# LKS32MC08x User Manual

© 2019, 版权归凌鸥创芯所有机密文件，未经许可不得扩散

# 目录

目录. II

表格目录

图片目录

# 1 文档约定

1.1 寄存器读写权限 11.2 缩略词汇表 1

# 2 地址空间 2

# 3 内核 3

3.1 ARM@CORTEXTM- M0 核心 3

3.2 NVIC控制器 3

3.3 异常和中断 4

3.4 SYSTICK定时器 5

3.4.1 SYST\_CSR 控制和状态寄存器 5

3.4.2 SYST\_RVR 重载值寄存器 6

3.4.3 SYST\_CVR 当前值寄存器 7

# 4 模拟电路 8

4.1 简述 8

4.2 POWER 电源管理系统 9

4.3 CLOCK 时钟系统 10

4.4 BGP 基准电压源 10

4.5 ADC 模数转换器 11

4.6 OPA 运算放大器 11

4.7 CMP 比较器 12

4.8 TMP 温度传感器 12

4.9 DAC 数模转换器 13

# 5 系统控制及时钟复位 15

5.1 时钟 15

5.1.1 时钟源. 15

5.1.2 功耗管理及休眠唤醒 17

5.1.2.1 休眠. 17

5.1.2.2 唤醒. 17

5.1.2.3 外设时钟门控. 17

5.1.2.4 外设时钟分频. 18

5.2 复位. .错误|未定义书签。

5.2.1 复位源. 18

5.2.1.1 硬件复位. 18

5.2.1.1.1 硬件复位架构. 18

5.2.1.1.2 硬件复位记录. 19

5.2.1.2 软件复位. 19

5.2.2 复位作用域. 19

5.3 寄存器. 20

5.3.1 地址分配. 20

5.3.2 SYS\_AFE\_CMP 模拟前端信息寄存器. 21

5.3.3 模拟寄存器概述. 22

5.3.4 SYS\_AFE\_REG0 模拟配置寄存器0. 22

5.3.5 SYS\_AFE\_REG1 模拟配置寄存器1. 23

5.3.6 SYS\_AFE\_REG2 模拟配置寄存器2. 24

5.3.7 SYS\_AFE\_REG3 模拟配置寄存器3. 24

5.3.8 SYS\_AFE\_REG4 模拟配置寄存器4. 26

5.3.9 SYS\_AFE\_REG5 模拟配置寄存器5. 26

5.3.10 SYS\_AFE\_REG6 模拟配置寄存器6. 28

5.3.11 SYS\_AFE\_REG7 模拟配置寄存器7. 29

5.3.12 SYS\_AFE\_DAC DAC 数字量寄存器. 30

5.3.13 SYS\_CLK\_CFG 时钟控制寄存器. 30

5.3.14 SYS\_RST\_CFG 复位控制寄存器. 31

5.3.15 SYS\_RST\_SRC 复位源记录寄存器. 32

5.3.16 SYS\_CLR\_RST 复位源记录清除寄存器. 32

5.3.17 SYS\_CLK\_DIV0 外设时钟分频寄存器0. 32 5.3.18 SYS\_CLK\_DIV2 外设时钟分频寄存器2. 33 5.3.19 SYS\_CLK\_FEN 外设时钟门控寄存器 33 5.3.20 SYS\_CLK\_SLP 休眠寄存器 34 5.3.21 SYS\_TRIM 校正模式寄存器 34 5.3.22 SYS\_SFT\_RST 软复位寄存器 35 5.3.23 SYS\_WR\_PROTECT 写保护寄存器 35 5.3.24 SYS\_AFE\_DAC\_AMC DAC 增益校正寄存器 36 5.3.25 SYS\_AFE\_DAC\_DC DAC 直流偏置寄存器 36

# 6 FLASH 38

6.1 概述 38

6.2 功能特点 38

6.2.1 功能描述 39

6.2.1.1 复位操作 39

6.2.1.2 休眠操作 39

6.2.1.3 FLASH 读取操作 40

6.2.1.4 FLASH 编程操作 40

6.2.1.5 FLASH 擦除操作 42

6.2.1.6 FLASH 预取操作 43

6.2.1.7 FLASH 加密保护 43

6.2.1.8 FLASH 在线升级(IAP) 44

6.2.1.8.1 开启中断的在线升级 44

6.2.1.8.2 关闭中断的在线升级 44

6.2.1.8.3 在线升级函数的位置 45

6.3 寄存器 45

6.3.1 地址分配 45

6.3.2 FLASH\_CFG 配置寄存器（推荐先读回，按或/与方式修改） 45

6.3.3 FLASH\_ADDR 地址寄存器 46

6.3.4 FLASH\_WDATA 与数据寄存器 47

6.3.5 FLASH\_RDATA 读数据寄存器 47

6.3.6 FLASH\_ERASE 擦除控制寄存器 476.3.7 FLASH\_PROTECT 加密状态寄存器 486.3.8 FLASH\_READY 工作状态寄存器 48

# 7 DMA 52

7.1 概述 527.2 请求 547.3 优先级 557.4 仲裁 557.5 中断 567.6 寄存器 567.6.1 地址分配 567.6.2 DMA\_CTRLDMA 控制寄存器 567.6.3 DMA\_IFDMA 中断标志寄存器 577.6.4 DMA 通道配置寄存器 577.6.4.1 DMA\_CCRx (where 577.6.4.2 DMA\_CTMSx (where 597.6.4.3 DMA\_CPARx (where 607.6.4.4 DMA\_CMARx (where 60

# 8 GPIO 62

8.1 概述 628.1.1 功能框图 628.1.2 产品特点 638.2 寄存器 648.2.1 地址分配 648.2.2 GPIOx PIE 648.2.3 GPIOx POE 658.2.4 GPIOx PDI 668.2.5 GPIOx PDO 668.2.6 GPIOx PUE 678.2.7 GPIOx PODE 67

8.2.8 GPIOx\_LCKR. .688.2.9 GPIOx\_F3210. .698.2.10 GPIOx\_F7654. .708.2.11 GPIOx\_FBA98. .708.2.12 GPIOx\_FFEDC. .718.2.13 外部中断、唤醒、锁定保护. .718.2.13.1 EXTI\_CR0. .718.2.13.2 EXTI\_CR1. .738.2.13.3 EXTI\_IF. .748.2.13.4 LCKR\_PRT. .758.2.13.5 WAKE\_POL. .768.2.13.6 WAKE\_EN. .768.3 实现说明. .778.3.1 上拉实现. .778.4 应用指南. .778.4.1 配置锁定. .778.4.2 外部中断. .798.4.3 使用GPIO的模拟功能. .799 CRC. .809.1 概述. .809.2 基本原理. .809.3 基本概念. .809.3.1 对应关系. .809.3.2 生成多项式. .819.3.3 校验码位数. .819.3.4 生成步骤. .819.4 寄存器. .829.4.1 地址分配. .829.4.2 寄存器描述. .839.4.2.1 CRC\_DR CRC 信息码寄存器. .83

9.4.2.2 CRC\_CR CRC 控制寄存器 839.4.2.3 CRC\_INIT CRC 初始码寄存器 849.4.2.4 CRC\_POL CRC 生成码寄存器 85

# 10 ADC 86

10.1 概述 86

10.1.1 功能框图 86

10.1.2 ADC 触发方式 88

10.1.3 ADC 通道选择 88

10.1.4 ADC 中断 89

10.1.5 ADC 输出数制 89

10.1.6 ADC 基准电压与量程 90

10.1.6.1 2.4V 基准电压模式 90

10.1.6.2 1.2V 基准电压模式 91

10.1.7 ADC 校正 91

10.1.8 ADC 配置流程 92

10.2 寄存器 93

10.2.1 地址分配 93

10.2.2 采样数据寄存器 94

10.2.2.1 ADC0\_DAT0 94

10.2.2.2 ADC0\_DAT1 94

10.2.2.3 ADC0\_DAT2 95

10.2.2.4 ADC0\_DAT3 95

10.2.2.5 ADC0\_DAT4 96

10.2.2.6 ADC0\_DAT5 96

10.2.2.7 ADC0\_DAT6 96

10.2.2.8 ADC0\_DAT7 97

10.2.2.9 ADC0\_DAT8 97

10.2.2.10 ADC0\_DAT9 98

10.2.2.11 ADC0\_DAT10 98

10.2.2.12 ADC0\_DAT11 98

10.2.2.13 ADC0\_DAT12. 99

10.2.2.14 ADC\_DAT13. 99

10.2.2.15 ADC0\_DAT14. 100

10.2.2.16 ADC0\_DAT15. 100

10.2.2.17 ADC0\_DAT16. 100

10.2.2.18 ADC0\_DAT17. 101

10.2.2.19 ADC0\_DAT18. 101

10.2.2.20 ADC0\_DAT19. 102

# 10.2.3 信号来源寄存器 102

10.2.3.1 ADC0\_CHN0. 102

10.2.3.2 ADC0\_CHN1. 103

10.2.3.3 ADC0\_CHN2. 103

10.2.3.4 ADC0\_CHN3. 104

10.2.3.5 ADC0\_CHN4. 104

10.2.3.6 ADC0\_CHN5. 105

10.2.3.7 ADC0\_CHN6. 105

10.2.3.8 ADC0\_CHN7. 106

10.2.3.9 ADC0\_CHN8. 106

10.2.3.10 ADC0\_CHN9. 107

# 10.2.4 分段通道数寄存器 108

10.2.4.1 ADC0\_CHNT0. 108

10.2.4.2 ADC0\_CHNT1. 108

10.2.5 配置寄存器 109

10.2.5.1 ADC0\_CFG. 109

10.2.5.2 ADC0\_TRIG. 109

10.2.6 增益选择寄存器 111

10.2.6.1 ADC0\_GAIN0. 111

10.2.6.2 ADC0\_GAIN1. 111

10.2.7 中断寄存器 112

10.2.7.1 ADC0\_IE. 112

# LKS32MC08x User Manual

10.2.7.2 ADC0\_IF. 113

10.2.8 软件触发寄存器 113

10.2.8.1 ADC0 SWT. 113

10.2.9 直流偏置寄存器 114

10.2.9.1 ADC0\_DC\_A0 114

10.2.9.2 ADC0\_DC\_A1 114

10.2.9.3 ADC0\_DC\_B0 115

10.2.9.4 ADC0\_DC\_B1 115

10.2.10 增益校正寄存器 115

10.2.10.1 ADC0\_AMC\_A0 115

10.2.10.2 ADC0\_AMC\_A1 116

10.2.10.3 ADC0\_AMC\_B0 116

10.2.10.4 ADC0\_AMC\_B1 117

10.2.11 通道0阈值寄存器 117

10.2.11.1 ADC0\_DATA0\_TH 117

10.3 实现说明 118

10.3.1 DMA请求 118

10.3.2 连续采样 118

10.4 应用指南 118

10.4.1 ADC采样触发模式 118

10.4.1.1 单段触发模式 120

10.4.1.2 两段触发模式 120

10.4.1.3 四段触发模式 121

10.4.2 中断 122

10.4.2.1 单段触发采样完成中断 122

10.4.2.2 两段触发采样完成中断 122

10.4.2.3 四段触发采样完成中断 122

10.4.3 配置修改 122

10.4.4 选择对应的模拟通道 122

11 通用定时器 124

11.1 概述. 124

11.1.1 功能框图. 124

11.1.1.1 总线接口模块. 124

11.1.1.2 寄存器模块. 124

11.1.1.3 IO 滤波模块. 125

11.1.1.4 通用定时器模块. 125

11.1.1.5 编码器模块. 125

11.1.1.6 时钟分频模块. 125

11.1.2 功能特点. 125

11.2 实现说明. 126

11.2.1 时钟分频. 126

11.2.2 滤波. 126

11.2.3 模式. 126

11.2.3.1 计数器. 126

11.2.3.2 比较模式. 127

11.2.3.3 捕获模式. 128

11.2.4 编码器. 129

11.2.4.1 正交编码信号. 129

11.2.4.2 符号加脉冲信号. 130

11.2.4.3 CCW/CW 双脉冲信号. 131

11.3 寄存器. 131

11.3.1 地址分配. 131

11.3.2 系统控制寄存器. 132

11.3.2.1 UTIMER\_CFG. 132

11.3.3 滤波控制寄存器. 133

11.3.3.1 UTIMER\_FLT\_TH01. 133

11.3.3.2 UTIMER\_FLT\_TH23. 134

11.3.4 定时器寄存器. 134

11.3.4.1 UTIMER\_UNT0\_CFG\_Timer0 配置寄存器. 134

11.3.4.2 UTIMER\_UNT1\_CFG\_Timer1 配置寄存器. 136

# 11.3.4.3 UTIMER\_UNT2\_CFG\_Timer2 配置寄存器 137

11.3.4.4 UTIMER\_UNT3\_CFG\_Timer3 配置寄存器 13811.3.4.5 UTIMER\_UNT0\_TH Timer 0 门限寄存器 13911.3.4.6 UTIMER\_UNT1\_TH Timer 1 门限寄存器 14011.3.4.7 UTIMER\_UNT2\_TH Timer 2 门限寄存器 14011.3.4.8 UTIMER\_UNT3\_TH Timer 3 门限寄存器 14111.3.4.9 UTIMER\_UNT0\_CNT Timer 0 计数寄存器 14111.3.4.10 UTIMER\_UNT1\_CNT Timer 1 计数寄存器 14111.3.4.11 UTIMER\_UNT2\_CNT Timer 2 计数寄存器 14211.3.4.12 UTIMER\_UNT3\_CNT Timer 3 计数寄存器 14211.3.4.13 UTIMER\_UNT0\_CMP0 Timer 0 通道 0 比较捕获寄存器 14311.3.4.14 UTIMER\_UNT0\_CMP1 Timer 0 通道 1 比较捕获寄存器 14311.3.4.15 UTIMER\_UNT1\_CMP0 Timer 1 通道 0 比较捕获寄存器 14411.3.4.16 UTIMER\_UNT1\_CMP1 Timer 1 通道 1 比较捕获寄存器 14411.3.4.17 UTIMER\_UNT2\_CMP0 Timer 2 通道 0 比较捕获寄存器 14511.3.4.18 UTIMER\_UNT2\_CMP1 Timer 2 通道 1 比较捕获寄存器 14511.3.4.19 UTIMER\_UNT3\_CMP0 Timer 3 通道 0 比较捕获寄存器 14611.3.4.20 UTIMER\_UNT3\_CMP1 Timer 3 通道 1 比较捕获寄存器 14611.3.4.21 UTIMER\_UNT0\_EVT Timer0 外部事件选择寄存器 14711.3.4.22 UTIMER\_UNT1\_EVT Timer1 外部事件选择寄存器 14711.3.4.23 UTIMER\_UNT2\_EVT Timer2 外部事件选择寄存器 14811.3.4.24 UTIMER\_UNT3\_EVT Timer3 外部事件选择寄存器 14811.3.5 编码器寄存器 14911.3.5.1 UTIMER\_ECD0\_CFG Encoder0 配置寄存器 14911.3.5.2 UTIMER\_ECD0\_TH Encoder0 计数门限寄存器 15011.3.5.3 UTIMER\_ECD0\_CNT Encoder0 计数值寄存器 15011.3.6 中断管理寄存器 15111.3.6.1 UTIMER\_IE 中断使能寄存器 15111.3.6.2 UTIMER\_IF 中断标志寄存器 15211.3.7 DMA 管理寄存器 153

11.3.7.1 UTIMER\_RE DMA 请求使能寄存器 153

# 12 HALL 信号处理模块 155

12.1 综述 155

12.2 实现说明 155

12.2.1 信号来源 155

12.2.2 工作时钟 155

12.2.3 信号滤波 155

12.2.4 捕获 156

12.2.5 中断 156

12.2.6 数据流程 157

12.3 寄存器 157

12.3.1 地址分配 157

12.3.2 HALL\_CFG HALL 模块配置寄存器 157

12.3.3 HALL\_INFO HALL 模块信息寄存器 158

12.3.4 HALL\_WIDTH HALL 宽度计数值寄存器 159

12.3.5 HALL\_TH HALL 模块计数器门限值寄存器 159

12.3.6 HALL\_CNT HALL 计数寄存器 160

# 13 MCPWM 161

13.1 概述 161

13.1.1 Base Counter 模块 162

13.1.2 Fail 信号处理 163

13.1.3 MCPWM 特殊输出状态 164

13.1.4 IO DRIVER 模块 164

13.1.4.1 MCPWM 波形输出- 中心对齐模式 165

13.1.4.2 MCPWM 波形控制- 中心对齐推挽模式 166

13.1.4.3 MCPWM 波形输出- 边沿对齐模式 166

13.1.4.4 MCPWM 波形控制- 边沿对齐推挽模式 167

13.1.4.5 MCPWM IO 死区控制 168

13.1.4.6 MCPWM IO 极性设置 168

13.1.4.7 MCPWM IO 自动保护 168

13.1.5 ADC Trigger Timer 模块. 169

13.1.6 MCPWM 主要事件列表. 169

13.2 寄存器. 169

13.2.1 地址分配. 169

13.2.2 MCPWM\_TH00. 171

13.2.3 MCPWM\_TH01. 172

13.2.4 MCPWM\_TH10. 172

13.2.5 MCPWM\_TH11. 173

13.2.6 MCPWM\_TH20. 173

13.2.7 MCPWM\_TH21. 173

13.2.8 MCPWM\_TH30. 174

13.2.9 MCPWM\_TH31. 174

13.2.10 MCPWM\_TMR0. 175

13.2.11 MCPWM\_TMR1. 175

13.2.12 MCPWM\_TMR2. 176

13.2.13 MCPWM\_TMR3. 176

13.2.14 MCPWM\_TH. 176

13.2.15 MCPWM\_UPDATE. 177

13.2.16 MCPWM\_IE. 178

13.2.17 MCPWM\_IF. 179

13.2.18 MCPWM\_EIE. 180

13.2.19 MCPWM\_EIF. 181

13.2.20 MCPWM\_RE. 182

13.2.21 MCPWM\_PP. 182

13.2.22 MCPWM\_IO01. 183

13.2.23 MCPWM\_IO23. 184

13.2.24 MCPWM\_SDCPG. 185

13.2.25 MCPWM\_TCLK. 185

13.2.26 MCPWM\_FAIL. 186

13.2.27 MCPWM\_PRT. 187

13.2.28 MCPWM\_CNT. 188 13.2.29 MCPWM\_DTH0. 188 13.2.30 MCPWM\_DTH01. 189 13.2.31 MCPWM\_DTH10. 189 13.2.32 MCPWM\_DTH11. 189 13.2.33 MCPWM\_DTH20. 190 13.2.34 MCPWM\_DTH21. 190 13.2.35 MCPWM\_DTH30. 191 13.2.36 MCPWM\_DTH31. 191

# 14 UART 192

14.1 概述 192

14.2 功能说明 192

14.2.1 发送 192 14.2.2 接收 192 14.2.3 波特率配置 192 14.2.4 收发端口互换(TX/RX互换) 193 14.2.5 DMA配置 193

# 14.3 寄存器 194

14.3.1 地址分配 194 14.3.2 UARTx\_CTRL UARTx 控制寄存器 194 14.3.3 UARTx\_DIVH UARTx 波特率设置高字节寄存器 195 14.3.4 UARTx\_DIVL UARTx 波特率设置低字节寄存器 195 14.3.5 UARTx\_BUFF UARTx 收发缓冲寄存器 196 14.3.6 UARTx\_ADR UARTx 地址匹配寄存器 196 14.3.7 UARTx\_STT UARTx 状态寄存器 196 14.3.8 UARTx\_IE UARTx 中断使能寄存器 197 14.3.9 UARTx\_IF UARTx 中断标志寄存器 198 14.3.10 UARTx\_INV UARTx IO 翻转寄存器 199

# 15 DSP 200

15.1 概述 200

15.1.1 功能框图. 201

15.1.2 DSP核寄存器 201

15.1.3 位宽. 202

15.1.4 指令周期. 202

15.1.5 地址空间. 202

15.2 寄存器. 203

15.2.1 地址分配. 203

15.2.2 DSP状态控制寄存器. 203

15.2.2.1 DSP\_SC. 203

15.2.3 DSPsin/cos相关寄存器. 204

15.2.3.1 DSP\_THETA. 204

15.2.3.2 DSP\_SIN. 205

15.2.3.3 DSP\_COS. 205

15.2.4 DSP arctan相关寄存器. 206

15.2.4.1 DSP\_X. 206

15.2.4.2 DSP\_Y. 206

15.2.4.3 DSP\_MOD. 207

15.2.4.4 DSP\_ARCTAN. 207

15.2.5 DSP除法相关寄存器. 208

15.2.5.1 DSP\_DID. 208

15.2.5.2 DSP\_DIS. 208

15.2.5.3 DSP\_QUO. 209

15.2.5.4 DSP\_REM. 209

15.2.6 DSP开方相关寄存器. 210

15.2.6.1 DSP\_RAD. 210

15.2.6.2 DSP\_SQRT. 210

15.3 DSP指令集. 211

15.3.1 Instruction Set Summary. 211

15.3.2 ADD. 212

15.3.2.1 编码. 212

15.3.2.2 汇编语法. .212 15.3.2.3 伪代码. .212

# 15.3.3 ADDI (reserved) .212

15.3.3.1 指令编码. .212 15.3.3.2 汇编语法. .212 15.3.3.3 伪代码. .212

# 15.3.4 SUB. .213

15.3.4.1 指令编码. .213 15.3.4.2 汇编语法. .213 15.3.4.3 伪代码. .213

# 15.3.5 ASR. .213

15.3.5.1 指令编码. .213 15.3.5.2 汇编语法. .213 15.3.5.3 伪代码. .213

# 15.3.6 ASRI. .214

15.3.6.1 指令编码. .214 15.3.6.2 汇编语法. .214 15.3.6.3 伪代码. .214

# 15.3.7 LSL. .214

15.3.7.1 指令编码. .214 15.3.7.2 汇编语法. .214 15.3.7.3 伪代码. .214

# 15.3.8 LSLI. .214

15.3.8.1 指令编码. .214 15.3.8.2 汇编语法. .215 15.3.8.3 伪代码. .215

# 15.3.9 MAC. .215

15.3.9.1 指令编码. .215 15.3.9.2 汇编语法. .215 15.3.9.3 伪代码. .215

# 15.3.10 MACI (reserved) 215

15.3.10.1 指令编码 216

15.3.10.2 汇编语法 216

15.3.10.3 伪代码 216

# 15.3.11 DIV 216

15.3.11.1 指令编码 216

15.3.11.2 汇编语法 216

15.3.11.3 伪代码 216

# 15.3.12 SAT 217

15.3.12.1 指令编码 217

15.3.12.2 汇编语法 217

15.3.12.3 伪代码 217

# 15.3.13 SATI (reserved) 217

15.3.13.1 指令编码 217

15.3.13.2 汇编语法 217

15.3.13.3 伪代码 217

# 15.3.14 SIN\_COS 217

15.3.14.1 指令编码 217

15.3.14.2 汇编语法 218

15.3.14.3 伪代码 218

# 15.3.15 ARCTAN 218

15.3.15.1 指令编码 218

15.3.15.2 汇编语法 218

15.3.15.3 伪代码 218

# 15.3.16 SQRT 218

15.3.16.1 指令编码 218

15.3.16.2 汇编语法 218

15.3.16.3 伪代码 218

# 15.3.17 LDRWI 219

15.3.17.1 指令编码 219

15.3.17.2 汇编语法. .219 15.3.17.3 伪代码. .219

# 15.3.18 LDRDHI. .219

15.3.18 LDRDHI. .219 15.3.18.1 指令编码. .219 15.3.18.2 汇编语法. .219 15.3.18.3 伪代码. .219

# 15.3.19 STRWI. .220

15.3.19 STRWI. .220 15.3.19.1 指令编码. .220 15.3.19.2 汇编语法. .220 15.3.19.3 伪代码. .220

# 15.3.20 STRDHI (reserved) .220

15.3.20.1 指令编码. .220 15.3.20.2 汇编语法. .220 15.3.20.3 伪代码. .220

# 15.3.21 JUMP (reserved) .221

15.3.21.1 指令编码. .221 15.3.21.2 汇编语法. .221 15.3.21.3 伪代码. .221

# 15.3.22 JUMPI. .221

15.3.22.1 指令编码. .221 15.3.22.2 汇编语法. .221 15.3.22.3 伪代码. .221

# 15.3.23 JLE. .221

15.3.23.1 指令编码. .221 15.3.23.2 汇编语法. .222 15.3.23.3 伪代码. .222

# 15.3.24 JLEI. .222

15.3.24.1 指令编码. .222 15.3.24.2 汇编语法. .222 15.3.24.3 伪代码. .222

15.3.25 IRQ. 222

15.3.25.1 指令编码. 222

15.3.25.2 汇编语法. 222

15.3.25.3 伪代码. 222

15.3.26 R（仅用于模拟器） 223

15.3.26.1 指令编码. 223

15.3.26.2 汇编语法. 223

15.3.26.3 伪代码. 223

15.3.27 BREAK（仅用于模拟器） 223

15.3.27.1 指令编码. 223

15.3.27.2 汇编语法. 223

15.3.27.3 伪代码. 223

15.3.28 END（仅用于模拟器） 223

15.3.28.1 指令编码. 223

15.3.28.2 汇编语法. 223

15.3.28.3 伪代码. 224

15.4 应用指南 224

15.4.1 访存寻址. 224

15.4.2 Load after Store. 225

15.4.3 多周期指令的的延迟提交. 225

15.4.4 数据填装. 225

16 I2C. 227

16.1 概述. 227

16.2 主要特性. 227

16.3 功能描述. 227

16.3.1 功能框图. 227

16.3.2 功能说明. 228

16.3.2.1 模式选择. 228

16.3.2.2 I2C 接口从模式. 229

16.3.2.2.1 从模式单字节传输. 230

# LKS32MC08x User Manual

16.3.2.2.2 从模式单字节发送 23116.3.2.2.3 从模式单字节接收 231

16.3.2.3 从模式 DMA 传输 231

16.3.2.3.1 从模式 DMA 发送 231

16.3.2.3.2 从模式 DMA 接收 232

16.3.2.4 I2C 接口主模式 232

16.3.2.4.1 主模式单字节传输 233

16.3.2.4.2 主模式单字节发送 233

16.3.2.4.3 主模式单字节接收 234

16.3.2.4.4 主模式 DMA 传输 234

16.3.2.4.5 主模式 DMA 发送 234

16.3.2.4.6 主模式 DMA 接收 235

16.3.2.5 I2C 总线异常处理 236

16.3.2.6 DMA 传输 236

16.3.2.7 CPU 传输 237

16.3.2.8 中断处理 237

16.3.2.9 通讯速度设置 237

16.4 寄存器 238

16.4.1 地址分配 238

16.4.2 寄存器说明 238

16.4.2.1 I2C0\_ADDR 地址寄存器 238

16.4.2.2 I2C0\_CFG 系统控制寄存器 238

16.4.2.3 I2C0\_SCR 状态控制寄存器 239

16.4.2.4 I2C0\_DATA 数据寄存器 240

16.4.2.5 I2C0\_MSCR 主模式寄存器 241

16.4.2.6 I2C0\_BCR DMA 传输控制寄存器 242

17 SPI 243

17.1 概述 243

17.2 主要特性 243

17.3 功能描述 243

17.3.1 功能框图. 243

17.3.2 功能说明. 244

17.3.2.1 全双工模式. 244 17.3.2.2 半双工模式. 245 17.3.2.3 片选信号. 246 17.3.2.4 通讯格式. 247 17.3.2.5 数据格式. 249 17.3.2.6 DMA传输. 249 17.3.2.7 CPU传输. 249 17.3.2.8 外部事件传输. 250 17.3.2.9 中断处理. 250 17.3.2.10 波特率设置. 251

17.4 寄存器. 251

17.4.1 地址分配. 251 17.4.2 SPI\_CFG SPI控制寄存器. 251 17.4.3 SPI\_IE SPI中继寄存器. 252 17.4.4 SPI\_DIV SPI波特率寄存器. 253 17.4.5 SPI\_TX\_DATA SPI数据发送寄存器. 254 17.4.6 SPI\_RX\_DATA SPI数据接收寄存器. 254 17.4.7 SPI\_SIZE SPI数据传输长度寄存器. 254

# 18 CMP. 256

18.1 概述. 256

18.2 寄存器. 258

18.2.1 地址分配. 258

18.2.2 CMP模块寄存器描述. 258

18.2.2.1 CMP\_IE. 258

18.2.2.2 CMP\_IF. 258

18.2.2.3 CMP\_TCLK. 259

18.2.2.4 CMP\_CFG. 260

18.2.2.5 CMP\_BLCWIN. 262

# 19 CAN. 264

19.1 概述. 26419.1 主要特性 26419.2 功能描述 26419.2.1 功能框图. 26419.2.2 功能说明. 26519.2.2.1 工作模式. 26519.2.2.2 DMA传输. 26519.2.2.3 CPU传输. 26519.2.2.4 中断处理. 26619.2.2.4.1 接收中断. 26619.2.2.4.2 发送中断. 26619.2.2.4.3 错误报警中断. 26619.2.2.4.4 数据溢出中断. 26719.2.2.4.5 仲裁丢失中断. 26719.2.2.4.6 总线错误中断. 26719.2.2.5 通讯采样设置. 26719.2.2.6 ID滤波. 26919.2.2.7 发送帧格式. 27119.2.2.8 接收帧格式. 27219.2.2.9 发送. 27319.2.2.10 接收. 27319.2.2.11 错误管理. 27419.2.2.12 错误计数. 27519.2.2.13 错误报警限制. 27619.2.2.14 被动错误. 27619.2.2.15 离线状态与离线恢复. 27619.2.3 寄存器. 27619.2.3.1 地址分配. 27619.2.3.2 寄存器说明. 278

19.2.3.2.1 CAN\_MOD 模式寄存器 .27819.2.3.2.2 CAN CMR 命令寄存器 .27819.2.3.2.3 CAN\_SR 状态寄存器 .27919.2.3.2.4 CAN\_IR 中断状态寄存器 .28019.2.3.2.5 CAN\_EIR 中断控制寄存器 .28119.2.3.2.6 CAN\_BTR0 波特率 0 控制寄存器 .28119.2.3.2.7 CAN\_BTR1 波特率 1 控制寄存器 .28219.2.3.2.8 CAN\_ALC 仲裁丢失捕捉寄存器 .28219.2.3.2.9 CAN\_ECC 错误码捕捉寄存器 .28319.2.3.2.10 CAN\_EVLR 错误&警告门限值寄存器 .28419.2.3.2.11 CAN\_RXERR 接收错误计数器寄存器 .28519.2.3.2.12 CAN\_TXERR 发送错误计数器寄存器 .28519.2.3.2.13 CAN\_TXRX0~CAN\_TXRXC 发送接收寄存器 .28619.2.3.2.14 CAN\_ACR ID 寄存器 .28619.2.3.2.15 CAN\_AMR ID 掩码寄存器 .28619.2.3.2.16 CAN\_RMC FIFO 有效接收数据寄存器 .28719.2.3.2.17 CAN\_RBSA 有效接收数据地址寄存器 .28719.2.3.2.18 CAN\_RFIFO0~CAN\_RFIFO31 RX FIFO 寄存器 .28819.2.3.2.19 CAN\_TFIFO0~CAN\_TFIFO12 TX FIFO 寄存器 .288

# 20 SIF 290

20.1 概述 .29020.2 主要特性 .29020.3 功能描述 .29020.3.1 功能框图 .29020.3.2 功能说明 .29020.3.3 模式选择 .29120.3.4 SIF 接口传输 .29120.3.5 中断处理 .29120.3.6 通讯速度设置 .29120.4 寄存器 .291

# LKS32MC08x User Manual

20.4.1 地址分配. 291

20.4.2 寄存器说明. 292

20.4.2.1 SIF\_CFG 配置寄存器. 292

20.4.2.2 SIF\_FREQ 波特率寄存器. 292

20.4.2.3 SIF\_IRQ 状态控制寄存器. 293

20.4.2.4 SIF\_WDATA 数据寄存器. 293

21 看门狗. 294

21.1 概述. 294

21.2 寄存器. 294

21.2.1 地址分配. 294

21.2.2 寄存器说明. 294

21.2.2.1 SYS\_WDT\_CLR 看门狗清零寄存器. 294

22 版本历史. 296

LKS32MC08x User Manual

# 表格目录

表2- 1系统地址空间分配. 2表3- 1中断号分布. 4表3- 2中断号分布. 5表3- 3SYST\_CSR控制和状态寄存器. 5表3- 4SYST\_RVR重载值寄存器. 6表3- 5SYST\_CVR当前值寄存器. 7表5- 1系统时钟源. 15表5- 2PLL作为MCLK时钟时的分频配置 16表5- 3硬件复位源. 18表5- 4复位作用域. 19表5- 5系统控制寄存器. 20表5- 6模拟前端信息寄存器SYS\_AFE\_CMP. 21表5- 7模拟配置寄存器0SYS\_AFE\_REG0. 22表5- 8模拟配置寄存器1SYS\_AFE\_REG1. 23表5- 9模拟配置寄存器2SYS\_AFE\_REG2. 24表5- 10模拟配置寄存器3SYS\_AFE\_REG3. 25表5- 11模拟配置寄存器4SYS\_AFE\_REG4. 26表5- 12模拟配置寄存器5SYS\_AFE\_REG5. 27表5- 13模拟配置寄存器6SYS\_AFE\_REG6. 28表5- 14模拟配置寄存器7SYS\_AFE\_REG7. 29表5- 20DAC数字量寄存器SYS\_AFE\_DAC. 30表5- 21时钟控制寄存器SYS\_CLK\_CFG. 30表5- 22复位控制寄存器SYS\_RST\_CFG. 31表5- 23复位源记录寄存器SYS\_RST\_SRC. 32表5- 24复位源记录清除寄存器SYS\_CLK\_RST. 32表5- 25外设时钟分频寄存器0SYS\_CLK\_DIV0. 33表5- 26外设时钟分频寄存器2SYS\_CLK\_DIV2. 33表5- 27外设时钟门控寄存器SYS\_CLK\_FEN. 33

表5- 28 休眠寄存器 SYS\_CLK\_SLP ….. 34表5- 29 校正模式寄存器 SYS\_TRIM ….. 34表5- 30 软复位寄存器 SYS\_SFT\_RST ….. 35表5- 31 写保护寄存器 SYS\_WR\_PROTECT ….. 35表5- 32 DAC 增益校正寄存器 SYS\_AFE\_DAC\_AMC ….. 36表5- 33 DAC 直流偏置寄存器 SYS\_AFE\_DAC\_DC ….. 36表6- 1 FLASH 访问空间分配表 ….. 40表6- 2 FLASH Sector 地址分配表 ….. 42表6- 3 IAP VTOR 寄存器描述 ….. 44表6- 4 FLASH 控制器模块寄存器列表 ….. 45表6- 5 配置寄存器 FLASH\_CFG ….. 45表6- 6 地址寄存器 FLASH\_ADDR ….. 46表6- 7 写数据寄存器 FLASH\_WDATA ….. 47表6- 8 读数据寄存器 FLASH\_RDATA ….. 47表6- 9 擦除控制寄存器 FLASH\_ERASE ….. 48表6- 10 加密状态寄存器 FLASH\_PROTECT ….. 48表6- 11 工作状态寄存器 FLASH\_READY ….. 48表7- 1 DMA 请求 ….. 54表7- 2 DMA 寄存器列表 ….. 56表7- 3 DMA 控制寄存器 DMA\_CTRL ….. 56表7- 4 DMA 中断标志寄存器 DMA\_IF ….. 57表7- 5 DMA 通道配置寄存器 DMA\_CCRx ….. 57表7- 6 DMA 通道请求信号 ….. 58表7- 7 DMA 传输次数寄存器 DMA\_CTMSx ….. 59表7- 8 DMA 外设地址寄存器 DMA\_CPARx ….. 60表7- 9 DMA 内存地址寄存器 DMA\_CMARx ….. 60表8- 1 GPIOx 寄存器列表 ….. 64表8- 2 GPIO 中断/唤醒/配置锁定模块寄存器列表 ….. 64表8- 3 GPIOx 输入使能寄存器 GPIOx PIE ….. 65表8- 4 GPIOx 输出使能寄存器 GPIOx POE ….. 65

表8- 5 GPIOx输入数据寄存器 GPIOx\_PDI. 66表8- 6 GPIOx输出数据寄存器 GPIOx\_PDO. 66表8- 7 GPIOx上拉使能寄存器 GPIOx\_PUE. 67表8- 8 GPIOx开漏使能寄存器 GPIOx\_PODE. 68表8- 9 GPIOx配置锁定寄存器 GPIOx\_LCKR. 68表8- 10 GPIOx功能选择寄存器 GPIOx\_F3210. 69表8- 11 GPIO功能复用. 70表8- 12 GPIOx功能选择寄存器 GPIOx\_F7654. 70表8- 13 GPIOx功能选择寄存器 GPIOx\_FBA98. 70表8- 14 GPIOx功能选择寄存器 GPIOx\_FFEDC. 71表8- 15外部中断配置寄存器EXTI\_CR0. 72表8- 16外部中断配置寄存器EXTI\_CR1. 73表8- 17外部中断标志寄存器EXTI\_IF. 74表8- 18锁定保护寄存器LCKR\_PRT. 75表8- 19外部唤醒源极性配置寄存器WAKE\_POL. 76表8- 20外部唤醒源使能寄存器WAKE\_EN. 76表8- 21 GPIO上拉资源分布表. 77表9- 1 CRC寄存器列表. 82表9- 2 CRC数据寄存器 CRC\_DR. 83表9- 3 CRC控制寄存器 CRC\_CR. 83表9- 4 CRC初始码寄存器 CRC\_INIT. 84表9- 5 CRC生成码寄存器 CRC\_POL. 85表10- 1 ADC分段触发方式触发源配置. 88表10- 2 四段触发分段采样通道数示例. 88表10- 3 ADC通道选择与寄存器配置对应关系. 89表10- 4 ADC输出数字量数制转换. 90表10- 5 ADC0寄存器列表. 93表10- 6采样数据寄存器ADC0\_DAT0. 94表10- 7采样数据寄存器ADC0\_DAT1. 95表10- 8采样数据寄存器ADC0\_DAT2. 95

表10- 9采样数据寄存器ADC0\_DAT3. .95表10- 10采样数据寄存器ADC0\_DAT4. .96表10- 11采样数据寄存器ADC0\_DAT5. .96表10- 12采样数据寄存器ADC0\_DAT6. .97表10- 13采样数据寄存器ADC0\_DAT7. .97表10- 14采样数据寄存器ADC0\_DAT8. .97表10- 15采样数据寄存器ADC0\_DAT9. .98表10- 16采样数据寄存器ADC0\_DAT10. .98表10- 17采样数据寄存器ADC0\_DAT11. .99表10- 18采样数据寄存器ADC0\_DAT12. .99表10- 19采样数据寄存器ADC0\_DAT13. .99表10- 20采样数据寄存器ADC0\_DAT14. .100表10- 21采样数据寄存器ADC0\_DAT15. .100表10- 22采样数据寄存器ADC0\_DAT16. .101表10- 23采样数据寄存器ADC0\_DAT17. .101表10- 24采样数据寄存器ADC0\_DAT18. .101表10- 25采样数据寄存器ADC0\_DAT19. .102表10- 26信号来源寄存器ADC0\_CHN0. .102表10- 27信号来源寄存器ADC0\_CHN1. .103表10- 28信号来源寄存器ADC0\_CHN2. .103表10- 29信号来源寄存器ADC0\_CHN3. .104表10- 30信号来源寄存器ADC0\_CHN4. .104表10- 31信号来源寄存器ADC0\_CHN5. .105表10- 32信号来源寄存器ADC0\_CHN6. .105表10- 33信号来源寄存器ADC0\_CHN7. .106表10- 34信号来源寄存器ADC0\_CHN8. .106表10- 35信号来源寄存器ADC0\_CHN9. .107表10- 36ADC采样信号通道选择. .107表10- 37分段通道数寄存器ADC0\_CHNT0. .108表10- 38分段通道数寄存器ADC0\_CHNT1. .108

表10- 39模式配置寄存器ADC0\_CFG. .109表10- 40触发控制寄存器ADC0\_TRIG. .109表10- 41增益选择寄存器ADC0\_GAIN0. .111表10- 42增益选择寄存器ADC0\_GAIN1. .111表10- 43中断使能寄存器ADC0\_IE. .112表10- 44中断标志寄存器ADC0\_IF. .113表10- 45软件触发寄存器ADC0 SWT. .113表10- 46直流偏置寄存器ADC0\_DC\_A0. .114表10- 47直流偏置寄存器ADC0\_DC\_A1. .114表10- 48直流偏置寄存器ADC0\_DC\_B0. .115表10- 49直流偏置寄存器ADC0\_DC\_B1. .115表10- 50增益校正寄存器ADC0\_AMC\_A0. .116表10- 51增益校正寄存器ADC0\_AMC\_A1. .116表10- 52增益校正寄存器ADC0\_AMC\_B0. .116表10- 53增益校正寄存器ADC0\_AMC\_B1. .117表10- 54通道0阈值寄存器ADC0\_DAT0\_TH. .117表10- 55ADC采样触发模式. .119表11- 1编码器正交编码工作模式. .129表11- 2编码器符号加脉冲工作模式. .130表11- 3编码器CCW/CW双脉冲工作模式. .131表11- 4通用定时器配置寄存器地址分配. .131表11- 5UTIMER配置寄存器UTIMER\_CFG. .133表11- 6滤波控制寄存器UTIMER\_FLT\_TH01. .133表11- 7滤波控制寄存器UTIMER\_FLT\_TH23. .134表11- 8Timer0配置寄存器UTIMER\_UNTO\_CFG. .135表11- 9Timer0配置寄存器UTIMER\_UNTO\_CFG. .136表11- 10Timer2配置寄存器UTIMER\_UNT2\_CFG. .137表11- 11Timer3配置寄存器UTIMER\_UNT3\_CFG. .138表11- 12Timer0门限寄存器UTIMER\_UNTO\_TH. .139表11- 13Timer1门限寄存器UTIMER\_UNT1\_TH. .140

表11- 14 Timer 2 门限寄存器UTIMER\_UNT2\_TH .140表11- 15 Timer 3 门限寄存器UTIMER\_UNT3\_TH .141表11- 16 Timer 0 计数寄存器UTIMER\_UNTO\_CNT. .141表11- 17 Timer 1 计数寄存器UTIMER\_UNT1\_CNT. .142表11- 18 Timer 2 计数寄存器UTIMER\_UNT2\_CNT. .142表11- 19 Timer 3 计数寄存器UTIMER\_UNT3\_CNT. .142表11- 20 Timer 0 通道0 比较捕获寄存器UTIMER\_UNTO\_CMP0. .143表11- 21 Timer 0 通道1 比较捕获寄存器UTIMER\_UNTO\_CMP1. .143表11- 22 Timer 1 通道0 比较捕获寄存器UTIMER\_UNT1\_CMP0. .144表11- 23 Timer 1 通道1 比较捕获寄存器UTIMER\_UNT1\_CMP1. .144表11- 24 Timer 2 通道0 比较捕获寄存器UTIMER\_UNT2\_CMP0. .145表11- 25 Timer 2 通道1 比较捕获寄存器UTIMER\_UNT2\_CMP1. .145表11- 26 Timer 3 通道0 比较捕获寄存器UTIMER\_UNT3\_CMP0. .146表11- 27 Timer 3 通道1 比较捕获寄存器UTIMER\_UNT3\_CMP1. .146表11- 28 Timer 0 外部事件选择寄存器UTIMER\_UNTO\_EVT .147表11- 29 Timer 1 外部事件选择寄存器UTIMER\_UNT1\_EVT .147表11- 30 Timer 2 外部事件选择寄存器UTIMER\_UNT2\_EVT .148表11- 31 Timer 3 外部事件选择寄存器UTIMER\_UNT3\_EVT .149表11- 32 Encoder0 配置寄存器UTIMER\_ECD0\_CFG. .149表11- 33 Encoder0 计数门限寄存器UTIMER\_ECD0\_TH. .150表11- 34 Encoder0 计数值寄存器UTIMER\_ECD0\_CNT. .150表11- 35 中断使能寄存器UTIMER\_IE. .151表11- 36 中断标志寄存器UTIMER\_IF. .152表11- 37 DMA 请求使能寄存器RE. .153表12- 1HALL 模块寄存器地址分配. .157表12- 2HALL 模块配置寄存器HALL\_CFG. .157表12- 3HALL 模块信息寄存器HALL\_INFO. .158表12- 4HALL 宽度计数值寄存器HALL\_WIDTH. .159表12- 5HALL 模块计数器门限值寄存器HALL\_TH. .160表12- 6HALL 计数寄存器HALL\_CNT. .160

表13- 1 MCPWM计数器阈值与事件对应表 .169表13- 2 MCPWM模块寄存器列表 .169表13- 3受MCPWM\_PRT保护的寄存器 .170表13- 4存在影子寄存器的寄存器 .171表13- 5 MCPWM\_TH00配置寄存器 .171表13- 6 MCPWM\_TH01配置寄存器 .172表13- 7 MCPWM\_TH10配置寄存器 .172表13- 8 MCPWM\_TH11配置寄存器 .173表13- 9 MCPWM\_TH20配置寄存器 .173表13- 10 MCPWM\_TH21配置寄存器 .174表13- 11 MCPWM\_TH30配置寄存器 .174表13- 12 MCPWM\_TH31配置寄存器 .174表13- 13 MCPWM\_TMR0配置寄存器 .175表13- 14 MCPWM\_TMR1配置寄存器 .175表13- 15 MCPWM\_TMR2配置寄存器 .176表13- 16 MCPWM\_TMR3配置寄存器 .176表13- 17 MCPWM\_TH配置寄存器 .177表13- 18 MCPWM\_UPDATE配置寄存器 .177表13- 19 MCPWM\_IE配置寄存器 .178表13- 20 MCPWM\_IF配置寄存器 .179表13- 21 MCPWM\_EIE配置寄存器 .180表13- 22 MCPWM\_EIF配置寄存器 .181表13- 23 MCPWM\_RE配置寄存器 .182表13- 24 MCPWM\_PP配置寄存器 .182表13- 25 MCPWM\_IO01配置寄存器 .183表13- 26 MCPWM\_IO23配置寄存器 .184表13- 27 MCPWM\_SDCFG配置寄存器 .185表13- 28 MCPWM\_TCLK配置寄存器 .185表13- 29 MCPWM\_FAIL配置寄存器 .186表13- 30 MCPWM\_PRT配置寄存器 .187

表13- 31 MCPWM\_CNT配置寄存器. .188表13- 32 MCPWM\_DTH00配置寄存器 .188表13- 33 MCPWM\_DTH01配置寄存器 .189表13- 34 MCPWM\_DTH10配置寄存器 .189表13- 35 MCPWM\_DTH11配置寄存器 .190表13- 36 MCPWM\_DTH20配置寄存器 .190表13- 37 MCPWM\_DTH21配置寄存器 .190表13- 38 MCPWM\_DTH30配置寄存器 .191表13- 39 MCPWM\_DTH31配置寄存器 .191表14- 1 UART波特率配置示例 .192表14- 2 UARTx地址分配列表 .194表14- 3 UARTx控制寄存器UARTx\_CTRL .194表14- 4 UARTx波特率设置高字节寄存器UARTx\_DIVH .195表14- 5 UARTx波特率设置低字节寄存器UARTx\_DIVL .195表14- 6 UARTx收发缓冲寄存器UARTx\_BUFF .196表14- 7 UARTx地址匹配寄存器UARTx\_ADR .196表14- 8 UARTx状态寄存器UARTx\_STT .197表14- 9 UARTx中断使能寄存器UARTx\_IE .197表14- 10 UARTx中断使能寄存器UARTx\_IF .198表14- 11 UARTx中断使能寄存器UARTx\_INV .199表15- 1 DSP核心寄存器 .201表15- 2 DSP地址空间 .202表15- 3 DSP寄存器列表 .203表15- 4 DSP状态控制寄存器DSP\_SC .204表15- 5 DSP sin/cos角度输入寄存器 .205表15- 6 DSP sin/cos正弦结果寄存器 .205表15- 7 DSP sin/cos余弦结果寄存器 .205表15- 8 DSP arctan/module坐标X输入寄存器 .206表15- 9 DSP arctan/module计算坐标Y输入寄存器 .206表15- 10 DSP arctan向量模结果sqrt 寄存器 .207

表15- 11 DSP arctan角度结果 arctan(Y/X) 角度寄存器. .207表15- 12DSP除法被除数寄存器 .208表15- 13DSP除法除数寄存器 .208表15- 14DSP除法商寄存器 .209表15- 15DSP除法余数寄存器 .209表15- 16DSP被开方数寄存器 .210表15- 17DSP平方根寄存器 .210表16- 1I2C寄存器地址分配表 .238表16- 2地址寄存器I2C0\_ADDR .238表16- 3系统控制寄存器I2C0\_CFG .239表16- 4状态控制寄存器I2C0\_SCR .239表16- 5数据寄存器I2C0\_DATA .240表16- 6主模式寄存器I2C0\_MSCR .241表16- 7DMA传输控制寄存器I2C0\_BCR .242表17- 1SPI模块控制寄存器列表 .251表17- 2系统控制寄存器SPI\_CFG .251表17- 3SPI\_IE中断寄存器 .252表17- 4SPI\_DIV控制寄存器 .253表17- 5SPI\_TX\_DATA数据发送寄存器 .254表17- 6SPI\_RX\_DATA数据接收寄存器 .254表17- 7SPI\_SIZE数据传输长度寄存器 .254表18- 1比较器模块寄存器列表 .258表18- 2比较器中断使能寄存器CMP\_IE .258表18- 3比较器中断标志寄存器CMP\_IF .258表18- 4比较器分频时钟控制寄存器CMP\_TCLK .259表18- 5比较器控制寄存器CMP\_CFG .260表18- 6比较器开窗控制寄存器CMP\_BLCWIN .262表19- 1发送帧结构 .271表19- 2发送SFF头信息 .271表19- 3发送EFF头信息 .272

表19- 4接收帧结构 .272表19- 5接收SFF头信息 .273表19- 6接收EFF头信息 .273表19- 7CAN寄存器地址分配 .276表19- 8模式寄存器CAN\_MOD .278表19- 9命令寄存器CAN\_CMR .278表19- 10状态寄存器CAN\_SR .279表19- 11中断状态寄存器CAN\_IR .280表19- 12中断控制寄存器CAN\_EIR .281表19- 13波特率0控制寄存器CAN\_BTR0 .281表19- 14波特率1控制寄存器CAN\_BTR1 .282表19- 15仲裁丢失捕捉寄存器CAN\_ALC .282表19- 16错误码捕捉寄存器CAN\_ECC .283表19- 17错误&警告门限值寄存器CAN\_EWLR .284表19- 18接收错误计数器寄存器CAN\_RXERR .285表19- 19发送错误计数器寄存器CAN\_TXERR .285表19- 20发送接收寄存器CAN\_TXRX .286表19- 21ID寄存器CAN\_ACR .286表19- 22ID掩码寄存器CAN\_AMR .287表19- 23FIFO有效接收数据寄存器CAN\_RMC .287表19- 24有效接收数据地址寄存器CAN\_RBSA .287表19- 25RXFIFO寄存器CAN\_RFIFO0~CAN\_RFIFO31 .288表19- 26TXFIFO寄存器CAN\_TFIFO0~CAN\_TFIFO12 .288表20- 1SIF模块控制寄存器列表 .291表20- 2地址寄存器SIF\_CFG .292表20- 3系统控制寄存器SIF\_FREQ .292表20- 4状态控制寄存器SIF\_IRQ .293表20- 5数据寄存器SIF\_WDATA .293表21- 1看门狗模块寄存器 .294表21- 2看门狗清零寄存器SYS\_WDT\_CLR .294

表 22- 1 文档版本历史 296

# 图片目录

图 4- 1 模拟电路功能框图 ….. 9图 4- 3 温度传感器曲线 ….. 12图 5- 1 时钟架构 ….. 16图 5- 2 硬件复位架构 ….. 19图 5- 3 SYS\_CLK\_CFG 不同配置时的系统主时钟分频波形 ….. 31图 6- 1 FLASH 存储体空间划分框图 ….. 38图 6- 2 FLASH 控制状态转换图 ….. 39图 6- 3 FLASH 间接读取操作流程图 ….. 40图 6- 4 FLASH 模块编程操作流程图 ….. 41图 6- 5 FLASH 模块编程操作流程图 ….. 42图 6- 6 FLASH 模块擦除操作流程图 ….. 43图 7- 1 multi- layer AHB lite 总线架构 ….. 52图 7- 2 DMA 地址递增控制 ….. 53图 7- 3 DMA 通道优先级 ….. 55图 8- 1 GPIO 功能框图 ….. 62图 10- 1 ADC 采集模块功能框图 ….. 87图 10- 2 ADC 中断产生 ….. 89图 10- 3 一倍增益设置下 ADC 模数转换数制量程 ….. 90图 10- 4 ADC 单段采样状态转移图 ….. 120图 10- 5 ADC 两段采样状态转移图 ….. 121图 10- 6 ADC 四段采样状态转移图 ….. 121图 11- 1 模块顶层功能框图 ….. 124图 11- 2 滤波示意图 ….. 126图 11- 3 通用计数器 ….. 127图 11- 4 比较模式 ….. 128图 11- 5 捕获模式 ….. 128图 11- 6 编码器只在 T1 时刻计数的正交编码信号计数情况 ….. 129图 11- 7 编码器在 T1 或 T2 时刻计数的正交编码信号计数情况 ….. 130

图11- 8编码器在T1上升下降沿都计数的符号加脉冲信号计数情况 .130图11- 9编码器在仅T1上升沿计数的符号加脉冲信号计数情况 .130图11- 10编码器仅在T1/T2上升沿计数的CCW/CW双脉冲信号计数情况 .131图11- 11编码器在T1/T2上升下降沿计数的CCW/CW双脉冲信号计数情况 .131图12- 17/5滤波模块框图 .156图12- 2数据流程框图 .157图13- 1MCPWM模块框图 .161图13- 2BaseCounter t0/t1时序 .162图13- 3MCPWM更新机制 .163图13- 4MCPWMFAIL逻辑示意图 .163图13- 5MCPWMFail信号滤波时钟生成逻辑 .164图13- 6IODriver模块数据流程图 .165图13- 7MCPWM时序 和 - 互补模式 .166图13- 8MCPWM时序 和 - 互补模式 .166图13- 9MCPWM时序边沿对齐模式 .167图13- 10MCPWM时序 和 边沿对齐推挽模式 .167图13- 11MCPWMIO控制示意图 .168图15- 1DSP模块功能框图 .201图15- 2CPU访问DSPMEM一致性问题 .226图15- 3增加Dummy写操作解决DSPMEM一致性问题 .226图15- 4CPU因命中cache无法读取DSPDATAMEM真实数据 .226图16- 1I2C模块顶层功能框图 .228图16- 2基本I2C传输时序图 .229图16- 3从模式下单字节传输示意图 .231图16- 4从模式下多字节发送示意图 .232图16- 5从模式下多字节接收示意图 .232图16- 6主模式下单字节传输示意图 .233图16- 7主模式下多字节发送示意图 .235图16- 8主模式下多字节接收示意图 .235

图17- 1SPI模块结构框图 .244图17- 2SPI接口全双工模式互连框图 .245图17- 3SPI接口半双工模式互连框图 .246图17- 4SPI模块Slave模式片选信号选择 .247图17- 5SPI模块Master模式片选信号选择 .247图17- 6SPI通讯信号极性相位(Polarity ,Phase .248图17- 7SPI通讯信号极性相位(Polarity ,Phase .248图17- 8SPI通讯信号极性相位(Polarity ,Phase .248图17- 9SPI通讯信号极性相位(Polarity ,Phase .249图17- 10SPI模块中断选信号产生图 .250图4- 2BEMFx\_MID信号 .257图18- 1比较器滤波时钟产生 .260图18- 2比较器控制及中断产生逻辑 .261图18- 3CMP与MCPWM的联动 .261图18- 4比较器开窗功能图示 .262图19- 1CAN模块顶层功能框图 .264图19- 2CAN模块Bit周期介绍图 .268图19- 3CAN模块ID滤波逻辑关系 .271图19- 4CAN模块错误管理 .274图20- 1SIF模块顶层功能框图 .290图20- 2SIF基本传输时序图 .291

# 1 文档约定

# 1.1 寄存器读写权限

RW 读/写，软件可以读写这些位。 RO 只读，软件只能读取这些位。 WO 只写，软件只能写入该位。读取该位时将返回默认值。 RW1C(Read and Write 1 to Clear) 可读，写 1 清零。

# 1.2 缩略词汇表

字：32位数据/指令。

半字：16位数据/指令。

字节：8位数据。

双字：64位数据。

ADC：Analog- Digital Converter，模数转换器

DAC：Digital- Analog Converter，数模转换器

BGP：Bandgap，带隙基准

WDT：Watch dog，看门狗

LSI：Low Speed Internal Clock，即 32kHz RC 时钟

HSI：High Speed Internal Clock，即 4MHz RC 时钟

HSE：High Speed External Clock，即 4~8MHz 外部晶振时钟

PLL：Phase Lock Loop Clock，即 96MHz 锁相环时钟，通常用作系统高速时钟

POR：Power- On Reset，即上电复位，芯片系统上电时产生的复位信号

NVR：Non- Volatile Register，flash 中区别于 main 区域之外的一块存储区域

IAP（在应用中编程）：IAP是指可以在用户程序运行期间对微控制器的Flash进行重新编程。ICP（在线编程）：ICP是指可以在器件安装于用户应用电路板上时使用JTAG协议、SWD协议或自举程序对微控制器的Flash进行编程。

CW：Clockwise，顺时针

CCW：Counter clockwise，逆时针

Option bytes：选项字节，保存在 Flash 中的 CPU 配置字节

# 2 地址空间

数据字节以小端格式存放在存储器中。一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间。

表2-1系统地址空间分配

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 外设 | 时钟/软复位 | 开始地址 | 结束地址 | 大小 | 说明 |
| FLASH | PLL/无 | 0x0000\_0000 | 0x0000\_FFFF | 64kB | FLASH 存储空间 |
| RAM | PLL/无 | 0x2000\_0000 | 0x2000\_1FFF | 8kB | RAM |
| SYS | PLL/无 | 0x4000\_0000 | 0x4000\_03FF | 1kB | SYSTEM control, Clock / Reset Management |
| FLSCR | PLL/无 | 0x4000\_0400 | 0x4000\_07FF | 1kB | FLASH control registers |
| SPI | FCLK[8]/sft rst[0] | 0x4001\_0000 | 0x4001\_03FF | 1kB | SPI interface |
| I2C | FCLK[0]/sft rst[0] | 0x4001\_0400 | 0x4001\_07FF | 1kB | I2C interface |
| CMP | PLL/无 | 0x4001\_0C00 | 0x4001\_0FFF | 1kB | Comparator |
| HALL | FCLK[1]/sft rst[1] | 0x4001\_1000 | 0x4001\_13FF | 1kB | HALL interface |
| ADC | ACLK | 0x4001\_1400 | 0x4001\_17FF | 1kB | ADC interface |
| TIMER | FCLK[2]/sft rst[2] | 0x4001\_1800 | 0x4001\_1BFF | 1kB | General Purpose Timer |
| MCPWM | FCLK[3]/sft rst[3] | 0x4001\_1C00 | 0x4001\_1FFF | 1kB | Motor Control Pulse Width Modulation |
| GPIO | PLL/无 | 0x4001\_2000 | 0x4001\_23FF | 1kB | General Purpose Input / Output |
| CRC | PLL/无 | 0x4001\_2400 | 0x4001\_27FF | 1kB | Cyclic Redundancy Check |
| UART0 | FCLK[4]/sft rst[4] | 0x4001\_2800 | 0x4001\_2BFF | 1kB |  |
| UART1 | FCLK[5]/sft rst[5] | 0x4001\_2C00 | 0x4001\_2FFF | 1kB |  |
| DMA | PLL/无 | 0x4001\_3000 | 0x4001\_33FF | 1kB |  |
| CAN | FCLK[7]/无 | 0x4001\_3400 | 0x4001\_37FF | 1kB |  |
| SIF | PLL/无 | 0x4001\_3800 | 0x4001\_3BFF | 1kB |  |
| DSP | FCLK[6]/无 | 0x4001\_4000 | 0x4001\_5FFF | 8kB |  |

# 3 内核

# 3.1 ARM@CortexTM-M0核心

Cortex- M0是Cortex- M家族中的M0系列。最大特点是低功耗的设计。Cortex- M0为32位、3级流水线RISC处理器，其核心仍为冯.诺依曼结构，是指令和数据共享同一总线的架构。作为新一代的处理器，Cortex- M0的设计进行了许多的改革与创新，如系统存储器地址映像(System AddressMap)、改善效率并增强确定性的嵌套向量中断系统(NVIC)与不可屏蔽中断(NMI)、全新的调试单元等等，都带给了使用者全新的体验和更便利、更有效率的操作。

Cortex- M0其核心架构为ARMv6M，其运算能力可以达到0.9DMIPS/MHz，而与其他的16位与8位处理器相比，由于CortexM0的运算性能大幅提高，所以在同样任务的执行上Cortex- M0只需较低的运行速度，而大幅降低了整体的动态功耗。

有关内核的更多信息请参考ARM官方技术手册”Cortex- M0 Technical Reference Manual”（汉译版本为”Cortex- M0技术参考手册”)。

本小节重点描述NVIC控制器、SysTick定时器、LKS32MC08x中断向量表定义及中断向量表重定义三个功能。

# 3.2 NVIC控制器

为了管理中断请求的优先级并处理其他异常，Cortex- M0处理器内置了嵌套中断控制器(NVIC)。NVIC的一些可编程控制器控制着中断管理功能，这些寄存器被映射到系统地址空间里，它们所处的区域被称为系统控制空间（SCS)。SCS的地址空间为：0xE000\_E000- 0xE000\_EFFF。

NVIC有以下特性：

灵活的中断管理；

Cortex- M0处理器中，每一个外部中断都可以被使能或者禁止，并且可以被设置为挂起状态或者清除状态。处理器的中断可以是电平信号的（在中断服务程序清除中断请求以前，外设的请求会一直保持)，也可以是脉冲信号的（最小一个始终周期)，这样中断控制器就可以处理任何中断源。

支持嵌套中断；

Cortex- M0处理器的任何中断都有一个固定或者可编程的中断优先级。当外部中断之类的异常发生时，NVIC将该异常的优先级与当前的优先级进行比较，如果新的优先级更高，当前的任务会被暂停，处理器进行核心寄存器入栈保持，然后开始处理新的异常程序，这个过程也被称为“抢占”。高优先级的中断完成后，异常返回就会执行，处理器自动进行出栈操作恢复刚才寄存器的值，并继续运行刚才的任务。这种机制并没有带来软件开销。

向量化的异常入口；

异常发生时，处理器需要定位异常对用的程序入口。传统的处理方式需要软件去完成。而M0处理器会从存储器的向量表中，自动定位异常的程序入口。从异常到异常的处理事件会被缩减。

中断屏蔽；

NVIC通过一组特殊寄存器提供了一种中断屏蔽机制，NVIC除了硬件错误（HardFault）和NMI之外，可以屏蔽所有的异常。有些操作，比如对时间敏感的控制任务或实时多媒体解码任务，不应该被打断，此时中断屏蔽的作用就表现了出来。具体参见ARM手册。

LKS32MC08x支持ARM公司推出的CMSIS(CortexMicrocontrollerSoftwareInterfaceStandard，即Cortex微控制器软件接口标准）函数。用户在工程中包含cm0\_core.h和cm0\_core.c代码，即可实现对NVIC模块的操作。

# 3.3 异常和中断

异常，会引起程序控制的变化。在异常发生时，处理器停止当前的任务，转而执行异常处理程序，异常处理完成后，会继续执行刚才的任务。异常分为很多种，中断是其中之一。Cortex- M0处理器最多支持32个外部中断（IRQ）和一个不可屏蔽中断（NMI)，中断事件的处理叫做中断服务程序(ISR)，中断一般由芯片上的硬件资源产生。

每一个异常都对应一个异常编号，异常编号还指明了异常向量的地址。异常编号和中断编号是相互独立的。系统异常使用负数定义，中断使用0- 31正数定义。复位是一种特殊的异常，数值为- 15。除了NMI，HardFault和复位，其他所有异常的优先级都是可编程的，NMI和HardFault的优先级是固定的，并且比其他异常的优先级高。

LKS32MC08x系列芯片共使用了其中的21个外部中断，后11个保留未使用。NMI未实现。最多支持4个中断优先级可供编程选择。

表3-1中断号分布

|  |  |  |  |
| --- | --- | --- | --- |
| 异常/中断号 | 说明 | 异常/中断号 | 说明 |
| -14 | NMI |  | |
| -13 | HardFault |
| -12 | 保留 |
| -11 |
| -10 |
| -9 |
| -8 |
| -7 |
| -6 |
| -5 | SVCall |
| -4 | 保留 |
| -3 |
| -2 | PendSV |
| -1 | SysTick |
| 0 | TIMERO | 16 | WAKEUP，系统唤醒中断 |
| 1 | TIMER1 | 17 | 电源电压过低 |
| 2 | TIMER2 | 18 | DMA |
| 3 | TIMER3 | 19 | CAN |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | ENCODERO | 20 | SIF |
| 5 | ENCODER1 | 21 | Reserved |
| 6 | I2C | 22 | Reserved |
| 7 | GPIO | 23 | Reserved |
| 8 | UART0 | 24 | Reserved |
| 9 | HALL | 25 | Reserved |
| 10 | SPI | 26 | Reserved |
| 11 | ADC | 27 | Reserved |
| 12 | DSP | 28 | Reserved |
| 13 | MCPWM | 29 | Reserved |
| 14 | UART1 | 30 | Reserved |
| 15 | CMP | 31 | Reserved |

异常发生时，处理器需要定位异常对应的程序入口。标准的 Cortex- M0处理器不支持中断向量表的重定义，即每个异常的入口地址是固定的。在线升级应用，将擦除FLASH部分内容，用户可能擦除/破坏中断向量表。因此，此种应用，一般推荐关闭中断。LKS32MC08x芯片增加了该功能，实现了中断向量表的重定义功能。即在升级过程中，可开启中断，编程更灵活。具体使用，参见本文档FLASH章节- - FLASH在线升级(IAP)部分的描述。

# 3.4 SysTick定时器

SysTick定时器是Cortex- M0处理器自带的系统滴答定时器。其存在的主要目的是为嵌入式操作系统提供 (即 )的定时节拍。当然，也可以做普通定时等其他用途。

可编程设置频率的RTOS定时器(例如 )，调用一个SysTick服务程序。简单计数器。软件可使用它测量时间（如：完成任务所需时间、已使用时间)。

SysTick定时器使用起来非常简单。它一共有三个寄存器：SYST\_CSR、SYST\_RVR、SYST\_CVR。寄存器的基地址为0xE000\_E000。LKS32MC08x芯片，针对实际应用，三个寄存器的配置如下。

表3-2中断号分布

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 描述 |
| SYST\_CSR | 0x10 | SysTick 定时器控制和状态寄存器 |
| SYST\_RVR | 0x14 | SysTick 定时器重载值寄存器 |
| SYST\_CVR | 0x18 | SysTick 定时器当前值寄存器 |

# 3.4.1 SYST\_CSR控制和状态寄存器

地址：0xE000\_E010

复位值：0x00000000

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3-3 SYST\_CSR 控制和状态寄存器 | | | | | | | | | | | | | |  |  |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TICKINT | ENABLE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW | RW |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [16] | FLAG | SYSTick 定时器计数器回零标志位，读清除。 0：没有回零 1：发生回零 |
| [15:3] | – | 保持为 0 |
| [2] | CLKSRC | SYSTick 定时器时钟选择信号。默认为 0。 0：CPU 时钟/8 1：CPU 时钟 |
| [1] | TICKINT | SYSTick 定时器中断使能信号。默认为 0。 0：关闭 1：使能 |
| [0] | ENABLE | SYSTick 定时器使能信号。默认为 0。 0：关闭 1：使能 |

# 3.4.2 SYST\_RVR重载值寄存器

地址：0xE000\_E014

复位值：0x00000000

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | RELOAD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:24] | – | 保持为 0 |
| [23:0] | RELOAD | SysTick 定时器周期寄存器，定时器在 0—RELOAD 之间计数。默认为 0。 |

# 3.4.3 SYST\_CVR当前值寄存器

地址：0xE000\_E018

复位值：0x00000000

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|  | | | | | | | | CURRENT | | | | | | | |
| RW | | | | | | | |
| 0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | RELOAD | | | | | | | |
|  | | | | | | | | RW | | | | | | | |
|  | | | | | | | | 0 | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:24] | – | 保持为 0 |
| [23:0] | CURRENT | 读取该寄存器，返回 SysTick 定时器内部计数值，默认为 0。对该寄存器写入任何值，将复位内部计数值，同时清除 SYST\_CSR.FLAG。 |

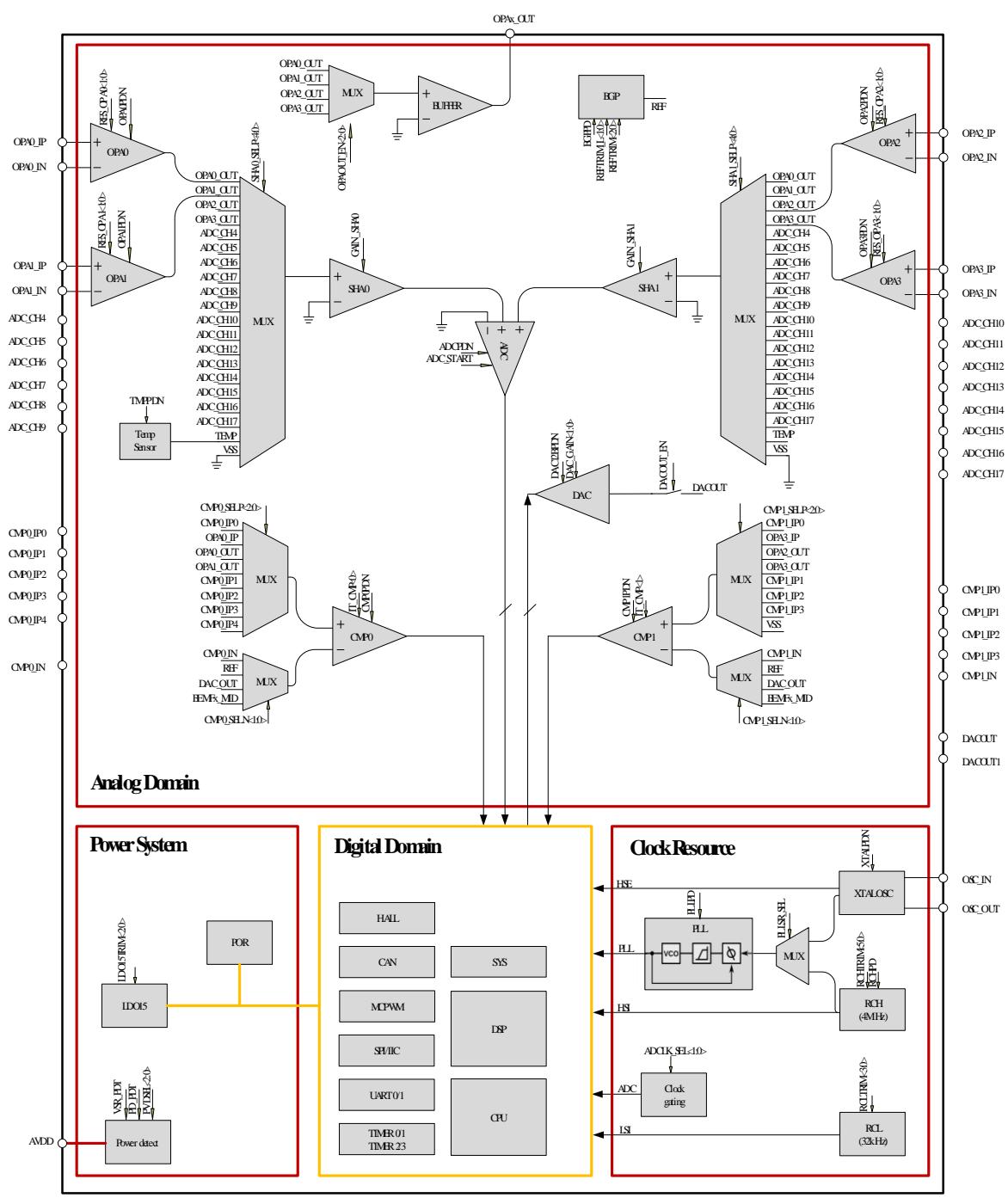
# 4 模拟电路

# 4.1 简述

模拟电路包含以下模块：

集成1路同步双采样的12bit SAR ADC，采样及转换速率3Msps。最多20通道 > 集成4路运算放大器，可设置为PGA模式 > 集成两路比较器，可设置迟滞模式 > 集成12bit数模转换器 > 内置 温度传感器 > 内置1.2V0.8%精度电压基准源

各个模块之间的相互关系、以及各模块的控制寄存器（寄存器的说明见下文“模拟寄存器表”）如下图所示。

  
图4-1模拟电路功能框图

# 4.2 POWER 电源管理系统

电源管理系统由电源检测模块（PVD）、上电/掉电复位模块（POR）组成。

POR 模块监测 AVDD 的电压，在 AVDD 电压低于 3.0V 时（例如上电之初，或者掉电时），为数字电路提供复位信号以避免数字电路工作产生异常。

PVD 模块对 5V 输入电源进行检测，如低于某一设定阈值，则产生报警（中断）信号以提醒 CPU，欠压只产生欠压标志，不会产生复位。中断提醒阈值可通过寄存器 PVDSEL[1:0] 设置为不同的电压。

PVD 模块可通过设置 PD\_PDT=1 关闭。

当发生电压过低事件时，会触发产生电源电压过低中断，对应中断号 17，详见第 3 章。

PVDSEL[1:0]/ PD\_PDT 的说明见模拟寄存器 SYS\_AFE\_REG6

电源电压过低标志见模拟前端信息寄存器 SYS\_AFE\_CMP

# 4.3 CLOCK 时钟系统

时钟系统包括内部 32kHzRC 时钟、内部 4MHzRC 时钟、外部 4~8MHz 晶体起振电路、PLL 电路组成。

32kHzRC 时钟 LSI 主要用于系统内的看门狗模块以及复位信号滤波等。4MHzRC 时钟可作为 CPU 主时钟使用，CPU 也可以使用 PLL 时钟，PLL 最高可提供 96MHz 的时钟。外部 4~8MHz 晶体起振电路作为备用时钟使用。

32kHz 和 4MHzRC 时钟均带有出厂校正，32kHzRC 时钟在- 40~105°C 范围内的精度为 ，4MHzRC 时钟在该温度范围的精度为 。

4MHzRC 时钟通过设置 RCHPD=‘0’ 打开（默认打开，写 1 关闭），RC 时钟需要 BGP 电压基准源模块提供基准电压和电流，因此开启 RC 时钟需要先开启 BGP 模块（BGPPD=‘0’）。芯片上电的默认状态下，4MHzRC 时钟和 BGP 模块都是开启的。32kHzRC 时钟始终开启，不能关闭。

PLL 对 4MHzRC 时钟进行倍频，给 CPU、ADC 等模块提供更高速的工作时钟。CPU 和 PWM 模块的最高时钟为 96MHz，ADC 模块最高时钟 48MHz，通过寄存器 ADCLKSEL[1:0] 可设置不同的 ADC 工作频率。

PLL 通过设置 PLLPDN=‘1’ 打开（默认关闭，设 1 打开），开启 PLL 模块之前，同样也需要开启 BGP 模块。开启 PLL 之后，PLL 需要 6us 的稳定时间来输出稳定时钟。芯片上电的默认状态下，RCH 时钟和 BGP 模块都是开启的，但 PLL 默认是关闭的，需要软件来开启。

晶体起振电路内置了放大器，需在 OSC\_IN/OSC\_OUT 管脚之间接入一个晶体，且在 OSC\_IN/OSC\_OUT 引脚各接一个 15pF 电容到地，设置 XTALPDN=1 即可起振。

ADCLKSEL<1:0>的说明见模拟寄存器 SYS\_AFE\_REG7

BGPPD/RCHPD/XTALPDN/PLLPDN 的说明见模拟寄存器 SYS\_AFE\_REG5

# 4.4 BGP 基准电压源

基准源电路(BGP REF Bandgap reference)为 ADC、DAC、RC 时钟、PLL、温度传感器、运算放大器、比较器和 FLASH 提供基准电压和电流，使用上述任何一个模块之前，都需要开启 BGP 基准电压源。

芯片上电的默认状态下，BGP 模块是开启的。通过设置 BGPPD=‘0’ 将基准源打开，从关闭到开启，BGP 需要约 2us 达到稳定。BGP 输出电压约 1.2V，精度为

BGPPD 的说明见模拟寄存器 SYS\_AFE\_REG5

# 4.5 ADC 模数转换器

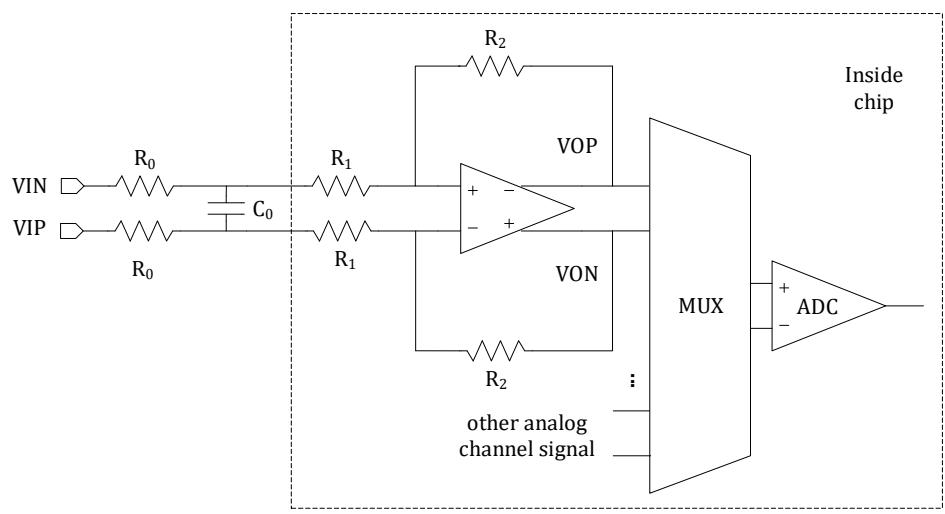
请参加第 10 章 ADC

# 4.6 OPA 运算放大器

芯片集成 4 路输入输出轨到轨（rail- to- rail）运算放大器，内置反馈电阻，外部引脚上需串联一个电阻 到信号源。反馈电阻 的阻值可通过寄存器 RES\_OPAx[1:0]设置，以实现不同的放大倍数。

RES\_OPAx<1:0>的说明见模拟寄存器 SYS\_AFE\_REG0

放大器的结构示意图如下所示：

  
图 4-2 放大器框图

图中两个 是片外需放置的电阻，阻值必须相等，最终的放大倍数为 。

对于 MOS 管电阻直接采样的应用，由于 MOS 下管关断、上管导通时信号会升高到数十 V 的电源电压，为减小此时往芯片引脚里流入的电流，建议接 的外部电阻。

对于分流电阻采样的应用，建议接 的外部电阻。 为信号滤波电容，和 形成一阶 RC 滤波电路。 的具体阻值可根据 的滤波常数而定。如果信号上噪声较小不需要滤波、或者信号需要很大的带宽（较快的响应速度），则 可以不加。

放大器可通过设置 OPAOUT\_EN<2:0>选择将 4 路放大器中的某一路输出信号通过 BUFFER 送至 P2.7 管脚口进行测量和应用（对应关系见 datasheet 芯片管脚说明）。因为有 BUFFER 存在，在运放正常工作模式下也可以选择送一路运放输出信号出来。

OPAOUT\_EN<2:0>的说明见模拟寄存器 SYS\_AFE\_REG2

芯片上电的默认状态下，放大器模块是关闭的。放大器可通过设置 OPAxPDN=1 (x=0,1,2,3) 打开。开启放大器之前，需要先开启 BGP 模块。

OPAxPDN 的说明见模拟寄存器 SYS\_AFE\_REG5

运放输入正负端内置钳位二极管，电机相线通过一个匹配电阻后直接接入输入端，从而简化了 MOSFET 电流采样的外置电路。

# 4.7 CMP 比较器

内置 2 路输入轨到轨（rail- to- rail）比较器，比较器比较速度可编程、迟滞电压可编程、信号源可编程。比较器的比较延时可设置为 。迟滞电压可设置为 。支持多种比较器输入信号来源。

# 4.8 TMP 温度传感器

芯片内置温度传感器，在 范围内典型精度为 。 范围内典型精度为 。

测量时需选择内部基准，且需要设置 SYS\_AFE\_REG1. GAIN\_REF=0，选择外部基准会引起结果的较大偏差。

芯片出厂前会经温度校正，校正值保存在 flash\_info 区(详见 6.3.9)。出厂较正时所用增益为 ADC\_GAIN=0，建议应用时也选择此增益，可以使较正值更准确。

芯片上电的默认状态下，温度传感器模块是关闭的。开启传感器之前，需要先开启 BGP 模块。

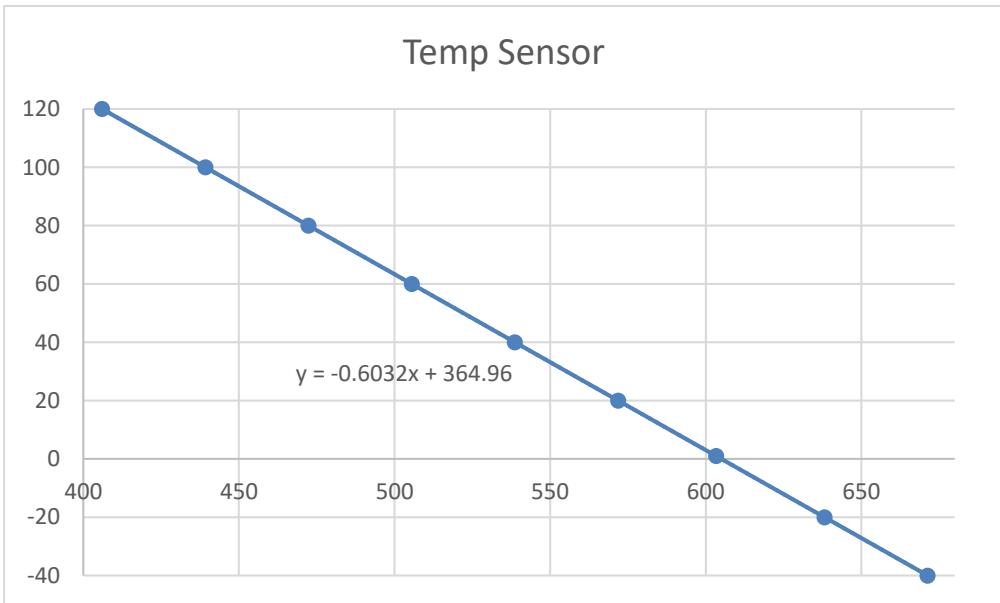
温度传感器通过设置 TMPPDN=1 打开，开启到稳定需要约 2us，因此需在 ADC 测量传感器之前 2us 打开。

温度传感器信号连至 ADC 的通道 18。

ADC 部分的设置参考模数转换器(ADC)章节

TMPPDN 的说明见模拟寄存器 SYS\_AFE\_REG5

温度传感器的典型曲线如下图所示：

  
图 4-2 温度传感器曲线

图中X轴为温度传感器的温度信号所对应的ADC值，Y轴为传感器所处的温度。测温时，按照如上要求配置传感器相关寄存器，并得到ADC值后，将ADC值作为X代入公式：

求得的Y值即为此时的温度。

公式中有两个系数， 。对于不同的芯片，b系数的值是不一样的。芯片出厂前会经过温度标定，将每颗芯片所对应的系数b写入flash的info区，地址为 。存储时，会将b系数小数点右移一位（乘10）存入info区，小数点后第二位不进行保存。

同时为方便客户操作，系数a也会存入flashinfo区，地址为 。存储时，将a系数小数点右移四位（乘10000）存入info区。

实际使用中，应从flashinfo区对应地址读出a/b系数，同时将读取到的ADC测到的当下温度传感器值代入公式，即可计算得到当下温度值，单位为摄氏度。计算时，需注意系数a/b在保存时小数点的位移数，即a系数应除以10000，b系数除以10。

注意，上述计算公式，基于ADC右对齐实现。若换成左对齐，ADC采样值需右移4位后，才能代入上述公式。

# 4.9 DAC数模转换器

芯片内置一路12bitDAC，输出信号的最大量程可通过寄存器DAC\_GAIN<1:0>设置为1.2V/3V/4.85V

12bitDAC可通过配置寄存器DACOUT\_EN ，将DAC输出送至P0.0管脚，可驱动 的负载电阻和50pF的负载电容。

DAC最大输出码率为1MHz。

芯片上电的默认状态下，DAC模块是关闭的。DAC可通过设置DAC12BPDN 打开，开启DAC模块之前，需要先开启BGF模块。

DAC的输入数字信号寄存器为SYS\_AFE\_DAC，低12BIT有效。信号范围是 。 对应零模拟量输出0V，0xFFF对应满量程模拟量输出为 ，如上文所述， 的值可由DAC\_GAIN<1:0>寄存器进行设置。每一档信号(LSB)所对应的模拟信号幅度为 。若SYS\_AFE\_DAC的数字值为Din，则该数字信号所对应的DAC输出模拟信号为 。

不同芯片，DAC存在制造偏差，为抵消偏差，DAC自带校准硬件模块。DAC输出遵循公式 。x为SYS\_AFE\_DAC填入值（理想值对应数字量）。a来自SYS\_AFE\_DAC\_AMC寄存器，b来自SYS\_AFE\_DAC\_DC。硬件根据SYS\_AFE\_DAC、SYS\_AFE\_DAC\_AMC和SYS\_AFE\_DAC\_DC进行乘加从而求得校正后的数字量，送至DAC输入端，使得DAC最终输出的模拟值就是填入的理想数字量对应的值。系统上电后，默认加载3V的校准值，若换成其它量程，软件需读取flashinfo区域，重新加载到相应寄存器。

地址 ，为3V量程的a参数，地址 ，为3V量程的b参数。

地址 0x00000334，为 1.2V 量程的 a 参数，地址 0x00000344，为 1.2V 量程的 b 参数。

地址 0x00000338，为 4.85V 量程的 a 参数，地址 0x00000348，为 4.85V 量程的 b 参数。

DAC 输出的模拟信号，除了可以送至 IO 口供外部模块使用外，还可通过配置寄存器连至芯片内部的 2 路比较器负端，作为比较器的基准信号使用。详见比较器章节。

DACOUT\_EN 的说明见模拟寄存器 SYS\_AFE\_REG3 DAC\_GAIN<1:0>的说明见模拟寄存器 SYS\_AFE\_REG1 DAC12BPDN 的说明见模拟寄存器 SYS\_AFE\_REG5 SYS\_AFE\_DAC 的说明见寄存器 SYS\_AFE\_DAC

# 5 系统控制及时钟复位

# 5.1 时钟

# 5.1.1 时钟源

如下表所示，系统包括5个时钟源。其中内部低速RC振荡时钟LSI（Low Speed Internal Clock）/内部高速RC振荡时钟HSI（High Speed Internal Clock）不会停振。外部晶振时钟HSE（High Speed External Clock）在极端工况下可能失效，仅部分应用会使用HSE。

表5-1系统时钟源

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时钟源 | 频率 | 来源 | 误差 | 说明 |
| LSI | 32kHz | 内部 RC 振荡器 | 常温 23~42kHz 全温度 16~48kHz | 内部系统管理时钟，用于看门狗模块，以及复位信号的滤波和展宽 |
| HSI | 4MHz | 内部 RC 振荡器 | -40~125° 误差&lt;±2.0% | 可作为PLL源时钟 |
| PLL | 96MHz | PLL 可钟 | 0 | PLL 输出时钟，以HSI/HSE 作为参考时钟，输出是HSI/HSE 时钟的24倍频，作为系统主时钟。 |
| HSE | 4~8MHz | 外部晶体振荡器 | 0 | 外部晶体，在对时钟精度有严格要求(例如ppm级别的精度要求)的应用下，可使用4MHz HSE 作为PLL 参考时钟来产生96MHz 的系统主时钟，不支持使用其他频率HSE 作为PLL 参考时钟 |
| SWD | 1MHz\* | 调试器 | - | SWD 的JTAG 时钟 |

\*SWD时钟速率典型值，实际大小与硬件环境有关。

另外，晶振起振电路可以支持 的外部晶振频率，但如果需要使用HSE作为PLL参考时钟，则仍需使用4MHz晶体，以提供与HSI相同频率的PLL参考时钟。如果直接使用HSE作为系统主时钟，则无此限制。

如

图5- 1所示，系统可以使用外部晶振时钟HSE或内部高速RC时钟HSI作为PLL的参考时钟，使用SYS\_AFE\_REG6. PLLSR\_SEL进行选择。同时，芯片内部可以检测HSE时钟是否停振，如果HSE失效会自动切换至HSI作为PLL参考时钟。PLL将4MHz的参考时钟CLK\_HS倍频24倍至96MHz。

PLL经过 分频后，可以得到 的高速时钟，SYS\_CLK\_CFG.CLK\_SEL在此分频后的高速时钟与4MHz的CLK\_HS之间进行二选一，作为系统主时钟MCLK。系统复位时，PLL默认关闭，HSI默认开启，系统选择HSI时钟，即4MHz作为系统主时钟进行工作，从而保证系统上电之

初功耗处于较低水平。

MCLK是系统主时钟。可以通过SYS\_CLK\_CFG寄存器CLK\_DIV位域控制进行n/8分频，可以产生12,24,36,48,60,72,84,96MHz等频率值。SYS\_CLK\_CFG\_CLK\_SEL表示选择PLL或CLK\_HS作为系统主时钟。当SYS\_CLK\_CFG\_CLK\_SEL为1时，SYS\_CLK\_CFG\_CLK\_DIV作为PLL的分频系数。当SYS\_CLK\_CFG\_CLK\_SEL为0时，SYS\_CLK\_CFG\_CLK\_DIV不起任何作用。

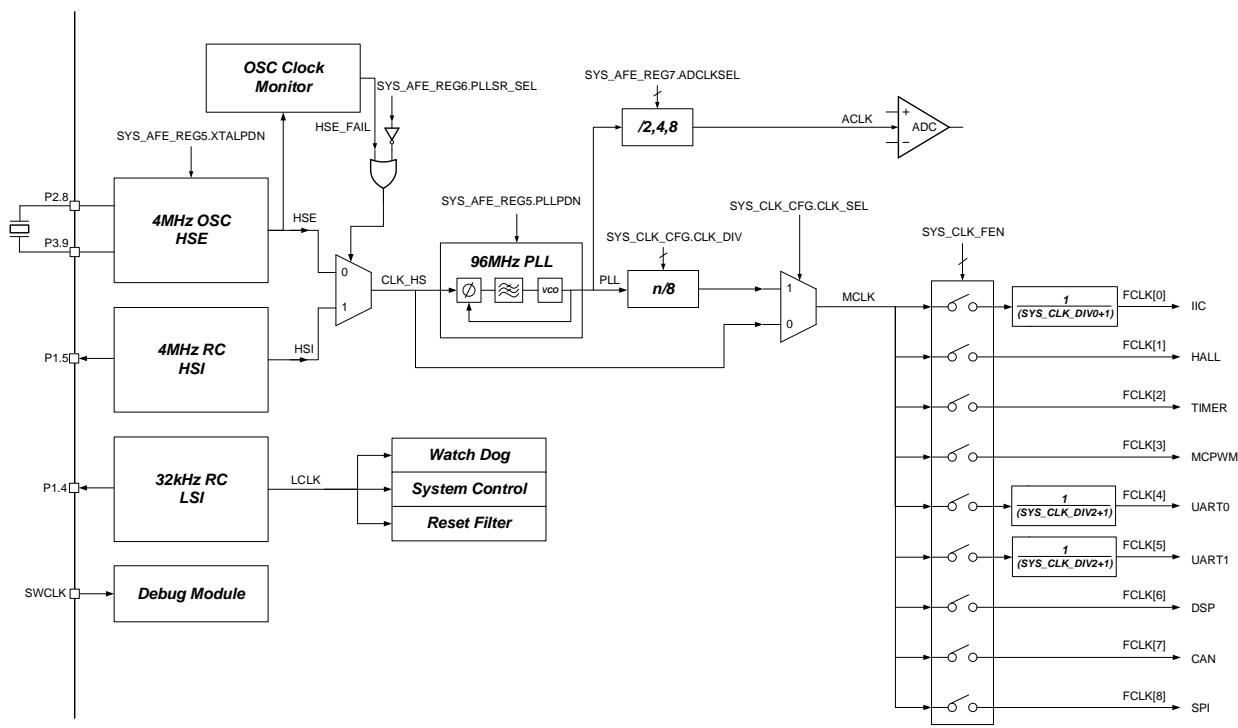
表5-2PLL作为MCLK时钟时的分频配置

|  |  |  |  |
| --- | --- | --- | --- |
| SYS\_CLK\_CFG | 分频系数 | 频率/MHz | 是否均匀 |
| 0x0101 | 1/8 | 12 | 是 |
| 0x0111 | 2/8 | 24 | 是 |
| 0x0115 | 3/8 | 36 | 否 |
| 0x0155 | 4/8 | 48 | 是 |
| 0x0157 | 5/8 | 60 | 否 |
| 0x0177 | 6/8 | 72 | 否 |
| 0x01FF | 7/8 | 84 | 否 |
| 0x01FF | 8/8 | 96 | 是 |

MCLK时钟经过SYS\_CLK\_FEN寄存器控制的开关之后供给外设。I2C时钟由SYS\_CLK\_DIV0寄存器控制可以进一步分频，UART时钟由SYS\_CLK\_DIV2寄存器控制可以进一步分频。

PLL输出的时钟经过SYS\_AFE\_REG7. ADCLKSEL控制的2/4/8分频后送至ADC（典型工作频率48MHz)，即ACLK。

内部32kHzRC产生一路LSI时钟LCLK，主要用于WDT工作时钟，以及部分系统控制，复位的滤波展宽等。

  
图5-1时钟架构

为了保证系统可靠工作，时钟系统有防止时钟被误操作关闭的机制。如当PLL被用作系统工作主时钟时，PLL本身无法被关闭，作为PLL参考时钟的HSI或HSE无法被软件关闭；当CLK\_HS作为系统工作主时钟时，作为CLK\_HS的HSI或HSE无法被关闭；当HSE被使用时，如果系统检测到HSE停振会自动切换到使用HSI时钟来代替HSE时钟。32kHz LSI时钟上电即工作，且无法关闭。SWCLK由调试器提供，频率可在调试界面进行选择。

为便于调试和出厂校正，高速RC时钟HSI和低速时钟LSI可以通过配置GPIO的第二功能通过芯片管脚输出。

# 5.1.2 功耗管理及休眠唤醒

# 5.1.2.1 休眠

MCLK可以通过配置进行门控，从而令包括CPU和所有外设在内的大部分数字电路处于休眠状态。门控时，PMU状态机依次关闭PLL，HSI/HSE,BGP等模拟模块，以降低功耗。

需要注意的是，系统的休眠模式仅仅关闭PLL，HSI，HSE等高速时钟，LSI时钟仍然存在。工作于LSI时钟的看门狗如果被使能，看门狗复位作为全局复位可以令系统回到初始状态重新开始工作。

向SYS\_CLK\_SLP寄存器写入0xDEAD可以令芯片准备进入休眠状态，之后立刻执行\_WFI0宏指令使得CPU停止取指。

在应用程序编写中请事先配置好唤醒条件。

\*建议参考官方休眠例程配置，休眠前需要先关闭所有数字模块的时钟，并关闭模拟ADC/OPA/CMP/DAC等模块。

# 5.1.2.2 唤醒

休眠后，外部IO事件、内部唤醒Timer均可以作为唤醒源

内部唤醒Timer为独立于UTimer模块的独立Timer，使用LSI时钟，不同于系统中的通用Timer工作于系统主时钟。唤醒Timer可以使用SYS\_RST\_CFG.WK\_INTV配置0.25s,0.5s,1s,2s,4s,8s,16s,32s共8档唤醒时间间隔，具体请参考5.2.14章节。

仅有P0[1:0]、P1[1:0]四个IO可以作为外部唤醒IO使用，可以配置独立的使能和极性。具体寄存器配置请参考8.2.13.5 WAKE\_POL和8.2.13.6 WAKE\_EN章节。需要注意的是，由于外部IO唤醒属于电平触发，如果外部IO在芯片休眠之前处于唤醒电平，会导致芯片休眠后立刻唤醒。

在应用程序编写中请尽量避免上电即进入休眠状态，如果使用内部唤醒Timer作为唤醒源，且唤醒后立即再次睡眠，会导致普通下载器无法进行连接调试，此时需要使用芯片供应商提供的离线下载器进行应用程序擦除改写。

# 5.1.2.3 外设时钟门控

外设时钟由系统高速时钟MCLK分频而来；当外设不需要使用时可以通过配置SYS\_CLK\_FEN寄存器门控关闭相应的外设时钟。对于每一个外设的工作时钟，均有一个时钟门控，共有8路可关闭的外设时钟分别供给不同的外设模块，详见

图5- 1时钟架构。门控时钟上电后默认是关闭的，使用相应外设模块之前需要由软件来开启。

I2C 使用 FCLK[0]Hall 模块使用 FCLK [1]Timer 模块使用 FCLK [2]MCPWM 模块使用 FCLK [3]UART0/UART1 分别使用 FCLK [4]/ FCLK [5]DSP 使用 FCLK [6]CAN 使用 FCLK [7]SPI 使用 FCLK [8]

# 5.1.2.4 外设时钟分频

部分外设有独立的时钟分频模块使得该模块可以工作在合适的时钟频率上。

其中 I2C 使用 SYS\_CLK\_DIV[0] 作为分频系数，UART0/1 共享 SYS\_CLK\_DIV[2] 作为分频系数。UART 的波特率在 UART 模块内部还有一个额外的分频器，详见

图 5- 1 时钟架构。

# 5.1.3 复位源

芯片的复位来源包括硬件复位与软件复位。

# 5.1.3.1 硬件复位

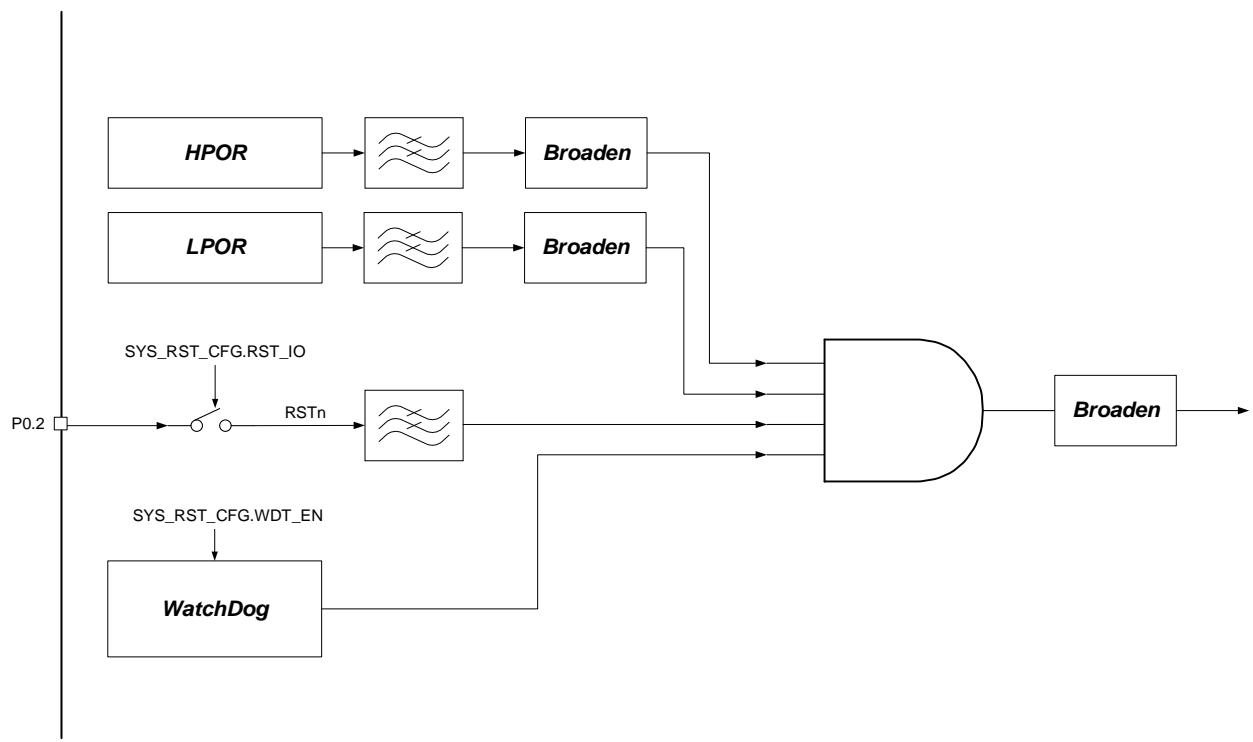
如表 5- 3 硬件复位源所示，系统包括 4 个硬件复位源，产生的复位均为芯片全局复位，复位产生后芯片程序计数器回到 0 地址，所有寄存器恢复到默认值。4 个硬件复位均为低电平有效。

表5-3硬件复位源

|  |  |  |
| --- | --- | --- |
| 名称 | 来源 | 说明 |
| LPORn | 内部 1.5V 电源管理 | 监控 1.5V 数字电源，低于 1.25V 时产生复位 |
| HPORn | 内部 3.3V 电源管理 | 监控 3.3V 电源，低于 3.0V 时产生复位 |
| RSTn | 外部按键 | 外部 RC 组成按键复位电路 |
| WDTn | 硬件看门狗 | 如果不进行软件喂狗则定时产生复位，复位间隔可配置 |

# 5.1.3.1.1 硬件复位架构

如下图所示，LPORn/HPORn 来自内部模拟电路，RSTn 来自外部按键经过滤波展宽预处理的复位信号进行与运算得到一个复位信号。P0.2 引脚持续时间小于 32us 的复位脉冲会被滤除，要求可靠复位宽度大于 200us。4 个复位信号复位等级和作用域一致，均为全局复位。

  
图5-2硬件复位架构

# 5.1.3.1.2硬件复位记录

SYS\_RST\_SRC寄存器用于保存硬件复位事件，当某个硬件复位发生后，SYS\_RST\_SRC对应位置位。SYS\_RST\_SRC寄存器本身无法被复位信号复位，只能通过向SYS\_CLR\_RST寄存器写入0xC4A0清空记录，复位记录可以方便地了解是否发生以及发生过何种复位。

# 5.1.3.2 软件复位

CPU的软复位操作可以使程序计数器(PC:ProgramCounter)回到0地址，但对所有外设中的寄存器没有影响。

在集成开发环境(IDE:IntegratedDevelopmentEnvironment)中的调试模式下，点击Reset与CPU软复位操作作用相同，仅仅使得PC回到0地址，对外设中的寄存器没有影响。但如果在bootloader中进行了外设模块的软复位，则会使得外设寄存器被复位为默认值。具体bootloader实现请咨询芯片供应商。

部分外设模块有模块级软复位，可以使用SYS\_SFT\_RST寄存器进行复位，写入对应的位，可以将模块状态机恢复到初始状态，同时将模块的寄存器恢复到默认值，详见5.2.22。

# 5.1.4 复位作用域

表5-4复位作用域

|  |  |
| --- | --- |
| 复位源 | 作用域 |
| LPORn | 内部 1.5V 电源管理，全局复位 |
| HPOR | 内部 3.3V 电源管理，全局复位，除极少数寄存器 |
| RSTn | 外部按键，全局复位，除极少数寄存器 |
| WDTn | 硬件看门狗，全局复位，除极少数寄存器 |

|  |  |
| --- | --- |
| SYS\_SFT\_RST.SPI\_SFT\_RST | SPI 模块 |
| SYS\_SFT\_RST.CAN\_SFT\_RST | CAN 模块 |
| SYS\_SFT\_RST.DSP\_SFT\_RST | DSP 模块 |
| SYS\_SFT\_RST.UART1\_SFT\_RST | UART1 模块 |
| SYS\_SFT\_RST.UART0\_SFT\_RST | UART0 模块 |
| SYS\_SFT\_RST.MCPWM\_SFT\_RST | MCPWM 模块 |
| SYS\_SFT\_RST.UTIMER\_SFT\_RST | UTIMER 模块 |
| SYS\_SFT\_RST.HALL\_SFT\_RST | HALL 模块 |
| SYS\_SFT\_RST.I2C\_SFT\_RST | I2C 模块 |
| NVIC\_SystemReset(); | CPU 软复位，仅复位 CPU 内核，将 PC 重置为 0，所有外设寄存器值仍然维持。 |

其中用于控制P0[2]作为GPIO使用还是作为外部复位脚使用的SYS\_RST\_CFG.RST\_IO，用于控制看门狗使能的SYS\_RST\_CFG.WDT\_EN，以及复位记录寄存器SYS\_RST\_SRC只受LPOR复位。

全局复位会复位全芯片寄存器，包括CPU内核寄存器以及所有外设寄存器，除上述极少数寄存器。

由于CPU软复位仅仅复位CPU内核，而不复位外设寄存器，因此建议重新烧录程序后使用掉电重新上电或外部复位的方式重置外设寄存器。

Flash存储内容，SRAM存储内容不受复位影响。

# 5.2 寄存器

# 5.2.1 地址分配

系统模块寄存器基地址为0x4000\_0000，寄存器列表如下：

表5-5系统控制寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
|  | 0x00~0x08 | 保留 |
|  | 0x10~0x14 | 保留 |
| SYS\_AFE\_CMP | 0x18 | 模拟前端信息寄存器 |
|  | 0x1C | 保留 |
| SYS\_AFE\_REG0 | 0x20 | 模拟配置寄存器 0 |
| SYS\_AFE\_REG1 | 0x24 | 模拟配置寄存器 1 |
| SYS\_AFE\_REG2 | 0x28 | 模拟配置寄存器 2 |
| SYS\_AFE\_REG3 | 0x2C | 模拟配置寄存器 3 |
| SYS\_AFE\_REG4 | 0x30 | 模拟配置寄存器 4 |
| SYS\_AFE\_REG5 | 0x34 | 模拟配置寄存器 5 |
| SYS\_AFE\_REG6 | 0x38 | 模拟配置寄存器 6 |
| SYS\_AFE\_REG7 | 0x3C | 模拟配置寄存器 7 |
|  | 0x54~0x78 | 保留 |
| SYS\_AFE\_DAC | 0x7C | DAC 数字量寄存器 |

|  |  |  |
| --- | --- | --- |
| SYS\_CLK\_CFG | 0x80 | 时钟控制寄存器 |
| SYS\_RST\_CFG | 0x84 | 复位控制寄存器 |
| SYS\_RST\_SRC | 0x88 | 复位源记录寄存器 |
| SYS\_CLR\_RST | 0x8C | 复位源记录清除寄存器 |
| SYS\_CLK\_DIV0 | 0x90 | 外设时钟分频寄存器 0 |
|  | 0x94 | 保留 |
| SYS\_CLK\_DIV2 | 0x98 | 外设时钟分频寄存器 2 |
| SYS\_CLK\_FEN | 0x9C | 外设时钟门控寄存器 |
| SYS\_CLK\_SLP | 0xA0 | 休眠寄存器 |
|  | 0xA4 | 保留 |
| SYS\_TRIM | 0xA8 | 校正模式寄存器 |
| SYS\_SFT\_RST | 0xAC | 软复位寄存器 |
| SYS\_WR\_PROTECT | 0xB0 | 写保护寄存器 |
| SYS\_DAC\_AMC | 0xB4 | DAC 增益校正寄存器 |
| SYS\_DAC\_DC | 0xB8 | DAC 直流偏置寄存器 |

# 5.2.2 SYS\_AFE\_CMP模拟前端信息寄存器

地址：0x4000\_0018

复位值：0x0

表5-6模拟前端信息寄存器SYS\_AFE\_CMP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMP1 | CMP0 | PWR\_WEAK |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RO | RO | RO |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 复位值 | 偏移 | 位置 | 权限 | 位名称 | 说明 |
| SYS\_AFE\_CMP | 0x0 | 0x18 | [31:16] | NA | 未使用 | |
| [15] | R | CMP1\_RESULT | CMP1 输出结果寄存器 |
| [14] | R | CMP0\_RESULT | CMP0 输出结果寄存器 |
| [13] | R | PWR\_WEAK | 供电低于掉电监测阈值 |
| [12:0] | NA | 未使用 | |

SYS\_AFE\_CMP中的CMP0/CMP1为比较器的原始输出，没有经过滤波处理。CMP0/CMP1也可以通过配置GPIO的第二功能（AF1）输出，具体输出引脚请查看器件DATASHEET。

PWR\_WEAK标志在当供电电压低于掉电监测电路设定的阈值时置位，同时将产生CPU掉电中断（中断号17）。在供电恢复，高于阈值后清零。此标志为只读位，无法软件清除。

掉电监测的使用请参考4.2POWER电源管理系统。

# 5.2.3 模拟寄存器概述

模拟寄存器 是开放给用户的寄存器，其中保留寄存器(Res)必须全部配置为0（芯片上电后会被复位为0)。其他寄存器根据应用场合需要进行配置。下面是各个模拟寄存器的详细说明。

下面是各个模拟寄存器的详细说明。

# 5.2.4 SYS\_AFE\_REG0模拟配置寄存器0

地址：0x4000\_0020

复位值：0x0

表5-7模拟配置寄存器0SYS\_AFE\_REG0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Resv. | Resv. | Resv. | Resv. | REF\_OPA3 | REF\_OPA2 | REF\_OPA1 | REF\_OPA0 |  |  |  |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:14] | Reserved | 保留位，需全部为'0' |
| [13:12] | Reserved | 保留位，需全部为'0' |
| [11:10] | Reserved | 保留位，需全部为'0' |
| [9:8] | Reserved | 保留位，需全部为'0' |
| [7:6] | REF(OPA3) | 运放3反馈电阻 00:200k:10.4k 01:190k:20.4k 10:180k:30.4k 11:170k:40.4k |
| [5:4] | RES(OPA2) | 运放2反馈电阻 00:200k:10.4k 01:190k:20.4k 10:180k:30.4k 11:170k:40.4k |
| [3:2] | RES(OPA1) | 运放1反馈电阻 00:200k:10.4k 01:190k:20.4k 10:180k:30.4k 11:170k:40.4k |
| [1:0] | RES(OPA0) | 运放0反馈电阻 00:200k:10.4k |

|  |  |  |
| --- | --- | --- |
|  |  | 01:190k:20.4k 10:180k:30.4k 11:170k:40.4k |

运放反馈电阻的比例精度为 ，具体精度数据可参考应用笔记 运放反馈电阻值精度》

# 5.2.5 SYS\_AFE\_REG1模拟配置寄存器1

地址：0x4000\_0024

复位值：0x0

表5-8模拟配置寄存器1SYS\_AFE\_REG1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Reserved | DAC\_GAIN | REFVDD5 | REF2VDD | GAIN\_REF | IT\_CMP | CMP\_FT |
| RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] | 未使用 | |
| [15] | Reserved | 保留位，需全部为'0' |
| [14] | Reserved | 保留位，需全部为'0' |
| [13] | Reserved | 保留位，需全部为'0' |
| [12] | Reserved | 保留位，需全部为'0' |
| [11:8] | Reserved | 保留位，需全部为'0' |
| [7:6] | DAC\_GAIN | DAC 输出档位配置 00: DAC 输出档位，满量程为 3V 01: DAC 输出档位，满量程为 1.2V 10: DAC 输出档位，满量程为 4.85V（注意，此时供电电源需为 5V，否则 DAC 输出异常） 11: 错误配置 |
| [5] | REFVDD5 | 电源作为 ADC REF 且电源为 5V 时，增加运放输出的信号范围 0: 运放最大输出范围为 3.3V 1: 增加运放最大输出范围到 4.8V，需在 REF2VDD 配置为'1'时才可设置为'1' |
| [4] | REF2VDD | 使用外部输入电源作为 ADC REF 1: 使用外部输入电源作为 ADC REF，此时 GAIN\_REF 配置无效； 0: 使用默认内部 REF 作为 ADC 基准 |
| [3] | GAIN\_REF | ADC 基准电压调节，采用默认配置 0: ×2 1: ×1 |
| [2:1] | IT\_CMP | IT\_CMP&lt;1&gt;：比较器 1 的比较速度选择 0: 150ns 1: 600ns |

|  |  |  |
| --- | --- | --- |
|  |  | IT\_CMP&lt;0&gt;:比较器0的比较速度选择 0:150ns 1:600ns |
| [0] | CMP\_FT | 使能比较器快速比较 1:在IT\_CMP&lt;1:0&gt;都为默认'00'的时候，比较器比较速度小于30ns 0:不使能，比较速度维持IT\_CMP设置里的参数 |

# 5.2.6 SYS\_AFE\_REG2模拟配置寄存器2

地址：0x4000\_0028

复位值：0x0

表5-9模拟配置寄存器2SYS\_AFE\_REG2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved |  | Reserved |  | Reserved |  |  | Reserved |  |  | OPAOUT\_EN |  |  |  |  |  |
| RW |  | RW |  | RW |  |  | RW |  |  | RW |  |  |  |  |  |
| 0 |  | 0 |  | 0 |  |  | 0 |  |  | 0 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:14] | Reserved | 保留位，需全部为'0' |
| [13] | Reserved | 保留位，需全部为'0' |
| [12] | Reserved | 保留位，需全部为'0' |
| [11:10] | Reserved | 保留位，需全部为'0' |
| [9:8] | Reserved | 保留位，需全部为'0' |
| [7] | Reserved | 保留位，需全部为'0' |
| [6] | Reserved | 保留位，需全部为'0' |
| [5:4] | Reserved | 保留位，需全部为'0' |
| [3] | Reserved | 保留位，需全部为'0' |
| [2:0] | OPAOUT\_EN | 使能OPAx输出信号送至IO口p2\_7 000:不输出; 001:输出OPA0信号到IO口; 010:输出OPA1信号到IO口; 011:输出OPA2信号到IO口; 100:输出OPA3信号到IO口; 101~111:禁止 |

# 5.2.7 SYS\_AFE\_REG3模拟配置寄存器3

地址：0x4000\_002C

复位值：0x0

表5-10模拟配置寄存器3SYS\_AFE\_REG3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | CMP1\_SELP | DACOUT\_EN | CMP0\_SELP | CMP\_HYS | Res. | CMP1\_SELN | CMP0\_SELN | Res. | LDOOUT\_EN |  |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [51:16] |  | 未使用 |
| [15] | Reserved | 保留位，需全部为'0' |
| [14:12] | CMP1\_SELF | 比较器1信号正端选择 000:连CMP1\_IPO 001:连OPA3\_IP 010:连OPA2\_OUT 011:连OPA3\_OUT 100:连CMP1\_IP1 101:连CMP1\_IP2 110:连CMP1\_IP3 111:连AVSS 说明:上述除AVSS/OPA2\_OUT/OPA3\_OUT 外都为管脚名称,请参看 datasheet 里管脚定义章节 |
| [11] | DACOUT\_EN | DAC 输出到IO 使能 0:不使能 1:使能输出到IO P0[0] |
| [10:8] | CMP0\_SELF | 比较器0信号正端选择 000:连CMP0\_IPO 001:连OPA0\_IP 010:连OPA0\_OUT 011:连OPA1\_OUT 100:连CMP0\_IP1 101:连CMP0\_IP2 110:连CMP0\_IP3 111:连CMP0\_IP4 说明:上述除OPA0\_OUT/OPA1\_OUT 外都为管脚名称,请参看 datasheet 里管脚定义章节 |
| [7] | CMP\_HYS | 比较器回差选择,采用默认配置 0:20mv 1:0mv |
| [6] | Reserved | 保留位,需全部为'0' |
| [5:4] | CMP1\_SELN | 比较器1信号负端选择 00:连CMP1\_IN 01:连REF 10:连DAC 输出 11:连BEMF1\_MID |

|  |  |  |
| --- | --- | --- |
|  |  | 说明:上述CMP1\_IN为管脚名称,请参看datasheet里管脚定义章节;REF为芯片内部1.2V BANDGAP基准源;DAC输出即为芯片内部DAC模块输出模拟信号;BEMF1\_MID为CMP1\_IP1, CMP1\_IP2, CMP1\_IP3信号经电阻星形连接后得到的平均值 |
| [3:2] | CMP0\_SELN | 比较器0信号负端选择 00:连CMP0\_IN 01:连REF 10:连DAC输出 11:连BEMF0\_MID 说明: 上述CMP0\_IN为管脚名称,请参看datasheet里管脚定义章节;REF为芯片内部1.2V BANDGAP基准源;DAC输出即为芯片内部DAC模块输出模拟信号;BEMF0\_MID为CMP0\_IP1, CMP0\_IP2, CMP0\_IP3信号经电阻星形连接后得到的平均值 |
| [1] | Reserved | 保留位,需全部为'0' |
| [0] | LDOOUT\_EN | LDO输出到IO使能 0:不输出 1:使能输出到IO P2.7 |

# 5.2.8 SYS\_AFE\_REG4模拟配置寄存器4

地址：0x4000\_0030

复位值：0x0

表5-11模拟配置寄存器4SYS\_AFE\_REG4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. |  | Res. |  |  |  |  |  |  | Res. |  |  |  |  |  |  |
| RW |  | RW |  |  |  |  |  |  | RW |  |  |  |  |  |  |
| 0 |  | 0 |  |  |  |  |  |  | 0 |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | Reserved | 保留位，需全部为'0' |
| [14] |  | 未使用 |
| [13] | Reserved | 保留位，需全部为'0' |
| [12:6] |  | 未使用 |
| [5] | Reserved | 保留位，需全部为'0' |
| [4:0] |  | 未使用 |

# 5.2.9 SYS\_AFE\_REG5模拟配置寄存器5

地址：0x4000\_0034

复位值：0x0

表5-12模拟配置寄存器5SYS\_AFE\_REG5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLL PD | XTAL PD | TMPPDN | DAC12BPDN | Res. | RCHPD | Res. | BGPPD | CMP1PDN | CMP0PDN | OPA3PDN | OPA2PDN | OPA1PDN | OPA0PDN | Res. | ADCPDN |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | PLLPDN | PLL关闭控制 0:关闭PLL（默认） 1:开启PLL |
| [14] | XTALPDN | 晶体起振电路开启使能 0:关闭（默认） 1:开启 |
| [13] | TMPPDN | 温度传感器开启使能 0:关闭 1:开启 |
| [12] | DAC12BPDN | 12BIT DAC 开启使能 0:关闭 1:开启 |
| [11] | Reserved | 保留位，需全部为'0' |
| [10] | RCHPD | RCH 时钟关闭使能 0:开启 1:关闭 |
| [9] | Reserved | 保留位，需全部为'0' |
| [8] | BGPPD | BGP 开启使能 0:开启 1:关闭 |
| [7] | CMP1PDN | CMP1 开启使能 0:关闭 1:开启 |
| [6] | CMP0PDN | CMP0 开启使能 0:关闭 1:开启 |
| [5] | OPA3PDN | OPA3 开启使能 0:关闭 1:开启 |
| [4] | OPA2PDN | OPA2 开启使能 |

|  |  |  |
| --- | --- | --- |
|  |  | 0:关闭 1:开启 |
| [3] | OPA1PDN | OPA1 开启使能 0:关闭 1:开启 |
| [2] | OPA0PDN | OPA0 开启使能 0:关闭 1:开启 |
| [1] | Reserved | 保留位，需全部为'0' |
| [0] | ADCPDN | ADC 开启使能 0:关闭 1:开启 |

如果SYS\_CLK\_CFG选择PLL时钟，则PLLPDN是被硬件控制的，软件配置PLLPDN关闭PLL无效。关闭PLL需要PLLPDN ，且SYS\_CLK\_CFG不选择PLL作为芯片主时钟，这两个条件都须满足。

同理，如果SYS\_CLK\_CFG选择了HRC时钟，则RCHPD是被硬件控制的，软件配置RCHPD关闭RCH无效。关闭PLL需要 ，且芯片进入休眠。

如果芯片主时钟为PLL时钟，且HRC为PLL参考时钟，则RCH也是被硬件控制的。

由于RCH和PLL依赖BGP（bandgap），所以BGPPD也是硬件控制的，在芯片使用了RCH或PLL时，软件配置BGPPD关闭BGP无效。关闭BGP需要先顺序关闭PLL和RCH，且芯片进入休眠。

# 5.2.10 SYS\_AFE\_REG6模拟配置寄存器6

地址：0x4000\_0038

复位值：0x0

表5-13模拟配置寄存器6SYS\_AFE\_REG6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLLSR\_SEL | Reserved | | | PVDSEL | | Reserved | | | Reserved | | VSR PDT | | PD PDT | |  |
| RW | RW | | | RW | | RW | | | RW | | RW | | RW | |  |
| 0 | 0 | | | 0 | | 0 | | | 0 | | 0 | | 0 | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | PLLSR\_SEL | PLL 时钟源选择 0: 使用 RCH 作为输入时钟源; 1: 使用 XTAL OSC 作为输入时钟源 |
| [14] | Reserved | 保留位，需全部为'0' |
| [13] | Reserved | 保留位，需全部为'0' |

|  |  |  |
| --- | --- | --- |
| [12] | Reserved | 保留位，需全部为'0' |
| [11] | Reserved | 保留位，需全部为'0' |
| [10] | Reserved | 保留位，需全部为'0' |
| [9:8] | PVDSEL | 电源掉电监测阈值选择 当 VSR\_PDT=0 时， 00: 4.5V 01: 4.2V 10: 3.9V 11: 3.6V 当 VSR\_PDT=1 时， 4 个档位与 DAC 设定值有关，分别为 00: 4.5/1.35*DAC V 01: 4.2/1.35*DAC V 10: 3.9/1.35*DAC V 11: 3.6/1.35*DAC V |
| [7] | Reserved | 保留位，需全部为'0' |
| [6:5] | Reserved | 保留位，需全部为'0' |
| [4:3] | Reserved | 保留位，需全部为'0' |
| [2] | Reserved | 保留位，需全部为'0' |
| [1] | VSR\_PDT | 掉电检测基准源选择，其中低功耗基准源为 1.35V 左右，DAC 输出则可以通过软件配置；低电压检测的阈值有 4 档可选，通过 PVDSEL 选择 0: 选择低功耗基准源； 1: 选择 DAC 输出作为掉电检测基准源 |
| [0] | PD\_PDT | 关闭掉电检测电路 0: 开启 1: 关闭 |

# 5.2.11 SYS\_AFE\_REG7模拟配置寄存器7

地址：0x4000\_003C

复位值：0x0

表5-14模拟配置寄存器7SYS\_AFE\_REG7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | | | ADCLKSEL | | | Reserved | | |  |
| RW | | | | | | | | | RW | | | RW | | |  |
| 0 | | | | | | | | | 0 | | | 0 | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | Reserved | 保留位，需全部为'0' |
| [14] | Reserved | 保留位，需全部为'0' |
| [13:8] | Reserved | 保留位，需全部为'0' |

|  |  |  |
| --- | --- | --- |
| [7:6] | Reserved | 保留位，需全部为'0' |
| [5:4] | ADCLKSEL | ADC 时钟频率选择 00: 48MHz 01: 禁止 10: 12MHz 11: 24MHz |
| [3:0] | Reserved | 保留位，需全部为'0' |

# 5.2.12 SYS\_AFE\_DAC DAC 数字量寄存器

地址：0x4000\_007C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |
| [11:0] | DAC\_IN | DAC 待转换的数字量输入 |

# 5.2.13 SYS\_CLK\_CFG 时钟控制寄存器

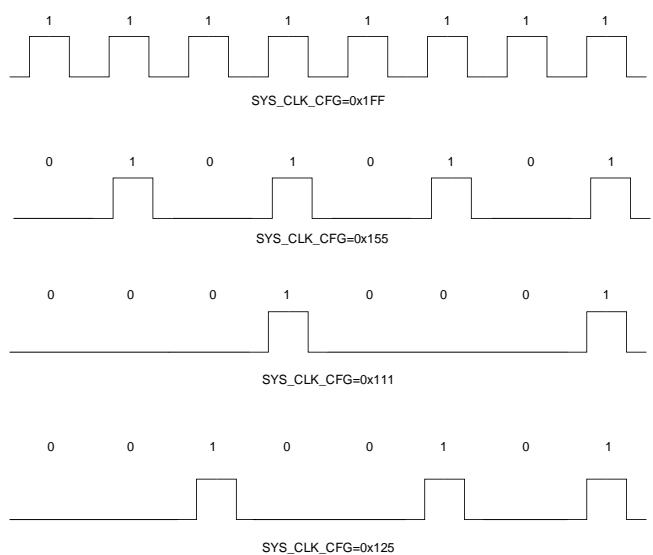
地址：0x4000\_0080

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  | CLK\_SEL |  | CLK\_DIV |  |  |  |  |  |  |  |
| RW |  | RW |  |  |  |  |  |  |
| 0 |  | 0 |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:9] |  | 未使用 |
| [8] | CLK\_SEL | CLK\_HS/PLL 选择，0: CLK\_HS, 1:PLL。默认选择 CLK\_HS，CLK\_HS 根据是否使用外部晶体时钟，可以是 HSI 或 HSE。 PLL 在上电后默认关闭，需要软件来开启 |
| [7:0] | CLK\_DIV | PLL 输出分频控制，选择 8 个时钟周期中，哪些周期输出时钟，例如 |

|  |  |  |
| --- | --- | --- |
|  |  | 8'b00000001，表示1/8分频，8'b00010001表示1/4分频，8'b00100101表示1/3分频，但不均匀。 |

  
图5-3 SYS\_CLK\_CFG不同配置时的系统主时钟分频波形

当使用4MHzHSI时钟作为系统主时钟时，SYS\_CLK\_CFG[7:0]的分频系数无效，最终输出的系统时钟频率即为4MHz。

# 5.2.14 SYS\_RST\_CFG复位控制寄存器

地址：0x4000\_0084

复位值：0x0

表5-17复位控制寄存器SYS\_RST\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:6] |  | 未使用 |
| [5] | RST\_IO | RSTn/P0[2]复用选择，0: RSTn, 1:P0[2] 当用作外部复位引脚时，片内上拉默认开启，且不可控。 当用作 GPIO P0[2]时，片内上拉默认关闭，且不可控。 |
| [4:2] | WK\_INTV | 休眠唤醒间隔设置 000: 0.25S 100: 4S 001: 0.5S 101: 8S 010: 1S 110: 16S 011: 2S 111: 32S |
| [1] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [0] | WDT\_EN | 看门狗使能，高有效。 |

# 5.2.15 SYS\_RST\_SRC 复位源记录寄存器

地址：0x4000\_0088

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 未使用 |
| [3] | WDT\_RST\_RCD | 看门狗复位发生标志，高有效 |
| [2] | KEY\_RST\_RCD | 按键复位发生标志，高有效 |
| [1] | HPOR\_RST\_RCD | HPOR 复位发生标志，高有效 |
| [0] | LPOR\_RST\_RCD | LPOR 复位发生标志，高有效 |

# 5.2.16 SYS\_CLR\_RST 复位源记录清除寄存器

地址：0x4000\_008C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSW | | | | | | | | | | | | | | |  |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PSW | 写入 0xC4A0，清除复位标志记录 请注意由于复位记录工作于低速时钟域，清除执行完成需要一定时间，不应清除后立即读记录状态 |

# 5.2.17 SYS\_CLK\_DIV0 外设时钟分频寄存器 0

地址：0x4000\_0090

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [51:16] |  | 未使用 |
| [15:0] | DIV0 | I2C 工作时钟=MCLK/(CLK\_DIV0+1)。其中 MCLK 由 SYS\_CLK\_CFG 分频系数决定 |

# 5.2.18 SYS\_CLK\_DIV2外设时钟分频寄存器2

地址：0x4000\_0098

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV2 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DIV2 | UART 工作时钟=MCLK/(CLK\_DIV2+1) UART0/UART1 共享此分频配置，波特率根据 UART 波特率寄存器进一步分频，其中 MCLK 由 SYS\_CLK\_CFG 分频系数决定 |

# 5.2.19 SYS\_CLK\_FEN外设时钟门控寄存器

地址：0x4000\_009C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | SPCLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | CAN\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | DSP\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | UART1\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | UART2\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | MCPWM\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | UTIMER\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | HALL\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | I2C\_CLK\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:9] |  | 未使用 |
| [8] | SPI\_CKL\_EN | SPI 时钟门控，1:使能时钟；0:禁用时钟 |
| [7] | CAN\_CLK\_EN | CAN 时钟门控，1:使能时钟；0:禁用时钟 |
| [6] | DSP\_CLK\_EN | DSP 时钟门控，1:使能时钟；0:禁用时钟 |
| [5] | UART1\_CLK\_EN | UART1 时钟门控，1:使能时钟；0:禁用时钟 |
| [4] | UART0\_CLK\_EN | UART0 时钟门控，1:使能时钟；0:禁用时钟 |
| [3] | MCPWM\_CLK\_EN | MCPWM 时钟门控，1:使能时钟；0:禁用时钟 |
| [2] | UTIMER\_CLK\_EN | UTIMER 时钟门控，1:使能时钟；0:禁用时钟 |
| [1] | HALL\_CLK\_EN | HALL 时钟门控，1:使能时钟；0:禁用时钟 |
| [0] | I2C\_CLK\_EN | I2C 时钟门控，1:使能时钟；0:禁用时钟 |

# 5.2.20 SYS\_CLK\_SLP 休眠寄存器

地址：0x4000\_00A0

复位值：0x0

表5-23休眠寄存器SYS\_CLK\_SLP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSW | | | | | | | | | | | | | | |  |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PSW | 写入密码 0xDEAD，系统关闭高速时钟，进入休眠状态 |

# 5.2.21 SYS\_TRIM 校正模式寄存器

地址：0x4000\_00A8

复位值：0x0

表5-24校正模式寄存器SYS\_TRIM

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | |  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:1] |  | 未使用 |
| [0] | TRIM | 芯片复位后，进入TRIM模式 TRIM结束后，通过软复位退出TRIM模式 |

# 5.2.22 SYS\_SFT\_RST 软复位寄存器

地址：0x4000\_00AC

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | I2C\_SFT\_RST |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:9] |  | 未使用 |
| [8] | SPI\_SFT\_RST | SPI 软复位，写 1 复位，再写 0 释放 |
| [7] | CAN\_SFT\_RST | CAN 软复位，写 1 复位，再写 0 释放 |
| [6] | DSP\_SFT\_RST | DSP 软复位，写 1 复位，再写 0 释放 |
| [5] | UART1\_SFT\_RST | UART1 软复位，写 1 复位，再写 0 释放 |
| [4] | UART0\_SFT\_RST | UART0 软复位，写 1 复位，再写 0 释放 |
| [3] | MCPWM\_SFT\_RST | MCPWM 软复位，写 1 复位，再写 0 释放 |
| [2] | UTIMER\_SFT\_RST | UTIMER 软复位，写 1 复位，再写 0 释放 |
| [1] | HALL\_SFT\_RST | HALL 软复位，写 1 复位，再写 0 释放 |
| [0] | I2C\_SFT\_RST | I2C 软复位，写 1 复位，再写 0 释放 |

注意，模块软复位在 SYS\_SFT\_RST 对应位写入 1 后会保持在复位状态，需要再次写入 0 才能解除复位状态。

# 5.2.23 SYS\_WR\_PROTECT 写保护寄存器

地址：0x4000\_00B0

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |
| --- |
| PSW |
| WO |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PSW | 除 SYS\_AFE\_REG3、SYS\_AFE\_DAC、SYS\_AFE\_DAC\_AMC、SYS\_AFE\_DAC\_DC 外，其他系统寄存器受写保护；写入前需先写入密码解除写保护写入 0x7A83，使能系统寄存器写操作写入 0xCAFE，使能看门狗 WDT\_CLR 寄存器写操作写入其它值，禁止寄存器写操作 |

# 5.2.24 SYS\_AFE\_DAC\_AMC DAC 增益校正寄存器

地址：0x4000\_00B4

复位值：0x0

表5-27DAC增益校正寄存器SYS\_AFE\_DAC\_AMC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |
| [9:0] | DAC\_AMC | DAC 增益校正值，10bit 无符号定点数，B[9]为整数部分，B[8:0]为小数部分 |

# 5.2.25 SYS\_AFE\_DAC\_DC DAC 直流偏置寄存器

地址：0x4000\_00B8

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  | DAC\_DC | | | | | | | |  |  |
| RW | |
| 0 | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:8] |  | 未使用 |
| [7:0] | DAC\_DC | DAC 直流偏置，8bit 有符号数，B[7]为符号位 |

DAC 增益校正，记模拟输出的 12bitDAC 数值为 DAC\_raw，经过校正后的 12bit DAC 数值为 DAC\_cali

DAC\_cali = DAC\_raw \* DAC\_AMC - DAC\_DC;

其中 DAC\_AMC 为 DAC 增益校正系数，为 10bit 定点无符号数，B[9]为整数，B[8:0]为小数，大小为 1 左右，例如 DAC\_AMC = 10’b10\_0001\_0000 = 1+1/32，或

DAC\_AMC = 10’b01\_1110\_1100 = 1- 5/128

DAC\_DC 位 DAC 直流偏置，为 8bit 有符号整数。

增益校正计算结果进行截断保留，最终 DAC\_cali 仍为 12bit 整数。

且增益校正加直流偏置校正后的数值会进行饱和处理，最大为 0xff，最小为 0x000。

需要注意的是，DAC 有三个输出档位，系统上电后，加载默认档位的 DAC 校准值，若切换到其它档位，请使用原厂提供的库函数。

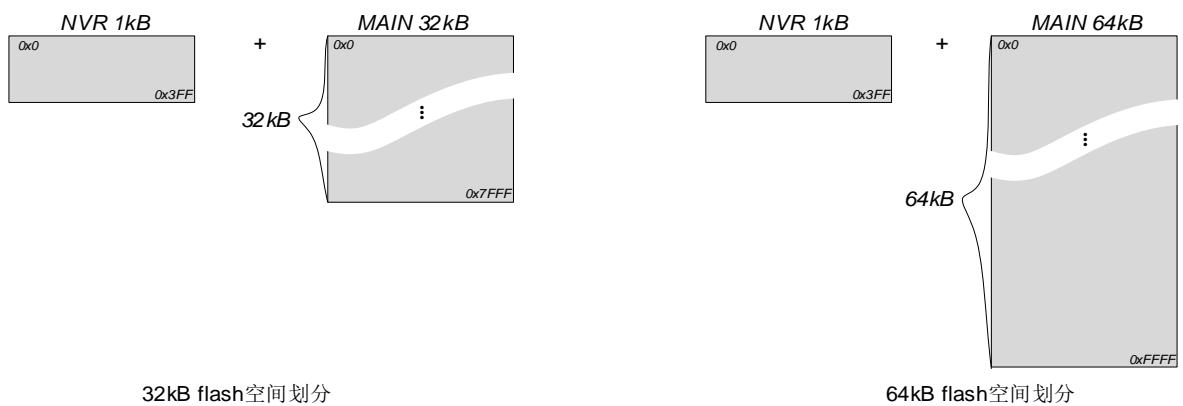
# 6 FLASH

# 6.1 概述

FLASH 存储体包含两个部分：NVR 和 MAIN。NVR 大小为 1KB，MAIN 为 32KB 或 64KB（不同型号）。

主闪存存储区（MAIN），包括应用程序和用户数据区

信息存储区（info区/NVR），预留给用户使用，大小为1KB，供客户存储私有信息

  
图6-1 FLASH存储体空间划分框图

* 可反复擦除写入不低于2万次- 室温数据保持长达100年- 单字节编程时间最长7.5us，Sector擦除时间最长5ms。- Sector大小512字节，可按Sector擦除写入，支持运行时编程，擦写一个Sector的同时读取访问另一个Sector- Flash数据防窃取（最后一个word须写入非0xFFFFFFFF的任意值）

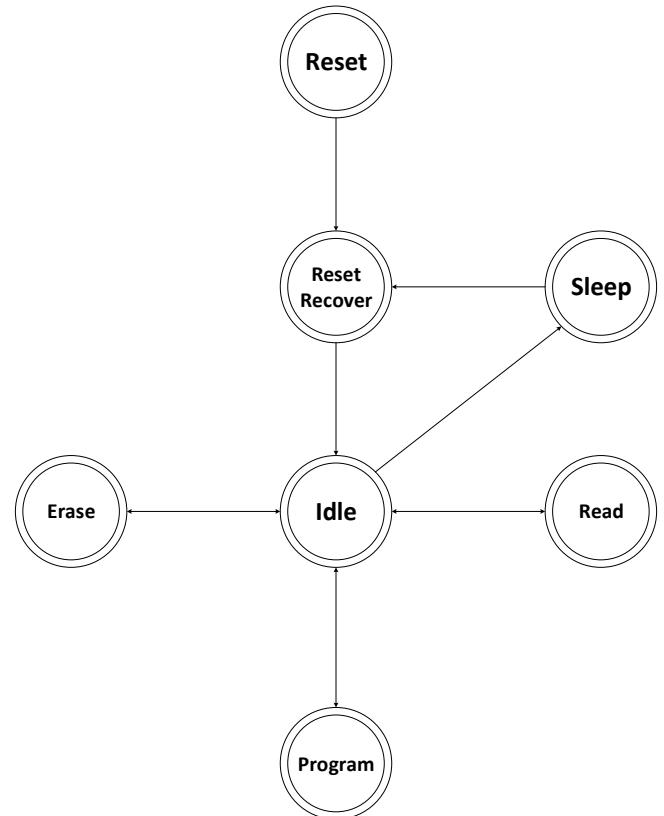
# 6.2 功能特点

FLASH控制器模块，主要实现对FLASH存储体的相关操作。包括：

* FLASH读取数据的操作，包括对NVR部分的读取和对MAIN部分的读取。
* FLASH写入数据的操作，包括对NVR部分的写入和对MAIN部分的写入。
* FLASH擦除操作，包括CHIP擦除和SECTOR擦除。NVR部分仅支持SECTOR擦除，MAIN部分支持CHIP擦除和SECTOR擦除。
* FLASH深度休眠的操作，以降低芯片的休眠功耗。
* FLASH 存储体内容的加密操作。- FLASH 的读取加速操作，以提升芯片整体运行效率。- FLASH 控制寄存器的访问。

# 6.2.1 功能描述

控制模块，实现了对 FLASH 存储体的复位/读出/写入/擦除/休眠等操作。如下为控制状态转换图：

  
图 6-2 FLASH 控制状态转换图

# 6.2.1.1 复位操作

系统完成复位后，FLASH 需要一段时间恢复。其目的是保证 FLASH 存储体内部的电路稳定。待内部稳定后，才可对 FLASH 执行操作。此恢复操作由硬件自动实现，无需软件干预。

# 6.2.1.2 休眠操作

FLASH 的休眠操作分成两个部分：Standby 和 Deep Sleep。当系统不执行对 FLASH 的操作时，FLASH 可自动进入 StandBy 状态（若开启预取，此功能失效）。当系统执行 Deep Sleep 操作时，将触发 FLASH 也进入 Deep Sleep，实现降低功耗的目的。FLASH 进入 Deep Sleep 的操作，硬件自动完成，无需软件介入。

当外界唤醒系统，同时将唤醒 FLASH。经过一段时间恢复后，FLASH 可执行正常操作。此唤醒恢复操作，硬件自动完成，无需软件介入。

# 6.2.1.3 FLASH读取操作

读操作为FLASH的基本操作。系统可通过两条路径访问FLASH内部的数据。

CPU通过AHB总线，直接对FLASH执行取指取数操作。取指宽度为32bit，且只能访问MAIN空间的数据。为了加快CPU的取指取数据的速度，硬件提供了加速的功能。

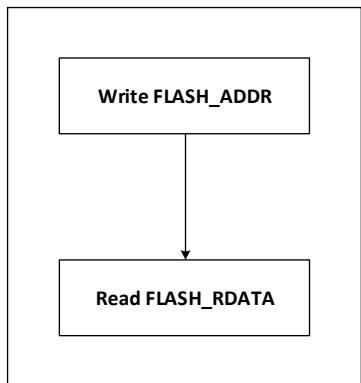
CPU通过AHB总线，访问控制器的寄存器，间接实现读取FLASH内部数据的操作。可以访问MAIN和NVR空间的数据；若执行连续读取操作，硬件可自动完成地址累加，无需每次都更新地址寄存器的值。

FLASH\_CFGREGION位指示当前访问的是哪个空间。具体表格如下：

表6-1FLASH访问空间分配表

|  |  |
| --- | --- |
| NVR (FLASH\_CFGREGION) | 访问区域 |
| 0 | MAIN 区域 |
| 1 | NVR 区域 |

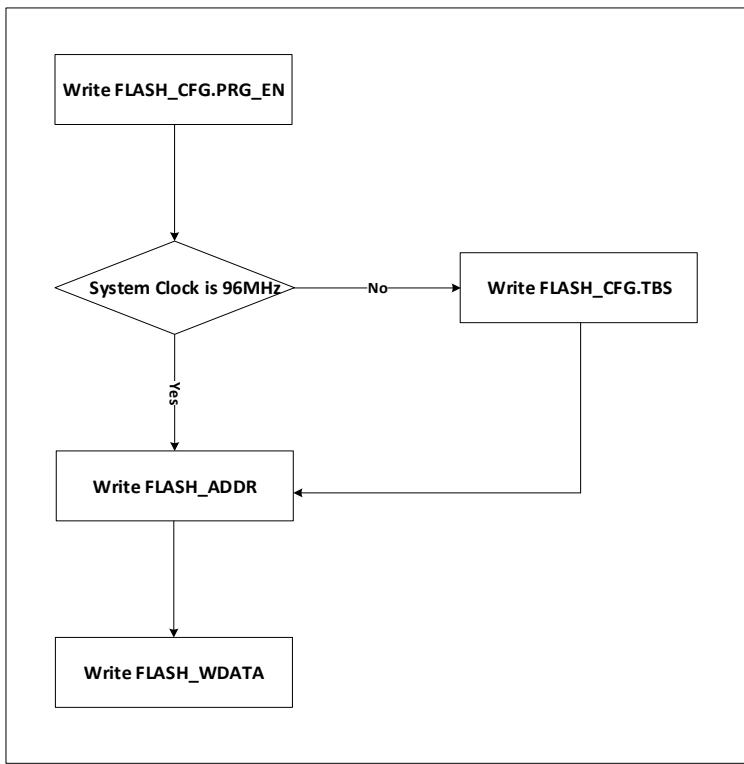
访问控制器的寄存器，实现间接读取FLASH内部数据的操作的执行流程如下：

  
图6-3FLASH间接读取操作流程图

# 6.2.1.4 FLASH编程操作

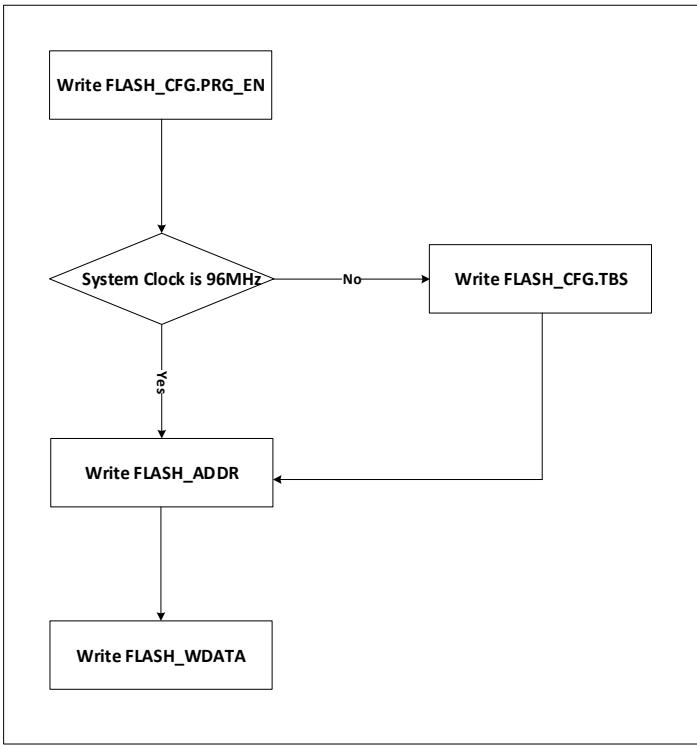
执行对FLASH存储体的编程操作。一般而言，我们先执行擦除操作，然后才能执行数据编程操作。同时，只能通过访问FLASH控制器的寄存器，实现编程操作。具体流程为：

控制寄存器CFG，开启编程使能 地址寄存器ADDR，写入编程地址 写数据寄存器WDATA，写入编程数据访问本控制器的寄存器，实现FLASH编程操作的执行流程如下：

  
图6-4 FLASH模块编程操作流程图

系统工作频率的判断，需要参考SYS\_CLK\_CFG的配置。FLASH写入/擦除操作的绝对时间是固定的，FLASH控制器需要保存这些绝对时间对应的计数值。FLASH\_CFG.TBS默认值是96MHz时钟频率下的计数值；当芯片工作在其他频率下时，需要配置FLASH\_CFG.TBS的值，以实现48MHz/24MHz和12MHz的计数值（其它频率暂不支持）。最终保证计数值的的值×时钟频率等于恒定的时间。不同频率下对应的的FLASH\_CFG.TBS值请参考6.3.2。切记，FLASH\_CFG.TBS仅能配置寄存器说明中提供的几组值，不能被写入其它值，否则可能导致FLASH编程/擦除失败。建议对FLASH\_CFG的操作，执行先读回，然后按照或/与的方法操作。另外，在执行FLASH的编程/擦除操作时，CPU将暂停工作直至FLASH的编程/擦除操作完毕。

图6- 4仅展示了一次编程的流程。若执行连续编程时，可以在写入FLASH\_ADDR寄存器前，配置FLASH\_CFG.ADR\_INC，开启地址自动递增模式，后续只需要反复写FLASHWDATA寄存器即可，FLASH\_ADDR每次写入一次数据会自动增加 。连续读操作类似。连续编程的流程如下：

  
图6-5 FLASH 模块编程操作流程图

# 6.2.1.5 FLASH 擦除操作

擦除操作为 FLASH 的基本操作。系统只能通过访问 FLASH 控制器的寄存器实现。具体流程为：

* FLASH 擦除操作使能- 地址寄存器 ADDR，写入擦除地址- 写擦除寄存器 ERASE，触发擦除操作

执行对 FLASH 存储体的擦除操作。擦除分成 Sector 和 FullChip。分别对应，512Byte 的擦除和 32KB/64kB 的擦除。通过配置 FLASH 控制寄存器决定执行哪一种类型的擦除操作。

下表为 Block 和 Secotor 地址分配空间。

表6-2FLASHSector地址分配表

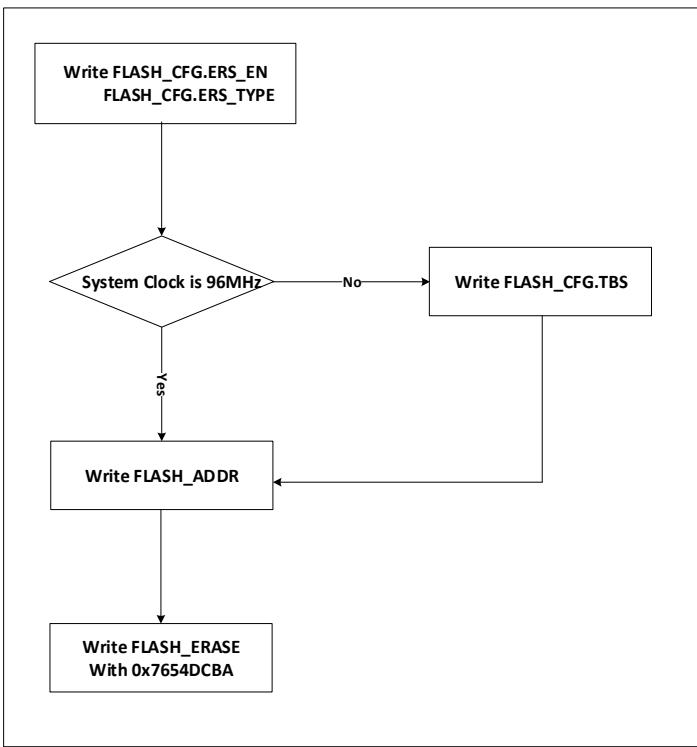
|  |  |  |
| --- | --- | --- |
| Name | Addresses | Size(Bytes) |
| Sector 0 | 0x0000 0000 - 0x0000 01FF | 512 |
| Sector1 | 0x0000 0200 - 0x0000 03FF | 512 |
| Sector2 | 0x0000 0400 - 0x0000 05FF | 512 |
| … | … | … |
| Sector127 | 0x0000 FE00 - 0x0000 FFFF | 512 |

NVR 区域只能实现 Sector 擦除；MAIN 区域可以实现 Sector 擦除和 FULL 擦除。具体表格如下：

|  |  |  |
| --- | --- | --- |
| NVR (FLASH\_CFGREGION) | Sector Erase | FULL Erase |
| 0 | Main 区域 | Main 区域 |

|  |  |  |
| --- | --- | --- |
| 1 | NVR 区域 | Main 区域 |

FLASH 擦除操作流程如下所示。

  
图6-6 FLASH 模块擦除操作流程图

若选择 Secotor 擦除，需要通过 FLASH\_ADDR 确定哪个 Secotor 被擦除，若是 Full Chip 模式的话，FLASH\_ADDR 的值将失效。FLASH\_ERASE 写入 0x7654DCBA 触发擦除操作。

# 6.2.1.6 FLASH 预取操作

因 FLASH 存储体的速度限制，无法达到 96MHz 的速度。当对 FLASH 进行读取操作时，需要大于 1 个时钟周期才能完成数据的读出。为了加快数据的读出，FLASH 控制器增加了预取功能。当 FLASH 控制器执行完当前读取操作后，在不影响正常程序执行的前提下，顺序预取下一个 WORD 的数据。预取操作的开启和关闭，只需要设置 FLASH\_CFG.PREF 即可。

# 6.2.1.7 FLASH 加密保护

若 FLASH 存储体内的数据处于加密状态，用户可执行解密操作，可对 FLASH 存储体内的数据进行解密。相反，若 FLASH 存储体内的数据处于解密状态，用户可执行加密操作，对 FLASH 存储体内的数据进行加密。默认情况下，FLASH 存储体内的数据处于加密状态。芯片上电复位后，硬件自动执行一次加密状态更新操作，是加密状态的话仍然加密状态，是解密的话状态变成解密状态。

FLASH 存储体共有 32kB/64kB 两个规格。无论哪种规格，相应规格中的最后一个 WORD 设计为加密字。当这个 WORD 内容为全 1 时，表明此时 FLASH 处于解密状态；当这个 WORD 的内容被写为非全 1 时，表明此时 FLASH 处于加密状态。若需要加密，执行最后一个 WORD 的编程，写入非全 1 的值，读取 FLASH\_PROTECT 寄存器，即触发一次加密状态更新，完成加密（读取 FLASH\_PROTECT 返回值无意义）。

对应的解密流程，分成两种情况。若最后一个 WORD 没有执行过编程写入非全 1 的操作，读取 FLASH\_PROTECT 寄存器，即完成解密更新（无需考虑此时返回值）。若已经执行过编程写入非全 1

的操作，那么只能执行擦除操作才能解除。先对 FLASH 执行擦除操作，将最后一个 WORD 恢复为全 1 值，然后读取 FLASH\_PROTECT 寄存器，即触发一次加密状态更新，完成解密（读取 FLASH\_PROTECT 返回值无意义）。

# 6.2.1.8 FLASH 在线升级(IAP)

IAP 模式，实现中断向量表的重映射。在 LKS32MC08x 系列芯片中，包含了系统寄存器 VTOR，其地址为 0xE000\_ED08。用于重新映射中断向量表入口地址。

表6-3IAPVTOR寄存器描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 复位值 | 偏移 | 位置 | 权限 | 说明 |
| VTOR | 0x0 |  | [31:7] | RW | 执行写入操作，写入中断向量表入口地址 |
| [6:0] | – | 默认写 0 |

默认值为 ，此时中断向量表入口地址为 。当写入非0值时，中断向量表入口地址将映射到写入值对应的地址上，立即生效。

在 LKS32MC08x 系列芯片中，因为有 VTOR 寄存器。用户可根据自己需求，更新整个 FLASH 的内容。在线升级过程中可以使用中断，也可以关闭中断。

# 6.2.1.8.1 开启中断的在线升级

推荐软件配置流程：

关闭 CPU 的中断控制器，暂时不接收新的中断响应；在新的中断入口地址处，放置中断处理函数代码；将新的中断入口地址写入 VTOR 寄存器；开启 CPU 的中断控制器，使能中断；用户跳转至在线升级函数，开始在线升级功能；完成升级，关闭 CPU 的中断控制器，配置 VTOR 为默认值 0；执行 CPU 软复位，系统 PC 重新从 0 地址开始执行升级后的程序；

# 6.2.1.8.2 关闭中断的在线升级

关闭 CPU 的中断控制器，暂时不接收新的中断响应；

用户跳转至在线升级函数，开始在线升级功能；如果在线升级使用了类似 UART 的外设通讯，需要 CPU 轮询处理 UART 的中断标志位。

执行 CPU 软复位，系统 PC 重新从 0 地址开始执行升级后的程序；

# 6.2.1.8.3在线升级函数的位置

如果需要将flash全部擦除，则需要将在线升级函数放置在RAM中，如果需要使用中断则新的中断向量入口地址也需要位于RAM地址空间。

如果只需要擦除应用程序占用的部分flash区域，则可以将在线升级函数放置在flash高段地址的空闲区域，使用块擦除flash旧的应用程序，写入新的应用程序。

# 6.3 寄存器

# 6.3.1 地址分配

FLASH控制器模块寄存器的基地址是0x4000\_0400，寄存器列表如下：

表6-4FLASH控制器模块寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| FLASH\_CFG | 0x00 | FLASH 配置寄存器 |
| FLASH\_ADDR | 0x04 | 地址寄存器 |
| FLASHWDATA | 0x08 | 写数据寄存器 |
| FLASH\_RDATA | 0x0C | 读数据寄存器 |
| FLASH\_ERASE | 0x10 | 擦除使能寄存器 |
| FLASH\_PROTECT | 0x14 | FLASH 保护状态寄存器 |
| FLASH\_READY | 0x18 | FLASH 闲忙状态寄存器 |

# 6.3.2 FLASH\_CFG配置寄存器（推荐先读回，按或/与方式修改）

地址：0x4000\_0400

复位值：0x00000060

表6-5配置寄存器FLASH\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERS\_EN |  |  | RMS\_EN |  |  | ADR\_INC |  |  | RREF |  |  |  |  |  |  |
| RW |  |  | RW |  |  | RW |  |  | RW |  |  |  |  |  |  |
| 0 |  |  | 0 |  |  | 0 |  |  | 0 |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERS\_TYPE |  |  | REGION |  |  |  |  |  | TBS | | | | | |  |
| RW |  |  | RW |  |  |  |  |  | RW | | | | | |  |
| 0 |  |  | 0 |  |  |  |  |  | 60 | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31] | ERS\_EN | FLASH 擦除使能。默认为 0。 0：关闭擦除 |

|  |  |  |
| --- | --- | --- |
|  |  | 1：开启擦除 |
| [27] | PRG\_EN | FLASH编程使能。默认为0。 0：关闭编程 1：开启编程 |
| [23] | ADR\_INC | FLASH地址递增使能。默认为0。 0：关闭递增使能 1：开启递增使能 当执行FLASH连续读写访问时，可以开启此功能减少对地址的操作。 |
| [19] | PREF | FLASH预取加速使能。默认为0。 0：关闭加速 1：开启加速 |
| [15] | ERS\_TYPE | FLASH擦除类型选择。默认为0。 0：Sector 1：FULL |
| [11] | REGION | 访问FLASH区域选择。默认为0。 0：MAIN 1：NVR |
| [6:0] | TBS | 编程/擦除时间基数寄存器，默认值为0x60。只能配成如下几个值 0x60：96Mhz系统频率下，FLASH编程/擦除时间基数配置值。 0x2F：48Mhz系统频率下，FLASH编程/擦除时间基数配置值。 0x17：24Mhz系统频率下，FLASH编程/擦除时间基数配置值。 0x0B：12Mhz系统频率下，FLASH编程/擦除时间基数配置值。 |

# 6.3.3 FLASH\_ADDR地址寄存器

地址：0x4000\_0404

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | ADDR | 地址寄存器。读/写/擦除操作对应的地址寄存器。因按照 WORD 操作，最低两位会被 FLASH 控制器忽略。 执行擦除操作时，需要根据擦除类型，地址需要对齐。一个 Sector 是 512-Byte。若执行 Sector 擦除，地址需要是 512 的整数倍（若带偏移，偏移量会被忽略）。全芯片擦除，不会参考这个寄存器的值 |

# 6.3.4 FLASHWDATA写数据寄存器

地址：0x4000\_0408

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| WDATA | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDATA | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | WDATA | 执行写入操作，写入 FLASH 的值 |

# 6.3.5 FLASH\_RDATA读数据寄存器

地址：0x4000\_040C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RDATA | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |
| 15 | -14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RDATA | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | RDATA | 执行读取操作，读出 FLASH 的值 |

# 6.3.6 FLASH\_ERASE擦除控制寄存器

地址：0x4000\_0410

复位值：0x0

表6-9擦除控制寄存器FLASH\_ERASE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERASE | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERASE | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | ERASE | 写入 0x7654DCBA，触发擦除操作 |

# 6.3.7 FLASH\_PROTECT 加密状态寄存器

地址：0x4000\_0414

复位值：0x0

表6-10加密状态寄存器FLASH\_PROTECT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PROTECT | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |
| <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <nl> |
| <lcel> | | | | | | | | | | | | | | | |
| <lcel> | | | | | | | | | | | | | | |  |
| <lcel> | | | | | | | | | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | READY |
| RO |
| 0 |
| 位置 | 位名称 | 说明 | |
| [31:1] |  | 未使用 | |
| [0] | READY | 1:FLASH 处于闲状态；0:FLASH 处于忙状态 | |

# 6.3.9 NVR校正值地址信息

LKS32MC08x产品存储芯片校准参数：

校准参数，每颗产品均独立校准，每颗产品不支持校准参数混用；校准参数，在出厂前进行写入，出厂后不支持编程和擦除，仅支持读取；校准参数，通过LKS公司提供的库函数进行读取访问；校准参数，推荐关闭系统中断后，执行访问操作；

lks32mc08x\_nvr.o文件包含了校准参数的读取函数：

读取函数：uint32\_tRead\_Trim(uint32\_t adr);

表6-12校准参数列表

|  |  |
| --- | --- |
| 地址 | 内容 |
| 0x0310 | ADC0\_DCO 校准值 |
| 0x0314 | ADC0\_DC1 校准值 |
| 0x0318 | ADC0\_AMC0 校准值 |
| 0x031C | ADC0\_AMC1 校准值 |
| 0x0320 | ADC1\_DC0 校准值 |
| 0x0324 | ADC1\_DC1 校准值 |
| 0x0328 | ADC1\_AMC0 校准值 |
| 0x032C | ADC1\_AMC1 校准值 |
| 0x0330 | DAC 选择 3.00V 档位，SYS\_AFE\_DAC\_AMC 校准值（扩大 512 倍结果） |
| 0x0334 | DAC 选择 1.20V 档位，SYS\_AFE\_DAC\_AMC 校准值（扩大 512 倍结果） |
| 0x0338 | DAC 选择 4.85V 档位，SYS\_AFE\_DAC\_AMC 校准值（扩大 512 倍结果） |
| 0x0340 | DAC 选择 3.00V 档位，SYS\_AFE\_DAC\_DC 校准值 |
| 0x0344 | DAC 选择 1.20V 档位，SYS\_AFE\_DAC\_DC 校准值 |
| 0x0348 | DAC 选择 4.85V 档位，SYS\_AFE\_DAC\_DC 校准值 |
| 0x0350 | OPA0，200K 欧姆 VS 10.4K 欧姆，GAIN 校准值（扩大 1000 倍结果），R0 为 0 欧姆 |
| 0x0354 | OPA0，190K 欧姆 VS 20.4K 欧姆，GAIN 校准值（扩大 1000 倍结果），R0 为 0 欧姆 |
| 0x0358 | OPA0，180K 欧姆 VS 30.4K 欧姆，GAIN 校准值（扩大 1000 倍结果），R0 为 0 欧姆 |

|  |  |
| --- | --- |
| 0x035C | OPA0, 170K 欧姆 VS 40.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0360 | OPA1, 200K 欧姆 VS 10.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0364 | OPA1, 190K 欧姆 VS 20.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0368 | OPA1, 180K 欧姆 VS 30.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x036C | OPA1, 170K 欧姆 VS 40.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0370 | OPA2, 200K 欧姆 VS 10.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0374 | OPA2, 190K 欧姆 VS 20.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0378 | OPA2, 180K 欧姆 VS 30.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x037C | OPA2, 170K 欧姆 VS 40.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0380 | OPA3, 200K 欧姆 VS 10.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0384 | OPA3, 190K 欧姆 VS 20.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0388 | OPA3, 180K 欧姆 VS 30.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x038C | OPA3, 170K 欧姆 VS 40.4K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆 |
| 0x0398 | 温度传感器, 斜率校准值 |
| 0x039C | 温度传感器, 斜率校准值 |
| 0x03B0 | 高 16-bit 存放 OPA1 共模电压, 低 16-bit 存放 OPA0 共模电压 (注意, 均是放大 10000 倍存入) |
| 0x03B4 | 高 16-bit 存放 OPA3 共模电压, 低 16-bit 存放 OPA2 共模电压 (注意, 均是放大 10000 倍存入) |
| 0x02C0 | OPA0, 200K 欧姆:10.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02C4 | OPA0, 190K 欧姆:20.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02C8 | OPA0, 180K 欧姆:30.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02CC | OPA0, 170K 欧姆:40.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02D0 | OPA1, 200K 欧姆:10.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02D4 | OPA1, 190K 欧姆:20.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02D8 | OPA1, 180K 欧姆:30.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02DC | OPA1, 170K 欧姆:40.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02E0 | OPA2, 200K 欧姆:10.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02E4 | OPA2, 190K 欧姆:20.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02E8 | OPA2, 180K 欧姆:30.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |
| 0x02EC | OPA2, 170K 欧姆:40.4K 欧姆。高 16 位为 R2 实际电阻值, 低 16 位为 R1 实际电阻值 (扩大 100 倍结果) |

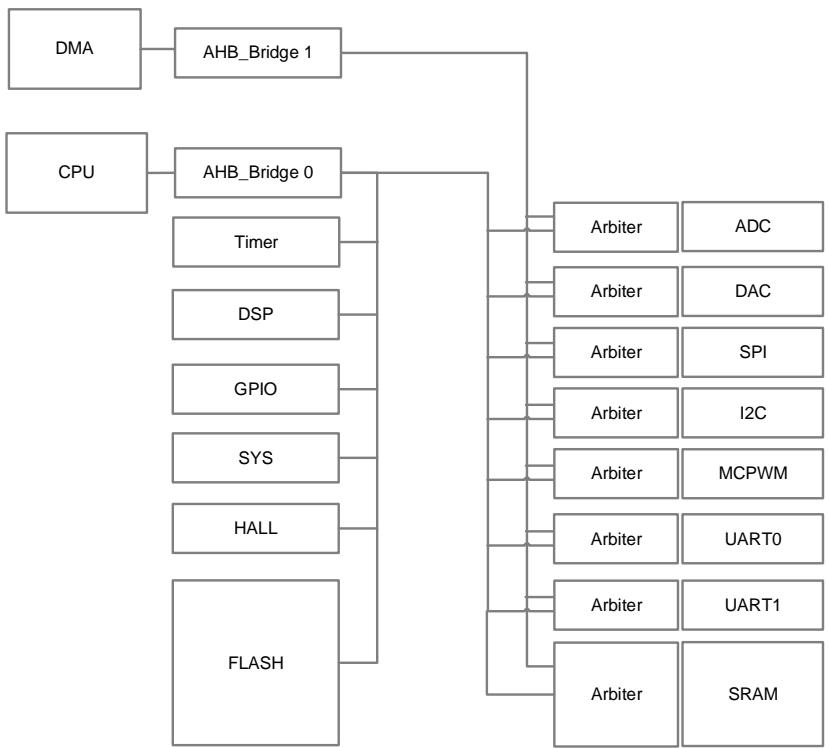
|  |  |
| --- | --- |
| 0x02F0 | OPA3, 200K 欧姆:10.4K 欧姆。高 16 位为 R2 实际电阻值，低 16 位为 R1 实际电阻值（扩大 100 倍结果） |
| 0x02F4 | OPA3, 190K 欧姆:20.4K 欧姆。高 16 位为 R2 实际电阻值，低 16 位为 R1 实际电阻值（扩大 100 倍结果） |
| 0x02F8 | OPA3, 180K 欧姆:30.4K 欧姆。高 16 位为 R2 实际电阻值，低 16 位为 R1 实际电阻值（扩大 100 倍结果） |
| 0x02FC | OPA3, 170K 欧姆:40.4K 欧姆。高 16 位为 R2 实际电阻值，低 16 位为 R1 实际电阻值（扩大 100 倍结果） |

# 7 DMA

# 7.1 概述

增加DMA后，总线上主设备由CPU增加为CPU和DMA，总线架构需要由AHBlite 演进为multi- layerAHBlite架构。如图7- 1所示。其中部分设备不需要被DMA访问，仅仅挂载于与CPU相连的AHBBridge0上。包括ADC/DAC/SPI/I2CMCPWM/UART/SRAM在内的设备被CPU和DMA共享访问，挂载于AHBBridge1上。而Timer/DSP/GPIO/Hall/Flash等设备，DMA无法访问。

从设备端的仲裁模块本质上相当于一个二选一的多路选择器。只有端口上有仲裁器的从设备才能被DMA访问到，否则无法被DMA访问，而所有外设设备均可被CPU访问。

  
图7-1 multi-layerAHBlite总线架构

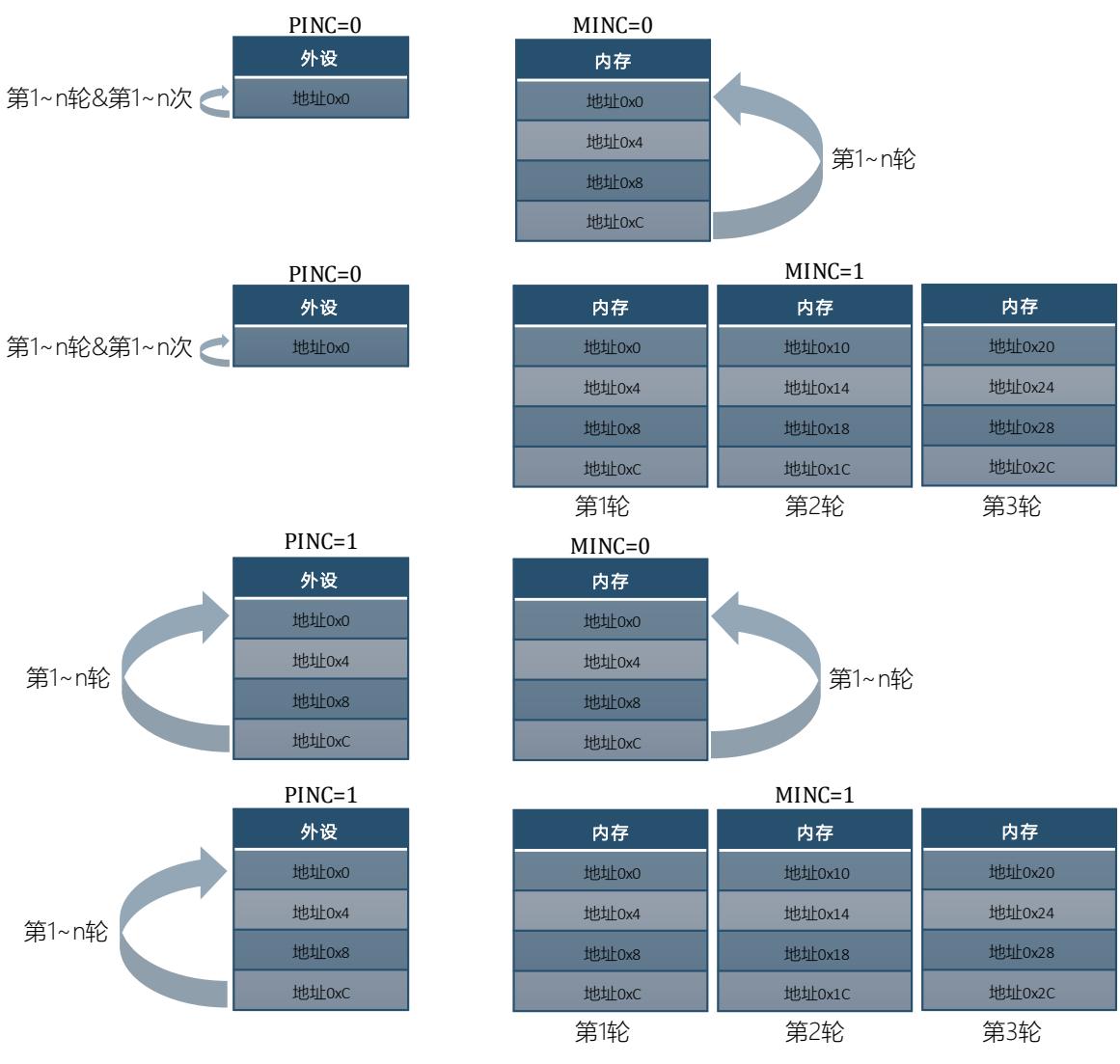
出于功耗控制考虑，DMA模块可以通过设置DMA\_CTRL.EN位为0来被禁用（要求关闭DMA使能前先关闭4个通道对应的使能DMA\_CCRx.EN），此时DMA时钟被门控关闭。DMA包含配置寄存器（相当于一个从设备，被CPU配置）和数据搬运模块（对于总线为主设备，向总线发起各种设备访问请求）。

DMA支持8，16或32bit(byte，half- word，orword）三种位宽的传输操作，通过配置DMA\_CCRx.PBTW和DMA\_CCRx.MBTW来选择外设和内存访问的位宽，外设访问的位宽与内存访问的位宽可以不同。

DMA每完成一次传输，地址根据DMA\_CCRx.PINC和DMA\_CCRx.MINC决定是否自动递增。所有的外设寄存器地址都是word对齐的，所以外设的地址递增永远是0/4（根据DMA\_CCRx.PINC=0/1设置)。例如对于UART/SPI/I2C，因为每次都是访问UART\_DATA或SPI/I2CFIFO接口的固定地址，DMA\_CCRx.PINC=0；如要访问ADC数据寄存器，则地址通常需要自动加4，需要设置

DMA\_CCRx.PINC=1。而对于内存，如果设置了DMA\_CCRx.MINC=1，每次地址递增的值，根据内存数据位宽(DMA\_CCRx.MBTW)设定，对于内存访问位宽为byte的情况，地址自动加1，对于half- word，地址自动加2，对于word，地址自动加4。

需要注意的是，我们将DMA的传输分为多个轮次，每轮传输内可以有多次传输。DMA\_CCRx.PINC用于控制外设地址每次传输是否递增。外设地址每轮传输一定重复上一轮的地址，即轮内可以递增，轮间不递增；DMA\_CCRx.MINC用于控制内存地址轮间是否递增，每次传输一定递增。

  
图7-2 DMA地址递增控制

DMA有循环模式和单次模式两种模式，由DMA\_CCRx.CIRC控制。循环模式下DMA完成一定大小的数据块搬运后重新开始下一轮搬运，如果是搬运数据到内存，则覆盖之前搬运至内存的数据；如果是搬运至外设，则重复一轮数据发射。以将ADC数据搬运至内存为例，设定数据块大小为16bit×12channel×8次，则在循环模式下DMA完成一轮96个half word搬运后重新开始下一轮搬运，并覆盖之前使用的内存地址，不设置DMA完成中断标志位。单次模式下，DMA完成一定大小的数据块搬运后即完成本次DMA操作，设置完成DMA完成中断标志位，同时硬件自动关闭对应的DMA通道，即硬件电路自动在该通道传输完成后将DMA\_CCRx.EN设为0。DMA需要搬运的轮次由DMA\_CTMS寄存器进行控制。

# 7.2 请求

DMA的请求分为软件请求和硬件请求两类，软件请求通过设置对应DMA通道的DMA\_CCRx.SW\_TRIG 来产生，写1即产生，软件触发位设置后需要软件清零。硬件请求通常为外设的中断事件，当特定的外设中断事件作为DMA传输请求时，通常要禁用对应事件的中断响应。而且硬件DMA请求信号经过DMA通道受理握手后会被DMA硬件清零，无法软件清除事件标志。

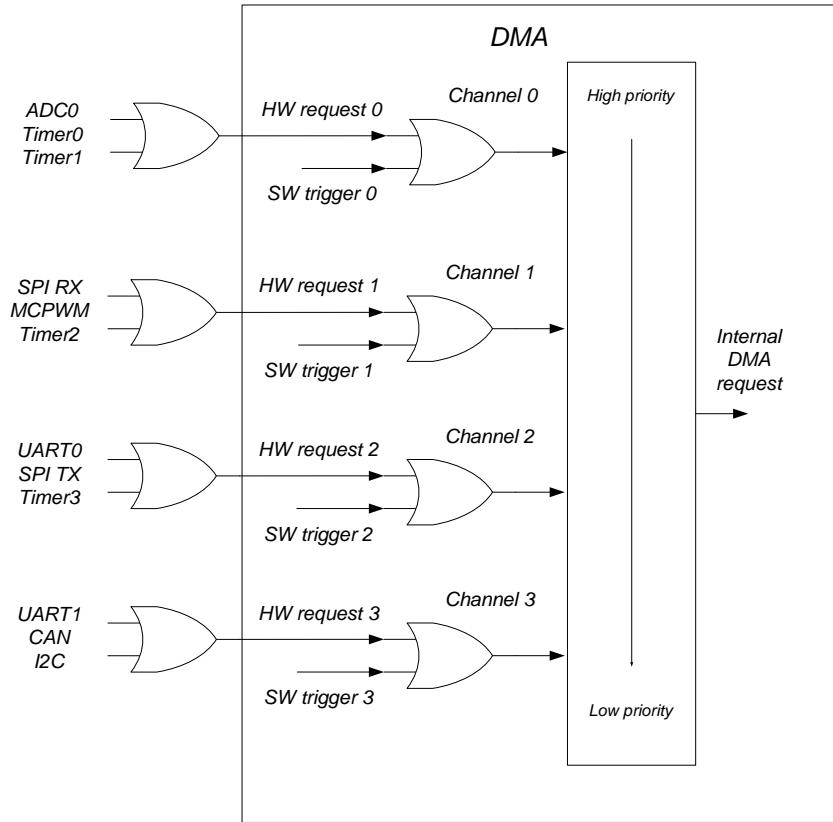
表7-1DMA请求

|  |  |
| --- | --- |
| 触发来源 | 描述 |
| 软件 | 软件触发时进行的搬运操作由配置寄存器DMA\_CCRx指定，当设置了DMA\_CCRx.SW\_TRIG，一旦通道被使能即开始进行DMA 操作。此版本DMA 暂不支持内存至内存搬运！！！ |
| ADC | ADC在单段触发模式下，一次完成若干个通道后采样后产生中断请求，由DMA把ADC转换后的值搬运到SRAM。ADC单段采样完成中断事件作为DMA请求信号，请求信号在得到DMA响应后由DMA将其清零，不要软件清零；注意此时需要软件同时禁用ADC采用完成中断，使中断不再被CPU响应。 |
| UART | 串口模块使用 UART\_IF触发DMA请求，如果DMA配置的传输方向是内存到串口，则使用UART发送完成事件产生DMA请求信号；如果传输方向是串口到内存，则应使用UART接收完成事件产生DMA请求信号。事件标志由DMA自动清零。UART在由DMA操作时应同样禁用相应中断防止被CPU响应。有两种方案可选： 方案1：若 UARTx\_IE.TX\_BUF\_EMPTY\_RE配置有效，UARTx模块将预取第一个字节，准备发送；一旦数据进入发射队列，UARTx\_IE.UARTx\_BUF 不能为空，硬件自动会请求DMA搬移下一个字节直至数据搬移完毕。DMA搬移完毕后，将产生DMA完成中断，但是UARTx很可能没有发送完毕最后一个字节，若立即操作UARTx，将可能会产生异常。建议在DMA中断处理程序中，开启UARTx\_IE.TX\_DONE\_IE中断，UARTx将最后一个字节发送完毕，产生发送完成中断，在UARTx中断处理函数里面，再关闭UARTx\_IE.TXT\_DONE\_IE。 方案2：若 UARTx\_IE.TX\_DONE\_RE配置有效。建议UART初始化阶段不要清除TX\_DONE标志。当前传输的数据长度为Len，DMA配置传输的字节数为Len，开启DMA中断，DMA传输完毕后，UART也发送完毕，软复位UARTx模块并重新初始化UARTx，可开启下一次UARTx的发送。 |
| SPI | SPI模块使用接收缓冲区满事件作为DMA请求信号，由于SPI是同时收发，所以接收缓冲区满既是接收完毕也是发送完毕的事件标志。读取SPI FIFO自动清除事件标志。 |
| I2C | I2C模块使用I2C0\_SCR.BYTE\_CMPLT即字节发送完成作为触发DMA请求，DMA自动清除I2C的请求标志。其他I2C中断事件仍由CPU响应 |
| Timer | Timer使用过零/比较/捕获事件作为DMA请求，具体DMA操作根据配置寄存器设定，通常作为定时事件（如每隔10ms触发一次DMA操作） |
| MCPWM | MCPWM模块使用过零/计数周期结束/4个ADC触发信号作为DMA请求，具体DMA操作根据配置寄存器设定 |
| CAN |  |

# 7.3 优先级

DMA的优先级采用固定优先级，优先级如图7- 3所示。为避免出现来不及响应某些外设请求的情况，在设计应用软件时应考虑任务实时性，每个通道不配置搬运过于大量的数据导致其他通道得不到及时响应。

如图7- 3所示，优先级由上至下降低。4个DMA通道中，优先级关系为：通道 通道 通道 通道3(>号表示优先级高于)。在DMA各通道内部，通常有3个硬件请求事件和一个软件请求事件，硬件请求优先级高于软件请求。3个硬件请求事件优先级相同，通常应用上面一个DMA通道配置一个硬件请求事件使能，多个硬件请求不应在一个通道内同时发生。

  
图7-3DMA通道优先级

# 7.4 仲裁

当DMA处于空闲状态，或刚刚完成某一通道的DMA传输后，若此时恰好有一个或多个DMA请求发生，应根据优先级设定进行仲裁，优先级高的外设请求率先得到DMA服务。比如ADC连续模式下，每完成一轮ADC数据搬运，ADC的采样完成事件标志被DMA清零，DMA回到空闲状态或转而服务其他外设请求；对于UART，每搬运一个byte重新仲裁；对于SPI/I2C每搬运完一个FIFO的数据，重新仲裁。

CPU访问RAM的优先级始终高于DMA。

为了避免CPU或DMA长期占用外设/SRAM，在外设/SRAM的端口仲裁模块中加入了时间片机制，即一个主设备占用一段时间后释放访问权，仲裁模块观测另一个主设备是否在请求访问，如果

是则转而允许另一个主设备访问，否则继续当前主设备未完成的访问。

# 7.5 中断

DMA的一个通道完成DMA操作后或出错则产生DMA中断。当DMA某一个通道完成操作后，会自动关闭该通道的使能为DMACCRx.EN。

# 7.6 寄存器

# 7.6.1 地址分配

DMA控制器模块寄存器的基地址是0x4001\_3000，寄存器列表如下：

表7-2DMA寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| DMACCR0 | 0x00 | DMA通道0通道配置寄存器 |
| DMACTMS0 | 0x04 | DMA通道0传输次数寄存器 |
| DMACPAR0 | 0x08 | DMA通道0外设地址寄存器 |
| DMACMAR0 | 0x0C | DMA通道0内存地址寄存器 |
| DMACCR1 | 0x10 | DMA通道1通道配置寄存器 |
| DMACTMS1 | 0x14 | DMA通道1传输次数寄存器 |
| DMACPAR1 | 0x18 | DMA通道1外设地址寄存器 |
| DMACMAR1 | 0x1C | DMA通道1内存地址寄存器 |
| DMACCR2 | 0x20 | DMA通道2通道配置寄存器 |
| DMACTMS2 | 0x24 | DMA通道2传输次数寄存器 |
| DMACPAR2 | 0x2B | DMA通道2外设地址寄存器 |
| DMACMAR2 | 0x2C | DMA通道2内存地址寄存器 |
| DMACCR3 | 0x30 | DMA通道3通道配置寄存器 |
| DMACTMS3 | 0x34 | DMA通道3传输次数寄存器 |
| DMACPAR3 | 0x38 | DMA通道3外设地址寄存器 |
| DMACMAR3 | 0x3C | DMA通道3内存地址寄存器 |
| DMACTRL | 0x40 | DMA控制寄存器 |
| DMAIF | 0x44 | DMA中断标志寄存器 |

# 7.6.2 DMA\_CTRLDMA控制寄存器

地址：0x4001\_3040

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7-3 DMA 控制寄存器 DMA\_CTRL | | | | | | | | | | | | |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  | PRIORITY | EN |  |

|  |  |  |
| --- | --- | --- |
|  | RW | RW |
| 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:2] |  | 未使用 |
| [1] | PRIORITY | 0:CPU 优先级高；1:DMA 优先级高，此版本必须设置 CPU 优先级高 |
| [0] | EN | DMA 使能 |

# 7.6.3 DMA\_IF DMA 中断标志寄存器

地址：0x4001\_3044

复位值：0x0

15 14 13 12 11 10 0 8 7 6 5 4 3 2 1 0

表7-4DMA中断标志寄存器DMA\_IF

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CH3\_EIF | CH2\_EIF | CH1\_EIF | CH0\_EIF | CH2\_FIF | CH2\_FIF | CH1\_FIF | CH0\_FIF |
| RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | CH3\_EIF | 通道3错误中断标志 |
| [6] | CH2\_EIF | 通道2错误中断标志 |
| [5] | CH1\_EIF | 通道1错误中断标志 |
| [4] | CH0\_EIF | 通道0错误中断标志 |
| [3] | CH3\_FIF | 通道3完成中断标志 |
| [2] | CH2\_FIF | 通道2完成中断标志 |
| [1] | CH1\_FIF | 通道1完成中断标志 |
| [0] | CH0\_FIF | 通道0完成中断标志 |

# 7.6.4 DMA 通道配置寄存器

# 7.6.4.1 DMA\_CCRx(where

地址分别是：0x4001\_3000，0x4001\_3010，0x4001\_3020，0x4001\_3030

复位值：0x0

表7-5DMA通道配置寄存器DMA\_CCRx

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SW\_TRIG | REQ\_EN | MBTW | PBTW | MINC | PINC | CIRC | DIR | TEIE | TCIE | EN |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | SW\_TRIG | 软件触发，高有效\*2 |
| [14:12] | REQ\_EN | 通道X三个硬件DMA请求使能\*1，高有效 |
| [11:10] | MBTW | 内存访问位宽，0:byte, 1:half-word, 2:word, 3:保留 |
| [9:8] | PBTW | 外设访问位宽，0:byte, 1:half-word, 2:word, 3:保留 |
| [7] | MINC | 内存地址第二轮是否在第一轮地址的基础上递增（轮内一定递增），高有效 |
| [6] | PINC | 指示外设地址每轮内是否递增（外设地址第二轮一定重复第一轮地址），高有效 |
| [5] | CIRC | 循环模式，高有效 |
| [4] | DIR | 传输方向，0:外设至内存，1:内存至外设 |
| [3] |  | 保留 |
| [2] | TEIE | 错误中断使能，高有效 |
| [1] | TCIE | 传输完成中断使能，高有效 |
| [0] | EN | 通道&amp;使能，高有效，在通道操作全部完成后由DMA清除此位 |

以通道0为例，DMA\_CCR0. REQ\_EN[2:0]分别为Timer1、Timer0、ADC0的DMA请求使能。且DMA的通道中配置的内存、外设地址应与使能的外设中断请求相对应，这一点需要由应用软件保证。其中软件请求始终处于使能状态，即软件写入DMA\_CCRx.SW\_TRIG位即开始一次DMA传输。来自外设的硬件请求进入DMA后通过或逻辑形成一个请求信号，每一个DMA通道同时只应使能一个硬件DMA请求。 软件触发标志DMA\_CCRx.SW\_TRIG写入1后需要软件清0。

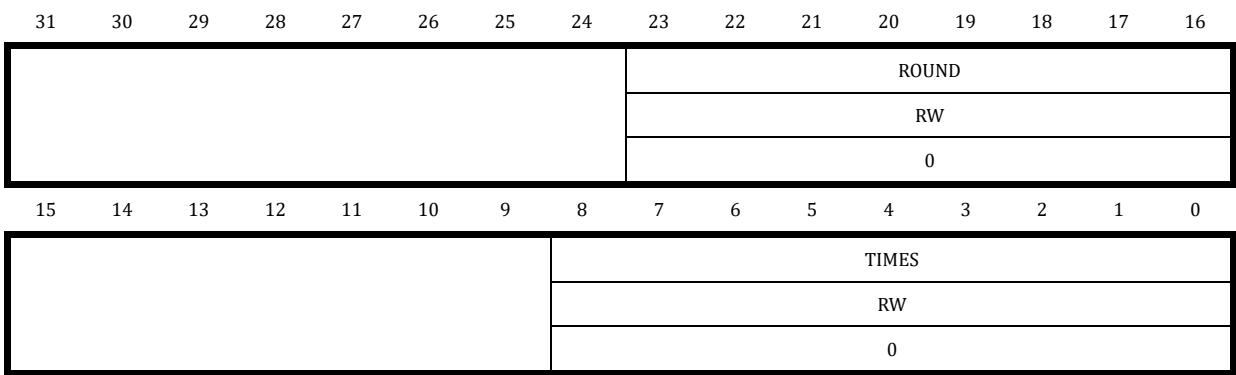
表7-6DMA通道请求信号

|  |  |  |
| --- | --- | --- |
| DMA 通道 | 设备请求信号编号 | 外设 |
| Channel 0 | 0 | ADC0 |
| 1 | Timer0 |
| 2 | Timer1 |
| Channel 1 | 0 | SPI\_RX |
| 1 | MCPWM |
| 2 | Timer2 |
| Channel 2 | 0 | UART0 |
| 1 | SPI\_TX |
| 2 | Timer3 |
| Channel 3 | 0 | UART1 |
| 1 | CAN |
| 2 | I2C |

# 7.6.4.2 DMA\_CTMSx (where

地址分别是：0x4001\_3004，0x4001\_3014，0x4001\_3024，0x4001\_3034

复位值：0x0



|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:24] |  | 未使用 |
| [23:16] | ROUND | DMA 通道 x 采样轮数 |
| [15:9] |  | 未使用 |
| [8:0] | TIMES | DMA 通道 x 每轮数据搬运次数，0～512。此寄存器在该通道使能后变为只读。 |

DMA\_CTMSx寄存器只有在通道禁用，即DMA\_CCRx.EN=0之后才可以写入数据。

以外设数据宽度为16，内存数据宽度为32，即CTMS.TIMES ，CTMS.ROUND 为例，DMA每轮需要读取外设数据16bit byte，写入内存数据32bit byte。一共需要搬运两轮，即读取外设64byte，输入内存128byte；

即使仅仅搬运一轮，也需要设置CTMS.ROUND ，而不能让其为0。

当设置DMA\_CCRx.CIRC=1（即循环模式）时，CTMS.ROUND不再起作用，相当于无限轮；其他情况需要相应设置CTMS.ROUND，如CTMS.ROUND ，用于搬运一轮数据。

当DMA\_CTRL=1且DMA\_CCRx.EN=0时，重新填写CTMSx值，可以将DMA内部已搬运的轮数次数计数器清零。

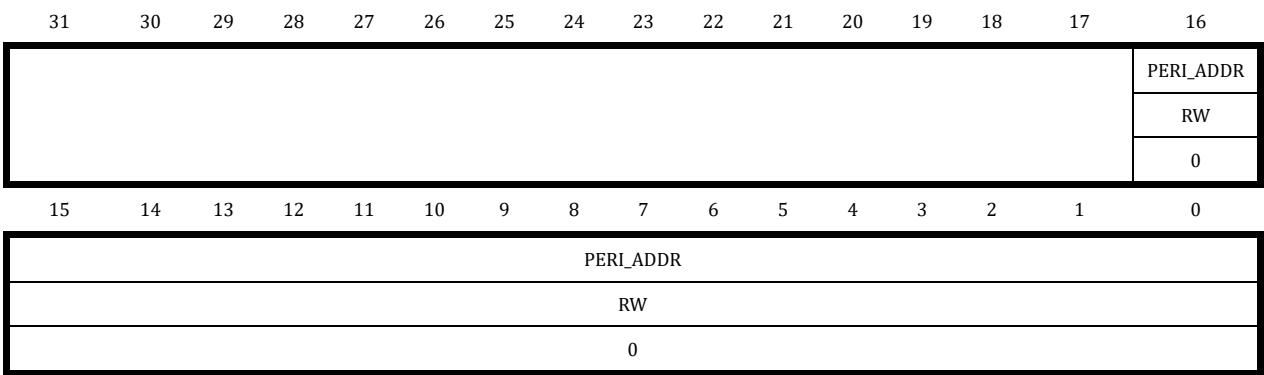
通常在开启DMA之前需要将触发DMA启动的外设接收中断标志位清零，防止之前留下的中断标志位成为DMA触发源造成一次错误触发。以UART为例，通常DMA被配置为多轮搬运，每轮搬运一次，每次搬运一个字节。如果UART接收FIFO不空，且UART\_IF[1]未清零，则使能DMA通道后立即发生一次DMA搬运，但此时的DMA搬运的数据是UART之前遗留的数据，并非实际需要搬运的数据，且此时DMA内部的轮次增1，可能会导致接收的UART数据帧发生错位。出现这种情况，可以通过重写DMA\_CTMS来讲DMA内部的轮次计数重置。

而在使用DMA通过UART发送数据时，通常使用的是UART TX\_FIFO空标志作为DMA请求，外设上电后发送FIFO空即有效，无须软件清除这一标志。否则无法向DMA产生第一次的发送请求。

# 7.6.4.3 DMA\_CPARx (where

地址分别是：0x4001\_3008，0x4001\_3018，0x4001\_3028，0x4001\_3038

复位值：0x0



|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:17] |  | 未使用 |
| [16:0] | PERI\_ADDR | DMA 通道 x 外设地址 |

当DMA\_CCRx.PBTW=7’b01时，即配置为以16bit为单位搬运外设数据。CPARx.PERI\_ADDR[0]值无效，外设地址会以2为单位递增。

当DMA\_CCRx.PBTW=7’b10时，即配置为以32bit为单位搬运外设数据。CPARx.PERI\_ADDR[1:0]值无效，外设地址会以4为单位递增。

注意：DMA\_CPARx寄存器只有在通道禁用，即DMA\_CCRx.EN=0之后才可以写入数据！！！

由于只进行内存与外设间的数据搬运，因此DMA\_CPAR只存储外设地址的低17位，高15位恒为 ，高20位可能为 (对应SYS寄存器)或 (对应外设寄存器)。

# 7.6.4.4 DMA\_CMARx (where

地址分别是：0x4001\_300C，0x4001\_301C，0x4001\_302C，0x4001\_303C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |

当 DMA\_CCRx.MBTW=2’b01 时, 即配置为以 16bit 为单位搬运内存数据。CMARx.MEM\_ADDR[0] 值无效, 内存地址会以 2 为单位递增。

当 DMA\_CCRx.MBTW=2’b10 时, 即配置为以 32bit 为单位搬运内存数据。CMARx.MEM\_ADDR[1:0] 值无效, 内存地址会以 4 为单位递增。

注意：DMA\_CMARx 寄存器只有在通道禁用, 即 DMA\_CCRx.EN=0 之后才可以写入数据!!!

由于只进行内存与外设间的数据搬运, 因此 DMA\_CMAR 只存储地址的低 13 位, 对应 SRAM8kB 地址空间。高 19 位恒为 0x10000。

# 8 GPIO

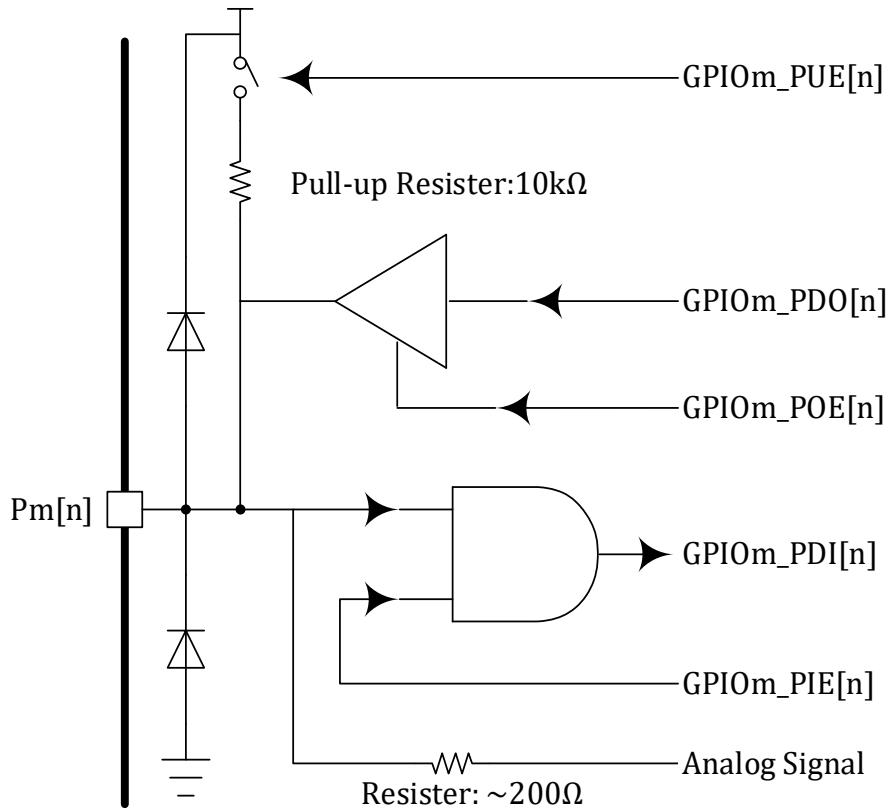
# 8.1 概述

LSK32MC08X系列芯片共集成了4组16bit位宽GPIO。P0.0/P0.1/P1.0/P1.14个GPIO可以作为系统的唤醒源。 共16个GPIO可以用作外部中断源输入。

其中P0.2既可以当做外部复位脚，也可以当做GPIO使用。可以通过软件设置SYS\_RST\_CFG.RST\_IO进行切换。上电后，P0.2默认用做外部复位脚，因此应用上应该注意上电后P0.2的外部信号不能一直为低，否则芯片会一直处于复位状态。复位释放后，软件可以将SYS\_RST\_CFG.RST\_IO置1，将P0.2切换为GPIO功能，SYS\_RST\_CFG受SYS\_WR\_PROTECT保护。

在某些低引脚数封装中，例如SSOP24，P2.15与SWDIO复用为同一个引脚。此种复用是通过封装直接bonding在一起实现的。芯片上电后，P2.15默认状态下输入输出全部禁用，此时不影响SWDIO参与SWD通讯。如果应用上需要将此引脚复用为GPIO，即使用P2.15，需要注意保留某种手段再次关闭P2.15的输出使能，以将引脚切换成SWDIO，否则可能导致芯片SWD无法使用。

# 8.1.1 功能框图

  
图8-1 GPIO功能框图

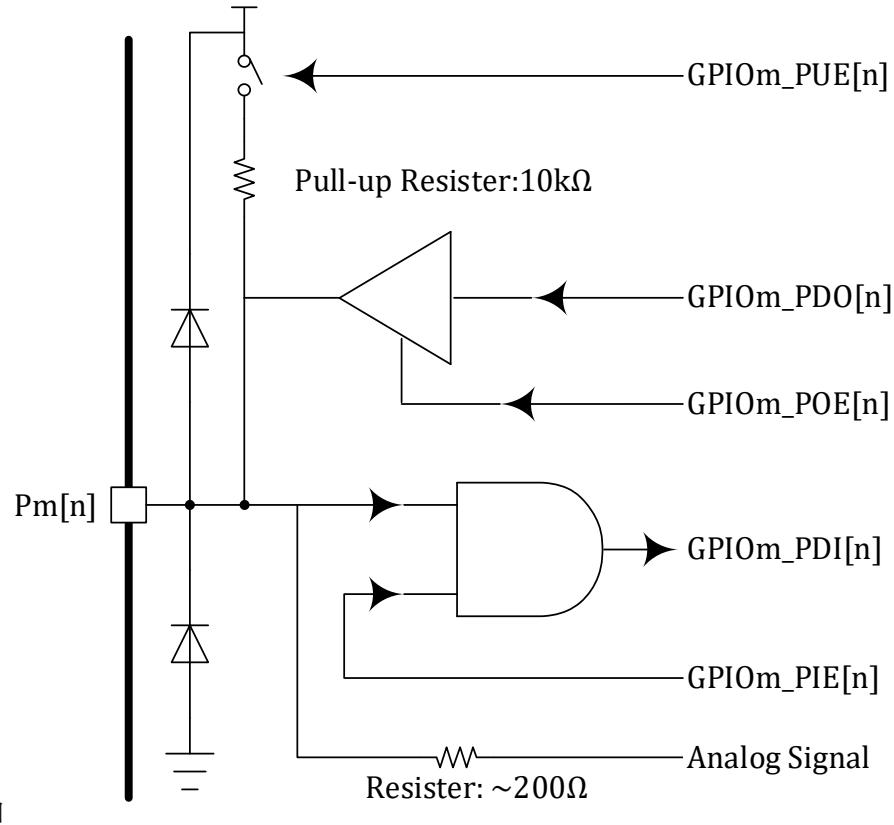


图8- 1所示， 为芯片PAD，m可以是 ，表示4组GPIO中的任意一组，n可以是 ，表示一组16bit GPIO中的一个IO。模拟信号通过一个电阻串联直接连接到PAD，电阻阻值为 。数字信号经过一个三态门输出，当输出使能GPIom\_POE[n] 时，buffer输出高阻态，否则buffer输出与GPIom\_PDO[n]电平相同。数字信号输入通过一个与门进入芯片内部，当GPIom\_PIE[n] 时，GPIom\_PDI[n]恒为0，当GPIom\_PIE[n] ，即输入使能打开时，GPIom\_PDI[n]的电平与 电平相同。芯片PAD可以配置上拉，P0[2]引脚因为复用为外部复位脚RSTN上拉电阻为 ，其余上拉电阻为 ，注意，并非所有PAD都配备上拉电阻，具体哪些PAD具有上拉电阻资源，请参考8.3.1上拉实现章节。没有上拉电阻的PAD也可以配置GPIom\_PUE[n]寄存器，但无实际作用。

# 8.1.2 产品特点

* 4组16bit GPIO- 支持开漏- 部分IO支持上拉- 支持配置锁定保护- 支持外部中断- 支持GPIO唤醒

# 8.2 寄存器

# 8.2.1 地址分配

GPIO0模块在芯片中的基地址是0x40012000。GPIO1模块在芯片中的基地址是0x40012040。GPIO2模块在芯片中的基地址是0x40012080。GPIO3模块在芯片中的基地址是0x400120C0。

GPIO0/1/2/3的寄存器定义完全相同，仅基地址不同，寄存器列表如下：

表8-1GPIOx寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| GPIOx\_PIE | 0x00 | GPIO x 输入使能 |
| GPIOx\_POE | 0x04 | GPIO x 输出使能 |
| GPIOx\_PDI | 0x08 | GPIO x 输入数据 |
| GPIOx\_PDO | 0x0C | GPIO x 输出数据 |
| GPIOx\_PUE | 0x10 | GPIO x 上拉使能 |
| GPIOx\_PODE | 0x18 | GPIO x 开漏使能 |
| GPIOx\_LCKR | 0x1C | GPIO x 配置锁定 |
| GPIOx\_F3210 | 0x20 | GPIO x [3:0] 功能选择 |
| GPIOx\_F7654 | 0x24 | GPIO x [7:4] 功能选择 |
| GPIOx\_FBA98 | 0x28 | GPIO x [11:8] 功能选择 |
| GPIOx\_FFEDC | 0x2C | GPIO x [15:12] 功能选择 |

GPIO中断/唤醒/配置锁定模块的基地址是0x40012100，寄存器列表如下：

表8-2GPIO中断/唤醒/配置锁定模块寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| EXTI\_CR0 | 0x00 | GPIO 0[7:0] 中断触发类型 |
| EXTI\_CR1 | 0x04 | GPIO 0[15:8] 中断触发类型 |
| EXTI\_IF | 0x08 | GPIO 中断标志 |
| LCKR\_PRT | 0x0C | GPIO 保护锁定配置 |
| WAKE\_POL | 0x10 | GPIO 唤醒信号极性 |
| WAKE\_EN | 0x14 | GPIO 唤醒使能 |

# 8.2.2 GPIOxPIE

地址分别是：0x4001\_2000，0x4001\_2040，0x4001\_2080，0x4001\_20C0

复位值：0x0

表8-3GPIOx输入使能寄存器GPIOxPIE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PIE15 | PIE14 | PIE13 | PIE12 | PIE11 | PIE10 | PIE9 | PIE8 | PIE7 | PIE6 | PIE5 | PIE4 | PIE3 | PIE2 | PIE1 | PIE0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [S1:16] |  | 未使用 |
| [15] | PIE15 | GPIO x[15] / Px[15] 输入使能 |
| [14] | PIE14 | GPIO x[14] / Px[14] 输入使能 |
| [13] | PIE13 | GPIO x[13] / Px[13] 输入使能 |
| [12] | PIE12 | GPIO x[12] / Px[12] 输入使能 |
| [11] | PIE11 | GPIO x[11] / Px[11] 输入使能 |
| [10] | PIE10 | GPIO x[10] / Px[10] 输入使能 |
| [9] | PIE9 | GPIO x[9] / Px[9] 输入使能 |
| [8] | PIE8 | GPIO x[8] / Px[8] 输入使能 |
| [7] | PIE7 | GPIO x[7] / Px[7] 输入使能 |
| [6] | PIE6 | GPIO x[6] / Px[6] 输入使能 |
| [5] | PIE5 | GPIO x[5] / Px[5] 输入使能 |
| [4] | PIE4 | GPIO x[4] / Px[4] 输入使能 |
| [3] | PIE3 | GPIO x[3] / Px[3] 输入使能 |
| [2] | PIE2 | GPIO x[2] / Px[2] 输入使能 |
| [1] | PIE1 | GPIO x[1] / Px[1] 输入使能 |
| [0] | PIE0 | GPIO x[0] / Px[0] 输入使能 |

# 8.2.3 GPIOxPOE

地址分别是：0x4001\_2004，0x4001\_2044，0x4001\_2084，0x4001\_20C4

复位值：0x0

表8-4GPIOx输出使能寄存器GPIOxPOE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| POE1 | POE1 | POE1 | POE1 | POE1 | POE1 | POE | POE | POE | POE | POE | POE | POE | POE | POE | POE |
| 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | POE15 | GPIO x[15] / Px[15] 输出使能 |
| [14] | POE14 | GPIO x[14] / Px[14] 输出使能 |

|  |  |  |
| --- | --- | --- |
| [13] | POE13 | GPIOx[13] / Px[13] 输出使能 |
| [12] | POE12 | GPIOx[12] / Px[12] 输出使能 |
| [11] | POE11 | GPIOx[11] / Px[11] 输出使能 |
| [10] | POE10 | GPIOx[10] / Px[10] 输出使能 |
| [9] | POE9 | GPIOx[9] / Px[9] 输出使能 |
| [8] | POE8 | GPIOx[8] / Px[8] 输出使能 |
| [7] | POE7 | GPIOx[7] / Px[7] 输出使能 |
| [6] | POE6 | GPIOx[6] / Px[6] 输出使能 |
| [5] | POE5 | GPIOx[5] / Px[5] 输出使能 |
| [4] | POE4 | GPIOx[4] / Px[4] 输出使能 |
| [3] | POE3 | GPIOx[3] / Px[3] 输出使能 |
| [2] | POE2 | GPIOx[2] / Px[2] 输出使能 |
| [1] | POE1 | GPIOx[1] / Px[1] 输出使能 |
| [0] | POE0 | GPIOx[0] / Px[0] 输出使能 |

# 8.2.4 GPIOx\_PDI

地址分别是：0x4001\_2008，0x4001\_2048，0x4001\_2088，0x4001\_20C8

复位值：0x0

表8-5GPIOx输入数据寄存器GPIOx\_PDI

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PDI | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PDI | GPIOx输入数据 |

当GPIOxPIE 时，GPIOx\_PDI读回为0。

# 8.2.5 GPIOx\_PDO

地址分别是：0x4001\_200C，0x4001\_204C，0x4001\_208C，0x4001\_20CC

复位值：0x0

表8-6GPIOx输出数据寄存器GPIOx\_PDO

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PDO | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PDO | GPIOx 输出数据 |

# 8.2.6 GPIOx\_PUE

地址分别是：0x4001\_2010，0x4001\_2050，0x4001- 2090，0x4001\_20D0

复位值：0x0

表8-7GPIOx上拉使能寄存器GPIOx\_PUE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PUE15 | PUE14 | PUE13 | PUE12 | PUE11 | PUE10 | PUE9 | PUE8 | PUE7 | PUE6 | PUE5 | PUE4 | PUE3 | PUE2 | PUE1 | PUE0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | PUE15 | GPIOx[15] / Px[15] 上拉使能 |
| [14] | PUE14 | GPIOx[14] / Px[14] 上拉使能 |
| [13] | PUE13 | GPIOx[13] / Px[13] 上拉使能 |
| [12] | PUE12 | GPIOx[12] / Px[12] 上拉使能 |
| [11] | PUE11 | GPIOx[11] / Px[11] 上拉使能 |
| [10] | PUE10 | GPIOx[10] / Px[10] 上拉使能 |
| [9] | PUE9 | GPIOx[9] / Px[9] 上拉使能 |
| [8] | PUE8 | GPIOx[8] / Px[8] 上拉使能 |
| [7] | PUE7 | GPIOx[7] / Px[7] 上拉使能 |
| [6] | PUE6 | GPIOx[6] / Px[6] 上拉使能 |
| [5] | PUE5 | GPIOx[5] / Px[5] 上拉使能 |
| [4] | PUE4 | GPIOx[4] / Px[4] 上拉使能 |
| [3] | PUE3 | GPIOx[3] / Px[3] 上拉使能 |
| [2] | PUE2 | GPIOx[2] / Px[2] 上拉使能 |
| [1] | PUE1 | GPIOx[1] / Px[1] 上拉使能 |
| [0] | PUE0 | GPIOx[0] / Px[0] 上拉使能 |

# 8.2.7 GPIOx\_PODE

地址分别是：0x4001\_2018，0x4001\_2058，0x4001\_2098，0x4001\_20D8

复位值：0x0

表8-8GPIOx开漏使能寄存器GPIOxPODE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PODE15 | PODE14 | PODE13 | PODE12 | PODE11 | PODE10 | PODE9 | PODE8 | PODE7 | PODE6 | PODE5 | PODE4 | PODE3 | PODE2 | PODE1 | PODE0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | PODE15 | GPIO x[15] / Px[15] 开漏使能 |
| [14] | PODE14 | GPIO x[14] / Px[14] 开漏使能 |
| [13] | PODE13 | GPIO x[13] / Px[13] 开漏使能 |
| [12] | PODE12 | GPIO x[12] / Px[12] 开漏使能 |
| [11] | PODE11 | GPIO x[11] / Px[11] 开漏使能 |
| [10] | PODE10 | GPIO x[10] / Px[10] 开漏使能 |
| [9] | PODE9 | GPIO x[9] / Px[9] 开漏使能 |
| [8] | PODE8 | GPIO x[8] / Px[8] 开漏使能 |
| [7] | PODE7 | GPIO x[7] / Px[7] 开漏使能 |
| [6] | PODE6 | GPIO x[6] / Px[6] 开漏使能 |
| [5] | PODE5 | GPIO x[5] / Px[5] 开漏使能 |
| [4] | PODE4 | GPIO x[4] / Px[4] 开漏使能 |
| [3] | PODE3 | GPIO x[3] / Px[3] 开漏使能 |
| [2] | PODE2 | GPIO x[2] / Px[2] 开漏使能 |
| [1] | PODE1 | GPIO x[1] / Px[1] 开漏使能 |
| [0] | PODE0 | GPIO x[0] / Px[0] 开漏使能 |

# 8.2.8 GPIOx\_LCKR

地址分别是：0x4001\_201C，0x4001\_205C，0x4001\_209C，0x4001\_20DC

复位值：0x0

表8-9GPIOx配置锁定寄存器GPIOx\_LCKR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLCKR15 | PLCKR14 | PLCKR13 | PLCKR12 | PLCKR11 | PLCKR10 | PLCKR9 | PLCKR8 | PLCKR7 | PLCKR6 | PLCKR5 | PLCKR4 | PLCKR3 | PLCKR2 | PLCKR1 | PLCKR0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | PLCKR15 | GPIO x[15] / Px[15] 配置锁定 |
| [14] | PLCKR14 | GPIO x[14] / Px[14] 配置锁定 |
| [13] | PLCKR13 | GPIO x[13] / Px[13] 配置锁定 |
| [12] | PLCKR12 | GPIO x[12] / Px[12] 配置锁定 |
| [11] | PLCKR11 | GPIO x[11] / Px[11] 配置锁定 |
| [10] | PLCKR10 | GPIO x[10] / Px[10] 配置锁定 |
| [9] | PLCKR9 | GPIO x[9] / Px[9] 配置锁定 |
| [8] | PLCKR8 | GPIO x[8] / Px[8] 配置锁定 |
| [7] | PLCKR7 | GPIO x[7] / Px[7] 配置锁定 |
| [6] | PLCKR6 | GPIO x[6] / Px[6] 配置锁定 |
| [5] | PLCKR5 | GPIO x[5] / Px[5] 配置锁定 |
| [4] | PLCKR4 | GPIO x[4] / Px[4] 配置锁定 |
| [3] | PLCKR3 | GPIO x[3] / Px[3] 配置锁定 |
| [2] | PLCKR2 | GPIO x[2] / Px[2] 配置锁定 |
| [1] | PLCKR1 | GPIO x[1] / Px[1] 配置锁定 |
| [0] | PLCKR0 | GPIO x[0] / Px[0] 配置锁定 |

配置保护，高有效；有效时 GPIO 输入/输出/上下拉/开漏/功能选择不能被修改；需要注意，只有在 LCKR\_PRT 写保护打开时才能改写 LCKR。

# 8.2.9 GPIOx\_F3210

地址分别是：0x4001\_2020，0x4001\_2060，0x4001\_20A0，0x4001\_20E0

复位值：0x0

表8-10GPIOx功能选择寄存器GPIOxF3210

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F3 |  |  | F2 |  |  |  | F1 |  |  | F0 |  |  |  |  |  |
| RW |  |  | RW |  |  |  | RW |  |  | RW |  |  |  |  |  |
| 0 |  |  | 0 |  |  |  | 0 |  |  | 0 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | F3 | GPIO x[3] / Px[3] 功能选择 |
| [11:8] | F2 | GPIO x[2] / Px[2] 功能选择 |
| [7:4] | F1 | GPIO x[1] / Px[1] 功能选择 |
| [3:0] | F0 | GPIO x[0] / Px[0] 功能选择 |

GPIO 引脚功能复用如表 8- 11 所示。

表8-11GPIO功能复用

|  |  |  |
| --- | --- | --- |
| GPIOx\_Fxxxx 配置值 | 第二功能代号 | 功能 |
| 0x0 | AF0 | 模拟功能 |
| 0x1 | AF1 | SYS\_AF，比较器及时钟等数字信号输出功能 |
| 0x2 | AF2 | HALL |
| 0x3 | AF3 | MCPWM |
| 0x4 | AF4 | UART |
| 0x5 | AF5 | SPI |
| 0x6 | AF6 | HC |
| 0x7 | AF7 | Timer0/Time1 |
| 0x8 | AF8 | Timer2/ TimeC3/QEP0/QEP1 |
| 0x9 | AF9 | ADC trigger debug |
| 0xA | AF10 | CAN |
| 0xB | AF11 | SIF |

# 8.2.10 GPIOx\_F7654

地址分别是：0x4001\_2024，0x4001\_2064，0x4001\_20A4，0x4001\_20E4

复位值：0x0

表8-12GPIOx功能选择寄存器GPIOxF7654

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F7 |  |  |  | F6 |  |  | F5 |  |  | F4 |  |  |  |  |  |
| RW |  |  |  | RW |  |  | RW |  |  | RW |  |  |  |  |  |
| 0 |  |  |  | 0 |  |  | 0 |  |  | 0 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | F7 | GPIO x[7] / Px[7] 功能选择 |
| [11:8] | F6 | GPIO x[6] / Px[6] 功能选择 |
| [7:4] | F5 | GPIO x[5] / Px[5] 功能选择 |
| [3:0] | F4 | GPIO x[4] / Px[4] 功能选择 |

# 8.2.11 GPIOx\_FBA98

地址分别是：0x4001\_2028，0x4001\_2068，0x4001\_20A8，0x4001\_20E8

复位值：0x0

表8-13GPIOx功能选择寄存器GPIOxFBA98

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F11 |  |  |  | F10 |  |  | F9 |  |  | F8 |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| RW | RW | RW | RW |
| 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | F11 | GPIO x[11] / Px[11] 功能选择 |
| [11:8] | F10 | GPIO x[10] / Px[10] 功能选择 |
| [7:4] | F9 | GPIO x[9] / Px[9] 功能选择 |
| [3:0] | F8 | GPIO x[8] / Px[8] 功能选择 |

# 8.2.12 GPIOxFFEDC

地址分别是：0x4001\_202C，0x4001\_206C，0x4001\_20AC，0x4001\_20EC

复位值：0x0

表8-14GPIOx功能选择寄存器GPIOxFFEDC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F15 |  |  | F14 |  |  |  | F13 |  |  |  |  | F12 |  |  |  |
| RW |  |  | RW |  |  |  | RW |  |  |  |  | RW |  |  |  |
| 0 |  |  | 0 |  |  |  | 0 |  |  |  |  | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | F15 | GPIO x[15] / Px[15] 功能选择 |
| [11:8] | F14 | GPIO x[14] / Px[14] 功能选择 |
| [7:4] | F13 | GPIO x[13] / Px[13] 功能选择 |
| [3:0] | F12 | GPIO x[12] / Px[12] 功能选择 |

GPIO的功能复用详细列表请见对应产品DATASHEET管脚分布章节。

# 8.2.13 外部中断、唤醒、锁定保护

P0.0/P0.1/P1.0/P1.1.4个GPIO可以作为系统的唤醒源。P0.15～P0.0共16个GPIO可以用作外部中断源输入。唤醒功能和外部中断功能使用的是IO的GPIO功能。GPIO第二功能可以配置为0。

# 8.2.13.1 EXTLCR0

地址：0x4001\_2100

复位值：0x0

表8-15外部中断配置寄存器EXTI\_CR0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T7 | T6 |  | T5 |  | T4 |  | T3 |  | T2 |  | T1 |  | T0 |  |  |
| RW | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  |  |
| 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:14] | T7 | GPIO 0[7]/P0[7]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [13:12] | T6 | GPIO 0[6]/P0[6]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [11:10] | T5 | GPIO 0[5]/P0[5]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [9:8] | T4 | GPIO 0[4]/P0[4]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [7:6] | T3 | GPIO 0[3]/P0[3]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [5:4] | T2 | GPIO 0[2]/P0[2]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [3:2] | T1 | GPIO 0[1]/P0[1]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |

|  |  |  |
| --- | --- | --- |
| [1:0] | T0 | GPIO 0[0]/P0[0]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |

# 8.2.13.2 EXTI\_CR1

地址：0x4001\_2104

复位值：0x0

表8-16外部中断配置寄存器EXTI\_CR1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T15 | T14 | T13 | T12 | T11 | T10 | T9 | T8 |  |  |  |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:14] | T15 | GPIO 0[15]/P0[15]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [13:12] | T14 | GPIO 0[14]/P0[14]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [11:10] | T13 | GPIO 0[13]/P0[13]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [9:8] | T12 | GPIO 0[12]/P0[12]外部中断触发类型选择 00:不触发 01:下降沿触发 10:上升沿触发 11:上升沿、下降沿均触发 |
| [7:6] | T11 | GPIO 0[11]/P0[11]外部中断触发类型选择 00:不触发 01:下降沿触发 |

|  |  |  |
| --- | --- | --- |
|  |  | 10：上升沿触发 11：上升沿、下降沿均触发 |
| [5:4] | T10 | GPIO 0[10]/P0[10]外部中断触发类型选择 00：不触发 01：下降沿触发 10：上升沿触发 11：上升沿、下降沿均触发 |
| [3:2] | T9 | GPIO 0[9]/P0[9]外部中断触发类型选择 00：不触发 01：下降沿触发 10：上升沿触发 11：上升沿、下降沿均触发 |
| [1:0] | T8 | GPIO 0[8]/P0[8]外部中断触发类型选择 00：不触发 01：下降沿触发 10：上升沿触发 11：上升沿、下降沿均触发 |

# 8.2.13.3 EXTI\_IF

地址：0x4001\_2108

复位值：0x0

表8-17外部中断标志寄存器EXTI\_IF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IF15 | IF14 | IF13 | IF12 | IF11 | IF10 | IF9 | IF8 | IF7 | IF6 | IF5 | IF4 | IF3 | IF2 | IF1 | IF0 |
| RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | IF15 | GPIO 0[15] / P0[15] 外部中断标志 中断标志高有效，写 1 清零 |
| [14] | IF14 | GPIO 0[14] / P0[14] 外部中断标志 中断标志高有效，写 1 清零 |
| [13] | IF13 | GPIO 0[13] / P0[13] 外部中断标志 中断标志高有效，写 1 清零 |
| [12] | IF12 | GPIO 0[12] / P0[12] 外部中断标志 中断标志高有效，写 1 清零 |
| [11] | IF11 | GPIO 0[11] / P0[11] 外部中断标志 |

|  |  |  |
| --- | --- | --- |
|  |  | 中断标志高有效，写1清零 |
| [10] | IF10 | GPIO 0[10] / P0[10] 外部中断标志 中断标志高有效，写1清零 |
| [9] | IF9 | GPIO 0[9] / P0[9] 外部中断标志 中断标志高有效，写1清零 |
| [8] | IF8 | GPIO 0[8] / P0[8] 外部中断标志 中断标志高有效，写1清零 |
| [7] | IF7 | GPIO 0[7] / P0[7] 外部中断标志 中断标志高有效，写1清零 |
| [6] | IF6 | GPIO 0[6] / P0[6] 外部中断标志 中断标志高有效，写1清零 |
| [5] | IF5 | GPIO 0[5] / P0[5] 外部中断标志 中断标志高有效，写1清零 |
| [4] | IF4 | GPIO 0[4] / P0[4] 外部中断标志 中断标志高有效，写1清零 |
| [3] | IF3 | GPIO 0[3] / P0[3] 外部中断标志 中断标志高有效，写1清零 |
| [2] | IF2 | GPIO 0[2] / P0[2] 外部中断标志 中断标志高有效，写1清零 |
| [1] | IF1 | GPIO 0[1] / P0[1] 外部中断标志 中断标志高有效，写1清零 |
| [0] | IF0 | GPIO 0[0] / P0[0] 外部中断标志 中断标志高有效，写1清零 |

# 8.2.13.4 LCKR\_PRT

地址：0x4001\_210C

复位值：0x0

表8-18锁定保护寄存器LCKR\_PRT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRT | | | | | | | | | | | | | | |  |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PRT | 配置锁定写保护；写入 0x5AC4 关闭写保护，然后才能修改 GPIO\_LCKR；写入任意其他数据开启写保护；B[0]状态指示当前写保护与否，高电平表示处于写保护，低电平表示处于非写保护。 |

# 8.2.13.5 WAKE\_POL

地址：0x4001\_2110

复位值：0x0

表8-19外部唤醒源极性配置寄存器WAKE\_POL

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 未使用 |
| [3] | GPIO1\_1\_POL | GPIO 1[1] / P1[1]外部唤醒触发电平选择 1：高电平；0：低电平 |
| [2] | GPIO1\_0\_POL | GPIO 1[0] / P1[0]外部唤醒触发电平选择 1：高电平；0：低电平 |
| [1] | GPIO0\_1\_POL | GPIO 0[1] / P0[1]外部唤醒触发电平选择 1：高电平；0：低电平 |
| [0] | GPIO0\_0\_POL | GPIO 0[0] / P0[0]外部唤醒触发电平选择 1：高电平；0：低电平 |

# 8.2.13.6 WAKE\_EN

地址：0x4001\_2114

复位值：0x0

表8-20外部唤醒源使能寄存器WAKE\_EN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | GPIO1\_1\_WKEN | GPIO1\_0\_WKEN | GPIO0\_1\_WKEN | GPIO0\_0\_WKEN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | RW | RW | RW | RW |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 未使用 |
| [3] | GPIO1\_1\_WKEN | GPIO 1[1] / P1[1] 外部唤醒使能 1：使能；0：禁用。 |
| [2] | GPIO1\_0\_WKEN | GPIO 1[0] / P1[0] 外部唤醒使能 1：使能；0：禁用。 |
| [1] | GPIO0\_1\_WKEN | GPIO 0[1] / P0[1] 外部唤醒使能 1：使能；0：禁用。 |

|  |  |  |
| --- | --- | --- |
| [0] | GPIO0\_0\_WKEN | GPIO 0[0] / P0[0] 外部唤醒使能 1：使能；0：禁用。 |

# 8.3 实现说明

# 8.3.1 上拉实现

LKS32MC08x系列芯片，通过内部模拟电路进行上拉功能实现。所有GPIO都有上拉控制寄存器PUE，但不是所有GPIO都有上拉电路。配备上拉功能的GPIO如下：

表8-21GPIO上拉资源分布表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| P0 |  |  |  |  |  |  |  | √ | √ |  |  |  |  |  | √ |  |
| P1 |  |  |  |  | √ | √ |  |  |  |  |  | √ |  |  | √ |  |
| P2 |  |  |  |  |  | √ |  | √ | √ | √ | √ |  |  | √ |  |  |
| P3 |  |  |  |  |  |  | √ |  |  |  |  |  |  |  |  |  |

# 8.4 应用指南

# 8.4.1 配置锁定

芯片提供对GPIO配置的保护功能。当LCKR\_PRT写保护使能时，3组GPIO的GPIO\_LCKR不能修改，GPIO\_PIE/GPIO\_PIOE/GPIO\_PUE/GPIO\_PDE/GPIO\_ODE/GPIO\_F3210/GPIO\_F7654/GPIO\_FBA98/GPIO\_FFEDC不能修改；

若需要修改GPIO配置，应先解除LCKR\_PRT写保护，然后将对应GPIO的GPIO\_LCKR写0，解除配置锁定，然后修改GPIO配置。

示例如下：

GPIO0\_PIE GPIO1\_PIE GPIO2\_PIE // // lock specific gpio GPIO0\_LCKR //lock gpio0 here GPIO1\_LCKR //lock gpio1 here GPIO2\_LCKR //lock gpio2 here

// modify to test if gpio config is lockedGPI00\_PIE = 0x3333;GPI01\_PIE = 0x0000;GPI02\_PIE = 0x0000;

//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

// read gpio config to flag PASS or FAIL

if(GPIO0\_PIE != 0x3233)FAIL;if(GPIO1\_PIE != 0x7777)FAIL;if(GPIO2\_PIE != 0x8000)FAIL;

LCKR\_PRT = 0x0000; // write any value other than 0x5AC4 to enable lock protectGPI00\_LCKR = 0x0000;GPI01\_LCKR = 0x0000;GPI02\_LCKR = 0x0000;

if(GPIO0\_LCKR != 0x0100)FAIL;if(GPIO1\_LCKR != 0xFFFF)FAIL;if(GPIO2\_LCKR != 0x8000)FAIL;

LCKR\_PRT = 0x5AC4; // disable protectGPI00\_LCKR = 0x0000;GPI01\_LCKR = 0x0000;GPI02\_LCKR = 0x0000;

if(GPIO0\_LCKR != 0x0000)FAIL;if(GPIO1\_LCKR != 0x0000)FAIL;if(GPIO2\_LCKR != 0x0000)FAIL;

while(i- - ); PASS;

# 8.4.2 外部中断

示例如下：

GPI00 PIE 0x0080; //使能P0[7]输入NVIC\_EnableIRQ(GPIO\_IRQn);//使能GPIO中断\_enable\_irqO; //使能中断 while(i- - );//P0[7]IO上外接方波信号EXTI ；//使能p0[7]上升沿触发，产生外部中断while(irq\_flag ；//外部信号翻转两次，产生两次中断，irq\_flag在GPIO中断处理程序中递增两次EXTI ；//使能p0[7]下降沿触发，产生外部中断while(irq\_flag ；EXTI ；//同时使能P0[7]上升沿、下降沿触发，产生外部中断while(irq\_flag ；EXTI ；//同时禁用P[7]上下沿触发，将无法产生外部中断 while(i- - );if(irq\_flag )FAIL; while(i- - );PASS;

# 8.4.3 使用GPIO的模拟功能

将GPIO的IE和OE关闭，即可使用模拟功能。此时，PAD通过内部电阻直接与模拟模块相连。

# 9 CRC

# 9.1 概述

CRC即循环冗余校验码（CyclicRedundancyCheck）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

利用CRC进行检错的过程可简单描述为：在发送端根据要传送的K位二进制码序列，以一定的规则产生一个校验用的R位监督码(CRC码)，附在原始信息后边，构成一个新的二进制码序列数共 位，然后发送出去。在接收端，根据信息码和CRC码之间所遵循的规则进行检验，以确定传送中是否出错。这个规则，在差错控制理论中称为”生成多项式”。

# 9.2 基本原理

循环冗余校验码（CRC）的基本原理是：在K位信息码后再拼接R位的校验码，整个编码长度为N位，因此，这种编码也叫（N，K）码。对于一个给定的（N，K）码，可以证明存在一个最高次幂为 的多项式 。根据 可以生成K位信息的校验码，而 叫做这个CRC码的生成多项式。校验码的具体生成过程为：假设要发送的信息用多项式 表示，将 左移R位（可表示成 ，这样 的右边就会空出R位，这就是校验码的位置。用 除以生成多项式 得到的余数就是校验码。

任意一个由二进制位串组成的代码都可以和一个系数仅为 和’1’取值的多项式一一对应。例如：代码1010111对应的多项式为 ，而多项式为 对应的代码101111。

# 9.3 基本概念

# 9.3.1 对应关系

多项式和二进制数有直接对应关系：X的最高幂次对应二进制数的最高位，以下各位对应多项式的各幂次，有此幂次项对应1，无此幂次项对应0。可以看出：X的最高幂次为R，转换成对应的二进制数有 位。

多项式包括生成多项式G(X)和信息多项式C(X)。

如生成多项式为 ，可转换为二进制数码11011。

而发送信息为101111，可转换为数据多项式为 0

# 9.3.2 生成多项式

生成多项式是接受方和发送方的一个约定，也就是一个二进制数，在整个传输过程中，这个数始终保持不变。

在发送方，利用生成多项式对信息多项式做模2除生成校验码。在接收方利用生成多项式对收到的编码多项式做模2除检测和确定错误位置。

应满足以下条件：

A、生成多项式的最高位和最低位必须为1。B、当被传送信息（CRC码）任何一位发生错误时，被生成多项式做除后应该使余数不为0。C、不同位发生错误时，应该使余数不同。D、对余数继续做除，应使余数循环。

# 9.3.3 校验码位数

CRC校验码位数 生成多项式位数- 1。注意有些生成多项式的简记式中将生成多项式的最高位1省略了。

# 9.3.4 生成步骤

1、将X的最高次幂为R的生成多项式G(X)转换成对应的 位二进制数。

2、将信息码左移R位，相当于对应的信息多项式

3、用生成多项式（二进制数）对信息码做除，得到R位的余数(注意：这里的二进制做除法得到的余数其实是模2除法得到的余数，并不等于其对应十进制数做除法得到的余数。)。

4、将余数拼到信息码左移后空出的位置，得到完整的CRC码。

[例]假设使用的生成多项式是 。4位的原始报文为1010，求编码后的报文。解：

1、将生成多项式 转换成对应的二进制除数1011。

2、此题生成多项式有4位（ ）(注意：4位的生成多项式计算所得的校验码为3位，R为校验码位数)，要把原始报文C(X)左移3（R）位变成1010000

3、用生成多项式对应的二进制数对左移3位后的原始报文进行模2除（高位对齐)，相当于按位异或：

1010000

1011

0001000

0001011

0000011

得到的余位011，所以最终编码为：1010011

POL=0x13，data=0x77

011101110000000

10010011

01111101000000

10010011

0110100100000

10010011

010000010000

10010011

00010001000

10010011

00011011

# 9.4 寄存器

# 9.4.1 地址分配

CRC的基地址是0x4001\_2400，寄存器列表如下：

表9-1CRC寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| CRC\_DR | 0x00 | CRC 数据（输入信息码/输出编码）寄存器 |
| CRC\_CR | 0x04 | CRC 控制寄存器 |
| CRC\_INIT | 0x08 | CRC 初始码寄存器 |
| CRC\_POL | 0x0C | CRC 生成多项式对应的二进制码寄存器 |

# 9.4.2 寄存器描述

9.4.2.1 CRC\_DR CRC 信息码寄存器

地址：0x4001\_2400复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表9-2 CRC 数据寄存器 CRC\_DR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | DR | 存放待编码的信息码和经 CRC 校验后的编码 |

CRC\_DR 寄存器既用于放入待校验数据，也用于返回校验结果。写入 CRC\_DR 寄存器即触发一次 CRC 计算。待编码数据应在 CR 等寄存器配置完成后最后写入，以触发 CRC 计算开始。

# 9.4.2.2 CRC\_CR CRC 控制寄存器

地址：0x4001\_2404

复位值：0x0

表9-3CRC控制寄存器CRC\_CR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | REV\_OUT\_TYPE |  |  | REV\_IN\_TYPE |  |  | POLY\_SIZE |  |  |  | RESET |  |  |  |
| RW | RW | RW | WO |  |  |  |
| 0 | 0 | 0 | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |
| [12] | REV\_OUT\_TYPE | 是否将 CRC 校验后的编码反转后输出，即 b[31]=b[0], b[30]=b[1],…[b0]=b[31] |
| [11:10] |  | 未使用 |
| [9:8] | REV\_IN\_TYPE | 待编码数据反转类型 00:不反转 |

|  |  |  |
| --- | --- | --- |
|  |  | 01:按字节反转，即 b[31]=b[24], b[30]=b[25], …, b[24]=b[31], …, b[7]=b[0], b[6]=b[1], …, b[0]=b[7] 10:按半字（16bit）反转，即 b[31]=b[16], b[30]=b[17], …, b[16]=b[31], …, b[15]=b[0], b[14]=b[1], …, b[0]=b[15] 11:按字反转，即 b[31]=b[0], b[30]=b[1], …, b[0]=b[31] |
| [7:6] |  | 未使用 |
| [5:4] | POLY\_SIZE | 输出编码（多项式）位宽 00:32bits 01:16bits 10:8bits 11:7bits |
| [3:1] |  | 未使用 |
| [0] | RESET | 与输入信息码进行 CRC 计算的数据来源 0:来自于上一次的计算结果 1:来自于 CRC\_INIT 写入 1 实现 CRC 数据重置并自动清零，读回恒为 0. |

同时需要注意的是，向CRC\_CR\_RESET写入1会将CRC\_INIT寄存器复位为0xFFFFFFFF。

如果需要清除CRC的计算结果，应向CRC\_CR\_RESET写入1，否则后续CRC计算会以之前的计算结果为初值进行。

# 9.4.2.3 CRC\_INIT CRC初始码寄存器

地址：0x4001\_2408

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表9-4 CRC 初始码寄存器 CRC\_INIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0xFFFFFFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| INIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RIW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

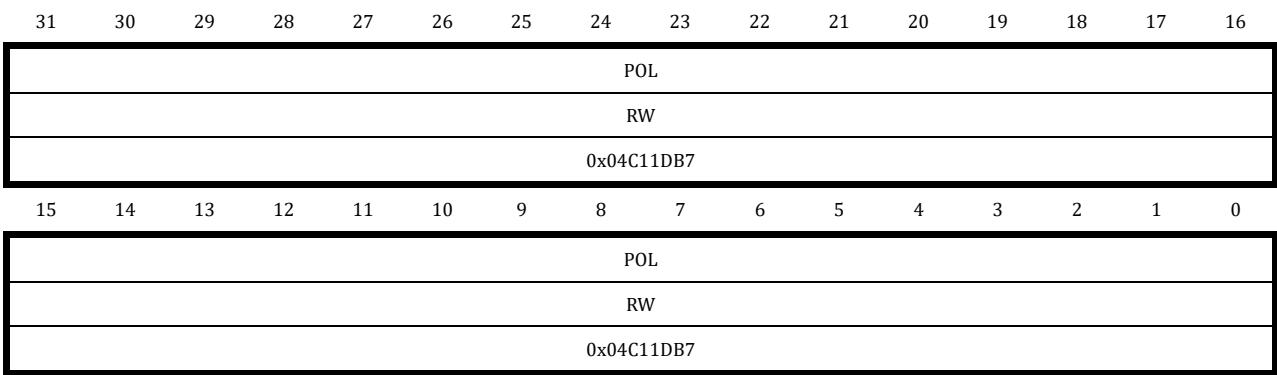
|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | INIT | 存放初始码 |

CRC\_DR与CRC\_INIT相异或后开始进行CRC校验计算。

# 9.4.2.4 CRC\_POL CRC生成码寄存器

地址：0x4001\_240C

复位值：0x0



|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | POL | 存放生成多项式对应的生成码 |

# 10 ADC

# 10.1概述

LKS32MC08x系列芯片集成了1个12bitSARADC，采用双路同步采样设计，可以同一个时刻采样两个通道的信号，每个采样电路对应10路输入通道，共20路。主要有以下特性：

3Msps采样及转换速率，48MHz工作频率。- 最多支持20通道选择。- 支持软件、硬件触发功能。- 可以与MCPWM、UTimer单元联动，触发指示信号ADC\_TRIGGER可通过GPIO送出用于调试- 支持单段、双段、四段自定义采样序列采样，序列次数和通道号可灵活配置。- 支持连续采样模式。- 支持DMA搬移数据。- 支持ADC硬件数值比较功能，可以设置上或下阈值，达到阈值触发ADC比较中断。- 支持左对齐、右对齐模式。

ADC采样的量词含义约定：

1次采样：完成对应的一个通道的模拟信号量到数据信号量的采样转换并存储数字量至ADC\_DATx寄存器；

1段采样：可能包含1次或若干次采样，若干次采样可以是相同的模拟量通道，也可以是不同的模拟量通道。采样开始通常由MCPWM、UTimer或软件进行触发，一个触发信号完成一段采样，采样完成后产生相应的段采样完成中断；以MCPWM触发的四段采样为例，每段采样3次（即完成3个模拟量的采样)，TADC[0]触发ADC开始第一段采样，第一段采样完成后ADC进入等待状态，等待TADC[1]触发事件发生；TADC[1]发生后，触发ADC开始第二段采样；同理，TADC[2]/TADC[3]分别触发第三段和第四段采样。

1轮采样：可能包含1段、2段或4段采样，每段分别由特定触发信号触发；ADC完成一轮采样后回归空闲状态等待下次触发。

# 10.1.1 功能框图

ADC接口包括20个数据寄存器（ADC20次采样各个通道模拟量对应的数字量)，以及若干控制寄存器。

ADC配备双采样电路，同一时刻进行两个通道的采样，然后依次先后完成到数字量的转换。

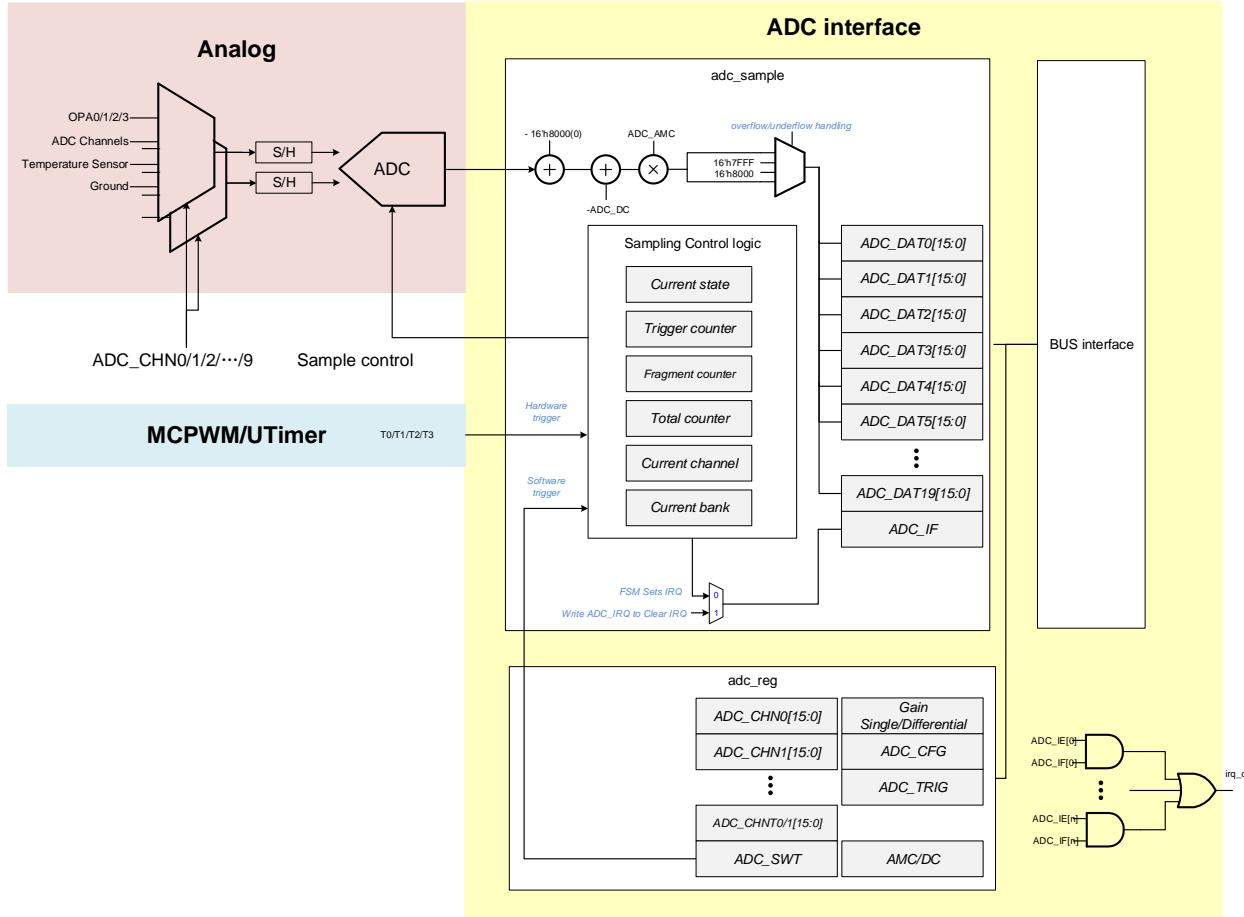
数据寄存器ADC\_DATx用于存储ADC第x个采样得到的数据量。被转换的模拟信号来源由寄存器ADC\_CHNx中的某5bit进行选择（详见9.2.3信号来源寄存器章节)。以ADC\_CHN0为例，位[4:0]选择第0个采样的模拟通道号，通道号CH0~CH19任选，若ADC\_CHN0[4:0]=0,ADC\_CHN0[12:8]=3，

则第0个采样的模拟量对应通道CH0，第1个采样的模拟量对应通道CH3，以此类推。注意，由于双采样电路同步工作，因此第0个采样和第1个采样实际是同时完成的，即采样电路一次就消费ADC\_CHNx选择的两个模拟通道。

分段采样次数寄存器ADC\_CHNTO/1控制每轮采样的个数，1表示1个采样，2表示2个采样，，12表示12个采样，，20表示20个采样。双采样电路一次完成两个采样，如果配置的采样个数是奇数，则最后一次仅完成一个采样的模拟信号的转换。

控制逻辑根据触发配置寄存器ADC\_TRIG选择来自MCPWM或通用定时器UTimer的触发信号启动一轮采样或者软件触发启动。MCPWM/UTimer会送出定时触发信号TADC[0]/TADC[1]/TADC[2]/TADC[3]，可选择TADC[0]/TADC[1]/TADC[2]/TADC[3]作为触发信号。

一段转换（一段内的所有通道采样转换完毕)完成，触发ADC转换完成中断。多段触发模式下，每一段转换完成可触发产生一个转换完成中断。

  
图10-1ADC采集模块功能框图

用户可以灵活配置采样顺序、以及采样信号来源，甚至实现对单个信号多次采样。同时每个通道的ADC增益也可以通过寄存器配置（两档增益可选)。控制寄存器使得用户可以配置采样个数，提高采样频率或降低采样功耗。

芯片上电的默认状态下，ADC模块是关闭的。通过将SYS\_AFE\_REG5. ADCPDN设置为1来开启ADC,ADC开启前，需要先开启BGP、4MRC时钟和PLL模块，并通过配置SYS\_AFE\_REG7. ADCCLKSEL选择ADC工作频率。

ADCPDN的说明见模拟寄存器5.2.9SYSAFE REG5

# ADCCLKSEL 的说明见模拟寄存器5.2.11SYSAFE\_REG7

ADC完成一次转换需要16个ADC时钟周期，其中13个为转换周期，3个为采样周期。在ADC时钟设为48MHz时，转换速率是3Msps。

# 10.1.2 ADC触发方式

支持单段触发、两段触发、四段触发完成采样单段触发可以设置触发事件发生次数，当触发事件发生一定次数后才开始采样两段触发的触发源只能为MCPWM/UTimer的定时信号 ，或两次软件触发四段触发的触发源只能为MCPWM/UTimer的定时信号 ，或四次软件触发每段触发完成均可配置产生中断触发指示信号可以通过GPIO送出用于调试

表10-1ADC分段触发方式触发源配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分段触发方式 | 当前段 | 可能的触发源 | | |
|  |  | MCPWM | UTIMER | 软件触发 |
| 单段触发 | - | N次T0 或N次T1 或N次T2 或N次T3 | N次T0 或N次T1 或N次T2 或N次T3 | 一次软件触发 |
| 两段触发 | 第一段 | T0 | T0 | 软件触发 |
| 第二段 | T1 | T1 | 软件触发 |
| 四段触发 | 第一段 | T0 | T0 | 软件触发 |
| 第二段 | T1 | T1 | 软件触发 |
| 第三段 | T2 | T2 | 软件触发 |
| 第四段 | T3 | T3 | 软件触发 |

其中N=ADC\_TRIG.SINGLE\_TNCT。可以同时开启MCPWM和UTIMER的触发使能，但由于时序控制的原因，通常不进行这样的配置。

各段采样的个数由ADC\_CHNTO/1进行控制。（1表示1个采样，2表示2个采样，，12表示12个采样)。

表10-2四段触发分段采样通道数示例

|  |  |  |  |
| --- | --- | --- | --- |
| 第n段 | ADC\_CHNTO/1 | 寄存器值 | 当前段采样个数 |
| 1 | ADC\_CHNTO[4:0] | 4 | 4 |
| 2 | ADC\_CHNTO[12:8] | 1 | 1 |
| 3 | ADC\_CHNT1[4:0] | 6 | 6 |
| 4 | ADC\_CHNT1[12:8] | 1 | 1 |

# 10.1.3 ADC通道选择

每个ADC模块有10个通道信号来源寄存器，控制采样序列 的信号选择。ADC\_CHN0控

制采样 ，ADC\_CHN1控制采样 ，…，ADC\_CHN9控制序列 。每个序列选择范围都是 ，对应通道 ，也就是可以对某一个通道进行多次采样。每一个采样对应一个结果寄存器，转换结束后，可以直接到对应结果寄存器中获取到ADC采样结果。

表10-3ADC通道选择与寄存器配置对应关系

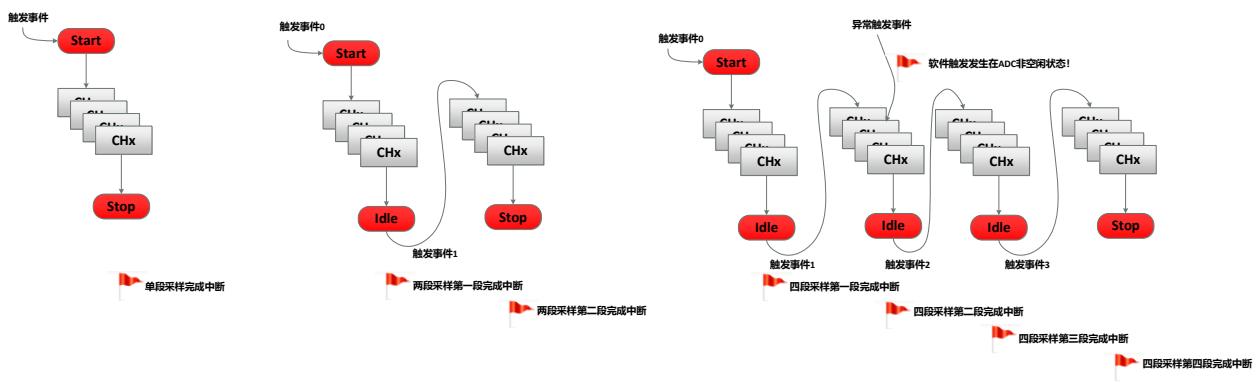
|  |  |  |
| --- | --- | --- |
| ADC 采样序列 | 对应的采样数据寄存器 | 对应信号来源寄存器 |
| 第 0 个采样 | ADC\_DAT0 | ADC\_CHN0[4:0] |
| 第 1 个采样 | ADC\_DAT1 | ADC\_CHN0[12:8] |
| 第 2 个采样 | ADC\_DAT2 | ADC\_CHN1[4:0] |
| 第 3 个采样 | ADC\_DAT3 | ADC\_CHN1[12:8] |
| …… | | |
| 第 18 个采样 | ADC\_DAT18 | ADC\_CHN9[4:0] |
| 第 19 个采样 | ADC\_DAT19 | ADC\_CHN9[12:8] |

# 10.1.4 ADC中断

ADC中断信号在每段采样完成后置高。

软件或硬件触发事件如果在ADC工作时发生，则产生异常触发中断

ADC\_DAT0具备阈值中断，阈值可以设置为上阈值/下阈值，通过ADC\_CFG[1]设置，当ADC\_DAT0超过ADC\_DAT0\_TH时发生中断。

  
图10-2ADC中断产生

# 10.1.5 ADC输出数制

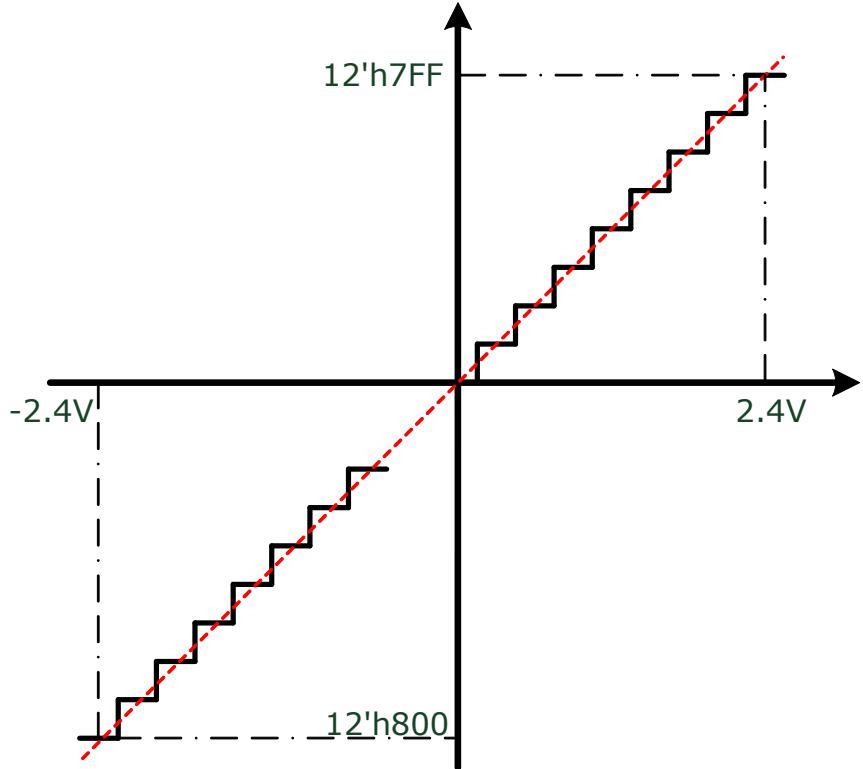
ADC输出数据为12bit补码，输入信号0对应12h’0000\_0000\_0000，以1倍增益配置为例，输入信号- 2.4V对应12h’1000\_0000\_0000，输入信号 对应12h’0111\_1111\_1111。ADC转换后的12bit补码需扩展为16BIT存入16bit位宽的采样数据寄存器，左对齐/右对齐可根据配置寄存器进行设置。以12’h1000\_0000\_1101为例，如果配置为左对齐，右侧补4个0，存入ADCx\_DAT的值为16’h1000\_0000\_1101\_0000；如果配置为右对齐，左侧进行符号扩展，存入ADCx\_DAT的值为16’h1111\_1000\_0000\_1101。推荐统一使用左对齐方式。但由于ADC存在增益校正系数，在增益校正值不为1(0x200)时，左对齐后4位可能不为全0。

需要注意的是，由于存在增益校正和直流偏置校正，ADC最终数据可能会超过12bit有符号数的表示范围，比如在右对齐的模式下，ADC某次转换的数字量可能为0xF745，此时直接进行低12bit

的截取取出 ，会使得负数被作为正数处理，即发生溢出错误。亦或者ADC某次转换的数字量可能为 ，此时直接进行低12bit的截取取出 ，会使得正数被错误地当做负数处理。因此需要将ADC数据作为16bit有符号数进行处理。

表10-4ADC输出数字量数制转换

|  |  |  |
| --- | --- | --- |
| ADC一倍增益输入模拟量数值/V | ADC 2/3 倍增益输入模拟量数值/V | 转为有符号数后的数值 |
| 2.4 | 3.6 | 12'h0111\_1111\_1111 |
| 0 | 0 | 12'h0000\_0000\_0000 |
| -2.4 | -3.6 | 12'h1000\_0000\_0000 |

  
图10-3一倍增益设置下ADC模数转换数制量程

# 10.1.6 ADC基准电压与量程

ADC有2.4V/1.2V两种基准电压配置，通过设置SYS\_AFE\_REG1. GAIN\_REF位实现选择，详见5.2.5。

# 10.1.6.12.4V基准电压模式

芯片工作在5V系统时，推荐使用2.4V的默认基准电压配置，此时ADC正常工作的最低供电电压为3.2V。

ADC有两种增益模式：高增益（1倍）和低增益（2/3倍），针对这两种增益，ADC的量程也相应有所区别。1倍增益模式下，对应最大 的输入信号幅度，2/3倍增益模式下，对应最大 的输入信号幅度。

在ADC采样通道配置为运放的输出信号时（即OPA0~OPA3），应选择合适的运放增益，使得具体应用上的最大信号可被放大到接近 的水平，同时将ADC配置为2/3倍增益。举例来说，相线电流最大100A（正弦波有效值），MOS内阻（假设为MOS内阻采样）为 ，则运放的最大

输入信号幅值为 。此时应该选择运放的放大倍数为4.5倍（放大倍数选择方式详见4.60PA运算放大器），则放大后的信号约为 。

如果因为客观原因，运放的输出信号经放大后，最大信号仍然小于 ，则应将ADC的增益配置为1倍。

在ADC采样通道配置为GPIO复用口输入的信号时，同样根据信号的最大幅度来选择ADC增益。由于IO口的限制，GPIO复用口输入的信号范围只能在 之间。

高低增益选择由ADC\_GAIN0/1增益寄存器进行控制。

# 10.1.6.2 1.2V基准电压模式

芯片工作在3.3V系统时，必须使用1.2V的默认基准电压配置，此时ADC正常工作的最低供电电压为 。通过设置SYS\_AFE\_REG1. GAIN\_REF 将ADC配置为1.2V基准电压模式。这一模式下相比2.4V模式，ADC的有效精度(ENOB)基本不变。

这一模式和2.4V模式的区别在于：

1）对于最常用的ADC2/3倍增益设置（对应 量程），没有任何影响。芯片会自动调整增益为1/3，因此同一个信号的ADC采样值不变，量程也不变。

2）对于ADC1倍增益设置，SYS\_AFE\_REG1. GAIN\_REF 后，使用该性增益设置的ADC通道，其量程变为 ，同一个信号对应的ADC值也会变大一倍

3）对于内部的温度传感器，ADC的增益设置不起作用。ADC采样温度传感器的转换后的数值需要除以2，再使用a/b系数计算温度。温度传感器a/b系数的描述详见4.8TMP温度传感器。由于温度系数a/b是在ADC2.4V基准电压下校正得到的，因此这种工作模式下温度精度会比2.4V基准略低，推荐使用2.4V基准。

# 10.1.7 ADC校正

ADC硬件接口模块可以进行直流偏置校正与增益校正

ADC\_AMC存储的是增益校正系数 ，为10bit无符号定点数，ADC\_AMC[9]为整数部分，ADC\_AMC[8:0]为小数部分。可以表示数值在1附近的定点数。

芯片出厂时已经过工厂标定，标定数据存放在NVR中，芯片上电会自动加载。ADC模块在初始化的时候，需要根据数据左右对齐模式配置DC offset，可以参看芯片供应商提供的库函数。

需要注意的是，ADC有高增益和低增益两挡配置，两种配置对应两套校正参数，每套校正数据分别包含一个DC offset(以下记为DC offset)和一个增益校正值 。同时每套校正参数包含两组DC/AMC分别对应采样电路a/b。高增益对应的校正系数为ADC\_DC\_A1/ADC\_AMC\_A1和ADC\_DC\_B1/ADC\_AMC\_B1，低增益对应的校正系数为ADC\_DC\_A0/ADC\_AMC\_A0和ADC\_DC\_B0/ADC\_AMC\_B0。

记ADC输出的数字量为 ， 对应的真实值为D， 为编码数制的0，则

最终硬件会将进行校正后的D存入相应的采样数据寄存器。ADC接口硬件电路会根据每个通道

的增益配置(ADC\_GAIN0/1)来自动选择 AMPcorrection 与 DCoffset。

# 10.1.8 ADC 配置流程

推荐配置流程：

1. 打开 ADC 模拟开关，选择 ADC 工作频率通过配置寄存器 SYS\_AFE\_REG5\_ADCPDN 可以开启 ADC，ADC 开启前，需要先开启 BGP、4M RC 时钟和 PLL 模块。通过配置寄存器 SYS\_AFE\_REG7\_ADCCLKSEL[5:4] 设置 ADC 工作频率，00 为 48MHz，10 为 12MHz，11 为 24MHz。
2. 配置 ADC 数据输出格式

ADC 的输出格式可配置为左对齐或者右对齐，配置的是 ADC0\_CFG\_DATA\_ALIGN 寄存器，0 为左对齐，1 为右对齐。

1. 配置 ADC 采样模式

ADC 的采样模式可配置为单段、双段、四段采样模式，配置的是 ADC0\_TRIG\_TRG\_MODE[13:12] 寄存器，00 为单段采样模式，01 为两段采样模式，11 为四段采样模式。

1. 配置 ADC 触发事件

ADC 采样的触发事件选择，一共有 8 种采样事件可选择，配置的是 ADC0\_TRIG 寄存器。单段采样模式下可以设置触发一次采样所需的事件数，配置的是 ADC0\_TRIG\_SINGLETCNT[11:8] 寄存器，设置范围是 0~15，0 表示一次事件即触发，15 表示 16 次事件才触发。

1. 配置 ADC 量程

ADC 的基准电压可以通过配置寄存器 SYS\_AFE\_REG1\_GAIN\_REF 进行选择，0 为 2.4V，1 为 1.2V。ADC 的两种增益模式可以通过配置 ADC\_GAIN0/1 增益寄存器进行逐个通道选择，0 为低增益（2/3 倍），1 为高增益（1 倍）。

1. 配置 ADC 通道数，选择采样信号源

配置各段采样模式下，采样的通道个数，配置的是 ADC0\_CHNT0、ADC0\_CHNT1 寄存器，设置范围是 1~20，1 代表一个通道。ADC 的采样信号源配置，通过配置 ADC0\_CHN0、ADC0\_CHN1 等寄存器选择，设置范围是 0~15。

1. 配置 ADC 中断

ADC 一共有七种中断：第一段~第四段采样完成中断、软件触发发生在非空闲状态下的中断、硬件触发发生在非空闲状态下的中断、ADC0\_DTA0 超阈值中断。通过配置 ADC0\_IE 寄存器可以使能以上中断。即使未开启中断，中断事件仍能置位 ADC0\_IE，但不会提出中断请求。

# 10.2 寄存器

# 10.2.1 地址分配

ADC在芯片中的基地址是0x4001\_1400，寄存器列表如下：

表10-5ADC0寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| ADC0\_DAT0 | 0x00 | ADC 第 0 个采样数据 |
| ADC0\_DAT1 | 0x04 | ADC 第 1 个采样数据 |
| ADC0\_DAT2 | 0x08 | ADC 第 2 个采样数据 |
| ADC0\_DAT3 | 0x0C | ADC 第 3 个采样数据 |
| ADC0\_DAT4 | 0x10 | ADC 第 4 个采样数据 |
| ADC0\_DAT5 | 0x14 | ADC 第 5 个采样数据 |
| ADC0\_DAT6 | 0x18 | ADC 第 6 个采样数据 |
| ADC0\_DAT7 | 0x1C | ADC 第 7 个采样数据 |
| ADC0\_DAT8 | 0x20 | ADC 第 8 个采样数据 |
| ADC0\_DAT9 | 0x24 | ADC 第 9 个采样数据 |
| ADC0\_DAT10 | 0x29 | ADC 第 10 个采样数据 |
| ADC0\_DAT11 | 0x2C | ADC 第 11 个采样数据 |
| ADC0\_DAT12 | 0x30 | ADC 第 12 个采样数据 |
| ADC0\_DAT13 | 0x34 | ADC 第 13 个采样数据 |
| ADC0\_DAT14 | 0x38 | ADC 第 14 个采样数据 |
| ADC0\_DAT15 | 0x3C | ADC 第 15 个采样数据 |
| ADC0\_DAT16 | 0x40 | ADC 第 16 个采样数据 |
| ADC0\_DAT17 | 0x44 | ADC 第 7 个采样数据 |
| ADC0\_DAT18 | 0x48 | ADC 第 18 个采样数据 |
| ADC0\_DAT19 | 0x4C | ADC 第 19 个采样数据 |
| ADC0\_CHN0 | 0x50 | ADC 第 0~1 采样信号选择 |
| ADC0\_CHN1 | 0x54 | ADC 第 2~3 采样信号选择 |
| ADC0\_CHN2 | 0x58 | ADC 第 4~5 采样信号选择 |
| ADC0\_CHN3 | 0x5C | ADC 第 6~7 采样信号选择 |
| ADC0\_CHN4 | 0x60 | ADC 第 8~9 采样信号选择 |
| ADC0\_CHN5 | 0x64 | ADC 第 10~11 采样信号选择 |
| ADC0\_CHN6 | 0x68 | ADC 第 12~13 采样信号选择 |
| ADC0\_CHN7 | 0x6C | ADC 第 14~15 采样信号选择 |
| ADC0\_CHN8 | 0x70 | ADC 第 16~17 采样信号选择 |
| ADC0\_CHN9 | 0x74 | ADC 第 18~19 采样信号选择 |
| ADC0\_CHNT0 | 0x78 | ADC 各种触发模式下前两段采样通道数 |
| ADC0\_CHNT1 | 0x7C | ADC 各种触发模式下后两段采样通道数 |
|  | 0x80 | 保留 |
|  | 0x84 | 保留 |
| ADC0\_GAIN0 | 0x88 | ADC 第 0~9 采样保持电路增益控制 |

|  |  |  |
| --- | --- | --- |
| ADC0\_GAIN1 | 0x8C | ADC 第 10~19 采样保持电路增益控制 |
| ADC0\_DC\_A0 | 0x90 | ADC 采样保持电路 A 非 1 倍增益 DC offset |
| ADC0\_DC\_A1 | 0x94 | ADC 采样保持电路 A 1 倍增益 DC offset |
| ADC0\_AMC\_A0 | 0x98 | ADC 采样保持电路 A 非 1 倍增益增益校正 |
| ADC0\_AMC\_A1 | 0x9C | ADC 采样保持电路 A 1 倍增益增益校正 |
| ADC0\_DC\_B0 | 0xA0 | ADC 采样保持电路 B 非 1 倍增益 DC offset |
| ADC0\_DC\_B1 | 0xA4 | ADC 采样保持电路 B 1 倍增益 DC offset |
| ADC0\_AMC\_B0 | 0xA8 | ADC 采样保持电路 B 非 1 倍增益增益校正 |
| ADC0\_AMC\_B1 | 0xA C | ADC 采样保持电路 B 1 倍增益增益校正 |
| ADC0\_IE | 0xB0 | ADC 中断使能 |
| ADC0\_IF | 0xB4 | ADC 中断标志 |
| ADC0\_CFG | 0xB8 | ADC 对齐模式配置 |
| ADC0\_TRIG | 0xBC | ADC 采样模式配置 |
| ADC0 SWT | 0xC0 | ADC 软件触发 |
| ADC0\_DAT0\_TH | 0xC4 | ADC 通道 0 阈值寄存器 |

# 10.2.2 采样数据寄存器

10.2.2.1 ADC0\_DAT0

地址：0x4001\_1400

复位值：0x0

表10-6采样数据寄存器ADC0\_DAT0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DATA0 | ADC 第 0 个采样数据 |

# 10.2.2.2 ADC0\_DAT1

地址：0x4001\_1404

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT1 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT1 | ADC 第 1 个采样数据 |

# 10.2.2.3 ADC0\_DAT2

地址：0x4001\_1408

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT2 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT2 | ADC 第 2 个采样数据 |

# 10.2.2.4 ADC0\_DAT3

地址：0x4001\_140C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT3 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | DAT3 | ADC 第 3 个采样数据 |

# 10.2.2.5 ADC0\_DAT4

地址：0x4001\_1410

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT4 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DATA4 | ADC 第 4 个采样数据 |

# 10.2.2.6 ADC0\_DAT5

地址：0x4001\_1414

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT5 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DATA5 | ADC 第 5 个采样数据 |

# 10.2.2.7 ADC0\_DAT6

地址：0x4001\_1418

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT6 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT6 | ADC 第 6 个采样数据 |

# 10.2.2.8 ADC0\_DAT7

地址：0x4001\_141C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT7 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT7 | ADC 第 7 个采样数据 |

# 10.2.2.9 ADC0\_DAT8

地址：0x4001\_1420

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT8 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | DAT8 | ADC 第 8 个采样数据 |

# 10.2.2.10 ADC0\_DAT9

地址：0x4001\_1424

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT9 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT9 | ADC 第 9 个采样数据 |

# 10.2.2.11 ADC0\_DAT10

地址：0x4001\_1428

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT10 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT10 | ADC 第 10 个采样数据 |

# 10.2.2.12 ADC0\_DAT11

地址：0x4001\_142C复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT11 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT11 | ADC 第 11 个采样数据 |

# 10.2.2.13 ADC0\_DAT12

地址：0x4001\_1430

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT12 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT12 | ADC 第 12 个采样数据 |

# 10.2.2.14 ADC\_DAT13

地址：0x4001\_1434

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT13 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | DAT13 | ADC 第 13 个采样数据 |

# 10.2.2.15 ADC0\_DAT14

地址：0x4001\_1438

复位值：0x0

表10-20采样数据寄存器ADC0\_DAT14

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT14 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT14 | ADC 第 14 个采样数据 |

# 10.2.2.16 ADC0\_DAT15

地址：0x4001\_143C

复位值：0x0

表10-21采样数据寄存器ADC0\_DAT15

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT15 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT15 | ADC 第 15 个采样数据 |

# 10.2.2.17 ADC0\_DAT16

地址：0x4001\_1440复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT16 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT16 | ADC 第 16 个采样数据 |

# 10.2.2.18 ADC0\_DAT17

地址：0x4001\_1444

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT17 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT17 | ADC 第 17 个采样数据 |

# 10.2.2.19 ADC0\_DAT18

地址：0x4001\_1448

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT18 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | DAT18 | ADC 第 18 个采样数据 |

# 10.2.2.20 ADC0\_DAT19

地址：0x4001\_144C

复位值：0x0

表10-25采样数据寄存器ADC0\_DAT19

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAT19 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT19 | ADC 第 19 个采样数据 |

# 10.2.3 信号来源寄存器

# 10.2.3.1 ADC0\_CHNO

地址：0x4001\_1450

复位值：0x0

表10-26信号来源寄存器ADC0\_CHNO

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | DS1 |  |  |  |  |  |  |  | DS0 |  |  |  |  |  |
| RW |  |  |  |  |  |  |  | RW |  |  |  |  |  |
| 0 |  |  |  |  |  |  |  | 0 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS1 | ADC 第 1 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS0 | ADC 第 0 个采样信号选择 |

ADC第0/1次采样是同步采样。

# 10.2.3.2 ADC0\_CHN1

地址：0x4001\_1454

复位值：0x0

表10-27信号来源寄存器ADC0\_CHN1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | DS2 | | | |  |  |
|  | RW | | | | | | | | | RW | | | |  |
|  | 0 | | | | | | | | | 0 | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS3 | ADC 第 3 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS2 | ADC 第 2 个采样信号选择 |

ADC第2/3次采样是同步采样。

# 10.2.3.3 ADC0\_CHN2

地址：0x4001\_1458

复位值：0x0

表10-28信号来源寄存器ADC0\_CHN2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS5 | ADC 第 5 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS4 | ADC 第 4 个采样信号选择 |

ADC第4/5次采样是同步采样。

# 10.2.3.4 ADC0\_CHN3

地址：0x4001\_145C复位值：0x0

表10-29信号来源寄存器ADC0\_CHN3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | RW |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS7 | ADC 第 7 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS6 | ADC 第 6 个采样信号选择 |

ADC第6/7次采样是同步采样。

# 10.2.3.5 ADC0\_CHN4

地址：0x4001\_1460

复位值：0x0

表10-30信号来源寄存器ADC0\_CHN4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS9 | ADC 第 9 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS8 | ADC 第 8 个采样信号选择 |

ADC第8/9次采样是同步采样。

# 10.2.3.6 ADC0\_CHN5

地址：0x4001\_1464

复位值：0x0

表10-31信号来源寄存器ADC0\_CHN5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | DS10 | | | |  |  |
|  | RW | | | | | | | | | RW | | | |  |
|  | 0 | | | | | | | | | 0 | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS11 | ADC 第 11 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS10 | ADC 第 10 个采样信号选择 |

ADC第10/11次采样是同步采样。

# 10.2.3.7 ADC0\_CHN6

地址：0x4001\_1468

复位值：0x0

表10-32信号来源寄存器ADC0\_CHN6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS13 | ADC 第 13 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS12 | ADC 第 12 个采样信号选择 |

ADC第12/13次采样是同步采样。

# 10.2.3.8 ADC0\_CHN7

地址：0x4001\_146C复位值：0x0

表10-33信号来源寄存器ADC0\_CHN7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | DS14 | | | |  |  |
|  | RW | | | | | | | | | RW | | | |  |
|  | 0 | | | | | | | | | 0 | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS15 | ADC 第 15 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS14 | ADC 第 14 个采样信号选择 |

ADC第14/15次采样是同步采样。

# 10.2.3.9 ADC0\_CHN8

地址：0x4001\_1470

复位值：0x0

表10-34信号来源寄存器ADC0\_CHN8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS17 | ADC 第 17 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS16 | ADC 第 16 个采样信号选择 |

ADC第16/17次采样是同步采样。

# 10.2.3.10 ADC0\_CHN9

地址：0x4001\_1474

复位值：0x0

表10-35信号来源寄存器ADC0\_CHN9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | DS18 | | | |  |  |
|  | RW | | | | | | | | | RW | | | |  |
|  | 0 | | | | | | | | | 0 | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | DS19 | ADC 第 19 个采样信号选择 |
| [7:5] |  | 未使用 |
| [4:0] | DS18 | ADC 第 18 个采样信号选择 |

ADC第18/19次采样是同步采样。

表10-36ADC采样信号通道选择

|  |  |  |
| --- | --- | --- |
| ADC采样电路输入信号正端选择 | 5'h00 | OPA0\_OUT |
| 5'h01 | OPA1\_OUT |
| 5'h02 | OPA2\_OUT |
| 5'h03 | OPA3\_OUT |
| 5'h04 | ADC\_CH4 |
| 5'h05 | ADC\_CH5 |
| 5'h06 | ADC\_CH6 |
| 5'h07 | ADC\_CH7 |
| 5'h08 | ADC\_CH8 |
| 5'h09 | ADC\_CH9 |
| 5'h0A | ADC\_CH10 |
| 5'h0B | ADC\_CH11 |
| 5'h0C | ADC\_CH12 |
| 5'h0D | ADC\_CH13 |
| 5'h0E | ADC\_CH14 |
| 5'h0F | ADC\_CH15 |
| 5'h10 | ADC\_CH16 |
| 5'h11 | ADC\_CH17 |
| 5'h12 | Temp |
| 5'h13 | VSS |

ADC采样电路输入信号的负端统一接地。

以单段触发采样8个采样为例，ADC0\_CHNx寄存器里设置的第0/1个采样是同步采样的，2/3是同步采样的，4/5是同步采样的，6/7是同步采样的。若设置的采样个数是奇数，则最后一次只仍同步采样两个通道，但只转换一个通道的采样值。

# 10.2.4 分段通道数寄存器

# 10.2.4.1 ADC0\_CHNT0

地址：0x4001\_1478

复位值：0x0

表10-37分段通道数寄存器ADC0\_CHNT0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | S2 | 两段或四段采样模式下第二段采样个数 |
| [7:5] |  | 未使用 |
| [4:0] | S1 | 单段、两段或四段采样模式下第一段采样个数 |

注意，ADC0\_CHNT0. S1在单段、两段、四段模式均不允许设置为0；ADC0\_CHN0. S2在两段、四段模式下不允许设置为0，在单段模式下由于没有用到这一字段可以为0。

# 10.2.4.2 ADC0\_CHNT1

地址：0x40011400\_0x7C

复位值：0x0

表10-38分段通道数寄存器ADC0\_CHNT1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |
| [12:8] | S4 | 四段采样模式下第四段采样个数 |

|  |  |  |
| --- | --- | --- |
| [7:5] |  | 未使用 |
| [4:0] | S3 | 四段采样模式下第三段采样个数 |

注意，ADC0\_CHNT1. S3 在四段模式不允许设置为 0；ADC0\_CHN1. S4 在四段模式下不允许设置为 0，在单段、两段模式下由于没有用到这两个字段可以为 0。

1 表示 1 个采样，2 表示 2 个采样，……，12 表示 12 个采样，……，20 表示 20 个采样。多次采样可以采样同一个模拟通道，也可以采样不同的模拟通道。

# 10.2.5 配置寄存器

# 10.2.5.1 ADC0\_CFG

地址：0x4001\_14B8

复位值：0x0

表10-39模式配置寄存器ADC0\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | FSM\_RESET |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | TH\_TYPE |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | DATA\_ALIGN |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:3] |  | 未使用 |
| [2] | FSM\_RESET | 状态机复位，软件写入后状态机回到 idle 状态，完成后自动清零 |
| [1] | TH\_TYPE | ADC0\_DAT0\_TH 作为上阈值或下阈值 1: 上阈值 0: 下阈值 |
| [0] | DATA\_ALIGN | ADC0\_DAT 对齐方式 0: 左对齐，右端补 4'h0， 1: 右对齐，左端补 4bit 符号位 |

# 10.2.5.2 ADC0\_TRIG

地址：0x4001\_14BC

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CON\_TRIG | TRG\_MODE | SINGLE\_TCNT | UTIMER3 CMP1\_E | UTIMER3 CMP0\_E | UTIMER2 CMP1\_E | UTIMER2 CMP0\_E | MCPWM\_TRG3\_E | MCPWM\_TRG2\_E | MCPWM\_TRG1\_E | MCPWM\_TRG0\_E |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:15] |  | 未使用 |
| [14] | CON\_TRIG | 连续触发使能，高有效 写1，ADC进入连续触发模式进行连续采样 写0，ADC停止采样 |
| [13:12] | TRG\_MODE | 触发模式 0:单段触发； 1:两段触发； 2:保留； 3:四段触发 |
| [11:8] | SINGLE\_TCNT | 单段触发模式下触发一次采样所需的事件数 (0表示1次事件即触发，15表示16次事件才触发) |
| [7] | UTIMER3\_CMP\_E | 4'b1xxx:UTimer3 CMP1 被使能 |
| [6] | UTIMER3\_CMP0\_E | 4'bx1xx:UTimer3 CMP0 被使能 |
| [5] | UTIMER2\_CMP1\_E | 4'bxx1x:UTimer2 CMP1 被使能 |
| [4] | UTIMER2\_CMP0\_E | 4'bxxx1:UTimer2 CMP0 被使能 |
| [3] | MCPWM\_TRG3\_E | 4'b1xxx: MCPWM T3 被使能 |
| [2] | MCPWM\_TRG2\_E | 4'bx1xx: MCPWM T2 被使能 |
| [1] | MCPWM\_TRG1\_E | 4'bxx1x: MCPWM T1 被使能 |
| [0] | MCPWM\_TRG0\_E | 4'bxxx1: MCPWM T0 被使能 |

TADC[3] = MCPWM\_T3 | UTimer3\_CMP1 TADC[2] = MCPWM\_T2 | UTimer3\_CMP0 TADC[1] = MCPWM\_T1 | UTimer2\_CMP1 TADC[0] = MCPWM\_T0 | UTimer2\_CMP0

来自两个不同设备的四个触发源相或之后送至ADC采样模块作为触发事件TADC[3:0]

在进入连续采样模式之前，须保证ADC采样模块处于空闲状态，推荐关闭ADC所有硬件触发使能，同时通过写ADC0\_CFG.FSM\_RESET=1；使得ADC采样状态机回到空闲状态。

MCPWM对ADC的触发信号可以通过配置GPIO为第9功能，即ADC\_TRIGGER功能送出用于捕捉调试。ADC触发信号在芯片内部为一个ADC时钟周期的窄脉冲。每发生一次ADC触发，ADC\_TRIGGER信号翻转一次，以便于输出捕捉。

UTimer 对 ADC 的触发信号可以通过配置 GPIO 为第 8 功能，即 Timer2/3 功能送出用于捕捉调试。

# 10.2.6 增益选择寄存器

# 10.2.6.1 ADC0\_GAIN0

地址：0x4001\_1488

复位值：0x0

表10-41增益选择寄存器ADC0\_GAIN0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:10] |  | 未使用 |
| [9] | DG9 | ADC0\_DAT9 采样保持电路增益选择 |
| [8] | DG8 | ADC0\_DAT8 采样保持电路增益选择 |
| [7] | DG7 | ADC0\_DAT7 采样保持电路增益选择 |
| [6] | DG6 | ADC0\_DAT6 采样保持电路增益选择 |
| [5] | DG5 | ADC0\_DAT5 采样保持电路增益选择 |
| [4] | DG4 | ADC0\_DAT4 采样保持电路增益选择 |
| [3] | DG3 | ADC0\_DAT3 采样保持电路增益选择 |
| [2] | DG2 | ADC0\_DAT2 采样保持电路增益选择 |
| [1] | DG1 | ADC0\_DAT1 采样保持电路增益选择 |
| [0] | DG0 | ADC0\_DAT0 采样保持电路增益选择 |

0:2/3 增益，1:1 倍增益。

# 10.2.6.2 ADC0\_GAIN1

地址：0x4001\_148C

复位值：0x0

表10-42增益选择寄存器ADC0\_GAIN1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | DG19 | DG18 | DG17 | DG16 | DG15 | DG14 | DG13 | DG12 | DG11 | DG10 |  |  |  |  |  |  |  |  |  |
|  |  | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名 称 | 说明 |
| [31:10] |  | 本使用 |
| [9] | DG19 | ADC0\_DAT19 采样保持电路增益选择 |
| [8] | DG18 | ADC0\_DAT18 采样保持电路增益选择 |
| [7] | DG17 | ADC0\_DAT17 采样保持电路增益选择 |
| [6] | DG16 | ADC0\_DAT16 采样保持电路增益选择 |
| [5] | DG15 | ADC0\_DAT15 采样保持电路增益选择 |
| [4] | DG14 | ADC0\_DAT14 采样保持电路增益选择 |
| [3] | DG13 | ADC0\_DAT13 采样保持电路增益选择 |
| [2] | DG12 | ADC0\_DAT12 采样保持电路增益选择 |
| [1] | DG11 | ADC0\_DAT11 采样保持电路增益选择 |
| [0] | DG10 | ADC0\_DAT10 采样保持电路增益选择 |

0:2/3 增益，1:1 倍增益。

# 10.2.7 中断寄存器

10.2.7.1 ADC0\_IE

地址：0x4001\_14B0

复位值：0x0

表10-43中断使能寄存器ADC0\_IE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | |
|  | TH\_IE | HERR\_IE | SERR\_IE | S4FINISH\_IE | S3FINISH\_IE | S2FINISH\_IE | S1FINISH\_IE |  |
| RW | RW | RW | RW | RW | RW | RW |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:7] |  | 未使用 |
| [6] | TH\_IE | ADC0\_DAT0 超阈值中断使能 |
| [5] | HERR\_IE | 硬件触发发生在非空闲状态中断使能 |
| [4] | SERR\_IE | 软件触发发生在非空闲状态中断使能 |
| [3] | S4FINISH\_IE | 第四段采样完成中断使能 |
| [2] | S3FINISH\_IE | 第三段采样完成中断使能 |
| [1] | S2FINISH\_IE | 第二段采样完成中断使能 |
| [0] | S1FINISH\_IE | 第一段采样完成中断使能 |

# 10.2.7.2 ADC0\_IF

地址：0x4001\_14B4

复位值：0x0

表10-44中断标志寄存器ADC0\_IF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  | TH\_IF | HERR\_IF | SERR\_IF | S4FINISH\_IF | S3FINISH\_IF | S2FINISH\_IF | S1FINISH\_IF |  |
|  |  |  |  |  |  |  |  | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |  |
|  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:7] |  | 未使用 |
| [6] | TH\_IF | ADC0\_DAT0 超阈值中断标志 |
| [5] | HERR\_IF | 硬件触发发生在非空闲状态中断标志 |
| [4] | SERR\_IF | 软件触发发生在非空闲状态中断标志 |
| [3] | S4FINISH\_IF | 第四段采样完成中断标志 |
| [2] | S3FINISH\_IF | 第三段采样完成中断标志 |
| [1] | S2FINISH\_IF | 第二段采样完成中断标志 |
| [0] | S1FINISH\_IF | 第一段采样完成中断标志 |

以上ADC0\_IF标志位，0：表示未发生中断，1：表示发生过中断，写1清零

# 10.2.8 软件触发寄存器

# 10.2.8.1 ADC0 SWT

地址：0x4001\_14C0

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWT | | | | | | | | | | | | | | |  |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | SWT | 写入数据为 0x5AA5 时，产生一次软件触发 |

注意，软件触发采集寄存器为只写寄存器，且只有写入数据为0x5AA5时产生软件触发事件，一次总线的写入产生一次软件触发，数据写入产生一个软件触发后寄存器自动清零，等待后续的软件触发到来。

# 10.2.9 直流偏置寄存器

芯片出厂时已经过工厂标定，标定数据存放在Flash info中，芯片上电会自动加载。ADC模块在初始化的时候，需要根据数据左右对齐模式配置DC offset，可以参看芯片供应商提供的库函数。

# 10.2.9.1 ADC0\_DC\_A0

地址：0x4001\_1490

复位值：0x0

表10-46直流偏置寄存器ADC0\_DC\_A0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DC\_OFFSET | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DC\_OFFSET | 采样电路 A 非一倍增益 ADC DC offset |

# 10.2.9.2 ADC0\_DC\_A1

地址：0x4001\_1494

复位值：0x0

表10-47直流偏置寄存器ADC0\_DC\_A1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DC OFFSET | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DC\_OFFSET | 采样电路 A 一倍增益 ADC DC offset |

# 10.2.9.3 ADC0\_DC\_B0

地址：0x4001\_14A0 复位值：0x0

表10-48直流偏置寄存器ADC0\_DC\_B0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DC\_OFFSET | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DC\_OFFSET | 采样电路 B 非一倍增益 ADC DC offset |

# 10.2.9.4 ADC0\_DC\_B1

地址：0x4001\_14A4

复位值：0x0

表10-49直流偏置寄存器ADC0\_DC\_B1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DC\_OFFSET | | | | | | | | | | | | | |  |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DC\_OFFSET | 采样电路 B 一倍增益 ADC DC offset |

# 10.2.10增益校正寄存器

# 10.2.10.1ADC0\_AMC\_A0

地址：0x4001\_1498

复位值：0x0

表10-50增益校正寄存器ADC0\_AMC\_A0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | AM\_CALL | | | | | | | | |  |
| RW | | | | | | | | |  |
| 0 | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |
| [9:0] | AM\_CALL | 采样电路 A 非一倍增益 ADC 增益校正寄存器 |

# 10.2.10.2 ADC0\_AMC\_A1

地址：0x4001\_149C

复位值：0x0

表10-51增益校正寄存器ADC0\_AMC\_A1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | A M CALL | | | | | | | | |  |
| RW | | | | | | | | |  |
| 0<lcol> | | | | | | | |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |
| [9:0] | AM\_CALL | 采样电路 A 一倍增益 ADC 增益校正寄存器 |

# 10.2.10.3 ADC0\_AMC\_B0

地址：0x4001\_14A8

复位值：0x0

表10-52增益校正寄存器ADC0\_AMC\_B0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | (AM\_CALL) | | | | | | | | |  |
| RW | | | | | | | | |  |
| 0<lcel"> | | | | | | | |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [9:0] | AM\_CALL | 采样电路 B 非一倍增益 ADC 增益校正寄存器 |

# 10.2.10.4 ADC0\_AMC\_B1

地址：0x4001\_14AC

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |
| [9:0] | AM\_CALL | 采样电路 B 一倍增益 ADC 增益校正寄存器 |

ADC0\_AMC存储的为增益校正系数 ，为10bit无符号定点数，ADCAMC[9]为整数部分，ADCAMC[8:0]为小数部分。所存值为1左右。

ADC有1倍增益和非1倍增益两种配置，两种配置对应两套校正参数，每套校正数据分别包含一个DC offset(以下记为DCoffset)和一个增益校正值 。

记ADC输出的数字量为 ， 对应的真实值为D。 为编码数制的0，则

因为有A/B两路采样电路，因此有两组DC和AMC校正值，硬件会在不同通道进行数据校正时自动选择对应的A/B路的DC/AMC。

# 10.2.11通道0阈值寄存器

10.2.11.1 ADC0\_DAT0\_TH

地址：0x4001\_14C4

复位值：0x0

表10-54通道0阈值寄存器ADC0\_DAT0\_TH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA0\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DAT0\_TH | 通道 0 阈值寄存器 |

用来配置ADC数值比较中断，当ADC0\_DAT0大于或小于ADC0\_DAT0\_TH设置的值时，可以触发ADC数值比较中断。根据ADC0\_CFG[1]配置，ADC0\_DAT0\_TH可作为ADC0\_DAT0的上调值或下阈值；在一些应用上可以替代硬件比较器。具体来说，当ADC0\_CFG.TH\_TYPE=1时，即ADC0\_DAT0\_TH为上调值，如果ADC0\_DAT0>ADC0\_DAT0\_TH，则产生阈值中断；当ADC0\_CFG.TH\_TYPE=0时，即ADC0\_DAT0\_TH为下阈值，如果ADC0\_DAT0<ADC0\_DAT0\_TH，则产生阈值中断。

# 10.3 实现说明

# 10.3.1 DMA请求

只有ADC\_IFS1FINISHLIF，即第一段采样完成事件可以作为DMA请求事件。因此只有单段触发采样完成，或多段触发采样第一段采样完成时会发起DMA请求，且这个请求与ADC\_IE设置无关。DMA响应请求后会通过内部的写清零直接将ADC\_IFS1FINISHLIF清零。

# 10.3.2 连续采样

当配置了连续采样模式时，即ADC\_TRIG.CON\_TRIG=1，ADC会反复采样单段模式下的各个通道，结束后立即开始下一轮单段采样，如果开启了单段采样中断，则会频繁发生中断，大部分情况下CPU来不及响应如此频繁的中断。软件写入ADC\_TRIG.CON\_TRIG=0，可以关闭连续采样。

连续采样模式可以配合ADC\_DAT0阈值中断进行使用，只有检测到信号超过阈值时才产生阈值中断通知CPU。

连续采样可以配置采样多个通道，具体采样通道的配置与单段采样模式相同。

# 10.4 应用指南

# 10.4.1 ADC采样触发模式

ADC支持一段、两段、四段采样模式，每段采样需要特定的外部事件来触发开始，每段采样支持不同采样次数和采样信号通道配置。ADC内部的状态转移描述如下，共有8个状态分别为采样状态03，空闲状态03。

第一次触发

来自MCPWM/UTimer的定时事件TADC[0]/TADC[1]/TADC[2]/TADC[3]可以触发ADC采样。可以选择四个触发源的任何一个或者几个触发采样。也可以通过向ADC SWT写入命令字的方式16’h5AA5软件触发ADC采样。

第一段采样

判断是否为一段采样。

是：采样次数达到预设值ADC\_CHNT0. S1即第一段采样次数后，ADC回到空闲状态0；若采样次数尚未达到预设值，继续采样。

否：采样次数达到预设值ADC\_CHNT.S1，ADC进入空闲状态1（两段或四段采样第一段完成，等待触发第二段)；采样次数未达到预设值，继续第一段采样。

第二段触发

进入第二段采样

第二段采样

第二段采样次数到达预设值ADC\_CHNT0. S2即第二段采样次数，判断是否为两段采样。

是：结束本次采样，回到空闲状态0。

否：进入空闲状态2，等待第三次触发及第四次触发完成采样。

第三段触发

进入第三段采样，如果到了次状态，则配置一定是四段采样。

第三段采样

第三段采样次数到达预设值ADC\_CHNT1. S3即第三段采样次数，进入空闲状态3。

第四段触发

第四段采样

第四段采样通道数到达预设值ADC\_CHNT1. S4即第四段采样次数，回到空闲状态0。

各种硬件触发模式的触发条件汇总如表10- 55所示。其中单段采样模式较为特殊，可以通过ADC0\_CFG寄存器设置，一次MCPWM/UTimer定时事件即触发采样，还是多次MCPWM/UTimer定时事件发生到一定次数才触发采样；而两段、四段采样模式仅支持一次MCPWM/UTimer事件即触发采样。

此外ADC模块也支持通过软件向ADC0 SWT写入0x5AA5的方式触发采样，软件触发也仅支持写入一次即触发。

表10-55ADC采样触发模式

|  |  |  |  |
| --- | --- | --- | --- |
|  | 单段触发 | 两段触发 | 四段触发 |
| Timer触发 | None (Timer trigger 使能未打开) | 第一段 TADC[0] 第二段 TADC[1] | 第一段 TADC[0] 第二段 TADC[1] 第三段 TADC[2] 第四段 TADC[3] |
| C 次 TADC[0] |
| C 次 TADC[1] |
| C 次 TADC[2] |
| C 次 TADC[3] |
| C 次\* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | TADC[0]/TADC[1]/ TADC[2]/TADC[3] |  |  |
| 软件触发 | 向 ADC\_SWT 写入 16'h5aa5 | 第一段向 ADC0\_SWT 写入 0x5AA5 第二段向 ADC0\_SWT 写入 0x5AA5 | 第一段向 ADC0\_SWT 写入 0x5AA5 第二段向 ADC0\_SWT 写入 0x5AA5 第三段向 ADC0\_SWT 写入 0x5AA5 第四段向 ADC0\_SWT 写入 0x5AA5 |

次通过ADC0\_TRIG.SINGLE\_TCNT设置。ADC0\_TRIG.SINGLE\_TCNT只在单段触发下使用，如果同时使能了TADC[3:0]，则4个触发源都会被计数，到SINGLE\_TCNT次触发一次ADC采样转换。

# 10.4.1.1 单段触发模式

单段触发收到一次触发完成一段采样动作，一段采样可能包含多次对模拟信号的采样，次数由分段采样次数寄存器配ADC\_CHNT0[4:0]进行配置，寄存器数值为 时，对应的采样次数为

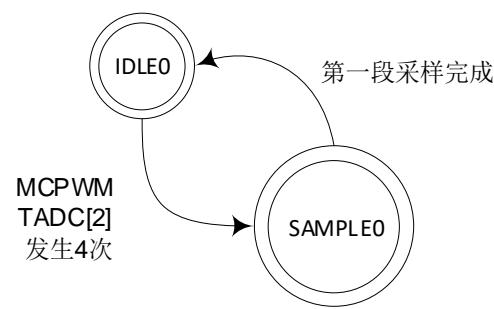
假设单段采样配置通道数目为4，则采样转换后的数据会依次填充到ADC0DAT0、ADC0DAT1、ADC0DAT2、ADC0DAT3。

触发事件可以是来自外部的MCPWM/UTimer定时信号TADC[0]、TADC[1]、TADC[2]、TADC[3]、发生到预设次数、或者为软件触发。

每个采样的信号源通过信号来源寄存器ADCCHN0/1/2. ../19进行配置选定，信号源的选定需在触发前完成，且在一次采样过程完成前不应该改变。

完成一段采样动作后，进入空闲状态，并产生采样完成中断。

以MCPWM触发单段采样为例，设置TADC[2]发生4次才进行触发，状态转移如图10- 4所示。

  
图10-4ADC单段采样状态转移图

# 10.4.1.2 两段触发模式

两段触发需要两次触发才能完成完整的一轮采样。第一个触发到达时进行第一段采样，第二个触发到达时进行第二段采样。

触发事件可以是来自外部的MCPWM/UTimer定时信号TADC[0]和TADC[1]或两次软件触发。

TADC[0]或软件触发发生后，先进行ADC\_CHNT0[4:0]次采样，完成后进入空闲状态并等待下一个触发信号的到来；TADC[1]或软件触发作为第二个触发信号发生后，再进行ADC\_CHNT0[12:8]次

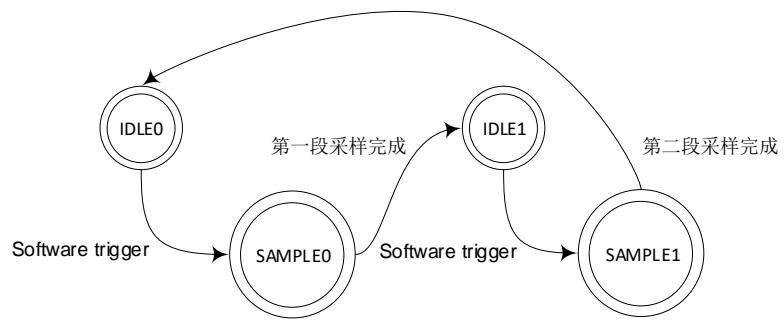
采样。采样次数均通过分段采样次数寄存器ADC\_CHNT0进行配置。

假设两段采样配置通道数目分别为2和3，则第一段采样转换后的数据会依次填充到ADC0\_DAT0、ADC0\_DAT1，第二段采样转换后的数据会依次填充到ADC0\_DAT2、ADC0\_DAT3、ADC0\_DAT4。

每个采样的信号源通过寄存器配置选定，信号源的选定需在触发前完成，且在一次采样过程完成前不应该改变。

软件触发较硬件触发的优先级低，在硬件触发采样的过程中发生软件触发，状态机不予处理，而产生一个错误中断。即只有状态机处于空闲状态时才会处理软件触发的采样请求。如果需要使用软件触发采样，需要确保硬件触发已经关闭。然后通过向ADC SWT寄存器写入0x5AA5以产生一次软件触发。

以两次软件触发两段采样为例，状态转移如图10- 5所示

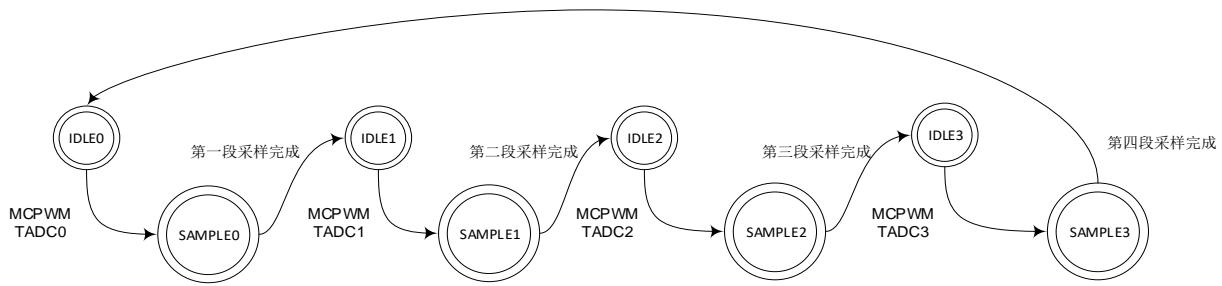
  
图10-5ADC两段采样状态转移图

# 10.4.1.3 四段触发模式

与两段触发类似。四段的触发源分别为TADC[0]、TADC[1]、TADC[2]、TADC[3]，且必须为MCPWM/UTimer TADC[0]/TADC[1]TADC[2]TADC[3]顺序触发ADC的四段采样；或者也可以是4次软件触发采样。四段采样的第一、二、三、四段采样次数分别为ADC\_CHNT0[4:0]、ADC\_CHNT0[12:8]、ADC\_CHNT1[4:0]、ADC\_CHNT1[12:8]。

假设四段采样配置通道数目分别为2、3、1、5，则第一段采样转换后的数据会依次填充到ADC0\_DAT0、ADC0\_DAT1，第二段采样转换后的数据会依次填充到ADC0\_DAT2、ADC0\_DAT3、ADC0\_DAT4，第三段采样转换后的数据会填充到ADC0\_DAT5，第四段采样转换后的数据会依次填充到ADC0\_DAT6、ADC0\_DAT7、ADC0\_DAT8、ADC0\_DAT9、ADC0\_DAT10

以MCPWMTADC[0]/TADC[1]/TADC[2]/TADC[3]触发四段采样为例的状态转移如图10- 6所示。

  
图10-6ADC四段采样状态转移图

为使用MCPWM定时器产生ADC采样触发信号，需要配置MCPWM\_TMR0/ MCPWM\_TMR1/ MCPWM\_TMR2/ MCPWM\_TMR3等寄存器，对应TADC0/1/2/3发生时的MCPWM计数器值，此外需要配置MCPWM\_TH设置计数器计数范围以及MCPWM\_TCLK设置计数时钟频率并使能时钟。

同理，如果使用UTimer对ADC进行定时触发，也需要配置UTIMER\_UNT2\_CMP0/UTIMER\_UNT2\_CMP1/UTIMER\_UNT3\_CMP0/UTIMER\_UNT3\_CMP1。

# 10.4.2 中断

# 10.4.2.1 单段触发采样完成中断

采样完成产生一个中断。

# 10.4.2.2 两段触发采样完成中断

第一段采样完成产生一个中断，第二段采样完成产生一个中断。

# 10.4.2.3 四段触发采样完成中断

第一段采样完成产生一个中断，第二段采样完成产生一个中断，第三段采样完成产生一个中断，第四段采样完成产生一个中断。

# 10.4.3 配置修改

建议在ADC中断中进行ADC0\_CHNTx的配置和修改，因为进入ADC中断后说明ADC此时已完成一段采样且处于空闲状态。而在主程序中，无法确认ADC运行状态，因此在主程序中如需修改ADC0\_CHNx和ADC0\_CHNT等寄存器，需要先关闭ADC触发，并向ADC0\_CFG.FSM\_RESET写入1，以复位ADC接口电路状态机，确保ADC不在工作状态。如果ADC在运行中配置发生变化会发生不可预判的行为。

示例程序如下

ADC0\_TRIG\_temp = ADC0\_TRIG; //保存ADC采样触发配置ADC0\_TRIG = 0x0000; //关闭ADC采样触发ADC0\_CFG |= 0x0004; //复位ADC接口电路状态机//此处进行的ADC采样通道和通道数的修改仅为示例ADC0\_CHNT0 = 0x0005 //修改ADC单段采样通道数为5ADC0\_CHN0 = 0x0305; //修改ADC第0/1次采样通道为模拟通道5和3ADC0\_CHN1 = 0x0604; //修改ADC第2/3次采样通道为模拟通道4和6

ADC0\_TRIG = ADC0\_TRIG\_temp; //恢复ADC采样触发配置

# 10.4.4 选择对应的模拟通道

ADC所采样信号对应的通道，请查阅表10- 36ADC采样信号通道选择以及DATASHEET中

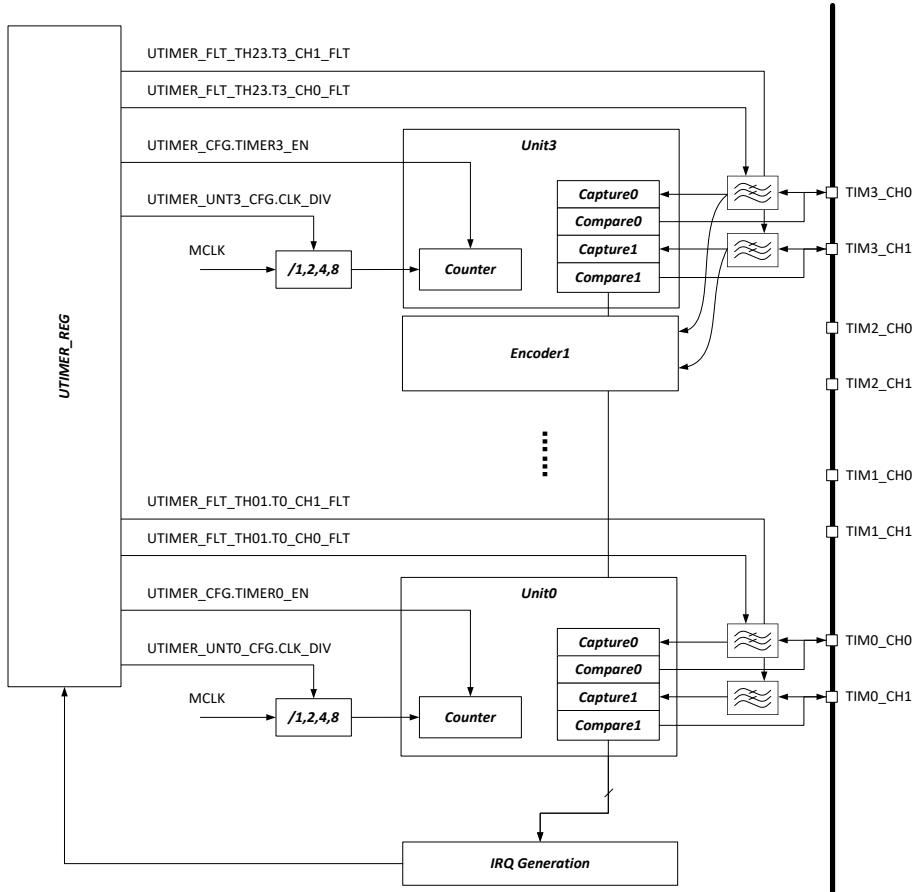
表 2.2 引脚功能选择。关闭对应 IO 的 IE 和 OE，即可使用其模拟功能。

# 11 通用定时器

# 11.1 概述

# 11.1.1 功能框图

如图 11- 1 所示，通用定时器 UTIMER 主要包括 4 个独立的 Timer，分别可以独立配置运行计数时钟和滤波常数。每个 Timer 可以用于输出特定周期占空比的波形，也可以捕获外部波形进行周期占空比的检测。

  
图 11-1 模块顶层功能框图

# 11.1.1.1 总线接口模块

总线接口模块包括：

将来自 AHB 总线的访问信号翻译为寄存器读写信号，控制寄存器模块的时钟，并对寄存器模块发起读写。

CG 时钟门控模块，在 AHB 总线无访问时，将寄存器模块时钟关闭以降低功耗。

# 11.1.1.2 寄存器模块

utimer\_reg 寄存器模块，实现

对各个子模块控制寄存器的读写。

对各个子模块状态、结果寄存器的访问。

对各个子模块中断信号的处理和中断产生。

# 11.1.1.3 IO 滤波模块

10 滤波模块对来自芯片外部的输入信号进行滤波，降低毛刺对定时器功能的影响。

# 11.1.1.4 通用定时器模块

utimer\_unt 模块实现了通用的定时器功能，包括比较和捕获工作模式，可以处理两个外部输入信号或者产生两个脉冲信号送到芯片外部。定时器模块中一共包括 4 个独立工作的通用定时器，Timer0/Unit0、Timer1/Unit1 为 16bit 位宽，Timer2/Unit2、Timer3/Unit3 为 32bit 位宽。每个定时器包含两个通道。

utimer\_unt 模块，支持外部事件触发开始计数。外部事件源头可配。当外部事件触发后，utimer\_unt 定时器开始自增。

# 11.1.1.5 编码器模块

编码器模块用于对芯片外部送入的编码器编码信号进行计数。定时器模块中集成了 2 个编码器模块。其中编码器的输入分别来自 Timer2 的通道 0/1 和 Timer3 的通道 0/1。使用编码器不影响 Timer 功能。

# 11.1.1.6 时钟分频模块

时钟分频模块用于产生时钟分频的各种信号。

# 11.1.2 功能特点

定时器模块有以下特点：

* 独立工作，可工作在不同频率下
* Timer0 和 Timer1 为 16bit 通用定时器
* Timer2 和 Timer3 为 32bit 通用定时器
* 每个通用定时器处理 2 个外部输入信号（捕获模式），或者产生 2 个输出信号（比较模式）
* 2 个独立工作计数器
* 对每个输入信号可以进行最大 120 个系统主时钟的滤波，即，当芯片工作在 96MHz 时钟频率下时，可以滤除 1.25us 宽度以下毛刺
* 支持 DMA 传输

# 11.2 实现说明

# 11.2.1 时钟分频

为了实现各个timer独立分频，且可以方便对中断/计数值进行写操作，采取了各个timer均工作在系统主频，但使用分频计数器来降低计数器计数频率。

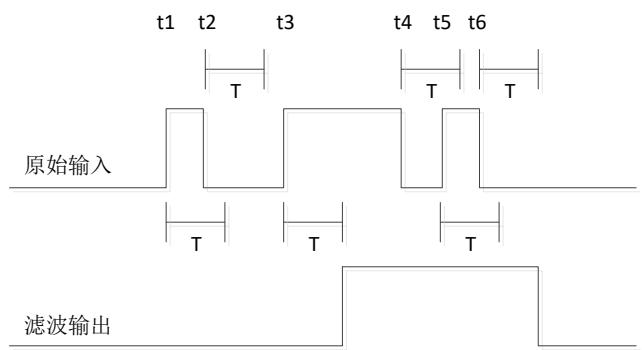
# 11.2.2 滤波

定时器模块共有8个/4对通道输入，定时器可以对每个输入进行不同程度的滤波。

通过配置滤波寄存器可以调整滤波宽度， 个系统时钟宽度。

输入信号滤波始终使用系统高速时钟，通常为96MHzPLL时钟，UTIMER\_UNTx\_CFG.CLK\_DIV对Timer计数时钟的分频对滤波时钟没有影响。

如图11- 2所示，原始输入信号在t1~t6几个时刻发生了翻转，滤波器宽度配置成T。可以看到只有t3和t6时刻发生的翻转维持了大于T的时间，因此从滤波器的输出看，信号仅发生了两次翻转。

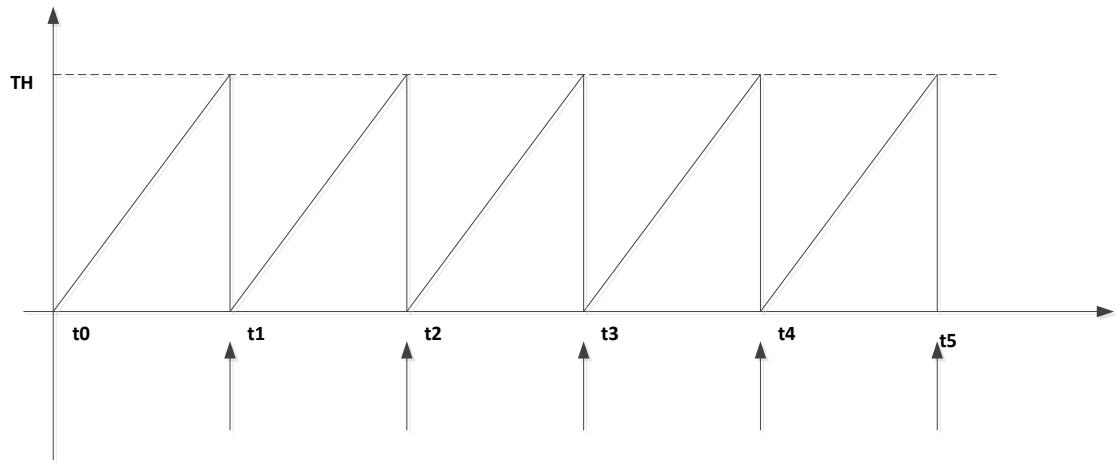
  
图11-2滤波示意图

# 11.2.3 模式

# 11.2.3.1计数器

Timer中的计数器采用递增方向计数。

计数器从0计数到TH值，再回到0重新开始计数，计数器回到0时，产生回零中断。实际计数周期为clk\_freq\*(TH+1)。

  
图11-3通用计数器

# 11.2.3.2 比较模式

比较模式下，计数器计数到UTIMER\_UNTx\_CMP值时，产生比较中断。比较模式可以驱动一个比较脉冲发生，在回零时，向IO口输出一个电平（极性可配置），在比较事件发生时，电平翻转，向IO口输出另一个电平。计数器回零时，仍然会产生回零中断。回零事件优先于比较事件，因此，两者同时发生时候（CMP==0），输出回零的值，而不会输出比较事件的值。

# 如果要实现某个Timer通道0输出全0：

1）可以设置UTIMER\_UNTx\_CFG.CH0\_POL=0

2）并设置UTIMER\_UNTx\_CMP0=UTIMER\_UNTx\_TH+1，。即Timer计数值回0时，通道输出0，且Timer计数值全程不命中CMP0。

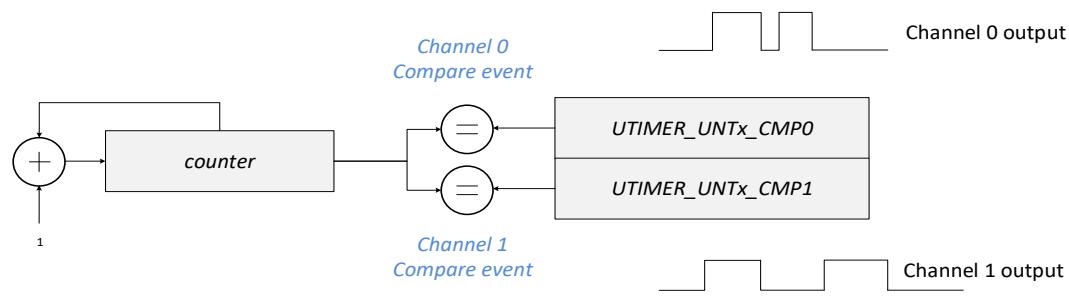
或设置UTIMER\_UNTx\_CMP0=0，即Timer命中CMP0和回零事件为相同事件。

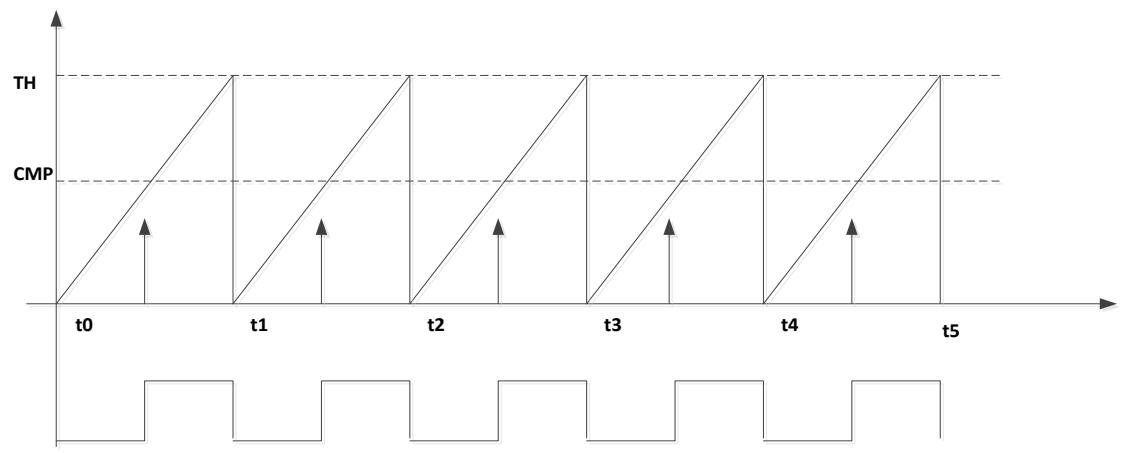
# 如果要实现某个Timer通道0输出全1：

1）可以设置UTIMER\_UNTx\_CFG.CH0\_POL=1

2）并设置UTIMER\_UNTx\_CMP0=UTIMER\_UNTx\_TH+1，。即Timer计数值回0时，通道输出1，且Timer计数值全程不命中CMP0。

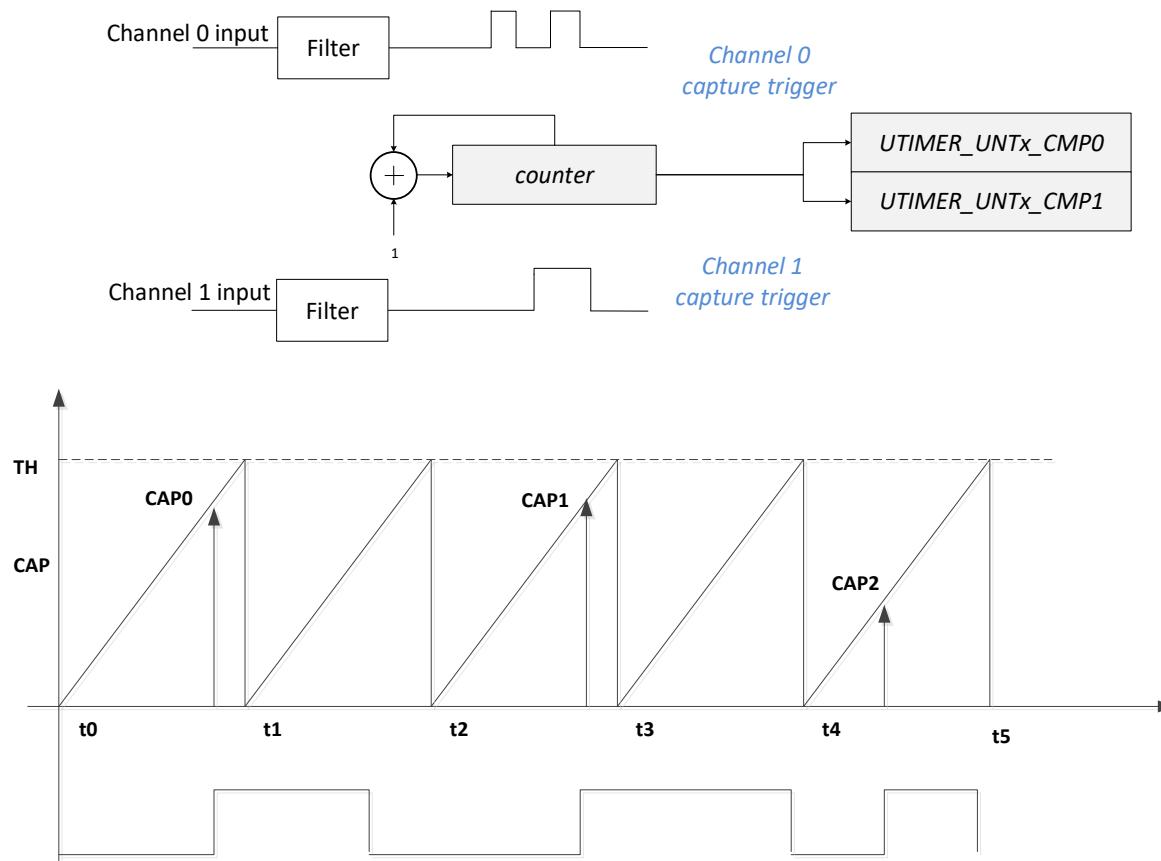
或设置UTIMER\_UNTx\_CMP0=0，即Timer命中CMP0和回零事件为相同事件。



  
图11-4 比较模式

# 11.2.3.3 捕获模式

捕获模式下，可以使用Timer来检测输入信号的上升/下降或者双沿，发生捕获事件（即输入信号电平变化）时，定时器计数值存入UTIMER\_UNTx\_CMP寄存器，并产生捕获中断。计数器回零时，仍然会产生回零中断。

  
图11-5 捕获模式

如图10- 5所示，定时器设置为上升沿捕获。在CAP0/CAP1/CAP2三个时刻点，捕获到输入信号发生上升沿变化，对应时刻点的定时器计数值将存入UTIMER\_UNTx\_CMP寄存器中。

# 11.2.4 编码器

编码器接口支持正交编码信号、符号加脉冲信号、CW/CCW双脉冲信号三种模式。

其中Encoder0的两个输入信号T1/T2分别来自Timer2Channel0/1对应的GPIO输入；Encoder1的T1/T2信号分别来自Timer3Channel0/1对应的GPIO输入。开启编码器功能时并不影响Timer功能的正常使用。

# 11.2.4.1 正交编码信号

正交编码信号多用于计数编码器圈数，输入为T1/T2两个信号，支持下表中两个模式。

概括来讲，T1/T2的跳变沿会导致计数器递增或递减。而计数器计数方向（递增或递减）由跳变信号之外的另一个稳态信号的电平高低决定。

如果T1发生了上升沿跳变，则看T2是高电平还是低电平，如果是高电平则计数器递减，如果是低电平计数器递增，T1下降沿计数器变化相反。

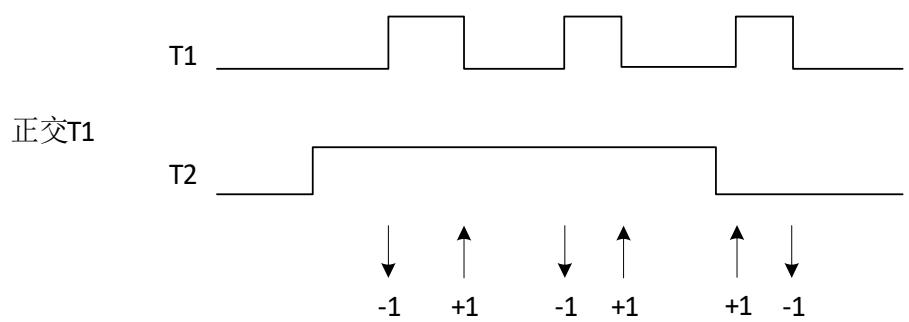
如果T2发生了上升沿跳变，则看T1是高电平还是低电平，如果是高电平则计数器递增，如果是低电平计数器递减，T2下降沿计数器变化相反。

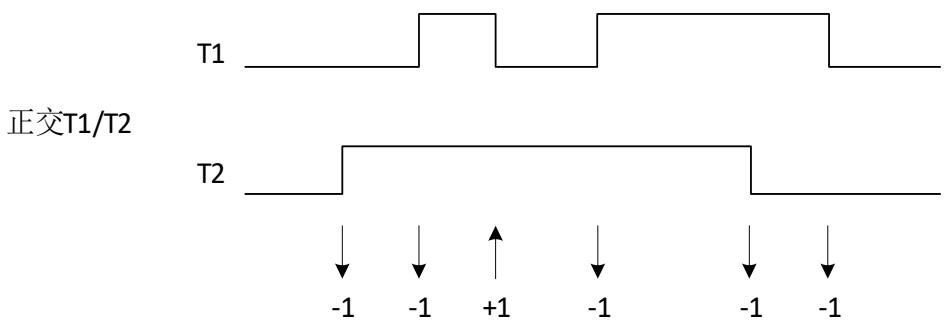
以下式子表示

CounterUp triggering edges) | (T2 triggering edges) Counter Down (T1 triggering edges) | (T2 triggering edges)

表11-1编码器正交编码工作模式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 计数模式 | T1/T2 电平状态(稳态信号) | T1 变化边沿状态 | | T2 变化边沿状态 | |
| 上升沿 | 下降沿 | 上升沿 | 下降沿 |
| 仅 T1 计数 | T2 高 | 递减 | 递增 | 不计数 | 不计数 |
| T2 低 | 递增 | 递减 | 不计数 | 不计数 |
| T1/T2 都计数 | T2 高 | 递减 | 递增 | 不计数 | 不计数 |
| T2 低 | 递增 | 递减 | 不计数 | 不计数 |
| T1 高 | 不计数 | 不计数 | 递增 | 递减 |
| T1 低 | 不计数 | 不计数 | 递减 | 递增 |

  
图11-6编码器只在T1时刻计数的正交编码信号计数情况

  
图11-7编码器在T1或T2时刻计数的正文编码信号计数情况

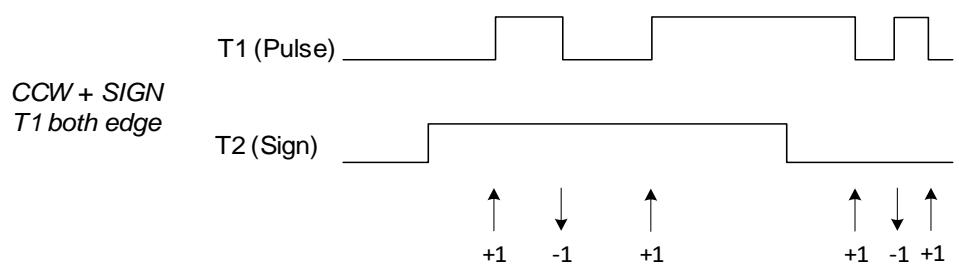
# 11.2.4.2符号加脉冲信号

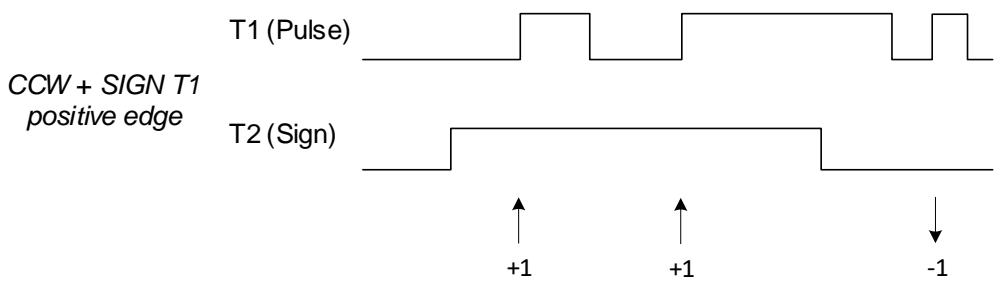
这种工作模式下，T1为脉冲信号，T2为符号信号。T1的边沿触发计数，T2电平控制计数方向，高则递增，低则递减。可以配置仅T1上升沿计数还是T1上升下降沿都计数。

CounterUp @ (T1 triggering edges) Counter Down @ (T1 triggering edges)

表11-2编码器符号加脉冲工作模式

|  |  |  |  |
| --- | --- | --- | --- |
| 计数模式 | T2 电平状态 (稳态信号) | T1 变化边沿状态 | |
| 上升沿 | 下降沿 |
| 仅 T1 上升沿 | 高 | 递增 | 不计数 |
| 低 | 递减 | 不计数 |
| T1 上升下降沿 | 高 | 递增 | 递减 |
| 低 | 递减 | 递增 |

  
图11-8编码器在T1上升下降沿都计数的符号加脉冲信号计数情况

  
图11-9编码器在仅T1上升沿计数的符号加脉冲信号计数情况

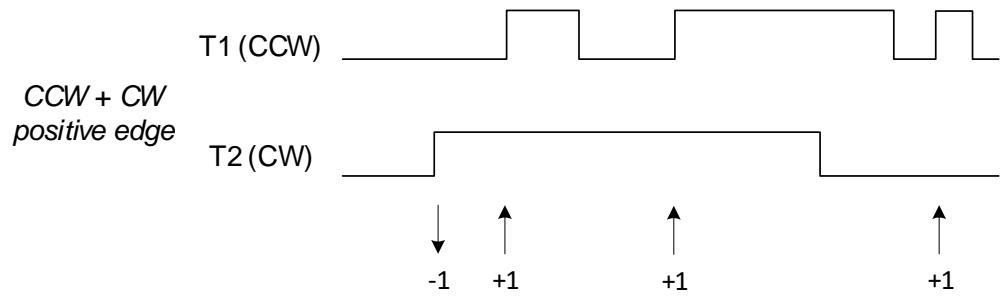
# 11.2.4.3 CCW/CW双脉冲信号

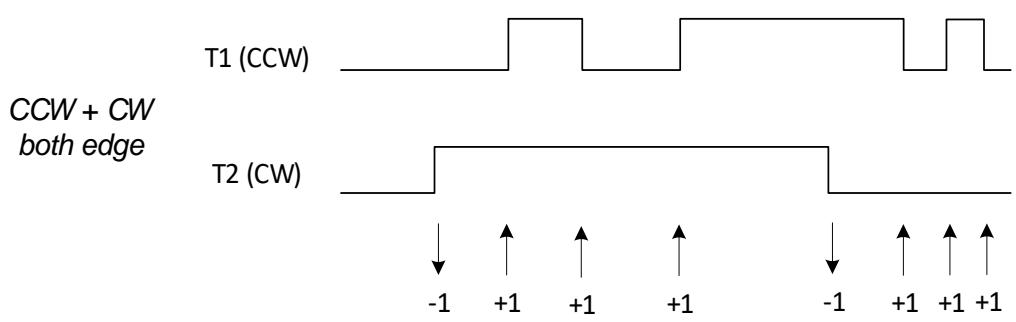
在T1跳变时计数器递增，在T2跳变时计数器递减。可以配置计数器仅在上升沿变化或者在上升下降沿都变化。以下式表示

CounterUp (T1 triggering edges) Counter Down (T2 triggering edges)

表11-3编码器CCW/CW双脉冲工作模式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 计数模式 | 变化边沿状态 | | | |
| T1上升沿 | T1下降沿 | T2上升沿 | T2下降沿 |
| T1/T2上升沿 | 递增 | 不计数 | 递减 | 不计数 |
| T1/T2上升下降沿 | 递增 | 递增 | 递减 | 递减 |

  
图11-10编码器仅在T1/T2上升沿计数的CCW/CW双脉冲信号计数情况

  
图11-11编码器在T1/T2上升下降沿计数的CCW/CW双脉冲信号计数情况

# 11.3 寄存器

# 11.3.1 地址分配

通用定时器模块在芯片中的基地址是0x4001\_1800，寄存器列表如下：

表11-4通用定时器配置寄存器地址分配

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 描述 |

|  |  |  |
| --- | --- | --- |
| UTIMER\_UNTO\_CFG | 0x00 | Timer0 配置寄存器 |
| UTIMER\_UNTO\_TH | 0x04 | Timer0 计数门限寄存器 |
| UTIMER\_UNTO\_CNT | 0x08 | Timer0 计数值寄存器 |
| UTIMER\_UNTO\_CMP0 | 0x0C | Timer0 比较/捕获寄存器 0 |
| UTIMER\_UNTO\_CMP1 | 0x10 | Timer0 比较/捕获寄存器 1 |
| UTIMER\_UNTO\_EVT | 0x14 | Timer0 外部事件选择寄存器 |
| UTIMER\_UNT1\_CFG | 0x20 | Timer1 配置寄存器 |
| UTIMER\_UNT1\_TH | 0x24 | Timer1 计数门限寄存器 |
| UTIMER\_UNT1\_CNT | 0x28 | Timer1 计数值寄存器 |
| UTIMER\_UNT1\_CMP0 | 0x2C | Timer1 比较/捕获寄存器 0 |
| UTIMER\_UNT1\_CMP1 | 0x30 | Timer1 比较/捕获寄存器 1 |
| UTIMER\_UNT1\_EVT | 0x34 | Timer1 外部事件选择寄存器 |
| UTIMER\_UNT2\_CFG | 0x40 | Timer2 配置寄存器 |
| UTIMER\_UNT2\_TH | 0x44 | Timer2 计数门限寄存器 |
| UTIMER\_UNT2\_CNT | 0x48 | Timer2 计数值寄存器 |
| UTIMER\_UNT2\_CMP0 | 0x4C | Timer2 比较/捕获寄存器 0 |
| UTIMER\_UNT2\_CMP1 | 0x50 | Timer2 比较/捕获寄存器 1 |
| UTIMER\_UNT2\_EVT | 0x54 | Timer2 外部事件选择寄存器 |
| UTIMER\_UNT3\_CFG | 0x60 | Timer3 配置寄存器 |
| UTIMER\_UNT3\_TH | 0x64 | Timer3 计数门限寄存器 |
| UTIMER\_UNT3\_CNT | 0x68 | Timer3 计数值寄存器 |
| UTIMER\_UNT3\_CMP0 | 0x6C | Timer3 比较/捕获寄存器 0 |
| UTIMER\_UNT3\_CMP1 | 0x70 | Timer3 比较/捕获寄存器 1 |
| UTIMER\_UNT3\_EVT | 0x74 | Timer3 外部事件选择寄存器 |
| UTIMER\_ECD0\_CFG | 0x80 | Encoder0 配置寄存器 |
| UTIMER\_ECD0\_TH | 0x84 | Encoder0 计数门限寄存器 |
| UTIMER\_ECD0\_CNT | 0x88 | Encoder0 计数值寄存器 |
| UTIMER\_ECD1\_CFG | 0x90 | Encoder1 配置寄存器 |
| UTIMER\_ECD1\_TH | 0x94 | Encoder1 计数门限寄存器 |
| UTIMER\_ECD1\_CNT | 0x98 | Encoder1 计数值寄存器 |
| UTIMER\_FLT\_TH01 | 0xA0 | 滤波门限寄存器 01 |
| UTIMER\_FLT\_TH23 | 0xA4 | 滤波门限寄存器 23 |
| UTIMER\_CFG | 0xF0 | 通用定时器配置寄存器 |
| UTIMER\_IE | 0xF4 | 中断使能寄存器 |
| UTIMER\_IF | 0xF8 | 中断标志寄存器 |
| UTIMER\_RE | 0xFC | DMA 管理寄存器 |

# 11.3.2 系统控制寄存器

# 11.3.2.1 UTIMER\_CFG

地址：0x4001\_18F0

复位值：0x0

表11-5UTIMER配置寄存器UTIMER\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:10] |  | 未使用 |
| [9] | ENC1\_EN | 1: 启动编码器 1, 0: 停止编码器 1 |
| [8] | ENC0\_EN | 1: 启动编码器 0, 0: 停止编码器 0 |
| [7] | TIMER3\_EN | Timer3 使能，当 TIMER3\_EN 为 0 时，Timer3 停止计数 |
| [6] | TIMER2\_EN | Timer2 使能，当 TIMER2\_EN 为 0 时，Timer2 停止计数 |
| [5] | TIMER1\_EN | Timer1 使能，当 TIMER1\_EN 为 0 时，Timer1 停止计数 |
| [4] | TIMER0\_EN | Timer0 使能，当 TIMER0\_EN 为 0 时，Timer0 停止计数 |
| [3:0] |  | 系统保留，推荐写入 0 |

当Timer停止计数时，Timer的寄存器配置不受影响，仍然保持；且在停止计数时仍可以通过软件修改Timer的寄存器配置。

# 11.3.3 滤波控制寄存器

# 11.3.3.1UTIMER\_FLT\_TH01

地址：0x4001\_18A0

复位值：0x0

表11-6滤波控制寄存器UTIMER\_FLT\_TH01

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T1\_CH1\_FLT |  |  | T1\_CH0\_FLT |  |  |  | T0\_CH1\_FLT |  |  |  | T0\_CH0\_FLT |  |  |  |  |
| RW |  |  | RW |  |  |  | RW |  |  |  | RW |  |  |  |  |
| 0 |  |  | 0 |  |  |  | 0 |  |  |  | 0 |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | T1\_CH1\_FLT | TIM1\_CH1 信号滤波宽度选择，取值范围 0～15。 [15:12]为 0 时，对 TIM1\_CH1 不进行滤波。 [15:12]不为 0 时，对 TIM1\_CH1 信号进行滤波：滤波宽度为 T1\_CH1\_FLT×8。当 TIM1\_CH1 电平稳定超过 T1\_CH1\_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM1\_CH1 信号值；否则，滤波器保 |

|  |  |  |
| --- | --- | --- |
|  |  | 持当前的输出不变。 |
| [11:8] | T1\_CH0\_FLT | TIM1\_CH0 信号滤波宽度选择。含义同 T1\_CH1\_FLT。 |
| [7:4] | T0\_CH1\_FLT | TIM0\_CH1 信号滤波宽度选择。含义同 T1\_CH1\_FLT。 |
| [3:0] | T0\_CH0\_FLT | TIM0\_CH0 信号滤波宽度选择。含义同 T1\_CH1\_FLT。 |

# 11.3.3.2UTIMER\_FLT\_TH23

地址：0x4001\_18A4

复位值：0x0

表11-7滤波控制寄存器UTIMER\_FLT\_TH23

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T3\_CH1\_FLT | | | | T3\_CH0\_FLT | | | T2\_CH1\_FLT | | | T2\_CH0\_FLT | | | | |  |
| RW | | | | RW | | | RW | | | RW | | | | |  |
| 0 | | | | 0 | | | 0 | | | 0 | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:12] | T3\_CH1\_FLT | TIM3\_CH1 信号滤波宽度选择，取值范围 0~15。 [15:12]为 0 时，对 TIM3\_CH1 不进行滤波。 [15:12]不为 0 时，对 TIM3\_CH1 信号进行滤波：滤波宽度为 T3\_CH1\_FLT×8。当 TIM3\_CH1 电平稳定超过 T3\_CH1\_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM3\_CH1 信号值；否则，滤波器保持当前的输出不变。 |
| [11:8] | T3\_CH0\_FLT | TIM3\_CH0 信号滤波宽度选择。含义同 T3\_CH1\_FLT。 |
| [7:4] | T2\_CH1\_FLT | TIM2\_CH1 信号滤波宽度选择。含义同 T3\_CH1\_FLT。 |
| [3:0] | T2\_CH0\_FLT | TIM2\_CH0 信号滤波宽度选择。含义同 T3\_CH1\_FLT。 |

# 11.3.4 定时器寄存器

Timer0/1完全相同，此处仅说明Timer0寄存器

Timer2/3完全相同，与Timer0/1不同之处在于Timer2/3计数器相关寄存器为32位宽，而Timer0/1计数器相关寄存器为16位宽。

Encoder0复用了Timer2的输入端口，Encoder1复用了Timer3的输入端口；开启Encoder功能时，并不影响对应Timer的正常使用。

# 11.3.4.1UTIMER\_UNTO\_CFGTimer0配置寄存器

地址：0x4001\_1800

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
|  |  | ETON |  | CLK\_DIV |  | CH1\_POL |  | CH1\_MODE |  | CH1\_RE\_CAP\_EN |  | CH1\_POL |  | CH0\_MODE |  | CH0\_RE\_CAP\_EN |  |
| RW | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  |
| 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |
| [11] | ETON | Timer0 计数器计数使能来源。 0，自动运行，持续计数； 1，等待外部事件触发计数一个 Timer 周期后停止。 外部事件通过 UTM ER\_UNTO\_EVT 寄存器进行配置。 |
| [10] |  | 系统保留，恒 0。 |
| [9:8] | CLK\_DIV | Timer0 计数器频率配置，计数器计数频率是系统主时钟频率的 1/2/4/8 分频： 00：1 分频，01：2 分频，10：4 分频，11：8 分频 |
| [7] | CH1\_POL | 通道 1 在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |
| [6] | CH1\_MODE | 通道 1 的工作模式： 0，比较模式，输出方波，在通道 1 计数器计数值等于 0 或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道 1 输入信号发生捕获事件时，将计数器计数值存入通道 1 比较捕获寄存器。 |
| [5] | CH1\_FE\_CAP\_EN | 通道 1 下降沿捕获事件使能。1：使能；0：关闭。 通道 1 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [4] | CH1\_RE\_CAP\_EN | 通道 1 上升沿捕获事件使能。1：使能；0：关闭。 通道 1 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |
| [3] | CH0\_POL | 通道 0 在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |
| [2] | CH0\_MODE | 通道 0 的工作模式： 0，比较模式，输出方波，在通道 0 计数器计数值等于 0 或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道 0 输入信号发生捕获事件时，将计数器计数值存入通道 0 比较捕获寄存器。 |
| [1] | CH0\_FE\_CAP\_EN | 通道 0 下降沿捕获事件使能。1：使能；0：关闭。 通道 0 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [0] | CH0\_RE\_CAP\_EN | 通道 0 上升沿捕获事件使能。1：使能；0：关闭。 |

|  |  |  |
| --- | --- | --- |
|  |  | 通道0输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |

# 11.3.4.2UTIMER\_UNT1\_CFGTimer1配置寄存器

地址：0x4001\_1820

复位值：0x0

表11-9Timer0配置寄存器UTIMER\_UNT0\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  | ETON |  | CLK\_DIV |  | CH1\_POL |  | CH1\_MODE |  | CH1\_FE\_CAP\_EN |  | CH1\_RE\_CAP\_EN |  | CH0\_POL |  | CH0\_MODE |  | CH0\_FE\_CAP\_EN |  | CH0\_RE\_CAP\_EN |  |
| RW | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  |  |  |
| 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |
| [11] | ETON | Timer1计数器计数使能来源。 0，自动运行，持续计数； 1，等待外部事件触发计数一个Timer周期后停止。 外部事件通过UTIMER\_UNT1\_EVT寄存器进行配置。 |
| [10] |  | 系统保留，恒0。 |
| [9:8] | CLK\_DIV | Timer1计数器频率配置，计数器计数频率是系统主频率的1/2/4/8分频： 00：1分频，01：2分频，10：4分频，11：8分频 |
| [7] | CH1\_POL | 通道1在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |
| [6] | CH1\_MODE | 通道1的工作模式： 0，比较模式，输出方波，在通道1计数器计数值等于0或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道1输入信号发生捕获事件时，将计数器计数值存入通道1比较捕获寄存器。 |
| [5] | CH1\_FE\_CAP\_EN | 通道1下降沿捕获事件使能。1：使能；0：关闭。 通道1输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [4] | CH1\_RE\_CAP\_EN | 通道1上升沿捕获事件使能。1：使能；0：关闭。 通道1输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |
| [3] | CH0\_POL | 通道0在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |

|  |  |  |
| --- | --- | --- |
| [2] | CH0\_MODE | 通道0的工作模式， 0，比较模式，输出方波，在通道0计数器计数值等于0或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道0输入信号发生捕获事件时，将计数器计数值存入通道0比较捕获寄存器。 |
| [1] | CH0\_FE\_CAP\_EN | 通道0下降沿捕获事件使能。1：使能；0：关闭。 通道0输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [0] | CH0\_RE\_CAP\_EN | 通道0上升沿捕获事件使能。1：使能；0：关闭。 通道0输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |

# 11.3.4.3UTIMER\_UNT2\_CFGTimer2配置寄存器

地址：0x4001\_1840

复位值：0x0

表11-10Timer2配置寄存器UTIMER\_UNT2\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | ETON | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  | RW |  | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |
|  |  |  |  | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |
| [11] | ETON | 0，自动运行，持续计数； 1，等待外部事件触发计数一个 Timer 周期后停止。 外部事件通过 UTM ER\_UNT2\_EVT 寄存器进行配置。 |
| [10] |  | 系统保留，恒 0。 |
| [9:8] | CLK\_DIV | Timer2 计数器频率配置，计数器计数频率是系统主频率的 1/2/4/8 分频： 00：1 分频，01：2 分频，10：4 分频，11：8 分频 |
| [7] | CH1\_POL | 通道 1 在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |
| [6] | CH1\_MODE | 通道 1 的工作模式： 0，比较模式，输出方波，在通道 1 计数器计数值等于 0 或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道 1 输入信号发生捕获事件时，将计数器计数值存入通道 1 比较捕获寄存器。 |

|  |  |  |
| --- | --- | --- |
| [5] | CH1\_FE\_CAP\_EN | 通道1下降沿捕获事件使能。1：使能；0：关闭。 通道1输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [4] | CH1\_RE\_CAP\_EN | 通道1上升沿捕获事件使能。1：使能；0：关闭。 通道1输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |
| [3] | CH0\_POL | 通道0在比较模式下的输出极性控制：当计数器计数值回零时的输出值。 |
| [2] | CH0\_MODE | 通道0的工作模式， 0，比较模式，输出方波，在通道0计数器计数值等于0或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道0输入信号发生捕获事件时，将计数器计数值存入通道0比较捕获寄存器。 |
| [1] | CH0\_FE\_CAP\_EN | 通道0下降沿捕获事件使能。1：使能；0：关闭。 通道0输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [0] | CH0\_RE\_CAP\_EN | 通道0上升沿捕获事件使能。1：使能；0：关闭。 通道0输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |

# 11.3.4.4UTIMER\_UNT3\_CFGTimer3配置寄存器

地址：0x4001\_1860

复位值：0x0

表11-11Timer3配置寄存器UTIMER\_UNT3\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  | ETON |  | CLK\_DIV |  | CH1\_POL |  | CH1\_MODE |  | CH1\_FE\_CAP\_EN |  | CH1\_RE\_CAP\_EN |  | CH0\_POL |  | CH0\_MODE |  | CH0\_FE\_CAP\_EN |  | CH0\_RE\_CAP\_EN |  |
| RW | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  | RW |  |  |  |
| 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |
| [11] | ETON | 0，自动运行，持续计数； 1，等待外部事件触发计数一个 Timer 周期后停止。 外部事件通过 UTIMER\_UNT3\_EVT 寄存器进行配置。 |
| [10] |  | 系统保留，恒 0。 |
| [9:8] | CLK\_DIV | Timer3 计数器频率配置，计数器计数频率是系统主频率的 1/2/4/8 分频： |

|  |  |  |
| --- | --- | --- |
|  |  | 00:1分频, 01:2分频, 10:4分频, 11:8分频 |
| [7] | CH1\_POL | 通道1在比较模式下的输出极性控制:当计数器计数值回零时的输出值。 |
| [6] | CH1\_MODE | 通道1的工作模式: 0,比较模式,输出方波,在通道1计数器计数值等于0或等于比较捕获寄存器值时发生翻转。 1,捕获模式,当通道1输入信号发生捕获事件时,将计数器计数值存入通道1比较捕获寄存器。 |
| [5] | CH1\_FE\_CAP\_EN | 通道1下降沿捕获事件使能。1:使能;0:关闭。 通道1输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [4] | CH1\_RE\_CAP\_EN | 通道1上升沿捕获事件使能。1:使能;0:关闭。 通道1输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |
| [3] | CH0\_POL | 通道0在比较模式下的输出极性控制:当计数器计数值回零时的输出值。 |
| [2] | CH0\_MODE | 通道0的工作模式, 0,比较模式,输出方波,在通道0计数器计数值等于0或等于比较捕获寄存器值时发生翻转。 1,捕获模式,当通道0输入信号发生捕获事件时,将计数器计数值存入通道0比较捕获寄存器。 |
| [1] | CH0\_FE\_CAP\_EN | 通道0下降沿捕获事件使能。1:使能;0:关闭。 通道0输入信号发生1→0跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。 |
| [0] | CH0\_RE\_CAP\_EN | 通道0上升沿捕获事件使能。1:使能;0:关闭。 通道0输入信号发生0→1跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。 |

在LKS32MC08x系列中，Timer0/1为16bit位宽；Timer2/3为32bit位宽，因此Timer2/3下列相关寄存器为32bit位宽。

# 11.3.4.5UTIMER\_UNTO\_THTimer0门限寄存器

地址：0x4001\_1804

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNTO\_TH | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:16] |  | 未使用 |
| [15:0] | UNTO\_TH | Timer 0 计数器计数门限。计数器从 0 计数到 UTIMER\_UNTO\_TH 值后再次回 0 开始计数。 |

# 11.3.4.6UTIMER\_UNT1\_THTimer1门限寄存器

地址：0x4001\_1824

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT1\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNT1\_TH | Timer 1 计数器计数门限。计数器从 0 计数到 UTIMER\_UNT1\_TH 值后再次回 0 开始计数。 |

11.3.4.7UTIMER\_UNT2\_THTimer2门限寄存器

地址：0x4001\_1844

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 10 | 19 | 18 | 17 | 16 |
| UNT2\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT2\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT2\_TH | Timer 2 计数器计数门限。计数器从 0 计数到 UTIMER\_UNT2\_TH 值后再次回 0 开始计数。 |

# 11.3.4.8UTIMER\_UNT3\_THTimer3门限寄存器

地址：0x4001\_1864

复位值：0x0

表11.15Timer3门限寄存器UTIMER\_UNT3\_TH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNT3\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT3\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT3\_TH | Timer 3 计数器计数门限。计数器从 0 计数到 UTIMER\_UNT3\_TH 值后再次回 0 开始计数。 |

# 11.3.4.9UTIMER\_UNTO\_CNTTimer0计数寄存器

地址：0x4001\_1808

复位值：0x0

表11-16Timer0计数寄存器UTIMER\_UNTO\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNTO\_CNT | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNTO\_CNT | Timer 0 计数器当前计数值。写操作可以写入新的计数值。 |

# 11.3.4.10UTIMER\_UNT1\_CNTTimer1计数寄存器

地址：0x4001\_1828

复位值：0x0

表11-17Timer1计数寄存器UTIMER\_UNT1\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT1\_CNT | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [S1:16] |  | 未使用 |
| [15:0] | UNT1\_CNT | Timer 1 计数器当前计数值。写操作可以写入新的计数值。 |

# 11.3.4.11UTIMER\_UNT2\_CNTTimer2计数寄存器

地址：0x4001\_1848

复位值：0x0

表11-18Timer2计数寄存器UTIMER\_UNT2\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 10 | 19 | 18 | 17 | 16 |  |  |
| UNT2\_CNT | | | | | | | | | | | | | | |  |  |  |
| RW | | | | | | | | | | | | | | |  |  |  |
| 0 | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT2\_CNT | | | | | | | | | | | | | | |  |  |  |
| RW | | 0 | | | | | | | | | | | | | |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT2\_CNT | Timer 2 计数器当前计数值。写操作可以写入新的计数值。 |

# 11.3.4.12UTIMER\_UNT3\_CNTTimer3计数寄存器

地址：0x4001\_1868

复位值：0x0

表11-19Timer3计数寄存器UTIMER\_UNT3\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 10 | 19 | 18 | 17 | 16 |
| UNT3\_CNT | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT3\_CNT | | | | | | | | | | | | | | |  |

|  |
| --- |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT3\_CNT | Timer3 计数器当前计数值。写操作可以写入新的计数值。 |

11.3.4.13UTIMER\_UNTO\_CMP0Timer0通道0比较捕获寄存器

地址：0x4001\_180C

复位值：0x0

表11- 20Timer0通道0比较捕获寄存器UTIMER\_UNTO\_CMP0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNTO\_CMP0 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNTO\_CMP0 | Timer 0 通道 0 工作在比较模式时，当计数器计数值等于UTIMER\_UNTO\_CMP0时，发生比较事件。 Timer 0 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入UTIMER\_UNTO\_CMP0寄存器。 |

11.3.4.14UTIMER\_UNTO\_CMP1Timer0通道1比较捕获寄存器

地址：0x4001\_1810

复位值：0x0

表11- 21Timer0通道1比较捕获寄存器UTIMER\_UNTO\_CMP1

|  |  |
| --- | --- |
| UNTO\_CMP1 | |
| RW | |
| 0 | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNTO\_CMP1 | Timer 0 通道 1 工作在比较模式时，当计数器计数值等于UTIMER\_UNTO\_CMP1时，发生比较事件。 |

|  |  |  |
| --- | --- | --- |
|  |  | Timer 0 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER\_UNTO\_CMP1 寄存器。 |

11.3.4.15UTIMER\_UNT1\_CMP0Timer1通道0比较捕获寄存器

地址：0x4001\_182C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表11-22 Timer 1 通道 0 比较捕获寄存器 UTIMER\_UNT1\_CMP0 | | | | | | | | | | | | |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT1\_CMP0 | | | | | | | | | | | | |  |  |  |
| RW | | | | | | | | | | | | |  |  |  |
| 0 | | | | | | | | | | | | |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNT1\_CMP0 | Timer 1 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER\_UNT1\_CMP0 时，发生比较事件。 Timer 1 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER\_UNT1\_CMP0 寄存器。 |

11.3.4.16UTIMER\_UNT1\_CMP1Timer1通道1比较捕获寄存器

地址：0x4001\_1830

复位值：0x0

表11- 23Timer1通道1比较捕获寄存器UTIMER\_UNT1\_CMP1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT1\_CMP1 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | UNT1\_CMP1 | Timer 1 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER\_UNT1\_CMP1 时，发生比较事件。 Timer 1 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER\_UNT1\_CMP1 寄存器。 |

# 11.3.4.17UTIMER\_UNT2CMPOTimer2通道0比较捕获寄存器

地址：0x4001\_184C

复位值：0x0

表11-24Timer2通道0比较捕获寄存器UTIMER\_UNT2CMP0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNT2\_CMP0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT2\_CMP0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT2\_CMP0 | Timer 2 通道 0 工作在比较模式时，当计数器计数值等于 UTM ER\_UNT2\_CMP0 时，发生比较事件。 Timer 2 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTM ER\_UNT2\_CMP0 寄存器。 |

11.3.4.18UTIMER\_UNT2CMP1Timer2通道1比较捕获寄存器

地址：0x4001\_1850

复位值：0x0

表11-25Timer2通道1比较捕获寄存器UTIMER\_UNT2CMP1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 0 | 19 | 18 | 17 | 16 |
| UNT2\_CMP1 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <lcel> | | | | | | | | | | | | | | | |
| <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <fcel> | <nl> |
| <lcel> | | | | | | | | | | | | | | | |
| <lcel> | | | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <lcel> | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT2\_CMP1 | Timer 2 通道 1 工作在比较模式时，当计数器计数值等于 UTM ER\_UNT2\_CMP1 时，发生比较事件。 Timer 2 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTM ER\_UNT2\_CMP1 寄存器。 |

# 11.3.4.19UTIMER\_UNT3CMPOTimer3通道0比较捕获寄存器

地址：0x4001\_186C

复位值：0x0

表11-26Timer3通道0比较捕获寄存器UTIMER\_UNT3CMP0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNT3\_CMP0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNT3\_CMP0 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT3\_CMP0 | Timer 3 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER\_UNT3\_CMP0 时，发生比较事件。 Timer 3 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER\_UNT3\_CMP0 寄存器。 |

11.3.4.20UTIMER\_UNT3CMP1Timer3通道1比较捕获寄存器

地址：0x4001\_1870

复位值：0x0

表11-27Timer3通道1比较捕获寄存器UTIMER\_UNT3CMP1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNT3\_CMP1 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <lcel> | | | | | | | | | | | | | | | |
| <fcel> | <fcel> | <lcel> | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | UNT3\_CMP1 | Timer 3 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER\_UNT3\_CMP1 时，发生比较事件。 |

|  |  |  |
| --- | --- | --- |
|  |  | Timer 3 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER\_UNT3\_CMP1 寄存器。 |

11.3.4.21UTIMER\_UNTO\_EVTTimer0外部事件选择寄存器

地址：0x4001\_1814

复位值：0x0

表11-28Timer0外部事件选择寄存器UTIMER\_UNT0\_EVT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | EVT\_SRC | | |  |
| RW | | |  |
| 0 | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:3] |  | 未使用 |
| [2:0] | EVT\_SRC | Timer0 外部事件选择寄存器，本寄存器需要配合UTIMER\_UNTO\_CFG[11]使用。U0CFG[11]为高时，根据本寄存器选择触发Timer0计数的事件。 0: MCPWM TADC[2]比较事件 1: MCPWM TADC[3]比较事件 2: TIMER1 通道0比较事件 3: TIMER1 通道1比较事件 4: TIMER2 通道0比较事件 5: TIMER2 通道1比较事件 6: TIMER3 通道0比较事件 7: TIMER3 通道1比较事件 |

11.3.4.22UTIMER\_UNT1\_EVTTimer1外部事件选择寄存器

地址：0x4001\_1834

复位值：0x0

表11-29Timer1外部事件选择寄存器UTIMER\_UNT1\_EVT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  | | | | | | |  | | | | | | | | | | | EVT\_SRC | | |
| RW | | |
| 0 | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:3] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [2:0] | EVT\_SRC | Timer1 外部事件选择寄存器，本寄存器需要配合UTIMER\_UNT1\_CFG[11]使用。UTIMER\_UNT1\_CFG[11]为高时，根据本寄存器选择触发 Timer1 计数的事件。 0：TIMER0 通道 0 比较事件 1：TIMER0 通道 1 比较事件 2：MCPWM TADC[2]比较事件 3：MCPWM TADC[3]比较事件 4：TIMER2 通道 0 比较事件 5：TIMER2 通道 1 比较事件 6：TIMER3 通道 0 比较事件 7：TIMER3 通道 1 比较事件 |

# 11.3.4.23UTIMER\_UNT2\_EVTTimer2外部事件选择寄存器

地址：0x4001\_1854

复位值：0x0

表11-30Timer2外部事件选择寄存器UTIMER\_UNT2\_EVT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:3] |  | 未使用 |
| [2:0] | EVT\_SRC | Timer2 外部事件选择寄存器，本寄存器需要配合UTIMER\_UNT2\_CFG[11]使用。UTIMER\_UNT2\_CFG[11]为高时，根据本寄存器选择触发 Timer2 计数的事件。 0：TIMER0 通道 0 比较事件 1：TIMER0 通道 1 比较事件 2：TIMER1 通道 0 比较事件 3：TIMER1 通道 1 比较事件 4：MCPWM TADC[2]比较事件 5：MCPWM TADC[3]比较事件 6：TIMER3 通道 0 比较事件 7：TIMER3 通道 1 比较事件 |

# 11.3.4.24UTIMER\_UNT3\_EVTTimer3外部事件选择寄存器

地址：0x4001\_1874

复位值：0x0

表11-31Timer3外部事件选择寄存器UTIMER\_UNT3\_EVT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [S1:3] |  | 未使用 |
| [2:0] | EVT\_SRC | Timer3 外部事件选择寄存器，本寄存器需要配合UTIMER\_UNT3\_CFG[11]使用。UTIMER\_UNT3\_CFG[11]为高时，根据本寄存器选择触发 Timer3 计数的事件。 0：TIMER0 通道 0 比较事件 1：TIMER0 通道 1 比较事件 2：TIMER1 通道 0 比较事件 3：TIMER1 通道 1 比较事件 4：TIMER2 通道 0 比较事件 5：TIMER2 通道 1 比较事件 6：MCPWM TADC[2]比较事件 7：MCPWM TADC[3]比较事件 |

# 11.3.5 编码器寄存器

Encoder0/1功能完全相同仅寄存器地址分配不同。

11.3.5.1UTIMER\_ECDx\_CFGEncoderX配置寄存器

UTIMER\_ECD0\_CFG地址：0x4001\_1880

UTIMER\_ECD1\_CFG地址：0x4001\_1890

复位值：0x0

表11-32EncoderX配置寄存器UTIMER\_ECDx\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | FE\_CNT\_EN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:11] |  | 未使用 |
| [10] | FE\_CNT\_EN | CCW+SIGN/CCW+CW 两种模式下，是否在下降沿进行计数（上升沿总是计数） |

|  |  |  |
| --- | --- | --- |
| [9:8] | MODE | EncoderX编码器模式选择 00: counting on T1, 01: counting on T1 &amp; T2 以上两种模式都为正交编码信号计数模式 10: CCW+SIGN,符号加脉冲信号计数模式 11: CCW+CW,CCW+CW双脉冲信号计数模式 |
| [7:0] |  | 未使用 |

11.3.5.2UTIMER\_ECDx\_THEncoderX计数门限寄存器

UTIMER\_ECD0\_TH地址：0x4001\_1884

UTIMER\_ECD1\_TH地址：0x4001\_1894

复位值：0x0

表11-33EncoderX计数门限寄存器UTIMER\_ECDx\_TH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ECD0\_TH | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | ECDx\_TH | EncoderX 计数门限 TH。编码器向上计数（增）到 TH 值后，再次向上计数会导致计数器回到 0。编码器向下计数（减）到 0 值后，再次向下计数会导致计数器回到 TH。 |

11.3.5.3UTIMER\_ECDxCNTEncoderX计数值寄存器

UTIMER\_ECD0\_CNT地址：0x4001\_1888

UTIMER\_ECD1\_CNT地址：0x4001\_1898

复位值：0x0

表11-34Encoder0计数值寄存器UTIMER\_ECD0\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ECD0\_CNT | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | ECDx\_CNT | EncoderX 计数值。 |

# 11.3.6 中断管理寄存器

中断管理寄存器包括中断标志寄存器UTIMER\_IF和中断使能寄存器UTIMER\_IE。两个寄存器各个比特对应相同的中断。

# 11.3.6.1UTIMER\_IE中断使能寄存器

地址：0x4001\_18F4

复位值：0x0

表11-35中断使能寄存器UTIMER\_IE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |
| ENC1\_OF\_IE |  | ENC1\_OF\_IE |  | ENC0\_OF\_IE |  | T3\_CH1\_IE |  | T3\_CH0\_IE |  | T2\_ZC\_IE |  | T1\_CH1\_IE |  | T1\_ZC\_IE |  | T0\_CH1\_IE |  | T0\_CH0\_IE |  | T0\_ZC\_IE |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | R |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | ENC1\_OF\_IE | Encoder1上溢出中断使能，高电平有效。当Encoder1计数器计数达到计数门限时，上计数事件触发上溢出中断。 |
| [14] | ENC1\_UF\_IE | Encoder1下溢出中断使能，高电平有效。当Encoder1计数器计数达到0时，下计数事件触发下溢出中断。 |
| [13] | ENC0\_OF\_IE | Encoder0上溢出中断使能，高电平有效。 |
| [12] | ENC0\_UF\_IE | Encoder0下溢出中断使能，高电平有效。 |
| [11] | T3\_CH1\_IE | Timer3通道1比较/捕获中断使能，高电平有效。 |
| [10] | T3\_CH0\_IE | Timer3通道0比较/捕获中断使能，高电平有效。 |
| [9] | T3\_ZC\_IE | Timer3计数器过0中断使能，高电平有效。 |
| [8] | T2\_CH1\_IE | Timer2通道1比较/捕获中断使能，高电平有效。 |
| [7] | T2\_CH0\_IE | Timer2通道0比较/捕获中断使能，高电平有效。 |
| [6] | T2\_ZC\_IE | Timer2计数器过0中断使能，高电平有效。 |
| [5] | T1\_CH1\_IE | Timer1通道1比较/捕获中断使能，高电平有效。 |
| [4] | T1\_CH0\_IE | Timer1通道0比较/捕获中断使能，高电平有效。 |
| [3] | T1\_ZC\_IE | Timer1计数器过0中断使能，高电平有效。 |
| [2] | T0\_CH1\_IE | Timer0通道1比较/捕获中断使能，高电平有效。 |
| [1] | T0\_CH0\_IE | Timer0通道0比较/捕获中断使能，高电平有效。 |
| [0] | T0\_ZC\_IE | Timer0计数器过0中断使能，高电平有效。 |

# 11.3.6.2UTIMER\_IF中断标志寄存器

地址：0x4001\_18F8

复位值：0x0

表11-36中断标志寄存器UTIMER\_IF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ENC1\_OF\_IF | ENC1\_OF\_IF | ENC1\_OF\_IF | ENC0\_OF\_IF | ENC0\_OF\_IF | T3\_OF\_IF | T3\_OF\_0\_IF | T3\_OF\_C\_F | T2\_OF\_IF | T2\_OF\_0\_IF | T2\_OF\_C\_F | T1\_OF\_IF | T1\_OF\_0\_IF | T1\_OF\_C\_F | T0\_OF\_IF | T0\_OF\_0\_IF |
| RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | ENC1\_OF\_IF | Encoder1 上溢出中断标志，高电平有效，对此 bit 写 1 可清 0 此 bit。当 Encoder1 计数器计数达到计数门限时，上计数事件触发上溢出中断。 |
| [14] | ENC1\_OF\_IF | Encoder1 下溢出中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。当 Encoder1 计数器计数达到 0 时，下计数事件触发下溢出中断。 |
| [13] | ENC0\_OF\_IF | Encoder0 上溢出中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。当 Encoder0 计数器计数达到计数门限时，上计数事件触发上溢出中断。 |
| [12] | ENC0\_UF\_IF | Encoder0 下溢出中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。当 Encoder0 计数器计数达到 0 时，下计数事件触发下溢出中断。 |
| [11] | T3\_CH1\_IF | Timer3 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [10] | T3\_CH0\_IF | Timer3 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [9] | T3\_ZC\_IF | Timer3 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [8] | T2\_CH1\_IF | Timer2 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [7] | T2\_CH0\_IF | Timer2 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [6] | T2\_ZC\_IF | Timer2 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [5] | T1\_CH1\_IF | Timer1 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [4] | T1\_CH0\_IF | Timer1 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [3] | T1\_ZC\_IF | Timer1 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [2] | T0\_CH1\_IF | Timer0 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 |

|  |  |  |
| --- | --- | --- |
|  |  | 此 bit。 |
| [1] | T0\_CH0\_IF | Timer0 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |
| [0] | T0\_ZC\_IF | Timer0 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。 |

其中Timer2通道0/1、Timer3通道0/1的比较事件可作为ADC采样触发事件UTimer\_T0/UTimer\_T1/UTimer\_T2/UTimer\_T3;

此4个事件与MCPWM产生的MCPWM\_T0/MCPWM\_T1/MCPWM\_T2/MCPWM\_T3在ADC内部经过使能控制后分别相或得到4个ADC采样触发事件TADC[0]/TADC[1]/TADC[2]/TADC[3]。

中断标志写1清零，一般不建议用如下 方式清零，因为 是先读取中断标志，将对应位改为1再写入清零，如果同时有其他中断标志置位，会被一起清零，而这通常不是软件所期望的。例如，如下写法本意是清零T0\_ZC\_IF，但如果同时T0\_CH0\_IF在写入执行前置1了，则软件先读取回UTIMER\_IF值为 ，然后执行或操作 ，然后写入，同时对T0\_CH0\_IF和T0\_ZC\_IF进行了清零，可能导致Timer少进入一次因捕获而产生的中断。

如果希望清零T0\_ZC\_IF标志位，应以如下方式，直接对BIT0写1.

# 11.3.7 DMA管理寄存器

中断管理寄存器包括中断标志寄存器IF和中断使能寄存器IE。两个寄存器各个比特对应相同的中断。

# 11.3.7.1UTIMER\_REDMA请求使能寄存器

地址：0x4001\_18FC

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | T3\_CH0\_DMA\_RE |  |  |  |  |  |  |  |  |  |  |  |  |  |
| T3\_CH0\_DMA\_RE |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:12] |  | 未使用 |

# LKS32MC08x User Manual

|  |  |  |
| --- | --- | --- |
| [11] | T3\_CH1\_DMA\_RE | Timer3 通道 1 比较/捕获 DMA 请求使能，高电平有效。 |
| [10] | T3\_CH0\_DMA\_RE | Timer3 通道 0 比较/捕获 DMA 请求使能，高电平有效。 |
| [9] | T3\_DMA\_RE | Timer3 计数器过 0 DMA 请求使能，高电平有效。 |
| [8] | T2\_CH1\_DMA\_RE | Timer2 通道 1 比较/捕获 DMA 请求使能，高电平有效。 |
| [7] | T2\_CH0\_DMA\_RE | Timer2 通道 0 比较/捕获 DMA 请求使能，高电平有效。 |
| [6] | T2\_DMA\_RE | Timer2 计数器过 0 DMA 请求使能，高电平有效。 |
| [5] | T1\_CH1\_DMA\_RE | Timer1 通道 1 比较/捕获 DMA 请求使能，高电平有效。 |
| [4] | T1\_CH0\_DMA\_RE | Timer1 通道 0 比较/捕获 DMA 请求使能，高电平有效。 |
| [3] | T1\_DMA\_RE | Timer1 计数器过 0 DMA 请求使能，高电平有效。 |
| [2] | T0\_CH1\_DMA\_RE | Timer0 通道 1 比较/捕获 DMA 请求使能，高电平有效。 |
| [1] | T0\_CH0\_DMA\_RE | Timer0 通道 0 比较/捕获 DMA 请求使能，高电平有效。 |
| [0] | T0\_DMA\_RE | Timer0 计数器过 0 DMA 请求使能，高电平有效。 |

# 12HALL信号处理模块

# 12.1综述

芯片共支持3路HALL信号输入。

对于输入的HALL传感器信号，所进行的处理包括：

滤波，消除HALL信号毛刺的影响

捕获，当HALL输入有变化时，记录当前的定时器值，并输出中断

溢出，当HALL信号长时间不发生变化导致计数器溢出时，输出中断

# 12.2 实现说明

# 12.2.1 信号来源

HALL信号来源于GPIO，对于每一路HALL信号，芯片有两个IO可以作为该信号的来源。通过配置GPIO寄存器，用户可以选择将其中一个GPIO的输入信号做为HALL信号使用。

详细管脚位置说明见芯片器件datasheet。

# 12.2.2 工作时钟

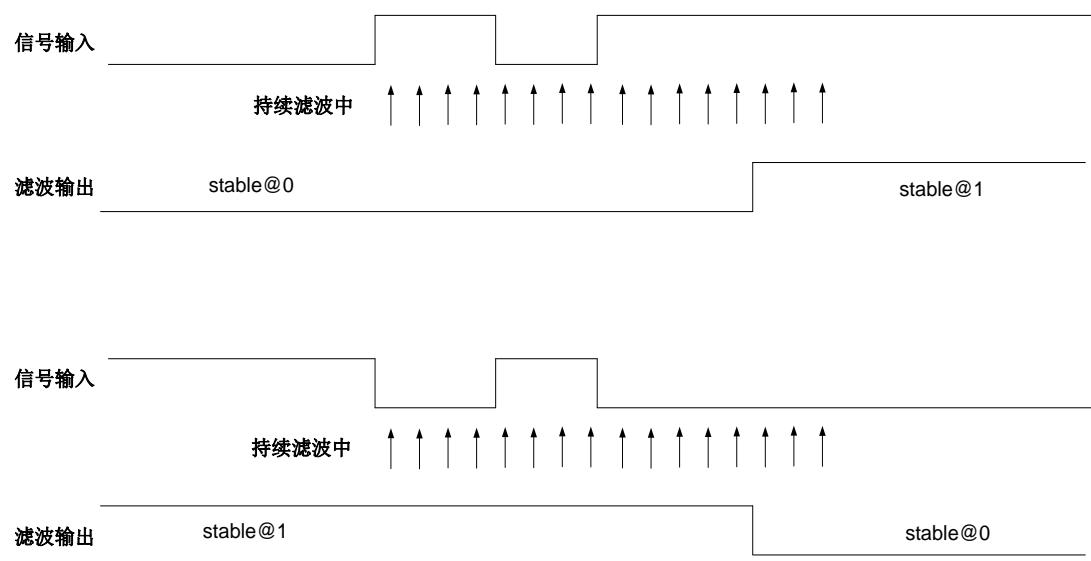
12.2.2 工作时钟HALL模块工作频率可调。通过配置HALL\_CFG.CLK\_DIV寄存器，可以选择系统主时钟的1/2/4/8分频作为HALL模块工作频率，滤波和计数均采用该频率工作。

# 12.2.3 信号滤波

滤波模块主要用于去除HALL信号上的毛刺。

滤波包括两级滤波器：

第一级采用7判5进行滤波，即连续7个采样点中，如果达到超过5个1则输出1，如果达到或超过5个0则输出0，否则输出保持上一次的滤波结果。具体如下图所示：

  
图12-17/5滤波模块框图

第二级采用连续滤波，在连续N个采样点中，如全为0则输出0，如全为1则输出1，否则输出保持上一次的滤波结果。

通过配置HALL\_CFG.FIL\_75可以选择是否使能第一级滤波器。

通过配置HALL\_CFG.FIL\_LEN可以配置第二级滤波器滤波深度，即连续采样个数。连续采样个数最大为 ，滤波时间常数计算公式如下：

举例，在96MHz工作频率下，周期Tclk是10.4ns，寄存器配置最大为32767，最长滤波宽度为约 。

通过访问HALL\_INFO.FIL\_DATA[2:0]可以捕捉滤波后的HALL信号；HALL\_INFO.RAW\_DATA[2:0]则是滤波前原始HALL输入信号，详见12.3.3。

# 12.2.4 捕获

捕获模块用于测量两次HALL信号变化之间的时间，其核心为一个24位计数器，在96MHz工作频率下，如果Hall时钟进行8分频，最大可以记录约 的时间宽度，达到10.42ns的时间分辨率。

HALL\_CNT从0开始计数，当发生HALL信号变化时，将此时刻的HALL\_CNT值保存到HALL\_WIDTH寄存器，将此时刻的HALL信号保存到HALL\_INFO.FIL\_DATA，输出HALL信号变化中断，HALL\_CNT重新从0开始计数。

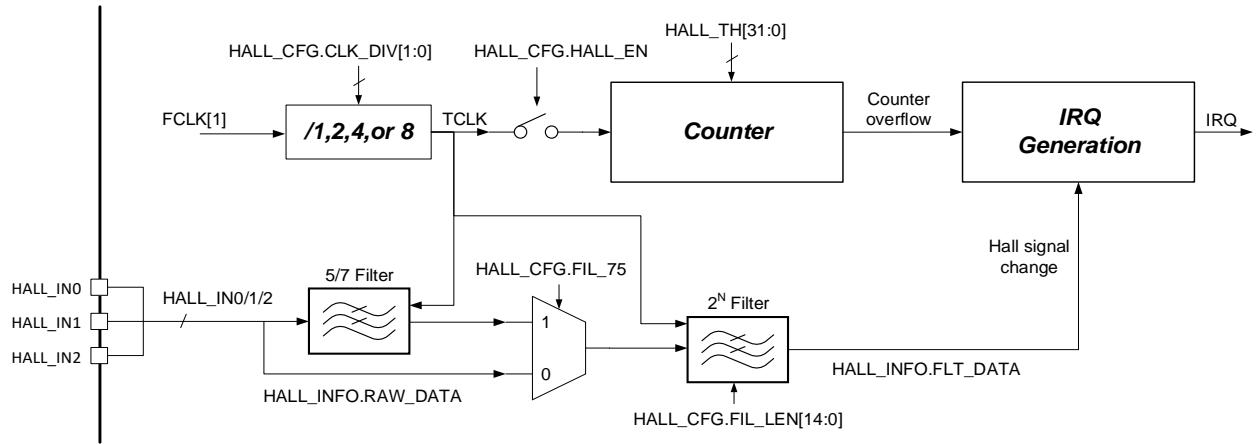
当计数器计数值达到HALL\_TH时，输出HALL计数器溢出中断，计数器重新从0开始计数。

# 12.2.5 中断

捕获、溢出事件触发中断，中断使能控制位位于HALL\_CFG.CHG\_IE和HALL\_CFG.OV\_IE，中断标志位位于HALL\_INFO.CHG\_IF和HALL\_INFO.OV\_IF。终端标志可以通过对HALL\_INFO.CHG\_IF和HALL\_INFO.OV\_IF写1清空。

# 12.2.6 数据流程

12.2.6 数据流程HALL 模块的数据流程如下图所示，FCLK[1]为受 SYS\_CLK\_FEN 门控控制的系统主时钟，通常为 96MHz 的 PLL 时钟。

  
图12-2数据流程框图

# 12.3 寄存器

# 12.3.1 地址分配

HALL模块寄存器的基地址是0x4001\_1000，寄存器列表如下：

表12-1HALL模块寄存器地址分配

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 描述 |
| HALL\_CFG | 0x00 | HALL 模块配置寄存器 |
| HALL\_INFO | 0x04 | HALL 模块信息寄存器 |
| HALL\_WIDTH | 0x08 | HALL 宽度计数值寄存器 |
| HALL\_TH | 0x0C | HALL 模块计数器门限值寄存器 |
| HALL\_CNT | 0x10 | HALL 计数寄存器 |

# 12.3.2 HALL\_CFGHALL模块配置寄存器

地址：0x4001\_1000

复位值：0x0

表12-2HALL模块配置寄存器HALL\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|  | SW\_IE | OV\_IE | CHG\_IE |  |  | HALL\_EN |  |  |  | FL\_75 |  |  | CLK\_DIV |  |  |
| RW | RW | RW | RW | RW | RW |  |
| 0 | 0 | 0 | 0 | 0 | 0 |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |
| --- | --- |
|  | FIL\_LEN\_ |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31] |  | 未使用 |
| [30] | SW\_IE | 软件触发HALL信号变化中断使能,高电平有效。有效此位后,INFO[18]写1,将手动产生HALL信号变化中断。 |
| [29] | OV\_IE | HALL计数器溢出中断使能开关。默认关闭。 1,使能 0,关闭 |
| [28] | CHG\_IE | HALL信号变化中断使能开关。默认关闭。 1,使能 0,关闭 |
| [27:25] |  | 未使用 |
| [24] | HALL\_EN | HALL模块使能开关。默认关闭。 1,使能 0,关闭 |
| [23:21] |  | 未使用 |
| [20] | FIL\_75 | 7/5滤波开关（连续采样7次，5次值一致）。默认关闭。 1,使能 0,关闭 |
| [19:18] |  | 未使用 |
| [17:16] | CLK\_DIV | HALL时钟分频系数 00:不分频 01:2分频 10:4分频 11:8分频 |
| [15] |  | 未使用 |
| [14:0] | FIL\_LEN | 滤波宽度，低于对应脉冲宽度的信号将被硬件自动过滤掉。滤波宽度的计算公式为[14:0]+1。 |

# 12.3.3 HALL\_INFO HALL 模块信息寄存器

地址：0x4001\_1004

复位值：0x0

表12-3HALL模块信息寄存器HALL\_INFO

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|  |  |  |  |  |  |  |  |  |  |  |  | SW\_IF | OV\_IF | CHG\_IF |  |
| WO | RW | RW |  |
| 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:19] |  | 未使用 |
| [18] | SW\_IF | 软件触发 HALL 信号变化中断。写 1 触发，自动清零。当 HALL\_CFG.SW\_IE=1 时，向 SW\_IF 写 1 可以触发 CHG\_IF 置位。需要使 能 HALL\_CFG.CHG\_IE，才能产生 HALL 中断。 |
| [17] | OV\_IF | HALL 计数器溢出事件标志，写 1 清空 |
| [16] | CHG\_IF | HALL 信号变化事件标志，写 1 清空 |
| [15:11] |  | 系统保留，必须写入 0，读出 0 |
| [10:8] | RAW\_DATA | HALL 值，未滤波结果 |
| [7:3] |  | 系统保留，必须写入 0，读出 0 |
| [2:0] | FLT\_DATA | HALL 值，滤波结果 |

# 12.3.4 HALL\_WIDTH HALL 宽度计数值寄存器

地址：0x4001\_1008

复位值：0x0

表12-4HALL宽度计数值寄存器HALL\_WIDTH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 0 | 19 | 18 | 17 | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | CAP\_CNT | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RO | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | CAP\_CNT | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RO | | | | | | | 0 | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:24] |  | 未使用 |
| [23:0] | CAP\_CNT | HALL 宽度计数器值 |

# 12.3.5 HALL\_TH HALL 模块计数器门限值寄存器

地址：0x4001\_100C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|  | | | | | | | | TH | | | | | | | |
| RW | | | | | | | |
| 0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | TH | | | | | | | |
| RW | | | | | | |  |
| 0 | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:24] |  | 未使用 |
| [23:0] | TH | HALL 计数器门限值 |

# 12.3.6 HALL\_CNT HALL 计数寄存器

地址：0x4001\_1010

复位值：0x0

表12-6HALL计数寄存器HALL\_CNT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 0 | 19 | 18 | 17 | 16 |
|  | | | | | | | | CNT | | | | | | | |
| RW | | | | | | | |
| 0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:24] |  | 未使用 |
| [23:0] | CNT | HALL 计数值，写入任意值可清零 |

# 13 MCPWM

# 13.1 概述

MCPWM 模块，是一个精确控制电机驱动波形输出的模块。

包含一个16位递增计数器，用于提供一个基础周期。计数器的时钟分频有1/2/4/8四种选项，产生的的分频时钟频率分别为96MHz、48MHz、24MHz和12MHz。

包含四组PWM生成模块。

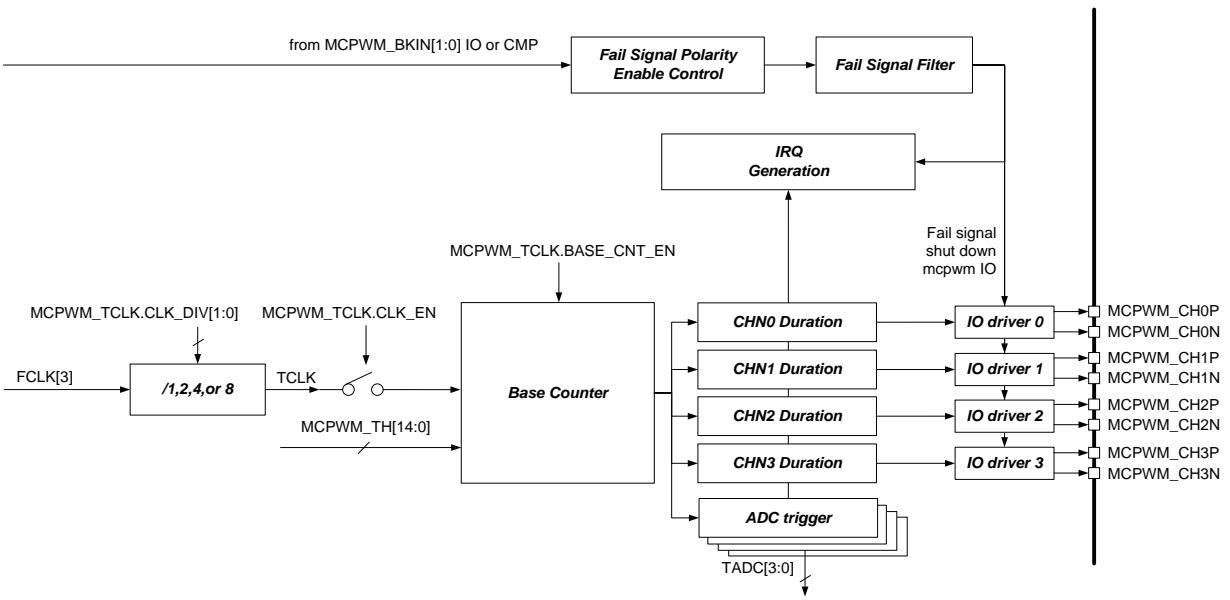
* 可以产生4对（互补信号）不交叠的或8路边沿对齐的（边沿模式）PWM信号；- 支持边沿对齐PWM- 中心对齐PWM- 移相PWM

同时可以产生4路与MCPWM同时基的定时信息，用于触发ADC模块同步采样，进行与MCPWM的联动。

包含一组急停保护模块，用于不依赖CPU软件的处理快速关断MCPWM模块输出。MCPWM模块可输入4路急停信号，其中两路来自芯片IO，两路来自片内比较器的输出。当急停事件发生时（支持有效电平极性选择），把所有MCPWM输出信号复位到规定状态，以避免短路发生。

对急停信号有独立滤波模块。

MCPWM的每个输出IO支持两种控制模式：PWM硬件控制或者软件直接控制（用于EABS软刹车，或BLDC方波换相控制）。

  
图13-1MCPWM模块框图

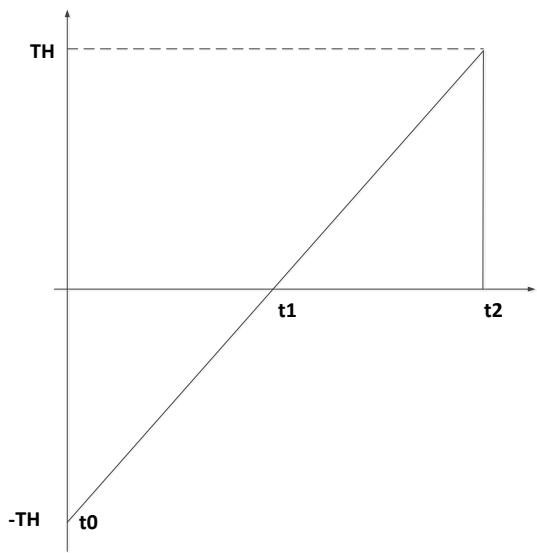
为了保证定时精度，通常采用96MHz的时钟作为MCPWM模块工作频率。

# 13.1.1 BaseCounter模块

该模块主要是由一个递增计数器组成，其计数门限值为MCPWM\_TH，计数器从t0时刻开始从- TH递增计数，在t1时刻过0点，在t2时刻计数到TH完成一次计数循环，回到- TH，重新开始计数。计数周期为 计数时钟周期。

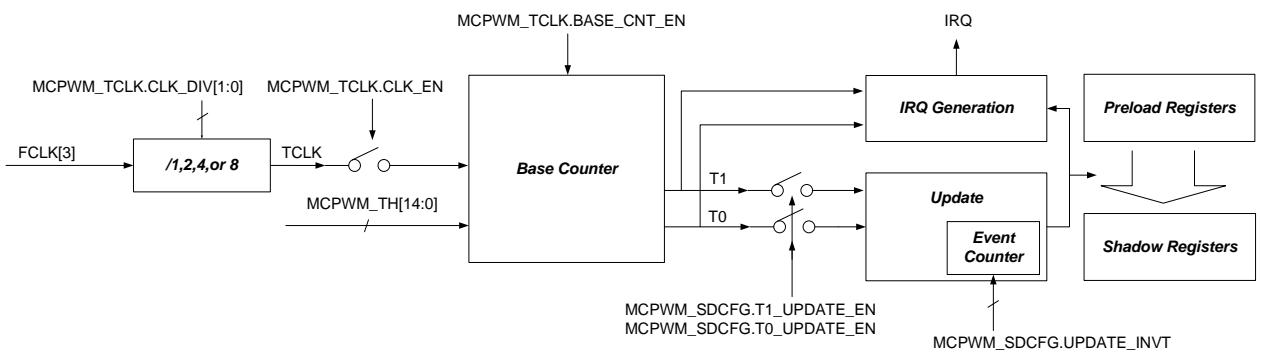
在t0/t1(本次t0即上一次t2)可产生定时事件中断，MCPWM\_IFT0\_IF和MCPWM\_IFT1\_IF将被置位。

可通过寄存器配置MCPWM\_TCLK.BASE\_CNT\_EN控制BaseCounter的启动和停止。

  
图13-2BaseCountert0/t1时序

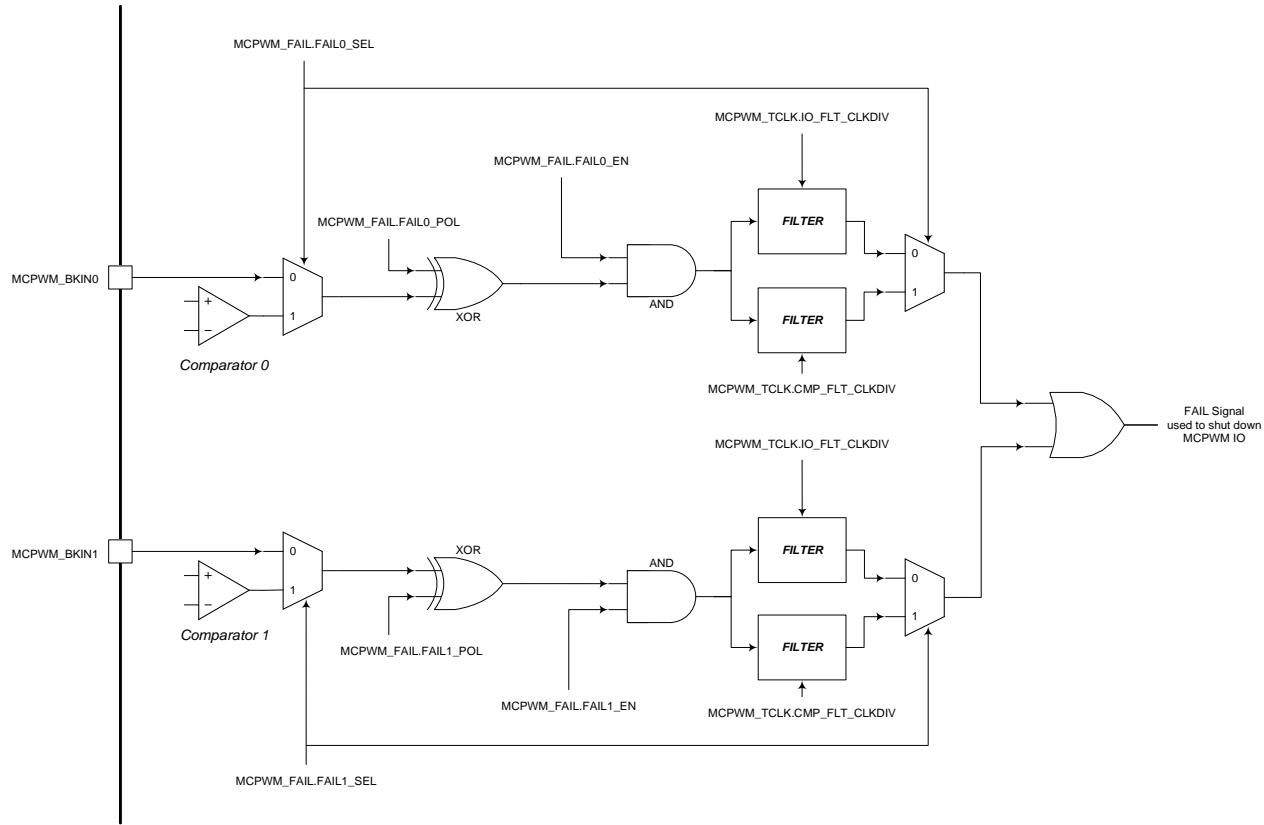
在运行MCPWM模块前，用户一般需将对应的比较门限值MCPWM\_TH00MCWM\_TH31，死区寄存器MCPWM\_DTH00MCWM\_DTH31以及PWM周期MCPWM\_TH配置好。在实际运行过程中，也可动态改变比较门限值和PWM周期寄存器，可通过写MCPWM\_UPDATE寄存器实现手动更新，也可以通过配置MCPWM\_SDCFG.T1\_UPDATE\_EN及MCPWM\_SDCFG.TO\_UPDATE\_EN进行硬件自动更新。硬件更新，仅在t0及t1时刻（可配置t0或t1更新和t0t1时刻都更新）才能产生更新事件，硬件把加载寄存器的值载入到实际运行的寄存器中。而更新事件的发生频率可以配置，即每间隔N个t0及t1时刻才发生更新。无论是否发生更新，t0/t1时刻均可产生相应的中断。若硬件把加载寄存器的值到载入实际运行的寄存器后，产生装载中断。

通过配置MCPWM\_SDCFG寄存器选择更新发生在t0或者t1或者二者皆可，配置更新间隔数，间隔数为1~16。最频繁的更新配置为更新发生在t0和t1，连续发生。最低速的更新配置为更新发生在t1，每16个t1更新一次。

  
图13-3 MCPWM更新机制

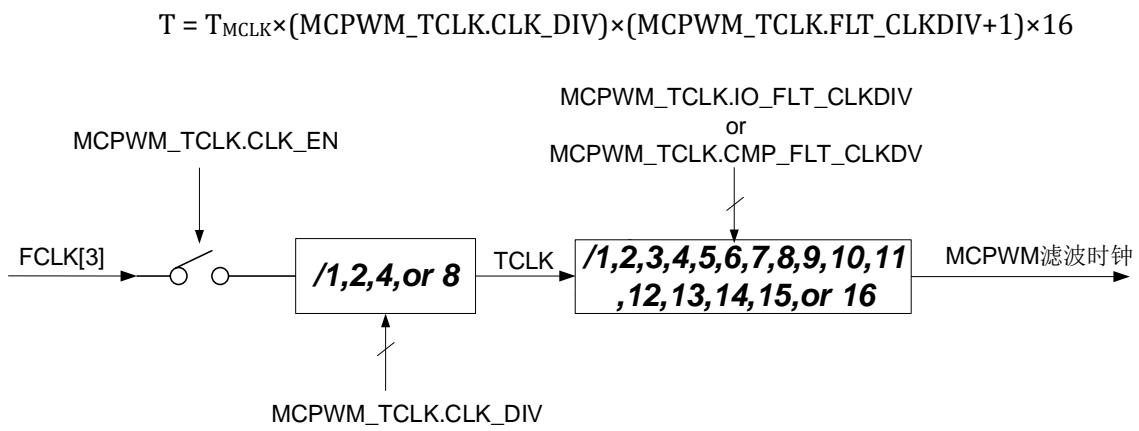
# 13.1.2 Fail信号处理

Fail信号即急停信号，主要用于在出现异常时迅速关断功率管，以免造成不可逆的硬件损毁。该信号处理模块主要是根据实际情况设置急停事件，实现快速关断PWM的输出。有两路fail信号输入MCPWM，即FAIL0和FAIL1，分别可以来自芯片IO MCPWM\_BKIN[1:0]或芯片内部比较器的输出CMP[1:0]。

  
图13-4 MCPWM FAIL逻辑示意图

Filter滤波模块的时钟，来自系统主时钟MCLK的门控时钟FCLK[3]，并经过两级分频，第一级分频由MCPWM\_TCLK.CLK\_DIV控制，进行1/2/4/8倍分频。第二级分频可实现1~16倍的分频，如果Fail信号来自MCPWM\_BKIN[1:0]则使用MCPWM\_TCLK.IO\_FLT\_CLKDIV[3:0]作为第二级的分频系数；如果Fail信号来自芯片内部比较器输出，则使用MCPWM\_TCLK.CMP\_FLT\_CLKDIV[3:0]作为第二级的分频系数，如图13- 5所示。

MCPWM 模块使用分频后的时钟进行 Fail 信号滤波，滤波宽度固定为 16 个周期，即输入信号必须保持至少 16 个时钟（两级分频后的时钟）周期稳定后，硬件才判定其为有效输入信号。滤波时间常数的公式为，其中 为 MCLK/FCLK[3] 的时钟周期，96MHz 对应 10.4ns。MCPWM\_TCLKFLT\_CLKDIV 根据配置情况可能是 MCPWM\_TCLK.IOFLT\_CLKDIV 或 MCPWM\_TCLK.CMPFLT\_CLKDIV。

  
图 13-5 MCPWM Fail 信号滤波时钟生成逻辑

一旦发生 Fail 事件，即

图 13- 4 最终输出 Fail 信号，硬件将 IO 输出强制变为 MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 寄存器所指定的故障缺省值，此时 MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 的值直接输出到 IO 口，不再受到 MCPWM\_FAIL.FAIL.POL 等极性控制的影响。

来自比较器的 FAIL 信号为模拟比较器输出的原始信号，未经过比较器数字接口模块的滤波处理，但是可以被 MCPWM 的通道信号进行开窗控制，开窗控制设置见比较器数字接口模块。FAIL 信号进入 MCPWM 模块后，可以通过设置 MCPWM\_TCLK 进行滤波。

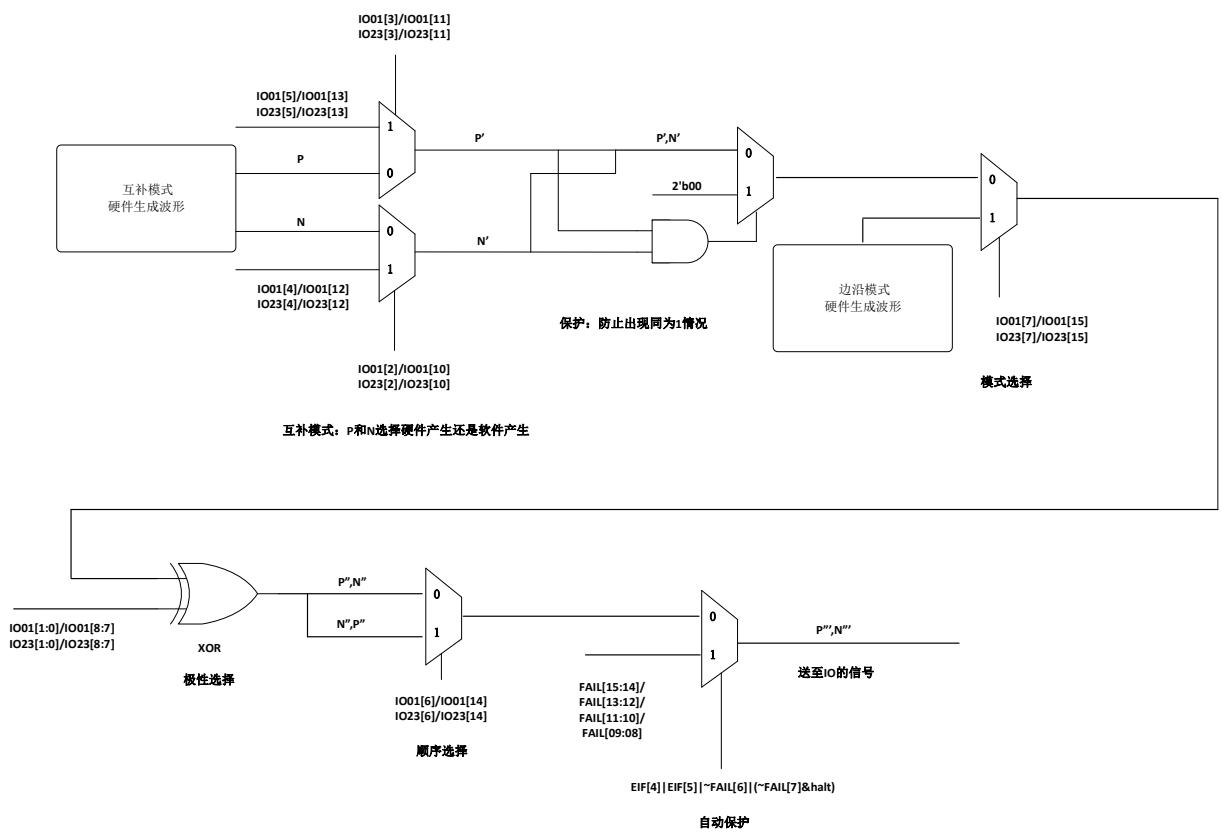
# 13.1.3 MCPWM 特殊输出状态

电机控制中经常会用到全零和全 1 输出状态，以下互补模式设置可以得到期望的输出。

1. 如果 ，芯片处于恒0状态（ 关闭， 开启)，无死区2.如果 ， ，芯片处于恒1状态（ 开启， 关闭)，无死区

# 13.1.4 IO DRIVER 模块

该模块根据实际 MCPWM 的寄存器配置情况，将 IO 设置到相应电平。IO Driver 模块的整体数据流程图如下：

  
图13-6 IO Driver 模块数据流程图

# 13.1.4.1 MCPWM波形输出-中心对齐模式

4个MCPWM IO Driver采用独立的控制门限，独立死区宽度（每一对互补IO的死区需要独立配置，即4个死区配置寄存器)，共享数据更新事件。

采用 和 控制第 个MCPWMIO的启动、关闭动作，n为 0

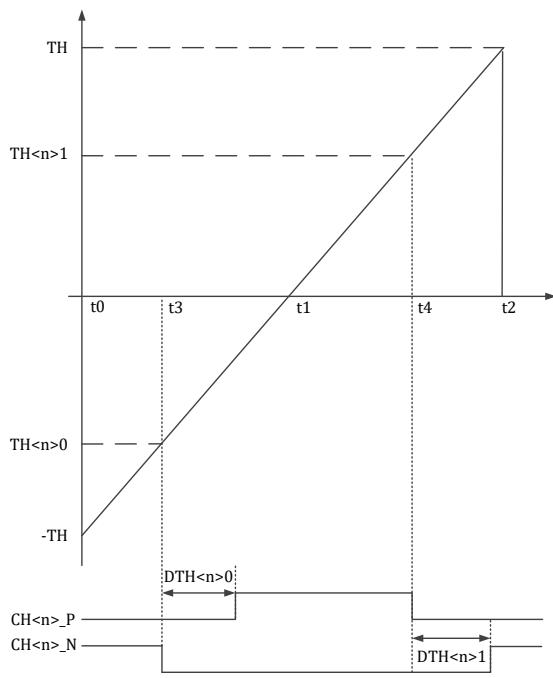
当计数器CNT值向上计数达到 时，在t3时刻关闭 ，经过死区延时DTH0，打开 0

当计数器CNT值向上计数达到 时，在t4时刻关闭 ，经过死区延时DTH1，打开 0

采用独立的启动和关闭时间控制，可以提供相位控制的能力。

死区延时保证 不会同时为高，避免短路发生。

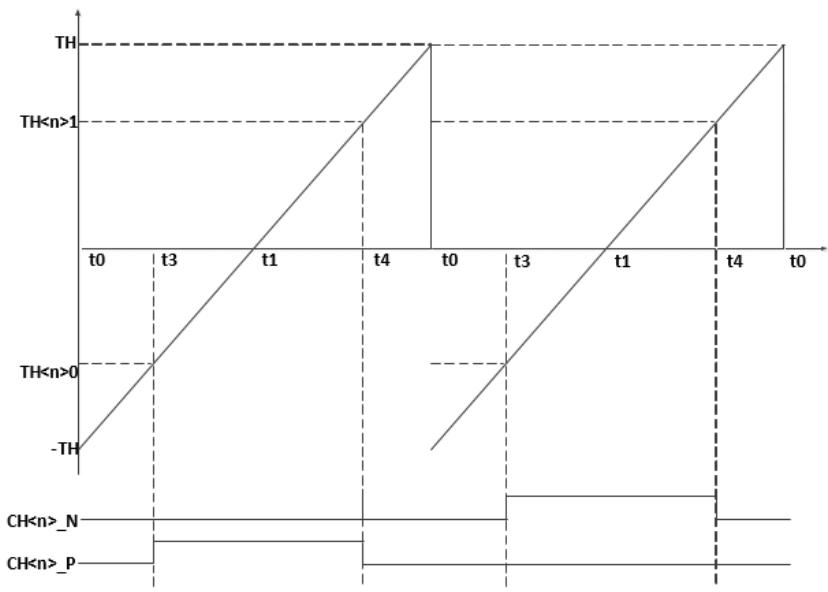
t3/t4时刻均会产生相应中断。

  
图13-7 MCPWM时序 和 -互补模式

13.1.4.2 MCPWM波形控制-中心对齐推挽模式

互补推挽模式。第一个周期内，在t3时刻 置1，在t4时刻， 变低。第二个周期内，在t3时刻 置1，在t4时刻， 变低。

t3/t4均会产生相应中断。

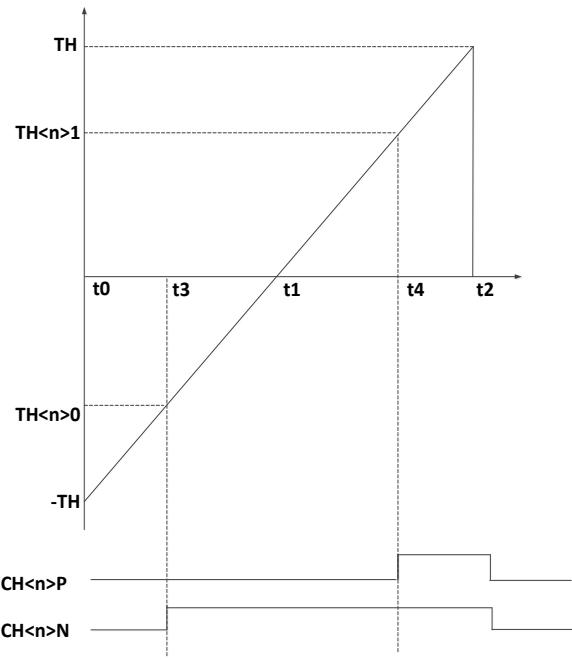
  
图13-8 MCPWM时序 和 -互补模式

13.1.4.3 MCPWM波形输出-边沿对齐模式

边沿对齐模式中，在t0时刻 同时置0，在t3时刻， 变高，在t4时刻， 变高。

t3/t4均会产生相应中断。

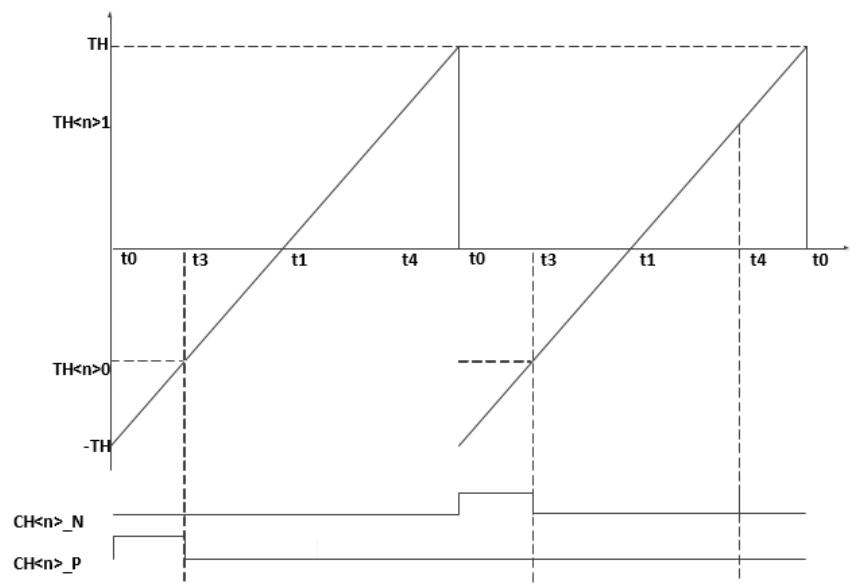
边沿对齐模式下， 无需死区保护。

  
图13-9 MCPWM时序边沿对齐模式

13.1.4.4 MCPWM波形控制- 边沿对齐推挽模式

边沿对齐推挽模式。第一个周期内，在t0时刻 置1，在t3时刻， 变低。第二个周期内，在t0时刻 置1，在t3时刻， 变低。

t0/t3均会产生相应中断。

  
图13-10 MCPWM时序 和 边沿对齐推挽模式

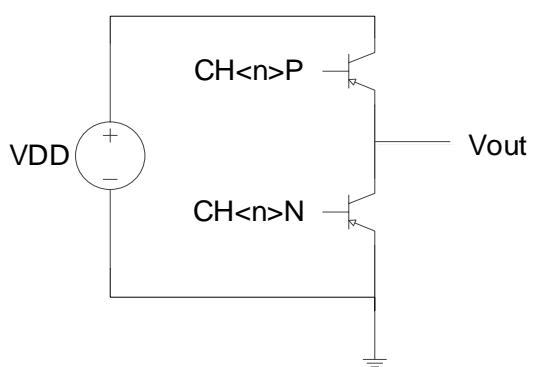
# 13.1.4.5 MCPWMIO死区控制

13.1.4.5 MCPWMIO死区控制MCPWMIO是一对互斥控制信号 ，控制如下图所示的电路，当 为高 为低时，Vout输出高（VDD）；当 为低 为高时，Vout输出低（VSS）；当 为高 为高时，Vout输出不确定，但是会产生VDD到VSS的短路；当 为低 为低时，Vout输出不确定。必须避免 同时为高的情况，死区的引入，可以有效避免VDD到VSS的短路。四组MCPWMIO的死区宽度可独立调整。对于互补模式MCPWMIO自动插入死区。对于边沿对齐模式，MCPWMIO无死区。

在IODriver模块中增加 冲突检测，发生冲突时，自动将IO拉低，同时给出错误中断（中断保持，直到CPU写0)。

MCPWMIO也可通过软件配置的方式输出，此时，死区控制通过软件实现，如果PWM模式为互补，仍然由硬件保证不同时为高或者为低。

，在IO上可以互换。

  
图13-11MCPWMIO控制示意图

# 13.1.4.6MCPWMIO极性设置

13.1.4.6 MCPWMIO极性设置CHP/CHN的有效电平可以配置为高有效/低有效，每个IO的有效电平单独可配。CHP/CHN输出到IO的位置通过软件配置可以互换。

# 13.1.4.7MCPWMIO自动保护

当发生急停事件（Fail事件），应立刻将 自动切换到关闭状态。需要注意关闭电平配置（MCPWM\_FAIL.CHxN\_DEFAULT和MCPWM\_FAIL.CHxP\_DEFAULT控制默认电平)。

芯片正常工作后，IO默认输出的电平是寄存器MCPWM\_FAIL.CHxN\_DEFAULT和MCPWM\_FAIL.CHxP\_DEFAULT指定值，当用户配置完毕，MCPWM正常工作后，配置MCPWM\_FAIL.MCPWM\_OE（即MOE）为1，IO输出电平受到MCPWMIO模块控制。

当发生Fail短路状况时，硬件立即切换到IO默认输出电平。当芯片调试中，CPU Halt时，PWM停止输出，输出FAIL[15:8]的值。

# 13.1.5 ADCTriggerTimer模块

13.1.5 ADC Trigger Timer 模块MCPWM 可以提供 ADC 采样控制。当计数器计数到 MCPWM\_TMR0/ MCPWM\_TMR1/ MCPWM\_TMR2/ MCPWM\_TMR3 时，可产生定时事件，触发 ADC 采样。该触发信号可同时输出到 GPIO，便于调试之用。输出的具体 GPIO，参见对应器件的 datasheet。

# 13.1.6 MCPWM主要事件列表

MCPWM模块，计数器阀值同对应事件如下表所述

表13-1MCPWM计数器阀值与事件对应表

|  |  |
| --- | --- |
| T0 | -MCPWM\_TH |
| T1 | 0 |
| TI00[0] | MCPWM\_TH00 |
| TI00[1] | MCPWM\_TH01 |
| TI01[0] | MCPWM\_TH10 |
| TI01[1] | MCPWM\_TH11 |
| TI02[0] | MCPWM\_TH20 |
| TI02[1] | MCPWM\_TH21 |
| TI03[0] | MCPWM\_TH30 |
| TI03[1] | MCPWM\_TH31 |
| TADC[0] | MCPWM\_TMR0 |
| TADC[1] | MCPWM\_TMR1 |
| TADC[2] | MCPWM\_TMR2 |
| TADC[3] | MCPWM\_TMR3 |

# 13.2 寄存器

# 13.2.1 地址分配

MCPWM模块寄存器的基地址是0x4001\_1C00，寄存器列表如下：

表13-2MCPWM模块寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| MCPWM\_TH00 | 0x00 | MCPWM CH0\_P 比较门限值寄存器 |
| MCPWM\_TH01 | 0x04 | MCPWM CH0\_N 比较门限值寄存器 |
| MCPWM\_TH10 | 0x08 | MCPWM CH1\_P 比较门限值寄存器 |
| MCPWM\_TH11 | 0x0C | MCPWM CH1\_N 比较门限值寄存器 |
| MCPWM\_TH20 | 0x10 | MCPWM CH2\_P 比较门限值寄存器 |

|  |  |  |
| --- | --- | --- |
| MCPWM\_TH21 | 0x14 | MCPWM CH2\_N 比较门限值寄存器 |
| MCPWM\_TH30 | 0x18 | MCPWM CH3\_P 比较门限值寄存器 |
| MCPWM\_TH31 | 0x1C | MCPWM CH3\_N 比较门限值寄存器 |
| MCPWM\_TMR0 | 0x20 | ADC 采样定时器比较门限 0 寄存器 |
| MCPWM\_TMR1 | 0x24 | ADC 采样定时器比较门限 1 寄存器 |
| MCPWM\_TMR2 | 0x28 | ADC 采样定时器比较门限 2 寄存器 |
| MCPWM\_TMR3 | 0x2C | ADC 采样定时器比较门限 3 寄存器 |
| MCPWM\_TH | 0x30 | MCPWM 门限值寄存器 |
| MCPWM\_UPDATE | 0x34 | MCPWM 加载控制寄存器 |
| MCPWM\_IE | 0x38 | MCPWM 中断控制寄存器 |
| MCPWM\_IF | 0x3C | MCPWM 中断标志位寄存器 |
| MCPWM\_EIE | 0x40 | MCPWM 异常中断控制寄存器 |
| MCPWM\_EIF | 0x44 | MCPWM 异常中断标志位寄存器 |
| MCPWM\_RE | 0x48 | MCPWM DMA 请求使能寄存器 |
| MCPWM\_PP | 0x4C | MCPWM 推挽模式使能寄存器 |
| MCPWM\_IO01 | 0x50 | MCPWM IO01 控制寄存器 |
| MCPWM\_IO23 | 0x54 | MCPWM IO23 控制寄存器 |
| MCPWM\_SDCFG | 0x58 | MCPWM 加载配置寄存器 |
| MCPWM\_TCLK | 0x60 | MCPWM 时钟分频控制寄存器 |
| MCPWM\_FAIL | 0x64 | MCPWM 短路控制寄存器 |
| MCPWM\_PRT | 0x74 | MCPWM 保护寄存器 |
| MCPWM\_CNT | 0x78 | MCPWM 计数器寄存器 |
| MCPWM\_DTH00 | 0x80 | MCPWM CH0 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH01 | 0x84 | MCPWM CH0 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH10 | 0x88 | MCPWM CH1 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH11 | 0x8C | MCPWM CH1 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH20 | 0x90 | MCPWM CH2 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH21 | 0x94 | MCPWM CH2 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH30 | 0x98 | MCPWM CH3 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH31 | 0x9C | MCPWM CH3 P 通道死区宽度控制寄存器 |

13-3受MCPWM\_PRT保护的寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| MCPWM\_TH | 0x30 | MCPWM 门限值寄存器 |
| MCPWM\_IE | 0x38 | MCPWM 中断控制寄存器 |
| MCPWM\_EIE | 0x40 | MCPWM 异常中断控制寄存器 |
| MCPWM\_RE | 0x48 | MCPWM DMA 请求使能寄存器 |
| MCPWM\_PP | 0x4C | MCPWM 推挽模式使能寄存器 |
| MCPWM\_IO01 | 0x50 | MCPWM IO01 控制寄存器 |
| MCPwm\_ID023 | 0x54 | MCPWM IO023 控制寄存器 |
| MCPWM\_SDCFG | 0x58 | MCPWM 加载配置寄存器 |
| MCPWM\_TCLK | 0x60 | MCPWM 时钟分频控制寄存器 |

|  |  |  |
| --- | --- | --- |
| MCPWM\_FAIL | 0x64 | MCPWM 短路控制寄存器 |
| MCPWM\_DTH00 | 0x80 | MCPWM CH0 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH01 | 0x84 | MCPWM CH0 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH10 | 0x88 | MCPWM CH1 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH11 | 0x8C | MCPWM CH1 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH20 | 0x90 | MCPWM CH2 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH21 | 0x94 | MCPWM CH2 P 通道死区宽度控制寄存器 |
| MCPWM\_DTH30 | 0x98 | MCPWM CH3 N 通道死区宽度控制寄存器 |
| MCPWM\_DTH31 | 0x9C | MCPWM CH3 P 通道死区宽度控制寄存器 |

表13-4存在影子寄存器的寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| MCPWM\_TH00 | 0x00 | MCPWM CH0\_P 比较门限值寄存器 |
| MCPWM\_TH01 | 0x04 | MCPWM CH0\_N 比较门限值寄存器 |
| MCPWM\_TH10 | 0x08 | MCPWM CH1\_P 比较门限值寄存器 |
| MCPWM\_TH11 | 0x0C | MCPWM CH1\_N 比较门限值寄存器 |
| MCPWM\_TH20 | 0x10 | MCPWM CH2\_P 比较门限值寄存器 |
| MCPWM\_TH21 | 0x14 | MCPWM CH2\_N 比较门限值寄存器 |
| MCPWM\_TH30 | 0x18 | MCPWM CH3\_P 比较门限值寄存器 |
| MCPWM\_TH31 | 0x1C | MCPWM CH3\_N 比较门限值寄存器 |
| MCPWM\_TMR0 | 0x20 | ADC 采样定时器比较门限 0 寄存器 |
| MCPWM\_TMR1 | 0x24 | ADC 采样定时器比较门限 1 寄存器 |
| MCPWM\_TMR2 | 0x28 | ADC 采样定时器比较门限 2 寄存器 |
| MCPWM\_TMR3 | 0x2C | ADC 采样定时器比较门限 3 寄存器 |
| MCPWM\_TH | 0x30 | MCPWM 门限值寄存器 |

# 13.2.2 MCPWM\_TH00

无写保护的寄存器

地址：0x4001\_1C00

复位值：0x0

表13-5MCPWM\_TH00配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH00 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |

|  |  |  |
| --- | --- | --- |
| [15:0] | TH00 | MCPWM CH0\_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.3 MCPWM\_TH01

无写保护的寄存器

地址：0x4001\_1C04

复位值：0x0

表13-6MCPWM\_TH01配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH01 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH01 | MCPWM CH0\_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.4 MCPWM\_TH10

无写保护的寄存器

地址：0x4001\_1C08

复位值：0x0

表13-7MCPWM\_TH10配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH10 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH10 | MCPWM CH1\_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.5 MCPWM\_TH11

无写保护的寄存器

地址：0x4001\_1C0C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH11 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH11 | MCPWM CH1\_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.6 MCPWM\_TH20

无写保护的寄存器

地址：0x4001\_1C10

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH20 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH20 | MCPWM CH2\_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.7 MCPWM\_TH21

无写保护的寄存器

地址：0x4001\_1C14

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表13-10 MCPWM\_TH21配置寄存器 | | | | | | | | | | | | |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH21 | | | | | | | | | | | | |  |  |  |
| RW | | | | | | | | | | | | |  |  |  |
| 0 | | | | | | | | | | | | |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH21 | MCPWM CH2\_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.8 MCPWM\_TH30

无写保护的寄存器

地址：0x4001\_1C18

复位值：0x0

表13-11MCPWM\_TH30配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH30 | | | | | | | | | | | | | |  |  |
| RW | | | | | | | | | | | | | |  |  |
| 0 | | | | | | | | | | | | | |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH30 | MCPWM CH3\_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.9 MCPWM\_TH31

无写保护的寄存器

地址：0x4001\_1C1C

复位值：0x0

表13-12MCPWM\_TH31配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH31 | | | | | | | | | | | | | |  |  |
| RW | | | | | | | | | | | | | |  |  |
| 0 | | | |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TH31 | MCPWM CH3\_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.10 MCPWM\_TMR0

无写保护的寄存器

地址：0x4001\_1C20

复位值：0x0

表13-13MCPWM\_TMR0配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMRO | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TMR0 | ADC 采样定时器比较门限 0 寄存器，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.11 MCPWM\_TMR1

无写保护的寄存器

地址：0x4001\_1C24

复位值：0x0

表13-14MCPWM\_TMR1配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMR1 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TMR1 | ADC 采样定时器比较门限 1 寄存器，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.12 MCPWM\_TMR2

无写保护的寄存器

地址：0x4001\_1C28

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMR2 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TMR2 | ADC 采样定时器比较门限 2 寄存器，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.13 MCPWM\_TMR3

无写保护的寄存器

地址：0x4001\_1C2C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMR3 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | TMR3 | ADC 采样定时器比较门限 3 寄存器，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.14 MCPWM\_TH

写保护的寄存器

地址：0x4001\_1C30

复位值：0x0

表13-17MCPWM\_TH配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:15] |  | 未使用 |
| [14:0] | TH | MCPWM 计数器门限值，15 位无符号数，MCPWM 实际运行系统中的计数器从-TH 计数到 TH；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。 |

# 13.2.15 MCPWM\_UPDATE

无写保护的寄存器

地址：0x4001\_1C34

复位值：0x0

表13-18MCPWM\_UPDATE配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | TH\_UPDATE | TMR3\_UPDATE | TMR2\_UPDATE | TMR1\_UPDATE | TMR0\_UPDATE | TH31\_UPDATE | TH30\_UPDATE | TH21\_UPDATE | TH20\_UPDATE | TH11\_UPDATE | TH10\_UPDATE | TH01\_UPDATE | TH00\_UPDATE |  |  |
| WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:13] |  | 未使用 |
| [12] | TH\_UPDATE | 手动将加载 MCPWM\_TH 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [11] | TMR3\_UPDATE | 手动将加载 MCPWM\_TMR3 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [10] | TMR2\_UPDATE | 手动将加载 MCPWM\_TMR2 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [9] | TMR1\_UPDATE | 手动将加载 MCPWM\_TMR1 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [8] | TMR0\_UPDATE | 手动将加载 MCPWM\_TMR0 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |

|  |  |  |
| --- | --- | --- |
| [7] | TH31\_UPDATE | 手动将加载 MCPWM\_TH31 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [6] | TH30\_UPDATE | 手动将加载 MCPWM\_TH30 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [5] | TH21\_UPDATE | 手动将加载 MCPWM\_TH21 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [4] | TH20\_UPDATE | 手动将加载 MCPWM\_TH20 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [3] | TH11\_UPDATE | 手动将加载 MCPWM\_TH11 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [2] | TH10\_UPDATE | 手动将加载 MCPWM\_TH10 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [1] | TH01\_UPDATE | 手动将加载 MCPWM\_TH01 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |
| [0] | TH00\_UPDATE | 手动将加载 MCPWM\_TH00 寄存器的内容到 MCPWM 运行系统中。 1：加载；0：不加载。 |

向MCPWM\_UPDATE对应位写1可以触发寄存器值从预装载寄存器写入影子寄存器，MCPWM\_UPDATE写入后自动清零。每写1一次，进行一次软件/手动触发。

# 13.2.16 MCPWM\_IE

写保护的寄存器

地址：0x4001\_1C38

复位值：0x0

表13-19MCPWM\_IE配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |
|  | SHADE\_IE | TMR3\_IE |  | TMR2\_IE | TMR1\_IE |  | TMR0\_IE |  | TMR1\_IE |  | TMR0\_IE |  | TMR1\_IE |  | TMR0\_IE |  | TLIE |  | TOIE |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:15] |  | 未使用 |
| [14] | SHADE\_IE | MCPWM\_TH/MCPWM\_TH00MCPWM\_TH31/MCPWM\_TMR0MCPWM\_TMR3 等寄存器更新到 MCPWM 实际运行系统的中断源使能。 1，使能；0，关闭。 |
| [13] | TMR3\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR3 中断源使能。 1，使能；0，关闭。 |
| [12] | TMR2\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR2 中断源使能。 1，使能；0，关闭。 |
| [11] | TMR1\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR1 中断源使能。 |

|  |  |  |
| --- | --- | --- |
|  |  | 1，使能；0，关闭。 |
| [10] | TMR0\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR0 中断源使能。1，使能；0，关闭。 |
| [9] | TH31\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH31 中断源使能。1，使能；0，关闭。 |
| [8] | TH30\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH30 中断源使能。1，使能；0，关闭。 |
| [7] | TH21\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH21 中断源使能。1，使能；0，关闭。 |
| [6] | TH20\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH20 中断源使能。1，使能；0，关闭。 |
| [5] | TH11\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH11 中断源使能。1，使能；0，关闭。 |
| [4] | TH10\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH10 中断源使能。1，使能；0，关闭。 |
| [3] | TH01\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH01 中断源使能。1，使能；0，关闭。 |
| [2] | TH00\_IE | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH00 中断源使能。1，使能；0，关闭。 |
| [1] | T1\_IE | t1 事件，计数器的计数值到达 0 中断源使能。1，使能；0，关闭。 |
| [0] | T0\_IE | t0 事件，计数器的计数值回到 MCPWM\_TH 中断源使能。1，使能；0，关闭。 |

# 13.2.17 MCPWM\_IF

无写保护的寄存器

地址：0x4001\_1C3C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | HADE\_IF | TMR3\_IF | TMR2\_IF | TMR1\_IF | TMRO\_IF | TH31\_IF | TH30\_IF | TH21\_IF | TH20\_IF | TH11\_IF | TH02\_IF | TH01\_IF | TH00\_IF | TL1F | TO1F |
| RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:15] |  | 未使用 |
| [14] | SHADE\_IF | MCPWM\_TH/MCPWM\_TH00MCPWM\_TH31/MCPWM\_TMR0MCPWM\_TMR3 等寄存器更新到 MCPWM 实际运行系统的中断源事件。 |

|  |  |  |
| --- | --- | --- |
|  |  | 1，发生；0，没发生。写1清零。 |
| [13] | TMR3\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR3 中断源事件。1，发生；0，没发生。写1清零。 |
| [12] | TMR2\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR2 中断源事件。1，发生；0，没发生。写1清零。 |
| [11] | TMR1\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR1 中断源事件。1，发生；0，没发生。写1清零。 |
| [10] | TMR0\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TMR0 中断源事件。1，发生；0，没发生。写1清零。 |
| [9] | TH31\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH31 中断源事件。1，发生；0，没发生。写1清零。 |
| [8] | TH30\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH30 中断源事件。1，发生；0，没发生。写1清零。 |
| [7] | TH21\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH21 中断源事件。1，发生；0，没发生。写1清零。 |
| [6] | TH20\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH20 中断源事件。1，发生；0，没发生。写1清零。 |
| [5] | TH11\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH11 中断源事件。1，发生；0，没发生。写1清零。 |
| [4] | TH10\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH10 中断源事件。1，发生；0，没发生。写1清零。 |
| [3] | TH01\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH01 中断源事件。1，发生；0，没发生。写1清零。 |
| [2] | TH00\_IF | MCPWM 实际运行系统中计数器的计数值等于 MCPWM\_TH00 中断源事件。1，发生；0，没发生。写1清零。 |
| [1] | T1\_IF | t1 事件，计数器的计数值到达0中断源事件。1，发生；0，没发生。写1清零。 |
| [0] | T0\_IF | t0 事件，计数器的计数值回到 MCPWM\_TH 中断源事件。1，发生；0，没发生。写1清零。 |

# 13.2.18 MCPWM\_EIE

写保护的寄存器

地址：0x4001\_1C40

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CH3\_SHORT\_IE |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CH2\_SHORT\_IE |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CH1\_SHORT\_IE |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CH0\_SHORT\_IE |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:6] |  | 未使用 |
| [5] | FAIL1\_IE | FAIL1 中断源使能。1，使能；0，关闭。 |
| [4] | FAIL0\_IE | FAIL0 中断源使能。1，使能；0，关闭。 |
| [3] | CH3\_SHORT\_IE | MCPWM CH3\_P 和 CH3\_N 同时有效，中断源使能。1，使能；0，关闭。 |
| [2] | CH2\_SHORT\_IE | MCPWM CH2\_P 和 CH2\_N 同时有效，中断源使能。1，使能；0，关闭。 |
| [1] | CH1\_SHORT\_IE | MCPWM CH1\_P 和 CH1\_N 同时有效，中断源使能。1，使能；0，关闭。 |
| [0] | CH0\_SHORT\_IE | MCPWM CH0\_P 和 CH0\_N 同时有效，中断源使能。1，使能；0，关闭。 |

# 13.2.19 MCPWM\_EIF

无写保护的寄存器

地址：0x4001\_1C44复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  | FAIL1\_IF | FAIL0\_IF | CH3\_SHORT\_IF | CH2\_SHORT\_IF | CH1\_SHORT\_IF | CH0\_SHORT\_IF |  |  |  |  |
| RW1C | RW1C | RW1C | RW1C | RW1C | RW1C |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:6] |  | 未使用 |
| [5] | FAIL1\_IF | FAIL1 中断源事件。1，发生；0，没发生。写 1 清零。 |
| [4] | FAIL0\_IF | FAIL0 中断源事件。1，发生；0，没发生。写 1 清零。 |
| [3] | CH3\_SHORT\_IF | MCPWM CH3\_P 和 CH3\_N 同时有效，中断源事件。 1，发生；0，没发生。写 1 清零。 |
| [2] | CH2\_SHORT\_IF | MCPWM CH2\_P 和 CH2\_N 同时有效，中断源事件。 1，发生；0，没发生。写 1 清零。 |
| [1] | CH1\_SHORT\_IF | MCPWM CH1\_P 和 CH1\_N 同时有效，中断源事件。 1，发生；0，没发生。写 1 清零。 |
| [0] | CH0\_SHORT\_IF | MCPWM CH0\_P 和 CH0\_N 同时有效，中断源事件。 1，发生；0，没发生。写 1 清零。 |

# 13.2.20 MCPWM\_RE

写保护的寄存器

地址：0x4001\_1C48

复位值：0x0

表13-23MCPWM\_RE配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  | ZC\_RE | STOV\_RE | TMR3\_RE | TMR2\_RE | TMR1\_RE | TMR0\_RE |  |
|  |  |  |  |  |  |  |  |  | RW | RW | RW | RW | RW | RW |  |
|  |  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:6] |  | 未使用 |
| [5] | ZC\_RE | DMA 请求使能信号。过零事件请求。写 1，使能；写 0，关闭。 |
| [4] | STOV\_RE | DMA 请求使能信号。起点事件请求。写 1，使能；写 0，关闭。 |
| [3] | TMR3\_RE | DMA 请求使能信号。ADC 通道 3 请求。写 1，使能；写 0，关闭。 |
| [2] | TMR2\_RE | DMA 请求使能信号。ADC 通道 2 请求。写 1，使能；写 0，关闭。 |
| [1] | TMR1\_RE | DMA 请求使能信号。ADC 通道 1 请求。写 1，使能；写 0，关闭。 |
| [0] | TMR0\_RE | DMA 请求使能信号。ADC 通道 0 请求。写 1，使能；写 0，关闭。 |

# 13.2.21 MCPWM\_PP

写保护的寄存器

地址：0x4001\_1C4C

复位值：0x0

表13-24MCPWM\_PP配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | IO3\_PPE | IO2\_PPE | IO1\_PPE | IO0\_PPE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW<fnl> |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 未使用 |
| [3] | IO3\_PPE | IO3 推挽模式使能信号。写 1，使能；写 0，关闭。 |
| [2] | IO2\_PPE | IO2 推挽模式使能信号。写 1，使能；写 0，关闭。 |
| [1] | IO1\_PPE | IO1 推挽模式使能信号。写 1，使能；写 0，关闭。 |
| [0] | IO0\_PPE | IO0 推挽模式使能信号。写 1，使能；写 0，关闭。 |

推挽模式使能信号。根据工作模式不同而不同。边沿模式，开启边沿模式的推挽模式；中心对齐，开启中心对齐的推挽模式。

# 13.2.22 MCPWM\_I001

写保护的寄存器

地址：0x4001\_1C50

复位值：0x0

表13-25MCPWM\_I001配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH1\_WM | CH1\_PN\_SW | CH1\_SCTRLP | CH1\_SCTRLN | CH1\_PS | CH1\_NS | CH1\_PP | CH1\_NP | CH0\_WM | CH0\_PN\_SW | CH0\_SCTRLP | CH0\_PS | CH0\_NS | CH0\_PP | CH0\_NP |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15] | CH1\_WM | CH1 工作模式选择。1：Edge 模式；0：互补模式。 |
| [14] | CH1\_PN\_SW | CH1 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出，N 通道的信号最后从 P 通道输出。1：互换；0：不互换。 |
| [13] | CH1\_SCTRLP | 当 CH1\_PS=1 时，输出到 CH1 P 通道的值。 |
| [12] | CH1\_SCTRLN | 当 CH1\_NS=1 时，输出到 CH1 N 通道的值。 |
| [11] | CH1\_PS | CH1 P 来源。1：来自 CH1\_SCTRLP；0：MCPWM 内部计数器产生。 |
| [10] | CH1\_NS | CH1 N 来源。1：来自 CH1\_SCTRLN；0：MCPWM 内部计数器产生。 |
| [9] | CH1\_PP | CH1 P 极性选择。1：CH1 P 信号取反输出；0：CH1 P 信号正常输出。 |
| [8] | CH1\_NP | CH1 N 极性选择。1：CH1 N 信号取反输出；0：CH1 N 信号正常输出。 |
| [7] | CH0\_WM | CH0 工作模式选择。1：Edge 模式；0：互补模式。 |
| [6] | CH0\_PN\_SW | CH0 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出，N 通道的信号最后从 P 通道输出。1：互换；0：不互换。 |
| [5] | CH0\_SCTRLP | 当 CH0\_PS=1 时，输出到 CH0 P 通道的值。 |
| [4] | CH0\_SCTRLN | 当 CH0\_NS=1 时，输出到 CH0 N 通道的值。 |
| [3] | CH0\_PS | CH0 P 来源。1：来自 CH0\_SCTRLP；0：MCPWM 实际运行系统中计数器产生。 |
| [2] | CH0\_NS | CH0 N 来源。1：来自 CH0\_SCTRLN；0：MCPWM 实际运行系统中计数器产生。 |
| [1] | CH0\_PP | CH0 P 极性选择。1：CH0 P 信号取反输出；0：CH0 P 信号正常输出。 |
| [0] | CH0\_NP | CH0 N 极性选择。1：CH0 N 信号取反输出；0：CH0 N 信号正常输出。极性选择跟随通道交换，例如 CH0 N 选择取反输出，同时选择了通道交换，则交换后的 CH0 N 仍是取反输出。 |

# 13.2.23 MCPWM\_IO23

写保护的寄存器

地址：0x4001\_1C54

复位值：0x0

|  |  |  |
| --- | --- | --- |
|  |  | 换，则交换后的 CH0 N 仍是取反输出。 |

# 13.2.24 MCPWM\_SDCFG

写保护的寄存器

地址：0x4001\_1C58

复位值：0x0

表13-27MCPWM\_SDCFG配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  | AUTO\_ERR\_CLR | T1\_UPDATE\_EN | T0\_UPDATE\_EN | UPDATE\_INTV |  |  |  |  |
| RW | RW | RW | RW | RW |  |  |  |
| 0 | 0 | 0 | 0 | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:7] |  | 未使用 |
| [6] | AUTO\_ERR\_CLR | AUTO\_ERR\_CLR 更新事件是否自动清除 MCPWM\_EIF[5:4]并置位MOE，恢复 MCPWM 信号输出。 1：使能自动故障清除功能；0：关闭自动故障清除功能。 |
| [5] | T1\_UPDATE\_EN | t1（过零）事件更新使能。1：使能；0，关闭。 |
| [4] | T0\_UPDATE\_EN | t0（起点）事件更新使能。1：使能；0，关闭。 |
| [3:0] | UPDATE\_INTV | 更新间隔。t0/t1 发生次数同 UPDATE\_INTV+1 相等时，，MCPWM 系统自动触发 MCPWM\_TH（包括 THxx）和 MCPWM\_TMR 寄存器加载到 MCPWM 运行系统的操作。若 B[5]和 B[4]均关闭，将不会触发此类型加载，只能手动触发加载。 |

# 13.2.25 MCPWM\_TCLK

写保护的寄存器

地址：0x4001\_1C60

复位值：0x0

表13-28MCPWM\_TCLK配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMP\_FLT\_CLKDIV | | | IO\_FLT\_CLKDIV | | |  | | | BASE\_CNT\_EN | | CLK\_EN | | CLK\_DIV | |  |
| RW | | | RW | | | RW | | RW | | RW | |  |
| 0 | | | 0 | | | 0 | | 0 | | 0 | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:16] |  | 未使用 |
| [15:12] | CMP\_FLT\_CLKDIV | 比较器输出的滤波时钟分频寄存器，基于系统时钟分频，影响 MCPWM\_FAIL[1:0]。计算公式如下： 系统时钟/(B[15:12]+1)。分频范围是1-16。 |
| [11:8] | IO\_FLT\_CLKDIV | GPIO输入的滤波时钟分频寄存器，基于系统时钟分频，影响 MCPWM\_FAIL[1:0]。计算公式如下： 系统时钟/(B[11:8]+1)。分频范围是1-16。 |
| [7:4] |  | 未使用 |
| [3] | BASE\_CNT\_EN | MCPWM 实际运行计数器使能开关。1：使能；0：关闭。 |
| [2] | CLK\_EN | MCPWM 工作时钟使能。1：使能；0：关闭。 |
| [1:0] | CLK\_DIV | MCPWM 工作时钟分频寄存器。 0：系统时钟 1：系统时钟/2 2：系统时钟/4 3：系统时钟/8 |

# 13.2.26 MCPWM\_FAIL

写保护的寄存器

地址：0x4001\_1C64

复位值：0x0

表13-29MCPWM\_FAIL配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CH3N\_DEFAULT |  | CH3P\_DEFAULT |  | CH2N\_DEFAULT |  | CH2P\_DEFAULT |  | CH1P\_DEFAULT |  | CHON\_DEFAULT |  | CHOP\_DEFAULT |  | HALT\_PRT |  | MCPWM\_OE |  | FAIL\_EN |  | FAIL\_OEN |  | FAIL\_LPOL |  | FAIL\_OPOL |  | FAIL\_I\_SEL |  | FAIL\_O\_SEL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | RW | RW | RW | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | RW | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | Rw | RW | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | RW | Rw | Rw | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | RW | RW | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | Rw | RW | Rw | Rw | Rw | RW | RW | R |

|  |  |  |
| --- | --- | --- |
|  |  | 出默认电平。默认电平输出不受 MCPWM\_IO01 和 MCPWM\_IO23 的 BIT0、BIT1、BIT8、BIT9、BIT6、BIT14 通道交换和极性控制的影响，直接控制通道输出。 |
| [7] | HALT\_PRT | CPU 进入 HALT 状态，MCPWM 输出值选择。 1：正常输出；0：强制 MCPWM 输出保护值。 |
| [6] | MCPWM\_OE | MOE 控制 MCPWM CHP 和 N 输出值。 1：输出 MCPWM 产生的正常信号 0：输出 CHxN\_DEFAULT 和 CHxP\_DEFAULT 默认值，此默认值不受极性/通道选择等控制。 MCPWM\_EIF\_FAIL1\_IF 和 MCPWM\_EIF\_FAIL0\_IF 任意一位变 1 将触发 MCPWM\_OE 变成 0，输出默认值。 |
| [5] | FAIL1\_EN | FAIL1 输入使能。1：使能；0：关闭。 |
| [4] | FAIL0\_EN | FAIL0 输入使能。1：使能；0：关闭。 |
| [3] | FAIL1\_POL | FAIL1 极性选择。1：信号极性取反输入，信号输入低为有效电平； 0：信号极性正常输入，信号输入高为有效电平。 |
| [2] | FAIL0\_POL | FAIL0 极性选择。1：信号极性取反输入，信号输入低为有效电平； 0：信号极性正常输入，信号输入高为有效电平。 |
| [1] | FAIL1\_SEL | FAIL1 来源选择。1：比较器 1 的结果；0：来自 GPIO 第 1 路。 |
| [0] | FAIL0\_SEL | FAIL0 来源选择。1：比较器 0 的结果；0：来自 GPIO 第 0 路。 |

MCPWM\_FAIL 可以用来设置紧急停车事件，封锁 MCPWM 的信号输出。急停事件主要有两个 FAIL0 和 FAIL1。共有 4 个信号来源，比较器 0 输出和比较器 1 输出以及 MCPWM\_BKIN0 和 MCPWM\_BKIN0。FAIL0 可以来自比较器 0 输出或芯片 IO MCPWM\_BKIN0，FAIL1 可以来自比较器 1 输出或芯片 IO MCPWM\_BKIN1。

FAIL 的输入信号可以使用数字滤波，滤波时钟的第一级分频由 MCPWM\_TCLKCLK\_DIV 寄存器设置。信号来源比较器 0 输出和比较器 1 输出的滤波时钟分频由 MCPWM\_TCLKCMP\_FLT\_CLKDIV 设置；信号来源 MCPWM\_BKIN0 和 MCPWM\_BKIN0 的滤波时钟分频由 MCPWM\_TCLK.IO\_FLT\_CLKDIV 设置。

最后 FAIL0, FAIL1 会对信号进行 16 个滤波时钟的滤波，即只有信号稳定时间超过 16 个滤波周期才能通过滤波器。即滤波宽度 = 滤波时钟周期 \* 16。

更多信息可以参考 13.1.2 Fail 信号处理。

# 13.2.27 MCPWM\_PRT

无写保护的寄存器

地址：0x4001\_1C74

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRT | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |

|  |
| --- |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | PRT | 写入 0xDEAD，解除 MCPWM 寄存器写保护；写入其它值，MCPWM 寄存器进入写保护。 |

# 13.2.28 MCPWM\_CNT

无写保护的寄存器

地址：0x4001\_1C78

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNT | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | CNT | MCPWM 实际运行系统中计数器的值。实际读出的计数范围为 0x8000-TH ~ 0x8000+TH，即读出值与 MCPWM 内部 CNT 计数值存在 0x8000 的偏移 |

# 13.2.29 MCPWM\_DTH00

写保护的寄存器

地址：0x4001\_1C80

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTH00 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:16] |  | 未使用 |
| [15:0] | DTH00 | MCPWM CH0 P 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.30 MCPWM\_DTH01

写保护的寄存器

地址：0x4001\_1C84

复位值：0x0

表13-33MCPWM\_DTH01配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DTH01 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH01 | MCPWM CH0 N 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.31 MCPWM\_DTH10

写保护的寄存器

地址：0x4001\_1C88

复位值：0x0

表13-34MCPWM\_DTH10配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 位置 | 位名称 | 说明 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [31:16] |  | 未使用 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [15:0] | DTH10 | MCPWM CH1 P 通道死区宽度控制寄存器，10bit 无符号数 |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 13.2.32 MCPWM\_DTH11

写保护的寄存器

地址：0x4001\_1C8C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTH11 | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH11 | MCPWM CH1 N 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.33 MCPWM\_DTH20

写保护的寄存器

地址：0x4001\_1C90

复位值：0x0

|  |
| --- |
| DTH20 |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH20 | MCPWM CH2 P 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.34 MCPWM\_DTH21

写保护的寄存器

地址：0x4001\_1C94

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTH21 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH21 | MCPWM CH2 N 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.35 MCPWM\_DTH30

写保护的寄存器

地址：0x4001\_1C98

复位值：0x0

表13-38MCPWM\_DTH30配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTH30 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH30 | MCPWM CH3 P 通道死区宽度控制寄存器，10bit 无符号数 |

# 13.2.36 MCPWM\_DTH31

写保护的寄存器

地址：0x4001\_1C9C

复位值：0x0

表13-39MCPWM\_DTH31配置寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTH31 | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | DTH31 | MCPWM CH3 N 通道死区宽度控制寄存器，10bit 无符号数 |

# 14 UART

# 14.1概述

UART特征如下：

全双工工作支持7/8位数据位支持1/2停止位支持奇/偶/无校验模式带1字节发送缓存带1字节接收缓存支持Multi- dropSlave/Master模式

# 14.2功能说明

# 14.2.1 发送

UART包括一个字节发送缓冲区，当发送缓冲区有数据时，UART将发送缓冲区的数据加载，并通过TX发送出去。

完成加载后，产生发送缓冲区空中断，此时，用户可以往发送缓冲区填入下一个需要发送的字节，这样，发送完成后，UART将加载这个字节进行发送。

完成发送后，会产生发送完成中断。

# 14.2.2 接收

UART包括一个字节的接收缓冲区，当完成一个字节的接收后，会产生接收中断，并将接收到字节存储到接收缓冲区，用户应当在UART接收完成下一个字节前完成此字节的读取，否则缓冲区会被写入新接收的字节。

# 14.2.3 波特率配置

UART输入时钟为系统主时钟，波特率通过两级分频实现。波特率 UART模块时钟/（ UARTx\_DIVH UARTx\_DIVL+1)可以通过SYS\_CLK\_DIV2对UART模块时钟进行分频，UART模块时钟 系统主时钟/(1+SYS\_CLK\_DIV2)

表14-1UART波特率配置示例

|  |  |  |  |
| --- | --- | --- | --- |
| UART 波特率 | SYS\_CLK\_DIV2 | UART\_DIVH | UART\_DIVL |

|  |  |  |  |
| --- | --- | --- | --- |
| 300 | 0x0007 | 0x9C | 0x3F |
| 600 | 0x0003 | 0x9C | 0x3F |
| 1200 | 0x0001 | 0x9C | 0x3F |
| 2400 | 0x0000 | 0x9C | 0x3F |
| 4800 | 0x0000 | 0x4E | 0x1F |
| 9600 | 0x0000 | 0x27 | 0x0F |
| 19200 | 0x0000 | 0x13 | 0x87 |
| 38400 | 0x0000 | 0x09 | 0xC3 |
| 43000 | 0x0000 | 0x08 | 0xB8 |
| 56000 | 0x0000 | 0x06 | 0xB1 |
| 57600 | 0x0000 | 0x06 | 0x82 |
| 115200 | 0x0000 | 0x03 | 0x40 |

注意，波特率配置系数仅为示例，可能不唯一。

# 14.2.4 收发端口互换(TX/RX互换)

UART模块支持Tx与Rx端口互换。通过将Tx对应的GPIO配置为输入使能，Rx对应的GPIO配置为输出使能，即可实现Tx与Rx端口的互换。此时，GPIO第二功能仍选择为UART，UART本身配置无需修改。

此外，如果要使用一个GPIO同时作为Tx和Rx，需要将IO分时复用为输入或输出，对应Rx或Tx，即可实现单口半双工逻辑。

# 14.2.5 DMA配置

UART模块，支持DMA操作。实现DMA搬移数据，能极大减轻CPU的负担。UART采用DMA操作，有以下注意事项。

接收和发送，不能同时使用DMA。只能接收，或者只能发送使用DMA。

发送模式，有两种配置DMA的方案。

方案1：若UARTx\_IE.TX\_BUF\_EMPTY\_RE配置有效。UARTx模块将预取第一个字节，准备发送；一旦数据进入发射队列，UARTx\_IE.UARTx\_BUF即为空，硬件自动会请求DMA搬移下一个字节直至数据搬移完毕。DMA搬移完毕后，将产生DMA完成中断，但是UARTx很可能没有发送完毕最后一个字节，若立即操作UARTx，将可能会产生异常。建议在DMA中断处理程序中，开启UARTx\_IE.TX\_DONE\_IE中断，UARTx将最后一个字节发送完毕，产生发送完成中断，在UARTx中断处理函数里面，再关闭UARTx\_IE.TX\_DONE\_IE。

方案2：若UARTx\_IE.TX\_DONE\_RE配置有效。建议UART初始化阶段不要清除TX\_DONE标志。当前传输的数据长度为Len，DMA配置传输的字节数为Len，开启DMA中断，DMA传输完毕后，UART也发送完毕，软复位UARTx模块并重新初始化UARTx，可开启下一次UARTx的发送。

# 14.3 寄存器

# 14.3.1 地址分配

UART0与UART1实现完全相同。UART0基地址0x4001\_2800。UART1基地址0x4001\_2C00。

表14-2UARTx地址分配列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| UARTx\_CTRL | 0x00 | UART 控制寄存器 |
| UARTx\_DIVH | 0x04 | UART 波特率设置高字节寄存器 |
| UARTx\_DIVL | 0x08 | UART 波特率设置低字节寄存器 |
| UARTx\_BUFF | 0x0C | UART 收发缓冲寄存器 |
| UARTx\_ADR | 0x10 | 485 通信地址匹配寄存器 |
| UARTx\_STT | 0x14 | UART 状态寄存器 |
| UARTx\_IE | 0x18 | UART 中断使能寄存器 |
| UARTx\_IF | 0x1C | UART 中断标志寄存器 |
| UARTx\_INV | 0x20 | UART IO 翻转使能 |

# 14.3.2 UARTx\_CTRLUARTx控制寄存器

UART0\_CTRL地址：0x4001\_2800

UART1\_CTRL地址：0x4001\_2C00

复位值：0x0

表14-3UARTx控制寄存器UARTx\_CTRL

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [6] | MDMASTER\_BIT9 | Multi-drop Master 模式时，第 9 个数据位值 |
| [5] | MD\_EN | 使能 Multi-drop，0:禁用，1:使能 |
| [4] | CK\_EN | 使能校验，0:禁用，1:使能 |

|  |  |  |
| --- | --- | --- |
| [3] | CK\_TYPE | 奇偶校验，0:EVEN 1:ODD |
| [2] | BIT\_ORDER | 先发送的比特，0:LSB,1:MSB |
| [1] | STOP\_LEN | 停止位长度，0:1bit,1:2bit |
| [0] | DAT\_LEN | 数据长度，0:8bit,1:7bit |

# 14.3.3 UARTx\_DIVH UARTx 波特率设置高字节寄存器

UART0\_DIVH 地址：0x4001\_2804 UART1\_DIVH 地址：0x4001\_2C04

复位值：0x0

表14-4UARTx波特率设置高字节寄存器UARTx\_DIVH

|  |  |  |
| --- | --- | --- |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | |
|  | | DIVH |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | DIVH | 波特率设置高字节 BAUDRATE =主时钟/(1+DIVL+256\*DIVH) |

# 14.3.4 UARTx\_DIVL UARTx 波特率设置低字节寄存器

UART0\_DIVL 地址：0x4001\_2808

UART1\_DIVL 地址：0x4001\_2C08

复位值：0x0

表14-5UARTx波特率设置低字节寄存器UARTx\_DIVL

|  |  |  |
| --- | --- | --- |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | |
|  | | DIVL |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | DIVL | 波特率设置低字节 BAUDRATE =主时钟/(1+256\* UARTx\_DIVH+UARTx\_DIVL) |

# 14.3.5 UARTxBUFFUARTx收发缓冲寄存器

UARTOBUFF地址：0x4001\_280C UART1BUFF地址：0x4001\_2C0C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | BUFF | | | | | |  |
| RW | | | | | |  |
| 0 | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | BUFF | 写:发送数据缓存读:接收数据寄存器 |

UART的Tx\_buffer和Rx\_buffer共享地址 地址。其中，Tx\_buffer是只写的，Rx\_buffer是只读的。因此读访问UARTxBUFF是访问UARTx\_RXBUFF，写访问UARTxBUFF是访问UARTx\_TX\_BUFF

# 14.3.6 UARTxADRUARTx地址匹配寄存器

UARTOADER地址：0x4001\_2810 UART1ADER地址：0x4001\_2C10

复位值：0x0

表14-7UARTx地址匹配寄存器UARTxADR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  | | | | | | |  | | | | | | | ADR | | | | | | |
| RW | | | | | | |  |  |  |  |  |  |  |
|  | | | | | | |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | ADR | 用作 485 通信时的匹配地址 |

# 14.3.7 UARTxSTTUARTx状态寄存器

UARTO\_STT地址：0x4001\_2814 UART1\_STT地址：0x4001\_2C14

复位值：0x0

表14-8UARTx状态寄存器UARTx\_STT

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 1 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | TX\_BUF\_EMPTY |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | TX\_BUF\_EMPTY |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:3] |  | 未使用 |
| [2] | ADR\_MATCH | Multi-drop 模式下，地址匹配标志位。 1：匹配；0：未匹配。 |
| [1] | TX\_DONE | 发送完成（此时发送缓存如不为空，则可以继续发送缓存中的数据） 1：完成；0：未完成。 |
| [0] | TX\_BUF\_EMPTY | 发送缓存状态位。 1：空；0：非空。 |

# 14-3.8 UARTx\_IE UARTx 中断使能寄存器

UART0\_IE地址：0x4001\_2818

UART1\_IE地址：0x4001\_2C18

复位值：0x0

表14-9UARTx中断使能寄存器UARTx\_IE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | TX\_BUF\_EMPTY\_RE | 发送缓冲区空 DMA 请求使能， 默认值为 0。 0：关闭；1：开启。 |
| [6] | RX\_DONE\_RE | 接收完成 DMA 请求使能， 默认值为 0。 |

|  |  |  |
| --- | --- | --- |
|  |  | 0：关闭；1：开启。 |
| [5] | TX\_DONE\_IE | 发送完成 DMA 请求使能，默认值为 0。 0：关闭；1：开启。 |
| [4] | CK\_ERR\_IE | 校验错误中断开关，默认值为 0。 0：关闭；1：开启。 |
| [3] | STOP\_ERR\_IE | 停止位错误中断开关，默认值为 0。 0：关闭；1：开启。 |
| [2] | TX\_BUF\_EMPTY\_IE | 发送缓冲区空中断开关，默认值为 0。 0：关闭；1：开启。 |
| [1] | RX\_DONE\_IE | 接收完成中断开关，默认值为 0。 0：关闭；1：开启。 |
| [0] | TX\_DONE\_IE | 发送完成中断开关，默认值为 0。 0：关闭；1：开启。 |

# 14.3.9 UARTx\_IF UARTx 中断标志寄存器

UART0\_IF 地址：0x4001\_281C

UART1\_IF 地址：0x4001\_2C1C

复位值：0x0

表14-10UARTx中断使能寄存器UARTx\_IF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 1 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | TXDONEIF | STOP\_ERRIF | TXBUF\_EMPTYIF | RXDONEIF | TXDONEIF |
|  |
|  |
|  |
|  | | | | | | | | | | |  |
|  |
|  |  |
|  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4] | CK\_ERR\_IF | 校验错误中断标志，高有效，写1清零。 |
| [3] | STOP\_ERR\_IF | 停止位错误中断标志，高有效，写1清零。 |
| [2] | TX\_BUF\_EMPTY\_IF | 发送缓冲区空中断标志，高有效，写1清零。 |
| [1] | RX\_DONE\_IF | 接收完成中断标志，高有效，写1清零。 |
| [0] | TX\_DONE\_IF | 发送完成中断标志，高有效，写1清零。 |

上电后发送缓冲区空标志TX\_BUF\_EMPTY\_IF和发送完成中断标志TX\_DONE\_IF默认为1。

中断标志写1清零，一般不建议用如下 方式清零，因为 是先读取中断标志，将对应位改为1再写入清零，如果同时有其他中断标志置位，会被一起清零，而这通常不是软件所期望的。例如，如下写法本意是清零TX\_DONE\_IF，但如果同时RX\_DONE\_IF在写入执行前置1了，则软件先读取

回 UARTx\_IF 值为 ，然后执行或操作 ，然后写入，同时对 RX\_DONE\_IF 和 TX\_DONE\_IF 进行了清零，可能导致 UART 少进入一次因接收数据产生的中断，从而少接收到一字节数据。

如果希望清零 TX\_DONE\_IF 标志位，应以如下方式，直接对 BIT0 写 1.

# 14.3.10 UARTx\_INV UARTx IO 翻转寄存器

UART0\_INV 地址：0x4001\_2820

UART1\_INV 地址：0x4001\_2C20

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RXD\_INV |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RXD\_INV |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW | RW |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:2] |  | 未使用 |
| [1] | TXD\_INV | TXD 输出极性使能开关， 默认值为 0。 0：正常输出；1：取反输出。 正常输出极性，是指软件发送 1，硬件发送的即是 1；取反输出极性，是指软件发送 1，硬件发送的是 0。 |
| [0] | RXD\_INV | RXD 输入极性使能开关， 默认值为 0。 0：正常输入；1：取反输入。 正常输入极性，是指硬件接收 1，软件接收的即是 1；取反输入极性，是指硬件接收 1，软件接收到的是 0。 |

# 15 DSP

# 15.1概述

DSP模块使用自主设计的DSP指令集，可以进行加法、乘累加、移位、饱和等单周期算术指令，以及除法、开方、三角函数等多周期算术运算指令；具备load/store等访存指令，无条件跳转以及条件跳转等分支指令，还有中断提起等杂项指令。有断点指令和寄存器赋值等伪指令可以在模拟器上用于调试。DSP有两种运行模式，自主运行、被动调用。

DSP有两种运行模式，自主运行、被动调用。

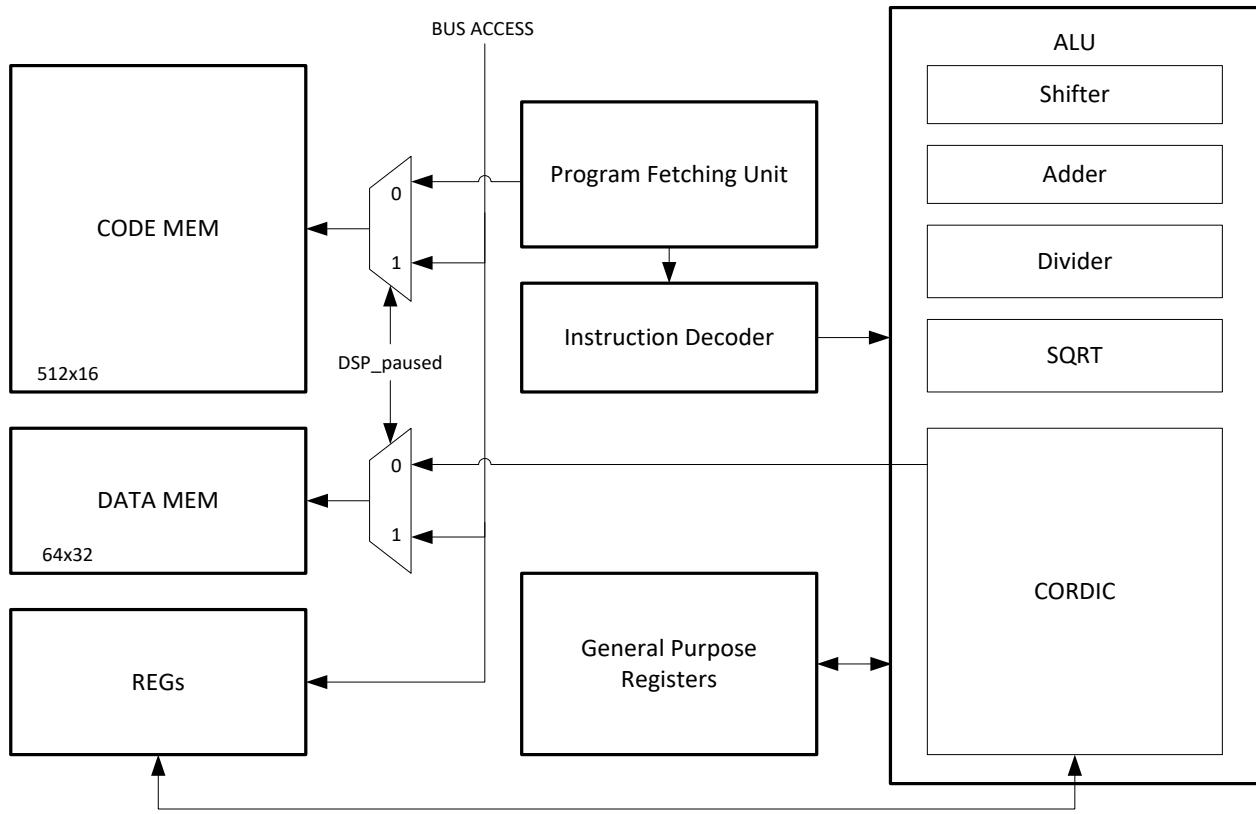
所谓自主运行即DSP读取CODEMEM中的指令和DATAMEM中的数据进行DSP程序执行，独立于ARM CortexM0，此时DSP\_SC.PAUSED ，即DSP处于运行状态；CODEMEM和DATAMEM允许DSP访问但不允许CPU访问改写。

被动调用是指DSP作为一个外设模块被ARM CortexM0所调用，CPU直接访问DSP内部的算术运算资源如除法、开方、三角函数等。此时DSP\_SC.PAUSED ，即DSP不运行DSP程序，处于暂停状态，CODEMEM和DATAMEM允许CPU进行访问改写，对于不进行DSP程序开发的用户，推荐使用此种模式，通过CPU运行的软件直接调用DSP的算术单元。

DSP配备独立的程序存储器(CODEMEM)和数据存储器(DATAMEM)。在DSP暂停即DSP\_SC.PAUSED 时可以通过CPU访问这两个独立的存储区域，在DSP初始化的过程中需要由CPU将DSP运行的程序和初始数据分别写入DSP的CODEMEM和DATAMEM。DSP具备提起中断的指令，中断置位后，DSP同时进入暂停状态，此时允许CPU通过总线接口访问DATAMEM与DSP进行数据交互，包括读取DSP运算结果，以及写入DSP后续运行所需的数据等。

此外，为充分灵活利用DSP，在DSP暂停时允许CPU通过DSP寄存器接口直接访问DSP除法器、开方器、三角函数等运算模块，即允许CPU将DSP当做简单的运算协处理模块使用。

# 15.1.1 功能框图

  
图15-1DSP模块功能框图

# 15.1.2 DSP核寄存器

表15-1DSP核心寄存器

|  |  |  |
| --- | --- | --- |
| Register | Bit Width | Usage |
| R0 | 32 | Always read as 0 |
| R1 | 32 |  |
| R2 | 32 |  |
| R3 | 32 |  |
| R4 | 32 |  |
| R5 | 32 |  |
| R6 | 32 | ARCTAN module destination |
| R7 | 32 | MAC result / DIV dividend |
| PC | 16 | Program Counter |

其中R0寄存器为常0寄存器，不能写入数据，读回恒为0。R0寄存器可作为特殊操作数构造一些伪指令。如

ADDRORORO相当于NOP

MACR1R2RO相当于MULR1R2，即乘累加中累加的数为0，变为乘法操作

R6寄存器在ARCTAN指令中固定用于存放向量模值

R7 寄存器在 MAC 指令中固定用于存放计算结果，在 DIV 指令中固定用于存放被除数操作数。以上约定是由于受定长指令编码限制，在 4 操作数指令中需要固定一个操作数使用约定寄存器。

# 15.1.3 位宽

除法的被除数和商位宽均为 32 位有符号数，除数和余数为 16 位有符号数。被开方数为 32 位无符号数，平方根为 16 位无符号数。乘累加的两个乘数均为 16 位有符号数，加数和结果为 32 位有符号数。三角函数 CORDIC 模块位宽为 16 位，Q15 定点数格式。

注意：在使用 DSP 算术运算资源时，无论是 CPU 调用还是通过 DSP 算术指令调用，请务必保证操作数不要超过表示范围，否则会出现计算异常。例如乘法的两个操作数为 16 位有符号数，除法的除数为 16 位有符号数。16 位有符号数的表示范围为 - 32768~32767，如果输入的 32 位数据超过这一范围，比如以 50000 或 - 40000 作为乘法的操作数，则会因源操作数被截取了低 16 位而出现计算错误。

除法（DIV 指令）的操作数，被除数范围是 ，除数的范围是 ；被除数不支持赋值为 ，除数不支持赋值为 。

减法（SUB 指令）的操作数，减数范围为 ，减数不支持赋值为 。

此外，由于 CORDIC 运算基于多次旋转逐次逼近，计算结果存在不超过 的计算误差。

# 15.1.4 指令周期

除法指令需要 10 个总线周期（96MHz）完成。

开方指令需要 8 个总线周期（96MHz）完成。

三角函数指令需要 8 个总线周期（96MHz）完成。

其余指令均为单周期指令

# 15.1.5 地址空间

表15-2DSP地址空间

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 空间大小 | 空间区域 | 实际存储体大小 |
| code\_mem | 2kB | 0x4001\_4000～0x4001\_47FF | 512 x 16bit |
| data\_mem | 2kB | 0x4001\_4800～0x4001\_4FFF | 64 x 32bit |
| reg | 2kB | 0x4001\_5000～0x4001\_57FF |  |
| Reserved | 2kB | 0x4001\_5800～0x4001\_5FFF |  |

DSP 的地址空间分为 4 段，分别为程序段、数据段、寄存器段、保留段，各段空间占用 2kB 的地址空间，但实际的 CODE MEM 和 DATA MEM 存储空间不足 2kB。程序段（CODE MEM）用于存放 DSP 运行所需程序代码，为单口 SRAM，单周期完成读写操作；数据段（DATA MEM）用于存放 DSP 运行所需数据，为单口 SRAM，单周期完成读写操作；寄存器段为允许 CPU 通过总线访问的 DSP 寄存器；保留段暂未使用。

CODEMEM位宽为16，但仍按word寻址，亦即地址每次增4。

DSP地址空间需要在适当的时机才能访问。其中DSP的状态控制寄存器可以在任何时间访问；DSP的CODEMEM、DATAMEM、以及寄存器段中CODIC三角函数模块、DIV除法器、SQRT开方器只有在DSP暂停即DSP\_SC.PAUSED 时才能进行寄存器访问。因为在DSP运行期间，所有的运算单元可能在被DSP使用，CPU同时通过寄存器接口进行访问会造成访问冲突。因此在DSP运行期间，即DSP\_SC.PAUSED 时，禁止通过寄存器接口访问DSP的运算单元。

DSP遇到中断指令会提起中断等待CPU处理，同时DSP进入暂停状态，DSP\_SC.PAUSED会置位为1。此外软件也可以在任意时刻通过写DSP\_SC.PAUSED 来使DSP进入暂停状态。这一机制主要是为了防止DSP写入的程序中没有IRQ指令，导致DSP在不掉电的情况下会永久运行。

# 15.2 寄存器

# 15.2.1 地址分配

DSP模块在芯片中的基地址是0x4001\_4000。DSP寄存器在芯片中的基地址是0x4001\_5000。

表15-3DSP寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| DSP\_SC | 0x00 | DSP 状态控制寄存器 |
| DSP\_THETA | 0x04 | DSP sin/cos 输入角度寄存器 |
| DSP\_X | 0x08 | DSP arctan/module 计算输入坐标 X 寄存器 |
| DSP\_Y | 0x0C | DSP arctan/module 计算输入坐标 Y 寄存器 |
| DSP\_SIN | 0x10 | DSP sin/cos 计算结果 sin 寄存器 |
| DSP\_COS | 0x14 | DSP sin/cos 计算结果 cos 寄存器 |
| DSP\_MOD | 0x18 | DSP arctan 计算结果 sqrt(X²+Y²) 寄存器 |
| DSP\_ARCTAN | 0x1C | DSP arctan 计算结果 arctan(Y/X) 角度寄存器 |
| DSP\_DID | 0x20 | DSP 除法操作被除数 |
| DSP\_DIS | 0x24 | DSP 除法操作除数 |
| DSP\_QUO | 0x28 | DSP 除法操作商 |
| DSP\_REM | 0x2C | DSP 除法操作余数 |
| DSP\_RAD | 0x30 | DSP 开方操作被开方数 |
| DSP\_SQRT | 0x34 | DSP 开方操作平方根 |

# 15.2.2 DSP状态控制寄存器

# 15.2.2.1 DSP\_SC

地址：0x4001\_5000

复位值：0x2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表15-4DSP状态控制寄存器DSP\_SC | | | | | | | | | | |  |  |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  | RESET\_PC | CORDIC\_MODE | PAUSED | IRQ |  |
| W0 | RW | RW | RWIC |  |
| 0 | 0 | 1 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 保留 |
| [3] | RESET\_PC | 当DSP暂停时，写1重置DSP PC到0地址 |
| [2] | CORDIC\_MODE | CORDIC mode, 0: arctan, 1: sin/cos |
| [1] | PAUSED | 指示DSP是否处于暂停状态，当DSP执行到IRQ指令时此bit置1，软件写可以将此bit置1。软件将此bit清零可以启动DSP运行 0：DSP正读取CODE MEM和DATA MEM自主运行DSP程序 1：DSP暂停取指令，允许ARM软件通过寄存器访问DSP内部算术单元，向寄存器写入操作数触发运算，读取寄存器取得运算结果 对于不编写DSP CODE MEM程序的用户，PAUSED需保持为1 |
| [0] | IRQ | DSP中断标志，写1清零 |

注意，复位后DSP处于暂停状态，即DSP\_SC.PAUSED 时；DSP\_SC.CORDIC\_MODE用于CPU通过寄存器接口访问CORDIC模块时，sin/cos mode和arctan mode的选择，CORDIC模块计算sin/cos或arctan使用的是相同的硬件电路。因此在进行某一种计算前，应通过配置DSP\_SC计算器进行适当模式选择。

只有当DSP处于暂停状态时，CPU才可以通过寄存器接口调用DSP。当DSP自主运行时，CPU无法通过寄存器接口访问DSP内部资源。

DSP\_SC.CORDIC\_MODE位仅在CPU通过寄存器接口调用DSPCORDIC单元时需要设置，DSP程序可以根据SIN\_COS或ARCTAN指令直接进行模式切换，DSP\_SC.CORDIC\_MODE不再起作用。

软件调用CORDIC模块计算sin/cos时以角度DSP\_THETA为输入，计算并输出sin/cos结果到DSP\_SIN/DSP\_COS寄存器；计算arctan时以坐标DSP\_X/DSP\_Y为输入，计算并输出角度theta=arctan 和module sqrt 到DSP\_ARCTAN和DSP\_MOD寄存器。

# 15.2.3DSPsin/cos相关寄存器

CORDIC模块计算sin/cos和arctan使用的是相同的数据通路，因此通过CPU使用CORDIC模块进行sin/cos计算，需要先将DSP\_SC.CORDIC\_MODE写为1，使CORDIC进入sin/cos模式。

# 15.2.3.1DSP\_THETA

地址：0x4001\_5004

复位值：0x0

表15-5DSPsin/cos角度输入寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| THETA | | | | | | | | | | | | | | |  |
| RW | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_THETA[15]}} |
| [15:0] | THETA | DSP sin/cos 输入角度寄存器 |

DSP\_THETA为16位有符号定点数，表示范围（ ）对应（ ）。

15.2.3.2DSP\_SIN

地址：0x4001\_5010

复位值：0x0

表15-6DSPsin/cos正弦结果寄存器

|  |
| --- |
| SIN |
| RO |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_SIN[15]}} |
| [15:0] | SIN | DSP sin/cos 计算结果 sin 寄存器 |

DSP\_SIN为16位有符号定点数，其中1bit符号位，15bit小数位；表示范围（ ）。

15.2.3.3DSP\_COS

地址：0x4001\_5014

复位值：0x0

表15-7DSPsin/cos余弦结果寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COS | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_COS[15]}} |
| [15:0] | COS | DSP sin/cos 计算结果 cos 寄存器 |

DSP\_COS为16位有符号定点数，其中1bit符号位，15bit小数位；表示范围（- 1～1）。

# 15.2.4 DSP arctan 相关寄存器

15.2.4.1 DSP\_X

地址：0x4001\_5008

复位值：0x0

表15-8DSParctan/module坐标X输入寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_X[15]}} |
| [15:0] | X | DSP arctan/module 计算输入坐标 X 寄存器 |

DSP\_X为16位有符号定点数，Q15格式，其中1bit符号位，15bit整数位，表示范围（- 32768～32767）。

15.2.4.2 DSP\_Y

地址：0x4001\_500C

复位值：0x0

表15-9DSParctan/module计算坐标Y输入寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_Y[15]}} |

|  |  |  |
| --- | --- | --- |
| [15:0] | Y | DSP arctan/module 计算输入坐标 Y 寄存器 |

DSP\_Y为16位有符号定点数，其中1bit符号位，15bit整数位，表示范围（- 32768～32767）。

15.2.4.3 DSP\_MOD

地址：0x4001\_5018

复位值：0x0

表15-10DSParctan向量模结果sqrt 寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MOD | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出为 0 |
| [15:0] | MOD | DSP arctan 计算结果 sqrt(X²+Y²) 寄存器 |

DSP\_MOD为16位有符号定点数，其中1bit符号位，15bit整数位，表示范围（- 32768～32767）。

需要注意的是，当计算arctan(Y/X)时，需要保证sqrt 也不超过32767，因此一般建议abs(DSP\_X)和abs(DSP\_Y)不超过 为宜。

15.2.4.4 DSP\_ARCTAN

地址：0x4001\_501C

复位值：0x0

表15-11DSParctan角度结果arctan(Y/X）角度寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ARCTAN | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_ARCTAN[15]}} |
| [15:0] | ARCTAN | DSP arctan 计算结果 arctan(Y/X) 角度寄存器 |

DSP\_ARCTAN为16位有符号定点数，表示范围（- 32768～32767）对应（- π～π）。

需要注意的是，当计算arctan(Y/X)时，需要保证sqrt 也不超过32767，因此一般建议abs(DSP\_X)和abs(DSP\_Y)不超过 为宜。

# 15.2.5 DSP除法相关寄存器

# 15.2.5.1 DSP\_DID

地址：0x4001\_5020

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DID | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DID | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | |  |  |
| 0 | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | DID | DSP 除法被除数寄存器 |

# 15.2.5.2 DSP\_DIS

地址：0x4001\_5024

复位值：0x0

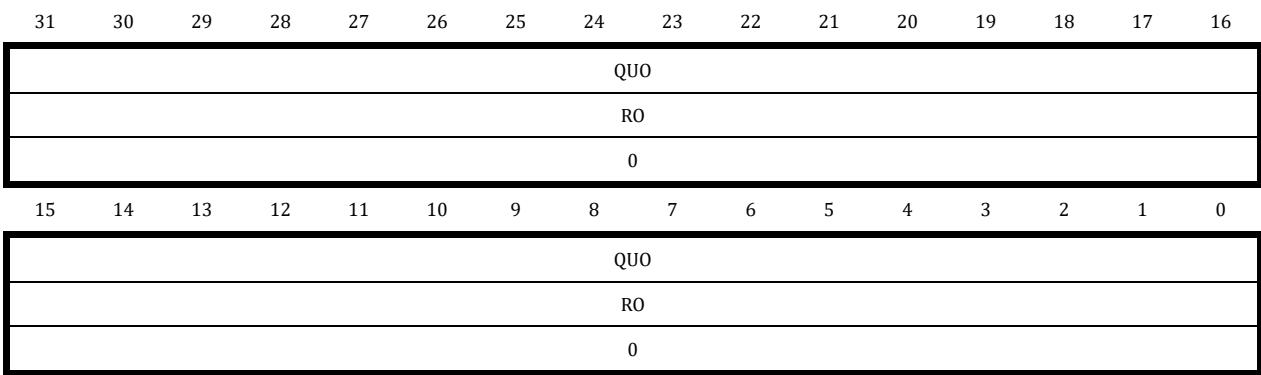
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIS | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <lcel> | | | | | | | | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_DIS[15]}} |
| [15:0] | DIS | DSP 除法除数寄存器 |

# 15.2.5.3 DSP\_QUO

地址：0x4001\_5028

复位值：0x0



|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | QUO | DSP 商寄存器 |

# 15.2.5.4 DSP\_REM

地址：0x4001\_502C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REM | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 权限 | 说明 |
| [31:16] |  | 保留，读出时为符号扩展，即{16{DSP\_REM[15]}} |
| [15:0] | REM | DSP 除法余数寄存器 |

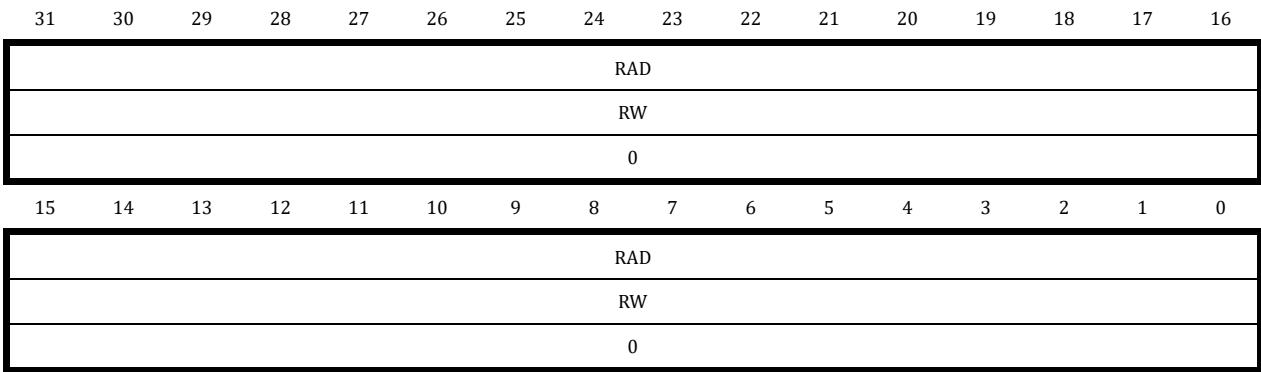
当CPU需要使用DSP除法器时，应首先保证DSP处于暂停状态。先向DSP写入被除数，后写入除数；写入除数可以触发一次除法操作，32/16位除法需要8周期完成，期间读取除法结果DSP\_QUO或DSP\_REM会使CPU进入等待状态，等待除法计算完成并通过总线返回计算结果。

# 15.2.6 DSP 开方相关寄存器

# 15.2.6.1 DSP\_RAD

地址：0x4001\_5030

复位值：0x0

  
表15-16DSP被开方数寄存器

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:0] | RAD | DSP 被开方数寄存器 |

# 15.2.6.2 DSP\_SQRT

地址：0x4001\_5034

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SQRT | | | | | | | | | | | | | | |  |
| RO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 保留，读出时恒为 0 |
| [15:0] | SQRT | DSP 平方根寄存器 |

当CPU需要使用DSP开方器时，应首先保证DSP处于暂停状态。向DSP写入被开方数；写入被开方数可以触发一次开方操作，32位开方需要8周期完成，期间读取开方结果DSP\_SQRT会使CPU进入等待状态，等待开方计算完成并通过总线返回计算结果。

# 15.3 DSP指令集

# 15.3.1 Instruction Set Summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation | Description | Assembler | | | | Cycles |
| Add |  | ADD | Rd1 | Rs1 | Rs2 | 1 |
| 5bit Immediate | ADDI | Rd1 | Rs1 | #<l>1 |  |
| Subtract |  | SUB | Rd1 | Rs1 | Rs2 | 1 |
| Shift | Arithmetic right shift | ASR | Rd1 | Rs1 | Rs2 | 1 |
| 5bit Immediate | ASRI | Rd1 | Rs1 | #<l>1 |  |
| Logical left shift | LSL | Rd1 | Rs1 | Rs2 | 1 |
| 5bit Immediate | LSLI | Rd1 | Rs1 | #<l>1 |  |
| Multiply and accumulation |  | MAC | Rs1 | Rs2 | Rs3 | 1 |
| 5bit Immediate | MACI | Rs1 | Rs2 | #<l>1 |  |
| Divide |  | DIV | Rd1 | Rs1 | Rs2 | 10 |
| Saturation |  | SAT | Rd1 | Rs1 | Rs2 | 1 |
| 4bit Immediate | SATI | Rd1 | #<l>1<l>1 |  |  |
| Cordic | SIN/COS | SIN\_COS | Rd1 | Rd2 | Rs1 | 8 |
| Arctan/Module | ARCTAN | Rd1 | Rs1 | Rs2 | 8 |
| Square root | Square root | SQRT | Rd1 | Rs1 | #<l>8 |  |
| Memory access | Load word | LDRWI | Rd1 | #<l>1 |  |  |
| Load double half words | LDRDHI | Rd1 | Rd2 | #<l>1 |  |
| Store word | STRWI | Rs1 | #<l>1 |  |  |
| Store double half words | STRDHI | Rs1 | Rs2 | #<l>1 |  |
| Branch | Unconditional Jump | JUMP | Rs1 | #<l>2 |  |  |
| Immediate | JUMPI | #<l>2 | | |  |
| Jump if less than or equal to | JLE | Rs1 | Rs2 | Rs3 | 2 |
| Immediate | JLEI | Rs1 | Rs2 | #<l>2 |  |
| Miscellaneous | Generate IRQ and Pause DSP | IRQ | #<l>1 | | |  |

DSP使用16bit定长编码指令，因为通用寄存器共8个，所以寄存器编码需要3bit。其中大部分指令为3操作数指令，包括两个源操作数寄存器和一个目的操作数寄存器；部分指令包含立即数；部分指令会涉及4个操作数，以乘加（MAC）操作为例， ，由于指令长度不够表示4个寄存器，将Rd固定为R7，在指令编码中不显示。其余4操作数指令还有ARCTAN/DIV。具体操作数分配见下文指令的详细解释。

# 15.3.2 ADD

15.3.2.1编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Rs2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 |

15.3.2.2汇编语法

ADD Rd1 Rs1 Rs2

15.3.2.3伪代码

结果防溢出保护，上溢出后 下溢出后

# 15.3.3 ADDI(reserved)

带立即数的加法指令在此版本DSP中保留，未实现。

15.3.3.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | Imm | | | | Rd1 | | | Rs1 | | | |

立即数为5bit有符号数，表示数值范围是

15.3.3.2汇编语法

ADDI Rd1 Rs1 Imm

15.3.3.3伪代码

结果防溢出保护，上溢出后

下溢出后

# 15.3.4 SUB

15.3.4.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | Rs2 |  | Rd1 |  | Rs1 |  |  |  |  |

15.3.4.2汇编语法

SUB Rd1 Rs1 Rs2

15.3.4.3伪代码

结果防溢出保护，上溢出后 下溢出后

# 15.3.5 ASR

15.3.5.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | Rs2 |  | Rd1 |  | Rs1 |  |  |  |  |

15.3.5.2汇编语法

ASR Rd1 Rs1 Rs2

15.3.5.3伪代码

算术右移指令只支持 bit的右移。

# 15.3.6 ASRI

15.3.6.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | Imm | | | | Rd1 | | | Rs1 | | | |

立即数为5bit无符号数，表示数值范围是

15.3.6.2汇编语法

ASRI Rd1 Rs1 Imm

15.3.6.3伪代码

带立即数的算术右移指令只支持0~31bit的右移。

# 15.3.7 LSL

15.3.7.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | Rs2 | | Rd1 | | | Rs1 | | | |

15.3.7.2汇编语法

LSL Rd1 Rs1 Rs2

15.3.7.3伪代码

逻辑左移只支持 bit左移。

# 15.3.8 LSLI

15.3.8.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 1 | Imm | Rd1 | Rs1 |

立即数为5bit无符号数，表示数值范围是

15.3.8.2汇编语法

LSLI Rd1 Rs1 Imm

15.3.8.3伪代码

带立即数的逻辑左移只支持 bit左移。

15.3.9 MAC

15.3.9.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Rs3 | Rs2 | Rs1 |  |  |  |  |  |

15.3.9.2汇编语法

MAC Rs1 Rs2 Rs3

15.3.9.3伪代码

结果防溢出保护，

上溢出后

下溢出后

其中Rs1、Rs2为16位有符号数，Rs3为32位有符号数。注意，作为操作数的Rs1和Rs2不要超出16位有符号数的表达范围。当Rs3为R0时，MAC可以当做乘法指令MUL使用。

# 15.3.10 MACI (reserved)

带立即数的乘加指令在此版本DSP中保留，未实现。

15.3.10.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | Imm | | | | Rs2 | | | Rs1 | | | |

立即数为5bit有符号数，表示数值范围是- 16~15.

15.3.10.2汇编语法

MACI Rs1 Rs2 Imm

15.3.10.3伪代码

结果防溢出保护，上溢出后 下溢出后

15.3.11 DIV

15.3.11.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd2 | Rd0 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Ri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rli | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rir | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrit | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrib | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rii | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | R ri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rni | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | <nl> |

15.3.11.2汇编语法

DIV Rd2 Rd1 Rs1

15.3.11.3伪代码

除法指令需要10个周期才会结束提交，在除法进行计算的过程中，DSP不应该再发射其他的多周期指令，即同一时刻只能由一个多周期指令在long- pipeline中。其他多周期指令包括三角函数指令以及开方指令。多周期指令可以后台运行，即多周期指令运算的同时，DSP可以继续执行其他的单周期指令。但同一时间只能有一条多周期指令在后台运行。在多周期指令计算期间，DSP仍可以使用多周期指令的目的操作数，但需要注意是当多周期计算完成提交结果的时候，目的操作数寄存器会被改写。

其中Rd7为32为有符号数，Rs1,为16为有符号数，Rd1为32位有符号数，Rd2为16位有符

号数。

# 15.3.12 SAT

15.3.12.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Rs2 |  | Rd1 |  | Rs1 |  |  |  |

15.3.12.2汇编语法

SAT Rd1 Rs1 Rs2

15.3.12.3伪代码

If Rd1 Rs1;else if Rd1 Rs2

# 15.3.13 SATI (reserved)

带立即数的饱和指令在此版本DSP中保留，未实现。

15.3.13.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | Imm2 | | | Imm1 | | | Rs1 | | | |  |

15.3.13.2汇编语法

SATI Rd1 Imm1 Imm2

15.3.13.3伪代码

If Rd1 mm1;else if Rd1 mm2

# 15.3.14 SIN\_COS

15.3.14.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd2 | Rd2 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd0 | Rd2 | Rd0 | Rd2 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rd1 | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rdi | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Ri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rli | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrit | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rir | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrib | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rii | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | R ri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rrri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | Rri | <nl> |

# 15.3.14.2汇编语法

SIN\_COS Rd1 Rd2 Rs1

15.3.14.3伪代码

Sin/cos指令8周期结束并提交，在其执行期间，DSP不应该发射其他的多周期指令。

Rd1 cos(Rs1);Rd2 sin(Rs1)

# 15.3.15 ARCTAN

15.3.15.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | Rs2 |  | Rd1 |  | Rs1 |  |  |  |  |

15.3.15.2汇编语法

ARCTAN Rd1 Rs1 Rs2

15.3.15.3伪代码

ARCTAN指令8周期结束并提交，在其执行期间，DSP不应该发射其他的多周期指令。Rd1 arctan(Rs2/Rs1); sqrt(Rs12+Rs22)

# 15.3.16 SQRT

15.3.16.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Rd1 | Rs1 |  |  |  |

15.3.16.2汇编语法

SQRT Rd1 Rs1

15.3.16.3伪代码

开方指令8周期结束并提交，在其执行期间，DSP不应该发射其他的多周期指令。

Rd1 sqrt(Rs1)

# 15.3.17 LDRWI

15.3.17.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 0 | Imm | | | | Rd1 | 0 | 0 | 0 | 0 |  |  |  |

15.3.17.2汇编语法

LDRWI Rd1 Imm

15.3.17.3伪代码

Rd1 word(SRAM[imm])

由于Load指令都是立即数指令，访问的数据地址可以在译码阶段产生，因此load操作可以在一个周期内完成。而类似的load指令，CPU需要访问寄存器来计算地址，因此需要2个周期。

立即数imm的表达范围是 。因为DSP datamem是64个32bitword构成。

# 15.3.18 LDRDHI

15.3.18.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 1 | Imm | | | | Rd1 | | | Rd2 | | | |  |

15.3.18.2汇编语法

LDRDHI Rd1 Rd2 Imm

15.3.18.3伪代码

LDRDHI Rd1 Rd2

word(SRAM[Imm]),

从datamem取回的32bit数据高16位会进行符号扩展为32位，然后赋值给Rd1，低16位会进行符号扩展为32位，然后赋值给Rd2。

立即数imm的表达范围是 。因为DSP datamem是64个32bitword构成。

# 15.3.19 STRWI

15.3.19.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 1 | Imm | | | | | 0 | 0 | 0 | Rs1 | | | |

15.3.19.2汇编语法

STRWI Rs1 Imm

15.3.19.3伪代码

word{SRAM[imm]}=Rs1

Store指令的数据来自寄存器，因此即使待写入地址在译码阶段产生也无法立即完成Store操作。需要将地址锁存一个周期，然后与写入的数据一起送至datamemory接口。因此Store指令后续如果立刻连接load指令会出现访问冲突。在设计汇编程序时应避免出现此种指令序列。如果STR指令后必须是LDR指令，可以在两者之间插入ADDR0R0R0指令作为指令bubble。

立即数imm的表达范围是 。因为DSPdatamem是64个32bitword构成。

# 15.3.20 STRDHI (reserved)

带立即数的存储双半字指令在此版本DSP中保留，未实现。

15.3.20.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | Imm | | | | | Rs1 | | | Rs2 | | | |

15.3.20.2汇编语法

STRDHI Rs1 Rs2 Imm

15.3.20.3伪代码

word{SRAM[imm]}={Rs1, Rs2}

立即数imm的表达范围是 。因为DSPdatamem是64个32bitword构成。

# 15.3.21JUMP(reserved)

基于寄存器的跳转指令在此版本DSP中保留，未实现。

15.3.21.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Rs1 |  |

15.3.21.2汇编语法

JUMP Rs1

15.3.21.3伪代码

15.3.22JUMPI

15.3.22.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |

15.3.22.2汇编语法

JUMPI Imm

15.3.22.3伪代码

15.3.23JLE

15.3.23.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | Rs3 | Rs2 | Rs1 |  |  |  |  |  |  |

15.3.23.2汇编语法

JLE Rs1 Rs2 Rs3

15.3.23.3伪代码

15.3.24 JLEI

15.3.24.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | Imm | | | Rs2 | | Rs1 | | | | | |

15.3.24.2汇编语法

JLEI Rs1 Rs2 Imm

15.3.24.3伪代码

15.3.25 IRQ

15.3.25.1指令编码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

15.3.25.2汇编语法

IRQ

15.3.25.3伪代码

IRQ指令会产生中断，等待CPU处理，并将DSP暂停运行。

# 15.3.26 R（仅用于模拟器）

15.3.26.1指令编码

无

15.3.26.2汇编语法

R50x5555 #Assign0x5555toR5 R120 #Assign20toR1

15.3.26.3伪代码

调试指令，仅在DSPEmulator中使用。可用于在DSP程序的任意位置将R1~R7置为任意指定数值。

# 15.3.27BREAK（仅用于模拟器）

15.3.27.1指令编码

无

15.3.27.2汇编语法

BREAK

15.3.27.3伪代码

调试指令，仅在DSPEmulator中使用。可用于在DSP程序的任意位置插入断点并打印R1~R7寄存器值。断点命中后按回车继续程序运行。

# 15.3.28END（仅用于模拟器）

15.3.28.1指令编码

无

15.3.28.2汇编语法

END

# 15.3.28.3伪代码

类似于断点指令，仅在DSP Emulator中使用。在指令模拟中遇到该指令即中断模拟器的运行，直接打印核心寄存器值。

# 15.4应用指南

# 15.4.1 访存寻址

DSP使用立即数寻址，由于DATAMEM地址空间有限，STR和LDR指令中的6bit立即数可以直接表示DATAMEM全部地址偏移。

以如下DATAMEM内容为例，第一行数据0x01000100对应0x0地址，第二行数据0x30005000对应0x1地址，第三行数据0x0003FFF8对应0x2地址。

使用LDRDHIR1R20x1可对R1,R2进行赋值 ，

使用STRWIR30x3可以将R3的32bit数据写入0x3地址并覆盖掉数据0xFFFE000。

需要注意的是，虽然DSP访存寻址每增加1是增加4Byte，对于CPU来说，每增加4地址才是增加4Byte。因此CPU访存DSPDATAMEM时，应该按照DSP\_DATA\_MEM\_BASE+offset\*4的方式进行计算。以如下DATAMEM内容为例，第二行数据0x30005000对应的CPU寻址地址为0x4001\_4804，第三行数据0x0003FFF8对应的CPU寻址地址为0x4001\_4808。

DATAMEM:

0x00100010 0x30005000 0x0003FFF8 0xFFFFE000 0x30004000 0x7FFFFFFF 0x7FFFFFF 0xF0000003 0x00007FFF #8 0x00008000 0x80000000 0x00000000

# 15.4.2 Load after Store

DSP STR 指令之后不能立即使用 LDR，可在其间插入 “ADD R0 R0 R0” 作为 “NOP” 指令使用。

# 15.4.3 多周期指令的的延迟提交

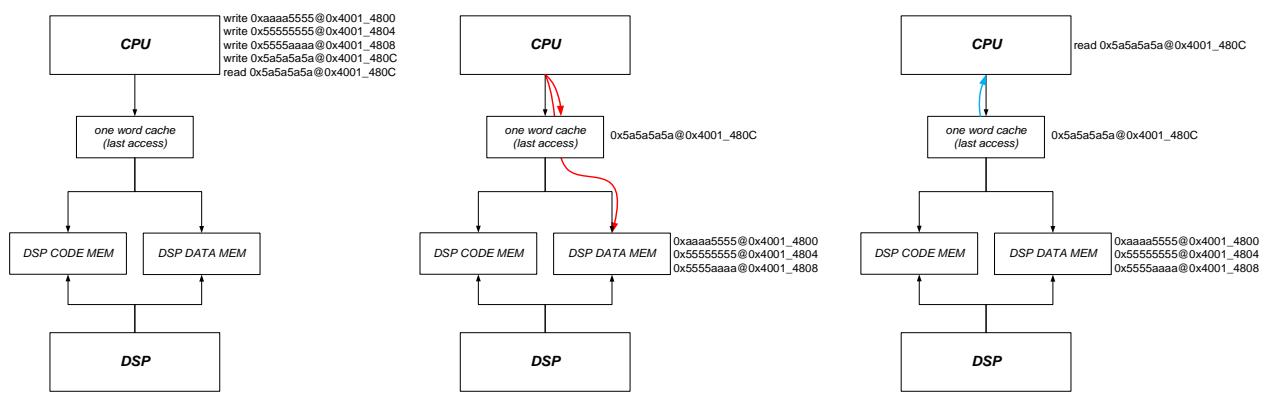
DSP的多周期算术指令包括除法、开方、三角函数。除法指令需要10个总线周期（96MHz）完成。开方指令需要8个总线周期（96MHz）完成。三角函数指令需要8个总线周期（96MHz）完成。

为了充分利用DSP性能，允许多周期指令后台执行，即多周期指令计算期间DSP仍可其他指令，而不会阻塞流水线。这需要在编程时对指令序列有所考虑，在多周期指令的发射和结果使用之间插入其他不相关指令。如下指令序列，其中ADD R0 R0 R0作用类似NOP，在实际应用中可以替换为其他指令。

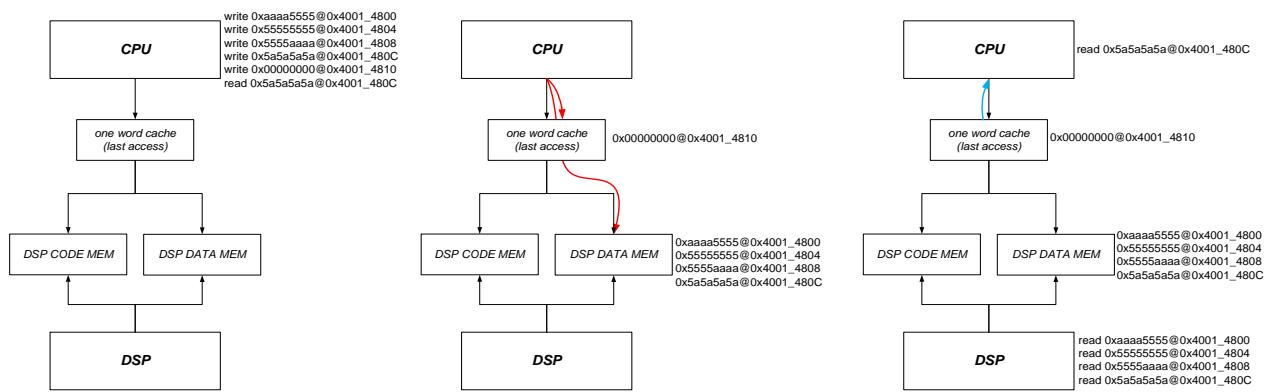
SIN\_COS R1 R2 R3 ADD R0 R0 R0 ADD R0 R0 R0 ADD R0 R0 R0 ADD R0 R0 R0 ADD R0 R0 R0 ADD R0 R0 R0 ADD R0 R0 R0 LDRWI R4 0x10 MAC R1 R4 R0

# 15.4.4 数据填装

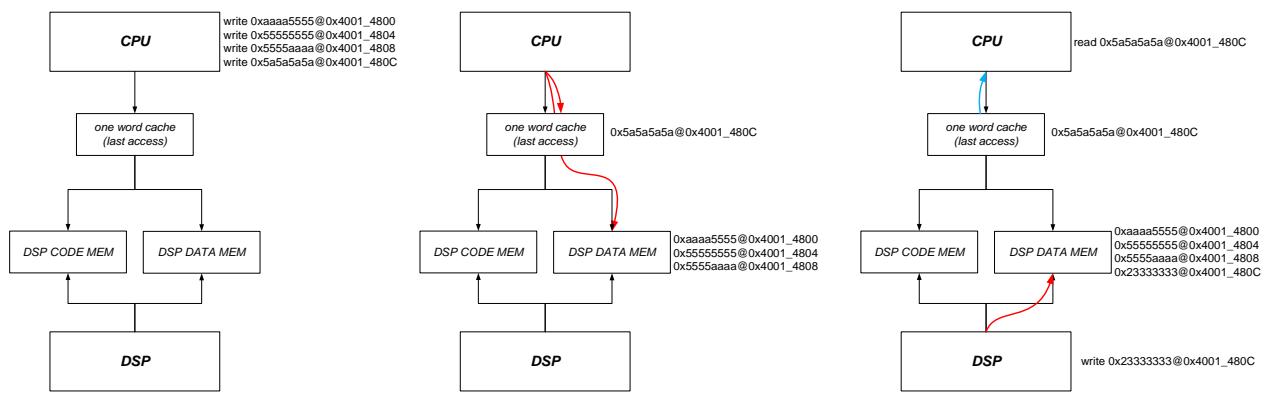
15.4.4 数据填装DATA MEM 支持按字、半字、字节写入。为了解决 AHB 总线协议写后立即读问题，SRAM 端口处有一个 word 深度的 cache。DSP CODE MEM 和 DATA MEM 由于允许 CPU 和 DSP 两个主设备访问，就存在一致性问题。举例来说，如图 15- 2 所示。CPU 先向 DSP DATAMEM 顺序写入 4 个 word，然后读取最后一个 word。由于 cache 的存在，最后一个 word 实际上留存在 cache 中，而没有真正写入 DSP DATAMEM，但 CPU 读取这一 word 数据时，命中 cache，所以直接从 cache 返回数据，读回的数据是正确的。但 DSP 读取 DATAMEM 时，则只能读取到前 3 个 word，留存在 cache 中的最后一个 word 是访问不到的。

  
图15-2 CPU访问DSPMEM一致性问题

因此要求最后一次写入后需要再写任意一个其他地址，从而将停留在cache中的数据真实地写入DSPMEM。对于单一主设备的系统来说，所有的读写操作都通过接口上的cache，所以不会出现一致性问题，但由于DSP中的MEM被CPU写入后是由DSP经由另外的通道进行访问，所以需要通过一次冗余的写入操作实现cache数据推送，如图15- 3所示。

  
图15-3增加Dummy写操作解决DSPMEM一致性问题

同理，由于CPU最后一次读写的地址信号会留存在Cache中，如果之后DSP在运行中修改了DATA MEM数据。而后CPU读取DSP DATA MEM数据，如果命中CPU最后一次读写的地址，CPU会去cache中取数据，而不会从DSP DATA MEM取数，即CPU认为cache数据仍然有效，如。因此应用上应该CPU最后一次写入一个dummy数据方式来规避此种情况出现。

  
图15-4 CPU因命中cache无法读取DSP DATA MEM真实数据

# 16 I2C

# 16.1概述

16.1 概述I2C 总线接口连接微控制器和串行 I2C 总线。它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁和定时。支持标准和快速两种模式。根据特定设备的需要，可以使用 DMA 以减轻 CPU 的负担。

# 16.2主要特性

多主机功能：该模块既可做主设备也可做从设备。

I2C主设备功能：产生时钟、START和STOP事件。

I2C从设备功能：可编程的I2C硬件地址比较（仅支持7位硬件地址）、停止位检测。

根据系统分频，实现不同的通讯速度。

状态标志：发送器/接收器模式标志、字节发送结束标志、I2C总线忙标志。

错误标志：主模式时的仲裁丢失、地址/数据传输后的应答(ACK)错误、检测到错位的起始或停止条件。

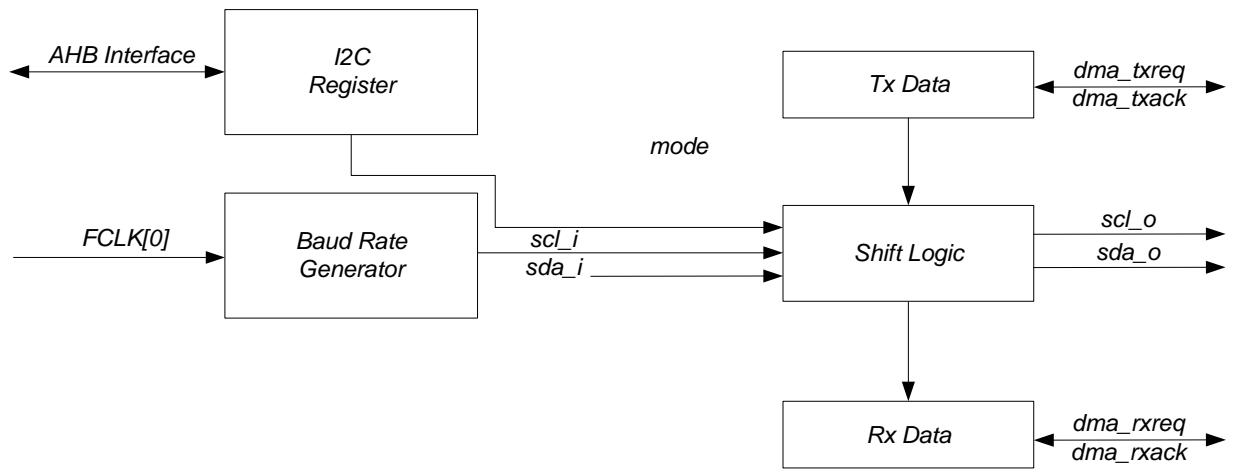
一个中断向量，包含五个中断源：总线错误中断源、完成中断源、NACK中断源、硬件地址匹配中断源和传输完成中断源。

具单字节缓冲器的DMA。

# 16.3功能描述

# 16.3.1 功能框图

本接口采用同步串行设计，实现CPU同外部设备之间的I2C传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。

  
图16-1 I2C模块顶层功能框图

I2C接口同外界通讯只有SCL和SDA两根信号线。SDA为双向复用信号线，受到sda\_oe控制。模块级，I2C接口信号包括，scl\_i，sda\_i，scl\_o，sda\_o和sda\_oe。

scl\_i：时钟信号。当I2C接口配置为从模式时，此为I2C总线的时钟输入信号。

sda\_i：数据信号。当I2C接口接收数据时（无论主模式还是从模式），此为I2C总线的数据输入信号。

scl\_o：时钟信号。当I2C接口配置为主模式时，此为I2C总线的时钟输出信号。

sda\_o：数据信号。当I2C接口发送数据时（无论主模式还是从模式），此为I2C总线的数据输出信号。

sda\_oe：数据使能信号。当sda\_o输出时，sda\_oe有效；当sda\_i输入时，sda\_oe无效。

# 16.3.2 功能说明

12C模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。可以开启或禁止中断。接口通过数据引脚(SDA)和时钟引脚(SCL)连接到I2C总线。

# 16.3.2.1 模式选择

接口可以下述4种模式中的一种运行：

从发送模式从接收模式主发送模式主接收模式

I2C接口默认主从均不使能。接口根据配置情况，进入主模式或者从模式。当仲裁丢失或产生停止信号时，主模式自动释放总线并产生相应异常中断。允许多主机功能。

主模式时，I2C接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

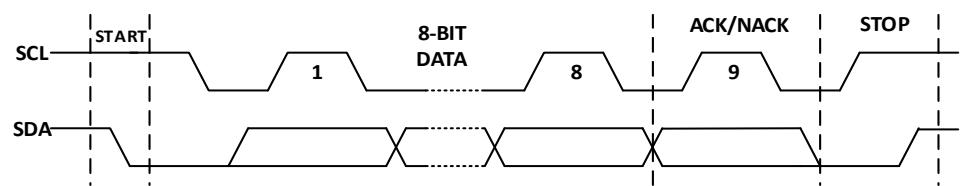
从模式时，I2C接口能识别它自己的地址(7位)。软件能够控制开启或禁止硬件地址比较功能，

硬件地址比较功能可降低CPU的负担。只有地址匹配才通知CPU进行相关处理。

数据和地址按8位/字节进行传输，高位在前。跟在起始条件后的1个字节是地址。地址只在主模式发送。

在一个字节传输的8个时钟后的第9个时钟期间，接收器必须回送一个应答位(ACK)给发送器。

软件可以开启或禁止应答(ACK)，并可以设置I2C接口的地址。

  
图16-2基本I2C传输时序图

I2C接口没有FIFO，若一次性发送大量数据，为了降低CPU的负担，需DMA配合。I2C接口支持DMA传输（多字节传输）和非DMA传输（单字节传输)。上述四种传输模式进一步扩展为：

从模式单字节发送，从模式DMA发送从模式单字节接收，从模式DMA接收主模式单字节发送，主模式DMA发送主模式单字节接收，主模式DMA接收

一般情况下，非DMA方式时，一次传输一个字节（可反复单次传输，需软件介入提供数据)。DMA方式，一次连续传输可以多字节(最大不超过16字节，极端情况一次传输一个字节，因无FIFO，每次DMA请求，仅传输一个字节，多轮完成本次数据传输)。

上述所有模式，遵循如下基本原则：

单字节发送，中断将在8- bit数据发送完毕且收到响应后（ACK/NACK均可）产生。- 单字节接收，中断将在8- bit数据接收完毕后产生。- DMA发送，正常情况下，中断将在数据发送完毕且收到响应后（ACK/NACK均可）产生。- DMA接收，正常情况下，中断将在数据接收完毕后产生。- 当I2C接口配置为主模式时，检测到错误后，I2C接口会主动释放总线，恢复到起始状态并产生中断信号。

# 16.3.2.2I2C接口从模式

默认情况下，I2C接口主模式和从模式均关闭。若工作在从模式，需使能从模式。为了产生正确的时序，必须通过系统寄存器SYS\_CLK\_DIV0设定I2C接口的工作时钟频率，I2C接口时钟基于系统高速主时钟进行分频，SYS\_CLK\_DIV0是I2C接口工作时钟的分频系数。

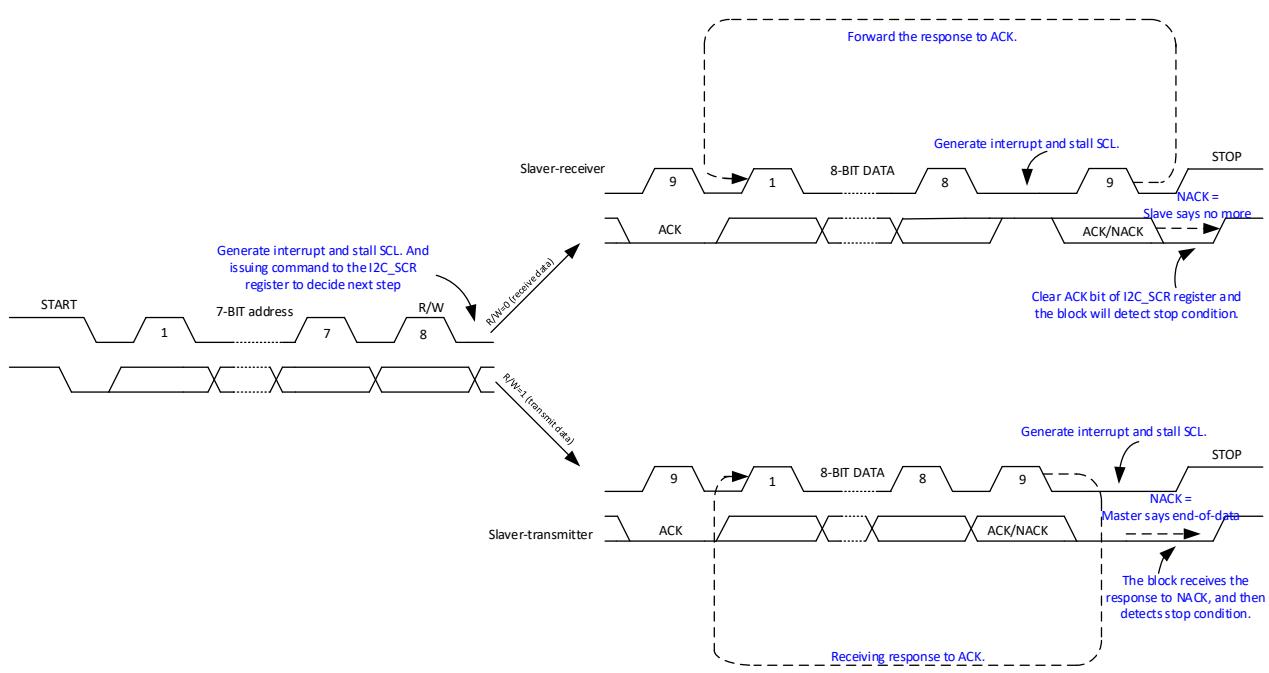
从模式下，I2C接口时刻在监控总线上的信号。一旦检测到起始条件，其将保存地址位数据和读写位数据。

* 从模式下，若硬件地址匹配功能开启，只有地址匹配的情况下，才会产生中断，通知 CPU 进行后续处理。若没有开启，每次收到地址及读写位数据，都将产生中断。- 从模式下，单字节接收模式。每次收到一个字节的数据后，产生中断，此时 I2C 接口可拉低 SCL，直至中断完成，继续后续操作。- 从模式下，单字节发送模式。每次发送一个字节完毕后且收到响应（ACK/NACK），产生中断，此时 I2C 接口可拉低 SCL，直至中断完成，继续后续操作。- 从模式下，DMA 接收模式。每次收到 SIZE 约定后的数据，产生中断，此时 I2C 接口可拉低 SCL，直至中断完成。- 从模式下，DMA 发送模式。每次发送 SIZE 约定后的数据并收到响应（ACK/NACK），产生中断，此时 I2C 接口可拉低 SCL，直至中断完成。

# 16.3.2.2.1 从模式单字节传输

单字节传输，并不意味着仅仅传输一个字节数据，其含义为每次传输完一个字节的数据后，将产生中断判断是否还要继续传输。单字节传输的极端情况是仅传输一个字节的数据。下图为单字节传输的总线示意图。从图可知，一般单字节传输的流程如下：

* 地址匹配，产生地址匹配中断，准备开始传输。- 若是接收模式，一个字节接收完毕后，产生中断，软件判断是否继续接收，返回 ACK/NACK 响应。- 若是发送模式，一个字节发送完毕后，等待响应（ACK/NACK），产生中断，根据响应判断后续操作。- 获得总线 STOP 事件，本次传输完成。



# 16.3.2.2.2 从模式单字节发送

地址匹配后，从发送器将字节从I2C\_DATA寄存器经由内部移位寄存器发送到SDA线上。在I2C\_DATA数据没有准备好之前，从设备可拉底SCL，直到待发送数据已写入I2C\_DATA寄存器。I2C接口在发送完毕每个字节后都执行下列操作：

如果接收到ACK位，装载下一个字节数据，继续传输。装载过程中，可拉底SCL。 如果接收到NACK位，停止下一个字节的装载。 等待STOP事件，停止本次传输。

# 16.3.2.2.3 从模式单字节接收

地址匹配后，从接收器将通过内部移位寄存器从SDA线接收到的数据存入I2C\_DATA寄存器。I2C接口在接收到每个字节后都执行下列操作：

如果设置了ACK位，在一个字节接收完毕后，产生一个ACK应答脉冲。 如果清除了ACK位，在一个字节接收完毕后，产生一个NACK应答脉冲。 等待STOP事件，结束本次传输。

# 16.3.2.3从模式DMA传输

从模式下，一般仅配置I2C时钟、从地址以及硬件地址匹配使能，等待总线有访问请求后，根据芯片实际情况决定是否响应本次传输请求。DMA传输，其含义为每次传输多个字节的数据后，将产生中断判断是否还要继续传输。DMA传输的极端情况是仅传输一个字节的数据。DMA传输，一般建议开启硬件地址比较功能，NACK中断，传输完成中断。一般DMA传输的流程如下：

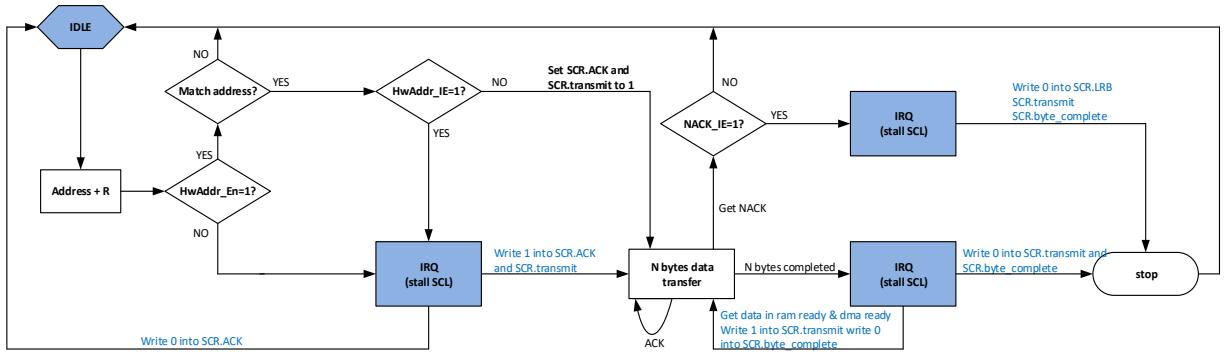
配置I2C从地址，使能I2C中断（可使能硬件地址比较中断)。地址匹配，产生I2C地址匹配中断，在中断处理函数中，配置DMA，准备好发送数据或者准备好接收地址。然后写I2CSCR，准备开始传输或者停止本次传输。 若是接收模式，I2C\_BOR\_BURST\_SIZE约定的字节接收完毕后，产生中断，软件判断是否继续接收，返回ACK/NACK响应。 若是发送模式，I2C\_BOR\_BURST\_SIZE约定的字节发送完毕后，等待响应（ACK/NACK），产生中断，根据响应判断后续操作。 获得总线完成标志，本次传输完成。

# 16.3.2.3.1 从模式DMA发送

地址匹配后，配置完毕DMA。通过发送DMA请求，将字节从RAM搬移到I2C\_DATA寄存器，然后经由内部移位寄存器发送到SDA线上。在I2C\_DATA数据没有准备好之前，从设备可拉底SCL，

直到待发送数据已写入 I2C\_DATA 寄存器。从模式下发送数据，需要软件协助触发第一次 DMA 搬移，配置 I2C\_BCR.BYTE\_CMPLT 为 1 即可。I2C 接口在发送完毕 I2C\_BCR.BURST\_SIZE 约定的字节数据后都执行下列操作：

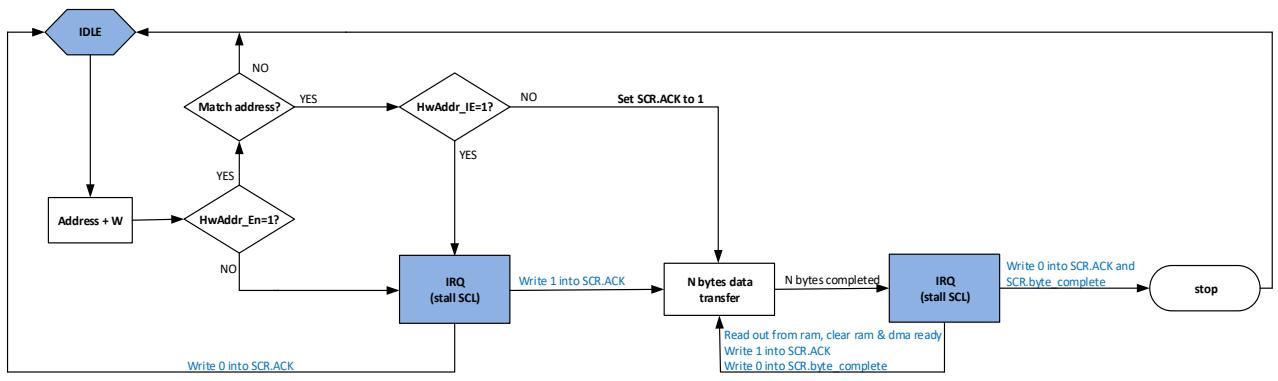
如果接收到ACK位，配置DMA，准备下一批数据，继续传输。准备过程中，可拉低SCL。 如果接收到NACK位，停止准备下一批数据，停止本次传输。

  
图16-4从模式下多字节发送示意图

# 16.3.2.3.2 从模式DMA接收

地址匹配后，配置好DMA，从SDA线接收到的数据先存入I2C\_DATA寄存器，然后通过DMA搬移到RAM。I2C接口在接收到一个字节后都将执行DMA请求。完成I2C\_BCR.BURST\_SIZE约定的数据传输后，执行下列操作：

如果设置了ACK位，在I2C\_BCR.BURST\_SIZE约定数据接收完毕后，产生一个ACK应答脉冲。 如果清除了ACK位，在I2C\_BCR.BURST\_SIZE约定数据接收完毕后，产生一个NACK应答脉冲。

  
图16-5从模式下多字节接收示意图

# 16.3.2.4I2C接口主模式

默认情况下，I2C接口主模式和从模式均关闭。若工作在主模式，需使能主模式。为了产生正确的时序，必须在系统寄存器CLK\_DIV0中设定I2C接口的工作时钟。

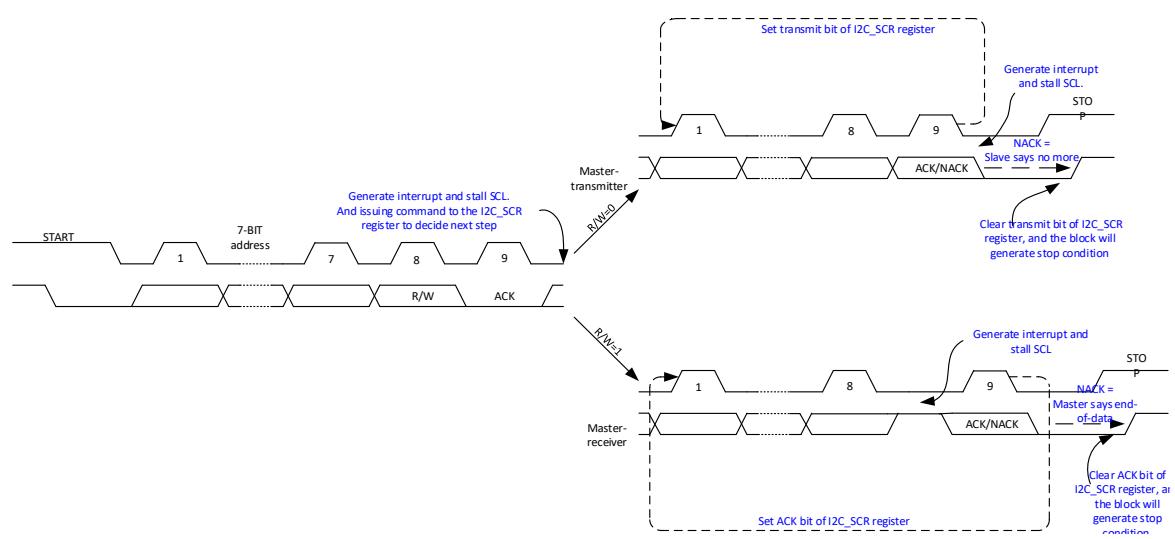
I2C接口执行主模式传输之前，需要判断总线是否空闲。可读取I2C\_MSCR寄存器的BIT3，查

询当前总线状态。若总线处于忙的状态，可以开启 I2C 中断，通过收到 STOP 中断事件判断总线是否空闲下来。只有空闲状态下，才能正常发送 START 状态，以及后续的数据。

# 16.3.2.4.1 主模式单字节传输

单字节传输，并不意味着仅仅传输一个字节数据，其含义为每次传输完一个字节的数据后，将产生中断判断是否还要继续传输。单字节传输的极端情况是仅传输一个字节的数据。图 6 为单字节传输的总线示意图。从图可知，一般单字节传输的流程如下：

* 判断总线是否空闲，若空闲，准备开始传输。- 若是接收模式，一个字节接收完毕后，产生中断，软件判断是否继续接收，返回 ACK/NACK 响应。- 若是发送模式，一个字节发送完毕后，等待响应（ACK/NACK），产生中断，根据响应判断后续操作。- 发送总线 STOP 事件，本次传输完成。

  
图 16-6 主模式下单字节传输示意图

# 16.3.2.4.2 主模式单字节发送

开始传输后，I2C 接口将字节从 I2C\_DATA 寄存器经由内部移位寄存器发送到 SDA 线上。在 I2C\_DATA 数据没有准备好之前，主设备可不产生 SCL 时钟信号，直到待发送数据已写入 I2C\_DATA 寄存器。I2C 接口在发送完毕每个字节后都执行下列操作：

* 如果接收到 ACK 位，装载下一个字节数据，继续传输。装载过程中，可拉底 SCL。
* 如果接收到 NACK 位，停止下一个字节的装载。
* 产生 STOP 事件，结束本次传输。

# 16.3.2.4.3 主模式单字节接收

开始传输后，I2C接口将通过内部移位寄存器从SDA线接收到的数据存入I2C\_DATA寄存器。I2C接口在接收到每个字节后都执行下列操作：

如果设置了ACK位，在一个字节接收完毕后，产生一个ACK应答脉冲。 如果清除了ACK位，在一个字节接收完毕后，产生一个NACK应答脉冲。 产生STOP事件，结束本次传输。

# 16.3.2.4.4 主模式DMA传输

DMA传输，其含义为每次传输多个字节的数据后，将产生中断判断是否还要继续传输。DMA传输的极端情况是仅传输一个字节的数据。DMA传输，一般建议开启NACK中断，传输完成中断。一般DMA传输的流程如下：

总线空闲，准备开始传输。- 若是接收模式，I2C\_BCR.BURST\_SIZE约定的字节接收完毕后，产生中断，软件判断是否继续接收，返回ACK/NACK响应。- 若是发送模式，I2C\_BCR.BURST\_SIZE约定的字节发送完毕后，等待响应（ACK/NACK），产生中断，根据响应判断后续操作。- 发送STOP事件，本次传输完成。

# 16.3.2.4.5 主模式DMA发送

总线空闲，配置完毕DMA。通过发送DMA请求，将字节从RAM搬移到I2C\_DATA寄存器，然后经由内部移位寄存器发送到SDA线上。在I2C\_DATA数据没有准备好之前，主设备可不产生SCL时钟，直到待发送数据已写入I2C\_DATA寄存器。I2C接口在发送完毕SIZE约定的字节数据后都执行下列操作：

如果接收到ACK位，配置DMA，准备下一批数据，继续传输。准备过程中，可拉低SCL。

如果接收到NACK位，停止加载下一批数据。

如果本次数据发送完毕，停止后续发送。

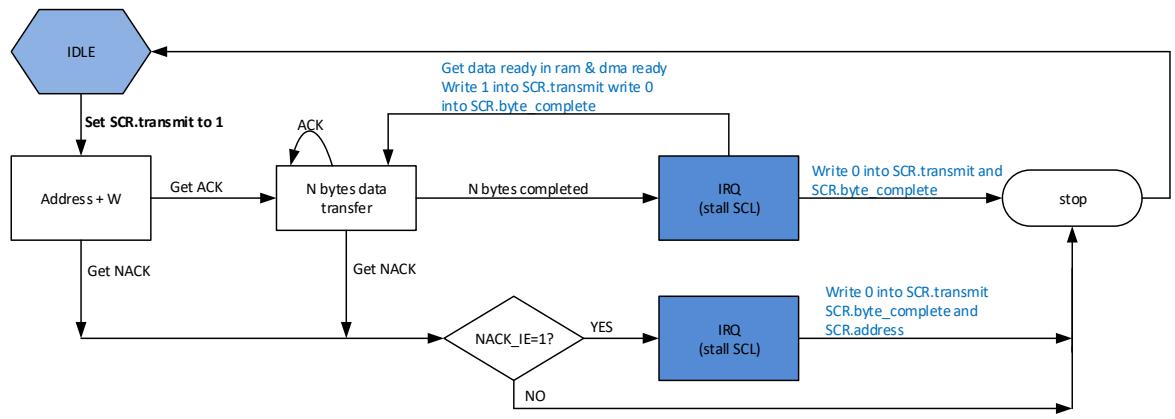
产生STOP事件，停止本次传输。

异常情况是：

若从设备地址不匹配或者从设备没有准备好，此时从设备将返回NACK

主设备产生STOP事件，停止本次传输。

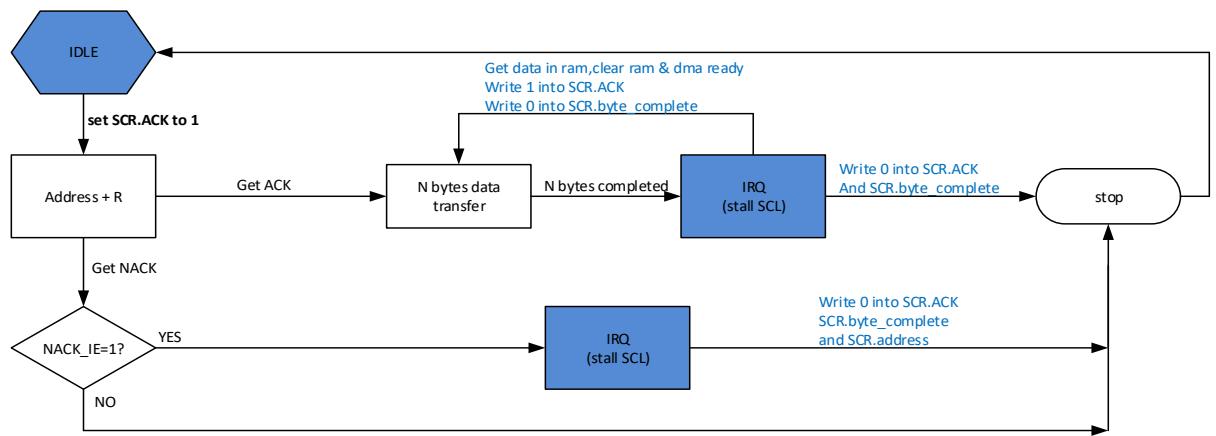
等待一段时间后，重新配置I2C寄存器，关闭DMA对应的通道使能信号，重新配置DMA寄存器，再次发送传输请求。关闭DMA对应通道是因为I2C有预取，DMA已经不是初始状态。

  
图16-7主模式下多字节发送示意图

# 16.3.2.4.6 主模式DMA接收

总线空闲，配置好DMA从SDA线接收到的数据先存入I2C\_DATA寄存器，然后通过DMA搬移到RAM。I2C接口在接收到一个字节后都将执行DMA请求。完成I2C\_BCR.BURST\_SIZE约定的数据传输后，执行下列操作：

* 如果设置了ACK位，在I2C\_BCR.BURST\_SIZE约定数据接收完毕后，产生一个ACK应答脉冲。- 如果清除了ACK位，在I2C\_BCR.BURST\_SIZE约定数据接收完毕后，产生一个NACK应答脉冲。- 产生STOP事件，停止本次传输。异常情况是：- 若从设备地址不匹配或者从设备没有准备好，此时从设备将返回NACK。- 主设备产生STOP事件，停止本次传输。- 等待一段时间后，重新配置I2C寄存器，再次发送传输请求。因为上一次没有收到有效数据，DMA并没有任何动作，所以不用重置DMA。

  
图16-8主模式下多字节接收示意图

# 16.3.2.5I2C总线异常处理

在一个地址或数据字节传输期间，当I2C接口检测到一个外部的停止或起始条件则产生总线错误。一般而言，产生总线错误是由于总线上有干扰、某些I2C设备没有同步于本I2C网络自行发送了START事件/STOP事件。根据I2C协议规定，发生总线错误的时候，在收到START事件/STOP事件后要重置本I2C设备的接口逻辑。对于从设备而言，这个操作是没有问题的；对于主设备而言，总线错误强行要求其释放总线并重置其I2C接口逻辑。因为主设备是不响应外部START和STOP事件的，发生总线错误后，需要中断处理函数处理本次异常，并指导主设备继续监视总线情况，以便后续执行I2C总线传输。

本I2C接口。主模式下，总线错误可被检测到同时总线错误中断也会产生；从模式下，总线错误将触发地址数据被接收，同时让I2C接口恢复空闲状态并产生中断。

# 16.3.2.6DMA传输

在大容量数据传输应用下，I2C接口支持DMA传输，减轻CPU的负担。一次传输，最大传输量为16字节，最小传输量为1字节。因I2C无FIFO，I2C发送一次DMA请求，DMA只能搬移一个字节数据。若要实现多字节搬移，DMA需配置为多轮搬移，每一轮搬移一个字节的方式。

在接收到新的数据后，硬件自动产生DMA请求，通过DMA模块将数据搬移到RAM中。在发送新数据前，硬件自动产生DMA请求，通过DMA模块将数据从RAM中搬移到I2C接口。

DMA传输需要配置DMA模块相应寄存器

因I2C接口支持DMA传输，也支持CPU传输。两者区别在于DMA传输，发送的数据来自DMA的搬移；CPU传输，发送的数据来自CPU的搬移。

DMA传输，推荐软件配置流程如下：

初始化DMA模块，将本次发送的数据来源，接收的数据去向配置好，传输长度配置完毕。

初始化GPIO模块，将I2C复用的GPIO配置完毕。

初始化I2C接口，I2C\_CFG/I2C\_BCR等寄存器配置完毕。

主模式下，触发I2C接口，进入发送状态；从模式下，等待主发送传输请求。

若I2C接口为从发送模式，需要考虑预取，I2C\_BCRB[4]为预取开关。一般从模式传输的时候，可以开启硬件地址比较(I2C\_A00R B[7])，可选择是否开启硬件地址比较中断使能(I2C\_BCR B[6])。

开启硬件地址比较中断，从设备收到的地址同自身匹配成功后，会产生中断。通知软件此时主设备请求获得从设备数据，软件判断是否接收，若接收就回ACK，软件需要I2C\_BCR B[4]置1，协助硬件预取第一次传输的数据；否则就回NACK。

关闭硬件地址比较中断，一旦总线上出现START事件，从设备硬件预取第一次传输数据，无论从设备是否地址匹配成功。匹配成功，也不会产生地址匹配中断，直接开始数据传输。匹配不成功，从设备不会传输数据。此时，若匹配不成功，因为I2C有预取的操作，为后续匹配成功后传输正常，需要对DMA进行清除操作。

# 16.3.2.7CPU传输

CPU传输，一次只能发送/接收1个字节，每次完成后需要通过中断或者轮询的方式判断传输是否完成。

CPU传输，推荐软件配置流程如下：

初始化GPIO模块，将I2C复用的GPIO配置完毕。- 初始化I2C接口，IE/CFG等寄存器配置完毕。- CPU触发I2C接口进入发送流程，发送的数据来自CPU对I2C\_DATA写入值。

# 16.3.2.8中断处理

I2C接口包含三种类型的中断事件，分别是：数据传输完成事件，总线错误事件、STOP事件、NACK事件和硬件地址匹配事件。

数据完成事件。当前数据传输完成，高电平有效，对I2CSCR的BIT0写0清除。- 总线错误事件。传输过程中，总线产生错误的START事件/STOP事件，高电平有效，对I2CSCR的BIT7写0清除。- STOP事件。当前数据传输完成，主设备发送STOP事件，从设备收到STOP事件并产生相应中断。高电平有效，对I2CSCR的BIT5写0清除。- NACK事件。发送端接收到NACK响应，表明接收端无法继续后续传输。高电平有效，对I2CSCR的BIT1写0清除。- 硬件地址匹配事件。从模式下接收到的地址同本设备地址匹配，产生相应中断。高电平有效，对I2CSCR的BIT3写0清除。- 使用DMA协助数据传输。若本模块为接收模式，I2C收到数据后还需要通过DMA搬移到RAM，此时数据最终完成是要看DMA是否搬移完成，若使用I2C的完成中断作为判断依据的话，推荐在中断处理函数中查询下DMA的状态。若本模块为发送模式，无此问题。直接使用I2C的完成中断作为判断依据即可。

# 16.3.2.9通讯速度设置

I2C模块的波特率时钟来自系统时钟的分频，分频寄存器为SYS模块的CLK\_DIV0。

I2C接口采用同步设计，需要对外部设备的信号进行同步采样，同步时钟为I2C接口工作时钟。数据和时钟信号的时钟频率为接口工作时钟/17。

I2C模块工作时钟频率 系统频率/

I2C波特率：I2C波特率 （系统频率/ ）/17

# 16.4 寄存器

# 16.4.1 地址分配

I2C 模块寄存器的基地址是 0x4001\_0400，寄存器列表如下：

表16-1I2C寄存器地址分配表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| I2C0\_ADDR | 0x00 | I2C 地址寄存器 |
| I2C0\_CFG | 0x04 | I2C 配置寄存器 |
| I2C0\_SCR | 0x08 | I2C 状态寄存器 |
| I2C0\_DATA | 0x0C | I2C 数据寄存器 |
| I2C0\_MSCR | 0x10 | I2C 主模式寄存器 |
| I2C0\_BCR | 0x14 | I2C DMA 传输控制寄存器 |

# 16.4.2 寄存器说明

16.4.2.1 I2C0\_ADDR 地址寄存器

地址：0x4001\_0400

复位值：0x0

表16-2地址寄存器I2C0\_ADDR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | ADDR\_CMP | ADDR | | | | | | | |  |
| RW | RW | | | | | | | |  |
| 0 | 0 | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | ADDR\_CMP | I2C 硬件地址比较使能开关，只有在 DMA 模式下开启才有效。默认值为 0。 1：使能 0：关闭 |
| [6:0] | ADDR | 从模式下，I2C 设备硬件地址。主模式下，仅在 DMA 模式下，才需填入从设备地址；否则，从设备地址写入 I2C\_DATA 寄存器。 |

# 16.4.2.2 I2C0\_CFG 系统控制寄存器

地址：0x4001\_0404

复位值：0x0

表16-3系统控制寄存器I2C0\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | IE | I2C 中断使能信号。默认值为 0。 1：使能 I2C 中断 0：关闭 I2C 中断 |
| [6] | TC\_IE | I2C 数据传输完成中断使能信号。默认值为 0。 1：使能此中断源 0：屏蔽此中断源 |
| [5] | BUS\_ERR\_IE | I2C 总线错误事件中断使能信号。默认值为 0。 1：使能此中断源 0：屏蔽此中断源 |
| [4] | STOP\_IE | I2CSTOP 事件中断使能信号。默认值为 0。 1：使能此中断源 0：屏蔽此中断源 |
| [3:2] |  | NA |
| [1] | MST\_MODE | I2C 主模式使能信号。默认值为 0。 1：使能主模式 0：关闭主模式 |
| [0] | SLV\_MODE | I2C 从模式使能信号。默认值为 0。 1：使能从模式 0：关闭从模式 |

# 16.4.2.3 I2C0\_SCR 状态控制寄存器

地址：0x4001\_0408

复位值：0x0

表16-4状态控制寄存器I2C0SCR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | STT\_ERR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | RW | RW | RW | RW | RW | RW | RX |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] | 未使用 | |
| [7] | STT\_ERR | 总线错误状态标志位，仅用于主模式下，写 0 清除。 0：无 START/STOP 总线错误 1：有 START/STOP 总线错误 |
| [6] | LOST\_ARB | 总线仲裁丢失状态标志位，仅用于主模式下，发生总线仲裁丢失事件将此位置 1，无中断事件产生，在字节完成中断中需查此位。 总线上任何 START 事件将导致硬件清除此位。 0：无总线仲裁丢失错误发生 1：有总线仲裁丢失错误发生 |
| [5] | STOP\_EVT | STOP 事件状态标志位，主从模式均可使用，写 0 清除。 0：无 STOP 事件 1：有 STOP 事件 |
| [4] | BYTE\_CMPLT | ACK 发生控制位，主从模式均可使用。发送前配置好改位。单字节模式下，发送字节完毕后硬件自动清除；多字节模式下，全部发送完毕后硬件自动清除。 0：字节发送完成，返回 NACK 回应 1：字节发送完成，返回 ACK 回应 |
| [3] | ADDR\_DATA | Address 数据标志位，主从模式均可使用，写 0 清除。 0：发送或接收到的数据非 Address 数据 1：发送或接收到的数据是 Address 数据 |
| [2] | DATA\_DIR | 发送或接收控制位，主从模式均可使用，通过此位决定数据传输方向。总线上 START 事件可清除此位，同时也可写 0 清除此位。 0：接收 1：触发发送 |
| [1] | RX\_ACK | 接收响应标志位，主从模式均可使用，总线上 START 事件可清除此位，同时也可写 0 清除此位。 0：本 I2C 接口发送数据，接收到 ACK 响应 1：本 I2C 接口发送数据，接收到 NACK 响应 |
| [0] | Done | 传输完成状态标志位，主从模式均可使用，总线上 START 事件可清除此位，同时也可写 0 清除此位。 0：传输未完成 1：传输已完成 |

一般，进入中断后，需读取 I2C\_SCR 寄存器，获得当前 I2C 总线状态及当前传输处于什么阶段；然后，对 I2C\_SCR 进行写操作，写入不同的值，软件通知硬件下一步如何处理。

# 16.4.2.4 I2C0\_DATA 数据寄存器

地址：0x4001\_040C

复位值：0x0

# 表16-5数据寄存器I2C0\_DATA

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |
| --- | --- |
|  | DATA |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | DATA | 数据寄存器，主从模式均可使用。对寄存器写入，数据进入 I2C 内部发送部分，无法直接读取到刚写入数据。读出操作，获得 I2C 接收到的数据。主/从接收模式：存放接收的数据。单字节传输，需等待完成中断；多字节传输，数据会被 DMA 搬移到 RAM，仅保留最后一个字节。从发送模式：存放发送的数据。一般先填好数据，然后触发 I2C\_SCR 的 BIT[2]。多字节传输，数据会被 DMA 从 RAM 搬移到 I2C\_DATA，然后由 I2C 接口发送。主发送模式：单字节传输，I2C\_DATA 填入地址；多字节传输，I2C\_ADDR 填入地址，I2C\_DATA 数据由 DMA 传输。 |

# 16.4.2.5 I2C0\_MSCR 主模式寄存器

地址：0x4001\_0410

复位值：0x0

表16-6主模式寄存器I2C0\_MSCR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | BUSY | MST\_CHECK | RESTART | START |  |  |  |
| RW | RW | RW | RW |  |  |  |
| 0 | 0 | 0 | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:4] |  | 未使用 |
| [3] | BUSY | I2C 总线，闲忙状态。1，检测到 START 事件，忙碌；0，检测到 STOP 事件，空闲。 |
| [2] | MST\_CHECK | 主模式争抢总线标志位。争抢到总线，置 1；STOP 事件或者发生总线冲突本模块释放总线，置 0。 |
| [1] | RESTART | 再次触发 START 事件，写 1 有效。发送 START 完毕，硬件清 0。I2C\_CFG[1] 置 1，才能实现写 1 操作。 |
| [0] | START | 触发 START 事件并发送地址数据至总线，写 1 有效。I2C\_CFG[1] 置 1，才能实现写 1 操作。 |

# 16.4.2.6I2C0\_BCRDMA传输控制寄存器

地址：0x4001\_0414

复位值：0x0

表16-7DMA传输控制寄存器I2C0BCR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | BURST\_NACK | I2C 传输，NACK 事件中断使能信号。 1：使能此中断源 0：屏蔽此中断源 |
| [6] | BURST\_ADDR\_CMP | I2C 传输，硬件地址匹配中断使能信号。 1：使能此中断源 0：屏蔽此中断源 |
| [5] | BUSRT\_EN | I2C 多数据传输使能，需要采用 DMA 方式。 1：使能 0：关闭 |
| [4] | SLV\_DMA\_PREF | I2C 多数据传输。从模式执行 DMA 方式发送，触发硬件预取第一个字节。硬件自动清零。 1：使能 0：关闭 |
| [3:0] | BURST\_SIZE | I2C 数据传输长度寄存器，用于多字节传输。 实际传输字节数 = B[3:0] + 1 |

# 17 SPI

# 17.1概述

17.1 概述SPI 接口主要使用在，外部设计采用 SPI 协议的应用场景下。SPI 的工作模式软件可选，默认为 SPI Motorola 模式。SPI 接口支持全双工传输和半双工传输。当接口配置为 Master 模式时，可发送时钟信号供外部 Slave 设备使用。

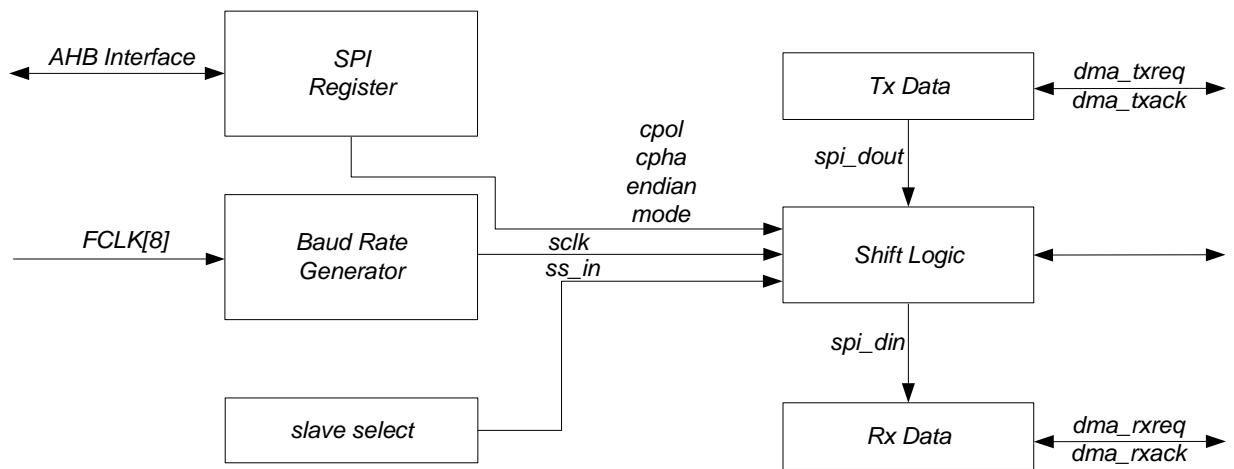
# 17.2主要特性

17.2 主要特性- 支持 Master 和 Slave 操作- 全双工传输，可根据应用情况，使用 3 根或者 4 根信号线- 支持半双工传输，可根据应用情况，使用 2 根信号线- 可编程的时钟极性和相位- 可编程的数据顺序：MSB 或 LSB- 最快传输速度为系统最高时钟频率的 1/8- 片选信号均可选。Master 模式下，片选信号可以软件控制也可以硬件产生；Slave 模式下，片选信号可以恒定有效，也可以来自外界设备- 无本地 FIFO，支持 DMA 操作。包含溢出检测和片选信号异常检测

# 17.3功能描述

# 17.3.1 功能框图

本接口采用同步串行设计，实现CPU同外部设备之间的SPI传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。



接口信号包括，spi\_din，spi dout，sclk\_in，sclk\_out，ss\_in和ss\_out。

spi\_din：接口接收的数据信号。同SPI协议比较，当接口配置为Master模式时，其等效为MISO；当接口配置为Slave模式时，其等效为MOSI。

spi dout：接口发送的数据信号。同SPI协议比较，当接口配置为Master模式时，其等效为MOSI；当接口配置为Slave模式时，其等效为MISO。

sclk\_in：接口接收的时钟信号。此时，接口的工作模式为Slave。非Slave模式下，此信号输入无效。

sclk\_out：接口发送的时钟信号。此时，接口的工作模式为Master，非Master模式下，此信号输出恒定为0。

ss\_in：接口接收的片选信号。此时，接口的工作模式为Slave。非Slave模式下，此信号输入无效。

ss\_out：接口发送的片选信号。此时，接口的工作模式为Master。非Master模式下，此信号输出恒定为1。

# 17.3.2 功能说明

# 17.3.2.1 全双工模式

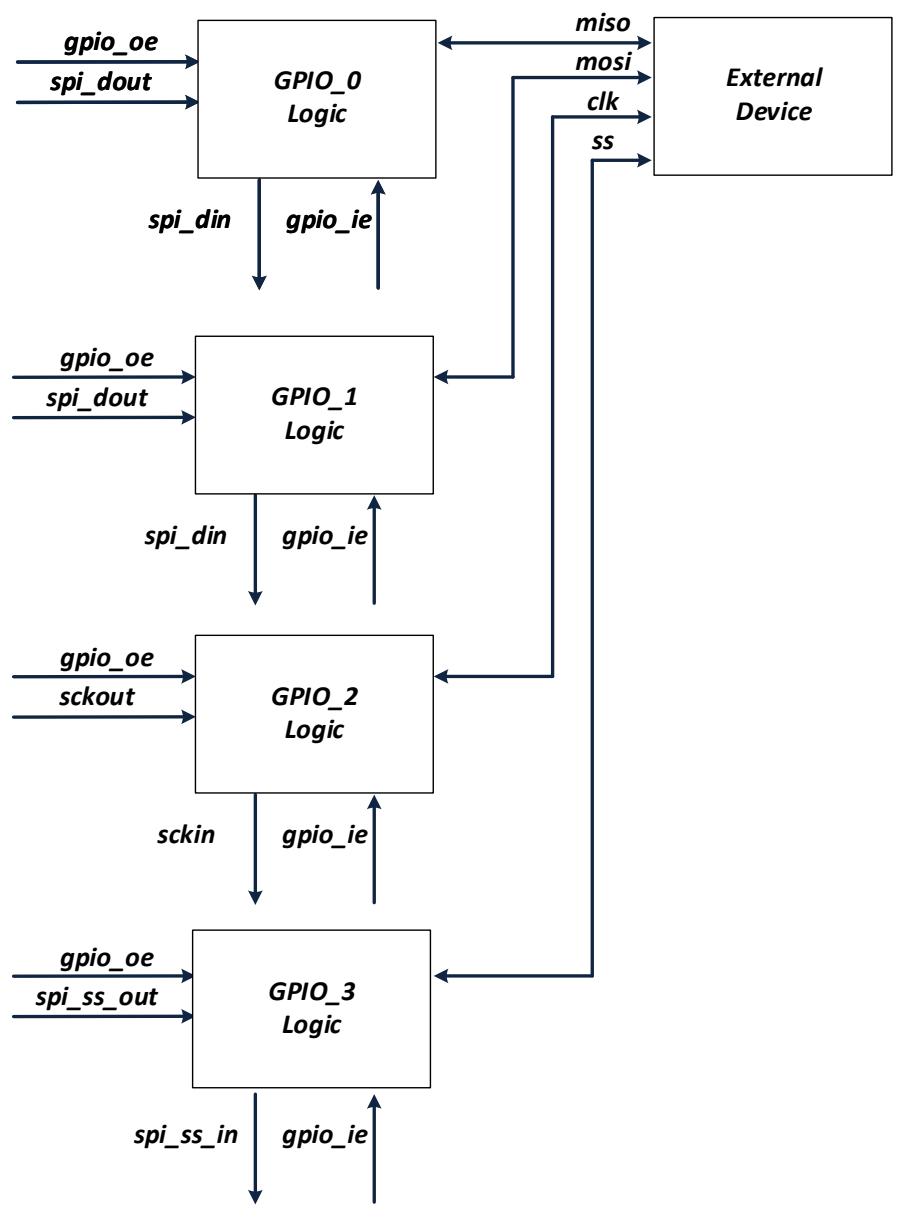
默认情况下，SPI接口配置为全双工模式。此时，数据传输需要两根数据线。数据信号的变化，发生在时钟信号的边沿，即同步于时钟信号。

接口为Master模式时：

spi\_dout为数据输出，接外部Slave设备的MOSIspi\_ss\_out为片选信号，根据应用情况选择是使用该信号还是软件控制其它GPIO实现

接口为Slave模式时：

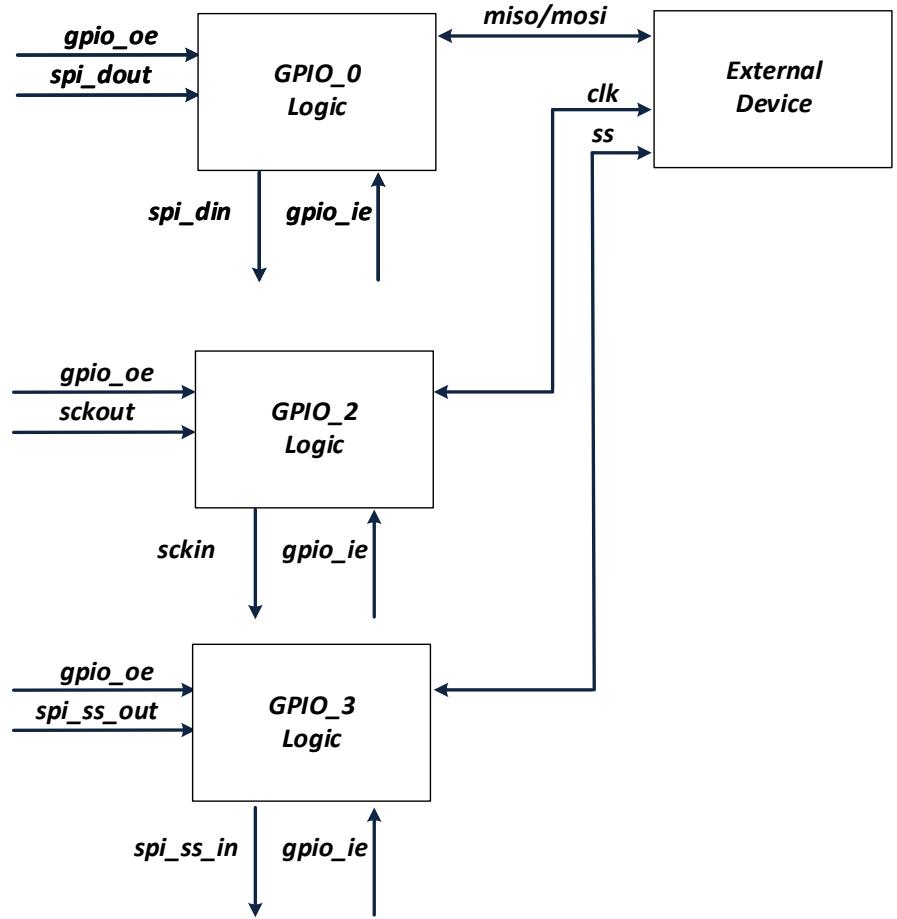
spi\_dout为数据输出，接外部Slave设备的MISOspi\_ss\_in为片选信号，根据应用情况是使用该信号还是片选恒有效

  
图17-2 SPI接口全双工模式互连框图

从上图可知，GPIO若配置为输出，则SPI接口可发送数据；GPIO若配置为输入，则SPI接口可接收数据。亦即SPI\_DI关闭GPIO输入使能且开启GPIO输出使能时，即可作为SPI\_DO使用，SPI\_DO关闭GPIO输出使能且开启GPIO输入使能时，即可作为SPI\_DI使用。

# 17.3.2.2 半双工模式

SPI接口可配置为半双工模式。此时，数据传输只需要一根数据线。数据信号的变化，发生在时钟信号的边沿，即同步于时钟信号。一次传输只能是一个方向的，要不就是发送，要不就是接收。

  
图17-3 SPI接口半双工模式互连框图

注意，上图中若本接口做Master，则clk为本接口的输出信号；若本接口做Slave，则clk为本接口的输入信号。

仅发送

CFG[7:6]配置为2，半双工发送模式有效。此时，本接口只能发送数据。GPIO\_0的oe使能，发送spi\_dout数据到外界；GPIO\_0的ie关闭，spi\_din恒定输入为0。此模式下，支持DMA传输、支持Master/Slave模式下的发送。

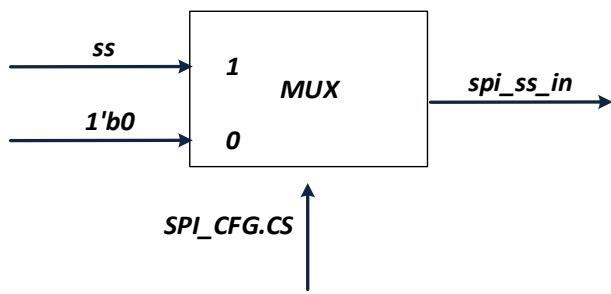
仅接收

CFG[7:6]配置为3，半双工接收模式有效。此时，本接口只能接收数据。GPIO\_0的oe关闭，spi\_dout无法发送数据到外界；GPIO\_0的ie开启，spi\_din接收来自外部的数据。此模式下，支持DMA传输、支持Master/Slave模式下的接收。

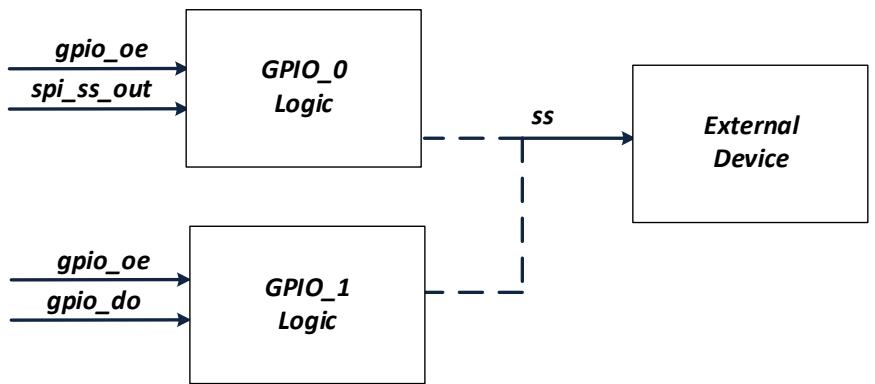
注意，全双工下是两个GPIO用于数据传输，半双工下可从中任意选一个GPIO用于数据传输。

# 17.3.2.3 片选信号

本接口做Slave模式时，片选信号可选，CFG[5]决定片选来源。ss为Master设备发出的选通使能信号，低电平有效。

  
图17-4 SPI 模块 Slave 模式片选信号选择

本接口做 Master 模式时，片选信号亦可选。模块硬件产生了标准的片选信号，实际应用可屏蔽此信号通过软件操作额外的 GPIO 实现。

  
图17-5 SPI 模块 Master 模式片选信号选择

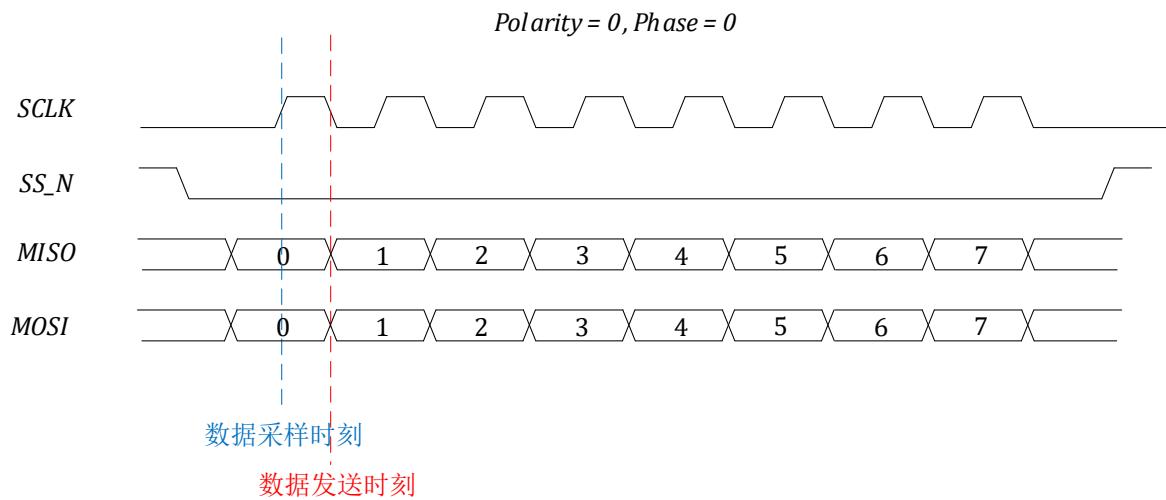
注意，图14- 5 虚线仅表示不确定。若使用 spi\_ss\_out 为 ss 的源头，那么将 GPIO\_0 同外界设备互连；若使用软件操作 GPIO 的方式，那么可将 GPIO\_1 同外界设备互连。

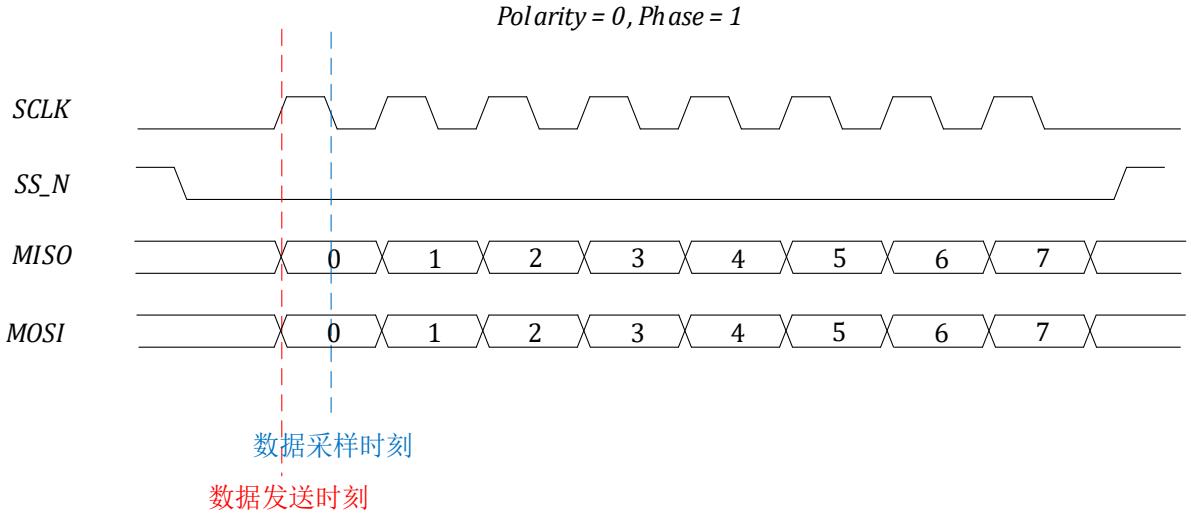
# 17.3.2.4 通讯格式

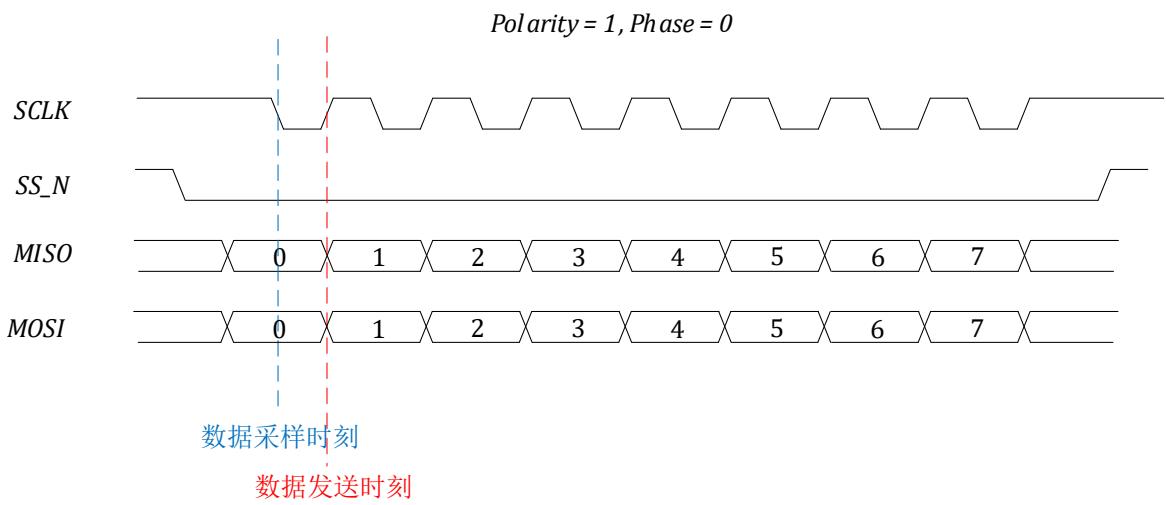
在 SPI 通讯过程中，发送或者接收操作均是基于 SPI 时钟的。通讯格式受到 CFG[3:2] 控制。CFG[3] 为 Phase 控制位，CFG[2] 为 Polarity 控制位。

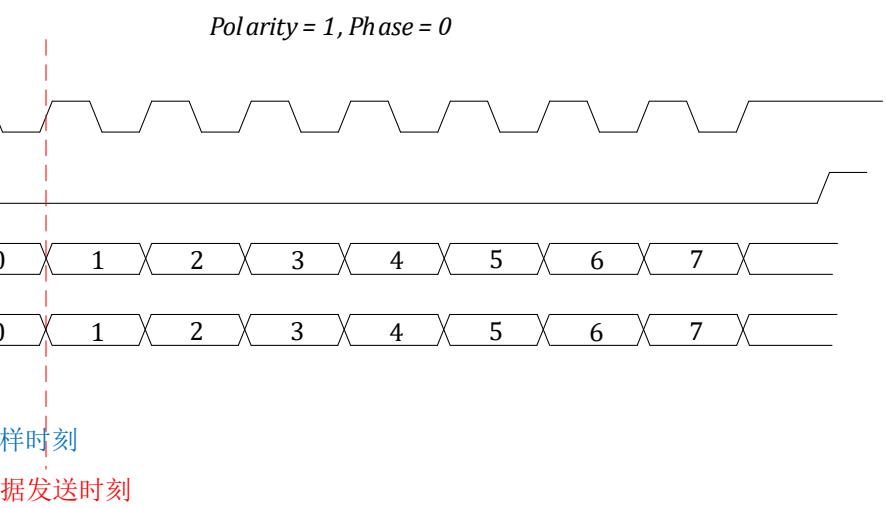
Polarity 控制了 SPI 时钟信号在默认情况下的电平状态。Polarity 为 0 时，默认时钟电平为低电平；Polarity 为 1 时，默认电平为高电平。

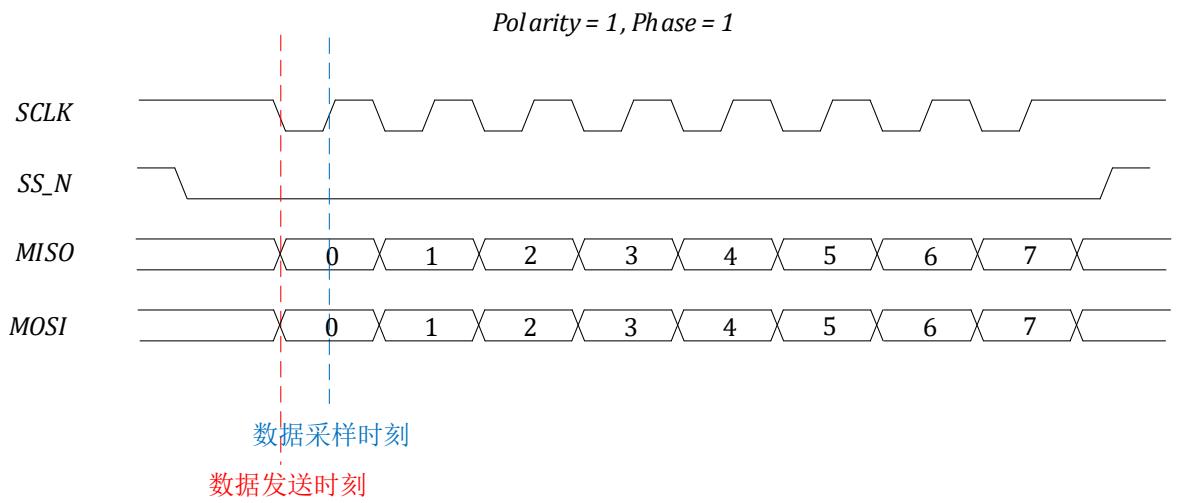
Phase 控制了 SPI 数据的发送/接收时刻。Phase 为 0 时，时钟从默认电平到第一个跳变边沿为采样数据时刻，Phase 为 1 时，时钟从默认电平到第一个跳变边沿为发送数据时刻。



  
图17-7SPI通讯信号极性相位(Polarity ,Phase

  
图17-8SPI通讯信号极性相位(Polarity ,Phase



  
图17-9 SPI通讯信号极性相位(Polarity , Phase )

# 17.3.2.5 数据格式

SPI数据传输格式分成两种：MSB和LSB。数据传输格式受到CFG[1]控制。注意，在数据传输过程中硬件自动实现传输格式的转换，无需软件转换。

# 17.3.2.6 DMA传输

在大容量数据传输应用下，SPI接口支持DMA传输，减轻CPU的负担。一次传输，最大传输量为255字节，最小传输量为1字节。在全双工模式下，接收和发送均可实现DMA传输；在半双工模式下，仅接收或发送实现DMA传输。

在接收到新的数据后，硬件自动产生DMA请求，通过DMA模块将数据搬移到RAM中。在发送新数据前，硬件自动产生DMA请求，通过DMA模块将数据从RAM中搬移到SPI接口。因SPI无FIFO，SPI发送一次DMA请求，DMA只能搬移一个字节数据。若要实现多字节搬移，DMA需配置为多轮搬移，每一轮搬移一个字节的方式。

DMA传输需要配置DMA模块相应寄存器

因SPI接口支持DMA传输，也支持CPU传输。两者区别在于- - DMA传输，发送的数据来自DMA的搬移；CPU传输，发送的数据来自CPU的搬移。

DMA传输，推荐软件配置流程如下：

* 初始化DMA模块，将本次发送的数据来源，接收的数据去向配置好，传输长度配置完毕。- 初始化GPIO模块，将SPI复用的GPIO配置完毕。- 初始化SPI接口，IE/CFG/BAUD/SIZE等寄存器配置完毕。- 触发SPI接口，进入发送/接收状态。触发条件是CPU对TX\_DATA寄存器执行写操作，因最终发送的数据来自DMA，本次CPU写入的数据不会混入SPI发送流程。

# 17.3.2.7 CPU传输

CPU传输，一次只能发送/接收1个字节，每次完成后需要通过中断或者轮询的方式判断传输

是否完成。

SPI接口支持DMA传输，也支持CPU传输。CPU传输，SPI接口无需触发发送状态，只要CPU将数据搬移到SPI接口，就可开始发送数据到外界。

CPU传输，推荐软件配置流程如下：

初始化GPIO模块，将SPI复用的GPIO配置完毕。

初始化SPI接口，IE/CFG/BAUD/SIZE等寄存器配置完毕。注意SIZE只能配置为1。

CPU对TX\_DATA寄存器执行写操作，触发SPI接口进入发送流程，发送的数据来自CPU对TX\_DATA写入值。

# 注意：若需要连续发送，则需要重新配置SIZE和TX\_DATA寄存器。1

# 17.3.2.8外部事件传输

17.3.2.8 外部事件传输在大容量数据传输应用下，SPI接口支持DMA传输。传输过程中，可被打断也可不被打断。所谓打断是指，当前字节传输完成，需等待外部事件才开始下一个字节的传输；不打断是指，当前字节传输完成，直接下一个字节的传输。打断模式需要同其它模块配合使用。例如定时器模块，定时器可触发下一字节的传输。打断模式仅在Master模式有效。IE[3]控制了是否使用打断模式。

# 17.3.2.9中断处理

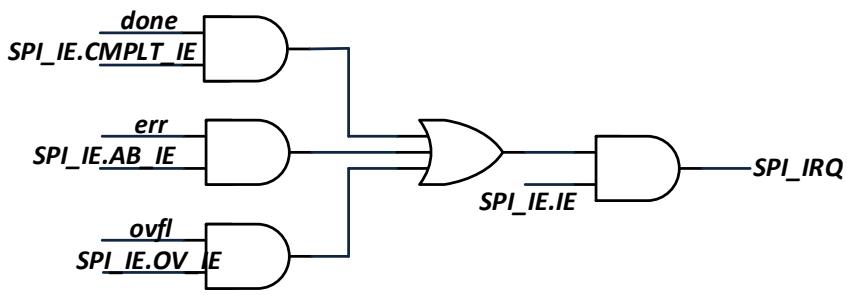
SPI接口包含三种类型的中断事件，分别是：数据传输完成事件，异常事件和溢出事件。

数据完成事件，当前数据传输完成。高电平有效，对IE[0]写1清除。

异常事件，SPI接口为Slave模式，若在传输过程中片选信号受到干扰，被拉高，将产生片选异常事件。高电平有效，对IE[1]写1清除。

溢出事件，数据没有及时通过DMA返回RAM，或者从RAM获得数据，将产生溢出事件。高电平有效，对IE[2]写1清除。

上述事件，默认是不触发SPI中断，可以通过配置IE[6:4]使能事件产生中断。

  
图17-10SPI模块中断选信号产生图

数据传输完毕后，可通过DMA中断或者SPI自身中断判断结束。

发送模式下，DMA先完成搬移操作，SPI后发送完毕。SPI发送完毕，可作为本次传输完成标

识。

接收模式下，SPI接收完毕，触发DMA搬移。DMA搬移完成，可作为本次传输完成标识。

溢出事件。在全双工模式下，发送和接收的DMA均有效，一般情况下不会发送溢出事件；在半双工模式下，只有发送（或接收），此时硬件屏蔽了接收（或发送）的溢出判断。

# 17.3.2.10波特率设置

SPI接口时钟通过对系统时钟分频获得，分频系数来自BAUD[5:0]。分频范围是 ，对应的BAUD[5:0]的值为 。

SPI协议为半拍协议，上升沿发送数据，下降沿采集数据；或者下降沿发送数据，上升沿采用数据。

SPI接口采用同步设计，需要对外部设备的信号进行同步采样，同步时钟为系统时钟。数据和时钟信号（此时为Slave模式）的同步，需要两拍系统时钟。考虑到时钟相位的偏移，此时需要一拍系统时钟的冗余，由此推导出最快的BAUD率为系统时钟的1/8，高电平周期为四拍系统时钟，低电平周期为四拍系统时钟。

# 17.4寄存器

# 17.4.1 地址分配

SPI模块寄存器的基地址是0x4001\_0000，寄存器列表如下：

表17-1SPI模块控制寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| SPI\_CFG | 0x00 | SPI 配置寄存器 |
| SPI\_IE | 0x04 | SPI 中断寄存器 |
| SPI\_DIV | 0x08 | SPI 波特率寄存器 |
| SPI\_TX\_DATA | 0x0C | SPI 发送数据寄存器 |
| SPI\_RX\_DATA | 0x10 | SPI 接收数据寄存器 |
| SPI\_SIZE | 0x14 | SPI 传输数据长度寄存器 |

# 17.4.2 SPI\_CFGSPI控制寄存器

地址：0x4001\_0000

复位值：0x0

表17-2系统控制寄存器SPI\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:6] | DUPLEX | 半双工模式设置 0X: 关闭半双工模式 10: 开启半双工模式，仅发送 11: 开启半双工模式，仅接收 |
| [5] | CS | SPI 从设备下，片选信号来源。默认值为 1。 1: Slave 模式下，片选信号来自 Master 设备 0: Slave 模式下，片选信号恒为有效值–0 |
| [4] | MS | SPI 主从模式选择。默认值为 0。 1: Master 模式 0: Slave 模式 |
| [3] | SAMPLE | SPI 相位选择。默认值为 0。 1: Phase 为 1 0: Phase 为 0 |
| [2] | CLK\_POL | SPI 极性选择。默认值为 0。 1: Polarity 为 1 0: Polarity 为 0 |
| [1] | ENDIAN | SPI 模块传输顺序。默认值为 0。 1: LSB，低位先传输 0: MSB，高位先传输 |
| [0] | EN | SPI 模块使能信号。默认值为 0。 1: 开启 SPI 模块 0: 关闭 SPI 模块 |

SPI\_CFG[3:2]对应的通讯波形极性和相位可参考17.3.2.4。

# 17.4.3 SPI\_IE SPI中断寄存器

地址：0x4001\_0004

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | IE | CMPI\_EIE | AR\_EIE | OV\_EIE | TANX\_TANX | CMPI\_EIE | AR\_EIE | OV\_EIE |  |
| RW | RW | RW | RW | RW | RW | RW | RW |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | IE | SPI 中断使能开关。默认值为 0。 1：使能 SPI 中断 |

|  |  |  |
| --- | --- | --- |
|  |  | 0：关闭 SPI 中断 |
| [6] | CMPLT\_IE | SPI 传输，完成事件中断使能信号。 1：使能此中断源 0：屏蔽此中断源 |
| [5] | AB\_IE | SPI 传输，异常事件中断使能信号。 1：使能此中断源 0：屏蔽此中断源 |
| [4] | OV\_IE | SPI 传输，溢出事件中断使能信号。默认值为 0。 1：使能此中断源 0：屏蔽此中断源 |
| [3] | TRANS\_TRIG | 传输触发选择。 1：外部触发 0：内部自动执行。仅主模式有效 |
| [2] | CMPLT\_IF | SPI 传输，完成事件。高电平有效，写 1 清除。 |
| [1] | AB\_IF | SPI 传输，异常事件。Slave 模式下，传输未完成，发生片选信号无效事件。高电平有效，写 1 清除。 |
| [0] | OV\_IF | SPI 传输，溢出事件。包含两种情况： 1. 前次接收的旧数据没有被取得走，本次接收的新数据已经到达。 2. 本次发送的数据已发送完毕，新数据没有准备好。高电平有效，写 1 清除。 |

# 17.4.4 SPI\_DIV SPI 设特率寄存器

地址：0x4001\_0008

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | TRANS\_MODE | | BAUD | | | | |  |
| RW | | RW | | | | |  |
| 0 | | 0 | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | TRANS\_MODE | SPI数据搬移方式。默认为0，DMA方式。 0：SPI接口支持DMA搬移数据到SPI接口，完成发送和接收。 1：SPI接口支持CPU搬移数据到SPI接口，完成发送和接收。 |
| [6] |  | 未使用 |
| [5:0] | BAUD | SPI传输波特率配置，SPI实际传输速度计算公式为： SPI传输速度=系统时钟/(2\*(BAUD+1)) 切记，BAUD的配置值不能小于3。 |

# 17.4.5 SPI\_TX\_DATA SPI数据发送寄存器

地址：0x4001\_000C

复位值：0x0

表17-5SPI\_TX\_DATA数据发送寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
|  | | | | | | | | | | | | | | | |  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | TX\_DATA | SPI数据发送寄存器。CPU方式，寄存器写入要发送的数据；DMA方式，对寄存器执行写入操作，仅仅是触发DMA搬移，要发送的数据来自DMA，非本寄存器。 |

# 17.4.6 SPI\_RX\_DATA SPI数据接收寄存器

地址：0x4001\_0010

复位值：0x0

表17-6SPI\_RX\_DATA数据接收寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  | | | | | | |  | | | | | | | RX\_DATA | | | | | | | |
| RW | | | | | | | |  |  |  |  |  |  |  |
| 0 | | | | | | | |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | RX\_DATA | SPI 数据接收寄存器 |

# 17.4.7 SPI\_SIZE SPI数据传输长度寄存器

地址：0x4001\_0014

复位值：0x0

表17-7SPI\_SIZE数据传输长度寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
| RW |

|  |  |
| --- | --- |
|  | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | SIZE | SPI 数据传输长度寄存器。CPU 方式，只能写入 1，一次传输一个字节的数据，传输完成后 SIZE 变成 0；DMA 方式，表示本次传输的数据总量，硬件每传输完成一个字节，SIZE 自减 1，传输完成后 SIZE 变成 0。 |

# 18 CMP

# 18.1概述

比较器信号处理模块（以下简称CMP模块，为便于区分，后续图示中模拟比较器使用Comparator表示，数字CMP模块使用CMP表示），用于处理两个模拟轨到轨比较器产生的输出信号，由一系列使能、极性控制、滤波等数字电路组成。

比较器可用于以下功能：

1. 反电势过零点检测2. 硬件过流检测3. 作为MCPWM的fail信号来源

比较器的主要特性如下：

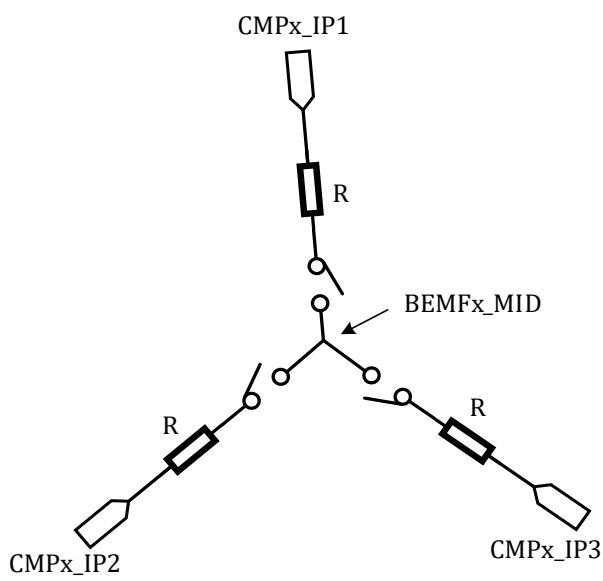
1. 每个比较器都具备多种的信号来源可供选择：

多路GPIO口输入信号 运放输出信号 运放正端输出信号1.2VBANDGAP基准源DAC输出信号

1. 比较速度可编程，迟滞电压可编程
2. 输出信号可滤波，滤波深度可选
3. 可产生CMP中断

需说明的是，CMP负输入端的BEMFx\_MID信号，是对正输入端信号CMPx\_IP1/ CMPx\_IP2/ CMPx\_IP3信号的平均，具体连接方式见图18- 1，图中电阻 欧，图中的开关只有在比较器负输入端信号选择为BEMFx\_MID之后才会导通，否则开关都处于断开状态。

BEMFx\_MID主要用于BLDC方波模式控制时，虚拟电机相线中心点电压，用于反电势过零点检测。三个相线分压后，分别接CMPx\_IP1、CMPx\_IP2、CMPx\_IP3，比较器负端选择BEMFx\_MID，比较器正端的多路选择器以分时复用的方式分别选择CMPx\_IP1、CMPx\_IP2、CMPx\_IP3，就可以比较出反电势过零点。

  
图18-1BEMFx\_MID信号

此模块的信号处理时钟由系统主时钟分频得到。模拟比较器未经滤波的原始输出值，可以通过读取寄存器SYSAFECMP值获得，同时也可以通过配置GPIO的第二功能通过P0.14和P2.3送出，具体GPIO第二功能配置及引进位置，请参考器件datasheet。

推荐配置流程：

1. 配置CMP的模拟开关芯片上电的默认状态下，比较器模块是关闭的，通过配置寄存器SYSAFE\_REG5.CMPxPDN可以打开比较器 ，代表CMP0/CMP1两个比较器)。开启比较器之前，需要先开启BGP模块。
2. 配置CMP的数字时钟开关、时钟滤波和信号输入开关

通过配置寄存器CMPTCLKCLKEN可以打开CMP的数字时钟开关，1表示打开，0表示关闭；通过配置寄存器CMPTCLK\_FILCLKDIV16[7:4]可以配置滤波时钟，数值设置范围是 ，0表示1分频滤波，15表示16分频滤波；通过配置寄存器CMPCFG.CMPxINEN可以打开信号输入开关，1表示打开，0表示关闭。

1. 配置CMP的比较速度和迟滞电压

比较器的比较延时可通过配置寄存器SYSAFE\_REG1. ITCM[1:0]设置为 。迟滞电压可通过配置寄存器SYSAFE\_REG3.CMP\_HYS设置为 。

1. 选择CMP的正负端信号来源

CMP的正端信号有8种信号来源可选择，可以通过配置寄存器SYSAFE\_REG3. CMPxSELP[2:0]进行设置，负端有4种信号来源可选择，可以通过配置寄存器SYSAFE\_REG3. CMPxSELN[1:0]进行设置。

1. 配置CMP的中断

通过配置寄存器CMPIE.CMPxIE可以打开CMP中断，1表示打开，0表示关闭。

# 18.2 寄存器

# 18.2.1 地址分配

CMP模块所涉及的寄存器包括两部分，一是18.1中提到的模拟配置寄存器，二是CMP\_IE等CMP模块寄存器。

CMP模块寄存器的基地址是0x4001\_0C00，寄存器列表如下：

表18-1比较器模块寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移地址 | 说明 |
| CMP\_IE | 0x00 | 比较器中断使能寄存器 |
| CMP\_IF | 0x04 | 比较器中断标志寄存器 |
| CMP\_TCLK | 0x08 | 比较器分频时钟控制寄存器 |
| CMP\_CFG | 0x0C | 比较器控制寄存器 |
| CMP\_BLCWIN | 0x10 | 比较器开窗控制寄存器 |

# 18.2.2 CMP模块寄存器描述

18.2.2.1 CMP\_IE

地址：0x4001\_0C00

复位值：0x0

表18-2比较器中断使能寄存器CMP\_IE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | CMP1\_IE | CMP0\_IE |  |
| RW | RW |  |
| 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:2] |  | 未使用 |
| [1] | CMP1\_IE | 比较器1中断使能，高有效 |
| [0] | CMP0\_IE | 比较器0中断使能，高有效 |

# 18.2.2.2 CMP\_IF

地址：0x4001\_0C04

复位值：0x0

表18-3比较器中断标志寄存器CMP\_IF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  | CMP1\_IF | CMP0\_IF |
| RW1C | RW1C |
| 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:2] |  | 未使用 |
| [1] | CMP1\_IF | 比较器 1 中断标志，高有效，写 1 清零 |
| [0] | CMP0\_IF | 比较器 0 中断标志，高有效，写 1 清零 |

# 18.2.2.3 CMP\_TCLK

地址：0x4001\_0C08

复位值：0x0

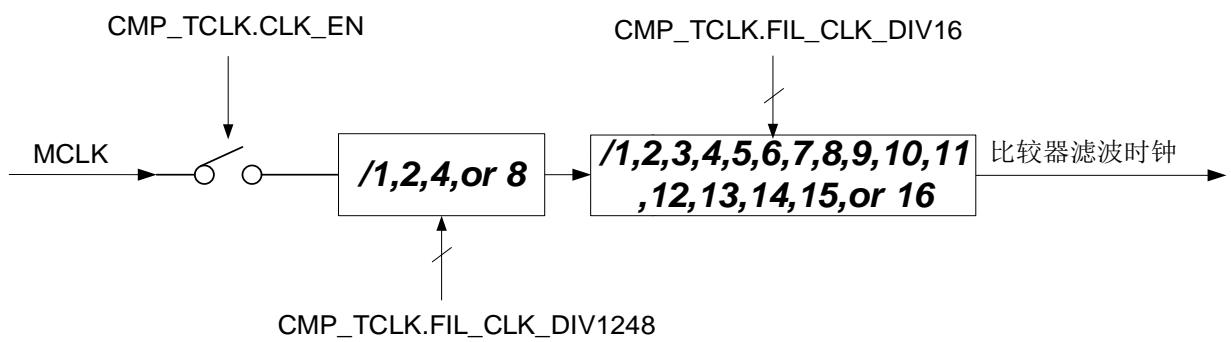
表18-4比较器分频时钟控制寄存器CMP\_TCLK

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:4] | FIL\_CLK\_DIV16 | 比较器滤波时钟分频，基于MCLK进行1~16分频，影响进入比较器中断的时间 |
| [3] | CLK\_EN | 时钟使能，高有效 |
| [2] |  | 未使用 |
| [1:0] | FIL\_CLK\_DIV1248 | 比较器滤波时钟分频，2'b00:1分频，2'b01:2分频，2'b10:4分频，2'b11:8分频 |

CMP滤波时钟频率

Freq(CMP\_Filter) = Freq(MCLK)/2CMP\_TCLK\_FIL\_CLK\_DIV1248 / (CMP\_TCLK\_FIL\_CLK\_DIV16+1)，其中MCLK为系统的主时钟，通常为96MHz全速时钟。需要注意的是，产生CMP滤波时钟需要使能CMP\_TCLK\_CLK\_EN位。

  
图18-2比较器滤波时钟产生

CMP模块使用此滤波时钟对模拟比较器的输出信号进行16时钟周期长的滤波，即只有信号稳定时间超过16个滤波时钟周期才能通过滤波器，CMP模块输出的滤波后的信号才会发生变化，如果输入信号稳定时间不足16个滤波时钟周期即发生变化，则CMP模块输出的滤波后的信号维持原值不变。即滤波宽度 滤波时钟周期

# 18.2.2.4CMP\_CFG

地址：0x4001\_0C0C

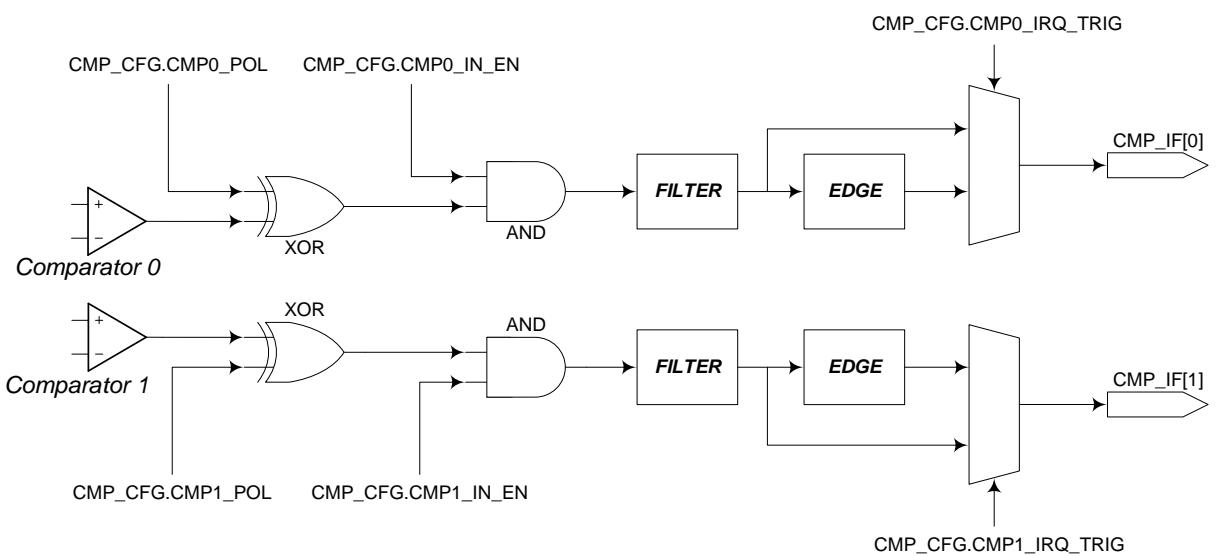
复位值：0x0

表18-5比较器控制寄存器CMP\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
|  |  |  |  |  | CMPLW\_PWM\_POL |  | CMPL\_IRQ\_TRIG |  | CMPL\_IN\_EN |  | CMPL\_POL |  | CMPL\_IRQ\_TRIG |  | CMPL\_IN\_EN |  | CMPL\_POL |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | CMP1\_W\_PWM\_POL | 比较器1开窗PWM信号极性选择，在CMP\_BLCWIN使能情况下使用 |
| [6] | CMP1\_IRQ\_TRIG | 比较器1中断触发类型，0:电平触发，1:边沿触发 |
| [5] | CMP1\_IN\_EN | 比较器1信号输入使能 |
| [4] | CMP1\_POL | 比较器1极性选择，0:高电平有效；1:低电平有效 |
| [3] | CMP0\_W\_PWM\_POL | 比较器0开窗PWM信号极性选择，在CMP\_BLCWIN使能情况下使用 |
| [2] | CMP0\_IRQ\_TRIG | 比较器0中断触发类型，0:电平触发，1:边沿触发 |
| [1] | CMP0\_IN\_EN | 比较器0信号输入使能 |
| [0] | CMP0\_POL | 比较器0极性选择，0:高电平有效；1:低电平有效 |

比较器的极性及使能控制如图 18- 3 所示。

  
图18-3比较器控制及中断产生逻辑

比较器模块与MCPWM模块可以联合动作，其中MCPWM模块的P管控制信号可以作为比较器开窗的控制信号。但比较器自身的中断信号产生于开窗控制无关，仅仅受CMP\_CFG寄存器影响。

MCPWM的fail信号可以来自GPIO，也可以来自比较器模块，使用MCPWM\_FAIL寄存器进行控制。如果MCPWM的fail信号来自比较器，则是经过比较器模块内部的开窗控制的。fail信号进入MCPWM后也会进行极性使能以及滤波等处理，与比较器模块类似但完全独立，由MCPWM内部的寄存器进行控制。MCPWM内部与fail相关的错误中断信号产生收到MCPWM内部有关极性使能滤波控制寄存器的影响。具体请参考MCPWM章节。

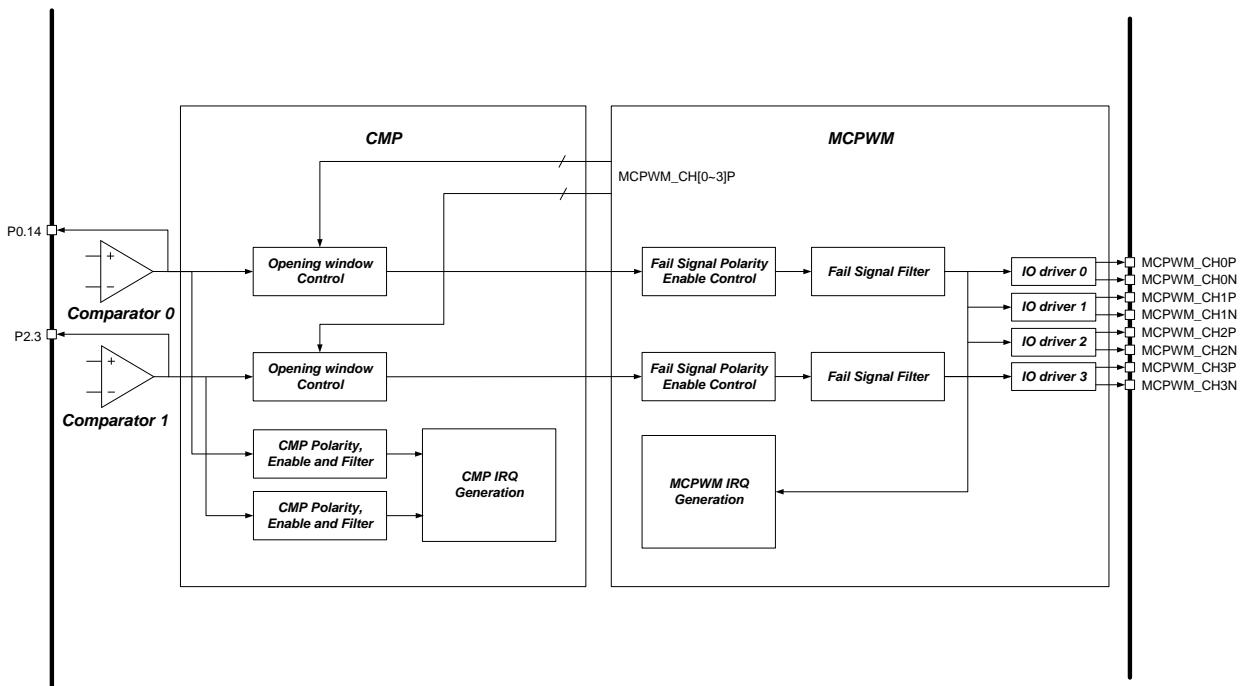
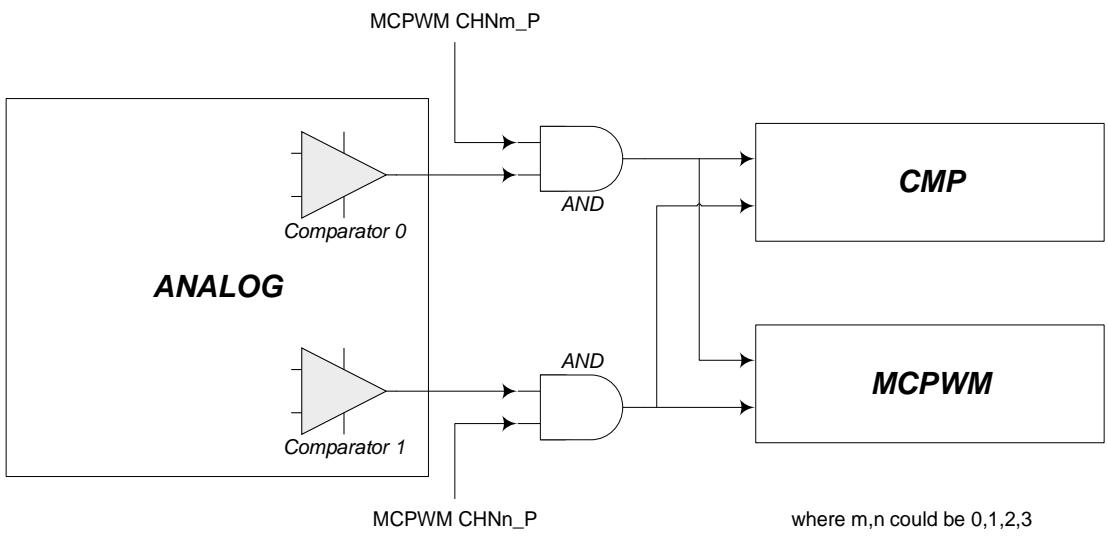
  
图18-4CMP与MCPWM的联动

图18- 4 CMP与MCPWM的联动对于比较器的开窗功能，若CMP\_CFG.CMP0\_PWM\_POL=1，则在对应MCPWM CHNx\_P信号为1时，比较器0可以产生比较信号输出，其他时刻比较信号为0；反之，若CMP\_CFG.CMP0\_PWM\_POL=0，则在对应MCPWM CHNx\_P信号为0时，比较器0可以产生比较信号输出，其他时刻比较信号为0。

比较器1的开窗控制信号极性由CMP\_CFG.CMP1\_PWM\_POL位进行控制，逻辑相同。

注意：CMP\_CFG.CMP0\_W\_PWM\_POL和CMP\_CFG.CMP1\_W\_PWM\_POL同时会影响送入MCPWM模块作为FAIL信号的比较器信号，如图18- 5所示。来自比较器的MCPWMFAIL信号为模拟比较器输出的原始信号，未经过比较器数字接口模块的滤波处理，但是可以被MCPWM的通道信号进行开窗控制，开窗控制设置见比较器数字接口模块。FAIL信号进入MCPWM模块后，可以通过设置MCPWM\_TCLK进行滤波。

  
图18-5比较器开窗功能图示

# 18.2.2.5CMP\_BLCWIN

地址：0x4001\_0C10

复位值：0x0

表18-6比较器开窗控制寄存器CMP\_BLCWIN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 保留 |
| [7] | CMP1\_CHN3P\_WIN\_EN | 使用 MCPWM 模块 CHN3\_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能 |
| [6] | CMP1\_CHN2P\_WIN\_EN | 使用 MCPWM 模块 CHN2\_P 通道输出的 P 管开关控制信号作 |

|  |  |  |
| --- | --- | --- |
|  |  | 为比较器 1 开窗使能 |
| [5] | CMP1\_CHN1P\_WIN\_EN | 使用 MCPWM 模块 CHN1\_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能 |
| [4] | CMP1\_CHN0P\_WIN\_EN | 使用 MCPWM 模块 CHN0\_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能 |
| [3] | CMP0\_CHN3P\_WIN\_EN | 使用 MCPWM 模块 CHN3\_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能 |
| [2] | CMP0\_CHN2P\_WIN\_EN | 使用 MCPWM 模块 CHN2\_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能 |
| [1] | CMP0\_CHN1P\_WIN\_EN | 使用 MCPWM 模块 CHN1\_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能 |
| [0] | CMP0\_CHN0P\_WIN\_EN | 使用 MCPWM 模块 CHN0\_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能 |

通常 CMP\_BLCWIN[3:0]或 CMP\_BLCWIN[7:4]中有 1bit 为 1，表明使用对应的 CHNx\_P 对比较器 0/1 的信号产生进行控制。如果 CMP\_BLCWIN[3:0]或 CMP\_BLCWIN[7:4]为 4’b0000，则表示比较器 0/1 比较信号的产生与 PWM 信号无关。

# 19 CAN

# 19.1概述

19.1 概述CAN 总线接口连接微控制器和串行 CAN 总线。本 CAN 模块根据特定设备的需要，可以使用 DMA 以减轻 CPU 的负担。使用 CAN 通信时需要用外部晶振作为参考时钟。

# 19.1主要特性

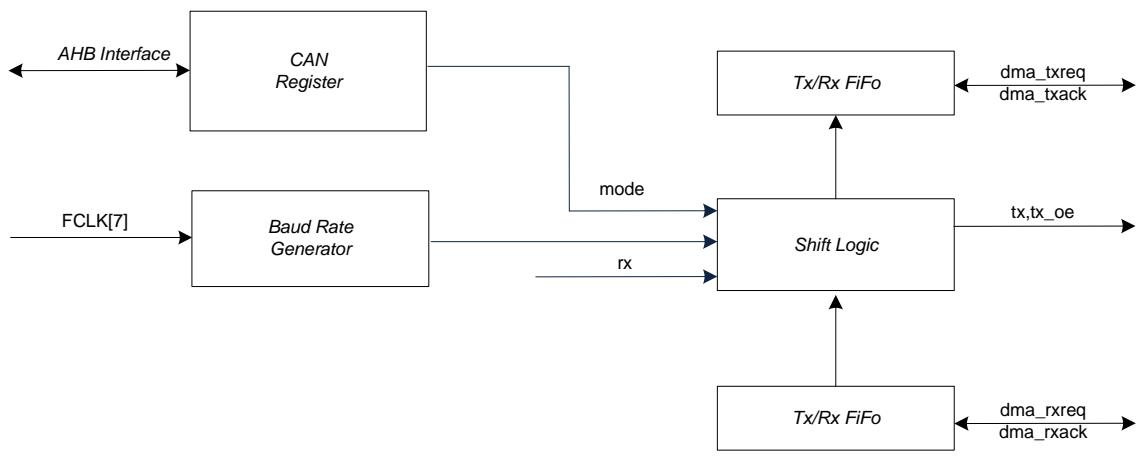
支持BOSCH2.0A和2.0B协议。2.0A等效CAN1.2，包含了11位ID格式；2.0B包含了11位ID和29位ID。

支持SJA1000大部分功能（部分不支持点在文档后续有说明）。- 有两种模式：工作模式和复位模式。- 支持监听和自测试。监听模式下，仅接收外界信号但不会返回任何信号。- 支持DMA功能。

# 19.2功能描述

# 19.2.1 功能框图

本接口采用同步串行设计，实现CPU同外部设备之间的CAN传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。

  
图19-1CAN模块顶层功能框图

CAN接口同外界通讯只有tx和rx两根信号线。tx发送有效数据时候，tx\_oe有效。tx不发送有效数据的时候，tx\_oe无效。

rx：数据信号。接收来自外界的CAN数据

tx：数据信号。发送CAN数据到CAN总线。

tx\_oe：数据使能信号。当tx输出时，tx\_oe有效；当tx无数据输出时，tx\_oe无效。

# 19.2.2 功能说明

CAN模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。可以开启或禁止中断。接口通过数据输出引脚(TX)和数据输入引脚(RX)连接到CAN总线的PHY芯片上。

# 19.2.2.1工作模式

CAN模块主要包含两个工作模式：正常工作模式和复位模式。

复位模式，时序参数、ID配置、错误统计值等均在此模式下设定好。CAN\_MOD.0为1，是复位模式。硬件复位结束后，CAN模块处于复位模式下。

正常工作模式，CAN\_MOD.0为0。此时，可以正常响应CAN总线请求。

在上述两个模式下，扩展了监听模式（ListenOnly）和自测试模式（SelfTest)。前者，类似一个搜集器，只接收CAN总线上的数据，不发送任何数据。后者，是内部自测试，发送的数据同时被自己接收，检测内部功能是否正确。

# 19.2.2.2DMA传输

在大容量数据传输应用下，CAN接口支持DMA传输，减轻CPU的负担。一次传输，SFF为11个字节，EFF为13个字节。

在接收到新的数据后，即CAN\_IR.RFIFO\_N\_EMPTY\_IF硬件自动产生DMA请求，通过DMA模块将数据搬移到RAM中。在发送新数据前，硬件自动产生DMA请求，通过DMA模块将数据从RAM中搬移到CAN接口。

DMA传输需要配置DMA模块相应寄存器

因CAN接口支持DMA传输，也支持CPU传输。两者区别在于- - DMA传输，发送的数据来自DMA的搬移；CPU传输，发送的数据来自CPU的搬移。

DMA传输，推荐软件配置流程如下：

初始化DMA模块，将本次发送的数据来源，接收的数据去向配置好，传输长度配置完毕。

初始化GPIO模块，将CAN复用的GPIO配置完毕。

初始化CAN接口，控制寄存器配置完毕。

触发CAN接口，进入发送状态。

本CAN模块设计的DMA不同于其它模块的DMA搬移操作，需要CPU介入部分搬移操作。假定当前配置CAN模块发送IN帧数据，那么第一帧数据需要CPU搬移到CAN模块寄存器中，后续帧（N- 1）的数据可由DMA实现搬移。

# 19.2.2.3CPU传输

CPU传输，根据SFF还是EFF，搬移11个字节或者13个字节；完成后需要通过中断或者轮询的方式判断传输是否完成。

CPU传输，推荐软件配置流程如下：

* 初始化 GPIO 模块，将 CAN 复用的 GPIO 配置完毕。- 初始化 CAN 接口，控制寄存器配置完毕。- CPU 触发 CAN 接口进入发送流程，发送的数据来自 CPU 对 CAN\_TXDATA 写入值。

19.2.2.4 中断处理

CAN 接口包含中断事件比较多，在 CAN\_IR 和 CAN\_EIR 寄存器描述中有相应说明。根据实际使用情况，开启对应的中断事件使能开关。

CAN控制器提供了以下七种中断：

·接收中断·发送中断·错误报警中断·数据溢出中断·被动错误中断·仲裁丢失中断·总线错误中断

只要在 CAN\_IR 一个或多个中断位为 1，CAN 控制器中的中断信号即为有效，当 CAN\_IR 中的所有位都被清除时，CAN 控制器中的中断信号则失效。寄存器 CAN\_IR 被读取后，其中的大多数中断位将自动清除。但是，接收中断不包括在内，直到通过 CAN\_CMR 的 bit2 写 1，读取所有接收报文后，接收中断位才能被清除。

# 19.2.2.4.1 接收中断

当 CAN 接收 FIFO 中有待读取帧时（CAN\_RMC > 0），都会触发接收中断。CAN\_RMC 中记录的帧数量包括接收 FIFO 中的有效帧和溢出帧。直到通过给 CAN\_CMR 的 bit2 写 1，清除所有挂起接收帧后，接收中断才会失效。

# 19.2.2.4.2 发送中断

每当发送缓冲器空闲，将其他帧加载到发送缓冲器中等待发送时，都会触发发送中断。以下情况下，发送缓冲器将变为空闲，同时发送中断将失效：

* 帧发送已成功完成（如，应答未发现错误）。任何发送失败将自动重发。- 单次发送已完成（CAN\_SR bit3 指示发送成功与否）。- CAN\_CMR 的 bit1 位，写 1，终止帧发送。

# 19.2.2.4.3 错误报警中断

每当寄存器 CAN\_IR 中 ON\_BUS 或 ERR\_OV 的位值改变时（如，从 0 变为 1 或反之），都会触发

此中断。根据触发时 ON\_BUS 或 ERR\_OV 的值分成以下几种情况：

* 如果 ERR\_OV = 0 且 ON\_BUS = 0:- 如果 CAN 控制器处于主动错误状态，则表示接收错误计数和发送错误计数的值都返回到了 CAN\_EWLR 所设的阈值之下。- 如果 CAN 控制器此前正处于总线恢复状态，则表示此时总线恢复已成功完成。- 如果 ERR\_OV = 1 且 ON\_BUS = 0: 则表示接收错误计数或发送错误计数的值超过了 CAN\_EWLR 所设的阈值。- 如果 ERR\_OV = 1 且 ON\_BUS = 1: 表示 CAN 控制器已进入 BUS\_OFF 状态（因发送错误计数 >= 256）。- 如果 ERR\_OV = 0 且 ON\_BUS = 1: 表示 BUS\_OFF 恢复期间，CAN 控制器的发送错误计数的数值已低于 CAN\_EWLR 所设的阈值。

# 19.2.2.4.4 数据溢出中断

接收 FIFO 满，若还收到有效帧，将触发数据溢出中断。

# 19.2.2.4.5 仲裁丢失中断

每当 CAN 控制器尝试发送帧且丢失仲裁时，都会触发此中断。CAN 控制器丢失仲裁的 bit 位置将自动记录在仲裁丢失捕捉寄存器 CAN\_ALC 中。仲裁丢失捕捉寄存器被清除（通过 CPU 读取该寄存器）之前，将不会再记录新发生的仲裁失败时的 bit 位置。

# 19.2.2.4.6 总线错误中断

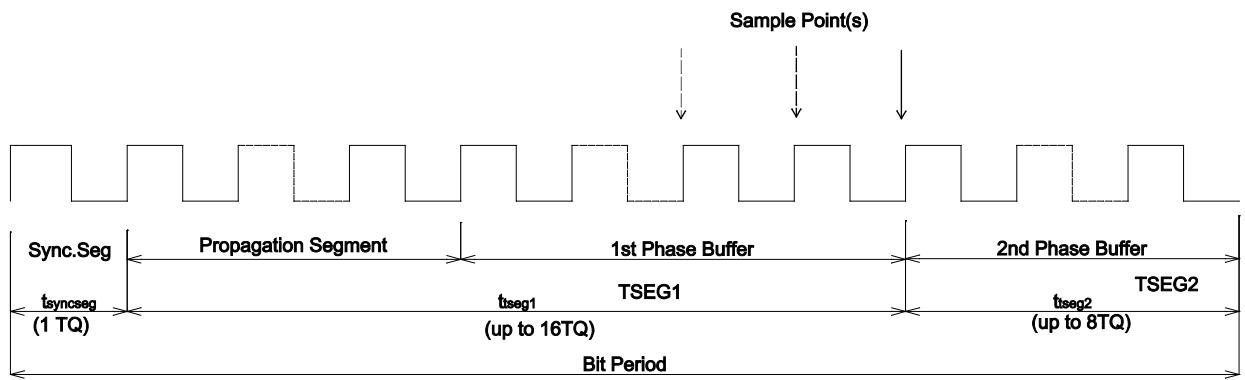
每当 CAN 控制器在 CAN 总线上检测到错误时，都会触发此中断。发生总线错误时，总线错误的类型和发生错误时的 bit 位置都将自动记录在错误捕捉寄存器 CAN\_ECC 中。错误捕捉寄存器被清除（通过 CPU 的读取）之前，将不会再记录新的总线错误信息。

# 19.2.2.5 通讯采样设置

CAN 的波特率设定，主要依靠 CAN\_BTR0 和 CAN\_BTR1 两个寄存器配合完成。CAN\_BTR0 主要是配置 TQ 参数（TQ 的计算见 BTR0 寄存器说明），CAN\_BTR1 主要处理 1- bit 数据的采样点、采样次数以及宽度信息。

CAN\_BTR0 设置传输基本时间单元参数 TQ: TQ = 2 \* Tcl \* (CAN\_BTR0. BAUDRATE + 1) LKS00x 时钟最快为 96M，对应的 Tcl 为 10.4ns，TQ 最大值为 1.3312us。

CAN\_BTR1 设置波特率：同时，可调节 BIT 信息中各个部分的宽度（TSEG1、TSEG2 和 Sync.Seg），找到理想采样点。

  
图19-2 CAN模块Bit周期介绍图

SEG1段时间计算公式：

SEG2段时间计算公式：

波特率计算公式为：CanBaudrate

CAN\_BTR0. SJW是SJW位，其为容差范围的配置寄存器。在一定波特率下通讯的各个设备，允许多大的通讯时间误差存在。

总线下界容差 总线波特率 总线上限容差

容差计算公式为：

CAN\_BTR1. SAM是SAM位，其位采样次数配置寄存器。0：一次；1：三次。根据实际使用情况配置即可，此位不参与波特率计算。

基于LKS08x芯片，常规波特率索引值如下：

|  |  |  |
| --- | --- | --- |
| Can 波特率 | BTR0 | BTR1 |
| 1Mbps | 0x05 | 0x14 |
| 800Kbps | 0x05 | 0x16 |
| 666Kbps | 0x85 | 0xB6 |
| 500Kbps | 0x05 | 0x1C |
| 400Kbps | 0x05 | 0xFA |
| 250Kbps | 0x0B | 0x1C |
| 200Kbps | 0x0B | 0xFA |
| 125Kbps | 0x17 | 0x1C |
| 100Kbps | 0x1D | 0x1C |
| 83.33Kbps | 0x17 | 0x6F |
| 80Kbps | 0x97 | 0xFF |
| 66.66Kbps | 0x1D | 0x6F |
| 50Kbps | 0x3B | 0x1C |
| 40Kbps | 0xAF | 0xFF |

理论上，可以通过配置寄存器实现1- bit数据的时间宽度在3TQ到25TQ之间。实际应用中，

如果遵循BOSCH标准，1- bit数据的时间宽度在8TQ到25TQ之间。若不遵循BOSCH标准，实际可达的范围在4TQ到25TQ之间。

# 19.2.2.6 ID滤波

CAN总线，可以挂载很多设备。通过ID号，不同设备可以知道当前总线上发送的帧是否需要被自己接收，或者发送出去的帧，是否有响应。

接收滤波器允许CAN控制器根据帧ID过滤接收帧（有时可以过滤帧的第一个数据字节和帧类型)。只有通过过滤的帧才能存储到接收FIFO中。接收滤波器的使用可以一定程度地减轻CAN控制器的运行负荷（如，可减少使用接收FIFO和发生接收中断的次数)，因为CAN控制器将只需要操作一小部分过滤后的帧。只有当CAN控制器处于复位模式时，才可以访问接收滤波器的配置寄存器。

CAN帧的ID号码，有两种长度- - 11位和29位。前者对应的SFF（standard frame format），后者对应EFF（extended frame format）。通过CAN\_ACR和CAN\_AMR判断当前CAN模块可接收的ID范围。CAN\_ACR列出了一个特定的ID，CAN\_AMR为MASK寄存器，标识CAN\_ACR中对应位数据同接收到的ID对应位数据需完全匹配，哪些位可以不用。极端情况就是CAN\_ACR上每一位都要匹配，或者每一位都不要匹配。CAN\_AMR对应位是0，表示接收到的ID对应位需要同CAN\_ACR对应位匹配；CAN\_AMR对应位是1，表示接收到的ID对应位不需要同CAN\_ACR对应位匹配。

CAN\_MOD.3决定了CAN\_ACR包含两个滤波ID还是一个滤波ID。为1的话，CAN\_ACR包含一个长的滤波ID；为0的话，CAN\_ACR包含了两个短的滤波ID。两个滤波ID情况下，只要接收帧的ID同其中一个匹配，就会被CAN接收。

SFF，单滤波ID数据格式

滤波ID设置

|  |  |  |  |
| --- | --- | --- | --- |
| CAN\_TXRX1 | CAN\_TXRX2 | CAN\_TXRX3 | CAN\_TXRX4 |
| ID.28..ID21 | ID.20..ID.18 RTR XXXX | Data Byte 1 | Data Byte 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| ACRO[7:0] | ACR1[7:4] (ACR1[3:0] unused) | ACR2[7:0] | ACR3[7:0] |
| AMRO[7:0] | AMR1[7:4] (AMR1[3:0] unused) | AMR2[7:0] | AMR3[7:0] |

此时，ACR2和ACR3若没有用到，对应的AMR2和AMR3需配置为全1。RTR是否需要匹配，结合实际情况配置AMR。

SFF，双滤波ID数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| CAN\_TXRX1 | CAN\_TXRX2 | CAN\_TXRX3 | CAN\_TXRX4 |
| ID.28..ID21 | ID.20..ID.18 RTR XXXX | Data Byte 1 | Data Byte 2 |

滤波ID1

|  |  |  |
| --- | --- | --- |
| ACR0[7:0] | ACR1[7:0] | ACR3[3:0] |
| AMR0[7:0] | AMR1[7:0] | AMR3[3:0] |

滤波ID2

|  |  |
| --- | --- |
| ACR2[7:0] | ACR3[7:4] |
| AMR2[7:0] | AMR3[7:4] |

此时，ACR若没有用到，对应的AMR需配置为全1。例如滤波ID1的ACR1[3:0]和ACR[3:0]。RTR是否需要匹配，结合实际情况配置AMR。

EFF，单滤波ID数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| CAN\_TXRX1 | CAN\_TXRX2 | CAN\_TXRX3 | CAN\_TXRX4 |
| ID.28..ID21 | ID.20..ID.13 | ID.12..ID.5 | ID.4..ID.0 RTR X X |

滤波ID

|  |  |  |  |
| --- | --- | --- | --- |
| ACR0[7:0] | ACR1[7:0] | ACR2[7:0] | ACR3[7:2] |
| AMR0[7:0] | AMR1[7:0] | AMR2[7:0] | AMR3[7:2] |

此时，ACR3[1:0]没有用到，对应的AMR需配置为全1。

EFF，双滤波ID数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| CAN\_TXRX1 | CAN\_TXRX2 | CAN\_TXRX3 | CAN\_TXRX4 |
| ID.28..ID21 | ID.20..ID.13 | ID.12..ID.5(not matched) | ID.4..ID.0 RTR X X(not matched) |

滤波ID1

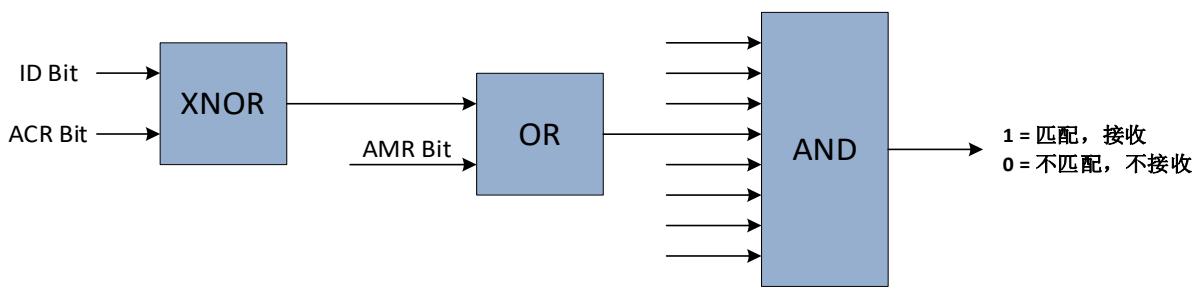
|  |  |
| --- | --- |
| ACR0[7:0] | ACR1[7:0] |
| AMR0[7:0] | AMR1[7:0] |

滤波ID2

|  |  |
| --- | --- |
| ACR2[7:0] | ACR3[7:0] |
| AMR2[7:0] | AMR3[7:0] |

此时，实际接收的ID12—ID5，不同ACR比较。

ACR和AMR的关系如下：

  
图19-3 CAN模块ID滤波逻辑关系

# 19.2.2.7 发送帧格式

发送帧分成ID部分和数据部分。第一个字节包含了帧分类等信息，确定是SFF（标准）帧还是EFF（扩展）帧；确定是远程帧，还是数据帧；以及，确定数据长度。SFF的ID长度为2个字节，EFF的ID长度为4个字节。数据长度为最大8个字节。

表19-1发送帧结构

|  |  |  |  |
| --- | --- | --- | --- |
| SFF | | EFF | |
| 地址 | 域 | 地址 | 域 |
| 0x40 | TX 帧信息 | 0x40 | TX 帧信息 |
| 0x44 | TX ID0 | 0x44 | TX ID0 |
| 0x48 | TX ID1 | 0x48 | TX ID1 |
| 0x4C | TX DATA0 | 0x4C | TX ID2 |
| 0x50 | TX DATA1 | 0x50 | TX ID3 |
| 0x54 | TX DATA2 | 0x54 | TX DATA0 |
| 0x58 | TX DATA3 | 0x58 | TX DATA1 |
| 0x5C | TX DATA4 | 0x5C | TX DATA2 |
| 0x60 | TX DATA5 | 0x60 | TX DATA3 |
| 0x64 | TX DATA6 | 0x64 | TX DATA4 |
| 0x68 | TX DATA7 | 0x68 | TX DATA5 |
| 0x6C | unused | 0x6C | TX DATA6 |
| 0x70 | unused | 0x70 | TX DATA7 |

# SFF帧非数据信息

表19-2发送SFF头信息

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN Address | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
| 0x40 | FF | RTR | X1 | X1 | DLC.3 | DLC.2 | DLC.1 | DLC.0 |
| 0x44 | ID.28 | ID.27 | ID.26 | ID.25 | ID.24 | ID.23 | ID.22 | ID.21 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x48 | ID.20 | ID.19 | ID.18 | X2 | X1 | X1 | X1 | X1 |

EFF帧非数据信息

表19-3发送EFF头信息

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN Address | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
| 0x40 | FF | RTR | X1 | X1 | DLC.3 | DLC.2 | DLC.1 | DLC.0 |
| 0x44 | ID.28 | ID.27 | ID.26 | ID.25 | ID.24 | ID.23 | ID.22 | ID.21 |
| 0x48 | ID.20 | ID.19 | ID.18 | ID.17 | ID.16 | ID.15 | ID.14 | ID.13 |
| 0x4C | ID.12 | ID.11 | ID.10 | ID.9 | ID.8 | ID.7 | ID.6 | ID.5 |
| 0x50 | ID.4 | ID.3 | ID.2 | ID.1 | ID.0 | X2 | X1 | X1 |

FF：1表示是EFF（扩展）帧，0表示是SFF（标准）帧。

RTR：1标识是remote（远程）帧，0表示是data（数据）帧。

DLC：表示此帧要发送数据的长度。最大为8个字节，最小为0个字节。

ID：帧标识号。SFF帧的ID长度为11位（ID.28到ID.18）。EFF帧的ID长度为29位（ID.28到ID.0）。高位优先发送。

DATA：数据。字节间顺序，从大到小，即TXDATA7先发送。字节内顺序，从高位到低位。

X2：最好同RTR值一致

X1：1或0均可

19.2.2.8接收帧格式

接收帧分成ID部分和数据部分。第一个字节包含了帧分类等信息，确定是SFF（标准）帧还是EFF（扩展）帧；确定是远程帧，还是数据帧；以及，确定数据长度。SFF的ID长度为2个字节，EFF的ID长度为4个字节。数据长度为最大8个字节。

表19-4接收帧结构

|  |  |  |  |
| --- | --- | --- | --- |
| SFF | | EFF | |
| 地址 | 域 | 地址 | 域 |
| 0x40 | RX 帧信息 | 0x40 | RX 帧信息 |
| 0x44 | RX ID0 | 0x44 | RX ID0 |
| 0x48 | RX ID1 | 0x48 | RX ID1 |
| 0x4C | RX DATA0 | 0x4C | RX ID2 |
| 0x50 | RX DATA1 | 0x50 | RX ID3 |
| 0x54 | RX DATA2 | 0x54 | RX DATA0 |
| 0x58 | RX DATA3 | 0x58 | RX DATA1 |
| 0x5C | RX DATA4 | 0x5C | RX DATA2 |
| 0x60 | RX DATA5 | 0x60 | RX DATA3 |
| 0x64 | RX DATA6 | 0x64 | RX DATA4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0x68 | RX DATA7 | 0x68 | RX DATA5 |
| 0x6C | unused | 0x6C | RX DATA6 |
| 0x70 | unused | 0x70 | RX DATA7 |

SFF帧非数据信息

表19-5接收SFF头信息

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN Address | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
| 0x40 | FF | RTR | 0 | 0 | DLC.3 | DLC.2 | DLC.1 | DLC.0 |
| 0x44 | ID.28 | ID.27 | ID.26 | ID.25 | ID.24 | ID.23 | ID.22 | ID.21 |
| 0x48 | ID.20 | ID.19 | ID.18 | RTR | 0 | 0 | 0 | 0 |

EFF帧非数据信息

表19-6接收EFF头信息

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CAN Address | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
| 0x40 | FF | RTR | 0 | 0 | DLC.3 | DLC.2 | D LC.1 | DLC.0 |
| 0x44 | ID.28 | ID.27 | ID.26 | ID.25 | ID.24 | ID.23 | ID.22 | ID.21 |
| 0x48 | ID.20 | ID.19 | ID.18 | ID.17 | ID.16 | ID.15 | ID.14 | ID.13 |
| 0x4C | ID.12 | ID.11 | ID.10 | ID.9 | ID.8 | ID.7 | ID.6 | ID.5 |
| 0x50 | ID.4 | ID.3 | ID.2 | ID.1 | ID.0 | RTR | 0 | 0 |

# 19.2.2.9发送

CAN模块要发送数据，必须在正常工作模式下。

执行发送操作，一般推荐开启发送中断源和错误警告数量中断源，而仲裁丢失和总线错误中断源头不强迫开启，CAN模块有自动重发机制。

发送的数据，存放在TXFIFO中，TXFIFO为32- Byte，及一次可以存几帧数据，且TXFIFO中的数据一旦被成功发送出去，就被CAN模块释放掉供新的数据写入。一帧数据长度为13个字节，其考虑到最大数据量为8个字节。在发送之前，需确保此时模块处于空闲状态。

发送帧的格式，分成标准帧（SFF）和扩展帧（EFF）。CAN\_CMR的BIT0置1，触发CAN模块发送数据。一旦CAN总线空闲，数据就会发出。传输完毕与否，一方面可以检查CAN\_SR的BIT2是否变成1（1表示空闲），也可以通过中断方式检查是否发送完毕。

在数据发送到总线前，设置CAN\_CMR的BIT1，可以停止发送，产生相应中断；通过CAN\_SR的BIT3，可知当前数据是否发送完毕。

# 19.2.2.10接收

CAN模块要接收数据，必须在正常工作模式下。

接收 FIFO 中存储的帧大小可以不同（3～13 byte 范围之间）。当接 FIFO 为满时（或剩余的空间不足以完全存储一帧），将触发溢出中断，后续的接收帧将丢失，直到接收 FIFO 中清除出足够的存储空间。接收 FIFO 中的第一条帧将被映射到 13- byte 的接收缓冲器中，直到该帧被清除（通过释放接收缓冲器指令）。清除后，接收缓冲器将继续映射接收 FIFO 中的下一条帧，接收 FIFO 中上一条已清除帧的空间将被释放。

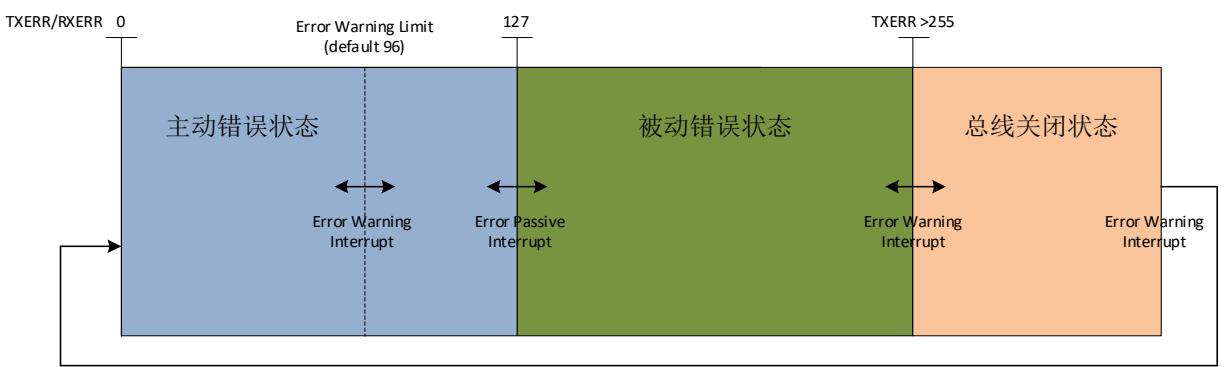
接收，开始于侦测到 Start 帧。接收到的数据，执行 ID 匹配操作：

匹配成功，存入 RX FIFO，RMC 递增。RX FIFO 为 32- Byte，即一次可以存几帧数据，且 RX FIFO 中的数据一旦被 DMA 成功取走，就被 CAN 模块释放掉供新的数据写入（非 DMA 模式，需要对 CAN\_CMR BIT2 写 1 释放 RX FIFO）。一帧数据长度为 13 个字节，其考虑到最大数据量为 8 个字节。RX FIFO 中有几个有效帧，通过 CAN\_AMC 寄存器可知。通过读取 CAN\_TXRX 寄存器可以获得最先被接收到的数据帧。若 RX FIFO 满了，抛弃此帧，RX FIFO 不接收，同时产生溢出中断（使能前提下）。

匹配失败，存入 RX FIFO，RMC 不会递增。因此，数据也会被后续帧覆盖。若 RX FIFO 满了，抛弃此帧，RX FIFO 不接收，不产生溢出中断（使能前提下）。

# 19.2.2.11 错误管理

CAN 协议要求每个节点中都包含发送错误计数和接收错误计数。这两个错误计数的数值决定了控制器当前的错误状态（如主动错误、被动错误、离线）。控制器将数值分别存储在 CAN\_RXERR 和 CAN\_TXERR 中，CPU 可随时进行读取。除了错误状态之外，控制器还提供错误报警限制的功能，这个功能可在 CAN 控制器进入被动错误状态之前，提醒用户，当前发生的严重总线错误。

  
图 19-4 CAN 模块错误管理

若发生总线错误，总线错误中断标志位（CAN\_IR.BUS\_ERR\_IF）置位，对应中断使能的话，将产生对应中断。当总线错误积累到一定数值，总线错误报警中断位（CAN\_IR.W\_ERR\_IF）置位，对应中断使能的话，将产生对应中断；CAN\_SR.ERROR\_OV 置位，表明错误超过警戒线。若继续增加，超过 127，进入被动错误状态，被动错误中断位（CAN\_IR.M\_ERR\_IF）置位，对应中断使能的话，将产生对应中断。若总线错误继续增加（每次增加总线错误中断标志位都会置位），发送错误超过 255，CAN\_SR.ON\_BUS 置位进入总线关闭状态。此时，开始离线恢复，从 127 开始递减，减到 0 完成恢复，CAN\_SR.ON\_BUS 和 CAN\_SR.ERROR\_OV 自动清除。

只要发生总线错误，总线错误中断标志位（CAN\_IR.BUS\_ERR\_IF）均置位。

# 19.2.2.12错误计数

发送错误计数和接收错误计数根据以下规则递增/递减。请注意，一帧传输中可应用多个规则。

1. 当接收器检测到错误时，接收错误计数数值将增加1。当检测到的错误为发送主动错误标志或过载标志期间的位错误除外。
2. 发送错误标志后，当接收器第一个检测到的位是显性位时，接收错误计数数值将增加8。
3. 当发送器发送错误标志时，发送错误计数数值增加8。但是，以下情况不适用于该规则：

·发送器为被动错误状态，因为在应答槽未检测到显性位而产生应答错误，且在发送被动错误标志时检测到显性位时，则发送错误计数数值不应增加。

·发送器在仲裁期间因填充错误而发送错误标志，且填充位本该是隐性位但是检测到显性位，则发送错误计数数值不应增加。

1. 若发送器在发送主动错误标志和过载标志时检测到位错误，则发送错误计数数值增加8。
2. 若接收器在发送主动错误标志和过载标志时检测到位错误，则接收错误计数数值增加8。
3. 任意节点在发送主动/被动错误标志或过载标志后，节点仅能承载最多7个连续显性位。在（发送主动错误标志或过载标志时）检测到第14个连续显性位，或在被动错误标志后检测到第8个连续显性位后，发送器将使其发送错误计数数值增加8，而接收器将使其接收错误计数数值增加8。每增加8个连续显性位的同时，（发送器的）发送错误计数和（接收器的）接收错误计数数值也将增加8。
4. 每当发送器成功发送帧后（接收到ACK，且直到EOF完成未发生错误），则发送错误计数数值将减小1，除非发送错误计数的数值已经为0。
5. 当接收器成功接收帧后（确认槽前未检测到错误，且成功发送ACK），则接收错误计数数值将相应减小。

·若接收错误计数数值位于 之间，则其值减小1。

·若接收错误计数数值大于127，则其值减小到127。

·若接收错误计数数值为0，则仍保持为0。

1. 当一个节点的发送错误计数和/或接收错误计数数值大于等于128时，该节点变为被动错误节点。导致节点发生上述状态切换的错误，该节点仍发送主动错误标志。请注意，一旦接收错误计数数值到达128，后续任何增加该值的动作都是无效的，直到接收错误计数数值返回到128以下。
2. 当某节点的发送错误计数数值大于等于256时，该节点将变为离线节点。
3. 当某被动错误节点的发送错误计数和接收错误计数数值都小于等于127，则该节点将变为主动错误节点。
4. 当离线节点在总线上检测到128次11个连续隐性位后，该节点可变为主动错误节点（发送错误计数和接收错误计数数值都重设为0）。

# 19.2.2.13错误报警限制

错误报警限制（CAN\_EWLR）为发送和接收的可配置闻值，若错误计数数值超过该闻值，将触发CAN总线错误中断，作为一个报警功能提示当前发生的严重CAN总线错误，且在CAN控制器进入被动错误状态之前被触发。CAN\_EWLR寄存器存储相应闻值，此寄存器的配置必须处于复位模式下。CAN\_EWLR寄存器默认数值为96。

当发送和/或接收错误的数值大于等于CAN\_EWLR寄存器对应的数值时，CAN\_IR.W\_ERR\_IF位将立即被置1。同理，当发送和/或接收错误的数值都小于CAN\_EWLR寄存器对应的数值时，CAN\_IR.W\_ERR\_IF位将立即复位为0。只要CAN\_IR.W\_ERR\_IF（CAN\_EIR.W\_ERR\_IF使能）位值变成1，便会触发错误报警中断。

# 19.2.2.14被动错误

19.2.2.14被动错误当发送或接收错误的数值大于127时，CAN控制器处于被动错误状态，CAN\_IR.M\_ERR\_IF位将立即被置1。同理，当发送或接收错误的数值都小于等于127时，CAN控制器进入主动错误状态。每当CAN控制器从主动错误状态变为被动错误状态，或反之时，CAN\_IR.M\_ERR\_IF位都将立即被置1。

# 19.2.2.15离线状态与离线恢复

当发送错误的数值大于255时，CAN控制器将进入离线状态。进入离线状态后，CAN控制器将自动进行以下动作：

·接收错误计数器的数值置为0·发送错误计数器的数值置为127·CAN\_IR.BUS\_ERR\_IF位置1·进入CAN的复位模式

每当CAN\_IR.BUS\_ERR\_IF/CAN\_IR.W\_ERR\_IF的数值发生变化时，都将触发错误报警中断。

为了返回主动错误状态，CAN控制器必须进行离线恢复。要启动离线恢复，首先需要退出复位模式，进入操作模式。然后要求CAN控制器在总线上检测到128次11个连续隐性位。每一次CAN控制器检测到11个连续隐性位时，发送错误计数器的数值都将减小，以追踪离线恢复进程。当离线恢复完成后（发送错误的数值从127减小到0），CAN\_IR.BUS\_ERR\_IF位将自动复位为0，从而触发错误报警中断。

# 19.2.3 寄存器

# 19.2.3.1地址分配

CAN模块寄存器的基地址是0x4001\_3400，寄存器列表如下。

表19-7CAN寄存器地址分配

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| CAN\_MOD | 0x000 | CAN 模式寄存器 |
| CAN\_CMR | 0x004 | CAN 命令寄存器 |

|  |  |  |
| --- | --- | --- |
| CAN\_SR | 0x008 | CAN 状态寄存器 |
| CAN\_IR | 0x00C | CAN 中断状态寄存器 |
| CAN\_IER | 0x010 | CAN 中断控制寄存器 |
| CAN\_BTR0 | 0x018 | CAN 总线时序控制寄存器 0 |
| CAN\_BTR1 | 0x01C | CAN 总线时序控制寄存器 1 |
| CAN\_ALC | 0x02C | CAN 仲裁丢失捕捉寄存器 |
| CAN\_ECC | 0x030 | CAN 错误码捕捉寄存器 |
| CAN\_EWLR | 0x034 | CAN 错误&amp;警告门限值设置寄存器 |
| CAN\_RXERRR | 0x038 | CAN 接收错误计数器 |
| CAN\_TXERR | 0x03C | CAN 发送错误计数器 |
| CAN\_TXRX0 | 0x040 | 正常工作模式下，CAN 发送帧格式寄存器/CAN 接收帧格式寄存器 |
| CAN\_TXRX1 | 0x044 | 正常工作模式下，CAN 发送数据寄存器 0/CAN 接收数据寄存器 0 |
| CAN\_TXRX2 | 0x048 | 正常工作模式下，CAN 发送数据寄存器 1/CAN 接收数据寄存器 1 |
| CAN\_TXRX3 | 0x04C | 正常工作模式下，CAN 发送数据寄存器 2/CAN 接收数据寄存器 2 |
| CAN\_TXRX4 | 0x050 | 正常工作模式下，CAN 发送数据寄存器 3/CAN 接收数据寄存器 3 |
| CAN\_TXRX5 | 0x054 | 正常工作模式下，CAN 发送数据寄存器 4/CAN 接收数据寄存器 4 |
| CAN\_TXRX6 | 0x058 | 正常工作模式下，CAN 发送数据寄存器 5/CAN 接收数据寄存器 5 |
| CAN\_TXRX7 | 0x05C | 正常工作模式下，CAN 发送数据寄存器 6/CAN 接收数据寄存器 6 |
| CAN\_TXRX8 | 0x060 | 正常工作模式下，CAN 发送数据寄存器 7/CAN 接收数据寄存器 7 |
| CAN\_TXRX9 | 0x064 | 正常工作模式下，CAN 发送数据寄存器 8/CAN 接收数据寄存器 8 |
| CAN\_TXRXA | 0x068 | 正常工作模式下，CAN 发送数据寄存器 9/CAN 接收数据寄存器 9 |
| CAN\_TXRXB | 0x06C | 正常工作模式下，CAN 发送数据寄存器 10/CAN 接收数据寄存器 10 |
| CAN\_TXRXC | 0x070 | 正常工作模式下，CAN 发送数据寄存器 11/CAN 接收数据寄存器 11 |
| CAN\_ACR0 | 0x040 | 复位模式下，CAN ID 码寄存器 0 |
| CAN\_ACR1 | 0x044 | 复位模式下，CAN ID 码寄存器 1 |
| CAN\_ACR2 | 0x048 | 复位模式下，CAN ID 码寄存器 2 |
| CAN\_ACR3 | 0x04C | 复位模式下，CAN ID 码寄存器 3 |
| CAN\_AMR0 | 0x050 | 复位模式下，CAN ID 掩码寄存器 0 |
| CAN\_AMR1 | 0x054 | 复位模式下，CAN ID 掩码寄存器 1 |
| CAN\_AMR2 | 0x058 | 复位模式下，CAN ID 掩码寄存器 2 |
| CAN\_AMR3 | 0x05C | 复位模式下，CAN ID 掩码寄存器 3 |
| CAN\_RMC | 0x074 | CAN FIFO 有效接收信息计数器 |
| CAN\_RBSA | 0x078 | CAN 第一条有效接收信息在 FIFO 中的地址寄存器 |
| CAN\_RFIFO0～CAN\_RFIFO31 | 0x080～0x144 | CAN RX FIFO 地址寄存器 |
| CAN\_TFIFO0～CAN\_TFIFO31 | 0x180～0x1B0 | CAN TX FIFO 地址寄存器 |

# 19.2.3.2 寄存器说明

# 19.2.3.2.1 CAN\_MOD 模式寄存器

地址：0x4001\_3400

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4] | SLEEP | 正常工作模式下写1，触发休眠操作，进入休眠模式。 |
| [3] | FLT\_ID | CAN滤波ID选择。默认值为0。 1：单滤波ID。一个32位长的滤波ID 0：双滤波ID。两个短滤波ID |
| [2] | TEST\_MODE | CAN工作模式选择。默认值为0。 1：自测试模式 0：正常工作模式 |
| [1] | FUNC\_MODE | CAN工作模式选择。默认值为0。 1：监听模式 0：正常工作模式 |
| [0] | MODE | CAN工作模式选择。默认值为1。 1：复位模式 0：正常工作模式 |

# 19.2.3.2.2 CAN\_CMR 命令寄存器

地址：0x4001\_3404

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | DMA\_EN | RX\_DUR\_TX | CLR\_OV | RELEASE\_FIFO | INTR\_TRAN | TRANS\_REQ |
| WO | WO | WO | WO | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |

|  |  |  |
| --- | --- | --- |
| [31:6] |  | 未使用 |
| [5] | DMA\_EN | 写 1，使能 DMA 功能 |
| [4] | RX\_DUR\_TX | 写 1，发送数据的同时也将数据接收回来 |
| [3] | CLR\_OV | 写 1，清除数据溢出标志位 |
| [2] | RELEASE\_FIFO | 写 1，释放 RIFIFO |
| [1] | INTR\_TRANS | 写 1，将中断取消未执行的发送传输 |
| [0] | TRANS\_REQ | 写 1，产生 CAN 发送传输请求 |

# 19.2.3.2.3 CAN\_SR状态寄存器

地址：0x4001\_3408

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  | ON\_BUS | ERR\_OV | TXING | RXING | TRANS\_DONE | TFIFO\_EMPTY | RFIFO\_EMPTY | DATA>AVAL | DATA |  |
| RO | RO | RO | RO | RO | RO | RO | RO | RO |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | ON\_BUS | 1：CAN 模块，处于 BUS OFF 状态，无数据发送接收动作 0：CAN 模块，可以发送数据到 CAN 总线，或接收 CAN 总线数据 |
| [6] | ERR\_OV | 1：CAN 传输产生的错误总数达到或超过 CAN\_EWL 规定值 0：CAN 传输产生的错误总数低于 CAN\_EWL 规定值 错误值低于 CAN\_EWLR，标志自动拉低 |
| [5] | TXING | 1：CAN 模块正在发送一帧数据 0：CAN 模块没有发送数据的动作 |
| [4] | RXING | 1：CAN 模块正在接收一帧数据 0：CAN 模块没有接收数据的动作 |
| [3] | TRANS\_DONE | 1：最近一次传输已完成 0：最近一次传输未完成 |
| [2] | TFIFO\_EMPTY | 1：TFIFO 空，可以写入发送数据 0：TFIFO 非空，内部数据没有发送完毕 |
| [1] | RFIFO\_EMPTY | 1：RFIFO 存入的帧太多，已满，导致数据丢失 0：RFIFO 未满 对 CAN\_CMR BIT[2]写 1，可清除此位（表示用户已经取走一帧数据，暂时 RFIFO 非满了） |

|  |  |  |
| --- | --- | --- |
| [0] | DATA>AVALI | 1: RFIFO存有一帧或多帧数据，可以通过RFIFO寄存器读取 0: RFIFO没有有效帧数据 |

# 19.2.3.2.4 CAN\_IR中断状态寄存器

地址：0x4001\_340C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| BUS\_ERR\_IF |  | LOST\_ERR\_IF |  | M\_ERR\_IF |  | WK\_IF |  | RFIFO\_OV\_IF |  | W\_ERR\_IF |  | TX\_DONE\_IF |  | RFIFO\_N\_EMPTY\_IF |  |
| RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | BUS\_ERR\_IF | 1：总线错误中断（CAN\_EIR.7为有效前提下） |
| [6] | LOST\_ARB\_IF | 1：丢失仲裁，改成接收模式（CAN\_EIR.6为有效前提下） |
| [5] | M\_ERR\_IF | 被动错误中断。若值为1，表明节点由于错误计数数值的变化，在主动错误状态与被动错误状态间发生了切换。 |
| [4] | WK\_IF | 1：CAN模块从休眠中唤醒（CAN\_EIR.4为有效前提下） |
| [3] | RFIFO\_OV\_IF | 1：RFIFO数据发生溢出（CAN\_EIR.3为有效前提下） |
| [2] | W\_ERR\_IF | 错误报警中断。若值为1，表明SR寄存器中CAN\_SR.7和CAN\_SR.6发生变化（0变为1或1变为0）。 1：CAN\_SR.7或者CAN\_SR.6有变化（CAN\_EIR.2为有效前提下） |
| [1] | TX\_DONE\_IF | 4：发送完毕当前帧（CAN\_EIR.1为有效前提下），可以执行新的数据发送任务 |
| [0] | RFIFO\_N\_EMP TY\_IF | 1：RFIFO有数据（CAN\_EIR.0为有效前提下） |

CAN\_IR寄存器，为读清除寄存器。只有BIT0- - RFIFO\_N\_EMPTY\_IF无法读清除，此位只能通过对CAN\_CMRBIT2写1清除，若RXFIFO有多帧接收的数据，CAN\_CMRBIT2写1也无法清除，除非RXFIFO全部读完，即读一帧数据写CAN\_CMRBIT2，反复迭代。

# 19.2.3.2.5 CAN\_EIR中断控制寄存器

地址：0x4001\_3410

复位值：0x0

表19-12中断控制寄存器CAN\_EIR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| BUS\_ERR\_IE |  | LOST\_ERR\_IE |  | M\_ERR\_IE |  | WK\_IE |  | RFIFO\_OV\_IE |  | CAN\_SRG7\_IE |  | TX\_DONE\_IE |  | RFIFO\_N\_EMPTY\_IE |  |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | BUS\_ERR\_IE | 1：总线错误中断，中断源使能 0：该中断源关闭 |
| [6] | LOST\_ERR\_IE | 1：丢失仲裁，改成接收模式，中断源使能 0：该中断源关闭 |
| [5] | M\_ERR\_IE | 1：被动错误中断源使能 0：该中断源关闭 |
| [4] | WK\_IE | 1：CAN 模块从休眠中唤醒，中断源使能 0：该中断源头关闭 |
| [3] | RFIFO\_OV\_IE | 1：RFIFO 数据发生溢出，中断源使能 0：该中断源头关闭 |
| [2] | W\_ERR\_IE | 1：错误报警中断，CAN\_SR.7 或者 CAN\_SR.6 有变化，中断源使能 0：该中断源头关闭 |
| [1] | TX\_DONE\_IE | 1：发送完毕当前帧，中断源使能 0：该中断源头关闭 |
| [0] | RFIFO\_N\_EMP TY\_IE | 1：RFIFO 有新的数据被接收到，中断源使能 0：该中断源头关闭 |

# 19.2.3.2.6 CAN\_BTR0波特率0控制寄存器

地址：0x4001\_3418

复位值：0x0

表19-13波特率0控制寄存器CAN\_BTR0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  | SJW | BAUDRATE |
| RW | RW |
| 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:6] | SJW | 同步跳转宽度配置。公式如下： Tsjw = TQ*(SJW + 1) 波特率配置，计算 CAN 传输基本时间单元参数 TQ,Tclk 为 CAN 模块的系统时钟频率 TQ = 2*Tclk\*(BAUDRATE+1) |
| [5:0] | BAUDRATE |

# 19.2.3.2.7 CAN\_BTR1波特率1控制寄存器

地址：0x4001\_341C

复位值：0x0

表19-14波特率1控制寄存器CAN\_BTR1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | OSR | SEG2 | | | SEG1 | | | |  |
| RW | RW | | | RW | | | |  |
| 0 | 0 | | | 0 | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7] | OSR | 1: 1-bit 数据被采样 3 次 0: 1-bit 数据被采样 1 次 |
| [6:4] | SEG2 | SEG2 段时间 Tseg2 = TQ\*(SEG2+1) |
| [3:0] | SEG1 | SEG1 段时间 Tseg1 = TQ\*(SEG1+1) |

# 19.2.3.2.8 CAN\_ALC仲裁丢失捕捉寄存器

地址：0x4001\_342C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表19-15仲裁丢失捕捉寄存器CAN\_ALC | | | | | | | | | | | | | | |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|  |  |
| --- | --- |
|  | LOST\_ARB |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4:0] | LOST\_ARB | 记录总线仲裁丢失的具体位置。 0：ID 的第一位 1：ID 的第二位 2：ID 的第三位 3：ID 的第四位 4：ID 的第五位 5：ID 的第六位 6：ID 的第七位 7：ID 的第八位 8：ID 的第九位 9：ID 的第十位 A：ID 的第十一位 B：SRTR 位 C：IDE 位 |

# 19.2.3.2.9 CAN\_ECC 错误码捕捉寄存器

地址：0x4001\_3430

复位值：0x0

表19-16错误码捕捉寄存器CAN\_ECC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
|  | | | | | | BUS\_ERR\_TYPE | ERR\_TIMING | ERR\_POSITION | | | | | | |  |
| RO | RO | RO | | | | | | |  |
| 0 | 0 | 0 | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:6] | BUS\_ERR\_TYPE | 总线错误类型 00: bit 错误 01: 格式错误 10: 填充错误 11: 其它类型错误 |
| [5] | ERR\_TIMING | 1: 总线错误发生在接收过程中 |

|  |  |  |
| --- | --- | --- |
|  |  | 0：总线错误发生在发送过程中 |
| [4:0] | ERR\_POSITION | 错误发生的位置 02：ID.28 到 ID.21 03：起始帧 04：SRTR 位 05：IDE 位 06：ID.20 到 ID.18 07：ID.17 到 ID.13 08：CRC 数据段 09：保留位，固定为 0 0A：Data Field 0B：Data Length Code 0C：RTR 位 0D：保留位，固定为 1 0E：ID.4 到 ID.0 0F：ID.12 到 ID.5 11：active error flag 12：Intermission 13：Tolerate dominant 位 16：passive error flag 17：Error delimiter 18：CRC delimiter 19：确认位 1A：帧结束 1B：确认 delimiter 1C：Overload flag |

# 19.2.3.2.10 CAN\_EWLR 错误&警告门限值寄存器

地址：0x4001\_3434

复位值：0x0

表19-17错误&警告门限值寄存器CAN\_EWLR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | EWLR | | | | | | | | |  |
| RW | | | | | | | | |  |
| 0 | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | EWLR | 正常工作模式下只读，复位模式下可读写。写入新值后，在正常工作模式下生效。 |

# 19.2.3.2.11 CAN\_RXERRR接收错误计数器寄存器

地址：0x4001\_3438

复位值：0x0

|  |  |  |
| --- | --- | --- |
|  | | RXERR |
| RW |
| 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | RXERRR | 正常工作模式下只读，复位模式下可读写。 错误数超过 CAN\_EWLR，CAN\_SR\_ERR\_OV 被拉高；数值超过 127，CAN 模块进入被动错误状态。复位或者 BUS OFF 事件，硬件清零本寄存器。本寄存器自动增减，具体增减规则见本章 19.2.2.12。 |

# 19.2.3.2.12 CAN\_TXERRR发送错误计数器寄存器

地址：0x4001\_343C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | TXERR | | | | | | |  |
| RW | | | | | | |  |
| 0 | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | TXERR | 正常工作模式下只读，复位模式下可读写。 错误数超过 CAN\_EWLR，CAN\_SR\_ERR\_OV 被拉高；数值超过 127，CAN 模块进入被动错误状态。数值超过 255，进入 BUS OFF 状态，CAN\_MOD.MODE 被硬件拉高，处于复位模式。CAN\_SR.ON\_BUS 被硬件自动拉高，本寄存器被硬件赋值为 127；复位，硬件清零本寄存器。具体增减规则见本章 19.2.2.12。 |

# 19.2.3.2.13 CAN\_TXRX0~CAN\_TXRXC 发送接收寄存器

地址：0x4001\_3440~0x4001\_3470

复位值：0x0

表19-20发送接收寄存器CAN\_TXRX

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | TXRX | 发送或者接收数据寄存器。 正常工作模式下，该寄存器映射的是RFIFO中第一个有效帧数据； 正常工作模式下，对该寄存器写入数据，直接存入TFIFO中。 |

# 19.2.3.2.14 CAN\_ACR ID 寄存器

地址：0x4001\_3440~0x4001\_344C

复位值：0x0

表19-21ID寄存器CANACR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | ACR | 仅复位模式下可用；复位模式下，可读写； 可接收的ID寄存器，当输入帧的ID同此匹配，则被CAN模块接受。配合AMR寄存器，可同一类ID匹配。 |

# 19.2.3.2.15 CAN\_AMR ID 椭码寄存器

地址：0x4001\_3450~0x4001\_345C

复位值：0x0

表19-22ID掩码寄存器CAN\_AMR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | |  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] | AMR | 未使用 |
| [7:0] | 仅复位模式下可用；复位模式下，可读写； 配合ACR寄存器，同输入ID进行匹配。AMR寄存器某一位为0，标识ID对应位需要同ACR匹配；为1，标识ID对应位不需要同ACR匹配。 |

# 19.2.3.2.16 CAN\_RMC FIFO 有效接收数据寄存器

地址：0x4001\_3474

复位值：0x0

表19-23FIFO有效接收数据寄存器CAN\_RMC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | FRAME\_CNT | | | | |  |  |
| RO | |
| 0 | |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4:0] | FRAME\_CNT | 正常工作模式和复位模式，此寄存器均只可读。 CAN收到一帧数据，计数器自动累加1；DMA方式读取完毕一帧数据后，计数器自动减1；CPU方式读取完毕一帧数据后，软件需将CAN-CMR[2]写1，手动实现计数器减1。 |

# 19.2.3.2.17 CAN\_RBSA 有效接收数据地址寄存器

地址：0x4001\_3478

复位值：0x0

表19-24有效接收数据地址寄存器CAN\_RBSA

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  | | | | | | |  | | | | | | | | | RBSA | | | | |
|  | | | | | | | | | RW | | | | |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4:0] | RBSA | 正常工作模式，只可读。复位模式，可读写 RFIFO的大小为32字节，本寄存器指明第一帧有效接收数据在RFIFO的位置。 |

# 19.2.3.2.18 CAN\_RFIFO0~CAN\_RFIFO31 RX FIFO 寄存器

地址：0x4001\_3480~0x4001\_34FC

复位值：0x0

表19-25RXFIFO寄存器CAN\_RFIFO0~CAN\_RFIFO31

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | RX\_DATA | RX FIFO 地址 正常工作模式下，只可读；复位模式下，可读写。 |

# 19.2.3.2.19 CAN\_TFIFO0~CAN\_TFIFO12 TX FIFO 寄存器

地址：0x4001\_3580~0x4001\_35B0

复位值：0x0

表19-26TXFIFO寄存器CAN\_TFIFO0~CAN\_TFIFO12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | TX\_DATA | TX FIFO 地址 正常工作模式下和复位模式下，均只可读。 |

LKS32MC08x User Manual

# 20SIF

# 20.1概述

20.1 概述SIF 总线接口，支持 SIF 协议，液晶显示器与电动车控制器之间的数据传输，实现其对运行状态和故障的检测。

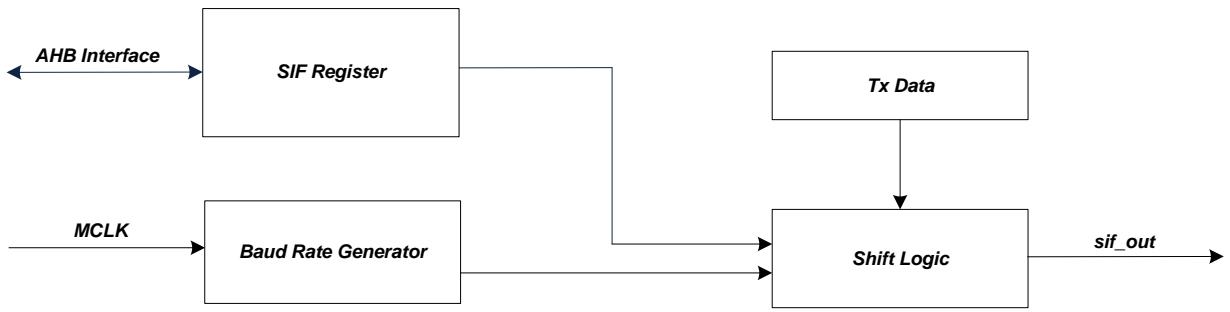
# 20.2主要特性

20.2 主要特性- 采用国际标准 SIF 通信协议，接口通用方便。- 主从方式采用单线单向传输，即只需要一根传输线路，电动车控制器为发送，仪表为接收方。- 传输线与电动车控制故障运行灯共用 I/O 口，不占用额外资源。- 传输波特率自适应范围宽，主机可以利用空闲时间发送数据。- 时间基准单位范围广，32us <Tosc < 320us。- 数据的电平遵守 TTL 规范。- 一个中断向量。

# 20.3功能描述

# 20.3.1 功能框图

本接口采用同步串行设计，实现CPU同外部设备之间的SIF传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。

  
图20-1SIF模块顶层功能框图

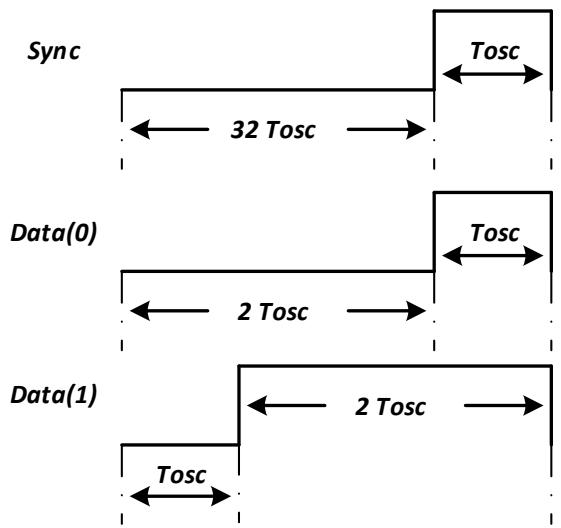
SIF接口同外界通讯只有sif\_out一根信号线

# 20.3.2 功能说明

20.3.2 功能说明SIF 接口仅支持主发送，模块发送数据，将数据从并行转换成串行。可以开启或禁止中断。接口通过数据引脚 sif\_out 连接到 SIF 总线。

# 20.3.3 模式选择

SIF接口仅有一种运行模式：主发送模式。因为SIF接口的传输速度很慢，不支持DMA传输。每次传输一个字节完毕后，产生中断信号。

  
图20-2SIF基本传输时序图

注：图17- 2占空比为2:1，可选配为3:1。SIF\_FREQ[0]控制占空比的选择。

# 20.3.4 SIF接口传输

默认情况下，SIF接口关闭。若需要发送数据，需要使能SIF接口。为了产生正确的时序，必须在SIF\_CFG中设定SIF接口的工作时钟。

# 20.3.5 中断处理

SIF接口仅含一种类型的中断事件- - 数据传输完成事件。

# 20.3.6 通讯速度设置

SIF接口的基本时间单位是32us- - Tosc。Tosc的产生来自系统时钟。在不同系统频率下，通过配置SIF\_FREQ寄存器，获得我们需要的基准时间。

# 20.4寄存器

# 20.4.1 地址分配

SIF模块寄存器的基地址是0x4001\_3800，寄存器列表如下：

表20-1SIF模块控制寄存器列表

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| SIF\_CFG | 0x00 | SIF 配置寄存器 |
| SIF\_FREQ | 0x04 | SIF 波特率配置寄存器 |
| SIF\_IRQ | 0x08 | SIF 中断寄存器 |
| SIF\_WDATA | 0x0C | SIF 发送数据寄存器 |

# 20.4.2 寄存器说明

20.4.2.1SIF\_CFG配置寄存器

地址：0x4001\_3800

复位值：0x0

表20-2地址寄存器SIF\_CFG

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | LAST\_BYTE | |  | ENDIAN | EN |
| RW | |  | RW | RW |
| 0 | |  | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4] | LAST\_BYTE | 当前字节传输完毕后，结束本次传输。SIF接口传输当前字节，字节传输完成后，产生中断。在中断处理程序中对该位写1，让SIF接口恢复默认状态。若不操作此位，表明还有数据要发送。该位读回永远是0。 |
| [3:2] |  | 未使用 |
| [1] | ENDIAN | SIF大小端设置。1，MSB；0，LSB。默认为0。 |
| [0] | EN | SIF模块使能。默认值为0。 1，使能 0，关闭 |

# 20.4.2.2SIF\_FREQ波特率寄存器

地址：0x4001\_3804

复位值：0x0

表20-3系统控制寄存器SIF\_FREQ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  | | | | | | |  | | | | | | TOSC | | | | |  | | DUTY |
| RW | | | | |  |  | RW |  |  |  |  |  |  |
| 0 | | | | |  |  | 0 |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:4] | TOSC | 96Mhz系统时钟下，基准时间单位Tosc设置。 01:32us; 02:64us; 03:96us; 04:128us 05:160us; 06:192us; 07:220us; 08:256us 09:288us; 10:320us; 其它均为32us。 |

|  |  |  |
| --- | --- | --- |
| [3:1] |  | 未使用 |
| [0] | DUTY | SIF 传输 Data 占空比 1: 占空比为 3:1 0: 占空比为 2:1 |

# 20.4.2.3SIF\_IRQ状态控制寄存器

地址：0x4001\_3808

复位值：0x0

表20-4状态控制寄存器SIF\_IRQ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | IF |  |  |  | IE |  |  |
| RW |  |  |  | RW |  |  |
| 0 |  |  |  | 0 |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:5] |  | 未使用 |
| [4] | IF | SIF 中断事件，高电平有效。写 1 清除。 |
| [3:1] |  | 未使用 |
| [0] | IE | SIF 中断使能信号，高电平有效。默认值为 0。 |

# 20.4.2.4SIF\_WDATA数据寄存器

地址：0x4001\_380C

复位值：0x0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|  | | | | | | |  | | | | | | WDATA | | | | | | | |
| RW | | | | | | | | |  |  |  |  |  |
| 0 | | | | | | | | |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:8] |  | 未使用 |
| [7:0] | WDATA | 数据寄存器，SIF接口使能后，对该寄存器写入发送数据将触发SIF的数据传输。 |

# 21看门狗

# 21.1概述

看门狗工作于低速RC时钟域LSI，即使用32kHz进行计数。支持2s、4s、8s、64s四档复位时间可选。

复位控制寄存器SYS\_RST\_CFC\_WDT\_EN可用于使能或禁用看门狗，SYS\_RST\_CFC\_WDT\_EN 则使能看门狗模块。复位源记录寄存器SYS\_RST\_SRC.WDT\_RST\_RCD记录了看门狗复位事件，SYS\_RST\_SRC.WDT\_RST\_RCD为高表示发生过看门狗复位。

看门狗复位是硬件全局复位，其作用域等同于外部引脚复位以及内部的上电复位。

# 21.2寄存器

# 21.2.1 地址分配

看门狗模块寄存器基地址为0x4000\_0000。

表21-1看门狗模块寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 偏移 | 说明 |
| SYS\_WDT\_CLR | 0x38 | 看门狗清零寄存器 |

# 21.2.2 寄存器说明

21.2.2.1SYS\_WDT\_CLR看门狗清零寄存器

表21-2看门狗清零寄存器SYS\_WDT\_CLR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDT\_CLR | | | | | | | | | | | | | | |  |
| WO | | | | | | | | | | | | | | |  |
| 0 | | | | | | | | | | | | | | |  |

|  |  |  |
| --- | --- | --- |
| 位置 | 位名称 | 说明 |
| [31:16] |  | 未使用 |
| [15:0] | WDT\_CLR | 写入字节 16'b0111\_1001\_1000\_1B2B1B0。高 13 位为密码，密码正确时，B[2:0]才能写入其中，B[2:1]为 MODE 2'b00: 64 秒复位 2'b01: 8 秒复位 |

|  |  |  |
| --- | --- | --- |
|  |  | 2'b10: 4秒复位 2'b11: 2秒复位 B[0]为CLR，写入1，则复位WDT计数器 |

看门狗清零前，需要向 SYS\_WR\_PROTECT 写入 0xCAFE，开启 WatchDog SYS\_WDT\_CLR 寄存器写入。

# 22 版本历史

表22-1文档版本历史

|  |  |  |
| --- | --- | --- |
| 时间 | 版本号 | 说明 |
| 2025.06.10 | 1.41 | 修改 CAN 波特率计算公式 |
| 2025.05.20 | 1.40 | 修订关于内部 4M RC 时钟精度，在 40～125℃范围内精度在±2.0%之内 |
| 2025.05.15 | 1.39 | 温度传感器章节增加说明 |
| 2025.05.13 | 1.38 | 增加关于 PVD 描述 |
| 2024.06.14 | 1.37 | 增加 NVR 校正值地址信息 |
| 2024.01.15 | 1.36 | 增加关于休眠模式的说明 增加关于 I2C 模块通讯速度的说明 |
| 2023.12.06 | 1.35 | CAN 常规波特率索引值修正，计算公式修正 |
| 2023.09.07 | 1.34 | 修订关于 ADC TIMER 触发的说明 增加 ADC 在 3.3V 供电时必须使用 1.2V 基准的说明 |
| 2023.08.15 | 1.33 | GPIO 功能框图增加 ESD 二极管 |
| 2023.03.18 | 1.32 | 修改时钟精度的描述 |
| 2023.02.28 | 1.31 | 修改 PVDSEL 的描述 |
| 2023.02.18 | 1.3 | 修改 MCPWM\_SDCFG 的描述 |
| 2023.02.11 | 1.29 | 增加关于关闭 PLL 等操作的说明 增加 CAN 通讯需要使用外部晶振的描述 修改 SYS\_AFE\_REG5\_RGPPD 的描述 |
| 2023.01.03 | 1.28 | 修改编码器寄存器描述 |
| 2022.11.07 | 1.27 | 增加 IO 与内部模拟电路间连接电阻描述 |
| 2022.10.29 | 1.26 | 修改 MCPWM\_DTHx0/MCPWM\_DTHx1 寄存器描述 |
| 2022.10.17 | 1.25 | 修订 DAC 校准输出公式 |
| 2022.09.23 | 1.24 | 修改 ADC 1.2V 基准时温度传感器使用说明 |
| 2022.08.18 | 1.23 | 增加 ADC/CMP 配置流程，简化 SYS\_AFE\_REG 部分描述 |
| 2022.04.08 | 1.22 | 增加 MCU 内核描述 |
| 2021.10.18 | 1.21 | 修订 FLASH 擦写次数的描述 |
| 2021.08.03 | 1.20 | 修订 CAN 模块部分描述 |
| 2021.06.07 | 1.19 | 修订 CAN 模块部分描述，修订 FLASH NVR 空间描述 |
| 2021.05.20 | 1.18 | 修订比较器 BEMFx\_MID 的描述 |
| 2020.06.12 | 1.17 | 增加了复位源的说明，修订 UTimer ETON 的说明，SYS\_AFE\_CMP 修改为 SYS\_AFE\_INFO，增加了 PWR\_WEAK 标志位的说明 |
| 2020.02.18 | 1.16 | 修订部分说明 |
| 2019.12.03 | 1.15 | 修订部分图示 |
| 2019.09.12 | 1.14 | 修订部分说明 |
| 2019.06.01 | 1.10 | 时钟部分章节修订 |
| 2019.03.18 | 1.0 | 针对发布的修订 |
| 2018.11.29 | 0.1 | 初始版本 |

# 免责声明

LKS和LKO为凌鸥创芯注册商标。

南京凌鸥创芯电子有限公司（以下简称：“Linko”）尽力确保本文档内容的准确和可靠，但是保留随时更改、更正、增强、修改产品和/或文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。

客户应针对应用需求选择合适的Linko产品，详细设计、验证和测试您的应用，以确保满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。

Linko在此确认未以明示或暗示方式授予Linko或第三方的任何知识产权许可。

Linko产品的转售，若其条款与此处规定不同，Linko对此类产品的任何保修承诺无效。

如有更早期版本文档，一切信息以此文档为准。