
FR800x 参考手册

支持 SIG MESH 的低功耗蓝牙 SOC 芯片

版本号： v1.1.6

日期： 2022.5



修订版本

版本号	日期	更新内容
V1.0	2021.6.1	首版
V1.0.1	2021.11.17	更新 UART 模块 5.4.2 DATA 和 DLL 寄存器定义
V1.1.0	2021.11.29	增加 DMA 的离散(Scatter)/聚合 (Gather) 功能 1. 增加 DMA 的 Scatter/gather 功能描述 2. 增加 SGRx 和 DSRx 寄存器 3. 修改 DMA 寄存器映像表，加入 SGRx 和 DSRx 寄存器 4. 修改 CTL 寄存器，支持 scatter/gather 功能 5.修改寄存器，主要有 12.4.9 RawTfr, 12.4.10 RawErr, 12.4.11 StatusTfr, 12.4.12 StatusErr, 12.4.13 MaskTfr, 12.4.14 MaskErr, 12.4.15 ClearTfr, 12.4.16 ClearErr, 12.4.17 StatusInt,12.4.19 ChEnReg 寄存器。
V1.1.1	2021.12.1	更新 SPI 模块 6.4.8 TXFTLR, 6.4.9 RXFTLR,6.4.10 TXFLR,6.4.11 RXFLR 寄存器； 7.4.5 TXFTLR, 7.4.6 RXFTLR, 7.4.7 TXFLR, 7.4.8 RXFLR 寄存器。
V1.1.2	2021.12.6	修改 12.3.2DMA 章节部分，修改传输总长度、数据位宽的关系。
V1.1.3	2022.1.4	修改 5.4.6 章节 MCR 寄存器描述
V1.1.4	2022.1.26	修改 1.2 章节 FIFO 描述，及 18 章 USB 端点描述
V1.1.5	2022.5.25	修改 ADKey 相关寄存器
V1.1.6	2022.6.20	USB 调试模式删除

目录

目录.....	I
表目录.....	XIII
图目录.....	XXVI
1. 系统概述.....	1
1.1. 概述.....	1
1.2. 主要特性.....	1
1.3. 存储器地址映像.....	5
1.4. 内核中断映像.....	6
2. 系统控制单元 (System Controller)	8
2.1. 概述.....	8
2.2. 系统时钟.....	8
2.2.1. 时钟源.....	8
2.2.2. 系统时钟树.....	9
2.2.3. 外设时钟选择.....	9
2.3. 系统复位.....	9
2.4. 系统低功耗.....	9
2.5. 系统控制寄存器.....	10
2.5.1. 系统寄存器映像.....	10
2.5.2. Clock Control (Offset 0x00)	11
2.5.3. Clock Enable (Offset 0x04)	12
2.5.4. QSPI0 Control (Offset 0x08)	13
2.5.5. Reset Control (Offset 0x0C)	14
2.5.6. APB Clock Enable (Offset 0x10)	15
2.5.7. DMA_REQ_MUX0 (Offset 0x14)	17
2.5.8. DMA_REQ_MUX1 (Offset 0x18)	18
2.5.9. DMA_REQ_MUX2 (Offset 0x1C)	18
2.5.10. PortA Function MUX (Offset 0x20)	19
2.5.11. PortB Function MUX (Offset 0x24)	19
2.5.12. PortC Function MUX (Offset 0x28)	20
2.5.13. PortD Function MUX (Offset 0x2C)	20
2.5.14. Portx Pull Enable (Offset 0x30)	20
2.5.15. Portx Pull Select (Offset 0x34)	21
2.5.16. PortE_CFG (Offset 0x38)	21
2.5.17. USB_OTG_CTRL (Offset 0x3C)	22
3. 通用输入输出接口 (GPIO)	24
3.1. 概述.....	24
3.2. 主要特性.....	24

3.3. 结构框图	24
3.4. GPIO 寄存器	25
3.4.1. GPIO 寄存器地址映像	25
3.4.2. PortA_DATA (Offset 0x00)	25
3.4.3. PortB_DATA (Offset 0x04)	25
3.4.4. PortC_DATA (Offset 0x08)	26
3.4.5. PortD_DATA (Offset 0x0C)	26
3.4.6. PortE_DATA (Offset 0x10)	26
3.4.7. PortA_OEN (Offset 0x20)	26
3.4.8. PortB_OEN (Offset 0x24)	27
3.4.9. PortC_OEN (Offset 0x28)	27
3.4.10. PortD_OEN (Offset 0x2C)	27
3.4.11. PortE_OEN (Offset 0x30)	28
3.4.12. Port_EXTI_EN0 (Offset 0x40)	28
3.4.13. Port_EXTI_EN1 (Offset 0x44)	28
4. 外部中断控制器 (EXTI)	30
4.1. 概述	30
4.2. 主要特性	30
4.3. EXTI 系统框图	30
4.4. EXTI 寄存器	30
4.4.1. EXTI 寄存器映像	30
4.4.2. EXTI_CTL0 (Offset 0x00)	31
4.4.3. EXTI_CTL1 (Offset 0x04)	31
4.4.4. EXTI_STS0 (Offset 0x08)	32
4.4.5. EXTI_STS1 (Offset 0x0C)	32
4.4.6. EXTI_TYPE0 (Offset 0x10)	32
4.4.7. EXTI_TYPE1 (Offset 0x14)	33
4.4.8. EXTI_TYPE2 (Offset 0x18)	34
4.4.9. EXTI_CNT0 ~ EXTI_CNT35 (Offset 0x20 ~ 0xAC)	34
5. 通用异步收发器 (UART)	36
5.1. 概述	36
5.2. 主要特性	36
5.3. 功能描述	36
5.3.1. 波特率计算	36
5.3.2. IrDA SIR 1.0 协议	36
5.4. Uart 寄存器	38
5.4.1. Uart 寄存器映像	38
5.4.2. DATA/DLL (Offset 0x00)	39
5.4.3. IER/DLH (Offset 0x04)	39
5.4.4. FCR/IID (Offset 0x08)	41
5.4.5. LCR (Offset 0x0C)	42

5.4.6. MCR (Offset 0x10)	43
5.4.7. LSR (Offset 0x14)	44
5.4.8. MSR (Offset 0x18)	45
5.4.9. USR (Offset 0x7C)	45
5.4.10. DLF (Offset 0xC0)	46
5.5. 使用流程	47
5.5.1. 发送数据流程	47
5.5.2. 接收数据流程	48
6. 串行外设接口主设备 (SPI_Master)	49
6.1. 概述	49
6.2. 主要特性	49
6.3. 功能描述	50
6.3.1. 发送和接收 FIFO 及中断	50
6.3.2. 传输模式	50
6.3.3. DMA 接口	51
6.4. SPI 寄存器	51
6.4.1. SPI 寄存器映像	51
6.4.2. CTRLR0 (Offset 0x00)	52
6.4.3. CTRLR1 (Offset 0x04)	54
6.4.4. SSIENR (Offset 0x08)	55
6.4.5. MWCR (Offset 0x0C)	55
6.4.6. SER (Offset 0x10)	55
6.4.7. BAUDR (Offset 0x14)	56
6.4.8. TXFTLR (Offset 0x18)	56
6.4.9. RXFTLR (Offset 0x1C)	56
6.4.10. TXFLR (Offset 0x20)	57
6.4.11. RXFLR (Offset 0x24)	57
6.4.12. SR (Offset 0x28)	57
6.4.13. IMR (Offset 0x2C)	58
6.4.14. ISR (Offset 0x30)	58
6.4.15. RISR (Offset 0x34)	58
6.4.16. TXOICR (Offset 0x38)	59
6.4.17. RXOICR (Offset 0x3C)	59
6.4.18. RXUICR (Offset 0x40)	59
6.4.19. ICR (Offset 0x48)	60
6.4.20. DMACR (Offset 0x4C)	60
6.4.21. DMATDLR (Offset 0x50)	60
6.4.22. DMARDLR (Offset 0x54)	60
6.4.23. DR(offset 0x60)	61
6.4.24. RX_SAMPLE_DLY (Offset 0xF0)	61
6.4.25. CTRLR2 (Offset 0xF4)	61

6.5. 使用流程	62
6.5.1. SPI 标准模式发送	62
6.5.2. SPI 标准模式接收	63
6.5.3. Dual (二线) 或 Quad (四线) 写	63
6.5.4. Dual (二线) 或 Quad (四线) 读	64
6.6. MOTOROLA SPI 时序	67
7. 串行外设接口从设备 (SPI_Slave)	68
7.1. 概述	68
7.2. 主要特性	68
7.3. 功能描述	69
7.3.1. 发送和接收 FIFO 及中断	69
7.3.2. 传输模式	69
7.3.3. DMA 接口	70
7.4. SPI 寄存器	70
7.4.1. SPI 寄存器映像	70
7.4.2. CTRLR0 (Offset 0x00)	71
7.4.3. SSIENR (Offset 0x08)	73
7.4.4. MWCR (Offset 0x0C)	73
7.4.5. TXFTLR (Offset 0x18)	73
7.4.6. RXFTLR (Offset 0x1C)	74
7.4.7. TXFLR (Offset 0x20)	74
7.4.8. RXFLR (Offset 0x24)	74
7.4.9. SR (Offset 0x28)	74
7.4.10. IMR (Offset 0x2C)	75
7.4.11. ISR (Offset 0x30)	75
7.4.12. RISR (Offset 0x34)	76
7.4.13. TXOICR (Offset 0x38)	76
7.4.14. RXOICR (Offset 0x3C)	76
7.4.15. RXUICR (Offset 0x40)	77
7.4.16. ICR (Offset 0x48)	77
7.4.17. DMACR (Offset 0x4C)	77
7.4.18. DMATDLR (Offset 0x50)	77
7.4.19. DMARDLR (Offset 0x54)	78
7.4.20. DR(offset 0x60)	78
7.5. 使用流程	78
7.6. MOTOROLA SPI 时序	80
8. 内部集成电路总线接口 (I2C)	81
8.1. 概述	81
8.2. 主要特性	81
8.3. 功能描述	82
8.3.1. I2C 时序控制	82

8.3.2. I2C_SDA_HOLD 功能.....	83
8.3.3. SCL、SDA 拉低超时功能.....	83
8.3.4. 7bit 地址格式、10bit 地址格式.....	85
8.3.5. 突刺抑制功能.....	85
8.4. I2C 寄存器.....	86
8.4.1. I2C 寄存器映像.....	86
8.4.2. CONTROL (Offset 0x00)	88
8.4.3. TAR (Offset 0x04)	89
8.4.4. SAR (Offset 0x08)	90
8.4.5. HS_MADDR (Offset 0x0C)	90
8.4.6. DATA_CMD (Offset 0x10)	90
8.4.7. SS_SCL_HCNT (Offset 0x14)	91
8.4.8. SS_SCL_LCNT (Offset 0x18)	91
8.4.9. FS_SCL_HCNT (Offset 0x1C)	92
8.4.10. FS_SCL_LCNT (Offset 0x20)	92
8.4.11. HS_SCL_HCNT (Offset 0x24)	92
8.4.12. HS_SCL_LCNT (Offset 0x28)	93
8.4.13. INTR_STAT (Offset 0x2C)	93
8.4.14. INTR_MASK (Offset 0x30)	96
8.4.15. RAW_INTR_STAT (Offset 0x34)	97
8.4.16. RX_TL (Offset 0x38)	97
8.4.17. TX_TL (Offset 0x3C)	98
8.4.18. CLR_INTR (Offset 0x40)	98
8.4.19. CLR_RX_UNDER (Offset 0x44)	98
8.4.20. CLR_RX_OVER (Offset 0x48)	98
8.4.21. CLR_TX_OVER (Offset 0x4C)	99
8.4.22. CLR_RD_REQ (Offset 0x50)	99
8.4.23. CLR_TX_ABRT (Offset 0x54)	99
8.4.24. CLR_RX_DONE (Offset 0x58)	99
8.4.25. CLR_ACTIVITY (Offset 0x5C)	100
8.4.26. CLR_STOP_DET (Offset 0x60)	100
8.4.27. CLR_START_DET (Offset 0x64)	100
8.4.28. ENABLE (Offset 0x6C)	101
8.4.29. STATUS (Offset 0x70)	101
8.4.30. TXFLR (Offset 0x74)	103
8.4.31. RXFLR (Offset 0x78)	103
8.4.32. SDA_HOLD (Offset 0x7C)	103
8.4.33. TX_ABRT_SOURCE (Offset 0x80)	104
8.4.34. SLV_DATA_NACK_ONLY (Offset 0x84)	105
8.4.35. DMA_CR (Offset 0x88)	106
8.4.36. DMA_TDLR (Offset 0x8C)	106

8.4.37. DMA_RDLR (Offset 0x90)	106
8.4.38. SDA_SETUP (Offset 0x94)	107
8.4.39. FS_SPKLEN (Offset 0xA0)	107
8.4.40. HS_SPKLEN (Offset 0xA4)	107
8.4.41. CLR_RESTART_DET (Offset 0xA8)	108
8.4.42. SCL_STUCK_AT_LOW_TIMEOUT (Offset 0xAC)	108
8.4.43. SDA_STUCK_AT_LOW_TIMEOUT (Offset 0xB0)	108
8.4.44. CLR_SCL_STUCK_DET (Offset 0xB4)	108
8.5. 使用流程	109
8.5.1. 主机使用流程	109
8.5.2. 从机使用流程	109
8.5.3. 工作时序图	110
9. 音频接口 (I2S)	113
9.1. 概述	113
9.2. 主要特性	113
9.3. 结构框图	113
9.4. 功能描述	114
9.4.1. CLK、WS 分频	114
9.4.2. WS_INV, DLY_EN, FMT 配置效果	114
9.5. I2S 寄存器	116
9.5.1. I2S 寄存器映像	116
9.5.2. CONTROL (Offset 0x00)	116
9.5.3. CLK_DIV (Offset 0x04)	117
9.5.4. WS_DIV (Offset 0x08)	117
9.5.5. DATA (Offset 0x0C)	117
9.5.6. STATUS (Offset 0x10)	118
9.5.7. INTE (Offset 0x14)	118
9.5.8. FIFO_CFG (Offset 0x20)	119
9.5.9. DMA_CFG (Offset 0x24)	119
10. 真随机数发生器 (TRNG)	121
10.1. 概述	121
10.2. 主要特性	121
10.3. TRNG 结构框图	121
10.4. TRNG 寄存器	121
10.4.1. TRNG 寄存器映像	121
10.4.2. RNG_IMR (Offset 0x100)	122
10.4.3. RNG_ISR (Offset 0x104)	123
10.4.4. RNG_ICR (Offset 0x108)	123
10.4.5. TRNG_CONFIG (Offset 0x10C)	124
10.4.6. TRNG_VALID (Offset 0x110)	124
10.4.7. HER_DATA0 ~ 5 (Offset 0x114 ~ 0x128)	124

10.4.8. RND_SOURCE_ENABLE (Offset 0x12C)	125
10.4.9. SAMPLE_CNT1 (Offset 0x130)	125
10.4.10. TRNG_RESET (Offset 0x140)	125
10.4.11. TRNG_BUSY (Offset 0x1B8)	125
10.4.12. RST_BITS_COUNTER (Offset 0x1BC)	126
11. 数字麦克风 (PDM)	127
11.1. 概述	127
11.2. 主要特性	127
11.3. PDM 系统框图	127
11.4. PDM 寄存器	127
11.4.1. PDM 寄存器映像	127
11.4.2. PDM_Control (Offset 0x00)	128
11.4.3. VOLUME (Offset 0x04)	129
11.4.4. VOLSTEP (Offset 0x08)	129
11.4.5. RX_DATA (Offset 0x0C)	130
11.4.6. FIFO_Clear (Offset 0x10)	130
11.4.7. FIFO_Status (Offset 0x14)	130
11.4.8. FIFO_INT_EN (Offset 0x18)	130
11.4.9. FIFO_Almost_Level (Offset 0x0x1C)	131
11.4.10. DMA_CFG (Offset 0x20)	131
12. DMA 控制器 (DMA)	132
12.1. 概述	132
12.2. 主要特性	132
12.3. 功能描述	132
12.3.1. DMA 优先级	132
12.3.2. 传输总长度、数据位宽、Burst length 三者关系	133
12.3.3. 链表传输功能	133
12.3.4. 离散与聚合功能	134
12.4. DMA 寄存器	136
12.4.1. DMA 寄存器映像	136
12.4.2. SARx (Offset 0x00, 0x58, 0xB0)	137
12.4.3. DARx (Offset 0x08, 0x60, 0xB8)	137
12.4.4. LLPx (Offset 0x10, 0x68, 0xC0)	137
12.4.5. CTLx (Offset 0x18, 0x70, 0xC8)	138
12.4.6. CFGx (Offset 0x40, 0x98, 0xF0)	140
12.4.7. SGRx(Offset 0x48, 0xA0, 0xF8)	142
12.4.8. DSRx(Offset 0x50, 0xA8, 0x100)	142
12.4.9. RawTfr (Offset 0x2C0)	142
12.4.10. RawErr (Offset 0x2E0)	143
12.4.11. StatusTfr (Offset 0x2E8)	143
12.4.12. StatusErr (Offset 0x308)	143

12.4.13. MaskTfr (Offset 0x310)	144
12.4.14. MaskErr (Offset 0x330)	144
12.4.15. ClearTfr (Offset 0x338)	145
12.4.16. ClearErr (Offset 0x358)	145
12.4.17. StatusInt (Offset 0x360)	145
12.4.18. DmaCfgReg (Offset 0x398)	146
12.4.19. ChEnReg (Offset 0x3A0)	146
12.5. 注意事项	147
13. 脉冲宽度调制 (PWM)	148
13.1. 概述	148
13.2. 主要特性	148
13.3. PWM 系统框图	149
13.4. PWM 寄存器	149
13.4.1. PWM 寄存器映像	149
13.4.2. PWM_Output_EN (Offset 0x04)	151
13.4.3. PWM_ENABLE (Offset 0x08)	152
13.4.4. PWM0~7_Posedge (Offset 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48)	152
13.4.5. PWM0~7_Negedge (Offset 0x14, 0x1C, 0x24, 0x2C, 0x34, 0x3C, 0x44, 0x4C)	152
13.4.6. PWM_Output_Select (Offset 0x90)	152
13.4.7. Capture_Prescale (Offset 0x94)	153
13.4.8. Capture_Status (Offset 0x98)	153
13.4.9. Capture_INT_EN (Offset 0x9C)	154
13.4.10. Capture_CTL (Offset 0xA0)	154
13.4.11. Capture_Value0~7 (Offset 0xA4 ~ 0xC0)	154
13.4.12. PWM_Update (Offset 0xE4)	155
13.4.13. PWM_DAC_Mode (Offset 0xE8)	155
13.4.14. PWM_Output_Value (Offset 0xF0)	155
13.4.15. PWM_Inverter_EN (Offset 0xF8)	156
13.4.16. PWM_CNT_EN (Offset 0xFC)	156
13.4.17. PWM0~7_Prescale (Offset 0x100, 0x108, 0x110, 0x118, 0x120, 0x128, 0x130, 0x138)	156
13.4.18. PWM0~7_Period (Offset 0x104, 0x10C, 0x114, 0x11C, 0x124, 0x12C, 0x134, 0x13C)	157
13.4.19. PWM0~7_DAC_DIV (Offset 0x180, 0x184, 0x188, 0x18C, 0x190, 0x194, 0x198, 0x19C)	157
13.5. 使用流程	157
13.5.1. PWM 模式	157
13.5.2. PWM_DAC 模式	158
14. 低功耗电源管理 (PMU)	160
14.1. 概述	160
14.2. 访问控制接口	160
14.3. PMU_IWDG	161

14.3.1. PMU_IWDG 功能描述.....	161
14.3.2. PMU_IWDG 系统框图.....	161
14.3.3. PMU_IWDG 寄存器映像.....	161
14.3.4. IWDG_CTRL (Block0, 0x26)	162
14.3.5. IWDG_LEN (Block0, 0x27, 0x28, 0x29)	162
14.3.6. IWDG_TOUT_LEN (Block0, 0x2A)	162
14.4. PMU_GPIO	163
14.4.1. PMU_GPIO 功能描述.....	163
14.4.2. PMU_GPIO 寄存器映像.....	163
14.4.3. PortA/B/C/D/E_SEL (Block0, 0x43, 0x44, 0x45, 0x46, 0x47)	164
14.4.4. PortA/B/C/D/E_PE (Block0, 0x48, 0x49, 0x4A, 0x4B, 0x4C)	164
14.4.5. PortA/B/C/D/E_PS (Block0, 0x4D, 0x4E, 0x4F, 0x50, 0x51)	165
14.4.6. PortA/B/C/D_Monitor_EN (Block0, 0x52, 0x53, 0x54, 0x55)	165
14.4.7. PortE_Monitor_EN (Block0, 0x56)	165
14.4.8. Analog_Monitor_EN (Block0, 0x57)	166
14.4.9. PortA/B/C/D_Reference_Status (Block0, 0x58, 0x59, 0x5A, 0x5B)	166
14.4.10. PortE_Reference_Status (Block0, 0x5C)	167
14.4.11. Analog_Reference_Status (Block0, 0x5D)	167
14.4.12. PortA/B/C/D/E_OEN (Block0, 0x66, 0x67, 0x68, 0x69, 0x6A)	168
14.4.13. PortA/B/C/D/E_Data (Block0, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F)	168
14.5. PMU_RTC	169
14.5.1. PMU_RTC 功能描述.....	169
14.5.2. PMU_RTC 寄存器映像.....	169
14.5.3. RTC_CTRL (Block1, 0x07)	169
14.5.4. RTC_CNT (Block1, 0x08, 0x09, 0x0A, 0x0B)	170
14.5.5. ALARM_A (Block1, 0x0C, 0x0D, 0x0E, 0x0F)	170
14.5.6. ALARM_B (Block1, 0x10, 0x11, 0x12, 0x13)	170
14.6. PMU_KeyScan	171
14.6.1. PMU_KeyScan 功能描述.....	171
14.6.2. PMU_KeyScan 寄存器映像.....	171
14.6.3. KeyScan_CTL (Block1, 0x18)	171
14.6.4. Scan_Interval (Block1, 0x19, 0x1A)	172
14.6.5. KeyScan_Config (Block1, 0x1B)	172
14.6.6. ROW_SEL (Block1, 0x1C)	173
14.6.7. COL_SEL (Block1, 0x1D, 0x1E, 0x1F)	173
14.6.8. KEY_VAL (Block1, 0x20 ~ 0x33)	173
14.7. PMU_PWM	174
14.7.1. PMU_PWM 功能描述.....	174
14.7.2. PMU_PWM 寄存器映像.....	174
14.7.3. PWM0 ~ 3_CTRL (Block1, 0x34, 0x35, 0x36, 0x37)	175
14.7.4. PWM0_PRC (Block1, 0x38, 0x39)	175

14.7.5. PWM0_HRC (Block1, 0x3A, 0x3B)	175
14.7.6. PWM1_PRC (Block1, 0x3C, 0x3D)	175
14.7.7. PWM1_HRC (Block1, 0x3E, 0x3F)	175
14.7.8. PWM2_PRC (Block1, 0x40, 0x41)	175
14.7.9. PWM2_HRC (Block1, 0x42, 0x43)	175
14.7.10. PWM3_PRC (Block1, 0x44, 0x45)	175
14.7.11. PWM3_HRC (Block1, 0x46, 0x47)	175
14.8. PMU_IOMux.....	176
14.8.1. 功能描述.....	176
14.8.2. PMU_IOMux 寄存器映像.....	176
14.8.3. PortA_IOMux (Block1, 0x6A, 0x6B)	176
14.8.4. PortB_IOMux (Block1, 0x6C, 0x6D)	177
14.8.5. PortC_IOMux (Block1, 0x6E, 0x6F)	177
14.8.6. PortD_IOMux (Block1, 0x70, 0x71)	178
14.8.7. PortE_IOMux (Block1, 0x72)	178
14.9. PMU_INT_CTRL.....	178
14.9.1. PMU_INT_CTRL 功能描述	178
14.9.2. PMU_INT_CTRL 寄存器映像	179
14.9.3. BAT_Full_Filter (Block1, 0x58)	179
14.9.4. ACOK_Filter (Block1, 0x59)	180
14.9.5. LVD_Filter (Block1, 0x5A)	180
14.9.6. OTD_Filter (Block1, 0x5B)	180
14.9.7. INT_EN0 (Block1, 0x5E)	180
14.9.8. INT_EN1 (Block1, 0x5F)	181
14.9.9. INT_CLR0 (Block1, 0x60)	182
14.9.10. INT_CLR1 (Block1, 0x61)	182
14.9.11. INT_STS0 (Block1, 0x62)	183
14.9.12. INT_STS1 (Block1, 0x63)	183
14.9.13. PortA/B/C/D_Monitor_Status (Block1, 0x64, 0x65, 0x66, 0x67)	183
14.9.14. PortE_Monitor_Status (Block1, 0x68)	184
14.9.15. ANA_Monitor_Status (Block1, 0x69)	184
15. 液晶屏驱动接口 (LCD)	186
15.1. 概述	186
15.2. 主要特性	186
15.3. LCD 结构框图	186
15.4. LCD 寄存器.....	187
15.4.1. LCD 寄存器地址映像.....	187
15.4.2. LCD_TXFF_PUSH (Offset 0x00)	187
15.4.3. LCD_CFG (Offset 0x04)	188
15.4.4. LCD_CFGWR_STATUS (Offset 0x08)	188
15.4.5. LCD_CFGRD_STATUS (Offset 0x0C)	188

15.4.6. LCD_APB_RD (Offset 0x10)	189
15.4.7. LCD_TXFF_AEMP_LV (Offset 0x14)	189
15.4.8. LCD_DAT_WR_LEN (Offset 0x18)	189
15.4.9. LCD_DAT_WR_CFG (Offset 0x1C)	189
15.4.10. LCD_DMA (Offset 0x20)	190
15.4.11. LCD_CSX (Offset 0x24)	191
15.4.12. LCD_CRM (Offset 0x28)	191
15.4.13. LCD_TXFF_CLR (Offset 0x2C)	192
15.4.14. LCD_INTF_CFG (Offset 0x30)	192
15.4.15. LCD_INT (Offset 0x34)	193
15.4.16. LCD_TE_VSYNC_CFG (Offset 0x38)	193
15.4.17. LCD_TE_HSYNC_CFG (Offset 0x3C)	194
15.5. 使用流程	194
15.5.1. 写命令或写参数	194
15.5.2. 写数据	195
15.5.3. 读数据	195
16. 通用定时器 (Timer)	197
16.1. 概述	197
16.2. 主要特性	197
16.3. 系统框图	197
16.4. Timer 寄存器	197
16.4.1. Timer 寄存器地址映像	197
16.4.2. LOAD_VALUE (Offset 0x00)	198
16.4.3. COUNT_VALUE (Offset 0x04)	198
16.4.4. CONTROL (Offset 0x08)	198
16.4.5. INT_CLR (Offset 0x0C)	199
16.5. 使用流程	199
17. 数模转换器 (ADC)	200
17.1. 概述	200
17.2. 主要特性	200
17.3. 结构框图	200
17.4. 功能描述	201
17.5. ADC 寄存器	201
17.5.1. ADC 寄存器映像	201
17.5.2. CONTROL (Offset 0x00)	202
17.5.3. DATA0 ~ DATA7 (Offset 0x04 ~ 0x20)	203
17.5.4. DATA (Offset 0x24)	204
17.5.5. FIFO_Status (Offset 0x28)	204
17.5.6. INT (Offset 0x2C)	205
17.5.7. INTC (Offset 0x30)	205
17.5.8. ADC_ANA_CTRL0 (Offset 0x34)	206

17.5.9. ADC_ANA_CTRL1 (Offset 0x38)	207
17.5.10. ADC_ANA_CTRL2 (Offset 0x3C)	207
18. USB 全速设备接口 (USB)	209
18.1.1. 概述	209
18.1.2. 主要特性	209
18.2. USB 系统框图	209
18.3. USB 寄存器	210
18.3.1. USB 寄存器地址映像	210
18.3.2. USB 公共寄存器 (Offset 0x00~0x0F)	212
18.3.3. USB 端点 0 控制状态寄存器 (Offset 0x10~0x1F)	218
18.3.4. USB 端点 1~5 控制状态寄存器 (Offset 0x10~0x1F)	220
18.3.5. USB 端点 FIFO 寄存器 (Offset 0x20~0x34)	228
18.4. 使用流程	229
18.4.1. 端点选择	229
18.4.2. 端点 FIFO 配置	229
18.4.3. Device 模式发送	229
18.4.4. Device 模式接收	230
18.4.5. Device 模式注意事项	230
18.4.6. Device 模式暂停	231
18.4.7. Device 模式 SOF 包	231
18.4.8. Host 模式接收	231
18.4.9. Host 模式发送	232
18.4.10. Host 模式传输调度	232
18.4.11. Host 模式 Reset	233
18.4.12. Host 模式 Suspend	233
19. 联系信息	234
20. 附录 I	235
21. 附录 II	240

表目录

表格 1-1 存储器映像.....	5
表格 1-2 中断映像表.....	6
表格 2-1 时钟源	8
表格 2-2 系统复位源.....	9
表格 2-3 低功耗模式.....	9
表格 2-4 系统寄存器映像.....	10
表格 2-5 时钟控制	11
表格 2-6 时钟使能	12
表格 2-7 QSPI0.....	13
表格 2-8 复位控制	14
表格 2-9 APB 时钟使能.....	15
表格 2-10 DM_REQ_MUX0	17
表格 2-11 DMA_REQ_MUX1	18
表格 2-12 DMA_REQ_MUX2	18
表格 2-13 PortA Function MUX	19
表格 2-14 PortB Function MUX	19
表格 2-15 PortC Function MUX.....	20
表格 2-16 PortD Function MUX.....	20
表格 2-17 Portx Pull Enable.....	20
表格 2-18 Portx Pull Select 寄存器	21
表格 2-19 PortE CFG 寄存器.....	21
表格 2-20 USB_OTG_CTRL 寄存器	22
表格 3-1 GPIO 寄存器地址映像	25
表格 3-2 PortA DATA 寄存器	25
表格 3-3 PortB DATA 寄存器	25
表格 3-4 PortC DATA 寄存器	26
表格 3-5 PortB DATA 寄存器	26

表格 3-6 PortE DATA 寄存器	26
表格 3-7 PortA OEN 寄存器	26
表格 3-8 PortB OEN 寄存器	27
表格 3-9 PortC OEN 寄存器	27
表格 3-10 PortD OEN 寄存器	27
表格 3-11 PortE OEN 寄存器	28
表格 3-12 Port_EXTI_EN0 寄存器	28
表格 3-13 Port_EXTI_EN1 寄存器	28
表格 4-1 EXTI 寄存器映像	30
表格 4-2 EXTI_CTL0 寄存器	31
表格 4-3 EXTI_CTL1 寄存器	31
表格 4-4 EXTI_STS0 寄存器	32
表格 4-5 EXTI_STS1 寄存器	32
表格 4-6 EXTI_TYPE0 寄存器	32
表格 4-7 EXTI_TYPE1 寄存器	33
表格 4-8 EXTI_TYPE2 寄存器	34
表格 4-9 EXTI_CNT0~EXTI_CNT35 寄存器	34
表格 5-1 Uart 寄存器映像	38
表格 5-2 DATA 寄存器	39
表格 5-3 DLL 寄存器	39
表格 5-4 IER 寄存器	39
表格 5-5 DLH 寄存器	40
表格 5-6 FCR 寄存器	41
表格 5-7 IID 寄存器	42
表格 5-8 LCR 寄存器	42
表格 5-9 MCR 寄存器	43
表格 5-10 LSR 寄存器	44
表格 5-11 MSR 寄存器	45
表格 5-12 USR 寄存器	45

表格 5-13 DLF 寄存器	46
表格 6-1 SPI 寄存器.....	51
表格 6-2 CTRLR0 寄存器	52
表格 6-3 CTRLR1 寄存器	54
表格 6-4 SSIENR 寄存器.....	55
表格 6-5 MWCR 寄存器	55
表格 6-6 SER 寄存器	55
表格 6-7 BAUDR 寄存器	56
表格 6-8 TXFTLR 寄存器.....	56
表格 6-9 RXFTLR 寄存器	56
表格 6-10 TXFLR 寄存器.....	57
表格 6-11 RXFLR 寄存器.....	57
表格 6-12 SR 寄存器	57
表格 6-13 IMR 寄存器	58
表格 6-14 ISR 寄存器.....	58
表格 6-15 RISR 寄存器.....	58
表格 6-16 TXOICR 寄存器.....	59
表格 6-17 RXOICR 寄存器.....	59
表格 6-18 RXUICR 寄存器.....	59
表格 6-19 ICR 寄存器	60
表格 6-20 DMACR 寄存器.....	60
表格 6-21 DMATDLR 寄存器	60
表格 6-22 DMARDLR 寄存器.....	60
表格 6-23 DR 寄存器	61
表格 6-24 RX_SAMPLE_DLY 寄存器.....	61
表格 6-25 CTRLR2 寄存器.....	61
表格 7-1 SPI 寄存器.....	70
表格 7-2 CTRLR0 寄存器	71
表格 7-3 SSIENR 寄存器.....	73

表格 7-4 MWCR 寄存器	73
表格 7-5 TXFTLR 寄存器.....	73
表格 7-6 RXFTLR 寄存器	74
表格 7-7 TXFLR 寄存器	74
表格 7-8 RXFLR 寄存器	74
表格 7-9 SR 寄存器.....	74
表格 7-10 IMR 寄存器.....	75
表格 7-11 ISR 寄存器.....	75
表格 7-12 RISR 寄存器.....	76
表格 7-13 TXOICR 寄存器.....	76
表格 7-14 RXOICR 寄存器.....	76
表格 7-15 RXUICR 寄存器.....	77
表格 7-16 ICR 寄存器	77
表格 7-17 DMACR 寄存器.....	77
表格 7-18 DMATDLR 寄存器	77
表格 7-19 DMARDLR 寄存器.....	78
表格 7-20 DR 寄存器	78
表格 8-1 I2C 时序参数与寄存器关系	82
表格 8-2 I2C 寄存器映像.....	86
表格 8-3 CONTROL 寄存器.....	88
表格 8-4 TAR 寄存器.....	89
表格 8-5 SAR 寄存器.....	90
表格 8-6 HS_MADDR 寄存器	90
表格 8-7 DATA_CMD 寄存器	90
表格 8-8 SS_SCL_HCNT 寄存器.....	91
表格 8-9 SS_SCL_LCNT 寄存器.....	91
表格 8-10 FS_SCL_HCNT 寄存器.....	92
表格 8-11 FS_SCL_LCNT 寄存器.....	92
表格 8-12 HS_SCL_HCNT 寄存器	92

表格 8-13 HS_SCL_LCNT 寄存器.....	93
表格 8-14 INTR_STAT 寄存器.....	93
表格 8-15 INTR_MASK 寄存器.....	96
表格 8-16 RAW_INTR_STAT 寄存器.....	97
表格 8-17 RX_TL 寄存器.....	97
表格 8-18 TX_TL 寄存器.....	98
表格 8-19 CLR_INTR 寄存器.....	98
表格 8-20 CLR_RX_UNDER 寄存器	98
表格 8-21 CLR_RX_OVER 寄存器	98
表格 8-22 CLR_TX_OVER 寄存器	99
表格 8-23 CLR_RD_REQ 寄存器.....	99
表格 8-24 CLR_TX_ABRT 寄存器	99
表格 8-25 CLR_RX_DONE 寄存器	99
表格 8-26 CLR_ACTIVITY 寄存器	100
表格 8-27 CLR_STOP_DET 寄存器.....	100
表格 8-28 CLR_START_DET 寄存器	100
表格 8-29 ENABLE 寄存器	101
表格 8-30 STATUS 寄存器.....	101
表格 8-31 TXFLR 寄存器.....	103
表格 8-32 RXFLR 寄存器.....	103
表格 8-33 SDA_HOLD 寄存器	103
表格 8-34 TX_ABRT_SOURCE 寄存器	104
表格 8-35 SLV_DATA_NACK_ONLY 寄存器	105
表格 8-36 DMA_CR 寄存器.....	106
表格 8-37 DMA_TDLR 寄存器.....	106
表格 8-38 DMA_RDLR 寄存器.....	106
表格 8-39 SDA_SETUP 寄存器	107
表格 8-40 FS_SPKLEN 寄存器.....	107
表格 8-41 HS_SPKLEN 寄存器	107

表格 8-42 CLR_RESTART_DET 寄存器	108
表格 8-43 SCL_STUCK_AT_LOW_TIMEOUT 寄存器	108
表格 8-44 SDA_STUCK_AT_LOW_TIMEOUT 寄存器	108
表格 8-45 CLR_SCL_STUCK_DET 寄存器	108
表格 9-1 I2S 寄存器映像	116
表格 9-2 CONTROL 寄存器	116
表格 9-3 CLK_DIV 寄存器	117
表格 9-4 WS_DIV 寄存器	117
表格 9-5 DATA 寄存器	117
表格 9-6 STATUS 寄存器	118
表格 9-7 INTE 寄存器	118
表格 9-8 FIFO_CFG 寄存器	119
表格 9-9 DMA_CFG 寄存器	119
表格 10-1 TRNG 寄存器映像	121
表格 10-2 RNG_IMR 寄存器	122
表格 10-3 RNG_ISR 寄存器	123
表格 10-4 RNG_ICR 寄存器	123
表格 10-5 TRNG_CONFIG 寄存器	124
表格 10-6 TRNG_VALID 寄存器	124
表格 10-7 HER_DATA0~5 寄存器	124
表格 10-8 RND_SOURCE_ENABLE 寄存器	125
表格 10-9 SAMPLE_CNT1 寄存器	125
表格 10-10 TRNG_RESET 寄存器	125
表格 10-11 TRNG_BUSY 寄存器	125
表格 10-12 RST_BITS_COUNTER 寄存器	126
表格 11-1 PDM 寄存器映像	128
表格 11-2 PDM_Control 寄存器	128
表格 11-3 VOLUME 寄存器	129
表格 11-4 VOLSTEP 寄存器	129

表格 11-5 RX_DATA 寄存器.....	130
表格 11-6 FIFO_Clear 寄存器.....	130
表格 11-7 FIFO_Status 寄存器.....	130
表格 11-8 FIFO_INT_EN 寄存器.....	130
表格 11-9 FIFO_Almost_Level 寄存器.....	131
表格 11-10 DMA_CFG 寄存器.....	131
表格 12-1 DMA 寄存器映像.....	136
表格 12-2 SARx 寄存器.....	137
表格 12-3 DARx 寄存器.....	137
表格 12-4 LLPx 寄存器.....	137
表格 12-5 CTLx 寄存器.....	138
表格 12-6 CFGx 寄存器.....	140
表格 12-7 RawTfr 寄存器.....	142
表格 12-8 RawErr 寄存器.....	143
表格 12-9 StatusTfr 寄存器.....	143
表格 12-10 StatusErr 寄存器.....	143
表格 12-11 MaskTfr 寄存器.....	144
表格 12-12 MaskErr 寄存器.....	144
表格 12-13 ClearTfr 寄存器.....	145
表格 12-14 ClearErr 寄存器.....	145
表格 12-15 StatusInt 寄存器.....	145
表格 12-16 DmaCfgReg 寄存器.....	146
表格 12-17 ChEnReg 寄存器.....	146
表格 13-1 PWM 寄存器映像.....	149
表格 13-2 PWM_Output_EN 寄存器.....	151
表格 13-3 PWM_ENABLE 寄存器.....	152
表格 13-4 PWM0~7_Posedge 寄存器.....	152
表格 13-5 PWM0~7_Negedge 寄存器.....	152
表格 13-6 PWM_Output_Select 寄存器.....	152

表格 13-7 Capture_Prescale 寄存器	153
表格 13-8 Capture_Status 寄存器	153
表格 13-9 Capture_INT_EN 寄存器	154
表格 13-10 Capture_CTL 寄存器	154
表格 13-11 Capture_Value0~7 寄存器	154
表格 13-12 PWM_Update 寄存器	155
表格 13-13 PWM_DAC_Mode 寄存器	155
表格 13-14 PWM_Output_Value 寄存器	155
表格 13-15 PWM_Inverter_EN 寄存器	156
表格 13-16 PWM_CNT_EN 寄存器	156
表格 13-17 PWM0~7_Prescale 寄存器	156
表格 13-18 PWM0~7_Period 寄存器	157
表格 13-19 PWM0~7_DAC_DIV 寄存器	157
表格 14-1 PMU_IWDG 寄存器映像	161
表格 14-2 IWDG_CTRL 寄存器	162
表格 14-3 IWDG_LEN 寄存器	162
表格 14-4 IWDG_TOUT_LEN 寄存器	162
表格 14-5 PMU_GPIO 寄存器映像	163
表格 14-6 PortA/B/C/D/E_SEL 寄存器	164
表格 14-7 PortA/B/C/D/E_PE 寄存器	164
表格 14-8 PortA/B/C/D/E_PS 寄存器	165
表格 14-9 PortA/B/C/D Monitor EN 寄存器	165
表格 14-10 PortE Monitor EN 寄存器	165
表格 14-11 Analog Monitor_EN 寄存器	166
表格 14-12 PortA/B/C/D Reference Status 寄存器	166
表格 14-13 PortE Reference Status 寄存器	167
表格 14-14 Analog Reference Status 寄存器	167
表格 14-15 PortA/B/C/D/E_OEN 寄存器	168
表格 14-16 PortA/B/C/D/E_Data 寄存器	168

表格 14-17 PMU_RTC 寄存器映像	169
表格 14-18 RTC_CTRL 寄存器	169
表格 14-19 RTC_CNT 寄存器	170
表格 14-20 ALARM_A 寄存器	170
表格 14-21 ALARM_B 寄存器	170
表格 14-22 PMU_KeyScan 寄存器映像	171
表格 14-23 KeyScan_CTL 寄存器	171
表格 14-24 Scan_Interval 寄存器	172
表格 14-25 KeyScan_Config 寄存器	172
表格 14-26 ROW_SEL 寄存器	173
表格 14-27 COL_SEL 寄存器	173
表格 14-28 KEY_VAL 寄存器	173
表格 14-29 PMU_PWM 寄存器映像	174
表格 14-30 PWM0~3_CTRL 寄存器	175
表格 14-31 PWM0~3_PRC 寄存器	175
表格 14-32 PWM0~3_HRC 寄存器	175
表格 14-33 PMU_IOMux 寄存器映像	176
表格 14-34 PortA IOMux 寄存器	176
表格 14-35 PortB IOMux 寄存器	177
表格 14-36 PortC IOMux 寄存器	177
表格 14-37 PortD IOMux 寄存器	178
表格 14-38 PortE IOMux 寄存器	178
表格 14-39 PMU_INT_CTRL 寄存器映像	179
表格 14-40 BAT_Full_Filter 寄存器	179
表格 14-41 ACOK_Filter 寄存器	180
表格 14-42 LVD_Filter 寄存器	180
表格 14-43 OTD_Filter 寄存器	180
表格 14-44 INT_EN0 寄存器	180
表格 14-45 INT_EN1 寄存器	181

表格 14-46 INT_CLR0 寄存器	182
表格 14-47 INT_CLR1 寄存器	182
表格 14-48 INT_STS0 寄存器	183
表格 14-49 INT_STS1 寄存器	183
表格 14-50 PortA/B/C/D_Monitor_Status 寄存器	183
表格 14-51 PortE_Monitor_Status 寄存器	184
表格 14-52 ANA_Monitor_Status 寄存器	184
表格 15-1 LCD 寄存器地址映像	187
表格 15-2 LCD_TXFF_PUSH 寄存器	187
表格 15-3 LCD_CFG 寄存器	188
表格 15-4 LCD_CFGWR_STATUS 寄存器	188
表格 15-5 LCD_CFGRD_STATUS 寄存器	188
表格 15-6 LCD_APB_RD 寄存器	189
表格 15-7 LCD_TXFF_AEMP_LV 寄存器	189
表格 15-8 LCD_DAT_WR_LEN 寄存器	189
表格 15-9 LCD_DAT_WR_CFG 寄存器	189
表格 15-10 LCD_DMA 寄存器	190
表格 15-11 LCD_CSX 寄存器	191
表格 15-12 LCD_CRM 寄存器	191
表格 15-13 LCD_TXFF_CLR 寄存器	192
表格 15-14 LCD_INTF_CFG 寄存器	192
表格 15-15 LCD_INT 寄存器	193
表格 15-16 LCD_TE_VSYNC_CFG 寄存器	193
表格 15-17 LCD_TE_HSYNC_CFG 寄存器	194
表格 16-1 Timer 寄存器地址映像	198
表格 16-2 LOAD_VALUE 寄存器	198
表格 16-3 COUNT_VALUE 寄存器	198
表格 16-4 CONTROL 寄存器	198
表格 16-5 INT_CLR 寄存器	199

表格 17-1 ADC 寄存器映像	201
表格 17-2 CONTROL 寄存器	202
表格 17-3 DATA0~DATA7 寄存器	203
表格 17-4 DATA 寄存器	204
表格 17-5 FIFO_Status 寄存器	204
表格 17-6 INT 寄存器	205
表格 17-7 INTC 寄存器	205
表格 17-8 ADC_ANA_CTRL0 寄存器	206
表格 17-9 ADC_ANA_CTRL1 寄存器	207
表格 17-10 ADC_ANA_CTRL2 寄存器	207
表格 18-1 USB 寄存器地址映像	210
表格 18-2 Faddr 寄存器	212
表格 18-3 Power 寄存器	212
表格 18-4 IntrTx1 寄存器	213
表格 18-5 IntrTx2 寄存器	214
表格 18-6 IntrRx1 寄存器	214
表格 18-7 IntrRx2 寄存器	214
表格 18-8 IntrUSB 寄存器	214
表格 18-9 IntrTx1E 寄存器	215
表格 18-10 IntrTx2E 寄存器	215
表格 18-11 IntrRx1E 寄存器	215
表格 18-12 IntrRx2E 寄存器	216
表格 18-13 IntrUSB_E 寄存器	216
表格 18-14 Frame1 寄存器	216
表格 18-15 Frame2 寄存器	217
表格 18-16 Index 寄存器	217
表格 18-17 DevCtl 寄存器	217
表格 18-18 Device Mode CSR0 寄存器	218
表格 18-19 Host Mode CSR0 寄存器	219

表格 18-20 CSR2 寄存器	219
表格 18-21 Count0 寄存器	220
表格 18-22 NAKLimit0 寄存器	220
表格 18-23 TxMaxP 寄存器	220
表格 18-24 Device Mode TxCSR1 寄存器	220
表格 18-25 Host Mode TxCSR1 寄存器	221
表格 18-26 TxCSR2 寄存器	222
表格 18-27 RxMaxP 寄存器	223
表格 18-28 Device Mode RxCSR1 寄存器	223
表格 18-29 Host Mode RxCSR1 寄存器	224
表格 18-30 RxCSR2 寄存器	224
表格 18-31 RxCount1 寄存器	225
表格 18-32 RxCount2 寄存器	225
表格 18-33 TxType 寄存器	226
表格 18-34 TxInterval 寄存器	226
表格 18-35 RxType 寄存器	226
表格 18-36 RxInterval 寄存器	227
表格 18-37 TxFIFO1、TxFIFO2 寄存器	227
表格 18-38 RxFIFO1、RxFIFO2 寄存器	228
表格 18-39 USB 端点 FIFO 寄存器	228
表格 20-1 PortA 功能复用	235
表格 20-2 PortB 复用功能	236
表格 20-3 PortC 复用功能	237
表格 20-4 PortD 复用功能	238
表格 20-5 PortE 复用功能	239
表格 21-1 PMU PortA 复用功能	240
表格 21-2 PMU PortB 复用功能	240
表格 21-3 PMU PortC 复用功能	240
表格 21-4 PMU PortD 复用功能	241

表格 21-5 PMU PortE 复用功能	241
------------------------------	-----

Freqchip Confidential

图目录

图 1-1 系统框图	1
图 2-1 系统时钟树	8
图 2-2 外设时钟配置.....	9
图 3-1 GPIO 结构框图.....	24
图 4-1 EXTI 系统框图	30
图 5-1 IrDA 框图	37
图 5-2 IrDA 时序示意图.....	38
图 6-1 SPI Master 接线示意图.....	49
图 6-2 指令和地址使用标准模式、数据使用多线模式	64
图 6-3 指令使用标准模式、地址和数据使用多线模式.....	64
图 6-4 指令、地址和数据使用多线模式.....	64
图 6-5 只有指令使用多线模式	64
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.....	65
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.....	65
图 6-8 读数据，指令、地址和数据使用多线模式	66
图 6-9 读数据，没有指令，没有地址，仅数据传输	66
图 6-10 MOTOROLA SPI 时序图.....	67
图 7-1 SPI Slave 接线示意图.....	68
图 7-2 MOTOROLA SPI 时序图.....	80
图 8-1 I2C 时序参数图.....	82
图 8-2 SDA_TX_HOLD 使用示意图	83
图 8-3 SDA_RX_HOLD 使用示意图	83
图 8-4 主机发送 9 个 CLK 试图恢复 SDA.....	84
图 8-5 7bit 地址示意图	85
图 8-6 10bit 地址示意图	85
图 8-7 突刺抑制示意图.....	86
图 8-8 DATA_CMD 寄存器示意图.....	111

图 8-9 主机发送并产生 STOP	111
图 8-10 主机接收并产生 STOP 时序.....	111
图 8-11 主机发送并产生 Restart 时序.....	112
图 8-12 主机接收并产生 Restart 时序.....	112
图 8-13 主机发送, 产生 STOP, TxFIFO 非空.....	112
图 8-14 主机接收, 产生 STOP, TxFIFO 非空.....	112
图 9-1 I2S 结构框图.....	113
图 9-2 I2S 时序图 (左对齐)	114
图 9-3 I2S 时序图 (左对齐, WS 反相)	115
图 9-4 I2S 时序图 (左对齐, WS 反相, DLY_EN = 1)	115
图 9-5 I2S 时序图 (右对齐)	115
图 10-1 TRNG 结构框图.....	121
图 11-1 PDM 系统框图.....	127
图 12-1 离散传输示例图.....	135
图 12-2 源聚合传输示例图.....	136
图 13-1 PWM 系统框图	149
图 13-2 PWM 时序图	158
图 13-3 PWM_DAC 时序.....	159
图 13-4 输入捕获时序图.....	159
图 14-1 PMU_IWDG 系统框图.....	161
图 15-1 LCD 结构框图.....	186
图 15-2 8080 写命令时序	194
图 15-3 8080 写数据时序	195
图 15-4 8080 读数据时序	196
图 16-1 Timer 系统框图.....	197
图 17-1 ADC 结构框图.....	200
图 18-1 USB 系统框图	210

1. 系统概述

1.1. 概述

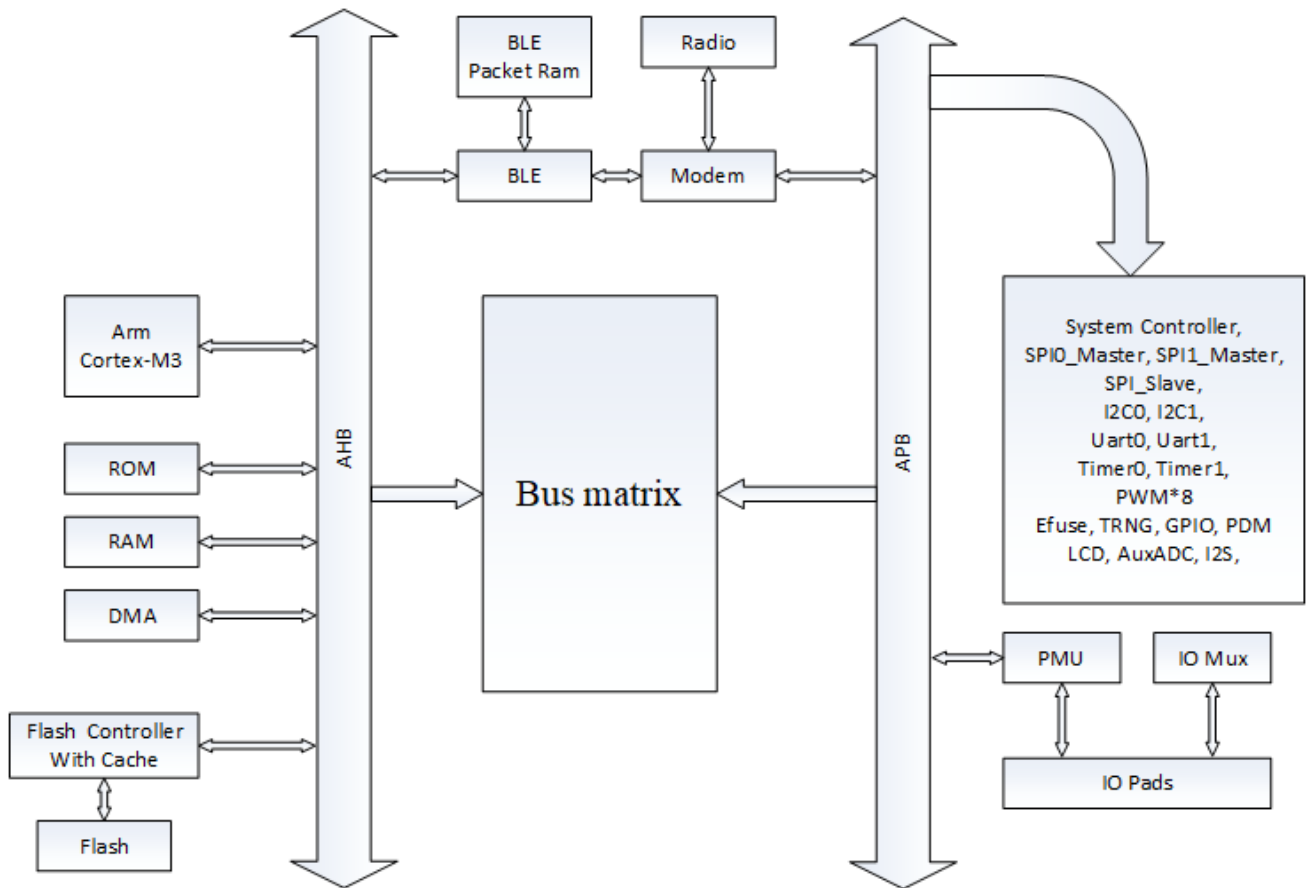


图 1-1 系统框图

1.2. 主要特性

- RF 收发器
 - Bluetooth Radio integrated balun
 - 外部 PA 支持
 - 发送功率 -20dBm~10dBm
 - 接收灵敏度 -96dBm@1Mbps
 - 接收灵敏度 -93dBm@2Mbps

- 接收灵敏度 -101dBm@125Kbps
- 接收灵敏度 -98dBm@500Kbps
- 电源管理 PMU
 - DC-DC Buck Inside
 - IO LDO
 - CORE/Analog LDO
 - Charger
 - Battery QS
 - Rcosc 32K
 - Power mode switch
 - BLE sniff mode clock & counter
- 基带和调制解调器
 - 蓝牙 5.2 BLE 兼容 1M/2M/125K/500K
 - 精度为 1dB 的 RSSI
 - RF rx & tx block 动态控制器
- CPU
 - Cortex-M3 内核
 - 48MHz、96MHz 可配置主频
- 存储器
 - 128KB ROM
 - RAM
 - ◆ 8KB (BLE Packet RAM)
 - ◆ 32KB (系统 RAM0) + 16KB (系统 RAM1) + 8KB (系统 RAM2)
 - ◆ 4KB (Cache)
 - ◆ 上述所有内存块都可以是保留单个内存块
- 外围接口
 - ◆ Timer*2
 - ✧ 系统 cpu/bus 时钟作为参考时钟
 - ✧ 带重载支持的 32 位计数器

- ◆ Efuse 128bits
- ◆ SPIM*2
 - 主时钟支持 96MHz(sclk up to 48MHz)
 - RX/TX FIFO 配置 深度 32，位宽 16bit
 - X2/X4 增强 SPI 模式
 - DMA 硬件握手
 - 传输数据 SPIM0->SPIM1 旁路总线矩阵
- ◆ Flash2LCD Engine
- ◆ 8080 Parallel interfaces with FIFO 32*32 & DMA handshake
- ◆ SPIS
 - RX/TX FIFO 配置 深度 16，位宽 16bit
 - 只支持标准的 SPI 访问模式
 - DMA 硬件握手
- ◆ QSPI*2
 - 可支持外接 PSRAM(FR8008xP 系列芯片已内置 PSRAM)
- ◆ I2C0
 - ✧ RX/TX FIFO 深度 32，位宽 8bit
 - ✧ 标准模式 & 高速模式
 - ✧ DMA 硬件握手
- ◆ I2C1
 - ✧ RX/TX FIFO 深度 32，位宽 8bit
 - ✧ 仅支持标准模式
 - ✧ DMA 硬件握手
- ◆ UART0
 - ✧ RX/TX FIFO 深度 32，宽度 8bit
 - ✧ 9 比特模式
 - ✧ 流控
 - ✧ IrDA SIR1.0
 - ✧ RS485 接口支持

- ✧ DMA 硬件握手
- ◆ UART1
 - ✧ RX/TX FIFO 深度 16，宽度 8bit
 - ✧ DMA 硬件握手
- ◆ GPIO*34(PA/PA/PC/PD/PE)
 - ✧ PA~PD *8, PE*2
 - ✧ 外部中断支持
- ◆ PDM
 - ✧ 16 位字 Mono PDM
 - ✧ RX FIFO 深度 32，位宽 16bit
 - ✧ 1M、1.5M、2M、3M 时钟支持
 - ✧ 可配置 62.5 或 125 下采样率转换
- ◆ TRNG
 - ✧ 当内核运行在 96MHz 时，产生 5K 位/秒的熵
 - ✧ 内置的硬件测试自相关和 CRNGT
 - ✧ 应用程序可以在 6 个 32 位只读寄存器中访问随机数
- ◆ 内部 FRSPIM
 - ✧ PMU 访问串行接口
 - ✧ Burst 支持@最大长度位 4 字节
 - ✧ CSNA/CSNB for PMU PK/PD
- ◆ AUXADC (SARADC)
 - ✧ 最大采样数据速率 1Mbps
 - ✧ 环回模式 for X8 input
 - ✧ Fixed mode for FIFO
 - ✧ 采样触发器遵从 PWM 脉冲
- ◆ PWMx8
 - ✧ 所有 PWM 通道的周期频率和占空比可以单独配置
 - ✧ 支持在一个 PWM 周期内的任意位置通过定时器触发 ADC 采样
 - ✧ 支持 2 路 PWM 组成总共 4 组互补输出,内置硬件逻辑电路设置死区

◆ GPIO Pads

- ✧ 用户可用 *34
- ✧ 支持中断源 *34
- ✧ 支持 SARADC*8
- ✧ UART/SPI/I2C/PDM/GPIO Mux 输出
- ✧ 上下拉浮动可配置

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
0x1000_0000 ~ 0x10FF_FFFF	Flash 最大支持 16M byte。 (不同封装存储有所不同)	AHB
0x1100_0000 ~ 0x1100_DFFF	RAM。 系统 RAM 底部的 8KB 可分配给 BLE Packet RAM。	
0x4000_4000 ~ 0x4000_5FFF	BLE Packet RAM。 BLE 包缓存内存，具有独占的 8KB。可扩展来自系统的 8KB 作为 BLE Packet RAM。	
0x2200_0000 ~ 0x22FF_FFFF	PSRAM/Flash Extend. 通过 QSPI1 扩展 RAM 或 Flash。增加扩展后可通过地址直接访问。	
0x2002_0000	DMA	
0x2004_0000	USB	
0x5000_0000	System Controller	APB
0x5001_0000	Timer	
0x5002_0000	PMU	
0x5002_8000	Efuse	
0x5003_0000	SPI0_Master	
0x5003_4000	SPI1_Master	
0x5003_8000	SPI_Slave	
0x5004_0000	I2C0	
0x5004_8000	I2C1	
0x5005_0000	Uart0	
0x5005_8000	Uart1	

起始地址	描述	总线
0x5006_0000	GPIO	
0x5007_0000	I2S	
0x5008_0000	PWM	
0x5009_0000	AuxADC	
0x500A_0000	PDM	
0x500C_0000	TRNG	
0x500D_0000	LCD	
0x500F_0000	QSPI0	

1.4. 内核中断映像

表格 1-2 中断映像表

中断号	中断源
0	BLE_Int
1	DMA_Int
2	Timer0_Int
3	Timer1_Int
4	I2C0_Int
5	I2C1_Int
6	SPI0_Master_Int
7	SPI1_Master_Int
8	SPI_Slave_Int
9	Uart0_Int
10	Uart1_Int
11	PDM_Int
12	TRAN_Int
13	ADC_Int
14	GPIO_Int
15	PMU_Int
16	FC_Int
17	
18	PWM_Int
19	USB_Int
20	

中断号	中断源
21	LCD_Int
22	QSPI0_Int
23	I2S_Int

Freqchip Confidential

2. 系统控制单元（System Controller）

2.1. 概述

时钟、复位以及芯片部分功能的复用配置，通过系统控制寄存器实现。

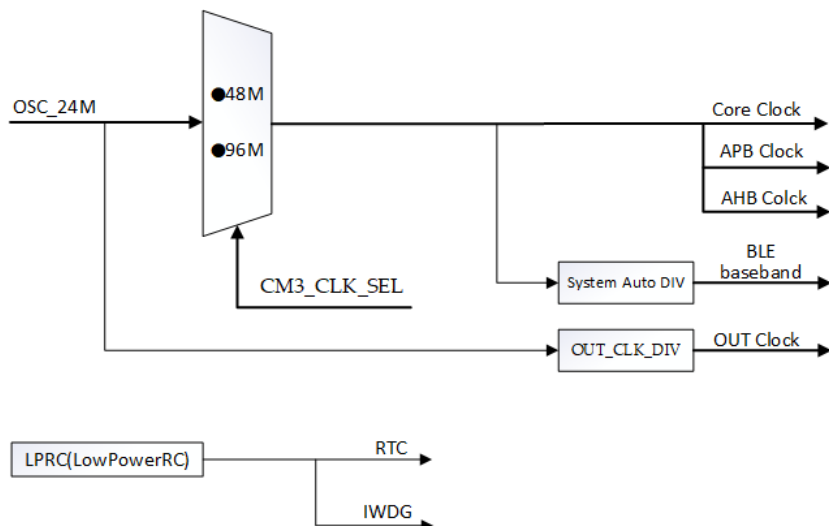


图 2-1 系统时钟树

2.2. 系统时钟

2.2.1. 时钟源

表格 2-1 时钟源

时钟名称	时钟频率	描述
OSC24M_CKDIG48M	48MHz	外部高速晶体经过内部 2 倍频产生的时钟
OSC24M_CKDIG96M	96MHz	外部高速晶体经过内部 4 倍频产生的时钟
RC64M	64MHz	内部 RC 时钟

2.2.2. 系统时钟树

2.2.3. 外设时钟选择

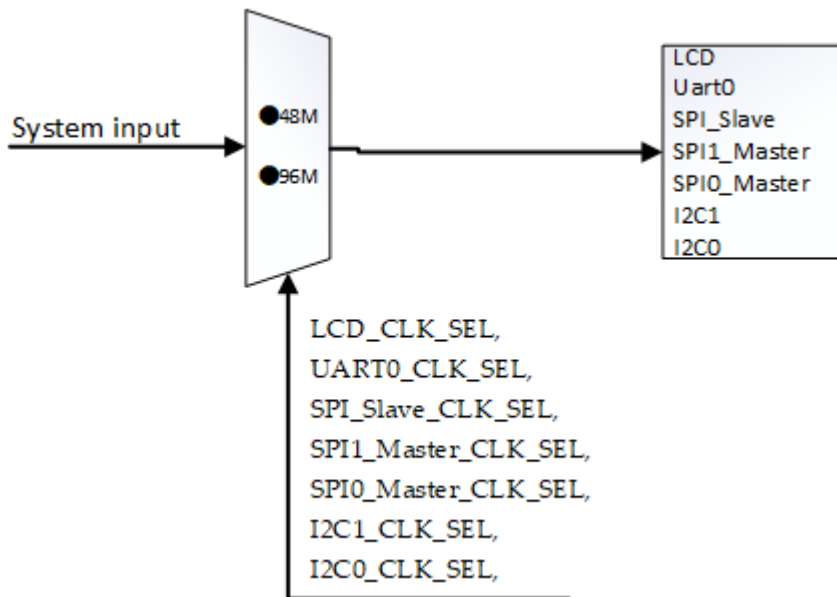


图 2-2 外设时钟配置

2.3. 系统复位

表格 2-2 系统复位源

复位源	描述
芯片上电 POR 复位	CPU 内核复位，所有外设复位，PMU 复位
外部 IO Reset 复位	
IWDG 复位	
软件调试复位	CPU 内核复位，所有外设复位，PMU 不复位

2.4. 系统低功耗

表格 2-3 低功耗模式

模式	模式描述	进入条件	退出条件
SLEEP	CPU 与外设接口模块掉电，部分 Memory 模块处于	系统应用调度空闲，不再需要发送或者接收外设数	睡眠定时器到达计数值唤醒；

	retention 模式；芯片 IO 根据选择配置支持外部唤醒	据，未再蓝牙收发时隙内	外部中断触发唤醒
SHUT-DOWN	CPU、外设接口模块、所有 Memory 均处于掉电模式	不再需要系统运行，不需要快速唤醒响应	外部中断唤醒或者定时器唤醒，触发系统启动

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：0x5000_0000

表格 2-4 系统寄存器映像

地址	寄存器	寄存器描述
0x5000_0000	Clock Control	时钟控制
0x5000_0004	Clock Enable	时钟使能
0x5000_0008	QSPI0 Control	杂项控制
0x5000_000C	Reset Control	复位控制
0x5000_0010	APB Clock Enable	APB 总线外设时钟使能
0x5000_0014	DMA_REQ_MUX0	DMA 请求号配置 0
0x5000_0018	DMA_REQ_MUX1	DMA 请求号配置 1
0x5000_001C	DMA_REQ_MUX2	DMA 请求号配置 2
0x5000_0020	PortA Function MUX	PortA 端口复用配置
0x5000_0024	PortB Function MUX	PortB 端口复用配置
0x5000_0028	PortC Function MUX	PortC 端口复用配置
0x5000_002C	PortD Function MUX	PortD 端口复用配置
0x5000_0030	Portx Pull Enable	Portx 端口上下拉使能
0x5000_0034	Portx Pull Select	Portx 端口上下拉选择
0x5000_0038	PortE_CFG	PortD 端口复用上下拉配置
0x5000_003C	USB_OTG_CTRL	USB PHY 配置

2.5.2. Clock Control (Offset 0x00)

表格 2-5 时钟控制

比特	名称	属性	复位值	描述
32:24	Reserved	R/W	8'h08	Reserved
23:16	OUT_CLK_DIV	R/W	8'h0C	输出时钟分频，只支持偶数分频。 $OUT_CLK = CLK_OSC / 2 * (OUT_CLK_DIV + 1)$
15:12	Reserved	R/W	4'h0	Reserved
11	RC64M_CAL_EN	R/W	1'b0	Reserved
10	LCD_CLK_SEL	R/W	1'b0	LCD 时钟选择。 0: 48MHZ 1: 96MHZ
9	UART0_CLK_SEL	R/W	1'b0	UART0 时钟选择。 0: 48MHZ 1: 96MHZ
8	SPI_Slave_CLK_SEL	R/W	1'b0	SPI Slave 时钟选择。 0: 48MHZ 1: 96MHZ
7	SPI1_Master_CLK_SEL	R/W	1'b0	SPI1 Master 时钟选择。 0: 48MHZ 1: 96MHZ
6	SPI0_Master_CLK_SEL	R/W	1'b0	SPI0 Master 时钟选择。 0: 48MHZ 1: 96MHZ
5	I2C1_CLK_SEL	R/W	1'b0	I2C1 时钟选择 0: 48MHZ 1: 96MHZ
4	I2C0_CLK_SEL	R/W	1'b0	I2C0 时钟选择 0: 48MHZ 1: 96MHZ
3	Reserved	R/W	1'b0	Reserved
2:0	CM3_CLK_SEL	R/W	3'b001	内核时钟选择。 3'b000: 不支持 3'b001: 16MHz 3'b010: 24MHz

				3'b011: 32MHz 3'b100: 48MHz 3'b101: 96MHz
--	--	--	--	---

2.5.3. Clock Enable (Offset 0x04)

表格 2-6 时钟使能

比特	名称	属性	复位值	描述
31	USB12_CLK_EN	R/W	1'b0	USB12M 时钟使能。 0: Disable 1: Enable
30	USB48_CLK_EN	R/W	1'b0	USB48M 时钟使能。
29	USB_HCLK_EN	R/W	1'b0	USB 主时钟使能
28	Reserved	R/W	1'b0	Reserved
27	Reserved	R/W	1'b0	Reserved
26	MDM_RXV1_CLK_EN	R/W	1'b0	Modem Rx VER1 时钟使能
25	MDM_RXV0_CLK_EN	R/W	1'b0	Modem Rx VER0 时钟使能
24	RF_CLK_EN	R/W	1'b0	FR 时钟使能
23	MDM_CLK_EN	R/W	1'b0	Modem 时钟使能
22	PMU_Interface_CLK_EN	R/W	1'b1	PMU 接口时钟使能
21	APB_CLK_EN	R/W	1'b1	APB 总线时钟使能
20	I2S_CLK_EN	R/W	1'b0	I2S 时钟使能
19	DMA_CLK_EN	R/W	1'b0	DMA 时钟使能
18	AHB_CLK_EN	R/W	1'b1	AHB 总线时钟使能
17	RAM_CLK_EN	R/W	1'b1	RAM 时钟使能
16	ROM_CLK_EN	R/W	1'b1	ROM 时钟使能
15	TRNG	R/W	1'b0	随机数使能
14	SPI_Slave_CLK_EN	R/W	1'b0	SPI Slave 时钟使能
13	SPI1_Master_CLK_EN	R/W	1'b0	SPI1 Master 时钟使能
12	SPI0_Master_CLK_EN	R/W	1'b0	SPI0 Master 时钟使能
11	I2C1_CLK_EN	R/W	1'b0	I2C1 时钟使能
10	I2C0_CLK_EN	R/W	1'b0	I2C0 时钟使能
9	OUT_CLK_EN	R/W	1'b0	时钟输出使能

8	PDM_CLK_EN	R/W	1'b0	PDM 数字麦克风时钟使能
7	UART0_CLK_EN	R/W	1'b1	Uart0 时钟使能
6	AUXADC_CLK_EN	R/W	1'b0	AUXADC 时钟使能
5	GPIO_CLK_EN	R/W	1'b0	GPIO 时钟使能
4	EFUSE_Interface_CLK_EN	R/W	1'b0	EFUSE 接口时钟使能
3	BLE_EM_CLK_EN	R/W	1'b1	BLE 内存时钟使能
2	BLE_CLK_EN	R/W	1'b1	BLE 主时钟使能
1	LCD_CLK_EN	R/W	1'b0	LCD 接口时钟使能
0	CM3_CLK_EN	R/W	1'b1	内核时钟使能

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

2.5.4. QSPI0 Control (Offset 0x08)

表格 2-7 QSPI0

比特	名称	属性	复位值	描述
31:28	QSPI0_SCLK_DLY_ADJ	R/W	4'h0	QSPI0 采样点延时。 (因用户外围硬件等影响, 可能导致线上信号达不到最理想的状态, 导致数据采样出现错误。此时调整此值, 用来达到准确的采样值)。
27:24	Reserved	R		厂商配置值, 禁止用户更改。
23	QSPI0_IO_CS_OEN	R/W	1'b1	QSPI0 CS 引脚使能 0: output enable 1: output disable
22	QSPI0_IO_SCLK_OEN	R/W	1'b1	QSPI0 SCLK 引脚使能 0: output enable 1: output disable
21:20	Reserved	R		厂商配置值, 禁止用户更改。
19	QSPI0_HGCLK_EN	R/W	1'b0	QSPI0 主时钟使能 0: CPU 非睡眠状态时时钟使能 1: 时钟总是使能
18	Reserved	R		厂商配置值, 禁止用户更改。
17	QSPI0_REF_GLCK_EN	R/W	1'b0	QSPI0 分频源时钟使能 0: disable

				1: enable
16:12	Reserved	R		厂商配置值，禁止用户更改。
11:10	QSPI0_REF_CLK_SEL	R/W	2'b00	QSPI0 分频源时钟选择 00: 48MHz 01: 96MHz 10: 128MHz 11: 192MHz
9:8	Reserved	R		厂商配置值，禁止用户更改。
7	QSPI0_REAMP	R/W	1'b0	QSPI0 映射 0: QSPI0 引脚映射在 PortC 引脚 1: 禁用
6:0	Reserved	R		厂商配置值，禁止用户更改。

2.5.5. Reset Control (Offset 0x0C)

表格 2-8 复位控制

比特	名称	属性	复位值	描述
31:27	Reserved	R/W	5'b0	Reserved
26	I2S_RST	R/W	1'b0	I2S 软件复位。 写 '1' 复位，硬件自动清零。
25	QSPI0_REF_RST	R/W	1'b0	QSPI0 复位。 写 '1' 复位，硬件自动清零。
24	LCD_PARA_RST	R/W	1'b0	LCD 接口复位。 写 '1' 复位，硬件自动清零。
23	Reserved	R/W	1'b0	Reserved
22	Reserved	R/W	1'b0	Reserved
21	TRNG_RST	R/W	1'b0	随机数发生器复位。 写 '1' 复位，硬件自动清零。
20	PDM_RST	R/W	1'b0	PDM 数字麦克风复位。 写 '1' 复位，硬件自动清零。
19	AUXADC_RST	R/W	1'b0	AUX-ADC 复位。 写 '1' 复位，硬件自动清零。
18	PWM_RST	R/W	1'b0	PWM 复位。

				写 ‘1’ 复位，硬件自动清零。
17	UART1_RST	R/W	1'b0	Uart1 复位。 写 ‘1’ 复位，硬件自动清零。
16	UART0_RST	R/W	1'b0	Uart0 复位。 写 ‘1’ 复位，硬件自动清零。
15	GPIO_RST	R/W	1'b0	GPIO 复位。 写 ‘1’ 复位，硬件自动清零。
14	I2C1_RST	R/W	1'b0	I2C1 复位。 写 ‘1’ 复位，硬件自动清零。
13	I2C0_RST	R/W	1'b0	I2C0 复位。 写 ‘1’ 复位，硬件自动清零。
12	SPI_SLAVE_RST	R/W	1'b0	SPI Slave 复位。 写 ‘1’ 复位，硬件自动清零。
11	SPI1_Master_RST	R/W	1'b0	SPI1_Master 复位。 写 ‘1’ 复位，硬件自动清零。
10	SPI0_Master_RST	R/W	1'b0	SPI0_Master 复位。 写 ‘1’ 复位，硬件自动清零。
9	EFUSE_Interface_RST	R/W	1'b0	EFUSE 接口复位。 写 ‘1’ 复位，硬件自动清零。
8	PMU_Interface_RST	R/W	1'b0	PMU 接口复位。 写 ‘1’ 复位，硬件自动清零。
7:0	Reserved	R/W	1'b0	Reserved

2.5.6. APB Clock Enable (Offset 0x10)

表格 2-9 APB 时钟使能

比特	名称	属性	复位值	描述
31	LCD_APB_CLKEN	R/W	1'b0	LCD 接口 APB 时钟使能 0: disable 1: enable
30	Reserved	R/W	1'b0	Reserved
29	PDM_APB_CLKEN	R/W	1'b0	PDM 数字麦克风 APB 时钟使能 0: disable 1: enable

28	AuxADC_APB_CLKEN	R/W	1'b0	AuxADC APB 时钟使能 0: disable 1: enable
27	PWM_APB_CLKEN	R/W	1'b0	PWM APB 时钟使能 0: disable 1: enable
26	GPIO_APB_CLKEN	R/W	1'b0	GPIO APB 时钟使能 0: disable 1: enable
25	UART1_APB_CLKEN	R/W	1'b0	Uart1 APB 时钟使能 0: disable 1: enable
24	UART0_APB_CLKEN	R/W	1'b1	Uart0 APB 时钟使能 0: disable 1: enable
23	I2C1_APB_CLKEN	R/W	1'b0	I2C1 APB 时钟使能 0: disable 1: enable
22	I2C0_APB_CLKEN	R/W	1'b0	I2C0 APB 时钟使能 0: disable 1: enable
21	SPI_Slave_APB_CLKEN	R/W	1'b0	SPI Slave APB 时钟使能 0: disable 1: enable
20	SPI1_Master_APB_CLKEN	R/W	1'b0	SPI1 Master APB 时钟使能 0: disable 1: enable
19	SPI0_Master_APB_CLKEN	R/W	1'b0	SPI0 Master APB 时钟使能 0: disable 1: enable
18	EFUSE_Interface_APB_CLKEN	R/W	1'b0	Efuse 接口 APB 时钟使能 0: disable 1: enable
17	PMU_Interface_APB_CLKEN	R/W	1'b1	PMU 接口 APB 时钟使能 0: disable

				1: enable
16	TIMER_APB_CLKEN	R/W	1'b0	Timer APB 时钟使能 0: disable 1: enable
15:8	GPIO_CLK_DIV	R/W	8'h08	GPIO 输入检测分频。 GPIO Clock = $APB_bus_clk / 2 * (GPIO_CLK_DIV + 1)$ 使用分频后的时钟检测 GPIO 输入信号，能更快速的检测引脚信号，在 GPIO 输入中断使用时，可进一步提高实时性。
7:5	Reserved	R/W	3'b0	Reserved
4	I2S_APB_CLKEN	R/W	1'b0	I2S APB 时钟使能 0: disable 1: enable
3	Reserved	R/W	1'b0	Reserved
2	Reserved	R/W	1'b0	Reserved
1	MEMORY_SEL	R/W	1'b0	8KB Ram 内存分享。 0: 给系统内存使用 1: 给蓝牙通讯包缓存使用
0	Reserved	R/W	1'b0	Reserved

2.5.7. DMA_REQ_MUX0 (Offset 0x14)

表格 2-10 DM_REQ_MUX0

比特	名称	属性	复位值	描述
31:28	SPI1_Master_TX_DMA_REQ	R/W	4'b0	SPI1_Master_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
27:24	SPI1_Master_RX_DMA_REQ	R/W	4'b0	SPI1_Master_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF
23:20	SPI0_Master_TX_DMA_REQ	R/W	4'b0	SPI0_Master_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
19:16	SPI0_Master_RX_DMA_REQ	R/W	4'b0	SPI0_Master_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF
15:12	UART1_TX_DMA_REQ	R/W	4'b0	Uart1_Tx DMA 请求分配。

				请求号: 0x0 ~ 0xF
11:8	UART1_RX_DMA_REQ	R/W	4'b0	Uart1_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF
7:4	UART0_TX_DMA_REQ	R/W	4'b0	Uart0_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
3:0	UART0_RX_DMA_REQ	R/W	4'b0	Uart0_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF

2.5.8. DMA_REQ_MUX1 (Offset 0x18)

表格 2-11 DMA_REQ_MUX1

比特	名称	属性	复位值	描述
31:28	PDM_Rx_DMA_REQ	R/W	4'b0	PDM 数字麦克风输入 DMA 请求分配 请求号: 0x0 ~ 0xF
27:24	AuxADC_RX_DMA_REQ	R/W	4'b0	Aux ADC RX DMA 请求分配 请求号: 0x0 ~ 0xF
23:20	SPI_Slave_TX_DMA_REQ	R/W	4'b0	SPI1_Slave_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
19:16	SPI_Slave_RX_DMA_REQ	R/W	4'b0	SPI1_Slave_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF
15:12	I2C1_TX_DMA_REQ	R/W	4'b0	I2C1_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
11:8	I2C1_RX_DMA_REQ	R/W	4'b0	I2C1_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF
7:4	I2C0_TX_DMA_REQ	R/W	4'b0	I2C0_Tx DMA 请求分配。 请求号: 0x0 ~ 0xF
3:0	I2C0_RX_DMA_REQ	R/W	4'b0	I2C0_Rx DMA 请求分配。 请求号: 0x0 ~ 0xF

2.5.9. DMA_REQ_MUX2 (Offset 0x1C)

表格 2-12 DMA_REQ_MUX2

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	4'b0	Reserved
15:12	I2S_TX_DMA_REQ	R/W	4'b0	I2S_TX DMA 请求分配

				请求号: 0x0 ~ 0xF
11:8	I2S_RX_DMA_REQ	R/W	4'b0	I2S_RX DMA 请求分配 请求号: 0x0 ~ 0xF
7:4	SPI1_Master_CMD_DMA_REQ	R/W	4'b0	SPI1_Master_CMD DMA 请求分配 请求号: 0x0 ~ 0xF
3:0	SPI0_Master_CMD_DMA_REQ	R/W	4'b0	SPI0_Master_CMD DMA 请求分配 请求号: 0x0 ~ 0xF

2.5.10. PortA Function MUX (Offset 0x20)

表格 2-13 PortA Function MUX

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](#)。

2.5.11. PortB Function MUX (Offset 0x24)

表格 2-14 PortB Function MUX

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](#)。

2.5.12. PortC Function MUX (Offset 0x28)

表格 2-15 PortC Function MUX

比特	名称	属性	复位值	描述
31:28	PC7_MUX	R/W	4'h8	默认调试引脚 SWDIO
27:24	PC6_MUX	R/W	4'h8	默认调试引脚 SWCLK
23:20	PC5_MUX	R/W	4'h0	PC0 复用功能配置。配置值：0x0 ~ 0xF
19:16	PC4_MUX	R/W	4'h0	PC0 复用功能配置。配置值：0x0 ~ 0xF
15:12	PC3_MUX	R/W	4'h0	PC0 复用功能配置。配置值：0x0 ~ 0xF
11:8	PC2_MUX	R/W	4'h0	PC0 复用功能配置。配置值：0x0 ~ 0xF
7:4	PC1_MUX	R/W	4'h0	PC0 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](#)。

2.5.13. PortD Function MUX (Offset 0x2C)

表格 2-16 PortD Function MUX

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](#)。

2.5.14. Portx Pull Enable (Offset 0x30)

表格 2-17 Portx Pull Enable

比特	名称	属性	复位值	描述
31:24	PD_PE	R/W	8'hFF	Port D 上下拉使能。1bit 对应 1 引脚。

				0: disable 1: enable
23:16	PC_PE	R/W	8'hFF	Port C 上下拉使能。1bit 对应 1 引脚。 0: disable 1: enable
15:8	PB_PE	R/W	8'hFF	Port B 上下拉使能。1bit 对应 1 引脚。 0: disable 1: enable
7:0	PA_PE	R/W	8'hFF	Port A 上下拉使能。1bit 对应 1 引脚。 0: disable 1: enable

2.5.15. Portx Pull Select (Offset 0x34)

表格 2-18 Portx Pull Select 寄存器

比特	名称	属性	复位值	描述
31:24	PD_PS	R/W	8'h00	Port D 上下拉选择。1bit 对应 1 引脚。 0: pull down 1: pull up
23:16	PC_PS	R/W	8'h00	Port C 上下拉选择。1bit 对应 1 引脚。 0: pull down 1: pull up
15:8	PB_PS	R/W	8'h00	Port B 上下拉选择。1bit 对应 1 引脚。 0: pull down 1: pull up
7:0	PA_PS	R/W	8'h00	Port A 上下拉选择。1bit 对应 1 引脚。 0: pull down 1: pull up

2.5.16. PortE_CFG (Offset 0x38)

表格 2-19 PortE CFG 寄存器

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

31:28	PE1_MUX	R/W	4'h00	PE1 复用功能配置。配置值：0x0 ~ 0xF
27:24	PE0_MUX	R/W	4'h00	PE0 复用功能配置。配置值：0x0 ~ 0xF
23:22	PE_PS	R/W	2'b00	Port E 上下拉选择。1bit 对应 1 引脚。 0: pull down 1: pull up
21:20	PE_PE	R/W	2'b00	Port E 上下拉使能。1bit 对应 1 引脚。 0: disable 1: enable
19:14	Reserved	R	6'h00	Reserved
13:8	QSPI_PS	R/W	6'h3F	QSPI 上下拉选择。
7:6	Reserved	R	2'h00	Reserved
5:0	QSPI_PE	R/W	6'h3F	QSPI 上下拉使能。

2.5.17. USB_OTG_CTRL (Offset 0x3C)

表格 2-20 USB_OTG_CTRL 寄存器

比特	名称	属性	复位值	描述
31:21	Reserved	R/W	11'h0	Reserved
20	USB_UART_SEL	R/W	1'b0	Reserved
19	USB_IO_PE1	R/W	1'b1	USB_DM 引脚上下拉使能 0: disable 1: enable
18	USB_IO_PE0	R/W	1'b1	USB_DP 引脚上下拉使能 0: disable 1: enable
17	USB_IO_PS1	R/W	1'b0	USB_DM 引脚上下拉选择 0: pull down 1: pull up
16	USB_IO_PS0	R/W	1'b1	USB_DP 引脚上下拉选择 0: pull down 1: pull up
15	USB_STDBY_REG	R/W	1'b0	USB 进入 standby 状态 0: standby diaable

				1: standby enable
14	USB_STDBY_SEL	R/W	1'b0	USB standby 控制选择 0: USB 控制器控制 1: 软件控制通过 bit[15]
13	USB_UART_EN	R/W	1'b0	Reserved
12	USB_PHY_SEL	R/W	1'b0	USB PHY 选择 0: 使用专用 PHY 1: 使用 GPIO 代替
11	OTG_CID_I	R/W	1'b0	CID 配置
10	OTG_VBUSLO_I	R/W	1'b0	VBUSLO 配置
9	OTG_VBUS_SES_I	R/W	1'b0	VBUS_SES 配置
8	OTG_VBUS_VAL_I	R/W	1'b0	VBUS_VAL 配置
7:0	USB_PHY_DAP_CFG	R/W	8'h09	USB PHY adapter 配置

3. 通用输入输出接口（GPIO）

3.1. 概述

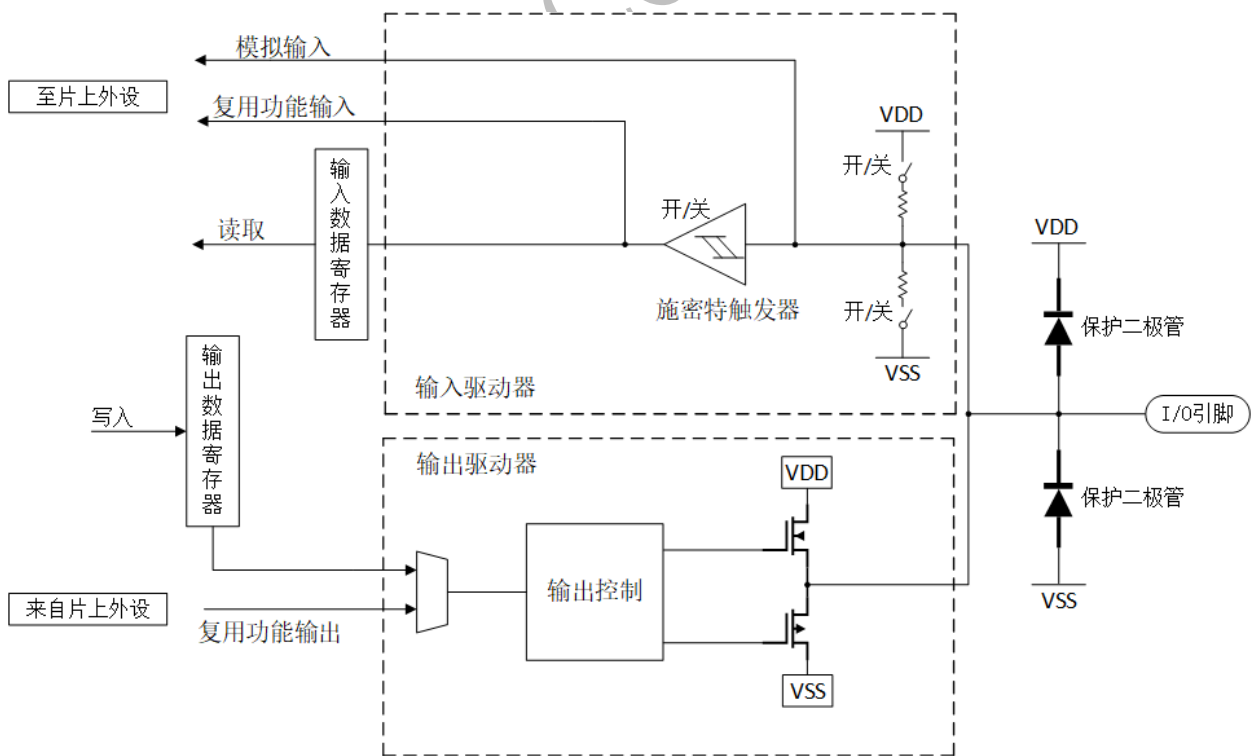
集成了通用数字 IO 的输入输出功能，同时在输入模式下，还支持外部中断检测处理（支持硬件防抖）。

GPIO 共有 5 组控制器，PortA、PortB、PortC、PortD、PortE。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

- 5 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

GPIO 寄存器基地址：0x5006_0000

表格 3-1 GPIO 寄存器地址映像

偏移	寄存器	寄存器描述
0x00	PortA_DATA	GPIO 端口 A 数据寄存器
0x04	PortB_DATA	GPIO 端口 B 数据寄存器
0x08	PortC_DATA	GPIO 端口 C 数据寄存器
0x0C	PortD_DATA	GPIO 端口 D 数据寄存器
0x10	PortE_DATA	GPIO 端口 E 数据寄存器
0x20	PortA_OEN	GPIO 端口 A 输出使能寄存器
0x24	PortB_OEN	GPIO 端口 B 输出使能寄存器
0x28	PortC_OEN	GPIO 端口 C 输出使能寄存器
0x2C	PortD_OEN	GPIO 端口 D 输出使能寄存器
0x30	PortE_OEN	GPIO 端口 E 输出使能寄存器
0x40	Port_EXTI_EN0	PortA~PortD EXTI 中断使能
0x44	Port_EXTI_EN1	PortE EXTI 中断使能

3.4.2. PortA_DATA (Offset 0x00)

表格 3-2 PortA DATA 寄存器

比特	名称	属性	复位值	描述
7:0	PA_DATA	R/W	8'h00	端口 A 输入输出数据寄存器 1bit 对应 1 引脚。写 '1' 输出高，写 '0' 输出低，读为当前引脚状态

3.4.3. PortB_DATA (Offset 0x04)

表格 3-3 PortB DATA 寄存器

比特	名称	属性	复位值	描述
7:0	PB_DATA	R/W	8'h00	端口 B 输入输出数据寄存器

比特	名称	属性	复位值	描述
				1bit 对应 1 引脚。写 ‘1’ 输出高，写 ‘0’ 输出低，读为当前引脚状态

3.4.4. PortC_DATA (Offset 0x08)

表格 3-4 PortC DATA 寄存器

比特	名称	属性	复位值	描述
7:0	PC_DATA	R/W	8'h00	端口 C 输入输出数据寄存器 1bit 对应 1 引脚。写 ‘1’ 输出高，写 ‘0’ 输出低，读为当前引脚状态

3.4.5. PortD_DATA (Offset 0x0C)

表格 3-5 PortB DATA 寄存器

比特	名称	属性	复位值	描述
7:0	PD_DATA	R/W	8'h00	端口 D 输入输出数据寄存器 1bit 对应 1 引脚。写 ‘1’ 输出高，写 ‘0’ 输出低，读为当前引脚状态

3.4.6. PortE_DATA (Offset 0x10)

表格 3-6 PortE DATA 寄存器

比特	名称	属性	复位值	描述
7:0	PE_DATA	R/W	8'h00	端口 E 输入输出数据寄存器 1bit 对应 1 引脚。写 ‘1’ 输出高，写 ‘0’ 输出低，读为当前引脚状态

3.4.7. PortA_OEN (Offset 0x20)

表格 3-7 PortA OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PA_OEN	R/W	8'h00	PortA 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。
--	--	--	--	----------------------

3.4.8. PortB_OEN (Offset 0x24)

表格 3-8 PortB OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PB_OEN	R/W	8'h00	PortB 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。

3.4.9. PortC_OEN (Offset 0x28)

表格 3-9 PortC OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PC_OEN	R/W	8'h00	PortC 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。

3.4.10. PortD_OEN (Offset 0x2C)

表格 3-10 PortD OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PD_OEN	R/W	8'h00	PortD 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。

3.4.11. PortE_OEN (Offset 0x30)

表格 3-11 PortE OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PE_OEN	R/W	8'h00	PortE 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。

3.4.12. Port_EXTI_EN0 (Offset 0x40)

表格 3-12 Port_EXTI_EN0 寄存器

比特	名称	属性	复位值	描述
31:24	PortD_EXTI_EN	R/W	8'h00	PortD EXTI 模式使能。 1bit 对应 1 引脚。 0: Disable。 1: Enable。
23:16	PortC_EXTI_EN	R/W	8'h00	PortC EXTI 模式使能。 1bit 对应 1 引脚。 0: Disable。 1: Enable。
15:8	PortB_EXTI_EN	R/W	8'h00	PortB EXTI 模式使能。 1bit 对应 1 引脚。 0: Disable。 1: Enable。
7:0	PortA_EXTI_EN	R/W	8'h00	PortA EXTI 模式使能。 1bit 对应 1 引脚。 0: Disable。 1: Enable。

3.4.13. Port_EXTI_EN1 (Offset 0x44)

表格 3-13 Port_EXTI_EN1 寄存器

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

31:4	Reserved	R/W	28'h00	Reserved
3:0	PortE_EXTI_EN	R/W	4'h0	PortE EXTI 模式使能。 1bit 对应 1 引脚。 0: Disable。 1: Enable。

Frequchip Confidential

4. 外部中断控制器（EXTI）

4.1. 概述

EXTI 支持 36 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

- 多达 36 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

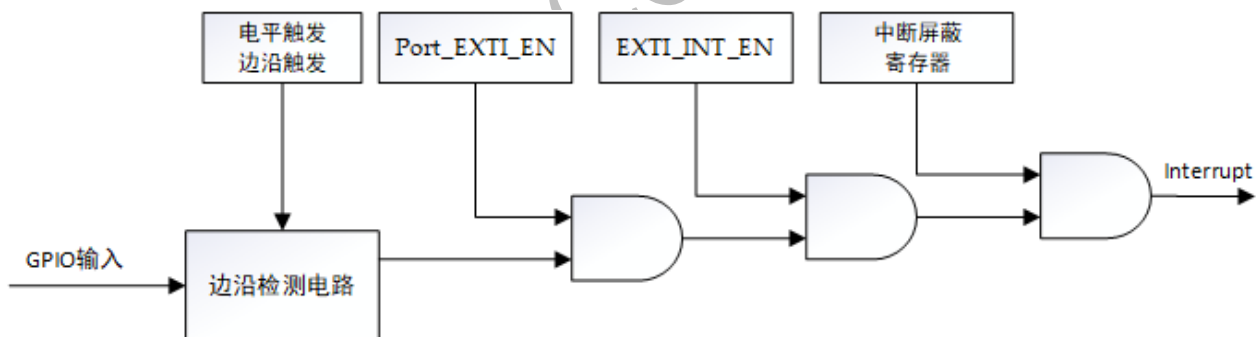


图 4-1 EXTI 系统框图

4.4. EXTI 寄存器

4.4.1. EXTI 寄存器映像

EXTI 寄存器基地址：0x5006_8000

表格 4-1 EXTI 寄存器映像

偏移	寄存器	寄存器描述
0x00	EXTI_CTL0	外部中断线使能寄存器 0

0x04	EXTI_CTL1	外部中断线使能寄存器 1
0x08	EXTI_STS0	中断状态寄存器 0
0x0C	EXTI_STS1	中断状态寄存器 1
0x10	EXTI_TYPE0	触发类型选择 0
0x14	EXTI_TYPE1	触发类型选择 1
0x18	EXTI_TYPE2	触发类型选择 2
0x20 ~ 0x3C	EXTI_CNT0 ~ EXTI_CNT7	中断线 0 ~ 7 防抖计数值
0x40 ~ 0x5C	EXTI_CNT8 ~ EXTI_CNT15	中断线 8 ~ 15 防抖计数值
0x60 ~ 0x7C	EXTI_CNT16 ~ EXTI_CNT23	中断线 16 ~ 23 防抖计数值
0x80 ~ 0x9C	EXTI_CNT24 ~ EXTI_CNT31	中断线 24 ~ 31 防抖计数值
0xA0 ~ 0xAC	EXTI_CNT32 ~ EXTI_CNT35	中断线 32 ~ 35 防抖计数值

4.4.2. EXTI_CTL0 (Offset 0x00)

表格 4-2 EXTI_CTL0 寄存器

比特	名称	属性	复位值	描述
31:0	EXTI_INT0_EN (EXTI 请求线 0 ~ 31)	R/W	32'h00	EXTI (0~31) 中断使能。 EXTI_INT0_EN[7:0] 对应引脚 PA7 ~ PA0。 EXTI_INT0_EN[15:8] 对应引脚 PB7 ~ PB0。 EXTI_INT0_EN[23:16] 对应引脚 PC7 ~ PC0。 EXTI_INT0_EN[31:24] 对应引脚 PD7 ~ PD0。 1bit 对应 1 条请求线。 0: Disable 1: Enable

4.4.3. EXTI_CTL1 (Offset 0x04)

表格 4-3 EXTI_CTL1 寄存器

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	28'h00	Reserved
3:0	EXTI_INT1_EN (EXTI 请求线 32 ~ 35)	R/W	4'h0	EXTI (32~35) 中断使能。 EXTI_INT1_EN[3:0] 对应引脚 PE3 ~ PE0。 1bit 对应 1 条请求线。

				0: Disable 1: Enable
--	--	--	--	-------------------------

4.4.4. EXTI_STSS0 (Offset 0x08)

表格 4-4 EXTI_STSS0 寄存器

比特	名称	属性	复位值	描述
31:0	EXTI_STSS0	R/W	32'h00	EXTI (0~31) 中断状态。 EXTI_STSS0 [7:0] 对应引脚 PA7 ~ PA0。 EXTI_STSS0 [15:8] 对应引脚 PB7 ~ PB0。 EXTI_STSS0 [23:16] 对应引脚 PC7 ~ PC0。 EXTI_STSS0 [31:24] 对应引脚 PD7 ~ PD0。 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。

4.4.5. EXTI_STSS1 (Offset 0x0C)

表格 4-5 EXTI_STSS1 寄存器

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	24'h00	Reserved
3:0	EXTI_STSS1	R/W	4'h0	EXTI (32~35) 中断状态。 EXTI_STSS1 [3:0] 对应引脚 PE3 ~ PE0。 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。

4.4.6. EXTI_TYPE0 (Offset 0x10)

表格 4-6 EXTI_TYPE0 寄存器

比特	名称	属性	复位值	描述
31:16	EXTI_PortB_TYPE	R/W	16'h00	PortB 管脚 EXTI 触发类型。 2bit 对应 1 引脚。 EXTI_PortB_TYPE[1:0]对应 PB0

				EXTI_PortB_TYPE[3:2]对应 PB1 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。
15:0	EXTI_PortA_TYPE	R/W	16'h00	PortA 管脚 EXTI 触发类型。 2bit 对应 1 引脚。 EXTI_PortA_TYPE[1:0]对应 PA0 EXTI_PortA_TYPE[3:2]对应 PA1 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。

4.4.7. EXTI_TYPE1 (Offset 0x14)

表格 4-7 EXTI_TYPE1 寄存器

比特	名称	属性	复位值	描述
31:16	EXTI_PortD_TYPE	R/W	16'h00	PortD 管脚 EXTI 触发类型。 2bit 对应 1 引脚。 EXTI_PortD_TYPE[1:0]对应 PD0 EXTI_PortD_TYPE[3:2]对应 PD1 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。
15:0	EXTI_PortC_TYPE	R/W	16'h00	PortC 管脚 EXTI 触发类型。 2bit 对应 1 引脚。

比特	名称	属性	复位值	描述
				EXTI_PortC_TYPE[1:0]对应 PC0 EXTI_PortC_TYPE[3:2]对应 PC1 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。

4.4.8. EXTI_TYPE2 (Offset 0x18)

表格 4-8 EXTI_TYPE2 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	EXTI_PortE_TYPE	R/W	8'h0	PortE 管脚 EXTI 触发类型。 2bit 对应 1 引脚。 EXTI_PortE_TYPE[1:0]对应 PE0 EXTI_PortE_TYPE[3:2]对应 PE1 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。

4.4.9. EXTI_CNT0 ~ EXTI_CNT35 (Offset 0x20 ~ 0xAC)

表格 4-9 EXTI_CNT0~EXTI_CNT35 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:8	PRE_VAL	R/W	8'h0	EXTI (0 ~ 35) 时钟预分频值。分频公式： $CLK = APB_CLK / (1 + PRE_VAL)$ 。

7:4	Reserved	R/W	4'h00	Reserved
3:0	DES_CNT	R/W	4'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

Frequchip Confidential

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

- Uart0 支持深度 32，宽度 8bit 的 FIFO
- Uart1 支持深度 16，宽度 8bit 的 FIFO
- Uart0 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。其中 Uart0 的时钟源可选 48MHz 或 96MHz，Uart1 只使用 APB_CLK。

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）

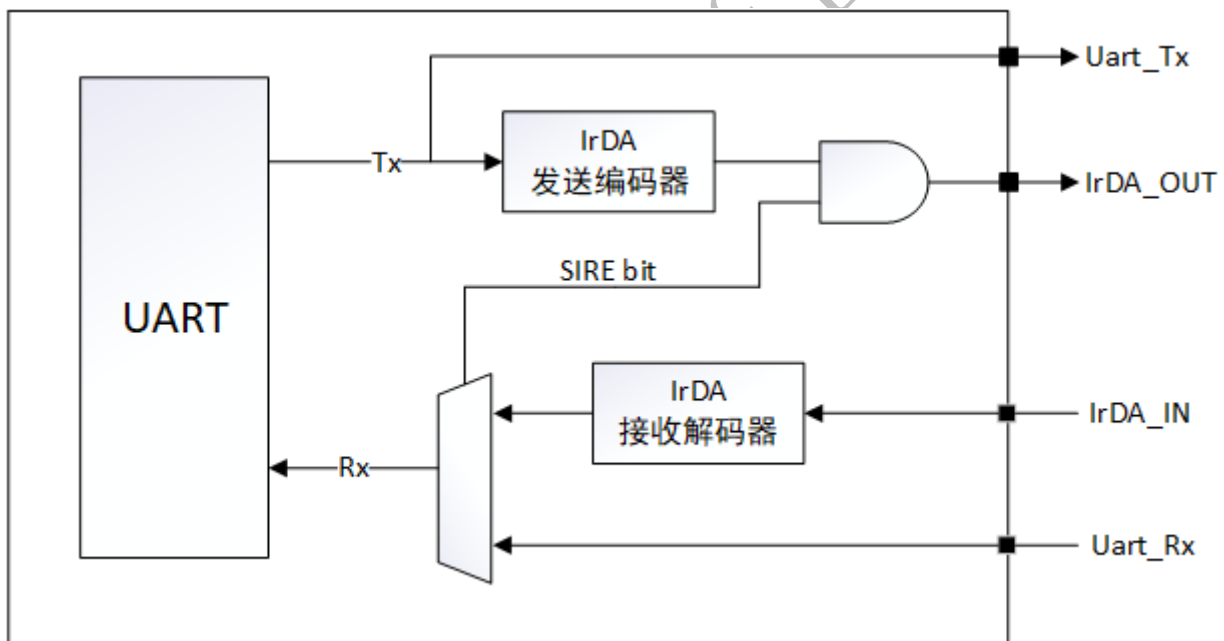
5.3.2. IrDA SIR 1.0 协议

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

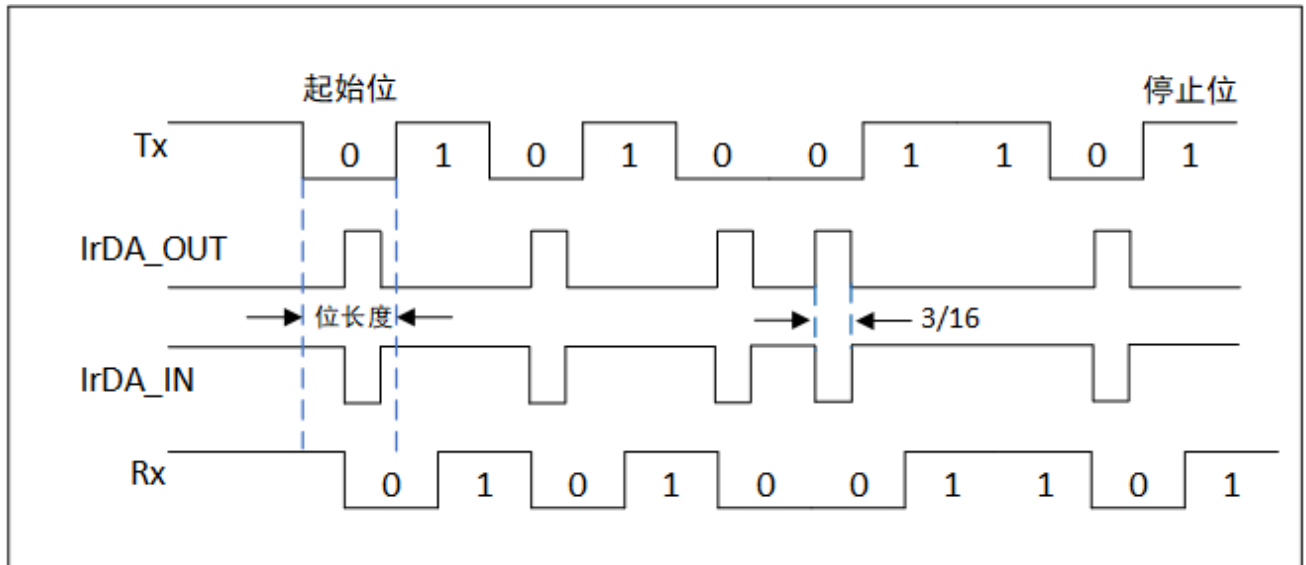


图 5-2 IrDA 时序示意图

5.4. Uart 寄存器

5.4.1. Uart 寄存器映像

Uart0 寄存器基地址: 0x5005_0000

Uart1 寄存器基地址: 0x5005_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)

表格 5-2 DATA 寄存器

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	数据接收发送。 读操作： 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。 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。 写操作： 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。 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。

表格 5-3 DLL 寄存器

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)

表格 5-4 IER 寄存器

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

比特	名称	属性	复位值	描述
				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，TxFIFO 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

表格 5-5 DLH 寄存器

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

比特	名称	属性	复位值	描述
				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’操作时将使用下列寄存器描述

表格 5-6 FCR 寄存器

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 RxFIFO，写 ‘0’ 无效。
0	FIFOE	W	1'b0	FIFO 使能。 0: FIFO Disable 1: FIFO Enable

‘读’操作时将使用下列寄存器描述

表格 5-7 IID 寄存器

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）

表格 5-8 LCR 寄存器

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

比特	名称	属性	复位值	描述
				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)

表格 5-9 MCR 寄存器

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。 (仅 Uart0 支持)
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。 (仅 Uart0 支持)
4:2	Reserved	R/W	0x0	Reserved
1	RTS	R/W	1'b0	RTS (Request to send) 信号控制。 不使用自动流控时为手动控制。 0: RTS Inactive 1: RTS Active

比特	名称	属性	复位值	描述
				注：使用自动流控时，若此位置'0' RTS 信号将无效。
0	Reserved	R/W	0x0	Reserved

5.4.7. LSR (Offset 0x14)

表格 5-10 LSR 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7	RFE	R	1'b0	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 0: RxFIFO 正常 1: RxFIFO 中存在错误
6	TEMT	R	1'b1	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。 0: 发送器非空 1: 发送器空
5	THRE	R	1'b1	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。
4	BI	R	1'b0	Break 错误。 接收数据时检测到连续的低电平信号产生 Break 信号。 读 LSR 寄存器清除 OE 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。

比特	名称	属性	复位值	描述
				读 LSR 寄存器清除 OE 位。
2	PE	R	1'b0	校验错误。 奇偶校验，0/1 校验错误。 读 LSR 寄存器清除 OE 位。
1	OE	R	1'b0	溢出错误。 如果在读取前一个数据之前接收了一个新的数据字符，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 Rx FIFO 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)

表格 5-11 MSR 寄存器

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)

表格 5-12 USR 寄存器

比特	名称	属性	复位值	描述
31:5	Reserved	R/W	0x0	Reserved
4	RFF	R	1'b0	Rx FIFO 满。 0: Rx FIFO 非满 1: Rx FIFO 满
3	RFNE	R	1'b0	Rx FIFO 非空。

比特	名称	属性	复位值	描述
				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)

表格 5-13 DLF 寄存器

比特	名称	属性	复位值	描述
31:4	Reserved	R/W	0x0	Reserved
3:0	DLF	R/W	4'h0	在计算波特率分频中加入小数。 0x0: $0/16 = 0.0000$ 0x1: $1/16 = 0.0625$ 0x2: $2/16 = 0.125$ 0x3: $3/16 = 0.1875$ 0x4: $4/16 = 0.25$ 0x5: $5/16 = 0.3125$ 0x6: $6/16 = 0.375$ 0x7: $7/16 = 0.4375$ 0x8: $8/16 = 0.5$ 0x9: $9/16 = 0.5625$ 0xA: $10/16 = 0.625$ 0xB: $11/16 = 0.6875$ 0xC: $12/16 = 0.75$ 0xD: $13/16 = 0.8125$ 0xE: $14/16 = 0.875$ 0xF: $15/16 = 0.9375$ 使用详情参照波特率计算章节

5.5. 使用流程

5.5.1. 发送数据流程

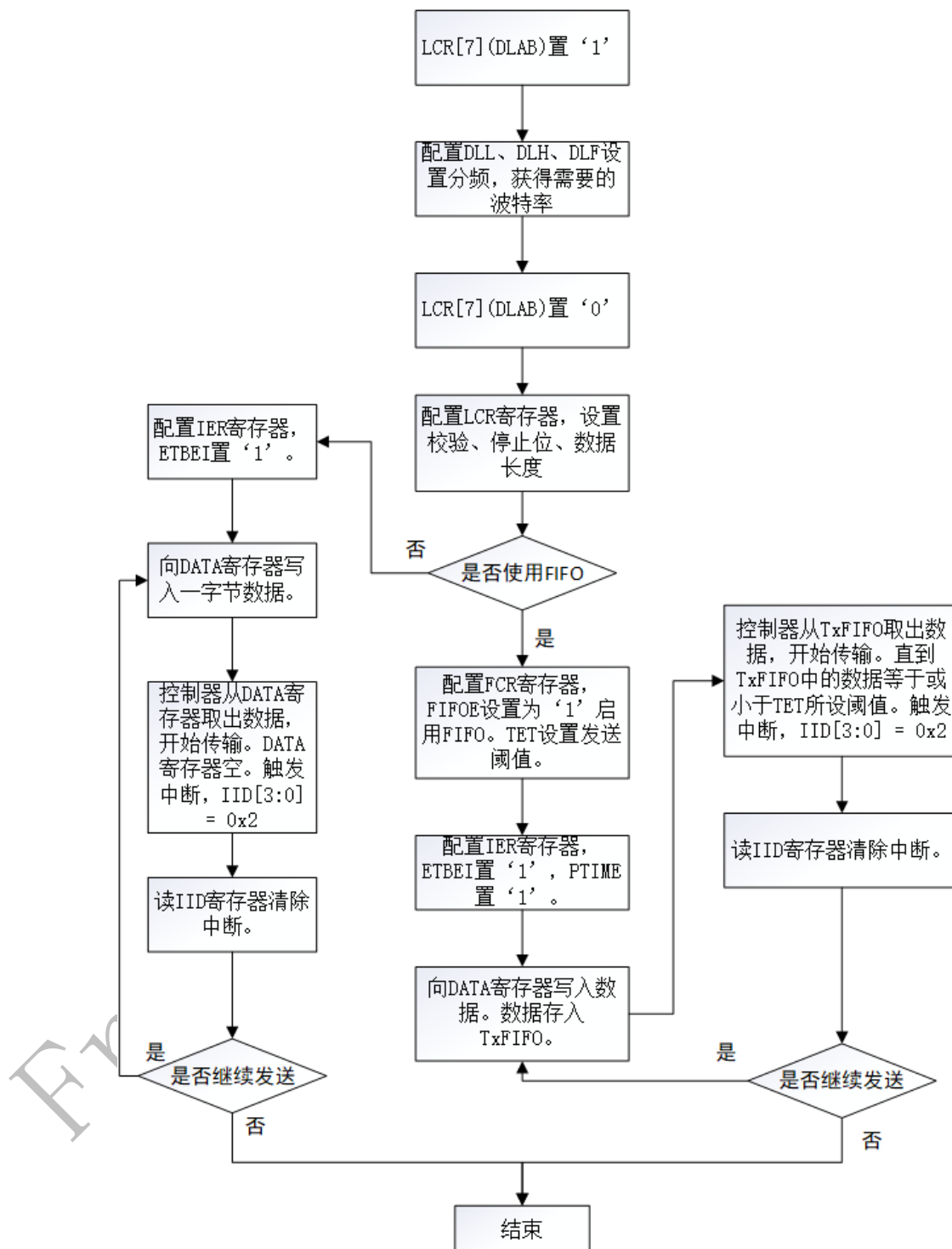


图 5-3 发送数据流程

5.5.2. 接收数据流程

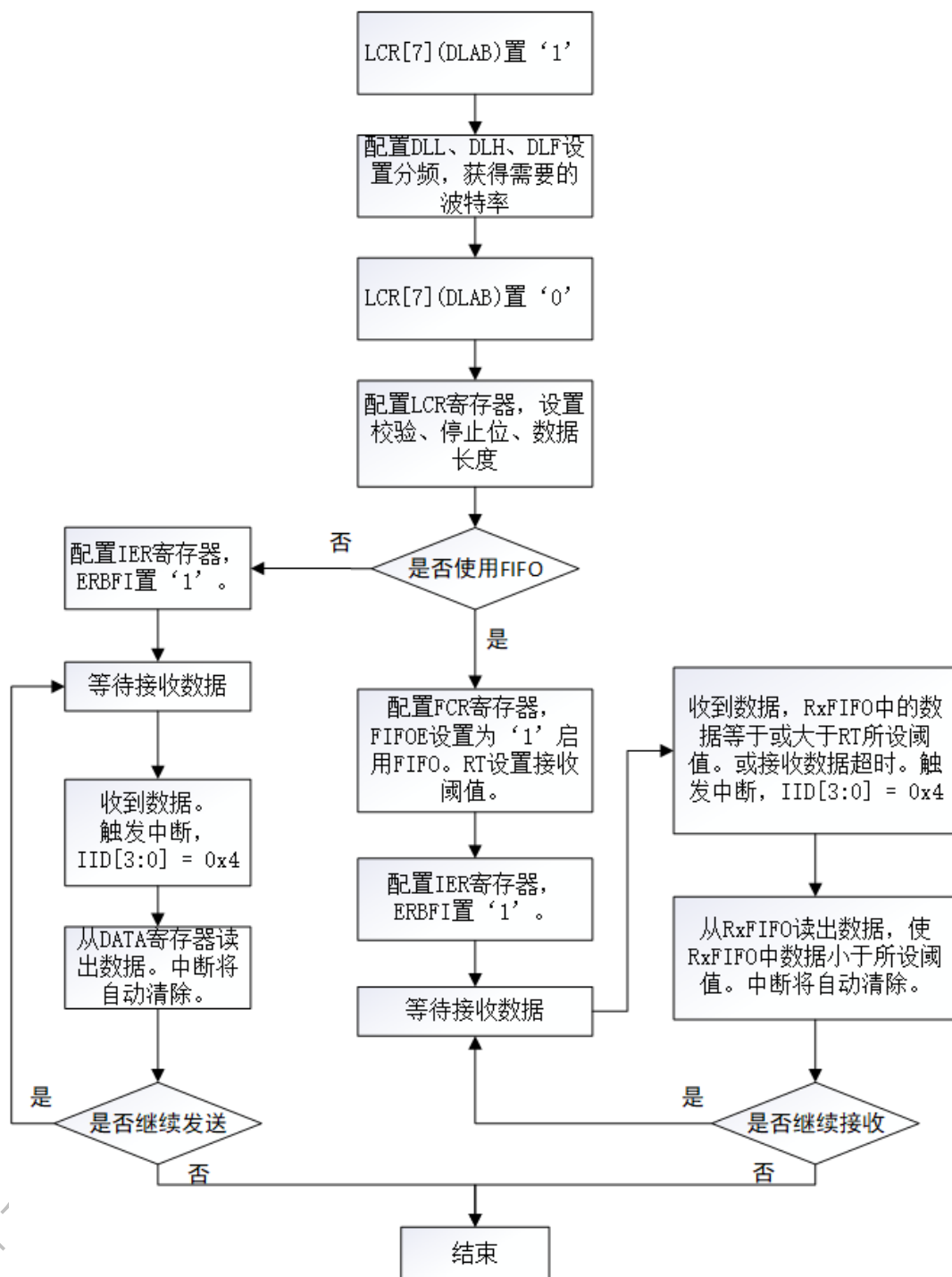


图 5-4 接收数据流程

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（SPIM）提供了一个符合行业标准的 4 线主设备 SPI 接口。支持 Motorola, TI, National Semiconductor Microwire 接口，允许与任何 SPI 从设备通信。除了标准的 8 位字长之外，SPI 主设备还支持可配置 4 至 16 位字长，用于与非标准的 SPI 字长进行通信。

SPIM 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）。

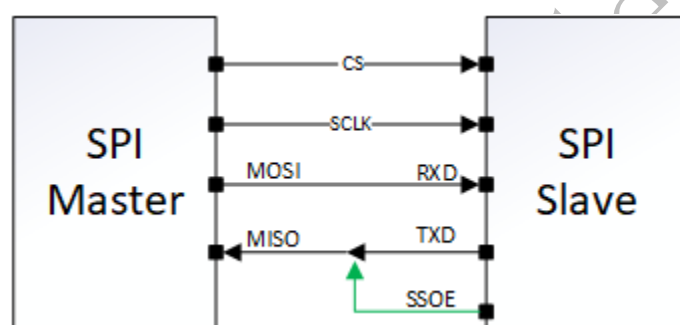


图 6-1 SPI Master 接线示意图

6.2. 主要特性

- 4 至 16 位数据宽度
- 比特率高达 48Mbps
- 支持 DMA
- 支持 Standard（一线）、Dual（二线）、Quad（四线）、接入方式
- 提供深度 32，位宽 16bit 的 FIFO

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。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 8 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 实体中的高八位需要忽略。

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 **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 中应该只有足够的数据帧来为 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

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

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：0x5003_4000

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

偏移	寄存器	寄存器描述
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（仅二线、四线模式下有效）

6.4.2. CTRLR0（Offset 0x00）

表格 6-2 CTRLR0 寄存器

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：保留

比特	名称	属性	复位值	描述
20:16	Reserved	R	0x0	Reserved
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 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
11	SRL	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
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

比特	名称	属性	复位值	描述
				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	DFS	R/W	4'h7	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

6.4.3. CTRLR1 (Offset 0x04)

表格 6-3 CTRLR1 寄存器

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)

表格 6-4 SSIENR 寄存器

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)

表格 6-5 MWCR 寄存器

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)

表格 6-6 SER 寄存器

比特	名称	属性	复位值	描述
31:1	Reserved	R	0x0	Reserved

比特	名称	属性	复位值	描述
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)

表格 6-7 BAUDR 寄存器

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

6.4.8. TXFTLR (Offset 0x18)

表格 6-8 TXFTLR 寄存器

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)

表格 6-9 RXFTLR 寄存器

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)

表格 6-10 TXFLR 寄存器

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)

表格 6-11 RXFLR 寄存器

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)

表格 6-12 SR 寄存器

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)

表格 6-13 IMR 寄存器

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)

表格 6-14 ISR 寄存器

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)

表格 6-15 RISR 寄存器

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)

表格 6-16 TXOICR 寄存器

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)

表格 6-17 RXOICR 寄存器

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)

表格 6-18 RXUICR 寄存器

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)

表格 6-19 ICR 寄存器

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)

表格 6-20 DMACR 寄存器

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)

表格 6-21 DMATDLR 寄存器

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)

表格 6-22 DMARDLR 寄存器

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)

表格 6-23 DR 寄存器

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)

表格 6-24 RX_SAMPLE_DLY 寄存器

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)

表格 6-25 CTRLR2 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
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: 8bie 指令 0x3: 16bit 指令

比特	名称	属性	复位值	描述
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 Dual（二线）、Quad（四线）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 选择数据位宽，SPI_FRF 选择一线二线四线，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 数据未满足或者为空时，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 选择数据位宽，SPI_FRF 选择一线二线四线，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）或 Quad（四线）写

当使用多线方式通讯时（SPI_FRF = 1 或 SPI_FRF = 2）。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：选择二线或四线
- 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。

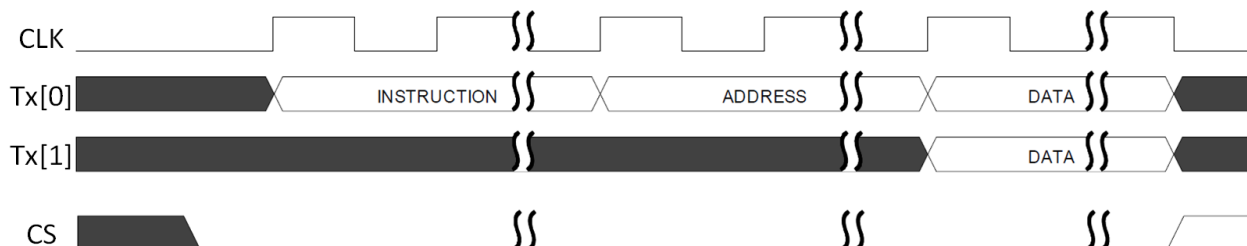


图 6-2 指令和地址使用标准模式、数据使用多线模式

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。

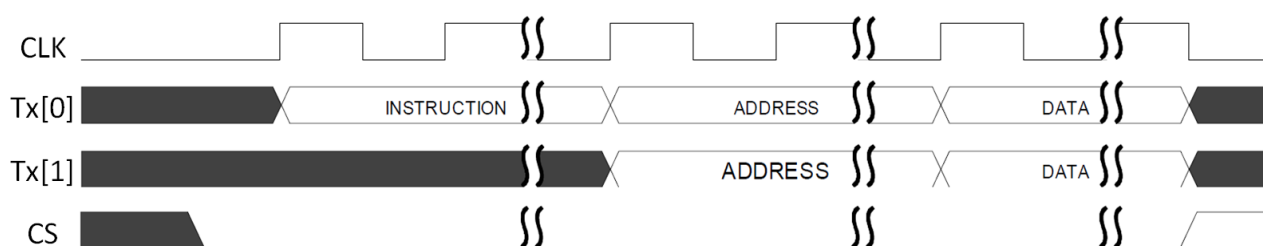


图 6-3 指令使用标准模式、地址和数据使用多线模式。

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

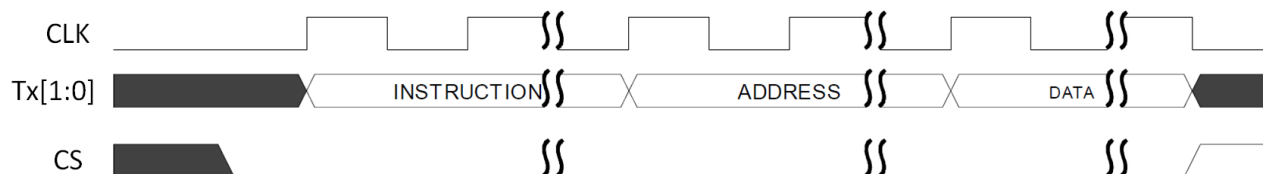


图 6-4 指令、地址和数据使用多线模式

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

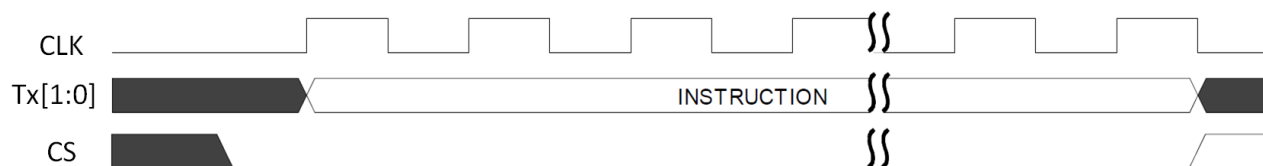


图 6-5 只有指令使用多线模式

6.5.4. Dual（二线）或 Quad（四线）读

当使用多线方式通讯时（SPI_FRF = 1 或 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：选择二线或四线
- 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 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 6-6 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。

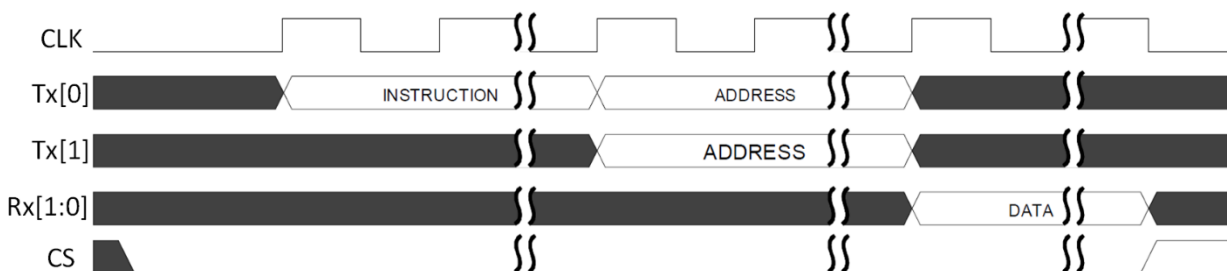


图 6-7 读数据、指令使用标准模式、地址和数据使用多线模式

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

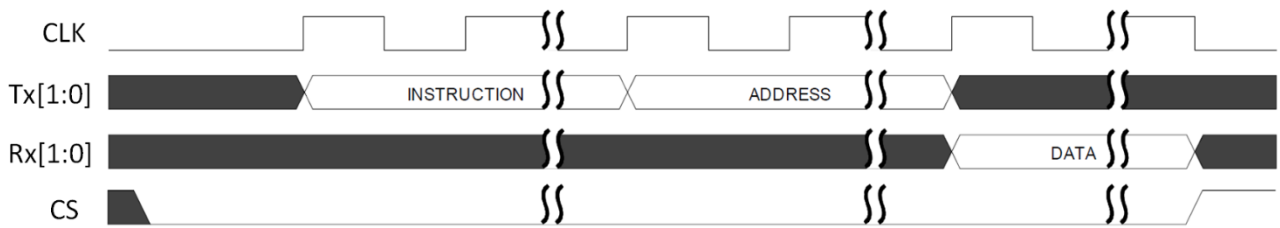


图 6-8 读数据，指令、地址和数据使用多线模式

Case 4: 没有指令，没有地址，仅数据传输。

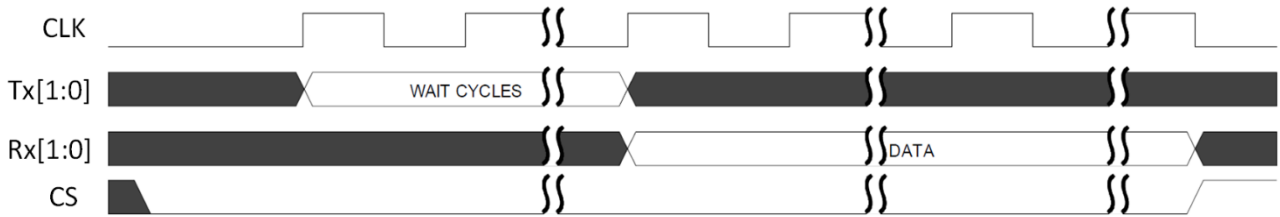


图 6-9 读数据，没有指令，没有地址，仅数据传输

6.6. MOTOROLA SPI 时序

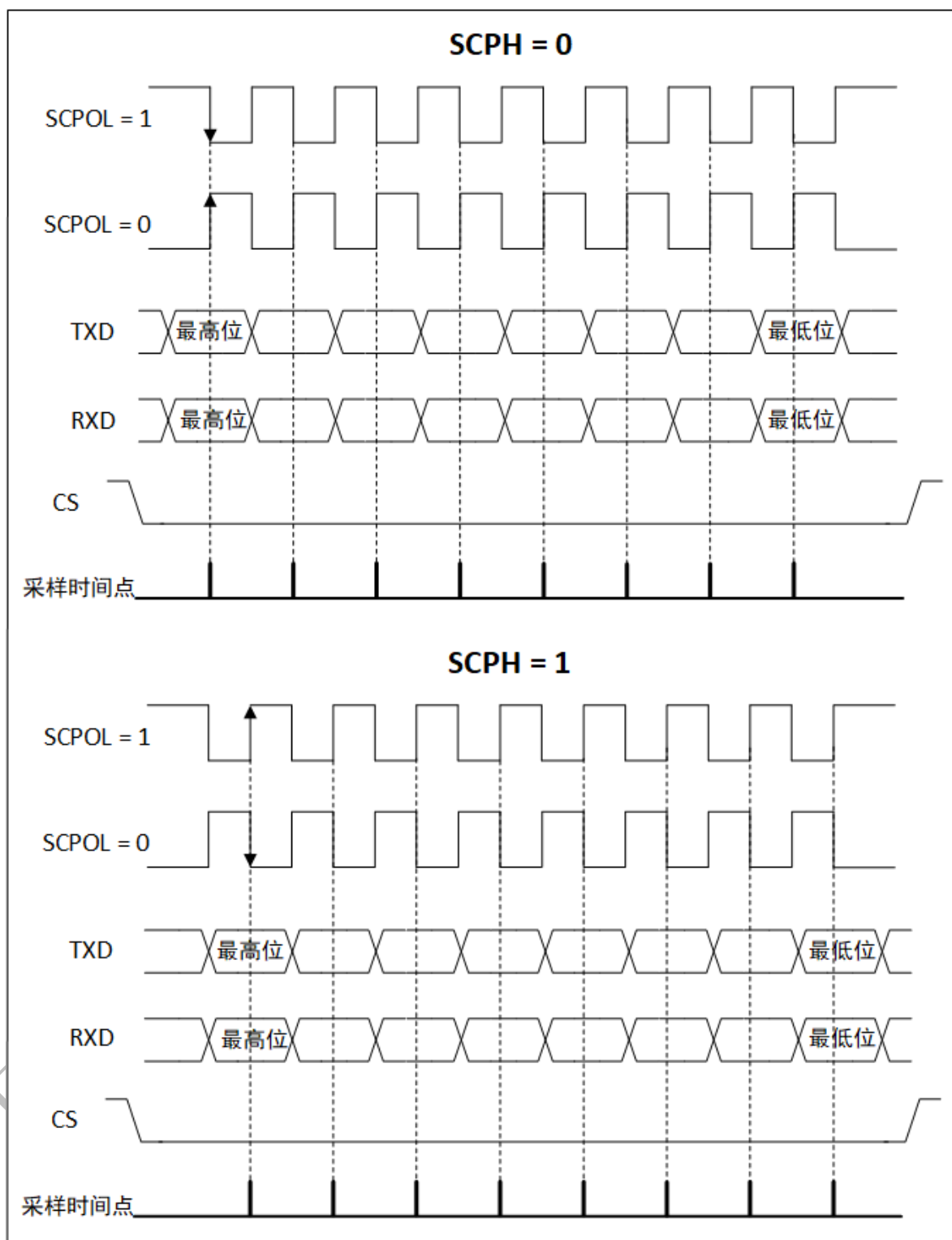


图 6-10 MOTOROLA SPI 时序图

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）提供了一个符合行业标准的 4 线从设备 SPI 接口。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，SPI 从设备还支持可配置 4 至 16 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）。

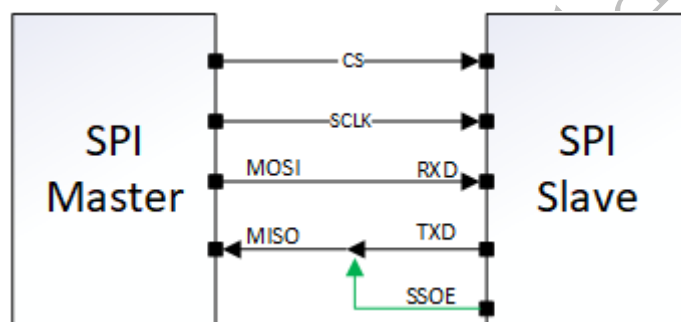


图 7-1 SPI Slave 接线示意图

7.2. 主要特性

- 4 至 16 位数据宽度
- 比特率高达 48 Mbps
- 支持 DMA
- 仅支持标准 SPI 接入方式（不支持 Dual 二线，Quad 四线方式）
- 提供深度 16，位宽 16bit 的 FIFO

7.3. 功能描述

7.3.1. 发送和接收 FIFO 及中断

SPIS 提供了深度 16，位宽 16bit 的 FIFO 缓存。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 8 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 实体中的高八位需要忽略。

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

SPIS 寄存器基地址：0x5003_8000

表格 7-1 SPI 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
------	----	-------

7.4.2. CTRLR0 (Offset 0x00)

表格 7-2 CTRLR0 寄存器

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。
23:16	Reserved	R	0x0	Reserved
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 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 注: 选择 Microwire 格式时有效。
11	SRL	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

比特	名称	属性	复位值	描述
10	SLV_OE	R/W	1'b0	从设备输出使能。 1: 输出不使能 0: 输出使能
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: 保留 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	DFS	R/W	4'h7	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

比特	名称	属性	复位值	描述
				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

7.4.3. SSIENR (Offset 0x08)

表格 7-3 SSIENR 寄存器

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)

表格 7-4 MWCR 寄存器

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)

表格 7-5 TXFTLR 寄存器

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)

表格 7-6 RXFTLR 寄存器

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)

表格 7-7 TXFLR 寄存器

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 7-8 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)

表格 7-9 SR 寄存器

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)

表格 7-10 IMR 寄存器

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)

表格 7-11 ISR 寄存器

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)

表格 7-12 RISR 寄存器

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)

表格 7-13 TXOICR 寄存器

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)

表格 7-14 RXOICR 寄存器

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)

表格 7-15 RXUICR 寄存器

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)

表格 7-16 ICR 寄存器

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)

表格 7-17 DMACR 寄存器

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)

表格 7-18 DMATDLR 寄存器

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)

表格 7-19 DMARDLR 寄存器

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)

表格 7-20 DR 寄存器

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。

7.5. 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。

Freqchip Confidential

7.6. MOTOROLA SPI 时序

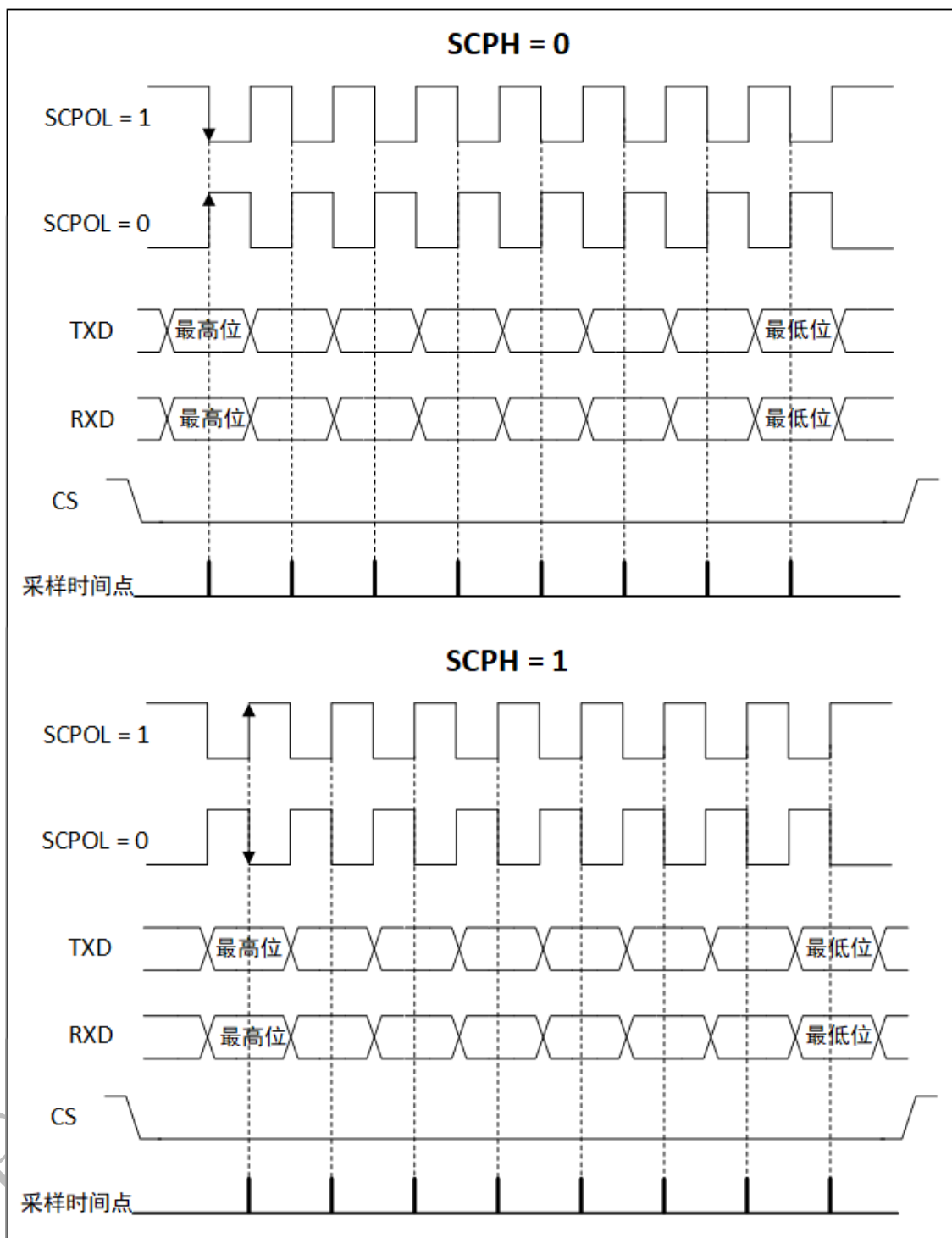


图 7-2 MOTOROLA SPI 时序图

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

- 支持标准、快速、高速三种速度标准
- 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

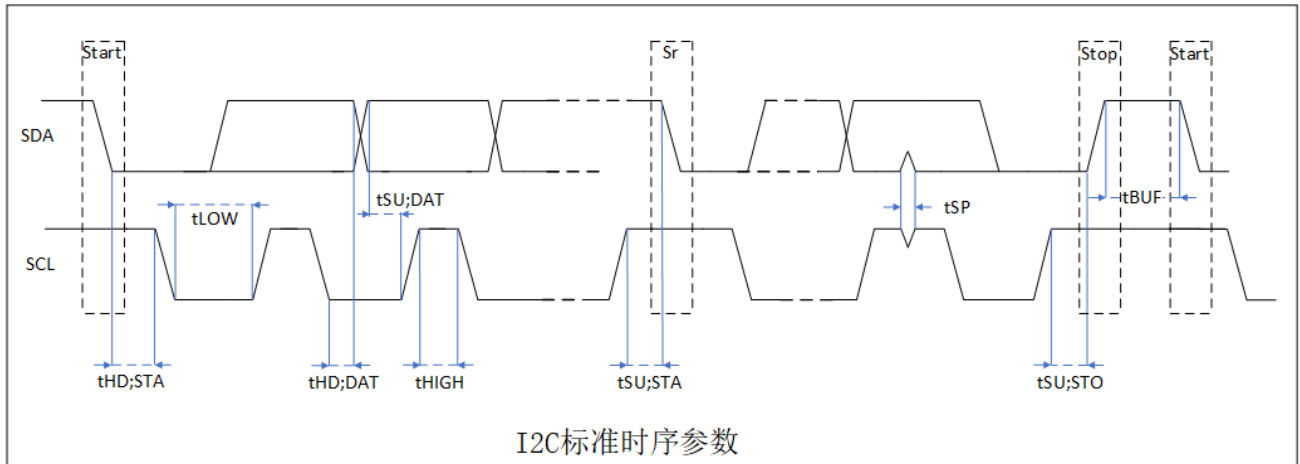


图 8-1 I2C 时序参数图

表格 8-1 I2C 时序参数与寄存器关系

参数	符号	Standard Speed	Fast Speed	High Speed
SCL 时钟低电平周期	tLOW	SS_SCL_LCNT	FS_SCL_LCNT	HS_SCL_LCNT
SCL 时钟高电平周期	tHIGH	SS_SCL_HCNT	FS_SCL_HCNT	HS_SCL_HCNT
重复起始条件的建立时间	tSU;STA	SS_SCL_LCNT	FS_SCL_HCNT	HS_SCL_LCNT
(重复) 起始条件的保持时间	tHD;STA	SS_SCL_HCNT	FS_SCL_HCNT	HS_SCL_LCNT
停止条件建立时间	tSU;STO	SS_SCL_HCNT	FS_SCL_HCNT	HS_SCL_LCNT
停止和启动条件之间的总线空闲时间	tBUF	SS_SCL_LCNT	FS_SCL_LCNT	
突刺长度	tSP	FS_SPKLEN	FS_SPKLEN	HS_SPKLEN
数据保持时间	tHD;DAT	SDA_HOLD	SDA_HOLD	SDA_HOLD
数据建立时间	tSU;DAT	SDA_SETUP	SDA_SETUP	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 表 7-1 所示，所调整的时序可参照 图 7-1 。上述寄存器调整的单位均以 APB_I2C_CLK 为单位。

$$\text{SCL HIGH TIME} = (\text{xx_SCL_HCNT} + \text{xx_SPKLEN} + 3) * \text{APB_I2C_CLK}$$

$$\text{SCL LOW TIME} = (\text{xx_SCL_LCNT} + 1) * \text{APB_I2C_CLK}$$

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。

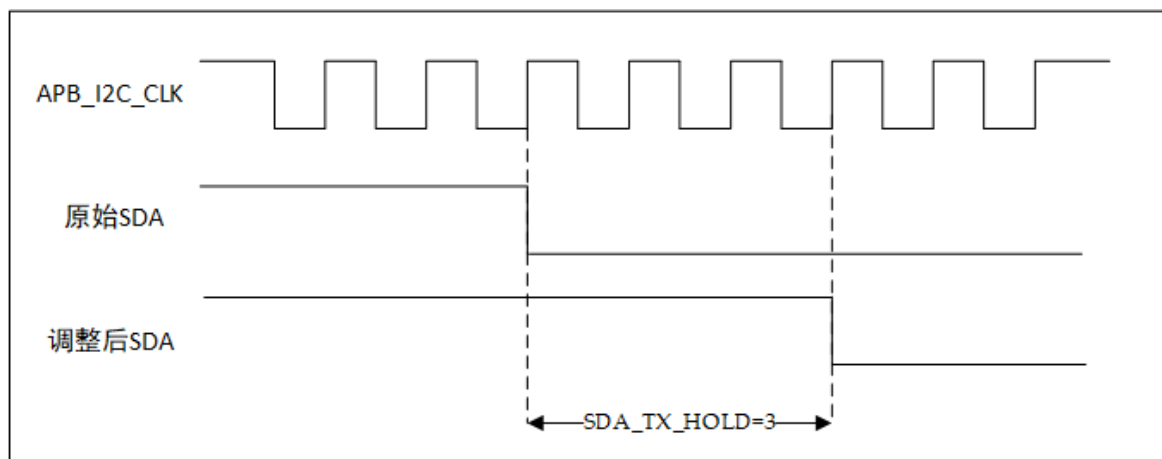


图 8-2 SDA_TX_HOLD 使用示意图

- 对于 SDA_RX_HOLD，将作用于 I2C 内部采样时间点。

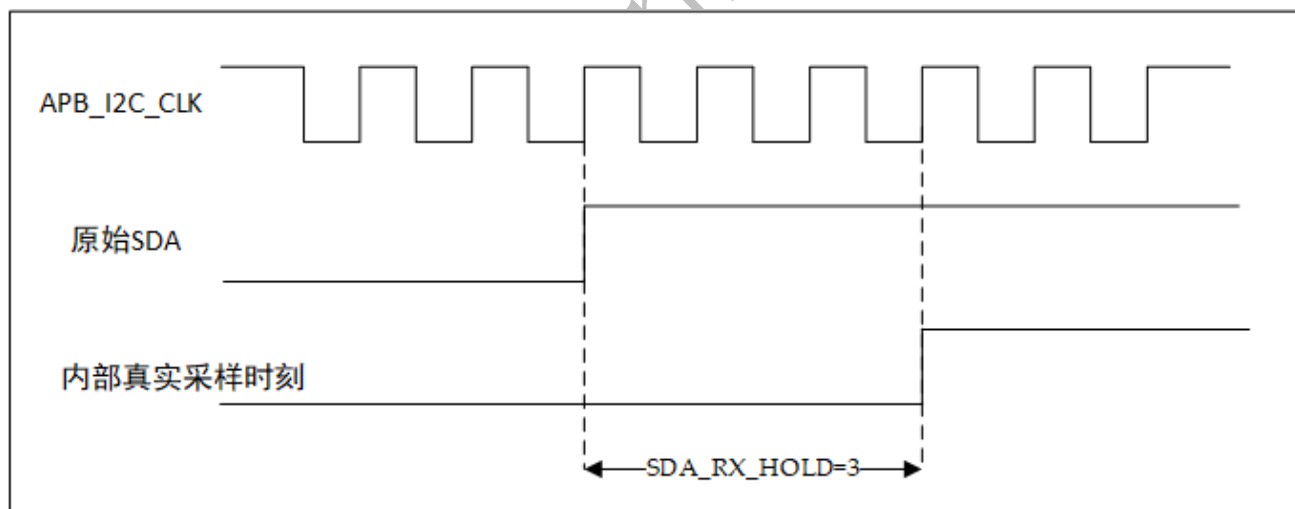


图 8-3 SDA_RX_HOLD 使用示意图

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。

注：此功能仅 I2C0 支持。

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。

注：此功能仅 I2C0 支持。

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。

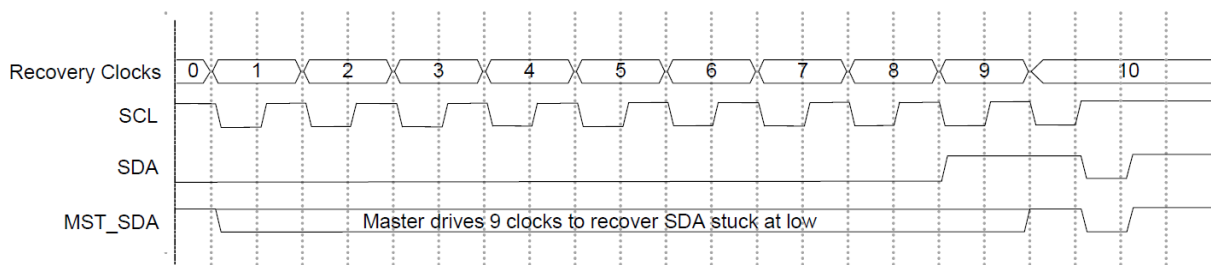


图 8-4 主机发送 9 个 CLK 试图恢复 SDA

8.3.4. 7bit 地址格式、10bit 地址格式

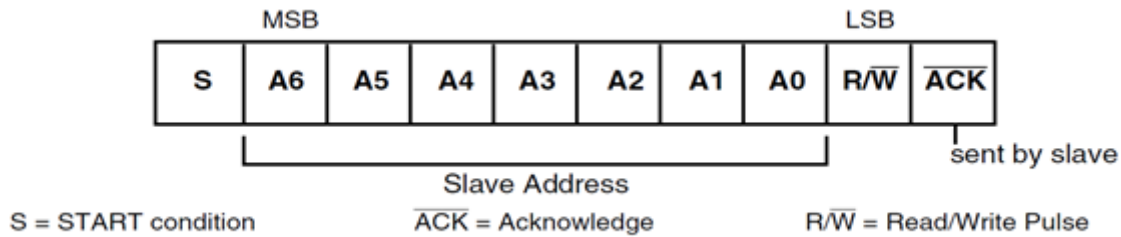


图 8-5 7bit 地址示意图

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。

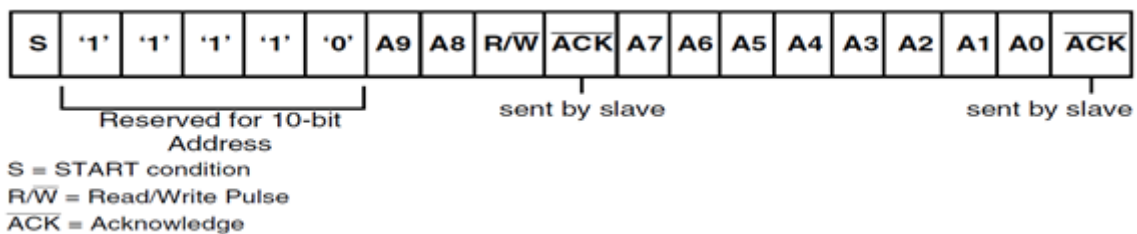


图 8-6 10bit 地址示意图

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 [FS SPKLEN](#)、[HS 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。

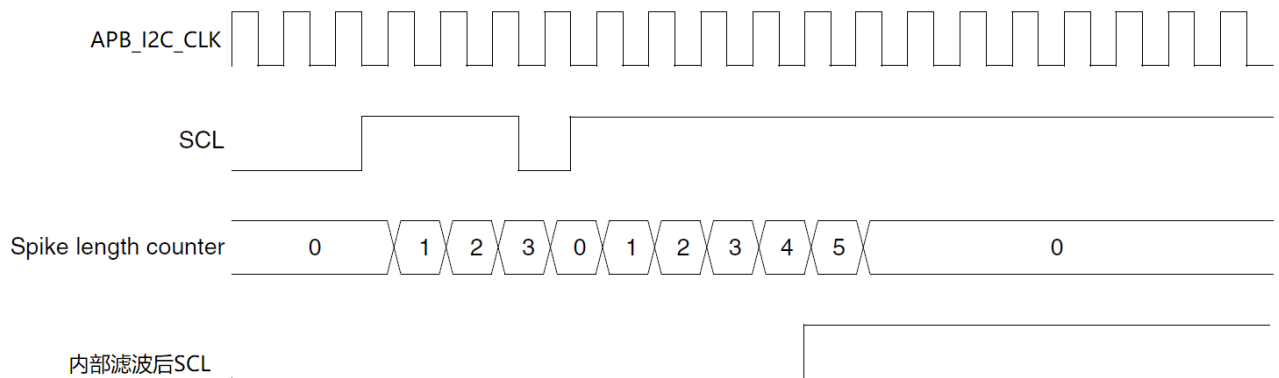


图 8-7 突刺抑制示意图

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：0x5004_0000

I2C1 寄存器基地址：0x5004_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
0x0C	HS_MADDR	高速模式主机码

0x10	DATA_CMD	数据、命令寄存器
0x14	SS_SCL_HCNT	Standard 模式下时钟信号高电平计数
0x18	SS_SCL_LCNT	Standard 模式下时钟信号低电平计数
0x1C	FS_SCL_HCNT	Fast 模式下时钟信号高电平计数 (仅 I2C0 支持)
0x20	FS_SCL_LCNT	Fast 模式下时钟信号低电平计数 (仅 I2C0 支持)
0x24	HS_SCL_HCNT	High Speed 模式下时钟信号高电平计数 (仅 I2C0 支持)
0x28	HS_SCL_LCNT	High Speed 模式下时钟信号低电平计数 (仅 I2C0 支持)
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	FS_SPKLEN	FS 峰值抑制寄存器
0xA4	HS_SPKLEN	HS 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)

表格 8-3 CONTROL 寄存器

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。 注：仅 I2C0 支持，仅主模式下有效。
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：从机只有在地址匹配后才能发出 STOP_DET 中断

				1: 从机总是能发出 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'b11	速度模式。 01: Standard (100kbit/s) 10: Fast (400 kbit/s <= Speed<=1000 kbit/s) 11: High Speed (3.4Mbit/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)

表格 8-4 TAR 寄存器

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)

表格 8-5 SAR 寄存器

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. HS_MADDR (Offset 0x0C)

表格 8-6 HS_MADDR 寄存器

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2:0	HS_MAR	R/W	3'b000	高速模式时，需要发送的主机码。 主机码低三位允许编程：0000_1xxx。

8.4.6. DATA_CMD (Offset 0x10)

表格 8-7 DATA_CMD 寄存器

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: 未接收到地址后的第一个字节 1: 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。

比特	名称	属性	复位值	描述
				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。

8.4.7. SS_SCL_HCNT (Offset 0x14)

表格 8-8 SS_SCL_HCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	SS_SCL_HCNT	R/W	16'h190	Standard 模式下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。

8.4.8. SS_SCL_LCNT (Offset 0x18)

表格 8-9 SS_SCL_LCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	SS_SCL_LCNT	R/W	16'h1D6	Standard 模式下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。

8.4.9. FS_SCL_HCNT (Offset 0x1C)

表格 8-10 FS_SCL_HCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	FS_SCL_HCNT	R/W	16'h3C	Fast 模式下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。

8.4.10. FS_SCL_LCNT (Offset 0x20)

表格 8-11 FS_SCL_LCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	FS_SCL_LCNT	R/W	16'h82	Fast 模式下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。

8.4.11. HS_SCL_HCNT (Offset 0x24)

表格 8-12 HS_SCL_HCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	HS_SCL_HCNT	R/W	16'h6	High Spee 模式下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。

8.4.12. HS_SCL_LCNT (Offset 0x28)

表格 8-13 HS_SCL_LCNT 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	HS_SCL_LCNT	R/W	16'h10	High Spee 模式下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。

8.4.13. INTR_STAT (Offset 0x2C)

表格 8-14 INTR_STAT 寄存器

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 (仅 I2C0 支持)
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

比特	名称	属性	复位值	描述
				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	从机发送时, 在从机发送完数据后, 主机回应 NAK, RX_DONE 置位。 一般用在一次传输的最后一个字节, 表示传输结束。 0: Status inactive 1: Status active

比特	名称	属性	复位值	描述
6	TX_ABRT	R	1'b0	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。 TX_ABRT_SOURCE 寄存器中可以查询发送中止的原因。 0: Status inactive 1: Status active
5	RD_REQ	R	1'b0	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。 0: Status inactive 1: Status active
4	TX_EMPTY	R	1'b0	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。 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。 0: Status inactive 1: Status active 注：此位会在 Tx FIFO 中数据超过阈值时自动清除。
3	TX_OVER	R	1'b0	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。 0: Status inactive 1: Status active
2	RX_FULL	R	1'b0	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。 注：此位会在 Rx FIFO 中数据小于阈值时自动清除。 0: Status inactive

比特	名称	属性	复位值	描述
				1: Status active
1	RX_OVER	R	1'b0	当 Rx FIFO 已满，并接收到了新的数据，RX_OVER 置位。 0: Status inactive 1: Status active 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。 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。
0	RX_UNDER	R	1'b0	Rx FIFO 空时，继续读取 Rx FIFO，RX_UNDER 状态置位。

8.4.14. INTR_MASK (Offset 0x30)

表格 8-15 INTR_MASK 寄存器

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	中断状态屏蔽。 0: 相应位中断状态屏蔽。 1: 相应位中断状态不屏蔽。 注：屏蔽后，INTR_STAT 寄存器中相应位，将会被屏蔽，不会置位。
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	

8.4.15. RAW_INTR_STAT (Offset 0x34)

表格 8-16 RAW_INTR_STAT 寄存器

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	

8.4.16. RX_TL (Offset 0x38)

表格 8-17 RX_TL 寄存器

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

8.4.17. TX_TL (Offset 0x3C)

表格 8-18 TX_TL 寄存器

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

8.4.18. CLR_INTR (Offset 0x40)

表格 8-19 CLR_INTR 寄存器

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。

8.4.19. CLR_RX_UNDER (Offset 0x44)

表格 8-20 CLR_RX_UNDER 寄存器

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。

8.4.20. CLR_RX_OVER (Offset 0x48)

表格 8-21 CLR_RX_OVER 寄存器

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。

8.4.21. CLR_TX_OVER (Offset 0x4C)

表格 8-22 CLR_TX_OVER 寄存器

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。

8.4.22. CLR_RD_REQ (Offset 0x50)

表格 8-23 CLR_RD_REQ 寄存器

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。

8.4.23. CLR_TX_ABORT (Offset 0x54)

表格 8-24 CLR_TX_ABORT 寄存器

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

8.4.24. CLR_RX_DONE (Offset 0x58)

表格 8-25 CLR_RX_DONE 寄存器

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。

8.4.25. CLR_ACTIVITY (Offset 0x5C)

表格 8-26 CLR_ACTIVITY 寄存器

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。

8.4.26. CLR_STOP_DET (Offset 0x60)

表格 8-27 CLR_STOP_DET 寄存器

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。

8.4.27. CLR_START_DET (Offset 0x64)

表格 8-28 CLR_START_DET 寄存器

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。

8.4.28. ENABLE (Offset 0x6C)

表格 8-29 ENABLE 寄存器

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 (仅 I2C0 支持)
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

8.4.29. STATUS (Offset 0x70)

表格 8-30 STATUS 寄存器

比特	名称	属性	复位值	描述
31:12	Reserved	R	0x0	Reserved

比特	名称	属性	复位值	描述
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 (仅 I2C0 支持)
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。 0：Tx FIFO 非空 1：Tx FIFO 空
1	TFNF	R	1'b0	Tx FIFO 满状态。 0：Tx FIFO 满

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

8.4.30. TXFLR (Offset 0x74)

表格 8-31 TXFLR 寄存器

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。

8.4.31. RXFLR (Offset 0x78)

表格 8-32 RXFLR 寄存器

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。

8.4.32. SDA_HOLD (Offset 0x7C)

表格 8-33 SDA_HOLD 寄存器

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。

8.4.33. TX_ABRT_SOURCE (Offset 0x80)

表格 8-34 TX_ABRT_SOURCE 寄存器

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 (仅 I2C0 支持)
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

比特	名称	属性	复位值	描述
				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

8.4.34. SLV_DATA_NACK_ONLY (Offset 0x84)

表格 8-35 SLV_DATA_NACK_ONLY 寄存器

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
--	--	--	--	-----------------------------------

8.4.35. DMA_CR (Offset 0x88)

表格 8-36 DMA_CR 寄存器

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

8.4.36. DMA_TDLR (Offset 0x8C)

表格 8-37 DMA_TDLR 寄存器

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

8.4.37. DMA_RDLR (Offset 0x90)

表格 8-38 DMA_RDLR 寄存器

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

8.4.38. SDA_SETUP (Offset 0x94)

表格 8-39 SDA_SETUP 寄存器

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	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。 注：在 I2C 未使能时配置（ENABLE = 0），否则无效。

8.4.39. FS_SPKLEN (Offset 0xA0)

表格 8-40 FS_SPKLEN 寄存器

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

8.4.40. HS_SPKLEN (Offset 0xA4)

表格 8-41 HS_SPKLEN 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:0	HS_SPKLEN	R/W	8'h1	在 HS 模式下运行时，突刺抑制逻辑过滤掉的最长峰值。最小值应不小于 1。 参照 I2C 时序的 tSP 部分，详情参照 I2C 时序控制 章节。 注：在 I2C 未使能时配置（ENABLE = 0），否

				则无效。
--	--	--	--	------

8.4.41. CLR_RESTART_DET (Offset 0xA8)

表格 8-42 CLR_RESTART_DET 寄存器

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。

8.4.42. SCL_STUCK_AT_LOW_TIMEOUT (Offset 0xAC)

表格 8-43 SCL_STUCK_AT_LOW_TIMEOUT 寄存器

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。

8.4.43. SDA_STUCK_AT_LOW_TIMEOUT (Offset 0xB0)

表格 8-44 SDA_STUCK_AT_LOW_TIMEOUT 寄存器

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。

8.4.44. CLR_SCL_STUCK_DET (Offset 0xB4)

表格 8-45 CLR_SCL_STUCK_DET 寄存器

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. 如果是 HIGH_SPEED 模式，配置 HS_MADDR 寄存器，设置主机码。
5. ENABLE 寄存器的[0]位写 ‘1’ 使能 I2C。
6. 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 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 TxFIFO 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 TxFIFO，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 Rx FIFO。
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。

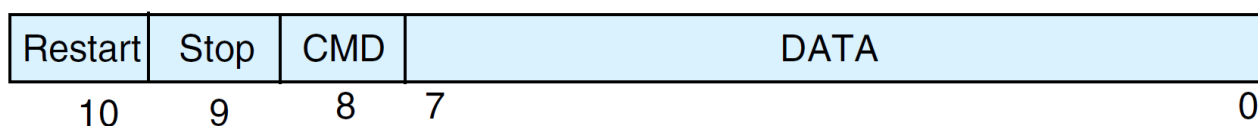


图 8-8 DATA_CMD 寄存器示意图

- 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用的情况下，STOP 后跟 START)

1. 主机发送，并产生 STOP 时序

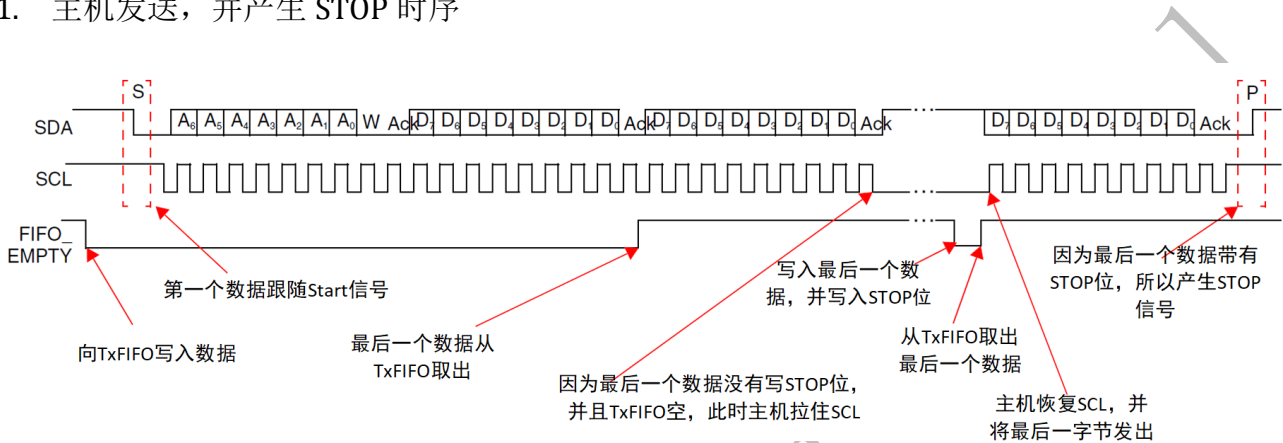


图 8-9 主机发送并产生 STOP

2. 主机接收，并产生 STOP 时序

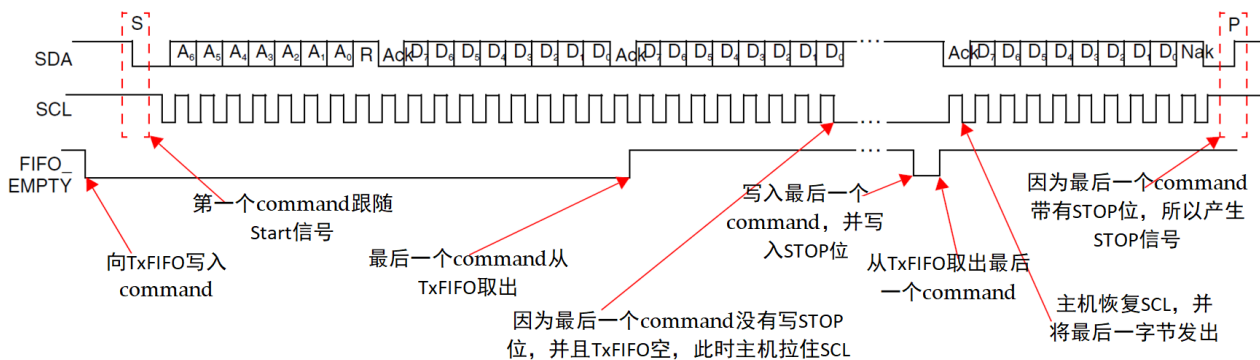


图 8-10 主机接收并产生 STOP 时序

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

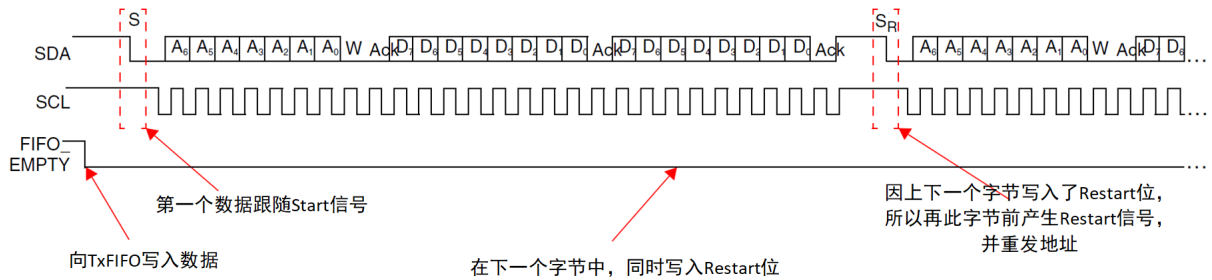


图 8-11 主机发送并产生 Restart 时序

4. 主机接收时, 并产生 Restart 时序

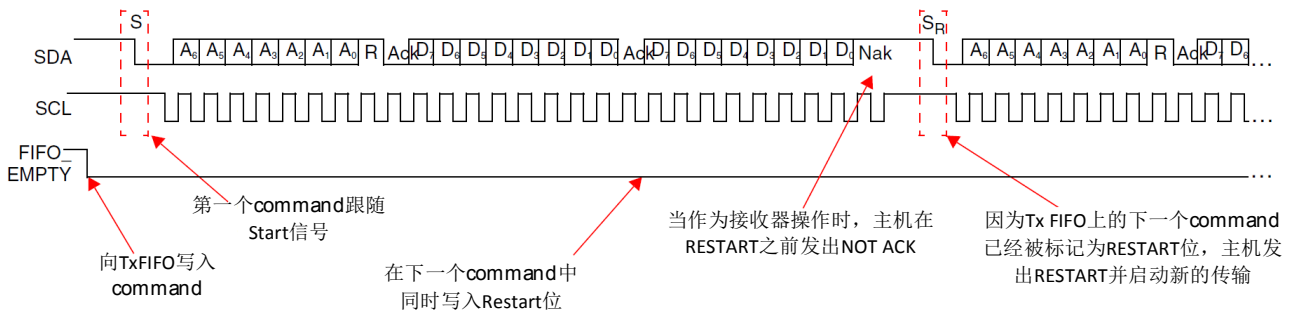


图 8-12 主机接收并产生 Restart 时序

5. 主机发送时, 产生 STOP, 但 Tx FIFO 非空

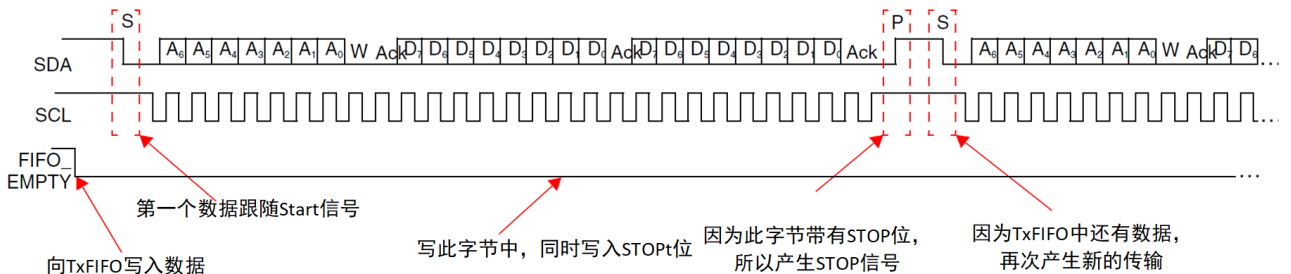


图 8-13 主机发送, 产生 STOP, Tx FIFO 非空

6. 主机接收时, 产生 STOP, 但 Tx FIFO 非空

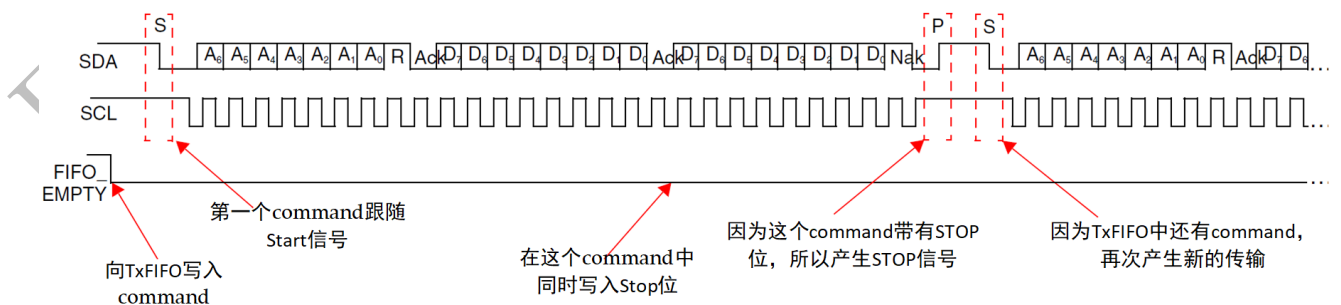


图 8-14 主机接收, 产生 STOP, Tx FIFO 非空

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），PCM（短帧），飞利浦格式。
- 仅支持有主模式下的 16bit 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

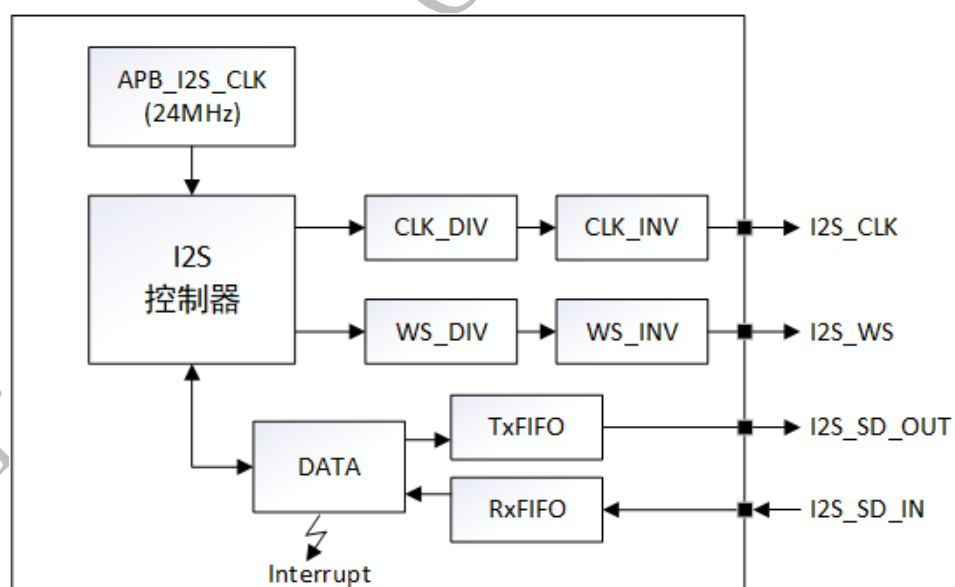


图 9-1 I2S 结构框图

9.4. 功能描述

9.4.1. CLK、WS 分频

时钟信号输出分频，与声道信号输出分频。

$$\text{CLK 分频计算公式: } \text{CLK} = \frac{\text{APB_I2S_CLK}}{2 \times (\text{CLK_DIV} + 1)}$$

$$\text{WS 分频计算公式: } \text{WS} = \frac{\text{CLK}}{(\text{WS_DIV} + 1)}$$

其中 APB_I2S_CLK 固定为 24MHz。

9.4.2. WS_INV, DLY_EN, FMT 配置效果

WS_INV = 0, DLY_EN = 0, FMT = 0

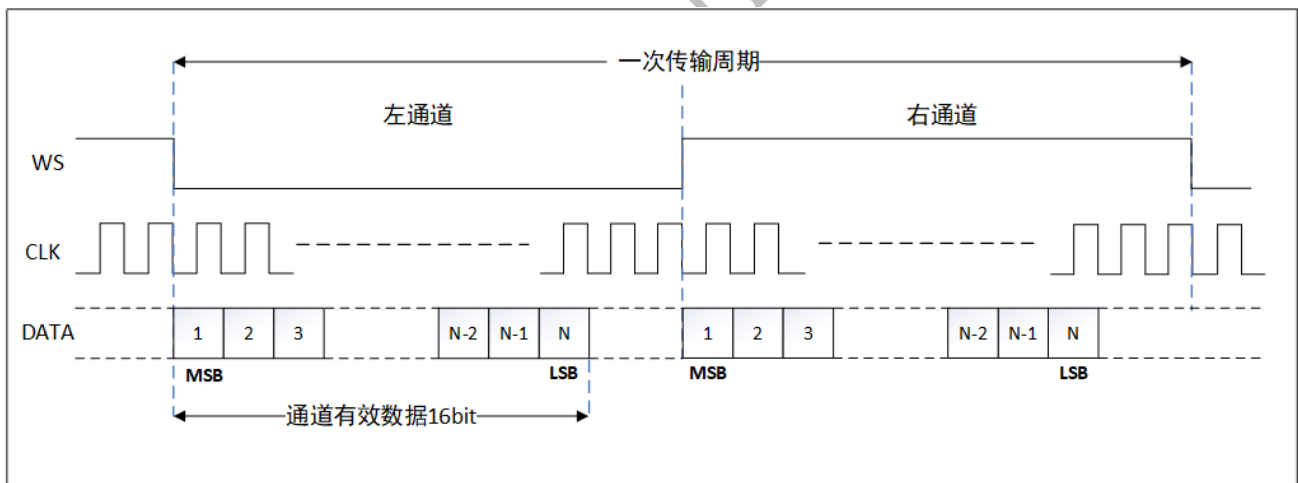


图 9-2 I2S 时序图（左对齐）

WS_INV = 1, DLY_EN = 0, FMT = 0

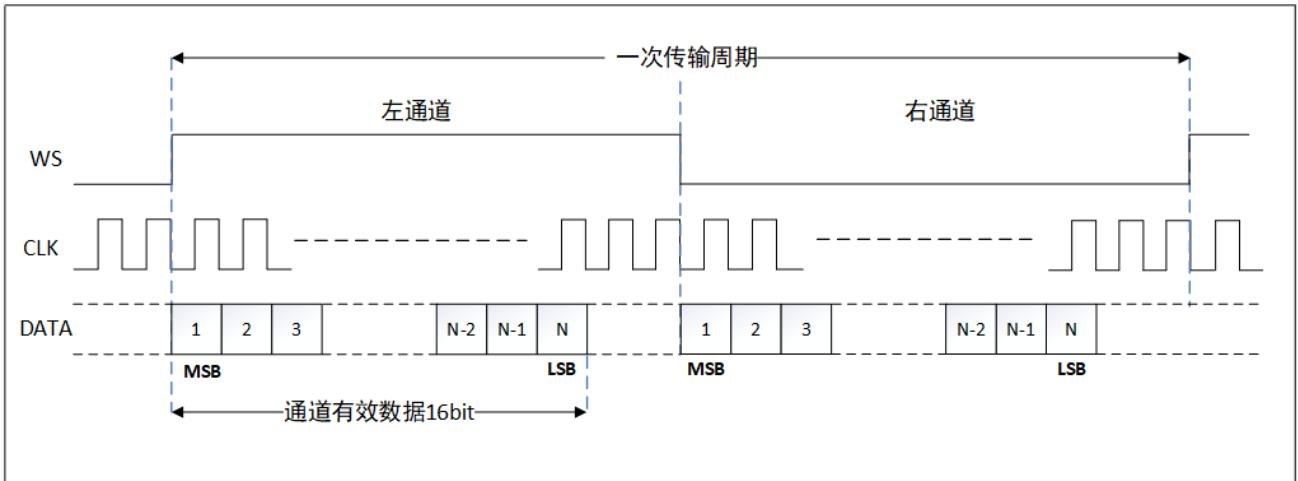


图 9-3 I2S 时序图（左对齐，WS 反相）

WS_INV = 1, DLY_EN = 1, FMT = 0

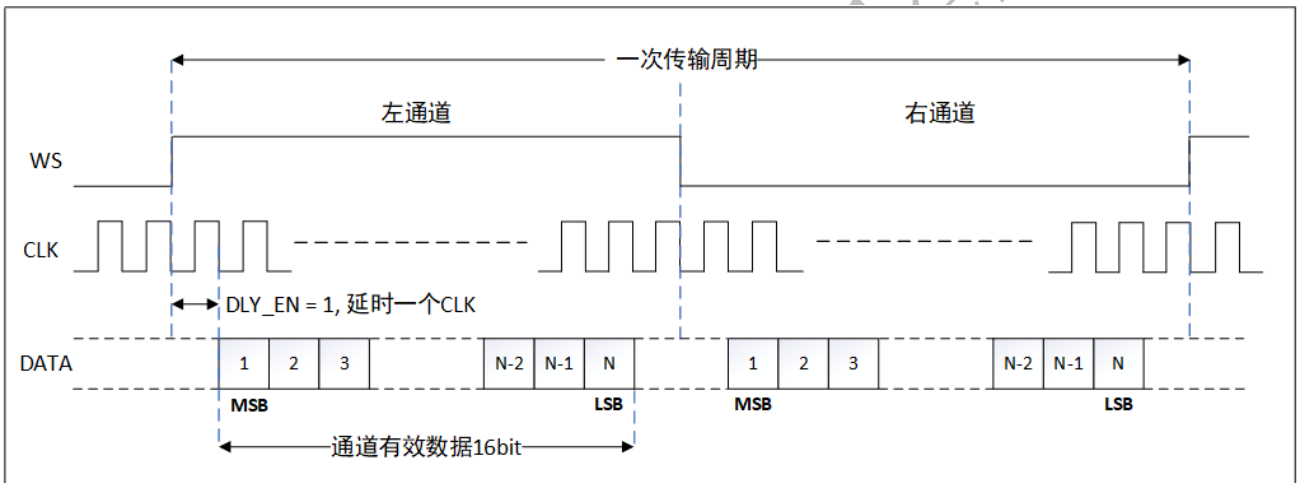
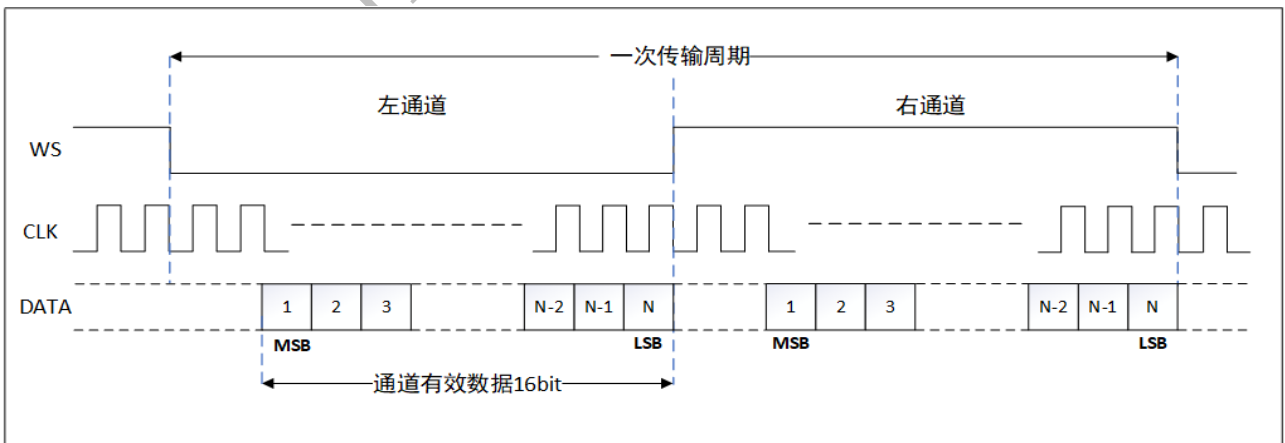


图 9-4 I2S 时序图（左对齐，WS 反相，DLY_EN = 1）



WS_INV = 0, DLY_EN = 0, FMT = 1

图 9-5 I2S 时序图（右对齐）

9.5. I2S 寄存器

9.5.1. I2S 寄存器映像

I2S 寄存器基地址：0x5007_0000

表格 9-1 I2S 寄存器映像

偏移	寄存器	寄存器描述
0x00	CONTROL	控制寄存器
0x04	CLK_DIV	时钟分频
0x08	WS_DIV	声道分频
0x0C	DATA	数据寄存器
0x10	STATUS	状态寄存器
0x14	INTE	中断使能寄存器
0x20	FIFO_CFG	FIFO 配置寄存器
0x24	DMA_CFG	DMA 配置寄存器

9.5.2. CONTROL (Offset 0x00)

表格 9-2 CONTROL 寄存器

比特	名称	属性	复位值	描述
31:11	Reserved	R/W	0x0	Reserved
10	HLSEL	R/W	1'b0	数据选择。 0: 接收数据时, DATA 寄存器储存方式: 右声道在高四字节, 左声道在低四字节。 1: 接收数据时, DATA 寄存器储存方式: 左声道在高四字节, 右声道在低四字节。
9	INT_EN	R/W	1'b0	I2S 总中断使能
8	TX_INT_EN	R/W	1'b0	发送中断使能
7	RX_INT_EN	R/W	1'b0	接收中断使能
6:5	Reserved	R/W	0x0	Reserved
4	CLK_INV	R/W	1'b0	0: 无影响 1: CLK 信号反相
3	WS_INV	R/W	1'b0	0: 无影响

比特	名称	属性	复位值	描述
				1: WS 信号反相
2	DLY_EN	R/W	1'b0	延时使能 0: 无影响 1: 发送数据帧时延时 1 个 CLK。
1	FMT	R/W	1'b0	格式选择。 0: PCM (短帧) 1: 左对齐
0	I2S_EN	R/W	1'b0	I2S 使能

注：当 FMT 选择左对齐，DLY_EN = 0 时，为 MSB（左对齐）标准。DLY_EN = 1 时为飞利浦标准。当 FMT 选择 PCM 时，通道宽度必须配置 $\geq 32\text{bit}$ 。

9.5.3. CLK_DIV (Offset 0x04)

表格 9-3 CLK_DIV 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
15:0	CLK_DIV	R/W	16'h00	CLK 时钟分频。 $\text{Clock} = \text{APB Clock} / (\text{CLK_DIV} + 1) * 2$

9.5.4. WS_DIV (Offset 0x08)

表格 9-4 WS_DIV 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
15:0	WS_DIV	R/W	16'h00	声道分频。 $\text{WS} = \text{Clock} / (\text{WS_DIV} + 1)$

9.5.5. DATA (Offset 0x0C)

表格 9-5 DATA 寄存器

比特	名称	属性	复位值	描述
31:0	DATA	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. STATUS (Offset 0x10)

表格 9-6 STATUS 寄存器

比特	名称	属性	复位值	描述
31:6	Reserved	R/W	0x0	Reserved
5	TX_FIFO_Empty	R	1'b0	TxFIFO 空
4	TX_FIFO_HalfEmpty	R	1'b0	TxFIFO 半空
3	TX_FIFO_Full	R	1'b0	TxFIFO 满
2	RX_FIFO_Empty	R	1'b0	RxFIFO 空
1	RX_FIFO_HalfFull	R	1'b0	RxFIFO 半满
0	RX_FIFO_Full	R	1'b0	RxFIFO 满

9.5.7. INTE (Offset 0x14)

表格 9-7 INTE 寄存器

比特	名称	属性	复位值	描述
31:6	Reserved	R/W	0x0	Reserved
5	TX_FIFO_Empty_INT	R	1'b0	TxFIFO 空中断使能 0: Disable 1: Enable
4	TX_FIFO_HalfEmpty_INT	R	1'b0	TxFIFO 半空中断使能 0: Disable 1: Enable
3	TX_FIFO_Full_INT	R	1'b0	TxFIFO 满中断使能 0: Disable 1: Enable
2	RX_FIFO_Empty_INT	R	1'b0	RxFIFO 空中断使能 0: Disable 1: Enable
1	RX_FIFO_HalfFull_INT	R	1'b0	RxFIFO 半满中断使能 0: Disable 1: Enable

比特	名称	属性	复位值	描述
0	RX_FIFO_Full_INT	R	1'b0	RxFIFO 满中断使能 0: Disable 1: Enable

9.5.8. FIFO_CFG (Offset 0x20)

表格 9-8 FIFO_CFG 寄存器

比特	名称	属性	复位值	描述
31:24	RxFIFO_CFG	R/W	8'h32	RxFIFO 满阈值。 RxFIFO 中数据 $\geq$ RxFIFO_CFG, 则 RX_FIFO_Full 状态置位。
23:8	Reserved	R/W	0x0	Reserved
7:0	TxFIFO_CFG	R/W	8'h0E	TxFIFO 空阈值。 TxFIFO 中数据 $\leq$ TxFIFO_CFG, 则 TX_FIFO_Empty 状态置位

9.5.9. DMA_CFG (Offset 0x24)

表格 9-9 DMA_CFG 寄存器

比特	名称	属性	复位值	描述
31:12	Reserved	R/W	0x0	Reserved
11:7	DMA_TX_Level	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
6:2	DMA_RX_Level	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
1	DMA_RX_EN	R/W	1'b0	DMA Rx 使能。 0: Disable 1: Enable
0	DMA_TX_EN	R/W	1'b0	DMA Tx 使能。 0: Disable

比特	名称	属性	复位值	描述
				1: Enable

Freqchip Confidential

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

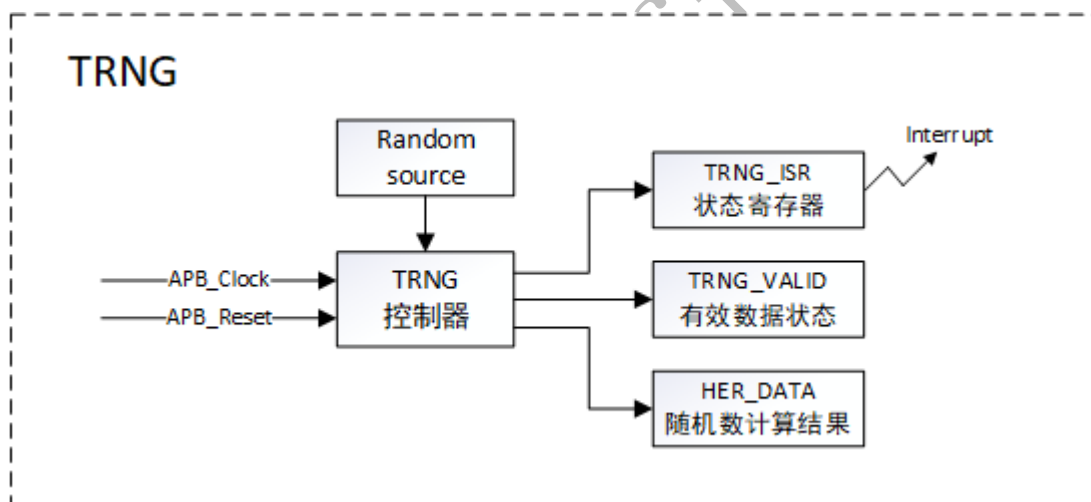


图 10-1 TRNG 结构框图

10.4. TRNG 寄存器

10.4.1. TRNG 寄存器映像

TRNG 寄存器基地址：0x500C_0000

表格 10-1 TRNG 寄存器映像

偏移	寄存器	寄存器描述
0x100	RNG_IMR	中断屏蔽寄存器

偏移	寄存器	寄存器描述
0x104	RNG_ISR	中断状态寄存器
0x108	RNG_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	HER_DATA0	随机数结果寄存器
0x118	HER_DATA1	
0x11C	HER_DATA2	
0x120	HER_DATA3	
0x124	HER_DATA4	
0x128	HER_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. RNG_IMR (Offset 0x100)

表格 10-2 RNG_IMR 寄存器

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. RNG_ISR (Offset 0x104)

表格 10-3 RNG_ISR 寄存器

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. RNG_ICR (Offset 0x108)

表格 10-4 RNG_ICR 寄存器

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)

表格 10-5 TRNG_CONFIG 寄存器

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)

表格 10-6 TRNG_VALID 寄存器

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. HER_DATA0 ~ 5 (Offset 0x114 ~ 0x128)

表格 10-7 HER_DATA0~5 寄存器

比特	名称	属性	复位值	描述
31:0	HER_DATA5 (0x124)	R	32'h00	随机数[191:160]
31:0	HER_DATA4 (0x124)	R	32'h00	随机数[159:128]
31:0	HER_DATA3 (0x120)	R	32'h00	随机数[127:96]
31:0	HER_DATA2 (0x11C)	R	32'h00	随机数[95:64]
31:0	HER_DATA1 (0x118)	R	32'h00	随机数[63:32]
31:0	HER_DATA0 (0x114)	R	32'h00	随机数[31:0]

10.4.8. RND_SOURCE_ENABLE (Offset 0x12C)

表格 10-8 RND_SOURCE_ENABLE 寄存器

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)

表格 10-9 SAMPLE_CNT1 寄存器

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)

表格 10-10 TRNG_RESET 寄存器

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)

表格 10-11 TRNG_BUSY 寄存器

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)

表格 10-12 RST_BITS_COUNTER 寄存器

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

11.3. PDM 系统框图

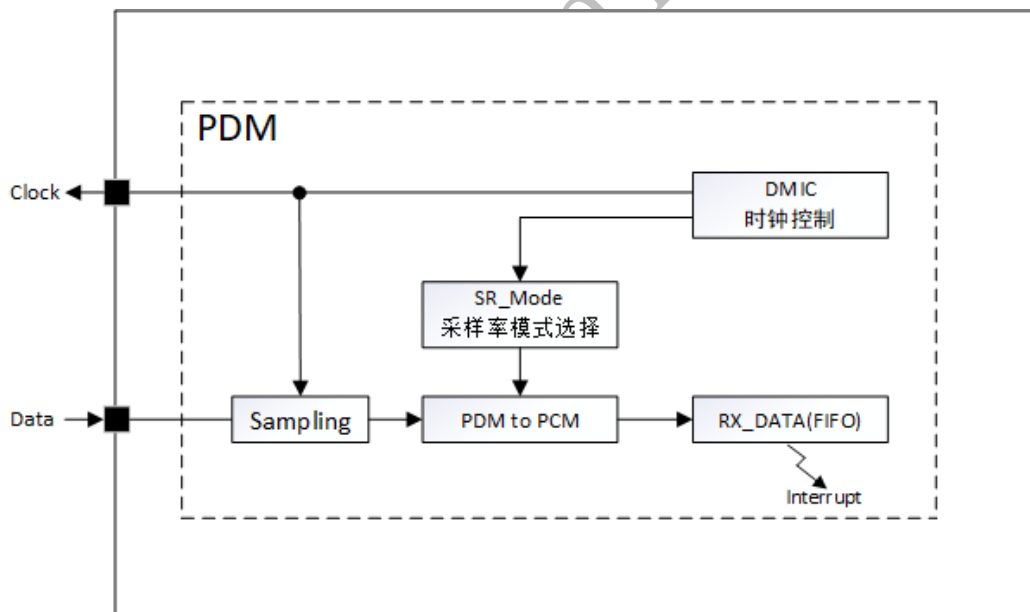


图 11-1 PDM 系统框图

11.4. PDM 寄存器

11.4.1. PDM 寄存器映像

PDM 寄存器基地址：0x500A_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	VOLUME	音量值
0x08	VOLSTEP	音量配置
0x0C	RX_DATA	接收数据寄存器
0x10	FIFO_Clear	FIFO 清除寄存器
0x14	FIFO_Status	FIFO 状态寄存器
0x18	FIFO_INT_EN	FIFO 中断使能寄存器
0x1C	FIFO_Almost_Level	FIFO 近满阈值设置寄存器
0x20	DMA_CFG	DMA 控制寄存器

11.4.2. PDM_Control (Offset 0x00)

表格 11-2 PDM_Control 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
7:6	CLK_CFG	R/W	2'b00	DMIC 时钟配置。 00: 1M 01: 1.5M 10: 2M 11: 3M
5	FIFO_Enable	R/W	1'b0	FIFO 使能。 0: Disable FIFO 1: Enable FIFO
4	ZERO_DET	R/W	1'b1	音量调节是否在过零点。 0: 随时调节音量 1: 过零点时调节音量
3	HPF_EN	R/W	1'b1	高通滤波器使能。 0: High-Pass Filter Disable 1: High-Pass Filter Enable
2	SR_Mode	R/W	1'b0	采样率模式选择。 0: 62.5

比特	名称	属性	复位值	描述
				1: 125 PCM 率 = DMIC_Clock / SR_Mode。 例：设置 DMIC 为 1M，SR_MODE 为 1。则采样率为 1M/125 = 8KHz。
1	EDGE_Select	R/W	1'b1	采样边沿选择。 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. VOLUME (Offset 0x04)

表格 11-3 VOLUME 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
11:0	Volume	R/W	12'h172	音量值。 音量值 = Volume_Gain / 256

11.4.4. VOLSTEP (Offset 0x08)

表格 11-4 VOLSTEP 寄存器

比特	名称	属性	复位值	描述
31:17	Reserved	R	0x0	Reserved
16	VOL_DIRECT	R/W	1'b0	直接调节音量。 调节音量时直接达到目标值 (Volume)
15:10	Reserved	R	0x0	Reserved
9:0	VOL_STEP	R/W	10'h01	逐渐调节音量。 调节音量时使用逐步递增或递减方式，最终达到目标值。每次递增或递减值 (VOL_STEP)

11.4.5. RX_DATA (Offset 0x0C)

表格 11-5 RX_DATA 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:0	RxData	R	1'b0	接收数据

11.4.6. FIFO_Clear (Offset 0x10)

表格 11-6 FIFO_Clear 寄存器

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	Read_FIFO_Pointer_CLR	R/W	1'b0	复位读 FIFO 指针 1: reset Read FIFO pointer 0: release
0	Write_FIFO_Pointer_CLR	R/W	1'b0	复位写 FIFO 指针 1: reset Write FIFO pointer 0: release

11.4.7. FIFO_Status (Offset 0x14)

表格 11-7 FIFO_Status 寄存器

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	FIFO_Empty	R/W	1'b0	FIFO 空
1	FIFO_Almost_Full	R/W	1'b0	FIFO 近满
0	FIFO_Full	R/W	1'b0	FIFO 满

11.4.8. FIFO_INT_EN (Offset 0x18)

表格 11-8 FIFO_INT_EN 寄存器

比特	名称	属性	复位值	描述
31:3	Reserved	R	0x0	Reserved
2	FIFO_Empty_INTEN	R/W	1'b0	FIFO 空中断使能

				0: Disable 1: Enable
1	FIFO_Almost_Full_INTEN	R/W	1'b0	FIFO 近满中断使能 0: Disable 1: Enable
0	FIFO_Full_INTEN	R/W	1'b0	FIFO 满中断使能 0: Disable 1: Enable

11.4.9. FIFO_Almost_Level (Offset 0x0x1C)

表格 11-9 FIFO_Almost_Level 寄存器

比特	名称	属性	复位值	描述
31:5	Reserved	R	0x0	Reserved
4:0	FIFO_Almost_Level	R/W	5'b10	FIFO 近满阈值设置。 当 FIFO 数据达到设置值时，FIFO_Stats 寄存器中 FIFO_Almost_Full 置位。

11.4.10. DMA_CFG (Offset 0x20)

表格 11-10 DMA_CFG 寄存器

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5	DMA_EN	R/W	1'b0	DMA 使能。 0: Disable 1: Enable
4:0	DMA_Rx_Level	R/W	5'h1E	DMA Rx 阈值设置。 当 FIFO 数据等于或大于设置值时，产生 DMA Rx 请求

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

- 3 个独立的可配置通道（请求）
- 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）
- 通道 0 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_MSIZE、DEST_MSIZE 为 Burst length。

例如：

BLOCK_TS = 5.

SRC_TR_WIDTH = 0x2. (4 byte)

DST_TR_WIDTH = 0x2. (4 byte)

SRC_MSIZE = 0x1. (待传输的数据项为 4)

DEST_MSIZE = 0x1. (待传输的数据项为 4)

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

DMA 收到一次传输请求后，将待传输的数据项 $4 \times 4 \text{ byte} = 16 \text{ byte}$ 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_MSIZE 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_MSIZE 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。

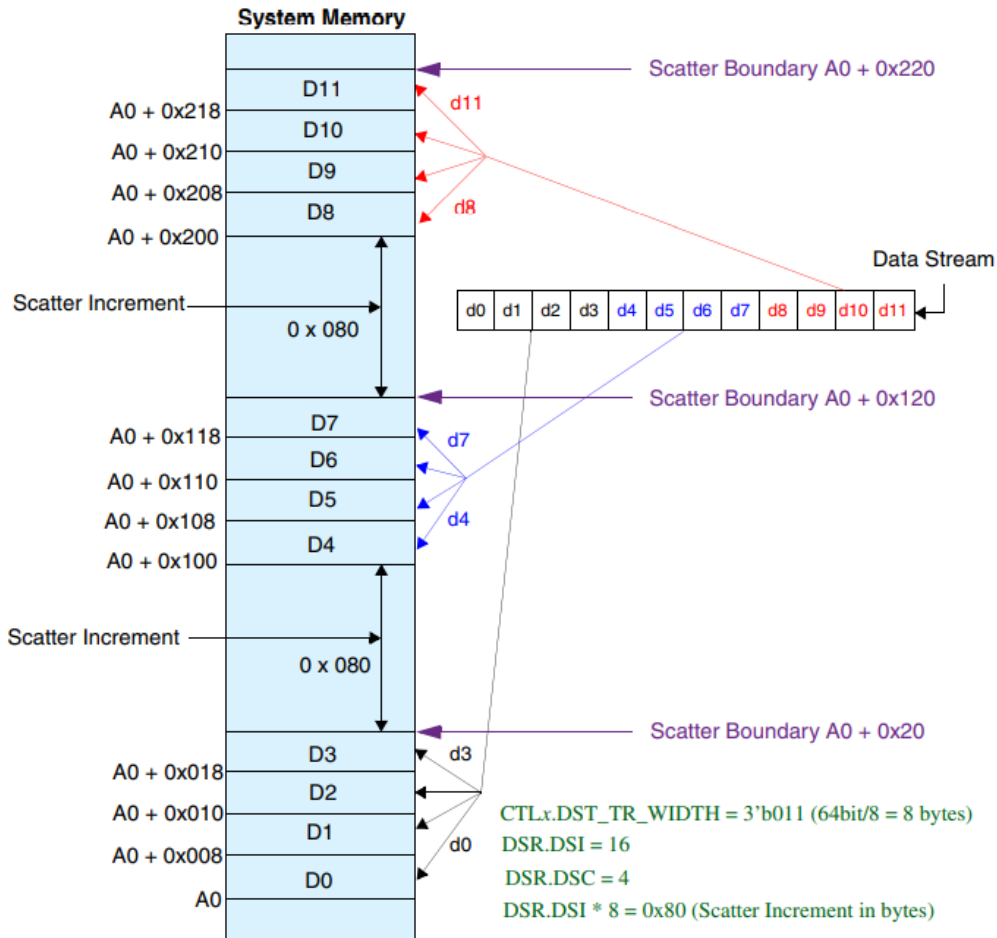


图 12-1 离散传输示例图

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。

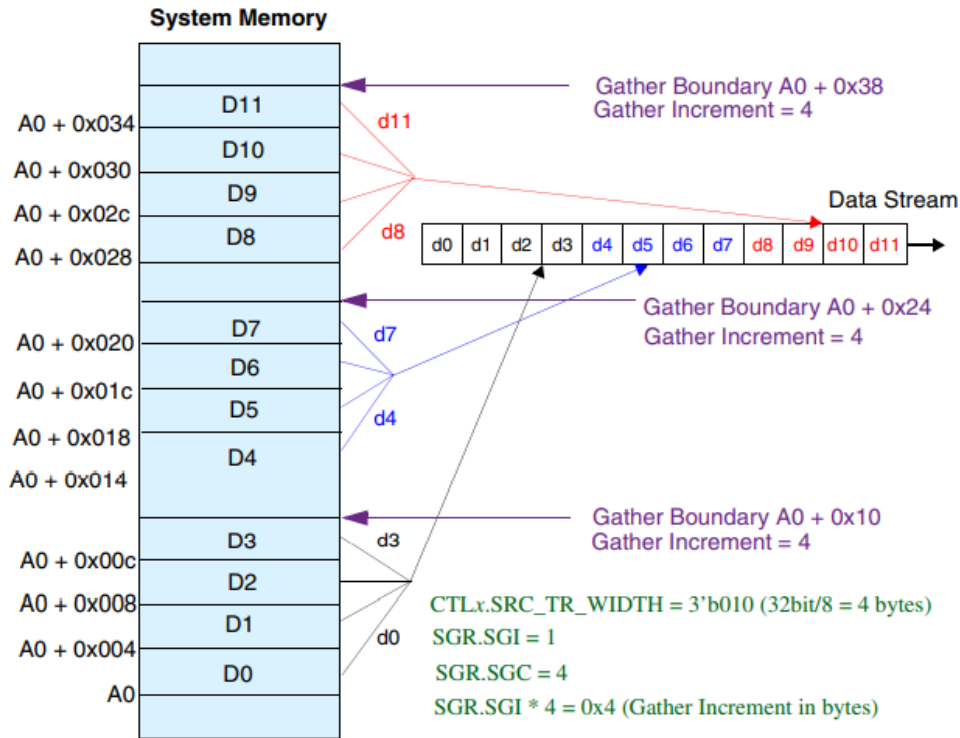


图 12-2 源聚合传输示例图

12.4. DMA 寄存器

12.4.1. DMA 寄存器映像

DMA 寄存器基地址: 0x2002_0000

表格 12-1 DMA 寄存器映像

偏移	寄存器	寄存器描述
0x00, 0x58, 0xB0	SARx	通道 0, 1, 2 源地址寄存器
0x08, 0x60, 0xB8	DARx	通道 0, 1, 2 目标地址寄存器
0x10, 0x68, 0xC0	LLPx	通道 0, 1, 2 链表地址寄存器
0x18, 0x70, 0xC8	CTLx	通道 0, 1, 2 控制寄存器
0x40, 0x98, 0xF0	CFGx	通道 0, 1, 2 配置寄存器
0x48, 0xA0, 0xF8	SGRx	通道 0, 1, 2 源聚合寄存器
0x50, 0xA8, 0x100	DSRx	通道 0, 1, 2 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

偏移	寄存器	寄存器描述
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)

表格 12-2 SARx 寄存器

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)

表格 12-3 DARx 寄存器

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)

表格 12-4 LLPx 寄存器

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

比特	名称	属性	复位值	描述
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)

表格 12-5 CTLx 寄存器

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	目标 Master 选择 0: AHB Master1 1: AHB Master2 2: AHB Master3 3: AHB Master4
22:20	TT_FC	R/W	3'b000	传输类型和流控制器选择 0: 内存 到 内存, DMA 控制 1: 内存 到 外设, DMA 控制 2: 外设 到 内存, DMA 控制 3: 外设 到 外设, DMA 控制
19	Reserved	R/W	0x0	Reserved
18	DST_SCATTER_EN	R/W	0x0	目的地离散使能。 0x0: 未使能。

比特	名称	属性	复位值	描述
				0x1: 使能。
17	SRC_GATHER_EN	R/W	0x0	源聚合使能。 0x0: 未使能。 0x1: 使能。
16:14	SRC_MSIZE	R/W	3'b000	源 Burst 传输长度选择。 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)。 0x0: 待传输的数据项为 1。 0x1: 待传输的数据项为 4。 0x2: 待传输的数据项为 8。 0x3: 待传输的数据项为 16。 0x4: 待传输的数据项为 32。 0x5: 待传输的数据项为 64。 注: SRC_MSIZE 应与 DEST_MSIZE 配置同。
13:11	DEST_MSIZE	R/W	3'b000	目标 Burst 传输长度选择。 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)。 0x0: 待传输的数据项为 1。 0x1: 待传输的数据项为 4。 0x2: 待传输的数据项为 8。 0x3: 待传输的数据项为 16。 0x4: 待传输的数据项为 32。 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

比特	名称	属性	复位值	描述
				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)

表格 12-6 CFGx 寄存器

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。 当目标地址是外设时，设置此位。ID 值由用户自定义。 与系统寄存器 DMA_REQ_MUX0， DMA_REQ_MUX1， DMA_REQ_MUX2，设置的 ID 相同即可。
42:39	SRC_PER	R/W	4'h0	源外设 ID。 当源地址是外设时，设置此位。ID 值由用户自定义。 与系统寄存器 DMA_REQ_MUX0， DMA_REQ_MUX1，

比特	名称	属性	复位值	描述
				DMA_REQ_MUX2, 设置的 ID 相同即可。
38:34	Reserved	R/W	0x0	Reserved
33	FIFO_MODE	R/W	1'b0	0x0: FIFO_MODE_0, 可用于指定传输宽度的单个 AHB 传输的空间/数据。 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
32:20	Reserved	R/W	0x0	Reserved
19	SRC_HS_POL	R/W	1'b0	源外设握手接口极性。 0x0: 源握手信号高有效 0x1: 源握手信号低有效 当源地址为外设时要设置此位。
18	DST_HS_POL	R/W	1'b0	目标外设握手接口极性。 0x0: 目标握手信号高有效 0x1: 目标握手信号低有效 当目标地址为外设时要设置此位。
17:12	Reserved	R/W	0x0	Reserved
11	HS_SEL_SRC	R/W	1'b0	源握手信号选择。 0x0: 硬件握手 0x1: 软件握手 当源地址为存储器时, 忽略此位。
10	HS_SEL_DST	R/W	1'b0	目标握手信号选择。 0x0: 硬件握手 0x1: 软件握手 当目标地址为存储器时, 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。

比特	名称	属性	复位值	描述
				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)

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)

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)

表格 12-7 RawTfr 寄存器

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)

表格 12-8 RawErr 寄存器

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)

表格 12-9 StatusTfr 寄存器

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)

表格 12-10 StatusErr 寄存器

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)

表格 12-11 MaskTfr 寄存器

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	对应通道传输完成中断状态掩码。 每 bit 对应一个通道，bit0 对应通道 0。 StatusTfr 寄存器的内容被 MaskTfr 寄存器的内容屏蔽。 0x0: 屏蔽中断状态 0x1: 不屏蔽中断状态

12.4.14. MaskErr (Offset 0x330)

表格 12-12 MaskErr 寄存器

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 MaskErr 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	对应通道传输错误中断状态掩码。 每 bit 对应一个通道，bit0 对应通道 0。 StatusErr 寄存器的内容被 MaskErr 寄存器的内容屏蔽。 0x0: 屏蔽中断状态 0x1: 不屏蔽中断状态

12.4.15. ClearTfr (Offset 0x338)

表格 12-13 ClearTfr 寄存器

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)

表格 12-14 ClearErr 寄存器

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)

表格 12-15 StatusInt 寄存器

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)

表格 12-16 DmaCfgReg 寄存器

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)

表格 12-17 ChEnReg 寄存器

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 MUX0](#)、[DMA REQ MUX1](#)、[DMA REQ MUX2](#) 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。

除了通用 PWM 模式，还有一种“PWM_DAC”模式，它创建一个很小的周期脉冲宽度。当与低通滤波器(如简单的 RC 电路)一起使用时，DAC 可以产生比标准 PWM 算法更好的带宽和纹波性能。这种类型的 DAC 非常适合于电源输出水平的微调。

13.2. 主要特性

- 8 路 PWM 通道输出或 8 路输入捕获
- 两种 PWM 输出模式可选（PWM 模式，PWM_DAC 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

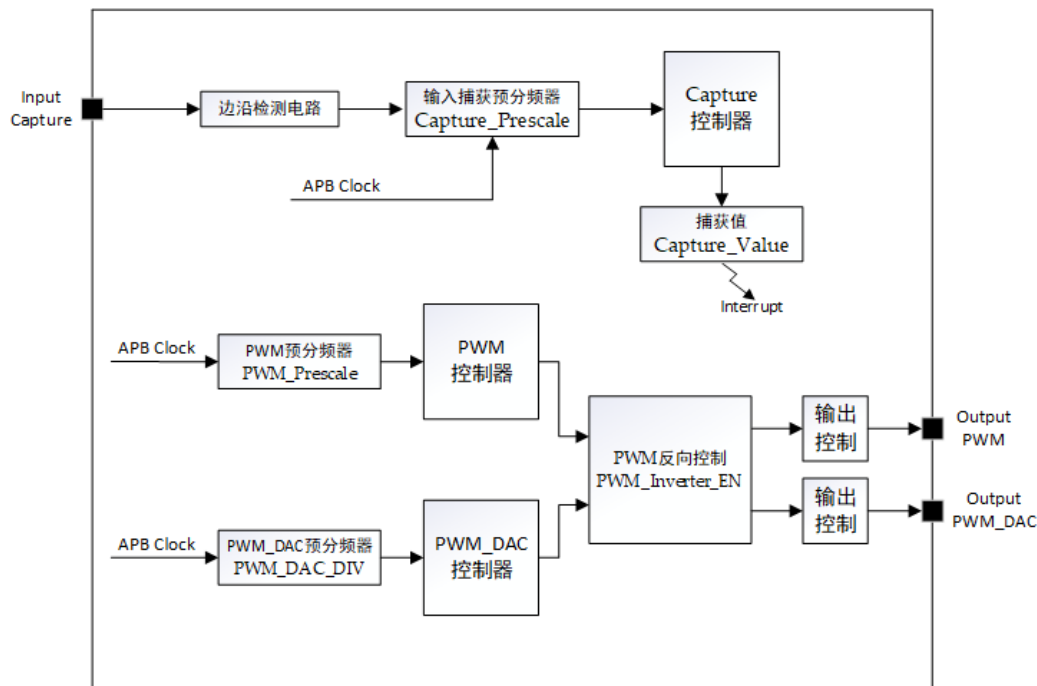


图 13-1 PWM 系统框图

13.4. PWM 寄存器

13.4.1. PWM 寄存器映像

PWM 寄存器基地址: 0x5008_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 ~ 7 上升沿位置寄存器 PWM 通道 0 ~ 7 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	

偏移	寄存器	寄存器描述
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	
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 ~ 7 预分频寄存器 PWM 通道 0 ~ 7 周期寄存器
0x104	PWM0_Period	
0x108	PWM1_Prescale	
0x10C	PWM1_Period	
0x110	PWM2_Prescale	
0x114	PWM2_Period	

偏移	寄存器	寄存器描述
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	
0x180	PWM0_DAC_DIV	PWM 通道 0~7 分频寄存器 (只有在 DAC 模式下有效)
0x184	PWM1_DAC_DIV	
0x188	PWM2_DAC_DIV	
0x18C	PWM3_DAC_DIV	
0x190	PWM4_DAC_DIV	
0x194	PWM5_DAC_DIV	
0x198	PWM6_DAC_DIV	
0x19C	PWM7_DAC_DIV	

13.4.2. PWM_Output_EN (Offset 0x04)

表格 13-2 PWM_Output_EN 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_Output_EN	R/W	8'hFF	PWM 输出使能: 1bit 对应 1 通道。 0: Enable 1: Disable

13.4.3. PWM_ENABLE (Offset 0x08)

表格 13-3 PWM_ENABLE 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_Enable	R/W	8'h00	通道使能。 1bit 对应 1 通道。 0: Disable 1: Enable

13.4.4. PWM0~7_Posedge (Offset 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48)

表格 13-4 PWM0~7_Posedge 寄存器

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 ~ PWM7_Posedge 寄存器描述相同。

13.4.5. PWM0~7_Negedge (Offset 0x14, 0x1C, 0x24, 0x2C, 0x34, 0x3C, 0x44, 0x4C)

表格 13-5 PWM0~7_Negedge 寄存器

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。 PWM_DAC 模式：高电平时间占周期的百分比。 0%~100% (0 ~ 0xFFFFFFFF)。

注：PWM0_Negedge ~ PWM7_Negedge 寄存器描述相同。

13.4.6. PWM_Output_Select (Offset 0x90)

表格 13-6 PWM_Output_Select 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_Output_Select	R/W	8'h00	PWM 输出控制选择。 1bit 对应 1 通道。 0: 使用 PWM_Posedge 和 PWM_Negedge 控制

比特	名称	属性	复位值	描述
				输出波形 1: 使用 PWM Output Value 控制输出波形。

13.4.7. Capture_Prescale (Offset 0x94)

表格 13-7 Capture_Prescale 寄存器

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 = divide by 1 (default) 0001 = divide by 2 0010 = divide by 4 0011 = divide by 8 0100 = divide by 16 0101 = divide by 32 0110 = divide by 64 0111 = divide by 128 1000 = divide by 256 1001 = divide by 512 1010 = divide by 1024 1011 = divide by 2048 Others = divide by 2048

13.4.8. Capture_Status (Offset 0x98)

表格 13-8 Capture_Status 寄存器

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)

表格 13-9 Capture_INT_EN 寄存器

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)

表格 13-10 Capture_CTL 寄存器

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)

表格 13-11 Capture_Value0~7 寄存器

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)

表格 13-12 PWM_Update 寄存器

比特	名称	属性	复位值	描述
31:24	Reserved	R/W	8'h00	Reserved
23:16	PWM_Status	R	8'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
15:8	Reserved	R/W	8'h00	Reserved
7:0	PWM_Update	R/W	8'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_DAC_Mode (Offset 0xE8)

表格 13-13 PWM_DAC_Mode 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	DAC_Mode_EN	R/W	8'h00	输出选择为 PWM 模式或 PWM_DAC 模式。 1bit 对应 1 通道。 0: PWM 模式 1: PWM_DAC 模式

13.4.14. PWM_Output_Value (Offset 0xF0)

表格 13-14 PWM_Output_Value 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_Output_Value	R/W	8'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
--	--	--	--	---

13.4.15. PWM_Inverter_EN (Offset 0xF8)

表格 13-15 PWM_Inverter_EN 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_Inverter_EN	R/W	8'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

13.4.16. PWM_CNT_EN (Offset 0xFC)

表格 13-16 PWM_CNT_EN 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	24'h00	Reserved
7:0	PWM_CNT_EN	R/W	8'hFF	PWM 周期、预分频计数器使能。 1bit 对应 1 通道。 0: Disable 1: Enable

13.4.17. PWM0~7_Prescale (Offset 0x100, 0x108, 0x110, 0x118, 0x120, 0x128, 0x130, 0x138)

表格 13-17 PWM0~7_Prescale 寄存器

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。 $PWMx_Clock = APB_Clock / PWM_Prescale + 1$

PWM0_Prescale ~ PWM7_Prescale 寄存器描述相同。

13.4.18. PWM0~7_Period (Offset 0x104, 0x10C, 0x114, 0x11C, 0x124, 0x12C, 0x134, 0x13C)

表格 13-18 PWM0~7_Period 寄存器

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 ~ PWM7_Period 寄存器描述相同。

13.4.19. PWM0~7_DAC_DIV (Offset 0x180, 0x184, 0x188, 0x18C, 0x190, 0x194, 0x198, 0x19C)

表格 13-19 PWM0~7_DAC_DIV 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	16'h00	Reserved
15:0	PWM_DIV	R/W	16'h08	PWM_DAC 模式下对应通道时钟分频。 $PWMx_Pulse = APB_Clock / 2 * (PWM_DIV + 1)$ 注： 此寄存器只选择 PWM_DAC 模式时有效。

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：

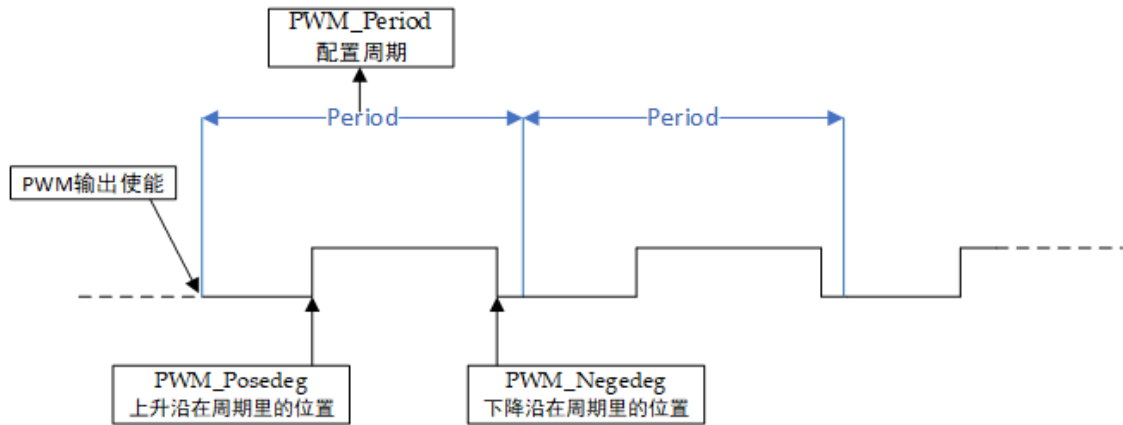


图 13-2 PWM 时序图

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. PWM_DAC 模式

PWM_DAC 模式可以创建一个很小的周期脉冲宽度。

使用 PWM_DAC 模式首先配置 [PWM DAC Mode](#) 寄存器，选择 PWM_DAC 模式。然后配置 [PWM DAC DIV](#) 寄存器设置时钟分频，得到高电平脉冲宽度 PWMx_Pulse。设置 [PWM_Negedge](#) 寄存器设置高电平在整个周期的占比（0%~100%）。

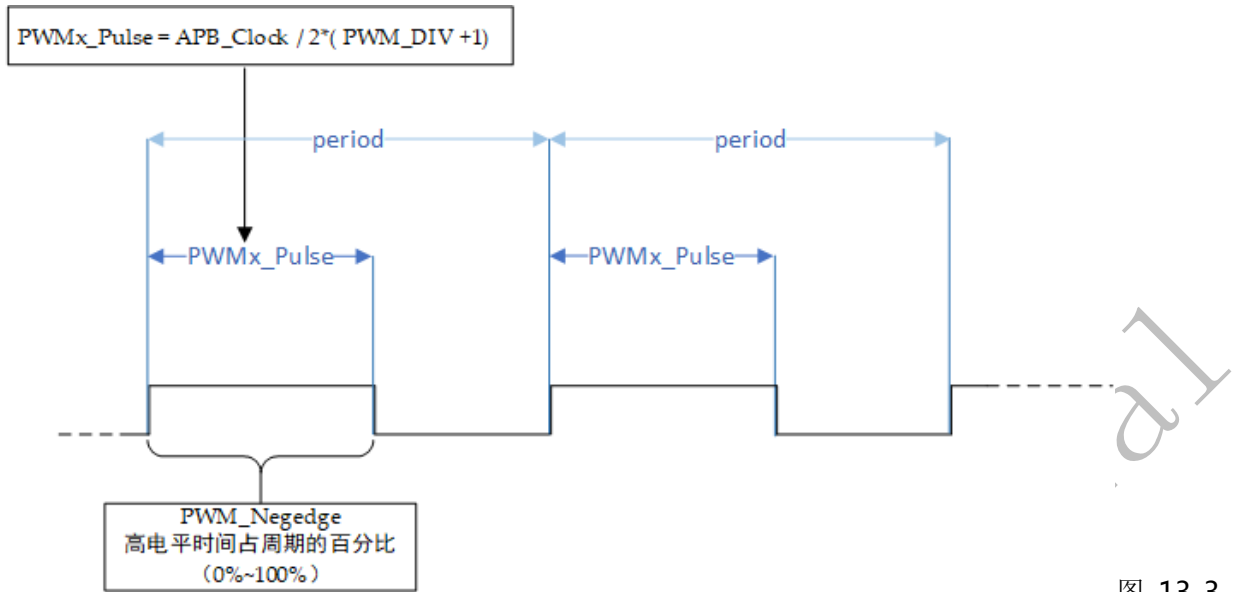


图 13-3

PWM_DAC 时序

Capture 输入捕获功能

FR8000 系列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 $APB_Clock / \text{Capture Prescale}$ 决定。

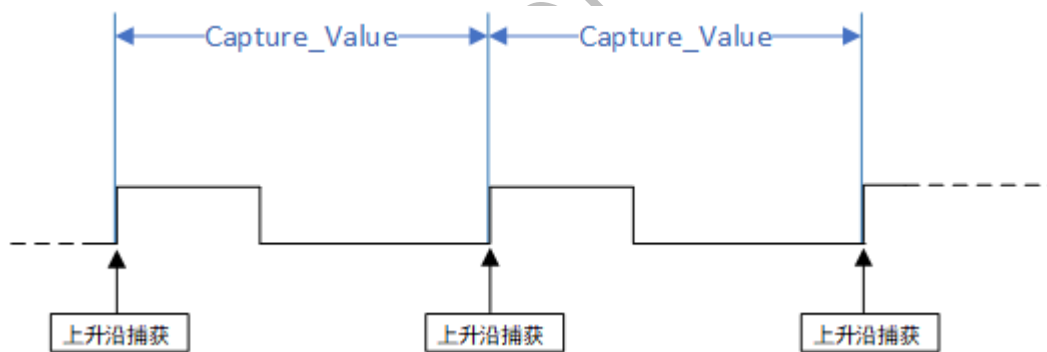


图 13-4 输入捕获时序图

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。

- 上电检测
- 电池充电管理
- 低电压报警
- 低功耗模式下电源模块控制
- SOC 供电模块管理：IO-LDO，DC-DC，DLDO，RFLDO，PKLDO
- 独立看门狗
- RTC
- 矩阵键盘扫描
- PWM
- IO 检测唤醒

14.2. 访问控制接口

PMU 内部存在两个模块，Block0 和 Block1，两个模块上分配了不同功能的寄存器。MCU 通过访问两个模块上的寄存器来控制 PMU 功能。

详细的寄存器地址及分配，请参考后续章节。

14.3. PMU_IWDG

14.3.1. PMU_IWDG 功能描述

IWDG（独立看门狗）使用内部低速时钟（LowPowerRC）。采用 24 位计数器，向下计数。当计数值从 IWDG_LEN 配置值计数到 0 时，产生中断信号。继续从 IWDG_TOUT_LEN 配置值向下计数，若计数到 0 时，产生系统复位。

注：不配置 IWDG_TOUT_LEN 寄存器将只能产生中断，无法产生系统复位。

14.3.2. PMU_IWDG 系统框图

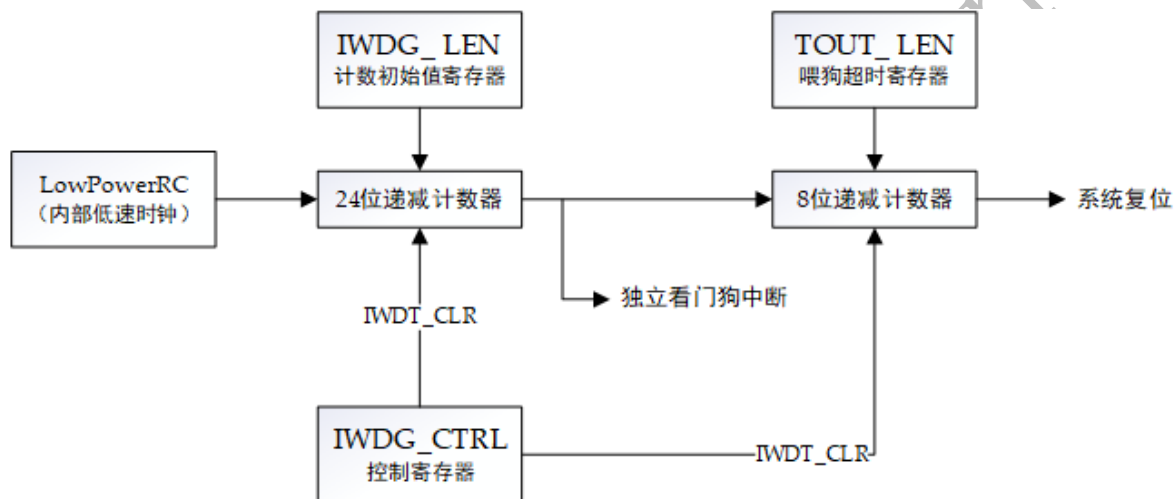


图 14-1 PMU_IWDG 系统框图

14.3.3. PMU_IWDG 寄存器映像

IWDG 寄存器在 PMU 的 **Block0** 区域。

表格 14-1 PMU_IWDG 寄存器映像

地址	寄存器	寄存器描述
0x26	IWDG_CTRL	独立看门狗控制寄存器
0x27	IWDG_LEN_0	独立看门狗计数初始值寄存器（LSB[7:0]）
0x28	IWDG_LEN_1	独立看门狗计数初始值寄存器（[15:8]）
0x29	IWDG_LEN_2	独立看门狗计数初始值寄存器（MSB[23:6]）
0x2A	IWDG_TOUT_LEN	独立看门狗超时寄存器

14.3.4. IWDG_CTRL (Block0, 0x26)

表格 14-2 IWDG_CTRL 寄存器

比特	名称	属性	复位值	描述
7:3	Reserved	R/W	6'h0	Reserved
2	IWDT_IRQ_EN	R/W	1'b0	独立看门狗中断使能
1	IWDT_CLR	R/W	1'b0	清计数，从配置值重新递减计数
0	IWDT_EN	R/W	1'b0	独立看门狗使能

14.3.5. IWDG_LEN (Block0, 0x27, 0x28, 0x29)

表格 14-3 IWDG_LEN 寄存器

比特	名称	属性	复位值	描述
7:0	IWDG_LEN_0 (0x29)	R/W	8'h00	初始计数值 MSB[23:16]
7:0	IWDG_LEN_0 (0x28)	R/W	8'h00	初始计数值[15:8]
7:0	IWDG_LEN_0 (0x27)	R/W	8'h00	初始计数值 LSB[7:0]

14.3.6. IWDG_TOUT_LEN (Block0, 0x2A)

表格 14-4 IWDG_TOUT_LEN 寄存器

比特	名称	属性	复位值	描述
7:0	TOUT_LEN	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

14.4. PMU_GPIO

14.4.1. PMU_GPIO 功能描述

需要使用 PMU 管理 GPIO 引脚时，通过 PMU_GPIO 系列寄存器进行配置。如上下拉配置，输入输出配置，功能复用选择，IO 引脚变化检测功能。

14.4.2. PMU_GPIO 寄存器映像

PMU_GPIO 寄存器在 PMU 的 **Block0** 区域。

表格 14-5 PMU_GPIO 寄存器映像

地址	寄存器	寄存器描述
0x43	PortA_SEL	PortA 控制选择
0x44	PortB_SEL	PortB 控制选择
0x45	PortC_SEL	PortC 控制选择
0x46	PortD_SEL	PortD 控制选择
0x47	PortE_SEL	PortE 控制选择
0x48	PortA_PE	PortA 上下拉使能
0x49	PortB_PE	PortB 上下拉使能
0x4A	PortC_PE	PortC 上下拉使能
0x4B	PortD_PE	PortD 上下拉使能
0x4C	PortE_PE	PortE 上下拉使能
0x4D	PortA_PS	PortA 上下拉选择
0x4E	PortB_PS	PortB 上下拉选择
0x4F	PortC_PS	PortC 上下拉选择
0x50	PortD_PS	PortD 上下拉选择
0x51	PortE_PS	PortE 上下拉选择
0x52	PortA_Monitor_EN	PortA 变化检测使能
0x53	PortB_Monitor_EN	PortB 变化检测使能
0x54	PortC_Monitor_EN	PortC 变化检测使能
0x55	PortD_Monitor_EN	PortD 变化检测使能
0x56	PortE_Monitor_EN	PortE 变化检测使能
0x57	Analog_Monitor_EN	模拟变化检测使能
0x58	PortA_Reference_Status	PortA 变化参考状态

0x59	PortB_Reference_Status	PortB 变化参考状态
0x5A	PortC_Reference_Status	PortC 变化参考状态
0x5B	PortD_Reference_Status	PortD 变化参考状态
0x5C	PortE_Reference_Status	PortE 变化参考状态
0x5D	Analog_Reference_Status	模拟变化参考状态
0x66	PortA_OEN	PortA 输入输出配置
0x67	PortB_OEN	PortB 输入输出配置
0x68	PortC_OEN	PortC 输入输出配置
0x69	PortD_OEN	PortD 输入输出配置
0x6A	PortE_OEN	PortE 输入输出配置
0x6B	PortA_Data	PortA 输出状态寄存器
0x6C	PortB_Data	PortB 输出状态寄存器
0x6D	PortC_Data	PortC 输出状态寄存器
0x6E	PortD_Data	PortD 输出状态寄存器
0x6F	PortE_Data	PortE 输出状态寄存器

14.4.3. PortA/B/C/D/E_SEL (Block0, 0x43, 0x44, 0x45, 0x46, 0x47)

表格 14-6 PortA/B/C/D/E_SEL 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_SEL	R/W	8'hFF	PortA 引脚控制选择: 0: 使用 PMU_GPIO 域寄存器控制 1: 使用系统域 GPIO 寄存器控制 Bit[0]对应 PortA0, 1bit 对应 1 引脚
7:0	PortB_SEL	R/W	8'hFF	同 PortA
7:0	PortC_SEL	R/W	8'hFF	同 PortA
7:0	PortD_SEL	R/W	8'hFF	同 PortA
1:0	PortE_SEL	R/W	2'b11	同 PortA

14.4.4. PortA/B/C/D/E_PE (Block0, 0x48, 0x49, 0x4A, 0x4B, 0x4C)

表格 14-7 PortA/B/C/D/E_PE 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_PE	R/W	8'hFF	PortA 上下拉使能: 0: 禁用上下拉 1: 使能上下拉 Bit[0]对应 PortA0, 1bit 对应 1 引脚

比特	名称	属性	复位值	描述
7:0	PortB_PE	R/W	8'hFF	同 PortA
7:0	PortC_PE	R/W	8'hFF	同 PortA
7:0	PortD_PE	R/W	8'hFF	同 PortA
1:0	PortE_PE	R/W	2'b11	同 PortA

14.4.5. PortA/B/C/D/E_PS (Block0, 0x4D, 0x4E, 0x4F, 0x50, 0x51)

表格 14-8 PortA/B/C/D/E_PS 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_PS	R/W	8'h00	PortA 上下拉选择: 0: 下拉 1: 上拉 Bit[0]对应 PortA0, 1bit 对应 1 引脚
7:0	PortB_PS	R/W	8'h00	同 PortA
7:0	PortC_PS	R/W	8'h00	同 PortA
7:0	PortD_PS	R/W	8'h00	同 PortA
1:0	PortE_PS	R/W	2'b00	同 PortA

14.4.6. PortA/B/C/D_Monitor_EN (Block0, 0x52, 0x53, 0x54, 0x55)

表格 14-9 PortA/B/C/D Monitor EN 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_Monitor_EN	R/W	8'h00	PortA 引脚检测状态变化使能: 0: Enable 1: Disable Bit[0]对应 PortA0, 1bit 对应 1 引脚
7:0	PortB_Monitor_EN	R/W	8'h00	同 PortA
7:0	PortC_Monitor_EN	R/W	8'h00	同 PortA
7:0	PortD_Monitor_EN	R/W	8'h00	同 PortA

14.4.7. PortE_Monitor_EN (Block0, 0x56)

表格 14-10 PortE Monitor EN 寄存器

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R/W	1'b0	Reserved

4	Reserved	R/W	1'b0	Reserved
3	USB_DM Monitor_EN	R/W	1'b0	USB_DM 引脚检测状态变化使能
2	USB_DP Monitor_EN	R/W	1'b0	USB_DP 引脚检测状态变化使能
1:0	PortE_Monitor_EN	R/W	2'b00	PortE 引脚检测状态变化使能： 0: Enable 1: Disable Bit[0]对应 PortE0，1bit 对应 1 引脚

14.4.8. Analog_Monitor_EN (Block0, 0x57)

表格 14-11 Analog_Monitor_EN 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	POFWARN	R/W	1'b0	低电压报警状态，检测变化使能
5	Charging	R/W	1'b0	充电中状态，检测变化使能
4	Reserved	R/W	1'b0	Reserved
3	OTD	R/W	1'b0	OTD 状态，检测变化使能
2	LVD	R/W	1'b0	LVD 状态，检测变化使能
1	ACOK	R/W	1'b0	ACOK 状态，检测变化使能
0	BAT_FULL	R/W	1'b0	BAT 状态，检测变化使能 0: Disable 1: Enable

14.4.9. PortA/B/C/D_Reference_Status (Block0, 0x58, 0x59, 0x5A, 0x5B)

表格 14-12 PortA/B/C/D Reference Status 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_Reference_Status	R/W	8'h00	PortA 引脚状态变化参考值。 写操作：写入参考状态值。 例：写入 0x01。并使能 PA0 引脚检测状态变化寄存器 ，则当检测到 PortA0 由高变低时触发中断。 读操作：引脚当前状态。
7:0	PortB_Reference_Status	R/W	8'h00	同 PortA
7:0	PortC_Reference_Status	R/W	8'h00	同 PortA
7:0	PortD_Reference_Status	R/W	8'h00	同 PortA

14.4.10. PortE_Reference_Status (Block0, 0x5C)

表格 14-13 PortE Reference Status 寄存器

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R/W	1'b0	Reserved
4	Reserved	R/W	1'b0	Reserved
3	USB_DM Reference_Status	R/W	1'b0	USB_DM 引脚状态变化参考值 写：写入参考值。 读：当前状态。
2	USB_DP Reference_Status	R/W	1'b0	USB_DP 引脚状态变化参考值 写：写入参考值。 读：当前状态。
1:0	PortE_Reference_Status	R/W	2'b00	PortE 引脚状态变化参考值。 写操作：写入参考状态值。 例：写入 0x01。并使能 PE0 引脚检测状态变化寄存器，则当检测到 PortE0 由高变低时触发中断。 读操作：引脚当前状态。

14.4.11. Analog_Reference_Status (Block0, 0x5D)

表格 14-14 Analog Reference Status 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	POFWARN Reference_Status	R/W	1'b0	低电压报警参考状态。 写：写入参考值。 读：当前状态。
5	Charging Reference_Status	R/W	1'b0	充电中参考状态 写：写入参考值。 读：当前状态。
4	Reserved	R/W	1'b0	Reserved
3	OTD Reference_Status	R/W	1'b0	OTD 参考状态 写：写入参考值。 读：当前状态。
2	LVD Reference_Status	R/W	1'b0	LVD 参考状态 写：写入参考值。 读：当前状态。
1	ACOK Reference_Status	R/W	1'b0	ACOK 参考状态 写：写入参考值。 读：当前状态。

0	BAT_FULL Reference_Status	R/W	1'b0	BAT_FULL 参考状态 写：写入参考值。 读：当前状态。
---	---------------------------	-----	------	--------------------------------------

14.4.12. PortA/B/C/D/E_OEN (Block0, 0x66, 0x67, 0x68, 0x69, 0x6A)

表格 14-15 PortA/B/C/D/E_OEN 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_OEN	R/W	8'hFF	PortA 输入/输出选择： 0: Output 1: Input Bit[0]对应 PortA0, 1bit 对应 1 引脚
7:0	PortB_OEN	R/W	8'hFF	同 PortA
7:0	PortC_OEN	R/W	8'hFF	同 PortA
7:0	PortD_OEN	R/W	8'hFF	同 PortA
1:0	PortE_OEN	R/W	2'b11	同 PortA

14.4.13. PortA/B/C/D/E_Data (Block0, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F)

表格 14-16 PortA/B/C/D/E_Data 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_Data	R/W	8'h00	PortA 输出状态： 0: 低电平 1: 高电平 Bit[0]对应 PortA0, 1bit 对应 1 引脚
7:0	PortB_Data	R/W	8'h00	同 PortA
7:0	PortC_Data	R/W	8'h00	同 PortA
7:0	PortD_Data	R/W	8'h00	同 PortA
1:0	PortE_Data	R/W	8'h00	同 PortA

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

RTC 寄存器在 PMU 的 **Block1** 区域。

表格 14-17 PMU_RTC 寄存器映像

地址	寄存器	寄存器描述
0x07	RTC_CTRL	RTC 控制寄存器
0x08	RTC_CNT	RTC 计数值[7:0]
0x09	RTC_CNT	RTC 计数值[15:8]
0x0A	RTC_CNT	RTC 计数值[23:16]
0x0B	RTC_CNT	RTC 计数值[31:24]
0x0C	ALARM_A	闹钟 A[7:0]
0x0D	ALARM_A	闹钟 A[15:8]
0x0E	ALARM_A	闹钟 A[23:16]
0x0F	ALARM_A	闹钟 A[31:24]
0x10	ALARM_B	闹钟 B[7:0]
0x11	ALARM_B	闹钟 B[15:8]
0x12	ALARM_B	闹钟 B[23:16]
0x13	ALARM_B	闹钟 B[31:24]

14.5.3. RTC_CTRL (Block1, 0x07)

表格 14-18 RTC_CTRL 寄存器

比特	名称	属性	复位值	描述
7:5	Reserved	R/W	3'b000	Reserved

4	RTC_EN	R/W	1'b0	RTC 使能。 0: 停止 RTC 计数 1: 开始 RTC 计数
3	Alarm_B_EN	R/W	1'b0	闹钟 B 使能。 0: Disable 1: Enable
2	Alarm_A_EN	R/W	1'b0	闹钟 A 使能。 0: Disable 1: Enable
1:0	Reserved	R/W	2'b00	Reserved

14.5.4. RTC_CNT (Block1, 0x08, 0x09, 0x0A, 0x0B)

表格 14-19 RTC_CNT 寄存器

比特	名称	属性	复位值	描述
7:0	RTC_CNT (0x0B)	R/W	8'h00	RTC 计数值[31:24]
7:0	RTC_CNT (0x0A)	R/W	8'h00	RTC 计数值[23:16]
7:0	RTC_CNT (0x09)	R/W	8'h00	RTC 计数值[15:8]
7:0	RTC_CNT (0x08)	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 (Block1, 0x0C, 0x0D, 0x0E, 0x0F)

表格 14-20 ALARM_A 寄存器

比特	名称	属性	复位值	描述
7:0	Alarm_A (0x0F)	R/W	8'h00	闹钟 A 设定值[31:24]
7:0	Alarm_A (0x0E)	R/W	8'h00	闹钟 A 设定值[23:16]
7:0	Alarm_A (0x0D)	R/W	8'h00	闹钟 A 设定值[15:8]
7:0	Alarm_A (0x0C)	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 (Block1, 0x10, 0x11, 0x12, 0x13)

表格 14-21 ALARM_B 寄存器

比特	名称	属性	复位值	描述
7:0	Alarm_B (0x13)	R/W	8'h00	闹钟 B 设定值[31:24]

7:0	Alarm_B (0x12)	R/W	8'h00	闹钟 B 设定值[23:16]
7:0	Alarm_B (0x11)	R/W	8'h00	闹钟 B 设定值[15:8]
7:0	Alarm_B (0x10)	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_KeyScan

14.6.1. PMU_KeyScan 功能描述

矩阵键盘使用内部低速时钟（LowPowerRC），最多支持 8 行，20 列。支持硬件滤波防抖，支持按下、松开中断响应。

14.6.2. PMU_KeyScan 寄存器映像

KeyScan 寄存器在 PMU 的 **Block1** 区域。

表格 14-22 PMU_KeyScan 寄存器映像

地址	寄存器	寄存器描述
0x18	KeyScan_CTL	矩阵键盘扫描控制寄存器
0x19	Scan_Interval	扫描间隔[7:0]
0x1A	Scan_Interval	扫描间隔[9:8]
0x1B	KeyScan_Config	扫描行，滤波配置
0x1C	ROW_SEL	行选择
0x1D	COL_SEL	列选择[7:0]
0x1E	COL_SEL	列选择[15:8]
0x1F	COL_SEL	列选择[19:16]
0x20 ~ 0x33	KEY_VAL	按键值

14.6.3. KeyScan_CTL (Block1, 0x18)

表格 14-23 KeyScan_CTL 寄存器

比特	名称	属性	复位值	描述
7:5	Reserved	R/W	2'b00	Reserved

比特	名称	属性	复位值	描述
4	KEY_Status_CLR	R/W	1'b0	按键状态清除。 当发生按下或抬起动作时置‘1’如使能了中断位，将开始一次中断请求。如不清除状态位，则会频繁的请求中断。
3	GHOST_KEY_EN	R/W	1'b0	GhostKey 处理使能。 检测存在 GhostKey 时，对应按键不会请求中断。
2	Scan_Lowpower_EN	R/W	1'b0	按键扫描低功耗模式。 在没有按键使用时停止列扫描，节省功耗，检测到有键按下时，立刻扫描键值。
1	Filter_EN	R/W	1'b0	滤波使能。 0: Disable 1: Enable
0	KeyScan_EN	R/W	1'b0	矩阵扫描使能。 0: Disable 1: Enable

14.6.4. Scan_Interval (Block1, 0x19, 0x1A)

表格 14-24 Scan_Interval 寄存器

比特	名称	属性	复位值	描述
1:0	Scan_Interval (0x1A)	R/W	2'b00	矩阵键盘扫描间隔[9:8]
7:0	Scan_Interval (0x19)	R/W	8'h00	矩阵键盘扫描间隔[7:0]

14.6.5. KeyScan_Config (Block1, 0x1B)

表格 14-25 KeyScan_Config 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	4'h0	Reserved
6:4	ROW_Select	R/W	3'b000	行配置。 001: PortA 当做行 010: PortB 当做行

				011: PortC 当做行 100: PortD 当做行
3:0	Filter_Value	R/W	4'h0	滤波防抖

14.6.6. ROW_SEL (Block1, 0x1C)

表格 14-26 ROW_SEL 寄存器

比特	名称	属性	复位值	描述
7:0	ROW_SEL	R/W	8'h00	行选择。最多支持 8 行。 1bit 代表 1 行。 例: ROW_Select = PortA 时。 ROW_SEL = 0x0F, 则选择 PA0~3 当做行 0~3。 ROW_SEL = 0x05, 则选择 PA0、PA2 当做行 0、行 2。

14.6.7. COL_SEL (Block1, 0x1D, 0x1E, 0x1F)

表格 14-27 COL_SEL 寄存器

比特	名称	属性	复位值	描述
3:0	COL_SEL (0x1F)	R/W	4'h0	列选择[19:16]。
7:0	COL_SEL (0x1E)	R/W	8'h00	列选择[15:8]。
7:0	COL_SEL (0x1D)	R/W	8'h00	列选择[7:0]。最多支持 20 列。 1bit 代表 1 列。 例: COL_SEL (0x1D) = 0xFF, 则选择了列 0 ~ 7, 使能了 8 列。 列对应引脚详见 附录 II 。

注：不能重复选择列引脚。比如 PA0、PC4 都可以作为 KeyScan_COL[0]，但同时只能选择一个。

14.6.8. KEY_VAL (Block1, 0x20 ~ 0x33)

表格 14-28 KEY_VAL 寄存器

比特	名称	属性	复位值	描述
7:0	KEY_VAL (0x20 ~ 0x33)	R/W	8'h00	按键值，1bit 对应一个按键。

14.7. PMU_PWM

14.7.1. PMU_PWM 功能描述

PMU 域提供了 4 路 PWM，使用内部低速时钟（LowPowerRC）。

14.7.2. PMU_PWM 寄存器映像

PMU_PWM 寄存器在 PMU 的 **Block1** 区域。

表格 14-29 PMU_PWM 寄存器映像

地址	寄存器	寄存器描述
0x34	PWM0_CTRL	PWM0 控制寄存器
0x35	PWM1_CTRL	PWM1 控制寄存器
0x36	PWM2_CTRL	PWM2 控制寄存器
0x37	PWM3_CTRL	PWM3 控制寄存器
0x38	PWM0_PRC	PWM0 周期计数[7:0]
0x39	PWM0_PRC	PWM0 周期计数[15:8]
0x3A	PWM0_HRC	PWM0 高电平计数[7:0]
0x3B	PWM0_HRC	PWM0 高电平计数[15:8]
0x3C	PWM1_PRC	PWM1 周期计数[7:0]
0x3D	PWM1_PRC	PWM1 周期计数[15:8]
0x3E	PWM1_HRC	PWM1 高电平计数[7:0]
0x3F	PWM1_HRC	PWM1 高电平计数[15:8]
0x40	PWM2_PRC	PWM2 周期计数[7:0]
0x41	PWM2_PRC	PWM2 周期计数[15:8]
0x42	PWM2_HRC	PWM2 高电平计数[7:0]
0x43	PWM2_HRC	PWM2 高电平计数[15:8]
0x44	PWM3_PRC	PWM3 周期计数[7:0]
0x45	PWM3_PRC	PWM3 周期计数[15:8]
0x46	PWM3_HRC	PWM3 高电平计数[7:0]
0x47	PWM3_HRC	PWM3 高电平计数[15:8]

14.7.3. PWM0 ~ 3_CTRL (Block1, 0x34, 0x35, 0x36, 0x37)

表格 14-30 PWM0~3_CTRL 寄存器

比特	名称	属性	复位值	描述
7:5	Reserved	R/W	3'b000	Reserved
4	PWM_INV	R/W	1'b0	反相输出使能。 0: 正常输出 1: 反相输出
3	PWM_RST	R/W	1'b0	PWM 计数值复位。 0: 停止复位 1: 计数值复位
2	PWM_MODE	R/W	1'b0	模式选择 0: 循环模式 1: 单次模式
1	PWM_OUT_EN	R/W	1'b0	PWM 输出使能 0: Output Disable 1: Output Enable
0	PWM_EN	R/W	1'b0	PWM 使能 0: Disable 1: Enable

注：4 路 PWM 控制器内容相同，PWM0~3_CTRL 参考上述内容。

14.7.4. PWM0_PRC (Block1, 0x38, 0x39)

14.7.5. PWM0_HRC (Block1, 0x3A, 0x3B)

14.7.6. PWM1_PRC (Block1, 0x3C, 0x3D)

14.7.7. PWM1_HRC (Block1, 0x3E, 0x3F)

14.7.8. PWM2_PRC (Block1, 0x40, 0x41)

14.7.9. PWM2_HRC (Block1, 0x42, 0x43)

14.7.10. PWM3_PRC (Block1, 0x44, 0x45)

14.7.11. PWM3_HRC (Block1, 0x46, 0x47)

表格 14-31 PWM0~3_PRC 寄存器

比特	名称	属性	复位值	描述
7:0	PWM3_PRC[15:8] (0x45)	R/W	8'h00	PWM3 周期计数时间，高八位。
7:0	PWM3_PRC[7:0] (0x44)	R/W	8'h00	PWM3 周期计数时间，低八位。

表格 14-32 PWM0~3_HRC 寄存器

比特	名称	属性	复位值	描述
7:0	PWM3_HRC[15:8] (0x47)	R/W	8'h00	PWM3 高电平保持时间，高八位。

7:0	PWM3_HRC[7:0] (0x46)	R/W	8'h00	PWM3 高电平保持时间，低八位。
-----	----------------------	-----	-------	-------------------

注：PWM 周期 = PWM3_PRC + 1。高电平保持时间 = PWM 周期 - PWM3_HRC

注：4 路 PWM PRC、HRC 内容相同，参考上述配置。

14.8. PMU_IOMux

14.8.1. 功能描述

当使用 PMU 控制 GPIO 时，可配置 PMU 功能的引脚复用，如 PWM 输出，KeyScan 矩阵键盘扫描。

14.8.2. PMU_IOMux 寄存器映像

PMU_IOMux 寄存器在 PMU 的 **Block1** 区域。

表格 14-33 PMU_IOMux 寄存器映像

地址	寄存器	寄存器描述
0x6A	PortA_IOMux	PA0~PA3 引脚复用功能选择
0x6B	PortA_IOMux	PA4~PA7 引脚复用功能选择
0x6C	PortB_IOMux	PB0~PB3 引脚复用功能选择
0x6D	PortB_IOMux	PB4~PB7 引脚复用功能选择
0x6E	PortC_IOMux	PC0~PC3 引脚复用功能选择
0x6F	PortC_IOMux	PC4~PC7 引脚复用功能选择
0x70	PortD_IOMux	PD0~PD3 引脚复用功能选择
0x71	PortD_IOMux	PD4~PD7 引脚复用功能选择
0x72	PortE_IOMux	PE0~PE1 引脚复用功能选择

14.8.3. PortA_IOMux (Block1, 0x6A, 0x6B)

表格 14-34 PortA IOMux 寄存器

比特	名称	属性	复位值	描述
7:6	PA7_IOMux (0x6B)	R/W	2'b00	PA7 在 PMU 中的复用配置
5:4	PA6_IOMux (0x6B)	R/W	2'b00	PA6 在 PMU 中的复用配置
3:2	PA5_IOMux (0x6B)	R/W	2'b00	PA5 在 PMU 中的复用配置

比特	名称	属性	复位值	描述
1:0	PA4_IOMux (0x6B)	R/W	2'b00	PA4 在 PMU 中的复用配置
7:6	PA3_IOMux (0x6A)	R/W	2'b00	PA3 在 PMU 中的复用配置
5:4	PA2_IOMux (0x6A)	R/W	2'b00	PA2 在 PMU 中的复用配置
3:2	PA1_IOMux (0x6A)	R/W	2'b00	PA1 在 PMU 中的复用配置
1:0	PA0_IOMux (0x6A)	R/W	2'b00	PA0 在 PMU 中的复用配置

具体功能详见[附录 II](#)。

14.8.4. PortB_IOMux (Block1, 0x6C, 0x6D)

表格 14-35 PortB IOMux 寄存器

比特	名称	属性	复位值	描述
7:6	PB7_IOMux (0x6D)	R/W	2'b00	PB7 在 PMU 中的复用配置
5:4	PB6_IOMux (0x6D)	R/W	2'b00	PB6 在 PMU 中的复用配置
3:2	PB5_IOMux (0x6D)	R/W	2'b00	PB5 在 PMU 中的复用配置
1:0	PB4_IOMux (0x6D)	R/W	2'b00	PB4 在 PMU 中的复用配置
7:6	PB3_IOMux (0x6C)	R/W	2'b00	PB3 在 PMU 中的复用配置
5:4	PB2_IOMux (0x6C)	R/W	2'b00	PB2 在 PMU 中的复用配置
3:2	PB1_IOMux (0x6C)	R/W	2'b00	PB1 在 PMU 中的复用配置
1:0	PB0_IOMux (0x6C)	R/W	2'b00	PB0 在 PMU 中的复用配置

具体功能详见[附录 II](#)。

14.8.5. PortC_IOMux (Block1, 0x6E, 0x6F)

表格 14-36 PortC IOMux 寄存器

比特	名称	属性	复位值	描述
7:6	PC7_IOMux (0x6F)	R/W	2'b00	PC7 在 PMU 中的复用配置
5:4	PC6_IOMux (0x6F)	R/W	2'b00	PC6 在 PMU 中的复用配置
3:2	PC5_IOMux (0x6F)	R/W	2'b00	PC5 在 PMU 中的复用配置
1:0	PC4_IOMux (0x6F)	R/W	2'b00	PC4 在 PMU 中的复用配置
7:6	PC3_IOMux (0x6E)	R/W	2'b00	PC3 在 PMU 中的复用配置
5:4	PC2_IOMux (0x6E)	R/W	2'b00	PC2 在 PMU 中的复用配置
3:2	PC1_IOMux (0x6E)	R/W	2'b00	PC1 在 PMU 中的复用配置

比特	名称	属性	复位值	描述
1:0	PC0_IOMux (0x6E)	R/W	2'b00	PC0 在 PMU 中的复用配置

具体功能详见[附录 II](#)。

14.8.6. PortD_IOMux (Block1, 0x70, 0x71)

表格 14-37 PortD IOMux 寄存器

比特	名称	属性	复位值	描述
7:6	PD7_IOMux (0x71)	R/W	2'b00	PD7 在 PMU 中的复用配置
5:4	PD6_IOMux (0x71)	R/W	2'b00	PD6 在 PMU 中的复用配置
3:2	PD5_IOMux (0x71)	R/W	2'b00	PD5 在 PMU 中的复用配置
1:0	PD4_IOMux (0x71)	R/W	2'b00	PD4 在 PMU 中的复用配置
7:6	PD3_IOMux (0x70)	R/W	2'b00	PD3 在 PMU 中的复用配置
5:4	PD2_IOMux (0x70)	R/W	2'b00	PD2 在 PMU 中的复用配置
3:2	PD1_IOMux (0x70)	R/W	2'b00	PD1 在 PMU 中的复用配置
1:0	PD0_IOMux (0x70)	R/W	2'b00	PD0 在 PMU 中的复用配置

具体功能详见[附录 II](#)。

14.8.7. PortE_IOMux (Block1, 0x72)

表格 14-38 PortE IOMux 寄存器

比特	名称	属性	复位值	描述
7:4	Reserved	R/W	4'h0	Reserved
3:2	PE1_IOMux (0x72)	R/W	2'b00	PE1 在 PMU 中的复用配置
1:0	PE0_IOMux (0x72)	R/W	2'b00	PE0 在 PMU 中的复用配置

具体功能详见[附录 II](#)。

14.9. PMU_INT_CTRL

14.9.1. 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；BAT_Full、ACOK、LVD、OTD 信号滤波配置。

BAT_Full: 充电满

ACOK: 充电线插入

LVD: 低电压预警

OTD: 高温预警

14.9.2. PMU_INT_CTRL 寄存器映像

PMU_INT_CTRL 寄存器在 PMU 的 **Block1** 区域。

表格 14-39 PMU_INT_CTRL 寄存器映像

地址	寄存器	寄存器描述
0x58	BAT_Full_Filter	BAT_Full 信号滤波配置
0x59	ACOK_Filter	ACOK 信号滤波配置
0x5A	LVD_Filter	LVD 信号滤波配置
0x5B	OTD_Filter	OTD 信号滤波配置
0x5E	INT_EN0	PMU 中断使能寄存器 0
0x5F	INT_EN1	PMU 中断使能寄存器 1
0x60	INT_CLR0	PMU 中断清除寄存器 0
0x61	INT_CLR1	PMU 中断清除寄存器 0
0x62	INT_STS0	PMU 中断状态寄存器 0
0x63	INT_STS1	PMU 中断状态寄存器 0
0x64	PortA_Monitor_Status	PortA 检测引脚状态变化, 结果值寄存器
0x65	PortB_Monitor_Status	PortB 检测引脚状态变化, 结果值寄存器
0x66	PortC_Monitor_Status	PortC 检测引脚状态变化, 结果值寄存器
0x67	PortD_Monitor_Status	PortD 检测引脚状态变化, 结果值寄存器
0x68	PortE_Monitor_Status	PortE 检测引脚状态变化, 结果值寄存器
0x69	ANA_Monitor_Status	模拟部分 检测状态变化, 结果值寄存器

14.9.3. BAT_Full_Filter (Block1, 0x58)

表格 14-40 BAT_Full_Filter 寄存器

比特	名称	属性	复位值	描述
7:0	BAT_Full_Filer	R/W	8'h00	充电电量满状态检测滤波。 滤波时钟使用 LowPowerRC 滤波时间 = BAT_Full_Filer * 0xFF + 0x40 个

				时钟周期。
--	--	--	--	-------

14.9.4. ACOK_Filter (Block1, 0x59)

表格 14-41 ACOK_Filter 寄存器

比特	名称	属性	复位值	描述
7:0	ACOK_Filter	R/W	8'h00	充电线插入状态检测滤波。 滤波时钟使用 LowPowerRC 滤波时间 = ACOK_Filter * 0xFF + 0x40 个时钟周期。

14.9.5. LVD_Filter (Block1, 0x5A)

表格 14-42 LVD_Filter 寄存器

比特	名称	属性	复位值	描述
7:0	LVD_Filter	R/W	8'h00	低电压状态检测滤波。 滤波时钟使用 LowPowerRC 滤波时间 = LVD_Filter * 0xFF + 0x40 个时钟周期。

14.9.6. OTD_Filter (Block1, 0x5B)

表格 14-43 OTD_Filter 寄存器

比特	名称	属性	复位值	描述
7:0	OTD_Filter	R/W	8'h00	高温状态检测滤波。 滤波时钟使用 LowPowerRC 滤波时间 = OTD_Filter * 0xFF + 0x40 个时钟周期。

14.9.7. INT_EN0 (Block1, 0x5E)

表格 14-44 INT_EN0 寄存器

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

7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R/W	1'b0	Reserved
4	Reserved	R/W	1'b0	Reserved
3	OTD_INT_EN	R/W	1'b0	OTD 状态，检测变化中断使能 功能详见 Analog Monitor EN 章节 功能详见 Analog Reference Status 章节
2	LVD_INT_EN	R/W	1'b0	LVD 状态，检测变化中断使能 功能详见 Analog Monitor EN 章节 功能详见 Analog Reference Status 章节
1	ACOK_INT_EN	R/W	1'b0	ACOK 状态，检测变化中断使能 功能详见 Analog Monitor EN 章节 功能详见 Analog Reference Status 章节
0	BAT_FULL_INT_EN	R/W	1'b0	BAT 状态，检测变化中断使能 功能详见 Analog Monitor EN 章节 功能详见 Analog Reference Status 章节

14.9.8. INT_EN1 (Block1, 0x5F)

表格 14-45 INT_EN1 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	IWDG_INT_EN	R/W	1'b0	独立看门狗中断使能
5	GPIO_Monitor_INT_EN	R/W	1'b0	GPIO 引脚状态检测变化中断使能
4	Reserved	R/W	1'b0	Reserved
3	KEY_SCAN_INT_EN	R/W	1'b0	键盘扫描中断使能
2	AlarmB_INT_EN	R/W	1'b0	闹钟 B 中断使能
1	AlarmA_INT_EN	R/W	1'b0	闹钟 A 中断使能
0	Reserved	R/W	1'b0	Reserved

14.9.9. INT_CLR0 (Block1, 0x60)

表格 14-46 INT_CLR0 寄存器

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R/W	1'b0	Reserved
4	Reserved	R/W	1'b0	Reserved
3	OTD_INT_CLR	R/W	1'b0	OTD 中断清除
2	LVD_INT_CLR	R/W	1'b0	LVD 中断清除
1	ACOK_INT_CLR	R/W	1'b0	ACOK 中断清除
0	BAT_FULL_INT_CLR	R/W	1'b0	BAT 中断清除。 此寄存器所有位，先写 '1' 清除状态。经 10us 延时后再写 '0'。

14.9.10. INT_CLR1 (Block1, 0x61)

表格 14-47 INT_CLR1 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	IWDG_INT_CLR	R/W	1'b0	独立看门狗中断清除
5	GPIO_Monitor_INT_CLR	R/W	1'b0	GPIO 引脚状态检测变化中断清除
4	Reserved	R/W	1'b0	Reserved
3	KEY_SCAN_INT_CLR	R/W	1'b0	键盘扫描中断清除
2	AlarmB_INT_CLR	R/W	1'b0	闹钟 B 中断清除
1	AlarmA_INT_CLR	R/W	1'b0	闹钟 A 中断清除 此寄存器所有位，先写 '1' 清除状态。经 10us 延时后再写 '0'。
0	Reserved	R/W	1'b0	Reserved

14.9.11. INT_STS0 (Block1, 0x62)

表格 14-48 INT_STS0 寄存器

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R	1'b0	Reserved
4	Reserved	R	1'b0	Reserved
3	OTD_INT_STS	R	1'b0	OTD 中断状态
2	LVD_INT_STS	R	1'b0	LVD 中断状态
1	ACOK_INT_STS	R	1'b0	ACOK 中断状态
0	BAT_FULL_INT_STS	R	1'b0	BAT 中断状态。

14.9.12. INT_STS1 (Block1, 0x63)

表格 14-49 INT_STS1 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	IWDG_INT_STS	R/W	1'b0	独立看门狗中断状态
5	GPIO_Monitor_INT_STS	R/W	1'b0	GPIO 引脚状态检测变化中断状态
4	Reserved	R/W	1'b0	Reserved
3	KEY_SCAN_INT_STS	R/W	1'b0	键盘扫描中断状态
2	AlarmB_INT_STS	R/W	1'b0	闹钟 B 中断状态
1	AlarmA_INT_STS	R/W	1'b0	闹钟 A 中断状态
0	Reserved	R/W	1'b0	Reserved

14.9.13. PortA/B/C/D_Monitor_Status (Block1, 0x64, 0x65, 0x66, 0x67)

表格 14-50 PortA/B/C/D_Monitor_Status 寄存器

比特	名称	属性	复位值	描述
7:0	PortA_Monitor_Status	R/W	8'h00	PortA 引脚检测状态: 0: 未发生改变 1: 发生改变 Bit[0]对应 PortA0, 1bit 对应 1 引脚

7:0	PortB_Monitor_Status	R/W	8'h00	同 PortA
7:0	PortC_Monitor_Status	R/W	8'h00	同 PortA
7:0	PortD_Monitor_Status	R/W	8'h00	同 PortA

14.9.14. PortE_Monitor_Status (Block1, 0x68)

表格 14-51 PortE_Monitor_Status 寄存器

比特	名称	属性	复位值	描述
7:6	Reserved	R/W	2'b00	Reserved
5	Reserved	R/W	1'b0	Reserved
4	Reserved	R/W	1'b0	Reserved
3	USB_DM_Monitor_Status	R/W	1'b0	0: USB_DM 引脚检测状态未发生变化 1: USB_DM 引脚检测状态发生变化
2	USB_DP_Monitor_Status	R/W	1'b0	0: USB_DP 引脚检测状态未发生变化 1: USB_DP 引脚检测状态发生变化
1:0	PortE_Monitor_Status	R/W	2'b00	PortE 引脚检测状态: 0: 未发生改变 1: 发生改变 Bit[0]对应 PortE0, 1bit 对应 1 引脚

14.9.15. ANA_Monitor_Status (Block1, 0x69)

表格 14-52 ANA_Monitor_Status 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R/W	1'b0	Reserved
6	POFWARN	R/W	1'b0	低电压报警状态, 检测变化。 0: 未发生改变 1: 发生改变
5	Charging	R/W	1'b0	充电中状态, 检测变化。 0: 未发生改变 1: 发生改变
4	Reserved	R/W	1'b0	Reserved
3	OTD	R/W	1'b0	OTD 状态, 检测变化。 0: 未发生改变 1: 发生改变
2	LVD	R/W	1'b0	LVD 状态, 检测变化。 0: 未发生改变

				1: 发生改变
1	ACOK	R/W	1'b0	ACOK 状态, 检测变化。 0: 未发生改变 1: 发生改变
0	BAT_FULL	R/W	1'b0	BAT 状态, 检测变化。 0: 未发生改变 1: 发生改变

15. 液晶屏驱动接口（LCD）

15.1. 概述

LCD 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。
- 支持 VSYNC、HSYNC 检测。
- 总线 8BIT、16BIT 可选。

15.3. LCD 结构框图

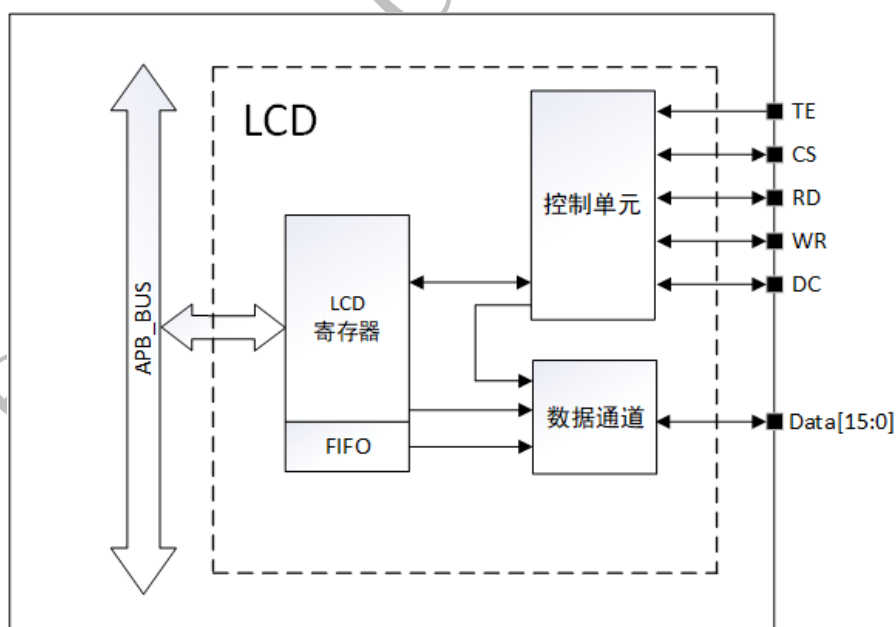


图 15-1 LCD 结构框图

15.4. LCD 寄存器

15.4.1. LCD 寄存器地址映像

LCD 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	LCD_TXFF_PUSH	Tx FIFO 寄存器
0x04	LCD_CFG	配置寄存器
0x08	LCD_CFGWR_STATUS	写状态寄存器
0x0C	LCD_CFGRD_STATUS	读状态寄存器
0x10	LCD_APB_RD	读数据寄存器
0x14	LCD_TXFF_AEMP_LV	Tx FIFO 剩余数据深度
0x18	LCD_DAT_WR_LEN	写数据长度寄存器
0x1C	LCD_DAT_WR_CFG	数据配置寄存器
0x20	LCD_DMA	DMA 配置寄存器
0x24	LCD_CSX	CS 线控制寄存器
0x28	LCD_CRM	读写时钟配置
0x2C	LCD_TXFF_CLR	Tx FIFO 清除寄存器
0x30	LCD_INTF_CFG	接口配置寄存器
0x34	LCD_INT	中断使能、查询、清除寄存器
0x38	LCD_TE_VSYNC_CFG	垂直同步检测配置寄存器
0x3C	LCD_TE_HSYNC_CFG	水平同步检测配置寄存器

15.4.2. LCD_TXFF_PUSH (Offset 0x00)

表格 15-2 LCD_TXFF_PUSH 寄存器

比特	名称	属性	复位值	描述
31:0	LCD_TXFF_PUSH_DATA	W	32'h0	Tx FIFO

15.4.3. LCD_CFG (Offset 0x04)

表格 15-3 LCD_CFG 寄存器

比特	名称	属性	复位值	描述
31	LCD_CFG_WRT	R/W	1'b0	1: 写 0: 读
30	LCD_CFG_CDX	R/W	1'b0	只在 LCD_CFG_WRT 为 1 时有效。 1: LCD_CFG_DATA 为参数 0: LCD_CFG_DATA 为命令
29:8	LCD_CFG_LEN	R/W	22'h00	只在 LCD_CFG_WRT 为 0 时有效，表示需要读取数据的个数。
7:0	LCD_CFG_DATA	R/W	8'h00	发送的命令或参数。

15.4.4. LCD_CFGWR_STATUS (Offset 0x08)

表格 15-4 LCD_CFGWR_STATUS 寄存器

比特	名称	属性	复位值	描述
31:1	Reserved	R	31'h00	Reserved
0	LCD_CFGWR_STATUS	R	1'b0	CFG WR 操作执行后 (LCD_CFG_WRT 为 1，写 LCD_CFG 寄存器)，该位置 0。当并行总线完成该次写操作后，该状态位置 1。

15.4.5. LCD_CFGRD_STATUS (Offset 0x0C)

表格 15-5 LCD_CFGRD_STATUS 寄存器

比特	名称	属性	复位值	描述
31:1	Reserved	R	31'h00	Reserved
0	LCD_CFGRD_STATUS	R	1'b0	CFG RD 操作执行后 (LCD_CFG_WRT 为 0，写 LCD_CFG 寄存器)，数据可读时，该状态位置 1。每次读取 LCD_APB_RD 后，该位清 0，当数据再次可读时，再次置为 1，直到所有 LCD_CFG_LEN 个数据都读取完毕

15.4.6. LCD_APB_RD (Offset 0x10)

表格 15-6 LCD_APB_RD 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	16'h00	Reserved
15:0	LCD_APB_RD_DATA	R	16'h00	当总线上数据被存储到寄存器后，LCD_CFGRD_STATUS 为 1，CPU 可通过该地址获取数据。读取后，LCD_CFGRD_STATUS 清 0

15.4.7. LCD_TXFF_AEMP_LV (Offset 0x14)

表格 15-7 LCD_TXFF_AEMP_LV 寄存器

比特	名称	属性	复位值	描述
31:5	Reserved	R	27'h00	Reserved
4:0	LCD_TXFF_AEMP_LV	R/W	5'h0	Tx FIFO 近空阈值。当 Tx FIFO 中数据等于或小于设置值时，Tx FIFO 近空状态置位。

15.4.8. LCD_DAT_WR_LEN (Offset 0x18)

表格 15-8 LCD_DAT_WR_LEN 寄存器

比特	名称	属性	复位值	描述
31:24	Reserved	R	8'h00	Reserved
23:0	LCD_DAT_WR_LEN	R/W	24'h00	LCD DAT WR 数据长度

15.4.9. LCD_DAT_WR_CFG (Offset 0x1C)

表格 15-9 LCD_DAT_WR_CFG 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R	0x0	Reserved
15:14	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]

比特	名称	属性	复位值	描述
				01: FIFO_DATA [15:8] 10: FIFO_DATA [23:16] 11: FIFO_DATA [31:24]
13:12	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]
11:10	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]
9:8	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]
7	DATA_BUS_16BIT	R/W	1'b0	1: 16 bit parallel data bus selected 0: 8 bit parallel data bus selected
6:4	WR_H_LEN	R/W	3'b1	DAT WR 操作时, 总线 WRCLK 高电平持续的 wrclk 个数
3	Reserved	R	1'b0	Reserved
2:0	WR_L_LEN	R/W	3'b1	DAT WR 操作时, 总线 WRCLK 低电平持续的 wrclk 个数

15.4.10. LCD_DMA (Offset 0x20)

表格 15-10 LCD_DMA 寄存器

比特	名称	属性	复位值	描述
31:6	Reserved	R	0x0	Reserved
5	DMA_CONTROL	R/W	1'b0	1: enable

				0: disable
4:0	DMA_TX_LEVEL	R/W	5'h10	当 TX FIFO 中数据个数小于等于该阈值时，发起 DMA Tx 请求。

15.4.11. LCD_CSX (Offset 0x24)

表格 15-11 LCD_CSX 寄存器

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

15.4.12. LCD_CRM (Offset 0x28)

表格 15-12 LCD_CRM 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R	0x0	Reserved
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

15.4.13. LCD_TXFF_CLR (Offset 0x2C)

表格 15-13 LCD_TXFF_CLR 寄存器

比特	名称	属性	复位值	描述
31:2	Reserved	R	0x0	Reserved
1	LCD_TXFF_PUSH_CLR	R/W	1'b0	复位写 FIFO 指针 1: reset push pointer 0: release
0	LCD_TXFF_POP_CLR	R/W	1'b0	复位读 FIFO 指针 1: reset pop pointer 0: release

15.4.14. LCD_INTF_CFG (Offset 0x30)

表格 15-14 LCD_INTF_CFG 寄存器

比特	名称	属性	复位值	描述
31:4	Reserved	R	0x0	Reserved
3	P68_RD	R/W	1'b0	6800 模式下, 1: 读 (CFG_RD 前配置为 1, 操作完成后配置为 0) 0: 写
2	P68_MODE	R/W	1'b0	8080 or 6800 select: 1: 6800 0: 8080
1	CS_POLARITY	R/W	1'b0	CS 极性选择: 1: high active 0: low active
0	DC_POLARITY	R/W	1'b0	DC 极性选择: 1: high active 0: low active

15.4.15. LCD_INT (Offset 0x34)

表格 15-15 LCD_INT 寄存器

比特	名称	属性	复位值	描述
31:12	Reserved	R	0x0	Reserved
11	Hsync_int_clear	W	1'b0	写 '1' 清除 Hsync 状态
10	Hsync	R	1'b0	水平同步状态
9	Hsync_int_enable	R/W	1'b0	水平同步状态中断使能。 0: Disable 1: Enable
8	Vsync_int_clear	W	1'b0	写 '1' 清除 Vsync 状态
7	Vsync	R	1'b0	垂直同步状态
6	Vsync_int_enable	R/W	1'b0	垂直同步状态中断使能。 0: Disable 1: Enable
5	txfifo_aempt	R	1'b1	Tx FIFO 近空状态
4	txfifo_aempt_int_enable	R/W	1'b0	Tx FIFO 近空状态中断使能。 0: Disable 1: Enable
3	txfifo_empty	R	1'b1	Tx FIFO 空状态
2	txfifo_empty_inten	R/W	1'b0	Tx FIFO 空状态中断使能。 0: Disable 1: Enable
1	txfifo_full	R	1'b0	Tx FIFO 满状态
0	txfifo_full_inten	R/W	1'b0	Tx FIFO 满状态中断使能 0: Disable 1: Enable

15.4.16. LCD_TE_VSYNC_CFG (Offset 0x38)

表格 15-16 LCD_TE_VSYNC_CFG 寄存器

比特	名称	属性	复位值	描述
31:24	Reserved	R	0x0	Reserved
23:0	TE_VSYNC_WIDTH_NUM	R/W	24'h1E	当 TE 信号高电平持续时间达到

				TE_VSYNC_WIDTH_NUM 个 pclk(cpu 时钟), 表示检测到 Vsync 宽脉冲, 若 Vsync 中断使能, 则相应中断置起
--	--	--	--	---

15.4.17. LCD_TE_HSYNC_CFG (Offset 0x3C)

表格 15-17 LCD_TE_HSYNC_CFG 寄存器

比特	名称	属性	复位值	描述
31:13	Reserved	R	0x0	Reserved
12	TE_HSYNC_POLARITY	R/W	1'b0	line number 计数的极性 1: 上升沿 line number + 1 0: 下降沿 line number + 1
11:0	TE_HSYNC_LINE_NUM	R/W	12'h01	Vsync 宽脉冲检测到后, 第 TE_HSYNC_LINE_NUM 个上升沿或下降沿, 若 te_hsync 中断使能, 则相应中断置起。

15.5. 使用流程

15.5.1. 写命令或写参数

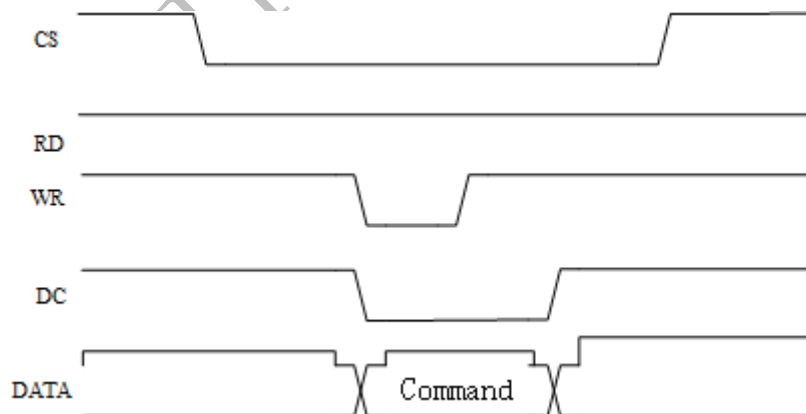


图 15-2 8080 写命令时序

1. 首先拉低 CSX 信号。
2. 配置 LCD_CFG 寄存器。因为是写命令、参数操作, LCD_CFG_WR 设置为 1。如果需要发送 CMD, 则 LCD_CFG_CDX 应配置为 0。如果是发送参数, 则 LCD_CFG_CDX 应配置

- 为 1。将命令或参数写入 LCD_CFG_DATA。
3. 读取 LCD_CFGWR_STATUS 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

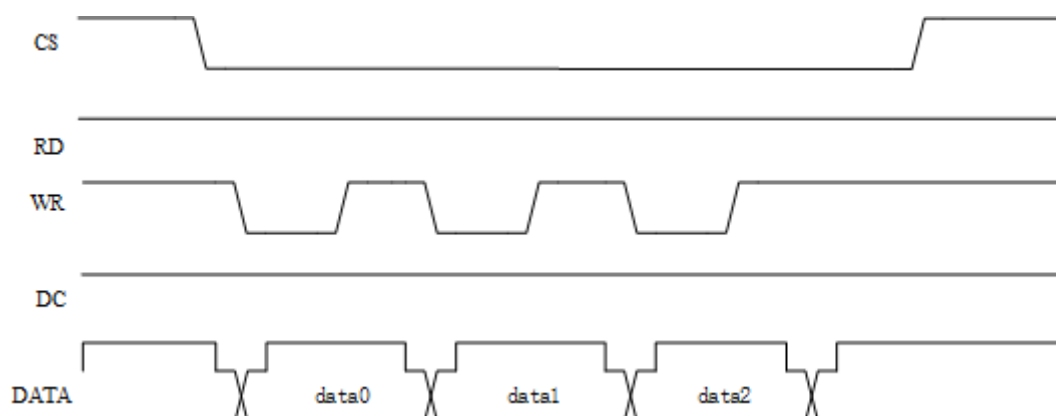


图 15-3 8080 写数据时序

1. 首先拉低 CSX 信号。
2. 根据所需传输数据的个数，配置 LCD_DAT_WR_LEN。
3. 向 FIFO 中写入数据。
4. 最后拉高 CSX 信号，结束。

15.5.3. 读数据

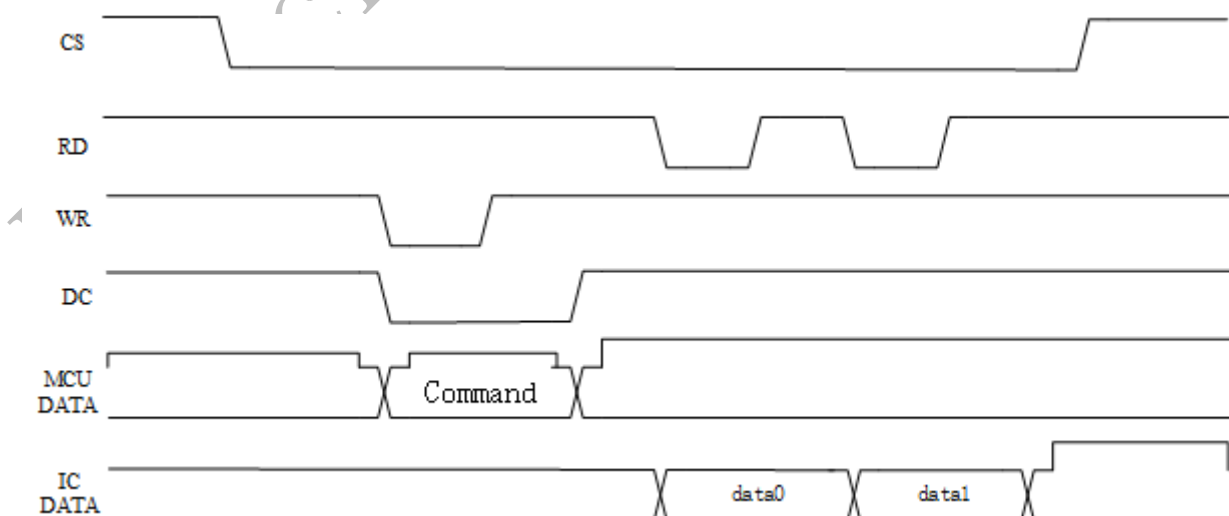


图 15-4 8080 读数据时序

1. 首先拉低 CSX 信号。
1. 配置 LCD_CFG 寄存器，因为是读操作，LCD_CFG_WR 设置为 0。LCD_CFG_CDX 无需配置。配置 LCD_CFG_LEN，写入需要读取数据的长度。向 LCD_CFG_DATA 写入读的 CMD（读数据操作，回先发送这个 CMD，然后进行读取操作）。
2. 读取 LCD_CFGRD_STATUS 状态，确保数据已准备好，读取 LCD_APB_RD_DATA 获取一次数据，重复该步骤直到读完所有 LCD_CFG_LEN 个数据（读取完所有数据后，LCD_CFGWR_STATUS 不会再置起）。
3. 最后拉高 CSX 信号，结束。

16. 通用定时器（Timer）

16.1. 概述

提供了两路 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. 系统框图

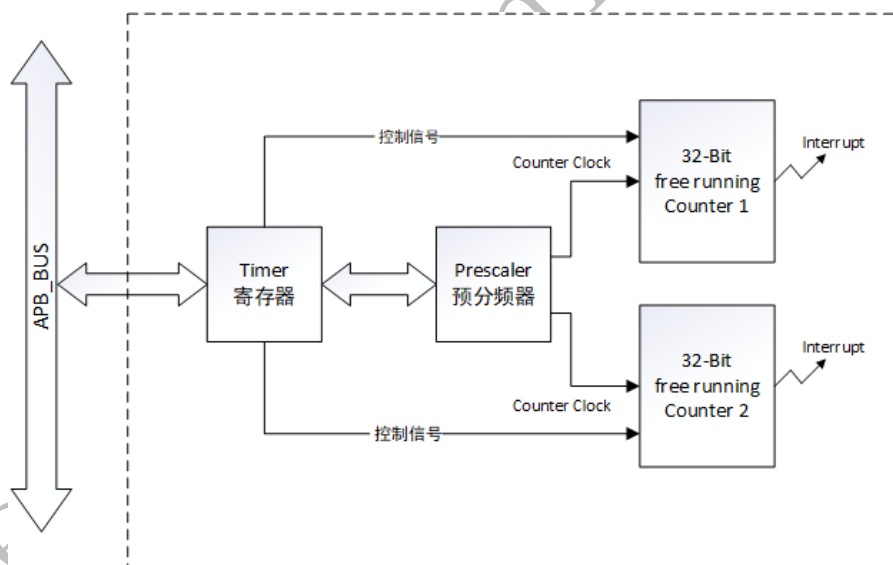


图 16-1 Timer 系统框图

16.4. Timer 寄存器

16.4.1. Timer 寄存器地址映像

Timer0 寄存器基地址：0x5001_0000

Timer0 寄存器基地址：0x5001_0020

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	中断控制，查询寄存器

16.4.2. LOAD_VALUE (Offset 0x00)

表格 16-2 LOAD_VALUE 寄存器

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

16.4.3. COUNT_VALUE (Offset 0x04)

表格 16-3 COUNT_VALUE 寄存器

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

16.4.4. CONTROL (Offset 0x08)

表格 16-4 CONTROL 寄存器

比特	名称	属性	复位值	描述
31:8	Reserved	R/W	0x0	Reserved
7	CNT_EN	R/W	1'b0	计数使能： 1：开始计数 0：停止计数
6	CNT_MODE	R/W	1'b0	计数模式选择： 0：单次计数 1：循环计数
5:4	Reserved	R/W	2'b0	Reserved

比特	名称	属性	复位值	描述
3:2	PRESCALE	R/W	2'b0	分频选择: 00: pclk 01: pclk/16 10: pclk/256
1:0	Reserved	R/W	2'b0	Reserved

16.4.5. INT_CLR (Offset 0x0C)

表格 16-5 INT_CLR 寄存器

比特	名称	属性	复位值	描述
31:16	Reserved	R/W	0x0	Reserved
15:0	INT_CLR	R/W	16'h00	写 '1' 清除中断状态。

16.5. 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，各通道的 A/D 转换可以连续循环模式执行。

17.2. 主要特性

- 10 位分辨率
- 8 个独立的采样通道
- 连续循环转换模式
- 深度 64，位宽 10bit 的数据 FIFO
- 支持电池电压，核心温度检测

17.3. 结构框图

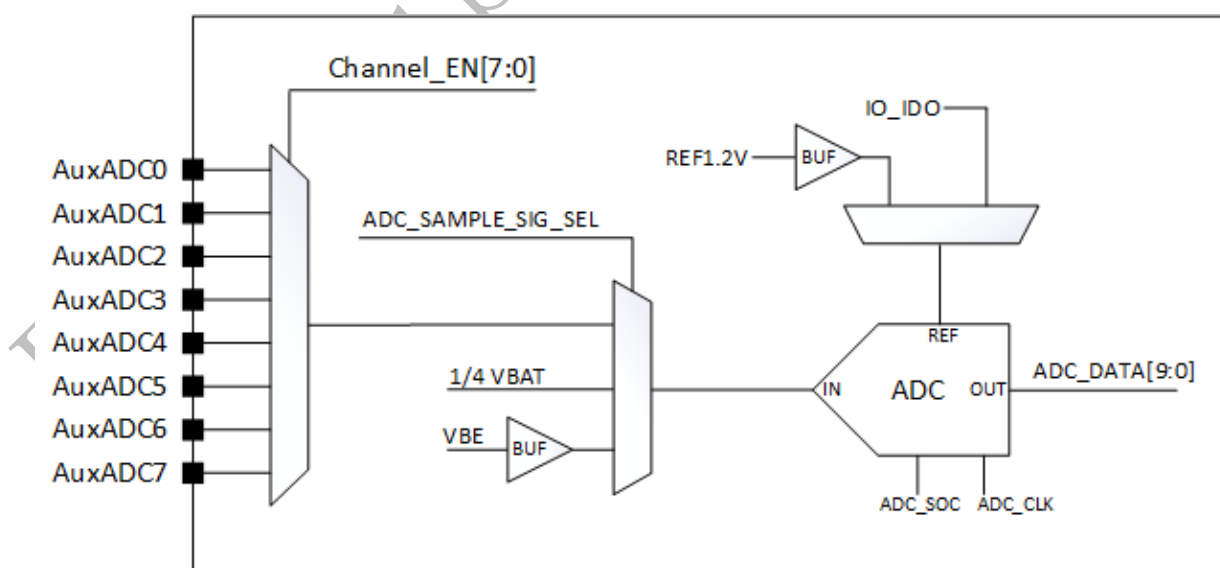


图 17-1 ADC 结构框图

17.4. 功能描述

对于连续循环转换模式。

$$\text{通道采样时钟 } \text{ADC_CLK} = \frac{48\text{MHz}}{(\text{CLK_DIV} + 1) \times 2}$$

$$\text{通道采样周期 } \text{ADC_Channel_Period} = \text{ADC_CLK} / 15$$

当使能 ADC_EN 开始采样后，ADC 控制器从通道 0 开始，将 Channel_EN 中使能的通道开始连续循环转换。每个通道转换结束后将转换值写入对应的通道数据寄存器中。若启用 FIFO，则转换完的数据将会储存在 ADC_FIFO 中，通过读 DATA 寄存器的[9:0]位可以将 FIFO 中的数据取出。

17.5. ADC 寄存器

17.5.1. ADC 寄存器映像

ADC 寄存器基地址：0x5009_0000

表格 17-1 ADC 寄存器映像

偏移	寄存器	寄存器描述
0x00	CONTROL	ADC 控制寄存器
0x04	DATA0	通道 0 ~ 7 数据寄存器
0x08	DATA1	
0x0C	DATA2	
0x10	DATA3	
0x14	DATA4	
0x18	DATA5	
0x1C	DATA6	
0x20	DATA7	
0x24	DATA	Data FIFO 寄存器
0x28	FIFO_Status	FIFO 状态寄存器
0x2C	INT	中断状态寄存器

偏移	寄存器	寄存器描述
0x30	INTC	中断状态清除寄存器
0x34	ADC_ANA_CTRL0	ADC 模拟控制寄存器 0
0x38	ADC_ANA_CTRL1	ADC 模拟控制寄存器 1
0x3C	ADC_ANA_CTRL2	ADC 模拟控制寄存器 2

17.5.2. CONTROL (Offset 0x00)

表格 17-2 CONTROL 寄存器

比特	名称	属性	复位值	描述
31	FIFO_CLR	R/W	1'b0	FIFO 清除。 0: 停止清除 1: 清除 FIFO 注: 需软件写 '1' 清除后再写 '0' 停止。
30	ADC_TRIG_SW	R/W	1'b0	转换触发条件。 0: 软件触发 1: 禁用 注: 软件通过将 ADC_EN 置 '1' 开启 ADC 转换。
29:24	DMA_Level	R/W	6'h30	DMA 搬运阈值。 FIFO 中数据等于或大于 DMA_Level, 产生一次 DMA 请求。
23	DMA_EN	R/W	1'b0	DMA 使能。
22	Reserved	R/W	0x0	Reserved
21:16	FIFO_Almost_Full_Level	R/W	6'h30	FIFO 近满阈值。 当 FIFO 中数据等于或大于 FIFO_Almost_Full_Level 时, FIFO_ALMOST_FULL 置位
15:8	Channel_EN	R/W	8'h00	通道使能。 1bit 对应 1 通道。 Bit[8] ---- Channel0 Bit[9] ---- Channel1 Bit[10] --- Channel2 Bit[11] --- Channel3

比特	名称	属性	复位值	描述
				Bit[12] --- Channel4 Bit[13] --- Channel5 Bit[14] --- Channel6 Bit[15] --- Channel7
7	ADC_EN	R/W	1'b0	ADC 转换使能。 0: 停止转换 1: 开始转换
6:4	Reserved	R/W	0x0	Reserved
3	DATA_VALID	R	1'b0	数据有效信号。 0: 未转换完成。 1: 内部信号转换完成标志。 注: 此位只在 ADC_MODE 配置为 '1' 时有效, 表示内部信号转换完成, 可以从 DATA 寄存器中读取转换结果值。
2	ADC_MODE	R/W	1'b0	ADC 模式选择。 0: 外部输入信号循环转换模式。 1: 内部信号循环转换模式。 注: 置 '1' 时, 测量 1/4 VBAT、VEB、。
1	FIFO_EN	R/W	1'b0	FIFO 使能。 0: 不启用 FIFO 1: 启用 FIFO 注: 启动 FIFO 后, 转换结果值将保存在 FIFO 中, 读 DATA 寄存器, 将 FIFO 中数据读出。
0	ADC_CK_EN	R/W	1'b0	ADC 采样时钟使能

17.5.3. DATA0 ~ DATA7 (Offset 0x04 ~ 0x20)

表格 17-3 DATA0~DATA7 寄存器

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	0x0	Reserved
9:0	Channel0 (Offset 0x04)	R/W	0x0	通道 0 采样转换数据。
9:0	Channel1 (Offset 0x08)	R/W	0x0	通道 1 采样转换数据。

比特	名称	属性	复位值	描述
9:0	Channel2 (Offset 0x0C)	R/W	0x0	通道 2 采样转换数据。
9:0	Channel3 (Offset 0x10)	R/W	0x0	通道 3 采样转换数据。
9:0	Channel4 (Offset 0x14)	R/W	0x0	通道 4 采样转换数据。
9:0	Channel5 (Offset 0x18)	R/W	0x0	通道 5 采样转换数据。
9:0	Channel6 (Offset 0x1C)	R/W	0x0	通道 6 采样转换数据。
9:0	Channel7 (Offset 0x20)	R/W	0x0	通道 7 采样转换数据。

注：不使用 FIFO 时，数据将存入上述寄存器中。

17.5.4. DATA (Offset 0x24)

表格 17-4 DATA 寄存器

比特	名称	属性	复位值	描述
31:10	Reserved	R	0x0	Reserved
9:0	ADC_FIFO	R	0x0	使能 FIFO 或测试内部信号时，转换结果将保存在 FIFO 中，读此寄存器，将结果值从 FIFO 中读出。

注：使用 FIFO 时，数据将存入上述寄存器中。

17.5.5. FIFO_Status (Offset 0x28)

表格 17-5 FIFO_Status 寄存器

比特	名称	属性	复位值	描述
31:7	Reserved	R	0x0	Reserved
4	FIFO_UNDER	R	1'b0	FIFO 下溢出
3	FIFO_OVER	R	1'b0	FIFO 溢出
2	FIFO_ALMOST_FULL	R	1'b0	FIFO 近满
1	FIFO_EMPTY	R	1'b1	FIFO 空
0	FIFO_FULL	R	1'b0	FIFO 满

17.5.6. INT (Offset 0x2C)

表格 17-6 INT 寄存器

比特	名称	属性	复位值	描述
31:23	Reserved	R	0x0	Reserved
22	SW	R	1'b0	对应中断状态
21	ERR	R	1'b0	
20	FIFO_UNDER	R	1'b0	
19	FIFO_OVER	R	1'b0	
18	FIFO_ALMOST_FULL	R	1'b0	
17	FIFO_EMPTY	R	1'b0	
16	FIFO_FULL	R	1'b0	
15:7	Reserved	R	0x0	Reserved
6	SW_INT	R/W	1'b0	软件触发转换中断使能。
5	ERR_INT	R/W	1'b0	ADC 错误中断使能。若检测到 ADC Error 应复位 ADC 模块。
4	FIFO_UNDER_INT	R/W	1'b0	FIFO under 中断使能。FIFO 空时读 DATA
3	FIFO_OVER_INT	R/W	1'b0	FIFO 溢出中断使能。
2	FIFO_ALMOST_FULL_INT	R/W	1'b0	FIFO 近满中断使能。
1	FIFO_EMPTY_INT	R/W	1'b0	FIFO 空中断使能
0	FIFO_FULL_INT	R/W	1'b0	FIFO 满中断使能。 0: Disable 1: Enable

17.5.7. INTC (Offset 0x30)

表格 17-7 INTC 寄存器

比特	名称	属性	复位值	描述
31:7	Reserved	R/W	0x0	Reserved
6	SW_CLR	W	1'b0	写 '1' 对应状态清除
5	ERR_CLR	W	1'b0	
4	FIFO_UNDER_CLR	W	1'b0	
3	FIFO_OVER_CLR	W	1'b0	

比特	名称	属性	复位值	描述
2	FIFO_ALMOST_FULL_CLR	W	1'b0	
1	FIFO_EMPTY_CLR	W	1'b0	
0	FIFO_FULL_CLR	W	1'b0	

17.5.8. ADC_ANA_CTRL0 (Offset 0x34)

表格 17-8 ADC_ANA_CTRL0 寄存器

比特	名称	属性	复位值	描述
31:16	Delay_For_Setup	R/W	16'h50	通道采集建立时间延时。 单位为 48MHz 时钟周期。
15:8	CLK_DIV	R/W	8'h5	ADC 采样时钟分频。 $48\text{MHz} / ((\text{CLK_DIV} + 1) * 2)$ 注: $\text{CLK_DIV} \geq 1$ 。
7:5	ADC_SAMPLE_SIG_SEL	R/W	3'h0	转换源选择。 0x3: IO (引脚电压) 0x6: VBE (温度) 0x7: 1/4 VBAT (1/4 电池电压)
4	ADC_SACLK_CTL	R/W	1'b1	ADC 内部时钟控制。总是置 '1'。
3	ADC_CLK_EDGE	R/W	1'b0	内部时钟转换边沿选择。 0: 上升沿时开始转换 1: 下降沿时开始转换
2	ADC_ISON	R/W	1'b0	隔离使能。 0: ADC 隔离 1: ADC 不隔离 注: ADC 正常工作时应配置为不隔离, 在不使用时应配置为隔离。
1	ADC_RSTN	R/W	1'b0	复位使能。 0: 复位 1: 不复位
0	ADC_POWER	R/W	1'b1	ADC 电源控制。 0: Power On 1: Power Down

17.5.9. ADC_ANA_CTRL1 (Offset 0x38)

表格 17-9 ADC_ANA_CTRL1 寄存器

比特	名称	属性	复位值	描述
31:10	Reserved	R/W	0x0	Reserved
9	ADC_REF1.2V_BUF_EN	R/W	1'b0	1.2V 参考电压 BUF 使能。 0: Disable 1: Enable
8:5	Reserved	R/W	0x0	Reserved
4	ADC_VBAT_DIV_EN	R/W	1'b0	1/4VBAT 分压使能。 0: Disable 1: Enable
3	ADC_REFH_MODE	R/W	1'b0	ADC 参考信号选择。 0: IO_IDO 1: 内部 1.2V
2	ADC_BUF_IN_SEL	R/W	1'b0	ADC BUF 输入选择。 0: 选择 VBE 1: 无效
1	ADC_BUF_EN	R/W	1'b0	ADC BUF 使能 0: Disable 1: Enable
0	Reserved	R/W	0x0	Reserved

17.5.10. ADC_ANA_CTRL2 (Offset 0x3C)

表格 17-10 ADC_ANA_CTRL2 寄存器

比特	名称	属性	复位值	描述
31:24	Reserved	R/W	0x0	Reserved
23:16	SAMPLE_POINT	R/W	8'h8	采样点数量
15:7	Reserved	R/W	0x0	Reserved
6:2	CONV_TIME	R/W	5'hE	转换超时时间
1	TOUT_MASK	R/W	1'b0	超时状态屏蔽。 0: 屏蔽 ERR 中断状态。 1: 不屏蔽 ERR 中断状态。

比特	名称	属性	复位值	描述
0	POLL_MODE	R/W	1'b0	循环采样。总是置 '0'。

Freqchip Confidential

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 0 和 5 对(端点 1~5)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

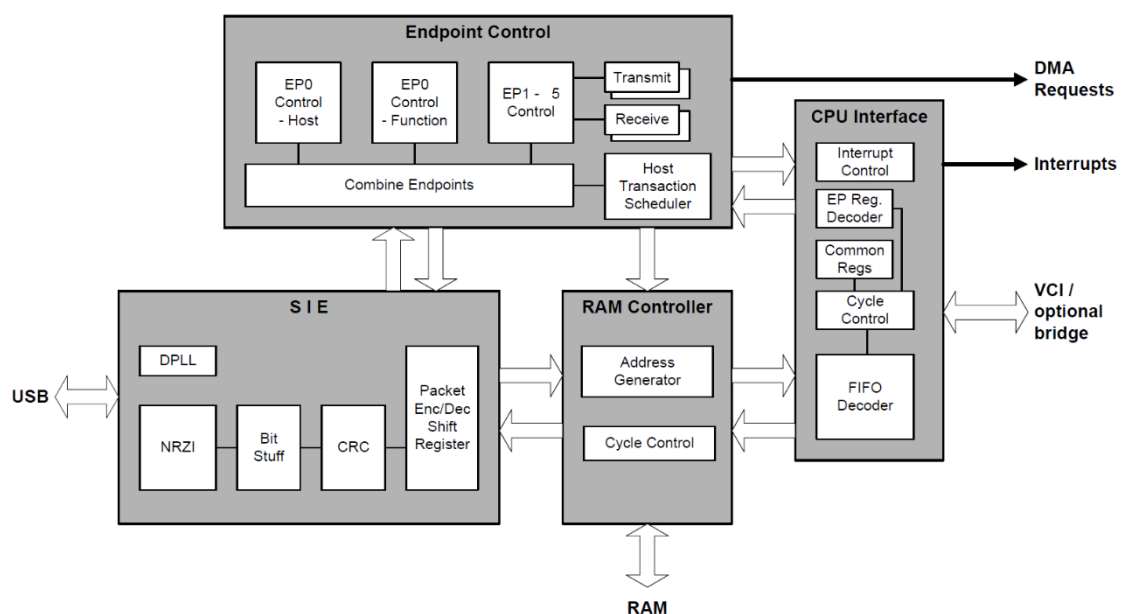


图 18-1 USB 系统框图

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：0x2004_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	IntrUSBE	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）		

偏移	寄存器	寄存器描述
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）

偏移	寄存器	寄存器描述
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）

表格 18-2 Faddr 寄存器

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）

表格 18-3 Power 寄存器

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）

比特	名称	Device Mode	Host Mode	复位值	描述
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
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 读取中断寄存器或清除该位或离开主机模式时，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)

表格 18-4 IntrTx1 寄存器

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)

表格 18-5 IntrTx2 寄存器

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)

表格 18-6 IntrRx1 寄存器

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)

表格 18-7 IntrRx2 寄存器

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

18.3.2.7. IntrUSB (Offset 0x06)

表格 18-8 IntrUSB 寄存器

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)

表格 18-9 IntrTx1E 寄存器

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)

表格 18-10 IntrTx2E 寄存器

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)

表格 18-11 IntrRx1E 寄存器

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)

表格 18-12 IntrRx2E 寄存器

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)

表格 18-13 IntrUSBE 寄存器

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

18.3.2.13. Frame1 (Offset 0x0C)

表格 18-14 Frame1 寄存器

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)

表格 18-15 Frame2 寄存器

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)

表格 18-16 Index 寄存器

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)

表格 18-17 DevCtl 寄存器

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。

比特	名称	属性	复位值	描述
				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. CSRO (Offset 0x11)

IN Device Mode:

表格 18-18 Device Mode CSRO 寄存器

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:

表格 18-19 Host Mode CSR0 寄存器

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)

表格 18-20 CSR2 寄存器

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)

表格 18-21 Count0 寄存器

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)

表格 18-22 NAKLimit0 寄存器

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)

表格 18-23 TxMaxP 寄存器

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:

表格 18-24 Device Mode TxCSR1 寄存器

比特	名称	属性	复位值	描述
7	Reserved	R	1'b0	Reserved

比特	名称	属性	复位值	描述
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:

表格 18-25 Host Mode TxCSR1 寄存器

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'。

18.3.4.3. TxCSR2 (Offset 0x12)

表格 18-26 TxCSR2 寄存器

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)

表格 18-27 RxMaxP 寄存器

比特	名称	属性	复位值	描述
7:0	Maximum Packet Size	R/W	8'h00	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。 最大数据包不应超过 USB 规范要求。 端点 FIFO 配置章节

18.3.4.5. RxCSR1 (Offset 0x14)

IN Device Mode:

表格 18-28 Device Mode RxCSR1 寄存器

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	当 STALL 发送后置位。 (写 '0' 清除)
5	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。
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	R	1'b0	当 RxPktRdy 置位并且数据包发生 CRC 错误或 bit 错误时置位。 清除 RxPktRdy 时，会将此位清除。 注： 只在同步传输时有效
2	OverRun	R	1'b0	当 Rx FIFO 无法装下接收到的数据包时置位。 (写 '0' 清除) 注： 只在同步传输时有效
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)
---	----------	---	------	---

IN Host Mode:

表格 18-29 Host Mode RxCSR1 寄存器

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)

表格 18-30 RxCSR2 寄存器

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

比特	名称	属性	复位值	描述
				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)

表格 18-31 RxCount1 寄存器

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)

表格 18-32 RxCount2 寄存器

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)

表格 18-33 TxType 寄存器

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)

表格 18-34 TxInterval 寄存器

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)

表格 18-35 RxType 寄存器

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)

表格 18-36 RxInterval 寄存器

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)

表格 18-37 TxTxFIFO1、TxTxFIFO2 寄存器

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。 <table><tr><th>AD[11 : 0]</th><th>Start Address</th></tr><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></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																			
7:0	AD[7:0]																			
TxFIFO2																				
3:0	AD[11:8]																			
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><tr><th>SZ[2 : 0]</th><th>端点 FIFO 大小（byte）</th></tr><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></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)

表格 18-38 RxFIFO1、RxFIFO2 寄存器

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。 <table><tr><th>AD[11 : 0]</th><th>Start Address</th></tr><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></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																			
7:0	AD[7:0]																			
RxFIFO2																				
3:0	AD[11:8]																			
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><tr><th>SZ[2 : 0]</th><th>端点 FIFO 大小（byte）</th></tr><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></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																			

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

表格 18-39 USB 端点 FIFO 寄存器

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. 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

网址： www.freqchip.com

销售： sales@freqchip.com

本文档的所有部分，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、仿制本文档的全部或部分。富芮坤保留在不另行通知的情况下随时对产品或本文档进行更改、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。

20. 附录 I

表格 20-1 PortA 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	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT
0x2	SPI0_M_CLK	SPI0_M_CS	SPI0_M_IO0	SPI0_M_IO1	SPI0_M_IO2	SPI0_M_IO3	SPI0_M_CLK	SPI0_M_CS
0x3	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO
0x4	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS
0x5	USB_DP	USB_DM	UART1_Rx	UART1_Tx	USB_DP	USB_DM	UART1_Rx	UART1_Tx
0x6	PWM0	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7
0x7	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA
0x8					CLK_OUT			CLK_OUT
0x9	IrDA_IN	IrDA_OUT				IrDA_IN	IrDA_OUT	
0xA	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN
0xB								
0xC								
0xD	LCD_CS	LCD_DC	LCD_WR	LCD_RD	LCD_TE	LCD_TE	LCD_CS	LCD_DC
0xE								
0xF								

表格 20-2 PortB 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	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT
0x2	SPI0_M_CLK	SPI0_M_CS	SPI0_M_IO0	SPI0_M_IO1	SPI0_M_IO2	SPI0_M_IO3	SPI0_M_CLK	SPI0_M_CS
0x3	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO
0x4	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS
0x5	USB_DP	USB_DM	UART1_Rx	UART1_Tx	USB_DP	USB_DM	UART1_Rx	UART1_Tx
0x6	PWM0	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7
0x7	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA
0x8					CLK_OUT			CLK_OUT
0x9	IrDA_IN	IrDA_OUT				IrDA_IN	IrDA_OUT	
0xA					I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN
0xB								
0xC								
0xD	LCD_D0	LCD_D1	LCD_D2	LCD_D3	LCD_D4	LCD_D5	LCD_D6	LCD_D7
0xE								
0xF								

表格 20-3 PortC 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	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT
0x2	SPI1_M_CLK	SPI1_M_CS	SPI1_M_IO0	SPI1_M_IO1	SPI1_M_IO2	SPI1_M_IO3	SPI1_M_CLK	SPI1_M_CS
0x3	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO
0x4	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS
0x5	LCD_CS	LCD_DC	UART1_Rx	UART1_Tx	LCD_WR	LCD_RD	UART1_Rx	UART1_Tx
0x6	PWM0	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7
0x7	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA
0x8							SWCLK	SWDIO
0x9	IrDA_IN	IrDA_OUT				IrDA_IN	IrDA_OUT	
0xA	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN
0xB								
0xC	QSPI0_IO3	QSPI0_CLK	QSPI0_CS	QSPI0_IO1	QSPI0_IO2	QSPI0_IO0		
0xD	LCD_D8	LCD_D9	LCD_D10	LCD_D11	LCD_D12	LCD_D13	LCD_D14	LCD_D15
0xE								
0xF								

表格 20-4 PortD 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	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT	I2C0_CLK	I2C0_DAT	I2C1_CLK	I2C1_DAT
0x2	SPI1_M_CLK	SPI1_M_CS	SPI1_M_IO0	SPI1_M_IO1	SPI1_M_IO2	SPI1_M_IO3	SPI1_M_CLK	SPI1_M_CS
0x3	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO	SPI_S_CLK	SPI_S_CS	SPI_S_MOSI	SPI_S_MISO
0x4	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS	UART0_Rx	UART0_Tx	UART0_RTS	UART0_CTS
0x5			UART1_Rx	UART1_Tx			UART1_Rx	UART1_Tx
0x6	PWM0	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7
0x7	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA	PDM_CLK	PDM_DATA
0x8	AuxADC7	AuxADC6	AuxADC5	AuxADC4	AuxADC3	AuxADC2	AuxADC1	AuxADC0
0x9	IrDA_IN	IrDA_OUT				IrDA_IN	IrDA_OUT	
0xA	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN	I2S_CK	I2S_WS	I2S_SD_OUT	I2S_SD_IN
0xB								
0xC								
0xD	LCD_WR	LCD_RD	LCD_TE	LCD_TE	LCD_CS	LCD_DC	LCD_WR	LCD_RD
0xE								
0xF								

表格 20-5 PortE 复用功能

引脚 功能	PE0	PE1						
0x0	GPIO_E0	GPIO_E1						
0x1	I2C0_CLK	I2C0_DAT						
0x2	SPI1_M_IO0	SPI1_M_IO1						
0x3	SPI_S_CLK	SPI_S_CS						
0x4	UART0_Rx	UART0_Tx						
0x5	UART1_Rx	UART1_Tx						
0x6	PWM0	PWM1						
0x7	PDM_CLK	PDM_DATA						
0x8								
0x9								
0xA								
0xB								
0xC								
0xD	LCD_TE	LCD_TE						
0xE								
0xF								

USB_PHY_DP	USB_DP	UART0_Rx	UART1_Rx
USB_PHY_DM	USB_DM	UART0_Tx	UART1_Tx

21. 附录 II

表格 21-1 PMU PortA 复用功能

功能 引脚	2'b00	2'b01	2'b10	2'b11
PA0	PMU_GPIO_A0	PMU_PWM0	KeyScan_COL[0]	
PA1	PMU_GPIO_A1	PMU_PWM1	KeyScan_COL[1]	
PA2	PMU_GPIO_A2	PMU_PWM2	KeyScan_COL[2]	
PA3	PMU_GPIO_A3	PMU_PWM3	KeyScan_COL[3]	
PA4	PMU_GPIO_A4	PMU_PWM0	KeyScan_COL[4]	
PA5	PMU_GPIO_A5	PMU_PWM1	KeyScan_COL[5]	
PA6	PMU_GPIO_A6	PMU_PWM2	KeyScan_COL[6]	
PA7	PMU_GPIO_A7	PMU_PWM3	KeyScan_COL[7]	

表格 21-2 PMU PortB 复用功能

功能 引脚	2'b00	2'b01	2'b10	2'b11
PB0	PMU_GPIO_B0	PMU_PWM0	KeyScan_COL[8]	
PB1	PMU_GPIO_B1	PMU_PWM1	KeyScan_COL[9]	
PB2	PMU_GPIO_B2	PMU_PWM2	KeyScan_COL[10]	
PB3	PMU_GPIO_B3	PMU_PWM3	KeyScan_COL[11]	
PB4	PMU_GPIO_B4	PMU_PWM0	KeyScan_COL[12]	
PB5	PMU_GPIO_B5	PMU_PWM1	KeyScan_COL[13]	
PB6	PMU_GPIO_B6	PMU_PWM2	KeyScan_COL[14]	
PB7	PMU_GPIO_B7	PMU_PWM3	KeyScan_COL[15]	

表格 21-3 PMU PortC 复用功能

功能 引脚	2'b00	2'b01	2'b10	2'b11
PC0	PMU_GPIO_C0	PMU_PWM0	KeyScan_COL[16]	
PC1	PMU_GPIO_C1	PMU_PWM1	KeyScan_COL[17]	
PC2	PMU_GPIO_C2	PMU_PWM2	KeyScan_COL[18]	
PC3	PMU_GPIO_C3	PMU_PWM3	KeyScan_COL[19]	
PC4	PMU_GPIO_C4	PMU_PWM0	KeyScan_COL[0]	
PC5	PMU_GPIO_C5	PMU_PWM1	KeyScan_COL[1]	
PC6	PMU_GPIO_C6	PMU_PWM2	KeyScan_COL[2]	
PC7	PMU_GPIO_C7	PMU_PWM3	KeyScan_COL[3]	

表格 21-4 PMU PortD 复用功能

功能 引脚	2'b00	2'b01	2'b10	2'b11
PD0	PMU_GPIO_D0	PMU_PWM0	KeyScan_COL[4]	
PD1	PMU_GPIO_D1	PMU_PWM1	KeyScan_COL[5]	
PD2	PMU_GPIO_D2	PMU_PWM2	KeyScan_COL[6]	
PD3	PMU_GPIO_D3	PMU_PWM3	KeyScan_COL[7]	
PD4	PMU_GPIO_D4	PMU_PWM0	KeyScan_COL[8]	
PD5	PMU_GPIO_D5	PMU_PWM1	KeyScan_COL[9]	
PD6	PMU_GPIO_D6	PMU_PWM2	KeyScan_COL[10]	
PD7	PMU_GPIO_D7	PMU_PWM3	KeyScan_COL[11]	

表格 21-5 PMU PortE 复用功能

功能 引脚	2'b00	2'b01	2'b10	2'b11
PE0	PMU_GPIO_E0	PMU_PWM0	KeyScan_COL[12]	
PE1	PMU_GPIO_E1	PMU_PWM1	KeyScan_COL[13]	