO2	SPIMX8_0_IO3	SPIMX8_0_SCLK	SPIMX8_0_CSN
0x8	PARAL_D4	PARAL_D5	PARAL_D6	PARAL_D7	PARAL_D8	PARAL_D9	PARAL_D10	PARAL_D11
0x9								
0xA			PDM0_SCK	PDM0_SDA			PDM1_SCK	PDM1_SDA
0xB	I2S0_SCLK	I2S0_FRM	I2S0_MOSI	I2S0_MISO				
0xC			SPDIF_OUT	SPDIF_IN				
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-4 PortB_H 复用功能

引脚 功能	PB8	PB9	PB10	PB11	PB12	PB13	PB14	PB15
0x0	GPIO_B8	GPIO_B9	GPIO_B10	GPIO_B11	GPIO_B12	GPIO_B13	GPIO_B14	GPIO_B15
0x1	UART2_RXD	UART2_TXD	UART2_CTS	UART2_RTS	UART3_RXD	UART3_TXD	UART3_CTS	UART3_RTS
0x2	SPIM1_SCLK	SPIM1_CSN	SPIM1_MOSI	SPIM1_MISO	SPIS0_SCLK	SPIS0_CSN	SPIS0_MOSI	SPIS0_MISO
0x3	I2C0_SCK	I2C0_SDA	I2C1_SCK	I2C1_SDA	I2C2_SCK	I2C2_SDA	I2C0_SCK	I2C0_SDA
0x4	PWM0_8	PWM0_9	PWM0_10	PWM0_11	PWM0_12	PWM0_13	PWM0_14	PWM0_15
0x5	MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX	SPIMX8_0_IO4	SPIMX8_0_IO5	SPIMX8_0_IO6	SPIMX8_0_IO7
0x6	SDIOH0_DAT[0/4]	SDIOH0_DAT[1/5]	SDIOH0_DAT[2/6]	SDIOH0_DAT[3/7]	SDIOH0_DAT[4/0]	SDIOH0_DAT[5/1]	SDIOH0_DAT[6/2]	SDIOH0_DAT[7/3]
0x7	SPIMX8_0_IO4	SPIMX8_0_IO5	SPIMX8_0_IO6	SPIMX8_0_IO7	SPIMX8_0_DQS		SPIMX8_1_SCLK	SPIMX8_1_CSN
0x8	PARAL_D12	PARAL_D13	PARAL_D14	PARAL_D15	PARAL_CSX	PARAL8080_WRX PARAL6800_R/W	PARAL8080_RDX PARAL6800_EN	PARAL_DCX
0x9			CM33_SWCK	CM33_SWD	QSPI1_IO3	QSPI1_IO0	QSPI1_IO2	QSPI1_IO1
0xA			PDM1_SCK	PDM1_SDA			PDM1_SCK	PDM1_SDA
0xB	I2S1_SCLK	I2S1_FRM	I2S1_MOSI	I2S1_MISO				
0xC			SPDIF_OUT	SPDIF_IN				
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

注：PB12 ~ PC15 为 1.8V 电压域。

表格 23-5 PortC_L 复用功能

引脚 功能	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0x0	GPIO_C0	GPIO_C1	GPIO_C2	GPIO_C3	GPIO_C4	GPIO_C5	GPIO_C6	GPIO_C7
0x1	UART2_RXD	UART2_TXD	UART2_CTS	UART2_RTS	UART3_RXD	UART3_TXD	UART3_CTS	UART3_RTS
0x2	SPIM1_SCLK	SPIM1_CSN	SPIM1_MOSI	SPIM1_MISO				
0x3	I2C3_SCK	I2C3_SDA	I2C4_SCK	I2C4_SDA	I2C5_SCK	I2C5_SDA	I2C3_SCK	I2C3_SDA
0x4	PWM1_0	PWM1_1	PWM1_2	PWM1_3	PWM1_4	PWM1_5	PWM1_6	PWM1_7
0x5	SPIMX8_0_IO0	SPIMX8_0_IO1	SPIMX8_0_IO2	SPIMX8_0_IO3	SPIMX8_0_SCLK	SPIMX8_0_CSN	SPIMX8_0_DQS	
0x6	SDIOH0_DAT[0/4]	SDIOH0_DAT[1/5]	SDIOH0_DAT[2/6]	SDIOH0_DAT[3/7]	SDIOH0_CLK	SDIOH0_CMD	SDIOH0_DAT[4/0]	SDIOH0_DAT[5/1]
0x7	SPIMX8_1_IO0	SPIMX8_1_IO1	SPIMX8_1_IO2	SPIMX8_1_IO3	SPIMX8_1_IO4	SPIMX8_1_IO5	SPIMX8_1_IO6	SPIMX8_1_IO7
0x8	PARAL_D0	PARAL_D1	PARAL_D2	PARAL_D3	PARAL_D4	PARAL_D5	PARAL_D6	PARAL_D7
0x9								
0xA			PDM1_SCK	PDM1_SDA			PDM1_SCK	PDM1_SDA
0xB	I2S1_SCLK	I2S1_FRM	I2S1_MOSI	I2S1_MISO				
0xC			SPDIF_OUT	SPDIF_IN				
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-6 PortC_H 复用功能

引脚 功能	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15
0x0	GPIO_C8	GPIO_C9	GPIO_C10	GPIO_C11	GPIO_C12	GPIO_C13	GPIO_C14	GPIO_C15
0x1	UART4_RXD	UART4_TXD	UART4_CTS	UART4_RTS	UART5_RXD	UART5_TXD	UART5_CTS	UART5_RTS
0x2	SPIM1_SCLK	SPIM1_CSN	SPIM1_MOSI	SPIM1_MISO	SPIM2_SCLK	SPIM2_CSN	SPIM2_MOSI	SPIM2_MISO
0x3	I2C4_SCK	I2C4_SDA	I2C5_SCK	I2C5_SDA	I2C3_SCK	I2C3_SDA	I2C4_SCK	I2C4_SDA
0x4	PWM1_8	PWM1_8	PWM1_8	PWM1_8	PWM1_8	PWM1_8	PWM1_8	PWM1_8
0x5	SPIS1_SCLK	SPIS1_CSN	SPIS1_IO0	SPIS1_IO1	SPIS1_IO2	SPIS1_IO3		
0x6	SDIOH0_DAT[6/2]	SDIOH0_DAT[7/3]	SDIOH1_CLK	SDIOH1_CMD	SDIOH1_DAT[0/4]	SDIOH1_DAT[1/5]	SDIOH1_DAT[2/6]	SDIOH1_DAT[3/7]
0x7	SPIMX8_1_SCLK	SPIMX8_1_CSN	SPIMX8_1_IO0	SPIMX8_1_IO1	SPIMX8_1_IO2	SPIMX8_1_IO3	SPIMX8_1_DQS	
0x8	PARAL_D8	PARAL_D9	PARAL_D10	PARAL_D11	PARAL_D12	PARAL_D13	PARAL_D14	PARAL_D15
0x9								
0xA			PDM2_SCK	PDM2_SDA			PDM2_SCK	PDM2_SDA
0xB	I2S1_SCLK	I2S1_FRM	I2S1_MOSI	I2S1_MISO	I2S2_SCLK	I2S2_FRM	I2S2_MOSI	I2S2_MISO
0xC			SPDIF_OUT	SPDIF_IN			SPDIF_OUT	SPDIF_IN
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-7 PortD_L 复用功能

引脚 功能	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7
0x0	GPIO_D0	GPIO_D1	GPIO_D2	GPIO_D3	GPIO_D4	GPIO_D5	GPIO_D6	GPIO_D7
0x1	UART4_RXD	UART4_TXD	UART4_CTS	UART4_RTS	UART5_RXD	UART5_TXD	UART5_CTS	UART5_RTS
0x2	SPIM2_SCLK	SPIM2_CSN	SPIM2_MOSI	SPIM2_MISO				
0x3	I2C5_SCK	I2C5_SDA	I2C3_SCK	I2C3_SDA	I2C4_SCK	I2C4_SDA	I2C5_SCK	I2C5_SDA
0x4	PWM1_0	PWM1_1	PWM1_2	PWM1_3	PWM1_4	PWM1_5	PWM1_6	PWM1_7
0x5	MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX	SPIS1_SCLK	SPIS1_CSN	SPIS1_IO0	SPIS1_IO1
0x6	SDIOH1_CLK	SDIOH1_CMD	SDIOH1_DAT[4/0]	SDIOH1_DAT[5/1]	SDIOH1_DAT[6/2]	SDIOH1_DAT[7/3]	SDIOH1_DAT[0/4]	SDIOH1_DAT[1/5]
0x7	SPIMX8_1_SCLK	SPIMX8_1_CSN	SPIMX8_1_IO0	SPIMX8_1_IO1	SPIMX8_1_IO2	SPIMX8_1_IO3	SPIMX8_1_IO4	SPIMX8_1_IO5
0x8	PARAL_D0	PARAL_D1	PARAL_D2	PARAL_D3	PARAL_D4	PARAL_D5	PARAL_D6	PARAL_D7
0x9								
0xA			PDM2_SCK	PDM2_SDA			PDM2_SCK	PDM2_SDA
0xB	I2S2_SCLK	I2S2_FRM	I2S2_MOSI	I2S2_MISO				
0xC			SPDIF_OUT	SPDIF_IN				
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

表格 23-8 PortD_H 复用功能

引脚 功能	PD8	PD9	PD10	PD11	PD12	PD13	PD14	PD15
0x0	GPIO_D8	GPIO_D9	GPIO_D10	GPIO_D11	GPIO_D12	GPIO_D13	GPIO_D14	GPIO_D15
0x1	UART4_RXD	UART4_TXD	UART4_CTS	UART4_RTS	UART5_RXD	UART5_TXD	UART5_CTS	UART5_RTS
0x2	SPIM2_SCLK	SPIM2_CSN	SPIM2_MOSI	SPIM2_MISO				
0x3	I2C3_SCK	I2C3_SDA	I2C4_SCK	I2C4_SDA	I2C5_SCK	I2C5_SDA	I2C3_SCK	I2C3_SDA
0x4	PWM1_8	PWM1_9	PWM1_10	PWM1_11	PWM1_12	PWM1_13	PWM1_14	PWM1_15
0x5	SPIS1_IO2	SPIS 1_IO3	SPIS 1_SCLK	SPIS 1_CSN	MCAN0_RX	MCAN0_TX	MCAN1_RX	MCAN1_TX
0x6	SDIOH1_DAT[2/6]	SDIOH1_DAT[3/7]	SDIOH1_CLK	SDIOH1_CMD	SDIOH1_DAT[4/0]	SDIOH1_DAT[5/1]	SDIOH1_DAT[6/2]	SDIOH1_DAT[7/3]
0x7	SPIMX8_1_IO6	SPIMX8_1_IO7	SPIMX8_1_DQS					
0x8	PARAL_CSX	PARAL8080_WRX PARAL6800_R/W	PARAL8080_RDX PARAL6800_EN	PARAL_DCX				
0x9					USB_DP	USB_DP	USB_DP	USB_DP
0xA			PDM2_SCK	PDM2_SDA			PDM2_SCK	PDM2_SDA
0xB	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK	I2S2_SCLK
0xC			SPDIF_OUT	SPDIF_IN			SPDIF_OUT	SPDIF_IN
0xD	CDC_I2STX_CLK	CDC_I2STX_FRM	CDC_I2STX_DO	CDC_I2SRX_DIN	CDC_I2SRX_FRM	CDC_I2SRX_CLK	CDC_CLKIN	CDC_CLKOUT
0xE								
0xF								

24. 附录 II

表格 24-1 PMU PortA 复用功能

引脚 \ 功能	2'b00	2'b01	2'b10	2'b11
PMU_IO0	PMU_IO0			
PMU_IO1	PMU_IO1			
PMU_IO2	PMU_IO2			
PMU_IO3	PMU_IO3			
PMU_IO4	PMU_IO4			
PMU_IO5	PMU_IO5			
PMU_IO6	PMU_IO6			
PMU_IO7	PMU_IO7			

25. 附录 III

