

V6201X 数据手册

版　　本：　　**V1.1**
发行日期：　　**2021.04.16**

改版记录

日期	版本	描述
2020.10.13	1.0	首次发布
2021.04.06	1.1	增加 V6201R

目录

改版记录 1

图索引 10

表索引 12

1. 概要 21

1.1. 概述说明 21
1.2. 特性 21
1.3. 功能框图 23
1.4. 系统框图 23
1.5. 应用场景 24
1.6. 电路特性 24
1.6.1. 绝对最大额定值 24
1.6.2. 电路特性 25
1.6.3. Flash 规格 25
1.6.4. SPI 时序规格 26
1.6.5. UART 时序特性 28
1.6.6. I^2C 时序特性 28

2. 引脚说明 31

2.1. V6201B 引脚分布图 31
2.2. V6201B 引脚说明 33
2.3. 控制寄存器 36
2.3.1. IO 驱动和转换速率控制寄存器 36
2.3.1.1. IODRVCTRL0 寄存器 36
2.3.1.2. IODRVCTRL1 寄存器 38
2.3.1.3. IODRVCTRL2 寄存器 39
2.3.2. 引脚功能多路复用控制寄存器 42
2.3.2.1. pmur_mfs0 寄存器 42
2.3.2.2. pmur_mfs1 寄存器 44
2.3.2.3. pmur_mfs2 寄存器 47
2.3.2.4. pmur_mfs3 寄存器 49
2.3.3. IO 开漏控制寄存器 50
2.3.3.1. ODMODE 寄存器 50

3. 电源供应 52

3.1. 上电时序（Power-on Sequence） 52

4. 时钟与复位 54

4.1. 系统时钟 54
4.1.1. 概述 54
4.1.2. 外部时钟源 54
4.1.3. PLL 编程值 55
4.1.4. 外设时钟参数寄存器（PDIV_VALUE） 55
4.1.4.1. PDIV_VALUE 寄存器 55
4.1.5. 时钟控制寄存器（AHBCLKG 和 APBCLKG） 56
4.1.5.1. AHBCLKG 寄存器 56

4.1.5.2. APBCLKG 寄存器 57
4.2. 系统复位 59
4.2.1. 软件复位 59
4.2.2. 系统复位控制寄存器 60
4.2.2.1. HWRST_MASK 寄存器 60
4.2.2.2. HWRST_MASK 寄存器 61
4.2.2.3. SWRST_MASK 寄存器 61
4.2.2.4. SWRST_CLR 寄存器 63
4.2.2.5. SWRST_STS 寄存器 63

5. 总线互连和内存映射 65
5.1. Cortex-M4 AHB / APB 总线上的内存空间地址映像 65
5.2. Cortex-M0 AHB 总线上的 AHB 主机模块 66
5.3. Cortex-M0 AHB/APB 总线上的内存空间地址映像 66
5.4. Cortex-M4 AHB/APB 总线内存映射图 68
5.5. Cortex-M0 AHB/APB 总线内存映射图 69

6. 启动重映射 70

7. Bootstrap 70
7.1. 概述 70
7.2. Bootstrap 架构与程序 72
7.3. 寄存器 74
7.3.1. STRAP 寄存器 74
7.3.2. HWSTRAP 寄存器 75

8. 功能说明 76
8.1. 版本控制寄存器 76
8.1.1. 概述 76
8.1.2. HWVCODE 寄存器 76
8.2. Cortex-M0 系统控制 77
8.2.1. TXEV/ RXEV 77
8.2.2. 中断控制 77
8.2.3. DMA 请求与确认 78
8.2.4. Cortex-M0 相关寄存器 79
8.2.4.1. STCALIB 寄存器 79
8.2.4.2. PCLKENPH 寄存器 79
8.2.4.3. STCLK_div 寄存器 79
8.3. Cortex-M4 系统控制 80
8.3.1. TXEV/RXEV 80
8.3.2. 中断控制 80
8.3.3. Cortex-M4 相关寄存器 82
8.3.3.1. TSVALUEB1 寄存器 82
8.3.3.2. TSVALUEB2 寄存器 82
8.3.3.3. STCALIB1 寄存器 82
8.3.3.4. STCALIB2 寄存器 83
8.3.3.5. STCALIB3 寄存器 84
8.4. 系统控制单元（SCU） 85
8.4.1. 概述 85

8.4.2. 系统控制寄存器 85
8.4.2.1. BOOTUPSTATUS 寄存器 85
8.4.2.2. PWRMODE 寄存器 86
8.4.2.3. SWRSTCTRL 寄存器 87
8.4.2.4. PLLCTRL 寄存器 88
8.4.2.5. WAKEUPEN 寄存器 89
8.4.3. IPs 时钟，多路选择器 & 复位控制 90
8.4.3.1. IPMISC_CTRL 寄存器 90
8.4.3.2. UART_CLKCTRL 寄存器 91
8.4.4. Vango 控制寄存器 91
8.4.4.1. GLB_COUNTER 寄存器 91
8.4.4.2. VSBCTRL0 寄存器 91
8.4.4.3. BSTREN 寄存器 92
8.4.4.4. SWRST_CLR 寄存器 92
8.4.4.5. CUSTREG4 寄存器 93
8.4.4.6. CUSTREG0 寄存器 93
8.5. 中断控制寄存器 95
8.5.1. 概述 95
8.5.2. 中断控制寄存器列表与描述 95
8.6. UART 97
8.6.1. 概述 97
8.6.2. 特性 97
8.6.3. 预分频器与波特率发生器 97
8.6.3.1. 预分频器 97
8.6.3.2. 波特率发生器 97
8.6.4. 寄存器摘要 97
8.6.5. 寄存器描述 98
8.6.5.1. RBR 寄存器 98
8.6.5.2. THR 寄存器 99
8.6.5.3. IER 寄存器 99
8.6.5.4. IIR 寄存器 101
8.6.5.5. FCR 寄存器 103
8.6.5.6. LCR 寄存器 104
8.6.5.7. MCR 寄存器 106
8.6.5.8. LSR 寄存器 106
8.6.5.9. MSR 寄存器 108
8.6.5.10. DLL 和 DLM 寄存器 109
8.6.5.11. PSR 寄存器 109
8.6.5.12. MDR 寄存器 110
8.7. SSP 112
8.7.1. 概述 112
8.7.2. 特性 112
8.7.3. SSP 串行时钟比率及比特率 112
8.7.4. Texas Instrument SSP 帧时序 113
8.7.5. Motorola SPI 帧时序 114
8.7.6. National Semiconductor MICROWIRE 帧时序 116
8.7.7. 寄存器摘要 117

8.7.8. 寄存器描述 118
8.7.8.1. CR0 寄存器 118
8.7.8.2. CR1 寄存器 120
8.7.8.3. CR2 寄存器 120
8.7.8.4. Status 寄存器 122
8.7.8.5. IntrCR 寄存器 123
8.7.8.6. IntrStatus 寄存器 124
8.7.8.7. TxRxDR 寄存器 125
8.7.8.8. CR3 寄存器 126
8.8. I^2C 127
8.8.1. 概述 127
8.8.2. 特性 127
8.8.3. 功能框图 127
8.8.3.1. 毛刺抑制电路 128
8.8.3.2. I^2C_MnS 电路 128
8.8.4. 寄存器摘要 128
8.8.5. 寄存器描述 129
8.8.5.1. CR 寄存器 129
8.8.5.2. SR 寄存器 131
8.8.5.3. CDR 寄存器 133
8.8.5.4. DR 寄存器 134
8.8.5.5. AR 寄存器 134
8.8.5.6. TGSR 寄存器 135
8.8.5.7. BMR 寄存器 136
8.8.5.8. BSTMR 寄存器 137
8.9. TMR 138
8.9.1. 概述 138
8.9.2. 特性 138
8.9.3. 寄存器摘要 139
8.9.4. 寄存器描述 139
8.9.4.1. INT_CSTAT 寄存器 139
8.9.4.2. TMn_CTRL 寄存器 140
8.9.4.3. TMn_CNTB 寄存器 142
8.9.4.4. TMn_CMPB 寄存器 142
8.9.5. Timer 基本操作流程 143
8.9.6. 自动重载和双缓冲 144
8.9.7. PWM 功能 144
8.9.8. 死区生成 145
8.10. 看门狗 146
8.10.1. 概述 146
8.10.2. 特性 146
8.10.3. 功能框图 146
8.10.4. 功能描述 146
8.10.5. 寄存器摘要 147
8.10.6. 寄存器描述 148
8.10.6.1. WdCounter 寄存器 148
8.10.6.2. WdLoad 寄存器 148

8.10.6.3. WdRestart 寄存器 149
8.10.6.4. WdCR 寄存器 149
8.10.6.5. WdStatus 寄存器 150
8.10.6.6. WdClear 寄存器 150
8.10.6.7. WdIntrlen 寄存器 151
8.11. GPIO 152
8.11.1. 概述 152
8.11.2. 特性 152
8.11.3. 功能框图 152
8.11.4. 连接图 154
8.11.5. 寄存器摘要 154
8.11.6. 寄存器描述 156
8.11.6.1. GpioDataOut 寄存器 156
8.11.6.2. GpioDataIn 寄存器 157
8.11.6.3. GpioDataSet 寄存器 157
8.11.6.4. GpioDataClear 寄存器 157
8.11.6.5. PinPullEnable 寄存器 158
8.11.6.6. PinPullType 寄存器 158
8.11.6.7. IntrEnable 寄存器 159
8.11.6.8. IntrRawState 寄存器 159
8.11.6.9. IntrMaskedState 寄存器 160
8.11.6.10. IntrMask 寄存器 160
8.11.6.11. IntrClear 寄存器 160
8.11.6.12. IntrTrigger 寄存器 161
8.11.6.13. IntrBoth 寄存器 161
8.11.6.14. IntrRiseNeg 寄存器 162
8.11.6.15. BounceEnable 寄存器 162
8.11.6.16. BouncePreScale 寄存器 163
8.11.7. 时序图 163
8.11.7.1. 写 GPIO 输出数据寄存器 163
8.11.7.2. 下降沿触发 164
8.11.7.3. 高电平触发 164
8.12. VGDMA 164
8.12.1. 概述 164
8.12.2. DMA 框图 165
8.12.3. Cortex-M4 DMA 寄存器（通道号 n=0~7） 165
8.12.3.1. Chn_MODE 寄存器 165
8.12.3.2. CHn_ADDRSRC 寄存器 166
8.12.3.3. CHn_ADDRDES 寄存器 166
8.12.3.4. CHn_BLENGTH 寄存器 167
8.12.3.5. CHn_PKTLENGTH 寄存器 167
8.12.3.6. DMA_RESET 寄存器 167
8.12.3.7. DMA_INTC 寄存器 168
8.12.4. Cortex-M0 DMA 寄存器（通道号 n=0~7） 170
8.12.4.1. Chn_MODE 寄存器 170
8.12.4.2. CHn_ADDRSRC 寄存器 170
8.12.4.3. CHn_ADDRDES 寄存器 171

8.12.4.4. CHn_BLENGTH 寄存器 171
8.12.4.5. CHn_PKTLENGTH 寄存器 171
8.12.4.6. DMA_RESET 寄存器 172
8.12.4.7. DMA_INTC 寄存器 172
8.13. 安全加速器 175
8.13.1. 为安全加速器服务的 DMA 的控制状态寄存器 175
8.13.2. DMA CSR 概述 175
8.13.3. 寄存器描述 177
8.13.3.1. DE1_CR 寄存器 177
8.13.3.2. DE1_TDBAR 寄存器 178
8.13.3.3. DE1_CTDAR 寄存器 178
8.13.3.4. DE1_TDR 寄存器 178
8.13.3.5. DE1_RDBAR 寄存器 179
8.13.3.6. DE1_CRDAR 寄存器 179
8.13.3.7. DE1_RDR 寄存器 179
8.13.3.8. IER 寄存器 180
8.13.3.9. ISR 寄存器 181
8.13.3.10. PROCR 寄存器 183
8.13.3.11. DE2_CR 寄存器 184
8.13.3.12. DE2_TDBAR 寄存器 184
8.13.3.13. DE2_CTDAR 寄存器 185
8.13.3.14. DE2_TDR 寄存器 185
8.13.3.15. DE2_RDBAR 寄存器 185
8.13.3.16. DE2_CRDAR 寄存器 186
8.13.3.17. DE2_RDR 寄存器 186
8.13.3.18. TXENG_CA_KR 寄存器 186
8.13.3.19. TXENG1_CA_SR 寄存器 187
8.13.3.20. TXENG2_CA_SR 寄存器 187
8.13.3.21. TXENG3_CA_SR 寄存器 187
8.13.3.22. TXENG_FIDLE_KR 寄存器 188
8.13.3.23. FSM_DE1 寄存器 188
8.13.3.24. FSM_DE2 寄存器 189
8.13.3.25. FSM_DE3 寄存器 189
8.13.3.26. FSM_PMH 寄存器 190
8.13.3.27. PMH_Debug1 寄存器 190
8.13.3.28. PMH_Debug2 寄存器 191
8.13.3.29. PMH_Debug3 寄存器 191
8.13.3.30. ESA_Debug_FSM 寄存器 191
8.13.3.31. ESA_Debug_CNT1 寄存器 192
8.13.3.32. ESA_Debug_CNT2 寄存器 192
8.13.3.33. SCR 寄存器 192
8.13.3.34. ZCCR 寄存器 193
8.13.3.35. PDCR 寄存器 193
8.13.3.36. GCCR 寄存器 193
8.13.3.37. PHY_CR1 寄存器 194
8.13.3.38. PHY_CR2 寄存器 194
8.13.3.39. PHY_IR1 寄存器 195

8.13.3.40. PHY_IR2 寄存器 195
8.13.3.41. PHY_CR3 寄存器 195
8.13.3.42. PHY_IR3 寄存器 196
8.13.3.43. RTL_VER 寄存器 196
8.13.3.44. ASIC_VER 寄存器 196
8.13.3.45. DE3_CR 寄存器 197
8.13.3.46. DE3_TDBAR 寄存器 197
8.13.3.47. DE3_CTDAR 寄存器 198
8.13.3.48. DE3_TDR 寄存器 198
8.13.3.49. DE3_RDBAR 寄存器 198
8.13.3.50. DE3_CRDAR 寄存器 199
8.13.3.51. DE3_RDR 寄存器 199
8.13.3.52. ECHO_CSR_CR 寄存器 199
8.13.3.53. ECHO_DMA_SWRSTCR 寄存器 200
8.13.3.54. ECHO_RXENG 寄存器 200
8.13.3.55. ECHO_RSVD_CSR 寄存器 202
8.14. CRC 204
8.14.1. 概述 204
8.14.2. CRC 加速器寄存器（Cortex-M4） 204
8.14.2.1. CRC_MODE 寄存器 204
8.14.2.2. CRC_POLY 寄存器 205
8.14.2.3. CRC_CODE 寄存器 205
8.14.2.4. CRC_ENTRY 寄存器 205
8.14.2.5. CRC_bLEN 寄存器 206
8.14.2.6. CRC_INIT_DATA 寄存器 206
8.14.3. CRC 加速器寄存器（Cortex-M0） 206
8.14.3.1. CRC_MODE 寄存器 206
8.14.3.2. CRC_POLY 寄存器 207
8.14.3.3. CRC_CODE 寄存器 208
8.14.3.4. CRC_ENTRY 寄存器 208
8.14.3.5. CRC_bLEN 寄存器 208
8.14.3.6. CRC_INIT_DATA 寄存器 209
8.15. SPI Flash 控制器 210
8.15.1. 概述 210
8.15.2. 功能框图 210
8.15.3. 寄存器摘要 210
8.15.4. 寄存器描述 211
8.15.4.1. FLASH_CMD 寄存器 211
8.15.4.2. FLASH_ADDR 寄存器 213
8.15.4.3. FLASH_CFG 寄存器 213
8.15.4.4. FLASH_CACHE 寄存器 215
8.15.4.5. FLASH_SR 寄存器 216
8.15.4.6. FLASH_ID 寄存器 216
8.15.4.7. FLASH_CACHEHIT 寄存器 217
8.15.4.8. FLASH_MAPOFFSET 寄存器 217
8.15.4.9. FLASH_BUFn 寄存器 218
8.16. 过零检测 220

8.16.1. 概述 220
8.16.2. 寄存器摘要 220
8.16.3. 寄存器描述 221
8.16.3.1. PH0_CTRL 寄存器 221
8.16.3.2. PH0_ZTIME0 寄存器 221
8.16.3.3. PH0_ZTIME1 寄存器 222
8.16.3.4. PH0_ZTIME2 寄存器 222
8.16.3.5. PH0_ZTIME3 寄存器 222
8.16.3.6. PH1_CTRL 寄存器 223
8.16.3.7. PH1_ZTIME0 寄存器 223
8.16.3.8. PH1_ZTIME1 寄存器 224
8.16.3.9. PH1_ZTIME2 寄存器 224
8.16.3.10. PH1_ZTIME3 寄存器 224
8.16.3.11. PH2_CTRL 寄存器 225
8.16.3.12. PH2_ZTIME0 寄存器 225
8.16.3.13. PH2_ZTIME1 寄存器 226
8.16.3.14. PH2_ZTIME2 寄存器 226
8.16.3.15. PH2_ZTIME3 寄存器 226
8.16.3.16. ZCDPHSEL 寄存器 227
8.17. 模拟前端 227
8.17.1. 概述 227
8.17.2. 特性 228
8.17.3. 功能描述 229
8.17.3.1. 总览 229
8.17.3.2. 省电模式 229
8.17.3.3. TX DAC 操作 230
8.17.3.4. PGA 增益控制操作 230
8.17.4. 信号描述 232
8.17.4.1. 电源引脚 232
8.17.4.2. 信号引脚 233
8.17.5. 电器规格 236
8.17.5.1. 直流特性 236
8.17.5.2. TX 路径特性 236
8.17.5.3. RX 路径特性 237
8.17.6. 时序特性 239
8.17.6.1. RX 239
8.17.6.2. TX 241
8.17.7. 电源和接地要求 243

9. 封装（QFN-68） 245

图索引

图 1-1. V6201B 功能框图.....23
图 1-2. 终端节点应用实例 – 单相电表.....23
图 1-3. 终端节点应用实例 – 三相电表.....23
图 1-4. 主模式 SPI 时序图.....27
图 1-5. 从模式 SPI 时序图.....28
图 1-6. UART0 时序图.....28
图 1-7. UART1 接收机时序图.....28
图 1-8. UART1 发送机时序图.....28
图 1-9. I^2C 时序图.....30
图 2-1. QFN-68 引脚分布图.....31
图 3-1. V6201B 上电流程图.....52
图 4-1. V6201B 时钟架构.....54
图 4-2. 通用复位系统.....59
图 5-1. Cortex-M4 AHB 总线的内存映像配置图.....68
图 5-2. Cortex-M0 AHB 总线的内存映像配置图.....69
图 6-1. 启动重映射图示.....70
图 7-1. Bootstrap.....72
图 8-1. 回送模式下 MCR 和 MSR 之间的相互作用.....106
图 8-2. TI SSP 单个传输时的帧格式.....113
图 8-3. TI SSP 连续传输时的帧格式.....113
图 8-4. 具有 SCLKPO 和 SCLKPH 不同组合的 SPI 帧格式.....114
图 8-5. 单次传输的摩托罗拉 SPI 帧格式.....115
图 8-6.摩托罗拉 SPI 帧格式（SCLKPH=0）的连续传输.....115
图 8-7.摩托罗拉 SPI 帧格式（SCLKPH=1）的连续传输.....115
图 8-8. 长度带循环填充的连续传输.....116
图 8-9. MICROWIRE 单个传输时的帧格式.....117
图 8-10. MICROWIRE 连续传输时的帧格式.....117
图 8-11. I^2C 功能框图.....128
图 8-12. TGSR，SCL，and SDA 之间的关系.....136
图 8-13. 等待状态后从机释放 SCL 时，TSR、SCL、SDA 的关系.....136

图 8-14. 功能框图 138
图 8-15. 定时器操作示例 143
图 8-16. 双缓冲函数的举例 144
图 8-17. PWM 函数示例 145
图 8-18. 启用的死区功能的波形 145
图 8-19. 功能框图 146
图 8-20. 功能模块图 153
图 8-21. 连接图 154
图 8-22. 写 GPIO 数据输出寄存器 163
图 8-23. 下降沿触发 164
图 8-24. 高电平触发 164
图 8-25. DMA 框图 165
图 8-26. SPI 闪存控制器功能框图 210
图 8-27. AFE 功能图 229
图 8-28. 正常工作状态下 RX 的时序图 240
图 8-29. 掉电模式下 RX 的唤醒时序图 241
图 8-30. 待机模式下 RX 的唤醒时序图 241
图 8-31. 正常工作状态下 DAC 的时序图 242
图 8-32. 掉电模式下 DAC 的唤醒时序图 243
图 8-33. 待机模式下 DAC 的唤醒时序图 243

表索引

表 1-1. 绝对最大额定值 24
表 1-2. 电路特性 25
表 1-3. Flash 内存规格 26
表 1-4. 主模式 SPI（MSPI）时序规格 26
表 1-5. 从模式 SPI（SSPI）时序规格 27
表 1-6. 正常模式 I^2C 时序特性 29
表 1-7. 快速模式 I^2C 时序特性 29
表 2-1. QFN-68（V6201B）引脚说明 33
表 2-2. IODRVCTRL0 寄存器 36
表 2-3. IODRVCTRL1 寄存器 39
表 2-4. IODRVCTRL2 寄存器 39
表 2-5. pmur_mfs0 寄存器（地址：0x54000820） 42
表 2-6. pmur_mfs1 寄存器（地址：0x54000824） 44
表 2-7. pmur_mfs2 寄存器（地址：0x54000864） 47
表 2-8. pmur_mfs3 寄存器（地址：0x54000880） 49
表 2-9. ODMODE 寄存器（地址：0x5400087C） 50
表 3-1. 上电时序说明 52
表 4-1. 外部时钟源 55
表 4-2. PLL 默认频率及编程值 55
表 4-3. PDIV_VALUE 寄存器（地址：0x54000838） 55
表 4-4. AHBCLKG 寄存器（地址：0x54000050） 56
表 4-5. APBCLKG 寄存器（地址：0x54000060） 57
表 4-6. 通用复位事件 59
表 4-7. HWRST_MASK 寄存器（地址：0x54000800） 60
表 4-8. HWRST_MASK 寄存器（地址：0x54000804） 61
表 4-9. SWRST_MASK 寄存器（地址：0x54000808） 62
表 4-10. SWRST_CLR 寄存器（地址：0x54000878） 63
表 4-11. SWRST_STS 寄存器（地址：0x54000884） 63
表 5-1. Cortex-M4 AHB/APB 总线上的寄存器空间地址映像 65
表 5-2. 在 Cortex-M0 AHB 总线上的 AHB 主机模块列表 66

表 5-3. Cortex-M0 AHB / APB 总线上的存储器空间地址映像.....66
表 7-1. Bootstrap 功能介绍.....71
表 7-2. STRAP 寄存器（地址：0x54000850）.....74
表 7-3. HWSTRAP 寄存器（地址：0x54000854）.....75
表 8-1. HWVCODE 寄存器（地址：0x700008FC）.....76
表 8-2. CPU TXEV/ RXEV 连接信号.....77
表 8-3. Cortex-M0 中断源.....77
表 8-4. 握手模式下的 VGDMA 请求与确认.....78
表 8-5. STCALIB 寄存器（地址：0x54020028）.....79
表 8-6. PCLKENPH 寄存器（地址：0x5402002C）.....79
表 8-7. STCLK_div 寄存器（地址：0x54020030）.....79
表 8-8. CPU TXEV/RXEV 连接讯号.....80
表 8-9. Cortex-M4 中断源.....80
表 8-10. TSVALUEB1 寄存器（地址：0x5400082C）.....82
表 8-11. TSVALUEB2 寄存器（地址：0x54000830）.....82
表 8-12. STCALIB1 寄存器（地址：0x5400083C）.....83
表 8-13. STCALIB2 寄存器（地址：0x54000840）.....83
表 8-14. STCALIB3 寄存器（地址：0x54000844）.....84
表 8-15. BOOTUPSTATUS 寄存器（地址：0x54000000）.....86
表 8-16. PWRMODE 寄存器（地址：0x54000020）.....86
表 8-17. SWRSTCTRL 寄存器（地址：0x5400002C）.....87
表 8-18. PLLCTRL 寄存器（地址：0x54000030）.....88
表 8-19. WAKEUPEN 寄存器（地址：0x5400081C）.....89
表 8-20. IPMISC_CTRL 寄存器（地址：0x5400084C）.....90
表 8-21. UART_CLKCTRL 寄存器（地址：0x54000868）.....91
表 8-22. GLB_COUNTER 寄存器（地址：0x54000858）.....91
表 8-23. VSBCTRL0 寄存器（地址：0x5400085C）.....92
表 8-24. BSTREN 寄存器（地址：0x54000860）.....92
表 8-25. SWRST_CLR 寄存器（地址：0x54000878）.....93
表 8-26. CUSTREG4 寄存器（地址：0x54000888）.....93
表 8-27. CUSTREG0 寄存器（地址：0x700008F0）.....93
表 8-28. 中断控制寄存器列表.....95

表 8-29. UART 寄存器摘要 98
表 8-30. RBR 寄存器（地址：0x000 读模式） 99
表 8-31. THR 寄存器（地址：0x000 写模式） 99
表 8-32. IER 寄存器（地址：0x004） 100
表 8-33. IIR 寄存器（地址：0x008 读模式） 101
表 8-34. 中断控制表说明 101
表 8-35. FCR 寄存器（地址：0x008） 103
表 8-36. 接收器的 FIFO 触发电平 103
表 8-37. 发送器的 FIFO 触发电平 104
表 8-38. LCR 寄存器（地址：0x00C） 104
表 8-39. 奇偶设定表说明 105
表 8-40. 字长度与停止位设定说明 105
表 8-41. MCR 寄存器（地址：0x010） 106
表 8-42. LSR 寄存器（地址：0x014） 107
表 8-43. MSR 寄存器（地址：0x018） 108
表 8-44. DLL 寄存器（地址：0x00，当 DLAB = 1 时） 109
表 8-45. DLM 寄存器（地址：0x04，当 DLAB = 1 时） 109
表 8-46. PSR 寄存器（地址：0x08，当 DLAB=1 时） 110
表 8-47. MDR 寄存器（地址：0x020） 110
表 8-48. FIFO 状态触发电平 111
表 8-49. SCLKDIV 中 SSPCLK 及 SCLK 的设置及比率 112
表 8-50. SSP 寄存器摘要 117
表 8-51. CR0 寄存器（地址：0x000） 118
表 8-52. CR1 寄存器（地址：0x004） 120
表 8-53. CR2 寄存器（地址：0x008） 120
表 8-54. Status 寄存器（地址：0x00C） 122
表 8-55. IntrCR 寄存器（地址：0x010） 123
表 8-56. IntrStatus 寄存器（地址：0x014） 124
表 8-57. TxRxDR 寄存器（地址：0x018） 125
表 8-58. CR3 寄存器（地址：0x01C） 126
表 8-59. I^2C 控制器的寄存器总表 128
表 8-60. CR 寄存器（地址：0x54040000） 129

表 8-61. SR 寄存器（地址：0x54040004）.....131
表 8-62. CDR 寄存器（地址：0x54040008）.....133
表 8-63. DR 寄存器（地址：0x5404000C）.....134
表 8-64. AR 寄存器（地址：0x54040010）.....135
表 8-65. TGSR 寄存器（地址：0x54040014）.....135
表 8-66. BMR 寄存器（地址：0x54040018）.....136
表 8-67. BSTMR 寄存器（地址：0x5404001C）.....137
表 8-68. Timer 控制器的寄存器摘要.....139
表 8-69. INT_CSTAT 寄存器（地址：0x54030000（Cortex-M4），0x54030000（Cortex-M0）） 140
表 8-70. TMn_CTRL 寄存器（地址：0x540300n0 （Cortex-M4），0x540300n0 （Cortex-M0）） 141
表 8-71. TMn_CNTB 寄存器（地址：0x540300n4（Cortex-M4），0x540300n4（Cortex-M0）） 142
表 8-72. TMn_CMPB 寄存器（地址：0x540300n8（Cortex-M4），0x540300n8（Cortex-M0）） 142
表 8-73. 看门狗控制器寄存器列表.....147
表 8-74. WdCounter 寄存器（地址：0x54020000）.....148
表 8-75. WdLoad 寄存器（地址：0x54020004）.....148
表 8-76. WdRestart 寄存器（地址：0x54020008）.....149
表 8-77. WdCR 寄存器（地址：0x5402000C）.....149
表 8-78. WdStatus 寄存器（地址：0x54020010）.....150
表 8-79. WdClear 寄存器（地址：0x54020014）.....150
表 8-80. WdIntrlen 寄存器（地址：0x54020018）.....151
表 8-81. GPIO 寄存器概况.....154
表 8-82. GpioDataOut 寄存器（地址：0x54050000 （Cortex-M4），0x54060000 （Cortex-M0））
.....156
表 8-83. GpioDataIn 寄存器（地址：0x54050004（Cortex-M4），0x54060004（Cortex-M0）） 157
表 8-84. GpioDataSet 寄存器（地址：0x54050010（Cortex-M4），0x54060010（Cortex-M0））
.....157
表 8-85. GpioDataClear 寄存器（地址：0x54050014（Cortex-M4），0x54060014（Cortex-M0））
.....158
表 8-86. PinPullEnable 寄存器（地址：0x54050018（Cortex-M4），0x54060018（Cortex-M0））
.....158
表 8-87. PinPullType 寄存器（地址：0x5405001C（Cortex-M4），0x5406001C（Cortex-M0））
.....158
表 8-88. IntrEnable 寄存器（地址：0x54050020（Cortex-M4），0x54060020（Cortex-M0）） 159

表 8-89. IntrRawState 寄存器（地址：0x54050024（Cortex-M4），0x54060024（Cortex-M0））.....159
表 8-90. IntrMaskedState 寄存器（地址：0x54050028（Cortex-M4），0x54060028（Cortex-M0））.....160
表 8-91. IntrMask 寄存器（地址：0x5405002C（Cortex-M4），0x5406002C（Cortex-M0））.....160
表 8-92. IntrClear 寄存器（地址：0x54050030（Cortex-M4），0x54060030（Cortex-M0））.....161
表 8-93. IntrTrigger 寄存器（地址：0x54050034（Cortex-M4），0x54060034（Cortex-M0））.....161
表 8-94. IntrBoth 寄存器（地址：0x54050038（Cortex-M4），0x54060038（Cortex-M0））.....161
表 8-95. IntrRiseNeg 寄存器（地址：0x5405003C（Cortex-M4），0x5406003C（Cortex-M0））.....162
表 8-96. BounceEnable 寄存器（地址：0x54050040（Cortex-M4），0x54060040（Cortex-M0））.....162
表 8-97. BouncePreScale 寄存器（地址：0x54050044（Cortex-M4），0x54060044（Cortex-M0））.....163
表 8-98. Chn_MODE 寄存器（地址：0x57000000+n*0x40）.....165
表 8-99. CHn_ADDRSRC 寄存器（地址：0x57000004+n*0x40）.....166
表 8-100. CHn_ADDRDES 寄存器（地址：0x57000008+n*0x40）.....166
表 8-101. CHn_BLENGTH 寄存器（地址：0x5700000C+n*0x40）.....167
表 8-102. CHn_PKTLENGTH 寄存器（地址：0x57000010+n*0x40）.....167
表 8-103. DMA_RESET 寄存器（地址：0x57000038）.....168
表 8-104. DMA_INTC 寄存器（地址：0x5700003C）.....168
表 8-105. Chn_MODE 寄存器（地址：0x77000000+n*0x40）.....170
表 8-106. CHn_ADDRSRC 寄存器（地址：0x77000004+n*0x40）.....170
表 8-107. CHn_ADDRDES 寄存器（地址：0x77000008+n*0x40）.....171
表 8-108. CHn_BLENGTH 寄存器（地址：0x7700000C+n*0x40）.....171
表 8-109. CHn_PKTLENGTH 寄存器（地址：0x77000010+n*0x40）.....171
表 8-110. DMA_RESET 寄存器（地址：0x77000038）.....172
表 8-111. DMA_INTC 寄存器（地址：0x7700003C）.....172
表 8-112. 安全加速器 DMA CSR 概述.....175
表 8-113. DE1_CR 寄存器（地址：0x59000000）.....177
表 8-114. DE1_TDBAR 寄存器.....178
表 8-115. DE1_CTDAR 寄存器.....178
表 8-116. DE1_TDR 寄存器.....178

表 8-117. DE1_RDBAR 寄存器.....179
表 8-118. DE1_CRDAR 寄存器.....179
表 8-119. DE1_RDR 寄存器.....179
表 8-120. IER 寄存器.....180
表 8-121. ISR 寄存器.....182
表 8-122. PROCR 寄存器.....183
表 8-123. DE2_CR 寄存器.....184
表 8-124. DE2_TDBAR 寄存器.....184
表 8-125. DE2_CTDAR 寄存器.....185
表 8-126. DE2_TDR 寄存器.....185
表 8-127. DE2_RDBAR 寄存器.....185
表 8-128. DE2_CRDAR 寄存器.....186
表 8-129. DE2_RDR 寄存器.....186
表 8-130. TXENG_CA_KR 寄存器.....187
表 8-131. TXENG1_CA_SR 寄存器.....187
表 8-132. TXENG2_CA_SR 寄存器.....187
表 8-133. TXENG3_CA_SR 寄存器.....188
表 8-134. TXENG_FIDLE_KR 寄存器.....188
表 8-135. FSM_DE1 寄存器.....188
表 8-136. FSM_DE2 寄存器.....189
表 8-137. FSM_DE3 寄存器.....189
表 8-138. FSM_PMH 寄存器.....190
表 8-139. PMH_Debug1 寄存器.....190
表 8-140. PMH_Debug2 寄存器.....191
表 8-141. PMH_Debug3 寄存器.....191
表 8-142. ESA_Debug_FSM 寄存器.....191
表 8-143. ESA_Debug_CNT1 寄存器.....192
表 8-144. ESA_Debug_CNT2 寄存器.....192
表 8-145. SCR 寄存器.....192
表 8-146. ESA_Debug_CNT2 寄存器.....193
表 8-147. PDCR 寄存器.....193
表 8-148. GCCR 寄存器.....193

表 8-149. PHY_CR1 寄存器 194
表 8-150. PHY_CR2 寄存器 194
表 8-151. PHY_IR1 寄存器 195
表 8-152. PHY_IR2 寄存器 195
表 8-153. PHY_CR3 寄存器 195
表 8-154. PHY_IR3 寄存器 196
表 8-155. RTL_VER 寄存器 196
表 8-156. ASIC_VER 寄存器 196
表 8-157. DE3_CR 寄存器 197
表 8-158. DE3_TDBAR 寄存器 197
表 8-159. DE3_CTDAR 寄存器 198
表 8-160. DE3_TDR 寄存器 198
表 8-161. DE3_RDBAR 寄存器 199
表 8-162. DE3_CRDAR 寄存器 199
表 8-163. DE3_RDR 寄存器 199
表 8-164. ECHO_CSR_CR 寄存器 200
表 8-165. ECHO_DMA_SWRSTCR 寄存器 200
表 8-166. ECHO_RXENG 寄存器 201
表 8-167. ECHO_RSVD_CSR 寄存器 202
表 8-168. CRC_MODE 寄存器（地址：0x58000000） 204
表 8-169. CRC_POLY 寄存器（地址：0x58000004） 205
表 8-170. CRC_CODE 寄存器（地址：0x58000008） 205
表 8-171. CRC_ENTRY 寄存器（地址：0x5800000C） 205
表 8-172. CRC_bLEN 寄存器（地址：0x58000010） 206
表 8-173. CRC_INIT_DATA 寄存器（地址：0x58000014） 206
表 8-174. CRC_MODE 寄存器（地址：0x7000F000） 207
表 8-175. CRC_POLY 寄存器（地址：0x7000F004） 207
表 8-176. CRC_CODE 寄存器（地址：0x7000F008） 208
表 8-177. CRC_ENTRY 寄存器（地址：0x7000F00C） 208
表 8-178. CRC_bLEN 寄存器（地址：0x7000F010） 208
表 8-179. CRC_INIT_DATA 寄存器（地址：0x7000F014） 209
表 8-180. SPI 闪存控制器的寄存器列表 210

表 8-181. FLASH_CMD 寄存器（地址：0x60FFF000）.....211
表 8-182. FLASH_ADDR 寄存器（地址：0x60FFF004）.....213
表 8-183. FLASH_CFG 寄存器（地址：0x60FFF008）.....213
表 8-184. FLASH_CACHE 寄存器（地址：0x60FFF00C）.....215
表 8-185. FLASH_SR 寄存器（地址：0x60FFF010）.....216
表 8-186. FLASH_ID 寄存器（地址：0x60FFF014）.....216
表 8-187. FLASH_CACHEHIT 寄存器（地址：0x60FFF018）.....217
表 8-188. FLASH_MAPOFFSET 寄存器（地址：0x60FFF020）.....218
表 8-189. FLASH_BUFn 寄存器（地址：0x60FFF100 ~ 0x60FFF1FC）.....218
表 8-190. 过零检测寄存器列表.....220
表 8-191. PH0_CTRL 寄存器（地址：0x70000000）.....221
表 8-192. PH0_ZTIME0 寄存器（地址：0x70000004）.....222
表 8-193. PH0_ZTIME1 寄存器（地址：0x70000008）.....222
表 8-194. PH0_ZTIME2 寄存器（地址：0x7000000C）.....222
表 8-195. PH0_ZTIME3 寄存器（地址：0x70000010）.....223
表 8-196. PH1_CTRL 寄存器（地址：0x70000014）.....223
表 8-197. PH1_ZTIME0 寄存器（地址：0x70000018）.....224
表 8-198. PH1_ZTIME1 寄存器（地址：0x7000001C）.....224
表 8-199. PH1_ZTIME2 寄存器（地址：0x70000020）.....224
表 8-200. PH1_ZTIME3 寄存器（地址：0x70000024）.....224
表 8-201. PH2_CTRL 寄存器（地址：0x70000028）.....225
表 8-202. PH2_ZTIME0 寄存器（地址：0x7000002C）.....225
表 8-203. PH2_ZTIME1 寄存器（地址：0x70000030）.....226
表 8-204. PH2_ZTIME2 寄存器（地址：0x70000034）.....226
表 8-205. PH2_ZTIME3 寄存器（地址：0x70000038）.....226
表 8-206. ZCDPHSEL 寄存器（地址：0x7000003C）.....227
表 8-207. PGA 增益控制设置.....230
表 8-208. 电源引脚.....232
表 8-209. 信号引脚.....233
表 8-210. 推荐工作条件.....236
表 8-211. TX 路径特性.....236
表 8-212. RX 路径特性.....237

表 8-213. RX 输入阻抗 239

表 8-214. RX 时序规格 240

表 8-215. TX 时序规格 242

1. 概要

1.1. 概述说明

V6201X 是一个兼具宽带及窄带电力线通信（PLC）的处理器芯片。V6201X 集成了一个 ARM M4 内核，一个 ARM M0 内核，一个高性能 DSP，最多三个 UART 接口，一个 SPI Master/Slave 控制器，一个 SPI Master 接口，一个 SPI Slave 接口，以及 PLC MAC/PHY 层功能和模拟前端，搭配 Vango 的线路驱动器芯片（V6100D）。它不仅形成了一个完整的调制解调器解决方案，支持宽带及窄带 PLC 标准，它还通过完整的带宽解决方案支持 Vango 开发的新一代电力线通信标准。专用于电力线通信的 V6201X PLC 处理器芯片满足智能电网安装中 AMI 网络的需求。

1.2. 特性

- 支持窄带 PLC（OFDM）标准：G3-PLC
- 支持宽带 PLC（OFDM）标准：HPLC
- 支持频带：CENELEC，FCC，G3M（高达 2 MHz），及宽带 PLC（高达 12 MHz）
- 支持调制：DBPSK/BPSK，DQPSK/QPSK，D8PSK/8PSK，16QAM
- 支持 G3-PLC 相容的 6LoWPAN 适配层，及优化组网和网状路由功能
- 支持 2 个 AES Engines

  高速 Engine 支持 AES-128 with ECB/CBC/CCM 模式

  慢速 Engine 支持 AES-128/192/256 with ECB 模式
- 支持高达 2048 浮点 FFT/IFFT 加速器
- 支持三相过零检测
- 支持 3 个 M4 内核的 UART 接口（UART0~UART2）

  UART1 是一个高速的 UART 可支持波特率高达 500 Kbps

  UART0 及 UART2 可支持波特率高达 115200 bps
- 提供一个可配置的 SPI Master（Slave）接口：

  配备有两个 CS 引脚，可以来控制无线收发器，计量或其他 SPI 装置
- 提供另一个 SPI master 接口：配备有两个 CS 引脚，可以用来控制无线收发器，计量或其他 SPI 装置
- 提供一个 SPI slave 接口：提供主处理器芯片另一个数据接口的选择弹性
- 2MB FLASH 内存，（512KB+64KB）DRAM 内存
- 支持经由 UART0 或 SSP2 Slave 接口进行 Flash 内存的在线程序设计（ISP）
- 高达 24 个可程序设计的 GPIOs 发挥最大塑性
- 3.3V 数字 I/O；UART 引脚耐压 5V

- 封装：

  V6201B：QFN-68

  V6201R：QFN-52

- 工作温度：-40~+85°C

1.3. 功能框图

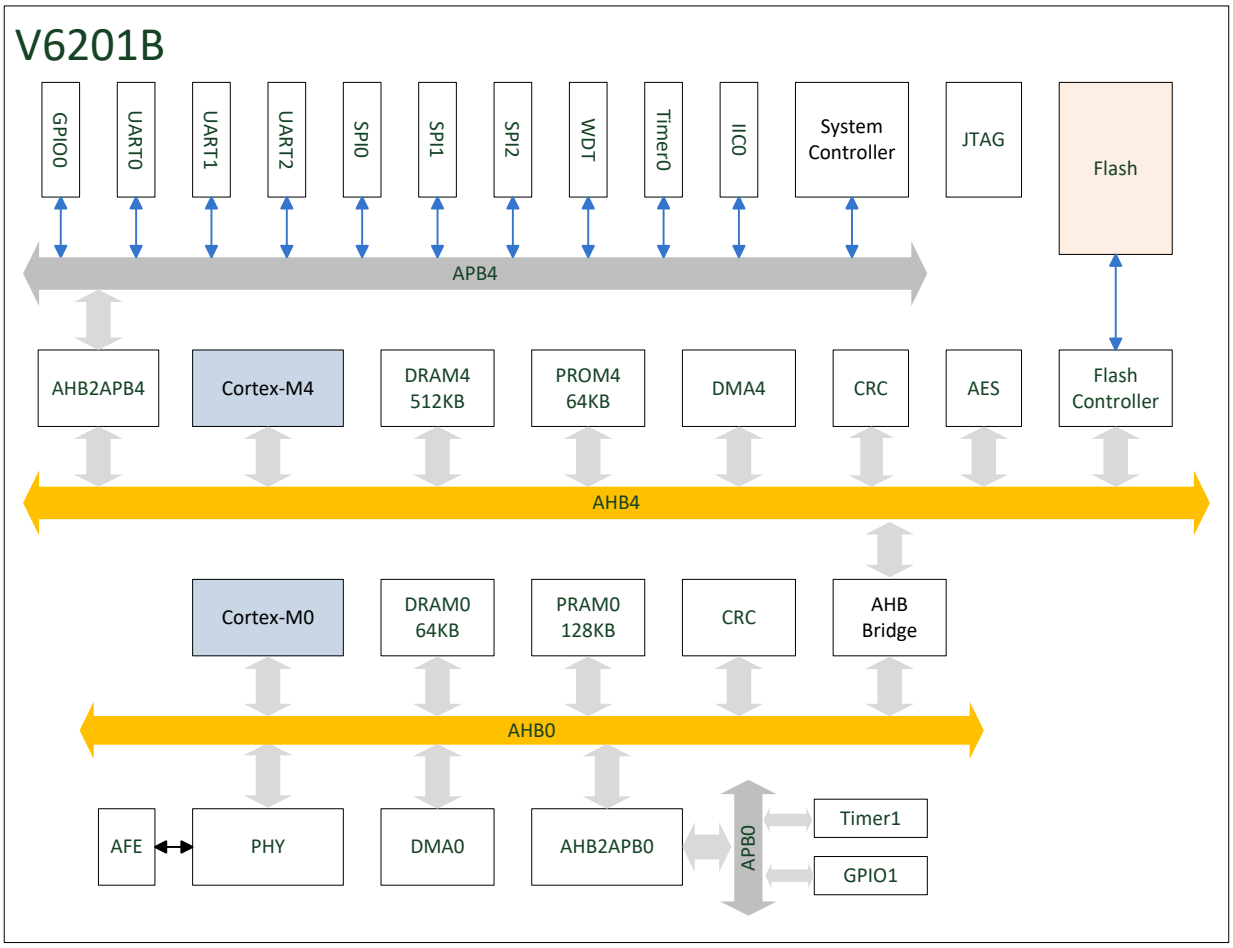


图 1-1. V6201X 功能框图

1.4. 系统框图

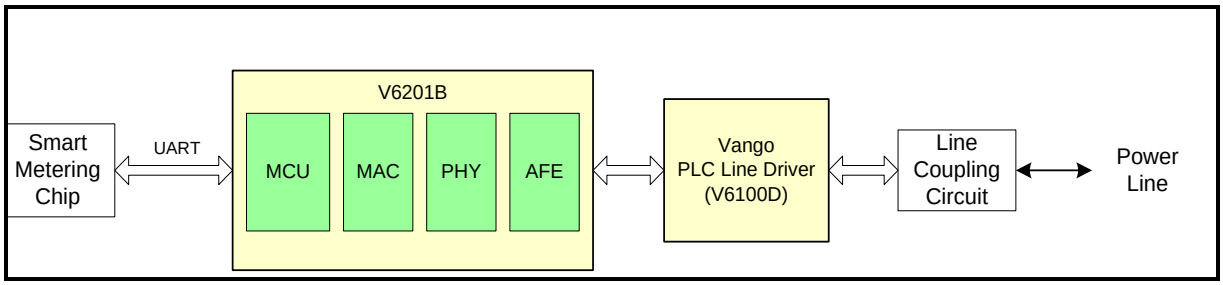


图 1-2. 终端节点应用实例 – 单相电表

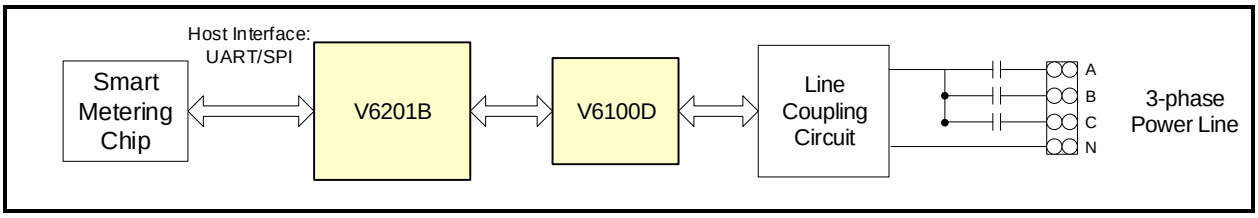


图 1-3. 终端节点应用实例 – 三相电表

1.5. 应用场景

- 智慧电网通信
- 高级计量基础设施（AMI）
- 智能表计
- 电力线通信数据集中器
- 路灯智控
- 智能家居

1.6. 电路特性

1.6.1. 绝对最大额定值

超出下列最大/最小值的工作条件可能会造成芯片的永久性损伤。

表 1-1. 绝对最大额定值

参数	最小	最大	单位	说明
VCC33A_AFE 引脚	-0.5	+4.6	V	模拟 3.3V 电源引脚 相对于地
VCC3IO_XTAL	-0.5	+4.6	V	PLL 1.1V 电源引脚
VCC3IO	-0.5	+4.6	V	数字 3.3V 电源引脚
VCC11A_PLL	-0.5	+1.5	V	PLL 1.1V 电源引脚
VCCK 引脚	-0.5	+1.5	V	内核电源引脚 相对于地
I/O 引脚	-0.5	+4.6	V	通用 I/O 引脚 相对于地
UART I/O 5V 引脚耐受度	-0.5	+5.8	V	5V I/O 引脚耐受度 相对于地
I_{IN}	-	+50	mA	直流输入电流
I_{OUT}	-	+50	mA	短路输出电流
工作温度	-40	+85	°C	-
储存温度	-65	+150	°C	-

结温	-	+125	°C	-
引脚温度（焊接，10 秒）	-	+270	°C	-

1.6.2. 电路特性

如无特别说明，以下表中所有最大/小值规格适用于整个推荐工作范围内（T = -40 °C ~ +85 °C）。以下表中所有典型值规格均在 TA = +25 °C 条件下测得。

表 1-2. 电路特性

参数	最小	典型	最大	单位	说明
VCC33A_AFE	2.97	3.3	3.63	V	-
VCC3IO_XTAL	2.97	3.3	3.63	V	-
VCC3IO	2.97	3.3	3.63	V	-
VCC11A_PLL	0.99	1.1	1.21	V	-
VCCK	0.99	1.1	1.21	V	-
工作电流（VCC33A_AFE）	-	130	150	mA	-
工作电流（VCC3IO）	-	8	-	mA	-
工作电流（VCCK）	-	90	150	mA	-
数字 IO，输出					
高电压输出，V_{OH}	2.4	-	-	V	-
I_{SOURCE}	-	4	16	mA	-
低电压输出，V_{OL}	-	-	0.4	V	-
I_{SINK}	-	4	16	mA	-
数字 IO，输入					
高电压输入，V_{IH}	2.0	-	-	V	-
低电压输入，V_{IL}	-	-	0.8	V	-
I/O 上拉电阻	40	75	190	KΩ	-
I/O 下拉电阻	30	75	190	KΩ	-

1.6.3. Flash 规格

如无另外说明，以下表中所有最大/小值规格适用于整个推荐工作范围内（T = -40 °C ~ +85 °C）。以下表中所有典型值规格均在 TA = +25 °C 条件下测得。

表 1-3. Flash 内存规格

参数	最小	典型	最大	单位	说明
Flash 内存					
读取时间	-	-	6	ns	VCC3IO= 2.97~3.63 V
擦/写次数	100k	-	-	cycle	-40 °C ~ +85 °C
数据保持时间	20	-	-	year	25 °C
页写入时间	-	0.4	3	ms	-
区擦除时间（4096 字节）	-	45	400	ms	-
全擦除时间	-	5	25	s	-

1.6.4. SPI 时序规格

如无另外说明，以下表中所有最大/小值规格适用于整个推荐工作范围内（T = -40 °C ~ +85 °C）。

表 1-4. 主模式 SPI（MSPI）时序规格

参数		最小	最大	单位
f_{MSPI}	MSPI_CLK 频率	-	8	MHz
t_{MCYC}	MSPI_CLK 周期时间	$1/f_{MSPI}$	-	ns
t_{MPH}	MSPI_CLK 脉冲高	$1/(2f_{MSPI})-5$	-	ns
t_{MPL}	MSPI_CLK 脉冲低	$1/(2f_{MSPI})-5$	-	ns
t_{MVLD}	SPI_CLK 上升之后 MSPI_MOSI 的有效时间	$1/(2f_{MSPI})$	-	ns
t_{MSETUP}	SPI_CLK 上升之前 MSPI_MISO 的设置时间	5	-	ns
t_{MHOLD}	SPI_CLK 上升之后 MSPI_MISO 的保持时间	5	-	ns
t_{MCS_SETUP}	SPI_CLK 上升之前 MSPI_CS 的设置时间	$1/f_{MSPI}$	-	ns
t_{MCS_HOLD}	SPI_CLK 下降之后 MSPI_CS 的保持时间	$1/f_{MSPI}$	-	ns

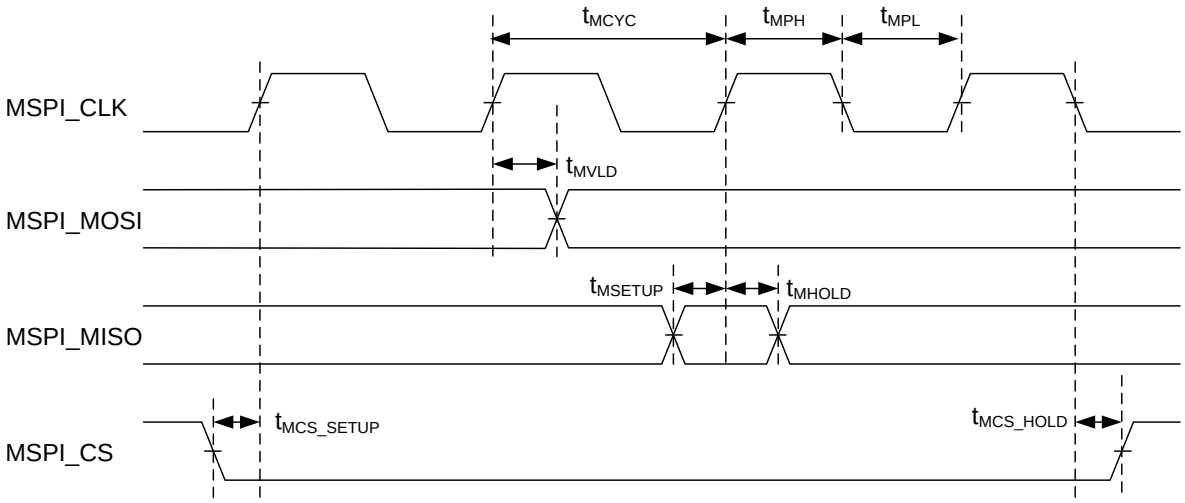


图 1-4. 主模式 SPI 时序图

表 1-5. 从模式 SPI（SSPI）时序规格

参数		最小	最大	单位
f_{SSPI}	SSPI_CLK 频率	-	8	MHz
t_{SCYC}	SSPI_CLK 时钟周期	$1/f_{SSPI}$	-	ns
T_{SPH}	SSPI_CLK 高电平时间	$1/(2f_{SSPI})$-5	-	ns
t_{SPL}	SSPI_CLK 低电平时间	$1/(2f_{SSPI})$-5	-	ns
t_{SVLD}	SSPI_CLK 时钟上升后 SSPI_MISO 的有效时间	$1/(2f_{SSPI})$	-	ns
T_{SSETUP}	SSPI_MOSI setup time before SSPI_CLK rising	5	-	ns
T_{SHOLD}	SSPI_MOSI hold time after SSPI_CLK rising	5	-	ns
T_{SCS_SETUP}	SSPI_CLK 时钟上升之前 SSPI_CS 建立时间	$1/f_{SSPI}$	-	ns
t_{SCS_HOLD}	SPI_CLK 下降之后 SSPI_CS 保持时间	$1/(2f_{SSPI})$	-	ns

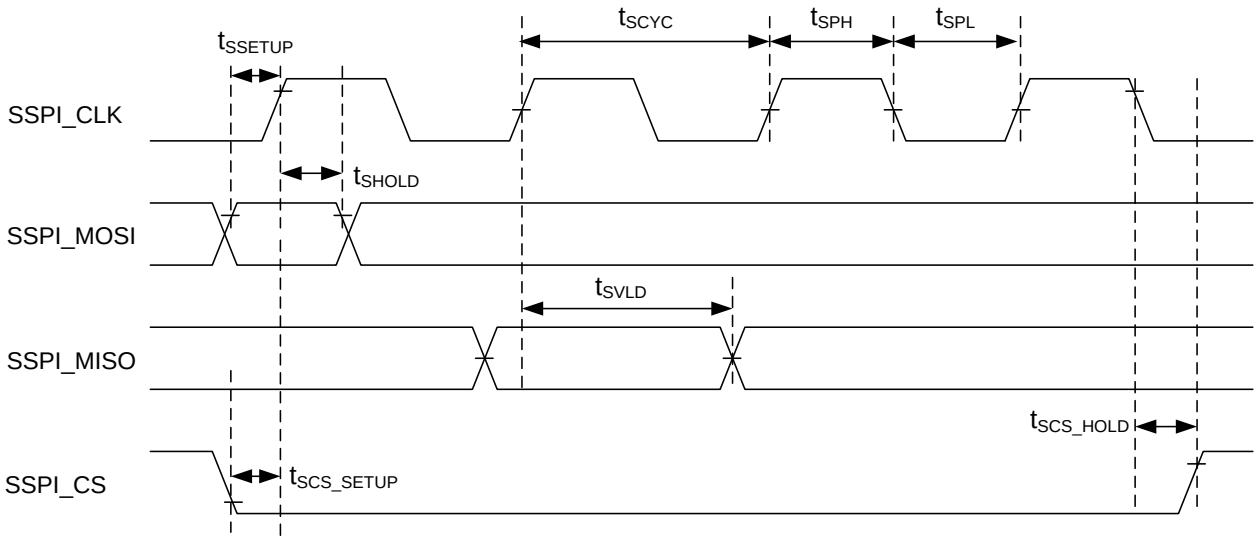


图 1-5. 从模式 SPI 时序图

1.6.5. UART 时序特性

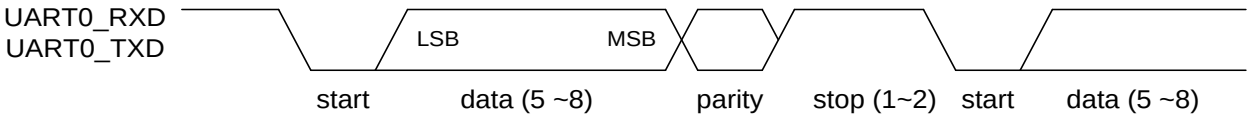


图 1-6. UART0 时序图

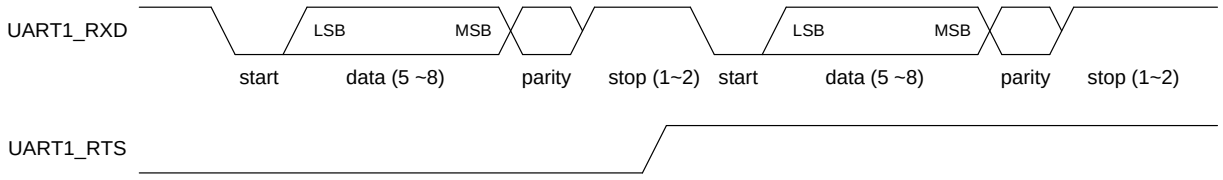


图 1-7. UART1 接收机时序图

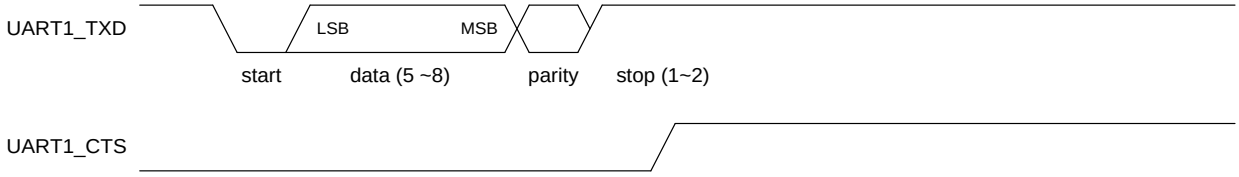


图 1-8. UART1 发送机时序图

1.6.6. I²C 时序特性

除另外注释，最大或最小值将适用在整个推荐的操作范围（T = -40 °C ~ +85 °C）。

表 1-6. 正常模式 I²C 时序特性

参数		最小值	最大值	单位
f_{SCL}	SCL 时钟频率	-	100	kHz
t_{SCL}	SCL 时钟周期	-	10	μs
t_{LOW}	SCL 低频时钟周期	4.7	-	μs
t_{HIGH}	SCL 高频时钟周期	4.0	-	μs
t_{FALL}	SCL 和 SDA 信号的下降时间	-	300	ns
t_{RISE}	SCL 和 SDA 信号的上升时间	-	1000	ns
t_{SU_STA}	重复一次启动状态的启动时间	4.7	-	μs
t_{HD_STA}	重复一次启动状态的持续时间	4.0	-	μs
t_{SU_DAT}	SDA 数据的启动时间	250	-	ns
t_{HD_DAT}	SDA 数据的持续时间	0	3.45	μs
t_{SU_STO}	停止状态的启动时间	4.0	-	μs
t_{BF}	停止状态与启动状态之间的总线空闲时间	4.7	-	μs

表 1-7. 快速模式 I²C 时序特性

参数		最小值	最大值	单位
f_{SCL}	SCL 时钟频率	-	400	kHz
t_{SCL}	SCL 时钟周期	-	2.5	μs
t_{LOW}	SCL 低频时钟周期	1.3	-	μs
t_{HIGH}	SCL 高频时钟周期	0.6	-	μs
t_{FALL}	SCL 和 SDA 信号的下降时间	-	300	ns
t_{RISE}	SCL 和 SDA 信号的上升时间	-	300	ns
t_{SU_STA}	重复一次启动状态的启动时间	0.6	-	μs
t_{HD_STA}	重复一次启动状态的持续时间	0.6	-	μs
t_{SU_DAT}	SDA 数据的启动时间	100	-	ns

t_{HD_DAT}	SDA 数据的持续时间	0	0.9	μs
t_{SU_STO}	停止状态的启动时间	0.6	-	μs
t_{BF}	停止状态与启动状态之间的总线空闲时间	1.3	-	μs

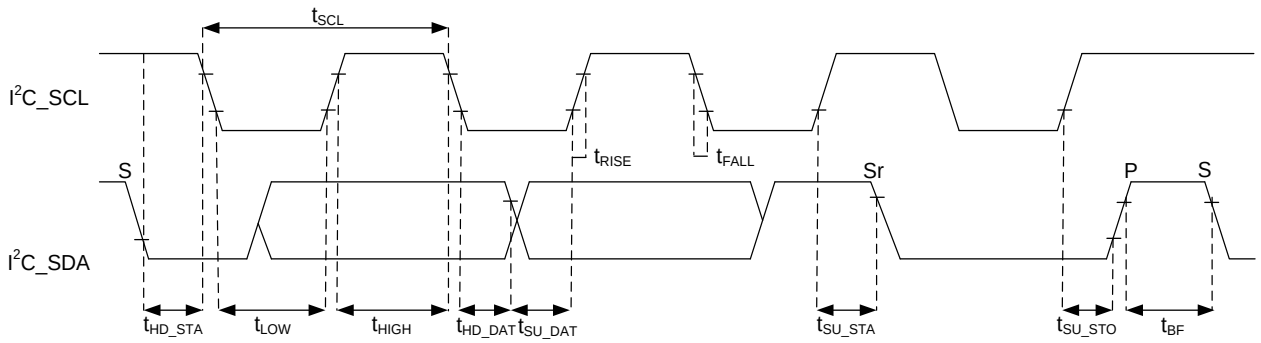


图 1-9. I²C 时序图

2.引脚说明

2.1. V6201B 引脚分布图

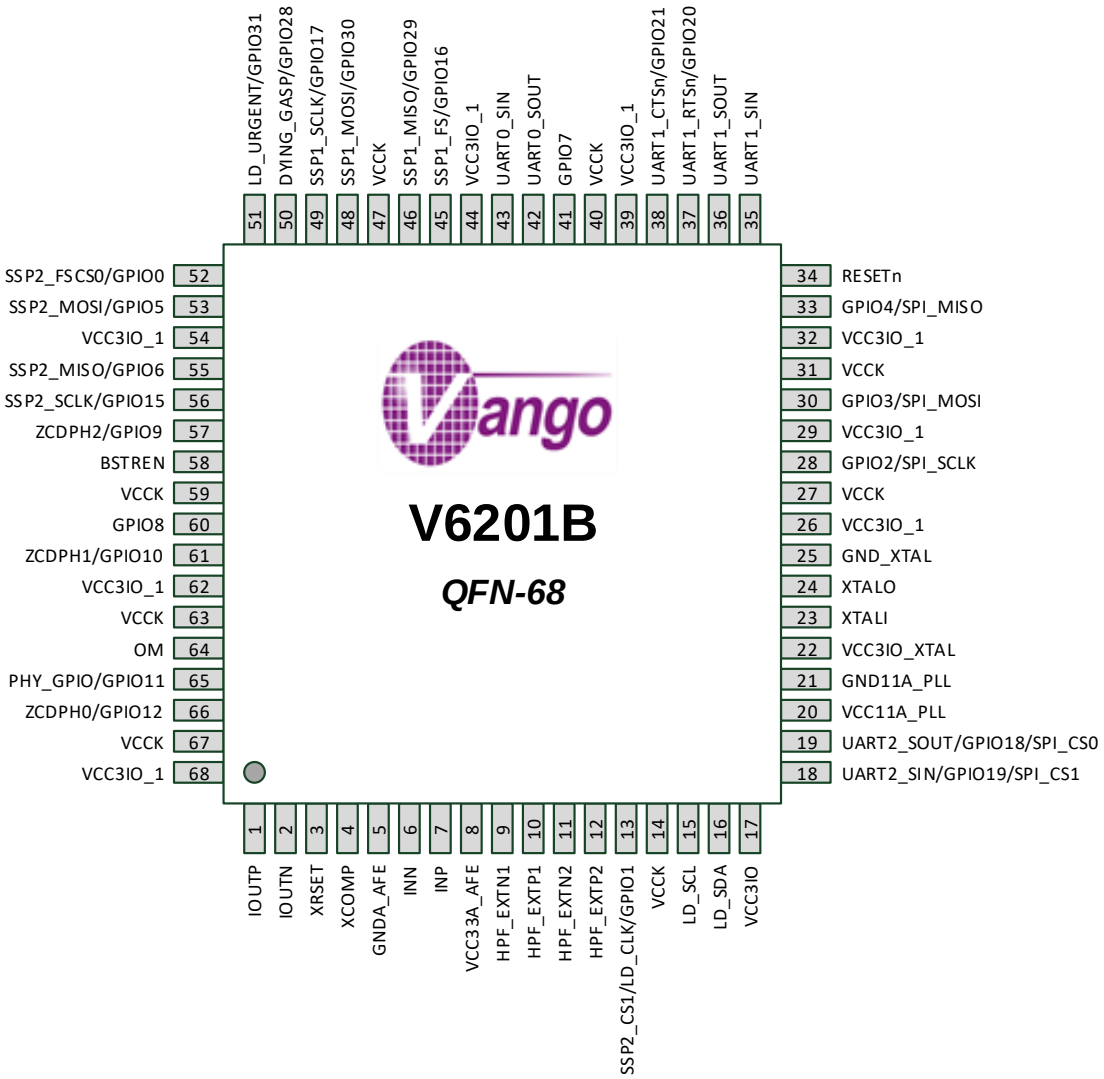


图 2-1. QFN-68 引脚分布图

2.2. V6201R 引脚分布图

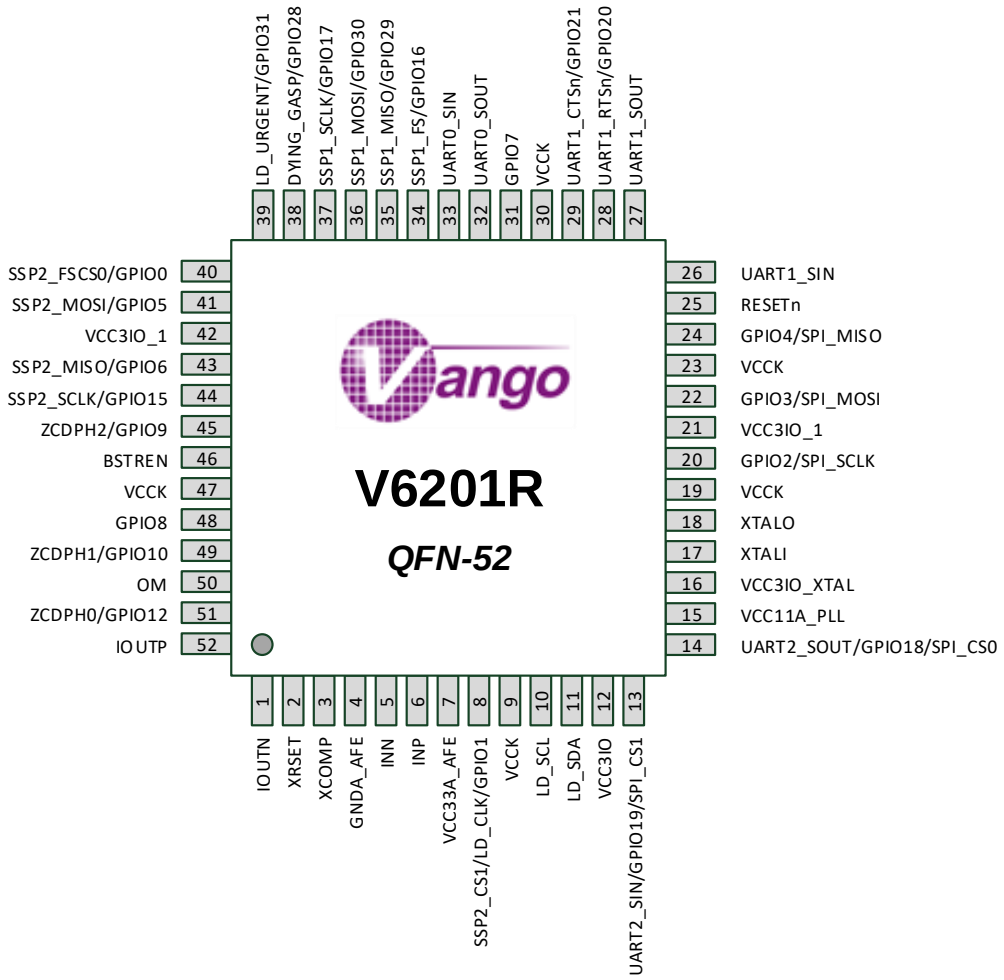


图 2-2. QFN-52 引脚分布图

2.3. V6201X 引脚说明

表 2-1. V6201X 引脚说明

（下表"类型"一栏中，I=输入；O=输出；P=电源；G=地。）

编号	V6201B	V6201R	名称	类型	功能说明
1	1	52	IOUTP	O	DAC 差分讯号正输出至线路驱动器芯片（V6100D）。
2	2	1	IOUTN	O	DAC 差分讯号负输出至线路驱动器芯片（V6100D）。
3	3	2	XRSET	O	可通过连接不同值的电阻器（RSET）来调节全量程输出电流。全量程输出电流的计算公式如下：IOUTFS = [(4095/80) * (VREF/RSET)]。例如：要获得 10 mA 的全量程电流，外部 RSET 应为 6.2 kΩ。
4	4	3	XCOMP	IO	带宽/降噪节点，新增一个对地 3.3μF 以上电容器（最小 ESR 限度>0.5 Ω）在 GNDA 上使其发挥最佳性能。
5	5	4	GNDA_AFE	G	模拟地。
6	6	5	INN	I	来自线路驱动器芯片的 ADC 差分讯号负输入。
7	7	6	INP	I	来自线路驱动器芯片的 ADC 差分讯号正输入。
8	8	7	VCC33A_AFE	P	模拟 3.3V 电源。
9	9		HPF_EXTN1	IO	高通滤波器外部管脚，PGA1 负输出。
10	10		HPF_EXTP1	IO	高通滤波器外部管脚，PGA1 正输出。
11	11		HPF_EXTN2	IO	高通滤波器外部管脚，PGA2 负输出。
12	12		HPF_EXTP2	IO	高通滤波器外部管脚，PGA2 正输出。
13	13	8	SSP2_CS1 LD_CLK GPIO1	IO O IO	Master/slave SSP2 主选择 1。 时钟输出至线路驱动器芯片（V6100D）。 通用 I/O。
14	14	9	VCCK	P	数字 1.1V 电源。
15	15	10	LD_SCL	O	线路驱动器芯片（V6100D）控制接口时钟输出。
16	16	11	LD_SDA	IO	线路驱动器芯片（V6100D）控制接口数据。
17	17	12	VCC3IO	P	数字 3.3V 电源。
18	18	13	UART2_SIN GPIO19 SPI_CS1	I IO O	UART2：数据输入。 通用 I/O。 Master SPI 0 选择 1。

编号	V6201B	V6201R	名称	类型	功能说明
19	19	14	UART2_SOUT GPIO18 SPI_CS0	O IO O	UART2：数据输出。 通用 I/O。 Master SSP0 选择 0。
20	20	15	VCC11A_PLL	P	模拟 PLL 1.1V 电源。
21	21		GND11A_PLL	G	模拟 PLL 模块地。
22	22	16	VCC3IO_XTAL	P	晶体震荡器 3.3V 电源。
23	23	17	XTALI	I	晶体振荡器的输入，25MHz。
24	24	18	XTALO	O	晶体振荡器的输出。
25	25		GND_XTAL	G	晶体振荡器地。
26	26		VCC3IO_1	P	数字 3.3V 电源。
27	27	19	VCCK	P	数字 1.1V 电源。
28	28	20	GPIO2 SSP_SCLK	IO O	通用 I/O。 Master SSP0 时钟输出。
29	29	21	VCC3IO_1	P	模拟 3.3V 电源。
30	30	22	GPIO3 SSP_MOSI	IO O	通用 I/O。 Master SSP0 数据输出。
31	31	23	VCCK	P	数字 1.1V 电源。
32	32		VCC3IO_1	P	数字 3.3V 电源。
33	33	24	GPIO4 SPI_MISO	IO I	通用 I/O。 Master SSP0 数据输入。
34	34	25	RESETn	I	复位输入（低电平有效）。
35	35	26	UART1_SIN	I	高速 UART 端口 1：数据输入。
36	36	27	UART1_SOUT	O	高速 UART 端口 1：数据输出。 默认为开漏设定，建议上拉电阻值 5.1KΩ.
37	37	28	UART1_RTSn GPIO20	O IO	高速 UART 端口 1：发送请求。 通用 I/O。
38	38	29	UART1_CTSn GPIO21	I IO	高速 UART 端口 1：清除发送。 通用 I/O。
39	39		VCC3IO_1	P	数字 3.3V 电源。
40	40	30	VCCK	P	数字 1.1V 电源。

编号	V6201B	V6201R	名称	类型	功能说明
41	41	31	GPIO7	IO	通用 I/O。
42	42	32	UART0_SOUT	O	UART 端口 0：数据输出。
43	43	33	UART0_SIN	I	UART 端口 0：数据输入。
44	44		VCC3IO_1	P	数字 3.3V 电源。
45	45	34	SSP1_FS GPIO16	I IO	Slave SSP1 帧同步输入。 通用 I/O。
46	46	35	SSP1_MISO GPIO29	O IO	Slave SSP1 数据输出。 通用 I/O。
47	47		VCCK	P	数字 1.1V 电源。
48	48	36	SSP1_MOSI GPIO30	I IO	Slave SSP1 数据输入。如不使用，需要调高。 通用 I/O。
49	49	37	SSP1_SCLK GPIO17	I IO	Slave SSP1 时钟输入。 通用 I/O。
50	50	38	DYING_GASP GPIO28	I IO	Dying gasp 输入。 通用 I/O。
51	51	39	LD_URGENT GPIO31	I IO	通讯线路过载保护讯号输入。 通用 I/O。
52	52	40	SSP2_FSCS0 GPIO0	IO IO	Master/slave SSP2 选择 0/帧同步。 通用 I/O。
53	53	41	SSP2_MOSI GPIO5	IO IO	SSP2 主数据输出/从数据输入。 通用 I/O。
54	54	42	VCC3IO_1	P	数字 3.3V 电源。
55	55	43	SSP2_MISO GPIO6	IO IO	SSP2 主数据输入/从数据输出。 通用 I/O。
56	56	44	SSP2_SCLK GPIO15	IO IO	SSP2 主时钟输出/从时钟输入。 通用 I/O。
57	57	45	ZCDPH2 GPIO9	I IO	相位 2 过零点侦测输入讯号。 通用 I/O。
58	58	46	BSTREN	I	必须连接到 GND。
59	59	47	VCCK	P	数字 1.1V 电源。

编号	V6201B	V6201R	名称	类型	功能说明
60	60	48	GPIO8	IO	通用 I/O。
61	61	49	ZCDPH1 GPIO10	I IO	相位 1 过零点侦测输入讯号。 通用 I/O。
62	62		VCC3IO_1	P	数字 3.3V 电源。
63	63		VCCK	P	数字 1.1V 电源。
64	64	50	OM	I	必须连接到 GND。
65	65		PHY_GPIO GPIO11	IO IO	PHY 通用 I/O。 通用 I/O。
66	66	51	ZCDPH0 GPIO12	I IO	相位 0 过零点侦测输入讯号。 通用 I/O。
67	67		VCCK	P	数字 1.1V 电源。
68	68		VCC3IO_1	P	数字 3.3V 电源。

2.4. 控制寄存器

2.4.1. IO 驱动和转换速率控制寄存器

2.4.1.1. IODRVCTRL0 寄存器

- 地址：0x54000810

  总线：Cortex-M4 APB

  名称：IODRVCTRL0

  描述：IO 驱动和转换速率控制。

表 2-2. IODRVCTRL0 寄存器

位	名称	类型	默认值	描述
31:28	ssp2_dcsr	R/W	0x0	[3]：SSP2 IO引脚转换速率 0：快速 1：慢速 [2:1]：SSP2 IO引脚驱动电流 00：4 mA

				01：8 mA 10：12 mA 11：16 mA [0]：保留
27:24	ssp1_dcsr	R/W	0x0	[3]：SSP1 IO引脚转换速率 0：快速 1：慢速 [2:1]：SSP1 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
23:20	ssp0_dcsr	R/W	0x0	[3]：SSP0 IO引脚转换速率 0：快速 1：慢速 [2:1]：SSP0 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
19:16	plc_dcsr	R/W	0x0	[3]：PLC IO引脚转换速率 0：快速 1：慢速 [2:1]：PLC IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
15:12	jtag_dcsr	R/W	0x0	[3]：JTAG IO引脚转换速率 0：快速 1：慢速 [2:1]：JTAG IO引脚驱动电流 00：4 mA

				01：8 mA 10：12 mA 11：16 mA [0]：保留
11:8	i2c_dcsr	R/W	0x0	[3]：I2C IO引脚转换速率 0：快速 1：慢速 [2:1]：I2C IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
7:4	gpio_dcsr	R/W	0x0	[3]：GPIO IO引脚转换速率 0：快速 1：慢速 [2:1]：GPIO IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
3:0	gmac_dcsr	R/W	0x0	[3]：GMAC IO引脚转换速率 0：快速 1：慢速 [2]：GMAC IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留

2.4.1.2. IODRVCTRL1 寄存器

- 地址：0x54000814

总线：Cortex-M4 APB

名称：IODRVCTRL1

描述：IO 驱动和转换速率控制。

表 2-3. IODRVCTRL1 寄存器

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:4	spi_prober_dcsr	R/W	0x0	[3]：SPI_PROBER IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_PROBER IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
3:0	ate_spi_flash_dcsr	R/W	0x0	[3]：ATE_SPI_FLASH IO引脚转换速率 0：快速 1：慢速 [2]：ATE_SPI_FLASH IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留

2.4.1.3. IODRVCTRL2 寄存器

- 地址：0x54000818

总线：Cortex-M4 APB

名称：IODRVCTRL2

描述：IO 驱动和转换速率控制。

表 2-4. IODRVCTRL2 寄存器

位	名称	类型	默认值	描述
31:28	sdr_dcsr	R/W	0x0	[3]：SDR IO引脚转换速率 0：快速 1：慢速

				[2:1]：SDR IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
27:24	uart6_dcsr	R/W	0x0	[3]：SPI_UART6 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART6 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
23:20	uart5_dcsr	R/W	0x0	[3]：SPI_UART5 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART5 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
19:16	uart4_dcsr	R/W	0x0	[3]：SPI_UART4 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART4 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
15:12	uart3_dcsr	R/W	0x0	[3]：SPI_UART3 IO引脚转换速率 0：快速 1：慢速

				[2:1]：SPI_UART3 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
11:8	uart2_dcsr	R/W	0x0	[3]：SPI_UART2 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART2 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
7:4	uart1_dcsr	R/W	0x0	[3]：SPI_UART1 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART1 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留
3:0	uart0_dcsr	R/W	0x0	[3]：SPI_UART0 IO引脚转换速率 0：快速 1：慢速 [2:1]：SPI_UART0 IO引脚驱动电流 00：4 mA 01：8 mA 10：12 mA 11：16 mA [0]：保留

2.4.2. 引脚功能多路复用控制寄存器

2.4.2.1. pmur_mfs0 寄存器

- 地址：0x54000820

  总线：Cortex-M4 APB

  名称：pmur_mfs0

  描述：引脚功能复用控制。

表 2-5. pmur_mfs0 寄存器（地址：0x54000820）

位	名称	类型	默认值	描述
31:30	pmur_mfs0	R/W	0x0	RMII_RXD1引脚复用选择。 00：RMII_RXD1 01：保留 10：保留 11：保留
29:28	pmur_mfs0	R/W	0x0	RMII_RXD0引脚复用选择。 00：RMII_RXD0 01：保留 10：保留 11：保留
27:26	pmur_mfs0	R/W	0x0	RMII_TX_EN引脚复用选择。 00：RMII_TX_EN 01：保留 10：保留 11：保留
25:24	pmur_mfs0	R/W	0x0	RMII_TXD1引脚复用选择。 00：RMII_TXD1 01：保留 10：保留 11：保留
23:22	pmur_mfs0	R/W	0x0	RMII_TXD0引脚复用选择。 00：RMII_TXD0 01：保留 10：保留 11：保留

21:20	pmur_mfs0	R/W	0x0	ATE_MISO引脚复用选择。 00：ATE_MISO 01：ssp1_miso 10：vango_probe_port[2] 11：GPIO29
19:18	pmur_mfs0	R/W	0x0	ATE_MOSI引脚复用选择。 00：ATE_MOSI 01：ssp1_mosi 10：vango_probe_port[1] 11：GPIO30
17:16	pmur_mfs0	R/W	0x0	ATE_CSN引脚复用选择。 00：ATE_CSN 01：sp1_fs 10：vango_probe_port[3] 11：GPIO16
15:14	pmur_mfs0	R/W	0x0	ATE_SCLK引脚复用选择。 00：ATE_SCLK 01：ssp1_sclk 10：vango_probe_port[0] 11：GPIO0_17
13:12	pmur_mfs0	R/W	0x0	DBGACK引脚复用选择。 00：GPIO0_14 01：DBGACK (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
11:10	pmur_mfs0	R/W	0x0	DBGRQ引脚复用选择。 00：GPIO0_13 01：DBGRQ (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
9:8	pmur_mfs0	R/W	0x0	TRSTn引脚复用选择。 00：GPIO12 01：TRSTn (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
7:6	pmur_mfs0	R/W	0x0	TDO引脚复用选择。 00：GPIO11

				01：TDO (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
5:4	pmur_mfs0	R/W	0x0	TDI引脚复用选择。 00：GPIO10 01：TDI (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
3:2	pmur_mfs0	R/W	0x0	TMS引脚复用选择。 00：GPIO9 01：TMS (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留
1:0	pmur_mfs0	R/W	0x0	TCK引脚复用选择。 00：GPIO8 01：TCK (CORTEXM4/CORTEXM0/DSP) 10：保留 11：保留

2.4.2.2. pmur_mfs1 寄存器

- 地址：0x54000824

  总线：Cortex-M4 APB

  名称：pmur_mfs1

  描述：引脚功能复用控制。

表 2-6. pmur_mfs1 寄存器（地址：0x54000824）

位	名称	类型	默认值	描述
31:30	pmur_mfs1	R/W	0x0	UART3_SIN引脚复用选择。 00：UART3_SIN 01：保留 10：保留 11：保留
29:28	pmur_mfs1	R/W	0x0	UART3_SOUT引脚复用选择。 00：UART3_SOUT 01：保留

				10：保留 11：保留
27:26	pmur_mfs1	R/W	0x0	UART2_SIN引脚复用选择。 00：GPIO19 01：UART2_SIN 10：UART6_SIN 11：SPI_CS1
25:24	pmur_mfs1	R/W	0x0	UART2_SOUT引脚复用选择。 00：GPIO18 01：UART2_SOUT 10：UART6_SOUT 11：SPI_CS0
23:22	pmur_mfs1	R/W	0x0	SSP2_FSCS0引脚复用选择。 00：SSP2_FSCS0 01：vango_probe_port[4] 10：GPIO0 11：保留
21:20	pmur_mfs1	R/W	0x0	SSP2_CS1引脚复用选择。 00：SSP2_CS1 01：LD_Clk 10：vango_probe_port[5] 11：GPIO1
19:18	pmur_mfs1	R/W	0x0	SSP2_MOSI引脚复用选择。 00：SSP2_MOSI 01：vango_probe_port[6] 10：GPIO5 11：保留
17:16	pmur_mfs1	R/W	0x0	SSP2_MISO引脚复用选择。 00：SSP2_MISO 01：vango_probe_port[7] 10：GPIO6 11：保留
15:14	pmur_mfs1	R/W	0x0	SSP2_SCLK引脚复用选择。 00：SSP2_SCLK 01：vango_probe_port[8] 10：GPIO15 11：保留

13:12	pmur_mfs1	R/W	0x0	I2C_SDA引脚复用选择。 00：I2C_SDA 01：保留 10：保留 11：保留
11:10	pmur_mfs1	R/W	0x0	I2C_SCL引脚复用选择。 00：I2C_SCL 01：保留 10：保留 11：保留
9:8	pmur_mfs1	R/W	0x0	RMII_MDIO引脚复用选择。 00：RMII_MDIO 01：保留 10：保留 11：保留
7:6	pmur_mfs1	R/W	0x0	RMII_MDC引脚复用选择。 00：RMII_MDC 01：保留 10：保留 11：保留
5:4	pmur_mfs1	R/W	0x0	RMII_REF_CLK引脚复用选择。 00：RMII_REF_CLK 01：保留 10：保留 11：保留
3:2	pmur_mfs1	R/W	0x0	RMII_RX_CRS_DV引脚复用选择。 00：RMII_RX_CRS_DV 01：保留 10：保留 11：保留
1:0	pmur_mfs1	R/W	0x0	RMII_RX_ER引脚复用选择。 00：RMII_RX_ER 01：保留 10：保留 11：保留

2.4.2.3. pmur_mfs2 寄存器

- 地址：0x54000864

  总线：Cortex-M4 APB

  名称：pmur_mfs2

  描述：引脚功能复用控制。

表 2-7. pmur_mfs2 寄存器（地址：0x54000864）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29:28	pmur_mfs2	R/W	0x0	GPIO4引脚复用选择。 00：GPIO4 01：SPI_MISO 10：vango_probe_port[10] 11：保留
27:26	pmur_mfs2	R/W	0x0	GPIO3引脚复用选择。 00：GPIO3 01：SPI_MOSI 10：vango_probe_port[12] 11：保留
25:24	pmur_mfs2	R/W	0x0	GPIO2引脚复用选择。 00：GPIO2 01：SPI_SCLK 10：vango_probe_port[9] 11：保留
23:20	-	-	-	保留。
19:18	pmur_mfs2	R/W	0x0	LD_URGENT_IN引脚复用选择。 00：LD_URGENT_IN 01：GPIO31 10：保留 11：保留
17:16	pmur_mfs2	R/W	0x0	LD_SDA引脚复用选择。 00：LD_SDA 01：保留 10：保留 11：保留

15:14	pmur_mfs2	R/W	0x0	LD_SCL引脚复用选择。 00：LD_SCL 01：保留 10：保留 11：保留
13:12	pmur_mfs2	R/W	0x0	DYING_GASP_IN引脚复用选择。 00：DYING_GASP_IN 01：GPIO28 10：保留 11：保留
11:10	pmur_mfs2	R/W	0x0	UART6_SIN引脚复用选择。 00：GPIO27 01：UART6_SIN 10：保留 11：保留
9:8	pmur_mfs2	R/W	0x0	UART6_SOUT引脚复用选择。 00：GPIO26 01：UART6_SOUT 10：保留 11：保留
7:6	pmur_mfs2	R/W	0x0	UART5_SIN引脚复用选择。 00：GPIO25 01：UART5_SIN 10：保留 11：保留
5:4	pmur_mfs2	R/W	0x0	UART5_SOUT引脚复用选择。 00：GPIO24 01：UART5_SOUT 10：保留 11：保留
3:2	pmur_mfs2	R/W	0x0	UART4_SIN引脚复用选择。 00：GPIO23 01：UART4_SIN 10：保留 11：保留
1:0	pmur_mfs2	R/W	0x0	UART4_SOUT引脚复用选择。 00：GPIO22

				01：UART4_SOUT 10：保留 11：保留

2.4.2.4. pmur_mfs3 寄存器

- 地址：0x54000880

  总线：Cortex-M4 APB

  名称：pmur_mfs3

  描述：引脚功能复用控制。

表 2-8. pmur_mfs3 寄存器（地址：0x54000880）

位	名称	类型	默认值	描述
31:12	-	-	-	保留。
11:10	pmur_mfs3	R/W	0x0	UART1_CTSn引脚复用选择。 00：GPIO21 01：UART1_CTSn 10：vango_probe_port[15] 11：保留
9:8	pmur_mfs3	R/W	0x0	UART1_RTSn引脚复用选择。 00：GPIO20 01：UART1_RTSn 10：vango_probe_port[11] 11：保留
7:6	pmur_mfs3	R/W	0x0	UART1_SOUT引脚复用选择。 00：UART1_SOUT 01：保留 10：保留 11：保留
5:4	pmur_mfs3	R/W	0x0	UART1_SIN引脚复用选择。 00：UART1_SIN 01：保留 10：保留 11：保留
3:2	pmur_mfs3	R/W	0x0	GPIO7引脚复用选择。 00：GPIO7

				01：保留 10：vango_probe_port[13] 11：保留
1:0	-	-	-	保留。

2.4.3. IO 开漏控制寄存器

2.4.3.1. ODMODE 寄存器

- 地址：0x5400087C

  总线：Cortex-M4 APB

  名称：ODMODE

  描述：开漏模式使能寄存器。

表 2-9. ODMODE 寄存器（地址：0x5400087C）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29	GPIO21_oden	R/W	0x0	GPIO21 设置为开漏模式。
28	UART1_RTSn_oden	R/W	0x0	UART1_RTSn/GPIO20 设置为开漏模式。
27	UART1_SOUT_oden	R/W	0x1	UART1_SOUT 设置为开漏模式。
26	UART1_SIN_oden	R/W	0x0	UART1_SIN 设置为开漏模式。
25	GPIO7_oden	R/W	0x0	GPIO7 设置为开漏模式。
24	GPIO6_oden	R/W	0x0	GPIO6 设置为开漏模式。
23	GPIO5_oden	R/W	0x0	GPIO5 设置为开漏模式。
22	GPIO4_oden	R/W	0x0	GPIO4 设置为开漏模式。
21	GPIO3_oden	R/W	0x0	GPIO3 设置为开漏模式。
20	GPIO2_oden	R/W	0x0	GPIO2 设置为开漏模式。
19	GPIO1_oden	R/W	0x0	GPIO1 设置为开漏模式。
18	GPIO0_oden	R/W	0x0	GPIO0 设置为开漏模式。
17	-	-	-	保留。
16	LD_SDA_oden	R/W	0x0	LD_SDA 设置为开漏模式。

15	LD_SCL_oden	R/W	0x0	LD_SCL 设置为开漏模式。
14:4	-	-	-	保留。
3	I2C_SDA_oden	R/W	0x1	I2C_SDA 设置为开漏模式。
2	I2C_SCL_oden	R/W	0x1	I2C_SCL 设置为开漏模式。
1:0	-	-	-	保留。

3. 电源供应

3.1. 上电时序（Power-on Sequence）

有关硬件重启机制，更多细节信息请参照

图 **3-1** 及表 3-1。当电源已开启并准备就绪时，上电复位（POR）在完成复位后，将进入等待晶体振荡器的稳定时间(T_OSC_RDY)周期。在晶体振荡器稳定之后，在 eFUSE 控制器完成 LDO 调整与系统控制器(SCU)在 T_RST_DEB 期间完成去抖动电路以生成 T_RSTn 之前，RSTn 被置为无效。同时，SCU 将持续等待占空比稳定时间周期直到拥有稳定的系统时钟源。之后，当 SCU 显示 scu_pllen 高位状态，即确认了 PLL 已准备就绪（T_PLL_LOCK）并开始系统启动时序。

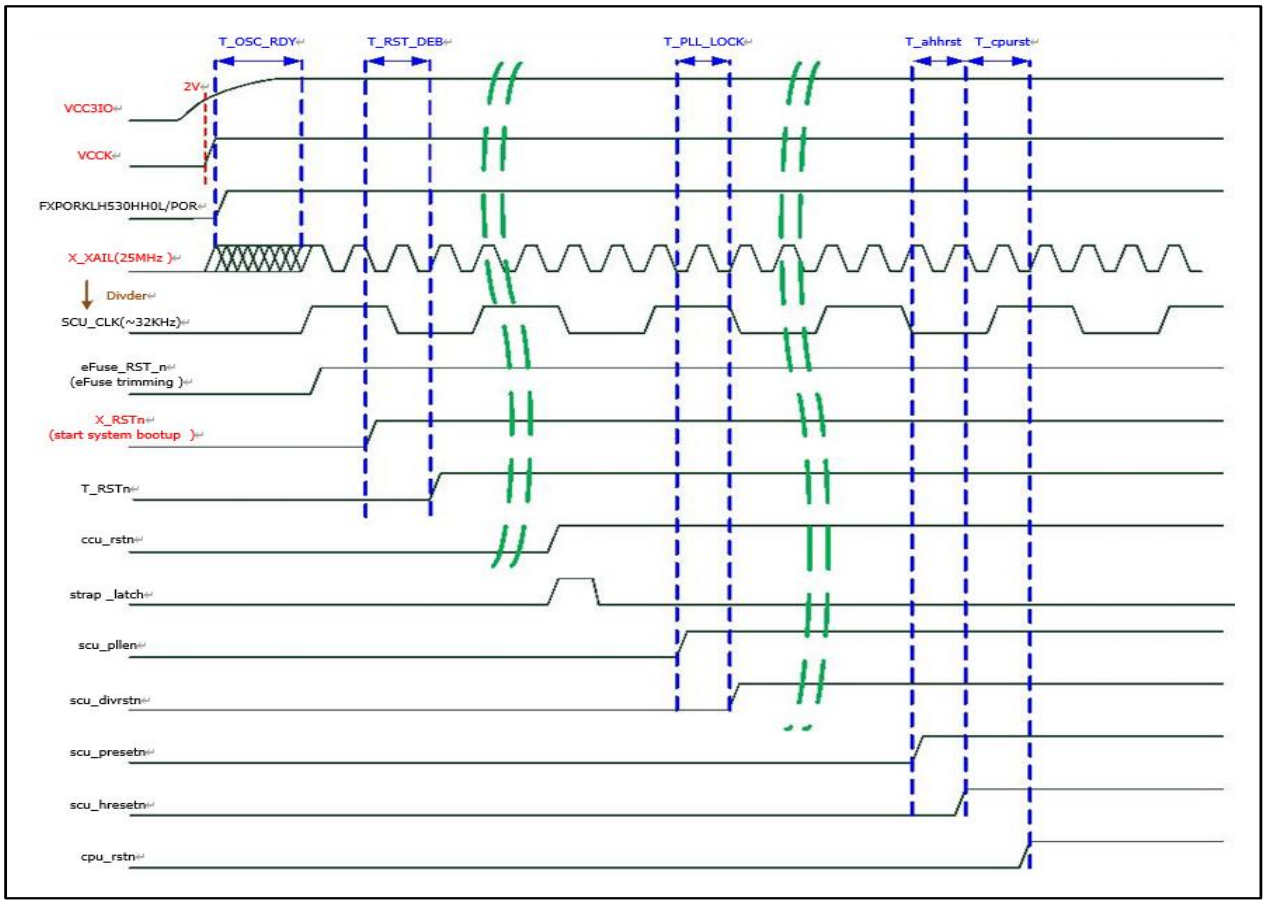


图 3-1. V6201X 上电流程图

表 3-1. 上电时序说明

参数	内容	数值	单位
T_OSC_RDY	上电后晶振时钟稳定时间	10	ms
T_RST_DEB	硬件复位信号去抖动时间	2	ms
T_PLL_LOCK	PLL 锁定时间	100	us
T_abhrst	APB 至 AHB 总线复位释放时间	1	PCLK 周期

T_cpurst	AHB 总线到 CPU 的复位释放时间	4	PCLK 周期

4. 时钟与复位

4.1. 系统时钟

4.1.1. 概述

晶体和 PLL 模块集成在 SoC 上层。SoC 上层从晶振提供 PLL 稳定的 25 MHz 时钟源，并从 PLL 提供 300 MHz 时钟源给系统总线上模块与分频器。其中，AHB 总线时钟和 APB 总线时钟将由 PLL 输出并通过使用时钟分频器来保持时钟之间的正确比率输出。在图 4-1 中，标记为"GCK"的区块是位于系统控制单元内的时钟门控寄存器。PLL 将为处理器，物理层单元，AHB 和 APB 总线生成时钟。如图 4-1 所示的时钟分配图描述了时钟分配网络。

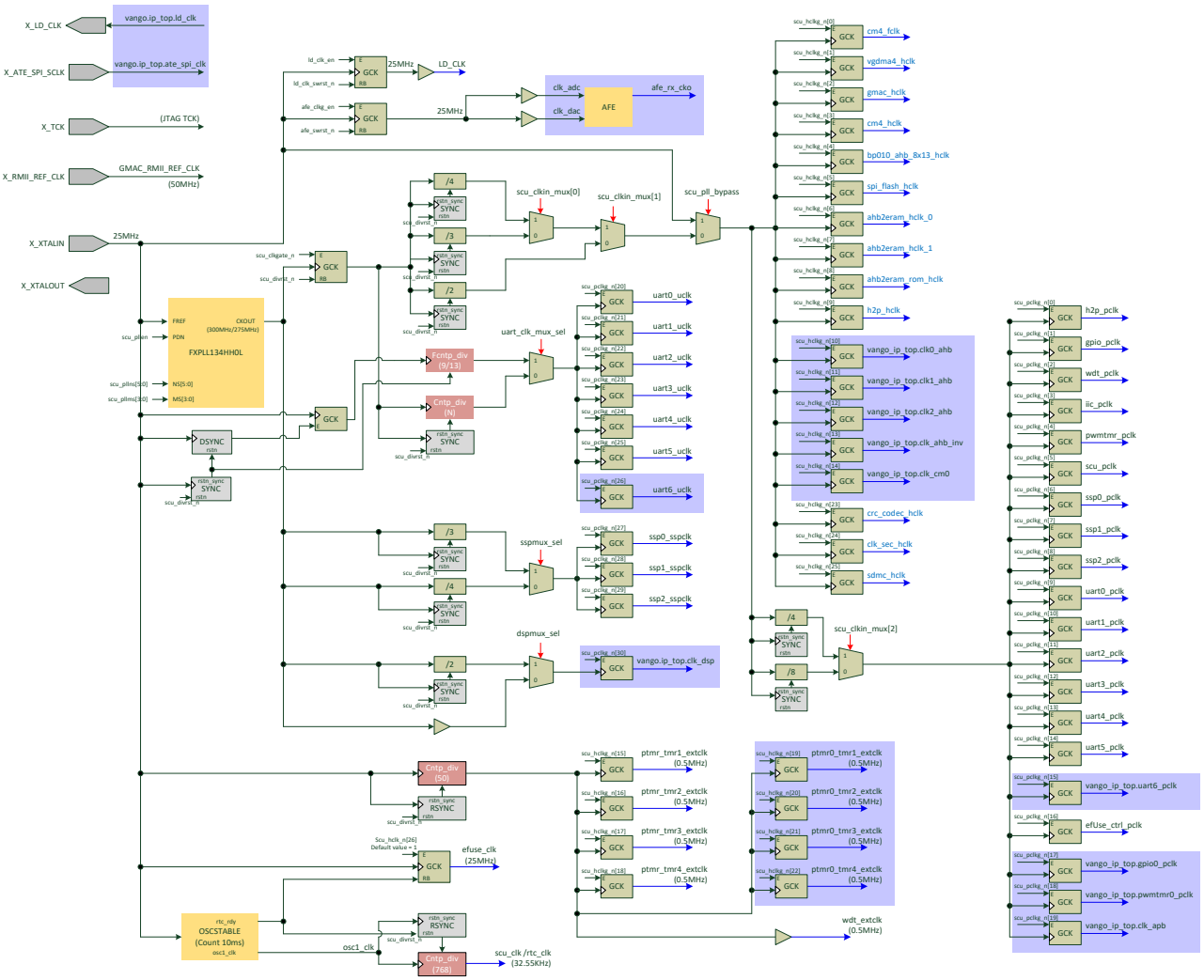


图 4-1. V6201X 时钟架构

4.1.2. 外部时钟源

芯片连接了一个外部时钟源。时钟源如表 4-1 所示。

表 4-1. 外部时钟源

时钟名称	频率	描述
XTALI/XTALO	25 MHz	用于驱动 PLL，CPU 和总线时钟源可以来自该 PLL。

4.1.3. PLL 编程值

表 4-2 定义了该器件中使用的 PLL 以及默认输出频率和相应的默认编程值。

表 4-2. PLL 默认频率及编程值

PLL 实例	NS[6:0]	MS[4:0]	CKOUT（MHz）	描述
FXPLL134HH0L	22/24	2	275/300	该PLL的输出固定为300 MHz或275 MHz，并驱动Cortex-M4内核处理器、Cortex-M0内核处理器、各自对应的总线结构（AHB和APB）和通信物理层模块。基于该PLL输出时钟的AMBA总线时钟如下： CPU FCLK = 150/137.5 MHz AHB总线Clk (hclk) = 150/137.5 MHz APB总线Clk (pclk) = 37.5/33.75 MHz 通信物理层模块 = 275/300 MHz 注：NS和MS的配置值可以由SCU控制。

4.1.4. 外设时钟参数寄存器（PDIV_VALUE）

4.1.4.1. PDIV_VALUE 寄存器

- 地址：0x54000838

  总线：Cortex-M4 APB

  名称：PDIV_VALUE

  描述：外设时钟分频的时钟参数控制。

表 4-3. PDIV_VALUE 寄存器（地址：0x54000838）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29:20	pvalue_scu_clk_div	R/W	0x300	SCU时钟分频系数。

				1：SCU clock = XTAL/2 2：SCU clock = XTAL/3 ….
19:10	pvalue_ptmr_clk_div	R/W	0x32	PWM定时器时钟分频系数。 1：PWM timer clock = XTAL/2 2：PWM timer clock = XTAL/3 ….
9:0	pvalue_uart_clk_div	R/W	0xF	UART时钟分频系数。 1：UART clock = PLL clock output/2 2：UART clock = PLL clock output/3 ….

4.1.5. 时钟控制寄存器（AHBCLKG 和 APBCLKG）

时钟控制寄存器用于控制 AHB 和 APB 总线时钟，scu_hclkg_n。

4.1.5.1. AHBCLKG 寄存器

- 地址：0x54000050

  总线：Cortex-M4 APB

  名称：AHBCLKG

  描述：AHB 时钟控制寄存器。

  （置 1：使能，置 0：关闭）

表 4-4. AHBCLKG 寄存器（地址：0x54000050）

位	名称	类型	默认值	描述
31:26	-	-	-	保留
25	HCLK_EN25	R/W	0x0	SDRAM 控制器时钟使能寄存器
24	HCLK_EN24	R/W	0x0	安全硬件加速器时钟使能寄存器
23	HCLK_EN23	R/W	0x1	Cortex-M4 内核 CRC 时钟使能寄存器
22	HCLK_EN22	R/W	0x1	Cortex-M0 内核 Timer1 extclk4 时钟使能寄存器
21	HCLK_EN21	R/W	0x1	Cortex-M0 内核 Timer1 extclk3 时钟使能寄存器
20	HCLK_EN20	R/W	0x1	Cortex-M0 内核 Timer1 extclk2 时钟使能寄存器
19	HCLK_EN19	R/W	0x1	Cortex-M0 内核 Timer1 extclk1 时钟使能寄存器

18	HCLK_EN18	R/W	0x1	Cortex-M4 内核 Timer0 extclk1 时钟使能寄存器
17	HCLK_EN17	R/W	0x1	Cortex-M4 内核 Timer0 extclk1 时钟使能寄存器
16	HCLK_EN16	R/W	0x1	Cortex-M4 内核 Timer0 extclk1 时钟使能寄存器
15	HCLK_EN15	R/W	0x1	Cortex-M4 内核 Timer0 extclk1 时钟使能寄存器
14	HCLK_EN14	R/W	0x1	Cortex-M0 内核 FCLK 时钟使能寄存器
13	HCLK_EN13	R/W	0x1	Cortex-M0 反向 HCLK 时钟使能寄存器
12	HCLK_EN12	R/W	0x1	保留
11	HCLK_EN11	R/W	0x1	Cortex-M0 内核总线 HCLK 时钟使能寄存器
10	HCLK_EN10	R/W	0x1	Cortex-M0 HCLK 时钟使能寄存器
9	HCLK_EN9	R/W	0x1	H2P HCLK 时钟使能寄存器
8	HCLK_EN8	R/W	0x1	AHB2ERAM (ROM) HCLK 时钟使能寄存器
7	HCLK_EN7	R/W	0x1	AHB2ERAM (DRAM1) HCLK 时钟使能寄存器
6	HCLK_EN6	R/W	0x1	AHB2ERAM (DRAM0) HCLK 时钟使能寄存器
5	HCLK_EN5	R/W	0x1	Flash 控制器 HCLK 时钟使能寄存器
4	HCLK_EN4	R/W	0x1	Cortex-M4 总线矩阵 HCLK 时钟使能寄存器
3	HCLK_EN3	R/W	0x1	Cortex-M4 内核 HCLK 时钟使能寄存器
2	HCLK_EN2	R/W	0x0	GMAC 时钟使能寄存器
1	HCLK_EN1	R/W	0x1	Cortex-M4 内核 DMA4 时钟使能寄存器
0	HCLK_EN0	R/W	0x1	Cortex-M4 内核 FCLK 时钟使能寄存器

4.1.5.2. APBCLKG 寄存器

- 地址：0x54000060

  总线：Cortex-M4 APB

  名称：APBCLKG

  描述：APB 时钟控制寄存器。

表 4-5. APBCLKG 寄存器（地址：0x54000060）

位	名称	类型	默认值	描述
31	PCLK_EN31	R/W	0x0	UART bypass 时钟使能寄存器
30	HCLK_EN30	R/W	0x1	通信物理层时钟使能寄存器
29	HCLK_EN29	R/W	0x1	SSP2 SSPCLK 时钟使能寄存器

28	HCLK_EN28	R/W	0x1	SSP1 SSPCLK 时钟使能寄存器
27	HCLK_EN27	R/W	0x1	SSP0 SSPCLK 时钟使能寄存器
26	-	-	-	保留
25	HCLK_EN25	R/W	0x1	UART5 UCLK 时钟使能寄存器
24	HCLK_EN24	R/W	0x1	UART4 UCLK 时钟使能寄存器
23	HCLK_EN23	R/W	0x1	UART3 UCLK 时钟使能寄存器
22	HCLK_EN22	R/W	0x1	UART2 UCLK 时钟使能寄存器
21	HCLK_EN21	R/W	0x1	UART1 UCLK 时钟使能寄存器
20	HCLK_EN20	R/W	0x1	UART0 UCLK 时钟使能寄存器
19	HCLK_EN19	R/W	0x1	Cortex-M0 内核 APB PCLK 时钟使能寄存器
18	HCLK_EN18	R/W	0x1	Cortex-M0 内核 Timer1 PCLK 时钟使能寄存器
17	HCLK_EN17	R/W	0x1	Cortex-M0 内核 GPIO1 PCLK 时钟使能寄存器
16	HCLK_EN16	R/W	0x1	保留
15	HCLK_EN15	R/W	0x1	保留
14	HCLK_EN14	R/W	0x1	UART5 PCLK 时钟使能寄存器
13	HCLK_EN13	R/W	0x1	UART4 PCLK 时钟使能寄存器
12	HCLK_EN12	R/W	0x1	UART3 PCLK 时钟使能寄存器
11	HCLK_EN11	R/W	0x1	UART2 PCLK 时钟使能寄存器
10	HCLK_EN10	R/W	0x1	UART1 PCLK 时钟使能寄存器
9	HCLK_EN9	R/W	0x1	UART0 PCLK 时钟使能寄存器
8	HCLK_EN8	R/W	0x1	SSP2 PCLK 时钟使能寄存器
7	HCLK_EN7	R/W	0x1	SSP1 PCLK 时钟使能寄存器
6	HCLK_EN6	R/W	0x1	SSP0 PCLK 时钟使能寄存器
5	HCLK_EN5	R/W	0x1	SCU PCLK 时钟使能寄存器
4	HCLK_EN4	R/W	0x1	Timer0 PCLK 时钟使能寄存器
3	HCLK_EN3	R/W	0x1	IIC0 PCLK 时钟使能寄存器
2	HCLK_EN2	R/W	0x1	WDT PCLK 时钟使能寄存器
1	HCLK_EN1	R/W	0x1	GPIO0 PCLK 时钟使能寄存器
0	HCLK_EN0	R/W	0x1	Cortex-M4 H2P PCLK 时钟使能寄存器

4.2. 系统复位

V6201X 支持一个外部复位事件。表 4-6 定义 V6201X 返回初始上电状态的复位事件，以及图 **4-2** 显示通用的复位系统。

表 4-6. 通用复位事件

复位事件	引脚/信号名称	事件未复位的功能	事件是否唤起初始化状态机	描述
Hardware Reset	RESETn	无，SCU 中的硬件复位事件状态除外。	是	此复位由外部引脚所提供。
Watchdog Reset	wdtrst_n	无，SCU 中的看门狗事件状态除外。	否	看门狗复位 IP 触发类似的硬件复位事件。
Software Reset	swrst_n	除复位目标以外的所有区块均处于未激活的非复位运行状态。	否	此复位允许软件复位单个 IP 区块。假设复位为异步且低电平有效，复位将同步到使用它的时钟域。

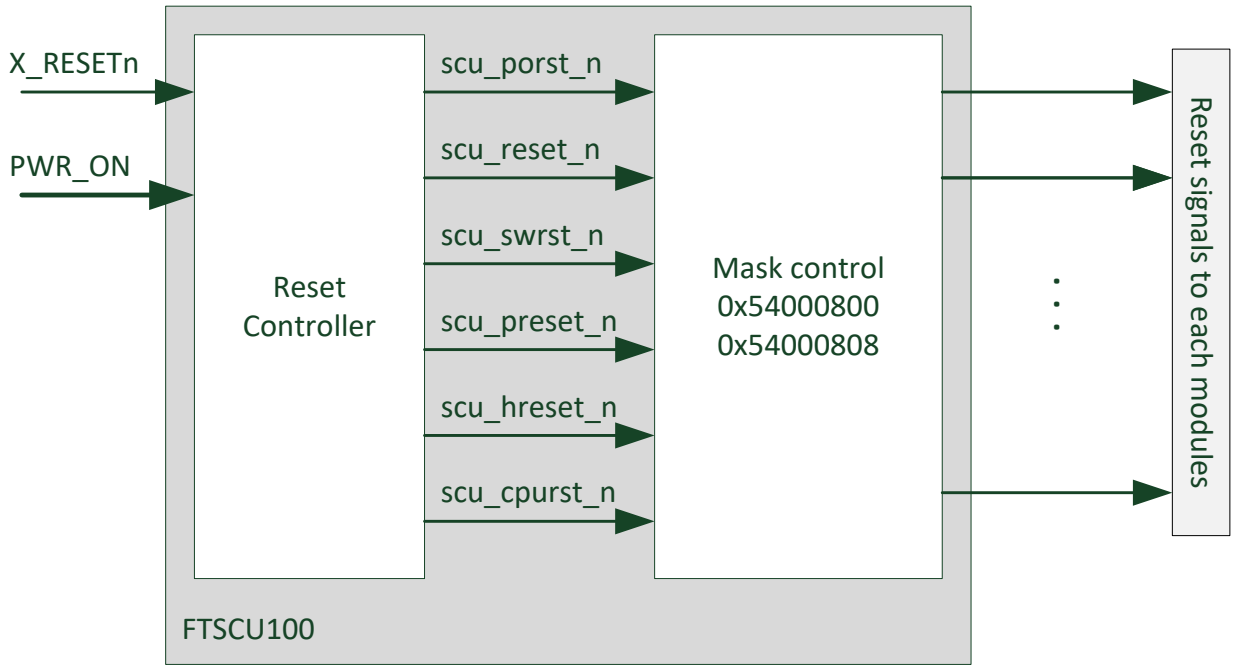


图 4-2. 通用复位系统

4.2.1. 软件复位

软件复位功能应作为系统控制单元（SCU）逻辑内的一部分。通过软件复位，CPU 可以复位特定的 IP 区块。软件复位在逻辑上与硬件复位进行“AND”运算，因此应为异步，低电平有效。硬件复位信号应具有与接收时钟域同步的复位后沿。

4.2.2. 系统复位控制寄存器

4.2.2.1. HWRST_MASK 寄存器

- 地址：0x54000800

  总线：Cortex-M4 APB

  名称：HWRST_MASK

  描述：硬件或软件重置阶段中的硬件复位屏蔽设置。

  （置 1：屏蔽，置 0：无屏蔽）

表 4-7. HWRST_MASK 寄存器（地址：0x54000800）

位	名称	类型	默认值	描述
31:30	-	-	-	保留
29	SDMC_HR	R/W	0x0	设置为屏蔽 SDMC 的复位信号。
28	security_top_HR	R/W	0x0	设置为屏蔽 AES 的复位信号。
27	crc_codec_HR	R/W	0x0	设置为屏蔽 CRC 的复位信号。
26	WDT_PR	R/W	0x0	设置为屏蔽 WDT 的复位信号。
25	SSP2_PR	R/W	0x0	设置为屏蔽 SSP2 的复位信号。
24	SSP1_PR	R/W	0x0	设置为屏蔽 SSP1 的复位信号。
23	SSP0_PR	R/W	0x0	设置为屏蔽 SSP0 的复位信号。
22	-	-	-	保留
21	UART5_PR	R/W	0x0	设置为屏蔽 UART5 的复位信号。
20	UART4_PR	R/W	0x0	设置为屏蔽 UART4 的复位信号。
19	UART3_PR	R/W	0x0	设置为屏蔽 UART3 的复位信号。
18	UART2_PR	R/W	0x0	设置为屏蔽 UART2 的复位信号。
17	UART1_PR	R/W	0x0	设置为屏蔽 UART1 的复位信号。
16	UART0_PR	R/W	0x0	设置为屏蔽 UART0 的复位信号。
15	Timer0_PR	R/W	0x0	设置为屏蔽 Timer0 的复位信号。
14	IIC0_PR	R/W	0x0	设置为屏蔽 IIC0 的复位信号。
13	GPIO0_PR	R/W	0x0	设置为屏蔽 GPIO0 的复位信号。
12	H2P_PR	R/W	0x0	设置为屏蔽 H2P bridge 的复位信号。
11	-	-	-	保留

10	vango_ip_top_HR	R/W	0x0	设置为屏蔽 MAC 的复位信号。
9	GMAC_HR	R/W	0x0	设置为屏蔽 GMAC 的复位信号。
8	H2P_HR	R/W	0x0	设置为屏蔽 H2P bridge 的复位信号。
7	vgdmac_m4_HR	R/W	0x0	设置为屏蔽 vgdmac_m4 的复位信号。
6	AHB_Matrix_HR	R/W	0x0	设置为屏蔽 AHB_Matrix 的复位信号。
5	FLASH_CTRL	R/W	0x0	设置为屏蔽 FLASH 的复位信号。
4	AHB2ERAM_ROM_HR	R/W	0x0	设置为屏蔽 CM4 ROM 的复位信号。
3	AHB2ERAM_1_HR	R/W	0x0	设置为屏蔽 CM4 DRAM4 bank0 的复位信号。
2	AHB2ERAM_HR	R/W	0x0	设置为屏蔽 CM4 DRAM4 bank1 的复位信号。
1	phy_top_BR	R/W	0x0	设置为屏蔽通信物理层的复位信号。
0	CORTEXM4_SR	R/W	0x0	设置为屏蔽 CORTEXM4 的复位信号。

4.2.2.2. HWRST_MASK 寄存器

- 地址：0x54000804

  总线：Cortex-M4 APB

  名称：HWRST_MASK (cont.)

  描述：硬件重置阶段中的复位屏蔽设置。

  （置 1：屏蔽，置 0：无屏蔽）

表 4-8. HWRST_MASK 寄存器（地址：0x54000804）

位	名称	类型	默认值	描述
31:3	-	-	-	保留。
2	MAC_HWR	R/W	0x0	设置为屏蔽 MAC 的硬件复位信号。
1	FLASH_CTRL	R/W	0x0	设置为屏蔽 FLASH 的硬件复位信号。
0	phy_top_HWR	R/W	0x0	设置为屏蔽通信物理层的硬件复位信号。

4.2.2.3. SWRST_MASK 寄存器

- 地址：0x54000808

  总线：Cortex-M4 APB

  名称：SWRST_MASK

  描述：软件复位屏蔽设定。

（置 1：屏蔽，置 0：无屏蔽）

表 4-9. SWRST_MASK 寄存器（地址：0x54000808）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29	SDMC_HR	R/W	0x0	设置为屏蔽 SDMC 的复位信号。
28	security_top_HR	R/W	0x0	设置为屏蔽 AES 的复位信号。
27	crc_codec_HR	R/W	0x0	设置为屏蔽 CRC 的复位信号。
26	WDT_PR	R/W	0x0	设置为屏蔽 WDT 的复位信号。
25	VGSPI_SSP2_PR	R/W	0x0	设置为屏蔽 SSP2 的复位信号。
24	VGSPI_SSP1_PR	R/W	0x0	设置为屏蔽 SSP1 的复位信号。
23	VGSPI_SSP0_PR	R/W	0x0	设置为屏蔽 SSP0 的复位信号。
22	-	-	-	保留。
21	UART5_PR	R/W	0x0	设置为屏蔽 UART5 的复位信号。
20	UART4_PR	R/W	0x0	设置为屏蔽 UART4 的复位信号。
19	UART3_PR	R/W	0x0	设置为屏蔽 UART3 的复位信号。
18	UART2_PR	R/W	0x0	设置为屏蔽 UART2 的复位信号。
17	UART1_PR	R/W	0x0	设置为屏蔽 UART1 的复位信号。
16	UART0_PR	R/W	0x0	设置为屏蔽 UART0 的复位信号。
15	Timer0_PR	R/W	0x0	设置为屏蔽 Timer0 的复位信号。
14	IIC0_PR	R/W	0x0	设置为屏蔽 IIC0 的复位信号。
13	GPIO_PR	R/W	0x0	设置为屏蔽 GPIO 的复位信号。
12	H2P_PR	R/W	0x0	设置为屏蔽 H2P 的复位信号。
11	-	-	-	保留。
10	vango_ip_top_HR	R/W	0x0	设置为屏蔽 MAC 的复位信号。
9	GMAC_HR	R/W	0x0	设置为屏蔽 GMAC 的复位信号。
8	H2P_HR	R/W	0x0	设置为屏蔽 H2P bridge 的复位信号。
7	vgdmac_m4_HR	R/W	0x0	设置为屏蔽 vgdmac_m4 的复位信号。
6	AHB_Matrix	R/W	0x0	设置为屏蔽 AHB_Matrix 的复位信号。
5	FLASH_CTRL	R/W	0x0	设置为屏蔽 FLASH 的复位信号。
4	AHB2ERAM_ROM_HR	R/W	0x0	设置为屏蔽 CM4 ROM 的复位信号。
3	AHB2ERAM_1_HR	R/W	0x0	设置为屏蔽 CM4 DRAM4 bank0 的复位

位	名称	类型	默认值	描述
				信号。
2	AHB2ERAM_HR	R/W	0x0	设置为屏蔽 CM4 DRAM4 bank1 的复位信号。
1	phy_top_BR	R/W	0x0	设置为屏蔽通信物理层的复位信号。
0	CORTEXM4_SR	R/W	0x0	设置为屏蔽 CORTEX-M4 的复位信号。

4.2.2.4. SWRST_CLR 寄存器

- 地址：0x54000878

  总线：Cortex-M4 APB

  名称：SWRST_CLR

  描述：清除软件复位状态。

表 4-10. SWRST_CLR 寄存器（地址：0x54000878）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29	sw_rst_status_clr	R/W	0x1	软复位状态清除寄存器。 该位置 1 清除 sw_rst_status （0x54000884）。
28	urgent_in_inv	R/W	0x1	设置为反向输入 LD_URGENT 事件信号。
27	dying_gasp_in_inv	R/W	0x1	设置为反向输入 DYING_GASP 事件信号。
26	phy_gpio_inv	R/W	0x1	设置为反向输入 PHY_GPIO 事件信号。
25:0	-	-	-	保留。

4.2.2.5. SWRST_STS 寄存器

- 地址：0x54000884

  总线：Cortex-M4 APB

  名称：SWRST_STS

  描述：软件复位状态。

表 4-11. SWRST_STS 寄存器（地址：0x54000884）

位	名称	类型	默认值	描述
31:1	-	-	-	保留。

位	名称	类型	默认值	描述
0	sw_rst_status	R	0x0	执行软件复位 （SCU，0x54000020）时软件复位状态检查。

5. 总线互连和内存映射

5.1. Cortex-M4 AHB / APB 总线上的内存空间地址映像

表 5-1. Cortex-M4 AHB/APB 总线上的寄存器空间地址映像

	SPIRIT 名称	描述	映像基地址	长度
AHB				
S0	AHB_Slave	SRAM 寄存器	0x0000_0000	64KB（remap=0x3）
S1	AHB_Slave	SRAM 寄存器	0x2000_0000	256KB
S2	AHB_Slave	SRAM 寄存器	0x3000_0000	256KB（remap=0x3）
S3	AHB_Slave	APB 总线寄存器	0x5400_0000	16MB
S4	AHB_Slave	GMAC 寄存器	0x5600_0000	16MB
S5	AHB_Slave	DMA 寄存器	0x5700_0000	16MB
S6	AHB_Slave	SPI Flash 寄存器	0x6000_0000	16MB
S7	AHB_Slave	AHB 寄存器	0x7000_0000	768MB
S8	AHB_Slave	AHB 寄存器	0xE010_0000	536MB
S9	AHB_Slave	CRC 寄存器	0x5800_0000	16MB
S10	AHB_Slave	AES 寄存器	0x5900_0000	16MB
S11	AHB_Slave	SDRAM REG 寄存器	0x5A00_0000	16MB
S12	AHB_Slave	SDRAM 寄存器	0x5B00_0000	8MB
APB				
S0	APB_Slave	系统控制寄存器	0x5400_0000	64KB
S1	APB_Slave	UART 寄存器	0x5401_0000	64KB
S2	APB_Slave	看门狗定时器	0x5402_0000	64KB
S3	APB_Slave	PWM 计时寄存器	0x5403_0000	64KB
S4	APB_Slave	I^2C 寄存器	0x5404_0000	64KB
S5	APB_Slave	GPIO 寄存器器	0x5405_0000	64KB
S6	APB_Slave	SSP0 寄存器	0x5406_0000	64KB
S7	APB_Slave	SSP1 寄存器	0x5407_0000	64KB

	SPIRIT 名称	描述	映像基地址	长度
S8	APB_Slave	UART 寄存器	0x5408_0000	64KB
S9	APB_Slave	UART 寄存器	0x5409_0000	64KB
S10	APB_Slave	UART 寄存器	0x540A_0000	64KB
S11	APB_Slave	UART 寄存器	0x540B_0000	64KB
S12	APB_Slave	UART 寄存器	0x540C_0000	64KB
S13	APB_Slave	SSP2 寄存器	0x540D_0000	64KB
S14	APB_Slave	eFUSE 寄存器	0x540E_0000	64KB

5.2. Cortex-M0 AHB 总线上的 AHB 主机模块

表 5-2. 在 Cortex-M0 AHB 总线上的 AHB 主机模块列表

Master 代号	Master 实例	数据长度
Master_00	u_cm0intimp	32
Master_01	u_h2hdec	32
Master_02	u_vgdmac_m0	32
Master_03	u_h2hdec	32

5.3. Cortex-M0 AHB/APB 总线上的内存空间地址映像

表 5-3. Cortex-M0 AHB / APB 总线上的存储器空间地址映像

	SPIRIT 名称	描述	映像基地址	长度
AHB				
S0	AHB_Slave	SRAM 寄存器	0x0000_0000	128KB
S1	AHB_Slave	SRAM 寄存器	0x2000_0000	32KB
S2	AHB_Slave	SRAM 寄存器	0x3000_0000	32KB
S3	AHB_Slave	APB 总线寄存器	0x5400_0000	1MB
S4	AHB_Slave	MAC CSR 寄存器	0x7000_0000	4KB
S5	AHB_Slave	DMA CSR 寄存器	0x7700_0000	16MB
S6	-	保留	-	-
S7	AHB_Slave	CRC CSR 寄存器	0x7000_F000	4KB

S8	-	保留	-	-
S9	AHB_Slave	PKT TX 寄存器输入	0x7000_1000	4KB
S10	AHB_Slave	PKT RX 寄存器输入	0x7000_2000	4KB
S11	AHB_Slave	SRAM 寄存器	0x8000_0000	512KB
APB				
S0	APB_Slave	UART 寄存器	0x5401_0000	64KB
S1	APB_Slave	PWM 计时寄存器	0x5403_0000	64KB
S2	APB_Slave	GPIO 寄存器	0x5406_0000	64KB

5.4. Cortex-M4 AHB/APB 总线内存映射图

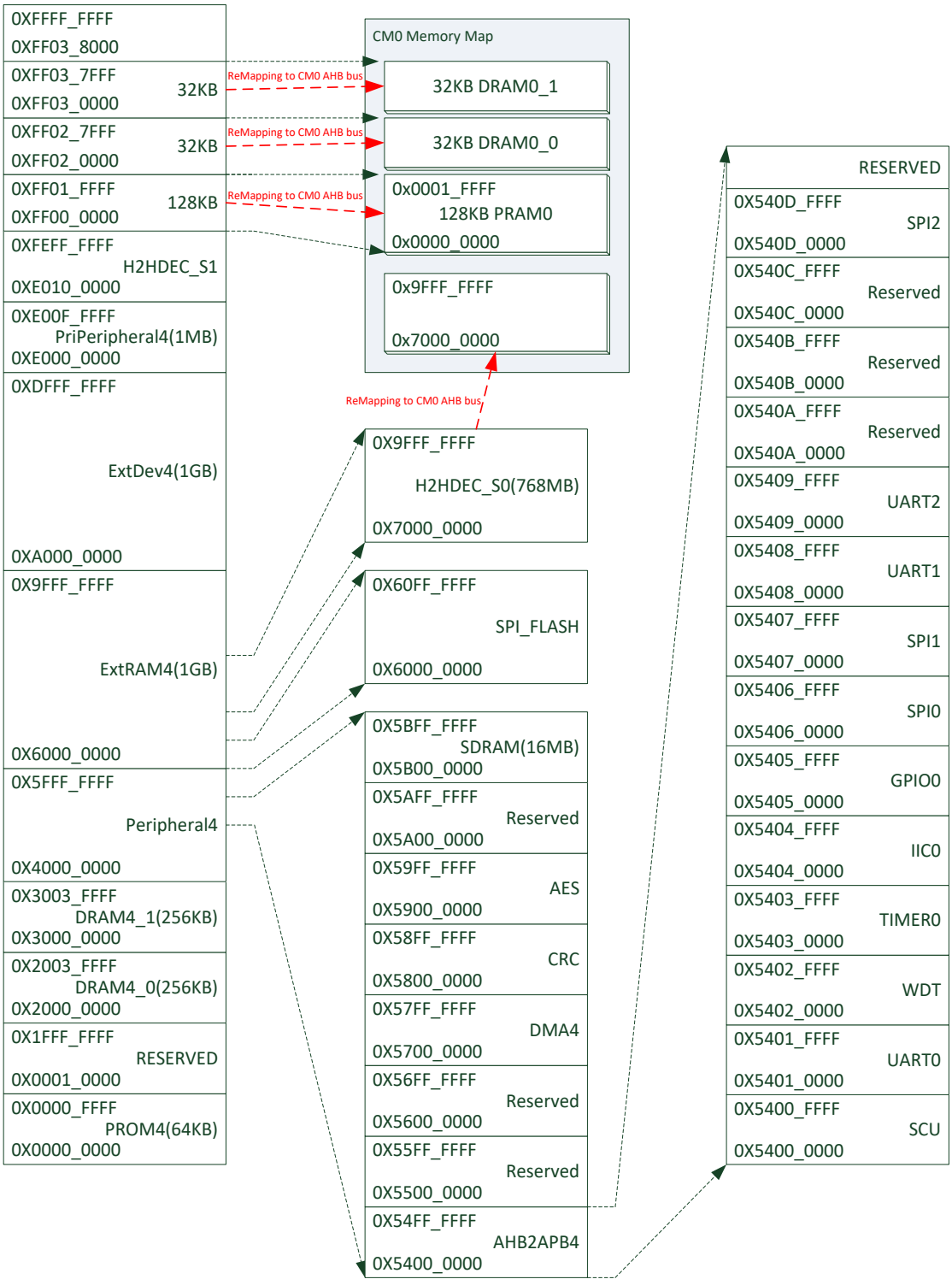


图 5-1. Cortex-M4 AHB 总线的内存映像配置图

5.5. Cortex-M0 AHB/APB 总线内存映射图

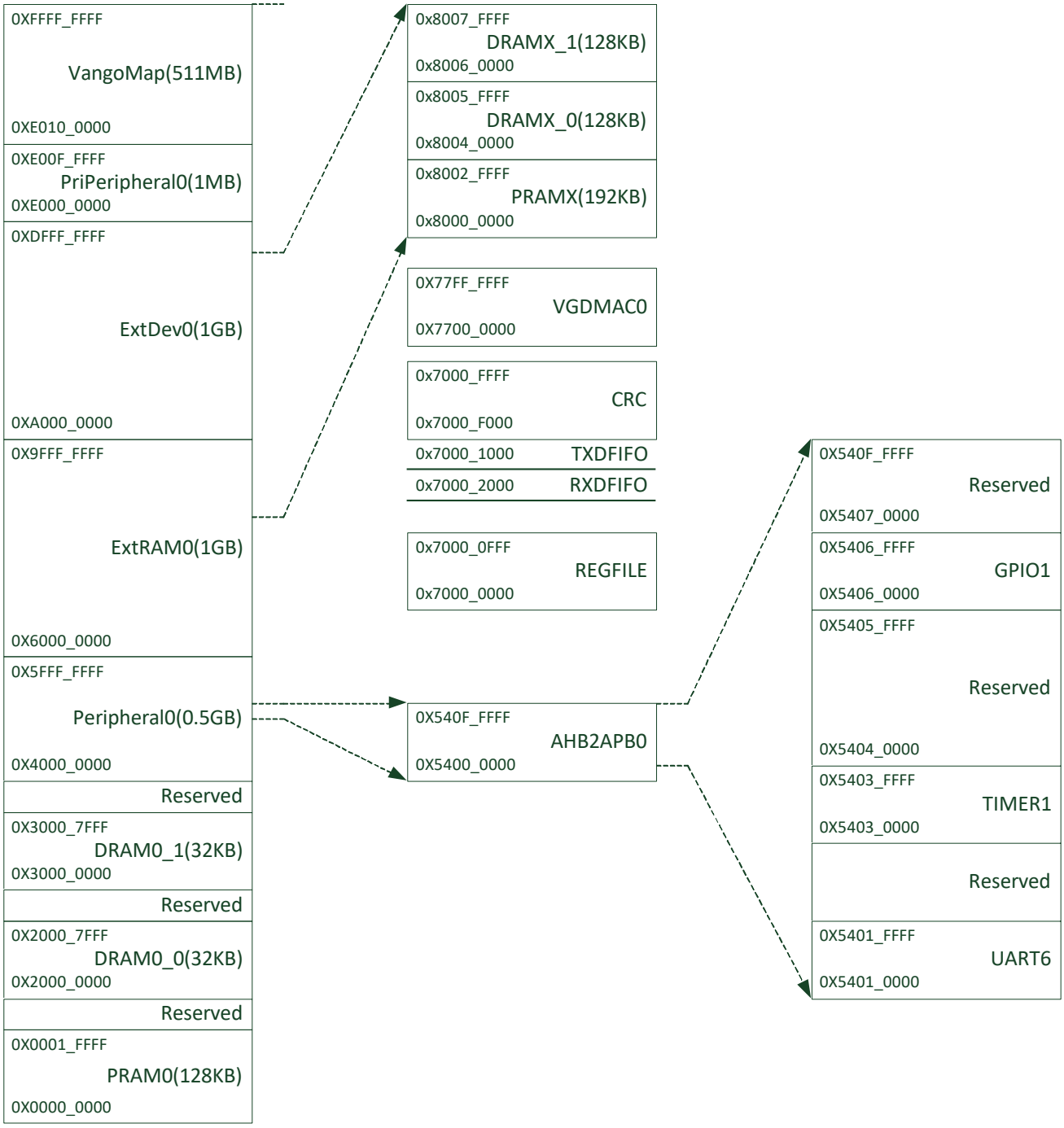


图 5-2. Cortex-M0 AHB 总线的内存映像配置图

6. 启动重映射

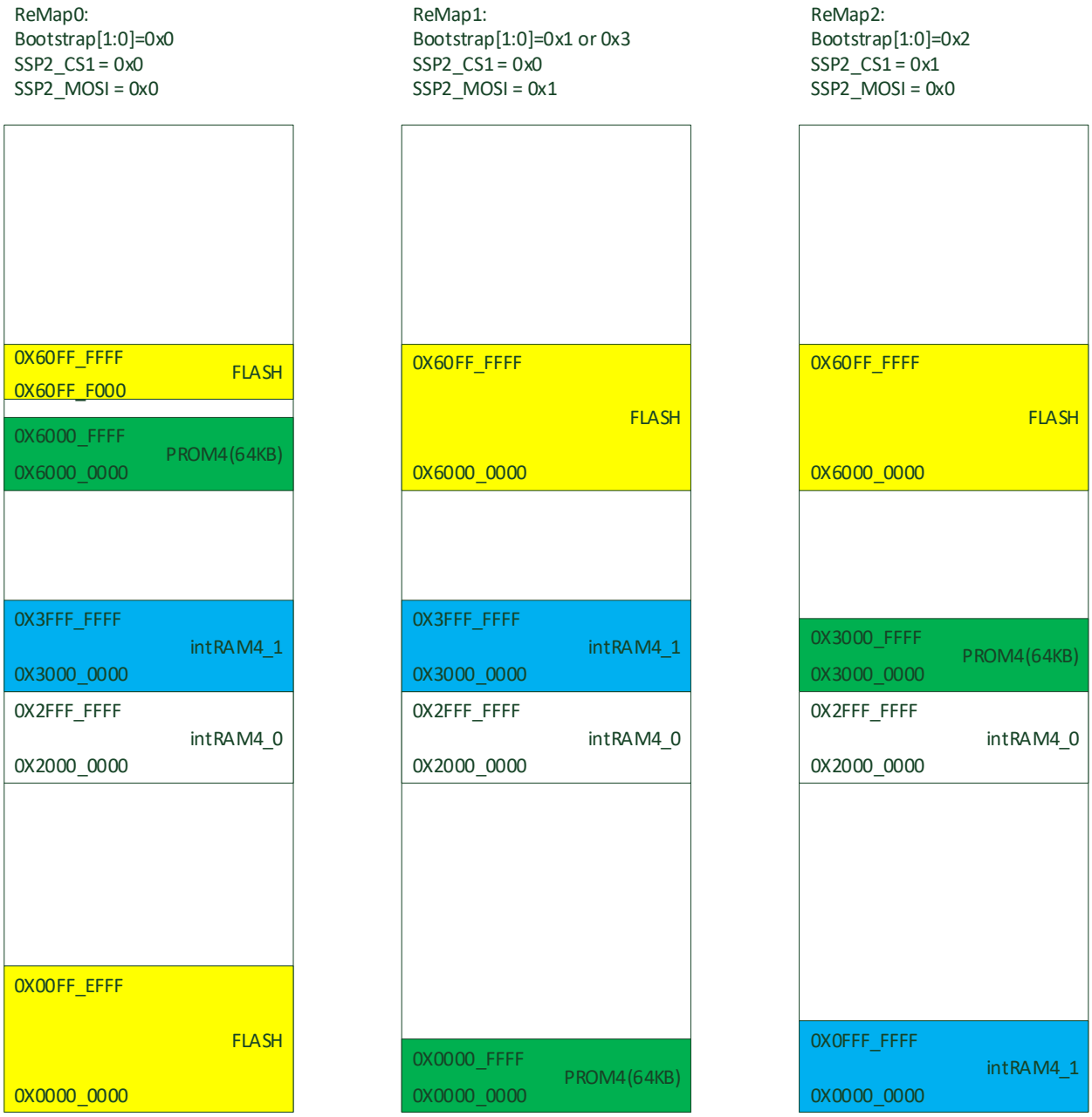


图 6-1. 启动重映射图示

7. Bootstrap

7.1. 概述

设置表 7-1 中描述的 bootstrap 引脚可以决定上电条件和所应用的功能。只有外部复位或上电复位（POR）会触发由 bootstrap 引脚设置的 bootstrap 事件。

表 **7-1. Bootstrap** 功能介绍

Bootstrap Pad	**Registers Map**	功能	描述
{SSP2_CS1,SSP2_MOSI}	bootstrap[1：0]	BootRomSel	0x0：eFlash 0x1：ROM 0x2：IntRAM1 0x3：ROM
{LD_SCL, UART1_RTSn}	bootstrap[3:2]	JTAGSel	当 SoC 处于调试模式时， 0x0：CM4 0x1：CM0，必须先设置 prog_mode（0x5400085C）以释放 M0。 0x2：保留 0x3：CM4
UART1_SOUT	bootstrap[4]	DebugMode	0x0：调试模式 0x1：正常模式
LD_SDA	bootstrap[5]	PLLBypassMode	0x0：此系统时钟来自 25MHz 的 Xtal 时钟。 0x1：此系统时钟来自 PLL。
SSP2_FSCS0	bootstrap[6]	SystemClkMode	0x0：以 PLL 的 300MHz 时钟启动。 0x1：以 PLL 的 275MHz 时钟启动。 当 PLL 被旁路时此模式无效。
GPIO7	bootstrap[7]	SWMode	根据固件实现过程。
BSTREN	-	BootstrapEnable	0x0：关闭 bootstrap 侦测。 在这种模式下，SCU 将不会采用以下 pad 的状态来设置 bootstrap 功能。开机后将取其默认值。 LD_SCL UART1_RSTn UART1_SOUT LD_SDA SSP2_FSCS0

			注：此模式不包括 SSP2_CS1，SSP2_MOSI 及 GPIO7。 0x1：开启 bootstrap 侦测。 在此模式下，SCU 将采用所有 pad 的状态来设置 bootstrap 功能。

7.2. Bootstrap 架构与程序

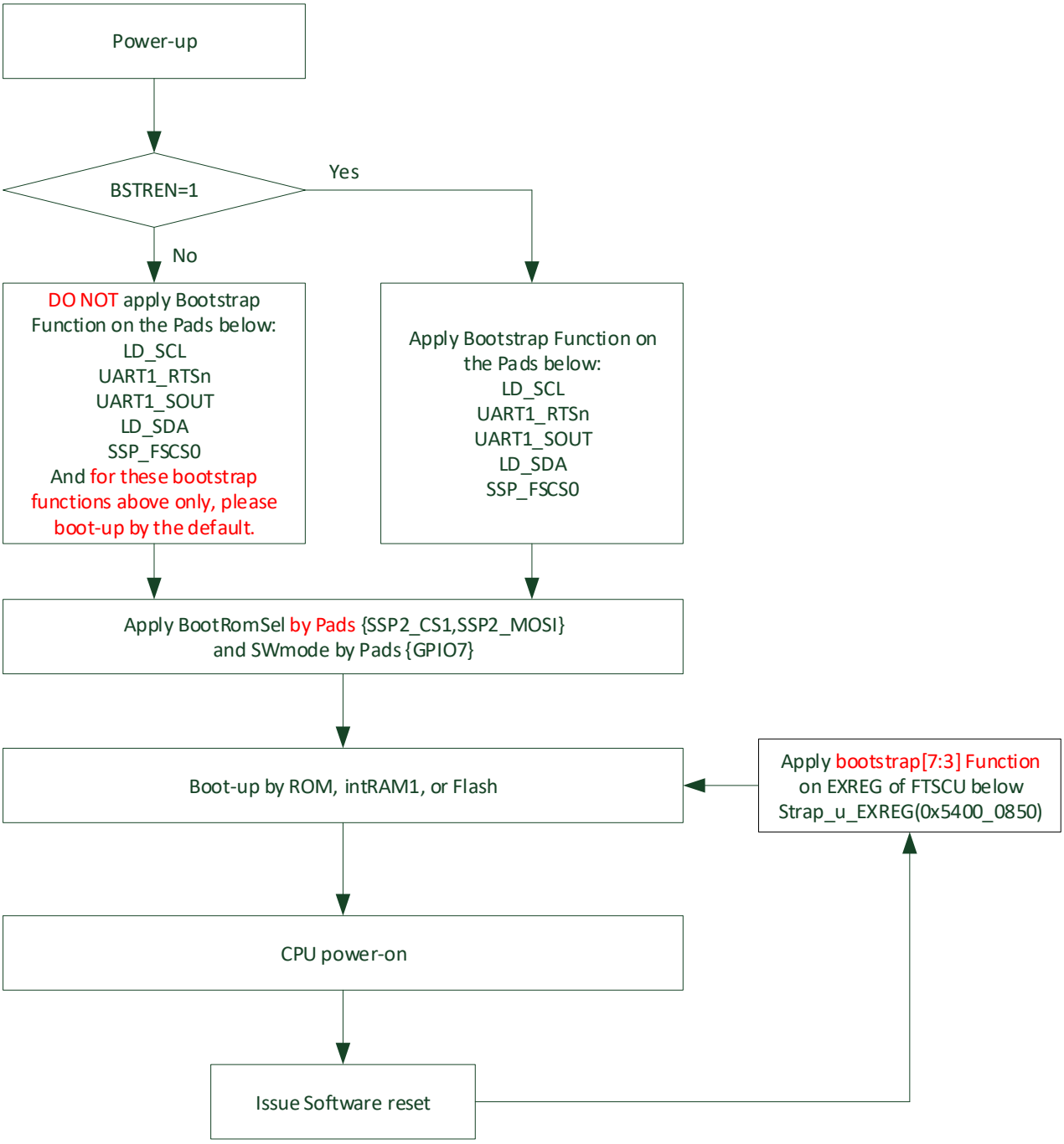


图 7-1. Bootstrap

7.3. 寄存器

7.3.1. STRAP 寄存器

- 地址：0x54000850

  总线：Cortex-M4 APB

  名称：SWSTRAP

  描述：软件 bootstrap 功能。

表 7-2. STRAP 寄存器（地址：0x54000850）

位	名称	类型	默认值	描述
31:16	-	-	-	保留
15	swstrap[7]	R/W	0x1	SW 模式选择。
14	swstrap[6]	R/W	0x1	0x0：以 PLL 的 300MHz 时钟启动。 0x1：以 PLL 的 275MHz 时钟启动（默认）。
13	swstrap[5]	R/W	0x1	0x0：此系统时钟来自 25MHz 的 Xtal 时钟。 0x1：此系统时钟来自 PLL（默认）。
12	swstrap[4]	R/W	0x1	0x0：调试模式。 0x1：正常模式（默认）。
11:10	swstrap[3:2]	R/W	0x3	当 SoC 为调试模式时， 0x0：CM4 0x1：CM0 0x2：保留 0x3：CM4（默认）
9:8	swstrap[1：0]	R/W	0x3	0x0：Flash 0x1：ROM 0x2：IntRAM1 0x3：ROM（默认）
7:0	-	-	-	保留。

7.3.2. HWSTRAP 寄存器

- 地址：0x54000854

  总线：Cortex-M4 APB

  名称：HWSTRAP

  描述：硬件 bootstrap 状态。

表 7-3. HWSTRAP 寄存器（地址：0x54000854）

位	名称	类型	默认值	描述
31:16	-	-	-	保留
15:8	strap_init	R	-	硬件 bootstrap 引脚之数值，仅在上电或外部复位中捕获。
7:0	-	-	-	

8. 功能说明

8.1. 版本控制寄存器

8.1.1. 概述

版本控制寄存器。

8.1.2. HWVCODE 寄存器

- 地址：0x700008FC

  总线：Cortex-M4 及 Cortex-M0 APB

  名称：HWVCODE

  描述：硬件版本控制码

表 8-1. HWVCODE 寄存器（地址：0x700008FC）

位	名称	类型	默认值	描述
31:0	HWVCODE	R	0x6770B26F	硬件版本控制码。

8.2. Cortex-M0 系统控制

8.2.1. TXEV/ RXEV

表 8-2. CPU TXEV/ RXEV 连接信号

Cortex-M0 内核	讯号源	Cortex-M4 内核
TXEV	cm0_int	RXEV
RXEV	cm4_int	TXEV

8.2.2. 中断控制

通过中断控制器提供给嵌入 Cortex-M0 内核的中断事件列表。表 8-3 列出了 V6201X 系统中可提供对 Cortex-M0 的中断请求 IP。

表 8-3. Cortex-M0 中断源

名称	IRQ	触发模式	触发电平	描述
cm4_intr	IRQ 0	LEVEL	高	从 CM4 TXEV 到 CM0 RXEV
gpio0_0_intr	IRQ 1	LEVEL	高	从 CM4，gpio0[25]
gpio0_1_intr	IRQ 2	LEVEL	高	从 CM4，gpio0[26]
-	-	-	-	保留。
reg00_intr	IRQ 4	LEVEL	高	当位于 0x7000_0800 的寄存器数据发生改变。
reg04_intr	IRQ 5	LEVEL	高	当位于 0x7000_0804 的寄存器数据发生改变。
gpio1_intr	IRQ 6	LEVEL	高	GPIO1 中断。
timer1_intr	IRQ 7	LEVEL	高	PWM/Timer1 的全局中断。
timer1_1_intr	IRQ 8	LEVEL	高	PWM/Timer1 通道 1。
timer1_2_intr	IRQ 9	LEVEL	高	PWM/Timer1 通道 2。
timer1_3_intr	IRQ 10	LEVEL	高	PWM/Timer1 通道 3。
timer1_4_intr	IRQ 11	LEVEL	高	PWM/Timer1 通道 4。
-	IRQ 12	-	-	保留。
-	IRQ 13	-	-	保留。
-	IRQ 14	-	-	保留。
-	IRQ 15	-	-	保留。

名称	IRQ	触发模式	触发电平	描述
dma0_intr	IRQ 16	LEVEL	高	DMA0 寄存器中断。
crc0_intr	IRQ 17	LEVEL	高	CRC0 中断。
-	IRQ 18	-	-	保留。
phy_intr	IRQ 19	LEVEL	高	PHY 事件中断。
tc_cmp_intr	IRQ 20	LEVEL	高	来自 TC 比较事件的中断。
IR3_intr	IRQ 21	LEVEL	高	来自 PHY_ IR3[31:16]比较事件的中断。
-	IRQ 22	-	-	保留。
-	IRQ 23	-	-	保留。
-	IRQ 24	-	-	保留。
-	IRQ 25	-	-	保留。
-	IRQ 25	-	-	保留。
-	IRQ 26	-	-	保留。
-	IRQ 27	-	-	保留。
-	IRQ 28	-	-	保留。
-	IRQ 29	-	-	保留。
-	IRQ 30	-	-	保留。
-	IRQ 31	-	-	保留。

8.2.3. DMA 请求与确认

表 8-4. 握手模式下的 VGDMA 请求与确认

DMA 通道	参数	描述
0	-	保留。
1	-	保留。
2	-	保留。
3	-	保留。
4	-	保留。
5	-	保留。
6	u_rxdfifo	为 PHY 的 PKTCTRL 接收端 FIFO 控制器请求和确认的 RX 通道。
7	u_txdfifo	为 PHY 的 PKTCTRL 传送端 FIFO 控制器请求和确认的 TX 通道。

8.2.4. Cortex-M0 相关寄存器

8.2.4.1. STCALIB 寄存器

- 地址：0x54020028

  总线：Cortex-M0 AHB

  名称：STCALIB

  描述：SysTick 计数器参数。

表 8-5. STCALIB 寄存器（地址：0x54020028）

位	名称	类型	默认值	描述
31:26	-	-	-	保留
25:0	STCALIB	R/W	0x270F	SysTick 计数器参数。

8.2.4.2. PCLKENPH 寄存器

- 地址：0x5402002C

  总线：Cortex-M0 AHB

  名称：PCLKENPH

  描述：在 APB 总线上 PCLKEN 的相位选择。

表 8-6. PCLKENPH 寄存器（地址：0x5402002C）

位	名称	类型	默认值	描述
31:3	-	-	-	保留
2:0	PCLKENPH	R/W	0x3	在 APB 总线上 PCLKEN 的相位选择。

8.2.4.3. STCLK_div 寄存器

- 地址：0x54020030

  总线：Cortex-M0 AHB

  名称：STCLK_div

  描述：SysTick 计数器的时钟分频器。

表 8-7. STCLK_div 寄存器（地址：0x54020030）

位	名称	类型	默认值	描述
31:10	-	-	-	保留。
9:0	STCLK_div	R/W	0x258	SysTick 计数器的时钟分频器。

8.3. Cortex-M4 系统控制

8.3.1. TXEV/RXEV

表 8-8. CPU TXEV/RXEV 连接讯号

Cortex-M0 内核	讯号源	Cortex-M4 内核
TXEV	cm0_int	RXEV
RXEV	cm4_int	TXEV

8.3.2. 中断控制

通过中断控制器提供给嵌入 Cortex-M4 内核的中断事件列表。表 8-9 列出了 V6201X 系统中可提供对 Cortex-M4 的中断请求 IP。

表 8-9. Cortex-M4 中断源

名称	IRQ	触发模式	触发电平	描述
sys_intr	IRQ 0	LEVEL	高	系统控制单元，SYSC int
cm0_intr	IRQ 1	LEVEL	高	从 CM0 TXEV 到 CM4 RXEV
gpio1_1_intr	IRQ 2	LEVEL	高	来自 CM0，（gpio1[1] | zcd_int） CM0 SW 应该透过写入 gpio1 [1] 0x1，然后写入 0x0 来触发脉冲，以发出中断。
gpio1_2_intr	IRQ 3	LEVEL	高	来自 CM0，（gpio1[2]） CM0 SW 应该透过写入 gpio1 [1] 0x1，然后写入 0x0 来触发脉冲，以发出中断。
gpio0_intr	IRQ 4	LEVEL	高	GPIO0 中断。
-	IRQ 5	-	-	保留。
-	IRQ 6	-	-	保留。
crc4_intr	IRQ 7	LEVEL	高	CRC4 中断。
aes_intr	IRQ 8	LEVEL	高	安全硬件加速器中断。

名称	IRQ	触发模式	触发电平	描述
-	IRQ 9	-	-	保留。
wdt_intr	IRQ 10	LEVEL	高	看门狗中断
-	IRQ 11	LEVEL	-	保留。
reg08_intr	IRQ 12	LEVEL	高	来自 CM0 寄存器：reg08_int 当位于 0x7000_0808 寄存器数据发生改变时。
ssp1_intr	IRQ 13	LEVEL	高	从模式之 SSP1
dma4_intr	IRQ 14	LEVEL	高	DMA4
ssp2_intr	IRQ 15	LEVEL	高	主模式/从模式之 SSP2
ssp0_intr	IRQ 16	LEVEL	高	主模式 SSP0
reg0C_intr	IRQ 17	LEVEL	高	来自 CM0 寄存器：reg0C_int 当位于 0x7000_080C 寄存器数据发生改变时。
-	IRQ 18	-	-	保留。
-	IRQ 19	-	-	保留。
Timer0_1_intr	IRQ 20	LEVEL	高	PWM/Timer0 通道 1
Timer0_2_intr	IRQ 21	LEVEL	高	PWM/Timer0 通道 2
Timer0_3_intr	IRQ 22	LEVEL	高	PWM/Timer0 通道 3
Timer0_4_intr	IRQ 23	LEVEL	高	PWM/Timer0 通道 4
iic0_intr	IRQ 24	LEVEL	高	I^2C
-	IRQ 25	-	-	保留
UART4_intr	IRQ 26	LEVEL	高	UART4
UART5_intr	IRQ 27	LEVEL	高	UART5
UART0_intr	IRQ 28	LEVEL	高	UART0
UART1_intr	IRQ 29	LEVEL	高	UART1
UART2_intr	IRQ 30	LEVEL	高	UART2
UART3_intr	IRQ 31	LEVEL	高	UART3

8.3.3. Cortex-M4 相关寄存器

8.3.3.1. TSVALUEB1 寄存器

- 地址：0x5400082C

  总线：Cortex-M4 APB

  名称：TSVALUEB1

  描述：全局时间戳记值。

表 8-10. TSVALUEB1 寄存器（地址：0x5400082C）

位	名称	类型	默认值	描述
31:0	TSVALUEB[31:0]	R/W	0x0	全局时间戳值。 如果需要全局时间戳，请连接到自然二进制计数值。如果不使用，用户必须将此信号绑定到0x00000000。计数值可以从与 HCLK 不同的时钟中导出，但必须与 HCLK 完全同步。

8.3.3.2. TSVALUEB2 寄存器

- 地址：0x54000830

  总线：Cortex-M4 APB

  名称：TSVALUEB2

  描述：全局时间戳记值（续）。

表 8-11. TSVALUEB2 寄存器（地址：0x54000830）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:0	TSVALUEB[47:32]	R/W	0x0	全局时间戳值。 如果需要全局时间戳，请连接到自然二进制计数值。如果不使用，用户必须将此信号绑定到0x00000000。计数值可以从与HCLK不同的时钟中导出，但必须与HCLK完全同步。

8.3.3.3. STCALIB1 寄存器

- 地址：0x5400083C

总线：Cortex-M4 APB

名称：STCALIB1

描述：Cortex-M4 内核外设参数控制寄存器。

表 8-12. STCALIB1 寄存器（地址：0x5400083C）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15	INTNMI (CORTEXM4)	R/W	0x0	不可屏蔽中断（CORTEXM4）。
14	RXEV(CORTEXM4)	R/W	0x0	从 WFE 指令唤醒。
13	EDBGRQ (CORTEXM4)	R/W	0x0	外部调试请求。 来自ETM的联合调试请求和多处理器调试支持，以连接到CoreSight嵌入式交叉调试器。如果不需要多处理器调试支持，EDBGRQ可以连接到ETM上的ETMDBGRQ。
12	DBGRESTART (CORTEXM4)	R/W	0x0	外部重新启动请求。 多处理器调试支持连接到CoreSight嵌入式交叉调试器。如果不需要多处理器调试支持，则DBGRESTART必须处于低位。
11:0	-	-	-	保留。

8.3.3.4. STCALIB2 寄存器

- 地址：0x54000840

  总线：Cortex-M4 APB

  名称：STCALIB2

  描述：Cortex-M4 内核外设参数控制器存器。

表 8-13. STCALIB2 寄存器（地址：0x54000840）

位	名称	类型	默认值	描述
31:26	-	-	-	保留。
25:0	STCALIB(CortexM4)	R/W	0x270F	STCALIB[25]： 表示未集成其他参考时钟源。如果STCLK已绑定，则拉高信号。 STCALIB[24]：

				如果系统计时器时钟、外部参考时钟或由STCALIB[25]指示的FCLK可以保证10 ms的精确倍数，则将此信号设为低电平。否则，将此信号设为高电平。 STCALIB[23:0]： 提供一个整数值，用于计算基准时钟的10毫秒（100Hz）延迟，如果基准时钟未实现，则提供FCLK。例如，如果未实现任何引用，且FCLK为50 MHz，则应用值0x07A11F。对于STCLK（1 MHz），STACALIB的默认值是26'h000270f。

8.3.3.5. STCALIB3 寄存器

- 地址：0x54000844

  总线：Cortex-M4 APB

  名称：STCALIB3

  描述：Cortex-M4 内核外设参数控制寄存器。

表 8-14. STCALIB3 寄存器（地址：0x54000844）

位	名称	类型	默认值	描述
31:11	-	-	-	保留。
10	SLEEPHOLDREQn (CortexM4)	R/W	0x0	请求延长睡眠时间。 仅当 CPU 处于睡眠模式时才有效。
9:0	STCLK_div	R/W	0x258	SysTick 计时器时钟分频系数。 1：SysTick 计时器时钟= CPU 时钟/ 2 2：SysTick 计时器时钟= CPU 时钟/ 3 …

8.4. 系统控制单元（SCU）

8.4.1. 概述

Vango SCU 是一系统控制单元，旨在为片上系统（SoC）提供电源和时钟管理，以接通芯片电源启动，开机设定，频率更改，软关闭（休眠或深度睡眠），及关闭电源（分流）等功能使用。

Vango SCU 支持以下功能：

- 上电启动时序
- 晶振与 PLL 控制
- 动态频率转换
- 以多个集群动态频率变化来加速或减速 CPU
- 以动态频率变化来加速或减速总线（Bus）
- 系统/外设时钟控制与生成
- 复位信号生成
- 操作模式及状态机控制
- 通用外设控制寄存器
- 内存数据保持
- 扩展寄存器接口

当打开系统电源时，VGSCU 将从引脚接收初始值，然后对 PLL/DLL 进行编程，使用系统启动时序为系统总线和处理器产生预期的时钟线。在完成 PLL/DLL 锁定程序后，VGSCU 将按时序取消复位信号。SoC 完成启动时序并进入工作状态后，CPU 将开始撷取代码以执行程序。

在工作状态下，VGSCU提供了几种时钟管理功能，以根据性能和电力需求要求来调整系统运行速度和总线频率。这些功能包括频率变化和时钟门控，以提供动态频率变化的安全控制程序。为了省电，可以在工作状态下透过寄存器编程来控制时钟里未使用的区域。

8.4.2. 系统控制寄存器

8.4.2.1. BOOTUPSTATUS 寄存器

- 地址：0x54000000

  名称：BOOTUPSTATUS

  描述：电源控制与时序启动状态。

表 8-15. BOOTUPSTATUS 寄存器（地址：0x54000000）

位	名称	类型	默认值	描述
31:11	-	-	-	保留。
10	SMR	R/W	0x0	Timer2 的唤醒始能中断。
9	WDR	R/W	0x0	通过看门狗复位启动。 1：看门狗复位将发生。 0：看门狗复位不会发生。
8	HWR	R/W	0x0	通过硬件复位启动。 1：将发生硬件复位。 0：将不会发生硬件复位。
7:0	-	-	-	保留。

8.4.2.2. PWRMODE 寄存器

- 地址：0x54000020

  总线：Cortex-M4 APB

  名称：PWRMODE

  描述：用于电源管理的电源模式控制寄存器。

表 8-16. PWRMODE 寄存器（地址：0x54000020）

位	名称	类型	默认值	描述
31:29	-	-	-	保留。
28	FCS_PLL_RSTn	R/W	0x0	PLL 频率转换时序复位。 如果不满足 PLL 输出频率，则应在 FCS 模式下复位 PLL。 0：复位 PLL。 1：在 FCS 中保持 PLL 活动状态。
27:24	REMAP	R/W	STRAP	目标代码重映射可编程寄存器。 在发出重新启动功能以更改 AHB 总线的地址重新映像后，用于设置端口的输出值 remap_out。寄存器内容应在重启命令之前保留。 0：Flash 1，3：ROM

				2：RAM
23:12	-	-	-	保留。
10	SW_RST	R/W	0x0	软件重置命令。 设置该位以发出复位信号，该间隔信号可以通过编程 SWRST_CTRL（偏移量 0x02C）进行调整，并在复位完成后由硬件清除。
9	-	-	-	保留。
8	PLL_UPDATE	R/W	0x0	PLL 更新命令。 PLL 设置值由 PLL 编程寄存器直接更新。
7	REBOOT	R/W	0x0	重启命令。 设置该位以发出复位以启动 CPU 和总线状态。 在组合命令中，在完成另一个操作时序后，将取消复位。但是，此操作禁止在睡眠模式中执行。
6	FCS	R/W	0x0	频率变化时序（FCS）。 设置该位以输入频率变化时序。完成 FCS 后将自动清除该位。 如果用户想透过下一个启动时序中的值来设置 PLL，则在设置软关断模式命令时应清除该位。
5:0	-	-	-	保留。

8.4.2.3. SWRSTCTRL 寄存器

- 地址：0x5400002C

  总线：Cortex-M4 APB

  名称：SWRSTCTRL

  描述：软件复位控制。

表 8-17. SWRSTCTRL 寄存器（地址：0x5400002C）

位	名称	类型	默认值	描述
31:16	SWRST_WAIT	R/W	0x0	APB 总线时钟的软件复位延迟周期。
15:0	SWRST_ACTIVE	R/W	0x0	APB 总线时钟的软件复位有效周期。

8.4.2.4. PLLCTRL 寄存器

- 地址：0x54000030

  总线：Cortex-M4 APB

  名称：PLLCTRL

  描述：PLL 控制寄存器用于设置 PLL 值。

表 8-18. PLLCTRL 寄存器（地址：0x54000030）

位	名称	类型	默认值	描述
31:24	PLLNS	R/W	0x16	设置嵌入式的 PLL 的 N 值以控制 PLL 频率输出。
23:16	PLLMS	R/W	0x2	设置嵌入式的 PLL 的 M 值，以控制 PLL 频率输出。
15:8	-	-	-	保留。
7:4	CLKIN_MUX	R/W	0x5	系统时钟源复用器（MUX）选择。 该字段在 FCS 模式下更改系统时钟源。 CLKIN_MUX[1：0]： 0，1：系统时钟 = 1/2 (PLL clock) 2：系统时钟 = 1/3 (PLL clock) 3：系统时钟 = 1/4 (PLL clock) CLKIN_MUX[2]： 0：APB 时钟 = 1/8 (AHB clock) 1：APB 时钟 = 1/4 (AHB clock) CLKIN_MUX[3]：保留。
3:2	-	-	-	保留。
1	PLLSTABLE	R	0x0	PLL 锁定和稳定状态。 1：PLL 是稳定的。 0：PLL 是不稳定的。
0	PLLEN	R/W	0x1	PLL 使能控制。当该位置 1 时，PLL 无法直接开启/关闭。该过程应伴随 FCS 命令以停止或启用 PLL。 1：使能 PLL。 0：关闭 PLL。

8.4.2.5. WAKEUPEN 寄存器

- 地址：0x5400081C

  总线：Cortex-M4 APB

  名称：WAKEUPEN

  描述：睡眠唤醒使能控制寄存器。

表 8-19. WAKEUPEN 寄存器（地址：0x5400081C）

位	名称	类型	默认值	描述
31:19	-	-	-	保留。
18	tm4_iwke	R/W	0x0	Timer 4 中断的唤醒使能。
17	tm3_iwke	R/W	0x0	Timer 3 中断的唤醒使能。
16	tm2_iwke	R/W	0x0	Timer 2 中断的唤醒使能。
15	tm1_iwke	R/W	0x0	Timer 1 中断的唤醒使能。
14	wdt_iwke	R/W	0x0	唤醒使能看门狗的中断。
13	gpio_iwke	R/W	0x0	唤醒使能 GPIO 的中断。
12	i2c_iwke	R/W	0x0	唤醒使能 I^2C 的中断。
11	gmac_iwke	R/W	0x0	唤醒使能 GMAC 的中断。
10	uart3_iwke	R/W	0x0	UART3 中断的唤醒使能。
9	uart2_iwke	R/W	0x0	UART2 中断的唤醒使能。
8	uart1_iwke	R/W	0x0	UART1 中断的唤醒使能。
7	uart0_iwke	R/W	0x0	UART0 中断的唤醒使能。
6	uart5_iwke	R/W	0x0	UART5 中断的唤醒使能。
5	uart4_iwke	R/W	0x0	UART4 中断的唤醒使能。
4	SSP1_iwke	R/W	0x0	SSP1 中断的唤醒使能。
3	vg0_iwke	R/W	0x0	vg_gpio_0 中断的唤醒使能。
2	vg1_iwke	R/W	0x0	vg_gpio_1 及 zcd 中断的唤醒使能。
1	CM0_txev_iwke	R/W	0x0	cm0_txev 中断的唤醒使能。
0	sysc_iwke	R/W	0x0	SYSC 中断的唤醒使能。

8.4.3. IPs 时钟，多路选择器 & 复位控制

8.4.3.1. IPMISC_CTRL 寄存器

- 地址：0x5400084C

  总线：Cortex-M4 APB

  名称：IPMISC_CTRL

  描述：IPs 时钟，多路复用器&复位控制。

表 8-20. IPMISC_CTRL 寄存器（地址：0x5400084C）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15	ld_clk_en	R/W	0x1	引脚 LD_CLK 复用使能。
14	ld_clk_swrst_n	R/W	0x1	LD_CLK 软件复位控制。
13:9	sdr_rdclk_delay_sel	R/W	0x0	SDRAM 控制器时钟延迟选择。
8:6	-	-	-	保留。
5	ssp_mux_sel	R/W	0x1	SSP 时钟选择。 1：PLL时钟输出/4 0：PLL 时钟输出/3
4	phy_mux_sel	R/W	0x0	PHY基带时钟选择。 1：PLL 时钟输出/2 0：PLL 时钟输出
3	uart_swrst_n	R/W	0x1	UART 软件复位。 0：复位使能 1：复位关闭
2	afe_clk_en	R/W	0x0	AFE 时钟使能。 0：关闭 1：使能
1	afe_swrst_n	R/W	0x1	AFE 软件复位。 0：复位始能 1：复位关闭
0	gatepclk_n	R/W	0x1	APB总线时钟门控信号。 它应该设定至1'b1。

8.4.3.2. UART_CLKCTRL 寄存器

- 地址：0x54000868

  总线：Cortex-M4 APB

  名称：UART_CLKCTRL

  描述：UART 外设参数时钟设控。

表 8-21. UART_CLKCTRL 寄存器（地址：0x54000868）

位	名称	类型	默认值	描述
31:21	-	-	-	保留。
20	uart_clk_mux_sel	R/W	0x0	UART 时钟选择。 1：XTAL 2：PLL 时钟输出。
19:10	ns_uart_clk_bypass_div_gen	R/W	0x9	uart_clk_bypass 的分频系数 N。
9:0	ms_uart_clk_bypass_div_gen	R/W	0xD	uart_clk_bypass 的分频系数 M。

8.4.4. Vango 控制寄存器

8.4.4.1. GLB_COUNTER 寄存器

- 地址：0x54000858

  总线：Cortex-M4 APB

  名称：GLB_COUNTER

  描述：基于 PHY 25MHz 广播的 PLC 全局计数器。

表 8-22. GLB_COUNTER 寄存器（地址：0x54000858）

位	名称	类型	默认值	描述
31:0	glb_count	R	0x0	从 PHY 广播的 PLC 全局计数器值。

8.4.4.2. VSBCTRL0 寄存器

- 地址：0x5400085C

  总线：Cortex-M4 APB

名称：VSBCTRL0

描述：Vango 自定控制寄存器 0

表 8-23. VSBCTRL0 寄存器（地址：0x5400085C）

位	名称	类型	默认值	描述
31:27	-	-	-	保留。
26	prog_mode_PHY	R/W	0x1	设置高位以开启 PHY 的存储器编程。
25	STCLK_u_CORTEXM_sel	R/W	0x0	SysTick 定时器时钟复用器（MUX）选择。 1：tmr1_out 0：分频器产生的 STCLK。
24	prog_en_CM0	R/W	0x1	设置高位以开启 Cortex-M0 的存储器编程。
23:0	-	-	-	保留。

8.4.4.3. BSTREN 寄存器

- 地址：0x54000860

  总线：Cortex-M4 APB

  名称：BSTREN

  描述：Pad 上的 Bootstrap 模式设置。

表 8-24. BSTREN 寄存器（地址：0x54000860）

位	名称	类型	默认值	描述
31:2	-	-	-	保留。
1	BSTREN	R	0x0	Bootstrap 使能。 1：使用 Bootstrap IO 引脚的开机电平状态选择开机功能 0：使用 Bootstrap 寄存器设定选择开机功能
0	-	-	-	保留。

8.4.4.4. SWRST_CLR 寄存器

- 地址：0x54000878

  总线：Cortex-M4 APB

  名称：SWRST_CLR

描述：软件复位状态清除。

表 8-25. SWRST_CLR 寄存器（地址：0x54000878）

位	名称	类型	默认值	描述
31:30	-	-	-	保留。
29	sw_rst_status_clr	R/W	0x1	软复位状态清除寄存器。 该位置为“1”以清除 sw_rst_status（0x54000884）。
28	urgent_in_inv	R/W	0x1	设置高位为反相输入 LD_URGENT 的事件信号。
27	dying_gasp_in_inv	R/W	0x1	设置高位为反相输入 DYING_GASP 的事件信号。
26	phy_gpio_inv	R/W	0x1	设置高位为反相输入 PHY_GPIO 的信号。
25:0	-	-	-	保留。

8.4.4.5. CUSTREG4 寄存器

- 地址：0x54000888

  总线：Cortex-M4 APB

  名称：CUSTREG4

  描述：自定义寄存器。

表 8-26. CUSTREG4 寄存器（地址：0x54000888）

位	名称	类型	默认值	描述
31:0	custreg4	R/W	0x0	Cortex-M4 内核 APB 总线上的自定义寄存器，只能由 POR 复位。

8.4.4.6. CUSTREG0 寄存器

- 地址：0x700008F0

  总线：Cortex-M0 AHB

  名称：CUSTREG0

  描述：自定义寄存器。

表 8-27. CUSTREG0 寄存器（地址：0x700008F0）

位	名称	类型	默认值	描述
31:0	custreg0	R/W	0x0	Cortex-M0 内核 AHB 总线上的自定义寄存器，只能

				由 POR 复位。

8.5. 中断控制寄存器

8.5.1. 概述

中断控制寄存器透过四个字的数据更改事件来实践，并用于在 MAC 和 PHY 层之间进行信息握手。

8.5.2. 中断控制寄存器列表与描述

表 8-28. 中断控制寄存器列表

地址	名称	类型	默认值	描述
0x70000800	reg00_intflag[31:0]	R/W	0x00000000	Reg00 的 32 位中断标志。 透过 CM4 设置每个位以触发对 CM0 的中断。
0x70000804	reg04_intflag[31:0]	R/W	0x00000000	Reg004 的 32 位中断标志。 透过 CM4 设置每个位以触发对 CM0 的中断。
0x70000808	reg08_intflag[31:0]	R/W	0x00000000	Reg08 的 32 位中断标志。 透过 CM0 设置每个位以触发对 CM4 的中断。
0x7000080C	reg0C_intflag[31:0]	R/W	0x00000000	Reg0C 的 32 位中断标志。 通过 CM0 设置每个位以触发对 CM4 的中断。
0x70000810	reg00_inten[31:0]	R/W	0x00000000	Reg00 的 32 位中断使能。
0x70000814	reg04_inten[31:0]	R/W	0x00000000	Reg04 的 32 位中断使能。
0x70000818	reg08_inten[31:0]	R/W	0x00000000	Reg08 的 32 位中断使能。
0x7000081C	reg0C_inten[31:0]	R/W	0x00000000	Reg0C 的 32 位中断使能。
0x70000820	reg00_intclr[31:0]	R/W	0x00000000	设置每个位以分别清除 reg00 的 32 位中断。 清除事件完成后，硬件将复位每个位。
0x70000824	reg04_intclr[31:0]	R/W	0x00000000	设置每个位以分别清除 reg04 的 32 位中断。 清除事件完成后，硬件将复位每个位。
0x70000828	reg08_intclr[31:0]	R/W	0x00000000	设置每个位以分别清除 reg08 的 32 位中断。

				清除事件完成后，硬件将复位每个位。
0x7000082C	reg0C_intclr[31:0]	R/W	0x00000000	设置每个位以分别清除 reg0C 的 32 位中断。 清除事件完成后，硬件将复位每个位。

8.6. UART

8.6.1. 概述

V6201X 支持 3 个 UART 接口。UART 控制器是一个串行通信组件，提供反向兼容 16550 UART 操作模式，以支持现有的通信软件。

8.6.2. 特性

UART 通信控制器提供以下功能：

- 支持 AMBA 2.0 APB 总线。
- 高达 460.8 kbps 可编程波特率。
- 支持仅对 RTS/CTS 的硬件流量控制。
- 可编程数据字长（5、6、7 或 8 位）。
- 可配置的停止位。
- TX/RX FIFOs 深度为 32 字节。

8.6.3. 预分频器与波特率发生器

8.6.3.1. 预分频器

预分频器使用1至2^5-1的除数对 **io_irda_uclk** 进行分频，并为波特率产生器产生一个预分频的时钟：**PB16XCLK**。

8.6.3.2. 波特率发生器

波特率产生器能够使用 1 到 2^{16}-1 的除数对预分频时钟 **PB16XCLK** 进行分频，并产生 16 倍的波特率时钟：**B16XCLK**。

8.6.4. 寄存器摘要

UART0 基地址：在 Coretx-M4 AMBA APB 上为 0x54010000

UART1 基地址：在 Cortex-M4 AMBA APB 上为 0x54080000

UART2 基地址：在 Coretx-M4 AMBA APB 上为 0x54090000

UART3 基地址：在 Coretx-M4 AMBA APB 上为 0x540A0000

UART4 基地址：在 Coretx-M4 AMBA APB 上为 0x540B0000

UART5 基地址：在 Coretx-M4 AMBA APB 上为 0x540C0000

表 8-29. UART 寄存器摘要

地址	名称	类型	默认值	描述
0x00	RBR	R	0x00	接收缓冲寄存器
	THR	W	0x00	发送保持寄存器
0x04	IER	R/W	0x00	中断使能寄存器
0x08	IIR	R	0x01	中断识别寄存器
	FCR	W	0x00	FIFO 控制寄存器
0x0C	LCR	R/W	0x00	线路控制寄存器
0x10	MCR	R/W	0x00	调制控制寄存器
0x14	LSR	R	0x60	线路状态寄存器
	TST	W	0x00	测试寄存器
0x18	MSR	R	0x00	调制状态寄存器
DLAB = 1 时可访问寄存器				
0x00	DLL	R/W	0x01	波特率除数锁存器（DL）之最低有效位（LSB）
0x04	DLM	R/W	0x00	波特率除数锁存器（DL）之最高有效位（MSB）
0x08	PSR	R/W	0x01	预分频寄存器

8.6.5. 寄存器描述

8.6.5.1. RBR 寄存器

- 地址：0x000 **读模式**

  名称：RBR

  描述：用户可以通过读取此"读位置"来获取数据；为 RX FIFO 的数据读取端口。

  - FIFOs 使能：

    一旦一个字符在接收移位寄存器中组装好，内容将被写入RXFIFO。这个RBR指的是当前读指针指示的RXFIFO位置。

- FIFOs关闭：

  一旦一个字符在接收移位寄存器中组装好，内容将被写入 RXFIFO。这个 RBR 指的是读指针"0"所指示的 RXFIFO 位置。

表 8-30. RBR 寄存器（地址：0x000 读模式）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:0	RBR	R	0x0	接收数据端口。

8.6.5.2. THR 寄存器

- 地址：0x000**写模式**

  名称：THR

  描述：用户可以将数据写入发送保持寄存器；为TX FIFO的数据写入端口。

  - FIFOs使能：

    一旦发送移位寄存器为空，THR的内容将被写入发送移位寄存器，以将发送的数据移出。该THR代表当前读取指针所指示的TXFIFO位置。

  - FIFOs关闭：

    一旦发送移位寄存器为空，THR的内容将被写入发送移位寄存器，以将发送的数据移出。该THR代表读取指标为"0"时所指示的TXFIFO位置。

如果发送的字符宽度小于八位，则必须向右对齐。左位 （即MSB）为 "无关位"。例如，如果字长为5位，则写入0xd3或0xf3将导致13h字符的发送。在将字符写入该寄存器之前，用户必须确保UART是准备好接受数据进行传输的。例如，检查THR空标志是否已设置至LSR中。

表 8-31. THR 寄存器（地址：0x000 写模式）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:0	THR	W	0x0	发送数据端口。

8.6.5.3. IER 寄存器

- 地址：0x004

名称：IER

描述：该寄存器分别使能每个可能的中断源。这些位中的任何一个为“1”时将使能相应的中断，而为“0”时将关闭相应的中断。以下是中断源的详细说明。

表 8-32. IER 寄存器（地址：0x004）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7	DSREn	R/W	0x0	DSR 流量控制使能。 1：使能 DSR 硬件流量控制。 0：关闭 DSR 硬件流量控制。 当 DSREn 置为“1”时，回送模式不得置为“1”并必须使能 FIFO（FCR[0]）。
6	DTREn	R/W	0x0	DTR 流量控制使能。 1：使能 DTR 硬件流控制。 0：关闭 DTR 硬件流控制。 当 DTREn 置为“1”时，回送模式不得置为“1”并必须使能 FIFO（FCR[0]）。
5	CTSEn	R/W	0x0	CTS 流量控制使能。 1：使能 CTS 硬件流量控制。 0：关闭 CTS 硬件流量控制。
4	RTSEn	R/W	0x0	RTS 流量控制使能。 1：使能 RTS 硬件流量控制。 0：关闭 RTS 硬件流量控制。 将 RTSEn 置为“1”时，不得将回送模式置为“1”且必须使能 FIFO（FCR[0]）。
3	MODEM Status	R/W	0x0	当设置为逻辑 1 时，此位启用 MODEM 状态中断。
2	Receiver Line Status	R/W	0x0	当该位置逻辑 1 时，使能接收器线路状态中断。
1	THR Empty	R/W	0x0	该位置逻辑 1 时，使能发送保持寄存器空中断。
0	Receiver Data Available	R/W	0x0	当该位置为逻辑 1 时，使能接收数据可用中断（以及 FIFO 模式下的字符接收超时中断）。

8.6.5.4. IIR 寄存器

- 地址：0x008 **读模式**

  名称：IIR

  描述：此寄存器的主要目的是识别当前具有最高优先级的中断。UART 控制器实践了一个从最高到最低优先级的四级优先级编码器，如下所示：

  (1.) 接收线路状态 （最高优先级）

  (2.) 接收数据就绪和字符接收超时 （第二优先级）

  (3.) 发送保持寄存器空中断 （第三优先级）

  (4.) 调制状态 （最低优先级）

表 8-33. IIR 寄存器（地址：0x008 读模式）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:6	FIFO mode enable	RO	0x0	当 FCR[0]置为 1 时，这两位将被设置为 1。
5	-	-	-	保留。
4	TX_FIFO_full	RO	0x0	TX FIFO 已满时，此位置“1”。
3	FIFO_mode_only	RO	0x0	在 16450 模式下，该位为 0。在 FIFO 模式下，当超时中断挂起时，此位与位 2 一起设置被设置为 1。
2:1	Intr_Iden_Code	RO	0x0	这些位标识了待处理的最高优先级中断。下表列出并描述了不同的中断条件，识别代码以及复位方法。请注意，要使一个中断源被视为待处理中断，必须使能 IER 中的相应位。
0	Intr_Pending	RO	0x1	该位可用于优先中断环境中，以表示此中断是否为待处理。 0：一个中断为待处理中，IIR 的内容可以用作指向适当中断服务程序的指标。 1：无中断待处理。

表 8-34. 中断控制表说明

IIR				中断设置与复位功能			
Bit3	**Bit2**	**Bit1**	**Bit0**	**优先级**	**中断类型**	**中断源描述**	**中断复位方法**
0	0	0	1	-	无	无中断待处理。	无
0	1	1	0	最高	接收电路状态	在 FIFO 接收数据前端，存在着数据溢出错误，基偶校验错误，成帧错误，或中断	读取线路状态寄存器（LSR）

						指示。 请注意在 LSR 中的 FIFO 错误标志并不会影响此中断事件，此中断事件仅与 RX FIFO 中的前端资料，与任何 LSR 位 1 到 4 中一个存在 1 有关。	
0	1	0	0	第二	接收数据就绪	在非 FIFO 模式下，RHR 寄存器中有可用的接收数据。 在 FIFO 模式下，接收 FIFO 中的字符数等于或大于 FCR 中编程的触发级别。当 FIFO 中的字数保持高于该值时，中断信号将保持活动状态；当微处理器读取必要的字以使 FIFO 中的字小于触发电平时，中断信号将被清除。 请注意这与 LSR 的位 0 没有直接关系，LSR 的位 0 始终表示已准备好至少一个字。	读取接收缓冲寄存器（RBR）
1	1	0	0	第二	字符接收超时	接收 FIFO 中至少有一个字符，并且在选定波特率下对应于四个字符的时间内，没有新的字符会被接收。 如果存在以下情况，将发生 FIFO 的超时中断： 1. 至少有一个字符在 FIFO 中。 2. 最近收到的串行字符比 4 个连续字符时间长（如果编程了 2 个停止位，则第二个停止位将被包括在此延时中）。 3. 最近一次 CPU 对 FIFO 的读取时间比 4 个连续字符时间长。	读取接收缓冲寄存器（RBR）。 或者，在 UART 模式里显示“字符接收超时”，复位 RX FIFO 可清除。
0	0	1	0	第三	发送保持寄存器空闲	在非 FIFO 模式下，1 字节 THR 为空位。在 FIFO 模式下，完整的 16 字节发送 FIFO 为空位，所以可将 1 到 16 个字符写入 THR。也就是说，LSR 中的 THR 空位为 1。	写入发送保存寄存器（THR）。 或者，如果此事件为当前显示的中断类型，则读取中断识别寄存器（IIR）也可将该中断清除（但不会清除 LSR 中的标志）。
0	0	0	0	第四	调制状态	在清除发送（CTS），数据准备就绪（DSR）或载波检测（CD）输入线或振铃指示器	读取调制状态寄存器（MSR）

						（RI）输入线的后沿中检测到更改。也就是说，至少一个 MSR 位 0 至 3 中是 1。	

8.6.5.5. FCR 寄存器

- 地址：0x008 写模式

  名称：FCR

  描述：这是与 IIR 相同位置的写寄存器（IIR 是读寄存器）。该寄存器用于使能和清除 FIFO，以及设置 RX FIFO 触发电平。如果用户复位 FIFO，则 FIFO 状态将不会清除。用户必须通过读取相应的寄存器来清除这些寄存器中的状态。

表 8-35. FCR 寄存器（地址：0x008）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:6	RXFIFO_TRGL	W	0x0	设置 RXFIFO 中断的触发级别。
5:4	TXFIFO_TRGL	W	0x0	设置 TXFIFO 中断的触发级别。
3	-	-	-	保留。
2	TX FIFO Reset	W	0x0	将该位设置为逻辑 1 会清除 TX FIFO 中的所有字节，并将计数器逻辑复位为 0。移位寄存器不会被清除，所以任何有效的接收位都将继续。 该位将自动归零。
1	RX FIFO Reset	W	0x0	将此位设置为逻辑 1 会清除 Rx FIFO 中的所有字节，并将计数器逻辑复位为 0。移位寄存器不会被清除，因此任何有效的接收位都将继续。将该位置 1 还将清除 FIFO 状态。 该位将自动归零。
0	FIFO Enable	W	0x0	将此位置为逻辑 1，将同时使能发送和接收 FIFO（以及 FIFO 状态）。更改该位将自动复位两个 FIFO。 在 FIR 模式下，设备驱动程序应始终将此位置为“1”。

表 8-36. 接收器的 FIFO 触发电平

FCR 代码		32 字节接收器 FIFO 触发电平
Bit 7	**Bit 6**	
0	0	1 字符

0	1	8 字符
1	0	16 字符
1	1	28 字符

表 8-37. 发送器的 FIFO 触发电平

FCR 代码		**32 字节发送器 FIFO 触发电平**
Bit 7	**Bit 6**	
0	0	1 字符
0	1	8 字符
1	0	16 字符
1	1	26 字符

8.6.5.6. LCR 寄存器

- 地址：0x00C

  名称：LCR

  描述：该寄存器控制发送的字符被串行化以及接收的字符被组合和检查的方式。

表 8-38. LCR 寄存器（地址：0x00C）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7	DLAB	R/W	0x0	除数锁存器访问位（DLAB）。 必须设置该位才能访问 DLL，DLM 和 PSR 寄存器为波特率分频器和预分频器编程分频常数。
6	-	-	-	保留。
5	Stick Parity	R/W	0x0	如果 bit[5:3]为逻辑 1，将发送并检查奇偶校验位为"0"。如果位 3 和位 5 为"1"，而位 4 为"0"，将发送并检查奇偶校验位为"1"。如果位 5 为"0"，则确认奇偶校验位将被关闭。
4	Even Parity	R/W	0x0	该位是偶数校验选择位。 如果位 3 为 "1" 而位 4 为 "0"，则将在数据字位和奇偶校验位中发送或检查逻辑高的奇数位。
3	Parity Enable	R/W	0x0	该位是奇偶校验使能位。

位	名称	类型	默认值	描述
				如果此位为“1”，则将在串行数据的最后一个数据字位和停止位之间产生 （发送数据） 或检查 （接收数据）奇偶校验位。如果位 3 为“1” 位 4 也为“1”，将在最后一个数据字位和停止位之间传输或检查偶数 1。
2	Stop Bits	R/W	0x0	该位选择要发送的停止位数量。 如果清零，将仅发送一个停止位。如果置位，将在下一个字符的起始位之前发送两个停止位（1.5 带 5 位数据）。接收器一次只能检查一个停止位。
1	WL1	R/W	0x0	该位与 WL0 界定要发送和接收的数据字长。
0	WL0	R/W	0x0	该位与 WL1 界定要发送和接收的数据字长。

表 8-39. 奇偶设定表说明

LCR 代码			**奇偶校验 （发送或检查位）**
Bit 5 奇偶校验位	**Bit 4 偶校验位**	**Bit 3 奇偶较验使能位**	
X	X	0	无发送或检查位
0	1	1	偶较验位
0	0	1	奇校验位
1	0	1	1
1	1	1	0

表 8-40. 字长度与停止位设定说明

LCR 代码			**字符长度 （Bits）**	**停止位数量**
Bit 2 停止位	**Bit 1 WL1**	**Bit 0 WL0**		
0	0	0	5	1
	0	1	6	
	1	0	7	
	1	1	8	
1	0	0	5	1.5
	0	1	6	2
	1	0	7	

	1	1	8	

8.6.5.7. MCR 寄存器

- 地址：0x010

  名称：MCR

  描述：通过写入该寄存器，用户可设置调制以控制输出。该寄存器也控制回送模式，并提供通用输出。

表 8-41. MCR 寄存器（地址：0x010）

位	名称	类型	默认值	描述
31:7	-	-	-	保留。
6	-	-	-	保留。
5	-	-	-	保留。
4	Loop	R/W	0x0	回送模式控制位。 回送模式旨在测试 UART 或 SIR 通信。
3:2	-	-	-	保留。
1	RTS	R/W	0x0	该位控制“请求发送”，低电平有效输出。
0	DTR	R/W	0x0	该位控制“数据终端就绪”，低电平有效输出。该位为 1 时输出为“0”。当该位清零时，输出“1”。

将此UART设置为回送模式时，会发生以下情况：

- 串行输出在内部连接到串行输入，所以发送的每个字符都会被回送并接收。
- 不使用输入引脚，并将输出引脚设置为“1”（无效状态）。

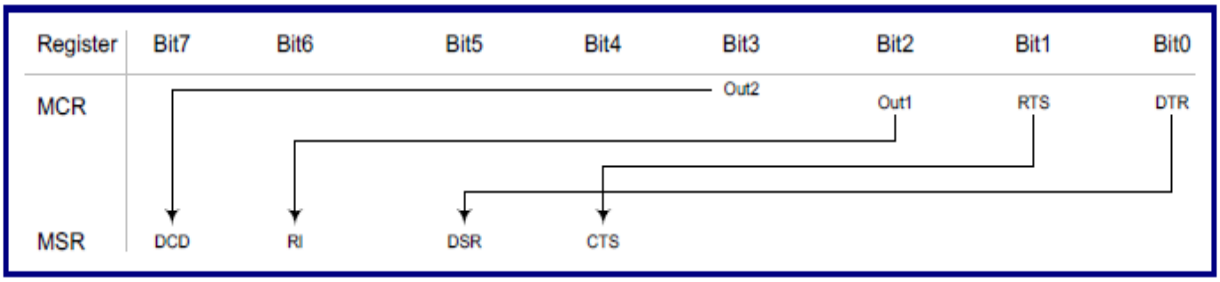


图 8-1. 回送模式下 MCR 和 MSR 之间的相互作用

8.6.5.8. LSR 寄存器

- 地址：0x014 写模式

名称：LSR

描述：该寄存器专门负责通知发送器和接收器的状态用户。为了获取有关接收到的字符信息，在读取从 RBR 接收到的字符之前必须先读取 LSR。

表 8-42. LSR 寄存器（地址：0x014）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7	FIFO Data Error	R	0x0	如果关闭了 FIFO（16450 模式），则该位将始终为零。 如果 FIFO 有效，则一旦 FIFO 中的任何数据字符发生奇偶校验或成帧错误，或发生中断指示有效，该位置 1。当 CPU 读取 LSR 并接收 FIFO 中的其余数据，且其余数据没有在这三个相关标志中的任何一个里头时，该位将被清除。
6	Transmitter Empty	R	0x1	当 THR（或 TX FIFO）和 TSR（发送移位寄存器） 均为空白时，该位为 1。将该位读为 1 表示在 io_irda_sout 输出引脚当前将不进行任何传输，并且传输线路处于空闲状态。一旦将新数据写入 THR，该位将清除。
5	THR Empty	R	0x1	该位表示 UART 已准备好接受新字符进行传输。此外，当发送保持寄存器空中断使能位（IER[1]）设置为高电平时，该位将导致 UART 向 CPU 发出中断。 在非 FIFO 模式下，只要 1 字节 THR 为空白，该位置 1。如果 THR 保留要发送的数据，则在该数据传递到 TSR 时该位将立即置 1。 在 FIFO 模式下，当发送 FIFO 完全为空白时，此位置 1；如果 FIFO 中至少有一个字节等待传递给 TSR 进行发送，则该位置 0。
4	-	-	-	保留。
3	Framing Error	RC	0x0	该位表示接收到的字符没有有效的停止位 （即，在（第一个） 停止位检测到的是 "0"，而不是"1" ）。该位以与奇偶校验错误位相同的方式在接收器 FIFO 中列队。当检测到帧错误时，接收器会尝试重新同步，如果下一个采样再次为零，则将其视为新起始位的起点。 该位在读取 LSR 后立即清除。
2	Parity Error	RC	0x0	设置此位后，根据 LCR 中的当前设置，表示接收到的字符奇偶校验为错误的。该位在接收 FIFO 中列队，因此它与出现错误的特定字符有相关联。 LSR 必须在 RBR 之前被读取：每次 RBR 读取字符时，下一个字符将传递到 FIFO 的前端，并且 LSR 会加载与该 FIFO 前端字符相对应的列队错误标志。 该位在读取 LSR 后立即清除。

1	Overrun Error	RC	0x0	将该位置 1 时，一字符将完全汇编在接收移位寄存器中，而没有可用空间将其放入接收 FIFO 或缓冲寄存器中。 如果 FIFO 是无效的，则只有 1 个字符的接收缓冲寄存器将可使用，并且该 RBR 中的未读数据不会被刚刚接收到的新字符所覆盖。 如果 FIFO 是有效的，则刚刚在接收移位寄存器中所接收的字符将会被覆盖，但是 FIFO 中已经存在的数据将不会被更改。一旦出现数据溢出情况，就会设置"溢出错误"标志。 如果该位为有效位，则不会在 FIFO 中列队。 一读取 LSR 后，该位将被清除。
0	Data Ready	R	0x0	如果已经接收到一个或多个字符并且该字符在接收 FIFO 中等待用户读取，则该位置 1。透过读取接收缓冲寄存器或 FIFO 中的所有数据，将其清除为逻辑 0。

8.6.5.9. MSR 寄存器

- 地址：0x018

  名称：MSR

  描述：此寄存器提供有关四个调制引脚输入状态的信息技术。四个最高有效位（MSB）直接提供了引脚的状态，而四个最低有效位（LSB）则提供了这些引脚变化的相关信息。

  如果由 IER 中的对应位允许，则这四个最低有效位（LSB）会产生一个中断（调制状态中断），只要其中任何一位为"1"，就会产生该中断。每当读取调制状态寄存器时，它们就会复位为逻辑 0。

表 8-43. MSR 寄存器（地址：0x018）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7	DCD	R	0x0	数据载波检测（DCD），是 io_irda_ndcd 输入的补位。
6	RI	R	0x0	振铃指示器（RI），是 io_irda_nri 输入的补位。
5	DSR	R	0x0	数据集就绪（DSR），是 io_irda_ndsr 输入的补位。
4	CTS	R	0x0	清除发送（CTS），是 io_irda_ncts 输入的补位。
3	Delta DCD	R	0x0	如果置位，则表示自上次微处理器读取该位以来，io_irda_ndcd 输入已更改。
2	Trailing edge R1	R	0x0	当在 io_irda_nri 输入引脚中检测到后沿时，该位置 1；也就是说当 io_irda_nri 从 0 变为 1 时，该位置 1。
1	Delta DSR	R	0x0	如果置位，则表示自上次微处理器读取该位以来，

位	名称	类型	默认值	描述
				io_irda_ndsr 输入已更改。
0	Delta CTS	R	0x0	如果置位，则表示自上次微处理器读取该位以来，io_irda_ncts 输入已更改。

8.6.5.10. DLL 和 DLM 寄存器

- 地址：0x000，0x004 （当 DLAB = 1 时）

  名称：DLL，DLM（当 DLAB=1 时）

  描述：除数锁存器是一个 16 位寄存器，其最高有效字节保存在 DLM 中，最低有效字节保存在 DLL 中。可以在 1 到 65535 之间设置分频系数，对这两个寄存器的访问分别位于地址 1 和 0，其访问条件取决于 LCR 寄存器中 DLAB 位的值。仅当该位为 1 时，才可以写入和读取 DLM 和 DLL 这两个寄存器；否则，这两个将访问 IER，RBR 和 THR。

  这两个寄存器与预分频寄存器（PSR）会一起选择通信发生的速度。这些波特率是传输字符的波特率，以及将要接收字符的预期波特率。其传输和接收仅会被一个波特率所定义。

  波特率定义为 io_irda_uclk 频率除以 16，再除以 PSR 寄存器的内容，然后再除以除数锁存器（Divisor Latch）的内容。当 DLM 和 DLL 编程设置为"0"时，没有输出时钟。为了节省 FIR 模式下的功耗，建议将 DLL 和 DLM 编程设置为"0"。

表 8-44. DLL寄存器（地址：0x00，当DLAB = 1时）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:0	DLL	RW	0x1	波特率除数锁存器的最低有效字节

表 8-45. DLM寄存器（地址：0x04，当DLAB = 1时）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:0	DLM	RW	0x1	波特率除数锁存器最高有效字节

8.6.5.11. PSR 寄存器

- 地址：0x008（当 DLAB = 1 时）

  名称：PSR

  描述：该 5 位寄存器 （PSR [4:0]）添加了第二个可编程除法系数以获得所需的波特率 （请参见上述对除数锁

存器的说明）。除法系数是该寄存器中保存的值，因此最大系数为 31，最小值为 0，位[7:5]始终为零。此为非标准寄存器 （即，工业标准 16550 UART 中不存在）。输入时钟 **io_irda_uclk** 会被 1 到 31 的整数整除。当 PSR 为 0 时，除数锁存器单元将没有时钟输入。因此将 PSR 设置为 0 时，向 DLL 和 DLM 的编程是无用的。PSR 寄存器的默认值为 1。

仅当 LCR 中的 DLAB 位置 1 时，才能访问该寄存器，否则将访问线路状态寄存器。

表 8-46. PSR寄存器（地址：0x08，当DLAB=1时）

位	名称	类型	默认值	描述
31:5	-	-	-	保留。
4:0	PSR	R/W	0x1	标度值。

8.6.5.12. MDR 寄存器

- 地址：0x020

  名称：MDR

  描述：该寄存器用于选择操作模式。

表 8-47. MDR 寄存器（地址：0x020）

位	名称	类型	默认值	描述
31:7	-	-	-	保留。
6	IR_INV_TX	R/W	0x0	当此位置位时，UART 在发送过程中产生反相的 FIR 或 SIR 脉冲。
5	FIR_INV_RX	R/W	0x0	该位用于支持具有相反极性且可接收信号的光学收发器（高电平而非低电平有效）。当此位置为 “1” 时，在接收器输入信号路径上放置一个变流器。在 SIR 模式下，该位为无用的。
4	DMA_EN	R/W	0x0	保留。
3	FMEND_MD	R/W	0x0	0：应用帧长计数器之方法 1：应用设置结束传输位之方法
2	SIP_BYCPU	R/W	0x0	0：当 TX FIFO 为空白时，处于传输模式的控制器会一直在传输帧的末端发送一个 1.6 μs 的脉冲。 1：1.6 μs 脉冲的传输取决于 ACR[4]。CPU 应该放置一个计时器，并至少每 500 ms 设置一次 ACR[4]位。
1：0	MODE_SEL	R/W	0x0	0x0：UART 0x1：SIR 0x2：FIR

位	名称	类型	默认值	描述
				0x3：保留

表 **8-48. FIFO** 状态触发电平

ACR 代码		**FIFO** 状态触发电平
Bit 6	**Bit 5**	
0	0	1
0	1	4
1	0	7
1	1	8

8.7. SSP

8.7.1. 概述

SSP 控制器是一个同步串行端口接口，允许主机处理器作为主机或从机。通过使用串行协议，可以将各种设备连接到该控制器。SSP 控制器支持德州仪器的同步串行端口（SSP），摩托罗拉的串行外围接口（SPI）。串行数据格式的范围从 4 位到 128 位长度。SSP 控制器可以使用片上 DMA 直接在外部处理器之间传输数据串行设备和系统内存，无需处理器的干预。

8.7.2. 特性

- 兼容 AMBA 2.0 APB 模式。
- 数据传输比特率（bit rate）之最高频率：主模式下为 16 MHz。
- 支持主从模式。
- 可编程串行位时钟极性、相位和频率。
- 可编程的串行位数据序列 （MSB 或 LSB 优先）。
- 发送/接收 FIFO 的可编程阈值中断。
- 独立编程中断之使能/关闭。
- 支持 16 字节的发送和接收数据的 FIFO 深度。

8.7.3. SSP 串行时钟比率及比特率

内部生成的串行时钟频率由控制寄存器 CR1 中的 SCLKDIV 所控制。

内部生成的串行时钟频率由以下公式所定义：

$$f_{SCLK} = \frac{f_{SSPCLK}}{2 \times (SCLKDIV + 1)}$$

当根据 SCLK 频率发送或接收数据时，SSP 的比特率将相等与 SCLK 的频率有关。如果指定了外部生成的串行时钟，则 SCLKDIV 将被忽略，并且所有传输将取决于外部提供的 SCLK。因为 SSP 必须同步外部生成的 SCLK 使用 SSPCLK，外部生成的 SCLK 的频率预计为比 SSPCLK 慢至少六倍。SSPCLK 需要 50%±5 占空比来同步外部生成 SCLK。下表表 1-1 显示了 SCLKDIV 信号的设置以及主模式和从模式下 SSPCLK 和 SCLK 的比率。当外部产生的 SCLK 具有更差的占空比时，SSPCLK/SCLK 的比率必须增加以进行同步。SCLK 的高/低水平应至少是 SSPCLK 的三倍。

表 8-49. SCLKDIV 中 SSPCLK 及 SCLK 的设置及比率

模式选择	SSPCLK/SCLK 的最小比率	SCLKDIV 设置

模式选择	SSPCLK/SCLK 的最小比率	SCLKDIV 设置
主模式	2 （SPI）	通过寄存器设置
从模式	6	无关，不着重

8.7.4. Texas Instrument SSP 帧时序

下图显示了使用 TI SSP 单个传输时的帧格式。

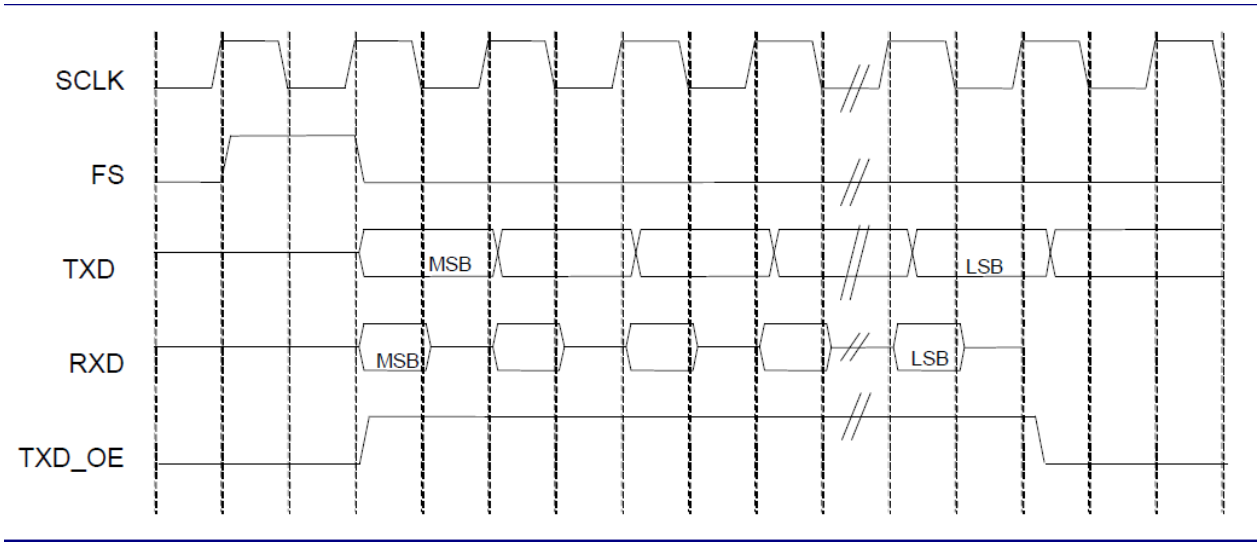


图 8-2. TI SSP 单个传输时的帧格式

当不存在数据传输时，FS 被强制为低。在主模式下，当传输 FIFO 不为空时，传输逻辑将从传输 FIFO 读取数据，并将数据写入移位寄存器。FS 将然后被拉高一个 SCLK 周期。在第一个 SCLK 周期之后，移位寄存器中的数据将移位在 SCLK 的每个上升沿通过 txd_r 引脚输出，直到数据达到串行数据的长度。串行数据的 MSB 将首先发送或接收。如果当前字的 LSB 被传输并且 FIFO 包含数据、帧/同步。为了实现连续传送，FS 在一个 SCLK 周期内将再次被拉高，FIFO 中的数据将被读取到。下图显示了连续传输波形。

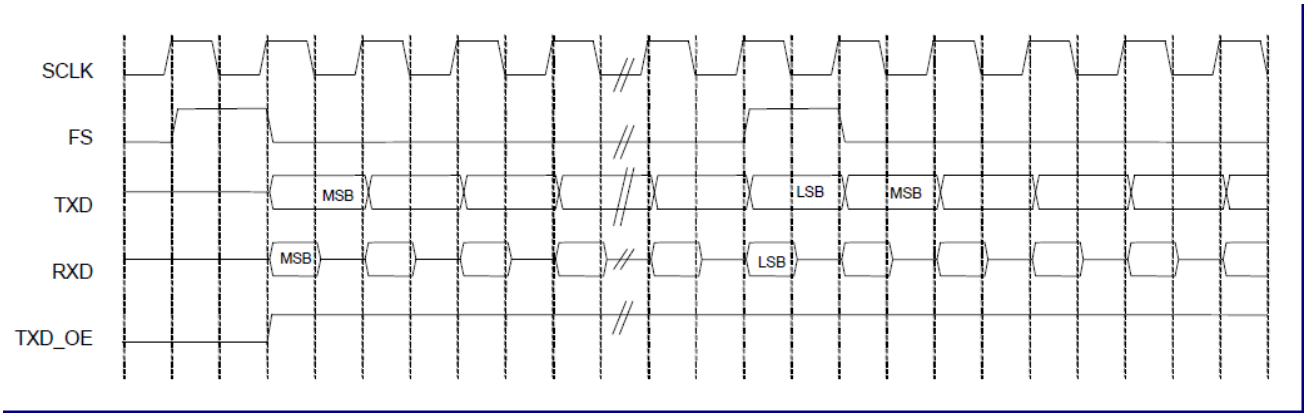


图 8-3. TI SSP 连续传输时的帧格式

在从属模式下，当帧/同步被采样到高，传输逻辑将读取传输 FIFO 中的数据并开始将其移出。如果传输 FIFO 为空，传输逻辑将尝试读取传输 FIFO 的最后一个字以将其移出。在这种情况下，如果启用下溢出中断功能，则会发出传输 FIFO 下溢出中断。接收逻辑将同时开始接收数据。在传输/接收期间，帧/同步在达到数据长度之前将视为不合格。

如果指定了 TI-SSP 帧格式，则无论控制器处于主模式还是从模式，位序列都将从 MSB 到 LSB。控制寄存器 0 中 LSB 的设置将被忽略，SCLKPH 和 SCLKPO 也将被忽略。

8.7.5. Motorola SPI 帧时序

摩托罗拉 SPI 帧格式是一种 4 线串行外设协议。帧格式完全由 SCLKPO 和 SCLKPH 控制。SCLKPO 控制空闲状态下的 SCLK 极性，SCLKPH 确定帧/同步之间的关系。和 SCLK。下图显示了具有 SCLKPO 和 SCLKPH 不同组合的四种 SPI 模式。

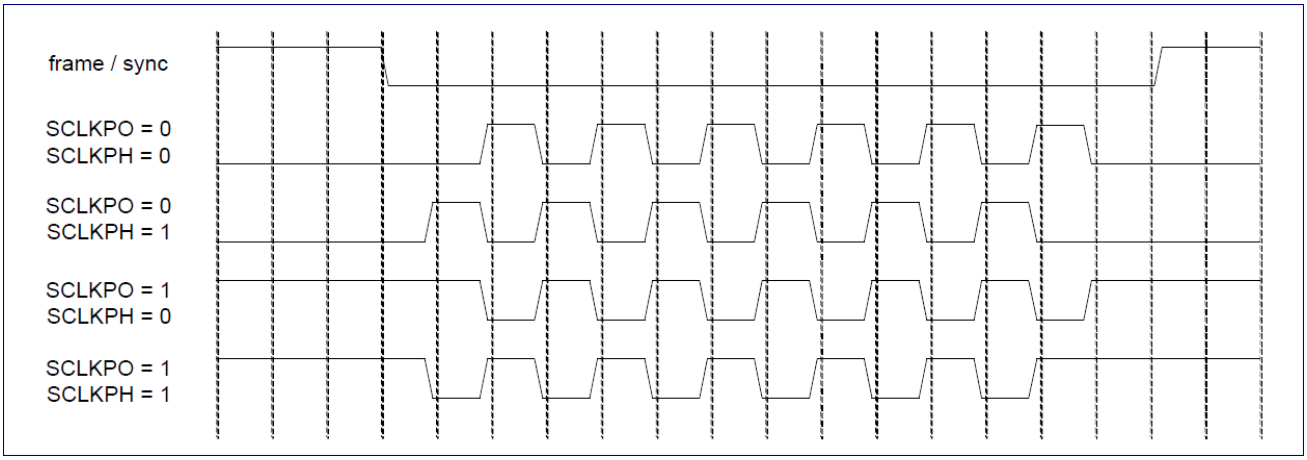


图 8-4. 具有 SCLKPO 和 SCLKPH 不同组合的 SPI 帧格式

如上所示，如果 SCLKPH 被指定为逻辑 1，帧/同步被激活，串行时钟将在 SCLK 周期的一半后开始运行。如果 SCLKPH 被指定为逻辑 0 并且帧/同步被激活，串行时钟将在一个 SCLK 周期后开始运行。如果 SCLKPO 和 SCLKPH 具有相同的值，则传输逻辑将在 SCLK 的下降沿传输数据，而接收逻辑将在 SCLK 的上升沿锁存数据。如果 SCLKPO 和 SCLKPH 的值相反，则发送逻辑将在 SCLK 的上升沿发送数据，接收逻辑将在 SCLK 的下降沿接收数据。

在主模式下，如果传输 FIFO 包含数据且 SSP 已启用，则传输逻辑将读取传输 FIFO 中的数据并在每个传输边缘将其移出（取决于 SCLKPO 和 SCLKPH）。接收逻辑将同时接收数据。发送或接收 LSB 后，帧/同步将保持低电平半个或一个 SCLK 周期，具体取决于 SCLKPH，如果发送 FIFO 中没有数据，则拉高电平。接收到的数据将写入接收 FIFO。如果传输 FIFO 中仍有数据可用，传输逻辑将重新启动以传输数据。

在从模式下，如果帧/同步被激活并且 SCLK 开始运行，则传输逻辑将尝试从传输 FIFO 转移数据，即使传输 FIFO 为空。如果传输 FIFO 已启用，将发出传输 FIFO 下溢出中断。接收逻辑将同时开始接收数据，接收的数据将写入接收 FIFO。

取决于控制寄存器 0 中 LSB 设置的位移位序列来自 MSB 或 LSB，而不管控制器处于主模式还是从模式。控制寄存器 0 中的 SPIFSPO 决定帧/同步信号的极性。

下图显示了单次传输的摩托罗拉 SPI 帧格式。

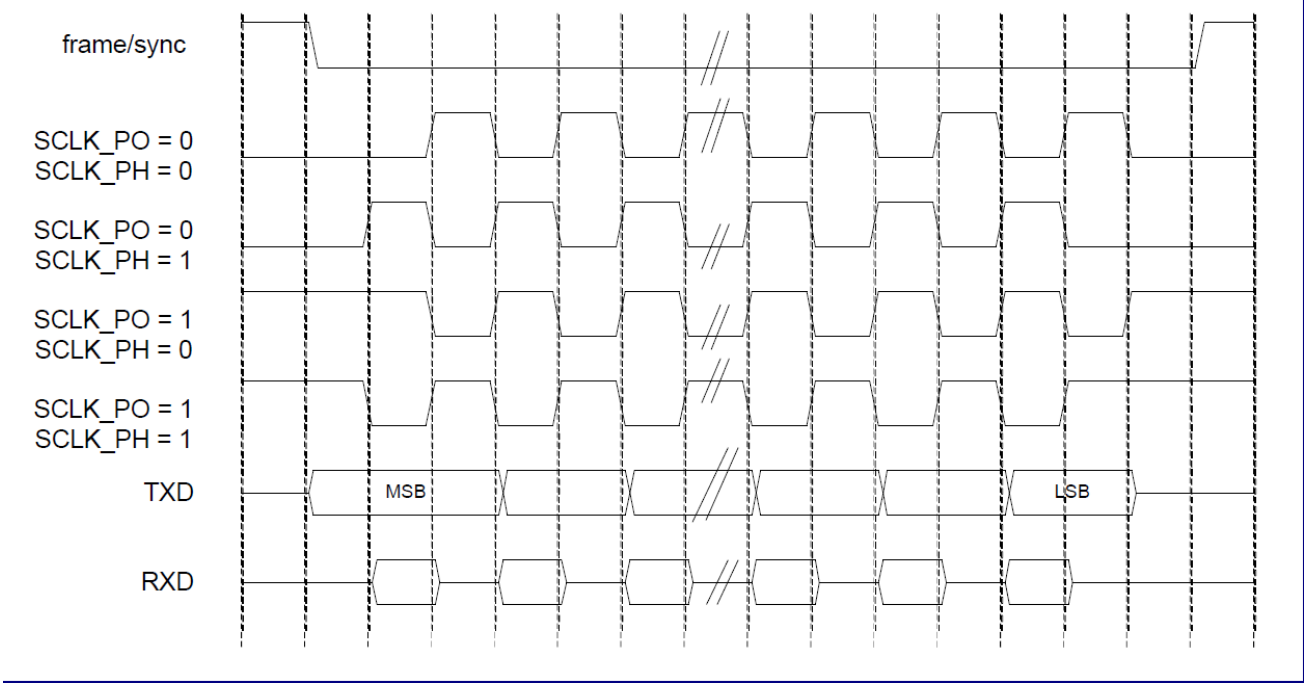


图 8-5. 单次传输的摩托罗拉 SPI 帧格式

对于连续传输，SCLKPH 决定每个连续传输之间的帧/同步。当 SCLKPH 指定为逻辑 0 时，帧/同步。下图所示，必须在每次连续传输之间取消断言和重新断言。当 SCLKPH 被指定为逻辑 1 时，帧/同步。

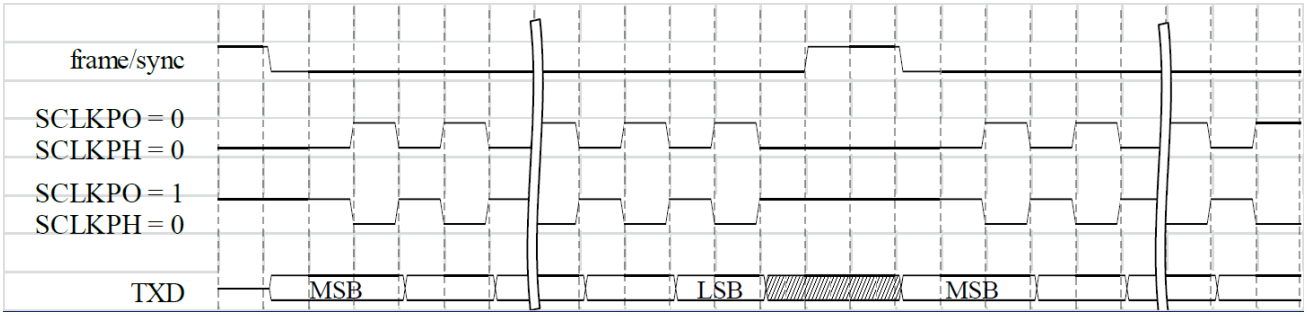


图 8-6.摩托罗拉 SPI 帧格式（SCLKPH=0）的连续传输

如下图所示，在每次连续传输之间可能保持低电平。

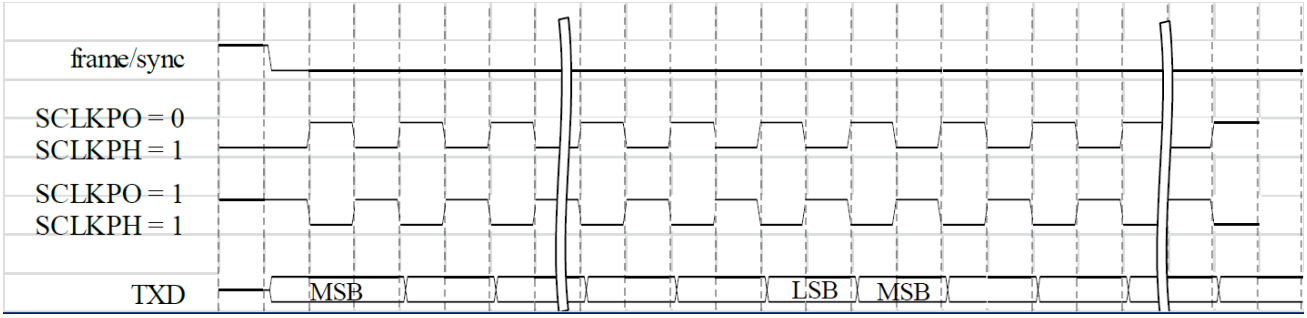


图 8-7.摩托罗拉 SPI 帧格式（SCLKPH=1）的连续传输

在主模式下，SSP 控制寄存器 3（0x1C）的 PCL 控制事务之间的从选择 de-assertion 时间，如图 1-9 所示。反断言时间等于（N+1）xSPI 时钟（SCLK），其中 N 是寄存器 0x1C 中的 PCL 值。对于 N=0 和 SCLKPH=1，如果在启用 TXEN 之前填充 TXFIFO，则将发生连续事务。

在 SPI 帧格式中，SCLK 空闲时将停止切换，帧/同步通常用作芯片选择信号。

在 SPI 主模式和仅 RX 模式下，控制器将保留一个串行数据长度（SDL）大小，以避免 RX 数据丢失。当接收 FIFO 空输入小于/等于 SDL/32（四舍五入）时，控制器将停止接收 RX 数据。此时，帧/同步将不启用。例如，如果 SDL 为 32 位，FIFO 深度为 32，则当接收 FIFO 空标志位等于 1 时，控制器将停止接收 RX 数据。此时，帧/同步将不会启用。

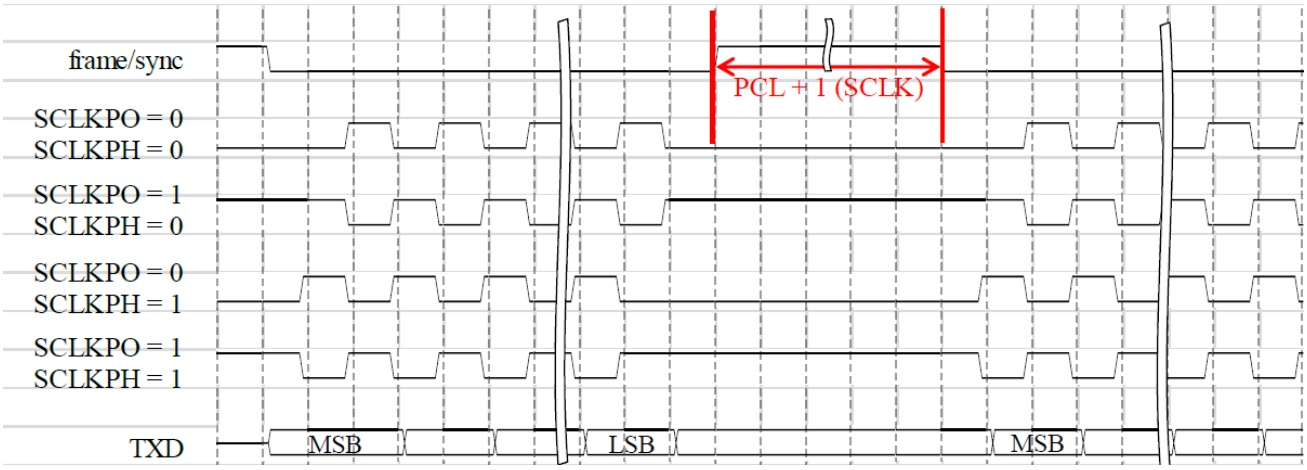


图 8-8. 长度带循环填充的连续传输

8.7.6. National Semiconductor MICROWIRE 帧时序

除了半双工传输外，国家半导体公司的 MICROWIRE 帧格式与摩托罗拉公司的 SPI 帧格式相似。在传输过程中，接收逻辑将停止，直到传输一个完整的字。当传输和接收移交时，插入一个额外的周期。数据将在 SCLK 的下降沿传输，接收的数据将在 SCLK 的上升沿锁存。

由于发送和接收数据长度可能不同，SSP 控制器为发送和接收提供独立的数据长度配置。

SSP 控制器采用国家半导体 MICROWIRE 帧格式时，定义了两个相位。

在主模式中，第一阶段是发送阶段。如果启用 SSP 且传输 FIFO 包含数据，传输逻辑将根据控制寄存器 1 中指定的填充数据长度（PDL）将数据从传输 FIFO 移到从设备。首先，接收逻辑停止。传输完成后，SSP 控制器将检查控制寄存器 1 中指定的串行数据长度（SDL）。如果 SDL 被指定为非零值，SSP 控制器将进入第二阶段接收数据并停止传输逻辑。接收完成后，如果传输 FIFO 中存在数据，SSP 控制器返回第一阶段传输数据。

在从模式中，第一阶段是接收阶段。如果启用 SSP 并且 SCLK 在激活帧/同步时开始运行，则接收逻辑根据控制寄存器 1 中指定的填充数据长度（PDL）开始接收数据。接收完成后，SSP 控制器将检查控制寄存器 1 中指定的串行数据长度（SDL）。如果 SDL 被指定为非零值，SSP 控制器将进入第二阶段传输数据并停止接收逻辑。传输完成后，如果 SCLK 和帧/同步处于活动状态，SSP 控制器将返回第一阶段接收数据。下图显示了国家半导体 MICROWIRE 的传输格式。控制寄存器 0 的 FSPO 设置为"1"。

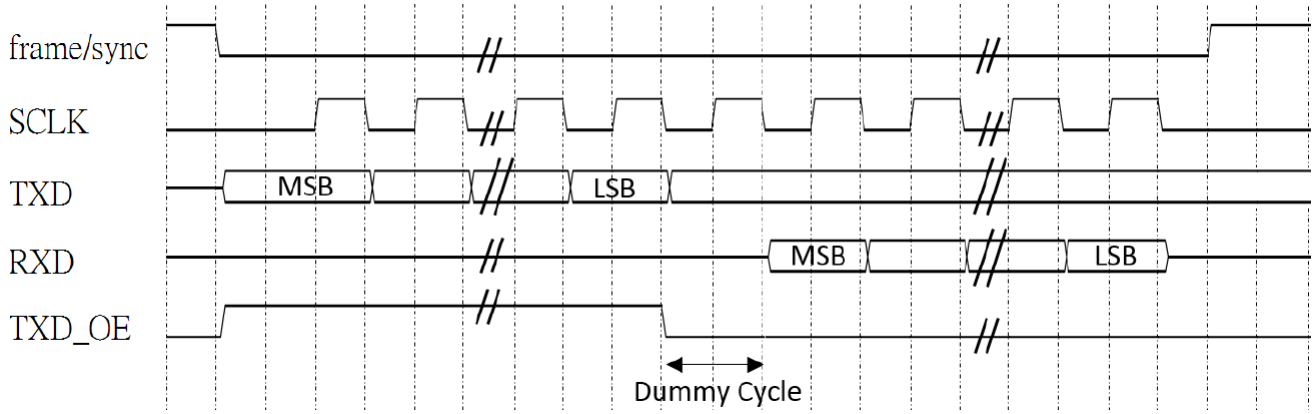


图 8-9. MICROWIRE 单个传输时的帧格式

如果传输 FIFO 中仍有数据，传输逻辑将继续向 TXD 传输数据并从 RXD 接收数据。下图显示了连续传输的帧格式。

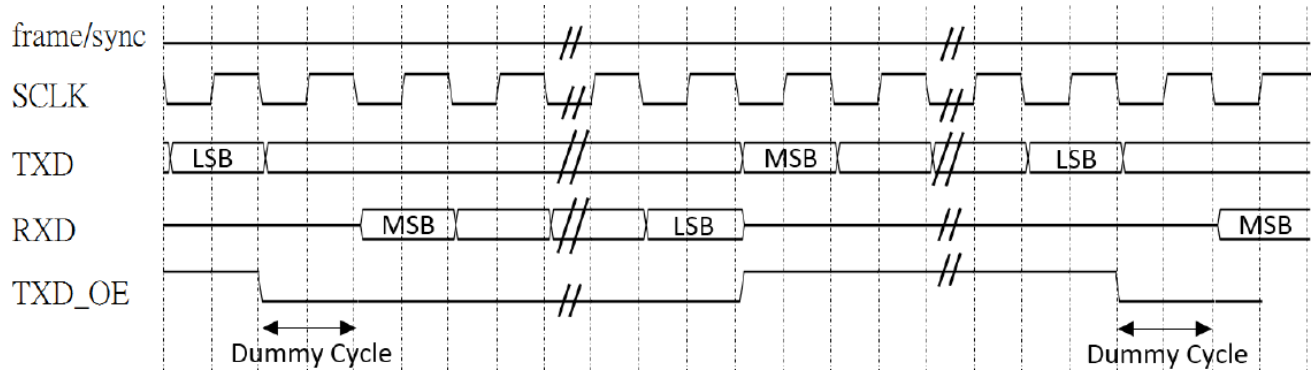


图 8-10. MICROWIRE 连续传输时的帧格式

如果指定了国家半导体 MICROWIRE 帧格式，则定义从 MSB 到 LSB 的位移位序列。控制寄存器 0 中的 LSB 设置将被忽略。SCLKPH 和 SCLKPO 也将被忽略。帧/同步的极性由用户指定。

8.7.7. 寄存器摘要

SSP0 基地址：在 Cortex-M4 AMBA APB 上为 0x54060000

SSP1 基地址：在 Cortex-M4 AMBA APB 上为 0x54070000

SSP2 基地址：Cortex-M4 AMBA APB 上为 0x540D0000

表 8-50. SSP 寄存器摘要

地址	名称	类型	默认值	描述
0x00	CR0	RW	0x0000010C	SSP 控制寄存器 0
0x04	CR1	RW	0x00078000	SSP 控制寄存器 1
0x08	CR2	RW/W1C	0x00000002	SSP 控制寄存器 2
0x0C	Status	R	0x00000002	SSP 状态寄存器
0x10	IntrCR	RW	0x00002100	SSP 中断控制寄存器

地址	名称	类型	默认值	描述
0x14	IntrStatus	R，RC	0x00000008	SSP 中断状态寄存器
0x18	TxRxDR	RW	NA	SSP 发送/接收数据寄存器
0x1C	CR3	RW	0x00000000	SSP 控制寄存器 3

8.7.8. 寄存器描述

8.7.8.1. CR0 寄存器

- 地址：0x000

  名称：CR0

  描述：此寄存器控制帧/同步的生成和性能状态。

表 8-51. CR0 寄存器（地址：0x000）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15	SPIFSPO	R/W	0x0	SPI 模式的帧/同步极性。第 5 位与 FSPO 不同。 如果该位设置为" 1"，则帧/同步将被视为高电平有效。 如果该位设置为" 0"，则帧/同步将被视为低电平有效。 仅当指定了 SPI 帧格式时，此位才有效。
14:12	FFMT	R/W	0x0	帧格式。 该寄存器根据以下编码定义预定义的帧格式： 000：德州仪器同步串行端口（SSP） 001：摩托罗拉串行外围接口（SPI） 010：国家半导体 MICROWIRE 011~111：未定义 如果设置了这些位，SSP 的操作将是不可预测的。
11	FLASH	R/W	0x0	此位表示当前应用程序是 SPI Flash。该位置 1 时 CS 由软件控制。

位	名称	类型	默认值	描述
				仅当指定了 SPI 帧格式时，此位才有效。
10:7	-	-	0x2	保留。
6	LSB	R/W	0x0	位序列指示器。 如果该位置“0”，则数据字的最高有效位（MSB）将首先被发送或接收。 如果该位置“1”，则数据字的最低有效位（LSB）将首先被发送或接收。 仅当指定了 SPI 帧格式时，此位才有效。
5	FSPO	R/W	0x0	帧/同步极性。 如果此位设置为“0”，帧/同步将被视为高电平有效。 如果此位设置为“1”，帧/同步将被视为低电平有效。 此位仅在指定 MWR 帧格式时生效。
4	-	-	-	保留。
3:2	OPM	R/W	0x3	操作模式。 这些位将定义以下操作模式： 00、01：从模式 10、11：主模式
1	SCLKPO	R/W	0x0	如果该位置“0”，则当 SPI 空闲时，SCLK 将保持低电平。 如果该位置“1”，则当 SPI 空闲时，SCLK 将保持高电平。 仅当指定了 SPI 帧格式时，此位才有效。
0	SCLKPH	R/W	0x0	SCLK 相位。 该位定义了 SCLK 和 SPI_CS 之间的关系。 如果此位置“0”，则当激活 SPI_CS 时，SCLK 将在一个 SCLK 周期时间后开始切换。 如果此位置“1”，则当激活 SPI_CS 时，SCLK 将在半个 SCLK 周期后开始运行。 仅当指定了 SPI 帧格式时，此位才有效。

8.7.8.2. CR1 寄存器

- 地址：0x004

  名称：CR1

  描述：此寄存器为定义时钟分频器和数据的传输长度。

表 8-52. CR1 寄存器（地址：0x004）

位	名称	类型	默认值	描述
31:24	PDL	R/W	0x00	填充数据长度 实际数据长度等于这些位加一。 只有在指定了国家半导体 MICROWIRE 帧格式时，这些位才是相关的。
23	-	-	-	保留。
22:16	SDL	R/W	0x07	串行数据长度。 这些位为定义发送/接收数据字的位长。实际数据长度等于这些位再加 1，但最小 SDL 值不应小于 4。 如果指定了 National Semiconductor MICROWIRE，则这些位将定义第二阶段的位长度（在主模式下接收或在从模式下传输）。
15:0	SCLKDIV	R/W	0x8000	串行时钟分频器。 它们不能设置为零，除非指定了 SPI 或 MWR 帧格式。

8.7.8.3. CR2 寄存器

- 地址：0x008

  名称：CR2

  描述：该寄存器为定义 SSP 使能/关闭以及输出发送数据的使能。

表 8-53. CR2 寄存器（地址：0x008）

位	名称	类型	默认值	描述
31:12	-	-	-	保留。
11:10	FSOS	R/W	0x0	帧/同步输出选择 指定 SPI 帧格式时，这两位设置为： 00：输出端口的帧/同步输出，fs_out_r

				01：输出端口的帧/同步输出，fs1_out_r 10\11：保留 只有指定了 SPI 帧格式，这些位才会生效。
9	FS	R/W	0x0	帧同步输出 该位控制 SPI 帧格式的帧/同步输出电平。 如果此位设置为"0"，则 FS 输出将为"低"。 如果此位设置为"1"，则 FS 输出将为"高"。
8	TXEN	R/W	0x0	传输功能使能位。 当指定 SPI 或 MICROWIRE 时，发送和接收功能将独立工作。该位控制传输功能。 1：传输功能已启用。 0：传输功能被禁用。 对于 SPI 和 MICROWIRE 帧格式，此位只能在 SSP 控制器空闲时更改。
7	RXEN	R/W	0x0	接收功能使能位。 当指定 SPI 或 MICROWIRE 帧格式时，发送和接收功能将独立工作。该位控制接收功能。 1：接收功能已启用。 0：接收功能被禁用。 对于 SPI 和 MICROWIRE 帧格式，此位只能在 SSP 控制器空闲时更改。
6	SSPRST	R/W	0x0	SSP 控制器状态机的软件复位。 向该位写 "1" 将复位 SPI 控制状态机。 向该位写 "0" 则为无效动作。 该位的读出值始终为 "0"。
5:4	-	-	-	保留。
3	TXFCLR	R/W	0x0	发送 FIFO 清除位。 此位仅于 "写" 模式。如果该位写 "1"，则发送 FIFO 中的所有数据将被清除。将此位写 "0" 则无效。读取该位的值始终为零。
2	RXFCLR	R/W	0x0	接收 FIFO 清除位。 这是仅于 "写" 模式。如果该位写 "1"，则接收 FIFO 中的所有数据将被清除。将此位写 "0" 无效。读取该位的值始终为零。
1	TXDOE	R/W	0x1	传输数据输出使能位。

				此位仅在指定从属模式时有效。在多从系统中，SSP 主机可以向系统内的所有从机广播消息，同时确保只有一个从机将数据传输到其串行输出线上。在这样的系统中，来自多个从机的 rxd 线路可以捆绑在一起。如果 SSP 从机不支持驱动 txd_r 线路，要在这样的系统中运行，则 TXDOE 可能会被清除。 设置 TXDOE 后，SSP 将在传输数据线上驱动数据。 未设置 TXDOE 时，SSP 将不会在传输数据线上驱动数据。 如果指定了 SPI 帧格式和从模式，则 TXDOE 应设置为高。
0	SSPEN	R/W	0x0	SSP 使能位。 如果此位设置为"1"，SSP 控制器将在可能的情况下开始发送/接收数据。 如果此位设置为"0"，串行数据将停止。 请注意，当 SSPEN 位在发送或接收数据操作中被清除时，SSP 将在当前字被发送或接收后停止发送或接收。

即使正常传输或接收正在进行，将"1"写入 SSPRST 将重置 SSP 状态机。通常，软件通过将 SSPEN 设置为"0"停止 SSP 控制器。对于 SSP 控制器的从模式，串行时钟将由外部主机控制，在某些情况下将停止动作，进而导致 SSP 状态机停止。将"1"写入该位会使 SSP 控制器返回空闲状态。

8.7.8.4. Status 寄存器

- 地址：0x00C

  名称：Status

  描述：该寄存器提供 SSP 内部状态，以便用户采取相应的操作。它显示了中断启用之前的原始状态。

表 8-54. Status 寄存器（地址：0x00C）

位	名称	类型	默认值	描述
31:18	-	-	-	保留。
17:12	TFVE	R	0x0	发送 FIFO 非空。 这些位表示发送 FIFO 中有等待发送的数据。
11:10	-	-	-	保留。
9:4	RFVE	R	0x0	接收 FIFO 非空。 这些位指示接收 FIFO 中有等待 DMA 或主机处理器读取的数据。
3	-	-	-	保留。
2	BUSY	R	0x0	BUSY 指示。

位	名称	类型	默认值	描述
				如果这个位被读到'1'，它将表明 SSP 正在发送和/或接收数据。 如果这个位被读到'0'，它将表明 SSP 是空闲的或禁用的。
1	TFNF	R	0x1	发送 FIFO 未满。 当传输 FIFO 可用于 DMA 或主机处理器写入时，此位将被设置为' 1 '。 当 FIFO 完全满时，此位将被清除为' 0 '。
0	RFF	R	0x0	接收 FIFO 满。 当接收 FIFO 满时，此位将被设置为' 1 '。 当 DMA 控制器或主机处理器从接收的 FIFO 读取数据时，此位将被清除为' 0 '。

8.7.8.5. IntrCR 寄存器

- 地址：0x010

  名称：IntrCR

  描述：这个寄存器控制 SSP 的中断生成。接收 FIFO 阈值取决于 RFTHOD 和 RFTHOD_UNIT。如果 RFTHOD_UNIT = 1，实际接收的 FIFO 门限将等于 RFTHOD 加 1。如果 RFTHOD_UNIT = 0，实际接收的 FIFO 门限将等于 RFTHOD。

表 8-55. IntrCR 寄存器（地址：0x010）

位	名称	类型	默认值	描述
31:18	-	-	-	保留。
17	RFTHOD_UNIT	R/W	0x0	接收 FIFO 阈值。 如果该位设置为 1，则 RxFIFO 阈值= RFTHOD + 1。 如果该位设置为 0，则 RxFIFO 阈值= RFTHOD。
16:12	TFTHOD	R/W	0x02	发送 FIFO 阈值。 如果发送 FIFO 中的有效数据等于或小于实际阈值，中断将会生效。如果该位设置为' 0 '，中断将被禁用。
11:7	RFTHOD	R/W	0x02	接收 FIFO 阈值。 如果在接收 FIFO 中的有效数据等于或大于实际

				阈值，中断将会生效。如果该位设置为' 0 '，中断将被禁用。
6	-	-	-	保留。
5	TFDMAEN	R/W	0x0	发送 DMA 请求使能。 如果该位设置为"1"，则当达到传输 FIFO 阈值时，将发出 DMA 请求。 如果此位设置为"0"，则不会发出 DMA 请求。
4	RFDMAEN	R/W	0x0	接收 DMA 请求使能。 如果此位设置为"1"，则在达到接收 FIFO 阈值时将发出 DMA 请求。 如果此位设置为"0"，则不会发出 DMA 请求。
3	TFTHIEN	R/W	0x0	发送 FIFO 阈值中断使能。 如果该位设置为'1'，当发送 FIFO 中的有效条目小于或等于阈值时，中断将被触发。
2	RFTHIEN	R/W	0x0	接收 FIFO 阈值中断使能。 如果该位设置为 1，当接收 FIFO 中的有效项大于或等于阈值时，中断将被触发。
1	TFURIEN	R/W	0x0	发送 FIFO 下溢出中断使能。 如果该位设置为'1'，发送 FIFO 下溢出将导致 SSP 触发中断。 如果该位设置为'0'，中断将被屏蔽，即使当发送 FIFO 下溢出发生。
0	RFORIEN	R/W	0x0	接收 FIFO 溢出中断使能。 如果该位设置为'1'，接收的 FIFO 溢出将导致 SSP 触发中断。 如果该位设置为'0'，中断将被屏蔽，即使当接收 FIFO 溢出发生。

8.7.8.6. IntrStatus 寄存器

- 地址：0x014

  名字：IntrStatus

  描述：这个寄存器显示当前的中断状态。

表 8-56. IntrStatus 寄存器（地址：0x014）

位	名称	类型	默认值	描述
31:4	-	-	-	保留。
3	TFTHI	RO	0x1	发送 FIFO 阈值状态指示。 如果发送 FIFO 等于或小于阈值，该位将被设置为' 1 '。 如果传输 FIFO 中的有效项大于 TFTHOD，则该位将自动清除。
2	RFTHI	RO	0x0	接收 FIFO 阈值状态指示。 如果接收的 FIFO 等于或大于阈值，这个位将被设置为' 1 '。 当上述条件被删除时，此位将被清除。
1	TFURI	RC	0x0	发送 FIFO 下溢出状态指示。 如果传输逻辑试图从空的传输 FIFO 中检索数据，此位将被设置为' 1 '。 在用户读取该位之前，该位永远不会被清除为' 0 '。
0	RFORI	RC	0x0	接收 FIFO 溢出状态指示。 如果接收逻辑尝试接收数据时，接收 FIFO 是满的，这个位将被设置为' 1 '。 在用户读取该位之前，该位永远不会被清除为' 0 '。

8.7.8.7. TxRxDR 寄存器

- 地址：0x018

  名字：TxRxDR

  描述：SSP 控制器分为发送 FIFO 和接收 FIFO，位宽是 32 位，可以单独配置。传输数据和接收数据占用相同的地址空间。写操作将数据写入发送 FIFO，读操作从接收 FIFO 读取数据。如果一个串行数据长度的大小小于 32 位，数据将在接收或传输期间自动正确对齐。当串行数据长度大于 32 位时，在收发数据时，会按照 32 位数据宽度从 LSB 处理到 MSB。

表 8-57. TxRxDR 寄存器（地址：0x018）

位	名称	类型	默认值	描述
31:0	TxRxDR	R/W	0x00000000	发送和接收数据

8.7.8.8. CR3 寄存器

- 地址：0x01C

  名字：CR3

  描述：该寄存器定义每次连续传输之间的 SPI 等待时间。

  - 对于 PCL> 0，所有 SPI 操作模式将在事务之间使用（PCL + 1）SPI 时钟生成离散事务
  - 对于 PCL = 0 和 SCLKPH = 1（SCLKPH 是 SSP 控制寄存器 0 的位[0]），如果在使能 TXEN 位（SSP 控制寄存器的位[8]）之前填充 TXFIFO，将发生连续事务。
  - 对于 PCL = 0 和 SCLKPH = 0，将发生多个离散事务。

表 8-58. CR3 寄存器（地址：0x01C）

位	名称	类型	默认值	描述
31:10	-	-	-	保留。
9:0	PCL	R/W	0x000	填充周期长度。 在主模式下，SPI 将在每次连续传输之间等待 (PCL + 1) SCLK 周期。

8.8. I²C

8.8.1. 概述

I²C 总线接口控制器允许主处理器作为 I²C 总线中的主机或从机。数据通过缓冲接口与 I²C 总线进行发送和接收。I²C（内部集成电路）总线接口处理单片机和串行 I²C 总线。它提供多主控功能，并控制所有 I²C 总线特定的排序、协议、仲裁和定时。支持标准模式（Sm），快速模式（Fm）和快速模式增强（Fm+）。

8.8.2. 特性

- 符合 AMBA 2.0 APB。
- 通过可编程时钟分频寄存器支持标准、快速和快速+模式（这 3 种模式称为 F/S 模式）。
- 通过可编程时钟分割寄存器支持 HS 模式。
- 支持 7/10 位寻址模式。
- 支持普通呼叫和启动字节。
- 支持毛刺抑制。
- 支持可编程从机地址。
- 支持主 TX、主 RX、从 TX 和从 RX 模式。
- 支持多主模式。

8.8.3. 功能框图

I²C 主控制模块包括 I²C 主控制模块和毛刺抑制模块。请参阅以下各小节以了解每个模块的详细信息。

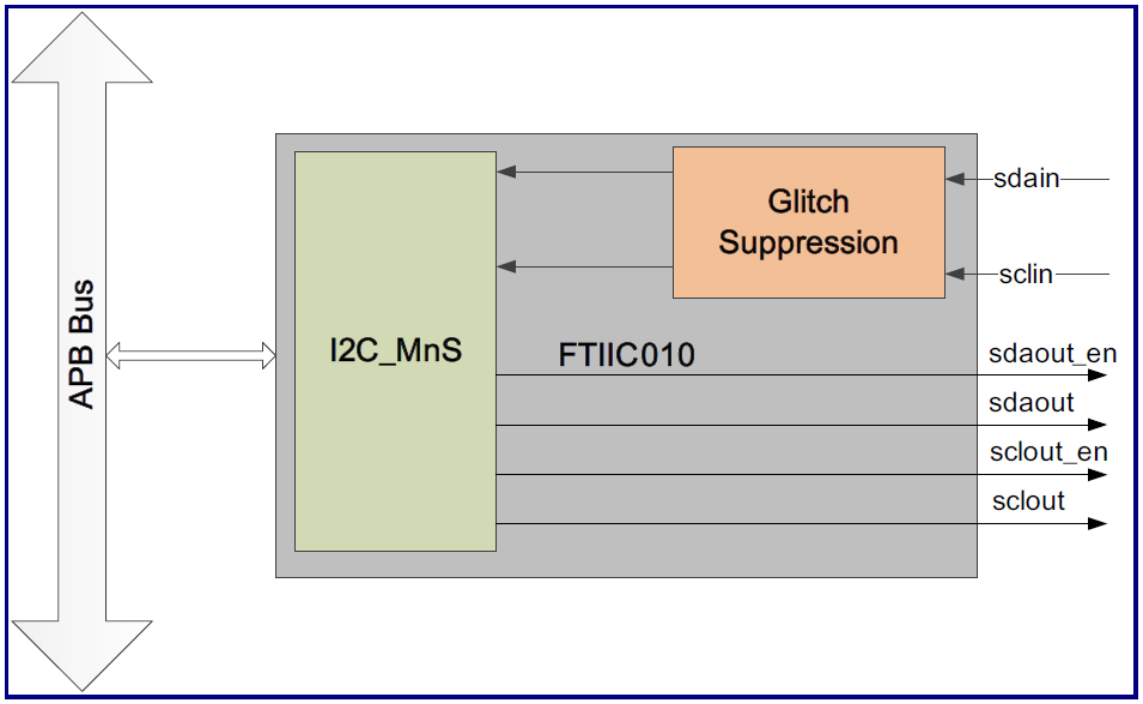


图 8-11. I²C 功能框图

8.8.3.1. 毛刺抑制电路

毛刺抑制电路用于消除 I²C 总线上的毛刺。根据 GSR 内部总线时钟周期抑制毛刺，其中 GSR 是寄存器 TGSR 偏移量 0x14 处的位[13:10]。

8.8.3.2. I²C_MnS 电路

I²C_MnS 电路用于从毛刺抑制电路中检测 sclin 和 sdain，生成 sclout、sclout_en 和 sdaout。此模块还包含寄存器文件并决定总线的状态。

8.8.4. 寄存器摘要

基地址：在 Cortex-M4 AMBA APB 上为 0x54040000

表 8-59. I²C 控制器的寄存器总表

位	名称	类型	默认值	描述
0x000	CR	R/W	0x00040000	I²C 控制寄存器 （CR）
0x004	SR	R/W1C R	0x00000000	I²C 状态寄存器 （SR）
0x008	CDR	R/W	0x00000008	I²C 时钟分频（CDR）

0x00C	DR	R/W	0x00000000	I²C 数据寄存器（DR）
0x010	SAR	R/W	0x00000000	I²C 地址寄存器（AR）
0x014	TGSR	R/W	0x00000401	I²C setup/hold 时间和毛刺抑制设置寄存器（TGSR）
0x018	BMR	R	0x00000003	I²C 时钟监测寄存器（BMR）
0x01C	BSTMR	R/W	0x00200101	I²C 块模式寄存器（BSTMR）

8.8.5. 寄存器描述

8.8.5.1. CR 寄存器

- 地址：0x54040000（Cortex-M4）

  名称：CR

  描述：主处理器使用 I²C 控制寄存器（CR）控制 I2C 接口控制器，用于从 I²C 总线发送和接收数据。将 MST_EN 设置为"0"以进入从机模式。启动和停止位只能在主机模式下设置。

表 8-60. CR 寄存器（地址：0x54040000）

位	名称	类型	默认值	描述
31:26	-	-	-	保留。
25:24	FLOW_CTRL_EN	R/W	0x0	0：保留。 1：主机 TX 块模式．在主模式下发出写块处理。 2：主机 RX 块模式．在主模式下发出读块处理。
23:22	-	-	-	保留。
21	SBI_EN	R/W	0x0	如果设置为 1，当 I²C 控制器在从模式下检测到 START 字节时，该位将使能 I²C 控制器中断主机处理器。
20	HSI_EN	R/W	0x0	如果设置为 1，当 I²C 控制器检测到从模式下的 HS-mode 代码时，该位将使能 I2C 控制器中断主机处理器。
19	HS_mode	R/W	0x0	该寄存器用于在 HS 模式和 F/S 模式之间切换。当用户将该寄存器设置为 1 时，I²C 控制器将使用 CDR[31:20]（0x08）生成相应的 HS 模式 SCL。
18	Arb_off	R/W	0x0	该寄存器用于忽略单个 I2C 主环境中的仲裁丢失检

				测。 尽管如此，SR（0x14）和 ISI2C 的位 10 将断言（如果 CR[13]为 1），以通知 CPU I^2C 总线已检测到仲裁丢失，但 IIC0 仍将完成整个 I^2C 协议。
17	Test_bit	R/W	0x0	特殊测试模式。 此位必须设置为"0"。
16	SDA_LOW	R/W	0x0	如果设置为 1，则 SDAout 将绑定到"0"。 对于正常情况，建议将该位设置为"0"。
15	SCL_LOW	R/W	0x0	如果设置为 1，则 SCLout 将绑定到"0"。 对于正常情况，建议将该位设置为"0"。
14	STARTI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器在 I^2C 总线上检测到启动条件时，该位将使能 I^2C 控制器中断主机处理器。
13	ALI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器在主模式下失去仲裁时，该位将使能 I^2C 控制器中断主机处理器。
12	SAMI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器检测到与 SAR 寄存器或通用调用地址匹配的从机地址（当设置 GC_EN 为 1 时），该位将使能 I^2C 控制器中断主机处理器。
11	STOPI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器在 I^2C 总线上检测到结束条件时，该位将使能 I^2C 控制器中断主机处理器。
10	NACKI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器检测到来自 I^2C 总线的 non-ACK 响应时，该位将使能 I^2C 控制器中断主机处理器。
9	TDI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器 DR 寄存器已从 I^2C 总线发送/接收一个数据字节，或在主 RX/TX 块模式下发送/接收到总数据计数（寄存器偏移量 0x1C 的位[13:8]），则该位将使能 I^2C 控制器中断主处理器。
8	BSTTHODI_EN	R/W	0x0	如果设置为 1，当 I^2C 控制器读取数据达到主 RX 块模式下的阈值（寄存器偏移量 0x1C 的位[5:0]）时，该位将使 I^2C 控制器中断主处理器。
7	TB_EN	R/W	0x0	当设置传输字节启用（TB_EN）时，I^2C 控制器准备接收或发送一个字节。否则，I^2C 控制器将通过在 I^2C 总线中拉低 SCLout 来插入等待状态。 请注意，当设备接收或发送一个字节或检测到停止/启动条件时，TB_EN 将被自动清除。
6	ACK/NACK	R/W	0x0	当 I^2C 控制器处于主接收或从接收模式时，I^2C 控制器发送的确认信号。如果 I^2C 控制器检测到 I^2C 总线上的停止条件，该位将自动清除为 0。 0：ACK

				1：NACK
5	STOP	R/W	0x0	当 I²C 处于主模式时，I²C 控制器在 I²C 总线上传输下一个数据字节后启动停止条件。
4	START	R/W	0x0	I²C 控制器在 I²C 总线空闲时启动一个启动条件，或者在主模式下传输 I²C 总线上的下一个数据字节后启动一个重复启动条件。
3	GC_EN	R/W	0x0	使能 I²C 控制器作为从机响应一般呼叫消息。
2	MST_EN	R/W	0x0	在主机模式工作时，使能 I²C 控制器时钟输出。
1	I2C_EN	R/W	0x0	使能 I²C 总线接口控制器。
0	I2C_RST	R/W	0x0	复位 I²C 总线接口控制器。此位将在两个 PCLK 时钟后自动清除。

8.8.5.2. SR 寄存器

- 地址：0x54040004（Cortex-M4）

  总线：Cortex-M4 APB

  名称：SR

  描述：I²C 中断通过 ISI2C 引脚发出信号。当 I²C 控制寄存器（CR）中的中断使能位和相应的状态寄存器都被设置时，将设置 ISI2C 引脚。当 I²C 控制器中断被断言时，软件读取 SR 位以检查 I²C 控制器的状态，并写入 1 以清除 SR。

  SR 用于通过读取寄存器状态清除以下中断：

  - 数据传输（TX/RX）完成。
  - 检测到从机地址。
  - 检测到总线 NACK。
  - 检测到启动条件。
  - 检测到停止条件。
  - 检测到仲裁丢失。
  - 检测到一般呼叫。
  - 检测到 START 字节。

表 8-61. SR 寄存器（地址：0x54040004）

位	名称	类型	默认值	描述

31:24	-	-	-	保留。
23	SBS	R/W1C	-	当 I²C 控制器检测到 I²C 总线上的 START 字节时，此信号置为 1。
22	HSS	R/W1C	-	当 I²C 控制器检测到 I²C 总线上的 HS 模式代码时，此信号置为 1。
21:12	-	-	-	保留。
11	START	R/W1C	0x0	当 I²C 控制器在 I²C 总线上检测到启动条件时，此信号置为 1。
10	AL	R/W1C	0x0	当 I²C 控制器在主模式下失去仲裁时，此信号置为 1。
9	GC	R/W1C	0x0	当 I²C 控制器接收到与一般呼叫地址相匹配的从机地址，并且 I²C 控制器处于从机模式时，此信号置为 1。
8	SAM	R/W1C	0x0	当 I²C 控制器处于从属模式时，该信号将在 I2C 控制器接收到与地址寄存器（AR）中的地址相匹配的从属地址时置为 1，并且它应该在 GC 命中时断言。
7	STOP	R/W1C	0x0	当 I²C 控制器检测到 I²C 总线中的停止条件时，此信号置为 1。
6	NACK	R/W1C	0x0	当 I²C 控制器在主模式下传输一个字节的数据后，I²C 控制器检测到来自从设备的 non-ACK 响应时，此信号置为 1。
5	TD	R/W1C	0x0	当数据完成传输（TX/RX）一个字节，或在主 RX/TX 块模式下完成总数据计数（寄存器偏移量 0x1C 的位[13:8]）时，此信号置为 1。
4	BSTTHOD_SR	R/W1C	0x0	当 I²C 控制器读取数据达到主 RX 块模式下的阈值（寄存器偏移量 0x1C 的位[5:0]）时，此信号置为 1。
3	BB	R	0x0	此信号将在 I²C 总线忙时置为 1，但 I²C 控制器不参与事务。
2	I2CB	R	0x0	此信号将在 I²C 控制器繁忙时置为 1，即在 START 和 STOP 之间的时间段内。
1	-	-	-	保留。
0	RW	R	0x0	当 I²C 控制器在 RX 或 TX 模式下工作时，此信号将置为 1。 0：主机处于 TX 模式，或从机处于 RX 模式 1：主机处于 RX 模式，或从机处于 TX 模式

8.8.5.3. CDR 寄存器

- 地址：0x54040008（Cortex-M4）

  总线：Cortex-M4 APB

  名称：CDR

  描述：I^2C 时钟分频寄存器（CDR）定义用于生成 I^2C SCL 时钟的分频值。此寄存器与内部 32 位计数器一起使用。当控制寄存器中的 MST_EN 位被设置为 1 时，该计数器从可编程值递减到零，然后重新加载可编程值并再次递减。每次计数器达到零时，SCL 从高到低或从高到低变化，具体取决于当前状态。该寄存器可配置为选择 I^2C 总线中所需的传输速度。

  请注意，由于控制器支持高达 3.4 MHz 的 I^2C 总线速度，最小 PCLK 必须大于 68 MHz。

表 8-62. CDR 寄存器（地址：0x54040008）

位	名称	类型	默认值	描述
31:28	DUTY_OFFSET	R/W	0x0	这个偏移量可以改变 I^2C 时钟占空比。用户必须配置此寄存器，以适应 I^2C 总线规范中声明的 SCL 时钟的低周期。 High period = ((COUNT) - DUTY_OFFSET) + GSR + 3) * PCLK period Low period = ((COUNT) + DUTY_OFFSET) + 1) * PCLK period 请注意：COUNT 必须大于 DUTY_OFFSET，((COUNT) - DUTY_OFFSET) 必须大于等于 2.
27:20	COUNTH	R/W	0x08	HS 模式的计数器阈值，用于从内部总线时钟 PCLK 生成 I^2C 时钟（SCLout）。PCLK 与 I^2C 总线时钟（SCLout）的关系如下式所示： SCLout (Hz) = PCLK (Hz)/(2 * COUNTH + GSR + 4) 请注意 COUNTH 的最小值必须等于或大于 GSR+TSR+4，其中 GSR 和 TSR 来自寄存器偏移量 0x14。 例如：如果 PCLK 为 60 MHz，COUNTH 为 7，GSR 为 2，则 SCLout 为 3.0 MHz。
19:0	COUNT	R/W	0x00008	F/S 模式的计数器阈值，用于从内部总线时钟 PCLK 生成 I^2C 时钟（SCLout）。PCLK 与 I^2C 总线时钟（SCLout）的关系如下式所示： SCLout (Hz) = PCLK (Hz)/(2 * COUNT + GSR + 4)

				请注意 COUN 的最小值必须等于或大于 GSR + TSR + 4，其中 GSR 和 TSR 来自寄存器偏移量 0x14。 例如： 如果 PCLK 为 20 MHz，则计数为 7，GSR 为 2，然后 SCLout 为 1 MHz。

8.8.5.4. DR 寄存器

- 地址：0x5404000C （Cortex-M4）

  总线：Cortex-M4 APB

  名称：DR

  描述：主机处理器使用 I^2C 数据寄存器（DR）从 I^2C 总线发送和接收数据。DR 由主机处理器或 I^2C 控制器移位寄存器（SHR）访问。来自 I^2C 总线接口的数据在收到完整的数据字节后由 DR 寄存器接收。从 I^2C 总线接口输出的数据由主机处理器写入 DR 寄存器并发送到串行总线。

  当 I^2C 控制器处于传输模式时，主机处理器通过 APB 总线将数据写入 DR 寄存器。当启动主转换或发出数据传输中断信号时，会发生这种情况。当控制寄存器中的传输字节启用（TB_EN）被设置时，数据从 DR 寄存器移动到 SHR 寄存器。当 I^2C 总线上传输了一个字节的数据时，就会发出数据传输中断信号。如果主机处理器在下一个字节包之前没有写入 DR，I^2C 控制器将插入一个等待状态，直到主机处理器写入 DR 并设置传输字节启用位（TB_EN）.

  当 I^2C 控制器处于接收模式时，处理器通过 APB 总线读取 DR 寄存器数据，当 DR 接收到的中断发出信号时，就会发生这种情况。当 I^2C 控制器接收到一个新的数据字节时，它将自动清除 TB_EN 并在 I^2C 总线上发出 ACK/NACK。在发出 ACK/NACK 之后，I^2C 控制器将向处理器插入 TD 中断。用户必须为 I^2C 总线上的下一个字节传输再次设置 TB_EN 位。

表 8-63. DR 寄存器（地址：0x5404000C）

位	名称	类型	默认值	描述
31:8	-	-	-	保留
7:0	DR	R/W	0x00	用于 I^2C 总线数据传输和接收的缓存。

8.8.5.5. AR 寄存器

- 地址：0x54040010 （Cortex-M4）

  总线：Cortex-M4 APB

  名称：AR

  描述：I^2C 地址寄存器定义 10 位 I^2C 控制器，该控制器在主 RX/TX 块模式下传输从 ID、1/2 索引、10 位寻址模式，定义 I^2C 控制器在从模式下运行时处理器响应的从地址，或定义在主块模式下传输的从机地址。主处理器在启

用 I²C 操作之前写入该寄存器。该寄存器完全可编程设置为一个值，而不是固定到系统中预先存在的从机外围地址。

表 8-64. AR 寄存器（地址：0x54040010）

位	名称	类型	默认值	描述
31:24	MEM_IDX2	R/W	0x00	主 RX/TX 块模式的第二个传输索引
23:16	MEM_IDX	R/W	0x00	主 RX/TX 块模式的第一个传输索引
15:14	-	-	-	保留。
13	M2BIDX_EN	R/W	0x0	设置为 1 以传输第二个索引字节
12	ADDR10EN	R/W	0x0	设置为 1 以启用 10 位寻址模式
11:10	-	-	-	保留。
9:7	ADDR2	R/W	0x0	最有效的 3 位地址，I²C 控制器在主 RX/TX 块模式下传输到该地址，或当 I²C 在从模式下工作时，如果 ADDR10EN='1'，则给出响应。 当 ADDR10EN='0'时，I²C 控制器忽略这三位。
6:0	ADDR	R/W	0x00	在 7 位寻址模式（ADDR10EN='0'）中，7 位地址用于在主块模式下传输的 I²C 控制器，或者当 I²C 在从模式下运行时，I²C 控制器给出响应。

8.8.5.6. TGSR 寄存器

- 地址：0x54040014（Cortex-M4）

  总线：Cortex-M4 APB

  名称：TGSR

  描述：I²C TGSR[13:10]定义了 I²C 总线接口具有内置毛刺抑制逻辑时 PCLK 时钟周期的值。根据 TGSR[13:10]*PCLK 时钟周期抑制毛刺。例如，使用 66MHz（15ns 周期）的 PCLK 时钟，并且 TGSR[13:10]=4'b100，则抑制 60ns 或更小的毛刺。使用 40MHz（25ns 周期）时钟，并且 TGSR[13:10]=4'b10，50ns 或更小的毛刺被抑制。这在 50ns 毛刺抑制规格范围内。唯一的限制是 COUNT 或 COUNTH>=4+GSR+TSR。请注意，当 I²C 为从机时，等待状态后 SCL 将被释放，然后计数 TSR+1 周期。

表 8-65. TGSR 寄存器（地址：0x54040014）

位	名称	类型	默认值	描述
31:14	-	-	-	保留。
13:10	GSR	R/W	0x01	当 I²C 总线接口具有内置的毛刺抑制逻辑时，这些位定义了 PCLK 时钟周期的值。根据“GSR*PCLK”时钟周期抑制毛刺。

9:0	TSR	R/W	0x001	这些位定义在驱动数据或确认时 SCL 和 SDA 之间的 PCLK 时钟周期的延迟值。 实际延迟值为 GSR+TSR+4。细节请参考 图 8-12 。 请注意，TSR 不能设置为"0"。

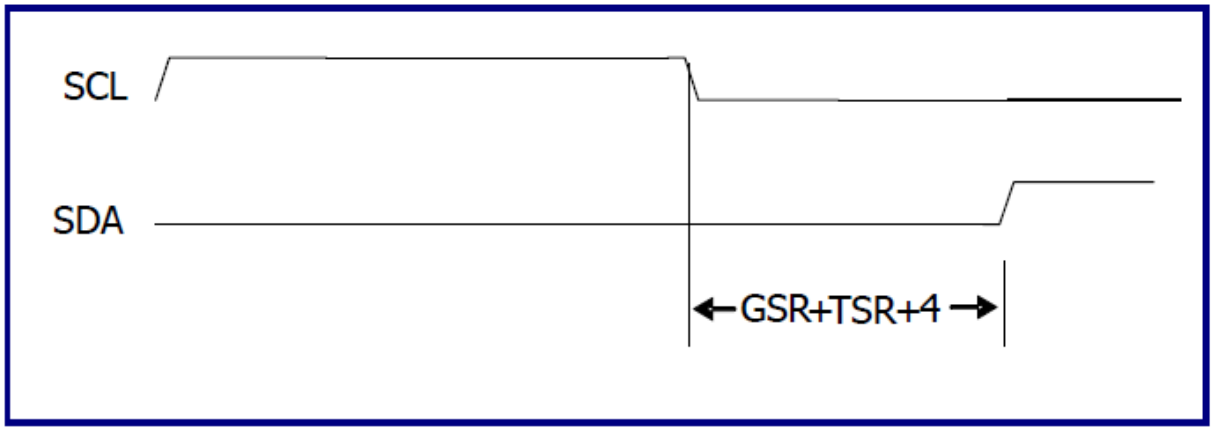


图 8-12. TGSR，SCL，and SDA 之间的关系

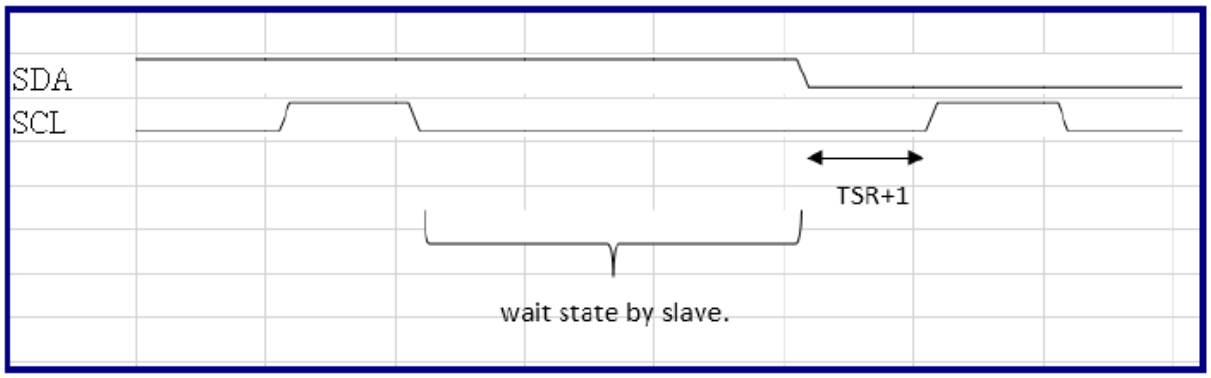


图 8-13. 等待状态后从机释放 SCL 时，TSR、SCL、SDA 的关系

8.8.5.7. BMR 寄存器

- 地址：0x54040018（Cortex-M4）

  总线：Cortex-M4 APB

  名称：BMR

  描述：I^2C 总线监测寄存器

表 8-66. BMR 寄存器（地址：0x54040018）

位	名称	类型	默认值	描述
31:2	-	-	-	保留。

位	名称	类型	默认值	描述
1	SCLin	R	0x1	该位连续反映 SCLin 引脚的值。
0	SDAin	R	0x1	该位连续反映 SDAin 引脚的值。

8.8.5.8. BSTMR 寄存器

- 地址：0x5404001C (Cortex-M4）

  总线：Cortex-M4 APB

  名称：BSTMR

  描述：I^2C 总线模式寄存器

表 8-67. BSTMR 寄存器（地址：0x5404001C）

位	名称	类型	默认值	描述
31:19	-	-	-	保留。
18:16	BUFHW	R	5	缓存深度 1：2 字节 2：4 字节 3：8 字节 4：16 字节 5：32 字节
15:14	-	-	-	保留。
13:8	BSTTDC	R/W	0x01	这些位是整块数据计数，必须小于或等于 BUFHW。请不要将它们设置为 0。
7:6	-	-	-	保留。
5:0	BSTTHOD	R/W	0x01	这些位必须小于或等于 BSTTDC。请注意，BSTTHOD 仅在主 RX 块模式下有效。请不要将它们设置为 0。

8.9. TMR

8.9.1. 概述

每个定时器都使用内部系统时钟（APB 的 PCLK）或外部时钟（ext_clk），并支持 PWM（脉冲宽度调制）功能，以生成用于电机控制或功率电平控制的 PWM 信号。

为了实现同步，建议时钟 PCLK 和 ext_clk 必须保持以下关系：ext_clk 循环时间/ PCLK 循环时间> 3。

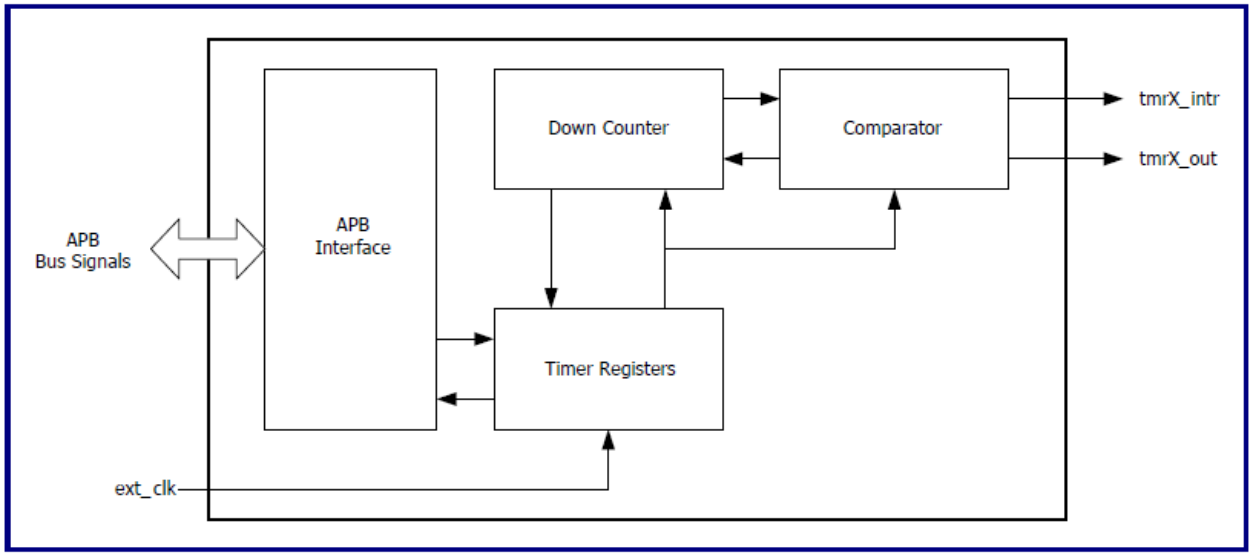


图 8-14. 功能框图

8.9.2. 特性

- 支持 AMBA 2.0 APB 总线。
- 支持多达 4 个独立的 32 位定时器。
- 每个定时器可选择内部或外部时钟源。
- 支持 PWM 功能。
- 可编程的占空比控制和 PWM 极性。
- 支持自动重载模式和单脉冲模式。
- 在 PWM 输出上支持死区发生器。

8.9.3. 寄存器摘要

表 8-68. Timer 控制器的寄存器摘要

基地址：在 Cortex-M4 AMBA APB 上为 0x54030000

基地址：在 Cortex-M0 AMBA APB 上为 0x54030000

偏移地址	名称	类型	默认值	描述
0x000	INT_CSTAT	R/W1C	0x00000000	TMR 的中断状态和控制寄存器
0x010	T1_CTRL	R/W	0x00000000	TMR 1 控制寄存器
0x014	T1_CNTB	R/W	0x00000000	TMR 1 计数缓冲寄存器
0x018	T1_CMPB	R/W	0x00000000	TMR 1 比较缓冲寄存器
0x01C	T1_CNTO	R	0x00000000	TMR 1 观察寄存器
0x020	T2_CTRL	R/W	0x00000000	TMR 2 控制寄存器
0x024	T2_CNTB	R/W	0x00000000	TMR 2 计数缓冲寄存器
0x028	T2_CMPB	R/W	0x00000000	TMR 2 比较缓冲寄存器
0x02C	T2_CNTO	R	0x00000000	TMR 2 观察寄存器
0x030	T3_CTRL	R/W	0x00000000	TMR 3 控制寄存器
0x034	T3_CNTB	R/W	0x00000000	TMR 3 计数缓冲寄存器
0x038	T3_CMPB	R/W	0x00000000	TMR 3 比较缓冲寄存器
0x03C	T3_CNTO	R	0x00000000	TMR 3 观察寄存器
0x040	T4_CTRL	R/W	0x00000000	TMR 4 控制寄存器
0x044	T4_CNTB	R/W	0x00000000	TMR 4 计数缓冲寄存器
0x048	T4_CMPB	R/W	0x00000000	TMR 4 比较缓冲寄存器
0x04C	T4_CNTO	R	0x00000000	TMR 4 观察寄存器

8.9.4. 寄存器描述

8.9.4.1. INT_CSTAT 寄存器

- 地址：0x54030000 （Cortex-M4）
- 地址：0x54030000 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：INT_CSTAT

描述：INT_CSTAT 寄存器显示每个定时器的单独中断状态。如果系统使用了 Timer 的全局中断，则 CPU 可以通过读取 INT_CSTAT 寄存器来检查发出中断的定时器。每个定时器的中断允许/禁止控制也可以在该寄存器中设置。

表 8-69. INT_CSTAT 寄存器（地址：0x54030000（Cortex-M4），0x54030000（Cortex-M0））

位	名称	类型	默认值	描述
31:4	-	-	-	保留。
3	TM4_INT_STAT	R/W1C	0x00000000	TMR4 中断状态。 写 1 清零 0:无效 1:TMR4 的计数器值达到零。
2	TM3_INT_STAT	R/W1C	0x00000000	TMR3 中断状态。 写 1 清零 0：无效 1：TMR4 的计数器值达到零。
1	TM2_INT_STAT	R/W1C	0x00000000	TMR2 中断状态。 写 1 清零 0：无效 1：TMR4 的计数器值达到零。
0	TM1_INT_STAT	R/W1C	0x00000000	TMR1 中断状态。 写 1 清零 0：无效 1：TMR1 计数器值达到零。

8.9.4.2. TMn_CTRL 寄存器

- 地址：0x540300n0 （Cortex-M4）, n = 1~4
- 地址：0x540300*n*0 （Cortex-M0）, *n* = 1~4

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：TM*n*_CTRL, *n* = 1~4

描述：该寄存器控制定时器的行为，包括启动，更新，反相输出，自动重载和死区。

表 8-70. TMn_CTRL 寄存器（地址：0x540300n0 （Cortex-M4），0x540300n0 （Cortex-M0））

位	名称	类型	默认值	描述
31:24	TMn_DZ	R/W	0x00	TMRn 死区长度。 = TMn_CTRL[31:24] - 1
23:9	-	-	-	保留。
8	TMn_PWM_EN	R/W	0x0	TMRn PWM 功能使能/关闭。 0：关闭 1：使能
7	-	-	-	保留
6	TMn_INT_MO	R/W	0x0	TMRn 中断模式。 0：电平（通过将'1'写入 TMX_INT_STAT 来清除中断） 1：脉冲
5	TMn_INT_EN	R/W	0x0	TMRn 中断使能/关闭。 0：关闭 1：使能
4	TMn_AUTO	R/W	0x0	TMRn 自动重新加载开/关。 0：单次 1：间隔模式 （自动重载）
3	TMn_OUT_INV	R/W	0x0	TMRn 输出逆变器开/关 0：逆变器关 1：逆变器开
2	TMn_UPDATE	R/W	0x0	TMRn 更新 TMX_CNTB 和 TMX_CMPB。向该寄存器写入"1"将触发手动更新。该位将在操作后自动清除。 0：无操作 1：手动更新
1	TMn_START	R/W	0x0	TMRn 使能/关闭。 0：关闭 1：使能
0	TMn_CLK_SRC	R/W	0x0	TMRn 时钟源。 0：PCLK

				1：ext_clk

8.9.4.3. TMn_CNTB 寄存器

- 地址：0x540300n4 （Cortex-M4），n = 1~4
- 地址：0x540300n4 （Cortex-M0），n = 1~4

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：TM*n*_CNTB，n = 1~4

描述：该寄存器存储计数器的起始值。计时器中的计数器将在以下情况下加载该值：

(a) 手动更新由 TMn_CTRL[TMn_UPDATE]触发。

(b) 使能自动重载并且计数器值达到零。

表 8-71. TMn_CNTB 寄存器（地址：0x540300n4（Cortex-M4），0x540300n4（Cortex-M0））

位	名称	类型	默认值	描述
31:0	TMn_CNTB	R/W	0x00000000	TMRn 计数器的起始值。

8.9.4.4. TMn_CMPB 寄存器

- 地址：0x540300n8 （Cortex-M4），n = 1~4
- 地址：0x540300n8 （Cortex-M0），n = 1~4

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：TM*n*_CMPB, n = 1, 2, 3, 4

描述：该寄存器存储计数器的比较值。当计数器值等于比较值时，将更改定时器输出的电平 tmrX_out。定时器中的比较器将在以下情况下加载该值：

(a) 手动更新由 TMn_CTRL[TMn_UPDATE]触发。

(b) 使能自动重载并且计数器值达到零。

表 8-72. TMn_CMPB 寄存器（地址：0x540300n8（Cortex-M4），0x540300n8（Cortex-M0））

位	名称	类型	默认值	描述
31:0	TMn_CMPB	R/W	0x00000000	TMRn 计数器的比较值

8.9.5. Timer 基本操作流程

下图描述了定时器操作的过程。

(1.) 启用自动重装功能(TMX_CTRL [TMX_AUTO])。将 TMX_CNTB 设置为 160(50 + 110),并将 TMX_CMPB 设置为 110。设置手动更新位(TMX_CTRL [TMX_UPDATE])和逆变器位开/关(TMX_CTRL [TMX_OUT_INV])。手动更新位将计数器和比较器的值分别设置为 TMX_CNTB 和 TMX_CMPB。

(2.) 通过设置 start 位和关闭手动更新来启动计时器。

(3.) 当计数器的值与比较器的值相同时，tmrX_out的逻辑电平将从低变为高。

(4.) 当计数器达到"0"时，计数器将自动重新加载 TMX_CNTB。同时，将产生中断请求，并将 tmrX_out 从高电平更改为低电平。

(5.) 在 ISR（中断服务程序）中，将 TMX_CNTB 和 TMX_CMPB 设置为 80（20 + 60）和 60，将在下一个持续时间使用。

(6.) 当计数器的值与比较器的值相同时，tmrX_out 的逻辑电平将从低变为高。

(7.) 当计数器达到"0"时，计数器将自动重新加载 TMX_CNTB。同时，将产生中断请求，并将 tmrX_out 从高电平更改为低电平。

(8.) 在 ISR 中，自动重载和中断请求将被禁用以停止计时器。

(9.) 当计数器的值与比较器的值相同时，tmrX_out 的逻辑电平将从低变为高。

(10.) 即使计数器达到"0"，也不会重新加载计数器，并且由于禁用了自动重新加载功能，计时器也停止了。

(11.) 不会产生中断请求。

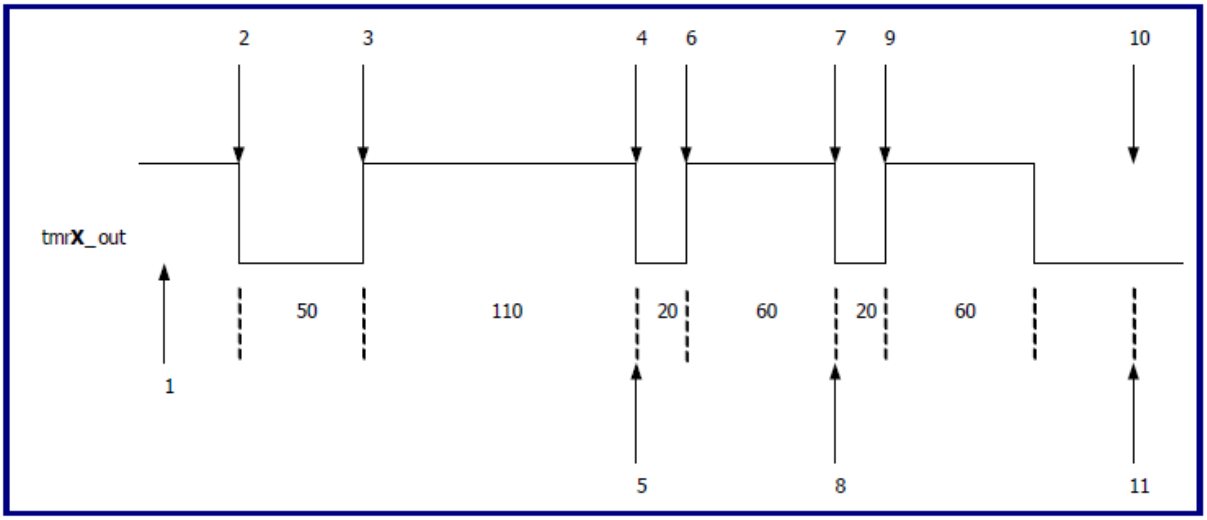


图 8-15. 定时器操作示例

8.9.6. 自动重载和双缓冲

计时器具有双重缓冲功能，可以在不停止当前计时器操作的情况下更改下一次计时器操作的重载值。因此，即使设置了新的时间值，当前的定时器操作也可以成功完成。

可以将定时器值写入TMX_CNTB（定时器计数缓冲寄存器），并且可以从TMX_CNTO（定时器计数观察寄存器）读取定时器的当前计数器值。读取TMX_CNTB时，读取的值是下一个计时器持续时间的重载值，而不是计数器的当前状态。

当计数器达到"0"且启用了自动重载功能时，自动重载功能可用于将 TMX_CNTB 复制到计数器中。当计数器达到"0"并且禁用自动重装功能时，计数器将停止运行。这称为"单次操作"。

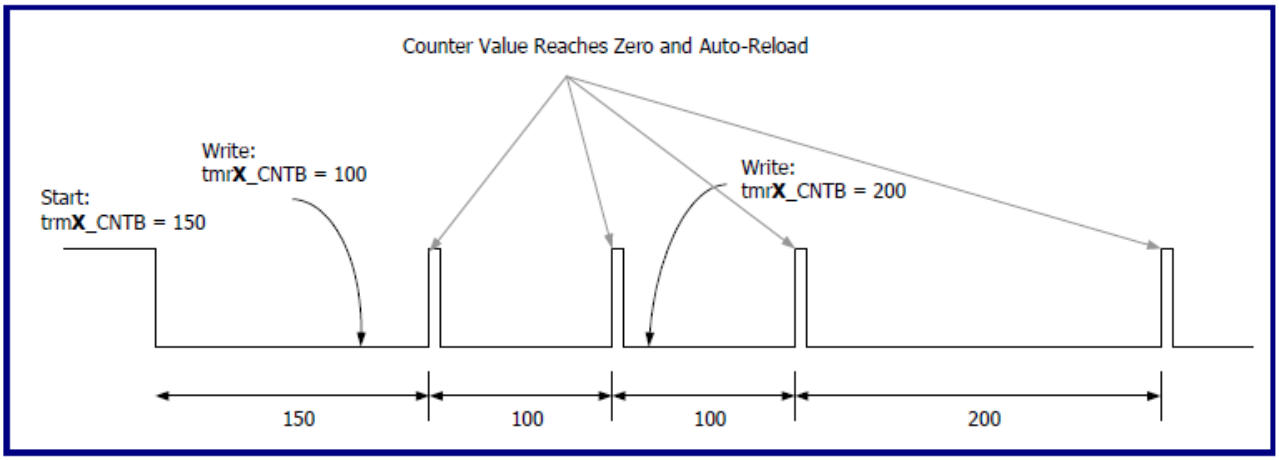


图 8-16. 双缓冲函数的举例

8.9.7. PWM 功能

可以通过使用TMX_CMPB实现PWM功能。禁用输出逆变器TMX_CTRL[TMX_OUT_INV]并启用计数器的手动更新或自动重载后，定时器输出tmrX_out将变为低电平。当计数器值等于比较值时，输出将变为高电平。该机制意味着PWM周期由TMX_CNTB确定，脉冲宽度由TMX_CMPB确定。

对于更高的PWM值，用户可以增加TMX_CMPB值。对于较低的PWM值，用户可以减小TMX_CMPB值。但是，如果启用了输出逆变器，则增量/减量可能相反。

ISR 可以将下一个 PWM 周期的双缓冲功能 TMX_CMPB 写入当前 PWM 周期的任何时间点。

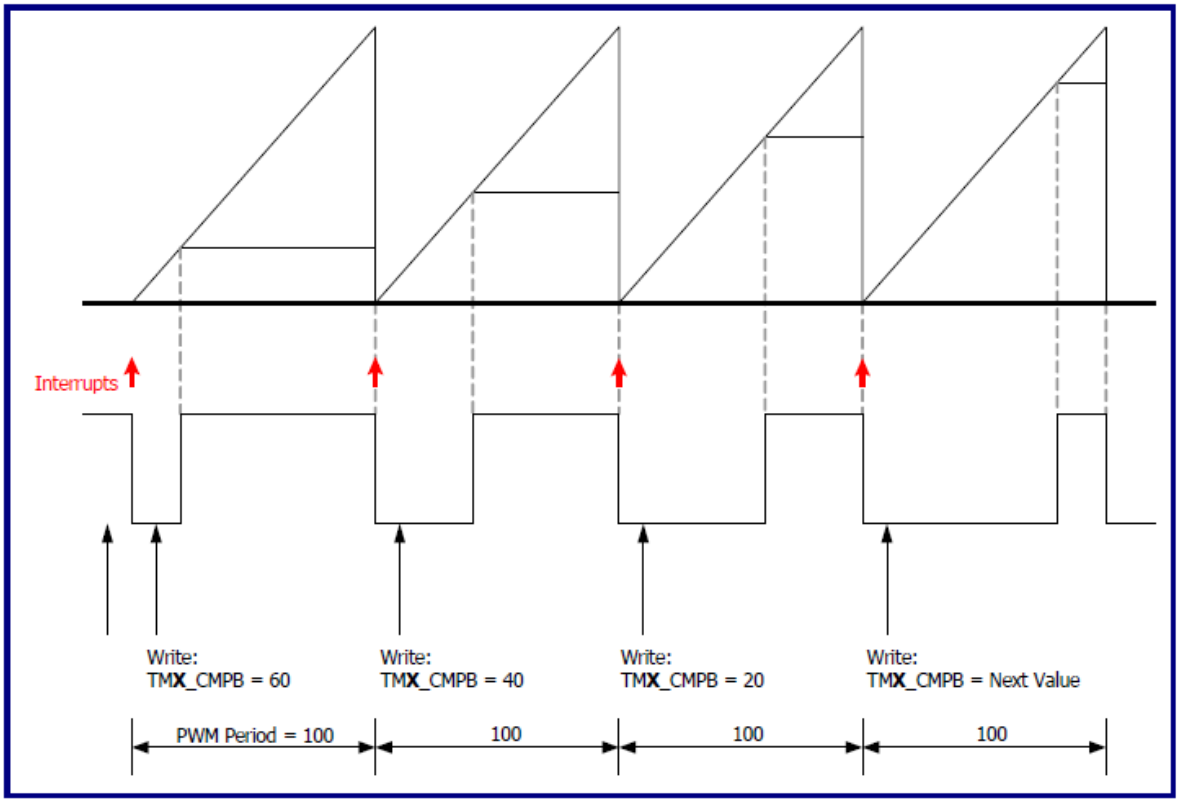


图 8-17. PWM 函数示例

8.9.8. 死区生成

死区用于PWM以控制功率设备。此功能用于在断开开关设备和接通开关设备之间插入时间间隔。该时间间隔禁止在短时间内同时打开这两个开关设备。

下图显示了启用死区功能时的波形。其中 tmrX_out 是 PWM 输出，tmrX_out_n 将是 tmrX_out 的反相。启用死区功能后，tmrX_out 和 tmrX_out_n 的输出波形将为 tmrX_out_DZ 和 tmrX_out_n_DZ。死区间隔永远无法同时打开这两个信号。为了确保功能正确，死区长度必须小于比较计数器的值； 否则，脉冲宽度小于死区的脉冲将消失。

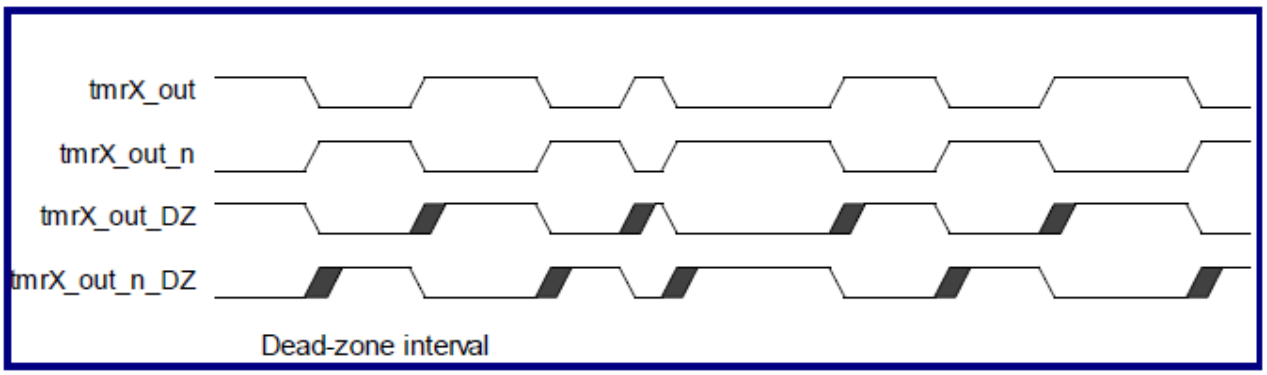


图 8-18. 启用的死区功能的波形

8.10. 看门狗

8.10.1. 概述

Vango WatchDog 计时器（WDT）用于防止软件陷入死锁时发生无限循环。在正常操作下，用户将在计数器递减计数到 0 之前定期重启 WDT。如果计数器计数到 0，则 WDT 会产生一个或多个信号，系统复位，系统中断和外部中断来复位系统，中断系统或相应地中断外部设备。

8.10.2. 特性

- 在超时后,输出将会是以下信号之一或者是组合:
  - 系统复位
  - 系统中断
  - 外部中断
- 32 位递减计数器
- 内部或外部时钟源选择
- 可变的复位超时时间
- 访问保护

8.10.3. 功能框图

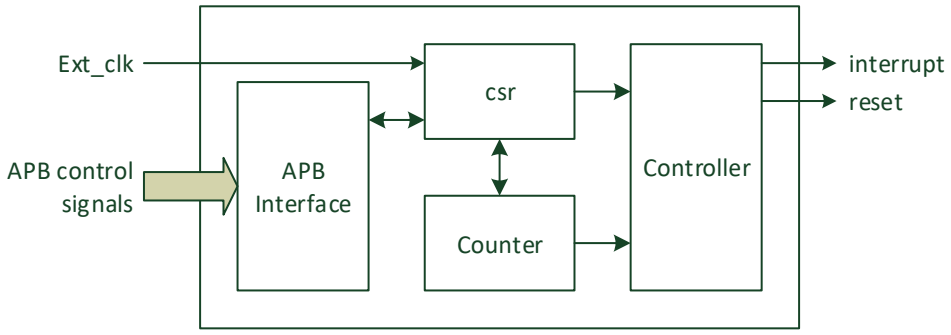


图 8-19. 功能框图

8.10.4. 功能描述

WDT 遵循 AMBA 规范。APB 接口可以从 AMBA APB 总线接收信号。复位后，WDT 寄存器将重新加载复位值。编程器打开 WatchDog 定时器控制寄存器中 WDT 的使能位后，WDT 计数器开始减少计数。WdStatus 位用于检查计数

器是否达到"0"。程序员可以通过向 WdClear 位写入 1 来清除 WdStatus 位。

为了防止意外复位，编程人员需要将 0x5AB9 作为密码写入 WdRestart 寄存器，以激活递减计数功能。如果 WdRestart 寄存器等于 0x5AB9，则 WdLoad 寄存器的值将被加载到 WatchDog 定时器的计数器中。WdRestart 寄存器的默认复位值为 0，并且每次写入后 WdRestart 寄存器将自动返回 0。WdLoad 寄存器的默认值设置为 0x3EF1480。程序员可以写入该寄存器，以自定义 WatchDog 定时器复位的操作。

8.10.5. 寄存器摘要

表 8-73. 看门狗控制器寄存器列表

基地址：0x54020000

偏移地址	寄存器名	类型	默认值	描述
0x00	WdCounter	R/W	0x03EF1480	看门狗定时器当前计数值
0x04	WdLoad	R/W	0x03EF1480	计数器自动重载值,默认为 0x3EF1480
0x08	WdRestart	W	0x00000000	看门狗计时器重启寄存器 对此寄存器写 0x5AB9,看门狗会自动将 WdLoad 寄存器的值加载到 WdCounter 寄存器中并重新开始计数
0x0C	WdCR	R/W	0x00000000	看门狗计时器控制寄存器
0x10	WdStatus	R	0x00000000	看门狗定时器状态寄存器。 当计数器达到"0"时，该位置 1。 0：未达到"0" 1：达到"0"
0x14	WdClear	R/W1C	0x00000000	看门狗计时器清除寄存器 向该寄存器写"1"将清除 WdStatus 寄存器。
0x18	WdIntrlen	R/W	0x000000FF	WatchDog 定时器中断长度。 该寄存器控制 wd_rst，wd_intr 和 wd_ext 的长度。默认值为 0xFF。

8.10.6. 寄存器描述

8.10.6.1. WdCounter 寄存器

- 地址：0x54020000

  总线：Cortex-M4 APB

  寄存器名称：WdCounter

  描述：WatchDog 计数值

表 8-74. WdCounter 寄存器（地址：0x54020000）

位	名称	类型	默认值	描述
31:0	WdCounter	R	0x03EF1480	WdCounter 包含当前计数器值。复位后，WdCounter 寄存器将设置为 0x3EF1480。程序员将 0x5AB9 写入 WdRestart 之后，WdLoad 的值将被加载到 WdCounter 中。一旦将 WdCR [0]（即看门狗定时器使能位）置 1，WdCounter 将开始减少。如果禁用了 WatchDog 计时器，则 WdCounter 将保留该值。如果设置了 WdCR[4]，则寄存器将由外部时钟驱动，并且 WdCounter 将以 EXTCLK 频率减小。该寄存器是只读的。

8.10.6.2. WdLoad 寄存器

- 地址：0x54020004

  总线：Cortex-M4 APB

  名称：WdLoad

  描述：看门狗加载值。

表 8-75. WdLoad 寄存器（地址：0x54020004）

Bits	Name	Access	Default	Description
31:0	WdLoad	R/W	0x03EF1480	WdLoad 寄存器包含将被加载到 WdCounter 中的值。 复位或重新启动时，WdLoad 的值将自动加载到 WdCounter 寄存器中。WdLoad 的复位值为 0x3EF1480。

8.10.6.3. WdRestart 寄存器

- 地址：0x54020008

  总线：Cortex-M4 APB

  名称：WdRestart

  描述：WatchDog 重启值。

表 8-76. WdRestart 寄存器（地址：0x54020008）

Bits	Name	Access	Default	Description
31:0	WdRestart	W	0x00000000	WdRestart 用于避免意外计数。如果程序员将 0x5AB9 写入该寄存器，则 WatchDog 定时器计数器会将 WdLoad 加载到 WdCounter 寄存器中，并且 WatchDog 定时器计数器将重新启动以减少计数。完成写周期后，WdRestart 将自动重置为"0"。

8.10.6.4. WdCR 寄存器

- 地址：0x5402000C

  总线：Cortex-M4 APB

  名称：WdCR

  描述：看门狗定时器控制器

表 8-77. WdCR 寄存器（地址：0x5402000C）

Bits	Name	Access	Default	Description
31:5	-	-	-	保留
4	WdClock	R/W	0x0	WatchDog 定时器时钟源位。 0：APB PCLK 1：EXTCLK（0.5MHz）
3	WdExt	R/W	0x0	WatchDog 定时器外部信号使能位。 0：禁用 1：启用
2	WdIntr	R/W	0x0	WatchDog 定时器系统中断使能位。 0：禁用 1：启用

Bits	Name	Access	Default	Description
1	WdRst	R/W	0x0	WatchDog 定时器系统复位使能位。 0：禁用 1：启用
0	WdEnable	R/W	0x0	看门狗定时器使能位。 0：禁用 1：启用

8.10.6.5. WdStatus 寄存器

- 地址：0x54020010

  总线：Cortex-M4 APB

  名称：WdStatus

  描述：看门狗状态寄存器

表 8-78. WdStatus 寄存器（地址：0x54020010）

Bits	Name	Access	Default	Description
31:1	-	-	-	保留.
0	WdStatus	R	0x0	WdStatus 寄存器将记录 WatchDog 计时器是否达到"0"。它是只读的。

8.10.6.6. WdClear 寄存器

- 地址：0x54020014

  总线：Cortex-M4 APB

  名称：WdClear

  描述：看门狗清除寄存器

表 8-79. WdClear 寄存器（地址：0x54020014）

Bits	Name	Access	Default	Description
31:1	-	-	-	保留
0	WdClear	R/W1C	0x0	当向该寄存器写入"1"时，WdStatus 寄存器将被清除。

8.10.6.7. WdIntrlen 寄存器

- 地址：0x54020018

  总线：Cortex-M4 APB

  名称：WdIntrlen

  详情：看门狗内部长度寄存器。

表 8-80. WdIntrlen 寄存器（地址：0x54020018）

Bits	Name	Access	Default	Description
31:8	-	-	-	保留.
7:0	WdIntrlen	R/W	0xFF	WdIntrlen 寄存器会确定 wd_rst，wd_intr 和 wd_ext 信号的声明持续时间。默认值为 0xFF，这意味着 wd_rst，wd_intr 和 wd_ext 的默认断言持续时间为 256 个时钟周期（PCLK）。

8.11. GPIO

8.11.1. 概述

GPIO 控制器是用户可编程的通用输入/输出控制器。它用于从系统和设备输入/输出数据。每个 GPIO 均可被编程为输入或输出。

GPIO 也可以是中断输入，支持上升沿，下降沿，两个沿以及高/低电平中断检测类型。

GPIO 最多提供 24 个可编程 I/O 端口。每个端口均可独立编程。

8.11.2. 特性

- 双向 I/O 引脚最多具有 24 个独立的输入，输出和输出使能总线。
- 当设置 INTR 选项并将其编程为输入引脚时，每个端口可以单独触发 GPIO 中断。
- 设置 INTR 选项时，每个中断都可以设置上升沿，下降沿，两个沿或高/低电平触发。
- 当设置了"Pull"选项时，可以将每个端口拉高或拉低（需要编程 I/O pad）。
- 设置 INTR 选项时，每个端口都可以选择预分频或 PCLK 时钟源。
- 输出数据位可以单独设置或清除。
- 硬件复位后，所有端口均设置为输入。

8.11.3. 功能框图

GPIO 支持数据输入/输出，数据位设置/清除，可变中断检测，"拉"功能和消抖。硬件配置可以除去**图 8-11** 中所示的灰色块。V6201X GPIO 的引脚号也可以根据用户要求进行设置。引脚号的可用范围是 1 到 24。

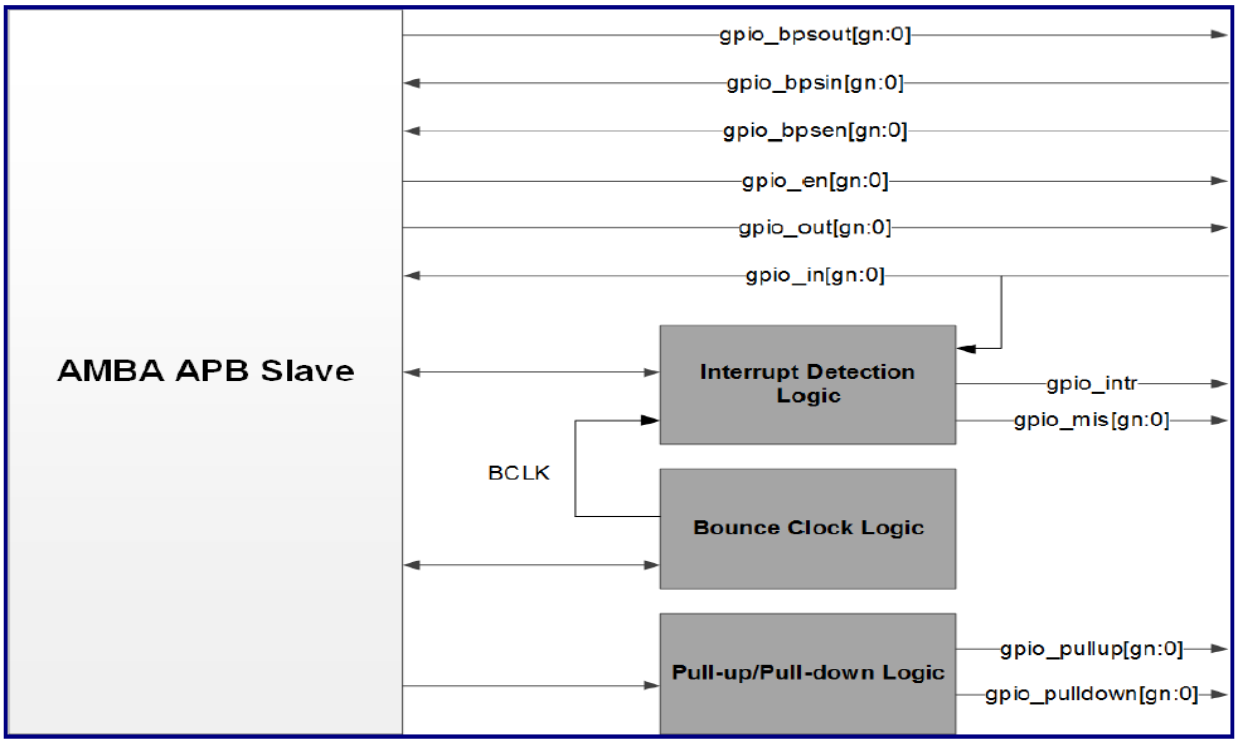

AMBA APB Slave
gpio_bpsout[gn:0]
gpio_bpsin[gn:0]
gpio_bpsen[gn:0]
gpio_en[gn:0]
gpio_out[gn:0]
gpio_in[gn:0]
Interrupt Detection Logic
gpio_intr
gpio_mis[gn:0]
BCLK
Bounce Clock Logic
Pull-up/Pull-down Logic
gpio_pullup[gn:0]
gpio_pulldown[gn:0]

图 **8-20.** 功能模块图

8.11.4. 连接图

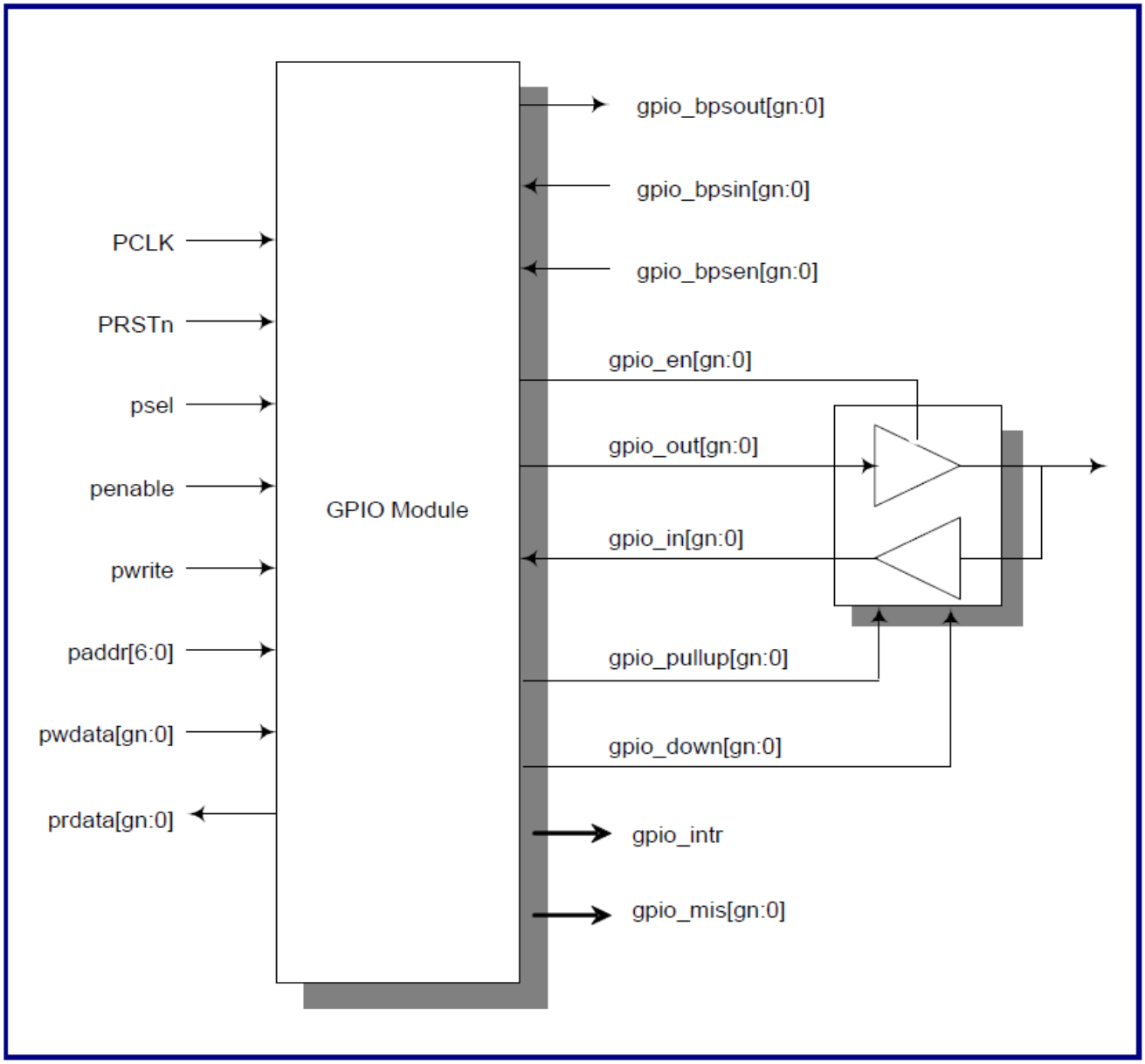


图 8-21. 连接图

8.11.5. 寄存器摘要

表 8-81. GPIO 寄存器概况

基地址：0x54050000 on Coretx-M4 AMBA APB

基地址：0x54060000 on Coretx-M0 AMBA APB

偏移地址	寄存器名	类型	默认值	描述

0x00	GpioDataOut	R/W	0x00000000	GPIO 输出数据寄存器
0x04	GpioDataIn	R	0x00000000	GPIO 输入数据寄存器
0x10	GpioDataSet	R/W1S	0x00000000	GPIO 数据位设置寄存器 写入该寄存器时，数据寄存器中的相应位将被设置为“1”，而其他位将保持不变。
0x14	GpioDataClear	R/W1C	0x00000000	GPIO 数据位清除寄存器 写入该寄存器时，数据寄存器中的相应位将被清除为 0，而其他位将保持不变。
0x18	PinPullEnable	R/W	0x00000000	GPIO 上拉寄存器 0：未上拉。 1：已上拉。
0x1C	PinPullType	R/W	0x00000000	GPIO 上拉/下拉寄存器 0：引脚被拉低。 1：引脚被拉高。
0x20	IntrEnable	R/W	0x00000000	GPIO 中断使能寄存器 0：禁止引脚中断。 1：使能引脚中断。
0x24	IntrRawState	R	0x00000000	GPIO 中断原始状态寄存器 0：未检测到中断。 1：检测到中断。
0x28	IntrMaskedState	R	0x00000000	GPIO 中断屏蔽状态寄存器 0：未检测到或屏蔽中断。 1：检测到中断且未被屏蔽。
0x2C	IntrMask	R/W	0x00000000	GPIO 中断屏蔽寄存器 0：屏蔽被禁用。 1：屏蔽被启用。
0x30	IntrClear	R/W1C	0x00000000	GPIO 中断清除寄存器 0：无效 1：清除中断
0x34	IntrTrigger	R/W	0x00000000	GPIO 中断触发方法寄存器 0：边沿触发 1：电平触发

0x38	IntrBoth	R/W	0x00000000	GPIO 边沿触发被单边沿或两个边沿中断 0：单边沿 1：双边沿
0x3C	IntrRiseNeg	R/W	0x00000000	GPIO 中断在上升沿或下降沿触发 0：上升沿 1：下降沿 GPIO 中断由高电平或低电平触发 0：高电平 1：低电平
0x40	BounceEnable	R/W	0x00000000	GPIO 预分频使能 当这个信号被启用时，PCLK 将被 BouncePreScale 时钟所除。该信号用于延长检测中断的时钟周期。 0:禁用 1:启用
0x44	BouncePreScale	R/W	0x00000000	GPIO 预分频 该信号用于调整不同的 PCLK 频率。 允许范围是 0x1 到 0xFFFFFF。

8.11.6. 寄存器描述

8.11.6.1. GpioDataOut 寄存器

- 地址：0x54050000（Cortex-M4）
- 地址：0x54060000（Cortex-M0）

  总线：Cortex-M4 APB/ Cortex-M0 APB

  名称：GpioDataOut

  描述：GpioDataOut 寄存器是 GPIO 数据输出寄存器。当 PinDir 指示该引脚为输出时，GpioDataOut 有效。当 PinDir 指示该引脚是输入时，GpioDataOut 可以保存数据。

表 8-82. GpioDataOut 寄存器（地址：0x54050000 （Cortex-M4），0x54060000 （Cortex-M0））

位	名称	类型	默认值	描述
31:0	GpioDataOut	R/W	0x00000000	GPIO 输出数据寄存器。

8.11.6.2. GpioDataIn 寄存器

- 地址：0x54050004（Cortex-M4）
- 地址：0x54060004（Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：GpioDataIn

描述：GpioDataIn 寄存器是 GPIO 数据寄存器。当 PinDir 指示该引脚为输入时，GpioDataIn 在 PCLK 上升沿工作。当 PinDir 指示该引脚为输出时，GpioDataIn 寄存器将为"无关"寄存器。

表 8-83. GpioDataIn 寄存器（地址：0x54050004（Cortex-M4），0x54060004（Cortex-M0））

位	名称	类型	默认值	描述
31:0	GpioDataIn	R	0x00000000	GPIO 输入数据寄存器。

8.11.6.3. GpioDataSet 寄存器

- 地址：0x54050010（Cortex-M4）
- 地址：0x54060010（Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：GpioDataSet

描述：GpioDataSet 寄存器是位操作逻辑。写入该地址时，如果 GpioDataSet 的某些位为 1，则 GpioDataOut 的相关位将被置 1。

表 8-84. GpioDataSet 寄存器（地址：0x54050010（Cortex-M4），0x54060010（Cortex-M0））

位	名称	类型	默认值	描述
31:0	GpioDataSet	R/W1S	0x00000000	-

8.11.6.4. GpioDataClear 寄存器

- 地址：0x54050014 （Cortex-M4）
- 地址：0x54060014 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：GpioDataClear

说明：GpioDataClear 寄存器是位操作逻辑。写入该地址时，如果 GpioDataClear 的某些位为 1，则 GpioDataOut 的相关位将被清除。

表 8-85. GpioDataClear 寄存器（地址：0x54050014（Cortex-M4），0x54060014（Cortex-M0））

位	名称	类型	默认值	描述
31:0	GpioDataClear	R/W1C	0x00000000	-

8.11.6.5. PinPullEnable 寄存器

- 地址：0x54050018 （Cortex-M4）
- 地址：0x54060018 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：PinPullEnable

Description：使能 IO 上下拉

表 8-86. PinPullEnable 寄存器（地址：0x54050018（Cortex-M4），0x54060018（Cortex-M0））

位	名称	类型	默认值	描述
31:0	PinPullEnable	R/W	0x00000000	0：禁止 1：使能

8.11.6.6. PinPullType 寄存器

- 地址：0x5405001C （Cortex-M4）
- 地址：0x5406001C （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：PinPullType

描述：此寄存器用于选择 IO 上拉还是下拉

表 8-87. PinPullType 寄存器（地址：0x5405001C（Cortex-M4），0x5406001C（Cortex-M0））

位	名称	类型	默认值	描述
31:0	PinPullType	R/W	0x00000000	0：下拉

位	名称	类型	默认值	描述
				1：上拉

8.11.6.7. IntrEnable 寄存器

- 地址：0x54050020 （Cortex-M4）
- 地址：0x54060020 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrEnable

说明：IntrEnable 寄存器控制启用或禁用中断检测逻辑。它是中断检测逻辑的掩码。当引脚方向为输入并且启用了中断检测时，引脚可以接受来自 pad 的中断。感测到的状态存储在 IntrMaskedState 寄存器中（由 IntrEnable 屏蔽）。在打开 IntrEnable 之前，程序员可以通过向 IntrClear 写一个“ 1”以确保初始状态来清除屏蔽状态。

表 8-88. IntrEnable 寄存器（地址：0x54050020（Cortex-M4），0x54060020（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrEnable	R/W	0x00000000	0：禁止 1：使能

8.11.6.8. IntrRawState 寄存器

- 地址：0x54050024 （Cortex-M4）
- 地址：0x54060024 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrRawState

说明：IntrRawState 寄存器是中断检测的原始结果。如果启用了 IntrEnable，则 IntrRawState 寄存器将反映中断检测状态。程序员可以查询该寄存器以检测中断。

表 8-89. IntrRawState 寄存器（地址：0x54050024（Cortex-M4），0x54060024（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrRawState	R	0x00000000	0：未检测到中断事件 1：检测到中断事件

8.11.6.9. IntrMaskedState 寄存器

- 地址：0x54050028 （Cortex-M4）
- 地址：0x54060028 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrMaskedState

说明：IntrMaskedState 寄存器是中断检测的屏蔽结果。IntrMaskedState 寄存器由 IntrEnable，IntrRawState 和 IntrMask 寄存器控制。

表 8-90. IntrMaskedState 寄存器（地址：0x54050028（Cortex-M4），0x54060028（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrMaskedState	R	0x00000000	0：未检测到或屏蔽中断。 1：检测到中断且未被屏蔽。

8.11.6.10. IntrMask 寄存器

- 地址：0x5405002C （Cortex-M4）
- 地址：0x5406002C （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrMask

说明：IntrMask 寄存器是中断检测的屏蔽寄存器。它屏蔽了 IntrRawState 寄存器。例如，如果 IntrEnable [0] ='1'，IntrRawState [0] ='1'，并且 IntrMask [0] ='1'，则 IntrMaskedState [0]将永远不会更改为'1'。

表 8-91. IntrMask 寄存器（地址：0x5405002C（Cortex-M4），0x5406002C（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrMask	R/W	0x00000000	0：屏蔽被禁用 1：屏蔽被启用

8.11.6.11. IntrClear 寄存器

- 地址：0x54050030 （Cortex-M4）
- 地址：0x54060030 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrClear

说明：IntrClear 寄存器是位操作逻辑。写入 IntrClear 地址时，IntrMaskedState 和 IntrRawState 的相关位将被清除。

表 8-92. IntrClear 寄存器（地址：0x54050030（Cortex-M4），0x54060030（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrClear	R/W1C	0x00000000	0：无效 1：清除中断

8.11.6.12. IntrTrigger 寄存器

- 地址：0x54050034 （Cortex-M4）
- 地址：0x54060034 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrTrigger

说明：IntrTrigger 寄存器指示每个引脚的中断触发方法。如果 IntrTrigger 为 0，则中断将在边沿触发；否则，中断将是电平触发。

表 8-93. IntrTrigger 寄存器（地址：0x54050034（Cortex-M4），0x54060034（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrTrigger	R/W	0x00000000	0：边沿触发 1：电平触发

8.11.6.13. IntrBoth 寄存器

- 地址：0x54050038 （Cortex-M4）
- 地址：0x54060038 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrBoth

说明：IntrBoth 寄存器指示边沿触发可以是单边沿，也可以是双边沿。如果 IntrTrigger 指示边沿触发且 IntrBoth 为 0，则中断边沿触发将由单个边沿完成。如果 IntrTrigger 指示边沿触发并且 IntrBoth 为 1，则中断边沿触发将由两个边沿完成。

表 8-94. IntrBoth 寄存器（地址：0x54050038（Cortex-M4），0x54060038（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrBoth	R/W	0x00000000	0：单边沿 1：双边沿

8.11.6.14. IntrRiseNeg 寄存器

- 地址：0x5405003C （Cortex-M4）
- 地址：0x5406003C （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：IntrRiseNeg

说明：IntrRiseNeg 寄存器指示边沿触发是在上升沿还是下降沿。如果 IntrTrigger 是边沿触发，且 IntrBoth 设置单边沿，IntrRiseNeg 是 0，此时中断触发边沿是上升沿。如果 IntrTrigger 是边沿触发，且 IntrBoth 设置单边沿，IntrRiseNeg 是 1，此时中断触发边沿是下降沿。

表 8-95. IntrRiseNeg 寄存器（地址：0x5405003C（Cortex-M4），0x5406003C（Cortex-M0））

位	名称	类型	默认值	描述
31:0	IntrRiseNeg	R/W	0x00000000	当中断设置边沿触发： 0：上升沿 1：下降沿 当中断设置电平触发 0：高电平 1：低电平

8.11.6.15. BounceEnable 寄存器

- 地址：0x54050040 （Cortex-M4）
- 地址：0x54060040 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：BounceEnable

说明：BounceEnable 寄存器控制滤波功能。如果 BounceEnable 使能，则中断检测将由外部时钟进行采样。分频数将由 BouncePreScale 寄存器控制。

表 8-96. BounceEnable 寄存器（地址：0x54050040（Cortex-M4），0x54060040（Cortex-M0））

位	名称	类型	默认值	描述
31:0	BounceEnable	R/W	0x00000000	Require the “INTR” option

8.11.6.16. BouncePreScale 寄存器

- 地址：0x54050044 （Cortex-M4）
- 地址：0x54060044 （Cortex-M0）

总线：Cortex-M4 APB/ Cortex-M0 APB

名称：BouncePreScale

说明：BouncePreScale 寄存器是一个滤波计时器的自动指示寄存器。它可以将 PCLK 扩展到 BouncePreScale 周期，该周期可用于调整不同机器中的中断采样时钟周期。复位值为0x7D0，这意味着如果APB时钟频率为66 MHz，则滤波时钟将被（0x7D0 +1）除以 32.98 kHz。程序员可以调整该寄存器以适合不同的系统。

表 8-97. BouncePreScale 寄存器（地址：0x54050044（Cortex-M4），0x54060044（Cortex-M0））

位	名称	类型	默认值	描述
31:0	BouncePreScale	R/W	0x000007D0	Require the “INTR” option

8.11.7.时序图

8.11.7.1. 写 GPIO 输出数据寄存器

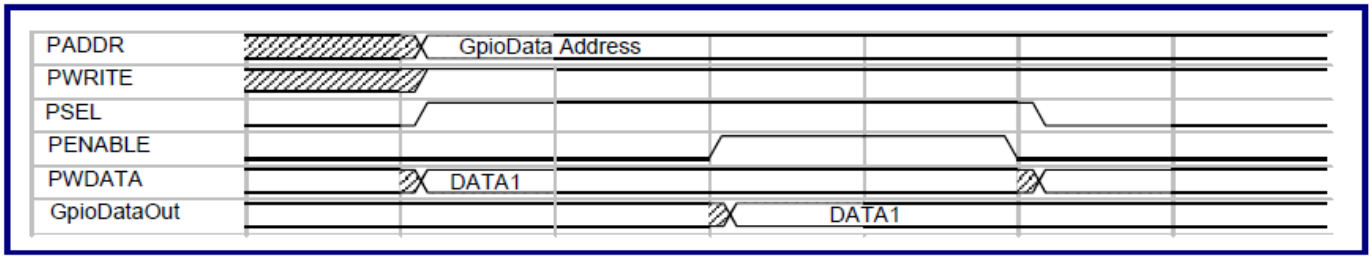


图 8-22. 写 GPIO 数据输出寄存器

8.11.7.2. 下降沿触发

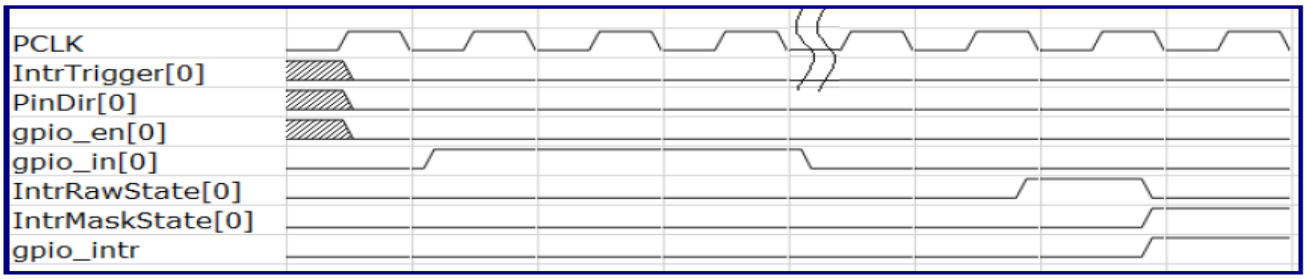


图 8-23. 下降沿触发

8.11.7.3. 高电平触发

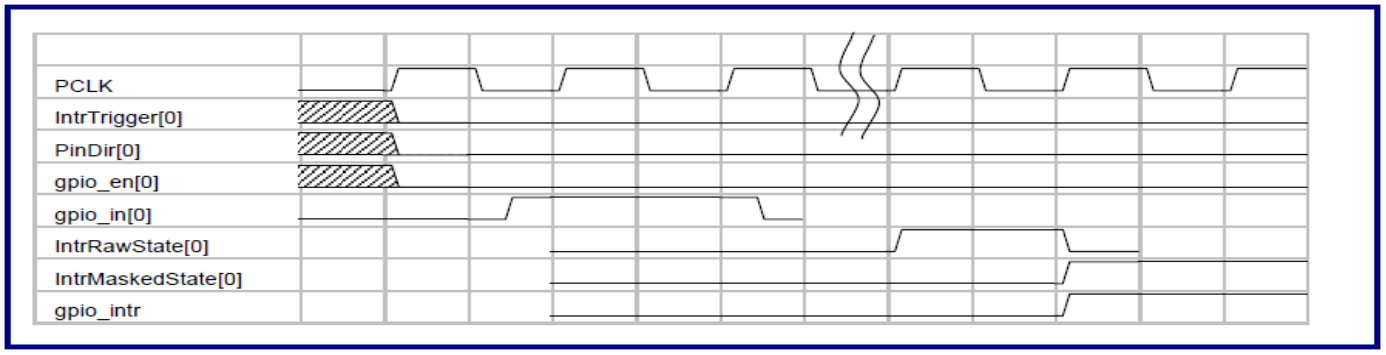


图 8-24. 高电平触发

8.12. VGDMA

8.12.1. 概述

V6201X 集成了 2 个 8 通道万高 DMA，一个在 Cortex-M0 AHB 总线上，另外一个在 Cortex-M4 AHB 总线上。

8.12.2. DMA 框图

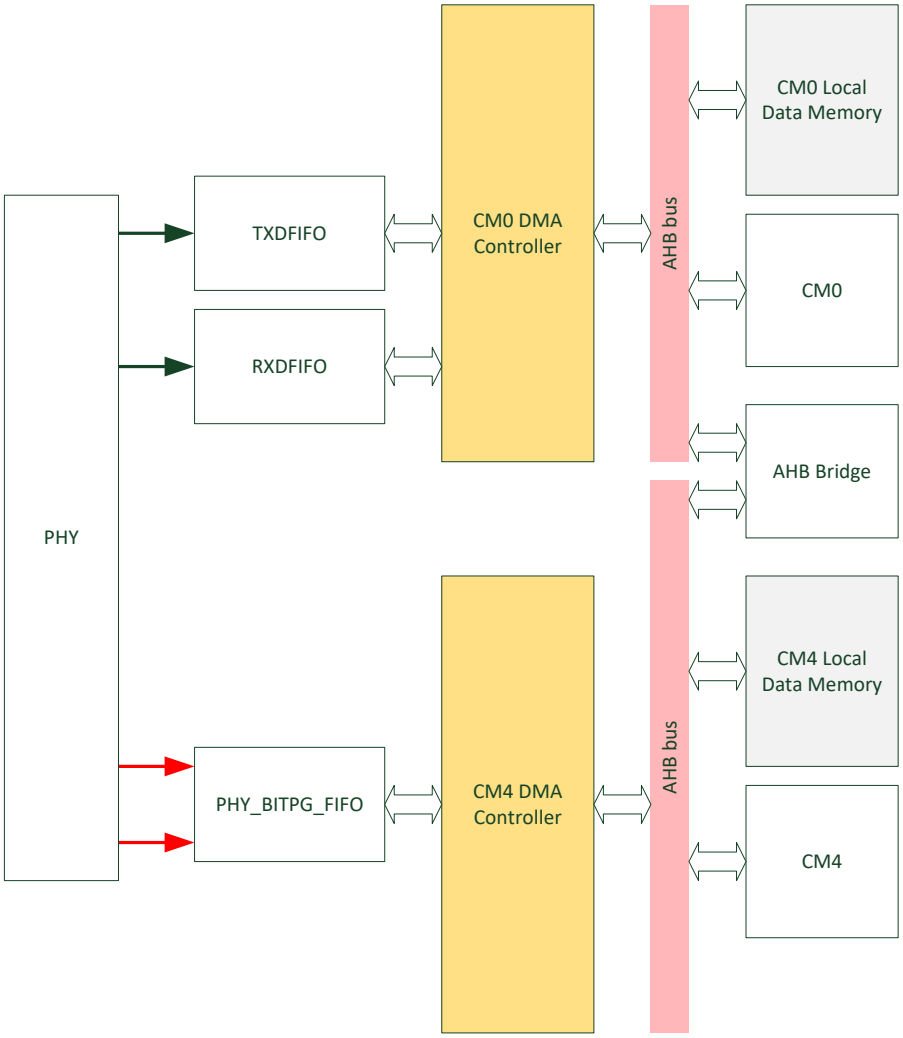


图 8-25. DMA 框图

8.12.3. Cortex-M4 DMA 寄存器（通道号 n=0~7）

8.12.3.1. Chn_MODE 寄存器

- 地址：0x57000000+n*0x40

  总线：Cortex-M4 AHB

  名称：Ch*n*_MODE

  描述：通道 n 模式控制。

表 8-98. Chn_MODE 寄存器（地址：0x57000000+n*0x40）

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:7	-	-	-	保留。
6	CHAIN	R/W	0x0	CHAIN 模式从通道 0 开始。 如果 n 为 0，通道 0 在接收到设置为 CHAIN 模式的最后一个通道的 dma_done 时自动开始传输，整个 CHAIN 传输将一直重复，直到其中一个通道的 CHAIN 模式设置为非活动（0x0）。
5	-	-	-	保留。
4	HS_EN	R/W	0x0	设置为在握手模式下运行。
3	FIXDES	R/W	0x0	固定目的地址为 ADDRDES。
2	FIXSRC	R/W	0x0	固定源地址为 ADDRSRC。
1	ABORT	R/W	0x0	中止通道数据传输。
0	TRANSEN	R/W	0x0	使能 DMA 数据传输，并在传输完成后由硬件自动清零。

8.12.3.2. CHn_ADDRSRC 寄存器

- 地址：0x57000004+*n**0x40

  总线：Cortex-M4 AHB

  名称：CH*n*_ADDRSRC

  描述：通道 n 传输的源地址。

表 8-99. CHn_ADDRSRC 寄存器（地址：0x57000004+n*0x40）

位	名称	类型	默认值	描述
31:0	ADDRSRC	R/W	0x00000000	传输的源地址。

8.12.3.3. CHn_ADDRDES 寄存器

- 地址：0x57000008+*n**0x40

  总线：Cortex-M4 AHB

  名称：CH*n*_ADDRDES

  描述：通道 n 传输的目的地址。

表 8-100. CHn_ADDRDES 寄存器（地址：0x57000008+n*0x40）

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:0	ADDRDES	R/W	0x00000000	传输的目的地址。

8.12.3.4. CHn_BLENGTH 寄存器

- 地址：0x5700000C+*n**0x40

  总线：Cortex-M4 AHB

  名称：CH*n*_BLENGTH

  描述：通道 n 传输的总数据长度（字节）。

表 8-101. CHn_BLENGTH 寄存器（地址：0x5700000C+n*0x40）

位	名称	类型	默认值	描述
31:17	-	-	-	保留。
16:0	BLENGTH	R/W	0x00000	传输的总数据长度（字节）。 范围：0~65536。

8.12.3.5. CHn_PKTLENGTH 寄存器

- 地址：0x57000010+*n**0x40

  总线：Cortex-M4 AHB

  名称：CH*n*_PKTLENGTH

  描述：通道 n 传输的最大数据包长度（字节）。

表 8-102. CHn_PKTLENGTH 寄存器（地址：0x57000010+n*0x40）

位	名称	类型	默认值	描述
31:17	-	-	-	保留。
10:0	PKTLENGTH	R/W	0x000	每次仲裁传输的最大数据包长度（字节）。 范围：0~512。

8.12.3.6. DMA_RESET 寄存器

- 地址：0x57000038

  总线：Cortex-M4 AHB

  名称：DMA_RESET

描述：复位 DMA 状态机。

表 8-103. DMA_RESET 寄存器（地址：0x57000038）

位	名称	类型	默认值	描述
31:1	-	-	-	保留
0	DMARESET	R/W	0x0	复位 DMA 控制器状态机，该位应该由软件清零。

8.12.3.7. DMA_INTC 寄存器

- 地址：0x5700003C

  总线：Cortex-M4 AHB

  名称：DMA_INTC

  描述：DMA 中断控制。

表 8-104. DMA_INTC 寄存器（地址：0x5700003C）

位	名称	类型	默认值	描述
31:24	-	-	-	保留。
23	CH7_INTEN	R/W	0x0	通道 7 中断使能。
22	CH6_INTEN	R/W	0x0	通道 6 中断使能。
21	CH5_INTEN	R/W	0x0	通道 5 中断使能。
20	CH4_INTEN	R/W	0x0	通道 4 中断使能。
19	CH3_INTEN	R/W	0x0	通道 3 中断使能。
18	CH2_INTEN	R/W	0x0	通道 2 中断使能。
17	CH1_INTEN	R/W	0x0	通道 1 中断使能。
16	CH0_INTEN	R/W	0x0	通道 0 中断使能。
15	CH7_INTCLR	R/W1C	0x0	清除 CH7_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。。
14	CH6_INTCLR	R/W1C	0x0	清除 CH6_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。。
13	CH5_INTCLR	R/W1C	0x0	清除 CH5_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。。
12	CH4_INTCLR	R/W1C	0x0	清除 CH4_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。

位	名称	类型	默认值	描述
11	CH3_INTCLR	R/W1C	0x0	清除 CH3_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
10	CH2_INTCLR	R/W1C	0x0	清除 CH2_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
9	CH1_INTCLR	R/W1C	0x0	清除 CH1_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
8	CH0_INTCLR	R/W1C	0x0	清除 CH0_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
7	CH7_INTR	R	0x0	通道 7 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
6	CH6_INTR	R	0x0	通道 6 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
5	CH5_INTR	R	0x0	通道 5 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
4	CH4_INTR	R	0x0	通道 4 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
3	CH3_INTR	R	0x0	通道 3 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
2	CH2_INTR	R	0x0	通道 2 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
1	CH1_INTR	R	0x0	通道 1 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
0	CH0_INTR	R	0x0	通道 0 中断状态标志位。 该位在传输完成或者中止时由硬件置位。

8.12.4. Cortex-M0 DMA 寄存器（通道号 n=0~7）

8.12.4.1. Chn_MODE 寄存器

- 地址：0x77000000+*n**0x40

  总线：Cortex-M0 AHB

  名称：Ch*n*_MODE

  描述：通道 n 模式控制。

表 8-105. Chn_MODE 寄存器（地址：0x77000000+n*0x40）

位	名称	类型	默认值	描述
31:6	-	-	-	保留。
5	HS_ATMODE	R/W	0x0	使用自动长度模式，在 PHY 的 FIFO 控制器使用握手模式的时候有效。
4	HS_EN	R/W	0x0	设置为在握手模式下运行。
3	FIXDES	R/W	0x0	固定目的地址为 ADDRDES。
2	FIXSRC	R/W	0x0	固定源地址为 ADDRSRC。
1	ABORT	R/W	0x0	中止 DMA 通道数据传输，当 ABORT 完成后 DMA 状态机将会自动产生中断。
0	TRANSEN	R/W	0x0	使能 DMA 数据传输，并在传输完成后由硬件自动清零。

8.12.4.2. CHn_ADDRSRC 寄存器

- 地址：0x77000004+n*0x40

  总线：Cortex-M0 AHB

  名称：CH*n*_ADDRSRC

  描述：通道 n 传输的源地址。

表 8-106. CHn_ADDRSRC 寄存器（地址：0x77000004+n*0x40）

位	名称	类型	默认值	描述
31:0	ADDRSRC	R/W	0x00000000	传输的源地址。

8.12.4.3. CHn_ADDRDES 寄存器

- 地址：0x77000008+*n**0x40

  总线：Cortex-M0 AHB

  名称：CH*n*_ADDRDES

  描述：通道 n 传输的目的地址。

表 8-107. CHn_ADDRDES 寄存器（地址：0x77000008+n*0x40）

位	名称	类型	默认值	描述
31:0	ADDRDES	R/W	0x00000000	传输的目的地址。

8.12.4.4. CHn_BLENGTH 寄存器

- 地址：0x7700000C+*n**0x40

  总线：Cortex-M0 AHB

  名称：CH*n*_BLENGTH

  描述：通道 n 传输的总数据长度（字节）。

表 8-108. CHn_BLENGTH 寄存器（地址：0x7700000C+n*0x40）

位	名称	类型	默认值	描述
31:17	-	-	-	保留。
16:0	BLENGTH	R/W	0x00000	传输的总数据长度（字节）。 范围：0~65536。

8.12.4.5. CHn_PKTLENGTH 寄存器

- 地址：0x77000010+*n**0x40

  总线：Cortex-M0 AHB

  名称：CH*n*_PKTLENGTH

  描述：通道 n 传输的最大数据包长度（字节）。

表 8-109. CHn_PKTLENGTH 寄存器（地址：0x77000010+n*0x40）

位	名称	类型	默认值	描述
31:11	-	-	-	保留。
10:0	PKTLENGTH	R/W	0x000	每次仲裁传输的最大数据包长度（字节）。

位	名称	类型	默认值	描述
				范围：0~512

8.12.4.6. DMA_RESET 寄存器

- 地址：0x77000038

  总线：Cortex-M0 AHB

  名称：DMA_RESET

  描述：复位 DMA 状态机。

表 8-110. DMA_RESET 寄存器（地址：0x77000038）

位	名称	类型	默认值	描述
31:1	-	-	-	保留。
0	DMARESET	R/W	0x0	复位 DMA 控制器状态机，该位应该由软件清零。

8.12.4.7. DMA_INTC 寄存器

- 地址：0x7700003C

  总线：Cortex-M0 AHB

  名称：DMA_INTC

  名称：DMA 中断控制。

表 8-111. DMA_INTC 寄存器（地址：0x7700003C）

位	名称	类型	默认值	描述
31:24	-	-	-	保留。
23	CH7_INTEN	R/W	0x0	通道 7 中断使能。
22	CH6_INTEN	R/W	0x0	通道 6 中断使能。
21	CH5_INTEN	R/W	0x0	通道 5 中断使能。
20	CH4_INTEN	R/W	0x0	通道 4 中断使能。
19	CH3_INTEN	R/W	0x0	通道 3 中断使能。
18	CH2_INTEN	R/W	0x0	通道 2 中断使能。
17	CH1_INTEN	R/W	0x0	通道 1 中断使能。
16	CH0_INTEN	R/W	0x0	通道 0 中断使能。

15	CH7_INTCLR	R/W1C	0x0	清除 CH7_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
14	CH6_INTCLR	R/W1C	0x0	清除 CH6_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
13	CH5_INTCLR	R/W1C	0x0	清除 CH5_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
12	CH4_INTCLR	R/W1C	0x0	清除 CH4_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
11	CH3_INTCLR	R/W1C	0x0	清除 CH3_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
10	CH2_INTCLR	R/W1C	0x0	清除 CH2_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
9	CH1_INTCLR	R/W1C	0x0	清除 CH1_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
8	CH0_INTCLR	R/W1C	0x0	清除 CH0_INTR 中断标志位。 该位在中断标志清除完成后由硬件自动清零。
7	CH7_INTR	R	0x0	通道 7 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
6	CH6_INTR	R	0x0	通道 6 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
5	CH5_INTR	R	0x0	通道 5 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
4	CH4_INTR	R	0x0	通道 4 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
3	CH3_INTR	R	0x0	通道 3 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
2	CH2_INTR	R	0x0	通道 2 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
1	CH1_INTR	R	0x0	通道 1 中断状态标志位。 该位在传输完成或者中止时由硬件置位。
0	CH0_INTR	R	0x0	通道 0 中断状态标志位。

				该位在传输完成或者中止时由硬件置位。

8.13. 安全加速器

8.13.1. 为安全加速器服务的 DMA 的控制状态寄存器

8.13.2. DMA CSR 概述

本节描述 DMA 操作所需的控制和状态寄存器（CSR）。寄存器只支持双字长地址寻址。DMA CSR 的基址是 0x59000000。DMA CSR 概述如下，DMA 引擎 1 可以被认为是安全加速器的入口，DMA 引擎 2 和 DMA 引擎 3 是保留的。

表 8-112. 安全加速器 DMA CSR 概述

地址	名称	描述
0x000	DE1_CR	DMA 引擎 1 控制寄存器。
0x004	DE1_TDBAR	DMA 引擎 1 传输描述符基地址。
0x008	DE1_CTDAR	DMA 引擎 1 传输描述符的当前地址。
0x00C	DE1_TDR	DMA 引擎 1 发送请求寄存器。
0x010	DE1_RDBAR	DMA 引擎 1 接收描述符基地址。
0x014	DE1_CRDAR	DMA 引擎 1 接收描述符的当前地址。
0x018	DE1_RDR	DMA 引擎 1 接收请求寄存器。
0x020	IER	中断使能寄存器。
0x024	ISR	中断状态寄存器。
0x030	PROCR	探测控制寄存器。
0x040	DE2_CR	DMA 引擎 2 控制寄存器。
0x044	DE2_TDBAR	DMA 引擎 2 传输描述符基地址。
0x048	DE2_CTDAR	DMA 引擎 2 传输描述符的当前地址。
0x04C	DE2_TDR	DMA 引擎 2 发送请求寄存器。
0x050	DE2_RDBAR	DMA 引擎 2 接收描述符基地址。
0x054	DE2_CRDAR	DMA 引擎 2 接收描述符的当前地址。
0x058	DE2_RDR	DMA 引擎 2 接收请求寄存器。
0x060	TXENG_CA_KR	TX 引擎当前地址触发寄存器。
0x064	TXENG1_CA_SR	TX 引擎 1 当前地址设置寄存器。
0x068	TXENG2_CA_SR	TX 引擎 2 当前地址设置寄存器。

地址	名称	描述
0x06C	TXENG3_CA_SR	TX 引擎 3 当前地址设置寄存器。
0x070	TXENG_FIDLE_KR	TXENG 强制返回空转井涌寄存器。
0x080	FSM_DE1	DMA 引擎 1 FSM 用于调试。
0x084	FSM_DE2	DMA 引擎 2 FSM 用于调试。
0x088	FSM_DE3	DMA 引擎 3 FSM 用于调试。
0x090	FSM_PMH	PMH FSM 用于调试。
0x094	PMH Debug1	PMH 调试 1。
0x098	PMH Debug2	PMH 调试 2。
0x09C	PMH Debug3	PMH 调试 3.
0x0A0	ESA_Debug_FSM	ESA_Debug_FSM.
0x0A4	ESA_Debug_CNT1	ESA_Debug_CNT1.
0x0A8	ESA_Debug_CNT2	ESA_Debug_CNT2.
0x0B0	SCR	系统控制寄存器。
0x0B4	ZCCR	过零控制寄存器。
0x0B8	PDCR	相位检测计数寄存器。
0x0BC	GCCR	门控时钟计数器寄存器。
0x0C0	PHY_CR1	PHY 控制寄存器 1。
0x0C4	PHY_CR2	PHY 控制寄存器 2。
0x0C8	PHY_IR1	PHY 信息寄存器 1。
0x0CC	PHY_IR2	PHY 信息寄存器 2。
0x0D0	PHY_CR3	PHY 控制寄存器 3。
0x0D4	PHY_IR3	PHY 信息寄存器 3。
0x0F8	RTL VER	RTL 版本。
0x0FC	ASIC_VER	ASIC 版本。
0x100	DE3_CR	DMA 引擎 3 控制寄存器。
0x104	DE3_TDBAR	DMA 引擎 3 传输描述符基址。
0x108	DE3_CTDAR	DMA 引擎 3 当前传输描述符地址。
0x10C	DE3_TDR	DMA 引擎 3 发送请求寄存器。
0x110	DE3_RDBAR	DMA 引擎 3 接收描述符的基址。
0x114	DE3_CRDAR	DMA 引擎 3 接收描述符的当前地址。

地址	名称	描述
0x118	DE3_RDR	DMA 引擎 3 接收请求寄存器。
0x120	ECHO_CSR_CR	CSR 访问控制寄存器。
0x130	ECHO_DMA_SWRSTCR	ECHO DMA 软复位控制寄存器。
0x140	ECHO_RXENG	ECHO RXENG MUX CSR。
0x1F0	ECHO_RSVD_CSR	保留。

8.13.3. 寄存器描述

8.13.3.1. DE1_CR 寄存器

- 地址：0x59000000

  总线：Cortex-M4 AHB

  名称：DE1_CR

  描述：DMA 引擎 1 控制寄存器。

表 8-113. DE1_CR 寄存器（地址：0x59000000）

位	名称	类型	默认值	描述
31:16	de1_dgl	R/W	0x0000	DMA引擎1描述符间隙长度。
15:10	-	R	0x00	保留。
9	de1_lenmode	R/W	0x0	DMA引擎1 RX DESC长度模式。 1:来自MOD 0:由DMA计数器
8	de1_lb	R/W	0x0	DMA 引擎 1 环回模式。
7:6	-	R	0x0	保留。
5	de1_txe	R/W	0x0	DMA 引擎 1 传输 DMA 使能。
4	de1_rxe	R/W	0x0	DMA 引擎 1 接收 DMA 使能。
3:2	-	R	0x0	保留。
1	de1_bs	R/W	0x0	DMA 引擎 1 字节交换。
0	de1_be	R/W	0x0	DMA Engine 1 大端模式。

8.13.3.2. DE1_TDBAR 寄存器

- 地址：0x59000004

  总线：Cortex-M4 AHB

  名称：DE1_TDBAR

  描述：DMA 引擎 1 传输描述符基址。

表 8-114. DE1_TDBAR 寄存器

位	名称	类型	默认值	描述
31:0	de1_tdbar	R/W	0x00000000	DMA 引擎 1 传输描述符基址。

8.13.3.3. DE1_CTDAR 寄存器

- 地址：0x59000008

  总线：Cortex-M4 AHB

  名称：DE1_CTDAR

  描述：DMA 引擎 1 当前传输描述符地址。

表 8-115. DE1_CTDAR 寄存器

位	名称	类型	默认值	描述
31:0	de1_ctdar	R	0x00000000	DMA 引擎 1 当前传输描述符地址。

8.13.3.4. DE1_TDR 寄存器

- 地址：0x5900000C

  总线：Cortex-M4 AHB

  名称：DE1_TDR

  描述：DMA 引擎 1 发送请求寄存器。

表 8-116. DE1_TDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。
0	de1_tdr	W1C	0x00000000	DMA 引擎 1 发送请求寄存器。 读取此寄存器时始终返回 0。 任何写操作都会触发 TXE 进入状态

位	名称	类型	默认值	描述
				TXE_RD_DESC.

8.13.3.5. DE1_RDBAR 寄存器

- 地址：0x59000010

  总线：Cortex-M4 AHB

  名称：DE1_RDBAR

  描述：DMA 引擎 1 接收描述符的基址。

表 8-117. DE1_RDBAR 寄存器

位	名称	类型	默认值	描述
31:0	de1_rdbar	R/W	0x00000000	DMA 引擎 1 接收描述符的基址。

8.13.3.6. DE1_CRDAR 寄存器

- 地址：0x59000014

  总线：Cortex-M4 AHB

  名称：DE1_CRDAR

  描述：DMA 引擎 1 接收描述符的当前地址。

表 8-118. DE1_CRDAR 寄存器

位	名称	类型	默认值	描述
31:0	de1_crdar	R/W	0x00000000	DMA 引擎 1 接收描述符的当前地址。

8.13.3.7. DE1_RDR 寄存器

- 地址：0x59000018

  总线：Cortex-M4 AHB

  名称：DE1_RDR

  描述：DMA 引擎 1 接收请求寄存器。

表 8-119. DE1_RDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。

位	名称	类型	默认值	描述
0	de1_rdr	R/W	0x00000000	DMA 引擎 1 接收轮询请求。 任何写操作都会触发 TXE 进入状态 RXE_RD_DESC。

8.13.3.8. IER 寄存器

- 地址：0x59000020

  总线：Cortex-M4 AHB

  名称：IER

  描述：中断使能寄存器。

表 8-120. IER 寄存器

位	名称	类型	默认值	描述
31	ier_zc	R/W	0x0	过零**中断使能寄存器**（上升和下降均可由 CSR 控制）。
30	ier_ld_urgent	R/W	0x0	用于 LD 紧急情况的中断使能寄存器。
29	ier_det_pmb	R/W	0x0	用于前导码检测的中断使能寄存器。
28	ier_dpid_alternate	R/W	0x0	DPID 备用的中断使能寄存器。
27	ier_bi_end	R/W	0x0	信标间隔结束指示器的中断使能寄存器。
26	ier_si_end	R/W	0x0	超帧结束指示中断使能寄存器。
25	ier_csma_fail	R/W	0x0	csma 失败中断使能寄存器。
24	ier_csma_success	R/W	0x0	csma 成功中断使能寄存器。
23	ier_kickphy	R/W	0x0	Kick PHY 中断使能寄存器。
22	ier_darq_swresp	R/W	0x0	DARQ SW 响应中断使能寄存器。
21	ier_sarq_noearlier	R/W	0x0	SARQ GOT 响应中断使能寄存器（NO EARLIER）。
20	ier_arq_acktimeout	R/W	0x0	ARQ Ack超时中断使能寄存器。
19	ier_arq_retryfail	R/W	0x0	ARQ重试失败中断使能寄存器（重试次数达到arq_maxretry）。
18	ier_arq_txsegdone	R/W	0x0	ARQ TX段完成（每段）中断使能寄存器[仅G3]。

位	名称	类型	默认值	描述
17	ier_arq_nack	R/W	0x0	ARQ NACK（每帧）中断使能寄存器。
16	ier_arq_ack	R/W	0x0	ARQ ACK（每帧）中断使能寄存器。
15:12	-	R	0x0	保留。
11	ier_de3_rxunavail	R/W	0x0	DMA引擎3 RX描述符不可用中断使能寄存器。
10	ier_de3_rxcmp	R/W	0x0	DMA 引擎 3 RX 描述符完成中断使能寄存器。
9	ier_de3_txunavail	R/W	0x0	DMA引擎3 TX描述符不可用中断使能寄存器。
8	ier_de3_txcmp	R/W	0x0	DMA 引擎 3 TX 描述符完成中断使能寄存器。
7	ier_de2_rxunavail	R/W	0x0	DMA 引擎 2 RX 描述符不可用中断使能寄存器。
6	ier_de2_rxcmp	R/W	0x0	DMA 引擎 2 RX 描述符完成中断使能寄存器。
5	ier_de2_txunavail	R/W	0x0	DMA 引擎 2 TX 描述符不可用中断使能寄存器。
4	ier_de2_txcmp	R/W	0x0	DMA 引擎 2 TX 描述符完成中断使能寄存器。
3	ier_de1_rxunavail	R/W	0x0	DMA 引擎 1 RX 描述符不可用中断使能寄存器。
2	ier_de1_rxcmp	R/W	0x0	DMA 引擎 1 RX 描述符完成中断使能寄存器。
1	ier_de1_txunavail	R/W	0x0	DMA 引擎 1 TX 描述符不可用中断使能寄存器。
0	ier_de1_txcmp	R/W	0x0	DMA 引擎 1 TX 描述符完成中断使能寄存器。

8.13.3.9. ISR 寄存器

- 地址：0x59000024

  总线：Cortex-M4 AHB

  名称：IER

  描述：中断状态寄存器。

表 8-121. ISR 寄存器

位	名称	类型	默认值	描述
31	isr_zc	W1C	0x0	过零中断状态寄存器（上升和下降均可由 CSR 控制）。
30	isr_ld_urgent	W1C	0x0	紧急 LD 的中断状态寄存器。
29	isr_det_pmb	W1C	0x0	前导码检测的中断状态寄存器。
28	isr_dpid_alternate	W1C	0x0	DPID 备用的中断状态寄存器。
27	isr_bi_end	W1C	0x0	信标间隔结束指示器的中断状态寄存器。
26	isr_si_end	W1C	0x0	超帧结束指示器的中断状态寄存器。
25	isr_csma_fail	W1C	0x0	CSMA 失败中断状态寄存器。
24	isr_csma_success	W1C	0x0	CSMA 成功中断状态寄存器。
23	isr_kickphy	W1C	0x0	Kick PHY 中断状态寄存器。
22	isr_darq_swresp	W1C	0x0	DARQ SW 响应中断状态寄存器。
21	isr_sarq_noearlier	W1C	0x0	SARQ GOT 响应中断状态寄存器（NO EARLIER）。
20	isr_arq_acktimeout	W1C	0x0	ARQ Ack 超时中断状态寄存器。
19	isr_arq_retryfail	W1C	0x0	ARQ 重试失败中断状态寄存器(重试次数达到 arq_maxretry)。
18	isr_arq_txsegdone	W1C	0x0	ARQ TX 段完成中断状态寄存器（每整段）[仅 G3]。
17	isr_arq_nack	W1C	0x0	ARQ NACK 中断状态寄存器 （每帧）。
16	isr_arq_ack	W1C	0x0	ARQ ACK 中断状态寄存器（每帧）。
15:12	-	R	0x0	保留。
11	isr_de3_rxunavail	W1C	0x0	DMA 引擎 3 接收描述符不可用中断状态寄存器。
10	isr_de3_rxcmp	W1C	0x0	DMA 引擎 3 RX 描述符完成中断状态寄存器。
9	isr_de3_txunavail	W1C	0x0	DMA 引擎 3 TX 描述符不可用中断状态寄存器。
8	isr_de3_txcmp	W1C	0x0	DMA 引擎 3 TX 描述符完成中断状态寄存器。
7	isr_de2_rxunavail	W1C	0x0	DMA 引擎 2 接收描述符不可用中断

位	名称	类型	默认值	描述
				状态寄存器。
6	isr_de2_rxcmp	W1C	0x0	DMA 引擎 2 RX 描述符完成中断状态寄存器。
5	isr_de2_txunavail	W1C	0x0	DMA 引擎 2 TX 描述符不可用中断状态寄存器。
4	isr_de2_txcmp	W1C	0x0	DMA 引擎 2 TX 描述符完成中断状态寄存器。
3	isr_de1_rxunavail	W1C	0x0	DMA 引擎 1 接收描述符不可用中断状态寄存器。
2	isr_de1_rxcmp	W1C	0x0	DMA 引擎 1 RX 描述符完成中断状态寄存器。
1	isr_de1_txunavail	W1C	0x0	DMA 引擎 1 TX 描述符不可用中断状态寄存器。
0	isr_de1_txcmp	W1C	0x0	DMA 引擎 1 TX 描述符完成中断状态寄存器。

8.13.3.10. PROCR 寄存器

- 地址：0x59000030

  总线：Cortex-M4 AHB

  名称：PROCR

  描述：探测控制寄存器。

表 8-122. PROCR 寄存器

位	名称	类型	默认值	描述
31:16	-	R	0x00	保留。
15:8	mac_probe_sel	R/W	0x0	MAC 探测组选择。
7:1	-	R	0x0	保留。
0	probe_sel	R/W	0x0	探测模式选择。 0: MAC 1: BBP

8.13.3.11. DE2_CR 寄存器

- 地址：0x59000040

  总线：Cortex-M4 AHB

  名称：DE2_CR

  描述：DMA 引擎 2 控制寄存器。

表 8-123. DE2_CR 寄存器

位	名称	类型	默认值	描述
31:16	de2_dgl	R/W	0x0000	DMA引擎2描述符间隙长度。
15:10	-	R	0x00	保留。
9	de2_lenmode	R/W	0x0	DMA引擎2 RX DESC长度模式。1：来自MOD 0：由DMA计数器
8	de2_lb	R/W	0x0	DMA 引擎 2 环回模式。
7:6	-	R	0x0	保留。
5	de2_txe	R/W	0x0	DMA 引擎 2 传输 DMA 使能。
4	de2_rxe	R/W	0x0	DMA 引擎 2 接收 DMA 使能。
3:2	-	R	0x0	保留。
1	de2_bs	R/W	0x0	DMA 引擎 2 字节交换。
0	de2_be	R/W	0x0	DMA 引擎 2 大端模式。

8.13.3.12. DE2_TDBAR 寄存器

- 地址：0x59000044

  总线：Cortex-M4 AHB

  名称：DE2_TDBAR

  描述：DMA 引擎 2 传输描述符基址。

表 8-124. DE2_TDBAR 寄存器

位	名称	类型	默认值	描述
31:0	de2_tdbar	R/W	0x00000000	DMA 引擎 2 传输描述符基址。

8.13.3.13. DE2_CTDAR 寄存器

- 地址：0x59000048

  总线：Cortex-M4 AHB

  名称：DE2_CTDAR

  描述：DMA 引擎 2 当前传输描述符地址。

表 8-125. DE2_CTDAR 寄存器

位	名称	类型	默认值	描述
31:0	de2_ctdar	R/W	0x00000000	DMA 引擎 2 当前传输描述符地址。

8.13.3.14. DE2_TDR 寄存器

- 地址：0x5900004C

  总线：Cortex-M4 AHB

  名称：DE2_TDR

  描述：DMA 引擎 2 发送请求寄存器。

表 8-126. DE2_TDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。
0	de2_tdr	R/W	0x00000000	DMA 引擎 2 发送请求寄存器。 读取此寄存器时始终返回 0。 任何写操作都会触发 TXE 进入状态 TXE_RD_DESC.

8.13.3.15. DE2_RDBAR 寄存器

- 地址：0x59000050

  总线：Cortex-M4 AHB

  名称：DE2_RDBAR

  描述：DMA 引擎 2 接收描述符的基址。

表 8-127. DE2_RDBAR 寄存器

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:0	de2_rdbar	R/W	0x00000000	DMA 引擎 2 接收描述符的基址。

8.13.3.16. DE2_CRDAR 寄存器

- 地址：0x59000054

  总线：Cortex-M4 AHB

  名称：DE2_CRDAR

  描述：DMA 引擎 2 接收描述符的当前地址。

表 8-128. DE2_CRDAR 寄存器

位	名称	类型	默认值	描述
31:0	de2_crdar	R/W	0x00000000	DMA 引擎 2 接收描述符的当前地址。

8.13.3.17. DE2_RDR 寄存器

- 地址：0x59000058

  总线：Cortex-M4 AHB

  名称：DE2_RDR

  描述：DMA 引擎 2 接收请求寄存器。

表 8-129. DE2_RDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。
0	de2_rdr	R/W	0x00000000	DMA 引擎 2 接收轮询请求。 任何写操作都会触发 TXE 进入状态 RXE_RD_DESC。

8.13.3.18. TXENG_CA_KR 寄存器

- 地址：0x59000060

  总线：Cortex-M4 AHB

  名称：TXENG_CA_KR

  描述：TX 引擎当前地址触发寄存器。

表 8-130. TXENG_CA_KR 寄存器

位	名称	类型	默认值	描述
31:3	-	R	0x0	保留。
2	te3_caddr_update	R/W	0x0	TXENG3 当前地址更新。
1	te2_caddr_update	R/W	0x0	TXENG2 当前地址更新。
0	te1_caddr_update	R/W	0x0	TXENG3 当前地址更新。

8.13.3.19. TXENG1_CA_SR 寄存器

- 地址：0x59000064

  总线：Cortex-M4 AHB

  名称：TXENG1_CA_SR

  描述：TX 引擎 1 当前地址设置寄存器。

表 8-131. TXENG1_CA_SR 寄存器

位	名称	类型	默认值	描述
31:0	te1_caddr	R/W	0x00000000	TX 引擎 1 当前地址设置寄存器。

8.13.3.20. TXENG2_CA_SR 寄存器

- 地址：0x59000068

  总线：Cortex-M4 AHB

  名称：TXENG2_CA_SR

描述：TX 引擎 2 当前地址设置寄存器。

表 8-132. TXENG2_CA_SR 寄存器

位	名称	类型	默认值	描述
31:0	te2_caddr	R/W	0x00000000	TX 引擎 2 当前地址设置寄存器。

8.13.3.21. TXENG3_CA_SR 寄存器

- 地址：0x5900006C

  总线：Cortex-M4 AHB

  名称：TXENG3_CA_SR

描述：TX 引擎 3 当前地址设置寄存器。

表 8-133. TXENG3_CA_SR 寄存器

位	名称	类型	默认值	描述
31:0	te3_caddr	R/W	0x00000000	TX 引擎 3 当前地址设置寄存器。

8.13.3.22. TXENG_FIDLE_KR 寄存器

- 地址：0x59000070

  总线：Cortex-M4 AHB

  名称：TXENG_FIDLE_KR

  描述：TXENG 强制返回空转井涌寄存器。

表 8-134. TXENG_FIDLE_KR 寄存器

位	名称	类型	默认值	描述
31:2	-	R	0x0	保留。
1	te2_forceidle_kick	R/W	0x0	TXENG2 强制回 idle kick。
0	-	R	0x0	保留。

8.13.3.23. FSM_DE1 寄存器

- 地址：0x59000080

  总线：Cortex-M4 AHB

  名称：FSM_DE1

  描述：DMA引擎1 FSM。

表 8-135. FSM_DE1 寄存器

位	名称	类型	默认值	描述
31:27	-	R	0x0	保留。
26:24	fsm_de1_rmd	R	0x0	RMD FSM。
23:21	-	R	0x0	保留。
20:16	fsm_de1_rxe	R	0x0	RXENG FSM。
15:11	-	R	0x0	保留。
10:8	fsm_de1_tmd	R	0x0	TMD FSM。

位	名称	类型	默认值	描述
7:5	-	R	0x0	保留。
4:0	fsm_de1_txe	R	0x0	TXENG FSM。

8.13.3.24. FSM_DE2 寄存器

- 地址：0x59000084

  总线：Cortex-M4 AHB

  名称：FSM_DE2

  描述：DMA引擎2 FSM。

表 8-136. FSM_DE2 寄存器

位	名称	类型	默认值	描述
31:27	-	R	0x0	保留。
26:24	fsm_de2_rmd	R	0x0	RMD FSM。
23:21	-	R	0x0	保留。
20:16	fsm_de2_rxe	R	0x0	RXENG FSM。
15:11	-	R	0x0	保留。
10:8	fsm_de2_tmd	R	0x0	TMD FSM。
7:5	-	R	0x0	保留。
4:0	fsm_de2_txe	R	0x0	TXENG FSM。

8.13.3.25. FSM_DE3 寄存器

- 地址：0x59000088

  总线：Cortex-M4 AHB

  名称：FSM_DE3

  描述：DMA引擎3 FSM。

表 8-137. FSM_DE3 寄存器

位	名称	类型	默认值	描述
31:27	-	R	0x0	保留。
26:24	fsm_de3_rmd	R	0x0	RMD FSM。

位	名称	类型	默认值	描述
23:21	-	R	0x0	保留。
20:16	fsm_de3_rxe	R	0x0	RXENG FSM。
15:11	-	R	0x0	保留。
10:8	fsm_de3_tmd	R	0x0	TMD FSM。
7:5	-	R	0x0	保留。
4:0	fsm_de3_txe	R	0x0	TXENG FSM。

8.13.3.26. FSM_PMH 寄存器

- 地址：0x59000090

  总线：Cortex-M4 AHB

  名称：FSM_PMH

  描述：PMH FSM。

表 8-138. FSM_PMH 寄存器

位	名称	类型	默认值	描述
31:20	-	R	0x0	保留。
19:16	fsm_pmh_pout	R	0x0	POUT FSM。
15:12	-	R	0x0	保留。
11:8	fsm_pmh_aes	R	0x0	AES FSM。
7:5	-	R	0x0	保留。
4:0	fsm_pmh_pin	R	0x0	PIN FSM。

8.13.3.27. PMH_Debug1 寄存器

- 地址：0x59000094

  总线：Cortex-M4 AHB

  名称：PMH_Debug1

  描述：PMH 调试寄存器 1。

表 8-139. PMH_Debug1 寄存器

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:0	pmh_debug1	R	0x00000000	PMH 调试寄存器 1。

8.13.3.28. PMH_Debug2 寄存器

- 地址：0x59000098

  总线：Cortex-M4 AHB

  名称：PMH_Debug2

  描述：PMH 调试寄存器 2。

表 8-140. PMH_Debug2 寄存器

位	名称	类型	默认值	描述
31:0	pmh_debug2	R	0x00000000	PMH 调试寄存器 2。

8.13.3.29. PMH_Debug3 寄存器

- 地址：0x5900009C

  总线：Cortex-M4 AHB

  名称：PMH_Debug3

  描述：PMH 调试寄存器 3.

表 8-141. PMH_Debug3 寄存器

位	名称	类型	默认值	描述
31:0	pmh_debug3	R	0x00000000	PMH 调试寄存器 3。

8.13.3.30. ESA_Debug_FSM 寄存器

- 地址：0x590000A0

  总线：Cortex-M4 AHB

  名称：ESA_Debug_FSM

  描述：ESA_Debug_FSM.

表 8-142. ESA_Debug_FSM 寄存器

位	名称	类型	默认值	描述
31:0	esa_debug_fsm	R	0x00000000	esa_debug_fsm。

8.13.3.31. ESA_Debug_CNT1 寄存器

- 地址：0x590000A4

  总线：Cortex-M4 AHB

  名称：ESA_Debug_CNT1

  描述：ESA 调试计数器 1。

表 8-143. ESA_Debug_CNT1 寄存器

位	名称	类型	默认值	描述
31:0	esa_debug_cnt1	R	0x00000000	ESA 调试计数器 1。

8.13.3.32. ESA_Debug_CNT2 寄存器

- 地址：0x590000A8

  总线：Cortex-M4 AHB

  名称：ESA_Debug_CNT2

  描述：ESA 调试计数器 2。

表 8-144. ESA_Debug_CNT2 寄存器

位	名称	类型	默认值	描述
31:0	esa_debug_cnt2	R	0x00000000	ESA 调试计数器 2。

8.13.3.33. SCR 寄存器

- 地址：0x590000B0

  总线：Cortex-M4 AHB

  名称：SCR

  描述：系统控制寄存器。

表 8-145. SCR 寄存器

位	名称	类型	默认值	描述
31:27	-	R	0x0	保留。
26:8	zc_hold	R/W	128000	保持128MHz时钟滴答声。
7:2	-	R	0x0	保留。
1	zc_both	R/W	0x0	1：检测上升和下降边缘。

位	名称	类型	默认值	描述
				0：则只检测到上升边。（对于中断）
0	zc_enb	R/W	0x0	过零检测使能。

8.13.3.34. ZCCR 寄存器

- 地址：0x590000B4

  总线：Cortex-M4 AHB

  名称：ZCCR

  描述：过零检测控制寄存器。

表 8-146. ESA_Debug_CNT2 寄存器

位	名称	类型	默认值	描述
31:0	zc_mask	R/W	0x00000000	过零输入屏蔽寄存器。

8.13.3.35. PDCR 寄存器

- 地址：0x590000B8

  总线：Cortex-M4 AHB

  名称：PDCR

  描述：相位检测计数器寄存器。

表 8-147. PDCR 寄存器

位	名称	类型	默认值	描述
31:24	-	R	0x0	保留。
23:0	zc_pdc_period_cnt	R	0x0	过零相位检测计数器。

8.13.3.36. GCCR 寄存器

- 地址：0x590000BC

  总线：Cortex-M4 AHB

  名称：GCCR

  描述：门控时钟计数器寄存器。

表 8-148. GCCR 寄存器

位	名称	类型	默认值	描述
31:6	-	R	0x0	保留。
5	ppc_cg_en	R/W	0x0	ppc csr门控时钟clk_ahb使能。 0：永远打开 1：当选中时打开
4	dma_cg_en	R/W	0x0	dma csr门控时钟clk_ahb使能。 0：永远打开 1：当选中时打开
3	dsp_ppc_cg_en	R/W	0x0	PPC门控时钟clk_dsp使能。
2	mac_sec_cg_en	R/W	0x0	Security门控时钟clk_mac使能。
1	ahb_ppc_cg_en	R/W	0x0	PP门控时钟clk_ahb使能。
0	ahb_sec_cg_en	R/W	0x0	Security门控时钟clk_ahb使能。

8.13.3.37. PHY_CR1 寄存器

- 地址：0x590000C0

  总线：Cortex-M4 AHB

  名称：PHY_CR1

  描述：PHY控制寄存器1。

表 8-149. PHY_CR1 寄存器

位	名称	类型	默认值	描述
31:0	phy_cr1	R/W	0x0	PHY控制寄存器1 DSP复位。

8.13.3.38. PHY_CR2 寄存器

- 地址：0x590000C4

  总线：Cortex-M4 AHB

  名称：PHY_CR2

  描述：PHY控制寄存器2 DSP复位。

表 8-150. PHY_CR2 寄存器

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:0	phy_cr2	R/W	0x0	PHY控制寄存器2 DSP复位。

8.13.3.39. PHY_IR1 寄存器

- 地址：0x590000C8

  总线：Cortex-M4 AHB

  名称：PHY_IR1

  描述：PHY信息寄存器1。

表 8-151. PHY_IR1 寄存器

位	名称	类型	默认值	描述
31:0	phy_ir1	R	0x0	PHY信息寄存器1。

8.13.3.40. PHY_IR2 寄存器

- 地址：0x590000CC

  总线：Cortex-M4 AHB

  名称：PHY_IR2

  描述：PHY信息寄存器2。

表 8-152. PHY_IR2 寄存器

位	名称	类型	默认值	描述
31:0	phy_ir2	R	0x0	PHY信息寄存器2。

8.13.3.41. PHY_CR3 寄存器

- 地址：0x590000D0

  总线：Cortex-M4 AHB

  名称：PHY_CR3

  描述：PHY控制寄存器3。

表 8-153. PHY_CR3 寄存器

位	名称	类型	默认值	描述
31:0	phy_cr3	R/W	0x0	PHY控制寄存器3。

8.13.3.42. PHY_IR3 寄存器

- 地址：0x590000D4

  总线：Cortex-M4 AHB

  名称：PHY_IR3

  描述：PHY信息寄存器3。

表 8-154. PHY_IR3 寄存器

位	名称	类型	默认值	描述
31:0	phy_ir3	R	0x0	PHY信息寄存器3。

8.13.3.43. RTL_VER 寄存器

- 地址：0x590000F8

  总线：Cortex-M4 AHB

  名称：RTL_VER

  描述：RTL版本。

表 8-155. RTL_VER 寄存器

位	名称	类型	默认值	描述
31:0	rtl_version	R	0xe0000324	RTL版本，可以用来标记最新RTL的发布日期。

8.13.3.44. ASIC_VER 寄存器

- 地址：0x590000FC

  总线：Cortex-M4 AHB

  名称：ASIC_VER

  描述：ASIC版本。

表 8-156. ASIC_VER 寄存器

位	名称	类型	默认值	描述
31:28	-	R	0x0	保留。
27:0	asic_ver	R	0x0	SIC版本。 第一个版本=0x6300E_A1，

位	名称	类型	默认值	描述
				0x6300EA是常量，0x1为ECO单元。

8.13.3.45. DE3_CR 寄存器

- 地址：0x59000100

  总线：Cortex-M4 AHB

  名称：DE3_CR

  描述：DMA 引擎 3 控制寄存器.

表 8-157. DE3_CR 寄存器

位	名称	类型	默认值	描述
31:16	de3_dgl	R/W	0x0000	DMA引擎3描述符间隙长度。
15:10	-	R	0x00	保留。
9	de3_lenmode	R/W	0x0	DMA引擎3 RX DESC长度模式。1：来自MOD 0：由DMA计数器
8	de3_lb	R/W	0x0	DMA 引擎 3 环回模式。
7:6	-	R	0x0	保留。
5	de3_txe	R/W	0x0	DMA 引擎 3 传输 DMA 使能。
4	de3_rxe	R/W	0x0	DMA 引擎 3 接收 DMA 使能。
3:2	-	R	0x0	保留。
1	de3_bs	R/W	0x0	DMA 引擎 3 字节交换。
0	de3_be	R/W	0x0	DMA 引擎 3 大端模式。

8.13.3.46. DE3_TDBAR 寄存器

- 地址：0x59000104

  总线：Cortex-M4 AHB

  名称：DE3_TDBAR

  描述：DMA 引擎 3 传输描述符基址。

表 8-158. DE3_TDBAR 寄存器

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:0	de3_tdbar	R/W	0x00000000	DMA 引擎 3 传输描述符基址。

8.13.3.47. DE3_CTDAR 寄存器

- 地址：0x59000108

  总线：Cortex-M4 AHB

  名称：DE3_CTDAR

  描述：DMA 引擎 3 当前传输描述符地址。

表 8-159. DE3_CTDAR 寄存器

位	名称	类型	默认值	描述
31:0	de3_ctdar	R/W	0x00000000	DMA 引擎 3 当前传输描述符地址。

8.13.3.48. DE3_TDR 寄存器

- 地址：0x5900010C

  总线：Cortex-M4 AHB

  名称：DE3_TDR

  描述：DMA 引擎 3 发送请求寄存器。

表 8-160. DE3_TDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。
0	de3_tdr	R/W	0x00000000	DMA 引擎 3 发送请求寄存器。 读取此寄存器时始终返回 0。 任何写操作都会触发 TXE 进入状态 TXE_RD_DESC.

8.13.3.49. DE3_RDBAR 寄存器

- 地址：0x59000110

  总线：Cortex-M4 AHB

  名称：DE3_RDBAR

描述：DMA 引擎 3 接收描述符的基址。

表 8-161. DE3_RDBAR 寄存器

位	名称	类型	默认值	描述
31:0	de3_rdbar	R/W	0x00000000	DMA 引擎 3 接收描述符的基址。

8.13.3.50. DE3_CRDAR 寄存器

- 地址：0x59000114

  总线：Cortex-M4 AHB

  名称：DE3_CRDAR

  描述：DMA 引擎 3 接收描述符的当前地址。

表 8-162. DE3_CRDAR 寄存器

位	名称	类型	默认值	描述
31:0	de3_crdar	R/W	0x00000000	DMA 引擎 3 接收描述符的当前地址。

8.13.3.51. DE3_RDR 寄存器

- 地址：0x59000118

  总线：Cortex-M4 AHB

  名称：DE3_RDR

  描述：DMA 引擎 3 接收请求寄存器。

表 8-163. DE3_RDR 寄存器

位	名称	类型	默认值	描述
31:1	-	R	0x0	保留。
0	de3_rdr	R/W	0x00000000	任何写操作都会触发 TXE 进入状态 RXE_RD_DESC.

8.13.3.52. ECHO_CSR_CR 寄存器

- 地址：0x59000120

  总线：Cortex-M4 AHB

  名称：ECHO_CSR_CR

描述：CSR 访问控制寄存器。

表 8-164. ECHO_CSR_CR 寄存器

位	名称	类型	默认值	描述
31:12	-	R	0x0	保留。
11:10	dly_rd	R/W	0x1	读延时
9:8	dly_wr	R/W	0x1	写延时
7:1	-	R	0x0	保留。
0	mux_access	R/W	0x0	0：通过 csr_done（默认）。 1：通过延时计数器

8.13.3.53. ECHO_DMA_SWRSTCR 寄存器

- 地址：0x59000130

  总线：Cortex-M4 AHB

  名称：ECHO_DMA_SWRSTCR

  描述：ECHO DMA 软复位控制寄存器

表 8-165. ECHO_DMA_SWRSTCR 寄存器

位	名称	类型	默认值	描述
31:16	dma_swrst_pattern	R/W	0x0	0x5AA5 以启用软件重置。
15:4	-	R	0x0	保留。
3	dma_rxe2_swrstn	R/W	0x1	软复位 DMA RX engine2 hclk 域。
2	dma_txe2_swrstn	R/W	0x1	软复位DMA TX engine2 hclk域。
1	dma_rxe1_swrstn	R/W	0x1	软复位 DMA RX engine1/3 hclk 域。
0	dma_txe1_swrstn	R/W	0x1	软复位 DMA TX engine1/3 hclk 域。

8.13.3.54. ECHO_RXENG 寄存器

- 地址：0x59000140

  总线：Cortex-M4 AHB

  名称：ECHO_DMA_SWRSTCR

描述：ECHO RXENG MUX CSR。

表 8-166. ECHO_RXENG 寄存器

位	名称	类型	默认值	描述
31:10	-	R	0x0	保留。
9	rxe_reclaim_en3	R/W	0x0	（ECHO）当状态=discard 时回收所有使用的 rx 数据包 0：A1 当使用多个描述时，将回收前一个错误描述。 1：ECHO。当使用多个描述时，回收所有使用的描述。
8	rxcmplastdesc3	R/W	0x0	（ECHO）仅在获得pktend时声明接收完成中断。 0：A1。将为每个接收描述符断言接收完成 1：ECHO。仅在数据包的最后一个描述符断言接受完成中断
7	rxe_reclaim_en2	R/W	0x1	（ECHO）当状态=discard 时回收所有使用的 rx 数据包 0：A1 当使用多个描述时，将回收前一个错误描述。 1：ECHO。当使用多个描述时，回收所有使用的描述。
6	rxcmplastdesc2	R/W	0x1	（ECHO）仅在获得pktend时声明接收完成中断。 0：A1。将为每个接收描述符断言接收完成 1：ECHO。仅在数据包的最后一个描述符断言接受完成中断
5	rxe_reclaim_en1	R/W	0x0	（ECHO）当状态=discard 时回收所有使用的 rx 数据包 0：A1 当使用多个描述时，将回收前一个错误描述。 1：ECHO。当使用多个描述时，回收所有使用的描述。
4	rxcmplastdesc1	R/W	0x0	（ECHO）仅在获得pktend时声明接收完成中断。 0：A1。将为每个接收描述符断言接收完成 1：ECHO。仅在数据包的最后一个

位	名称	类型	默认值	描述
				描述符断言接受完成中断
3	-	R	0x0	保留。
2	mux_denpost3	R/W	0x0	RXENG3 1：A1； 0：Echo，增加 den_post_r 检测数据输入（默认）
1	mux_denpost2	R/W	0x0	RXENG2. 1：A1； 0：Echo，增加 den_post_r 检测数据输入（默认）
0	mux_denpost1	R/W	0x0	RXENG1。 1：A1； 0：Echo，增加den_post_r检测数据输入（默认）

8.13.3.55. ECHO_RSVD_CSR 寄存器

- 地址：0x590001F0

  总线：Cortex-M4 AHB

  名称：ECHO_RSVD_CSR

  描述：保留给 ECO。

表 8-167. ECHO_RSVD_CSR 寄存器

位	名称	类型	默认值	描述
31	dma_rsvd_b31	R/W	0x0	保留给 ECO。
30	dma_rsvd_b30	R/W	0x0	保留给 ECO。
29	dma_rsvd_b29	R/W	0x0	保留给ECO。
28	dma_rsvd_b28	R/W	0x0	保留给ECO。
27	dma_rsvd_b27	R/W	0x0	保留给ECO。
26	dma_rsvd_b26	R/W	0x0	保留给 ECO。
25	dma_rsvd_b25	R/W	0x0	保留给ECO。
24	dma_rsvd_b24	R/W	0x0	保留给 ECO。
23	dma_rsvd_b23	R/W	0x0	保留给 ECO。

位	名称	类型	默认值	描述
22	dma_rsvd_b22	R/W	0x0	保留给 ECO。
21	dma_rsvd_b21	R/W	0x0	保留给 ECO。
20	dma_rsvd_b20	R/W	0x0	保留给ECO。
19	dma_rsvd_b19	R/W	0x0	保留给ECO。
18	dma_rsvd_b18	R/W	0x0	保留给ECO。
17	dma_rsvd_b17	R/W	0x0	保留给 ECO。
16	dma_rsvd_b16	R/W	0x0	保留给ECO。
15	dma_rsvd_b15	R/W	0x0	保留给 ECO。
14	dma_rsvd_b14	R/W	0x0	保留给 ECO。
13	dma_rsvd_b13	R/W	0x0	保留给 ECO。
12	dma_rsvd_b12	R/W	0x0	保留给ECO。
11	dma_rsvd_b11	R/W	0x0	保留给 ECO。
10	dma_rsvd_b10	R/W	0x0	保留给 ECO。
9	dma_rsvd_b9	R/W	0x0	保留给 ECO。
8	dma_rsvd_b8	R/W	0x0	保留给ECO。
7	dma_rsvd_b7	R/W	0x0	保留给 ECO。
6	dma_rsvd_b6	R/W	0x0	保留给ECO。
5	dma_rsvd_b5	R/W	0x0	保留给 ECO。
4	dma_rsvd_b4	R/W	0x0	保留给ECO。
3	dma_rsvd_b3	R/W	0x0	保留给 ECO。
2	dma_rsvd_b2	R/W	0x0	保留给ECO。
1	dma_rsvd_b1	R/W	0x0	保留给 ECO。
0	dma_rsvd_b0	R/W	0x0	保留给ECO。

8.14. CRC

8.14.1. 概述

V6201X 集成两个 CRC 加速器，支持 CRC16，CRC24，和 CRC32。一个挂在 Cortex-M0 的 AHB 总线上，另一个挂在 Cortex-M4 的 AHB 总线上。

8.14.2. CRC 加速器寄存器（Cortex-M4）

8.14.2.1. CRC_MODE 寄存器

- 地址：0x58000000

  总线：Cortex-M4 AHB

  寄存器名：CRC_MODE

  描述：CRC 加速器模式

表 8-168. CRC_MODE 寄存器（地址：0x58000000）

位	名称	类型	默认值	描述
31:11	-	-	-	保留。
10	CRC_XOREN	R/W	0x0	设置对输出代码执行 0xFFFFFFFF 的异或操作。
9	CRC_bREV	R/W	0x0	设置为对输出代码执行位反转。
8	-	-	-	保留。
7	CRC_RESET	R/W1C	0x0	Set 复位 CRC 状态机。 该位将被 HW 清除。
6	CRC_FLAG	R	0x0	该位将在 CRC 计算完成后设置，并在 CRC 码被读出后由 HW 清除。
5	CRC_MSBIN	R/W	0x0	设置将每个字节的 MSB 首先转移到 CRC 引擎。
4:0	CRC_TYPE	R/W	0x04	执行 CRC（CRC_TYPE+1）计算，仅支持 3 种类型： 0x0F：CRC16 0x17：CRC24 0x1F：CRC32

8.14.2.2. CRC_POLY 寄存器

- 地址：0x58000004

  总线：Cortex-M4 AHB

  名称：CRC_POLY

  描述：CRC 多项式系数。

表 8-169. CRC_POLY 寄存器（地址：0x58000004）

位	名称	类型	默认值	描述
31:0	CRC_POLY	R/W	0x00001021	CRC 计算的多项式系数。 例： CRC16：G(x)=x^16+x^12+x^5+1 CRC_POLY[15:0]=0x00001021

8.14.2.3. CRC_CODE 寄存器

- 地址：0x58000008

  总线：Cortex-M4 AHB

  名称：CRC_CODE

  描述：CRC 输出码

表 8-170. CRC_CODE 寄存器（地址：0x58000008）

位	名称	类型	默认值	描述
31:0	CRC_CODE	R	0x00000000	CRC 输出代码。 FW 需要自己指定所需的位范围。以 CRC16 为例，取 CRC_CODE[15:0]。

8.14.2.4. CRC_ENTRY 寄存器

- 地址：0x5800000C

  总线：Cortex-M4 AHB

  名称：CRC_ENTRY

  描述：被转移到 CRC 引擎的数据的入口点地址。

表 8-171. CRC_ENTRY 寄存器（地址：0x5800000C）

位	名称	类型	默认值	描述
31:0	CRC_ENTRY	W	0x00000000	被转移到 CRC 引擎中的数据的入口点地址。

8.14.2.5. CRC_bLEN 寄存器

- 地址：0x58000010

  总线：Cortex-M4 AHB

  名称：CRC_bLEN

  描述：CRC 计算的总比特长度。

表 8-172. CRC_bLEN 寄存器（地址：0x58000010）

位	名称	类型	默认值	描述
31:24	-	-	-	保留。
23:0	CRC_bLEN	R/W	0x00000000	CRC 计算的总比特长度。

8.14.2.6. CRC_INIT_DATA 寄存器

- 地址：0x58000014

  总线：Cortex-M4 AHB

  名称：CRC_INIT_DATA

  描述：CRC 计算的初始数据。

表 8-173. CRC_INIT_DATA 寄存器（地址：0x58000014）

位	名称	类型	默认值	描述
31:0	CRC_INIT_DATA	R/W	0x00000000	CRC 计算的初始数据。

8.14.3.CRC 加速器寄存器（Cortex-M0）

8.14.3.1. CRC_MODE 寄存器

- 地址：0x7000F000

  总线：Cortex-M0 AHB

  名称：CRC_MODE

描述：CRC 计算模式。

表 8-174. CRC_MODE 寄存器（地址：0x7000F000）

位	名称	类型	默认值	描述
31:11	-	-	-	保留。
10	CRC_XOREN	R/W	0x0	设置为对输出代码执行 0xFFFFFFFF 的异或操作。
9	CRC_bREV	R/W	0x0	设置为对输出代码执行位反转。
8	-	-	-	保留。
7	CRC_RESET	R/W1C	0x0	设置复位 CRC 状态机。 该位将被 HW 清除。
6	CRC_FLAG	R	0x0	该位将在 CRC 计算完成后设置，并在 CRC 码被读出后由 HW 清除。
5	CRC_MSBIN	R/W	0x0	设置将每个字节的 MSB 首先转移到 CRC 引擎。
4:0	CRC_TYPE	R/W	0x04	执行 CRC(CRC_TYPE+1)计算，仅支持 3 种类型： 0x0F：CRC16 0x17：CRC24 0x1F：CRC32

8.14.3.2. CRC_POLY 寄存器

- 地址：0x7000F004

  总线：Cortex-M0 AHB

  名称：CRC_POLY

  描述：CRC 多项式系数。

表 8-175. CRC_POLY 寄存器（地址：0x7000F004）

位	名称	类型	默认值	描述
31:0	CRC_POLY	R/W	0x00001021	CRC 多项式系数。 例： CRC16：G(x)=x^16+x^12+x^5+1 CRC_POLY[15:0]=0x00001021

8.14.3.3. CRC_CODE 寄存器

- 地址：0x7000F008

  总线：Cortex-M0 AHB

  名称：CRC_CODE

  描述：CRC 输出码。

表 8-176. CRC_CODE 寄存器（地址：0x7000F008）

位	名称	类型	默认值	描述
31:0	CRC_CODE	R	0x00000000	CRC 输出代码。 FW 需要自己指定所需的位范围。 以 CRC16 为例，取 CRC_CODE[15:0]。

8.14.3.4. CRC_ENTRY 寄存器

- 地址：0x7000F00C

  总线：Cortex-M0 AHB

  名称：CRC_ENTRY

  描述：被转移到 CRC 引擎的数据的入口点地址。

表 8-177. CRC_ENTRY 寄存器（地址：0x7000F00C）

位	名称	类型	默认值	描述
31:0	CRC_ENTRY	W	0x00000000	被转移到 CRC 引擎的数据的入口点地址。

8.14.3.5. CRC_bLEN 寄存器

- 地址：0x7000F010

  总线：Cortex-M0 AHB

  名称：CRC_bLEN

  描述：CRC 计算的总比特长度。

表 8-178. CRC_bLEN 寄存器（地址：0x7000F010）

位	名称	类型	默认值	描述
31:24	-	-	-	保留。
23:0	CRC_bLEN	R/W	0x00000000	CRC 计算的总比特长度。

8.14.3.6. CRC_INIT_DATA 寄存器

- 地址：0x7000F014

  总线：Cortex-M0 AHB

  名称：CRC_INIT_DATA

  描述：CRC 计算的初始数据。

表 8-179. CRC_INIT_DATA 寄存器（地址：0x7000F014）

位	名称	类型	默认值	描述
31:0	CRC_INIT_DATA	R/W	0x00000000	CRC 计算的初始数据。

8.15. SPI Flash 控制器

8.15.1. 概述

SPI 闪存控制器用于控制外部 SPI flash 的访问和编程。SPI 闪存控制器在命令模式或存储器模式下支持多种类型的传输模式。并且，cache 控制器可提高代码执行性能。

8.15.2. 功能框图

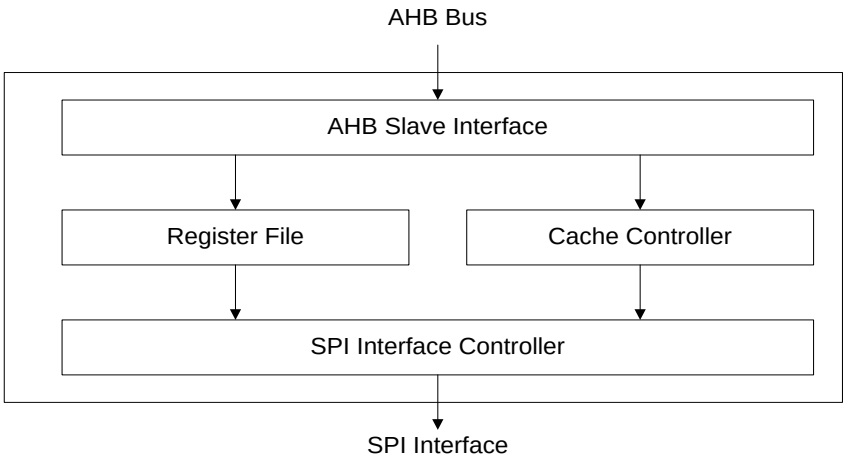


图 8-26. SPI 闪存控制器功能框图

8.15.3. 寄存器摘要

表 8-180. SPI 闪存控制器的寄存器列表

基地址: 0x60FFF000

地址	名称	类型	默认值	描述
0x000	FLASH_CMD	R/W	0x00000000	Flash 控制器命令寄存器
0x004	FLASH_ADDR	R/W	0x00000000	Flash 控制器地址寄存器
0x008	FLASH_CFG	R/W	0x00000733	Flash 控制器配置寄存器
0x00C	FLASH_CACHE	R/W	0x00000001	Flash cache 配置寄存器
0x010	FLASH_SR	R	-	Flash 控制器状态寄存器
0x014	FLASH_ID	R	-	Flash 控制器 ID 寄存器
0x018	FLASH_CACHEHIT	R	-	Flash 控制器 cache 命中率计数数字
0x020	FLASH_MAPOFFSET	R/W	0x00000000	Flash 启动偏移

0x100	FLASH_BUF0	R/W	0x00000000	用于 Flash 读/写的 Buffer0
0x104	FLASH_BUF1	R/W	0x00000000	用于 Flash 读/写的 Buffer1
0x108	FLASH_BUF2	R/W	0x00000000	用于 Flash 读/写的 Buffer2
…	…	…	…	…
0x1FC	FLASH_BUF63	R/W	0x00000000	用于 Flash 读/写的 Buffer63

8.15.4. 寄存器描述

8.15.4.1. FLASH_CMD 寄存器

- 地址：0x60FFF000

  总线：AHB

  名称：FLASH_CMD

  描述：flash 控制器的命令

表 8-181. FLASH_CMD 寄存器（地址：0x60FFF000）

位	名称	类型	默认值	描述
31	ACT	R/W	0x0	Flash 命令有效寄存器。向该寄存器写入 1 将启动在 FLASH_CMD 寄存器中定义的 flash 命令。命令处理完成后，该位将自动清除为 0。因此，程序员可以查询该寄存器以确认命令是否完成。
30:25	-	-	-	保留。
24	POLL	R/W	0x0	Flash 轮询类型命令控制寄存器。 0：非轮询类型命令。 1：轮询类型命令。 对于诸如扇区擦除或页面编程之类的轮询类型命令，BSY 位即 SPI flash 的状态寄存器将被置为有效，直到操作完成。在繁忙期间，如果用户读取存储区，如执行程序代码，将获得未知值，这将导致异常行为。对于这种情况，程序员可以将该位置 1 以启用硬件自动轮询模式，硬件将轮询 BSY 状态寄存器，直到清除了 BSY 位。在轮询期间，内存访问将一直保留到轮询完成为止。这对于应用程序内程序（IAP）非常有用。
23:16	LENGTH	R/W	0x00	命令数据长度寄存器。该寄存器用于控制读取或写入命令的数据长度。该寄存器仅在 CMDDATA 为 1

				时有效。 数据长度 = (LENGTH[7:0] + 1) 个字节
15	-	-	-	保留。
14:12	CMDMODE	R/W	0x0	命令模式控制寄存器。 0：1I1O 模式 1：1I1O 快速模式 2：1I2O 模式 3：2IO 模式 4：1I4O 模式 5：4IO 模式 其他：保留 对于页面编程命令，1I2O 或 1I4O 模式意味着地址部分是 1 位模式，而程序数据部分是 2 位或 4 位模式。如果选择了 2IO 或 4IO 模式，则意味着地址部分和数据部分都是 2 位或 4 位输入。应该注意的是，并非所有的 SPI 闪存都支持 2IO 或 4IO 模式页面编程命令。
11	CMDADDR4	R/W	0x0	地址字节数控制寄存器，仅当 CMDADDR 设置为 1 时，此寄存器才有效。 0：3 个字节的地址用于命令访问 1：4 个字节的地址用于命令访问
10	CMDADDR	R/W	0x0	命令地址控制。 0：命令发送时没有地址 1：命令与地址一起发送
9	CMDWR	R/W	0x0	该寄存器仅在 CMDDATA 为 1 时有效。 0：读命令 1：写命令
8	CMDDATA	R/W	0x0	命令数据控制。 0：发送不带数据的命令 1：发送带有数据的命令
7:0	CMDID	R/W	0x0	命令 ID 寄存器。 该寄存器用于控制发送到 SPI flash 的实际命令 ID。在将 ACT 位置 1 之前，必须正确设置每个不同 CMDID 的组合设置。

8.15.4.2. FLASH_ADDR 寄存器

- 地址：0x60FFF004

  总线：Cortex-M4 AHB

  名称：FLASH_ADDR

  描述：Flash 控制器的 SPI 协议中的地址字段。

表 8-182. FLASH_ADDR 寄存器（地址：0x60FFF004）

位	名称	类型	默认值	描述
31:0	ADDR	R/W	0x00000000	当 CMDADDR 位设置为 1 时，此寄存器用于控制 SPI 命令中的地址字段。当 CMDADDR4 为 0 时，将仅使用 ADDR [23:0]，而当 CMDADDR4 设置为 1，使用所有 ADDR [31:0]。

8.15.4.3. FLASH_CFG 寄存器

- 地址：0x60FFF008

  总线：Cortex-M4 AHB

  名称：FLASH_CFG

  描述：Flash 控制器的配置。

表 8-183. FLASH_CFG 寄存器（地址：0x60FFF008）

位	名称	类型	默认值	描述
31:28	-	-	-	保留。
27:24	CSNWIDTH	R/W	0x8	SPI 协议的 CSN 的高状态宽度（以周期为单位）。 0：1 个 HCLK 周期 1：2 个 HCLK 周期 2：3 个 HCLK 周期 … 15：16 个 HCLK 周期
23	MEMADDR4	R/W	0x0	内存访问地址字节数控制。 0：3 个字节的地址用于内存访问 1：4 个字节的地址用于内存访问控制
22	-	-	-	保留。

21	PERFENHC	R/W	0x0	性能增强控制。 仅当 TYPE 设置为 1 且 MEMMODE 设置为 5 时，该位才有效。当该位设置为 1 时，硬件将自动进入 SPI 性能模式，并跳过随后的 SPI 命令字节进行的存储器访问。当任何其他命令访问处于活动状态时，它将自动退出性能增强模式。 0：禁用性能增强模式 1：启用性能增强模式
20	-	-	-	保留。
19:18	MEMLAT	R/W	0x0	存储器访问数据锁存时序控制。 0：在内部 SSPCLK 的下降沿。 1：内部 SSPCLK 下降沿后 0.5 个周期 2：内部 SSPCLK 下降沿后 1 个周期 3：内部 SSPCLK 下降沿后 1.5 个周期
17:16	CMDLAT	R/W	0x0	命令访问数据锁存时序控制。 0：在内部 SSPCLK 的下降沿。 1：内部 SSPCLK 下降沿后 0.5 个周期 2：内部 SSPCLK 下降沿后 1 个周期 3：内部 SSPCLK 下降沿后 1.5 个周期
15	-	-	-	保留。
14:12	MEMMODE	R/W	0x0	存储器模式控制寄存器。 0：1I1O 模式 1：1I1O 快速模式 2：1I2O 模式 3：2IO 模式 4：1I4O 模式 5：4IO 模式 其他：保留
11	-	-	-	保留。
10:8	SIZE	R/W	0x7	Flash 大小控制寄存器。 0：512 Kbits. 64KB 1：1 Mbits. 128KB

				2：2 Mbits. 256KB 3：4 Mbits. 512KB 4：8 Mbits. 1MB 5：16 Mbits. 2MB 6：32 Mbits. 4MB 7：64 Mbits. 8MB
7:4	MEMCLK	R/W	0x3	内存访问时钟速度控制。 0：HCLK/1 1：HCLK/2 2：HCLK/3 … 15：HCLK/16
3:0	CMDCLK	R/W	0x3	命令访问时钟速度控制。 0：HCLK 1：HCLK/2 2：HCLK/3 … 15：HCLK/16

8.15.4.4. FLASH_CACHE 寄存器

- 地址：0x60FFF00C

  总线：Cortex-M4 AHB

  名称：FLASH_CACHE

  描述：flash cache 配置控制寄存器

表 8-184. FLASH_CACHE 寄存器（地址：0x60FFF00C）

位	名称	类型	默认值	描述
31:3	-	-	-	保留。
2	CACHE_HITEN	R/W	0x0	Cache 命中率计算使能寄存器。 0：关闭 1：使能

位	名称	类型	默认值	描述
1	CACHE_SIZE	R/W	0x0	Cache 大小控制寄存器。 0：8KB 1：4KB
0	CACHE_EN	R/W	0x1	Cache 使能控制寄存器。 0：关闭 1：使能 默认情况下，启用缓存。如果 flash 中的任何内容通过擦除或编程命令进行了更新，则必须禁用 cache 并再次启用它，以防止出现脏数据问题。

8.15.4.5. FLASH_SR 寄存器

- 地址：0x60FFF010

  总线：Cortex-M4 AHB

  名称：FLASH_SR

  描述：flash 控制器的状态寄存器。

表 8-185. FLASH_SR 寄存器（地址：0x60FFF010）

位	名称	类型	默认值	描述
31:8	-	-	-	保留。
7:0	SR	R	-	SPI flash 状态寄存器。 该寄存器显示最新的 SPI flash 状态，该寄存器的任何读取都会触发 SPI 总线上的读取状态命令，并在此寄存器上显示状态。

8.15.4.6. FLASH_ID 寄存器

- 地址：0x60FFF014

  总线：Cortex-M4 AHB

  名称：FLASH_ID

  描述：flash 控制器下发的 flash 命令 0x9F 的 flash JEDEC ID。

表 8-186. FLASH_ID 寄存器（地址：0x60FFF014）

位	名称	类型	默认值	描述

位	名称	类型	默认值	描述
31:24	-	-	-	保留。
23:16	MANU_ID	R	-	SPI flash 制造商 JEDEC ID 寄存器。 该寄存器显示 SPI flash 的制造商 ID，对该寄存器的任何读取都会触发 SPI 总线上的读取 ID 命令 0x9F，并在该寄存器上显示制造商 ID。
15:0	DEV_ID	R	-	SPI flash 设备 JEDEC ID 寄存器。 该寄存器显示 SPI flash 的设备 ID，对该寄存器的任何读取都会触发 SPI 总线上的读取 ID 命令 0x9F，并在该寄存器上显示设备 ID。

8.15.4.7. FLASH_CACHEHIT 寄存器

- 地址：0x60FFF018

  总线：Cortex-M4 AHB

  名称：FLASH_CACHEHIT

  描述：cache 控制器的命中率。

表 8-187. FLASH_CACHEHIT 寄存器（地址：0x60FFF018）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:0	HITRATE	R	-	Cache 命中率状态寄存器。 当 CACHE_HITEN 位置 1 时，内部 cache 命中率计算电路将处于活动状态。该寄存器存储每 65536 次访问中命中 cache 的次数。

8.15.4.8. FLASH_MAPOFFSET 寄存器

- 地址：0x60FFF020

  总线：Cortex-M4 AHB

  名称：FLASH_MAPOFFSET

  描述：Flash 启动按偏移量重新映射. FLASH_MAPOFFSET 将被映射到新的 0x0000_0000，并且 0x0000_0000 ∽（BootStartPos-1）之间的块将被重新映射到闪存的末尾，以实现地址包装功能。

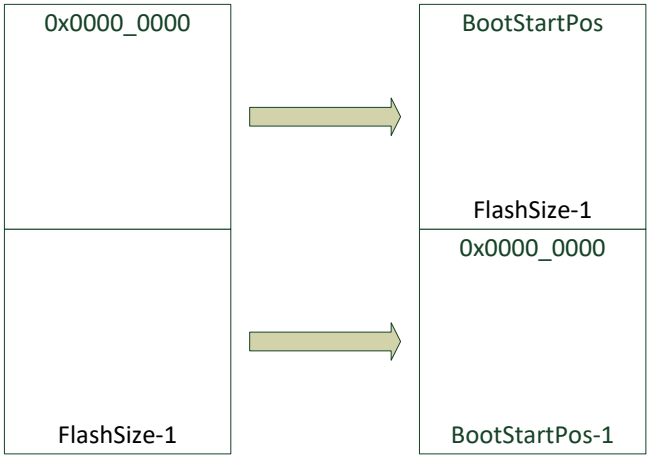


表 8-188. FLASH_MAPOFFSET 寄存器（地址：0x60FFF020）

位	名称	类型	默认值	描述
31:20	-	-	-	保留。
19:0	FLASH_MAPOFFSET	R/W	0x00000000	Flash 启动将从以下位置物理启动： BootStartPos = bootup_pos << 4 Unit：16-Byte 该寄存器不能通过软件复位来复位。

注意：发出重新启动任务后，当重新映射偏移量 bootup_pos [19:0]设置为非零值时，应将 FLASH_CFG [10:8]中的闪存大小写入正确的值。

8.15.4.9. FLASH_BUFn 寄存器

- 地址：0x60FFF100 ~ 0x60FFF1FC

  总线：Cortex-M4 AHB

  名称：FLASH_BUF*n*, *n*：0~63

  描述：Flash 控制器的第 n 个缓冲区，用于 SPI Flash 数据访问

表 8-189. FLASH_BUFn 寄存器（地址：0x60FFF100 ~ 0x60FFF1FC）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:0	BUFn n：0~63	R/W	-	Flash 读/写缓冲寄存器 0∽63。 对于带有数据的命令，这些缓冲区用于存储发送数据或接收数据。对于写操作，必须在 ACT 位置 1 之前将发送数据写入 BUFx。发送顺序为 BUF0 [7:0] => BUF0 [15:8] => BUF0 [23:16] => BUF0 [31:24] =>

位	名称	类型	默认值	描述
				BUF1 [7:0] =>……。=> BUF63 [31:24]。对于读取操作，在将 ACT 位清零之后，接收数据将以与发送序列相同的顺序放入 BUFx 中。

8.16. 过零检测

8.16.1. 概述

8.16.2. 寄存器摘要

表 8-190. 过零检测寄存器列表

基地址: 0x70000000

地址	名称	类型	默认值	描述
0x000	PH0_CTRL	R/W	0x00000000	Phase 0 控制寄存器。
0x004	PH0_ZTIME0	R	0x00000000	Phase 0 基于 25MHz 的过零时间计数器 0。
0x008	PH0_ZTIME1	R	0x00000000	Phase 0 基于 25MHz 的过零时间计数器 1。
0x00C	PH0_ZTIME2	R	0x00000000	Phase 0 基于 25MHz 的过零时间计数器 2。
0x010	PH0_ZTIME3	R	0x00000000	Phase 0 基于 25MHz 的过零时间计数器 3。
0x014	PH1_CTRL	R/W	0x00000000	Phase 1 控制寄存器。
0x018	PH1_ZTIME0	R	0x00000000	Phase 1 基于 25MHz 的过零时间计数器 0。
0x01C	PH1_ZTIME1	R	0x00000000	Phase 1 基于 25MHz 的过零时间计数器 1。
0x020	PH1_ZTIME2	R	0x00000000	Phase 1 基于 25MHz 的过零时间计数器 2。
0x024	PH1_ZTIME3	R	0x00000000	Phase 1 基于 25MHz 的过零时间计数器 3。
0x028	PH2_CTRL	R/W	0x00000000	Phase 2 控制寄存器。
0x02C	PH2_ZTIME0	R	0x00000000	Phase 2 基于 25MHz 的过零时间计数器 0。
0x030	PH2_ZTIME1	R	0x00000000	Phase 2 基于 25MHz 的过零时间计数器 1。
0x034	PH2_ZTIME2	R	0x00000000	Phase 2 基于 25MHz 的过零时间计数器 2。

0x038	PH2_ZTIME3	R	0x00000000	Phase 2 基于 25MHz 的过零时间计数器 3。
0x03C	ZCDPHSEL	R	0x00000000	PHY 的 TDP 过零相位选择。

8.16.3. 寄存器描述

8.16.3.1. PH0_CTRL 寄存器

- 地址：0x70000000

  总线：Cortex-M0 AHB

  名称：PH0_CTRL

  描述：Phase 0 控制寄存器。

表 8-191. PH0_CTRL 寄存器（地址：0x70000000）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:12	PH0_INTBL	R	0x0	Phase0 中断嵌套表。
11:9	-	-	0x0	保留。
8	PH0_INTR	R	0x0	Phase0 中断。
7	PH0_INTFLU	R/W	0x0	设置为刷新嵌套中断表。
6	PH0_INTCLR	R/W	0x0	设置为清除中断。
5:4	PH0_MODE	R/W	0x0	Phase0 检测模式。 0x0：上升沿过零检测 0x1：下降沿过零检测 0x2：双边沿过零检测
3:2	PH0_ID	R	0x0	当前记录索引。
1	PH0_INTEN	R/W	0x0	设置为使能过零触发 ISR。
0	PH0_EN	R/W	0x0	设置为使能 HW 捕捉过零检测。

8.16.3.2. PH0_ZTIME0 寄存器

- 地址：0x70000004

  总线：Cortex-M0 AHB

名称：PH0_ZTIME0

描述：基于 25MHz 计数器的过零时间。

表 8-192. PH0_ZTIME0 寄存器（地址：0x70000004）

位	名称	类型	默认值	描述
31:0	PH0_ZTIME0	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.3. PH0_ZTIME1 寄存器

- 地址：0x70000008

  总线：Cortex-M0 AHB

  名称：PH0_ZTIME1

  描述：基于 25MHz 计数器的过零时间。

表 8-193. PH0_ZTIME1 寄存器（地址：0x70000008）

位	名称	类型	默认值	描述
31:0	PH0_ZTIME1	R	0x00000000	描述：基于 25MHz 计数器的过零时间。

8.16.3.4. PH0_ZTIME2 寄存器

- 地址：0x7000000C

  总线：Cortex-M0 AHB

  名称：PH0_ZTIME2

  描述：基于 25MHz 计数器的过零时间。

表 8-194. PH0_ZTIME2 寄存器（地址：0x7000000C）

位	名称	类型	默认值	描述
31:0	PH0_ZTIME2	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.5. PH0_ZTIME3 寄存器

- 地址：0x70000010

  总线：Cortex-M0 AHB

  名称：PH0_ZTIME3

  描述：基于 25MHz 计数器的过零时间。

表 8-195. PH0_ZTIME3 寄存器（地址：0x70000010）

位	名称	类型	默认值	描述
31:0	PH0_ZTIME3	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.6. PH1_CTRL 寄存器

- 地址：0x70000014

  总线：Cortex-M0 AHB

  名称：PH1_CTRL

  描述：Phase 1 控制寄存器。

表 8-196. PH1_CTRL 寄存器（地址：0x70000014）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:12	PH1_INTBL	R	0x0	Phase1 中断嵌套表。
11:9	-	-	0x0	保留。
8	PH1_INTR	R	0x0	Phase1 中断。
7	PH1_INTFLU	R/W	0x0	设置为刷新嵌套中断表。
6	PH1_INTCLR	R/W	0x0	设置为清除中断。
5:4	PH1_MODE	R/W	0x0	Phase1 检测模式。 0x0：上升沿过零检测 0x1：下降沿过零检测 0x2：双边沿过零检测
3:2	PH1_ID	R	0x0	当前记录索引。
1	PH1_INTEN	R/W	0x0	设置为使能过零触发 ISR。
0	PH1_EN	R/W	0x0	设置为使能 HW 捕捉过零检测。

8.16.3.7. PH1_ZTIME0 寄存器

- 地址：0x70000018

  总线：Cortex-M0 AHB

  名称：PH1_ZTIME0

  描述：基于 25MHz 计数器的过零时间。

表 8-197. PH1_ZTIME0 寄存器（地址：0x70000018）

位	名称	类型	默认值	描述
31:0	PH1_ZTIME0	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.8. PH1_ZTIME1 寄存器

- 地址：0x7000001C

  总线：Cortex-M0 AHB

  名称：PH1_ZTIME1

  描述：基于 25MHz 计数器的过零时间。

表 8-198. PH1_ZTIME1 寄存器（地址：0x7000001C）

位	名称	类型	默认值	描述
31:0	PH1_ZTIME1	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.9. PH1_ZTIME2 寄存器

- 地址：0x70000020

  总线：Cortex-M0 AHB

  名称：PH1_ZTIME2

  描述：基于 25MHz 计数器的过零时间。

表 8-199. PH1_ZTIME2 寄存器（地址：0x70000020）

位	名称	类型	默认值	描述
31:0	PH1_ZTIME2	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.10. PH1_ZTIME3 寄存器

- 地址：0x70000024

  总线：Cortex-M0 AHB

  名称：PH1_ZTIME3

  描述：基于 25MHz 计数器的过零时间。

表 8-200. PH1_ZTIME3 寄存器（地址：0x70000024）

位	名称	类型	默认值	描述

31:0	PH1_ZTIME3	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.11. PH2_CTRL 寄存器

- 地址：0x70000028

  总线：Cortex-M0 AHB

  名称：PH2_CTRL

  描述：Phase 2 控制寄存器。

表 8-201. PH2_CTRL 寄存器（地址：0x70000028）

位	名称	类型	默认值	描述
31:16	-	-	-	保留。
15:12	PH2_INTBL	R	0x0	Phase2 中断嵌套表。
11:9	-	-	0x0	保留。
8	PH2_INTR	R	0x0	Phase2 中断。
7	PH2_INTFLU	R/W	0x0	设置为刷新嵌套中断表。
6	PH2_INTCLR	R/W	0x0	设置为清除中断。
5:4	PH2_MODE	R/W	0x0	Phase2 检测模式。 0x0：上升沿过零检测 0x1：下降沿过零检测 0x2：双边沿过零检测。
3:2	PH2_ID	R	0x0	当前记录索引。
1	PH2_INTEN	R/W	0x0	设置为使能过零触发 ISR。
0	PH2_EN	R/W	0x0	设置为使能 HW 捕捉过零检测。

8.16.3.12. PH2_ZTIME0 寄存器

- 地址：0x7000002C

  总线：Cortex-M0 AHB

  名称：PH2_ZTIME0

  描述：基于 25MHz 计数器的过零时间。

表 8-202. PH2_ZTIME0 寄存器（地址：0x7000002C）

位	名称	类型	默认值	描述
31:0	PH2_ZTIME0	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.13. PH2_ZTIME1 寄存器

- 地址：0x70000030

  总线：Cortex-M0 AHB

  名称：PH2_ZTIME1

  描述：基于 25MHz 计数器的过零时间。

表 8-203. PH2_ZTIME1 寄存器（地址：0x70000030）

位	名称	类型	默认值	描述
31:0	PH2_ZTIME1	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.14. PH2_ZTIME2 寄存器

- 地址：0x70000034

  总线：Cortex-M0 AHB

  名称：PH2_ZTIME2

  描述：基于 25MHz 计数器的过零时间。

表 8-204. PH2_ZTIME2 寄存器（地址：0x70000034）

位	名称	类型	默认值	描述
31:0	PH2_ZTIME2	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.15. PH2_ZTIME3 寄存器

- 地址：0x70000038

  总线：Cortex-M0 AHB

  名称：PH2_ZTIME3

  描述：基于 25MHz 计数器的过零时间。

表 8-205. PH2_ZTIME3 寄存器（地址：0x70000038）

位	名称	类型	默认值	描述
31:0	PH2_ZTIME3	R	0x00000000	基于 25MHz 计数器的过零时间。

8.16.3.16. ZCDPHSEL 寄存器

- 地址：0x7000003C

  总线：Cortex-M0 AHB

  名称：ZCDPHSEL

  描述：PHY 的 TDP 过零相位选择。

表 8-206. ZCDPHSEL 寄存器（地址：0x7000003C）

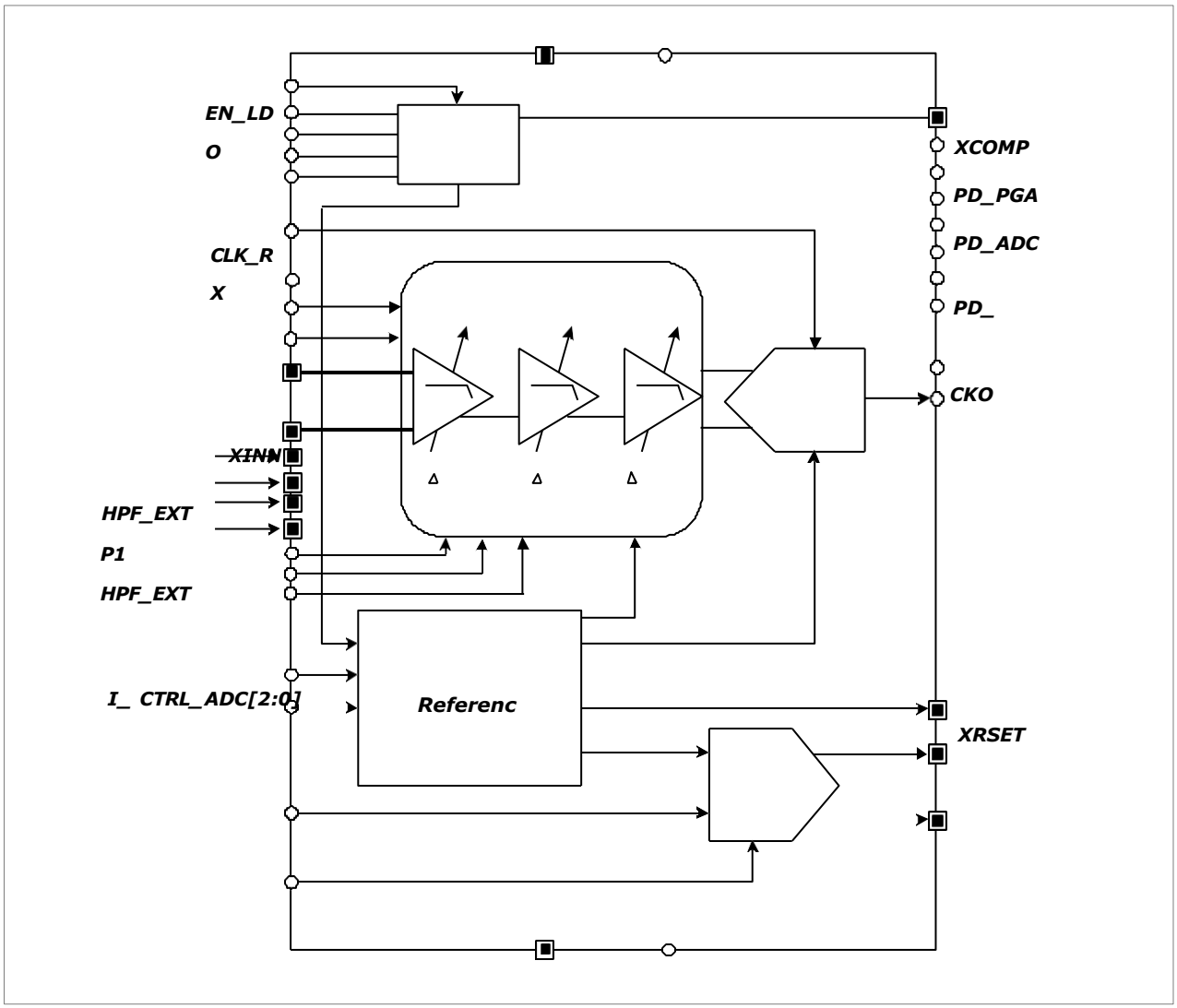
位	名称	类型	默认值	描述
31:2	-	-	-	保留。
1:0	ZCDPHSEL	R	0x0	PHY 的 TDP 过零检测的相位选择。 0：phase 0 1：phase 1 2：phase 2

8.17. 模拟前端

8.17.1. 概述

V6201X内置了用于收发器应用的混合信号模拟前端（AFE）。数字接口非常灵活，允许简单的接口支持半双工或全双工数据传输。该AFE支持ADC和DAC功能。省电模式包括减少单个功能块的功耗或关闭半双工应用中未使用的功能块的能力。TX信号路径是一个12位TXDAC。TXDAC提供差分电流输出，可直接将其输出到外部负载。

接收路径包括可编程放大器（RXPGA），可调低通滤波器（LPF），固定高通滤波器（HPF）和 12 位 ADC。低噪声 RXPGA 的可编程增益范围为-18 dB 至+54 dB，步进为 1 dB。接收路径的 LPF 截止频率可以设置为 5 MHz，7.5 MHz 或 12 MHz 左右。HPF 的截止频率可以设置为 100 kHz 左右，或者简单地绕过或通过外部电容设置。12 位 ADC 的动态性能高达 25 MSPS。RXPGA 和 ADC 均提供可扩展的功耗，可实现功耗/性能优化。



8.17.2.特性

- UMC 40 nm LP / RVT Low-K 逻辑工艺
- 适用于宽带调制解调器的低成本 3.3 V CMOS AFE
- 包含更新速度为 100 MSPS 的 12 位 DAC
- 包含 50 MSPS 的 12 位 ADC
- 低噪声 RXPGA 范围：-18 dB∽+54 dB
- 三阶可编程低通滤波器
- 灵活的数字数据路径接口
- 各种掉电/降电模式

8.17.3. 功能描述

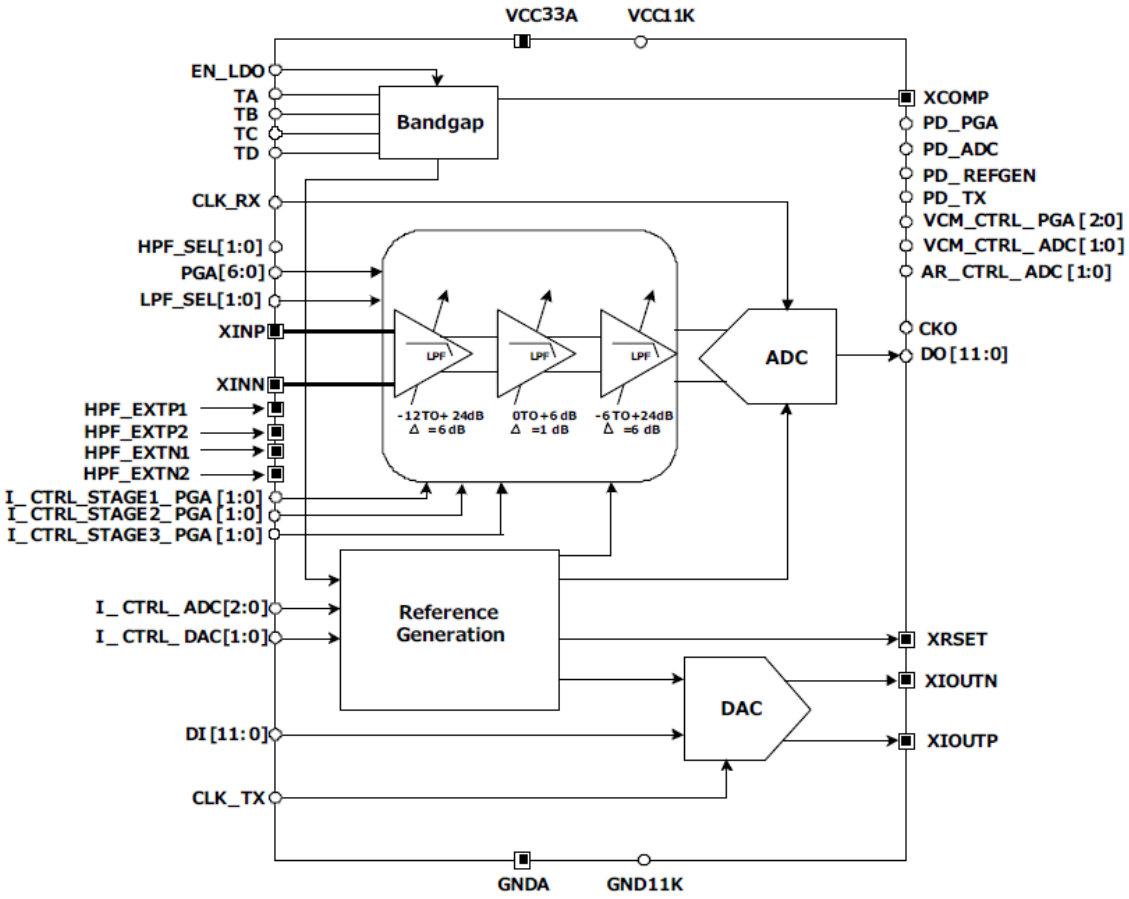


图 8-27. AFE 功能图

8.17.3.1. 总览

AFE 是一种混合信号模拟前端（AFE）IC，适用于需要 TX 路径和 RX 路径功能且数据速率分别高达 25 MSPS 和 25 MSPS 的收发器应用。

AFE 由 UMC 40 nm LP / RVT Low-K 逻辑工艺制造，以实现尺寸减小。它利用可编程增益放大器（PGA），其可编程增益范围为-18 dB 至+54 dB（步长为 1 dB），12 位流水线模数转换器（ADC）和 12 位数模转换器（DAC），以实现针对扫描仪，视频和 CMOS 成像应用的信号处理解决方案。

8.17.3.2. 省电模式

AFE 提供两种省电模式：掉电模式和待机模式。在掉电模式下，ADC / DAC / PGA 和内部基准电压源被禁用。因此，整个 IP 仅消耗泄漏电流和最小的功耗。启用此 IP 时，唤醒时间与外部旁路电容器的充电直接相关。待机模式的唤醒时间比掉电模式要短，通过禁用 ADC / DAC / PGA 以及待机模式可使内部基准保持开启状态。在这种情况下，AFE 将消耗少量电流用于正常偏置操作。

8.17.3.3. TX DAC 操作

DAC具有一对来自高阻抗电流源的差分模拟输出。这些输出能够直接驱动50Ω负载，例如双端接100Ω同轴电缆。如果需要单端输出，则另一个输出应直接连接至GNDA或通过50Ω电阻负载。

用户必须选择RSET值和输出电阻负载，以使输出电压水平降至符合的值，1V。高于上限的输出电压会严重降低AC性能。因此，在相同的电压信号摆幅的情况下，较低的电压信号电平是优选的。

8.17.3.4. PGA 增益控制操作

表 8-207. PGA 增益控制设置

PGA Gain （dB）	PGA[6:0]
-18	0x00
-17	0x01
-16	0x02
-15	0x03
-14	0x04
-13	0x05
-12	0x06
-11	0x07
-10	0x08
-9	0x09
-8	0x0A
-7	0x0B
-6	0x0C
-5	0x0D
-4	0x0E
-3	0x0F
-2	0x10
-1	0x11
0	0x12
1	0x13
2	0x14

3	0x15
4	0x16
5	0x17
6	0x18
7	0x19
8	0x1A
9	0x1B
10	0x1C
11	0x1D
12	0x1E
13	0x1F
14	0x20
15	0x21
16	0x22
17	0x23
18	0x24
19	0x25
20	0x26
21	0x27
22	0x28
23	0x29
24	0x2A
25	0x2B
26	0x2C
27	0x2D
28	0x2E
29	0x2F
30	0x30
31	0x31
32	0x32
33	0x33

34	0x34
35	0x35
36	0x36
37	0x37
38	0x38
39	0x39
40	0x3A
41	0x3B
42	0x3C
43	0x3D
44	0x3E
45	0x3F
46	0x40
47	0x41
48	0x42
49	0x43
50	0x44
51	0x45
52	0x46
53	0x47
54	0x48

8.17.4. 信号描述

8.17.4.1. 电源引脚

表 **8-208.** 电源引脚

信号	方向	描述
VCC33A	I	引脚电源 2.97 V~3.63 V 模拟电源

信号	方向	描述
		添加 10μF 和 0.1μF 电容用于去耦
GNDA	I	引脚地 模拟地 添加 10μF 和 0.1μF 电容用于去耦
VCC11K	I	内核电源 0.99 V～1.21 V 数字电源
GND11K	I	内核地 0.99 V～1.21 V 数字地

8.17.4.2. 信号引脚

表 8-209. 信号引脚

信号	方向	描述
XCOMP	IO	双向引脚 带宽/降噪节点 在 GNDA 中增加一个 3.3μF 电容（最小 ESR> 0.5Ω）以获得最佳性能。
XRSET	O	输出引脚 可以通过连接一个具有不同值的电阻器（RSET）来调节满量程输出电流。 满量程输出电流的计算公式如下：IOUTFS = [（4095/80）*（VREF / RSET）] 例如，要获得 10 mA 的满量程电流，外部 RSET 应为 6.2kΩ。
XIOUTP XIOUTN	O O	输出引脚 TX 的差分模拟输出 XIOUTP：TX 正电流输出 XIOUTN：TX 负电流输出
XINP XINN	I I	输入引脚 RX 的差分模拟输入 XINP：RX 正输入

信号	方向	描述
		XINN：RX 负输入
HPF_EXTP1 HPF_EXTP2	IO IO	双向引脚 高通滤波器外部引脚 HPF_EXTP1：PGA1 正输出 HPF_EXTP2：PGA2 正输入
HPF_EXTN1 HPF_EXTN2	IO IO	双向引脚 高通滤波器外部焊盘 HPF_EXTN1：PGA1 负输出 HPF_EXTN2：PGA2 负输入
PD_ADC	I	RX_ADC 掉电模式控制 0：正常运行模式（默认） 1：RX_ADC 掉电模式
PD_TX	I	TX 掉电模式控制 0：正常运行模式（默认） 1：TX 掉电模式
PD_PGA	I	RX_PGA 掉电模式控制 0：正常运行模式（默认） 1：RX_PGA 掉电模式
PD_REFGEN	I	REFGEN 掉电模式控制 0：正常运行模式（默认） 1：REFGEN 掉电模式
VCM_CTRL_PGA[2:0]	I	PGA VCM 驱动控制 VCM_CTRL_PGA [2:0] = 0b001（默认值）
AR_CTRL_ADC[1:0]	I	RX_ADC 模拟输入范围控制（VPP） 0x0：1.2V（默认）
VCM_CTRL_ADC[1:0]	I	RX_ADC 模拟输入共模控制 默认值：0x2
HPF_SEL[1:0]	I	高通滤波器控制 00：100 kHz 01：禁止

信号	方向	描述
		10：外部模式 11：关闭
LPF_SEL[1:0]	I	低通滤波器-3dB 截止频率选择信号 00：5 MHz 01：7.5 MHz 10：15 MHz 11：禁止
PGA[6:0]	I	PGA 增益控制
CLK_RX	I	RX 时钟输入
CLK_TX	I	TX 时钟输入
CKO	O	RX 时钟输出
DI[11:0]	I	AFE TX 数字输入代码
DO[11:0]	O	AFE RX 数字输出代码
I_CTRL_ADC[2:0]	I	功耗调整 默认值:0x3 （50μA）
I_CTRL_DAC[2:0]	I	增益误差调制 默认值:0x1
I_CTRL_STAGE1_PGA[1:0]	I	OP 电流控制 默认值:0x3 （40μA）
I_CTRL_STAGE2_PGA[1:0]	I	OP 电流控制 默认值:0x2 （20μA）
I_CTRL_STAGE3_PGA[1:0]	I	OP 电流控制 默认值:0x2 （40μA）
TA TB TC TD	I I I I	修整端口 默认值:0x0
EN_LDO	I	Bandgap 使能信号 0：禁用 1：启用（默认值）

8.17.5. 电器规格

8.17.5.1. 直流特性

表 8-210. 推荐工作条件

术语	定义	条件	最小值	典型值	最大值	单位
VCC33A	模拟电源电压	-	2.97	3.3	3.63	V
VCC11K	输入核心电源电压	-	0.99	1.1	1.21	V
TJ	工作结温	-	-40	+25	+125	°C
IVCC33A	电流消耗	正常运行 （TX）	-	12	TBD	mA
		正常运行 （RX）	-	39	TBD	mA
		待机模式	-	0.9	TBD	mA
		掉电模式	-	0.05	TBD	μA
IVCC11K	电流消耗	正常运行 （TX）	-	2.4	-	mA
		正常运行 （RX）	-	0.1	-	mA
		待机模式	-	0.3	-	mA
		掉电模式	-	0.3	-	μA

8.17.5.2. TX 路径特性

除非另有说明，否则所有测试均在以下条件下进行：VCC33A = 2.97 V∽3.63 V，VCC11K = 1.1 V±10%，TJ = -40°C∽125°C，通过 3.3μF 电容将 XCOMP 连接至 GNDA（ 最小 ESR> 0.5Ω），RSET = 6.2kΩ，RL = 50Ω。

表 8-211. TX 路径特性

描述	条件	最小值	典型值	最大值	单位
TX DAC 直流特性					
解析度	-	-	-	12	Bit
更新率	-	-	25	100	MSPS
满量程输出电流（IOUT_FS）	-	-	-	10	mA
输出电阻负载（RL）	-	-	-	50	Ω

描述	条件	最小值	典型值	最大值	单位
偏移误差	-	-10	-	10	mV
增益误差	-	-	± 5	-	% FS
电压符合范围	-	-0.5	-	0.5	V
Monotonic	-	设计保证	-	-	-
TX DAC 交流特性					
总谐波失真（THD）	Fout = 5.6 MHz	-	-57	TBD	dB
	Fout = 12 MHz （模拟）	-	-57	TBD	dB
无杂散动态范围（SFDR）	Fout = 5.6 MHz	TBD	62	-	dB
	Fout = 12 MHz （模拟）	TBD	62	-	dB
参考					
内部参考电压	-	-	1.22	-	V
参考误差	-	-	-	5	%

8.17.5.3. RX 路径特性

表 8-212. RX 路径特性

描述	条件	最小值	典型值	最大值	单位
RX 输入特性					
输入电压范围	RXPGA gain = -18 dB at VCC33A = 3.63 V. 注意：对于其他 VCC33A，请遵循： Vp-p (Max.) = (VCC33A + 0.34)*2.	-	-	7.94	VP-P
	RXPGA gain = +54 dB			2.0	mVP-P
输入电压范围		-	0.5*VCC33A	-	V
RXPGA 特性					
解析度	-	-	-	7	Bit
增益步长	-	1	-	-	dB
最小增益	-	-	-18	-	dB
最大增益	-	-	54	-	dB

描述	条件	最小值	典型值	最大值	单位
Monotonic	-	设计保证	-	-	-
输入参考噪声（f = 0.01 Hz∽20 MHz） *布局后仿真结果	RXPGA gain = +54 dB	TBD	4	-	$nV/\sqrt{Hz}$
RXPGA 增益= 0 dB 时的三阶 LPF 截止频率(-3 dB） *布局后仿真结果	LPF_SEL[1:0]=0x0	-	7.55	-	MHz
	LPF_SEL[1:0]=0x1	-	11.70		MHz
	LPF_SEL[1:0]=0x2	-	19.40		MHz
一阶 HPF 截止频率（-3 dB） *布局后仿真结果	HPF_SEL[1:0] = 00	-	98	-	kHz
ADC 直流特性					
解析度	-	-	-	12	Bit
转换率		10	25	50	MSPS
RXPGA 增益= 0 dB （满量程= 1.0 V p-p） Fin = 3.1 MHz; FADC = 25.0 MSPS	信噪比和失真度（SINAD）	-	54	-	dB
	无杂散动态范围（SFDR）	-	64	-	dB
RXPGA 增益= 0 dB （满量程= 1.0 V p-p） Fin = 6.1 MHz; FADC = 25.0 MSPS *布局后的模拟结果 I_CTRL_STAGE1_PGA [1：0] ='0x3' I_CTRL_STAGE2_PGA [1：0] ='0x3' I_CTRL_STAGE3_PGA [1：0] ='0x3'	信噪比和失真度（SINAD）	-	54	-	dB
	无杂散动态范围（SFDR）	-	64	-	dB
RXPGA 增益= 54 dB （满量程= 1.0 V p-p） Fin = 6.1 MHz; FADC = 25.0 MSPS	信噪比和失真度（SINAD）	-	23	-	dB

描述	条件	最小值	典型值	最大值	单位
*仅保证设计且仅用于 RD 测试					
RXPGA 增益= 54 dB （满量程= 1.0 V p-p） Fin = 6.1 MHz; FADC = 25.0 MSPS *仅保证设计且仅用于 RD 测试 *布局后的模拟结果 I_CTRL_STAGE1_PGA [1：0] ='0x3' I_CTRL_STAGE2_PGA [1：0] ='0x3' I_CTRL_STAGE3_PGA [1：0] ='0x3'	信噪比和失真度（SINAD）	-	23	-	dB

表 **8-213. RX** 输入阻抗

PGA 增益（dB）	RX 输入阻抗 （kΩ）
+24 ~ +54	0.5
+18 ~ +23	1
+12 ~ +17	2
+6 ~ +11	4
-6 ~ +5	8
-12 ~ -7	16
-18 ~ -13	32

8.17.6.时序特性

8.17.6.1. RX

除非另有说明，否则所有测试均在以下条件下进行：VCC33A = 2.97V∽3.63V，VCC11K = 1.1V±10%，TJ =

-40°C∽125°C，通过 3.3μF 电容将 XCOMP 连接至 GNDA（ 最小 ESR> 0.5Ω），RSET =6.2kΩ，RL = 50Ω。

表 8-214. RX 时序规格

符号	描述	条件	最小值	典型值	最大值	单位
t_{CP}	时钟周期	-	-	20	-	ns
t_{CH}	时钟高脉冲宽度	-	-	10	-	ns
t_{CL}	时钟低脉冲宽度	-	-	10	-	ns
	Clock duty	-	48.5	50	51.5	%
t_{OD}	AFE_RX 输出延时	-	3.65	4.70	6.90	ns
t_{setup}	数字输出的建立时间	-	-	-	3.10	ns
t_{hold}	数字输出的保持时间	-	-	-	13.65	ns
t_{LAT}	时钟延迟	-	-	5.5	-	CLK
t_{PDWU}	掉电模式下的唤醒时间	Clock rate = 50 MHz	-	1	3	ms
t_{STBYWU}	待机模式下的唤醒时间	Clock rate = 50 MHz	-		10	us

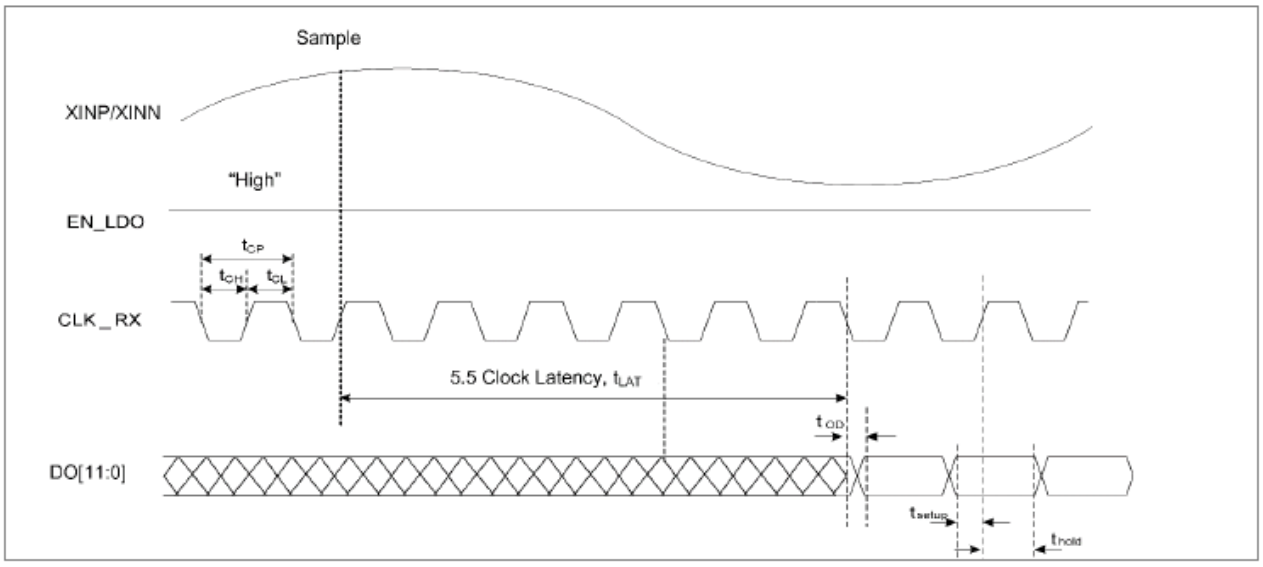


图 8-28. 正常工作状态下 RX 的时序图

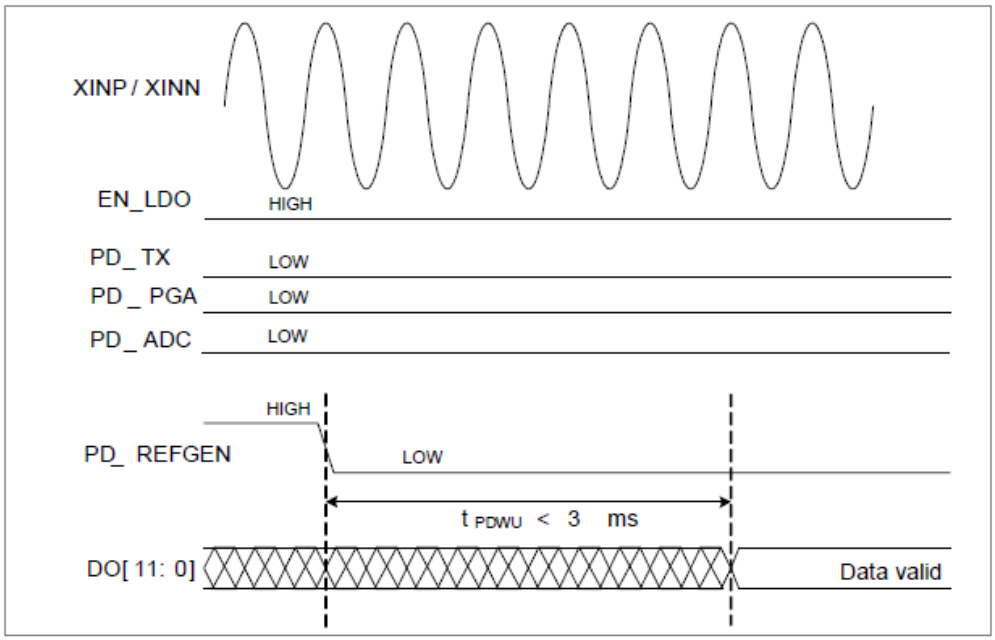


图 8-29. 掉电模式下 RX 的唤醒时序图

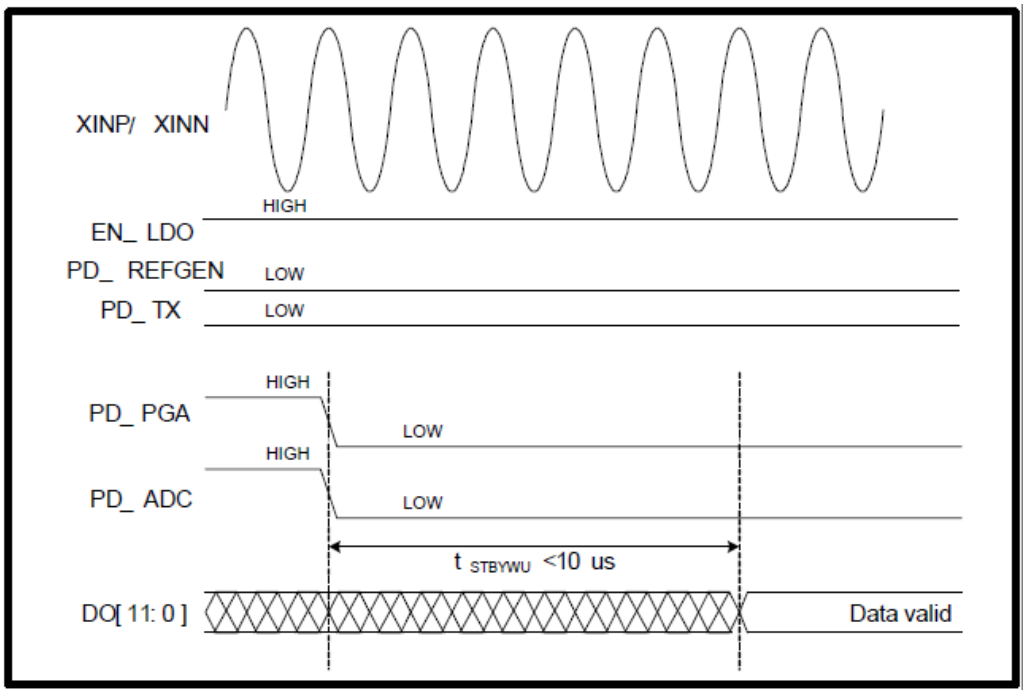


图 8-30. 待机模式下 RX 的唤醒时序图

8.17.6.2. TX

除非另有说明，否则所有测试均在以下条件下进行：VCC33A = 2.97V∽3.63V，VCC11K = 1.1V±10%，TJ = -40°C∽125°C，通过 3.3μF 电容将 XCOMP 连接至 GNDA（ 最小 ESR> 0.5Ω），RSET =6.2kΩ，RL = 50Ω。

表 **8-215. TX** 时序规格

符号	描述	条件	最小值	典型值	最大值	单位
t_S	数据建立时间	-	0.5		-	ns
t_H	数据保持时间	-	0.5		-	ns
t_{PD}	传播延迟时间	-	-	1.5	-	ns
t_{SET}	输出建立时间达到 0.1%	-	-	35	-	ns
t_{LAT}	Latency	-	-	0.5	-	CLK
	Clock duty	TJ = 25 °C	45	50	55	%
t_{STBYWU}	待机模式下的唤醒时间		-		5	μs
t_{PDWU}	掉电模式下的唤醒时间	-	-	-	3	ms

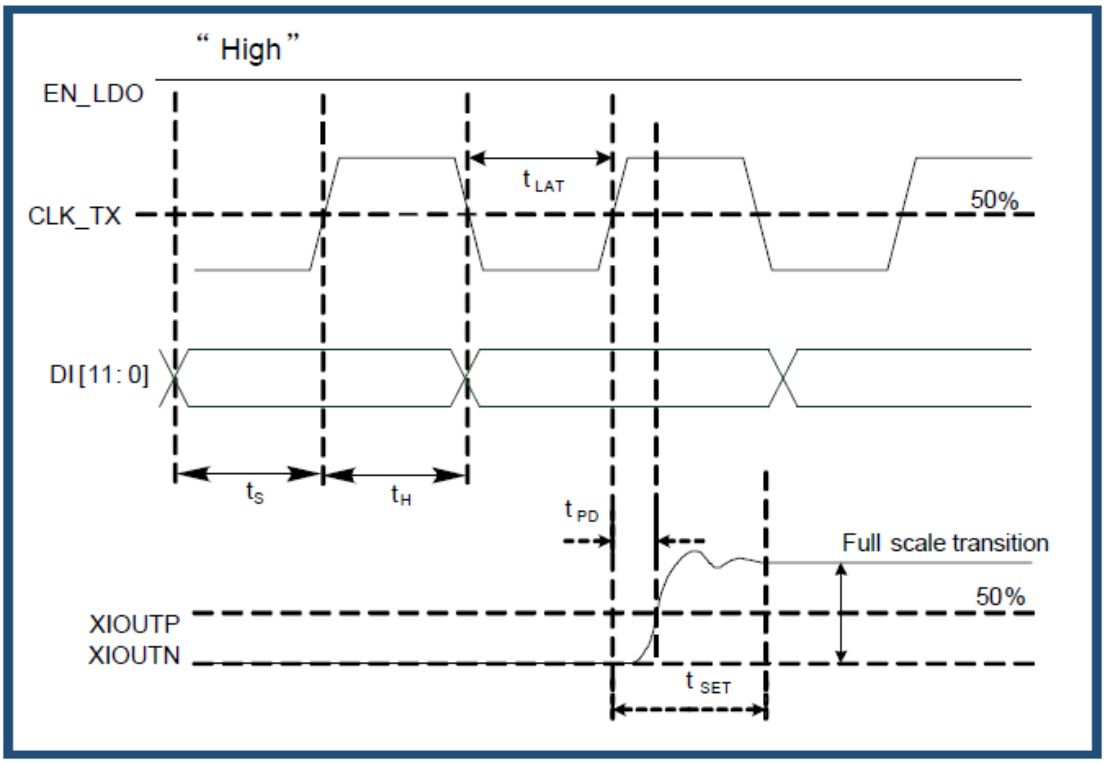


图 **8-31.** 正常工作状态下 **DAC** 的时序图

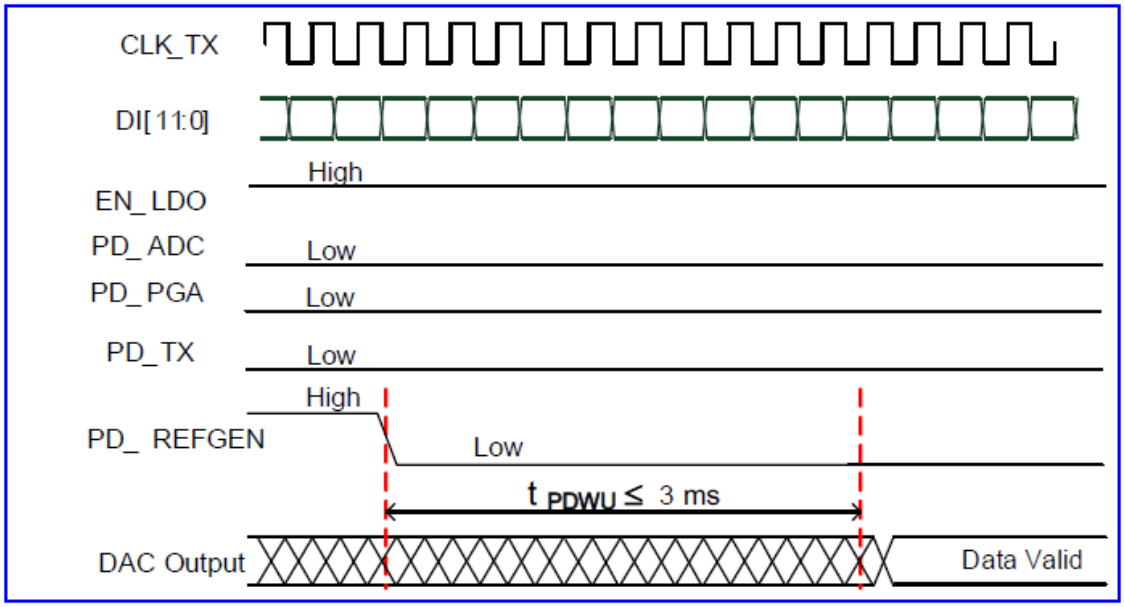


图 8-32. 掉电模式下 DAC 的唤醒时序图

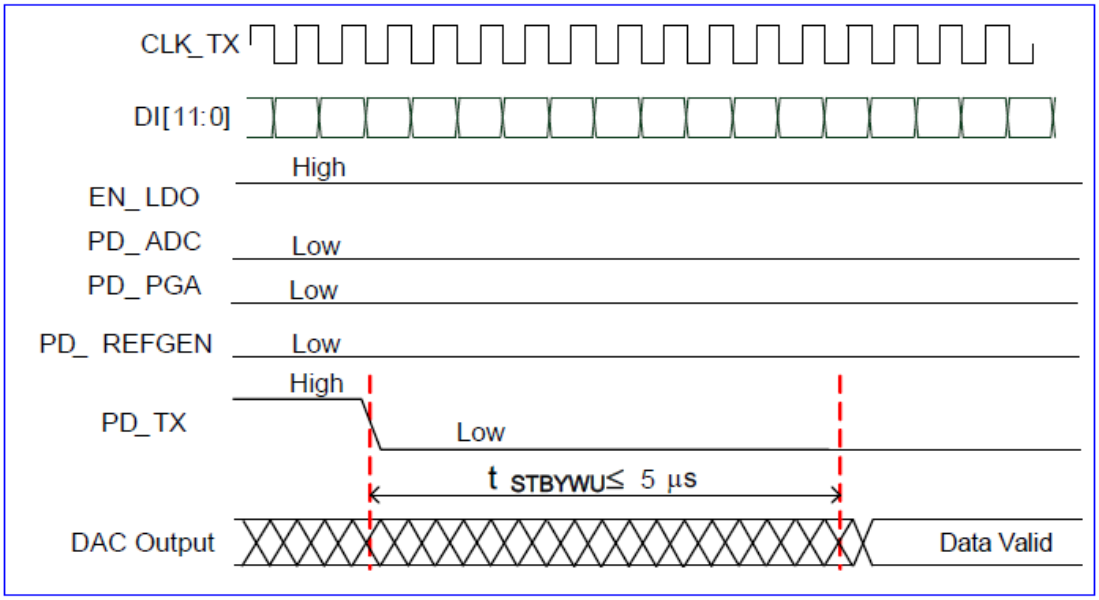


图 8-33. 待机模式下 DAC 的唤醒时序图

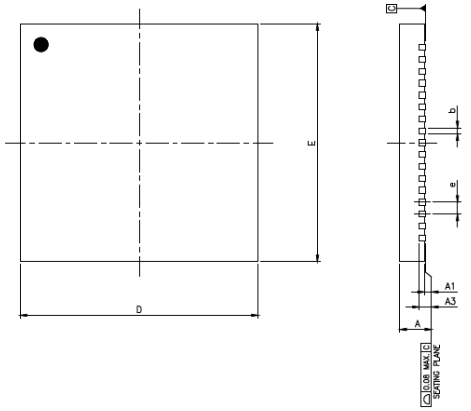
8.17.7. 电源和接地要求

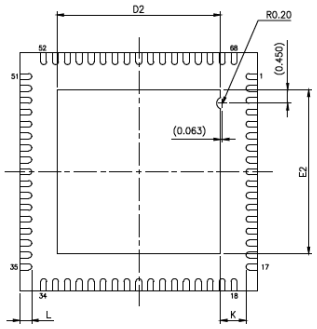
该IP有两个电源域：一个模拟域（VCC33A/GNDA）和一个数字域（VCC11K/GND11K）。这两个电源域与外部I/O电源环隔离。此外，这些电源域之间的钳位单元也包括在内，以提供更好的ESD抗扰性。AFE在VCC11K和GND11K之间没有钳位。

电源域之一 VCC11K 或 GND11K 由 ASIC 电源提供。建议 VCC11K/GND11K 的电源总线宽度至少为 30 µm，两个电源节点的允许 IR 压降为 20 mV。较短的键合线将有助于减少寄生电感。

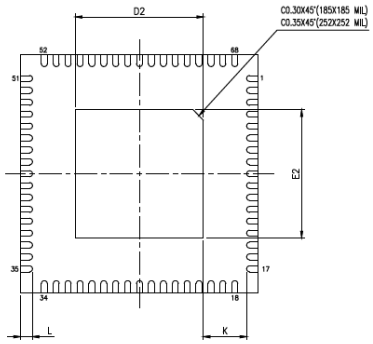
9.封装

9.1. V6201B（QFN-68）





PAD SIZE : 236X23* MIL



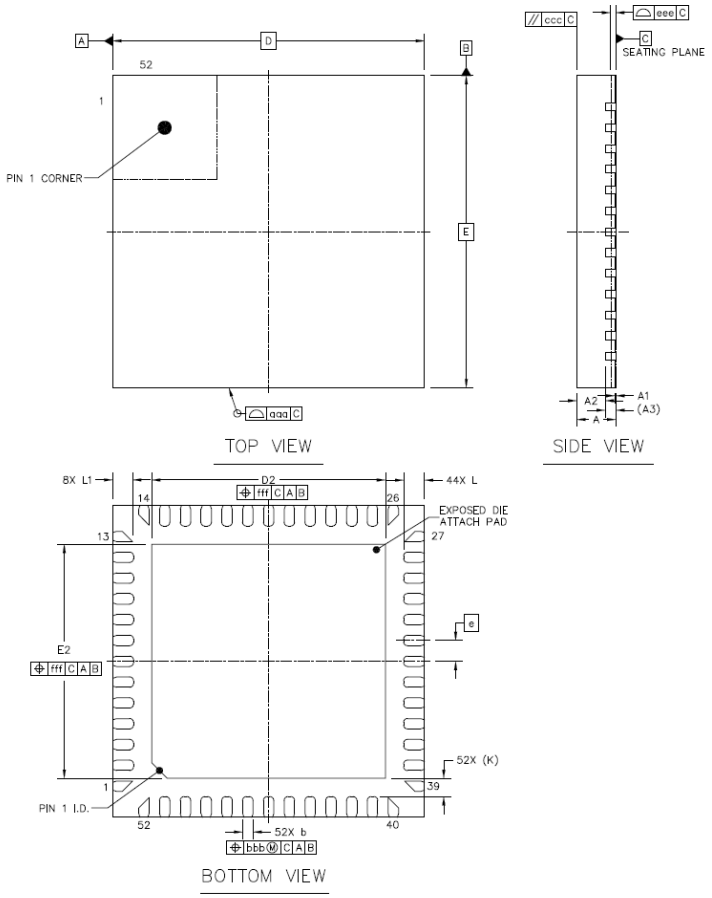
PAD SIZE : 185X18* MIL / 252X25* MIL

	PACKAGE TYPE					
JEDEC OUTLINE	MO-220			MO-220		
PKG CODE	WQFN(X868)			VQFN(Y868)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.203 REF.			0.203 REF.		
b	0.15	0.20	0.25	0.15	0.20	0.25
D	8.00 BSC			8.00 BSC		
E	8.00 BSC			8.00 BSC		
e	0.40 BSC			0.40 BSC		
L	0.35	0.40	0.45	0.35	0.40	0.45
K	0.20	—	—	0.20	—	—

PAD SIZE	D2			E2			LEAD FINISH		JEDEC CODE
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
6 185X18* MIL	4.25	4.30	4.35	4.25	4.30	4.35	V	X	N/A
6 236X23* MIL	5.45	5.50	5.55	5.45	5.50	5.55	V	X	N/A
6 252X25* MIL	6.15	6.20	6.25	6.15	6.20	6.25	V	X	VLLE-1

6 "*"表示汎用字元,此汎用字元可能被其它不同字元所取代,實際的字元請參照bonding diagram所示.
"*" is an universal character, which means maybe replaced by specific character, the actual character please refers to the bonding diagram.

9.2. V6201R（QFN-52）



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D		6 BSC	
	Y	E		6 BSC	
LEAD PITCH		e		0.4 BSC	
EP SIZE	X	D2	4.4	4.5	4.6
	Y	E2	4.4	4.5	4.6
LEAD LENGTH		L	0.35	0.4	0.45
		L1	0.28	0.38	0.48
LEAD TIP TO EXPOSED PAD EDGE		K		0.35 REF	
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		ccc		0.1	
COPLANARITY		eee		0.08	
LEAD OFFSET		bbb		0.07	
EXPOSED PAD OFFSET		fff		0.1	