AC7840x 技术参考手册

文档版本：3.0.13

发布日期：2025-04-23

© 2013 - 2025 杰发科技

本文档包含杰发科技的专有信息。未经授权，严禁复制或披露本文档包含的任何信息。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。

修订信息

版本	日期	作者	描述
1.0.0	2022-12-12	AutoChips	文档初版
2.0.0	2023-03-31	AutoChips	1. CKGEN 章节 HSE 时钟添加说明：或外部最高 50MHz 时钟输入 2. flash 15.2 节“寿命：≥ 10000 周期”，改为“寿命：P-Flash ≥ 10,000 周期，D-Flash ≥ 100,000 周期” 3. 27.6.6 章节 修改从机 SCL 拉伸使能的注意事项 4. 修改 15.6.14 章节寄存器表格默认值 5. 修改 16.6.2.1 寄存器 FUNC_FORMAT 的描述：指针和复制方法交取值调换 6. 修改 16.4.4.1-16.4.4.4 章节表格中 PAGE_LENGTH 所在的地址从 EF 字节修改到 CD 字节 7. 10.4.14、10.3.3 和 32.4.1 节：PWM Clock MUX 删除 SPLLDIV1 的配置 8. 25.4.10 节：修改表 25-7“标志”列，且补充 UART 中断至 13 个 9. 29.5.6 和 29.5.7 节：补充 EIO_I2S 使用 DMA 时的注意事项
2.0.1	2023-04-26	AutoChips	1. 6.4.2 节增加 MLMDR0 寄存器 2. 20.5 节补充“DMA 搬运 SRAM 数据时，限制 SRAM 数据的存放位置只能在 SRAM_U 区” 3. ADC：修改 30.6.14 节与 30.6.15 节校准寄存器 0 和 1 描述 4. ACMP：修改 31.6.1 节 ACMP_CR0 寄存器迟滞默认档位为 10mV 5. 17.3.1、17.4.1、17.4.3：完善 ERR_ST 域中 2bit 和 1bit SRAM ECC 错误的状态值
2.0.2	2023-08-08	AutoChips	1. CAN：26.5.1 表 26-13 中 tSeg_2 数值最小值改为 2，删除表 26-16 中 8M 波特率示例 2. EIO：29.6.17 节 bit1:0 TSTART 改为 bit1 TSTART 3. DMA：20.6.2 节寄存器，加入长度错误 bit 位 4. 3.2 节去掉 SPI3 的中断 5. MBIST & ABIST 功能释放： ①6 章节，添加 MBIST Cache 相关状态寄存器 ②15.4.2 添加 MBIST ABIST 地址 ③15.5 添加 MBIST ABIST 配置 6. UART：25.1.1 节修正型号选择表格 7. ADC：30.4.5 节修正 ADC 校准公式

© 2013 - 2024 杰发科技有限公司

3.0.0	2023-10-16	AutoChips	1. 第 12 和 13 章：补充 AC78408 和 AC78409 型号支持的 standby 模式描述、以及相关寄存器 2. CTU 24.4.6 节：补充 TIMER 捕获 RTC 时钟源的限制 3. ADC：30.4.5 节更新校准电压只能用 VREF 做参考；30.6.13 节序列长度寄存器补充 bit18-16 内部通道电压值描述 4. RTC 36.3/36.4.1/36.4.8/36.6.1 节：更新 RTC 对于 Standby 功能的适配说明
3.0.1	2023-12-20	AutoChips	1. 章节“32.2 特性”中，“支持正交解码（AB 相输入引脚映射到每个 PWM 模块的 CH0 和 CH1）”修改为“支持正交解码（ABZ 相输入引脚映射到每个 PWM 模块的 CH0、CH1 和 CH2）” 2. 章节“32.4.3 工作模式”表格中，Hall 功能行 channel 0~3 改成 0~2，channel 4~7 改成 3~7. 3. 章节“32.4.10 HALL 模式”中，补充说明通道 3~7 禁止使用的原因：“而通道 3~7 无法正常使用，因为 Hall 模式使能后，每次通道中断触发后会自动复位 CNT 计数器”；“使能 CH0SCR[CHRSTEN]和 CH1SCR[CHRSTEN]位”后面补充“（Hall 模式下硬件会自动使能它们，用户无需配置这两位）” 4. 章节“12.2.4” 增加 standby 唤醒 IO 复用功能表
3.0.2	2023-12-28	AutoChips	20.2 修正 DMA 特性描述
3.0.3	2024-01-22	AutoChips	新增 15.5.25，15.5.26 章节，补充 CSE 描述
3.0.4	2024-03-01	AutoChips	表 17.1，修正寄存器名称拼写错误
3.0.5	2024-03-25	AutoChips	10.4.18 修正寄存器 LD_DLY_SEL 名称拼写错误
3.0.6	2024-05-08	AutoChips	16.4.1.4 添加秘钥属性 bit 位说明
3.0.7	2024-06-14	AutoChips	16.4.4.1，16.4.4.2，16.4.4.3，16.4.4.4 更新描述
3.0.8	2024-06-25	AutoChips	19.4.4 纠正部分复用功能描述错误
3.0.9	2024-08-21	AutoChips	30.4.12 修正温度计算公式 22.4.5 补充注意事项说明 36.6.1 补充 RTC_CTRL 寄存器注意说明
3.0.10	2024-09-02	AutoChips	16.4.4.9 说明 CMD_LOAD_KEY 次数限制 16.5.2.1 更新授权密钥表
3.0.11	2024-10-31	AutoChips	28.6.6 SPI HREQ 极性寄存器描述修正
3.0.12	2025-04-03	AutoChips	12.2.4 更新表 12-1 PB0/PB1 描述
3.0.13	2025-04-23	AutoChips	12.2.1 补充说明 Standby 后不会保持待机前 IO 状态。

© 2013 - 2024 杰发科技有限公司

版权声明

本参考包含杰发科技的机密信息。禁止未经授权使用或披露本手册包含的信息。 对因未经杰发科技授权而全部或部分披露此文档内容而给杰发科技带来的任何损失或损害，杰发科技将追究责任。

杰发科技保留对此处任何信息进行更改的权利，此处的信息如有变更，恕不另行通知。杰发科技对使用或依赖此处包含的信息不承担任何责任。

本参考手册的所有信息均“按原样”提供，不提供任何形式的明示、暗示、法定或其他形式的保证。杰发科技明确拒绝对适销性，非侵权性和针对特定用途的适用性方面的的所有暗示保证。杰发科技对本手册可能使用、包含或提供的任何第三方软件不提供任何担保，并且用户同意仅向该等第三方寻求与此相关的任何担保索赔。杰发科技对于根据用户规格或为符合特定标准或公开论坛而产生的任何交付物，也不承担任何责任。

© 2013 - 2024 杰发科技有限公司

文档目录

修订信息 2
版权声明 4
文档目录 5
插图目录 37
表格目录 42
1 文中的约定（Conventions） 53
1.1 术语缩写 53
1.2 寄存器描述表中使用的缩写 54
1.3 数字表示方法 54
2 简介（Overview） 55
2.1 概要 55
2.2 模块概述 55
3 内核和系统（Core And System） 57
3.1 系统架构 57
3.2 嵌套中断向量控制器 (NVIC) 58
3.3 NMI 61
3.4 Systick 61
3.5 UUID 61
3.6 启动配置 61
4 存储器 62
4.1 存储器组织 62
4.2 存储器映射 62
4.3 内置 SRAM 63
4.4 片内 Flash 存储器 63
5 安全概览（Safety Overview） 64
5.1 简介 64
5.2 安全描述 64
5.2.1 内核自测(SCST) 64

© 2013 - 2024 杰发科技有限公司

5.2.2 访问保护 65
5.2.3 SMU 65
5.2.4 ECC 66
5.2.5 CRC 66
5.2.6 时钟监控 66
5.2.7 电源监控 66
5.2.8 看门狗 66
5.2.9 系统资源的多样性 67
6 杂项控制器(MCM) 68
6.1 简介 68
6.2 特性 68
6.3 功能描述 68
6.3.1 浮点运算单元 FPU 异常 68
6.3.2 MBIST 异常状态 68
6.3.3 Cache 69
6.4 寄存器定义 69
6.4.1 FPU 中断状态/控制寄存器(MCM_MISCR) 69
6.4.2 存储描述寄存器 0(MCM_MLMDR0) 71
6.4.3 MBIST 状态寄存器(MCM_MBIST) 72
6.4.4 缓存 ECC 错误状态寄存器(MCM_LMPEIR) 72
6.4.5 缓存 ECC 错误地址寄存器(MCM_LMFAR) 73
6.4.6 NMI 控制寄存器(MCM_MNCR) 73
7 外设桥（PBR） 75
7.1 简介 75
7.2 特性 75
7.3 结构框图 76
7.4 功能描述 76
7.4.1 主机非法访问 76
7.4.2 调试模式 76
7.5 应用说明 76

© 2013 - 2024 杰发科技有限公司

7.6 寄存器定义 77
7.6.1 Core 权限配置寄存器(PBR_MPR_CORE) 77
7.6.2 Debugger 权限配置寄存器(PBR_MPR_DEBUG) 78
7.6.3 DMA 权限配置寄存器(PBR_MPR_DMA) 79
7.6.4 外设访问权限配置寄存器 A(PBR_PACRA) 80
7.6.5 外设访问权限配置寄存器 B(PBR_PACRB) 82
7.6.6 外设访问权限配置寄存器 C(PBR_PACRC) 84
7.6.7 外设访问权限配置寄存器 D(PBR_PACRD) 87
7.6.8 外设访问权限配置寄存器 E(PBR_PACRE) 89
7.6.9 外设访问权限配置寄存器 F(PBR_PACRF) 91
7.6.10 外设访问权限配置寄存器 G(PBR_PACRG) 93
8 存储保护单元（MPU） 95
8.1 简介 95
8.2 特性 95
8.3 功能描述 95
8.3.1 MPU 主机/从机端口分配 95
8.3.2 命中认定 96
8.3.3 违规访问 97
8.3.4 Region 0 的配置限制 97
8.3.5 重叠区域(Overlapping) 98
8.3.6 权限配置复用寄存器 98
8.4 应用说明 98
8.4.1 编程指南 98
8.5 寄存器定义 99
8.5.1 控制/错误状态寄存器(MPU_CESR) 100
8.5.2 错误地址寄存器，Slave n(MPU_EAR0-MPU_EAR2) 101
8.5.3 错误信息寄存器，Slave n(MPU_EDR0-MPU_EDR2) 102
8.5.4 MASTER PID 配置寄存器(MPU_MPID) 103
8.5.5 起始地址寄存器(MPU_RGD0_WORD0-MPU_RGD7_WORD0) 103
8.5.6 结束地址寄存器(MPU_RGD0_WORD1) 104

© 2013 - 2024 杰发科技有限公司

8.5.7 结束地址寄存器(MPU_RGD1_WORD1-MPU_RGD7_WORD1) 104
8.5.8 权限配置寄存器(MPU_RGD0_WORD2) 105
8.5.9 权限配置寄存器(MPU_RGD1_WORD2-MPU_RGD7_WORD2) 107
8.5.10 RGD PID/使能配置寄存器(MPU_RGD0_WORD3) 109
8.5.11 RGD PID/使能配置寄存器(MPU_RGD1_WORD3-MPU_RGD7_WORD3) 110
8.5.12 权限配置复用寄存器(MPU_RGD0_AAC) 111
8.5.13 权限配置复用寄存器(MPU_RGD1_AAC-MPU_RGD7_AAC) 113
9 复位（RCM） 116
9.1 简介 116
9.2 特性 116
9.3 结构框图 116
9.4 功能描述 117
9.4.1 上电复位 (POR) 117
9.4.2 系统复位(System Reset) 118
9.4.3 复位范围 119
9.4.4 中断请求 120
9.4.5 调试模式 120
9.5 寄存器定义 121
9.5.1 复位控制寄存器(RCM_CTRL) 121
9.5.2 复位使能寄存器(RCM_EN) 123
9.5.3 复位状态寄存器(RCM_STATUS) 125
10 时钟（CKGEN） 128
10.1 简介 128
10.2 结构框图 128
10.3 功能描述 129
10.3.1 SPLL 功能 129
10.3.2 外设复位功能 130
10.3.3 外设时钟 131
10.3.4 调试模式 135
10.3.5 编程指南 135

© 2013 - 2024 杰发科技有限公司

10.4 寄存器定义 136
10.4.1 控制寄存器(CKGEN_CTRL) 137
10.4.2 低功耗时钟选择寄存器(CKGEN_LP_CLK_MUX) 139
10.4.3 外设时钟使能寄存器 0(CKGEN_PERI_CLK_EN0) 140
10.4.4 外设时钟使能寄存器 1(CKGEN_PERI_CLK_EN1) 142
10.4.5 外设时钟使能寄存器 2(CKGEN_PERI_CLK_EN2) 143
10.4.6 外设复位寄存器 0(CKGEN_PERI_SFT_RST0) 145
10.4.7 外设复位寄存器 1(CKGEN_PERI_SFT_RST1) 147
10.4.8 外设复位寄存器 2(CKGEN_PERI_SFT_RST2) 148
10.4.9 时钟分频控制寄存器 1(CKGEN_CLK_DIV1) 149
10.4.10 时钟分频控制寄存器 2(CKGEN_CLK_DIV2) 150
10.4.11 外设时钟选择寄存器 0(CKGEN_PERI_CLK_MUX0) 151
10.4.12 外设时钟选择寄存器 1(CKGEN_PERI_CLK_MUX1) 153
10.4.13 外设时钟选择寄存器 2(CKGEN_PERI_CLK_MUX2) 154
10.4.14 外设时钟选择寄存器 3(CKGEN_PERI_CLK_MUX3) 155
10.4.15 时钟输出配置寄存器(CKGEN_CLK_OUT_CFG) 157
10.4.16 外设时钟分频寄存器(CKGEN_PERI_CLK_DIV) 159
10.4.17 PLL 配置寄存器 0(ANA_SPLL_CFG0) 161
10.4.18 PLL 配置寄存器 1(ANA_SPLL_CFG1) 161
11 时钟监控单元（CMU） 163
11.1 简介 163
11.2 特性 163
11.3 结构框图 164
11.4 功能描述 165
11.4.1 被监控的时钟树 165
11.4.2 工作流程 166
11.4.3 中断请求 167
11.4.4 调试模式 167
11.5 应用说明 167
11.5.1 被监控的时钟丢失 167

© 2013 - 2024 杰发科技有限公司

11.5.2 窗口长度 168
11.5.3 阈值 168
11.5.4 编程指南 169
11.6 寄存器定义 169
11.6.1 配置寄存器 (CMU_CR) 169
11.6.2 参考计数配置寄存器(CMU_RCCR) 170
11.6.3 高阈值配置寄存器 (CMU_HTCR) 171
11.6.4 低阈值配置寄存器 (CMU_LTCR) 171
11.6.5 状态寄存器 (CMU_SR) 172
12 电源模式（Power Modes） 173
12.1 简介 173
12.2 功能描述 173
12.2.1 功耗模式 173
12.2.2 模式转换 173
12.2.3 休眠应答机制 174
12.2.4 唤醒 175
12.2.5 中断请求 176
12.3 应用说明 176
12.3.1 进入和退出停止模式 176
12.3.2 进入和退出待机模式 177
12.3.3 各模式下的模块功能 177
13 系统电源管理（SPM） 180
13.1 简介 180
13.2 特性 180
13.3 应用说明 180
13.3.1 晶体振荡器（XOSC） /系统时钟（SYSPLL） 电源控制 180
13.3.2 内部高速时钟（HSI） / 内部超高速时钟（VHSI） 电源控制 180
13.3.3 低电压检测(LVR/LVD) 181
13.4 寄存器定义 181
13.4.1 电源管理配置寄存器 0 (SPM_PWR_MGR_CFG0) 182

© 2013 - 2024 杰发科技有限公司

13.4.2 电源管理配置寄存器 1 (SPM_PWR_MGR_CFG1) 183
13.4.3 外设休眠应答状态寄存器 (SPM_PERIPH_SLEEP_ACK_STATUS) 185
13.4.4 外设休眠应答使能寄存器 (SPM_PERIPH_SLEEP_ACK_EN) 186
13.4.5 状态寄存器 (SPM_STATUS) 189
13.4.6 待机唤醒使能寄存器 (SPM_STB_WP_EN) 190
13.4.7 待机唤醒状态寄存器 (SPM_STB_WP_STATUS) 192
14 安全管理单元(SMU) 195
14.1 简介 195
14.2 特性 195
14.3 结构框图 196
14.4 功能描述 196
14.4.1 错误信号源 196
14.4.2 复位触发 197
14.4.3 硬件错误源列表 197
14.5 应用说明 199
14.6 寄存器定义 200
14.6.1 单点错误状态寄存器 (SMU_SFES) 201
14.6.2 单点错误状态影子寄存器 (SMU_SFESS) 201
14.6.3 潜在错误状态寄存器 (SMU_LFES) 202
14.6.4 潜在错误状态影子寄存器 (SMU_LFESS) 203
14.6.5 单点错误解锁时序寄存器 (SMU_LKSEQ0) 204
14.6.6 潜在错误解锁时序寄存器 (SMU_LKSEQ1) 204
14.6.7 软件单点错误使能寄存器 (SMU_SWSFE) 205
14.6.8 软件单点错误使能影子寄存器 (SMU_SWSFES) 206
14.6.9 软件潜在错误使能寄存器 (SMU_SWLFE) 206
14.6.10 软件潜在错误使能影子寄存器 (SMU_ SWLFES) 207
14.6.11 单点错误复位使能寄存器 (SMU_SFRSTEN) 207
14.6.12 潜在错误复位使能寄存器 (SMU_LFRSTEN) 208
14.6.13 系统复位请求计数阈值寄存器(SMU_SRSTCNTVAL) 209
14.6.14 系统复位请求计数寄存器(SMU_SRSTCNT) 209

© 2013 - 2024 杰发科技有限公司

14.6.15 系统复位请求计数影子寄存器(SMU_SRSTCNTS) 210
14.6.16 单点错误通道自检测寄存器(SMU_PATHCHK0) 210
14.6.17 潜在错误通道自检测寄存器(SMU_PATHCHK1) 211
15 片内 Flash（Embedded Flash） 212
15.1 简介 212
15.2 特性 212
15.3 结构框图 212
15.4 功能描述 213
15.4.1 P-Flash 组织 213
15.4.2 P-Flash 信息区组织 213
15.4.3 P-Flash 安全信息区 213
15.4.4 D-Flash 组织 214
15.4.5 Flash 命令 214
15.4.6 中断请求 215
15.4.7 调试模式 216
15.5 应用说明 216
15.5.1 页擦除 217
15.5.2 块擦除 217
15.5.3 写保护信息页擦除 218
15.5.4 全部块擦除 218
15.5.5 页编程 219
15.5.6 Section 编程 219
15.5.7 一次性编程 219
15.5.8 Section 验证 220
15.5.9 块验证 220
15.5.10 全部块验证 220
15.5.11 验证 Backdoor Key 221
15.5.12 擦除分区 221
15.5.13 编程分区 222
15.5.14 设置 FlexRAM 功能 223
15.5.15 Read while write(RWW) 224

© 2013 - 2024 杰发科技有限公司

15.5.16 ECC 224
15.5.17 写保护 225
15.5.18 失能写保护 226
15.5.19 读保护 226
15.5.20 失能读保护 226
15.5.21 使能 MBIST 227
15.5.22 失能 MBIST 227
15.5.23 使能 ABIST 227
15.5.24 失能 ABIST 228
15.5.25 设置 UID、Secret key 以及 Lock 228
15.5.26 擦除 UID 和 Secret KEY 228
15.6 寄存器定义 229
15.6.1 状态寄存器（FLASH_STAT） 229
15.6.2 安全寄存器（FLASH_SEC） 232
15.6.3 解锁寄存器（FLASH_KEYULK） 233
15.6.4 配置寄存器（FLASH_CNFG） 233
15.6.5 命令 ID 寄存器（FLASH_CMD） 235
15.6.6 地址寄存器（FLASH_ADDR） 235
15.6.7 低 32-bit 数据寄存器（FLASH_DATA0） 236
15.6.8 高 32-bit 数据寄存器（FLASH_DATA1） 236
15.6.9 长度寄存器（FLASH_LEN） 237
15.6.10 命令触发寄存器（FLASH_CST） 237
15.6.11 P-Flash 写保护寄存器（FLASH_PPROT） 238
15.6.12 D-Flash 寄存器（FLASH_DPROT） 238
15.6.13 ECC 错误地址寄存器（FLASH_DFADR） 239
15.6.14 分区信息寄存器（FLASH_PART） 240
16 硬件加密模块(CSE) 241
16.1 简介 241
16.2 特性 241
16.3 结构框图 242
16.4 功能描述 242

© 2013 - 2024 杰发科技有限公司

16.4.1 秘钥存储与管理 242
16.4.2 安全引导模式 244
16.4.3 PRAM 接口 245
16.4.4 CSE 命令 246
16.4.5 中断请求 270
16.5 应用说明 270
16.5.1 CSE 启用与关闭 270
16.5.2 秘钥管理 271
16.5.3 安全引导 272
16.5.4 数据通信安全保护 273
16.5.5 Flash 数据完整性保护 273
16.6 寄存器定义 275
16.6.1 CSE 状态寄存器（CSESTAT） 275
16.6.2 PRAM 寄存器 276
17 静态存储器（SRAM） 282
17.1 简介 282
17.2 特性 282
17.3 功能描述 283
17.3.1 ECC 错误状态 283
17.3.2 ECC 错误地址 283
17.3.3 ECC 错误处理 283
17.3.4 中断请求 283
17.3.5 调试模式 284
17.4 寄存器定义 284
17.4.1 SRAM_L ECC 错误状态和 1-bit 错误地址寄存器（CH0_STATUS0） 284
17.4.2 SRAM_L ECC 中断标志和 2-bit 错误地址寄存器（CH0_STATUS1） 285
17.4.3 SRAM_U ECC 错误状态和 1-bit 错误地址寄存器（CH1_STATUS0） 285
17.4.4 SRAM_U ECC 中断状态和 2-bit 错误地址寄存器（CH1_STATUS1） 286
17.4.5 SRAM ECC 中断使能和状态清除寄存器（ECC_ERR_CTRL） 287
18 错误注入模块(EIM) 289

© 2013 - 2024 杰发科技有限公司

18.1 简介 289
18.2 特性 289
18.3 结构框图 289
18.4 功能描述 290
18.4.1 错误注入类型 290
18.4.2 双重使能机制 290
18.4.3 调试模式 290
18.5 应用说明 290
18.5.1 错误注入配置流程 290
18.5.2 错误注入关闭流程 291
18.6 寄存器定义 291
18.6.1 EIM 全局使能控制寄存器（EIM_GLB_EN） 292
18.6.2 EIM 通道使能控制寄存器（EIM_CH_EN） 292
18.6.3 SRAM_L 数据错误注入寄存器（EIM0_INJ_DATA） 293
18.6.4 SRAM_L 地址错误注入寄存器（EIM0_INJ_ADDR） 293
18.6.5 SRAM_L 校验码错误注入寄存器（EIM0_INJ_ECC） 294
18.6.6 SRAM_U 数据错误注入寄存器（EIM1_INJ_DATA） 294
18.6.7 SRAM_U 地址错误注入寄存器（EIM1_INJ_ADDR） 295
18.6.8 SRAM_U 校验码错误注入寄存器（EIM1_INJ_ECC） 295
19 通用输入/输出（GPIO） 297
19.1 简介 297
19.2 特性 297
19.3 结构框图 298
19.4 功能描述 299
19.4.1 外部中断 299
19.4.2 DMA 请求 299
19.4.3 数字滤波 299
19.4.4 复用功能 300
19.4.5 中断请求 308
19.4.6 低功耗模式 308

© 2013 - 2024 杰发科技有限公司

19.4.7 调试模式 309
19.5 应用说明 309
19.5.1 开漏输出 309
19.5.2 编程指南 309
19.6 寄存器定义 310
19.6.1 端口配置寄存器(PORT_PCR) 311
19.6.2 端口中断状态标志寄存器 (PORT_ISFR) 313
19.6.3 端口数字滤波使能寄存器(PORT_DFER) 313
19.6.4 端口数字滤波时钟寄存器(PORT_DFCR) 314
19.6.5 端口数字滤波宽度寄存器(PORT_DFWR) 314
19.6.6 端口输出数据寄存器(GPIO_PODR) 315
19.6.7 端口输出置位寄存器(GPIO_PSOR) 316
19.6.8 端口输出复位寄存器(GPIO_PROR) 316
19.6.9 端口输出翻转寄存器(GPIO_PIOR) 317
19.6.10 端口输入数据寄存器(GPIO_PIDR) 317
19.6.11 端口输出使能寄存器(GPIO_POER) 318
19.6.12 端口输入使能寄存器(GPIO_PIER) 319
20 直接存储器访问（DMA） 320
20.1 简介 320
20.2 特性 320
20.3 结构框图 321
20.4 功能描述 321
20.4.1 DMA 请求列表 321
20.4.2 仲裁器 323
20.4.3 中断请求 323
20.4.4 低功耗模式 323
20.4.5 调试模式 324
20.5 应用说明 324
20.5.1 软复位和硬复位 325
20.5.2 暂停和恢复 325

© 2013 - 2024 杰发科技有限公司

20.5.3 通道循环 325
20.5.4 I2C 使用 DMA 326
20.5.5 DMA 触发模式 326
20.5.6 配置指南 326
20.6 寄存器定义 327
20.6.1 通用 DMA 复位寄存器(DMA_TOP_RST) 328
20.6.2 通道状态寄存器(DMA_STATUS) 329
20.6.3 通道中断使能寄存器(DMA_INTEN) 330
20.6.4 通道复位寄存器(DMA_RST) 331
20.6.5 通道停止寄存器(DMA_STOP) 332
20.6.6 通道配置寄存器(DMA_CONFIG) 333
20.6.7 通道长度寄存器(DMA_CHAN_LENGTH) 334
20.6.8 通道源起始地址寄存器(DMA_SSTART_ADDR) 334
20.6.9 通道源结束地址寄存器(DMA_SEND_ADDR) 335
20.6.10 通道目标起始地址寄存器(DMA_DSTART_ADDR) 335
20.6.11 通道使能寄存器(DMA_CHAN_ENABLE) 336
20.6.12 通道数据传输数目寄存器 (DMA_DATA_TRANS_NUM) 337
20.6.13 通道 FIFO 数据剩余数目寄存器(DMA_INTER_FIFO_DATA_LEFT_NUM) 337
20.6.14 通道目标结束地址寄存器(DMA_DEND_ADDR) 338
20.6.15 通道地址偏移寄存器(DMA_ADDR_OFFSET) 338
20.6.16 通道复用器配置寄存器(DMA_MUX_CFG) 339
21 循环冗余校验模块（CRC） 340
21.1 简介 340
21.2 特性 340
21.3 结构框图 340
21.4 功能描述 340
21.4.1 转置特性 340
21.4.2 转置类型 341
21.4.3 结果反码 342
21.4.4 支持 DMA 342
21.4.5 低功耗模式 342

© 2013 - 2024 杰发科技有限公司

21.4.6 调试模式 342
21.5 应用说明 343
21.5.1 CRC 初始化 343
21.5.2 CRC 多项式配置 343
21.5.3 CRC 校验 343
21.5.4 编程指南 343
21.6 寄存器定义 344
21.6.1 数据寄存器(CRC_DATA) 344
21.6.2 多项式寄存器(CRC_POLY) 345
21.6.3 控制寄存器(CRC_CTRL) 346
22 看门狗模块（WDG） 348
22.1 简介 348
22.2 特性 348
22.3 结构框图 348
22.4 功能描述 349
22.4.1 基本看门狗 349
22.4.2 看门狗默认超时行为 349
22.4.3 窗口看门狗 349
22.4.4 备份复位 349
22.4.5 快速测试模式 349
22.4.6 中断请求 350
22.4.7 低功耗模式 350
22.4.8 调试模式 351
22.5 应用说明 351
22.5.1 配置看门狗 351
22.5.2 刷新看门狗 351
22.5.3 看门狗中断 352
22.5.4 编程指南 352
22.6 寄存器定义 352
22.6.1 配置寄存器 0(WDG_CS0) 353

© 2013 - 2024 杰发科技有限公司

22.6.2 配置寄存器 1(WDG_CS1) 354
22.6.3 计数器寄存器(WDG_CNT) 355
22.6.4 超时值寄存器(WDG_TOVAL) 356
22.6.5 窗口寄存器(WDG_WIN) 356
23 外部看门狗（EWDG） 357
23.1 简介 357
23.2 特性 357
23.3 结构框图 357
23.4 功能描述 358
23.4.1 基本功能 358
23.4.2 刷新操作 359
23.4.3 EWDG_in 359
23.4.4 Write-Once 359
23.4.5 无效配置 360
23.4.6 中断请求 360
23.4.7 低功耗模式 360
23.4.8 调试模式 360
23.5 应用说明 360
23.5.1 编程指南 360
23.6 寄存器定义 361
23.6.1 控制寄存器（EWDG_CTRL） 361
23.6.2 计数器寄存器（EWDG_SERV） 362
23.6.3 CMPL 寄存器（EWDG_CMPL） 363
23.6.4 CMPH 寄存器（EWDG_CMPH） 363
23.6.5 预分频器寄存器（EWDG_CLKPRESCALER） 364
24 连接单元（CTU） 365
24.1 简介 365
24.2 特性 365
24.3 结构框图 366
24.4 功能描述 366

© 2013 - 2024 杰发科技有限公司

24.4.1 TRGMUX 触发源存储器映射表 366
24.4.2 TRGMUX 触发目标选择 367
24.4.3 ADC 硬件触发 368
24.4.4 PWM 触发 369
24.4.5 PWM 同步 371
24.4.6 RTC 时钟捕获 371
24.4.7 ACMP 输出捕获 372
24.4.8 软件触发 372
24.4.9 寄存器锁定 372
24.4.10 中断请求 372
24.4.11 低功耗模式 373
24.4.12 调试模式 373
24.5 应用说明 373
24.5.1 UART 调制 373
24.5.2 ACMP HALL 测量 374
24.5.3 PWM 调制 375
24.6 寄存器定义 375
24.6.1 TRGMUX DMA 配置寄存器(TRGMUX_DMA) 376
24.6.2 TRGMUX EXTOUT0 配置寄存器(TRGMUX_EXTOUT0) 377
24.6.3 TRGMUX EXTOUT1 配置寄存器(TRGMUX_EXTOUT1) 378
24.6.4 TRGMUX ADC0 规则组配置寄存器(TRGMUX_ADC0_REG) 379
24.6.5 TRGMUX ADC0 注入组配置寄存器(TRGMUX_ADC0_INJ) 380
24.6.6 TRGMUX ADC1 规则组配置寄存器(TRGMUX_ADC1_REG) 381
24.6.7 TRGMUX ADC1 注入组配置寄存器(TRGMUX_ADC1_INJ) 382
24.6.8 TRGMUX ACMP 配置寄存器(TRGMUX_ACMP) 383
24.6.9 TRGMUX PWM0 配置寄存器(TRGMUX_PWM0) 383
24.6.10 TRGMUX PWM1 配置寄存器(TRGMUX_PWM1) 384
24.6.11 TRGMUX PWM2 配置寄存器(TRGMUX_PWM2) 385
24.6.12 TRGMUX PWM3 配置寄存器(TRGMUX_PWM3) 386
24.6.13 TRGMUX PWM4 配置寄存器(TRGMUX_PWM4) 387
24.6.14 TRGMUX PWM5 配置寄存器(TRGMUX_PWM5) 388

© 2013 - 2024 杰发科技有限公司

24.6.15 TRGMUX TIMER 配置寄存器(TRGMUX_TIMER) 388
24.6.16 TRGMUX PCT 配置寄存器(TRGMUX_PCT) 389
24.6.17 TRGMUX UART0 配置寄存器(TRGMUX_UART0) 390
24.6.18 TRGMUX UART1 配置寄存器(TRGMUX_UART1) 390
24.6.19 TRGMUX PDT0 配置寄存器(TRGMUX_PDT0) 391
24.6.20 TRGMUX PDT1 配置寄存器(TRGMUX_PDT1) 392
24.6.21 TRGMUX EIO 配置寄存器(TRGMUX_EIO) 392
24.6.22 CTU 配置寄存器(CTU_CFG) 394
24.6.23 CTU 软件触发源控制寄存器(CTU_SW) 397
24.6.24 PWM 调制配置寄存器(PWM_MODULATION) 398
24.6.25 ADC 同步模式控制寄存器 (ADC_SYNC) 400
25 通用异步收发器（UART） 401
25.1 简介 401
25.2 特性 401
25.3 结构框图 403
25.4 功能描述 403
25.4.1 输入输出定时 404
25.4.2 噪声检测（Noise Detection） 404
25.4.3 波特率描述 405
25.4.4 单线半双工功能 406
25.4.5 硬件流控制功能 407
25.4.6 RS485 功能 407
25.4.7 LIN 功能 408
25.4.8 IRDA 功能 410
25.4.9 数据、地址匹配功能 411
25.4.10 中断请求 411
25.4.11 低功耗模式 412
25.4.12 调试模式 413
25.5 应用说明 413
25.5.1 波特率配置说明 413
25.5.2 UART 配置说明 414

© 2013 - 2024 杰发科技有限公司

25.6 寄存器定义 415
25.6.1 RX/TX 数据寄存器(UART_RBR/THR) 416
25.6.2 分频器低 8 位寄存器(UART_DIV_L) 417
25.6.3 分频器高 8 位寄存器(UART_DIV_H) 417
25.6.4 控制寄存器 0(UART_LCR0) 418
25.6.5 控制寄存器 1(UART_LCR1) 419
25.6.6 FIFO 控制寄存器(UART_FCR) 420
25.6.7 硬件流使能寄存器(UART_EFR) 421
25.6.8 中断使能寄存器(UART_IER) 422
25.6.9 线路状态寄存器 0(UART_LSR0) 423
25.6.10 线路状态寄存器 1(UART_LSR1) 425
25.6.11 采样计数器寄存器(UART_SMP_CNT) 427
25.6.12 UART 地址匹配寄存器(UART_ADDR) 427
25.6.13 UART 数据匹配寄存器(UART_DATA) 428
25.6.14 保护时间寄存器(UART_RS485_GUARD) 429
25.6.15 休眠使能寄存器(UART_SLEEP_EN) 430
25.6.16 DMA 使能寄存器(UART_DMA_EN) 430
25.6.17 小数分频器寄存器(UART_DIV_FRAC) 431
25.6.18 UART 匹配控制寄存器(UART_MTCHCR) 431
25.6.19 RS485 控制寄存器(UART_RS485CR) 432
25.6.20 RS485 延迟时间寄存器(UART_ CNTR) 433
25.6.21 空闲中断使能寄存器(UART_IDLE) 433
25.6.22 LIN 控制寄存器 (UART_LINCR) 434
25.6.23 LIN 同步间隔段控制寄存器(UART_BRKLGH) 435
25.6.24 UART 接收最小脉宽(UART_PMIN0) 436
25.6.25 UART 接收最小脉宽(UART_PMIN1) 436
25.6.26 UART 传输最小脉宽(UART_PWDTH) 437
26 控制器局域网（CAN） 438
26.1 简介 438
26.1.1 CAN 控制器 438
26.1.2 CAN 协议 438

© 2013 - 2024 杰发科技有限公司

26.1.3 型号选择 438
26.2 特性 439
26.3 结构框图 440
26.4 功能描述 441
26.4.1 消息缓冲区 441
26.4.2 总线关闭状态 445
26.4.3 接收过滤器 445
26.4.4 消息接收 447
26.4.5 处理消息接收 447
26.4.6 DMA 接收 447
26.4.7 消息发送 448
26.4.8 消息发送中止 448
26.4.9 STB 满 449
26.4.10 错误处理 450
26.4.11 扩展功能 451
26.4.12 时间戳 454
26.4.13 ECC 保护 455
26.4.14 中断请求 455
26.4.15 低功耗模式 456
26.4.16 调试模式 456
26.5 应用说明 457
26.5.1 CAN 位时间 457
26.5.2 软件复位 460
26.6 寄存器定义 463
26.6.1 控制寄存器 0(CAN_CTRL0) 464
26.6.2 控制寄存器 1(CAN_CTRL1) 469
26.6.3 低速波特率配置寄存器(CAN_SBITRATE) 472
26.6.4 快速波特率配置寄存器 (CAN_FBITRATE) 473
26.6.5 错误状态寄存器(CAN_ERRINFO) 474
26.6.6 接收过滤器控制寄存器 0(CAN_ACFCTRL0) 475
26.6.7 接收过滤器控制寄存器 1(CAN_ACFCTRL1) 476

© 2013 - 2024 杰发科技有限公司

26.6.8 接收过滤器控制寄存器 2(CAN_ACFCTRL2) 477
26.6.9 过滤器代码 ACODE 寄存器(CAN_ACF) 478
26.6.10 过滤器掩码 AMASK 寄存器(CAN_ACF) 478
26.6.11 版本及存储保护寄存器(CAN_VERMEM) 479
26.6.12 存储错误注入寄存器(CAN_MEMES) 481
26.6.13 唤醒配置寄存器(CAN_WAKEUP) 482
27 集成电路总线（I2C） 483
27.1 简介 483
27.2 特性 483
27.3 结构框图 484
27.3.1 I2C 信号组成 484
27.3.2 波特率组成 485
27.3.3 数据流程 486
27.4 功能描述 486
27.4.1 主机模式 486
27.4.2 从机模式 487
27.4.3 DMA 发送/接收 488
27.4.4 SMBus 490
27.4.5 中断请求 490
27.4.6 低功耗模式 491
27.4.7 调试模式 491
27.5 应用说明 491
27.5.1 数据传输 491
27.5.2 应答控制 493
27.5.3 编程指南 493
27.6 寄存器定义 494
27.6.1 地址寄存器 0（I2C_ADDR0） 495
27.6.2 地址寄存器 1（I2C_ADDR1） 495
27.6.3 波特率配置寄存器 0（I2C_SAMPLE_CNT） 496
27.6.4 波特率配置寄存器 1（I2C_STEP_CNT） 497

© 2013 - 2024 杰发科技有限公司

27.6.5 控制寄存器 0（I2C_CTRL0） 497
27.6.6 控制寄存器 1（I2C_CTRL1） 499
27.6.7 控制寄存器 2（I2C_CTRL2） 500
27.6.8 控制寄存器 3（I2C_CTRL3） 502
27.6.9 状态寄存器 0（I2C_STATUS0） 1
27.6.10 状态寄存器 1（I2C_STATUS1） 1
27.6.11 毛刺滤波配置寄存器（I2C_DGLCFG） 1
27.6.12 数据寄存器（I2C_DATA） 1
27.6.13 起始与停止信号控制寄存器（I2C_STARTSTOP） 1
28 串行外设接口（SPI） 1
28.1 简介 1
28.2 特性 1
28.3 结构框图 1
28.4 功能描述 1
28.4.1 数据流 & 算法 1
28.4.2 输入输出时序 1
28.4.3 主机 SCK 输出时序设置 1
28.4.4 主机模式故障检测 1
28.4.5 管脚数据方向配置 1
28.4.6 SPI 接收数据匹配 1
28.4.7 中断请求 1
28.4.8 低功耗模式 1
28.4.9 调试模式 1
28.5 应用说明 1
28.5.1 主机 CS 连续模式 1
28.5.2 主机 CS 非连续输出模式 1
28.5.3 主机 2/4 bit 传输模式 1
28.5.4 从机模式 1
28.5.5 DMA 模式 1
28.6 寄存器定义 1

© 2013 - 2024 杰发科技有限公司

28.6.1 配置寄存器 0(SPI_CFG0) 1
28.6.2 配置寄存器 1(SPI_CFG1) 1
28.6.3 命令寄存器 (SPI_CMD) 1
28.6.4 状态寄存器(SPI_STATUS) 1
28.6.5 数据寄存器(SPI_DATA) 1
28.6.6 配置寄存器 2(SPI_CFG2) 1
28.6.7 数据匹配寄存器(SPI_DMV) 1
29 增强型 IO(EIO) 1
29.1 简介 1
29.2 特性 1
29.3 结构框图 1
29.4 功能描述 1
29.4.1 功能时钟 1
29.4.2 软件复位 1
29.4.3 移位操作 1
29.4.4 定时器操作 1
29.4.5 引脚操作 1
29.4.6 支持中断与 DMA 1
29.4.7 外设触发 1
29.4.8 输出触发信号 1
29.4.9 中断请求 1
29.4.10 低功耗模式 1
29.4.11 调试模式 1
29.5 应用说明 1
29.5.1 UART 发送 1
29.5.2 UART 接收 1
29.5.3 SPI 主模式 1
29.5.4 SPI 从模式 1
29.5.5 I2C 主模式 1
29.5.6 I2S 主模式 1
29.5.7 I2S 从模式 1

© 2013 - 2024 杰发科技有限公司

29.6 寄存器定义 1
29.6.1 控制寄存器(EIO_CTRL) 1
29.6.2 引脚寄存器(EIO_PIN) 1
29.6.3 移位状态寄存器(EIO_SHIFTSTAT) 1
29.6.4 移位错误状态寄存器(EIO_SHIFTERR) 1
29.6.5 定时器状态寄存器(EIO_TIMSTAT) 1
29.6.6 移位状态中断使能寄存器(EIO_SHIFTSIEN) 1
29.6.7 移位错误状态中断使能寄存器(EIO_SHIFTEIEN) 1
29.6.8 定时器中断使能寄存器(EIO_TIMIEN) 1
29.6.9 移位状态 DMA 使能寄存器(EIO_SHIFTSDEN) 1
29.6.10 移位控制寄存器 n(EIO_SHIFTCTL0 – EIO_SHIFTCTL3) 1
29.6.11 移位配置寄存器 n(EIO_SHIFTCFG0 – EIO_SHIFTCFG3) 1
29.6.12 移位缓冲寄存器 n(EIO_SHIFTBUF0 – EIO_SHIFTBUF3) 1
29.6.13 移位缓冲位交换寄存器 n(EIO_SHIFTBUFBIS0 – EIO_SHIFTBUFBIS3) 1
29.6.14 移位缓冲字节交换寄存器 n(EIO_SHIFTBUFBYS0 – EIO_SHIFTBUFBYS3) .1
29.6.15 移位缓冲位字节交换寄存器 n(EIO_SHIFTBUFBBS0 – EIO_SHIFTBUFBBS3) 1
29.6.16 定时控制寄存器 n(EIO_TIMCTL0 – EIO_TIMCTL3) 1
29.6.17 定时器配置寄存器 n(EIO_TIMCFG0 – EIO_TIMCFG3) 1
29.6.18 定时器比较值寄存器 n(EIO_TIMCMP0 – EIO_TIMCMP3) 1
30 模数转换器（ADC） 1
30.1 简介 1
30.2 特性 1
30.3 结构框图 1
30.4 功能描述 1
30.4.1 上电时序 1
30.4.2 工作模式 1
30.4.3 触发方式 1
30.4.4 模拟监控器 1
30.4.5 校准 1
30.4.6 数据对齐方式 1

© 2013 - 2024 杰发科技有限公司

30.4.7 采样转换时间 1
30.4.8 同步触发模式 1
30.4.9 交织模式 1
30.4.10 硬件均值功能 1
30.4.11 注入组数据偏置功能 1
30.4.12 温度传感器 1
30.4.13 中断请求 1
30.4.14 低功耗模式 1
30.4.15 调试模式 1
30.5 应用说明 1
30.5.1 重置和使能 1
30.5.2 ADC 上电延时 1
30.5.3 DMA 访问 1
30.6 寄存器定义 1
30.6.1 状态寄存器(ADC_STR) 1
30.6.2 控制寄存器 0(ADC_CTRL0) 1
30.6.3 控制寄存器 1(ADC_CTRL1) 1
30.6.4 注入组偏移寄存器(ADC_IOFRx) 1
30.6.5 高阈值寄存器(ADC_AMOHR) 1
30.6.6 低阈值寄存器(ADC_AMOLR) 1
30.6.7 采样时间寄存器 0(ADC_SPT0) 1
30.6.8 采样时间寄存器 1(ADC_SPT1) 1
30.6.9 采样时间寄存器 2(ADC_SPT2) 1
30.6.10 采样时间寄存器 3(ADC_SPT3) 1
30.6.11 规则组序列配置寄存器 (ADC_RSQRx) 1
30.6.12 注入组序列配置寄存器(ADC_ISQRx) 1
30.6.13 序列长度寄存器(ADC_SQL) 1
30.6.14 校准寄存器 0(ADC_CALI0) 1
30.6.15 校准寄存器 1(ADC_CALI1) 1
30.6.16 规则组数据寄存器(ADC_RDRx) 1
30.6.17 注入组数据寄存器(ADC_IDRx) 1

© 2013 - 2024 杰发科技有限公司

30.6.18 ADC analog 配置寄存器 0 (AUXADC_CFG0) 1
30.6.19 ADC analog 配置寄存器 1 (AUXADC_CFG1) 1
30.6.20 ADC analog 配置寄存器 4 (AUXADC_CFG4) 1
30.6.21 ADC analog 配置寄存器 5(AUXADC_CFG5) 1
31 模拟比较器（ACMP） 1
31.1 简介 1
31.2 特性 1
31.3 结构框图 1
31.4 功能描述 1
31.4.1 普通模式 1
31.4.2 轮询模式 1
31.4.3 霍尔输出 1
31.4.4 迟滞 1
31.4.5 窗口控制模式 1
31.4.6 中断请求 1
31.4.7 低功耗模式 1
31.4.8 调试模式 1
31.5 应用说明 1
31.5.1 轮询模式下霍尔输出 1
31.5.2 数字滤波器 1
31.6 寄存器定义 1
31.6.1 配置寄存器 0(ACMP_CR0) 1
31.6.2 配置寄存器 1(ACMP_CR1) 1
31.6.3 配置寄存器 2(ACMP_CR2) 1
31.6.4 配置寄存器 3(ACMP_CR3) 1
31.6.5 配置寄存器 4(ACMP_CR4) 1
31.6.6 数据输出寄存器(ACMP_DR) 1
31.6.7 状态寄存器(ACMP_SR) 1
31.6.8 轮询分频器寄存器(ACMP_FD) 1
31.6.9 霍尔输出 A 设置寄存器(ACMP_OPA) 1

© 2013 - 2024 杰发科技有限公司

31.6.10 霍尔输出 B 设置寄存器(ACMP_OPB) 1
31.6.11 霍尔输出 C 设置寄存器(ACMP_OPC) 1
31.6.12 数字滤波器时钟配置寄存器(ACMP_CLK) 1
31.6.13 ACMP analog 配置寄存器(ACMPDAC_CFG) 1
32 脉宽调制（PWM） 1
32.1 简介 1
32.2 特性 1
32.3 结构框图 1
32.4 功能描述 1
32.4.1 时钟源 1
32.4.2 计数器 1
32.4.3 工作模式 1
32.4.4 输入捕获模式 1
32.4.5 输出比较模式 1
32.4.6 边沿对齐 PWM 模式(EPWM) 1
32.4.7 中心对齐 PWM 模式(CPWM) 1
32.4.8 组合模式 1
32.4.9 双边沿捕获模式 1
32.4.10 HALL 模式 1
32.4.11 正交解码模式 1
32.4.12 写保护 1
32.4.13 初始化 1
32.4.14 极性控制 1
32.4.15 输出屏蔽 1
32.4.16 软件输出控制 1
32.4.17 初始化触发器 1
32.4.18 通道匹配触发器 1
32.4.19 最大值触发器 1
32.4.20 故障控制 1
32.4.21 PWM 同步 1
32.4.22 特性优先级 1

© 2013 - 2024 杰发科技有限公司

32.4.23 PWM 抖动 1
32.4.24 DMA 功能 1
32.4.25 全局时基 1
32.4.26 中断请求 1
32.4.27 低功耗模式 1
32.4.28 调试模式 1
32.5 寄存器定义 1
32.5.1 初始化寄存器(PWM_INIT) 1
32.5.2 计数器寄存器(PWM_CNT) 1
32.5.3 最大计数值寄存器(PWM_MCVR) 1
32.5.4 通道状态和控制寄存器(PWM_CHnSCR) 1
32.5.5 通道值寄存器(PWM_CHnV) 1
32.5.6 计数器初始值寄存器(PWM_CNTIN) 1
32.5.7 捕获和比较状态寄存器(PWM_STR) 1
32.5.8 功能选择寄存器(PWM_FUNCSEL) 1
32.5.9 同步寄存器(PWM_SYNC) 1
32.5.10 通道输出初始状态寄存器(PWM_OUTINIT) 1
32.5.11 输出屏蔽控制寄存器(PWM_OMCR) 1
32.5.12 模式选择寄存器(PWM_MODESEL) 1
32.5.13 死区插入控制寄存器(PWM_DTSET0) 1
32.5.14 死区插入控制寄存器(PWM_DTSET1) 1
32.5.15 外部触发器寄存器(PWM_EXTTRIG) 1
32.5.16 通道输出极性控制寄存器(PWM_CHOPOLCR) 1
32.5.17 故障检测状态寄存器(PWM_FDSR) 1
32.5.18 输入捕获滤波器控制(PWM_CAPFILTER) 1
32.5.19 故障滤波器和故障使能寄存器(PWM_FFAFER) 1
32.5.20 正交解码器接口配置寄存器(PWM_QDI) 1
32.5.21 配置寄存器(PWM_CONF) 1
32.5.22 故障输入极性寄存器(PWM_FLTPOL) 1
32.5.23 同步配置寄存器(PWM_SYNCONF) 1
32.5.24 反相控制寄存器(PWM_INVCR) 1

© 2013 - 2024 杰发科技有限公司

32.5.25 通道软件输出控制寄存器(PWM_CHOSWCR) 1
32.5.26 通道抖动控制寄存器 0(PWM_DITHER0) 1
32.5.27 通道抖动控制寄存器 1(PWM_DITHER1) 1
32.5.28 通道抖动控制寄存器 2(PWM_DITHER2) 1
32.5.29 PWM DMA 控制寄存器(PWM_DMACTRL) 1
33 可编程延时定时器(PDT) 1
33.1 简介 1
33.2 特性 1
33.3 结构框图 1
33.4 功能描述 1
33.4.1 计数器时钟 1
33.4.2 计数器 1
33.4.3 模数值 1
33.4.4 可编程延时 1
33.4.5 触发模式 1
33.4.6 工作模式 1
33.4.7 脉冲输出 1
33.4.8 寄存器更新 1
33.4.9 中断请求 1
33.4.10 低功耗模式 1
33.4.11 调试模式 1
33.5 应用说明 1
33.5.1 ADC 定时采样控制功能编程指南 1
33.6 寄存器定义 1
33.6.1 状态控制寄存器(PDT_SC) 1
33.6.2 模数值寄存器(PDT_MOD) 1
33.6.3 触发延时寄存器 0(PDT_DLY0) 1
33.6.4 触发延时寄存器 1(PDT_DLY1) 1
33.6.5 触发延时寄存器 2(PDT_DLY2) 1
33.6.6 触发延时寄存器 3(PDT_DLY3) 1

© 2013 - 2024 杰发科技有限公司

33.6.7 触发延时寄存器 4(PDT_DLY4) 1
33.6.8 触发延时寄存器 5(PDT_DLY5) 1
33.6.9 触发延时寄存器 6(PDT_DLY6) 1
33.6.10 触发延时寄存器 7(PDT_DLY7) 1
33.6.11 中断触发延时寄存器(PDT_IDLY) 1
33.6.12 脉冲输出延时寄存器(PDT_PODLY) 1
33.6.13 触发使能寄存器(PDT_DLYn_EN) 1
33.6.14 脉冲输出使能寄存器(PDT_POEN) 1
33.6.15 计数器寄存器(PDT_CNT) 1
34 周期性中断定时器（Timer） 1
34.1 简介 1
34.2 结构框图 1
34.3 特性 1
34.4 功能描述 1
34.4.1 时钟源 1
34.4.2 工作模式 1
34.4.3 触发输入选择 1
34.4.4 超时值设置 1
34.4.5 实时计数值可获取可清零 1
34.4.6 触发输出 1
34.4.7 控制位 1
34.4.8 链模式 1
34.4.9 中断请求 1
34.4.10 低功耗模式 1
34.4.11 调试模式 1
34.5 应用说明 1
34.5.1 编程指南 1
34.5.2 详细时序图 1
34.6 寄存器定义 1
34.6.1 模块控制寄存器(TIMER_CR) 1

© 2013 - 2024 杰发科技有限公司

34.6.2 中断标志寄存器 (TIMER_SR) 1
34.6.3 中断使能寄存器(TIMER_IER) 1
34.6.4 通道使能寄存器(TIMER_ENR) 1
34.6.5 装载值寄存器(TIMER_CHx_TVAL) 1
34.6.6 当前值寄存器(TIMER_CHx_CVAL) 1
34.6.7 通道控制寄存器(TIMER_CHx_CTRL) 1
35 脉冲计数定时器（PCT） 1
35.1 简介 1
35.2 特性 1
35.3 结构框图 1
35.4 功能描述 1
35.4.1 时钟源 1
35.4.2 工作模式 1
35.4.3 触发/脉冲源选择 1
35.4.4 触发/脉冲极性配置 1
35.4.5 超时值设置 1
35.4.6 Free-running 模式 1
35.4.7 分频/滤波功能 1
35.4.8 实时计数值可获取 1
35.4.9 触发输出 1
35.4.10 中断请求 1
35.4.11 低功耗模式 1
35.4.12 调试模式 1
35.5 应用说明 1
35.5.1 时钟计数器配置流程 1
35.5.2 脉冲计数器配置流程 1
35.6 寄存器定义 1
35.6.1 控制寄存器(PCT_CSR) 1
35.6.2 分频滤波设置寄存器(PCT_PSR) 1
35.6.3 超时值寄存器(PCT_CMR) 1

© 2013 - 2024 杰发科技有限公司

35.6.4 当前值寄存器 (PCT_CNR) 1
36 实时计数器（RTC） 1
36.1 简介 1
36.2 特性 1
36.3 结构框图 1
36.4 功能描述 1
36.4.1 时钟源选择 1
36.4.2 软件复位 1
36.4.3 计时特性 1
36.4.4 闹钟事件 1
36.4.5 寄存器写保护 1
36.4.6 RTC 时钟输出 1
36.4.7 中断请求 1
36.4.8 低功耗模式 1
36.4.9 调试模式 1
36.5 应用说明 1
36.5.1 RTC 基本使用 1
36.5.2 RTC 低功耗唤醒 1
36.5.3 编程指南 1
36.6 寄存器定义 1
36.6.1 控制寄存器（RTC_CTRL） 1
36.6.2 闹钟寄存器（RTC_TAR） 1
36.6.3 实时计数寄存器（RTC_TC） 1
36.6.4 预分频器寄存器（RTC_PSR） 1
36.6.5 预分频计数寄存器（RTC_PSC） 1
36.6.6 状态寄存器（RTC_SR） 1
36.6.7 锁定寄存器（RTC_LR） 1
37 调试(Debug) 1
37.1 简介 1
37.2 特性 1

© 2013 - 2024 杰发科技有限公司

37.3 功能描述 1
37.3.1 调试安全性 1
37.3.2 调试复位 1
37.3.3 低功耗调试 1

© 2013 - 2024 杰发科技有限公司

插图目录

图 3-1 系统结构框图 57
图 5-1 功能安全结构框图 64
图 7-1 外设桥结构示意图 76
图 8-1 重叠区域权限示意图 98
图 8-2 MPU 编程指南 99
图 9-1 RCM 模块结构框图 117
图 10-1 时钟控制结构 128
图 10-2 系统时钟示意图 129
图 11-1 CMU 模块结构框图 164
图 11-2 被监控的时钟树 165
图 11-3 CMU 工作流程 166
图 12-1 模式转换图 174
图 14-1 SMU 模块结构框图 196
图 14-2 复位触发框图 197
图 14-3 SMU 复位触发软件配置流程 199
图 15-1 Flash 结构框图 212
图 15-2 操作流程 217
图 15-3 CSE 分区组织图 223
图 16-1 CSE 模块框图 242
图 16-2 数据通信安全保护 273
图 16-3 Flash 数据完整性保护 274
图 18-1 EIM 结构框图 289
图 19-1 GPIO 结构框图 298
图 19-2 GPIO 复用功能 300
图 20-1 DMA 结构框图 321
图 20-2 DMA 配置指南 324
图 20-3 DMA 通道循环 325
图 20-4 DMA 通道触发：正常操作 326
图 20-5 DMA 通道触发：忽略触发 326
图 21-1 CRC 结构框图 340
图 21-2 [TOTW] 或 [TOTR]为 01 341
图 21-3 [TOTW] 或 [TOTR]为 10 341
图 21-4 [TOTW] 或 [TOTR]为 11 342
图 22-1 WDG 结构框图 348
图 23-1 EWDG 结构框图 357
图 23-2 计数器小于 CMPL 时刷新 358
图 23-3 计数器达到了 CMPH 358
图 23-4 EWDG_in 有效时刷新 358
图 23-5 EWDG_in 有效时刷新 359
图 23-6 EWDG_out 复位状态 359
图 24-1 CTU 结构框图 366
图 24-2 ADC 硬件触发 369
图 24-3 PWM_FAULT 触发输入 370
图 24-4 RTC 时钟捕获 371
图 24-5 PWM 捕获 ACMP 输出 372
图 24-6 UART 调制 373
图 24-7 ACMP HALL 输出至 PWM 374
图 24-8 PWM 调制 375

© 2013 - 2024 杰发科技有限公司

图 25- 1 UART 结构框图 403
图 25- 2 UART 传送器流程 404
图 25- 3 UART 接收器流程 404
图 25- 4 UART 噪声检测 405
图 25- 5 单向半双工连接 406
图 25- 6 硬件流控制连接 407
图 25- 7 硬件流控制原理 407
图 25- 8 单字节数据传输 408
图 25- 9 多字节数据传输 408
图 25- 10 实际电路连接 408
图 25- 11 LIN 帧流程 409
图 25- 12 IRDA 发送和接收 410
图 25- 13 IRDA 发送编码 410
图 25- 14 IRDA 接收解码 410
图 25- 15 UART 通讯信号连接图 412
图 25- 16 UART 唤醒芯片的典型流程 412
图 25- 17 波特率发生器框图 413
图 26- 1 CAN 控制器结构框图 440
图 26- 2 监听模式 452
图 26- 3 环回模式 452
图 26- 4 时间戳时钟配置 454
图 26- 5 CAN 位定时 457
图 27- 1 I2C 结构框图 484
图 27- 2 START 和 STOP 条件 484
图 27- 3 数据传输格式 485
图 27- 4 波特率生成 485
图 27- 5 发送器数据流程 486
图 27- 6 接收器数据流程 486
图 27- 7 主机组合模式 487
图 27- 8 主机发送器情形 1 488
图 27- 9 主机发送器情形 2 488
图 27- 10 主机接收器 489
图 27- 11 从机发送器 489
图 27- 12 从机接收器 490
图 27- 13 主机写从机模式的 BND 序列 492
图 27- 14 主机读从机模式的 BND 序列 492
图 27- 15 典型 I2C 从机中断程序 493
图 28- 1 SPI 系统连接 1
图 28- 2 SPI 结构框图 1
图 28- 3 主机 1-bit 模式数据流 1
图 28- 4 从机数据流 1
图 28- 5 主机 2-bit 模式数据流 1
图 28- 6 主机 4-bit 模式数据流 1
图 28- 7 CPHA=0 传输格式 1
图 28- 8 CPHA=1 传输格式 1
图 28- 9 波特率生成 1
图 28- 10 在模式故障检测使能时的 SCK 输出时序 1
图 28- 11 模式故障检测限制 1
图 28- 12 唤醒序列 1
图 28- 13 CS 连续模式 1
图 29- 1 EIO 结构框图 1

© 2013 - 2024 杰发科技有限公司

图 29-2 移位器结构框图 1
图 29-3 定时器结构框图 1
图 30-1 ADC 结构框图 1
图 30-2 ADC 上电时序 1
图 30-3 规则组序列 1
图 30-4 有效规则组序列 1
图 30-5 注入组序列 1
图 30-6 有效注入组序列 1
图 30-7 Mode 1 工作流程 1
图 30-8 Mode 2 工作流程 1
图 30-9 Mode 3 注入组扫描模式工作流程 1
图 30-10 Mode 3 在 ADC 空闲状态下具有注入触发的工作流程 1
图 30-11 Mode 3 注入组间隔模式工作流程 1
图 30-12 Mode 3 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程 1
图 30-13 Mode 4 工作流程 1
图 30-14 Mode 5 注入组扫描模式工作流程 1
图 30-15 Mode 5 在 ADC 空闲状态下具有注入触发的工作流程 1
图 30-16 Mode 5 注入组间隔模式工作流程 1
图 30-17 Mode 5 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程 1
图 30-18 Mode 6 操作流程 1
图 30-19 Mode 7 操作流程 1
图 30-20 Mode 8 操作流程 1
图 30-21 电平触发模式下监控区域 1
图 30-22 边沿触发模式下监控区域 1
图 30-23 ADC GEOE 校准转换示意 1
图 30-24 12-bit 配置下的 ADC 数据排布 1
图 30-25 10- bit 配置下的 ADC 数据排布 1
图 30-26 8- bit 配置下的 ADC 数据排布 1
图 30-27 ADC 同步触发时序图 1
图 30-28 硬件交织 1
图 30-29 硬件均值时序图 1
图 31-1 ACMP 结构框图(包含 DAC 和 CMP) 1
图 31-2 ACMP 内部比较器结构框图 1
图 31-3 轮询模式工作流程图 1
图 31-4 下降沿迟滞模式 1
图 31-5 双边沿迟滞模式 1
图 31-6 窗口控制模式示意图 1
图 31-7 ACMP 数字滤波器 1
图 32-1 PWM 结构框图 1
图 32-2 PWM 时钟框图 1
图 32-3 向上计数 1
图 32-4 向上-向下计数 1
图 32-5 工作模式配置 1
图 32-6 输入捕获模式(CHRSTEN = 0) 1
图 32-7 输入捕获模式(CHRSTEN = 1) 1
图 32-8 匹配设置输出比较模式 1
图 32-9 匹配清除输出比较模式 1
图 32-10 匹配翻转输出比较模式 1
图 32-11 EPWM 波形 1
图 32-12 CPWM 波形 1
图 32-13 向上计数组合模式输出波形 1

© 2013 - 2024 杰发科技有限公司

图 32- 14 (CNTIN < CHnV / CH(n + 1)V < MCVR) & (CHnV < CH(n + 1)V)条件下输出波形 1
图 32- 15 (CNTIN < CHnV < MCVR) & (CH(n + 1)V = MCVR)条件下输出波形 1
图 32- 16 (CHnV = CNTIN) & (CNTIN < CH(n + 1)V < MCVR)条件下输出波形 1
图 32- 17 (CNTIN < CHnV / CH(n+1)V < MCVR) & (CHnV > CH(n + 1)V)条件下输出波形 1
图 32- 18 (CH(n + 1)V < CNTIN) & (CNTIN < CHnV < MCVR)条件下输出波形 1
图 32- 19 (CH(n + 1)V > MCVR) & (CNTIN < CHnV < MCVR)条件下输出波形 1
图 32- 20 向上-向下计数区间范围 1
图 32- 21 CHn 匹配点 DIR = 1(Up)，CH(n + 1)匹配点 DIR = 1(Up) 1
图 32- 22 CHn 匹配点 DIR = 1(Up)，CH(n + 1)匹配点 DIR = 0(Down) 1
图 32- 23 CHn 匹配点 DIR = 0(Down)，CH(n + 1)匹配点 DIR = 1(Up) 1
图 32- 24 CHnV 匹配点 DIR = 0(Down)，CH(n + 1)V 匹配点 DIR = 0(Down) 1
图 32- 25 互补模式输出 1
图 32- 26 互补死区时间插入 1
图 32- 27 CH(n + 1)V 在下一周期匹配的输出波形 1
图 32- 28 多通道之间相位偏移输出波形 1
图 32- 29 双边沿捕获模式(CHnSCR[CHRSTEN] = 0 & CHn+1SCR[CHRSTEN] = 0) 1
图 32- 30 双边沿捕获模式(CHnSCR[CHRSTEN] = 0 & CHn+1SCR[CHRSTEN] = 1) 1
图 32- 31 两种常见的 Hall 安装方式 1
图 32- 32 计数和方向编码模式 1
图 32- 33 ABZ 编码模式 1
图 32- 34 向上计数 PWM counter 溢出 1
图 32- 35 向下计数 PWM counter 溢出 1
图 32- 36 最大值触发器 1
图 32- 37 FAULTIN 1
图 32- 38 PWM 同步信号选择 1
图 32- 39 HWTRIGMODE = 0 硬件触发事件 1
图 32- 40 软件触发事件 1
图 32- 41 边界周期与加载点 1
图 32- 42 PWM_MCVR 寄存器同步流程 1
图 32- 43 PWM_CNT 寄存器同步流程 1
图 32- 44 PWM_OMCR 寄存器同步流程 1
图 32- 45 PWM_INVCR 寄存器同步流程 1
图 32- 46 PWM_CHOSWCR 寄存器同步流程 1
图 32- 47 PWM_CHOPOLCR 寄存器同步流程 1
图 32- 48 PWM 特性优先级 1
图 32- 49 PWM 向上计数_边沿抖动波形 1
图 32- 50 PWM 向上-向下计数_边沿抖动波形 1
图 32- 51 PWM 向上计数_周期抖动波形 1
图 32- 52 PWM 向上-向下计数_周期抖动波形 1
图 32- 53 PWM DMA 信号 1
图 32- 54 PWM DMA Read 1
图 32- 55 PWM DMA Write 1
图 33- 1 PDT 结构框图 1
图 33- 2 PDT 脉冲输出(PODLY2 > PODLY1) 1
图 33- 3 PDT 脉冲输出(PODLY1 > PODLY2) 1
图 33- 4 PDT_SC[LDMOD] = 0h 条件下寄存器缓存更新 1
图 33- 5 PDT_SC[LDMOD] = 1h 条件下寄存器缓存更新 1
图 33- 6 PDT_SC[LDMOD] = 2h 条件下寄存器缓存更新 1
图 33- 7 PDT_SC[LDMOD] = 3h 条件下寄存器缓存更新 1
图 33- 8 PDT 延时采样 1
图 34- 1 Timer 结构框图 1

© 2013 - 2024 杰发科技有限公司

图 34- 2 链模式 1
图 34- 3 Mode0：TSOT=0,TROT=0,TSOI=0 1
图 34- 4 Mode0：TSOT=0,TROT=0,TSOI=1 1
图 34- 5 Mode0：TSOT=0,TROT=1,TSOI=0 1
图 34- 6 Mode0：TSOT=0,TROT=1,TSOI=1 1
图 34- 7 Mode0：TSOT=1,TROT=0,TSOI=0 1
图 34- 8 Mode0：TSOT=1,TROT=0,TSOI=1 1
图 34- 9 Mode0：TSOT=1,TROT=1,TSOI=0 1
图 34- 10 Mode0：TSOT=1,TROT=1,TSOI=1 1
图 34- 11 Mode1：TSOT=0,TROT=0,TSOI=0 1
图 34- 12 Mode2：TSOT=X,TROT=X,TSOI=0 1
图 34- 13 Mode2：TSOT=X,TROT=X,TSOI=1 1
图 34- 14 Mode3：TSOT=X,TROT=0,TSOI=0 1
图 34- 15 Mode3：TSOT=X,TROT=1,TSOI=0 1
图 34- 16 Mode3：TSOT=X,TROT=0,TSOI=1 1
图 35- 1 PCT 结构框图 1
图 36- 1 AC78408/9 RTC 结构框图 1
图 36- 2 AC78406/7 RTC 结构框图 1

© 2013 - 2024 杰发科技有限公司

表格目录

表 2-1 AC7840x 模块概述 55
表 3-1 AC7840x 中断向量表 58
表 4-1 小端格式存放方式 62
表 4-2 存储器映射表 62
表 6-1 MCM 寄存器映射 69
表 6-2 MCM_MISCR 寄存器 69
表 6-3 MCM_MLMDR0 寄存器 71
表 6-4 MCM_MBIST 寄存器 72
表 6-5 MCM_LMPEIR 寄存器 72
表 6-6 MCM_LMFAR 寄存器 73
表 6-7 MCM_MNCR 寄存器 73
表 7-1 PBR 寄存器映射 77
表 7-2 PBR_MPR_CORE 寄存器 77
表 7-3 PBR_MPR_DEBUG 寄存器 78
表 7-4 PBR_MPR_DMA 寄存器 79
表 7-5 PBR_PACRA 寄存器 80
表 7-6 PBR_PACRB 寄存器 82
表 7-7 PBR_PACRC 寄存器 84
表 7-8 PBR_PACRD 寄存器 87
表 7-9 PBR_PACRE 寄存器 89
表 7-10 PBR_PACRF 寄存器 91
表 7-11 PBR_PACRG 寄存器 93
表 8-1 主机访问分配 96
表 8-2 从机端口分配 96
表 8-3 存储器保护单元（MPU）寄存器映射 99
表 8-4 MPU_CESR 寄存器 100
表 8-5 MPU_EARn 寄存器 101
表 8-6 MPU_EDRn 寄存器 102
表 8-7 MPU_MPID 寄存器 103
表 8-8 MPU_RGDn_WORD0 寄存器 103
表 8-9 MPU_RGD0_WORD1 寄存器 104
表 8-10 MPU_RGDn_WORD0 寄存器 104
表 8-11 MPU_RGD0_WORD2 寄存器 105
表 8-12 MPU_RGDn_WORD2 寄存器 107
表 8-13 MPU_RGD0_WORD3 寄存器 109
表 8-14 MPU_RGDn_WORD3 寄存器 110
表 8-15 MPU_RGD0_AAC 寄存器 111
表 8-16 MPU_RGDn_AAC 寄存器 113
表 9-1 系统复位范围 119
表 9-2 RCM 中断汇总 120
表 9-3 复位（Reset）寄存器映射 121
表 9-4 RCM_CTRL 寄存器 121
表 9-5 RCM_EN 寄存器 123
表 9-6 RCM_STATUS 寄存器 125
表 10-1 典型 SPLL 配置参考表 130
表 10-2 外设时钟 131
表 10-3 外设时钟示意图 133
表 10-4 时钟寄存器映射 136

© 2013 - 2024 杰发科技有限公司

表 10-5 CKGEN_CTRL 寄存器 137
表 10-6 CKGEN_LP_CLK_MUX 寄存器 139
表 10-7 CKGEN_PERI_CLK_EN0 寄存器 140
表 10-8 CKGEN_PERI_CLK_EN1 寄存器 142
表 10-9 CKGEN_PERI_CLK_EN2 寄存器 143
表 10-10 CKGEN_PERI_SFT_RST0 寄存器 145
表 10-11 CKGEN_PERI_SFT_RST1 寄存器 147
表 10-12 CKGEN_PERI_SFT_RST2 寄存器 148
表 10-13 CKGEN_CLK_DIV1 寄存器 149
表 10-14 CKGEN_CLK_DIV2 寄存器 150
表 10-15 CKGEN_PERI_CLK_MUX0 寄存器 151
表 10-16 CKGEN_PERI_CLK_MUX1 寄存器 153
表 10-17 CKGEN_PERI_CLK_MUX2 寄存器 154
表 10-18 CKGEN_PERI_CLK_MUX3 寄存器 155
表 10-19 CKGEN_CLK_OUT_CFG 寄存器 157
表 10-20 CKGEN_PERI_CLK_DIV 寄存器 159
表 10-21 ANA_SPLL_CFG0 寄存器 161
表 10-22 ANA_SPLL_CFG1 寄存器 161
表 11-1 CMU 寄存器映射 169
表 11-2 CMU_CR 寄存器 169
表 11-3 CMU_RCCR 寄存器 170
表 11-4 CMU_HTCR 寄存器 171
表 11-5 CMU_LTCR 寄存器 171
表 11-6 CMU_SR 寄存器 172
表 12-1 支持 Standby 唤醒的 IO 复用功能 175
表 12-2 SPM 中断汇总 176
表 12-3 各模式下的模块功能 177
表 13-1 SPM 寄存器映射 181
表 13-2 SPM_PWR_MGR_CFG0 寄存器 182
表 13-3 SPM_PWR_MGR_CFG1 寄存器 183
表 13-4 SPM_PERIPH_SLEEP_ACK_STATUS 寄存器 185
表 13-5 SPM_PERIPH_SLEEP_ACK_EN 寄存器 186
表 13-6 SPM_STATUS 寄存器 189
表 13-7 SPM_STB_WP_EN 寄存器 190
表 13-8 SPM_STB_WP_STATUS 寄存器 192
表 14-1 硬件单点错误源列表 197
表 14-2 硬件潜在错误源列表 198
表 14-3 SMU 寄存器映射 200
表 14-4 SMU_SFES 寄存器 201
表 14-5 SMU_SFESS 寄存器 201
表 14-6 SMU_LFES 寄存器 202
表 14-7 SMU_LFESS 寄存器 203
表 14-8 SMU_ LKSEQ0 寄存器 204
表 14-9 SMU_ LKSEQ1 寄存器 204
表 14-10 SMU_ SWSFE 寄存器 205
表 14-11 SMU_ SWSFES 寄存器 206
表 14-12 SMU_ SWLFE 寄存器 206
表 14-13 SMU_ SWLFES 寄存器 207
表 14-14 SMU_SFRSTEN 寄存器 207
表 14-15 SMU_LFRSTEN 寄存器 208
表 14-16 SMU_SRSTCNTVAL 寄存器 209

© 2013 - 2024 杰发科技有限公司

表 14-17 SMU_SRSTCNT 寄存器 209
表 14-18 SMU_SRSTCNTS 寄存器 210
表 14-19 SMU_PATHCHK0 寄存器 210
表 15-1 P-Flash 存储器组织 213
表 15-2 P-Flash 信息区组织 213
表 15-3 P-Flash 安全信息区 213
表 15-4 D-Flash 存储器组织 214
表 15-5 Flash 命令 ID 214
表 15-6 Flash 中断汇总 215
表 15-7 编程分区配置 222
表 15-8 FlexRAM 功能配置 224
表 15-9 RWW 224
表 15-10 写保护寄存器设置写保护 225
表 15-11 写保护信息区设置写保护 225
表 15-12 UID 和 Secret KEY 地址信息 228
表 15-13 Flash 寄存器映射 229
表 15-14 FLASH_STAT 寄存器 229
表 15-15 FLASH_SEC 寄存器 232
表 15-16 FLASH_KEYULK 寄存器 233
表 15-17 FLASH_CNFG 寄存器 233
表 15-18 FLASH_CMD 寄存器 235
表 15-19 FLASH_ADDR 寄存器 235
表 15-20 FLASH_DATA0 寄存器 236
表 15-21 FLASH_DATA1 寄存器 236
表 15-22 FLASH_LEN 寄存器 237
表 15-23 FLASH_CST 寄存器 237
表 15-24 FLASH_PPROT 寄存器 238
表 15-25 FLASH_DPROT 寄存器 238
表 15-26 FLASH_DFADR 寄存器 239
表 15-27 FLASH_PART 寄存器 240
表 16-1 CSE 模块秘钥 242
表 16-2 CSE 模块秘钥描述 243
表 16-3 CSE 模块秘钥属性描述 244
表 16-4 PARM 格式 245
表 16-5 CSE 命令头 245
表 16-6 KeyID 组成 246
表 16-7 CMD_ENC_ECB 详细参数 246
表 16-8 CMD_EN_ECB 输入参数格式 247
表 16-9 CMD_ENC_ECB 输出参数格式 247
表 16-10 CMD_ENC_CBC 详细参数 248
表 16-11 CMD_ENC_CBC 输入参数格式 248
表 16-12 CMD_ENC_CBC 连续输入参数格式 248
表 16-13 CMD_ENC_CBC 输出参数格式 249
表 16-14 CMD_ENC_CBC 连续输出参数格式 249
表 16-15 CMD_DEC_ECB 详细参数 249
表 16-16 CMD_DEC_ECB 输入参数格式 249
表 16-17 CMD_DEC_ECB 输出参数格式 250
表 16-18 CMD_DEC_CBC 详细参数 250
表 16-19 CMD_DEC_CBC 输入参数格式 250
表 16-20 CMD_DEC_ECB 连续输入参数格式 251
表 16-21 CMD_DEC_CBC 输出参数格式 251

© 2013 - 2024 杰发科技有限公司

表 16- 22 CMD_DEC_CBC 连续输出参数格式 251
表 16- 23 CMD_GENERATE_MAC 详细参数 252
表 16- 24 CMD_GENERATE_MAC 输入参数格式 252
表 16- 25 CMD_GENERATE_MAC 输出参数格式 252
表 16- 26 CMD_GENERATE_MAC 连续输入参数格式 253
表 16- 27 CMD_GENERATE_MAC 连续输出参数格式 253
表 16- 28 CMD_GENERATE_MAC（Point Method）详细输入参数格式 253
表 16- 29 CMD_GENERATE_MAC（Point Method）详细输出参数格式 254
表 16- 30 CMD_VERIFY_MAC 详细参数 255
表 16- 31 CMD_VERIFY_MAC 输入参数格式 255
表 16- 32 CMD_VERIFY_MAC 输出参数格式 255
表 16- 33 CMD_VERIFY_MAC 连续输入参数格式 256
表 16- 34 CMD_VERIFY_MAC 连续输出参数格式 256
表 16- 35 CMD_VERIFY_MAC（Pinter Method）输入参数格式 257
表 16- 36 CMD_VERIFY_MAC（Pinter Method）输出参数格式 257
表 16- 37 CMD_LOAD_KEY 详细参数 258
表 16- 38 CMD_LOAD_KEY 输入参数格式 258
表 16- 39 CMD_LOAD_KEY 输出参数格式 258
表 16- 40 CMD_LOAD_PLAIN_KEY 详细参数 259
表 16- 41 CMD_LOAD_PLAIN_KEY 输入参数格式 259
表 16- 42 CMD_EXPORT_RAM_KEY 详细参数 259
表 16- 43 CMD_EXPORT_RAM_KEY 输入参数格式 260
表 16- 44 CMD_EXPORT_RAM_KEY 输出参数格式 260
表 16- 45 CMD_INIT_RNG 详细参数 261
表 16- 46 CMD_INIT_RNG 输入参数格式 261
表 16- 47 CMD_EXTEND_SEED 详细参数 261
表 16- 48 CMD_EXTEND_SEED 输入参数格式 262
表 16- 49 CMD_RND 详细参数 262
表 16- 50 CMD_RND 输入参数格式 262
表 16- 51 CMD_RND 输出参数格式 263
表 16- 52 CMD_BOOT_FAILURE 详细参数 263
表 16- 53 CMD_BOOT_FAILURE 输入参数格式 263
表 16- 54 CMD_BOOT_OK 详细参数 264
表 16- 55 CMD_BOOT_OK 输入参数格式 264
表 16- 56 CMD_GET_ID 详细参数 264
表 16- 57 CMD_GET_ID 输入参数格式 264
表 16- 58 CMD_GET_ID 输出参数格式 265
表 16- 59 CMD_BOOT_DEFINE 详细参数 266
表 16- 60 CMD_BOOT_DEFINE 输入参数格式 266
表 16- 61 CMD_DBG_CHAL 详细参数 266
表 16- 62 CMD_DBG_CHAL 输入参数格式 267
表 16- 63 CMD_DBG_CHAL 输出参数格式 267
表 16- 64 CMD_DBG_AUTH 详细参数 268
表 16- 65 CMD_DBG_AUTH 输入参数格式 268
表 16- 66 CMD_MP_COMPRESS 详细参数 268
表 16- 67 CMD_MP_COMPRESS 输入参数格式 269
表 16- 68 CMD_MP_COMPRESS 输出参数格式 269
表 16- 69 CMD_MP_COMPRESS 连续输入参数格式 269
表 16- 70 CMD_MP_COMPRESS 连续输出参数格式 270
表 16- 71 CSE 中断汇总 270
表 16- 72 CSE Key Size 说明 270

© 2013 - 2024 杰发科技有限公司

表 16-73 授权秘钥 271
表 16-74 CSE 状态寄存器 275
表 16-75 CSESTAT 中标志位规则 276
表 16-76 PRAM 寄存器映射 276
表 16-77 CMD_PG0_WORD0 寄存器 277
表 16-78 CMD_PG0_WORD1 寄存器 278
表 16-79 CMD_PG0_WORD2 寄存器 280
表 16-80 CMD_PG0_WORD3 寄存器 280
表 16-81 CMD_PGm_WORDn 寄存器 281
表 17-1 SRAM 中断汇总 284
表 17-2 SRAM 寄存器映射 284
表 17-3 CH0_STATUS0 寄存器 284
表 17-4 CH0_STATUS1 寄存器 285
表 17-5 CH1_STATUS0 寄存器 285
表 17-6 CH1_STATUS1 寄存器 286
表 17-7 ECC_ERR_CTRL 寄存器 287
表 18-1 EIM 寄存器映射 291
表 18-2 EIM 全局使能控制寄存器 292
表 18-3 EIM 通道使能控制寄存器 292
表 18-4 SRAM_L 数据错误注入寄存器 293
表 18-5 SRAM_L 地址错误注入寄存器 293
表 18-6 SRAM_L 校验码错误注入寄存器 294
表 18-7 SRAM_U 数据错误注入寄存器 294
表 18-8 SRAM_U 地址错误注入寄存器 295
表 18-9 SRAM_U 校验码错误注入寄存器 295
表 19-1 GPIO 复用功能描述 300
表 19-2 GPIO 模块中断汇总 308
表 19-3 GPIO 模块唤醒源 309
表 19-4 GPIO 寄存器映射 310
表 19-5 PORT_PCR 寄存器 311
表 19-6 PORT_ISFR 寄存器 313
表 19-7 PORT_DFER 寄存器 313
表 19-8 PORT_DFCR 寄存器 314
表 19-9 PORT_DFWR 寄存器 314
表 19-10 GPIO_PODR 寄存器 315
表 19-11 GPIO_PSOR 寄存器 316
表 19-12 GPIO_PROR 寄存器 316
表 19-13 GPIO_PIOR 寄存器 317
表 19-14 GPIO_PIDR 寄存器 317
表 19-15 GPIO_POER 寄存器 318
表 19-16 GPIO_PIER 寄存器 319
表 20-1 DMA 请求列表 321
表 20-2 DMA 中断汇总 323
表 20-3 DMA 寄存器映射 327
表 20-4 DMA_TOP_RST 寄存器 328
表 20-5 DMA_STATUS 寄存器 329
表 20-6 DMA_INTEN 寄存器 330
表 20-7 DMA_RST 寄存器 331
表 20-8 DMA_STOP 寄存器 332
表 20-9 DMA_CONFIG 寄存器 333
表 20-10 DMA_CHAN_LENGTH 寄存器 334

© 2013 - 2024 杰发科技有限公司

表 20-11 DMA_SSTART_ADDR 寄存器 334
表 20-12 DMA_SEND_ADDR 寄存器 335
表 20-13 DMA_DSTART_ADDR 寄存器 335
表 20-14 DMA_CHAN_ENABLE 寄存器 336
表 20-15 DMA_DATA_TRANS_NUM 寄存器 337
表 20-16 DMA_INTER_FIFO_DATA_LEFT_NUM 寄存器 337
表 20-17 DMA_DEND_ADDR 寄存器 338
表 20-18 DMA_ADDR_OFFSET 寄存器 338
表 20-19 DMA_MUX_CFG 寄存器 339
表 21-1 CRC 寄存器映射 344
表 21-2 CRC_DATA 寄存器 344
表 21-3 CRC_POLY 寄存器 345
表 21-4 CRC_CTRL 寄存器 346
表 22-1 WDG 中断汇总 350
表 22-2 低功耗模式 WDG 计数行为 350
表 22-3 看门狗低功耗模式下唤醒源 351
表 22-4 WDG 寄存器映射 352
表 22-5 WDG_CS0 寄存器 353
表 22-6 WDG_CS1 寄存器 354
表 22-7 WDG_CNT 寄存器 355
表 22-8 WDG_TOVAL 寄存器 356
表 22-9 WDG_WIN 寄存器 356
表 23-1 EWDG 中断汇总 360
表 24-1 TRGMUX 触发源存储器映射表 367
表 24-2 CTU 支持触发目标选择的模块 368
表 24-3 CTU 模块低功耗模式 373
表 24-4 CTU 寄存器映射 375
表 24-5 TRGMUX_DMA 寄存器 376
表 24-6 TRGMUX_EXTOUT0 寄存器 377
表 24-7 TRGMUX EXTOUT1 寄存器 378
表 24-8 TRGMUX_ADC0_REG 寄存器 379
表 24-9 TRGMUX ADC0_INJ 寄存器 380
表 24-10 TRGMUX_ADC1_REG 寄存器 381
表 24-11 TRGMUX_ADC1_INJ 寄存器 382
表 24-12 TRGMUX_ACMP 寄存器 383
表 24-13 TRGMUX_PWM0 寄存器 383
表 24-14 TRGMUX_PWM1 寄存器 384
表 24-15 TRGMUX_PWM2 寄存器 385
表 24-16 TRGMUX_PWM3 寄存器 386
表 24-17 TRGMUX_PWM4 寄存器 387
表 24-18 TRGMUX_PWM5 寄存器 388
表 24-19 TRGMUX_TIMER 寄存器 388
表 24-20 TRGMUX_PCT 寄存器 389
表 24-21 TRGMUX_UART0 寄存器 390
表 24-22 TRGMUX_UART1 寄存器 390
表 24-23 TRGMUX_PDT0 寄存器 391
表 24-24 TRGMUX_PDT1 寄存器 392
表 24-25 TRGMUX_EIO 寄存器 392
表 24-26 CTU_CFG 寄存器 394
表 24-27 CTU_SW 寄存器 397
表 24-28 PWM_MODULATION 寄存器 398

© 2013 - 2024 杰发科技有限公司

表 24-29 ADC_SYNC 寄存器 400
表 25-1 功能分类和配置 401
表 25-2 UART 输入输出定时 404
表 25-3 UART 的典型波特率及误差率（总线时钟为 24MHz） 405
表 25-4 UART 的典型波特率及误差率（总线时钟为 48MHz） 405
表 25-5 UART 的典型波特率及误差率（总线时钟为 60MHz） 406
表 25-6 UART 中断汇总 411
表 25-7 UART 唤醒源 412
表 25-8 UART 寄存器映射 415
表 25-9 UART_RBR/THR 寄存器 416
表 25-10 UART_DIV_L 寄存器 417
表 25-11 UART_DIV_H 寄存器 417
表 25-12 UART_LCR0 寄存器 418
表 25-13 UART_LCR1 寄存器 419
表 25-14 UART_FCR 寄存器 420
表 25-15 UART_EFR 寄存器 421
表 25-16 UART_IER 寄存器 422
表 25-17 UART_LSR0 寄存器 423
表 25-18 UART_LSR1 寄存器 425
表 25-19 UART_SMP_CNT 寄存器 427
表 25-20 UART_ADDR 寄存器 427
表 25-21 UART_DATA 寄存器 428
表 25-22 UART_RS485_GUARD 寄存器 429
表 25-23 UART_SLEEP_EN 寄存器 430
表 25-24 UART_DMA_EN 寄存器 430
表 25-25 UART_DIV_FRAC 寄存器 431
表 25-26 UART_MTCHCR 寄存器 431
表 25-27 UART_RS485CR 寄存器 432
表 25-28 UART_ CNTR 寄存器 433
表 25-29 UART_IDLE 寄存器 433
表 25-30 UART_LINCR 寄存器 434
表 25-31 UART_BRKLGH 寄存器 435
表 25-32 UART_PMIN0 寄存器 436
表 25-33 UART_PMIN1 寄存器 436
表 25-34 UART_PWDTH 寄存器 437
表 26-1 CAN 型号选择 438
表 26-2 接收缓冲区寄存器 RBUF – 标准格式（RO） 441
表 26-3 接收缓冲区寄存器 RBUF – 扩展格式（RO） 441
表 26-4 发送缓冲区寄存器 TBUF – 标准格式（RW） 442
表 26-5 发缓冲区寄存器 TBUF – 扩展格式（RW） 443
表 26-6 RBUF 和 TBUF 的控制位 444
表 26-7 DLC 定义（基于 CAN 2.0 和 CAN FD 规范） 445
表 26-8 过滤器设置示例 446
表 26-9 CAN 中断汇总 455
表 26-10 CAN 唤醒中断 456
表 26-11 CAN 模块唤醒源 456
表 26-12 CAN 定时段 457
表 26-13 CAN 控制器定时配置 458
表 26-14 48MHz can_clk 的示例设置 459
表 26-15 8MHz can_clk 的示例设置 459
表 26-16 48MHz can fd clk 的示例设置 460

© 2013 - 2024 杰发科技有限公司

表 26- 17 软件复位 460
表 26- 18 CAN-CRTL 寄存器映射 463
表 26- 19 CAN_CTRL0 寄存器 464
表 26- 20 CAN_CTRL1 寄存器 469
表 26- 21 CAN_SBITRATE 寄存器 472
表 26- 22 CAN_FBITRATE 寄存器 473
表 26- 23 CAN_ERRINFO 寄存器 474
表 26- 24 CAN_ACFCTRL0 寄存器 475
表 26- 25 CAN_ACFCTRL1 寄存器 476
表 26- 26 CAN_ACFCTRL2 寄存器 477
表 26- 27 CAN_ACF 寄存器 478
表 26- 28 CAN_ACF 寄存器 478
表 26- 29 CAN_VERMEM 寄存器 479
表 26- 30 CAN_MEMES 寄存器 481
表 26- 31 CAN_WAKEUP 寄存器 482
表 27- 1 I2C 中断汇总 490
表 27- 2 I2C 唤醒源 491
表 27- 3 I2C 寄存器映射 494
表 27- 4 I2C_ADDR0 寄存器 495
表 27- 5 I2C_ADDR1 寄存器 495
表 27- 6 I2C_SAMPLE_CNT 寄存器 496
表 27- 7 I2C_STEP_CNT 寄存器 497
表 27- 8 I2C_CTRL0 寄存器 497
表 27- 9 I2C_CTRL1 寄存器 499
表 27- 10 I2C_CTRL2 寄存器 500
表 27- 11 I2C_CTRL3 寄存器 502
表 27- 12 I2C_STATUS0 寄存器 503
表 27- 13 I2C_STATUS1 寄存器 505
表 27- 14 I2C_DATA 寄存器 507
表 27- 15 I2C_STARTSTOP 寄存器 508
表 28- 1 中断汇总 517
表 28- 2 SPI 唤醒源 517
表 28- 3 SPI 寄存器映射 520
表 28- 4 SPI_CFG0 寄存器 521
表 28- 5 SPI_CFG1 寄存器 522
表 28- 6 SPI_CMD 寄存器 524
表 28- 7 SPI_STATUS 寄存器 525
表 28- 8 SPI_DATA 寄存器 527
表 28- 9 SPI_CFG2 寄存器 527
表 28- 10 SPI DMV 寄存器 530
表 29- 1 EIO 提供资源 531
表 29- 2 EIO 标志位说明 536
表 29- 3 EIO 中断汇总 537
表 29- 4 EIO 模块唤醒源 537
表 29- 5 UART 发送配置示例 538
表 29- 6 UART 接收配置示例 539
表 29- 7 带有 RTS 的 UART 接收配置 539
表 29- 8 SPI 主模式(CPHA=0)配置 540
表 29- 9 SPI 主模式(CPHA=1)配置 541
表 29- 10 SPI 从模式(CPHA=0)配置 542
表 29- 11 SPI 从模式(CPHA=1)配置 543

© 2013 - 2024 杰发科技有限公司

表 29- 12 I2C 主模式配置 544
表 29- 13 I2S 主模式配置 544
表 29- 14 I2S 从模式配置 546
表 29- 15 EIO 寄存器映射 547
表 29- 16 EIO_CTRL 寄存器 548
表 29- 17 EIO_PIN 寄存器 549
表 29- 18 EIO_SHIFTSTAT 寄存器 549
表 29- 19 EIO_SHIFTERR 寄存器 550
表 29- 20 EIO_TIMSTAT 寄存器 551
表 29- 21 EIO_SHIFTSIEN 寄存器 552
表 29- 22 EIO_SHIFTEIEN 寄存器 552
表 29- 23 EIO_TIMIEN 寄存器 553
表 29- 24 EIO_SHIFTSDEN 寄存器 553
表 29- 25 EIO_SHIFTCTLn 寄存器 554
表 29- 26 EIO_SHIFTCFGn 寄存器 555
表 29- 27 EIO_SHIFTBUFn 寄存器 556
表 29- 28 EIO_SHIFTBUFBISn 寄存器 557
表 29- 29 EIO_SHIFTBUFBYSn 寄存器 558
表 29- 30 EIO_SHIFTBUFBBSn 寄存器 558
表 29- 31 EIO_TIMCTLn 寄存器 559
表 29- 32 EIO_TIMCFGn 寄存器 561
表 29- 33 EIO_TIMCMPn 寄存器 562
表 30- 1 工作模式配置表 567
表 30- 2 不同触发方式下的响应行为 574
表 30- 3 模拟监控通道配置 575
表 30- 4 ADC 采样率示例 580
表 30- 5 ADC 模块唤醒源 583
表 30- 6 ADC 重置范围 584
表 30- 7 ADC 寄存器映射 585
表 30- 8 ADC_STR 寄存器 586
表 30- 9 ADC_CTRL0 寄存器 587
表 30- 10 ADC_CTRL1 寄存器 588
表 30- 11 ADC_IOFRx（x= 0 ~ 3）寄存器 589
表 30- 12 ADC_AMOHR 寄存器 590
表 30- 13 ADC_AMOLR 寄存器 590
表 30- 14 ADC_SPT0 寄存器 591
表 30- 15 ADC_SPT1 寄存器 592
表 30- 16 ADC_SPT2 寄存器 593
表 30- 17 ADC_SPT3 寄存器 594
表 30- 18 ADC_RSQRx 寄存器 595
表 30- 19 ADC_ISQRx 寄存器 596
表 30- 20 ADC_SQL 寄存器 597
表 30- 21 ADC_CALI0 寄存器 598
表 30- 22 ADC_CALI1 寄存器 598
表 30- 23 ADC_RDRx(x=0~23)寄存器 599
表 30- 24 ADC_IDRx (x=0 ~ 3)寄存器 599
表 30- 25 AUXADC_CFG0 寄存器 600
表 30- 26 AUXADC_CFG1 寄存器 601
表 30- 27 AUXADC_CFG4 寄存器 602
表 30- 28 AUXADC_CFG5 寄存器 603
表 31- 1 ACMP 中断汇总 610

© 2013 - 2024 杰发科技有限公司

表 31- 2 ACMP 模块唤醒源 610
表 31- 3 ACMP 寄存器映射 611
表 31- 4 ACMP_CR0 寄存器 612
表 31- 5 ACMP_CR1 寄存器 614
表 31- 6 ACMP_CR2 寄存器 615
表 31- 7 ACMP_CR3 寄存器 616
表 31- 8 ACMP_CR4 寄存器 617
表 31- 9 ACMP_DR 寄存器 617
表 31- 10 ACMP_SR 寄存器 618
表 31- 11 ACMP_FD 寄存器 619
表 31- 12 ACMP_OPA 寄存器 619
表 31- 13 ACMP_OPB 寄存器 620
表 31- 14 ACMP_OPC 寄存器 621
表 31- 15 ACMP_CLK 寄存器 622
表 31- 16 ACMPDAC_CFG 寄存器 622
表 32- 1 组合模式软件输出控制行为 647
表 32- 2 PWM 故障源 648
表 32- 3 PWM 中断汇总 663
表 32- 4 PWM 低功耗模式 663
表 32- 5 PWM 寄存器映射 664
表 32- 6 PWM_INIT 寄存器 665
表 32- 7 PWM_CNT 寄存器 667
表 32- 8 PWM_MCVR 寄存器 668
表 32- 9 PWM_CHnSCR 寄存器 668
表 32- 10 PWM_CHnV 寄存器 670
表 32- 11 PWM_CNTIN 寄存器 670
表 32- 12 PWM_STR 寄存器 671
表 32- 13 PWM_FUNCSEL 寄存器 672
表 32- 14 PWM_SYNC 寄存器 675
表 32- 15 PWM_OUTINIT 寄存器 677
表 32- 16 PWM_OMCR 寄存器 678
表 32- 17 PWM_MODESEL 寄存器 680
表 32- 18 PWM_DTSET 寄存器 0 685
表 32- 19 PWM_DTSET 寄存器 1 686
表 32- 20 PWM_EXTTRIG 寄存器 688
表 32- 21 PWM_CHOPOLCR 寄存器 690
表 32- 22 PWM_FDSR 寄存器 691
表 32- 23 PWM_CAPFILTER 寄存器 693
表 32- 24 PWM_FFAFER 寄存器 694
表 32- 25 PWM_QDI 寄存器 696
表 32- 26 PWM_CONF 寄存器 697
表 32- 27 PWM_FLTPOL 寄存器 699
表 32- 28 PWM_SYNCONF 寄存器 700
表 32- 29 PWM_INVCR 寄存器 702
表 32- 30 PWM_CHOSWCR 寄存器 703
表 32- 31 PWM_DITHER0 寄存器 705
表 32- 32 PWM_DITHER1 寄存器 706
表 32- 33 PWM_DITHER2 寄存器 707
表 32- 34 PWM_DMACTRL 寄存器 708
表 33- 1 加载模式 714
表 33- 2 PDT 中断汇总 717

© 2013 - 2024 杰发科技有限公司

表 33- 3 PDT 模块低功耗模式 717
表 33- 4 PDT 寄存器映射 719
表 33- 5 PDT_SC 寄存器 720
表 33- 6 PDT_MOD 寄存器 722
表 33- 7 PDT_DLY0 寄存器 722
表 33- 8 PDT_DLY1 寄存器 723
表 33- 9 PDT_DLY2 寄存器 723
表 33- 10 PDT_DLY3 寄存器 724
表 33- 11 PDT_DLY4 寄存器 724
表 33- 12 PDT_DLY5 寄存器 725
表 33- 13 PDT_DLY6 寄存器 725
表 33- 14 PDT_DLY7 寄存器 726
表 33- 15 PDT_IDLY 寄存器 726
表 33- 16 PDT_PODLY 寄存器 726
表 33- 17 PDT_DLYn_EN 寄存器 727
表 33- 18 PDT_POEN 寄存器 728
表 33- 19 PDT_CNT 寄存器 729
表 34- 1 Timer 中断汇总 735
表 34- 2 Timer 唤醒源 735
表 34- 3 Timer 控制位和工作模式组合 736
表 34- 4 Timer 寄存器映射 744
表 34- 5 TIMER_CR 寄存器 745
表 34- 6 TIMER_SR 寄存器 746
表 34- 7 TIMER_IER 寄存器 747
表 34- 8 TIMER_ENR 寄存器 748
表 34- 9 TIMER_CHx_TVAL 寄存器 749
表 34- 10 TIMER_CHx_CVAL 寄存器 749
表 34- 11 TIMER_CHx_CTRL 寄存器 750
表 35- 1 PCT 中断汇总 756
表 35- 2 PCT 唤醒源 756
表 35- 3 PCT 寄存器映射 758
表 35- 4 PCT_CSR 寄存器 758
表 35- 5 PCT_PSR 寄存器 760
表 35- 6 PCT_CMR 寄存器 762
表 35- 7 PCT_CNR 寄存器 762
表 36- 1 RTC 中断汇总 766
表 36- 2 RTC 唤醒源 766
表 36- 3 RTC 寄存器映射 767
表 36- 4 RTC_CTRL 寄存器 768
表 36- 5 RTC_TAR 寄存器 769
表 36- 6 RTC_TC 寄存器 770
表 36- 7 RTC_PSR 寄存器 770
表 36- 8 RTC_PSC 寄存器 771
表 36- 9 RTC_SR 寄存器 771
表 36- 10 RTC_LR 寄存器 772

© 2013 - 2024 杰发科技有限公司

1 文中的约定（Conventions）

1.1 术语缩写

缩写	全称	描述
AAI	Auto Address Increment	地址自增模式
AHB	Advanced High Performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	高级外设总线
CAN	Controller Area Network	控制器局域网
CKGEN	Clock Generator	时钟产生器
ECC	Error Checking and Correction	错误检测以及纠正
EIM	Error Injection Module	错误注入模块
FBKDIV	Feedback Divider	反馈除法器
HSE	High Speed External Clock	高速外部时钟
HSI	High Speed Internal Clock	高速内部时钟
ISP	In-System Programming	在系统编程
LIN	Local Interconnect Network	局域互联网络
LRU	Least Recently Used	最少最近使用
LSI	Low Speed Internal Clock	低速内部时钟
LVR	Low Voltage Reset	低电压复位
NMI	Non Maskable Interrupt	不可屏蔽中断
OSC	Oscillator	振荡器
PCT	Pulse Count Timer	脉冲计数定时器
POR	Power On Reset	上电复位
POSDIV	Post Divider	预置除法器
PREDIV	Previous Divider	前置除法器
LVD	Low Voltage Detect	低电压检测
SPM	System Power Manager	系统电源管理
SRAM	Static Random-Access Memory	静态随机存取存储器
SPLL	System Phase Locked Loop	系统锁相环
VCO	Voltage Controlled Oscillator	压控振荡器
VHSI	Very High Speed Internal Clock	超高速内部时钟
XOSC	External Crystal Oscillator	晶体振荡器
MCM	Miscellaneous Control Module	杂项控制器
MPU	Memory Protect Unit	存储保护单元
EIO	Enhanced Input and Output	增强型输入输出
I2C	Inter-Integrated Circuit	集成电路总线
SCL	Serial Clock	串行时钟线

© 2013 - 2024 杰发科技有限公司

缩写	全称	描述
SDA	Serial Data	串行数据线
SMBus	System Management Bus	系统管理总线
MSB	Most Significant Bit	最高有效位
ACK	Acknowledge	应答
NACK	Not Acknowledge	无应答
MCU	Microcontroller Unit	微控制单元
RTC	Real Time Clock	实时计数器

1.2 寄存器描述表中使用的缩写

缩写	全称	描述
RW	Read /Write	软件能读写此位
RO	Read-only	软件只能读此位
WO	Write-only	软件只能写此位，读此位将返回复位值
R/W1C	Read/Write 1 clear	软件可以读此位，也可以通过写'1'清除此位，写 0 对此位无影响
R/W0C	Read /Write 0 clear	软件可以读此位，也可以通过写'0'清除此位，写 1 对此位无影响
RC	Read clear	软件读清除，写 0 对此位无影响
RC/W0C	Read clear/Write 0 clear	软件读清除或写 0 清除位
RC/W1C	Read clear/Write 1 clear	软件读清除或写 1 清除位

1.3 数字表示方法

缩写	全称	描述
b	Binary number	二进制表示方法，比如：0101b，表示二进制 0101
d	Decimal number	十进制表示方法，统一不带后缀，比如：5，表示十进制 5
h	Hexadeimal number	十六进制表示方法： ■ 表格里用后缀 h，比如：6h，表示十六进制 6 ■ 其它地方用前缀 0x，比如：0x6，表示十六进制 6

© 2013 - 2024 杰发科技有限公司

2 简介（Overview）

2.1 概要

AC7840x 是采用 ARM Cortex™-M4F 内核的高性能、低功耗 MCU。

- 频率高达 120MHz
- 工作温度范围支持-40℃ ~ +125℃
- 工作电压支持 2.7V ~ 5.5V
- 提供 LQFP64/100/144 三种封装
- 支持 ASIL-B 等级的安全功能 ISO26262

2.2 模块概述

表 2-1 是 AC7840x 最大封装的模块概述。

表 2-1 AC7840x 模块概述

模块	说明
内核	■ ARM Cortex™-M4F 32位MCU 内核 – 高达120 MHz CPU频率 – 集成数字信号处理器(DSP) – 支持浮点单元(FPU) – 可配置嵌套向量中断控制器(NVIC)
存储器	■ 高达 1024KB(P-Flash) Flash存储器，带ECC ■ 128KB(D-Flash)Flash存储器，带ECC，可用于分区后存储CSE数据 ■ 高达 64KB(SRAM_L) +60KB(SRAM_U) SRAM存储器，带ECC ■ 4KB FlexRAM存储器，可用于SRAM或做为CSE保存数据访问入口 ■ 4KB程序Cache
时钟	■ 外部晶振或谐振器(HSE) – 范围：4 MHz ~ 30 MHz ■ 外部输入时钟 – <=50 MHz ■ 内置高频振荡器(HSI) – 8 MHz振荡器 ■ 内置超高频振荡器(VHSI) – 48 MHz振荡器 ■ 内置低频振荡器(LSI) – 128 KHz振荡器 ■ 系统锁相环(SPLL) – 最大频率120MHz – 支持HSE或HSI为输入时钟

© 2013 - 2024 杰发科技有限公司

模块	说明
系统	■ 1个外设桥(PBR)模块 ■ 1个存储保护(MPU)模块 ■ 1个看门狗(WDG)模块 ■ 1个外部看门狗(EWDG)模块 ■ 1个16通道的DMA模块 ■ 1个连接单元(CTU)模块 ■ 1个CRC校验模块 ■ 1个系统电源管理(SPM)模块 ■ 1个安全管理(SMU)单元 ■ 1个时钟监控(CMU)单元 ■ 1个硬件加密(CSE)模块
电源管理	■ 6种电源管理模式：RUN，VLPR，STOP1，STOP2，VLPS，Standby模式(Standby模式仅AC78408/9支持) ■ 低电压检测和复位功能(LVD/LVR)
模拟	■ 2个12位模数转换器(ADC)，每个ADC最多支持24路外部通道+4路内部通道 ■ 1个模拟比较器(ACMP)，内置1个8位的数模转换器(DAC) ■ 1个温度传感器
定时器	■ 6个全功能8通道脉宽调制(PWM)控制器 ■ 4个32bit周期性中断定时器(TIMER) ■ 1个16bit脉冲计数器(PCT) ■ 实时计数器(RTC) ■ 2个可编程延时定时器(PDT) ■ 系统滴答定时器(SysTick)
通信接口	■ 3个串行外设接口(SPI)模块 ■ 1个内部集成电路(I2C)模块 ■ 4个通用异步收发器(UART)模块，都支持LIN ■ 最多4个局域网控制器(CAN)模块，都支持CANFD ■ 1个增强型IO(EIO)模块，支持模拟UART, I2C, SPI, I2S, PWM等协议
通用接口	■ 最多128个通用输入输出控制器(GPIO) ■ 不可屏蔽中断(NMI) ■ 5个中断(IRQ)
调试接口	■ 串行调试 SWD&JTAG 接口

© 2013 - 2024 杰发科技有限公司

3 内核和系统（Core And System）

3.1 系统架构

AC7840x 系列 MCU 的 Crossbar Switch(AXBS-Lite)是改进的 ARM AMBA 总线矩阵，用于互联 CPU 内核与系统存储器及外设模块的访问(读写)。

- AXBS-Lite 的主机(Master)可以主动发起数据访问请求，而从机(Slave)则只能被动接受访问；
- 每个 Crossbar Switch 的端口都有自己的控制总线、地址总线和数据总线；
- 不同的主机(Core/Debugger/DMA)可以同时访问不同的从机，从而保证 MCU 系统工作带宽；
- 主机对从机的访问受 MPU 保护，对外设桥 PBR 以及外设的访问不受 MPU 保护；
- 外设桥 PBR 保护对外设的访问。

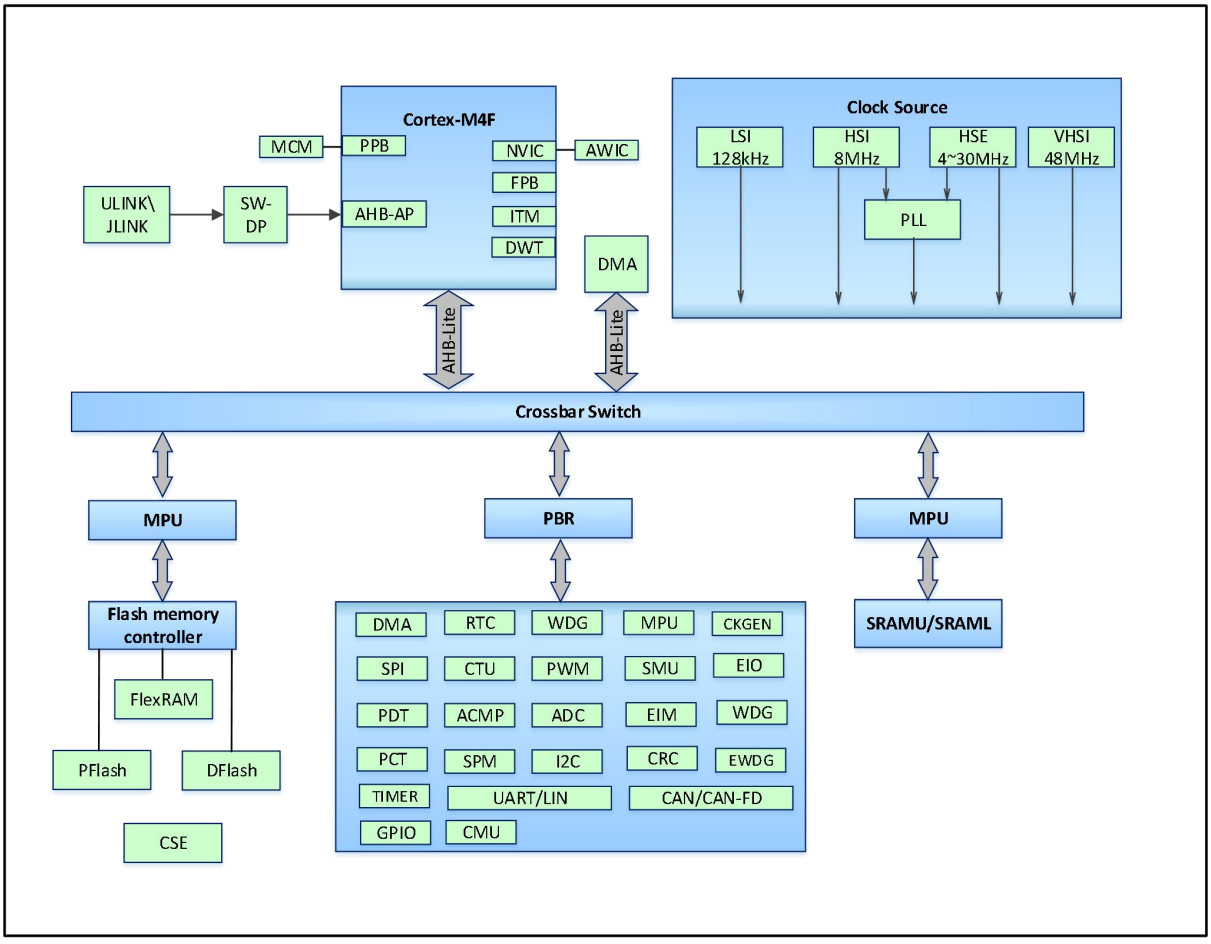


图 3-1 系统结构框图

© 2013 - 2024 杰发科技有限公司

3.2 嵌套中断向量控制器 (NVIC)

【特性】

- 93 个可屏蔽中断通道（不包括 16 个 Cortex™-M4F 中断）
- 16 个可编程优先等级（使用了 4 位 中断优先级）
- 低延迟的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

NVIC 和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理新的中断。

NVIC 管理着包括内核异常在内的所有中断。更多关于异常和 NVIC 编程的内容，请参考《ARM Cortex™-M4F 技术参考手册》中的嵌套向量中断控制器章节。表 3-1 是 AC7840x 最大封装的中断向量表。

表 3-1 AC7840x 中断向量表

中断向量号	优先级	优先级类型	名称	说明
	-3	固定	Reset	复位
	-2	固定	NMI	不可屏蔽中断
	-1	固定	HardFault	所有类型的错误
	0	可设置	MemManage	存储器管理
	0	可设置	BusFault	预取指失败，存储器访问失败
	0	可设置	UsageFault	未定义的指令或非法状态
	-	-	-	保留
	0	可设置	SVCall	通过 SWI 指令调用的系统服务
	0	可设置	Debug Monitor	调试监控器
	-	-	-	保留
	0	可设置	PendSV	可挂起的系统服务
	0	可设置	SysTick	系统嘀嗒定时器
0	0	可设置	DMA_Channel0_IRQHandler	DMA 通道 0 全局中断
1	0	可设置	DMA_Channel1_IRQHandler	DMA 通道 1 全局中断
2	0	可设置	DMA_Channel2_IRQHandler	DMA 通道 2 全局中断
3	0	可设置	DMA_Channel3_IRQHandler	DMA 通道 3 全局中断
4	0	可设置	DMA_Channel4_IRQHandler	DMA 通道 4 全局中断
5	0	可设置	DMA_Channel5_IRQHandler	DMA 通道 5 全局中断
6	0	可设置	DMA_Channel6_IRQHandler	DMA 通道 6 全局中断
7	0	可设置	DMA_Channel7_IRQHandler	DMA 通道 7 全局中断

© 2013 - 2024 杰发科技有限公司

中断向量号	优先级	优先级类型	名称	说明
8	0	可设置	DMA_Channel8_IRQHandler	DMA 通道 8 全局中断
9	0	可设置	DMA_Channel9_IRQHandler	DMA 通道 9 全局中断
10	0	可设置	DMA_Channel10_IRQHandler	DMA 通道 10 全局中断
11	0	可设置	DMA_Channel11_IRQHandler	DMA 通道 11 全局中断
12	0	可设置	DMA_Channel12_IRQHandler	DMA 通道 12 全局中断
13	0	可设置	DMA_Channel13_IRQHandler	DMA 通道 13 全局中断
14	0	可设置	DMA_Channel14_IRQHandler	DMA 通道 14 全局中断
15	0	可设置	DMA_Channel15_IRQHandler	DMA 通道 15 全局中断
16	0	可设置	PORTA_IRQHandler	PORTA 外部中断
17	0	可设置	PORTB_IRQHandler	PORTB 外部中断
18	0	可设置	PORTC_IRQHandler	PORTC 外部中断
19	0	可设置	PORTD_IRQHandler	PORTD 外部中断
20	0	可设置	PORTE_IRQHandler	PORTE 外部中断
21	0	可设置	UART0_IRQHandler	UART0 全局中断
22	0	可设置	UART1_IRQHandler	UART1 全局中断
23	0	可设置	UART2_IRQHandler	UART2 全局中断
24	0	可设置	UART3_IRQHandler	UART3 全局中断
25	-	-	-	保留
26	-	-	-	保留
27	0	可设置	SPI0_IRQHandler	SPI0 全局中断
28	0	可设置	SPI1_IRQHandler	SPI1 全局中断
29	0	可设置	SPI2_IRQHandler	SPI2 全局中断
30	-	-	-	保留
31	0	可设置	I2C0_IRQHandler	I2C0 全局中断
32	-	-	-	保留
33	0	可设置	EIO_IRQHandler	EIO 全局中断
34	0	可设置	CAN0_IRQHandler	CAN0 全局中断
35	0	可设置	CAN0_Wakeup_IRQHandler	CAN0 唤醒中断
36	0	可设置	CAN1_IRQHandler	CAN1 全局中断
37	0	可设置	CAN1_Wakeup_IRQHandler	CAN1 唤醒中断
38	0	可设置	CAN2_IRQHandler	CAN2 全局中断
39	0	可设置	CAN2_Wakeup_IRQHandler	CAN2 唤醒中断
40	0	可设置	CAN3_IRQHandler	CAN3 全局中断
41	0	可设置	CAN3_Wakeup_IRQHandler	CAN3 唤醒中断
42	-	-	-	保留
43	-	-	-	保留
44	-	-	-	保留
45	-	-	-	保留
46	0	可设置	PDT0_IRQHandler	PDT0 全局中断
47	0	可设置	PDT1_IRQHandler	PDT1 全局中断
48	0	可设置	ADC0_IRQHandler	ADC0 全局中断
49	0	可设置	ADC1_IRQHandler	ADC1 全局中断

© 2013 - 2024 杰发科技有限公司

中断向量号	优先级	优先级类型	名称	说明
50	0	可设置	ACMP_IRQHandler	ACMP 全局中断
51	0	可设置	WDG_IRQHandler	WDG 全局中断
52	0	可设置	EWDG_IRQHandler	EWDG 全局中断
53	0	可设置	MCM_IRQHandler	MCM 全局中断
54	0	可设置	LVD_IRQHandler	LVD 中断
55	0	可设置	SPM_IRQHandler	SPM 中断
56	0	可设置	RCM_IRQHandler	RCM 全局中断
57	0	可设置	PWM0_Overflow_IRQHandler	PWM0 溢出中断
58	0	可设置	PWM0_Channel_IRQHandler	PWM0 通道中断
59	0	可设置	PWM0_Fault_IRQHandler	PWM0 错误中断
60	0	可设置	PWM1_Overflow_IRQHandler	PWM1 溢出中断
61	0	可设置	PWM1_Channel_IRQHandler	PWM1 通道中断
62	0	可设置	PWM1_Fault_IRQHandler	PWM1 错误中断
63	0	可设置	PWM2_Overflow_IRQHandler	PWM2 溢出中断
64	0	可设置	PWM2_Channel_IRQHandler	PWM2 通道中断
65	0	可设置	PWM2_Fault_IRQHandler	PWM2 错误中断
66	0	可设置	PWM3_Overflow_IRQHandler	PWM3 溢出中断
67	0	可设置	PWM3_Channel_IRQHandler	PWM3 通道中断
68	0	可设置	PWM3_Fault_IRQHandler	PWM3 错误中断
69	0	可设置	PWM4_Overflow_IRQHandler	PWM4 溢出中断
70	0	可设置	PWM4_Channel_IRQHandler	PWM4 通道中断
71	0	可设置	PWM4_Fault_IRQHandler	PWM4 错误中断
72	0	可设置	PWM5_Overflow_IRQHandler	PWM5 溢出中断
73	0	可设置	PWM5_Channel_IRQHandler	PWM5 通道中断
74	0	可设置	PWM5_Fault_IRQHandler	PWM5 错误中断
75	-	-	-	保留
76	-	-	-	保留
77	-	-	-	保留
78	-	-	-	保留
79	-	-	-	保留
80	-	-	-	保留
81	0	可设置	RTC_IRQHandler	RTC 全局变量
82	0	可设置	PCT_IRQHandler	PCT 全局变量
83	0	可设置	TIMER_Channel0_IRQHandler	TIMER 通道 0 全局中断
84	0	可设置	TIMER_Channel1_IRQHandler	TIMER 通道 1 全局中断
85	0	可设置	TIMER_Channel2_IRQHandler	TIMER 通道 2 全局中断
86	0	可设置	TIMER_Channel3_IRQHandler	TIMER 通道 3 全局中断
87	0	可设置	CSE_IRQHandler	CSE 中断
88	0	可设置	Flash_ECC_IRQHandler	Flash_ECC 中断
89	0	可设置	Flash_IRQHandler	Flash 中断
90	0	可设置	Flash_Collision_IRQHandler	Flash 命令/读写冲突中断
91	0	可设置	ECC_1BIT_IRQHandler	ECC_1BIT 错误中断

© 2013 - 2024 杰发科技有限公司

中断向量号	优先级	优先级类型	名称	说明
92	0	可设置	ECC_2BIT_IRQHandler	ECC_2BIT 错误中断

3.3 NMI

NMI(Non-maskable interrupt)不可屏蔽中断，NVIC 中断请求与 GPIO 引脚 PD3 相连。当 PD3 引脚复用为 NMI 功能，并且输入低电平时，一直触发 NMI 中断。

3.4 Systick

Systick 系统滴答时钟， Systick 的时钟源始终是内核时钟 CORE_CLK。

3.5 UUID

UUID 共 128Bit，由随机数产生，可作为芯片唯一的识别标识。 用户可通过片内 Flash 读操作接口进行访问。 UUID 信息存放在 0x00201800~0x00201810 连续的空间(4*32Bit)。

3.6 启动配置

AC7840x 只能从片内 Flash 存储器启动，不可配置。

© 2013 - 2024 杰发科技有限公司

4 存储器

4.1 存储器组织

AC7840x 的数据字节在存储器中以小端格式存放。小端是指数据的低位保存在内存的低地址中，而数据的高位保存在内存的高地址中。例如，16bit 宽的数 0x1234 在小端格式 CPU 内存中的存放方式（假设从地址 0x4000 开始存放）为：

表 4-1 小端格式存放方式

内存地址	0x4000	0x4001
存放内容	0x34	0x12

4.2 存储器映射

程序存储器(P-Flash)、数据存储器(D-Flash)、寄存器和输入输出（I/O）接口统一编址在 4G 字节的线性地址空间里。

表 4-2 为 AC7840x 设备整个存储器空间的映射表，没有分配片上存储和外设的存储器区域，被称之为保留区。

表 4-2 存储器映射表

空间	起始地址	结束地址	说明	保护
Flash	0x0000_0000	0x000F_FFFF	P-Flash Main	MPU
	0x0010_0000	0x001F_FFFF	保留	保留
	0x0020_0000	0x0020_3FFF	P-Flash Info	MPU
	0x0020_4000	0x00FF_FFFF	保留	保留
	0x0100_0000	0x0101_FFFF	D-Flash Main	MPU
	0x0102_0000	0x0102_3FFF	D-Flash Info	MPU
	0x0102_4000	0x01FF_FFFF	保留	保留
	0x0200_0000	0x13FF_FFFF	保留	保留
	0x1400_0000	0x1400_0FFF	FlexRAM	MPU
	0x1400_1000	0x1400_107F	CSE_PRAM	MPU
	0x1400_1080	0x17FF_FFFF	保留	保留
	0x1800_0000	0x1BFF_FFFF	保留	保留
Code RAM	0x1C00_0000	0x1FFE_FFFF	保留	保留
	0x1FFF_0000	0x1FFF_FFFF	SRAM_L	MPU
	0x2000_0000	0x2000_EFFF	SRAM_U	MPU
	0x2000_F000	0x3FFF_FFFF	保留	保留
外设区域	0x4000_0000	0x400F_FFFF	PBR	外设桥
	0x4010_0000	0x5FFF_FFFF	保留	保留

© 2013 - 2024 杰发科技有限公司

空间	起始地址	结束地址	说明	保护
保留	0x6000_0000	0x9FFF_FFFF	保留	保留
保留	0xA000_FFFF	0xDFFF_FFFF	保留	保留
私有总线	0xE000_0000	0xE00F_FFFF	私有总线	M4 Core
保留	0xE010_0000	0xFFFF_FFFF	保留	保留

4.3 内置 SRAM

AC7840x 内置一个 128Kbyte 的静态 SRAM(其中包含了 4KB 的 FlexRAM)，它可以以字节、半字（16 位）或全字（32 位）访问。该静态 SRAM 分为 SRAM_L 和 SRAM_U 两个区（地址以 0x20000000 为分界点）。

更多细节请参考 17 静态存储器（SRAM）章节。

4.4 片内 Flash 存储器

高性能片内 Flash 模块的主要特性：

- P-Flash 最大支持 1MB（包括 2 个 bank 即 512KB+512KB）
- D-Flash 最大支持 128KB

Flash 控制器特性：

- Flash 编程/ 擦除操作
- 读/写保护
- 擦除及空白检查
- 缓存控制器用以提升读效率，最优效率为零等待

更多细节请参考 15 片内 Flash（Embedded Flash）章节。

© 2013 - 2024 杰发科技有限公司

5 安全概览（Safety Overview）

5.1 简介

AC7840x 是根据 ISO 26262 开发的，具有针对 ISO26262 ASIL-B 完整性级别的综合安全概念。

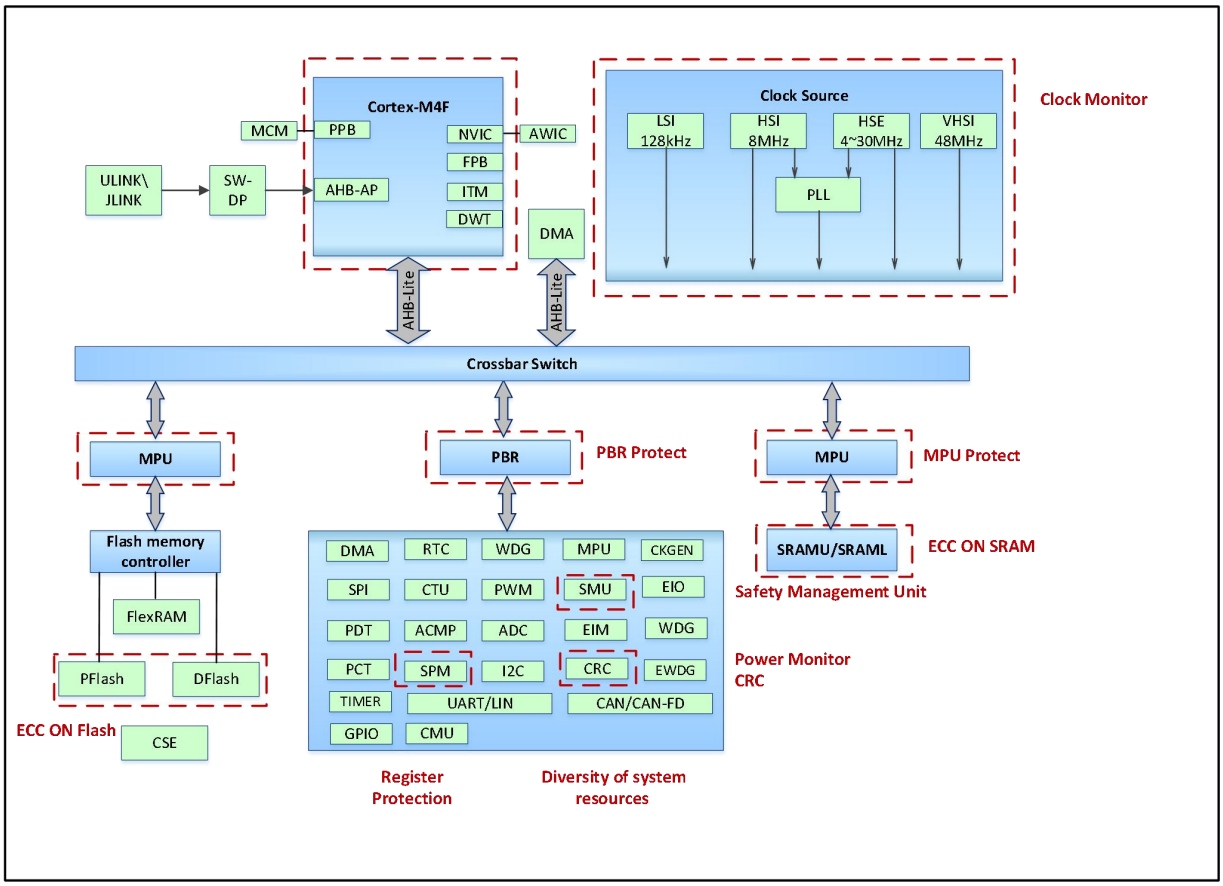


图 5-1 功能安全结构框图

5.2 安全描述

5.2.1 内核自测(SCST)

SCST(Cortex-M4 Structural Core Self Test) 是 AutoChips 提供的软件产品，用于检测 M4 内核中的硬件永久性故障。采用一组固定的操作数来比较 SCST 的执行结果，用户根据该结果判断内核自测是否成功。SCST 软件包是脱离上下文的安全元素，根据 ASIL-B 开发的。

© 2013 - 2024 杰发科技有限公司

5.2.2 访问保护

5.2.2.1 MPU

对于需要满足 ASIL-B 的应用程序，AC7840x 提供了系统内存保护单元 (System MPU) 用于访问保护。MPU 负责分配访问权限并确保只有授权的软件任务才可以访问其分配的资源。

- MPU 提供内存保护，它把内存分成了 8 个 Region 区域。用户可单独对这 8 个 Region 区域分配不同的访问权限以及分配每个 Region 覆盖的资源
- 对每个 Master (Core/Debugger/DMA)都可以单独分配不同的访问权限
- MPU 可以防止非安全 Master DMA 的非法访问

更多细节请参考 8 存储保护单元（MPU）章节。

5.2.2.2 外设桥(PBR)

外设桥(PBR)用于 Master(Core/Debugger/DMA)对外设寄存器的访问保护，以防止用户软件错误的修改外设寄存器而导致的功能异常。

更多细节请参考 7 外设桥（PBR）章节。

5.2.2.3 寄存器保护

AC7840x 支持对关键或者有安全功能需要的寄存器提供寄存器保护，例如：RCM、CKGEN、MCM、MPU、WDG 等。保护的方式有以下几种：

- 只能被特权模式访问
- 只能被 32 位访问
- 对寄存器进行 LOCK 锁住
- 需要特殊秘钥才能解锁寄存器的访问

更多细节请参考各个模块的寄存器定义章节。

5.2.3 SMU

安全管理单元（SMU）主要的作用是从硬件功能安全模块接收错误相关的信号，从而产生中断或复位请求。

更多细节请参考 14 安全管理单元(SMU) 章节。

© 2013 - 2024 杰发科技有限公司

5.2.4 ECC

SRAM 支持 ECC 功能，用于 SRAM 数据、地址、ECC 校验码的差错检测和修正，1-bit 错误可以纠正，2-bit 错误可以检测。

Flash 支持 ECC 功能，用于 Flash 数据、ECC 校验码的差错检测和修正，1-bit 错误可以纠正，2-bit 错误可以检测。

5.2.5 CRC

CRC 通过计算数据的 CRC 校验值并将其与之前计算好的 CRC 进行比较，可用于检测传输中数据或存储的数据是否被意外篡改。同时，CRC 模块支持 DMA 传输，用户可以使用 DMA 模块将其他外设寄存器值搬运至 CRC 模块来计算校验值，由此检测其他外设模块的寄存器值是否被意外篡改。

更多细节请参考 21 循环冗余校验模块（CRC） 章节。

5.2.6 时钟监控

时钟监控单元可以对下面的时钟进行监测：

- SPLL: 监测 SPLL 的时钟是否丢失，或超出上下阈值范围
- HSE: 监测 HSE 的时钟是否丢失，或超出上下阈值范围
- VHSI: 监测 VHSI 的时钟是否丢失，或超出上下阈值范围

更多细节请参考 11 时钟监控单元（CMU）章节。

5.2.7 电源监控

AC7840x 提供了电源监控(POR、LVR、LVD) 对电源的低电压检测，即当电源电压 VCC 电压过低时，芯片会发生复位。这样当 VCC 异常时，低电压检测可以让芯片处于一个复位的安全状态。

5.2.8 看门狗

看门狗提供了程序运行时间监控以及外部电路监控功能。AC7840x 提供两类看门狗：

- 看门狗模块（WDG）：它提供了一种安全特性，确保软件按照计划执行，且 CPU 不会陷入无限循环；
- 外部看门狗（EWDG）：它提供一个独立输出信号，该信号可用来关断外部电路起到保护作用。

更多细节请参考 22 看门狗模块（WDG） 和 23 外部看门狗 (EWDG) 章节。

© 2013 - 2024 杰发科技有限公司

5.2.9 系统资源的多样性

功能安全设计通常会在系统中进行冗余设计， AC7840x 提供多种系统资源来支持用户的冗余设计。

- 数字输入可以被复制以获得冗余的安全输入。
- 对至关重要的数字输出可以冗余地写入，或者与回读操作相结合。
- ADC 可以用过采样的方式来检测瞬时误差的影响。在测量功能安全相关的信号时，可以通过两个 ADC 同时测量同一个模拟信号得到冗余的转换结果。
- AC7840x 提供了 SPI、I2C、UART、EIO(EIO 可以模拟 UART/I2C/SPI/I2S/PWM)多样的通信模块来支持系统设计出于安全原因的通信通道冗余。

© 2013 - 2024 杰发科技有限公司

6 杂项控制器(MCM)

6.1 简介

MCM (Miscellaneous Control Module) 杂项控制模块主要包括了对 FPU 的控制，MBIST 的状态显示及 Cache 的异常状态显示。

6.2 特性

- 支持 FPU IDC/IXC/UFC/OFC/DZC/IOC 异常状态获取和中断处理。
- 支持 MBIST(Memory built-in self-test)结果状态显示。
- 支持 Cache 的 ECC 1bit 2bit 异常状态及错误地址显示。

6.3 功能描述

6.3.1 浮点运算单元 FPU 异常

- FPU 异常标志来源于 FPSCR 寄存器。 FPU 异常标志位不能通过操作 MCM 寄存器清除，可以通过修改栈帧中 FPCSR 的值来清除。下列错误的 FPU 运算会产生相应的异常标志：
  - 异常输入 (IDC)
  - 不精确结果 (IXC)
  - 下溢出(UFC)
  - 上溢出(OFC)
  - 除以零 (DZC)
  - 无效运算 (IOC)
- MCM 对 6 种使能了相应的 FPU 异常中断的异常标志位进行或运算，异常标志位置 1 将会触发 MCM 中断。

6.3.2 MBIST 异常状态

- MBIST(Memory built-in self-test)是一种有效的检测手段对于测试片内 flash，当芯片上电后，软件可以通过 MCM 寄存器获取 MBIST 的运行状态。

© 2013 - 2024 杰发科技有限公司

6.3.3 Cache

- 该缓存具有双向集关联缓存结构，Cache 总大小为 4 K 字节。缓存具有 32bit 地址数据路径以及 16 字节的行大小。缓存标签和数据存储使用单端口模式同步 RAM。

6.4 寄存器定义

MCM 寄存器只能在特权模式下访问，用户模式下访问触发 hardfault。

表 6-1 MCM 寄存器映射

MCM 基地址：0xE008000C

地址	名称	宽度	描述
MCM 基地址+0x4	MCM_MISCR	32	FPU 中断状态/控制寄存器
MCM 基地址+0x3F4	MCM_MLMDR0	32	存储描述寄存器 0
MCM 基地址+0x3F8	MCM_MBIST	32	MBIST 状态寄存器
MCM 基地址+0x404	MCM_LMPEIR	32	缓存 ECC 错误状态寄存器
MCM 基地址+0x408	MCM_LMFAR	32	缓存 ECC 错误地址寄存器
MCM 基地址+0x474	MCM_MNCR	32	NMI 控制寄存器

6.4.1 FPU 中断状态/控制寄存器(MCM_MISCR)

表 6-2 MCM_MISCR 寄存器

MCM_MISCR　　FPU 中断状态/控制寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	FIDCEN			FIXCEN	FUFCEN	FOFCEN	FDZCEN	FIOCEN								
访问	RW			RW	RW	RW	RW	RW								
复位值	0			0	0	0	0	0								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	FIDC			FIXC	FUFC	FOFC	FDZC	FIOC								
访问	RO			RO	RO	RO	RO	RO								
复位值	0			0	0	0	0	0								

字段	说明
31 FIDCEN	**FPU 异常输入中断使能** 0: 不使能 1: 使能
28 FIXCEN	**FPU 不精确结果中断使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 不使能 1: 使能
27 FUFCEN	**FPU 下溢中断使能** 0: 不使能 1: 使能
26 FOFCEN	**FPU 上溢中断使能** 0: 不使能 1: 使能
25 FDZCEN	**FPU 除以零中断使能** 0: 不使能 1: 使能
24 FIOCEN	**FPU 无效运算中断使能** 0: 不使能 1: 使能
15 FIDC	**FPU 异常输入标志位** 0: 无异常 1: 发生异常 需软件清除 FPSCR[IDC]，该位才能被清除
12 FIXC	**FPU 不精确结果标志位** 0: 无异常 1: 发生异常 需软件清除 FPSCR[IXC]，该位才能被清除
11 FUFC	**FPU 下溢标志位** 0: 无异常 1: 发生异常 需软件清除 FPSCR[UFC]，该位才能被清除
10 FOFC	**FPU 上溢标志位** 0: 无异常 1: 发生异常 需软件清除 FPSCR[OFC]，该位才能被清除
9 FDZC	**FPU 除以零标志位**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 无异常 1: 发生异常 需软件清除 FPSCR[DZC]，该位才能被清除
8 FIOC	**FPU 无效运算标志位** 0: 无异常 1: 发生异常 需软件清除 FPSCR[IOC]，该位才能被清除

6.4.2 存储描述寄存器 0(MCM_MLMDR0)

表 6-3 MCM_MLMDR0 寄存器

MCM_MLMDR0　　存储描述寄存器 0　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																LOCK
访问																R/W
复位值																0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													UREEN		LREEN	
访问													RW		RW	
复位值													0		0	

字段	说明
16 LOCK	**锁定此寄存器(MCM_MLMDR0)** 0: 锁定，本寄存器无法被写操作 1: 未锁定，本寄存器可被写操作 **说明：锁定后，只有复位才能清除**
3 UREEN	**SRAM_U 读 ECC 使能** 0: 不使能 1: 使能
1 LREEN	**SRAM_L 读 ECC 使能** 0: 不使能 1: 使能

© 2013 - 2024 杰发科技有限公司

6.4.3 MBIST 状态寄存器(MCM_MBIST)

表 6-4 MCM_MBIST 寄存器

MCM_MBIST **MBIST 状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															MBIST_FAIL	MBIST_DONE
访问															RO	RO
复位值															0	0

字段	说明
1 MBIST_FAIL	**MBIST 结果状态位** 0: MBIST 测试成功，当 MBIST_DONE 为 1 时 1: MBIST 测试失败，当 MBIST_DONE 为 1 时
0 MBIST_DONE	**MBIST 使能状态位** 0: 未使能 MBIST 功能 1: 已使能 MBIST 功能并完成测试

6.4.4 缓存 ECC 错误状态寄存器(MCM_LMPEIR)

表 6-5 MCM_LMPEIR 寄存器

MCM_LMPEIR **缓存 ECC 错误状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称								SRAM1_ECC[24:23]		SRAM0_ECC[22:21]		TAG_ECC[20:19]				
访问								W1C		W1C		W1C				
复位值								0	0	0	0	0	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																
访问																
复位值																

字段	说明
24:23	**缓存空间 RAM1 数据 ECC 错误**

© 2013 - 2024 杰发科技有限公司

字段	说明
SRAM1_ECC	
	01: 2bit ECC 错误
	10: 1bit ECC 错误
22:21 SRAM0_ECC	**缓存空间 RAM0 数据 ECC 错误**
	01: 2bit ECC 错误
	10: 1bit ECC 错误
20:19 TAG_ECC	**缓存标签 数据 ECC 错误**
	01: 2bit ECC 错误
	10: 1bit ECC 错误

6.4.5 缓存 ECC 错误地址寄存器(MCM_LMFAR)

表 6-6 MCM_LMFAR 寄存器

MCM_LMFAR **缓存 ECC 错误地址寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CEEA[31:16]															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CEEA[15:0]															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 CEEA	**缓存 ECC 错误地址**

6.4.6 NMI 控制寄存器(MCM_MNCR)

表 6-7 MCM_MNCR 寄存器

MCM_MNCR **NMI 控制寄存器** **复位值：0x00000000**

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																NMI_EN
访问																RW
复位值																0

字段	说明
0 NMI_EN	**NMI 使能** 0: 不使能 1: 使能 配置 NMI pinmux 后使能

© 2013 - 2024 杰发科技有限公司

7 外设桥（PBR）

7.1 简介

外设桥(Peripheral Bridge)：是连接主机与外设之间的桥梁，用于对外设提供访问许可保护。

7.2 特性

PBR 模块主要的性能包括：

- 3 个主机(Core, Debugger & DMA) 访问许可配置
  - 特权等级(Privilege level)
  - 受信任的写(Trusted for write)
  - 受信任的读(Trusted for read)
- 外设访问权限配置(PBR 寄存器要求主机必须为特权模式以及受信任访问)
  - 特权模式保护(Supervisor protect)
  - 写保护(Write protect)
  - 信任访问保护(Trusted protect)

© 2013 - 2024 杰发科技有限公司

7.3 结构框图

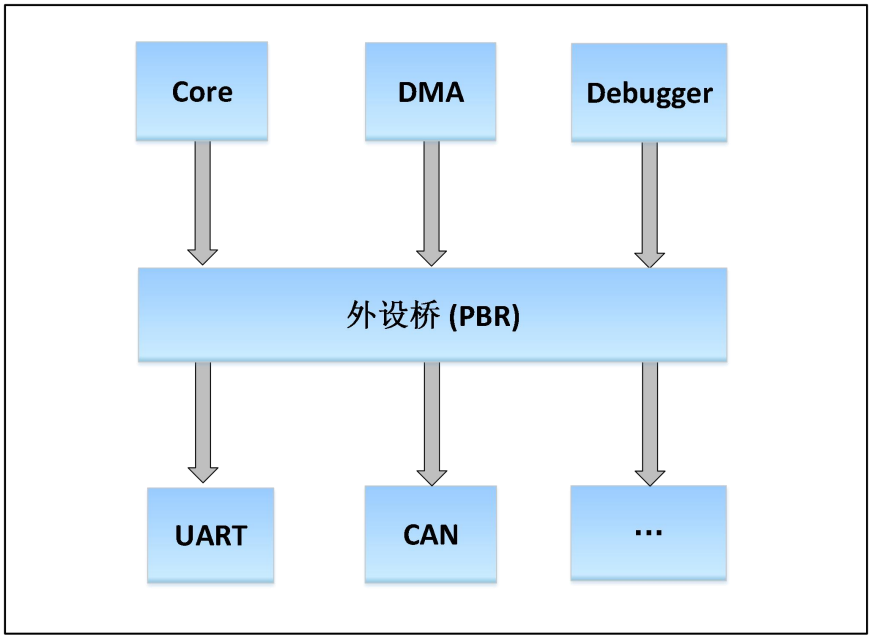


图 7-1 外设桥结构示意图

7.4 功能描述

7.4.1 主机非法访问

外设桥可用于防止主机非法访问，保护外设寄存器不被篡改。

- Core：CPU 非法读或写均会触发 Hardfault。
- DMA：DMA 非法读会报告源地址总线错误，DMA 非法写会报告目的地址总线错误。错误状态可通过查询 DMA_STATUS 获取。
- Debugger：通过 Memory 窗口非法读将显示问号；通过 Memory 窗口非法写无效。

7.4.2 调试模式

PBR 模块没有 debug 配置，在进入 debug 模式时可正常运行。

7.5 应用说明

PBR 寄存器包含主机权限配置寄存器和外设访问权限寄存器。可参考如下步骤配置 PBR 寄存器。

1. **配置外设访问权限寄存器：**

© 2013 - 2024 杰发科技有限公司

1.1. 读取外设访问权限寄存器

1.2. 修改对应外设的访问权限

1.3. 将修改后的值重新写入外设访问权限寄存器

2. 配置主机权限：



说明：

- 主机必须为特权模式以及受信任访问，PBR 寄存器才能配置使用。
- 主机权限配置寄存器的配置一定要放到最后，否则可能导致后续无法再配置外设权限访问寄存器。

2.1. 配置寄存器设置主机访问权限。

7.6 寄存器定义

PBR 寄存器要求主机必须为特权模式以及受信任访问。

表 7-1 PBR 寄存器映射

PBR 基地址：0x4008A000

地址	名称	宽度	描述
PBR 基地址+0x00	PBR_MPR_CORE	32	Core 权限配置寄存器
PBR 基地址+0x04	PBR_MPR_DEBUG	32	Debugger 权限配置寄存器
PBR 基地址+0x08	PBR_MPR_DMA	32	DMA 权限配置寄存器
PBR 基地址+0x10	PBR_PACRA	32	外设访问权限配置寄存器 A
PBR 基地址+0x14	PBR_PACRB	32	外设访问权限配置寄存器 B
PBR 基地址+0x20	PBR_PACRC	32	外设访问权限配置寄存器 C
PBR 基地址+0x24	PBR_PACRD	32	外设访问权限配置寄存器 D
PBR 基地址+0x28	PBR_PACRE	32	外设访问权限配置寄存器 E
PBR 基地址+0x30	PBR_PACRF	32	外设访问权限配置寄存器 F
PBR 基地址+0x34	PBR_PACRG	32	外设访问权限配置寄存器 G

7.6.1 Core 权限配置寄存器(PBR_MPR_CORE)

表 7-2 PBR_MPR_CORE 寄存器

PBR_MPR_CORE　　Core 权限配置寄存器　　复位值：

0x00000007

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														MTR	MTW	MPL
访问														RW	RW	RW
复位值														1	1	1

字段	说明
2 MTR	**Core 受信任的读** 0: 主机不受信任的读 1: 主机受信任的读
1 MTW	**Core 受信任的写** 0: 主机不受信任的写 1: 主机受信任的写
0 MPL	**Core 特权等级** 0: 强制主机为用户模式 1: 不强制(主机保持原有模式)

7.6.2 Debugger 权限配置寄存器(PBR_MPR_DEBUG)

表 7-3 PBR_MPR_DEBUG 寄存器

PBR_MPR_DEBUG　　Debugger 权限配置寄存器　　复位值：

0x00000007

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														MTR	MTW	MPL
访问														RW	RW	RW
复位值														1	1	1

字段	说明
2 MTR	**Debugger 受信任的读** 0: 主机不受信任的读

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 主机受信任的读
1 MTW	**Debugger 受信任的写** 0: 主机不受信任的写 1: 主机受信任的写
0 MPL	**Debugger 特权等级** 0: 强制主机为用户模式 1: 不强制(主机保持原有模式)

7.6.3 DMA 权限配置寄存器(PBR_MPR_DMA)

表 7-4 PBR_MPR_DMA 寄存器

PBR_MPR_DMA **DMA 权限配置寄存器** **复位值：0x00000007**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														MTR	MTW	MPL
访问														RW	RW	RW
复位值														1	1	1

字段	说明
2 MTR	**DMA 受信任的读** 0: 主机不受信任的读 1: 主机受信任的读
1 MTW	**DMA 受信任的写** 0: 主机不受信任的写 1: 主机受信任的写
0 MPL	**DMA 特权等级** 0: 强制主机为用户模式 1: 不强制(主机保持原有模式)

© 2013 - 2024 杰发科技有限公司

7.6.4 外设访问权限配置寄存器 A(PBR_PACRA)

表 7-5 PBR_PACRA 寄存器

PBR_PACRA 外设访问权限配置 A 复位值：**0x00444444**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称										PWM5_SP	PWM5_WP	PWM5_TP		PWM4_SP	PWM4_WP	PWM4_TP
访问										RW	RW	RW		RW	RW	RW
复位值										1	0	0		1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PWM3_SP	PWM3_WP	PWM3_TP		PWM2_SP	PWM2_WP	PWM2_TP		PWM1_SP	PWM1_WP	PWM1_TP		PWM0_SP	PWM0_WP	PWM0_TP
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值		1	0	0		1	0	0		1	0	0		1	0	0

字段	说明
22 PWM5_SP	**PWM5 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
21 PWM5_WP	**PWM5 写保护配置** 0: 允许写访问 1: 写保护
20 PWM5_TP	**PWM5 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
18 PWM4_SP	**PWM4 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
17 PWM4_WP	**PWM4 写保护配置** 0: 允许写访问 1: 写保护
16 PWM4_TP	**PWM4 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

© 2013 - 2024 杰发科技有限公司

字段	说明
14 PWM3_SP	**PWM3 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
13 PWM3_WP	**PWM3 写保护配置** 0: 允许写访问 1: 写保护
12 PWM3_TP	**PWM3 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 PWM2_SP	**PWM2 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 PWM2_WP	**PWM2 写保护配置** 0: 允许写访问 1: 写保护
8 PWM2_TP	**PWM2 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
6 PWM1_SP	**PWM1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
5 PWM1_WP	**PWM1 写保护配置** 0: 允许写访问 1: 写保护
4 PWM1_TP	**PWM1 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 PWM0_SP	**PWM0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
1 PWM0_WP	**PWM0 写保护配置** 0: 允许写访问 1: 写保护
0	**PWM0 受信任访问配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
PWM0_TP	0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

7.6.5 外设访问权限配置寄存器 B(PBR_PACRB)

表 7-6 PBR_PACRB 寄存器

PBR_PACRB **外设访问权限配置 B** **复位值：**
0x04444444

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称						SMU_SP	SMU_WP	SMU_TP		EIM_SP	EIM_WP	EIM_TP		MPU_SP	MPU_WP	MPU_TP
访问						RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值						1	0	0		1	0	0		1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PDT1_SP	PDT1_WP	PDT1_TP		PDT0_SP	PDT0_WP	PDT0_TP		GPIO_SP	GPIO_WP	GPIO_TP		CRC_SP	CRC_WP	CRC_TP
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值		1	0	0		1	0	0		1	0	0		1	0	0

字段	说明
26 SMU_SP	**SMU 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
25 SMU_WP	**SMU 写保护配置** 0: 允许写访问 1: 写保护
24 SMU_TP	**SMU 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
22 EIM_SP	**EIM 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
21 EIM_WP	**EIM 写保护配置** 0: 允许写访问 1: 写保护

© 2013 - 2024 杰发科技有限公司

字段	说明
20 EIM_TP	**EIM 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
18 MPU_SP	**MPU 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
17 MPU_WP	**MPU 写保护配置** 0: 允许写访问 1: 写保护
16 MPU_TP	**MPU 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
14 PDT1_SP	**PDT1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
13 PDT1_WP	**PDT1 写保护配置** 0: 允许写访问 1: 写保护
12 PDT1_TP	**PDT1 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 PDT0_SP	**PDT0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 PDT0_WP	**PDT0 写保护配置** 0: 允许写访问 1: 写保护
8 PDT0_TP	**PDT0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
6 GPIO_SP	**GPIO 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
5	**GPIO 写保护配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
GPIO_WP	0: 允许写访问 1: 写保护
4 GPIO_TP	**GPIO 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 CRC_SP	**CRC 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
1 CRC_WP	**CRC 写保护配置** 0: 允许写访问 1: 写保护
0 CRC_TP	**CRC 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

7.6.6 外设访问权限配置寄存器 C(PBR_PACRC)

表 7-7 PBR_PACRC 寄存器

PBR_PACRC **外设访问权限配置 C** **复位值：**

0x44444404

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		CAN0_SP	CAN0_WP	CAN0_TP		CTU_SP	CTU_WP	CTU_TP		ACMP_SP	ACMP_WP	ACMP_TP		ADC1_SP	ADC1_WP	ADC1_TP
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值		1	0	0		1	0	0		1	0	0		1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		ADC0_SP	ADC0_WP	ADC0_TP		FLASH_SP	FLASH_WP	FLASH_TP						CLK_SP	CLK_WP	CLK_TP
访问		RW	RW	RW		RW	RW	RW						RW	RW	RW
复位值		1	0	0		1	0	0						1	0	0

字段	说明
30 CAN0_SP	**CAN0 特权模式配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 无特权模式访问要求 1: 有特权模式访问要求
29 CAN0_WP	**CAN0 写保护配置** 0: 允许写访问 1: 写保护
28 CAN0_TP	**CAN0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
26 CTU_SP	**CTU 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
25 CTU_WP	**CTU 写保护配置** 0: 允许写访问 1: 写保护
24 CTU_TP	**CTU 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
22 ACMP_SP	**ACMP 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
21 ACMP_WP	**ACMP 写保护配置** 0: 允许写访问 1: 写保护
20 ACMP_TP	**ACMP 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
18 ADC1_SP	**ADC1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
17 ADC1_WP	**ADC1 写保护配置** 0: 允许写访问 1: 写保护
16 ADC1_TP	**ADC1 受信任访问配置** 0: 允许不受信任的主机访问

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 不允许不受信任的主机访问
14 ADC0_SP	**ADC0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
13 ADC0_WP	**ADC0 写保护配置** 0: 允许写访问 1: 写保护
12 ADC0_TP	**ADC0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 FLASH_SP	**FLASH 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 FLASH_WP	**FLASH 写保护配置** 0: 允许写访问 1: 写保护
8 FLASH_TP	**FLASH 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 CLK_SP	**CLOCK 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
1 CLK_WP	**CLOCK 写保护配置** 0: 允许写访问 1: 写保护
0 CLK_TP	**CLOCK 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

© 2013 - 2024 杰发科技有限公司

7.6.7 外设访问权限配置寄存器 D(PBR_PACRD)

表 7-8 PBR_PACRD 寄存器

PBR_PACRD 外设访问权限配置 D 复位值：**0x44400444**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		EIO_SP	EIO_WP	EIO_TP		RTC_SP	RTC_WP	RTC_TP		SPM_SP	SPM_WP	SPM_TP				
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW				
复位值		1	0	0		1	0	0		1	0	0				
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						CAN3_SP	CAN3_WP	CAN3_TP		CAN2_SP	CAN2_WP	CAN2_TP		CAN1_SP	CAN1_WP	CAN1_TP
访问						RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值						1	0	0		1	0	0		1	0	0

字段	说明
30 EIO_SP	**EIO 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
29 EIO_WP	**EIO 写保护配置** 0: 允许写访问 1: 写保护
28 EIO_TP	**EIO 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
26 RTC_SP	**RTC 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
25 RTC_WP	**RTC 写保护配置** 0: 允许写访问 1: 写保护
24 RTC_TP	**RTC 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
22 SPM_SP	**SPM 特权模式配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
21 SPM_WP	**SPM 写保护配置** 0: 允许写访问 1: 写保护
20 SPM_TP	**SPM 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 CAN3_SP	**CAN3 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 CAN3_WP	**CAN3 写保护配置** 0: 允许写访问 1: 写保护
8 CAN3_TP	**CAN3 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
6 CAN2_SP	**CAN2 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
5 CAN2_WP	**CAN2 写保护配置** 0: 允许写访问 1: 写保护
4 CAN2_TP	**CAN2 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 CAN1_SP	**CAN1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
1 CAN1_WP	**CAN1 写保护配置** 0: 允许写访问 1: 写保护
0 CAN1_TP	**CAN1 受信任访问配置** 0: 允许不受信任的主机访问

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 不允许不受信任的主机访问

7.6.8 外设访问权限配置寄存器 E(PBR_PACRE)

表 7-9 PBR_PACRE 寄存器

PBR_PACRE **外设访问权限配置 E** **复位值：**

0x04044444

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称						I2C0_SP	I2C0_WP	I2C0_TP						SPI2_SP	SPI2_WP	SPI2_TP
访问						RW	RW	RW						RW	RW	RW
复位值						1	0	0						1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SPI1_SP	SPI1_WP	SPI1_TP		SPI0_SP	SPI0_WP	SPI0_TP		EWDG_SP	EWDG_WP	EWDG_TP		WDG_SP	WDG_WP	WDG_TP
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值		1	0	0		1	0	0		1	0	0		1	0	0

字段	说明
26 I2C0_SP	**I2C0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
25 I2C0_WP	**I2C0 写保护配置** 0: 允许写访问 1: 写保护
24 I2C0_TP	**I2C0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
18 SPI2SP	**SPI2 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
17 SPI2_WP	**SPI2 写保护配置** 0: 允许写访问 1: 写保护
16 SPI2_TP	**SPI2 受信任访问配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
14 SPI1_SP	**SPI1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
13 SPI1_WP	**SPI1 写保护配置** 0: 允许写访问 1: 写保护
12 SPI1_TP	**SPI1 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 SPI0_SP	**SPI0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 SPI0_WP	**SPI0 写保护配置** 0: 允许写访问 1: 写保护
8 SPI0_TP	**SPI0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
6 EWDG_SP	**EWDG 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
5 EWDG_WP	**EWDG 写保护配置** 0: 允许写访问 1: 写保护
4 EWDG_TP	**EWDG 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 WDG_SP	**WDG 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
1 WDG_WP	**WDG 写保护配置** 0: 允许写访问

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 写保护
0 WDG_TP	**WDG 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

7.6.9 外设访问权限配置寄存器 F(PBR_PACRF)

表 7-10 PBR_PACRF 寄存器

PBR_PACRF **外设访问权限配置 F** **复位值：0x00444444**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称										UART3_SP	UART3_WP	UART3_TP		UART2_SP	UART2_WP	UART2_TP
访问										RW	RW	RW		RW	RW	RW
复位值										1	0	0		1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		UART1_SP	UART1_WP	UART1_TP		UART0_SP	UART0_WP	UART0_TP		DMA_SP	DMA_WP	DMA_TP		TIMER_SP	TIMER_WP	TIMER_TP
访问		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW
复位值		1	0	0		1	0	0		1	0	0		1	0	0

字段	说明
22 UART3_SP	**UART3 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
21 UART3_WP	**UART3 写保护配置** 0: 允许写访问 1: 写保护
20 UART3_TP	**UART3 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
18 UART2_SP	**UART2 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
17	**UART2 写保护配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
UART2_WP	0: 允许写访问 1: 写保护
16 UART2_TP	**UART2 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
14 UART1_SP	**UART1 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
13 UART1_WP	**UART1 写保护配置** 0: 允许写访问 1: 写保护
12 UART1_TP	**UART1 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
10 UART0_SP	**UART0 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
9 UART0_WP	**UART0 写保护配置** 0: 允许写访问 1: 写保护
8 UART0_TP	**UART0 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
6 DMA_SP	**DMA 特权模式配置** 0: 无特权模式访问要求 1: 有特权模式访问要求
5 DMA_WP	**DMA 写保护配置** 0: 允许写访问 1: 写保护
4 DMA_TP	**DMA 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 TIMER_SP	**TIMER 特权模式配置**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 无特权模式访问要求 1: 有特权模式访问要求
1 TIMER_WP	**TIMER 写保护配置** 0: 允许写访问 1: 写保护
0 TIMER_TP	**TIMER 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

7.6.10 外设访问权限配置寄存器 G(PBR_PACRG)

表 7-11 PBR_PACRG 寄存器

PBR_PACRG **外设访问权限配置 G** **复位值：**

0x00000044

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										CMU_SP	CMU_WP	CMU_TP		PCT_SP	PCT_WP	PCT_TP
访问										RW	RW	RW		RW	RW	RW
复位值										1	0	0		1	0	0

字段	说明
6 CMU_SP	**CMU 特权模式配置** 0: 无特权模式读访问要求，有写访问要求 1: 有特权模式访问要求
5 CMU_WP	**CMU 写保护配置** 0: 允许写访问 1: 写保护
4 CMU_TP	**CMU 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问
2 PCT_SP	**PCT 特权模式配置** 0: 无特权模式访问要求

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 有特权模式访问要求
1 PCT_WP	**PCT 写保护配置** 0: 允许写访问 1: 写保护
0 PCT_TP	**PCT 受信任访问配置** 0: 允许不受信任的主机访问 1: 不允许不受信任的主机访问

© 2013 - 2024 杰发科技有限公司

8 存储保护单元（MPU）

8.1 简介

MPU (Memory Protect Unit)，内存保护单元作为 MCU 的安全机制之一，用于提供内存区域保护功能。AC7840x 的 MPU 模块并未使用 ARM 官方版本仅支持对 CPU 内核存储器访问权限管理的 MPU IP，而是使用本司自己设计的系统 MPU IP，以控制/保护总线矩阵主机(Master)对存储器映射地址区域的访问权限。可以对每个 Crossbar Master(CPU 内核、Debugger 调试器接口、DMA)都可以配置其对每个保护存储器区域的不同访问权限。

文档中，术语 MPU 是指 AC7840x 的系统 MPU。

8.2 特性

- 8 个可编程保护区域 (Region 0 ~ Region 7)。
- 每个 Region 区域的大小：32 Bytes ~ 4G Bytes，遵循 32 字节对齐原则。
- 主机(Master) Core/Debugger/DMA 可以分别配置为特权模式和用户模式下是否可以读(R)、写(W)、执行(X)的属性，DMA 访问固定为特权模式。
- 权限配置复用寄存器 (RGD0_AAC ~ RGD7_AAC)。
- 对于重叠区域(Overlapping)，允许权限叠加。
- 若总线主机(Master)访问未命中的存储器映射区域，或者该访问违反存储器映射区域的访问权限。则会发现访问错误，并返回错误信息 。
- 每个从机(Slave)端口提供错误信息寄存器，以捕获上一次访问错误的地址、属性和其他信息。
- 总线访问 PID Hit 命中功能，每个 Region 可单独设置 PID 编号/掩码/PID 使能位。
- 每个 Region 拥有单独的使能/关闭控制位。
- 全局的 MPU 使能/关闭控制位。

8.3 功能描述

8.3.1 MPU 主机/从机端口分配

主机(Master)对从机(Salve)的访问受 MPU 的保护，MPU 主机端口分配以及访问类型见表 8-1，MPU Slave 端口分配见表 8-2。

© 2013 - 2024 杰发科技有限公司

表 8-1 主机访问分配

Master ID	总线 Master	可能的访问类型							是否支持 PID
		用户	特权	数据	指令	读(R)	写(W)	执行(X)	
0	Core	√	√	√	√	√	√	√	√
1	Debugger	√	√	√	√	√	√	√	√
2	DMA	×	√	√	×	√	√	×	×

说明：
DMA 访问固定为特权模式，没有指令访问和执行，也不支持 PID 。但 DMA 访问总是 PID 命中

表 8-2 从机端口分配

MPU 从机端口分配	描述
MPU slave port 0	Flash Controller
MPU slave port 1	SRAM_L
MPU slave port 2	SRAM_U

说明：
数据是通过 FLASH 控制器写入 FLASH，所以不受 MPU 保护。也就是说 MPU 不支持 FLASH 写保护，可以支持 FLASH 的读保护。

8.3.2 命中认定

8.3.2.1 区域命中认定(Region Hit)

RGDn_WORD0[S_ADDR]指定起始地址，RGDn_WORD1[E_ADDR]指定结束地址，RGDn_WORD3[RGD_EN]是使能位；如果当前总线主机(Master)访问的地址 addr[31:5]满足以下逻辑运算时，称作区域命中(region_hit)：

Region_Hit= ((addr[31:5] >= RGDn_WORD0[S_ADDR]) && (addr[31:5] <=

说明：
MPU 不会检查 E_ADDR ≤S_ADDR 的情况。

RGDn_WORD1[E_ADDR])) && RGDn_WORD3[RGD_EN]

© 2013 - 2024 杰发科技有限公司

8.3.2.2 PID 命中认定(PID Hit)

MPU 还将检查 PID 和 PIDMASK 比特配置的可选的处理识别号(PID)与当前发起访问的总线主机(Master)的 MPID 是否一致，这一过程称作 PID 命中(pid_hit)，其具体检查控制逻辑如下(对于没有 PID 的主机， MPU 默认其为 PID 命中成功)：

Pid_Hit = ~RGDn_WORD2[MxPID_EN] || ((MxPID|RGDn_WORD3[RGD_PIDMASK]) == (RGDn_WORD3[RGD_PID] || RGDn_WORD3[RGD_PIDMASK]))

8.3.3 违规访问

不同的主机违规访问导致的结果不一样，如下：

- Core：Core 非法读或写均会触发 Hardfault。
- DMA：DMA 非法读会报告源地址错误，DMA 非法写会报告目的地址错误。
- Debugger：通过仿真器 Memory 窗口非法读将显示问号；通过 Memory 窗口非法写无效。

8.3.3.1 未命中违规

当区域命中(Region_Hit)和 PID 命中(Pid_Hit)都成功时，这次主机(Master)访问命中(hit)成功。否则命中(Hit)失败，此次访问终止，触发总线错误，并产生错误信息（见寄存器 EAR0-EAR2/ EDR0-EDR2)。

8.3.3.2 权限违规

在访问命中(hit)成功的时候，MPU 的控制逻辑同时按照寄存器(RGD0_WORD2-RGD7_WORD2/ RGD0_AAC-RGD7_AAC)定义的各区域访问权限对当前访问进行权限认定，如果违反权限，则此次访问终止，触发总线错误，并产生错误信息（见寄存器 EAR0-EAR2/ EDR0-EDR2)。

8.3.4 Region 0 的配置限制

- RGD0_WORD0 寄存器默认为 0x0000_0000，Core 与 DMA 写 RGD0_WORD0 会触发总线错误，Debugger 可读可写。
- RGD0_WORD1 寄存器默认为 0xFFFF_FFFF， Core 与 DMA 写 RGD0_WORD1 会触发总线错误，Debugger 可读可写。
- RGD0_WORD2 寄存器默认为 0x0003_FFFF， Core 与 DMA 写 RGD0_WORD2 中的 M1UX/M1UR/M1UW/M1SX/M1SR/M1SW/M1PID_EN 不会生效，Debugger 可读可写。

上述配置限制的设计是为了保证 Debugger 总是可以对存储器映射区域进行访问。

© 2013 - 2024 杰发科技有限公司

8.3.5 重叠区域(Overlapping)

对于重叠区域(Overlapping)，允许权限叠加(例如：Region 0 可读，Region 1 可写，Region 0 与 Region 1 的重叠区域可读可写)。

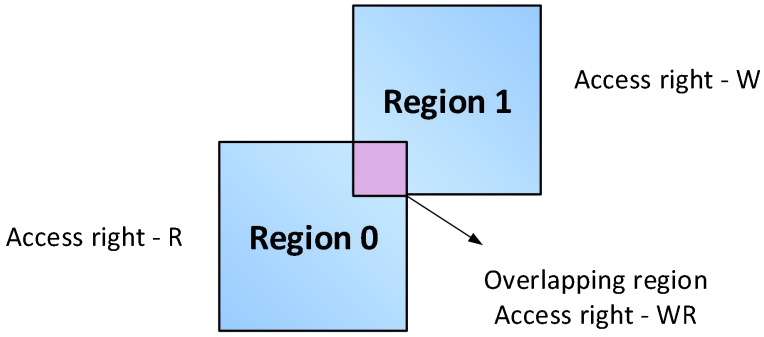


图 8-1 重叠区域权限示意图

8.3.6 权限配置复用寄存器

RGDn_AAC 为 RGDn_WORD2 的复用寄存器，即：配置 RGDn_WORD2 寄存器会同步到 RGDn_AAC，配置 RGDn_AAC 寄存器也会同步到 RGDn_WORD2 寄存器。二者的区别为:配置 RGDn_WORD2 硬件会自动清除相应的 RGD_EN 使能位，而配置 RGDn_AAC 则不会影响。用户可以根据需要选择 RGDn_WORD2 或者 RGDn_AAC 去配置 Region 的权限。

8.4 应用说明

8.4.1 编程指南

1. 所有需要访问的地址区域都需要配置，否则会因为命中规则，导致未配置的地址区域不可访问。
2. 由于 Region0 的配置限制，不能使用 Region 0 去保护用户区域，其它 Region 可以任意选择。
3. 需要修改 Region 0 的权限，否则会由于重叠区域的规则，所有区域可读可写可执行。
4. 配置访问权限时，用户可以选择 权限配置寄存器(MPU_RGDn_WORD2) 或者 权限配置复用寄存器(MPU_RGDn_AAC) 其中一个来进行权限配置。
5. 地址需保证 32 字节对齐。
6. 如果用户错误的保护了代码执行区域，则可能会触发内核死锁 lockup。

© 2013 - 2024 杰发科技有限公司

配置流程：

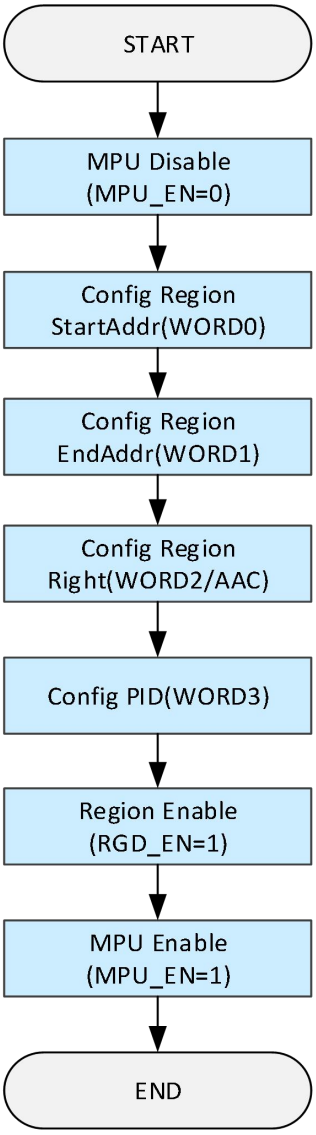


图 8-2 MPU 编程指南

8.5 寄存器定义

MPU 寄存器只能在特权模式下访问，用户模式下访问触发 hardfault。

MPU 寄存器只支持 32 位访问。

Core/DMA 写 RGD0_WORD0/WORD1/WORD3 会触发总线错误。

表 8-3 存储器保护单元（MPU）寄存器映射

MPU 基地址：0x40087000

地址	名称	宽度	描述

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
MPU 基地址+0x00	MPU_CESR	32	控制/错误状态寄存器
MPU 基地址+0x04~0x0C	MPU_EAR0-MPU_EAR2	32	错误地址寄存器，Slave 端口 n
MPU 基地址+0x10~0x18	MPU_EDR0-MPU_EDR2	32	错误信息寄存器，Slave 端口 n
MPU 基地址+0x1C	MPU_MPID	32	Master PID 寄存器
MPU 基地址+0x20~3C	MPU_RGD0_WORD0-MPU_RGD7_WORD0	32	起始地址寄存器
MPU 基地址+0x40~5C	MPU_RGD0_WORD1-MPU_RGD7_WORD1	32	结束地址寄存器
MPU 基地址+0x60~7C	MPU_RGD0_WORD2-MPU_RGD7_WORD2	32	权限配置寄存器
MPU 基地址+0x80~9C	MPU_RGD0_WORD3-MPU_RGD7_WORD3	32	PID/Region 使能配置寄存器
MPU 基地址+0x400~41C	MPU_RGD0_AAC-MPU_RGD7_AAC	32	权限配置复用寄存器

8.5.1 控制/错误状态寄存器(MPU_CESR)

表 8-4 MPU_CESR 寄存器

MPU_CESR **控制/错误状态寄存器** **复位值：**
0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SLAVE2	SLAVE1	SLAVE0													
访问	R/W 1C	R/W 1C	R/W 1C													
复位值	0	0	0													
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																MPU_EN
访问																RW
复位值																1

字段	说明
31 SLAVE2	**错误标志位 2** 0: 没有发生错误 1: 访问 Salve Port 2 SRAM_U 发生错误 写“1”清除该标志位
30 SLAVE1	**错误标志位 1** 0: 没有发生错误

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 访问 Salve Port 1 SRAM_L 发生错误 写“1”清除该标志位
29 SLAVE0	**错误标志位 0** 0: 没有发生错误 1: 访问 Salve Port 0 Flash Controller 发生错误 写“1”清除该标志位
0 MPU_EN	**MPU 全局使能** 0: MPU 不使能 1: MPU 使能

8.5.2 错误地址寄存器，Slave n(MPU_EAR0-MPU_EAR2)

表 8-5 MPU_EARn 寄存器

MPU_EARn **错误地址寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SLAVEn ERROR ADDR															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SLAVEn ERROR ADDR															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 SLAVEn ERROR ADDR	**错误地址** 发生访问错误的地址

© 2013 - 2024 杰发科技有限公司

8.5.3 错误信息寄存器，Slave n(MPU_EDR0-MPU_EDR2)

表 8-6 MPU_EDRn 寄存器

MPU_EDRn 错误信息寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	EFLAG								EACD							
访问	R/W1C								RO							
复位值	0								0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	EPID								EMN			EATTR				ERW
访问	RO								RO			RO				R
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31 EFLAG	**错误标志位** 0: 没有产生错误 1: 发生访问错误 写“1”清除该标志位
23:16 EACD	**错误访问的 Region 信息** 0-255: 指示发生错误访问的 Region 例如：bit16 为 1，bit17 为 1 表示发生错误访问的是 Region 0 和 Region 1
15:8 EPID	**错误访问的 Master PID 信息** 0-255: 指示发生错误访问的 Master PID
7:5 EMN	**错误访问的 Master ID 信息** 001b: Core 010b: Debugger 100b: DMA
4:1 EATTR	**发生错误访问的属性** 0001b: 用户模式指令访问 0010b: 用户模式数据访问 0100b: 特权模式指令访问 1000b: 特权模式数据访问
0 ERW	**错误访问类型** 0: 读 1: 写

© 2013 - 2024 杰发科技有限公司

8.5.4 MASTER PID 配置寄存器(MPU_MPID)

表 8-7 MPU_MPID 寄存器

MPU_MPID **MASTER PID 配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	M1PID								M0PID							
访问	RW								RW							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15:8 M1PID	**Debugger PID** 0-255: Debugger PID
7:0 M0PID	**Core PID** 0-255: Core PID **说明：Master DMA 没有 PID，即 DMA 访问时，PID 总是命中**

8.5.5 起始地址寄存器(MPU_RGD0_WORD0-MPU_RGD7_WORD0)

表 8-8 MPU_RGDn_WORD0 寄存器

MPU_RGDn_WORD0 **起始地址寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	S_ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	S_ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:5 S_ADDR	**起始地址** 保护区域的起始地址

© 2013 - 2024 杰发科技有限公司

字段	说明
	为了保证 32 字节地址对齐，Bit4-Bit0 固定为 0，不可配置

8.5.6 结束地址寄存器(MPU_RGD0_WORD1)

表 8-9 MPU_RGD0_WORD1 寄存器

MPU_RGD0_WORD1　　结束地址寄存器　　复位值：0xFFFFFFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	E_ADDR															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	E_ADDR															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
31:5 E_ADDR	**结束地址** 保护区域的结束地址 为了保证 32 字节地址对齐，Bit4-Bit0 固定为 1，不可配置

8.5.7 结束地址寄存器(MPU_RGD1_WORD1-MPU_RGD7_WORD1)

表 8-10 MPU_RGDn_WORD0 寄存器

MPU_RGDn_WORD0　　结束地址寄存器　　复位值：0x0000001F

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	E_ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	E_ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

字段	说明
31:5 E_ADDR	**结束地址** 保护区域的结束地址

© 2013 - 2024 杰发科技有限公司

字段	说明
	为了保证 32 字节地址对齐，Bit4-Bit0 固定为 1，不可配置

8.5.8 权限配置寄存器(MPU_RGD0_WORD2)

表 8-11 MPU_RGD0_WORD2 寄存器

MPU_RGD0_WORD2　　权限配置寄存器　　复位值：0x0003FFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	M0PID_EN	M1PID_EN													M2SW	M2SR
访问	RW	RW													RW	RW
复位值	0	0													1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	M2SX				M1SW	M1SR	M1SX	M1UW	M1UR	M1UX	M0SW	M0SR	M0SX	M0UW	M0UR	M0UX
访问	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	1				1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
31 M0PID_EN	**Master Core PID 使能** 0: 不使能 1: 使能
30 M1PID_EN	**Master Debugger PID 使能** 0: 不使能 1: 使能 **说明：该位只能被 Master Debugger 修改**
17 M2SW	**Master DMA 特权模式写访问** 0: 禁止 1: 允许
16 M2SR	**Master DMA 特权模式读访问** 0: 禁止 1: 允许
15 M2SX	**Master DMA 特权模式可执行访问** 0: 禁止 1: 允许
11	**Master Debugger 特权模式写访问**

© 2013 - 2024 杰发科技有限公司

字段	说明
M1SW	
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
10 M1SR	**Master Debugger 特权模式读访问**
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
9 M1SX	**Master Debugger 特权模式可执行访问**
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
8 M1UW	**Master Debugger 用户模式写访问**
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
7 M1UR	**Master Debugger 用户模式读访问**
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
6 M1UX	**Master Debugger 用户模式可执行访问**
	0: 禁止
	1: 允许
	说明：该位只能被 Master Debugger 修改
5 M0SW	**Master Core 特权模式写访问**
	0: 禁止
	1: 允许
4 M0SR	**Master Core 特权模式读访问**
	0: 禁止
	1: 允许
3 M0SX	**Master Core 特权模式可执行访问**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 禁止 1: 允许
2 M0UW	**Master Core 用户模式写访问** 0: 禁止 1: 允许
1 M0UR	**Master Core 用户模式读访问** 0: 禁止 1: 允许
0 M0UX	**Master Core 用户模式可执行访问** 0: 禁止 1: 允许

8.5.9 权限配置寄存器(MPU_RGD1_WORD2-MPU_RGD7_WORD2)

表 8- 12 MPU_RGDn_WORD2 寄存器

MPU_RGDn_WORD2　　权限配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	M0PID_EN	M1PID_EN													M2SW	M2SR
访问	RW	RW													RW	RW
复位值	0	0													0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	M2SX				M1SW	M1SR	M1SX	M1UW	M1UR	M1UX	M0SW	M0SR	M0SX	M0UW	M0UR	M0UX
访问	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0				0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31 M0PID_EN	**Master Core PID 使能** 0: 不使能 1: 使能
30 M1PID_EN	**Master Debugger PID 使能** 0: 不使能 1: 使能
17 M2SW	**Master DMA 特权模式写访问**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 禁止 1: 允许
16 M2SR	**Master DMA 特权模式读访问** 0: 禁止 1: 允许
15 M2SX	**Master DMA 特权模式可执行访问** 0: 禁止 1: 允许
11 M1SW	**Master Debugger 特权模式写访问** 0: 禁止 1: 允许
10 M1SR	**Master Debugger 特权模式读访问** 0: 禁止 1: 允许
9 M1SX	**Master Debugger 特权模式可执行访问** 0: 禁止 1: 允许
8 M1UW	**Master Debugger 用户模式写访问** 0: 禁止 1: 允许
7 M1UR	**Master Debugger 用户模式读访问** 0: 禁止 1: 允许
6 M1UX	**Master Debugger 用户模式可执行访问** 0: 禁止 1: 允许
5 M0SW	**Master Core 特权模式写访问** 0: 禁止 1: 允许
4 M0SR	**Master Core 特权模式读访问** 0: 禁止 1: 允许
3 M0SX	**Master Core 特权模式可执行访问** 0: 禁止

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 允许
2 M0UW	**Master Core 用户模式写访问** 0: 禁止 1: 允许
1 M0UR	**Master Core 用户模式读访问** 0: 禁止 1: 允许
0 M0UX	**Master Core 用户模式可执行访问** 0: 禁止 1: 允许

8.5.10 RGD PID/使能配置寄存器(MPU_RGD0_WORD3)

表 8-13 MPU_RGD0_WORD3 寄存器

MPU_RGD0_WORD3　　RGD PID/使能配置寄存器　　复位值：0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RGD_PID								RGD_PIDMASK							
访问	RW								RW							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																RGD_EN
访问																RW
复位值																1

字段	说明
31:24 RGD_PID	**Region PID** 0-255: Region PID
23:16 RGD_PIDMASK	**PID 掩码** 0-255: PID 掩码， 如果置位了 PID Mask 中的某个位，则在比较时将忽略相应的 PID 位
0 RGD_EN	**RGD 使能位** 0: 不使能 1: 使能 **说明：1. 该位只能被 Master Debugger 修改** **2. 改变 RGD0_WORD0/WORD1/WORD2 寄存器的值， 该位会被硬件自动清零**

© 2013 - 2024 杰发科技有限公司

8.5.11 RGD PID/使能配置寄存器(MPU_RGD1_WORD3-MPU_RGD7_WORD3)

表 8-14 MPU_RGDn_WORD3 寄存器

MPU_RGDn_WORD3　　RGD PID/使能配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RGD_PID								RGD_PIDMASK							
访问	RW								RW							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																RGD_EN
访问																RW
复位值																0

字段	说明
31:24 RGD_PID	**Region PID** 0-255: Region PID
23:16 RGD_PIDMASK	**PID 掩码** 0-255: PID 掩码， 如果置位了 PID Mask 中的某个位，则在比较时将忽略相应的 PID 位
0 RGD_EN	**RGD 使能位** 0: 不使能 1: 使能 **说明：改变 RGDn_WORD0/WORD1/WORD2 寄存器的值， 该位会被硬件自动清零**

© 2013 - 2024 杰发科技有限公司

8.5.12 权限配置复用寄存器(MPU_RGD0_AAC)

表 8-15 MPU_RGD0_AAC 寄存器

MPU_RGD0_AAC 权限配置寄存器 复位值：0x0003FFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	M0PID_EN	M1PID_EN													M2SW	M2SR
访问	RW	RW													RW	RW
复位值	0	0													1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	M2SX				M1SW	M1SR	M1SX	M1UW	M1UR	M1UX	M0SW	M0SR	M0SX	M0UW	M0UR	M0UX
访问	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	1				1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
31 M0PID_EN	**Master Core PID 使能** 0: 不使能 1: 使能
30 M1PID_EN	**Master Debugger PID 使能** 0: 不使能 1: 使能 **说明：该位只能被 Master Debugger 修改**
17 M2SW	**Master DMA 特权模式写访问** 0: 禁止 1: 允许
16 M2SR	**Master DMA 特权模式读访问** 0: 禁止 1: 允许
15 M2SX	**Master DMA 特权模式可执行访问** 0: 禁止 1: 允许
11 M1SW	**Master Debugger 特权模式写访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**

© 2013 - 2024 杰发科技有限公司

字段	说明
10 M1SR	**Master Debugger 特权模式读访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**
9 M1SX	**Master Debugger 特权模式可执行访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**
8 M1UW	**Master Debugger 用户模式写访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**
7 M1UR	**Master Debugger 用户模式读访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**
6 M1UX	**Master Debugger 用户模式可执行访问** 0: 禁止 1: 允许 **说明：该位只能被 Master Debugger 修改**
5 M0SW	**Master Core 特权模式写访问** 0: 禁止 1: 允许
4 M0SR	**Master Core 特权模式读访问** 0: 禁止 1: 允许
3 M0SX	**Master Core 特权模式可执行访问** 0: 禁止 1: 允许
2 M0UW	**Master Core 用户模式写访问** 0: 禁止

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 允许
1 M0UR	**Master Core 用户模式读访问** 0: 禁止 1: 允许
0 M0UX	**Master Core 用户模式可执行访问** 0: 禁止 1: 允许

8.5.13 权限配置复用寄存器(MPU_RGD1_AAC-MPU_RGD7_AAC)

表 8-16 MPU_RGDn_AAC 寄存器

MPU_RGDn_AAC **权限配置复用寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	M0PID_EN	M1PID_EN													M2SW	M2SR
访问	RW	RW													RW	RW
复位值	0	0													0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	M2SX				M1SW	M1SR	M1SX	M1UW	M1UR	M1UX	M0SW	M0SR	M0SX	M0UW	M0UR	M0UX
访问	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0				0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31 M0PID_EN	**Master Core PID 使能** 0: 不使能 1: 使能
30 M1PID_EN	**Master Debugger PID 使能** 0: 不使能 1: 使能
17 M2SW	**Master DMA 特权模式写访问** 0: 禁止 1: 允许
16 M2SR	**Master DMA 特权模式读访问** 0: 禁止 1: 允许

© 2013 - 2024 杰发科技有限公司

字段	说明
15 M2SX	**Master DMA 特权模式可执行访问** 0: 禁止 1: 允许
11 M1SW	**Master Debugger 特权模式写访问** 0: 禁止 1: 允许
10 M1SR	**Master Debugger 特权模式读访问** 0: 禁止 1: 允许
9 M1SX	**Master Debugger 特权模式可执行访问** 0: 禁止 1: 允许
8 M1UW	**Master Debugger 用户模式写访问** 0: 禁止 1: 允许
7 M1UR	**Master Debugger 用户模式读访问** 0: 禁止 1: 允许
6 M1UX	**Master Debugger 用户模式可执行访问** 0: 禁止 1: 允许
5 M0SW	**Master Core 特权模式写访问** 0: 禁止 1: 允许
4 M0SR	**Master Core 特权模式读访问** 0: 禁止 1: 允许
3 M0SX	**Master Core 特权模式可执行访问** 0: 禁止 1: 允许
2 M0UW	**Master Core 用户模式写访问** 0: 禁止 1: 允许
1	**Master Core 用户模式读访问**

© 2013 - 2024 杰发科技有限公司

字段	说明
M0UR	
	0: 禁止
	1: 允许
0	**Master Core 用户模式可执行访问**
M0UX	
	0: 禁止
	1: 允许

© 2013 - 2024 杰发科技有限公司

9 复位（RCM）

9.1 简介

复位控制模块（Reset Control Module）主要是用于管理各复位请求，并且支持复位前产生中断。

9.2 特性

- 上电复位（POR）：
  - POR_Reset：IC 上电复位
- 系统复位：
  - External Reset：外部复位脚复位，低电平有效
  - LVR Reset：低压检测复位
  - Software Reset：Cortex™-M4F 软件系统复位
  - ECC 2 Bit Error Reset：SRAM ECC 检测到 2 BIT 错误复位
  - Lock up Reset：Cortex™-M4F 死锁复位
  - PLL Unlock Reset：PLL 失锁复位
  - VHSI Loss Reset：当检测到 VHSI 时钟异常时，将会产生一个可屏蔽的系统复位
  - Watchdog Reset：看门狗定时器复位
  - XOSC Loss Reset：当检测到外部 XOSC 时钟异常时，将会产生一个可屏蔽的系统复位
  - SMU Error Reset：安全管理模块收到内部错误报告后，将会产生一个可屏蔽的系统复位
  - ACK Error Reset：进入低功耗模式时，一个或多个外设回复 ACK 超过 1 秒，将会产生一个可屏蔽的系统复位

  每个系统复位源在 RCM_STATUS 寄存器都有相应的状态位。

9.3 结构框图

外部 Reset，POR 和 LVD 信号默认是高电平，任意一信号为低电平，即产生复位信号；其他复位请求信号则是默认是低电平，任意一信号发生请求，则产生 rst_pre 复位信号，并且可设置复位延迟时间。

© 2013 - 2024 杰发科技有限公司

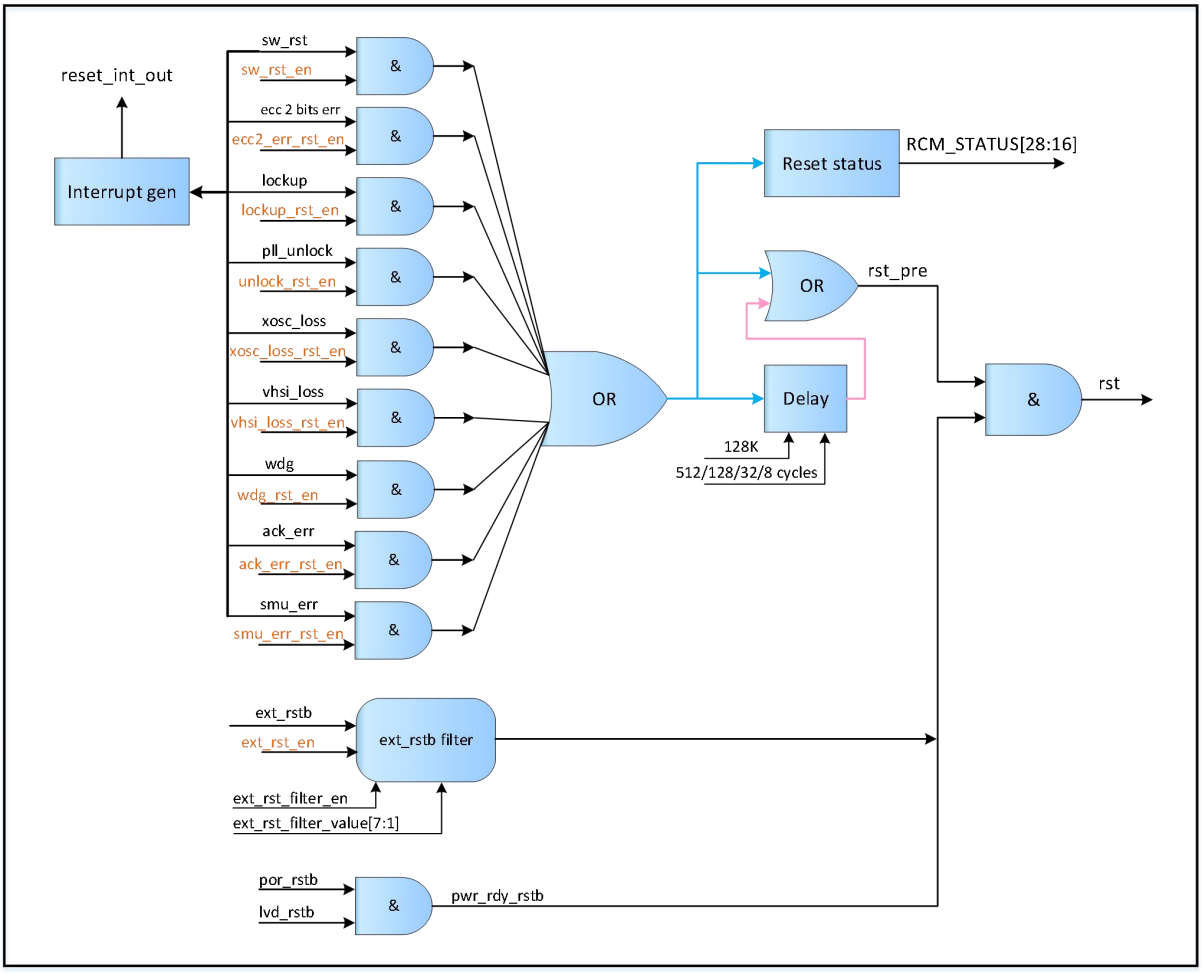


图 9-1 RCM 模块结构框图

9.4 功能描述

9.4.1 上电复位 (POR)

产生 POR 必须同时满足以下两个条件：

1. MCU 初次上电；
2. 电源电压下降到低于上电复位电压电平（VPOR）。

随着电源电压上升，低电压检测（LVR）会将 MCU 保持在复位状态，直到电源电压上升至高于低电压检测（LVR）的低阀值，具体请参考《ATC_AC7840x_Datasheet_CH》表 6-2 LVD/POR/AVDD 电压告警规格。

© 2013 - 2024 杰发科技有限公司

9.4.2 系统复位(System Reset)

系统复位开始时，片上稳压器处于完全稳压状态，系统时钟来源于 VHSI。处理器退出复位状态后，执行下列操作：

1. 从向量表偏移 0 处读取 SP(SP_main)初始值；
2. 从向量表偏移 4 处读取程序计数器 (PC) 初始值；
3. 连接寄存器 (LR) 设置为 0xFFFF_FFFF。

9.4.2.1 外部引脚复位 (External Reset)

MCU 专用引脚，用于复位和重启 MCU 所有的模块。由于该引脚是低电平有效，建议在外部复位电路中添加上拉电阻以防止噪声，或开启外部复位过滤器。可以通过设置 RCM_EN[EXT_RST_EN]=0 屏蔽外部复位功能。

9.4.2.2 低压检测复位 (LVR Reset)

集成了一个低压保护系统，以便在电源电压发生变化期间保护存储器内容和控制 MCU 系统状态。该系统由上电复位(POR)电路和 LVR 电路组成。

具体数值，请参考《ATC_AC7840x_Datasheet_CH》5.1.1 DC 特性章节。

9.4.2.3 软件系统复位（Software Reset）

配置 RCM_EN[SW_RST_EN]=1 将使能 Cortex™-M4F 软件系统复位。将内核发出软件复位请求以产生系统复位。

9.4.2.4 ECC 2 bit 错误复位（2 Bit Error Reset）

配置 RCM_EN[ECC2_ERR_RST_EN]=1 将使能 ECC 2bits 错误复位。当 SRAM ECC 检测到 2 BIT 错误后，将发出系统复位请求以产生系统复位。

9.4.2.5 死锁复位(Lock up Reset)

死锁（Lock up） 表明内核出现严重错误。这是由于激活处理器内置系统状态保护硬件后出现无法恢复的异常情况而导致内核锁定的结果。

死锁（Lock up）出现时，如果 RCM_EN[LOCKUP_RST_EN]=1，则会产生系统复位。

9.4.2.6 PLL 失锁复位（PLL Unlock Reset）

配置 RCM_EN[PLL_UNLOCK_RST_EN]=1 将使能 PLL 失锁复位。当 PLL 检测到失锁错误时，将发出系统复位请求以产生系统复位。

© 2013 - 2024 杰发科技有限公司

9.4.2.7 VHSI 丢失复位（VHSI Loss Reset）

配置 RCM_EN[VHSI_LOSS_RST_EN]=1 将使能 VHSI 时钟丢失复位。当 VHSI 发生丢失时，由 CMU 模块发出系统复位请求以产生系统复位。

9.4.2.8 看门狗定时器复位（Watchdog Reset）

看门狗定时器 (WDG)通过软件定时刷新来对系统进行监控。

如果周期性刷新没有出现，看门狗将发送系统复位。

更多细节的部分，请参考章节 22 看门狗模块（WDG）。

9.4.2.9 晶体振荡器（XOSC） 监控器功能

配置 CKGEN_CTRL[XOSC_MON_EN]=1 使能 XOSC 监控系统。使能 XOSC 监控系统后，时钟检测器在 HSE 振荡器启动延迟后启用，并在此振荡器停止时被禁用。如果在 HSE 振荡器时钟上检测到故障，则该振荡器会被自动禁用，如果 RCM_EN[XOSC_LOSS_RST_EN]=1，则会产生系统复位。

9.4.2.10 SMU 错误复位（SMU Error Reset）

配置 RCM_EN[SMU_ERR_RST_EN]=1 将使能 SMU 错误复位。当安全管理模块收到内部错误报告后，将发出系统复位请求以产生系统复位。

9.4.2.11 应答错误复位（ACK Error Reset）

配置 RCM_EN[ACK_ERR_RST_EN]=1 将使能休眠应答错误复位。当进入低功耗模式时，一个或多个外设回复 ACK 超过 1 秒，将发出系统复位请求以产生系统复位。

9.4.3 复位范围

9.4.3.1 上电复位

POR 复位，整个芯片都会复位，因此除有特别说明的少数寄存器(比如涉及 CSE 模块的 FLASH 分区信息寄存器、Flash 写保护寄存器)外，其他所有寄存器都会恢复默认值。

9.4.3.2 系统复位

系统复位，除了下面的寄存器没有被复位，其它寄存器都会恢复默认值：

表 9-1 系统复位范围

模块	寄存器

© 2013 - 2024 杰发科技有限公司

模块	寄存器
RCM	RCM_CTRL
	RCM_EN
CKGEN	CKGEN_LP_CLK_
SMU	SMU_SFESS
	SMU_LFESS
	SMU_SRSTCNTS
RTC	所有寄存器都不复位
PCT	所有寄存器都不复位

9.4.4 中断请求

RCM 模块支持复位前产生中断的功能，中断的请求源参见表 9-2。

表 9-2 RCM 中断汇总

中断源	标志	本地使能	全局使能
软件系统复位前产生中断	SW_RST_INT	SW_RST_INT_EN	RCM_CTRL[RST_GLB_INT_EN]
Lock Up 复位前产生中断	LOCKUP_RST_INT	LOCKUP_RST_INT_EN	
ACK Error 复位前产生中断	ACK_ERR_RST_INT	ACK_ERR_RST_INT_EN	
看门狗复位前产生中断	WDG_RST_INT	WDG_RST_INT_EN	
XOSC Loss 复位前产生中断	XOSC_LOSS_RST_INT	XOSC_LOSS_RST_INT_EN	
PLL Unlock 复位前产生中断	PLL_UNLOCK_RST_INT	PLL_UNLOCK_RST_INT_EN	
VHSI Loss 复位前产生中断	VHSI_LOSS_RST_INT	VHSI_LOSS_RST_INT_EN	
SMU 错误复位前产生中断	SMU_ERR_RST_INT	SMU_ERR_RST_INT_EN	
ECC 2 bits 错误复位前产生中断	ECC2_ERR_RST_INT	ECC2_ERR_RST_INT_EN	

下面以软件复位前产生中断为例，配置步骤如下：

1. 复位前产生中断全局使能，配置 RCM_CTRL[RST_GLB_INT_EN]为 1。
2. 软件复位前产生中断使能，配置 RCM_CTRL[SW_RST_INT_EN]为 1。
3. 设置复位延迟时间，配置 RCM_CTRL[RST_DLY_TIME]。
4. 当产生软件复位请求时，会先进入 RCM_IRQHandler 中断，延时时间达到后，如果该复位使能，则进行系统复位。

9.4.5 调试模式

RCM 模块不支持 debug 功能控制，不受调试状态的影响。

© 2013 - 2024 杰发科技有限公司

9.5 寄存器定义

表 9-3 复位（Reset）寄存器映射

RCM 基地址：0x40000010

地址	名称	宽度（位）	描述
RCM 基地址+0x04	RCM_CTRL	32	芯片复位控制
RCM 基地址+0x08	RCM_EN	32	芯片复位使能
RCM 基地址+0x0C	RCM_STATUS	32	芯片复位状态

9.5.1 复位控制寄存器(RCM_CTRL)

表 9-4 RCM_CTRL 寄存器

RCM_CTRL　　复位控制寄存器　　Reset：0x83000002

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RST_CLK_AUTO_SEL								RST_GLB_INT_EN	RST_CLK_SEL		ECC2ERR_RST_INT_EN	SMUERR_RST_INT_EN		VHSILOSS_RST_INT_EN	PLL_UNLOCK_RST_INT_EN
访问	RW								RW	RW		RW	RW		RW	RW
复位值	1								0	0		0	0		0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	XOSCLOSS_RST_INT_EN		WDGRST_INT_EN	ACKERR_RST_INT_EN	LOCKUP_RST_INT_EN	SW_RST_INT_EN	RST_DLY_TIME		EXT_RST_FILTER_VAL							EXT_RST_FILTER_EN
访问	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0		0	0	0	0	0	0	0	0	0	0	0	0	1	0

字段	说明
31 RST_CLK_AUTO_SEL	**复位时钟自动选择使能** 0: 复位时钟由软件选择，请参考 RST_CLK_SEL 位域的描述 1: 复位时钟由硬件自动选择。当 HSI 打开时，复位时钟为 HSI 8M。当 HSI 关闭时， 复位时钟自动切到 LSI 128K
23 RST_GLB_INT_EN	**复位(前)产生中断全局使能** 0: 复位全局中断不使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 复位全局中断使能
22 RST_CLK_SEL	**复位时钟软件选择** 0 : LSI 128K 时钟源 1 : HSI 8M 时钟源 当 RST_CLK_AUTOSEL 为 0 时，软件选择才有效。
20 ECC2_ERR_RST_INT_EN	**SRAM ECC 2 bits 错误复位(前)产生中断使能** 0 : 不使能 1 : 使能
19 SMU_ERR_RST_INT_EN	**SMU 错误复位(前)产生中断使能** 0 : 不使能 1 : 使能
17 VHSI_LOSS_RST_INT_EN	**VHSI 异常复位(前)产生中断使能** 0 : 不使能 1 : 使能
16 PLL_UNLOCK_RST_INT_EN	**PLL 失锁复位(前)产生中断使能** 0 : 不使能 1 : 使能
15 XOSC_LOSS_RST_INT_EN	**XOSC 异常复位(前)产生中断使能** 0 : 不使能 1 : 使能
13 WDG_RST_INT_EN	**看门狗复位(前)产生中断使能** 0 : 不使能 1 : 使能
12 ACK_ERR_RST_INT_EN	**进入低功耗 ACK 超时复位(前)产生中断使能** 0 : 不使能 1 : 使能
11 LOCKUP_RST_INT_EN	**CPU 死锁复位(前)产生中断使能** 0 : 不使能 1 : 使能
10 SW_RST_INT_EN	**软件系统复位(前)产生中断使能** 0 : 不使能 1 : 使能 **说明：这里的软件系统复位指的是用户设置内核 SYSRESETREQ bit 为 1，**

© 2013 - 2024 杰发科技有限公司

字段	说明
	产生的复位。
9:8 RST_DLY_TIME	**复位延迟时间** 00b : 8 * LPI 128K 01b : 32 * LPI 128K 10b : 128 * LPI 128K 11b : 512 * LPI 128K
7:1 EXT_RST_FILTER_VAL	**外部引脚复位滤波时钟周期** 以 LPI 128 K 为计数时钟，寄存器 VAL-1 表示可以被滤掉的信号，如 VAL=2 表示可以滤掉 1 个周期长的复位信号，VAL=0 表示可以滤掉 127 个周期长的复位信号。
0 EXT_RST_FILTER_EN	**外部引脚复位滤波使能** 0 : 不使能 1 : 使能

9.5.2 复位使能寄存器(RCM_EN)

表 9-5 RCM_EN 寄存器

RCM_EN　　复位使能寄存器　　Reset：0x000007EF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						ECC2_ERR_RST_EN	SMU_ERR_RST_EN	EXT_RST_EN	VHSI_LOSS_RST_EN	PLL_UNLOCK_RST_EN	XOSC_LOSS_RST_EN		WDG_RST_EN	ACK_ERR_RST_EN	LOCKUP_RST_EN	SW_RST_EN
访问						RW	RW	RW	RW	RW	RW		RW	RW	RW	RW
复位值						1	1	1	1	1	1		1	1	1	1

字段	说明
10 ECC2_ERR_RST_EN	**SRAM ECC 2 bits 错误复位使能** 0 : 不使能 1 : 使能
9 SMU_ERR_RST_EN	**SMU 错误复位使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：不使能 1：使能
8 EXT_RST_EN	**外部复位使能** 0：不使能 1：使能
7 VHSI_LOSS_RST_EN	**VHSI 异常复位使能** 0：不使能 1：使能 **说明：VHSI 异常信号由 CMU 模块产生。**
6 PLL_UNLOCK_RST_EN	**PLL 失锁复位使能** 0：不使能 1：使能
5 XOSC_LOSS_RST_EN	**XOSC 异常复位使能** 0：不使能 1：使能
3 WDG_RST_EN	**WDG 复位使能** 0：不使能 1：使能
2 ACK_ERR_RST_EN	**进入低功耗 ACK 超时复位使能** 0：不使能 1：使能
1 LOCKUP_RST_EN	**CPU 死锁复位使能** 0：不使能 1：使能
0 SW_RST_EN	**软件系统复位使能** 0：不使能 1：使能 **说明：这里的软件系统复位指的是用户设置内核 SYSRESETREQ bit 为 1，产生的复位。**

© 2013 - 2024 杰发科技有限公司

9.5.3 复位状态寄存器(RCM_STATUS)

表 9-6 RCM_STATUS 寄存器

RCM_STATUS 复位状态寄存器 **Reset：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RST_STATUS_CLR			ECC2_ERR_RST	SMU_ERR_RST	EXT_RST	VHSI_LOSS_RST	PLL_UNLOCK_RST	XOSC_LOSS		WDG_RST	ACK_ERR_RST	LOCKUP_RST	SW_RST	LVR_RST	POR_RST
访问	RW			RO	RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO
Reset	0			0	0	0	0	0	0		0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RST_INT_STATUS_CLR					ECC2_ERR_RST_INT	SMU_ERR_RST_INT		VHSI_LOSS_RST_INT	PLL_UNLOCK_RST_INT	XOSC_LOSS_RST_INT		WDG_RST_INT	ACK_ERR_RST_INT	LOCKUP_RST_INT	SW_RST_INT
访问	RW					RO	RO		RO	RO	RO		RO	RO	RO	RO
Reset	0					0	0		0	0	0		0	0	0	0

字段	说明
31 RST_STATUS_CLR	**清除所有复位状态** 0 : 不清除 1 : 清除
28 ECC2_ERR_RST	**SRAM ECC 2 bits 错误复位状态** 0 : 无效 1 : 有效
27 SMU_ERR_RST	**SMU 错误复位状态** 0 : 无效 1 : 有效
26 EXT_RST	**外部引脚复位状态** 0 : 无效

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：有效
25 VHSI_LOSS_RST	**VHSI 异常复位状态** 0：无效 1：有效
24 PLL_UNLOCK_RST	**PLL 失锁复位状态** 0：无效 1：有效
23 XOSC_LOSS_RST	**XOSC 异常复位状态** 0：无效 1：有效
21 WDG_RST	**看门狗复位状态** 0：无效 1：有效
20 ACK_ERR_RST	**进入低功耗 ACK 超时复位状态** 0：无效 1：有效
19 LOCKUP_RST	**CPU 死锁复位状态** 0：无效 1：有效
18 SW_RST	**软件系统复位状态** 0：无效 1：有效 **说明：这里的软件系统复位指的是用户设置内核 SYSRESETREQ bit 为 1，产生的复位。**
17 LVR_RST	**LVR 复位状态** 0：无效 1：有效
16 POR_RST	**POR 复位状态** 0：无效 1：有效
15 RST_INT_STATUS_CLR	**清除所有复位中断状态** 0：不清除 1：清除
10	**SRAM ECC 2 bits 错误复位中断标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
ECC2_ERR_RST_INT	
	0：无效
	1：有效
9 SMU_ERR_RST_INT	**SMU 错误复位中断标志**
	0：无效
	1：有效
7 VHSI_LOSS_RST_INT	**VHSI 异常复位中断标志**
	0：无效
	1：有效
6 PLL_UNLOCK_RST_INT	**PLL 失锁复位中断标志**
	0：无效
	1：有效
5 XOSC_LOSS_RST_INT	**XOSC 异常复位中断标志**
	0：无效
	1：有效
3 WDG_RST_INT	**看门狗复位中断标志**
	0：无效
	1：有效
2 ACK_ERR_RST_INT	**进入低功耗 ACK 超时复位中断标志**
	0：无效
	1：有效
1 LOCKUP_RST_INT	**CPU 死锁复位中断标志**
	0：无效
	1：有效
0 SW_RST_INT	**软件系统复位中断标志**
	0：无效
	1：有效

© 2013 - 2024 杰发科技有限公司

10 时钟（CKGEN）

10.1 简介

时钟控制模块为 MCU 提供时钟源选择。该模块包含高速内部时钟（HSI），超高速内部时钟（VHSI），高速外部时钟（HSE），低速内部时钟（LSI）和系统 PLL 时钟（SPLL）来作为时钟源，其中 SPLL 可选择内部或外部时钟作为参考时钟。这些时钟通过选择器和分配器，生成所有模块的时钟和频率。

10.2 结构框图

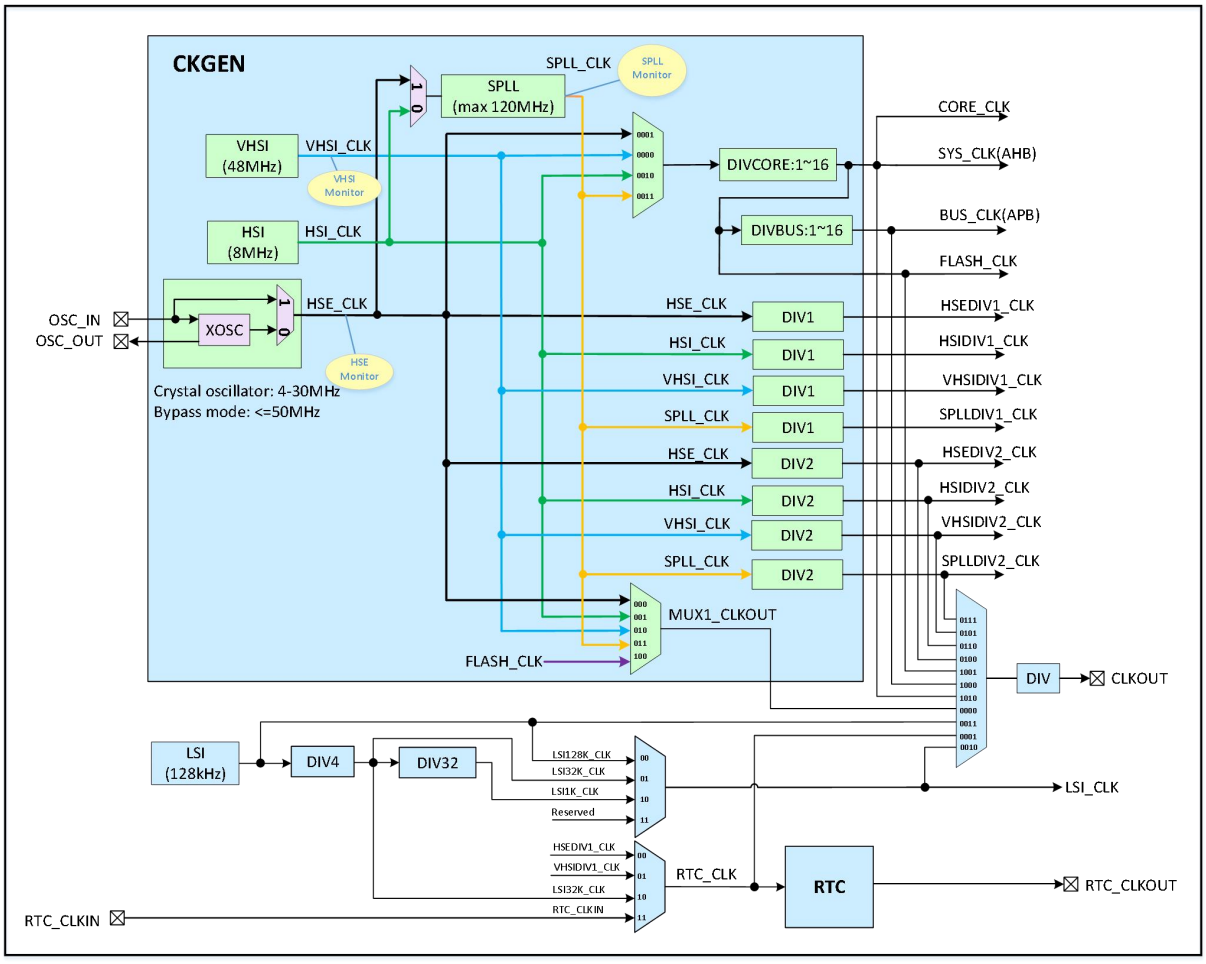


图 10-1 时钟控制结构

该模块包含如下时钟源：

- 高速内部时钟（HSI）：内部 RC 振荡器提供 8MHz 时钟源
- 超高速内部时钟（VHSI）：内部 RC 振荡器提供 48MHz 时钟源，上电或复位默认的系统时钟源

© 2013 - 2024 杰发科技有限公司

- 外部高速时钟（HSE）：外部 OSC 提供 4MHz ~30MHz 晶振或外部最高 50MHz 的时钟输入
- 低速内部时钟（LSI）：内部低速 RC 振荡器提供 128KHz 时钟源
- 系统 PLL 时钟（SPLL）：提供高达 120MHz 的高速时钟

每个外设都有专用的时钟使能信号来控制时钟的开关，请参阅 10.4 寄存器定义 章节以了解详细地址。

说明：

- 系统时钟高达 120MHz
- SYS_CLK(AHB)最高可达 120MHz
- BUS_CLK(APB)最高可达 60MHz。因此当 SYS_CLK 为 60MHz 时，BUSCLK_DIV 可以配置为 1 分频；当 SYS_CLK 为大于 60MHz 时，BUSCLK_DIV 推荐配置为 2 分频。

10.3 功能描述

10.3.1 SPLL 功能

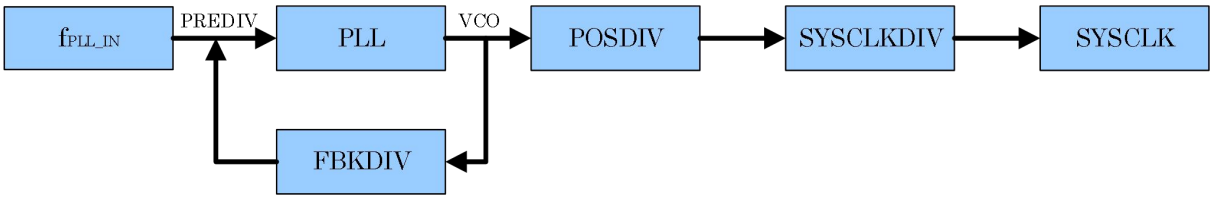


图 10-2 系统时钟示意图

- f_{PLL_IN} 为 PLL 输入频率，参考频率 $f_{REF} = f_{PLL_IN} / PREDIV$；
- $VCO = f_{REF} * FBKDIV$；
- 输出频率 $f_{OUT} = VCO / POSDIV$；
- 系统时钟 $SYSCLK = f_{OUT} / SYSCLKDIV$。

注意：

- 输入频率为 4MHz~48MHz，PLL 参考频率需小于 12MHz；
- VCO 的频率范围是 0.5GHz ~1.5GHz；
- 表中的 PREDIV, FBKDIV, POSDIV 为分频/倍频值，不是寄存器值。

© 2013 - 2024 杰发科技有限公司

表 10-1 典型 SPLL 配置参考表

f_{PLL_IN} (MHz)	PREDIV	FBKDIV	VCO(MHz)	POSDIV	Frequency (MHz)	Sysclk_div	System Clock(MHz)
8	1	120	960	8	120	1	120
8	1	96	768	8	96	1	96
8	1	120	960	12	80	1	80
8	1	128	1024	16	64	1	64
8	1	96	768	16	48	1	48
16	2	120	960	8	120	1	120
16	2	96	768	8	96	1	96
16	2	120	960	12	80	1	80
16	2	128	1024	16	64	1	64
16	2	96	768	16	48	1	48
4	1	240	960	8	120	1	120
4	1	192	768	8	96	1	96
4	1	240	960	12	80	1	80
4	1	128	512	8	64	1	64
4	1	192	768	16	48	1	48
12	1	100	1200	10	120	1	120
12	2	96	576	6	96	1	96
12	1	80	960	12	80	1	80
12	1	64	768	12	64	1	64
12	2	96	576	6	96	2	48
30	4	160	1200	10	120	1	120
30	4	128	960	10	96	1	96
30	4	128	960	12	80	1	80
30	4	128	960	20	48	1	48

10.3.2 外设复位功能

上电或复位，外设默认在非复位状态，外设可以正常工作。用户可以配置 CKGEN_PERI_SFT_RST0, CKGEN_PERI_SFT_RST1, CKGEN_PERI_SFT_RST2 寄存器对外设进行单独的复位。如：用户软件对 CKGEN_PERI_SFT_RST0 [SRST_UART0] 写“0”，UART0 模块变为复位状态。然后软件写“1”，UART0 变为非复位状态，所有寄存器恢复默认值。

© 2013 - 2024 杰发科技有限公司

10.3.3 外设时钟

外设的时钟包含功能时钟和接口时钟，如果外设的接口时钟没有使能，访问寄存器会出现 hardfault。表 10-2 总结了所有模块的功能时钟选择相关功能。表 10-3 总结了各外设的时钟选择示意图。

表 10-2 外设时钟

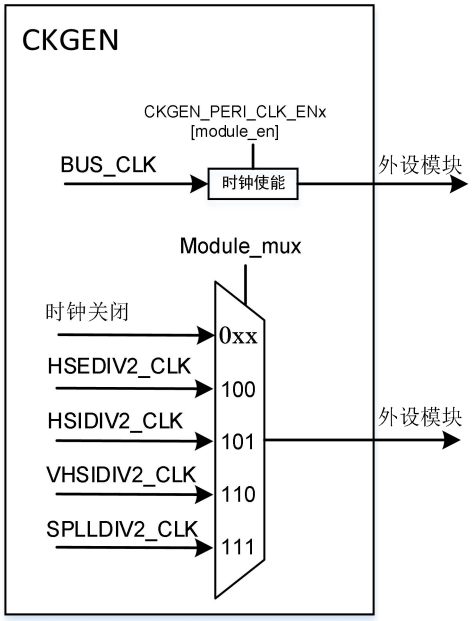
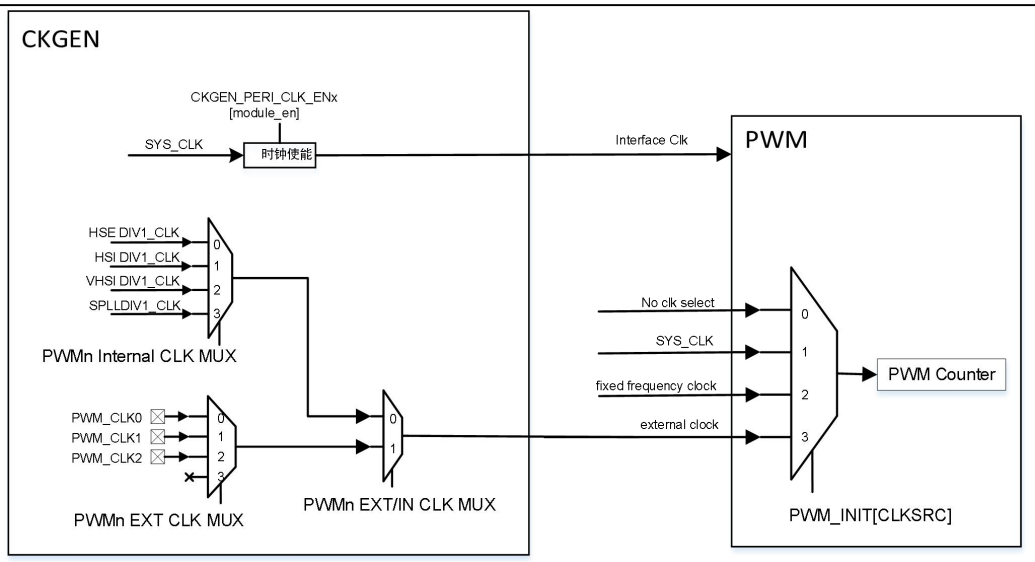
模块名	接口时钟	外设接口时钟是否需要使能 可通过配置 CKGEN_PERI_CLK_ENx 寄存器使能	外设功能时钟选项 可通过配置 CKGEN_PERI_CLK_MUXx 寄存器选项	最大功能时钟频率
UART	BUS_CLK	是	HSEDIV2_CLK, HSIDIV2_CLK, VHSIDIV2_CLK, SPLLDIV2_CLK	不超过 BUS_CLK
SPI	BUS_CLK	是	HSEDIV2_CLK, HSIDIV2_CLK, VHSIDIV2_CLK, SPLLDIV2_CLK	不超过 BUS_CLK
I2C	BUS_CLK	是	HSEDIV2_CLK, HSIDIV2_CLK, VHSIDIV2_CLK, SPLLDIV2_CLK	不超过 BUS_CLK
EIO	BUS_CLK	是	HSEDIV2_CLK HSIDIV2_CLK VHSIDIV2_CLK SPLLDIV2_CLK	不超过 BUS_CLK
CAN	BUS_CLK	是	HSEDIV2_CLK SYS_CLK SPLLDIV2_CLK	80M
TIMER	BUS_CLK	是	HSEDIV2_CLK HSIDIV2_CLK VHSIDIV2_CLK SPLLDIV2_CLK	不超过 BUS_CLK
PCT	BUS_CLK	是	HSEDIV2_CLK, HSIDIV2_CLK, VHSIDIV2_CLK, SPLLDIV2_CLK	不超过 BUS_CLK
PWM	SYS_CLK	是	HSEDIV1_CLK, HSIDIV1_CLK, VHSIDIV1_CLK, SPLLDIV1_CLK, PWM_CLKx PWM_INIT[CLKSRC]: SYS_CLK	SPLL_CLK
RTC	BUS_CLK	是	HSEDIV1_CLK, VHSIDV1_CLK, LSI32K_CLK, RTC_CLKIN,	不超过 BUS_CLK

© 2013 - 2024 杰发科技有限公司

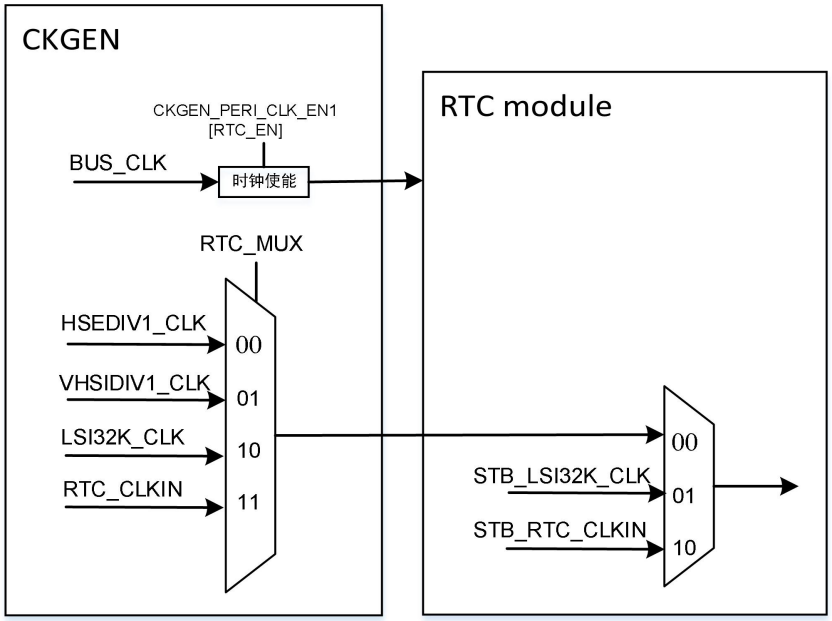
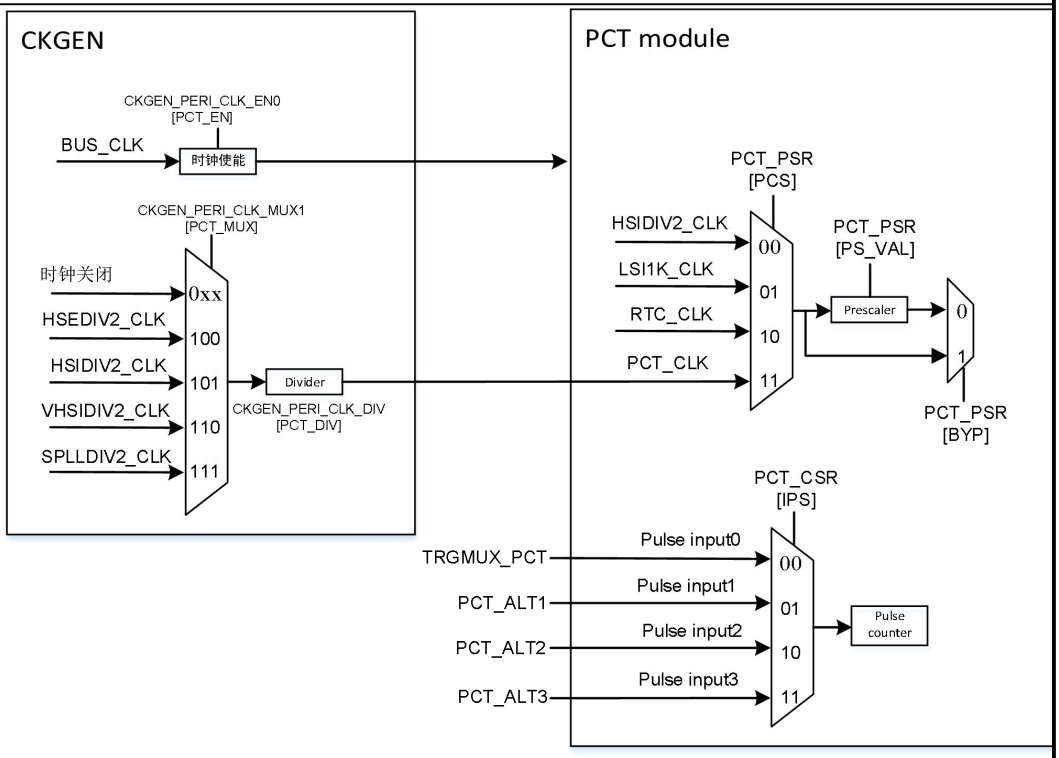
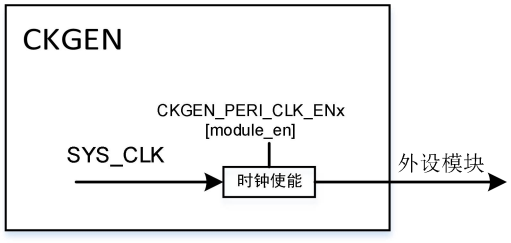
模块名	接口时钟	外设接口时钟是否需要使能	外设功能时钟选项	最大功能时钟频
			RTC_CTRL[CLKSEL]	
DMA	BUS_CLK	是	-	SYS_CLK
GPIO	SYS_CLK	是	-	SYS_CLK
WDG	BUS_CLK	是	WDG_CS1[CLK]: LSI_CLK BUS_CLK HSI_CLK HSE_CLK	BUS_CLK
EWDG	BUS_CLK	是	-	LSI
CRC	BUS_CLK	是	-	SYS_CLK
SMU	BUS_CLK	是	-	SYS_CLK
MPU	BUS_CLK	否	-	SYS_CLK
CMU	BUS_CLK	否	-	SYS_CLK
CTU	BUS_CLK	是	-	SYS_CLK
PDT	SYS_CLK	是	-	SYS_CLK
ADC	BUS_CLK	是	HSEDIV2_CLK HSIDIV2_CLK VHSIDIV2_CLK SPLLDIV2_CLK	不超过 BUS_CLK
ACMP	BUS_CLK	是	-	SYS_CLK

© 2013 - 2024 杰发科技有限公司

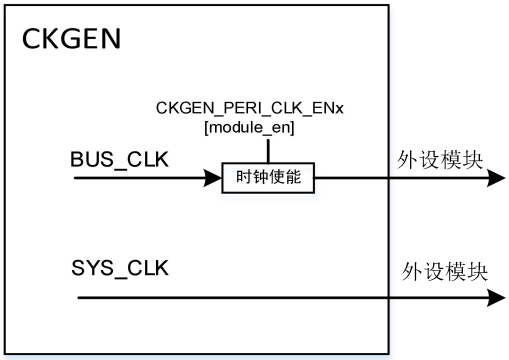
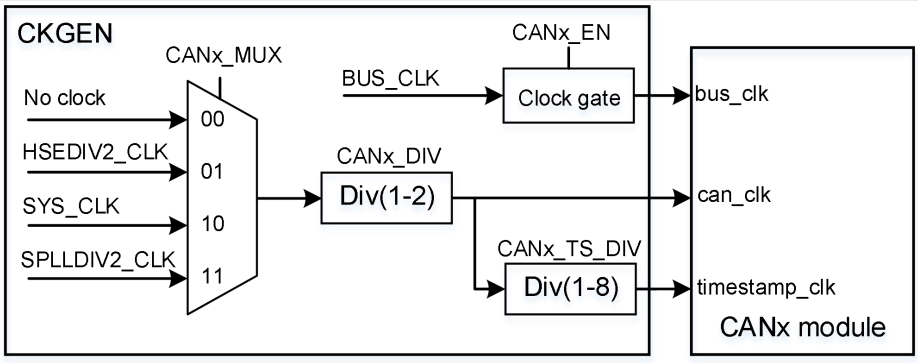
表 10-3 外设时钟示意图

模块	时钟示意图
UART SPI I2C ADC TIMER EIO	
PWM	

© 2013 - 2024 杰发科技有限公司

模块	时钟示意图
RTC	
PCT	
GPIO PDT	

© 2013 - 2024 杰发科技有限公司

模块	时钟示意图
DMA CRC CTU ACMP SMU CMU	
CAN	

10.3.4 调试模式

CKGEN 模块不支持 debug 功能控制，不受调试状态的影响。

10.3.5 编程指南

CKGEN 软件流程配置如下：

1. 先确定系统时钟源是 HSI, VHSI, HSE 还是 SPLL。
2. 如果系统时钟源需要选择为 HSE 或者 SPLL 时钟源为 HSE，先配置寄存器 SPM_PWR_MGR_CFG1[XOSC_HSEBYP]为 0，再配置 SPM_PWR_MGR_CFG1[XOSC_HSEEN] 使能 HSE 时钟。最后通过 SPM_PWR_MGR_CFG1[XOSC_RDY]位是否为 1，检查 HSE 是否稳定。
3. 如果系统时钟源需要选择为 SPLL，根据需要设置的系统时钟频率，参考 10.3.1 SPLL 功能 章节描述，先计算 SPLL 相关的分频系数，并通过配置 ANA_SPLL_CFG0 寄存器，配置 PLL 相关分频系数。再配置寄存器 CKGEN_CTRL[PLL_REF_SEL]，选择 SPLL 的时钟源为 HSI 或 HSE。然后，使能 SPLL 时钟输出，即写寄存器 SPM_PWR_MGR_CFG1[SPLL_EN]为 1。最后，通过 SPM_PWR_MGR_CFG1[PLL_RDY]位是否为 1，检查 SPLL 是否稳定。

© 2013 - 2024 杰发科技有限公司

4. 如果系统时钟源需要选择为 VHSI，先使能 VHSI 时钟，即写寄存器 SPM_PWR_MGR_CFG1[VHSI_EN]为 1。然后通过 SPM_PWR_MGR_CFG1[VHSI_RDY]位是否为 1，检查 VHSI 是否稳定。
5. 如果系统时钟源需要选择为 HSI，通过 SPM_PWR_MGR_CFG1[HSI_RDY]位是否为 1，检查 HSI 是否稳定。
6. 配置系统时钟源。系统时钟源 RUN 模式下的时钟源配置寄存器 CKGEN_CTRL [SYSCLK_SRC_SEL_RUN]位。系统时钟源 VLPR 模式下的时钟源配置寄存器 CKGEN_CTRL [SYSCLK_SRC_SEL_VLPR]位。
7. 根据需要的 SYS_CLK 频率，设置 SYS_CLK 分频，即配置寄存器 CKGEN_CTRL[SYSCLK_DIV]位。
8. 根据需要的 BUS_CLK 频率，设置 BUS_CLK 分频。 RUN 模式下分频需要配置寄存器 CKGEN_CTRL[BUS_CLK_DIV_RUN]位。VLPR 模式下分频需要配置寄存器 CKGEN_CTRL[BUS_CLK_DIV_VLPR]位。
9. 根据应用需要，配置时钟分频，通过配置 CKGEN_CLK_DIV1， CKGEN_CLK_DIV2 寄存器。

10.4 寄存器定义

CKGEN 寄存器只能在特权模式下访问，用户模式访问会产生 hardfault。

表 10-4 时钟寄存器映射

CKGEN 基地址：0x40000000

ANA 基地址：0x40008800

地址	名称	宽度	描述
CKGEN 基地址+0x00	CKGEN_CTRL	32	时钟控制寄存器
CKGEN 基地址+0x04	CKGEN_LP_CLK_MUX	32	低功耗时钟选择寄存器
CKGEN 基地址+0x08	CKGEN_PERI_CLK_EN0	32	外设时钟使能控制 0
CKGEN 基地址+0x0C	CKGEN_PERI_CLK_EN1	32	外设时钟使能控制 1
CKGEN 基地址+0x10	CKGEN_PERI_CLK_EN2	32	外设时钟使能控制 2
CKGEN 基地址+0x20	CKGEN_PERI_SFT_RST0	32	外设软件复位控制 0
CKGEN 基地址+0x24	CKGEN_PERI_SFT_RST1	32	外设软件复位控制 1
CKGEN 基地址+0x28	CKGEN_PERI_SFT_RST2	32	外设软件复位控制 2
CKGEN 基地址+0x2C	CKGEN_CLK_DIV1	32	时钟分频控制寄存器 1
CKGEN 基地址+0x30	CKGEN_CLK_DIV2	32	时钟分频控制寄存器 2
CKGEN 基地址+0x34	CKGEN_PERI_CLK_MUX0	32	外设时钟选择寄存器 0
CKGEN 基地址+0x38	CKGEN_PERI_CLK_MUX1	32	外设时钟选择寄存器 1
CKGEN 基地址+0x3C	CKGEN_PERI_CLK_MUX2	32	外设时钟选择寄存器 2
CKGEN 基地址+0x40	CKGEN_PERI_CLK_MUX3	32	外设时钟选择寄存器 3
CKGEN 基地址+0x48	CKGEN_CLK_OUT_CFG	32	时钟输出配置寄存器
CKGEN 基地址+0x4C	CKGEN_PERI_CLK_DIV	32	外设时钟分频寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
ANA 基地址+0xC0	ANA_SPLL_CFG0	32	SPLL 配置寄存器 0
ANA 基地址+0xC4	ANA_SPLL_CFG1	32	SPLL 配置寄存器 1

10.4.1 控制寄存器(CKGEN_CTRL)

表 10-5 CKGEN_CTRL 寄存器

CKGEN_CTRL 控制寄存器 复位值：

0x08000010

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LOCK		SYSCLK_SRC_SEL_VLPR				SYSCLK_SRC_SEL_RUN				XOSC_MON_EN	PLL_REF_SEL	SYSCLK_DIV			
访问	RW		RW				RW				RW	RW	RW			
复位值	0		0	0	1	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SYSCLK_DIV_VLPR								BUS_CLK_DIV_RUN				BUS_CLK_DIV_VLPR			
访问	RW								RW				RW			
复位值	0	0	0	0					0	0	0	1	0	0	0	0

字段	说明
31 LOCK	**锁** 0: lock 不使能 1: lock 使能 Lock 使能时，不能改写该寄存器。
29:26 SYSCLK_SRC_SEL_VLPR	**VLPR 模式系统时钟源选择** 0000b: 保留 0001b: 保留 0010b: HSI CLK 0011b: 保留
25:22 SYSCLK_SRC_SEL_RUN	**RUN 模式系统时钟源选择** 0000b: VHSI CLK 0001b: HSE CLK 0010b: HSI CLK 0011b: PLL CLK
21 XOSC_MON_EN	**XOSC 监控使能** 0：不使能 1：使能

© 2013 - 2024 杰发科技有限公司

字段	说明
20 PLL_REF_SEL	**SPLL 参考时钟输入选择** 0：HSI 1：HSE
19:16 SYSCLK_DIV	**系统时钟分频** 0000b : 分频系数为 1 0001b : 分频系数为 2 0010b : 分频系数为 3 0011b : 分频系数为 4 …… 1111b : 分频系数为 16
15:12 SYSCLK_DIV_VLPR	**VLPR 模式系统时钟分频** 0000b : 分频系数为 1 0001b : 分频系数为 2 0010b : 分频系数为 3 0011b : 分频系数为 4 …… 1111b : 分频系数为 16
7:4 BUS_CLK_DIV_RUN	**RUN 模式 BUS CLK 分频** 0000b : 分频系数为 1 0001b : 分频系数为 2 0010b : 分频系数为 3 0011b : 分频系数为 4 …… 1111b : 分频系数为 16
3:0 BUS_CLK_DIV_VLPR	**VLPR 模式 BUS CLK 分频** 0000b : 分频系数为 1 0001b : 分频系数为 2 0010b : 分频系数为 3 0011b : 分频系数为 4 …… 1111b : 分频系数为 16

© 2013 - 2024 杰发科技有限公司

10.4.2 低功耗时钟选择寄存器(CKGEN_LP_CLK_MUX)

表 10-6 CKGEN_LP_CLK_MUX 寄存器

CKGEN_LP_CLK_MUX 低功耗时钟选择寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													RTC_CLK_MUX		LSI_CLK_MUX	
访问													RW		RW	
复位值													0	0	0	0

字段	说明
3:2 RTC_CLK_MUX	**RTC 时钟源选择** 00b: HSEDIV1_CLK 01b: VHSIDV1_CLK 10b: LSI32K_CLK 11b: RTC_CLKIN
1:0 LSI_CLK_MUX	**LSI 时钟源选择** 00b: LSI 128K Clock 01b: LSI 32K Clock 10b: LSI 1K Clock 11b: Reserve

说明：

该寄存器为 write once, 只有 POR 复位才会复位该寄存器。

© 2013 - 2024 杰发科技有限公司

10.4.3 外设时钟使能寄存器 0(CKGEN_PERI_CLK_EN0)

表 10-7 CKGEN_PERI_CLK_EN0 寄存器

CKGEN_PERI_CLK_EN0 外设时钟使能寄存器 0 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称												PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN
访问												RW	RW	RW	RW	RW
复位值												0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PWM0_EN				PCT_EN		I2C0_EN	SPI2_EN	SPI1_EN	SPI0_EN			UART3_EN	UART2_EN	UART1-EN	UART0_EN
访问	RW				RW		RW	RW	RW	RW			RW	RW	RW	RW
复位值	0				0		0	0	0	0			0	0	0	0

字段	说明
20 PWM5_EN	PWM5 BUS CLK 使能 0：不使能 1：使能
19 PWM4_EN	PWM4 BUS CLK 使能 0：不使能 1：使能
18 PWM3_EN	PWM3 BUS CLK 使能 0：不使能 1：使能
17 PWM2_EN	PWM2 BUS CLK 使能 0：不使能 1：使能
16 PWM1_EN	PWM1 BUS CLK 使能 0：不使能 1：使能
15 PWM0_EN	PWM0 BUS CLK 使能 0：不使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：使能
11 PCT_EN	PCT BUS CLK 使能 0：不使能 1：使能
9 I2C0_EN	I2C0 BUS CLK 使能 0：不使能 1：使能
8 SPI2_EN	SPI2 BUS CLK 使能 0：不使能 1：使能
7 SPI1_EN	SPI1 BUS CLK 使能 0：不使能 1：使能
6 SPI0_EN	SPI0 BUS CLK 使能 0：不使能 1：使能
3 UART3_EN	UART3 BUS CLK 使能 0：不使能 1：使能
2 UART2_EN	UART2 BUS CLK 使能 0：不使能 1：使能
1 UART1_EN	UART1 BUS CLK 使能 0：不使能 1：使能
0 UART0_EN	UART0 BUS CLK 使能 0：不使能 1：使能

说明：

读写外设寄存器之前需使能对应外设的时钟，否则将产生 Hardfault。

© 2013 - 2024 杰发科技有限公司

10.4.4 外设时钟使能寄存器 1(CKGEN_PERI_CLK_EN1)

表 10-8 CKGEN_PERI_CLK_EN1 寄存器

CKGEN_PERI_CLK_EN1 外设时钟使能寄存器 1 复位值：0x00000060

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				CAN3_EN	CAN2_EN	CAN1_EN	CAN0_EN	CRC_EN	EWDG_EN	WDG_EN	GPIO_EN			DMA_EN	RTC_EN	
访问				RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	
复位值				0	0	0	0	0	0	1	1			0	0	

字段	说明
12 CAN3_EN	CAN3 BUS CLK 使能 0：不使能 1：使能
11 CAN2_EN	CAN2 BUS CLK 使能 0：不使能 1：使能
10 CAN1_EN	CAN1 BUS CLK 使能 0：不使能 1：使能
9 CAN0_EN	CAN0 BUS CLK 使能 0：不使能 1：使能
8 CRC_EN	CRC BUS CLK 使能 0：不使能 1：使能
7 EWDG_EN	EWDG BUS CLK 使能 0：不使能 1：使能
6 WDG_EN	WDG BUS CLK 使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：不使能 1：使能
5 GPIO_EN	GPIO AHB CLK 使能 0：不使能 1：使能
2 DMA_EN	DMA BUS CLK 使能 0：不使能 1：使能
1 RTC_EN	RTC BUS CLK 使能 0：不使能 1：使能

说明：
读写外设寄存器之前需使能对应外设的时钟，否则将产生 Hardfault。

10.4.5 外设时钟使能寄存器 2(CKGEN_PERI_CLK_EN2)

表 10-9 CKGEN_PERI_CLK_EN2 寄存器

CKGEN_PERI_CLK_EN2　　外设时钟使能寄存器 2　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称													SMU_EN			
访问													RW			
复位值													0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		EIO_EN	TIMER_EN	ADC1_EN	ADC0_EN	PDT1_EN	PDT0_EN	ACMP_EN							CTU_EN	
访问		RW	RW	RW	RW	RW	RW	RW							RW	
复位值		0	0	0	0	0	0	0							0	

字段	说明
19 SMU_EN	**SMU BUS CLK 使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：不使能 1：使能
14 EIO_EN	**EIO BUS CLK 使能** 0：不使能 1：使能
13 TIMER_EN	**TIMER BUS CLK 使能** 0：不使能 1：使能
12 ADC1_EN	**ACD1 BUS CLK 使能** 0：不使能 1：使能
11 ADC0_EN	**ACD0 BUS CLK 使能** 0：不使能 1：使能
10 PDT1_EN	**PDT1 BUS CLK 使能** 0：不使能 1：使能
9 PDT0_EN	**PDT0 BUS CLK 使能** 0：不使能 1：使能
8 ACMP_EN	**ACMP BUS CLK 使能** 0：不使能 1：使能
1 CTU_EN	**CTU BUS CLK 使能** 0：不使能 1：使能

© 2013 - 2024 杰发科技有限公司

10.4.6 外设复位寄存器 0(CKGEN_PERI_SFT_RST0)

表 10-10 CKGEN_PERI_SFT_RST0 寄存器

CKGEN_PERI_SFT_RST0 外设复位寄存器 0 复位值：0x001F8BCF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称												SRST_PWM5	SRST_PWM4	SRST_PWM3	SRST_PWM2	SRST_PWM1
访问												RW	RW	RW	RW	RW
复位值												1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SRST_PWM0				SRST_PCT		SRST_I2C0	SRST_SPI2	SRST_SPI1	SRST_SPI0			SRST_UART3	SRST_UART2	SRST_UART1	SRST_UART0
访问	RW				RW		RW	RW	RW	RW			RW	RW	RW	RW
复位值	1				1		1	1	1	1			1	1	1	1

字段	说明
20 SRST_PWM5	**PWM5 软件复位** 0：复位有效 1：复位无效
19 SRST_PWM4	**PWM4 软件复位** 0：复位有效 1：复位无效
18 SRST_PWM3	**PWM3 软件复位** 0：复位有效 1：复位无效
17 SRST_PWM2	**PWM2 软件复位** 0：复位有效 1：复位无效
16 SRST_PWM1	**PWM1 软件复位** 0：复位有效 1：复位无效
15 SRST_PWM0	**PWM0 软件复位** 0：复位有效

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：复位无效
11 SRST_PCT	**PCT 软件复位** 0：复位有效 1：复位无效
9 SRST_I2C0	**I2C0 软件复位** 0：复位有效 1：复位无效
8 SRST_SPI2	**SPI2 软件复位** 0：复位有效 1：复位无效
7 SRST_SPI1	**SPI1 软件复位** 0：复位有效 1：复位无效
6 SRST_SPI0	**SPI0 软件复位** 0：复位有效 1：复位无效
3 SRST_UART3	**UART3 软件复位** 0：复位有效 1：复位无效
2 SRST_UART2	**UART2 软件复位** 0：复位有效 1：复位无效
1 SRST_UART2	**UART1 软件复位** 0：复位有效 1：复位无效
0 SRST_UART0	**UART0 软件复位** 0：复位有效 1：复位无效

© 2013 - 2024 杰发科技有限公司

10.4.7 外设复位寄存器 1(CKGEN_PERI_SFT_RST1)

表 10-11 CKGEN_PERI_SFT_RST1 寄存器

CKGEN_PERI_SFT_RST1 外设复位寄存器 1 复位值：**0x00001FF4**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				SRST_CAN3	SRST_CAN2	SRST_CAN1	SRST_CAN0	SRST_CRC	SRST_EWDG	SRST_WDG	SRST_GPIO			SRST_DMA		
访问				RW	RW	RW	RW	RW	RW	RW	RW			RW		
复位值				1	1	1	1	1	1	1	1			1		

字段	说明
12 SRST_CAN3	**CAN3 软件复位** 0：复位有效 1：复位无效
11 SRST_CAN2	**CAN2 软件复位** 0：复位有效 1：复位无效
10 SRST_CAN1	**CAN1 软件复位** 0：复位有效 1：复位无效
9 SRST_CAN0	**CAN0 软件复位** 0：复位有效 1：复位无效
8 SRST_CRC	**CRC 软件复位** 0：复位有效 1：复位无效
7 SRST_EWDG	**EWDG 软件复位** 0：复位有效 1：复位无效
6 SRST_WDG	**WDG 软件复位**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：复位有效 1：复位无效
5 SRST_GPIO	**GPIO 软件 AHB 复位** 0：复位有效 1：复位无效
2 SRST_DMA	**DMA 软件复位** 0：复位有效 1：复位无效

10.4.8 外设复位寄存器 2(CKGEN_PERI_SFT_RST2)

表 10-12 CKGEN_PERI_SFT_RST2 寄存器

CKGEN_PERI_SFT_RST2　　外设复位寄存器 2　　复位值：0x00007F02

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SRST_EIO	SRST_TIMER	SRST_ADC1	SRST_ADC0	SRST_PDT1	SRST_PDT0	SRST_ACMP							SRST_CTU	
访问		RW	RW	RW	RW	RW	RW	RW							RW	
复位值		1	1	1	1	1	1	1							1	

字段	说明
14 SRST_EIO	**EIO 软件复位** 0：复位有效 1：复位无效
13 SRST_TIMER	**TIMER 软件复位** 0：复位有效 1：复位无效
12 SRST_ADC1	**ADC1 软件复位** 0：复位有效 1：复位无效
11 SRST_ADC0	**ADC0 软件复位** 0：复位有效

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：复位无效
10 SRST_PDT1	**PDT1 软件复位** 0：复位有效 1：复位无效
9 SRST_PDT0	**PDT0 软件复位** 0：复位有效 1：复位无效
8 SRST_ACMP	**ACMP 软件复位** 0：复位有效 1：复位无效
1 SRST_CTU	**CTU 软件复位** 0：复位有效 1：复位无效

10.4.9 时钟分频控制寄存器 1(CKGEN_CLK_DIV1)

表 10-13 CKGEN_CLK_DIV1 寄存器

CKGEN_CLK_DIV1 **时钟分频控制寄存器 1** **复位值：0x00000001**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									HSE_DIV1						HSI_DIV1	
访问									RW						RW	
复位值									0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HSI_DIV1				VHSI_DIV1						SPLL_DIV1					
访问	RW				RW						RW					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

字段	说明
23:18 HSE_DIV1	**HSE_DIV1** 0: 1 分频 1: 2 分频 2: 3 分频 3: 4 分频 …… 63: 64 分频
17:12	**HSI_DIV1**

© 2013 - 2024 杰发科技有限公司

字段	说明
HSI_DIV1	
	0: 1 分频
	1: 2 分频
	2: 3 分频
	3: 4 分频
	……
	63: 64 分频
11:6 VHSI_DIV1	**VHSI_DIV1**
	0: 1 分频
	1: 2 分频
	2: 3 分频
	3: 4 分频
	……
	63: 64 分频
5:0 SPLL_DIV1	**SPLL_DIV1**
	0: 1 分频
	1: 2 分频
	2: 3 分频
	3: 4 分频
	……
	63: 64 分频

10.4.10 时钟分频控制寄存器 2(CKGEN_CLK_DIV2)

表 10-14 CKGEN_CLK_DIV2 寄存器

CKGEN_CLK_DIV2 **时钟分频控制寄存器 2** **复位值：0x00000001**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									HSE_DIV2						HSI_DIV2	
访问									RW						RW	
复位值									0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HSI_DIV2				VHSI_DIV2						SPLL_DIV2					
访问	RW				RW						RW					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

字段	说明
23:18 HSE_DIV2	**HSE_DIV2**
	0: 1 分频
	1: 2 分频

© 2013 - 2024 杰发科技有限公司

字段	说明
	2: 3 分频 3: 4 分频 …… 63: 64 分频
17:12 HSI_DIV2	**HSI_DIV2** 0: 1 分频 1: 2 分频 2: 3 分频 3: 4 分频 …… 63: 64 分频
11:6 VHSI_DIV2	**VHSI_DIV2** 0: 1 分频 1: 2 分频 2: 3 分频 3: 4 分频 …… 63: 64 分频
5:0 SPLL_DIV2	**SPLL_DIV2** 0: 1 分频 1: 2 分频 2: 3 分频 3: 4 分频 …… 63: 64 分频

10.4.11 外设时钟选择寄存器 0(CKGEN_PERI_CLK_MUX0)

表 10- 15 CKGEN_PERI_CLK_MUX0 寄存器

CKGEN_PERI_CLK_MUX0　　外设时钟选择寄存器 0　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		ADC1_MUX				ADC0_MUX				SPI2_MUX				SPI1_MUX		
访问		RW				RW				RW				RW		
复位值		0	0	0		0	0	0		0	0	0		0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SPI0_MUX				TIMER_MUX								I2C0_MUX		
访问		RW				RW								RW		
复位值		0	0	0		0	0	0						0	0	0

© 2013 - 2024 杰发科技有限公司

字段	说明
30: 28 ADC1_MUX	**ADC1 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
26: 24 ADC0_MUX	**ADC0 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
22: 20 SPI2_MUX	**SPI2 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
18: 16 SPI1_MUX	**SPI1 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
14:12 SPI0_MUX	**SPI0 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
10: 8 TIMER_MUX	**TIMER 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
2: 0 I2C0_MUX	**I2C0 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK

© 2013 - 2024 杰发科技有限公司

字段	说明
	101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK

10.4.12 外设时钟选择寄存器 1(CKGEN_PERI_CLK_MUX1)

表 10- 16 CKGEN_PERI_CLK_MUX1 寄存器

CKGEN_PERI_CLK_MUX1 外设时钟选择寄存器 1 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														EIO_MUX		
访问														RW		
复位值	0	0	0	0										0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PCT_MUX							CAN3_MUX		CAN2_MUX		CAN1_MUX		CAN0_MUX	
访问		RW							RW		RW		RW		RW	
复位值		0	0	0					0	0	0	0	0	0	0	0

字段	说明
18: 16 EIO_MUX	**EIO 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
14: 12 PCT_MUX	**PCT 时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
7: 6 CAN3_MUX	**CAN3 时钟源选择** 00b: 时钟关闭 01b: HSEDIV2_CLK 10b: SYS_CLK 11b: SPLLDIV2_CLK
5: 4 CAN2_MUX	**CAN2 时钟源选择** 00b: 时钟关闭 01b: HSEDIV2_CLK

© 2013 - 2024 杰发科技有限公司

字段	说明
	10b: SYS_CLK 11b: SPLLDIV2_CLK
3: 2 CAN1_MUX	**CAN1 时钟源选择** 00b: 时钟关闭 01b: HSEDIV2_CLK 10b: SYS_CLK 11b: SPLLDIV2_CLK
1: 0 CAN0_MUX	**CAN0 时钟源选择** 00b: 时钟关闭 01b: HSEDIV2_CLK 10b: SYS_CLK 11b: SPLLDIV2_CLK

10.4.13 外设时钟选择寄存器 2(CKGEN_PERI_CLK_MUX2)

表 10-17 CKGEN_PERI_CLK_MUX2 寄存器

CKGEN_PERI_CLK_MUX2　　外设时钟选择寄存器 2　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		UART3_MUX				UART2_MUX				UART1_MUX				UART0_MUX		
访问		RW				RW				RW				RW		
复位值		0	0	0		0	0	0		0	0	0		0	0	0

字段	说明
14:12 UART3_MUX	UART3 **时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
10:8 UART2_MUX	UART2 **时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
6:4	UART1 **时钟源选择**

© 2013 - 2024 杰发科技有限公司

字段	说明
UART1_MUX	0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK
2:0 UART0_MUX	UART0 **时钟源选择** 0xxb: 时钟关闭 100b: HSEDIV2_CLK 101b: HSIDIV2_CLK 110b: VHSIDIV2_CLK 111b: SPLLDIV2_CLK

10.4.14 外设时钟选择寄存器 3(CKGEN_PERI_CLK_MUX3)

表 10- 18 CKGEN_PERI_CLK_MUX3 寄存器

CKGEN_PERI_CLK_MUX3　　外设时钟选择寄存器 3　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			PWM5_EXT_IN	PWM5_INT		PWM5_EXT		PWM4_EXT_IN	PWM4_INT		PWM4_EXT		PWM3_EXT_IN	PWM3_INT		PWM4_EXT
访问			RW	RW		RW		RW	RW		RW		RW	RW		RW
复位值			0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PWM4_EXT	PWM2_EXT_IN	PWM2_INT		PWM2_EXT		PWM1_EXT_IN	PWM1_INT		PWM1_EXT		PWM0_EXT_IN	PWM0_INT		PWM0_EXT	
访问	RW	RW	RW		RW		RW	RW		RW		RW	RW		RW	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
29 PWM5_EXT_IN	**PWM5 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
28:27 PWM5_INT	**PWM5 内部时钟选择** 00b: HSEDIV1_CLK

© 2013 - 2024 杰发科技有限公司

字段	说明
	01b: HSIDIV1_CLK 10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
26:25 PWM5_EXT	**PWM5 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2
24 PWM4_EXT_IN	**PWM4 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
23:22 PWM4_INT	**PWM4 内部时钟选择** 00b: HSEDIV1_CLK 01b: HSIDIV1_CLK 10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
21:20 PWM4_EXT	**PWM4 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2
19 PWM3_EXT_IN	**PWM3 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
18:17 PWM3_INT	**PWM3 内部时钟选择** 00b: HSEDIV1_CLK 01b: HSIDIV1_CLK 10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
16:15 PWM3_EXT	**PWM3 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2
14 PWM2_EXT_IN	**PWM2 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
13:12 PWM2_INT	**PWM2 内部时钟选择** 00b: HSEDIV1_CLK 01b: HSIDIV1_CLK

© 2013 - 2024 杰发科技有限公司

字段	说明
	10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
11:10 PWM2_EXT	**PWM2 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2
9 PWM1_EXT_IN	**PWM1 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
8:7 PWM1_INT	**PWM1 内部时钟选择** 00b: HSEDIV1_CLK 01b: HSIDIV1_CLK 10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
6:5 PWM1_EXT	**PWM1 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2
4 PWM0_EXT_IN	**PWM0 内部外部时钟选择** 0: 内部时钟 1: 外部时钟
3:2 PWM0_INT	**PWM0 内部时钟选择** 00b: HSEDIV1_CLK 01b: HSIDIV1_CLK 10b: VHSIDIV1_CLK 11b: SPLLDIV1_CLK
1:0 PWM0_EXT	**PWM0 外部时钟选择** 00b: PWM_CLK0 01b: PWM_CLK1 10b: PWM_CLK2

10.4.15 时钟输出配置寄存器(CKGEN_CLK_OUT_CFG)

表 10-19 CKGEN_CLK_OUT_CFG 寄存器

CKGEN_CLK_OUT_CFG　　时钟输出配置寄存器　　复位值：0x00000000

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					ENABLE	DIV			MUX2					MUX1		
访问					RW	RW			RW					RW		
复位值					0	0	0	0	0	0	0	0		0	0	0

字段	说明
11 ENABLE	**时钟输出使能** 0: 不使能 1: 使能
10:8 DIV	**时钟输出分频** 000b: 1 分频 001b: 2 分频 010b: 3 分频 …… 111b: 8 分频
7:4 MUX2	**时钟输出 MUX2 选择器** 0: CLK_OUT_MUX1 1: RTC_CLK 2: LSI_CLK 3: LSI128K_CLK 4: HSEDIV2_CLK 5: VHSIDIV2_CLK 6: HSIDIV2_CLK 7: SPLLDIV2_CLK 8: BUS_CLK 9: FLASH_CLK 10: SYS_CLK
2:0 MUX1	**时钟输出 MUX1 选择器** 000b: HSE CLK 001b: HSI CLK 010b: VHSI CLK 011b: SPLL CLK 100b: FLASH CLK

© 2013 - 2024 杰发科技有限公司

10.4.16 外设时钟分频寄存器(CKGEN_PERI_CLK_DIV)

表 10-20 CKGEN_PERI_CLK_DIV 寄存器

CKGEN_PERI_CLK_DIV 外设时钟分频寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	TPIU_DIV				PCT_DIV										CAN3_TS_DIV	
访问	RW				RW										RW	
复位值	0	0	0	0	0	0	0	0							0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		CAN2_TS_DIV			CAN1_TS_DIV			CAN0_TS_DIV					CAN3_DIV	CAN2_DIV	CAN1_DIV	CAN0_DIV
访问	RW	RW			RW			RW					RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0			0	0	0	0

字段	说明
31:28 TPIU_DIV	**TPIU 时钟分频** 0000b: 1 分频 0001b: 2 分频 0010b: 3 分频 ..... 1111b: 16 分频
27:24 PCT_DIV	**PCT 时钟分频** 0000b: 1 分频 0001b: 2 分频 0010b: 3 分频 ..... 1111b: 16 分频
17:15 CAN3_TS_DIV	**CAN3 时间戳时钟分频** 000b: 1 分频 001b: 2 分频 010b: 3 分频 .... 111b: 8 分频
14:12 CAN2_TS_DIV	**CAN2 时间戳时钟分频** 000b: 1 分频 001b: 2 分频 010b: 3 分频 ....

© 2013 - 2024 杰发科技有限公司

字段	说明
	111b: 8 分频
11:9 CAN1_TS_DIV	**CAN1 时间戳时钟分频** 000b: 1 分频 001b: 2 分频 010b: 3 分频 .... 111b: 8 分频
8:6 CAN0_TS_DIV	**CAN0 时间戳时钟分频** 000b: 1 分频 001b: 2 分频 010b: 3 分频 .... 111b: 8 分频
3 CAN3_DIV	**CAN3 功能时钟分频** 0: 1 分频 1: 2 分频
2 CAN2_DIV	**CAN2 功能时钟分频** 0: 1 分频 1: 2 分频
1 CAN1_DIV	**CAN1 功能时钟分频** 0: 1 分频 1: 2 分频
0 CAN0_DIV	**CAN0 功能时钟分频** 0: 1 分频 1: 2 分频

© 2013 - 2024 杰发科技有限公司

10.4.17 PLL 配置寄存器 0(ANA_SPLL_CFG0)

表 10-21 ANA_SPLL_CFG0 寄存器

ANA_SPLL_CFG0 **SPLL 配置寄存器 0** **复位值：0x14079018**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PREDIV		POSDIV							FBKDIV						
访问	RW		RW							RW						
复位值	0	0	0	1	0	1	0			0	0	0	0	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																
访问	RW															
复位值	1															

字段	说明
31：30 PREDIV	**PLL 预分频比** 00b：Fref = Fin/1 01b：Fref = Fin/2 1xb：Fref = Fin/4
29：25 POSDIV	**PLL 单端时钟输出的分频比** 00000b: 无 00001b: VCO/2 00010b: VCO/4 00011b: VCO/6 …… 11110b: VCO/60 11111b: VCO/62
22：15 FBKDIV	**PLL 反馈系数** 05h: 系数为 5 06h: 系数为 6 07h: 系数为 7 08h: 系数为 8 …… ffh: 系数为 255

10.4.18 PLL 配置寄存器 1(ANA_SPLL_CFG1)

表 10-22 ANA_SPLL_CFG1 寄存器

© 2013 - 2024 杰发科技有限公司

ANA_SPLL_CFG1 **SPLL 配置寄存器 1** **复位值：0x00F001C2**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								LD_EN	LD_SAMP0_EN	LD_SAMP1_EN			LD_DLY_SEL			
访问								RW	RW	RW			RW			
复位值								1	1	1			0	0	1	

字段	说明
8 LD_EN	**PLL Lock 检测使能** 0： 关闭 1： 使能
7 LD_SAMP0_EN	**PLL Lock 检测采样 0 使能** 0：关闭 1：使能
6 LD_SAMP1_EN	**PLL Lock 检测采样 1 使能** 0： 关闭 1： 使能
3：1 LD_DLY_SEL	**PLL Lock 检测比较时间选择** 000b: 300ps 001b: 600ps 010b: 900ps 011b: 1200ps 100b: 2200ps 101b: 3200ps 110b: 4200ps 111b: 5200ps

© 2013 - 2024 杰发科技有限公司

11 时钟监控单元（CMU）

11.1 简介

时钟监控单元(Clock Monitor Unit, CMU)主要有两大功能：

- 检测时钟频率的偏差是否超出最大的允许范围；
- 检测时钟是否丢失。

11.2 特性

- 如果被监控的频率大于参考时钟的最大值（HFREF），产生一个频率大于高阈值（FHH）事件
- 如果被监控的频率小于参考时钟的最小值（LFREF），产生一个频率小于低阈值（FLL）事件
- 当被监控的时钟丢失时，会产生 FLL 事件
- 参考时钟计数的窗口时间支持软件配置

© 2013 - 2024 杰发科技有限公司

11.3 结构框图

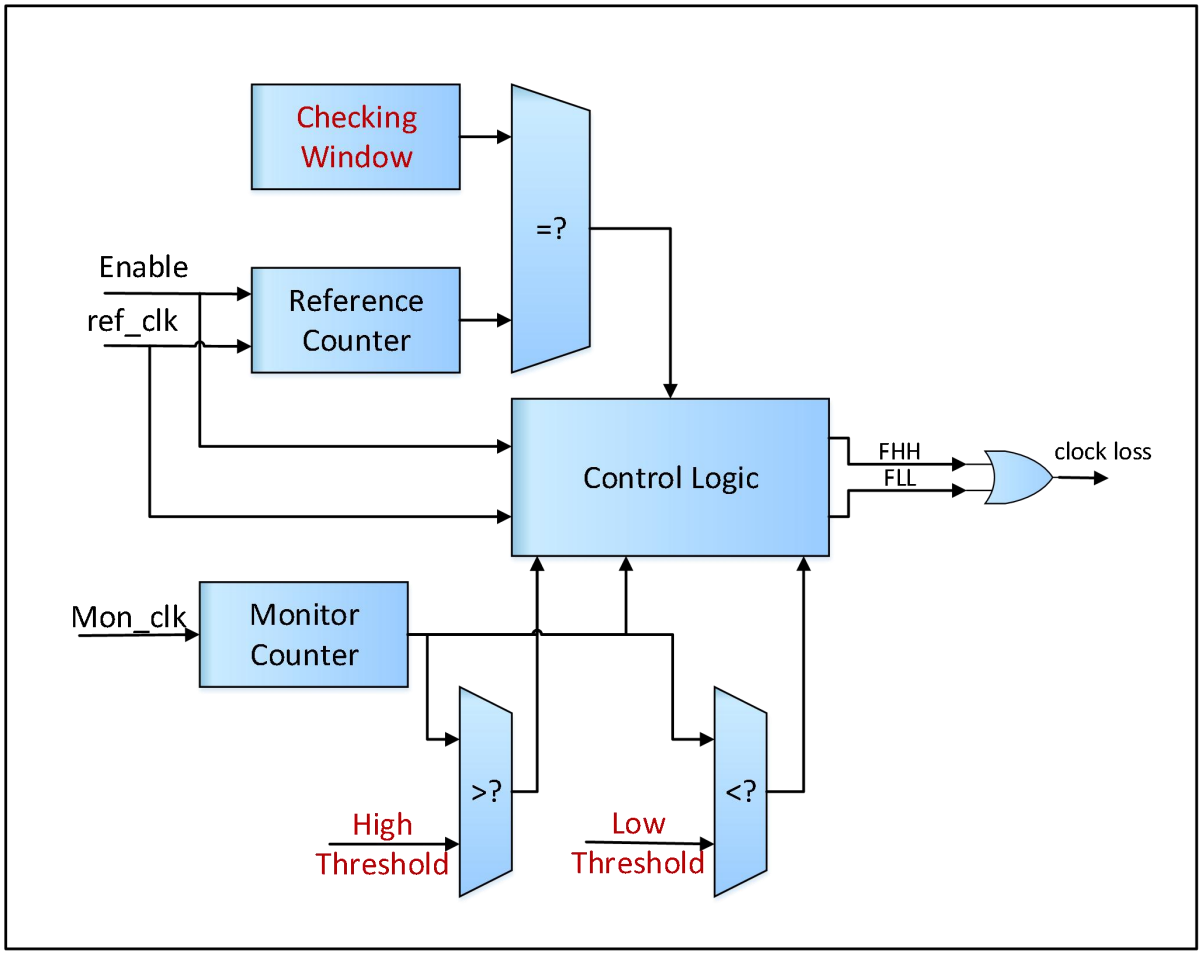


图 11-1 CMU 模块结构框图

该模块的主要输入信号有两个：参考时钟 ref_clk 和被监测的时钟 mon_clk。CMU 模块会把监控的实时计数值和用户配置的 2 个阈值进行比较，然后产生 FHH（实时计数大于配置的向上的阈值）或 FLL（实时计数小于配置的向下的阈值）事件。

© 2013 - 2024 杰发科技有限公司

11.4 功能描述

11.4.1 被监控的时钟树

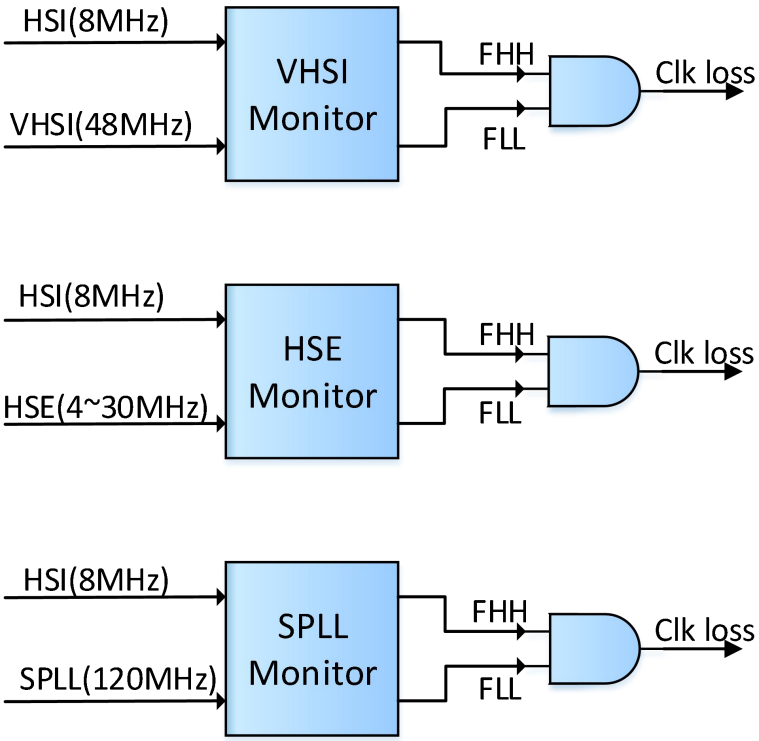


图 11-2 被监控的时钟树

CMU 模块使用 HSI 作为参考时钟。因为有 3 个 CMU 模块，且寄存器都是独立，因此支持同时监测 VHSI, HSE 和 SPLL。

© 2013 - 2024 杰发科技有限公司

11.4.2 工作流程

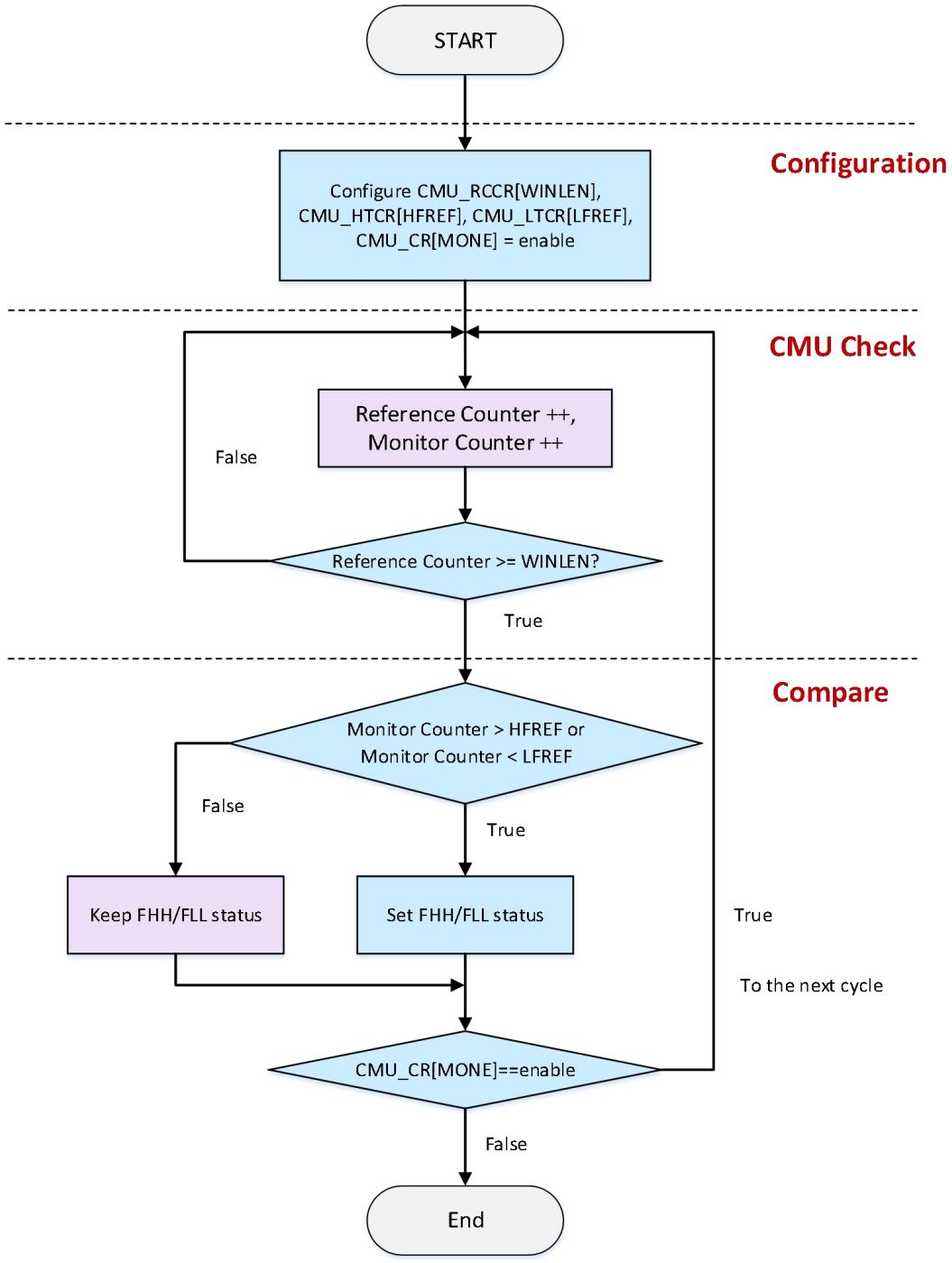


图 11-3 CMU 工作流程

当软件配置 CMU_CR[MONE] 为非 0 值，CMU 模块开始进行频率的检测。CMU 对被监控的时钟执行连续周期性的检测。每个检测窗口时间内，都会执行以下的操作：

- Stage1：在窗口时间内 CMU_RCCR[WINLEN]，CMU 会对参考时钟进行计数。同时，CMU 也会对被监控时钟进行计数。

© 2013 - 2024 杰发科技有限公司

- Stage2：窗口时间达到后，CMU 会把被监控的时钟计数值和用户配置的阈值进行比较，具体的比较方式如下：
  1. 如果被监控的时钟计数值大于高阈值 CMU_HTCR[HFREF]，FHH 事件将会产生。
  2. 如果被监控的时钟计数值小于低阈值 CMU_LTCR[LFREF]，FLL 事件将会产生。
  3. 当 FHH 或 FLL 事件被触发， clock loss 事件也会被触发。

11.4.3 中断请求

CMU 模块不支持中断。

11.4.4 调试模式

CMU 模块不支持 Debug 位控制，不受调试状态的影响。

11.5 应用说明

11.5.1 被监控的时钟丢失

如果被监视的时钟在被比较之前（Stage 2）就丢失了，CMU 在触发 FLL 事件前，会执行扩展的 Stage 2 阶段， 等待 CMU_RCCR[WINLEN] * 2 的参考时钟周期。

被监控的时钟丢失（Loss of clock, 简称 LOC）举例如下：

- 前提条件：
  - CMU 已经使能，即 CMU_CR[MONE]为非 0 值
  - 被监控的时钟丢失
  - LOC 时间要求为 5 μs
  - 假设 BUS clock 为 48 MHz， 则 $T_{BUS} = \frac{1}{48e6}$
  - 参考时钟固定为 HSI 8 MHz, $T_{REF} = \frac{1}{8e6}$
- 计算的公式如下：
  - $LOC > (2 * WINLEN + 2) * T_{REF} + 3 * T_{BUS}$

    因此， $WINLEN < \frac{5-3*0.020833}{2*0.125} - 1 = 18.75$, CMU_RCCR[WINLEN] 需要设置为小于 18。

© 2013 - 2024 杰发科技有限公司

11.5.2 窗口长度

用户在配置窗口长度 CMU_RCCR[WINLEN]时，需要考虑应用对事件响应的要求和精度。当 CMU_RCCR[WINLEN]配置为比较大的值时，检测的时间会长，对被监控的时钟检测精度会比较高。当 CMU_RCCR[WINLEN]配置为比较小的值时，检测的时间比较短，响应比较快。

CMU_RCCR[WINLEN]的最小值要求为：

$$WINLEN \geq 6 + 3 * \frac{F_{REF}}{F_{MON}}$$

11.5.3 阈值

高低阈值的配置，需要考虑下面的因素：

- 被监控的时钟频率可容忍的偏差
- 参考时钟频率(HSI)的最大偏差
- CMU 模块本身的偏差。CMU 频率监测预期的最大偏差为±3 个理想的监控时钟周期

下面是配置 CMU_HTCR[HFREF]和 CMU_LTCR[LFREF] 的例子：

- 参考时钟固定为 HSI: $F_{REF} = 8$ MHz
- 被监控的时钟假设为： $F_{MON} = 48$ MHz
- CMU_RCCR[WINLEN] = 200， 窗口长度默认值为 200
- 由于 $\frac{1}{F_{REF}} * \text{WINLEN} = \frac{1}{F_{MON}} * Monitor\ counter$

  因此，理想的被监控的时钟计数$Monitor\ counter = \frac{F_{Mon}}{F_{Ref}} * \text{WINLEN} = 1200$

考虑参考时钟和被监控时钟的组合总偏差为 ±5%：

- CMU 模块本身的精度偏差为 3 个 cycles
- 最小的被监控时钟计数 = 理想的被监控的时钟计数$Monitor\ counter$ * (100% - 5%) – 3 = 1137, 即 CMU_LTCR[LFREF] = 1137
- 最大的被监控时钟计数 = 理想的被监控的时钟计数$Monitor\ counter$ * (100% + 5%) + 3 = 1263, 即 CMU_HTCR[HFREF] = 1263

© 2013 - 2024 杰发科技有限公司

11.5.4 编程指南

CMU 软件流程配置如下：

1. 检查 CMU_SR 寄存器，保证 CMU_SR[WORK]在停止状态，并且 CMU_SR[FHH]和 CMU_SR[FLL]状态标志位都已经被清零。
2. 配置窗口长度 CMU_RCCR[WINLEN]，频率检查窗口的持续时间。
3. 配置阈值寄存器 CMU_HTCR[HFREF]和 CMU_LTCR[LFREF]。
4. CMU 监控使能，即 CMU_CR[MONE]为非 0 值。
5. 如果检测到时钟异常，会置位 CMU_SR[FHH]或 CMU_SR[FLL]或 CMU_SR[CLKLOSS]。
6. 如果用户需要重新设置相关的配置寄存器，在 CMU_SR[WORK] 在启动状态后，再配置 CMU_CR[MONE]=0， 停止 CMU 的时钟监控。

11.6 寄存器定义

表 11-1 CMU 寄存器映射

CMU_VHSI 基地址： 0x40000800
CMU_HSE 基地址： 0x40000A00
CMU_SPLL 基地址： 0x40000C00

地址	名称	宽度	描述
CMU_xx 基地址 + 0x00	CMU_CR	32	配置寄存器
CMU_xx 基地址 + 0x04	CMU_RCCR	32	参考计数配置寄存器
CMU_xx 基地址 + 0x08	CMU_HTCR	32	高阈值配置寄存器
CMU_xx 基地址 + 0x0C	CMU_LTCR	32	低阈值配置寄存器
CMU_xx 基地址 + 0x18	CMU_SR	32	状态寄存器

说明：
上表中，xx=VHSI、HSE、SPLL。

11.6.1 配置寄存器 (CMU_CR)

表 11-2 CMU_CR 寄存器

CMU_CR **配置寄存器** **复位值：0x00000000**

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														MONE		
访问														RW		
复位值														0	0	0

字段	说明
2:0 MONE	**时钟监控使能** 000b: 不使能 others: 使能

11.6.2 参考计数配置寄存器(CMU_RCCR)

表 11-3 CMU_RCCR 寄存器

CMU_RCCR **参考计数配置寄存器** **复位值：0x000000C8**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	WINLEN															
访问	RW															
复位值	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0

字段	说明
15:0 WINLEN	**参考计数窗口长度配置** 频率监测运行的参考时钟计数的数目。该位域定义一个频率检查窗口的持续时间。

© 2013 - 2024 杰发科技有限公司

11.6.3 高阈值配置寄存器 (CMU_HTCR)

表 11-4 CMU_HTCR 寄存器

CMU_HTCR 高阈值配置寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									HFREF[23:16]							
访问									RW							
复位值									0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HFREF[15:0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
23:0 HFREF	**高频率参考阈值** 定义被监控的时钟频率的高参考阈值。在监控的窗口时间内，如果被监控的时钟计数大于该阈值，则产生 FHH 事件。

11.6.4 低阈值配置寄存器 (CMU_LTCR)

表 11-5 CMU_LTCR 寄存器

CMU_LTCR 低阈值配置寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									LFREF[23:16]							
访问									RW							
复位值									0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	LFREF[15:0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
23:0 LFREF	**低频率参考阈值** 定义被监控的时钟频率的低参考阈值。在监控的窗口时间内，如果被监控的时钟计数小于该阈值，则产生 FLL 事件。

© 2013 - 2024 杰发科技有限公司

11.6.5 状态寄存器 (CMU_SR)

表 11-6 CMU_SR 寄存器

CMU_SR 状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												WORK	CHECKDONE	CLKLOSS	FHH	FLL
访问												RO	RO	RO	R/W1C	R/W1C
复位值												0	0	0	0	0

字段	说明
4 WORK	**工作状态** 0: 时钟监测停止 1: 时钟监测启动
3 CHECKDONE	**检测状态** 0: 模块未完成一次时钟检测 1: 模块至少完成一次时钟检测
2 CLKLOSS	**时钟丢失状态** 0: 时钟没有丢失 1: 时钟已丢失 **说明： FHH 或 FLL 事件产生，都会触发时钟丢失状态位置位。并且该信号硬件上已经接到 SMU 模块的单点故障通道 1~3。**
1 FHH	**频率大于高频率参考阈值状态** 0: 无 FHH 事件 1: FHH 事件已产生
0 FLL	**频率小于低频率参考阈值状态** 0: 无 FLL 事件 1: FLL 事件已产生

© 2013 - 2024 杰发科技有限公司

12 电源模式（Power Modes）

12.1 简介

本章介绍 AC7840x 各种电源模式及其各个模块在这些模式下的功能。

12.2 功能描述

12.2.1 功耗模式

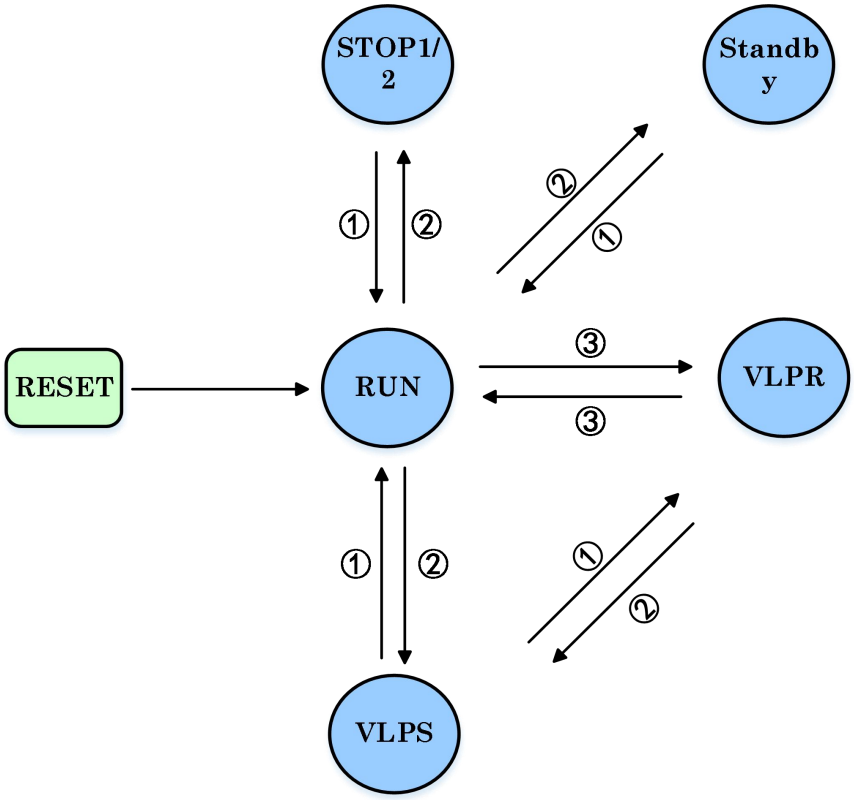
支持 RUN、STOP1、STOP2、VLPR、VLPS、Standby 6 种运行模式，分别是：

- **运行模式（RUN）** – CPU 可在全速状态下运行，MCU 复位后，默认工作模式为 RUN 模式，使用 VHSI 作为系统时钟。
- **停止模式 1（STOP1）** – CPU 进入深度休眠模式，内核时钟（Core Clock）、系统时钟（System Clock）和总线时钟（Bus Clock）均关闭。
- **停止模式 2（STOP2）** – CPU 进入深度休眠模式，内核时钟（Core Clock）和系统时钟（System Clock）关闭，总线时钟（Bus Clock）开启
- **低速运行模式（VLPR）** – CPU 可在低速状态下运行，此模式下 PLL 和 VHSI 及外部 XOSC 被关闭，只能使用 HSI 作为系统时钟。
- **低功耗停止模式（VLPS）** – CPU 进入深度休眠模式，支持部分模块中断唤醒。
- **待机模式（Standby）** – CPU 断电，特定 GPIO 和 RTC 唤醒（详见 13.4.6），GPIO 会掉电不会保持待机前 IO 状态，保留地址 0x1FFF8000 到 0x1FFFFFFF 共 32K 的 SRAM 数据，其余地址 SRAM 数据均会丢失（Standby 模式仅 AC78408/9 支持）。

12.2.2 模式转换

通过配置寄存器 SPM_PWR_MGR_CFG0[POWR_MODE] 可以进入各种功耗模式，停止模式可以通过唤醒信号唤醒回到运行模式。用户可以通过 SPM_STATUS[CURR_MODE] 寄存器查看当前系统的运行模式，模式转换如图 12-1 所示。

© 2013 - 2024 杰发科技有限公司



① 唤醒
② SPM_PWR_MGR_CFG0[POWR_MODE] + DEEPSLEEP + WFI
③ SPM_PWR_MGR_CFG0[POWR_MODE]

图 12-1 模式转换图

12.2.3 休眠应答机制

如果系统尝试进入 STOP1/STOP2/VLPS/Standby 模式，但有模块在 1s 的时间未回复休眠应答信号 ACK，则进入休眠失败，退出休眠回到运行模式。并产生休眠超时错误，SPM_STATUS[SLP_ACK_TOF]会被置位，根据 SPM_PWR_MGR_CFG0[SLP_TIMOUT_SEL]的配置，产生 SPM 中断或复位。

具体哪个模块没有回复休眠应答信号，可以查询 SPM_PERIPH_SLEEP_ACK_STATUS 得知。

举例：当 DMA 使能时，DMA 处于忙状态。此时进入 STOP1/STOP2/VLPS/Standby 模式，DMA 不会回复休眠应答信号 ACK。SPM 会等待 1s 的时间，如果在 1s 的时间内 DMA 仍未回复 ACK。则产生休眠超时错误，产生 SPM 中断或者复位。

关于各模块的休眠应答机制信息详见各模块的<低功耗模式>章节。

© 2013 - 2024 杰发科技有限公司

12.2.4 唤醒

系统进入停止模式(STOP1/STOP2/VLPS)后，能够被中断唤醒。因为 STOP2 不会关闭 BUS CLOCK，所以所有中断均可以唤醒 STOP2。但要注意的是：并不是所有的中断都能唤醒 STOP1/VLPS，只支持部分中断唤醒 STOP1/VLPS。

系统进入待机模式（Standby）后，可以通过配置 SPM_STB_WP_EN 选择开启 15 个 GPIO（PA12, PB0, PB1, PB12, PD3, PC2, PC3, PC6, PC7, PC16, PC17, PD6, PD7, PE4, PE5）或 RTC 来唤醒，唤醒后程序复位重新运行。其中 GPIO 可选择开启上升沿，下降沿，双边沿唤醒。该模式下唤醒后保留地址 0x1FFF8000 到 0x1FFFFFFF 的 32K SRAM 数据。

关于各模块的中断唤醒信息详见各模块的<低功耗模式>章节。

表 12-1 支持 Standby 唤醒的 IO 复用功能

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
5	8	8	PE5	HIGH_Z	GPIO	PWM_CLK2	PWM2_CH0	PWM2_CH3	CAN0_TX	EIO_D7	EWDG_IN
6	9	9	PE4	HIGH_Z	GPIO	HIGH_Z	PWM2_CH1	PWM2_CH2	CAN0_RX	EIO_D6	EWDG_OUT_b
20	29	42	PC3	ADC0_IN11/ACMP0_IN4	GPIO	PWM0_CH3	CAN0_TX	UART0_TX	HIGH_Z	HIGH_Z	HIGH_Z
21	30	43	PC2	ADC0_IN10/ACMP0_IN5	GPIO	PWM0_CH2	CAN0_RX	UART0_RX	HIGH_Z	HIGH_Z	HIGH_Z
22	31	44	PD7	ACMP0_IN6	GPIO	UART2_TX	HIGH_Z	PWM2_FLT3	HIGH_Z	HIGH_Z	HIGH_Z
23	32	45	PD6	ACMP0_IN7	GPIO	UART2_RX	HIGH_Z	PWM2_FLT2	HIGH_Z	HIGH_Z	HIGH_Z
27	43	56	PC17	ADC0_IN15	GPIO	PWM1_FLT3	CAN2_TX	HIGH_Z	HIGH_Z	UART3_TX	HIGH_Z
28	44	57	PC16	ADC0_IN14	GPIO	PWM1_FLT2	CAN2_RX	HIGH_Z	HIGH_Z	UART3_RX	HIGH_Z
33	53	77	PB1	ADC0_IN5/ADC1_INTERLEAVE1	GPIO	UART0_TX	SPI0_SOUT	PWM_CLK0	CAN0_TX	PWM4_CH5	HIGH_Z

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
34	54	78	PB0	ADC0_IN4/ ADC1_INTERLEAVE0	GPIO	UART0_RX	SPI0_CS0	PCT0_ALT3	CAN0_RX	PWM4_CH6	HIGH_Z
43	68	98	PB12	ADC1_IN7	GPIO	PWM0_CH0	PWM3_FLT2	CAN2_RX	HIGH_Z	HIGH_Z	HIGH_Z
45	70	101	PD3	ADC1_IN3	GPIO	PWM3_CH5	SPI1_CS0	EIO_D5	EIO_D7	TRGMUX_IN4	NMI_b
51	80	117	PC7	ADC1_IN5	GPIO	UART1_TX	CAN1_TX	PWM3_CH3	HIGH_Z	PWM1_CH0	HIGH_Z
52	81	118	PC6	ADC1_IN4	GPIO	UART1_RX	CAN1_RX	PWM3_CH2	HIGH_Z	PWM1_CH1	HIGH_Z
56	90	134	PA12	HIGH_Z	GPIO	PWM1_CH6	CAN1_RX	HIGH_Z	HIGH_Z	PWM2_CH1	HIGH_Z

12.2.5 中断请求

SPM 模块共有 2 种中断类型。

表 12-2 SPM 中断汇总

中断源	标志	本地使能	全局使能
休眠超时	SLP_ACK_TOF	SLP_TIMOUT_SEL= 0	无
LVD 低电压检测	无	默认使能	无

说明：
休眠超时触发 SPM_IRQHandler 中断，LVD 低电压检测触发 LVD_IRQHandler 中断。

12.3 应用说明

12.3.1 进入和退出停止模式

1. 使能 SPM 电源控制 SPM_PWR_MGR_CFG0[PWR_EN]；
2. 在 WFI 指令前使用 SPM_PWR_MGR_CFG0[POWR_MODE]设置停止模式；
3. 调用 DEEPSLEEP + WFI 指令进入深度休眠模式；

© 2013 - 2024 杰发科技有限公司

4. 处理器通过中断唤醒退出停止模式，唤醒后执行 WFI 下一条语句。

12.3.2 进入和退出待机模式

1. 初始化需要待机唤醒的 GPIO 或 RTC，并配置待机唤醒使能 SPM_STB_WP_EN，可配置上升沿，下降沿，双边沿唤醒；
2. 使能 SPM 电源控制 SPM_PWR_MGR_CFG0[PWR_EN]；
3. 在 WFI 指令前使用 SPM_PWR_MGR_CFG0[POWR_MODE]设置待机模式；
4. 调用 DEEPSLEEP + WFI 指令进入深度休眠模式；
5. 收到唤醒信号后，MCU 复位重新运行代码，可通过待机唤醒状态 SPM_STB_WP_STATUS 查看唤醒状态。

说明：
全速运行：模块的接口时钟与功能时钟均可正常提供，模块正常运行
异步运行：模块的接口时钟被关闭，功能时钟可正常提供，模块正常运行
关闭：模块被失能或者模块的时钟被关闭，模块不可运行
关电：模块关电

12.3.3 各模式下的模块功能

表 12-3 各模式下的模块功能

模块	RUN	STOP1/STOP2	VLPR	VLPS	Standby
CPU	全速运行	深度睡眠 (DeepSleep)	全速运行	深度睡眠 (DeepSleep)	关电
MPU	全速运行	关闭	全速运行	关闭	关电
CoreSight	全速运行	关闭	全速运行	关闭	关电
NVIC	全速运行	关闭	全速运行	关闭	关电
CTU	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
SPM	全速运行	全速运行	全速运行	全速运行	运行
CMU	全速运行	STOP1 关闭 /STOP2 全速运行	全速运行	关闭	关电
SMU	全速运行	关闭	全速运行	关闭	关电
PBR	全速运行	关闭	全速运行	关闭	关电
LVD/LVR	全速运行	全速运行	全速运行	LVD 关闭 / LVR 全速运行	关电
POR	全速运行	全速运行	全速运行	全速运行	运行

© 2013 - 2024 杰发科技有限公司

模块	RUN	STOP1/STOP2	VLPR	VLPS	Standby
T-sensor	全速运行	全速运行	全速运行	关闭	关电
DMA	全速运行	关闭	全速运行	关电	关电
WDG	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
EWDG	全速运行	STOP1 计数停止，/STOP2 全速运行	全速运行	计数停止	关电
LSI(128K)	全速运行	全速运行	全速运行	全速运行	运行
SPLL	全速运行	全速运行	关闭	关闭	关电
HSI(8M)	全速运行	全速运行	全速运行	全速运行(可选开启或关闭)	关电
VHSI(48M)	全速运行	全速运行	关闭	关闭	关电
HSE(XOSC)	全速运行	全速运行	关闭	关闭	关电
Core clock	全速运行	关闭	全速运行（最大 8M）	关闭	关电
System clock	全速运行	关闭	全速运行（最大 8M）	关闭	关电
Bus clock	全速运行	STOP1 关闭 /STOP2 全速运行	全速运行（最大 8M）	关闭	关电
UART (LIN)	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
SPI	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
I2C	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
EIO	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
CAN	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	关闭(支持低电平唤醒)	关电
CRC	全速运行	关闭	全速运行	关电	关电
PWM	全速运行	关闭	全速运行	关电	关电
TIMER	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
PDT	全速运行	关闭	全速运行	关闭	关电
PCT	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	关电
RTC	全速运行	STOP1 异步运行 /STOP2 全速运行	全速运行	异步运行	可选开启
ADC	全速运行	STOP1 异步运行 /STOP2 全速运行 (触发转换存在初始化延时)	触发转换存在初始化延时	异步运行(触发转换存在初始化延时)	关电
ACMP_DAC	全速运行	全速运行	仅支持低速比较	仅支持低速比较	关电

© 2013 - 2024 杰发科技有限公司

模块	RUN	STOP1/STOP2	VLPR	VLPS	Standby
GPIO	全速运行	STOP1 异步运行/STOP2 全速运行	全速运行	低速异步运行	部分开启
Flash	全速运行	进入低功耗省电模式	只支持读，不支持擦除和编程	进入低功耗省电模式	关电
CSE	全速运行	进入低功耗省电模式	只支持读，不支持擦除和编程	进入低功耗省电模式	关电
Memory	全速运行	进入低功耗省电模式	只支持读，不支持擦除和编程	进入低功耗省电模式	关电
RAM	全速运行	进入低功耗省电模式	全速运行	进入低功耗省电模式	部分开启

© 2013 - 2024 杰发科技有限公司

13 系统电源管理（SPM）

13.1 简介

系统电源管理 SPM (System Power Management)为软件开发人员提供了灵活的系统管理，包含休眠/唤醒功能、电源域管理和各模块功耗控制。

13.2 特性

- 支持各模式(STOP1/STOP2/VLPS/Standby/VLPR/RUN)下电源管理
- 支持低功耗的进入与退出的电源管理
- 支持时钟(HSI/VHSI/XOSC/PLL)的开启与关闭

13.3 应用说明

13.3.1 晶体振荡器（XOSC）/系统时钟（SYSPLL） 电源控制

- XOSC/SPLL 默认处于关闭状态。需要时， 通过配置 SPM_PWR_MGR_CFG1，打开或关闭。

  SPM 寄存器 SPM_PWR_MGR_CFG1：

  – XOSC_HSEEN：外部高速时钟使能。

  – XOSC_HSEBYP：外部高速时钟旁路。

  – SPLL_EN：SPLL 使能。

  当相应的位设置为 1 时， SPM 将按照上电顺序为 XOSC/SPLL 供电，可能需要花上一些时间。因此，在使用之前，软件需要等待 XOSC/SPLL 上电完成和时钟就绪。

- XOSC/PLL 上电状态可通过读取 SPM 寄存器 SPM_PWR_MGR_CFG1 来确定。

  – XOSC_RDY：外部高速时钟就绪标志。

  – SPLL_RDY：SPLL 时钟就绪标志。

  例如，在时钟源切换到 SPLL 时钟之前，应首先为 SPLL 供电，并等待 SPLL 时钟稳定。

13.3.2 内部高速时钟（HSI） / 内部超高速时钟（VHSI） 电源控制

- HSI（8MHz）和 VHSI（48MHz） 默认处于开启状态，芯片上电默认以 VHSI 做为系统时钟源。需要时， 通过配置 SPM_PWR_MGR_CFG1，打开或关闭。

  SPM 寄存器 SPM_PWR_MGR_CFG1：

© 2013 - 2024 杰发科技有限公司

- VHSI_EN： 内部超高速时钟使能。
- HSI_EN_NORMAL： RUN/STOP1/STOP2/VLPR 模式下 HSI 时钟使能。
- HSI_EN_VLPS： VLPS 模式下 HSI 时钟使能。

HSI 有两个使能位，分别对应不同的运行模式。如果需要在 VLPS 模式下进一步降低功耗，则需要让 HSI_EN_VLPS = 0。

- HSI/VHSI 上电状态可通过读取 SPM 寄存器 SPM_PWR_MGR_CFG1 来确定。
  - HSI_RDY：HSI 时钟就绪标志。
  - VHSI_RDY：VHSI 时钟就绪标志。

13.3.3 低电压检测(LVR/LVD)

- LVR (Low Voltage Reset) 低电压复位
  - 当芯片 VCC 欠压至 LVR 阈值以下，触发 LVR 复位。
  - LVR 共有两档阈值可选，见寄存器 SPM_PWR_MGR_CFG1[LVR_THL]。LVR 的低档(LVR_THL = 1)是为芯片进一步降低 VCC 供电来降低功耗设计的，建议在 VLPR/VLPS 模式下使用。
- LVD (Low Voltage Detect) 低电压检测
  - 当芯片 VCC 欠压至 LVD 阈值以下，触发 LVD 中断。
  - LVD 共有两档阈值可选，见寄存器 SPM_PWR_MGR_CFG1[LVD_THL]。LVD 的档位配置只能被 POR 复位。

13.4 寄存器定义

SPM 寄存器只能在特权模式下访问，用户模式下访问触发 hardfault。

表 13-1 SPM 寄存器映射

SPM 基地址： 0x40008000

地址	名称	宽度	描述
SPM 基地址 + 0x00	SPM_PWR_MGR_CFG0	32	电源管理器配置寄存器 0
SPM 基地址 + 0x04	SPM_PWR_MGR_CFG1	32	电源管理器配置寄存器 1
SPM 基地址 + 0x0C	SPM_PERIPH_SLEEP_ACK_STATUS	32	外设休眠应答状态寄存器
SPM 基地址 + 0x10	SPM_PERIPH_SLEEP_ACK_EN	32	外设休眠应答使能寄存器
SPM 基地址 + 0x14	SPM_STATUS	32	状态寄存器
SPM 基地址 + 0x28	SPM_STB_WP_EN	32	待机唤醒使能寄存器(仅 AC78408/9 支持)
SPM 基地址 + 0x2C	SPM_STB_WP_STATUS	32	待机唤醒状态寄存器(仅 AC78408/9 支持)

© 2013 - 2024 杰发科技有限公司

13.4.1 电源管理配置寄存器 0 (SPM_PWR_MGR_CFG0)

表 13-2 SPM_PWR_MGR_CFG0 寄存器

SPM_PWR_MGR_CFG0　　电源管理配置寄存器 0　　复位值：0x00000011

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																RESET_MASK
访问																RW
复位值																0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						POWER_MODE						SLP_TIMOUT_SEL			AWIC_EN	PWR_EN
访问						RW						RW			RW	RW
复位值						0						1			0	1

字段	说明
16 RESET_MASK	**屏蔽外部复位** 0: 不使能 1: 使能
10:8 POWER_MODE	**设置系统的运行模式** 0: RUN 模式 1: VLPR 模式 2: STOP1 模式 3: STOP2 模式 4: VLPS 模式 5: Standby 模式 **说明：所有复位后总是 RUN 模式**
4 SLP_TIMOUT_SEL	**休眠超时后的行为选择** 0: 休眠超时后，产生 SPM 中断 1: 休眠超时后，产生复位
1	**使能内核 AWIC 模块**

© 2013 - 2024 杰发科技有限公司

字段	说明
AWIC_EN	
	0: 不使能
	1: 使能
0 PWR_EN	**SPM 电源控制使能** 0：禁用 1：使能 SPM 电源控制

13.4.2 电源管理配置寄存器 1 (SPM_PWR_MGR_CFG1)

表 13-3 SPM_PWR_MGR_CFG1 寄存器

SPM_PWR_MGR_CFG1　　电源管理配置寄存器 1　　复位值：0x00000700

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			XOSC_HSEEN	XOSC_HSEBYP	SPLL_EN	XOSC_RDY	VHSI_RDY	HSI_RDY	SPLL_RDY							
访问			R/W	R/W	R/W	RO	RO	RO	RO							
复位值			0	0	0	0	0	0	0							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						VHSI_EN	HSI_EN_NORMAL	HSI_EN_VLPS							LVD_THL	LVR_THL
访问						RW	RW	RW							RW	RW
复位值						1	1	1							0	0

字段	说明
29 XOSC_HSEEN	**外部高速时钟使能** 0: 不使能 XOSC 1: 使能 XOSC
28 XOSC_HSEBYP	**外部高速时钟旁路** 0: 禁用外部时钟旁路振荡器 1: 使能外部时钟旁路振荡器
27 SPLL_EN	**SPLL 时钟使能** 0: 不使能 SPLL 1: 使能 SPLL
26 XOSC_RDY	**XOSC 时钟就绪标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：未就绪 1：就绪
25 VHSI_RDY	**VHSI 时钟就绪标志** 0：未就绪 1：就绪
24 HSI_RDY	**HSI 时钟就绪标志** 0：未就绪 1：就绪
23 SPLL_RDY	**SPLL 时钟就绪标志** 0：未就绪 1：就绪
10 VHSI_EN	**VHSI 时钟使能** 0: 不使能 VHSI(48MHZ) 1: 使能 VHSI(48MHZ) **说明：当系统时钟源为 VHSI 时，软件对该位清零不生效**
9 HSI_EN_NORMAL	**RUN/STOP1/STOP2/VLPR 模式下 HSI 时钟使能** 0: 不使能 HSI(8MHZ) 1: 使能 HSI(8MHZ) **说明：1. 不建议关闭该位，SPM 默认使用该时钟** **2. 当系统时钟源为 HSI 时，软件对该位清零不生效**
8 HSI_EN_VLPS	**VLPS 模式下 HSI 时钟使能** 0: 不使能 HSI(8MHZ) 1: 使能 HSI(8MHZ) **说明：该位只在 VLPS 模式下生效**
1 LVD_THL	**LVD 阈值选择** 0: 低档 2.9v 1: 高档 4.5v **说明：该位只能被 POR 复位**
0 LVR_THL	**LVR 阈值选择** 0: 高档 2.65V 1: 低档 2.22V **说明：LVR 的低档建议在 VLPR/VLPS 模式下使用**

© 2013 - 2024 杰发科技有限公司

13.4.3 外设休眠应答状态寄存器 (SPM_PERIPH_SLEEP_ACK_STATUS)

表 13-4 SPM_PERIPH_SLEEP_ACK_STATUS 寄存器

SPM_PERIPH_SLEEP_ACK_STATUS 外设休眠应答状态寄存器 复位值：**0xFFFFFFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称													FLASH	EIO	DMA	
访问													RO	RO	RO	
复位值													1	1	1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		UART3	UART2	UART1	UART0			CAN3	CAN2	CAN1	CAN0	SPI2	SPI1	SPI0		I2C0
访问		RO	RO	RO	RO			RO	RO	RO	RO	RO	RO	RO		RO
复位值		1	1	1	1			1	1	1	1	1	1	1		1

字段	说明
19 FLASH	**FLASH 空闲状态** 0: 忙 1: 空闲
18 EIO	**EIO 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
17 DMA	**DMA 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
14 UART3	**UART3 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
13 UART2	**UART2 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
12 UART1	**UART1 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
11 UART0	**UART0 休眠应答状态** 0: 未回 ACK

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 已回 ACK
8 CAN3	**CAN3 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
7 CAN2	**CAN2 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
6 CAN1	**CAN1 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
5 CAN0	**CAN0 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
4 SPI2	**SPI2 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
3 SPI1	**SPI1 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
2 SPI0	**SPI0 休眠应答状态** 0: 未回 ACK 1: 已回 ACK
0 I2C0	**I2C0 休眠应答状态** 0: 未回 ACK 1: 已回 ACK

13.4.4 外设休眠应答使能寄存器 (SPM_PERIPH_SLEEP_ACK_EN)

表 13-5 SPM_PERIPH_SLEEP_ACK_EN 寄存器

SPM_PERIPH_SLEEP_ACK_EN　　外设休眠应答使能寄存器　　复位值：0x000E79FD

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称													FLASH	EIO	DMA	
访问													RW	RW	RW	
复位值													1	1	1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		UART3	UART2	UART1	UART0			CAN3	CAN2	CAN1	CAN0	SPI2	SPI1	SPI0		I2C0
访问		RW	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW		RW
复位值		1	1	1	1			1	1	1	1	1	1	1		1

字段	说明
19 FLASH	**FLASH 空闲使能** 0: 不使能 1: 使能
18 EIO	**EIO 休眠应答使能** 0: 不使能 1: 使能
17 DMA	**DMA 休眠应答使能** 0: 不使能 1: 使能
14 UART3	**UART3 休眠应答使能** 0: 不使能 1: 使能
13 UART2	**UART2 休眠应答使能** 0: 不使能 1: 使能
12 UART1	**UART1 休眠应答使能** 0: 不使能 1: 使能
11 UART0	**UART0 休眠应答使能** 0: 不使能 1: 使能
8 CAN3	**CAN3 休眠应答使能** 0: 不使能 1: 使能
7	**CAN2 休眠应答使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
CAN2	
	0: 不使能
	1: 使能
6 CAN1	**CAN1 休眠应答使能** 0: 不使能 1: 使能
5 CAN0	**CAN0 休眠应答使能** 0: 不使能 1: 使能
4 SPI2	**SPI2 休眠应答使能** 0: 不使能 1: 使能
3 SPI1	**SPI1 休眠应答使能** 0: 不使能 1: 使能
2 SPI0	**SPI0 休眠应答使能** 0: 不使能 1: 使能
0 I2C0	**I2C0 休眠应答使能** 0: 不使能 1: 使能

© 2013 - 2024 杰发科技有限公司

13.4.5 状态寄存器 (SPM_STATUS)

表 13-6 SPM_STATUS 寄存器

SPM_STATUS 状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												SLP_ACK_TOF		CURR_MODE		
访问												R/W 1C		RO		
复位值												0		0	0	0

字段	说明
4 SLP_ACK_TOF	**休眠超时标志位** 0: 各个模块均回复 ACK，未发生休眠超时 1: 有模块未回复 ACK，发生休眠超时
2:0 CURR_MODE	**当前系统的运行模式** 0: RUN 模式 1: VLPR 模式 2: STOP1 模式 3: STOP2 模式 4: VLPS 模式 5: Standby 模式

© 2013 - 2024 杰发科技有限公司

13.4.6 待机唤醒使能寄存器 (SPM_STB_WP_EN)

表 13-7 SPM_STB_WP_EN 寄存器

SPM_STB_WP_EN 待机唤醒使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RTC		PE5		PE4		PD7		PD6		PC17		PC16		PC7	
访问	RW		RW		RW		RW		RW		RW		RW		RW	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PC6		PC3		PC2		PD3		PB12		PB1		PB0		PA12	
访问	RW		RW		RW		RW		RW		RW		RW		RW	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31: 30 RTC	**RTC 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 保留 11b: 双边沿唤醒使能
29: 28 PE5	**PE5 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
27: 26 PE4	**PE4 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
25: 24 PD7	**PD7 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
23: 22 PD6	**PD6 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能

© 2013 - 2024 杰发科技有限公司

字段	说明
21: 20 PC17	**PC17 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
19: 18 PC16	**PC16 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
17: 16 PC7	**PC7 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
15: 14 PC6	**PC6 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
13: 12 PC3	**PC3 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
11: 10 PC2	**PC2 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
9: 8 PD3	**PD3 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
7: 6 PB12	**PB12 唤醒使能** 00b: 不使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
5: 4 PB1	**PB1 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
3: 2 PB0	**PB0 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能
1: 0 PA12	**PA12 唤醒使能** 00b: 不使能 01b: 上升沿唤醒使能 10b: 下降沿唤醒使能 11b: 双边沿唤醒使能

13.4.7 待机唤醒状态寄存器 (SPM_STB_WP_STATUS)

表 13-8 SPM_STB_WP_STATUS 寄存器

SPM_STB_WP_STATUS　　待机唤醒状态寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																STB_WP_FLG
访问																R/WC
复位值																0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RTC	PE5	PE4	PD7	PD6	PC17	PC16	PC7	PC6	PC3	PC2	PD3	PB12	PB1	PB0	PA12
访问	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
16 STB_WP_FLG	**Standby 唤醒标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 无效 1: 有效
15 RTC	**RTC 唤醒标志** 0: 无效 1: 有效
14 PE5	**PE5 唤醒标志** 0: 无效 1: 有效
13 PE4	**PE4 唤醒标志** 0: 无效 1: 有效
12 PD7	**PD7 唤醒标志** 0: 无效 1: 有效
11 PD6	**PD6 唤醒标志** 0: 无效 1: 有效
10 PC17	**PC17 唤醒标志** 0: 无效 1: 有效
9 PC16	**PC16 唤醒标志** 0: 无效 1: 有效
8 PC7	**PC7 唤醒标志** 0: 无效 1: 有效
7 PC6	**PC6 唤醒标志** 0: 无效 1: 有效
6 PC3	**PC3 唤醒标志** 0: 无效 1: 有效
5 PC2	**PC2 唤醒标志** 0: 无效

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 有效
4 PD3	**PD3 唤醒标志** 0: 无效 1: 有效
3 PB12	**PB12 唤醒标志** 0: 无效 1: 有效
2 PB1	**PB1 唤醒标志** 0: 无效 1: 有效
1 PB0	**PB0 唤醒标志** 0: 无效 1: 有效
0 PA12	**PA12 唤醒标志** 0: 无效 1: 有效

© 2013 - 2024 杰发科技有限公司

14 安全管理单元(SMU)

14.1 简介

安全管理单元（Safety Management Unit，SMU）主要的作用是从硬件功能安全模块软件功能安全接收错误相关的信号，从而复位请求。该复位请求会送到 RCM 模块，由 RCM 模块进行复位。

14.2 特性

- 支持单点故障以及多点故障状态检测
- 当收到从硬件功能安全模块发出的错误信号，产生复位请求
- 支持软件配置捕获到一定次数的错误信号，才产生复位请求
- 复位功能可屏蔽
- 寄存器写保护功能
- SMU 功能支持自检测
- 带影子寄存器，可以保存 SMU 复位前的值（包含错误状态，错误次数）

© 2013 - 2024 杰发科技有限公司

14.3 结构框图

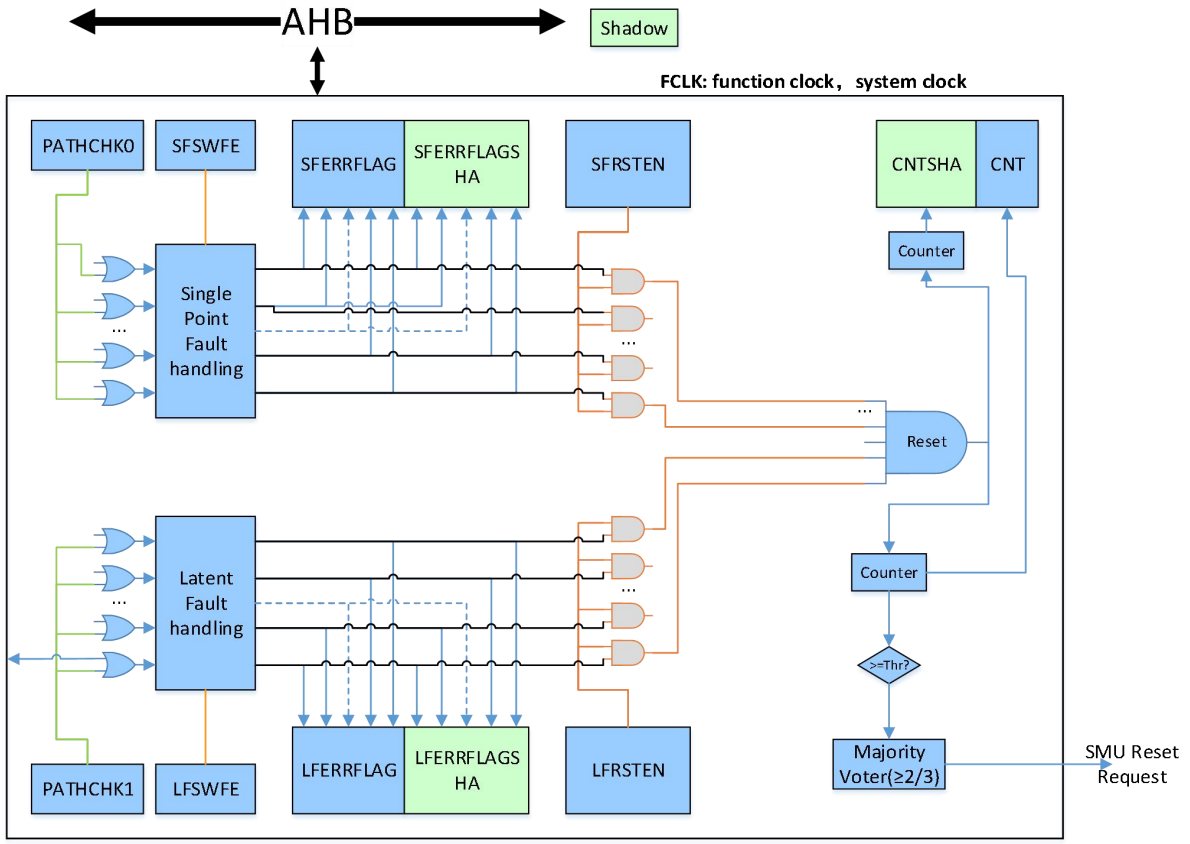


图 14-1 SMU 模块结构框图

SMU 模块在下面的情况下，可以产生复位请求：

1. 自检测功能：用户可以通过配置 SMU_PATHCHK0 寄存器测试所有单点故障通道的通路，也可以通过配置 SMU_PATHCHK1 寄存器测试所有潜在故障通道的通路。
2. 软件的功能安全：当软件安全错误产生后，通过配置 SMU_SWSFE 和 SMU_SWLFE 可以置位错误状态。如果达到错误计数阈值，可以触发复位请求。
3. 硬件的功能安全：当硬件安全模块错误产生后，可以置位错误状态。如果达到错误计数阈值，可以触发复位请求。

14.4 功能描述

14.4.1 错误信号源

SMU 模块错误的信号源有：

1. 硬件功能安全模块：硬件单点错误信号或硬件潜在错误信号。

© 2013 - 2024 杰发科技有限公司

2. 软件故障源：软件单点错误信号或软件潜在错误信号。
3. 自检测寄存器：单点错误通道自检测寄存器 SMU_PATHCHK0 和潜在错误通道自检测寄存器 SMU_PATHCHK1。

14.4.2 复位触发

单点错误通道或潜在错误通道， 当产生错误时，都可以选择是否触发复位。用户可以通过配置单点错误复位使能寄存器 SMU_SFRSTEN 和潜在错误复位使能寄存器 SMU_LFRSTEN 进行选择 。

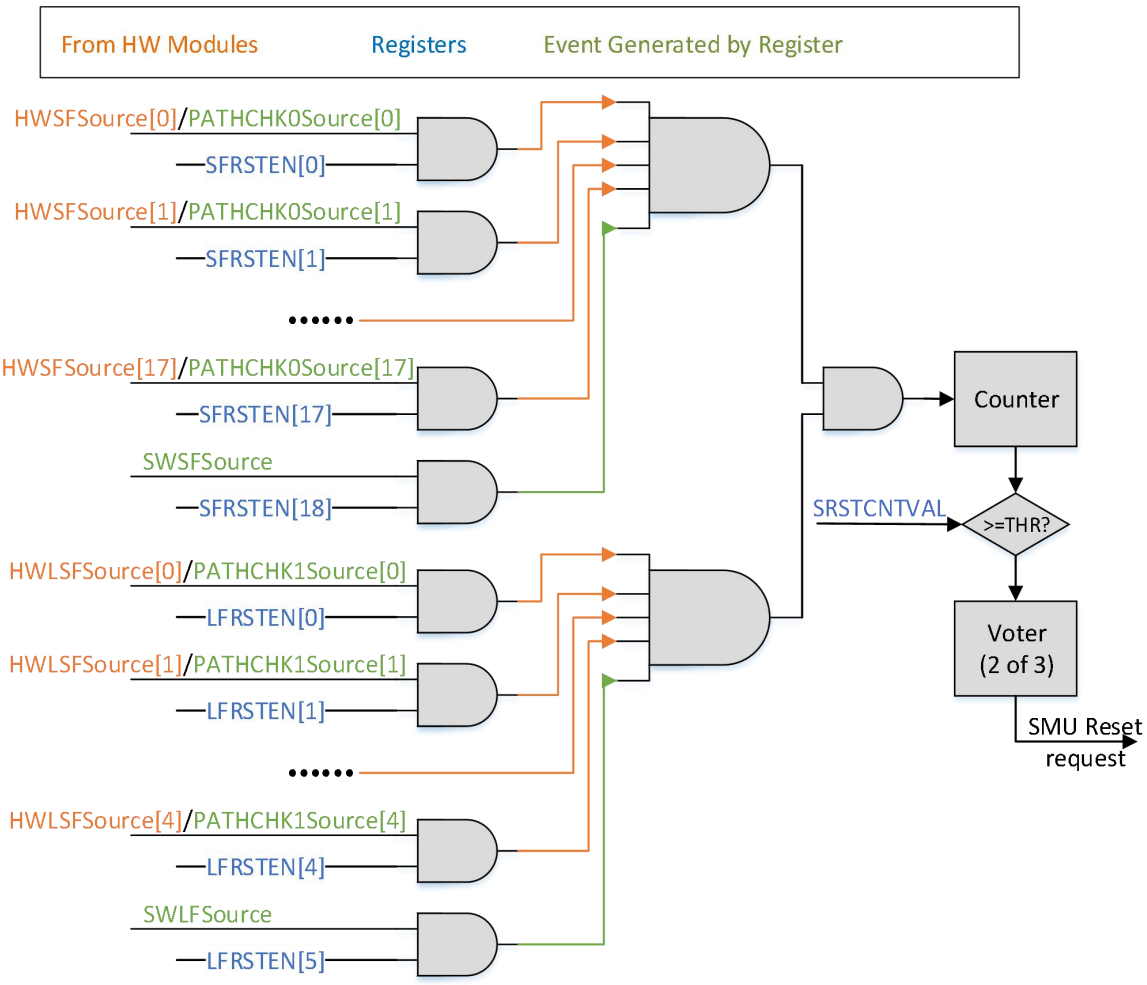


图 14-2 复位触发框图

14.4.3 硬件错误源列表

表 14-1 硬件单点错误源列表

通道号	模块	条件	中断或复位使能 (SFINTEN, SFRSTEN)
0	Analog	PLL 时钟失锁	bit 0

© 2013 - 2024 杰发科技有限公司

通道号	模块	条件	中断或复位使能 (SFINTEN, SFRSTEN)
1	System PLL Clock Monitor	PLL 时钟丢失（由 CMU 模块的时钟丢失事件触发）	bit 1
2	HSE Clock Monitor	HSE 时钟丢失（由 CMU 模块的时钟丢失事件触发）	bit 2
3	VHSI Clock Monitor	VHSI 时钟丢失（由 CMU 模块的时钟丢失事件触发）	bit 3
4	SRAM	ECC 1 bit 错误	bit 4
5	SRAM	ECC 2 bits 错误	bit 5
6	E2E	端到端奇偶校验错误	bit 6
7	DMA	奇偶校验错误	bit 7
8	Flash	ECC 错误	bit 8
9	PDT1	在一定的时间计数内，触发的次数与配置的次数不一致，产生错误	bit 9
10	PDT0	在一定的时间计数内，触发的次数与配置的次数不一致，产生错误	bit 10
11	WDG	看门狗超时	bit 11
12	Analog	Low voltage detection of VVDD after MTCMOS long chain	bit 12
13	Analog	DIGLDO 对于 FLASH 供电电压的低电压检测	bit 13
14	Analog	FLHLDO 低电压检测信号	bit 14
15	Cache	奇偶校验错误	bit 15
16	CPU	CPU Lock Up	bit 16
17	Flex SRAM	ECC 错误	bit 17

表 14- 2 硬件潜在错误源列表

通道号	模块	条件	中断或复位使能 (LFINTEN, LFRSTEN)
0	SRAM	EIM 2 bits ECC 错误	bit 0
1	SRAM	EIM 1 bit ECC 错误	bit 1
2	Flash	EIM ECC 错误	bit 2
3	ABIST	ABIST 错误	bit 3
4	MBIST	MBIST 错误	bit 4

© 2013 - 2024 杰发科技有限公司

14.5 应用说明

SMU 复位触发的软件配置流程如下：

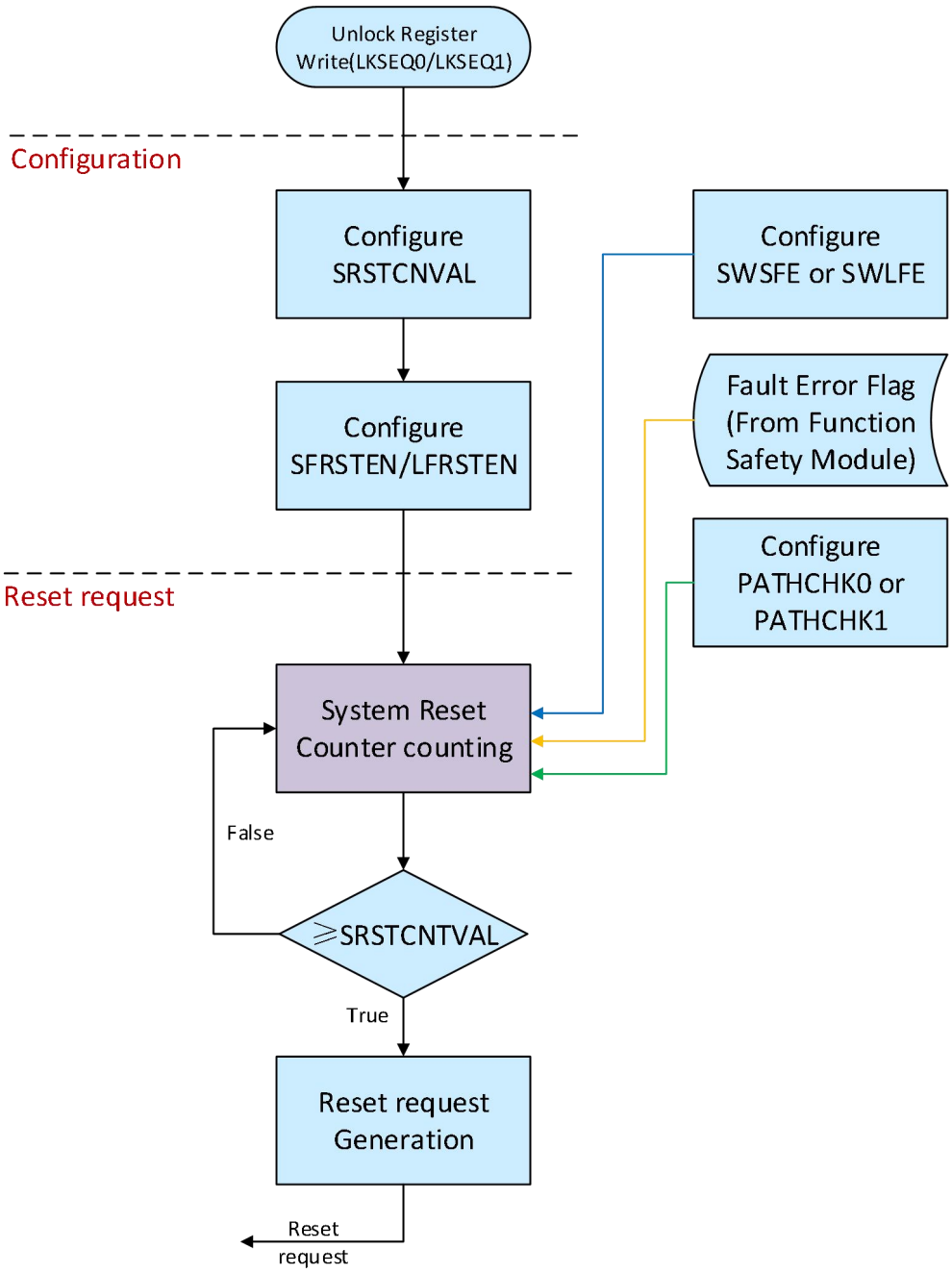


图 14-3 SMU 复位触发软件配置流程

1. 对单点错误通道和潜在错误通道的相关配置寄存器进行解锁，通过配置 SMU_LKSEQ0，SMU_LKSEQ1 寄存器进行解锁。
2. 配置系统复位请求计数阈值寄存器 SMU_SRSTCNTVAL，该配置决定了错误通道向 SMU 模块发出系统复位的请求达到该阈值后，才能真正触发系统复位。

© 2013 - 2024 杰发科技有限公司

3. 配置复位使能，单点错误通道可以通过配置 SMU_SFRSTEN 寄存器进行使能。潜在错误通道可以通过配置 SMU_LFRSTEN 寄存器进行使能。

4. 当硬件功能安全模块发出错误信号，或通过配置单点错误通道自检测寄存器 SMU_PATHCHK0，潜在错误通道自检测寄存器 SMU_PATHCHK1，或者通过配置软件单点错误使能寄存器 SMU_SWSFE，软件潜在错误使能寄存器 SMU_SWLFE，即可以触发复位请求。当复位请求达到配置的阈值后，才能触发系统复位。

5. 用户可以通过检查单点错误状态寄存器 SMU_SFES，潜在错误状态寄存器 SMU_LFES 进行检查是否产生错误的状态。这两个寄存器在系统复位后，寄存器的值会被复位。如果用户需要读取复位前的错误状态值，可以通过读取单点错误状态影子寄存器 SMU_SFESS 和潜在错误状态影子寄存器 SMU_LFESS， 影子寄存器只有 POR 才能复位。

说明：
配置自检测寄存器 SMU_PATHCHK1 bit3,bit4 不支持计数并触发复位，只支持触发 SMU_LFES 错误状态位。

14.6 寄存器定义

表 14-3 SMU 寄存器映射

SMU 基地址： 0x40089000

地址	名称	宽度	描述
SMU 基地址 + 0x00	SMU_SFES	32	单点错误状态寄存器
SMU 基地址 + 0x04	SMU_SFESS	32	单点错误状态影子寄存器
SMU 基地址 + 0x08	SMU_LFES	32	潜在错误状态寄存器
SMU 基地址 + 0x0C	SMU_LFESS	32	潜在错误状态影子寄存器
SMU 基地址 + 0x10	SMU_LKSEQ0	32	单点错误解锁时序寄存器
SMU 基地址 + 0x14	SMU_LKSEQ1	32	潜在错误解锁时序寄存器
SMU 基地址 + 0x18	SMU_SWSFE	32	软件单点错误使能寄存器
SMU 基地址 + 0x1C	SMU_SWSFES	32	软件单点错误使能影子寄存器
SMU 基地址 + 0x20	SMU_SWLFE	32	软件潜在错误使能寄存器
SMU 基地址 + 0x24	SMU_SWLFES	32	软件潜在错误使能影子寄存器
SMU 基地址 + 0x30	SMU_SFRSTEN	32	单点错误复位使能寄存器
SMU 基地址 + 0x34	SMU_LFRSTEN	32	潜在错误复位使能寄存器
SMU 基地址 + 0x38	SMU_SRSTCNTVAL	32	系统复位请求计数阈值寄存器
SMU 基地址 + 0x3C	SMU_SRSTCNT	32	系统复位请求计数寄存器
SMU 基地址 + 0x40	SMU_SRSTCNTS	32	系统复位请求计数影子寄存器
SMU 基地址 + 0x44	SMU_PATHCHK0	32	单点错误通道自检测寄存器
SMU 基地址 + 0x48	SMU_PATHCHK1	32	潜在错误通道自检测寄存器

© 2013 - 2024 杰发科技有限公司

说明：
所有的影子寄存器，只有 POR 才能复位。其它的寄存器，系统复位即可以复位。

14.6.1 单点错误状态寄存器 (SMU_SFES)

表 14-4 SMU_SFES 寄存器

SMU_SFES **单点错误状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														SWSFES	HWSFES[17:16]	
访问														W1C	W1C	
复位值														0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HWSFES[15:0]															
访问	W1C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
18 SWSFES	**软件单点错误状态** 触发软件功能安全故障（向 SMU_SWSFE 寄存器写“1”）时置位。写“1”可以清除该状态位。
17:0 HWSFES	**硬件单点错误状态** 每个 bit 代表一个单点错误通道的状态，高有效。写“1”可以清除该状态位。

14.6.2 单点错误状态影子寄存器 (SMU_SFESS)

表 14-5 SMU_SFESS 寄存器

SMU_SFESS **单点错误状态影子寄存器** **复位值：0x00000000**

© 2013 - 2024 杰发科技有限公司

位	31	30	29		27	26	25	24	23	22	21	20	19	18	17	16
名称														SWSFES	HWSFES[17:16]	
访问														W1C	W1C	
复位值														0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HWSFES[15:0]															
访问	W1C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
18 SWSFES	**软件单点错误状态** 触发软件功能安全故障（向 SMU_SWSFE 寄存器写“1”）时置位。写“1”可以清除该状态位。 **说明：该寄存器只有 POR 才能复位。**
17:0 HWSFES	**硬件单点错误状态** 每个 bit 代表一个单点错误通道的状态，高有效。写“1”可以清除该状态位。 **说明：该寄存器只有 POR 才能复位。**

14.6.3 潜在错误状态寄存器 (SMU_LFES)

表 14-6 SMU_LFES 寄存器

SMU_LFES **潜在错误状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称											SWLFES	HWLFES[4:0]				
访问											W1C	W1C				
复位值											0	0	0	0	0	0

© 2013 - 2024 杰发科技有限公司

字段	说明
5 SWLFES	**软件潜在错误状态** 触发软件功能安全故障（向 SMU_SWLFE 寄存器写“1”）时置位。写“1”可以清除该状态位。 **说明：该寄存器只有 POR 才能复位。**
4:0 HWLFES	**硬件潜在错误状态** 每个 bit 代表一个潜在错误通道的状态，高有效。写“1”可以清除该状态位。

14.6.4 潜在错误状态影子寄存器 (SMU_LFESS)

表 14-7 SMU_LFESS 寄存器

SMU_LFESS **潜在错误状态影子寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称											SWLFES	HWLFES[4:0]				
访问											W1C	W1C				
复位值											0	0	0	0	0	0

字段	说明
5 SWLFES	**软件潜在错误状态** 触发软件功能安全故障（向 SMU_SWLFE 寄存器写“1”）时置位。写“1”可以清除该状态位。 **说明：该寄存器只有 POR 才能复位。**
4:0 HWLFES	**硬件潜在错误状态** 每个 bit 代表一个潜在错误通道的状态，高有效。写“1”可以清除该状态位。 **说明：该寄存器只有 POR 才能复位。**

© 2013 - 2024 杰发科技有限公司

14.6.5 单点错误解锁时序寄存器 (SMU_LKSEQ0)

表 14-8 SMU_ LKSEQ0 寄存器

SMU_LKSEQ0 单点错误解锁时序寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													LKSEQ0			
访问													WR			
复位值													0	0	0	0

字段	说明
3:0 LKSEQ0	**单点错误通道解锁时序** 解锁时序步骤如下： 1. 先写 0xA5 2. 再写 0x5A 其它错误的写时序都会导致单点错误通道相关的配置寄存器上锁。

14.6.6 潜在错误解锁时序寄存器 (SMU_LKSEQ1)

表 14-9 SMU_ LKSEQ1 寄存器

SMU_LKSEQ1 潜在错误解锁时序寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													LKSEQ1			
访问													WR			
复位值													0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
3:0 LKSEQ1	**潜在错误通道解锁时序** 解锁时序步骤如下： 1. 先写 0x5A 2. 再写 0xA5 其它错误的写时序都会导致潜在错误通道相关的配置寄存器上锁。

14.6.7 软件单点错误使能寄存器 (SMU_SWSFE)

表 14-10 SMU_ SWSFE 寄存器

SMU_SWSFE **软件单点错误使能寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SWSFE							
访问									WR							
复位值									0	0	0	0	0	0	0	0

字段	说明
7:0 SWSFE	**软件单点错误使能通道** 当发生软件单点错误，在对应通道写“1”触发 SMU 错误。

© 2013 - 2024 杰发科技有限公司

14.6.8 软件单点错误使能影子寄存器 (SMU_SWSFES)

表 14-11 SMU_ SWSFES 寄存器

SMU_SWSFES 软件单点错误使能影子寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SWSFES							
访问									R							
复位值									0	0	0	0	0	0	0	0

字段	说明
7:0 SWSFES	软件单点错误使能影子寄存器 说明：该寄存器只有 POR 才能复位。

14.6.9 软件潜在错误使能寄存器 (SMU_SWLFE)

表 14-12 SMU_ SWLFE 寄存器

SMU_SWLFE 软件潜在错误使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SWLFE							
访问									WR							
复位值									0	0	0	0	0	0	0	0

字段	说明
7:0 SWLFE	软件潜在错误使能通道 当发生软件潜在错误，在对应通道写“1”触发 SMU 错误。

© 2013 - 2024 杰发科技有限公司

14.6.10 软件潜在错误使能影子寄存器 (SMU_ SWLFES)

表 14-13 SMU_ SWLFES 寄存器

SMU_SWLFES 软件潜在错误使能影子寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SWLFES							
访问									R							
复位值									0	0	0	0	0	0	0	0

字段	说明
7:0 SWLFES	**软件潜在错误使能影子寄存器** **说明：该寄存器只有 POR 才能复位。**

14.6.11 单点错误复位使能寄存器 (SMU_SFRSTEN)

表 14-14 SMU_SFRSTEN 寄存器

SMU_SFRSTEN 单点错误复位使能寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														SWSFRSTE	HWSFSRTE[17:16]	
访问														WR	WR	
复位值														0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	HWSFRSTE[15:0]															
访问	WR															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
18 SWSFRSTE	**软件单点错误通道复位使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	软件单点错误通道的复位使能配置，高有效。
17:0 HWSFRSTE	**硬件单点错误通道复位使能** 每个 bit 代表一个单点错误通道的复位使能配置，高有效。

14.6.12 潜在错误复位使能寄存器 (SMU_LFRSTEN)

表 14-15 SMU_LFRSTEN 寄存器

SMU_LFRSTEN　　潜在错误复位使能寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称											SWLFRSTE	HWLFRSTE[4:0]				
访问											WR	WR				
复位值											0	0	0	0	0	0

字段	说明
5 SWLFRSTE	**软件潜在错误通道复位使能** 软件潜在错误通道的复位使能配置，高有效。
4:0 HWLFINTE	**硬件潜在错误通道复位使能** 每个 bit 代表一个潜在错误通道的复位使能配置，高有效。

© 2013 - 2024 杰发科技有限公司

14.6.13 系统复位请求计数阈值寄存器(SMU_SRSTCNTVAL)

表 14-16 SMU_SRSTCNTVAL 寄存器

SMU_SRSTCNTVAL 系统复位请求计数阈值寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SRSTCNTVAL			
访问													WR			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
3:0 SRSTCNTVAL	**系统复位请求计数阈值** 当系统复位计数大于该阈值时，才会产生复位的请求。

14.6.14 系统复位请求计数寄存器(SMU_SRSTCNT)

表 14-17 SMU_SRSTCNT 寄存器

SMU_SRSTCNT 系统复位计数寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SRSTCNT			
访问													WR			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
3:0 SRSTCNT	**系统复位请求计数** 系统复位请求实时的计数值。 对该寄存器写 0xFFFFFFFF 可以清除该计数。

© 2013 - 2024 杰发科技有限公司

14.6.15 系统复位请求计数影子寄存器(SMU_SRSTCNTS)

表 14-18 SMU_SRSTCNTS 寄存器

SMU_SRSTCNTS 系统复位计数影子寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SRSTCNTS			
访问													WR			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
3:0 SRSTCNTS	**系统复位请求计数** 系统复位请求实时的计数值。 **说明：对该寄存器写 0xFFFFFFFF 可以清除该计数，并且只有 POR 才能复位该寄存器，系统复位无法复位该寄存器。**

14.6.16 单点错误通道自检测寄存器(SMU_PATHCHK0)

表 14-19 SMU_PATHCHK0 寄存器

SMU_PATHCHK0 单点错误通道自检测寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PATHCHK0EN														PATHCHK0SEL[17:16]	
访问	WR														WR	
复位值	0	0	0	0											0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PATHCHK0SEL[15:0]															
访问	WR															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
31:28 PATHCHK0EN	**硬件单点错误通道自检测开关** 1111b：自检功能使能 其它值，自检测功能都是关闭。
17:0 PATHCHK0SEL	**硬件单点错误通道自检测选择** 每个 bit 代表一个单点错误通道是否选择，高有效。

14.6.17 潜在错误通道自检测寄存器(SMU_PATHCHK1)

表 14-20 SMU_PATHCHK1 寄存器

SMU_PATHCHK1　　潜在错误通道自检测寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PATHCHK1EN															
访问	WR															
复位值	0	0	0	0												
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												PATHCHK1SEL[4:0]				
访问												WR				
复位值												0	0	0	0	0

字段	说明
31:28 PATHCHK1EN	**硬件潜在错误通道自检测开关** 1111b：自检功能使能 其它值，自检测功能都是关闭。
4:0 PATHCHK1SEL	**硬件潜在错误通道自检测选择** 每个 bit 代表一个单点错误通道是否选择，高有效。 **说明：bit3,bit4 不支持计数并触发复位，只支持触发 SMU_LFES 错误状态位。**

© 2013 - 2024 杰发科技有限公司

15 片内 Flash（Embedded Flash）

15.1 简介

片内 Flash 控制器是 Cortex™-M4F 和片内 Flash 之间的桥梁，在实际应用中，用户代码存放于片内 Flash 中，片内 Flash 启动作为唯一启动模式，在 AC7840x 系列产品中含两块 Flash 即 P-Flash 和 D-Flash。

15.2 特性

- Flash 存储器
  - P-Flash 最大支持 1MB(包括 2 个 bank 即 512KB+512KB)
  - D-Flash 最大支持 128KB
  - 寿命：P-Flash ≥ 10,000 周期，D-Flash ≥ 100,000 周期
  - 页容量：每页 2048 字节
- Flash 控制器
  - 操作列表：
    - 擦除：页擦除，块擦除，写保护信息页擦除，所有块的擦除
    - 编程：页编程，一次性编程，Section 编程，编程最小单位 64-bit
    - 读：支持 8-bit/16-bit/32-bit 宽度读数据
    - 验证：Section(可以指定验证长度)验证，块擦除验证，全部块擦除验证
  - 包含预取缓冲区(Cache)

15.3 结构框图

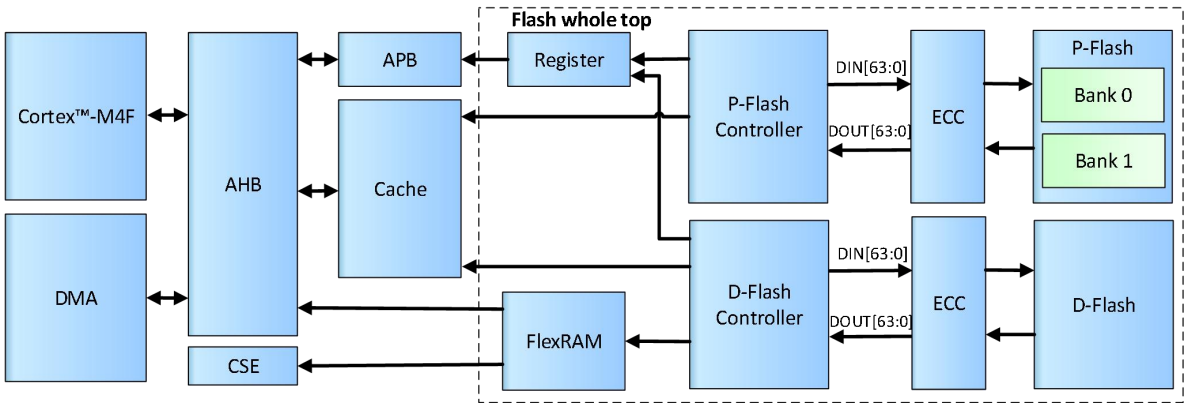


图 15-1 Flash 结构框图

© 2013 - 2024 杰发科技有限公司

15.4 功能描述

15.4.1 P-Flash 组织

表 15-1 P-Flash 存储器组织

P-Flash 存储器	名称	地址	尺寸 (字节)	用户权限 (说明)
主存储区	Page 1	0x0000 0000 ~ 0x0000 07FF	2K	Erase all blocks / Block erase/ Page erase/ Program/ Read/ Verify
	Page 2	0x0000 0800 ~ 0x0000 0FFF	2K	
	Page 3	0x0000 1000 ~ 0x0000 17FF	2K	
	Page 4	0x0000 1800 ~ 0x0000 1FFF	2K	
	…		2K	
	Page 512	0x000F F800 ~ 0x000F FFFF	2K	
信息区	Security INFO	0x0020 0000 ~ 0x0020 07FF	2K	Erase all blocks / Program/Read
	Write Protection	0x0020 0800 ~ 0x0020 0FFF	2K	Erase all blocks / Write protection page erase/ Program/Read
	Once Program Field	0x0020 1000 ~ 0x0020 17FF	2K	Read/Once program

15.4.2 P-Flash 信息区组织

表 15-2 P-Flash 信息区组织

地址范围	起始地址	位	大小（字节）	信息
0x0020 0000~0x0020 07FF	0x0020 0000		8	工厂分析授权码
	0x0020 0008		8	Backdoor key
	0x0020 0010	1：0	2	读保护使能位
		3：2		Debug 解除读保护使能位
		5：4		验证 backdoor key 使能位
		7：6		工厂分析使能位
		11：8		MBIST 使能位
		15：12		ABIST 使能位
0x0020 0800~0x0020 0FFF	0x0020 0800		4	P-Flash 写保护信息
	0x0020 0804		1	D-Flash 写保护信息
0x0020 1000~0x0020 17FF	0x0020 1000		2K	一次性编程区域

15.4.3 P-Flash 安全信息区

表 15-3 P-Flash 安全信息区

© 2013 - 2024 杰发科技有限公司

地址	位	描述	值
0x0020 0010	1：0	使能/失能 MCU 读保护，复位后生效，如果使能读保护则生效则不可以通过 JTAG/SWD 方式访问整个 MCU 空间	11b：无读保护(默认) 其他值：有读保护
	3：2	使能/失能通过 JTAG/SWD 方式解除 MCU 读保护	11b：使能(默认) 其他值：失能
	5：4	使能/失能 backdoor 功能，失能后无法通过验证 backdoor key 方式解除读保护	11b：失能(默认) 其他值：使能
	7：6	使能/失能工厂分析功能	11b：使能(默认) 其他值：失能

15.4.4 D-Flash 组织

表 15-4 D-Flash 存储器组织

D-Flash 存储区	名称	地址	大小(字节)	用户权限 (说明)
主存储区	Page 1	0x0100 0000 ~ 0x0100 07FF	2K	Erase All blocks/ Block Erase/ Page Erase/ Program/Read/Verify
	Page 2	0x0100 0800 ~ 0x0100 0FFF	2K	
	Page 3	0x0100 1000 ~ 0x0100 17FF	2K	
	…			
	Page 64	0x0101 F800 ~ 0x0101 FFFF	2K	

15.4.5 Flash 命令

表 15-5 Flash 命令 ID

操作类型	命令	命令 ID	描述
擦除	页擦除	0x01	按页擦除 P-Flash 或者 D-Flash
	块擦除	0x02	擦除整个 P-Flash 或者整个 D-Flash
	写保护信息页擦除	0x03	擦除范围是写保护选项字节区域，地址范围为：0x0020 0800 ~ 0x0020 0FFF
	全部块擦除	0x04	擦除所有块(P-Flash & D-Flash)的主存储区
编程	页编程	0x10	按照 64-bit 对齐方式编程 P-Flash 或者 D-Flash
	Section 编程	0x11	通过 FlexRAM 来进行批量编程 P-Flash 或者 D-Flash
	一次性编程	0x12	编程一次性可编程区域，地址范围为：0x0020 1000~0x0020 17FF
	设置 UID、Secret Key、Lock	0x33	设置 UID、Secret Key、Lock
验证	Section 验证擦除	0x20	验证指定长度(64-bit 为单位)空间是否被擦除
	块擦除验证	0x21	验证整个 P-Flash 主区域或者 D-Flash 主区是否被擦除
	全部块擦除验证	0x22	验证整个 P-Flash 主区域以及 D-Flash 主区是否被擦除
	验证 backdoor key	0x23	根据指定的秘钥验证是否和存储在 Info 区的秘钥匹配
分区	擦除分区	0x30	擦除分区
	分区	0x31	编程分区

© 2013 - 2024 杰发科技有限公司

操作类型	命令	命令 ID	描述
	设置 FlexRAM 功能	0x32	设置 FlexRAM 功能

15.4.6 中断请求

如表 15-6 所示，Flash 共支持三个中断向量。

表 15-6 Flash 中断汇总

中断源	标志	本地使能	全局使能
命令完成中断	FLASH_STAT[CCIF]	FLASH_CNFG[CCIE]	
冲突中断	FLASH_STAT[COLLERRF]	FLASH_CNFG[COLLE]	
ECC 2-bit 错误中断	FLASH_STAT[DFDIF]	FLASH_CNFG[DFDIE]	

1. 命令完成中断

   该中断比较特殊，由于命令完成标记 CCIF 默认值为 1，不可以软件清除该标记，有且只有当正在执行 Flash 相关命令时才会置位为 0，当设置 FLASH_CNFG[CCIE]=1 时该中断会一直产生，所以建议该中断使用方式如下：

   - 设置 1 到寄存器 FLASH_CST 触发执行 Flash 命令；
   - 当 CCIF=0 时设置寄存器 FLASH_CNFG[CCIE]=1 且使能对应的 NVIC 中断；
   - 在命令完成中断处理程序中设置寄存器 FLASH_CNFG[CCIE]=0 以及禁用对应的 NVIC 中断。

2. 冲突中断

   当执行一个 Flash 命令没有完成(即 CCIF=0)时又触发了另一个命令的执行，此时就会产生冲突中断，建议该中断使用方式如下：

   - 设置 1 到寄存器 FLASH_CST 触发执行 Flash 命令；
   - 检查 CCIF=0 时使能寄存器 FLASH_CNFG[COLLE]以及使能对应的 NVIC 中断；
   - 在冲突中断函数中检查寄存器 FLASH_STAT[COLLERRF]是否为置位为 1 来判断判断是否产生冲突中断；
   - 向寄存器 FLASH_STAT[COLLERRF]写 1 清除冲突中断标记；
   - 如果想禁用冲突中断，可设置寄存器 FLASH_CNFG[COLLE] 以及对应的 NVIC 中断为禁用状态。

3. ECC 2-bit 错误中断

   ECC 中断相关描述请参考 15.5.16 ECC 章节。

© 2013 - 2024 杰发科技有限公司

15.4.7 调试模式

Flash 模块没有调试模式配置，在进入调试模式时可正常运行。

15.5 应用说明

为了防止用户误操作 Flash，配置 Flash 寄存器前首先需要解锁否则无法写 Flash 寄存器，解锁方式参考如下步骤 1 所描述。

为了保证 Flash 正常运行，务必在系统时钟初始化完成后解锁 Flash，然后配置寄存器 FLASH_CNFG 的 CLKFREQ 位，具体配置请参考该寄存器位说明。

Flash 命令操作流程均相同，操作步骤如下：

1. 通过向寄存器 FLASH_KEYUNLK 写入正确解锁秘钥（依次写入 0xAC7840 和 0x01234567）解锁 Flash；
2. 通过检测 FLASH_STAT 寄存器的 CCIF 位来判断是否当前有命令正在执行。如果有命令正在执行(CCIF=0)则返回错误状态，否则(CCIF=1)执行步骤 3；
3. 设置 Flash 相关配置寄存器以及命令寄存器 FLASH_CMD，然后向寄存器 FLASH_CST 为 1 触发 Flash 命令执行。后续章节不同的命令仅针对该步骤做详细描述，其它通用的步骤不在做描述；
4. 通过检查 FLASH_STAT 寄存器的 CCIF 位来判断该命令是否执行完成；
5. 当操作完成 Flash 寄存器后，可以通过依次向 FLASH_KEYUNLK 寄存器依次写入非正确秘钥（正确解锁秘钥参见步骤 1）即可锁定 Flash。

© 2013 - 2024 杰发科技有限公司

命令操作流程如下：

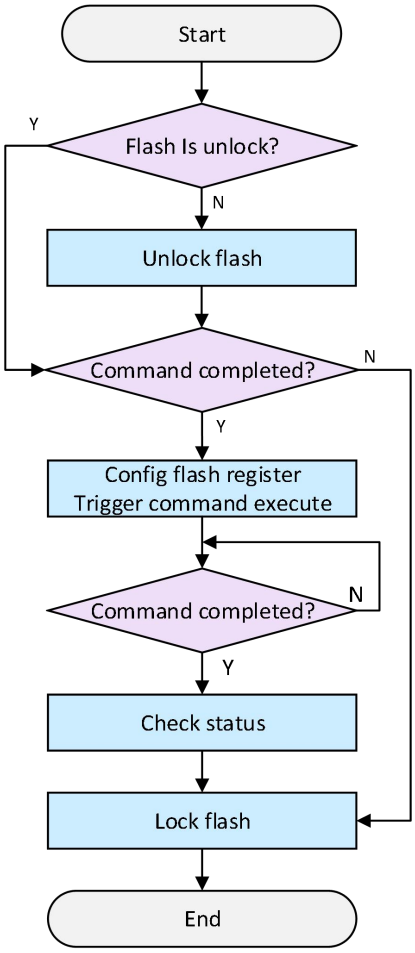


图 15- 2 操作流程

15.5.1 页擦除

页擦命令能把指定地址所在页的数据全部擦除，该命令配置步骤如下：

1. 设置页擦除地址到寄存器 FLASH_ADDR,该地址需要 8 字节对齐，否则执行页擦除命令时会产生访问错误即置位 FLASH_STAT[ACCERF];
2. 设置页擦除命令(0x01)到寄存器 FLASH_CMD；
3. 设置 1 到寄存器 FLASH_CST 触发页擦除命令的执行。

15.5.2 块擦除

块擦除可以单独擦除 P-Flash 或者 D-Flash 主存储区的全部数据。

- 当有分区时，执行 D-Flash 块擦除会失败且会产生访问错误即置位 FLASH_STAT[ACCERRF]；

© 2013 - 2024 杰发科技有限公司

- 当 P-Flash 或者 D-Flash 被设置为写保护时候执行块擦除会失败且会置位 FLASH_STAT[PVIOLF]。

该命令配置步骤如下：

1. 设置块擦除地址到寄存器 FLASH_ADDR，该地址可以是 P-Flash 或者 D-Flash 范围内任意地址；
2. 设置块擦除命令(0x02)到寄存器 FLASH_CMD；
3. 设置 1 到寄存器 FLASH_CST 触发块擦除命令的执行。

15.5.3 写保护信息页擦除

写保护信息页擦除命令仅用于擦除写保护信息所在页，擦除地址范围为 0x0020 0800 ~ 0x0020 0FFF，擦除完成后复位系统将会禁用写保护，同时会同步写保护信息到写保护寄存器，配置步骤如下：

1. 设置选项页擦除地址到寄存器 FLASH_ADDR；
2. 设置选项页擦除命令(0x03)到寄存器 FLASH_CMD；
3. 设置 1 到寄存器 FLASH_CST 触发写保护信息页擦除命令的执行。

15.5.4 全部块擦除

全部块擦除命令能擦除整个 Flash 主存储区以及信息区，如果使能 CSE（调用分区命令时分配了 CSE 空间），此时执行所有块擦除会产生访问错误即置位 FLASH_STAT[ACCERF]。当使能了读保护后且没有使能 CSE，此时通过调用该命令即可擦除读保护信息区的数据从而解除读保护，该命令成功执行的前提条件如下：

- 编程分区时没有给 CSE 分配空间，详细分区方式参考对应章节。
- 如果当前已经使能 CSE 则需要成功关闭 CSE 功能，关闭 CSE 功能方式请参考 16 硬件加密模块(CSE) 章节。

  该命令擦除范围如下：

  1. P-Flash 主存储区;
  2. P-Flash 信息区 (0x0020 0000 ~ 0x0020 0FFF) ;
  3. D-Flash 主存储区；
  4. 分区信息；

  配置步骤如下：

  1. 设置全部块擦除命令(0x04)到寄存器 FLASH_CMD；
  2. 设置 1 到寄存器 FLASH_CST 触发全部块擦除命令的执行。

© 2013 - 2024 杰发科技有限公司

15.5.5 页编程

页编程命令用于编程数据到 Flash 中，最小的编程单位为 64-bit，可编程范围如下：

- P-Flash 主存储区；
- P-Flash 信息区 (0x00200000 ~ 0x00200FFF) ；
- D-Flash 主存储区。

配置步骤如下：

1. 设置编程地址到寄存器 FLASH_ADDR，该地址需要 8 字节对齐，否则执行页编程命令时会产生访问错误即置位 FLASH_STAT[ACCERF]；
2. 设置编程数据低 32-bit 到寄存器 FLASH_DATA0；
3. 设置编程数据高 32-bit 到寄存器 FLASH_DATA1
4. 设置页编程命令(0x10)到寄存器 FLASH_CMD；
5. 设置 1 到寄存器 FLASH_CST 触发页编程命令的执行。

15.5.6 Section 编程

Section 编程能够按照 8 字节对齐方式把 FlexRAM 中的数据编程到指定 Flash(P-Flash 或者 D-Flash)地址空间，需要注意的是：该编程方式不支持跨页编程，如果编程范围超过 flash 物理页地址范围，则只会编程本页，其余未编程的数据不会继续编程，配置步骤如下：

1. 设置 FlexRAM 功能为普通 SRAM(命令 ID 为 0x32)，详情参考 15.5.14 设置 FlexRAM 功能 章节；
2. 写数据到 FlexRAM；
3. 设置编程长度到寄存器 FLASH_LEN，最小长度 1 即 64-bit，最大为 256 即编程 Flash 的一页；
4. 设置 Flash 编程目标起始地址到寄存器 FLASH_ADDR，该地址需要 8 字节对齐，否则执行 Section 编程命令时会产生访问错误即置位 FLASH_STAT[ACCERF]；
5. 设置页编程命令(0x11)到寄存器 FLASH_CMD；
6. 设置 1 到寄存器 FLASH_CST 触发 Section 编程命令的执行。

15.5.7 一次性编程

P-Flash 信息区提供了 2KB 的空间作为一次性编程区域，其地址范围为：0x0020 1000 ~ 0x0020 17FF，该区域仅仅只可以编程一次，不支持任何擦除机制，该区域编程最小位宽为 64-bit，编程地址需要 8 字节对齐，当所编程的 8 个地址空间数据非全 1 时候则编程失败且会置位访问错误标记寄存器 FLASH_STAT[ACCERRF]。一次性编程区域的读操作和读普通 Flash 区域数据相同。

可编程范围如下：

© 2013 - 2024 杰发科技有限公司

- P-Flash 信息区 (0x0020 1000 ~ 0x0020 17FF) 。

配置步骤如下：

1. 设置编程地址到寄存器 FLASH_ADDR；
2. 设置编程数据低 32-bit 到寄存器 FLASH_DATA0；
3. 设置编程数据高 32-bit 到寄存器 FLASH_DATA1；
4. 设置页编程命令(0x12)到寄存器 FLASH_CMD；
5. 设置 1 到寄存器 FLASH_CST 触发一次性编程命令的执行。

15.5.8 Section 验证

Section 验证程命令用于验证 Flash 主存储区是否被擦除，需要注意的该命令不支持跨 bank 验证，该命令可验证范围如下：

- P-Flash 主存储区 (0x0000 0000 ~ 0x000F FFFF) ；
- D-Flash 主存储区 (0x0100 0000 ~ 0x0101 FFFF) 。

配置步骤如下：

1. 设置验证地址到寄存器 FLASH_ADDR；
2. 设置验证长度到寄存器 FLASH_LEN；
    - 如果验证地址在 P-Flash 上则最多一次性验证 P-Flash 的 bank0 或者 bank1；
    - 如果验证地址在 D-Flash 上则最多一次性验证整个 D-Flash；
    - 验证长度最小值为 1 即 8 个地址。
3. 设置 section 验证命令(0x20)到寄存器 FLASH_CMD；
4. 设置 1 到寄存器 FLASH_CST 触发 section 验证命令的执行。

15.5.9 块验证

块验证程命令用于验证 Flash 主存储区中整个 P-Flash 或者整个 D-Flash 是否被擦除
配置步骤如下：

1. 设置验证地址到寄存器 FLASH_ADDR，可以指定 P-Flash/D-Flash 主存储区任意地址；
2. 设置验证命令(0x21)到寄存器 FLASH_CMD；
3. 设置 1 到寄存器 FLASH_CST 触发块验证命令的执行。

15.5.10 全部块验证

全部块验证程命令用于验证 P-Flash 以及 D-Flash 主存储区是否被擦除，该命令可验证范围如下：

© 2013 - 2024 杰发科技有限公司

- P-Flash 主存储区 (0x0000 0000 ~ 0x000F FFFF) ；
- D-Flash 主存储区 (0x010 00000 ~ 0x0101 FFFF) 。

配置步骤如下：

1. 设置验证命令(0x22)到寄存器 FLASH_CMD；
2. 设置 1 到寄存器 FLASH_CST 触发所有块验证命令的执行。

15.5.11 验证 Backdoor Key

验证 backdoor key 命令可以在 Flash 数据不被擦除的前提下临时解除芯片读保护，在内核和外设不复位前提下可以 debug，当复位后芯片又会恢复到读保护状态。所以该命令使用在读保护生效的前提下，该命令能正确执行需要具备以下条件：

1. 默认验证 backdoor key 功能是被禁用，所以首先需要通过编程命令写地址 0x0020 0010 的 bit5 以及 bit4 的值为非 11b 来使能验证 backdoor key 功能；
2. 通过页编程命令(0x10)编程地址 0x0020 0008 从而设置信息区 backdoor key 且复位后生效；
3. 触发验证命令执行时的 64-bit key 值和信息区所存在信息区(地址为 0x0020 0008)的 key 值匹配；

当 Backdoor key 使能位为非 11b 时：

- 如果用于验证的 backdoor key 和存储在信息区的 key 相等，则读保护状态会被临时解除直到复位，此过程 Flash 的数据不受任何影响；
- 如果用于验证的 backdoor key 是 64-bit 的全 1 或者全 0，则不会解除读保护状态；
- 如果用于验证的 backdoor key 和 Flash 信息区的 key 不匹配，则在下次执行验证 backdoor key 命令前必须复位，否则该命令不会再次被执行。

配置步骤如下：

1. 设置验证命令(0x23)到寄存器 FLASH_CMD；
2. 设置用户提供的 64-bit backdoor key 到数据寄存器 FLASH_DATA0 和 FLASH_DATA1 中；
3. 设置 1 到寄存器 FLASH_CST 触发验证 backdoor key 命令的执行。

15.5.12 擦除分区

擦除分区命令用来擦除分区且复位后生效，擦除分区完成后如下存储空间会发生变化：

- FlexRAM 自动被设置为普通 SRAM 功能；
- 备份区数据全部被擦除；
- 非备份区的 D-Flash 空间数据不受影响。

以下情况会导致擦除分区失败：

- 有使能 CSE 功能即分区时分配了 CSE 空间。

© 2013 - 2024 杰发科技有限公司

配置步骤如下：

1. 设置擦除分区命令(0x30)到寄存器 FLASH_CMD；
2. 设置 1 到寄存器 FLASH_CST 触发擦除分区命令的执行；
3. 复位后生效。

15.5.13 编程分区

编程分区命令用于把 D-Flash 和 FlexRAM 分区成 CSE 使用且复位后生效，编程分区命令执行前必须保证当前无任何分区，否则会产生访问错误即置位 FLASH_SATA[ACCERRF]。编程分区生效后如下存储空间会发生变化：

- FlexRAM 区域自动设置为 CSE 访问使用；
- D-Flash 的一部分或者全部空间被用于备份区；
- 未分配给备份区使用的 D-Flash 空间数据不受影响。

配置步骤如下：

1. 设置分区命令(0x31)到寄存器 FLASH_CMD；
2. 设置 CSE 分区大小、SFE 到寄存器 FLASH_DATA0 和 FLASH_DATA1 寄存器，具体设置参考表 15-7；

表 15-7 编程分区配置

寄存器	地址	位	信息
FLASH_DATA0	0x4000 201C	31：24	安全标志扩展（CSE 模块使用）
		23：0	保留
FLASH_DATA1	0x4000 2020	31：8	保留
		7：0	CSE 秘钥大小 • 0x0： CSE 秘钥大小为 0 byte • 0x1：CSE 秘钥大小为 128 bytes • 0x2：CSE 秘钥大小为 256 bytes • 0x3：CSE 秘钥大小为 512 bytes

3. 设置 1 到寄存器 FLASH_CST 触发编程分区命令的执行。

© 2013 - 2024 杰发科技有限公司

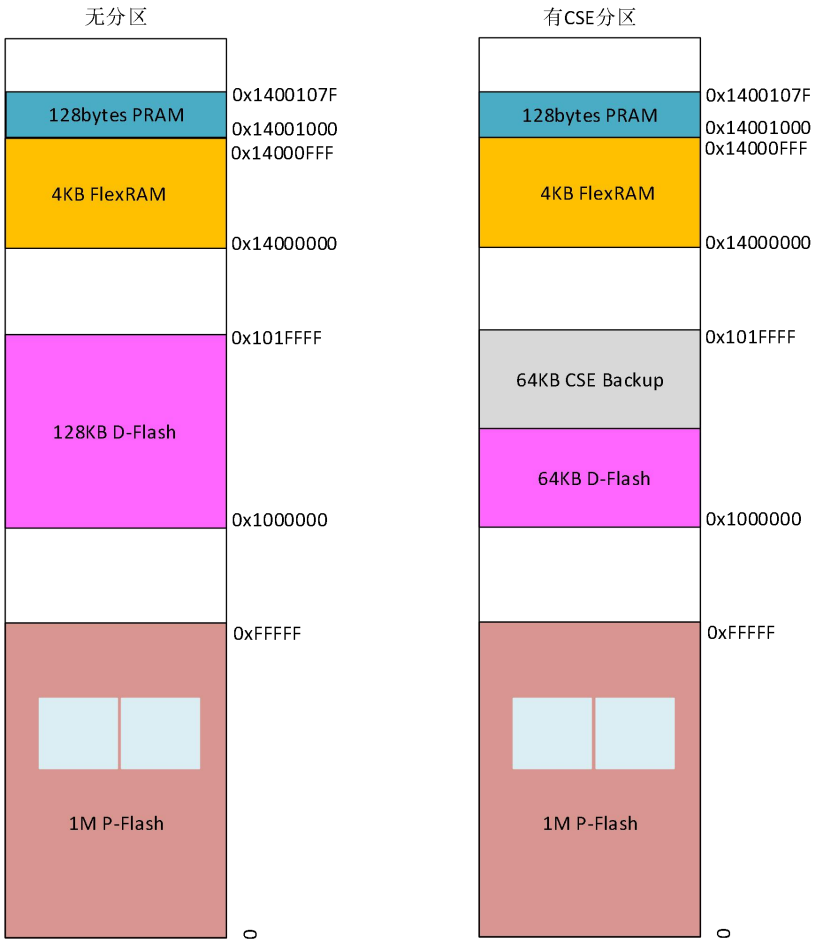


图 15-3 CSE 分区组织图

15.5.14 设置 FlexRAM 功能

FlexRAM 是一个 4KB 大小的 SRAM，其地址为 0x1400 0000 ~ 0x1400 0FFF，FlexRAM 具有如下功能：

- FlexRAM 设置为普通 SRAM 后其内容会被清除，此时 FlexRAM 功能和普通 SRAM 相同；
- FlexRAM 设置为 CSE 访问使用；

配置步骤如下：

1. 设置 FlexRAM 功能命令(0x32)到寄存器 FLASH_CMD；
2. 设置 FlexRAM 功能码到寄存器 FLASH_ADDR；
3. 设置 1 到寄存器 FLASH_CST 触发设置 FlexRAM 功能命令的执行，具体可以参考表 15-8 配置。

© 2013 - 2024 杰发科技有限公司

表 15-8 FlexRAM 功能配置

寄存器	起始地址	位	信息
FLASH_ADDR	0x4000 2018	[7：0]	FlexRAM 功能码 • 0x0：作为 CSE 功能 • 0xFF：普通 SRAM 功能
		[31：8]	保留

15.5.15 Read while write(RWW)

1. 编程或者擦除 P-Flash 其中一个 bank 或者是 D-Flash 期间支持读 P-Flash 另一个 bank 中的数据；
2. 编程和擦除 P-Flash 期间支持读 D-Flash 和 FlexRAM；
3. 当 FlexRAM 配置为普通 RAM 功能后，编程/擦除 P-Flash 或者 D-Flash 期间支持读写 FlexRAM；
4. P-Flash 和 D-Flash 不支持同时读/编程/擦除。

详情如表 15-9 所示：

表 15-9 RWW

		P-Flash bank1			D-Flash			FlexRAM	
		Read	Program	Erase	Read	Program	Erase	Read	RAM-Write
P-Flash bank0	Read	N/A	OK	OK	N/A	OK	OK	N/A	N/A
	Program	OK	N/A	N/A	OK	N/A	N/A	OK	OK
	Erase	OK	N/A	N/A	OK	N/A	N/A	OK	OK
D-Flash	Read	N/A	OK	OK	N/A	N/A	N/A	N/A	N/A
	Program	OK	N/A	N/A	N/A	N/A	N/A	OK	OK
	Erase	OK	N/A	N/A	N/A	N/A	N/A	OK	OK
Flex RAM	Read	N/A	OK	OK	N/A	OK	OK	N/A	N/A
	RAM-Write	N/A	OK	OK	N/A	OK	OK	N/A	N/A

说明：

- N/A：不支持该功能
- OK：支持此功能。

15.5.16 ECC

FLASH 支持 ECC 功能，相关功能如下：

1. ECC 1-bit 错误自动纠正且没有任何寄存器指示是否发生 ECC 1-bit 错误；
2. ECC 2-bit 错误无法纠正且会置位寄存器 FLASH_STAT[DFDIF]，该标记位写 1 清零；

© 2013 - 2024 杰发科技有限公司

3. ECC 2-bit 错误会记录 FLASH 发生该错误的物理地址。ECC 2-bit 错误的物理地址记录在寄存器 FLASH_DFADR，该寄存器写 0 清零；
4. 如果通过寄存器 FLASH_CNFG[DFDIE]使能 ECC 中断且使能对应的 NVIC 中断，则当 ECC 检测到 2-bit 错误时会产生 ECC 中断且置位寄存器 FLASH_STAT[DFDIF]，建议在中断中写 1 清除该中断标记；
5. 软件配置 SMU_LFRSTEN 对应位为 1 时，当检测到 ECC 2-bit 错误时经由 SMU 触发系统复位。

15.5.17 写保护

- 默认 P-Flash、D-Flash 写保护失能。
- 使能/使能写保护有两种方式：
  - 写保护寄存器设置写保护，详情参见表 15-10。
  - 写保护信息区设置写保护，详情参见表 15-11。

表 15-10 写保护寄存器设置写保护

写保护寄存器	FLASH_PPROT	FLASH_DPROT
寄存器地址	0x4000 2034	0x4000 2038
寄存器有效位宽	32-bit	8-bit
使能写保护方式	寄存器对应位写 0	寄存器对应位写 0
失能写保护方式	寄存器对应位写 1	寄存器对应位写 1
写保护位管控大小	32KB(1MB P-Flash)	16KB(128KB D-Flash)
保护位和区域关系	寄存器低位保护低地址区域 寄存器高位保护高地址区域	寄存器低位保护低地址区域 寄存器高位保护高地址区域
写保护生效时机	立即生效	立即生效

表 15-11 写保护信息区设置写保护

写保护信息区	P-Flash 写保护信息区	D-Flash 写保护信息区
地址	0x0020 0800	0x0020 0804
有效位宽	32-bit	8-bit
使能写保护方式	使用页编程命令对应位编程 0	使用页编程命令对应位编程 0
失能写保护方式	使用写保护信息页擦除 命令擦除该地址所在页	使用写保护信息页擦除 命令擦除该地址所在页
写保护位管控大小	32KB(1MB P-Flash)	16KB(128KB D-Flash)
保护位和区域关系	低位保护低地址区域 高位保护高地址区域	低位保护低地址区域 高位保护高地址区域
写保护生效时机	复位后生效	复位后生效

1. P-Flash 写保护由 32-bit 控制， 整个 P-Flash（地址范围参考 P-Flash 组织）被划分成 32 个区域，如果 P-Flash 空间大小为 1MB，则每个写保护位控制 32KB 大小的区域；
2. D-Flash 写保护由 8-bit 控制，整个 D-Flash（地址范围参考 D-Flash 组织）被划分成 8 个区域，

© 2013 - 2024 杰发科技有限公司

每个写保护位控制 D-Flash 空间大小除以 8 的区域，例如如果分区后 D-Flash 大小为 64KB 则每个写保护位控制 64KB/8=8KB 大小的区域；

3. 如果设置对应区域为写保护(写保护位为 0) 擦除和编程操作该 Flash 区域会失败且置位 FLASH_STAT[PVIOLF]；
4. 通过写保护寄存器设置写保护后会立即生效且不会同步到写保护信息区；
5. 通过写保护信息区设置写保护后会在复位后生效且同步写保护信息到写保护寄存器。
6. 通过写保护信息区设置写保护生效后，仍然可以通过写保护寄存器修改写保护状态，但是复位后写保护状态以写保护信息区为准直到再次更新写保护状态。

15.5.18 失能写保护

失能写保护方式有以下几种方式：

1. 程序调用全部块擦除命令(命令 ID 为 0x04)，请参考对应章节了解该命令使用流程;
2. 如果当前写保护状态是通过写保护信息区设置的，则可调用写保护信息页擦除命令失能写保护，请参考对应章节了解该命令使用流程;
3. 如果当前写保护状态是通过写保护寄存器设置的，则可通过写保护寄存器对应位为 1b 即可失能写保护。

15.5.19 读保护

通过页编程命令编程安全信息区即地址为 0x0020 0010 的 bit[1：0]为非 11b 就可以设置读保护，读保护需要复位后生效，读保护生效后无法通过 JTAG/SWD 访问整个 MCU 空间，软件可以通过读取寄存器 FLASH_SEC[RDPROT]获取读保护状态。

15.5.20 失能读保护

失能读保护方式有以下几种方式：

1. 执行全部块擦除 失能读保护，请参考对应章节了解该命令使用流程;
2. 执行验证 Backdoor Key 失能读保护，请参考对应章节了解该命令使用流程；
3. 通过编译工具失能读保护，该方式失能读保护的前提条件如下：
   - 必须通过页编程命令编程安全信息区即地址为 0x0020 0010 的 bit[3：2]为 11b（默认为 11b），否则无法通过该方式失能读保护。此种方式失能读保护效果和执行全部块擦除相同，所以失能读保护生效后擦除的区域请查看全部块擦除章节；
   - 编程分区时没有给 CSE 分配空间，详细分区方式参考对应章节；
   - 如果当前已经使能 CSE 则需要成功关闭 CSE 功能，关闭 CSE 功能方式请参考 16 硬件加密模块(CSE) 章节。

© 2013 - 2024 杰发科技有限公司

编译工具失能读保护流程如下：

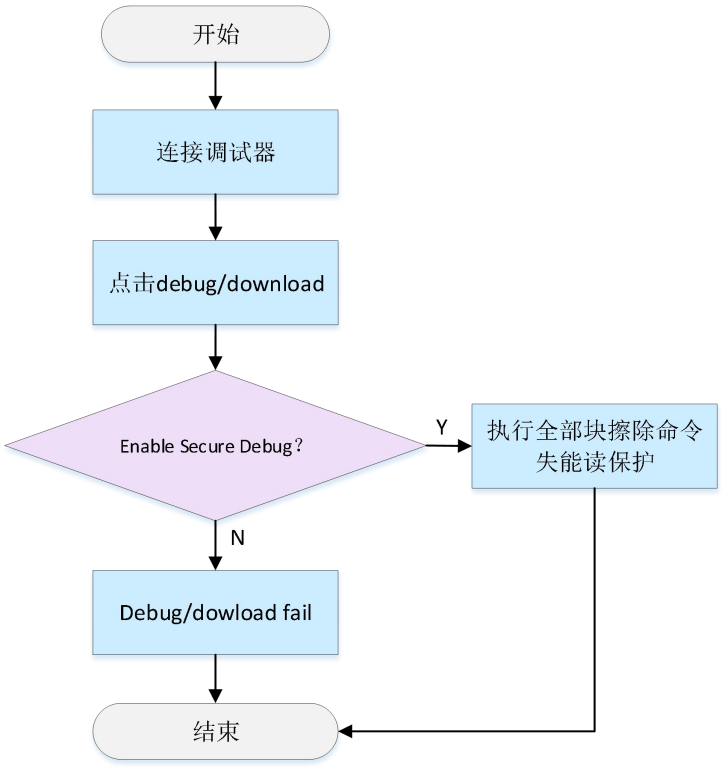


图 15-5 失能读保护流程

15.5.21 使能 MBIST

通过页编程命令编程安全信息区即地址为 0x0020 0010 的 bit[11：8]为 0000b 就可以设置 MBIST 使能，MBIST 测试需要复位后生效，每次复位后进入程序之前硬件会进行 MBIST 测试，软件可以通过读取寄存器 MCM_MBIST 获取测试结果或者通过读取寄存器 SMU_LFES 的 bit[4]获取测试结果。

15.5.22 失能 MBIST

失能 MBIST 有以下方式：

1. 执行全部块擦除失能 MBIST，请参考对应章节了解该命令使用流程;

15.5.23 使能 ABIST

通过页编程命令编程安全信息区即地址为 0x0020 0010 的 bit[15：12]为 0000b 就可以设置 ABIST 使能，ABIST 测试需要复位后生效，每次复位后进入程序之前硬件会进行 ABIST 测试，软件可以通过读取寄存器 SMU_LFES 的 bit[3]获取测试结果。

© 2013 - 2024 杰发科技有限公司

15.5.24 失能 ABIST

失能 ABIST 有以下方式：

1. 执行全部块擦除 失能 ABIST，请参考对应章节了解该命令使用流程;

15.5.25 设置 UID、Secret key 以及 Lock

UID、Secret key 以及 Lock 的地址信息如下表，其中 Lock 为 UID 和 Secret Key 的锁定位，决定了 UID 和 Secret Key 是否可以被改写。当 Lock = 64'hffff ffff ffff ffff 时，表示解锁状态，否则（不等于全 F）为上锁状态。解锁状态时，可调用页擦除命令(0x01)命令对这一页进行擦除，也可以调用设置 UID、Secret Key、Lock (0x33)命令分别对 UID、Secret Key、Lock 进行编程。

上电后硬件会自动读出 Lock。若对 Lock 的值进行了更改，则更改后的效果将在下一次上电后生效。一旦 Lock 被改写成了非全 F，则 UID、Secret Key、Lock 就不能再更改了，擦除和编程命令都无效，并且会报 access error

表 15-12 UID 和 Secret KEY 地址信息

地址	描述
0x0102_0800	SECRET KEY
0x0102_0808	
0x0102_0810	UID
0x0102_0818	
0x0102_0820	Lock

设置 UID、Secret key 以及 Lock 的流程如下：

1. 设置对应的地址到寄存器 FLASH_ADDR，该地址需要 8 字节对齐，否则执行页编程命令时会产生访问错误即置位 FLASH_STAT[ACCERF]；

2. 设置编程数据低 32-bit 到寄存器 FLASH_DATA0 ；

3. 设置编程数据高 32-bit 到寄存器 FLASH_DATA1 ；

4. 设置 UID、Secret Key、Lock 编程命令(0x33)到寄存器 FLASH_CMD；

说明：

- UID 和 Secret Key 只能通过 CSE 读，code 读该地址只会显示为 FFFFFFFFFFFFFFFF

5. 设置 1 到寄存器 FLASH_CST 触发页编程命令的执行。

15.5.26 擦除 UID 和 Secret KEY

解锁状态时，可调用页擦除命令(0x01)命令对 UID 和 Secret KEY 进行擦除，步骤如下：

© 2013 - 2024 杰发科技有限公司

1. 设置当前页的地址到寄存器 FLASH_ADDR,该地址需要 8 字节对齐，否则执行页擦除命令时会产生访问错误即置位 FLASH_STAT[ACCERF];

2. 设置页擦除命令(0x01)到寄存器 FLASH_CMD；

3. 设置 1 到寄存器 FLASH_CST 触发页擦除命令的执行。

15.6 寄存器定义

表 15-13 Flash 寄存器映射

Flash 基地址：0x40002000

地址	名称	宽度	描述
Flash 基地址+0x00	FLASH_STAT	32	Flash 状态寄存器
Flash 基地址+0x04	FLASH_SEC	32	Flash 安全寄存器
Flash 基地址+0xC	FLASH_KEYUNLK	32	Flash 解锁寄存器
Flash 基地址+0x10	FLASH_CNFG	32	Flash 配置寄存器
Flash 基地址+0x14	FLASH_CMD	32	Flash 命令寄存器
Flash 基地址+0x18	FLASH_ADDR	32	Flash 地址寄存器
Flash 基地址+0x1C	FLASH_DATA0	32	Flash 低 32 位数据寄存器
Flash 基地址+0x20	FLASH_DATA1	32	Flash 高 32 位数据寄存器
Flash 基地址+0x24	FLASH_LEN	32	Flash 长度寄存器
Flash 基地址+0x30	FLASH_CST	32	Flash 命令触发寄存器
Flash 基地址+0x34	FLASH_PPROT	32	P-Flash 写保护寄存器
Flash 基地址+0x38	FLASH_DPROT	32	D-Flash 写保护寄存器
Flash 基地址+0x40	FLASH_DFADR	32	Flash ECC 2-bit 错误地址寄存器
Flash 基地址+0x48	FLASH_PART	32	Flash 分区寄存器

15.6.1 状态寄存器（FLASH_STAT）

表 15-14 FLASH_STAT 寄存器

FLASH_STAT **状态寄存器** **复位值：0x000 0600A**

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		LOCK	CCIF	COLLERRF	ACCERRF	VERIFYERR	PVIOLF	DFDIF	FRAMEC2F	FRAMECC1F				BDKERRF	RAMRDY	EMEMRDY
访问		RO	RO	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C				RO	RO	RO
复位值		1	1	0	0	0	0	0	0	0				0	1	0

字段	说明
14 LOCK	**Flash 锁定状态寄存器** 0：Flash 未锁住，能写 Flash 寄存器 1：Flash 锁住，不能写 Flash 寄存器
13 CCIF	**FLASH 命令完成标志** 0：Flash 命令未完成 1：Flash 命令完成
12 COLLERRF	**冲突标记，当一个命令未完成时启动了另外一个命令会置位该位** 0：没有检测到冲突错误 1：检测到冲突错误 写 1 可清除该标志。
11 ACCERRF	**访问错误标记，例如错误的命令码，非对齐的 Flash 操作地址，或者不合法的地址都会置位该标记** 0：没有检测到访问错误 1：检测到访问错误 写 1 可清除该标志。
10 VERIFYERR	**验证标记，执行验证命令时所验证的区域非全 1 时会置位该位** 0：没有检测到验证错误 1：检测到验证错误 写 1 可清除该标志。
9 PVIOLF	**写保护错误标记，当编程/擦除有写保护的 Flash 区域时会置位该位** 0：没有检测到写保护错误 1：检测到写保护错误

© 2013 - 2024 杰发科技有限公司

字段	说明
	写 1 可清除该标志。
8 DFDIF	**FLASH ECC 2-bit 错误标记** 0：没有检测到 FLASH ECC 2-bit 错误 1：检测到 FLASH ECC 2-bit 错误 写 1 可清除该标志。
7 FRAMECC2F	**FlexRAM ECC 2-bit 错误标记** 0：没有检测到 FlexRAM ECC 2-bit 错误 1：检测到 FlexRAM ECC 2-bit 错误 写 1 可清除该标志。
6 FRAMECC1F	FlexRAM **ECC 1-bit 错误标记** 0：没有检测到 FlexRAM ECC 1-bit 错误 1：检测到 FlexRAM ECC 1-bit 错误 写 1 可清除该标志。
3 DFRDY	**D-Flash 空闲标记** 0：D-Flash 非空闲 1：D-Flash 空闲
2 BDKERRF	**Backdoor key 验证失败标记** 0：没有检测到 backdoor key 验证错误 1：检测到 backdoor key 验证错误 仅可被系统复位清除该标志。
1 RAMRDY	**FlexRAM 的状态标记** 0：FlexRAM 用于 CSE 1：FlexRAM 用于普通 SRAM
0 PRDY	**分区状态标记** 0：CSE 忙碌或者无 CSE 分区 1：CSE 空闲

© 2013 - 2024 杰发科技有限公司

15.6.2 安全寄存器（FLASH_SEC）

表 15-15 FLASH_SEC 寄存器

FLASH_SEC 安全寄存器 **复位值：0x000 0FF00**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	GRANTEN		BDKEYEN		EAUEN		RDPROT									
访问	RO		RO		RO		RO									
复位值	1	1	1	1	1	1	1	1								

字段	说明
15：14 GRANTEN	**Flash 工厂失效分析授权使能/失能** 11b： 使能 其它：**失能**
13：12 BDKEN	**Flash backdoor key 使能/失能** 11b：失能 其它：使能
11：10 EAUEN	**通过 JTAG/SWD 方式解除 MCU 读保护使能/失能** 11b： 使能 其它：失能
9：8 RDPROT	**读保护指示位** 11b： 读保护失能 其它：读保护使能

© 2013 - 2024 杰发科技有限公司

15.6.3 解锁寄存器（FLASH_KEYULK）

表 15-16 FLASH_KEYULK 寄存器

FLASH_KEYULK 解锁寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	KEYULK															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	KEYULK															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 KEYULK	当 STAT[LOCK]寄存器为 1 时，依次向解锁寄存器写入解锁秘钥 0xAC7840 和 0x01234567 可以解锁 Flash 寄存器，此时 STAT[LOCK]寄存器值变为 0。 当向解锁寄存器写入的数据非正确的解锁秘钥则会锁住 FLASH 寄存器。

15.6.4 配置寄存器（FLASH_CNFG）

表 15-17 FLASH_CNFG 寄存器

FLASH_CNFG 配置寄存器 复位值：0x000 03000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CLKFREQ								PERABORT		LVDDETEN	EMDFIF	CSECCIE	CCIE	COLLE	DFDIE
访问	RW								RW		RW	RW	RW	RW	RW	RW
复位值	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0

字段	说明
15：8 CLKFREQ	**时钟频率** 由于 Flash 的时钟和系统时钟同步，而 Flash 硬件的编程和擦除需要周期为 1μs 的脉冲，所以如果系统时钟频率为 100MHZ，则必须写入 0x64 到该寄存器，默认

© 2013 - 2024 杰发科技有限公司

字段	说明
	值为 0x30(48MHZ)。
7 PERABORT	**取消页擦除** 该寄存器写 1 可以取消当前正在擦除的页，取消完成后该位自动清 0 且该页 Flash 的内容未知。
5 LVDDETEN	**LVD 检测使能** 1：启用检测 LVD 0：禁用检测 LVD LVD 期间不允许执行擦除/编程操作。如果在擦除过程中发生 LVD，擦除操作将立即中止。若 LVD 发生在编程过程中，则会结束当前的编程且会忽略下一个编程请求。
4 EMDFIE	**模拟 ECC 2-bit 错误** 0：禁用模拟 ECC 2-bit 错误 1：使能模拟 ECC 2-bit 错误 允许用户模拟置位 FLASH_STAT[DFDIF]标记以检查 ECC 中断程序。
3 CSECCIE	**CSE 命令完成中断使能** 0：禁用 CSE 命令完成中断 1：使能 CSE 命令完成中断
2 CCIE	**命令完成中断使能** 0：禁用命令完成中断 1：使能命令完成中断
1 COLLE	**Flash 冲突中断使能** 0：禁用 Flash 冲突中断 1：使能 Flash 冲突中断
0 DFDIE	**ECC 2-bit 错误中断使能** 0：禁用 ECC 2-bit 错误中断 1：使能 ECC 2-bit 错误中断

© 2013 - 2024 杰发科技有限公司

15.6.5 命令 ID 寄存器（FLASH_CMD）

表 15-18 FLASH_CMD 寄存器

FLASH_CMD 命令 ID 寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CMD															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMD															
访问	R/W															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 CMD	Flash 命令 ID 寄存器，请参考 Flash 命令 ID 表，使用其它 ID 则会置位寄存器 STAT[ACCERRF]产生访问错误。

15.6.6 地址寄存器（FLASH_ADDR）

表 15-19 FLASH_ADDR 寄存器

FLASH_ADDR 地址寄存器 复位值：0x000 00000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	ADDR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 ADDR	FLASH 操作地址寄存器，地址对齐规则详见应用说明章节。

© 2013 - 2024 杰发科技有限公司

15.6.7 低 32-bit 数据寄存器（FLASH_DATA0）

表 15-20 FLASH_DATA0 寄存器

FLASH_DATA0 低 32-bit 数据寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DATA0															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA0															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 DATA0	Flash 数据寄存器低 32-bit。

15.6.8 高 32-bit 数据寄存器（FLASH_DATA1）

表 15-21 FLASH_DATA1 寄存器

FLASH_DATA1 高 32-bit 数据寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DATA1															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA1															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 DATA1	Flash 数据寄存器高 32-bit。

© 2013 - 2024 杰发科技有限公司

15.6.9 长度寄存器（FLASH_LEN）

表 15-22 FLASH_LEN 寄存器

FLASH_LEN 长度寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LEN															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	LEN															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 LEN	Section 编程命令和 section 验证命令会用到该寄存器，最小的单元为 64-bit，例如当这个寄存器值是 0x10,则表示编程/验证 16 个 64-bit 的数据。

15.6.10 命令触发寄存器（FLASH_CST）

表 15-23 FLASH_CST 寄存器

FLASH_CST 命令触发寄存器寄存器 复位值：0x0000 0000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																START
访问																RW
复位值																0

字段	说明
0 START	**触发命令执行** 该寄存器写 1 将会触发 Flash 命令的执行，如果当前 Flash 正处于忙碌状态则该命令将会被取消，当该命令执行完成，该寄存器会自动被清零。

© 2013 - 2024 杰发科技有限公司

15.6.11 P-Flash 写保护寄存器（FLASH_PPROT）

表 15-24 FLASH_PPROT 寄存器

FLASH_PPROT **P-Flash 写保护寄存器** **复位值：0xFFFF FFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PPROT															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PPROT															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
31：0 PPROT	**P-Flash 写保护状态值** 0：使能写保护 1：禁用写保护 该寄存器把 P-Flash 划分成 32 个区域，如果 P-Flash 大小是 1M 则每个区域大小为 32KB，每一个 bit 代表着一个区域的写保护状态。 MCU 复位后，该寄存器将会同步 Flash 信息区的写保护信息，通过寄存器设置的写保护立即生效。 擦除或者编程写保护的区域，将会触发写保护错误（置位 FLASH_STAT[PVIOLF]）。

15.6.12 D-Flash 寄存器（FLASH_DPROT）

表 15-25 FLASH_DPROT 寄存器

FLASH_DPROT **D-Flash 写保护寄存器** **复位值：0x0000 00FF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DPROT							
访问									RW							
复位值									1	1	1	1	1	1	1	1

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
7：0 DPROT	**D-Flash 写保护状态值** 0：使能写保护 1：禁用写保护 该寄存器把 D-Flash 划分成 8 个区域，如果 D-Flash 大小是 128KB 则每个区域大小为 16KB，每一个 bit 代表着一个区域的写保护状态。 MCU 复位后，该寄存器将会同步 Flash 信息区的写保护信息，通过寄存器设置的写保护立即生效。 如果擦除或者编程写保护的区域，将会触发写保护错误(置位 FLASH_STAT[PVIOLF])。

15.6.13 ECC 错误地址寄存器（FLASH_DFADR）

表 15- 26 FLASH_DFADR 寄存器

FLASH_DFADR **ECC 2-bit 错误地址寄存器** **复位值：0x0000 0000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DFADR															
访问	R/W0C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DFADR															
访问	R/W0C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 DFADR	**ECC 2-bit 错误地址** 当 Flash 产生 2-bit ECC 错误时候会将错误的地址记录在该寄存器。 该寄存器可以写 0 清除。

© 2013 - 2024 杰发科技有限公司

15.6.14 分区信息寄存器（FLASH_PART）

表 15-27 FLASH_PART 寄存器

FLASH_PART 分区信息寄存器 复位值：0x0000 FFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															CSESIZE	
访问															RO	
复位值															0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SFE								EPART				DPART			
访问	RO								RO				RO			
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
17：16 CSESIZE	**CSE 秘钥分配空间大小** 00b：没有分配 CSE 秘钥空间 01b：分配 128B CSE 秘钥空间 10b：分配 256B CSE 秘钥空间 11b：分配 512B CSE 秘钥空间
15：8 SFE	**安全标志扩展** 0h：秘钥的 Verify Only 属性设置功能关闭 1h：秘钥的 Verify Only 属性设置功能打开
7：4 EPART	**FlexRAM 分区 size** 1111b：FlexRAM 不用于 CSE 0010b：FlexRAM 用于 CSE
3：0 DPART	**D-Flash 分配给 CSE 保存秘钥的空间** 1000b：D-Flash 64KB 分配给 CSE 存储秘钥 1111b：D-Flash 0KB 分配给 CSE 存储秘钥 其他值：保留

© 2013 - 2024 杰发科技有限公司

16 硬件加密模块(CSE)

16.1 简介

CSE（Cryptographic Service Engine）是硬件加密模块，符合 HIS-SHE specification 1.1 rev 439 和 GM-SHE+ 安全规范标准，本文不对这些规范本身做详细介绍，若想了解更多协议本身的内容，请参考相关协议规范。

16.2 特性

- 符合 HIS-SHE specification 1.1 rev 439 和 GM-SHE+ 安全规范标准
- 支持多达 17 个用户秘钥
- 支持 AES-128 ECB(Electronic Codebook Book)算法加密和解密
- 支持 AES-128 CBC(Cipher Block Chaining)算法加密和解密
- 支持 AES-128 CMAC (Cipher-based Message Authentication Code)生成和验证
- 支持真随机数的生成
- 支持 Miyaguchi-Preneel 压缩算法 API
- 支持三种安全引导（ Secure Boot）模式：
  - 顺序安全引导模式（ Sequential Boot Mode）
  - 严格的顺序安全引导模式（ Strict Sequential Boot Mode）
  - 并行安全引导模式（ Parallel Boot Mode）

© 2013 - 2024 杰发科技有限公司

16.3 结构框图

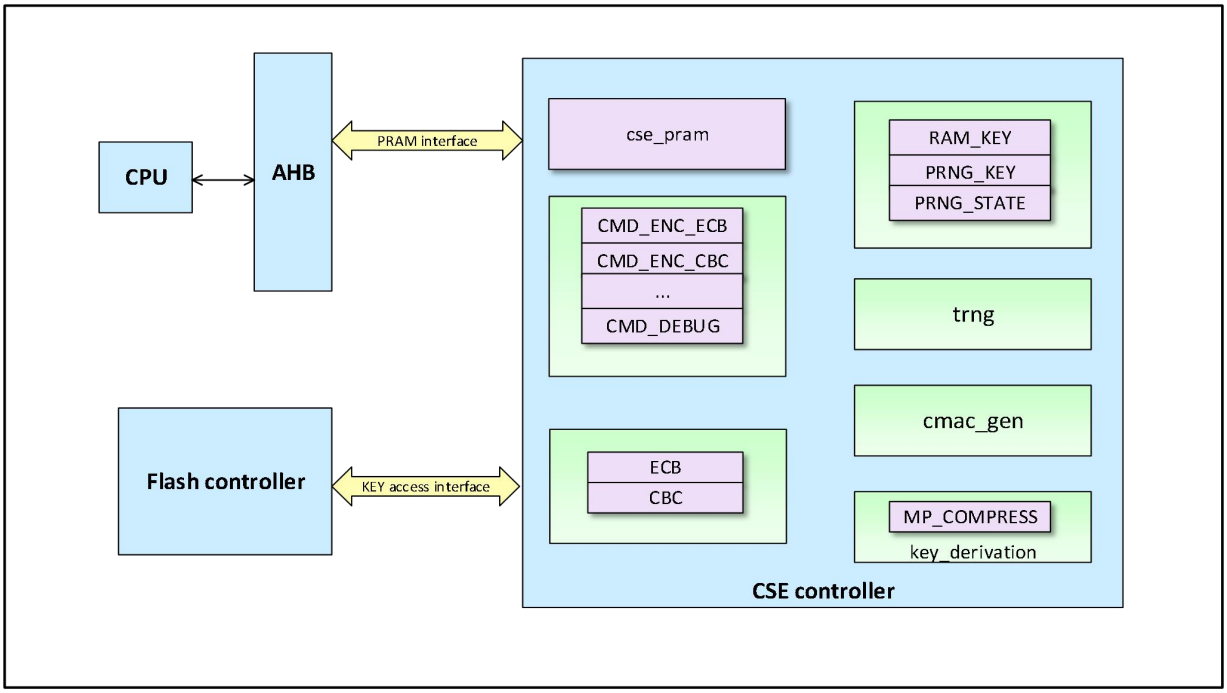


图 16-1 CSE 模块框图

16.4 功能描述

16.4.1 秘钥存储与管理

CSE 模块总共可存存储 24 个秘钥，其中有两个作为保留秘钥，下表没有列出。

表 16-1 CSE 模块秘钥

		SECRET_KEY	UID	MASTER_ECU_KEY	BOOT_MAC_KEY	BOOT_MAC	KEY_<1~10>	KEY_<11~17>	RAM_KEY
KBS		X	X	X	X	X	1'b0	1'b1	X
Sub_Key_ID		0	0	1	2	3	4~13	4~10	15
Key Size(bytes)		16	15	16	16	16	16*10	16*7	16
Memory Type		ROM (Segment)	ROM (Segment)	no-volatile					volatile
Key Counter(bits)				28	28	28	28	28	
Key Attributes (bits)	Write Protection			√	√	√	√	√	
	Boot Protection						√	√	

© 2013 - 2024 杰发科技有限公司

		SECRE	UID	MASTER_	BOOT_M	BOOT	KEY_	KEY_	RAM_
	Debugger Protection				√	√	√	√	
	Wildcard			√	√	√	√	√	
	Key Usage						√	√	
	Verify Only						√	√	
Factory Default Status		Written by user	Written by user	Empty	Empty	Empty	Empty	Empty	Undefined after reset

表 16- 2 CSE 模块秘钥描述

秘钥名称	描述
SECRET_KEY	器件在制造过程中写入的一个随机值，该值不可见且不能被更改，用于生成特殊的秘钥，比如伪随机数生成器
MASTER_ECU_KEY	可以使用 MASTER_ECU_KEY 作为授权秘钥，用于更新其他所有的用户秘钥，或者恢复 CSE 模块的秘钥至出厂状态
BOOT_MAC_KEY	用于安全引导中计算和验证 Bootloader 代码的 MAC 值
BOOT_MAC	安全引导中 Bootloader 代码的 MAC 值
KEY_<n>	17 个用户秘钥
PRNG_SEED	随机数生成器的种子，用户不能访问
RAM_KEY	存储在 RAM 中可随时更改的秘钥
PRNG_KEY	随机数生成器内部使用的秘钥，用户不能访问
KEY_ID	用于标识正在使用、更新或授权更新的秘钥

16.4.1.1 用户秘钥 ID

SHE 规范支持 10 个用户秘钥，AC7840x 扩展到最大支持 17 个用户秘钥。

用户秘钥由两部分组成：

- 第一个是**用户秘钥 ID**（即每个秘钥的固定编号，以下简称 KEY_ID）。

  KEY_ID 就像 C 语言中的指针，在程序应用中，访问 KEY_ID 代替直接访问 key 值。

  每个 KEY_ID 由两部分组成：KBS （秘钥块编号）和 Sub_Key_ID。

  因为 AC7840x CSE 模块的秘钥比 HIS-SHE 标准多 7 个用户秘钥，为了管理总共 17 个用户秘钥，使用 KBS 进行秘钥块的选择，这样 KBS 和 Sub _Key_ID 的组合 KEY_ID(KBS，Sub_Key_ID)就可以对所有的 17 个秘钥进行索引，其中高 4 位代表 KBS，低四位代表 Sub_Key_ID，比如：用户秘钥 1 的 KEY_ID 为 0x04（0x0，0x4 ），秘钥 11 的 KEY_ID 为 0x14 （0x1，0x4）。

- 第二个是 **key 值**，是真正的用户秘钥（128bit）。

© 2013 - 2024 杰发科技有限公司

16.4.1.2 CSE UID

CSE 模块中固化了一个 120-bit 长度的唯一 UID，只能由 CMD_GET_ID 命令获取，且只能读不能改写。

16.4.1.3 CSE 空间

所有秘钥存放于分区空间中，该空间只能由 CSE 模块访问，不能直接由 CPU 访问或其它模块访问。

16.4.1.4 秘钥属性

每个秘钥都有 6 个可配置属性。这些属性决定了如何使用以及在什么条件下能够使用该秘钥。

表 16-3 CSE 模块秘钥属性描述

属性	bit 位	描述
Write Protection	bit 5	设置此属性（置 1），该秘钥不允许再被改变，也不能删除秘钥和删除分区，此操作不可逆，设置后则任何方式都无法修改，设置时要小心
Boot Protection	bit 4	设置此属性（置 1），如果安全引导失败，则无法使用该秘钥
Debugger Protection	bit 3	设置此属性（置 1），如果有调试器（JTAG/SWD）连接到 MCU，则无法使用该秘钥
Key Usage	bit 2	此属性确定该秘钥是用于加/解密还是用于 CMAC 生成/验证。如果设置此属性（置 1），则秘钥用于生成和验证 CMAC 值。如果不设置此属性，则秘钥用于加密和解密
Wildcard	bit 1	设置此属性（置 1），则无法通过提供特殊的配符 UID（即 UID=0）来更新秘钥，必须获取芯片真实的 UID 来更新秘钥
Verify Only	bit 0	此属性是 GM-SHE+规范要求。如果设置此属性（置 1），则这个 Key 不能用于生成 MAC（CMD_GENERATE_MAC）命令，而只能用于验 证 MAC（CMD_VERIFY_MAC）命令；如果不设置此属性（置 0），Key 都可用于生成 MAC 和验证 MAC 该属性功能使用的前提是：在分区时配置 SFE(Security Flag Extension)=0x01

16.4.2 安全引导模式

16.4.2.1 顺序引导模式

顺序引导模块（Sequential Boot Mode）是 MCU 复位或通电后，先执行 CSE 模块安全引导对 bootloader 进行验证，经验证后按正常流程执行用户代码。如果验证失败，则用户代码无法使用具有启动保护属性（ BOOT_PROT）的秘钥。

© 2013 - 2024 杰发科技有限公司

16.4.2.2 严格引导模式

严格引导模式（Strict Sequential Boot Mode）是 MCU 复位或通电后，先执行 CSE 模块安全引导对 bootloader 进行验证，经验证后按正常流程执行用户代码。如果验证失败 CPU 就要重新 reset，不会执行用户代码。

16.4.2.3 并行引导模式

并行引导模式（Parellel Boot Mode）是 MCU 复位或通电后，安全引导和应用程序启动应并行进行，如果安全引导失败，应用程序也应启动，但将无法使用已经设置引导保护属性（ BOOT_PROT）的秘钥。

16.4.3 PRAM 接口

16.4.3.1 PRAM 接口介绍

1. PRAM 是一段连续的寄存器地址，由 8 个页组成，每页 128 位。PRAM 的起始地址为 0x14001000；
2. 应用程序通过 PRAM 接口来执行 CSE 模块相关命令。
3. PRAM 第一个页总是从命令头开始，这必须是写入 PRAM 的数据序列中写入的最后一个数据，因为写入命令头会触发锁定 PRAM 接口，导致 CSE 命令开始执行。所以，要发送 CSE 命令前，应当首先输入数据信息，然后输入数据消息长度信息，最后写入命令头 ，触发 CSE 操作。

表 16-4 PARM 格式

位	[127:0]															
位	31:24	23:17	15:8	7:0	31:24	23:16	15:8	7:0	31:24	23:17	15:8	7:0	31:24	23:17	15:8	7:0
字	Word 0				Word 1				Word 2				Word 3			
字节	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 页	FuncID	Format	CallSeq	KEYID	Error Bits		Command Specific									
备注	命令头（Command Header）						控制字段									
1 页	与 CSE 模块交互的数据（输入输出）															
2 页	与 CSE 模块交互的数据（输入输出）															
…	与 CSE 模块交互的数据（输入输出）															
6 页	与 CSE 模块交互的数据（输入输出）															
7 页	与 CSE 模块交互的数据（输入输出）															

16.4.3.2 命令头（ Command Header ）格式

CSE 模块的所有命令必循包含 6 个字节的命令头，命令头的各个字节的含义如表 16-5 所示。

表 16-5 CSE 命令头

© 2013 - 2024 杰发科技有限公司

字节					
0	1	2	3	4	5
Command Word				Error Bits	
FuncID	Func Format	CallSeq	KEY_ID		

1. FuncID：长度为 8 位的功能标识字段，根据 HIS-SHE 标注指定 CSE 模块要执行的命令，如 FunID 为 0x01 时，则意味着进行 ECB 模式的加密操作，其有效值范围为 0x01~0x16；

2. FuncFormat：FuncFormat 指定数据在 CSE 模块和应用程序之间传输方式。设置为 0x00 表示所有数据和命令通过复制的方式传入到 CSE 模块（常用方式）。设置为 0x01 则表示使用指针的方法将数据和命令传入 CSE 模块（仅适用于命令 CMD_GENERATE_MAC 和 CMD_VERIFY_MAC）；

3. CallSeq：CallSeq 用于当数据量超过 PRAM 大小（128 个字节），数据需要分多次写入到 PRAM 时，用于指定当前数据是第一次写入还是后续的写入， 取值如下：

   0x00：第一次数据写入

   0x01：后续数据写入

   后续数据写入时，只改 CallSeq，其余信息必须保持不变。

4. KEY_ID： 分为 KBS 和 Sub_Key_ID 两部分：其中 Sub_Key_ID 是指向要使用的秘钥的值（4-bit）， KBS 是秘钥所在区域的选择，位定义如下：

表 16-6 KeyID 组成

位	7	6	5	4	3	2	1	0
值	0	0	0	KBS	Sub_Key_ID			

注意：
Bit[7：5]必须始终为 0。写入 1 会导致寻址错误。如果 KBS 不用于特定秘钥，则用户必须将 0 写入 Bit[4]。

5. Error Bits：16-bit 错误信息，当检查到命令执行完成后，通过读取这 16 位的错误信息字段，检查执行过程中是否产生了错误。

16.4.4 CSE 命令

16.4.4.1 CMD_ENC_ECB

CMD_ENC_ECB 功能是实现 AES-128 ECB 加密。使用 KEY_ID 标识的秘钥对给定的明文 PLAIN_TEXT（128 位）进行加密，返回密文 CIPHER_TEXT（128 位）。其中“PAGE_LENGTH”表示有多少个 128 位数据页，所有数据必须按 128 位取整（不足 128 位的由应用程序填充）。

表 16-7 CMD_ENC_ECB 详细参数

© 2013 - 2024 杰发科技有限公司

参数	方向	位宽
KEY_ID	IN	5
PAGE_LENGTH	IN	16
PLAIN_TEXT	IN	n*128
CIPHER_TEXT	OUT	n*128
CIPHER_TEXT = $ENC_{ECB,KEY,\ KEY_ID}$ (PLAIN_TEXT)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR ,ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-8 CMD_EN_ECB 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x01	0x00	0x00	KeyID	Error Bits								PAGE_LENGTH			
1	PLAIN_TEXT 1 [0:15]															
2	PLAIN_TEXT 2 [0:15]															
3	PLAIN_TEXT 3 [0:15]															
4	PLAIN_TEXT 4 [0:15]															
5	PLAIN_TEXT 5 [0:15]															
6	PLAIN_TEXT 6 [0:15]															
7	PLAIN_TEXT 7 [0:15]															

表 16-9 CMD_ENC_ECB 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x01	0x00	0x00	KeyID	Error Bits											
1	CIPHER_TEXT 1 [0:15]															
2	CIPHER_TEXT 2 [0:15]															
3	CIPHER_TEXT 3 [0:15]															
4	CIPHER_TEXT 4 [0:15]															
5	CIPHER_TEXT 5 [0:15]															
6	CIPHER_TEXT 6 [0:15]															
7	CIPHER_TEXT 7 [0:15]															

16.4.4.2 CMD_ENC_CBC

CMD_ENC_CBC 功能是实现 AES-128 CBC 加密。使用 KEY_ID 标识的秘钥和初始向量（IV）对给定的多个明文 PLAIN_TEXT（128 位）进行加密，返回密文 CIPHER_TEXT（128 位）。其中“PAGE_LENGTH”表示有多少个 128 位数据页，所有数据必须按 128 位取整（不足 128 位的由应用程序填充）。此命令也可以通过使用 CallSeq 字段扩展为 0x01 以实现连续数据页加密。

© 2013 - 2024 杰发科技有限公司

表 16-10 CMD_ENC_CBC 详细参数

参数	方向	位宽
KEY_ID	IN	5
PAGE_LENGTH	IN	16
PLAIN_TEXT	IN	n*128
IV	IN	128
CIPHER_TEXT	OUT	n*128
CIPHER_TEXT = $ENC_{CBC,KEY,\ KEY_ID,IV}$ (PLAIN_TEXT)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-11 CMD_ENC_CBC 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x02	0x00	0x00	KeyID	Error Bits								PAGE_LENGTH			
1	IV [0:15]															
2	PLAIN_TEXT 1 [0:15]															
3	PLAIN_TEXT 2 [0:15]															
4	PLAIN_TEXT 3 [0:15]															
5	PLAIN_TEXT 4 [0:15]															
6	PLAIN_TEXT 5 [0:15]															
7	PLAIN_TEXT 6 [0:15]															

表 16-12 CMD_ENC_CBC 连续输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x02	0x00	0x01	KeyID	Error Bits											
1	PLAIN_TEXT 7 [0:15]															
2	PLAIN_TEXT 8 [0:15]															
3	PLAIN_TEXT 9 [0:15]															
4	PLAIN_TEXT 10 [0:15]															
5	PLAIN_TEXT 11 [0:15]															
6	PLAIN_TEXT 12 [0:15]															
7	PLAIN_TEXT 13 [0:15]															

© 2013 - 2024 杰发科技有限公司

表 16-13 CMD_ENC_CBC 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x02	0x00	0x00	KeyID	Error Bits											
1	IV [0:15]															
2	CIPHER_TEXT 1 [0:15]															
3	CIPHER_TEXT 2 [0:15]															
4	CIPHER_TEXT 3 [0:15]															
5	CIPHER_TEXT 4 [0:15]															
6	CIPHER_TEXT 5 [0:15]															
7	CIPHER_TEXT 6 [0:15]															

表 16-14 CMD_ENC_CBC 连续输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x02	0x00	0x01	KeyID	Error Bits											
1	CIPHER_TEXT 7 [0:15]															
2	CIPHER_TEXT 8 [0:15]															
3	CIPHER_TEXT 9 [0:15]															
4	CIPHER_TEXT 10 [0:15]															
5	CIPHER_TEXT 11 [0:15]															
6	CIPHER_TEXT 12 [0:15]															
7	CIPHER_TEXT 13 [0:15]															

16.4.4.3 CMD_DEC_ECB

CMD_DEC_ECB 功能是实现 AES-128 ECB 解密。使用 KEY_ID 标识的秘钥对给定的密文 CIPHER_TEXT（128 位）进行解密，返回明文 PLAIN_TEXT（128 位）。所有数据必须按 128 位取整（不足 128 位的由应用程序填充）。

表 16-15 CMD_DEC_ECB 详细参数

参数	方向	位宽
KEY_ID	IN	5
PAGE_LENGTH	IN	16
CIPHER_TEXT	IN	n*128
PLAIN_TEXT	OUT	n*128
PLAIN_TEXT = $DEC_{ECB,KEY,\ KEY_ID}$ (CIPHER_TEXT)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR ,ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-16 CMD_DEC_ECB 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

© 2013 - 2024 杰发科技有限公司

0	0x03	0x00	0x00	KeyID	Error Bits							PAGE_LENGTH		
1	CIPHER_TEXT 1 [0:15]													
2	CIPHER_TEXT 2 [0:15]													
3	CIPHER_TEXT 3 [0:15]													
4	CIPHER_TEXT 4 [0:15]													
5	CIPHER_TEXT 5 [0:15]													
6	CIPHER_TEXT 6 [0:15]													
7	CIPHER_TEXT 7 [0:15]													

表 16-17 CMD_DEC_ECB 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x03	0x00	0x00	KeyID	Error Bits											
1	PLAIN_TEXT 1 [0:15]															
2	PLAIN_TEXT 2 [0:15]															
3	PLAIN_TEXT 3 [0:15]															
4	PLAIN_TEXT 4 [0:15]															
5	PLAIN_TEXT 5 [0:15]															
6	PLAIN_TEXT 6 [0:15]															
7	PLAIN_TEXT 7 [0:15]															

16.4.4.4 CMD_DEC_CBC

CMD_DEC_CBC 功能是实现 AES-128 CBC 解密。使用 KEY_ID 标识的秘钥和初始向量（IV）对给定的多个密文 CIPHER_TEXT（128 位）进行解密，返回多个明文 PLAIN_TEXT（128 位）。所有数据必须按 128 位取整（不足 128 位的由应用程序填充）。此命令也可以通过使用 CallSeq 字段扩展为 0x01 以实现连续数据页解密。

表 16-18 CMD_DEC_CBC 详细参数

参数	方向	位宽
KEY_ID	IN	5
PAGE_LENGTH	IN	16
CIPHER_TEXT	IN	n*128
IV	IN	128
PLAIN_TEXT	OUT	n*128
PLAIN_TEXT = $DEC_{CBC,KEY,\ KEY_ID,IV}$ (CIPHER_TEXT)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-19 CMD_DEC_CBC 输入参数格式

© 2013 - 2024 杰发科技有限公司

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x04	0x00	0x00	KeyID	Error Bits								PAGE_LENGTH			
1	IV [0:15]															
2	CIPHER_TEXT 1 [0:15]															
3	CIPHER_TEXT 2 [0:15]															
4	CIPHER_TEXT 3 [0:15]															
5	CIPHER_TEXT 4 [0:15]															
6	CIPHER_TEXT 5 [0:15]															
7	CIPHER_TEXT 6 [0:15]															

表 16-20 CMD_DEC_ECB 连续输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x04	0x00	0x01	KeyID	Error Bits								PAGE_LENGTH			
1	CIPHER_TEXT 7 [0:15]															
2	CIPHER_TEXT 8 [0:15]															
3	CIPHER_TEXT 9 [0:15]															
4	CIPHER_TEXT 10 [0:15]															
5	CIPHER_TEXT 11 [0:15]															
6	CIPHER_TEXT 12 [0:15]															
7	CIPHER_TEXT 13 [0:15]															

表 16-21 CMD_DEC_CBC 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x04	0x00	0x00	KeyID	Error Bits											
1	IV [0:15]															
2	PLAIN_TEXT 1 [0:15]															
3	PLAIN_TEXT 2 [0:15]															
4	PLAIN_TEXT 3 [0:15]															
5	PLAIN_TEXT 4 [0:15]															
6	PLAIN_TEXT 5 [0:15]															
7	PLAIN_TEXT 6 [0:15]															

表 16-22 CMD_DEC_CBC 连续输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x04	0x00	0x01	KeyID	Error Bits											
1	CIPHER_TEXT 7 [0:15]															
2	CIPHER_TEXT 8 [0:15]															
3	CIPHER_TEXT 9 [0:15]															

© 2013 - 2024 杰发科技有限公司

4	CIPHER_TEXT 10 [0:15]
5	CIPHER_TEXT 11 [0:15]
6	CIPHER_TEXT 12 [0:15]
7	CIPHER_TEXT 13 [0:15]

16.4.4.5 CMD_GENERATE_MAC

CMD_GENERATE_MAC 功能是对指定的 MESSAGE 生成 MAC 值。使用秘钥 (KEY_ID) 对长度为 MESAGE_LENGTH 位的 MESSAGE 进行编码，其中 MESAGE_LENGTH 必须按 128 位取整（不足 128 位的由应用程序填充），最后生成 128 位 MAC 值 。

表 16-23 CMD_GENERATE_MAC 详细参数

参数	方向	位宽
KEY_ID	IN	5
MESSAGE_LENGTH	IN	32
MESSAGE (DATA <n>)	IN	n*128
MAC	OUT	128
MAC =$CMAC_{KEY, KEY_ID}$(MESSAGE, MESSAGE_LENGTH)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR ,ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-24 CMD_GENERATE_MAC 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x00	0x00	KeyID	Error Bits								MESSAGE_LENGTH			
1	DATA 1 [0:15]															
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															
5	DATA 5 [0:15]															
6	DATA 6 [0:15]															
7	DATA 7 [0:15]															

表 16-25 CMD_GENERATE_MAC 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x00	0x00	KeyID	Error Bits								MESSAGE_LENGTH			
1	DATA 1 [0:15]															
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															

© 2013 - 2024 杰发科技有限公司

5	DATA 5 [0:15]
6	DATA 6 [0:15]
7	DATA 7 [0:15]

表 16-26 CMD_GENERATE_MAC 连续输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x00	0x01	KeyID	Error Bits								MESSAGE_LENGTH			
1	DATA 8 [0:15]															
2	DATA 9 [0:15]															
3	DATA 10 [0:15]															
4	DATA 11 [0:15]															
5	DATA 12 [0:15]															
6	DATA 13 [0:15]															
7	DATA 14 [0:15]															

表 16-27 CMD_GENERATE_MAC 连续输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x00	0x01	KeyID	Error Bits								MESSAGE_LENGTH			
1	DATA 8 [0:15]															
2	MAC [0:15]															
3	DATA 10 [0:15]															
4	DATA 11 [0:15]															
5	DATA 12 [0:15]															
6	DATA 13 [0:15]															
7	DATA 14 [0:15]															

16.4.4.6 CMD_GENERATE_MAC (Pointer Method)

CMD_GENERATE_MAC 功能是对指定的 P-Flash 数据生成 MAC 值。使用“指针”方法指向 P-Flash 地址。“P-Flash 起始地址”必须是 128 位对齐。

如果在非 P-Flash 位置执行 CMD_GENERATE_MAC 指针方法，结果将是不可预测的。

表 16-28 CMD_GENERATE_MAC（Point Method）详细输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x01	0x00	KeyID	Error Bits								MESSAGE_LENGTH			
1	Flash Start Address															
2																
3																

© 2013 - 2024 杰发科技有限公司

4	
5	
6	
7	

表 16-29 CMD_GENERATE_MAC（Point Method）详细输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x05	0x01	0x00	KeyID	Error Bits								MESSAGE_LENGTH			
1	Flash Start Address															
2	MAC [0:15]															
3																
4																
5																
6																
7																

16.4.4.7 CMD_VERIFY_MAC

CMD_VERIFY_MAC 功能是验证指定 MESSAGE 的 MAC 值，该 MESSAGE 可以是多个 128 位数据块，长度由 MESSAGE_LENGTH 指定，MESSAGE_LENGTH 必须是 128 位对齐，预期 MAC 值 放在所有 MESSAGE 最后一个条目，

第一个示例显示 6 个 DATA 块后跟预期的 MAC，输出在操作结束时用 MAC 验证状态覆盖 DATA1 块。

第二个示例通过从第一个示例连续来显示 DATA1-10，但 DATA7 将被写入来代替第一个示例中的 MAC 值，然后 DATA8 从下一页继续，最后以 DATA10 结尾，在 DATA 10 条目之后跟随预期的 MAC 值。

© 2013 - 2024 杰发科技有限公司

表 16-30 CMD_VERIFY_MAC 详细参数

参数	方向	位宽
KEY_ID	IN	5
MESSAGE_LENGTH	IN	32
MESSAGE (DATA <n>)	IN	n*128
MAC	IN	128
MAC_LENGTH (MAC Length[6:0])	IN	7
VERIFICATION_STATUS (Verification Status[0])	OUT	1
MAC_{calc} = TRUNCATE($CMAC_{KEY, KEY_ID}$ (MESSAGE, MESSAGE_LENGTH), MAC_LENGTH) MAC_{ref} = TRUNCATE (MAC, MAC_LENGTH) VERIFICATION_STATUS = (0 != (MAC_{calc} - MAC_{ref}))		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR ,ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-31 CMD_VERIFY_MAC 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x00	0x00	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1	DATA 1 [0:15]															
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															
5	DATA 5 [0:15]															
6	DATA 6 [0:15]															
7	MAC [0:15]															

表 16-32 CMD_VERIFY_MAC 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x00	0x00	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1					Verification Status		DATA 1 [6:15]									
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															
5	DATA 5 [0:15]															
6	DATA 6 [0:15]															
7	MAC [0:15]															

© 2013 - 2024 杰发科技有限公司

表 16-33 CMD_VERIFY_MAC 连续输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x00	0x01	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1	DATA 8 [0:15]															
2	DATA 9 [0:15]															
3	DATA 10 [0:15]															
4	MAC [0:15]															
5																
6																
7																

表 16-34 CMD_VERIFY_MAC 连续输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x00	0x01	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1					Verification Status		DATA 8 [6:15]									
2	DATA 9 [0:15]															
3	DATA 10 [0:15]															
4	MAC [0:15]															
5																
6																
7																

16.4.4.8 CMD_VERIFY_MAC(Pointer Method)

CMD_VERIFY_MAC（Pointer Method）功能是对验证指定 MESSAGE 的 MAC 值，指定的 MSESSAGE 数据放在 P-Flash 中，使用“指针”方法指向 P-Flash 地址。“P-Flash 起始地址”必须是 128 位对齐的。如果在非 P-Flash 位置执行 CMD_VERIFY_MAC 指针方法，结果将是不可预测的。

© 2013 - 2024 杰发科技有限公司

表 16-35 CMD_VERIFY_MAC（Pinter Method）输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x01	0x00	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1	Flash Start Address															
2	MAC [0:15]															
3																
4																
5																
6																
7																

表 16-36 CMD_VERIFY_MAC（Pinter Method）输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x06	0x01	0x00	KeyID	Error Bits				MAC Length				MESSAGE_LENGTH			
1	Flash Start Address				Verification Status											
2	MAC [0:15]															
3																
4																
5																
6																
7																

16.4.4.9 CMD_LOAD_KEY

CMD_LOAD_KEY 功能是根据 SHE 规范更新内部秘钥。

注意：使用 CMD_LOAD_KEY 更新次数不能超过 340 次，若更新次数达到 340 次时，需要使用 CMD_DBG_AUTH 删除所有存储的秘钥，才能重新继续更新秘钥，或者每次 CMD_LOAD_KEY 更新后都复位 MCU。

© 2013 - 2024 杰发科技有限公司

表 16-37 CMD_LOAD_KEY 详细参数

参数	方向	位宽
KEY_ID	IN	5
M1	IN	128
M2	IN	256
M3	IN	128
M4	OUT	256
M5	OUT	128
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_NOT_AVAILABLE, ERC_KEY_INVALID,ERC_KEY_WRITE_PROTECTED, ERC_KEY_UPDATE_ERROR, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-38 CMD_LOAD_KEY 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x07	0x00	0x01	KeyID	Error Bits											
1		M1 [0:15]														
2		M2 [0:31]														
3																
4		M3 [0:15]														
5																
6																
7																

表 16-39 CMD_LOAD_KEY 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x07	0x00	0x01	KeyID	Error Bits											
1	M1 [0:15]															
2	M2 [0:31]															
3																
4	M3 [0:15]															
5	M4 [0:31]															
6																
7	M5 [0:15]															

© 2013 - 2024 杰发科技有限公司

16.4.4.10 CMD_LOAD_PLAIN_KEY

CMD_LOAD_PLAIN_KEY 功能是加载一个不加密的秘钥，并对秘钥进行验证。秘钥是以明文形式加载。普通秘钥只能装入 RAM_KEY slot（根据 SHE 规范）。该命令为 RAM_KEY 秘钥设置普通秘钥状态位。此命令没有返回值。

表 16-40 CMD_LOAD_PLAIN_KEY 详细参数

参数	方向	位宽
KEY	IN	128
KEY_{RAM_KEY}=key RAM_KEY_PLAIN = 1		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-41 CMD_LOAD_PLAIN_KEY 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x08	0x00	0x00	KeyID	Error Bits											
1	Plain Key [0:15]															
2																
3																
4																
5																
6																
7																

16.4.4.11 CMD_EXPORT_RAM_KEY

CMD_EXPORT_RAM_KEY 功能是将 RAM_KEY 导出为受 SECRET_KEY 秘钥保护的格式。
可以使用 CMD_LOAD_KEY 命令再次导入秘钥。RAM_KEY 只有在以纯文本形式写入时才能导出，即使用 CMD_LOAD_PLAIN_KEY。

表 16-42 CMD_EXPORT_RAM_KEY 详细参数

参数	方向	位宽
M1	OUT	128
M2	OUT	256
M3	OUT	128
M4	OUT	256
M5	OUT	128

© 2013 - 2024 杰发科技有限公司

K1 = KDF(KEY_{SECRET_KEY}, KEY_UPDATE_ENC_C) K2 = KDF(KEY_{SECRET_KEY}, KEY_UPDATE_MAC_C) C_{ID} = 0 (28-bits) F_{ID} = 0 (5-bits) M1 = UID | ID_{RAM_KEY} | ID_{SECRET_KEY} M2 = $ENC_{CBC,K1,IV=0}$ (C_{ID} | F_{ID} | "0...0"$_{95}$ | KEY_{RAM_KEY}) = ENC $_{CBC, K1, IV=0}$ ("0 ...0"$_{128}$ | KEY_{RAM_KEY}) M3 = $CMAC_{K2}$ (M1 | M2) K3 = KDF (KEY_{RAM_KEY} , KEY_UPDATE_ENC_C) K4 = KDF (KEY_{RAM_KEY} , KEY_UPDATE_MAC_C) M4 = UID | ID_{RAM_KEY} | ID_{SECRET_KEY} | ENC $_{EBC, K3}$ (C_{ID}) M5 = $CMAC_{K4}$(M4)
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_INVALID, ERC_KEY_EMPTY, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR

表 16-43 CMD_EXPORT_RAM_KEY 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x09	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

表 16-44 CMD_EXPORT_RAM_KEY 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x09	0x00	0x00	KeyID	Error Bits											
1	M1 [0:15]															
2	M2 [0:31]															
3																
4	M3 [0:15]															
5	M4 [0:31]															
6																
7	M5 [0:15]															

16.4.4.12 CMD_INIT_RNG

CMD_INIT_RNG 功能是初始化并派生 PRNG 的秘钥。它必须在 CMD_RND 命令之前调用，并且在每次断电或复位之后调用。此命令没有返回值。

© 2013 - 2024 杰发科技有限公司

表 16-45 CMD_INIT_RNG 详细参数

参数	方向	位宽
NONE		
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_MEMORY_FAILURE, ERC_BUSY,ERC_GENERAL_ERROR		

表 16-46 CMD_INIT_RNG 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0A	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

16.4.4.13 CMD_EXTEND_SEED

CMD_EXTEND_SEED 功能是通过将以前的种子值和提供的熵压缩为新的种子来扩展 PRNG 的种子。然后通过调用 CMD_RND 命令，使用这个新种子生成一个随机数。在扩展种子之前，必须用 CMD_INIT_RNG 初始化随机数生成器。此命令没有返回值。

表 16-47 CMD_EXTEND_SEED 详细参数

参数	方向	位宽
ENTROPY	IN	128
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_RNG_SEED, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

© 2013 - 2024 杰发科技有限公司

表 16-48 CMD_EXTEND_SEED 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0B	0x00	0x00	KeyID	Error Bits											
1	ENTROPY [0:15]															
2																
3																
4																
5																
6																
7																

16.4.4.14 CMD_RND

CMD_RND 功能是生成 128 位随机值。随机数生成器必须由 CMD_INIT_RNG 初始化，然后才能提供随机数。

表 16-49 CMD_RND 详细参数

参数	方向	位宽
RND	OUT	128

IF ($SREG_{RND_INIT}$ == 0)
$PRNG_STATE_i = ENC_{ECB,KEY\ PRNG_KEY}(PRNG_STATE_{i-1})$
$RND = PRNG_STATE_i$
END IF		

Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_RNG_SEED, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-50 CMD_RND 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0C	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

© 2013 - 2024 杰发科技有限公司

表 16-51 CMD_RND 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0C	0x00	0x00	KeyID	Error Bits											
1	RND [0:15]															
2																
3																
4																
5																
6																
7																

16.4.4.15 CMD_SECURE_BOOT

CSE 模块不支持此命令，CSE 模块通过命令 CMD_BOOT_DEFINE 设置安全启动类型，详细请参阅命令 CMD_BOOT_DEFINE。

16.4.4.16 CMD_BOOT_FAILURE

CMD_BOOT_FAILURE 功能是在自动安全启动完成后（根据 CMD_BOOT_DEFINE 配置）,检查其它信息块失败时，可以使用此命令设置为安全启动失败。

表 16-52 CMD_BOOT_FAILURE 详细参数

参数	方向	位宽
NONE		
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_NO_SECURE_BOOT, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-53 CMD_BOOT_FAILURE 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0E	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

© 2013 - 2024 杰发科技有限公司

16.4.4.17 CMD_BOOT_OK

CMD_BOOT_OK 功能是在自动安全启动完成后（根据 CMD_BOOT_DEFINE 配置）,检查其他信息块成功时，可以使用此命令设置为安全启动成功。

表 16-54 CMD_BOOT_OK 详细参数

参数	方向	位宽
NONE		
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_NO_SECURE_BOOT, ERC_BUSY,ERC_GENERAL_ERROR		

表 16-55 CMD_BOOT_OK 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x0F	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

16.4.4.18 CMD_GET_ID

GET_ID 功能是获取 UID，如果 MASTER_ECU_KEY 为空，则返回值设置为“0”。“KeyID”输入在此命令中被忽略，因为主秘钥是唯一要考虑的有效秘钥。

表 16-56 CMD_GET_ID 详细参数

参数	方向	位宽
CHALLENGE	IN	128
ID	OUT	120
SREG	OUT	8
MAC	OUT	128
ID = UID MAC = $CMAC_{KEY_MASTER_ECU_KEY}$(CHALLENGE | ID | SREG)		
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_NOT_AVAILABLE, ERC_MEMORY_FAILURE, ERC_GENERAL_ERROR		

表 16-57 CMD_GET_ID 输入参数格式

© 2013 - 2024 杰发科技有限公司

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x10	0x00	0x00	KeyID	Error Bits											
1	CHALLENGE [0:15]															
2																
3																
4																
5																
6																
7																

表 16-58 CMD_GET_ID 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x09	0x00	0x00	KeyID	Error Bits											
1	CHALLENGE [0:15]															
2	ID [0:14]															SREG
3	MAC [0:15]															
4																
5																
6																
7																

16.4.4.19 CMD_BOOT_DEFINE

CMD_BOOT_DEFINE 不是 SHE 标准的命令，用于定义受安全启动保护的启动代码大小和安全启动类似（严格、串行、并行、无）。

1. 警告：如果选择“严格启动”选项，且提供的 CMAC 与计算的 CMAC 不匹配，则 MCU 会一直处理于复位状态。由于一旦选择了“严格”启动后，启动类型就无法更改，因此在选择“严格”启动之前，要确保 CMAC 和启动过程设置正确，此外，即使设置后不会更改严格启动模模式，但仍可以更改代码内容和关联的 BOOT_MAC 和 SIZE。如果更新代码、BOOT_MAC 和 SIZE 之间发生复位或中断，将导致 BOOT_MAC 与计算的 CMAC 不匹配，MCU 会一直处理于复位状态。

2. 安全启动在 P-Flash 上运行，从地址“0”开始，大小为 Boot_SIZE。其他地址的后续代码检查可通过 CMD_VEFIFY_MAC 命令运行。整个安全启动过程结束后可能软件执行命令 CMD_BOOT_OK 或 CMD_BOOT_FAILURE 设置安全启动状态。

© 2013 - 2024 杰发科技有限公司

表 16-59 CMD_BOOT_DEFINE 详细参数

参数	方向	位宽
BOOT_SIZE (# Bytes*8, or Number of Bits : maximum size is 512 KB)	IN	32
Boot Flavor 00b : Strict (SHE Optional boot - OTP : Once set can't be changed) 01b : Serial Boot method 10b : Parallel Boot method 11b : No Boot defined (MASTER_ECU_KEY) not initialized yet, or non-CSE enabled part	IN	8
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_KEY_INVALID, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-60 CMD_BOOT_DEFINE 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x11	0x00	0x00	KeyID	Error Bits											
1												Boot Flavor	BOOT_SIZE			
2																
3																
4																
5																
6																
7																

16.4.4.20 CMD_DBG_CHAL

CSE 模块把 SHE 规范中的 CMD_DEBUG 命令被修改并分为两个命令：CMD_DBG_CHAL 和 CMD_DBG_AUTH。发出 CMD_DBG_CHAL 命令请求一个随机数，用户将使用该随机数以及 MASTER_ECU_KEY 和 UID 来使用 CMD_DBG_AUTH 命令。

表 16-61 CMD_DBG_CHAL 详细参数

参数	方向	位宽
CHALLENGE	OUT	128
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_WRITE_PROTECTED, ERC_RNG_SEED, ERC_NO_DEBUGGING, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

© 2013 - 2024 杰发科技有限公司

表 16-62 CMD_DBG_CHAL 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x12	0x00	0x00	KeyID	Error Bits											
1																
2																
3																
4																
5																
6																
7																

表 16-63 CMD_DBG_CHAL 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x12	0x00	0x00	KeyID	Error Bits											
1	CHALLENGE [0:15]															
2																
3																
4																
5																
6																
7																

16.4.4.21 CMD_DBG_AUTH

注意：

- 如果存在任何秘钥属性设置了 WRITE_PROTECTION 保护标志，则 CMD_DEBUG 不会执行擦除动作，所有涉及擦除 CSE 秘钥的命令都不能成功执行（比如 Erase partition)。
- CSE 确认验证后，闪存秘钥将被擦除，使零件恢复出厂状态。如果身份验证失败，则无法执行该命令，且不提供任何故障分析模式。
- 该命令忽略 BOOT_PROTECTION 和 DEBUGGER_PROTECTION 保护标志以及空秘钥。

CMD_DBG_AUTH 命令删除存储的所有秘钥。

© 2013 - 2024 杰发科技有限公司

表 16-64 CMD_DBG_AUTH 详细参数

参数	方向	位宽
AUTHORIZATION	IN	128
SHE & Backend: K = KDF($KEY_{MASTER_ECU_KEY}$, DEBUG_KEY_C) SHE: CHALLENGE = CMD_RND() Backend: AUTHORIZATION = $CMAC_K$(CHALLENGE | UID)		
Error Codes: ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_WRITE_PROTECTED, ERC_RNG_SEED, ERC_NO_DEBUGGING, ERC_MEMORY_FAILURE, ERC_BUSY, ERC_GENERAL_ERROR		

表 16-65 CMD_DBG_AUTH 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x13	0x00	0x00	KeyID	Error Bits											
1	AUTHORIZATION [0:15]															
2																
3																
4																
5																
6																
7																

16.4.4.22 CMD_MP_COMPRESS

MP_COMPRESS 功能是提供 Miyaguchi-Prenell 压缩接口，如果信息不足完整的 128 位块大小要求，用户/应用程序负责填充，此命令不关心 KeyID。

表 16-66 CMD_MP_COMPRESS 详细参数

参数	方向	位宽
DATA <n>	IN	n*128
PAGE_LENGTH	IN	16
MP_COMPRESS	OUT	128
Error Codes : ERC_NO_ERROR, ERC_SEQUENCE_ERROR, ERC_GENERAL_ERROR		

© 2013 - 2024 杰发科技有限公司

表 16-67 CMD_MP_COMPRESS 输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x16	0x00	0x00	KeyID	Error Bits										PAGE_LENGTH	
1	DATA 1 [0:15]															
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															
5	DATA 5 [0:15]															
6	DATA 6 [0:15]															
7	DATA 7 [0:15]															

表 16-68 CMD_MP_COMPRESS 输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x16	0x00	0x00	KeyID	Error Bits										PAGE_LENGTH	
1	MP_COMPRESS [0:15]															
2	DATA 2 [0:15]															
3	DATA 3 [0:15]															
4	DATA 4 [0:15]															
5	DATA 5 [0:15]															
6	DATA 6 [0:15]															
7	DATA 7 [0:15]															

表 16-69 CMD_MP_COMPRESS 连续输入参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x16	0x00	0x01	KeyID	Error Bits										PAGE_LENGTH	
1	DATA 8 [0:15]															
2	DATA 9 [0:15]															
3	DATA 10 [0:15]															
4	DATA 11 [0:15]															
5	DATA 12 [0:15]															
6	DATA 13 [0:15]															
7	DATA 14 [0:15]															

© 2013 - 2024 杰发科技有限公司

表 16-70 CMD_MP_COMPRESS 连续输出参数格式

page	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x16	0x00	0x01	KeyID	Error Bits										PAGE_LENGTH	
1	MP_COMPRESS [0:15]															
2	DATA 9 [0:15]															
3	DATA 10 [0:15]															
4	DATA 11 [0:15]															
5	DATA 12 [0:15]															
6	DATA 13 [0:15]															
7	DATA 14 [0:15]															

16.4.5 中断请求

通过向寄存器 FLASH_CNFG[CSECCIE]设置为 1 使能中断，通过向寄存器 CSE_STAT [CCIF]写 1 清除中断标志。

表 16-71 CSE 中断汇总

中断源	标志	本地使能	全局使能
命令完成中断	CSE_STAT[CCIF]	FLASH_CNFG[CSECCIE]	

16.5 应用说明

16.5.1 CSE 启用与关闭

16.5.1.1 启用

由于 CSE 模块所使用的秘钥都是存在分区空间中，这部分空间由 CSE 模块管理，以保证秘钥的安全，在使用之前，必须在 D-Flash 中先分区，同时分区启用 CSE 模块。通过 Flash 命令 ID 0x31 来分区和启用 CSE 模块，与 CSE 相关参数描述如下：

- **CSE Keys Size**

表 16-72 CSE Key Size 说明

	Number of Flash Keys	Number of Bytes
00b	Zero	0
01b	1 to 5 keys	128 Bytes

© 2013 - 2024 杰发科技有限公司

10b	1 to 10 Keys	256 Bytes
11b	1 to 24 Keys	512 Bytes

- **SFE(Security Flag Extension)**

此功能是 SHE 规范的扩展，对于标准 SHE，用户应将其配置为“未启用”，即 SFE == 0;

此功能是为了添加将秘钥标记为“VERIFY_ONLY”属性，也就是说，用户可以配置 SFE == 0x01，然后使用 LOAD_KEY 命令向秘钥添加一个新属性，表示该秘钥只能由 CMD_VERIFY_MAC 命令使用，不能用于 CMD_GENERATE_MAC 命令。如果 KEY_USAGE 属性为“0”，则 VERIFY_ONLY 属性无效。

16.5.1.2 关闭

CSE 功能模块关闭方法下：

1. 先通过命令 CMD_DBG_CHAL 和 CMD_DBG_AUTH 移除所有秘钥；
2. 然后通过 Flash 命令 ID 0x30 擦除 Flash 分区，如果不执行这步骤，CSE 模块还会占用空间。

如果任何一个秘钥属性设置为“WRITE_PROTECTION”，就永久不能关闭 CSE 模块。

16.5.2 秘钥管理

16.5.2.1 秘钥加载

分区使能 CSE 模块后，默认用户秘钥均为空白，必须先加载秘钥到 CSE 模块，否则无法直接使用用户秘钥进行后续操作。为了保证秘钥的安全，HIS-SHE 标准要求在尝试更新秘钥之前必须提供一个授权秘钥。可以使用这个秘钥原来的值或 MASTER_ECU_KEY 作为授权秘钥（两者使用一个即可）。MASTER_ECU_KEY 可以作为所有秘钥的授权秘钥，用于操作其他秘钥（更新 RAM_KEY 不需要授权秘钥）。比如可以使用 MASTER_ECU_KEY 作为授权秘钥来更新 BOOT_MAC_KEY，BOOT_MAC 和用户的 KEY_1 至 KEY_17。各个秘钥的授权秘钥如表 16-73 所示。

表 16-73 授权秘钥

待更新的秘钥	授权秘钥				
	MASTER_ECU_KEY	BOOT_MAC_KEY	BOOT_MAC	KEY_<N>	RAM_KEY
MASTER_ECU_KEY	√				
BOOT_MAC_KEY	√	√			
BOOT_MAC	√				
KEY_<N>	√			√	
RAM_KEY					

加载用户秘钥至 CSE 模块中时，并不是直接将秘钥明文加载至 CSE 模块中（除了 RAM_KEY 外），为了确保秘钥的机密性、完整性、真实性，并防止连续攻击，添加秘钥必须使用 HIS-SHE 规范中定义的协议（HIS-SHE 功能规范，v1.1 9.1 Description of memory update protocol）。通过对原始的

© 2013 - 2024 杰发科技有限公司

秘钥进行一系列的计算，生成 M1、M2、M3、M4、M5 共 5 组数据 ，其中 M1~M3 作为输入，加载到 CSE 模块中，加载完成后 CSE 模块会返回 M4,M5，通过返回的 M4,M5 与之前计算的 M4,M5 进行比较，如果返回的数据 与 之前计算的数据相同，则说明秘钥已经正确的加载至 CSE 模块中。由于通过原始秘钥计算出 M1~M5 值的这个过程是不可逆的，所以整个过程是安全的。

16.5.2.2 秘钥删除

通过命令 CMD_DBG_CHAL 和 CMD_DBG_AUTH 移除所有秘钥。

16.5.3 安全引导

16.5.3.1 使能安全引导模式

使用 CMD_BOOT_DEFINE 命令配置安全引导模式和 Boot 程序的大小。BOOT_MAC_KEY 是用于验证引导代码的秘钥 ，在使能安全引导模式之前必须先加载此秘钥。

16.5.3.2 加载 BOOT_MAC

加载 BOOT_MAC 有手动和自动两种方式：

- 手动加载 BOOT_MAC 步骤：
  1. 下载需要保护的 Booloader 程序到 P-Flash 中；
  2. 将 BOOT_MAC_KEY 写入到 CSE 模块中；
  3. 使用 CMD_BOOT_DEFINE 命令配置安全启动模式和 Bootloader 大小。
  4. 使用 BOOT_MAC_KEY 和 CMD_GENERATE_MAC(Pointer Method)计算要保护代码的 MAC 值;
  5. 加载计算的 MAC 值到 BOOT_MAC 中；
  6. 复位设备；
  7. CSE 模块会验证定义的 Bootloader 程序的 MAC 值和存储的 BOOT_MAC 值是否匹配，如果匹配，设置 CSESTAT[BOK]=1，表示安全启动验证成功，如果不相同，则设置为 CSESTAT[BOK]=0，表示安全启动验证失败 。
- 自动加载 BOOT_MAC 步骤：
  1. 下载需要保护的 Booloader 程序到 P-Flash 中；
  2. 将 BOOT_MAC_KEY 写入到 CSE 模块中；
  3. 使用 CMD_BOOT_DEFINE 命令配置安全启动模式和 Bootloader 大小。

© 2013 - 2024 杰发科技有限公司

4. 复位设备，CSE 模块自动计算 MAC 并加载在 BOOT_MAC 中；

5. 复位设置，CSE 模块会验证定义的 Bootloader 程序的 MAC 值和存储的 BOOT_MAC 值是否匹配，如果匹配，设置 CSESTAT[BOK]=1，表示安全启动验证成功，如果不相同，则设置为 CSESTAT[BOK]=0，表示安全启动验证失败 。

16.5.4 数据通信安全保护

使用 CSE 模块对 Central ECU 与 Sensor ECU 之前通信数据进行加密，随机数可以防止连续的攻击，通信加密能够防止通信被窃听，整个机制结合确保了数据的完整性和真实性。

1 Central ECU 产生一个随机数 RND，通过接口（比如 CAN）发给 Sensor ECU；

2 Sensor ECU 收到随机数 RND，使用 Key #对 Sensor 数据进行加密；

3 Sensor ECU 把加密后的数据和收到的随机数据 RND 传给 Central ECU；

4 Central ECU 收到随机数 RND，先对比上次发给 Sensor ECU 的 RND，如果不一致就报错，如果一致，就使用 Key #对收到加密数据进行解密。

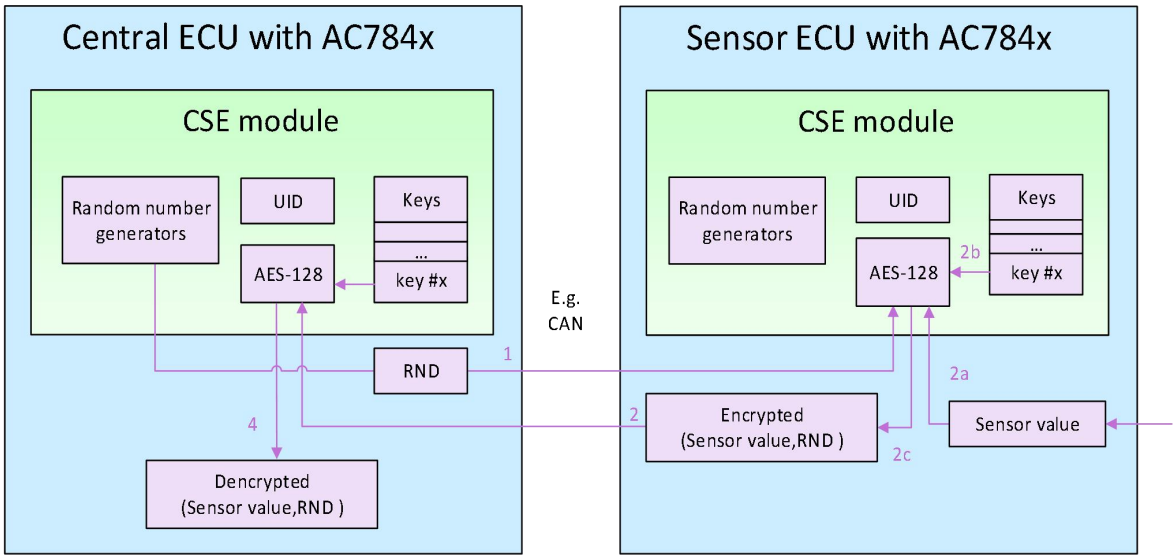


图 16- 2 数据通信安全保护

16.5.5 Flash 数据完整性保护

通过 AES-128 CMAC 生成和验证功能，对 Flash 数据进行完整性校验和保护，比如以下对代码进行完整性保护。

1 安全启动机制验证 Bootloader 的完整性；

2 Bootloader 启动之后，使用 CSE 模块的命令 CMD_VERIFY_MAC(Pointer Method)可对 part #1 代码进行验证；

© 2013 - 2024 杰发科技有限公司

3 验证 part #1 通过，则运行代码 part #1；

4 在代码 part #1 对 part #2 验证；

5 类似方法逐步验证到 part #n+1。

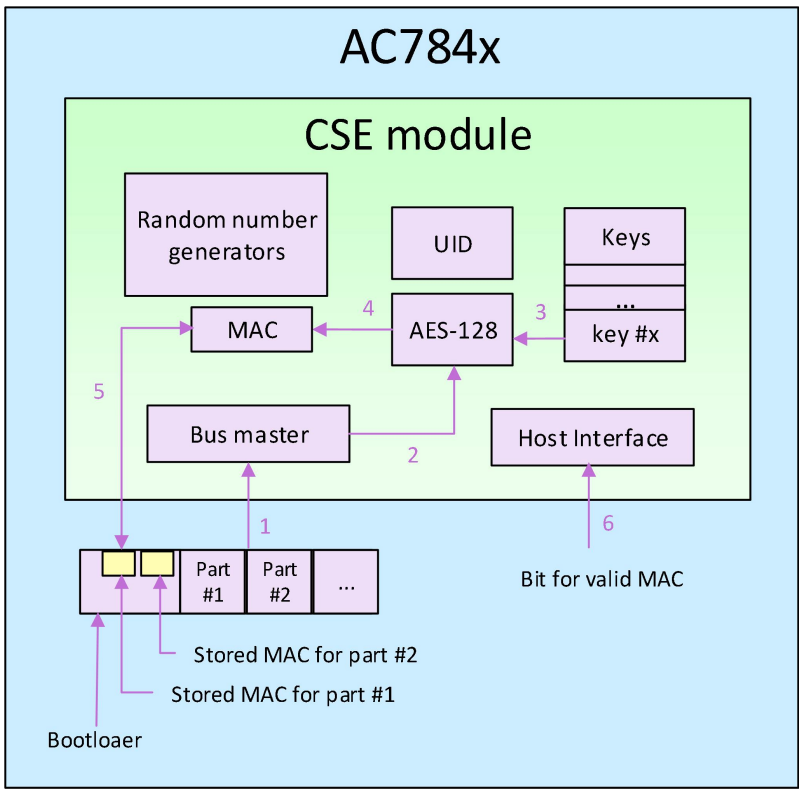


图 16-3 Flash 数据完整性保护

© 2013 - 2024 杰发科技有限公司

16.6 寄存器定义

16.6.1 CSE 状态寄存器（CSESTAT）

CSE 状态寄存器地址：0x4000 2008

表 16- 74 CSE 状态寄存器

CSESTAT **CSE 状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								CCIF	IDB	EDB	RIN	BOK	BFN	BIN	SB	BSY
访问								R/W 1c	RO							
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8 CCIF	**CSE 命令完成中断标志** 0：CSE 命令未完成 1：CSE 命令完成 **说明：写 1 清零**
7 IDB	**CSE 内部 debug** 0：CSE 内部 debug 关闭 1：CSE 内部 debug 打开
6 EDB	**CSE 外部 debug** 0：外部 debug 未连接 1：外部 debug 连接
5 RIN	**CSE 随机数初始化** 0：随机数未初始化 1：随机数已经初始化
4 BOK	**CSE 安全引导正常** 0：安全引导失败 1：安全引导正常
3 BFN	**CSE 安全引导完成**

© 2013 - 2024 杰发科技有限公司

字段	说明
	CSEBFN 由命令 BOOT_FAILUER 或 BOOT_OK 设置，或在安全引导过程中 BOOT_MAC 校验失败。 0：安全引导未完成 1：安全引导完成
2 BIN	**CSE 安全引导初始化** 0：安全引导未初始化 1：安全引导初始化完成
1 SB	**CSE 安全引导激活** 0：安全引导未激活 1：安全引导激活
0 BSY	**CSE 模块忙状态** 0：CSE 命令执行完成 1：CSE 命令执行过程中

表 16- 75 CSESTAT 中标志位规则

场景	SB	BIN	BFN	BOK	Error Code	描述
未使用安全引导	0	0	0	0	0	未使用安全引导功能
BOOT_MAC 为空	1	1	1	0	未返回错误	加载 BOOT_MAC 到 CSE 模化未成功
BOOT_MAC 不匹配	1	0	1	0	未返回错误	安全引导校验 BOOT_MAC 失败
BOOT_MAC 验证成功	1	0	0	1	未返回错误	安全引导校验 BOOT_MAC 成功
BOOT_MAC_KEY 为空	0	0	1	0	NO_SECURE_BOOT	安全引导退出，并返回 错误状态

16.6.2 PRAM 寄存器

表 16- 76 PRAM 寄存器映射

PRAM 基地址：0x1400 1000

地址	名称	宽度	描述
PRAM 基地址 + 0x00	CMD_PG0_WORD0	32	寄存器 0 页 0 字
PRAM 基地址 + 0x04	CMD_PG0_WORD1	32	寄存器 0 页 1 字
PRAM 基地址 + 0x08	CMD_PG0_WORD2	32	寄存器 0 页 2 字
PRAM 基地址 + 0x0C	CMD_PG0_WORD3	32	寄存器 0 页 3 字
PRAM 基地址 + m*0x10 + n*0x04	CMD_PGm_WORDn	32	寄存器 m 页 n 字

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
（其中“m”从 1 到 7，”n”从 0 到 3）			

16.6.2.1 寄存器 0 页 0 字（CMD_PG0_WORD0）

表 16- 77 CMD_PG0_WORD0 寄存器

CMD_PG0_WORD0 寄存器 0 页 0 字 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	KEY_ID								CALL_SEQ							
访问	RW								RW							
复位值	0								0							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	FUNC_FORMAT								FUNC_ID							
访问	RW								RW							
复位值	0								0							

字段	说明
31:24 KEY_ID	**秘钥 ID 号** KEY_ID = {KBS, Sub_Key_ID } Key name KBS Sub_Key_ID SECRET_KEY x 0h MASTER_ECU_KEY x 1h BOOT_MAC_KEY x 2h BOOT_MAC x 3h KEY_01-KEY_10 0 4h-Dh KEY_11-KEY_17 1 4h-Ah RESERVED 1 Bh-Eh RESERVED 0 Eh RAM_KEY x Fh
23:16 CALL_SEQ	**调用序列** 0 : 第 1 次调用 1 : 第 2 到 n 次调用
15:08 FUNC_FORMAT	**函数格式** 0h：复制调用方法 1h：指针调用方法 指定如何将数据传输到 CSE 或从 CSE 传输数据。有两个用例。一个（最常见的）是复制所有数据和命令函数调用方法，另一个是指针和函数调用方法。
7:00 FUNC_ID	**函数 ID 号** 1h：ENC_ECB

© 2013 - 2024 杰发科技有限公司

字段	说明
	2h：ENC_CBC 3h：DEC_ECB 4h：DEC_CBC 5h：GENERATE_MAC 6h：VERIFY_MAC 7h：LOAD_KEY 8h：LOAD_PLAIN_KEY 9h：EXPORT_RAM_KEY Ah：INIT_RND Bh：EXTEND_SEED Ch：RND Dh：Reserved Eh：BOOT_FAILURE Fh： BOOT_OK 10h：GET_ID 11h：BOOT_DEFINE 12h：DBG_CHAL 13h：DBG_AUTH 14h：CANCEL 15h： MP_COMPRESS 16h – FFh：Reserved

16.6.2.2 寄存器 0 页 1 字（CMD_PG0_WORD1）

表 16-78 CMD_PG0_WORD1 寄存器

CMD_PG0_WORD1	寄存器 0 页 1 字	复位值：0x00000000

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				ERC_GENERAL_ERROR	ERC_MEMORY_FAILURE	ERC_BUSY	ERC_NO_DEBUGGING	ERC_RNG_SEED	ERC_KEY_UPDATE_ERROR	ERC_KEY_WRITE_PROTECTED	ERC_NO_SECURE_BOOT	ERC_KEY_EMPTY	ERC_KEY_INVALID	ERC_KET_NOT_AVALIABLE	ERC_SEQ_ERROR	ERC_NO_ERROR
访问				RO												
复位值				0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
12 ERC_GENERAL_ERROR	**ERC_GENERAL_ERROR** 如果在 CSE 内检测到上述错误代码未涵盖的错误，则返回此错误代码
11 ERC_MEMORY_FAILURE	**ERC_MEMORY_FAILURE** 如果保存 CSE 数据区域检测到物理错误，则可以返回此错误代码
10 ERC_BUSY	**ERC_BUSY** NA（不支持此错误状态）
9 ERC_NO_DEBUGGING	**ERC_NO_DEBUGGING** 如果协议未成功，无法进行内部调试，则返回错误代码
8 ERC_RNG_SEED	**ERC_RNG_SEED** 如果之前未初始化种子，调用 CMD_RND 和 CMD_DEBUG 将返回错误代码
7 ERC_KEY_UPDATE_ERROR	**ERC_KEY_UPDATE_ERROR** 当验证消息时出错而导致秘钥更新未成功时，将返回此错误
6 ERC_KEY_WRITE_PROTECTED	**ERC_KEY_WRITE_PROTECTED** 当对已经写保护秘钥进行更新时，将返回此错误
5 ERC_NO_SECURE_BOOT	**ERC_NO_SECURE_BOOT** NA（不支持此错误状态）
4 ERC_KEY_EMPTY	**ERC_KEY_EMPTY**

© 2013 - 2024 杰发科技有限公司

字段	说明
	当使用尚未初始化的秘钥时，将返回此错误代码
3 ERC_KEY_INVALID	**ERC_KEY_INVALID** 当使用秘钥执行不允许的操作时，将返回此错误代码
2 ERC_KET_NOT_AVALIABLE	**ERC_KET_NOT_AVALIABLE** 当使用由于启动失败而锁定的秘钥时，则返回此错误代码
1 ERC_SEQ_ERROR	**ERC_SEQ_ERROR** 当命令或子命令的序列超出范围时，C 将返回此错误代码 顺序，例如，当调用一个函数时，另一个函数仍在运
0 ERC_NO_ERROR	**ERC_NO_ERROR** 未发生错误

16.6.2.3 寄存器 0 页 2 字（CMD_PG0_WORD2）

表 16-79 CMD_PG0_WORD2 寄存器

CMD_PG0_WORD2 **寄存器 0 页 2 字** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CMD_SPECIFIC															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMD_SPECIFIC															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 CMD_SPECIFIC	**CSE 命令指定数据** 参考 16.4.4 CSE 命令 描述

16.6.2.4 寄存器 0 页 3 字（CMD_PG0_WORD3）

表 16-80 CMD_PG0_WORD3 寄存器

CMD_PG0_WORD3 **寄存器 0 页 3 字** **复位值：0x00000000**

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CMD_SPECIFIC															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMD_SPECIFIC															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 CMD_SPECIFIC	**CSE 命令指定数据** 参考 16.4.4 CSE 命令 描述

16.6.2.5 寄存器 m 页 n 字（CMD_PGm_WORDn）

表 16-81 CMD_PGm_WORDn 寄存器

CMD_PGm_WORDn **寄存器 m 页 n 字** **复位值班：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CMD_DATA															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMD_DATA															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 CMD_DATA	**CSE 命令指定数据** 参考 16.4.4 CSE 命令 描述

© 2013 - 2024 杰发科技有限公司

17 静态存储器（SRAM）

17.1 简介

SRAM 分为 64Kbyte SRAM_L、60Kbyte SRAM_U 区域、和 4KB FlexRAM，总计 128Kbyte。

地址分区如下：

- SRAM_L: 0x1FFF_0000~0x1FFF_FFFF；
- SRAM_U: 0x2000_0000 ~0x2000_EFFF；
- FlexRAM: 0x1400_0000 ~0x1400_0FFF

SRAM 支持 ECC（Error Correcting Code），用于 SRAM 的差错检测和修正。SRAM 出错的时候一般不会造成整个 SRAM 不能访问或是全部出错，而是整个 SRAM 中只有一个或几个 bit 出错。



说明：
FlexRAM 一般用于 CSE 模块，也可以用作普通 SRAM。

ECC_SRAM 采用汉明码 ECC 检测算法，可以纠正 1-bit 错误，检测 2-bit 及以下的错误。

17.2 特性

SRAM 支持字节（8-bit）、半字（16-bit）、字（32-bit）访问。
ECC 支持：

- 1-bit 和 2-bit 检错
  - 可以检出 1-bit 和 2-bit 错误状态
  - 软件可以通过寄存器读取出错的地址以及错误的状态
- 1-bit 错误状态纠错
  - 当检测出 1-bit 错误状态时，硬件自动纠错
  - 2-bit 错误状态硬件不可以纠错
- 1-bit 和 2-bit 错误状态中断
  - 软件可以配置当 ECC_SRAM 检测到 1-bit 和 2-bit 错误时是否产生中断
  - 软件可以配置在 ECC_SRAM 检测到 2-bit 错误时是否触发系统复位
  - 软件可以配置当 ECC_SRAM 检测到 1-bit 和 2-bit 错误时是否经由 SMU 触发系统复位
- 2-bit 以上错误状态不能保证检测

© 2013 - 2024 杰发科技有限公司

17.3 功能描述

下述描述中 x=0/1，分别对应 SRAM_L 和 SRAM_U 区域的 ECC；下面描述中 CH0 和 CH1 分别对应 SRAM_L 和 SRAM_U；

y=1/2，分别代表 1-bit 和 2-bit 错误。

17.3.1 ECC 错误状态

软件可以查询 CH0_STATUS0[ERR_ST]和 CH1_STATUS0[ERR_ST]的值来判断当前 SRAM 的错误状态。

- 如果[ERR_ST]=000b，表明 SRAM ECC 没有检测到错误；
- 如果[ERR_ST]=x01b，表明 SRAM ECC 检测到 2-bit 错误；
- 如果[ERR_ST]= 010b/011b/11xb/100b，表明 SRAM ECC 检测到 1-bit 错误。

在系统不复位情况下，[ERR_ST]会保留最近一次检测到的错误状态，通过向 ECC_ERR_CTRL[ECCx_ST_CLR2]写入 1 可以清除错误状态。

17.3.2 ECC 错误地址

- 软件可以查询 CH0_STATUS0[ERR_ADDR0]和 CH1_STATUS0[ERR_ADDR0] 的值，乘以 4 后再加上对应 SRAM 区基地址，得到 SRAM 1-bit 错误地址；
- 查询 CH0_STATUS1 [ERR_ADDR1]和 CH1_STATUS1[ERR_ADDR1] 的值，乘以 4 后再加上对应 SRAM 区基地址，得到 SRAM 2-bit 错误地址。

在系统不复位情况下，[ERR_ADDR0]和[ERR_ADDR1]会保留最近一次检测到的错误地址，软件通过向 ECC_ERR_CTRL[ECCx_ST_CLR2]写入 1 可以清除错误地址。

17.3.3 ECC 错误处理

当使能 ECC 2-bit 错误复位功能（即 RCM_EN [ECC2_ERR_RST_EN]=1），2-bit 错误将会产生系统复位。

当使能 ECC 1-bit 或 2-bit SMU 复位功能（即配置 SMU_LFRSTEN 对应位为 1），1-bit 或 2-bit 错误都可由 SMU 触发系统复位。

17.3.4 中断请求

当配置 ECC_ERR_CTRL [ECCx_yBIT_IRQ_EN]=1，且对应 SRAM 区检测到 ECC y-bit 错误时，相应的中断标志将被置位，即寄存器 CH0_STATUS1[ERR_STS$_{(y-1)}$]和 CH1_STATUS1[ERR_STS$_{(y-1)}$]=1。

通过向 ECC_ERR_CTRL[ECCx_ST_CLR1]写 1 清除对应的中断标志。

© 2013 - 2024 杰发科技有限公司

表 17-1 SRAM 中断汇总

中断源	标志	使能	全局使能
SRAM_L 1-bit 错误	CH0_STATUS1 [ERR_STS0]	ECC0_1BIT_IRQ_EN	
SRAM_L 2-bit 错误	CH0_STATUS1 [ERR_STS1]	ECC0_2BIT_IRQ_EN	
SRAM_U 1-bit 错误	CH1_STATUS1 [ERR_STS0]	ECC1_1BIT_IRQ_EN	
SRAM_U 2-bit 错误	CH1_STATUS1[ERR_STS1]	ECC1_2BIT_IRQ_EN	

17.3.5 调试模式

SRAM 模块没有调试模式配置，在进入调试模式时可正常运行。

17.4 寄存器定义

SRAM 寄存器只能在特权模式下访问，用户模式下访问触发 hardfault。
SRAM 寄存器只支持 32 位访问。

表 17-2 SRAM 寄存器映射

ECC_SRAM 基地址：0x40088100

地址	名称	宽度	描述
ECC_SRAM 基地址+0x0	CH0_STATUS0	32	SRAM_L ECC 错误状态和 1-bit 错误地址寄存器
ECC_SRAM 基地址+0x4	CH0_STATUS1	32	SRAM_L ECC 中断标志和 2-bit 错误地址寄存器
ECC_SRAM 基地址+0x8	CH1_STATUS0	32	SRAM_U ECC 错误状态和 1-bit 错误地址寄存器
ECC_SRAM 基地址+0xC	CH1_STATUS1	32	SRAM_U ECC 中断标志和 2-bit 错误地址寄存器
ECC_SRAM 基地址+0x10	ECC_ERR_CTRL	32	ECC 中断使能和状态清除寄存器

17.4.1 SRAM_L ECC 错误状态和 1-bit 错误地址寄存器（CH0_STATUS0）

表 17-3 CH0_STATUS0 寄存器

CH0_STATUS0　　CH0 错误状态和 1-bit 错误地址寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														ERR_ST		
访问														RO		
复位值														0		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			ERR_ADDR0[13:0]													
访问			RO													
复位值			0													

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
18:16 ERR_ST	**SRAM_L ECC 错误状态** 000b: 没有错误 x01b: 2-bit 错误 010b/011b/11xb/100b: 1-bit 错误
13:0 ERR_ADDR0	**SRAM_L 1-bit 错误地址** **说明： (ERR_ADDR0*4 + 0x1FFF0000) 是发生 1-bit 错误的 SRAM_L 地址。**

17.4.2 SRAM_L ECC 中断标志和 2-bit 错误地址寄存器（CH0_STATUS1）

表 17-4 CH0_STATUS1 寄存器

CH0_STATUS1　　CH0 中断标志和 2-bit 错误地址寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															ERR_ST1	ERR_ST0
访问															RO	RO
复位值															0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			ERR_ADDR1[13:0]													
访问			RO													
复位值			0													

字段	说明
17 ERR_STS1	**SRAM_L 2-bit 错误中断标志位** 0: 没有错误 1: 2-bit 错误
16 ERR_STS0	**SRAM_L 1-bit 错误中断标志位** 0: 没有错误 1: 1-bit 错误
13:0 ERR_ADDR1	**SRAM_L 2-bit 错误地址** **说明： (ERR_ADDR1*4 + 0x1FFF0000) 是发生 2-bit 错误的 SRAM_L 地址。**

17.4.3 SRAM_U ECC 错误状态和 1-bit 错误地址寄存器（CH1_STATUS0）

表 17-5 CH1_STATUS0 寄存器

© 2013 - 2024 杰发科技有限公司

CH1_STATUS0　　CH1 错误状态和 1-bit 错误地址寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														ERR_ST		
访问														RO		
复位值														0		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			ERR_ADDR0[13:0]													
访问			RO													
复位值			0													

字段	说明
18:16 ERR_ST	**SRAM_U ECC 错误状态** 000b: 没有错误 x01b: 2-bit 错误 010b/011b/11xb/100b: 1-bit 错误
13:0 ERR_ADDR0	**SRAM_U 1-bit 错误地址** **说明： (ERR_ADDR0*4 + 0x20000000) 是发生 1-bit 错误的 SRAM_U 地址。**

17.4.4　SRAM_U ECC 中断状态和 2-bit 错误地址寄存器（CH1_STATUS1）

表 17- 6 CH1_STATUS1 寄存器

CH1_STATUS1　　ECC1 中断标志和 2-bit 错误地址寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															ERR_ST1	ERR_ST0
访问															RO	RO
复位值															0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			ERR_ADDR1[13:0]													
访问			RO													
复位值			0													

字段	说明
17 ERR_STS1	**SRAM_U 2-bit 错误中断标志位** 0: 没有错误 1: 2-bit 错误
16 ERR_STS0	**SRAM_U 1-bit 错误中断标志位** 0: 没有错误

© 2013 - 2024 杰发科技有限公司

字段	说明
	1: 1-bit 错误
13:0 ERR_ADDR1	**SRAM_U 2-bit 错误地址** **说明： (ERR_ADDR1*4 + 0x20000000) 是发生 2-bit 错误的 SRAM_U 地址。**

17.4.5 SRAM ECC 中断使能和状态清除寄存器（ECC_ERR_CTRL）

表 17- 7 ECC_ERR_CTRL 寄存器

ECC_ERR_CTRL　　ECC 中断使能和状态清除寄存器　　复位值：0x00000303

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					ECC1_ST_CLR2	ECC1_ST_CLR1	ECC1_2BIT_IRQ_EN	ECC1_1BIT_IRQ_EN					ECC0_ST_CLR2	ECC0_ST_CLR1	ECC0_2BIT_IRQ_EN	ECC0_1BIT_IRQ_EN
访问					WO	WO	RW	RW					WO	WO	RW	RW
复位值					0	0	1	1					0	0	1	1

字段	说明
11 ECC1_ST_CLR2	**SRAM_U ECC 错误状态和地址清除** **说明：写 1 清除**
10 ECC1_ST_CLR1	**SRAM_U ECC 错误中断标志清除** **说明：写 1 清除**
9 ECC1_2BIT_IRQ_EN	**SRAM_U ECC 2-bit 错误中断使能** 0: 禁止 1: 使能
8 ECC1_1BIT_IRQ_EN	**SRAM_U ECC 1-bit 错误中断使能** 0: 禁止 1: 使能
3 ECC0_ST_CLR2	**SRAM_L ECC 错误状态和地址清除** **说明：写 1 清除**
2	**SRAM_L ECC 错误中断标志清除**

© 2013 - 2024 杰发科技有限公司

字段	说明
ECC0_ST_CLR1	
	说明：写 1 清除
1 ECC0_2BIT_IRQ_EN	**SRAM_L ECC 2-bit 错误中断使能** 0: 禁止 1: 使能
0 ECC0_1BIT_IRQ_EN	**SRAM_L ECC 1-bit 错误中断使能** 0: 禁止 1: 使能

© 2013 - 2024 杰发科技有限公司

18 错误注入模块(EIM)

18.1 简介

EIM（error injection module）支持对 SRAM 的数据、地址、ECC 校验码进行选择性错误注入，可以用于验证 ECC 机制。

18.2 特性

- 对 SRAM 的数据、地址、ECC 校验码进行选择性错误注入
- 双重使能机制

18.3 结构框图

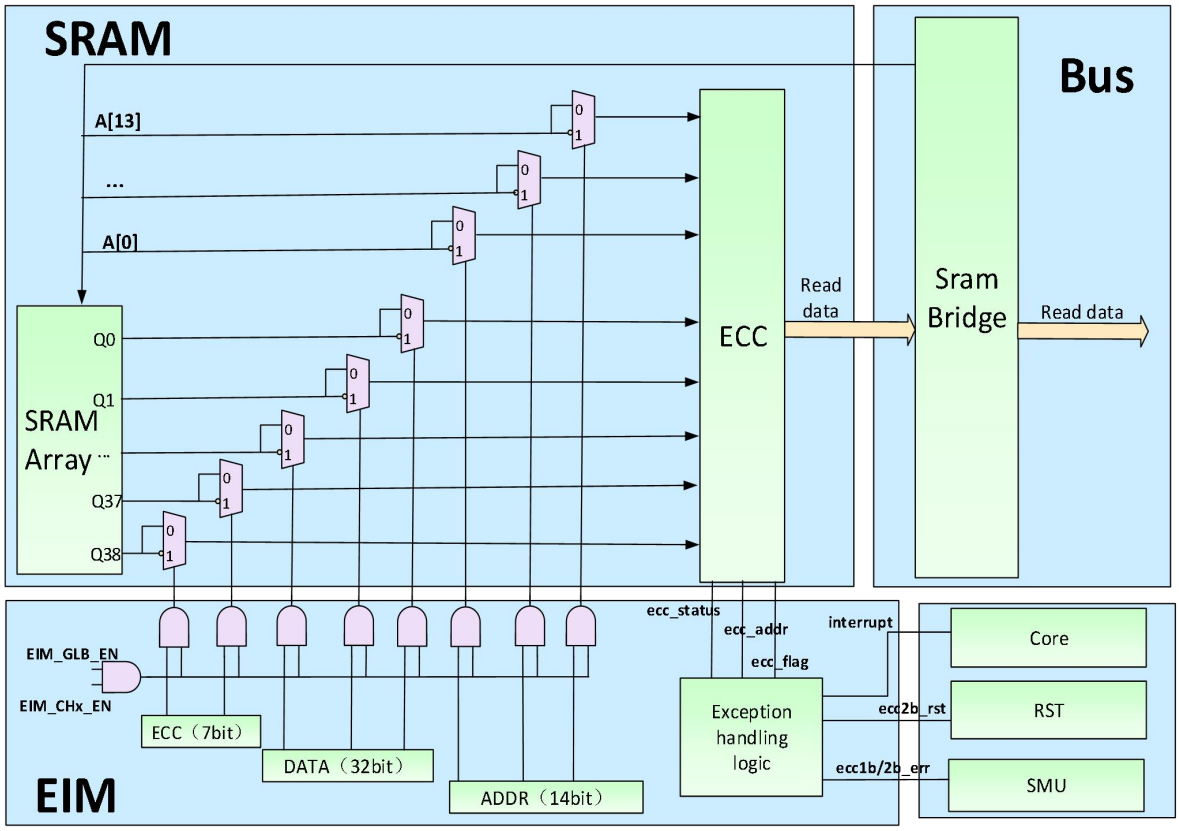


图 18-1 EIM 结构框图

© 2013 - 2024 杰发科技有限公司

18.4 功能描述

18.4.1 错误注入类型

两个错误注入通道：SRAM_L 和 SRAM_U。

对 SRAM 的数据、地址、ECC 校验码进行选择性错误注入，可以通过配置相应通道的 EIMx_INJ_DATA、EIMx_INJ_ADDR、EIMx_INJ_ECC 寄存器来定义要反转的位，注入 1-bit、2-bit、或多 bits 错误。

18.4.2 双重使能机制

防止错误注入功能的意外启用和重新配置，EIM 通过实施两阶段启用机制：

- 全局使能控制
- SRAM_L 和 SRAM_U 的通道使能

18.4.3 调试模式

SRAM 模块没有调试模式配置，在进入调试模式时可正常运行。

18.5 应用说明

18.5.1 错误注入配置流程

1. 配置数据、地址、ECC 校验码对应位是否反转；
2. 打开要进行错误注入的 SRAM 通道的使能位 EIM_CH_EN[EIMx_EN]；
3. 打开 EIM 全局使能位 EIM_GLB_EN[EIM_GLB_EN]；
4. 读取对应 SRAM 通道某一地址，在 ECC 状态、地址和中断标志寄存器里检查错误信息；
5. 进入错误处理流程。

下面举例说明使用 EIM 对 SRAM_L 制造 2-bit 数据错误来验证 ECC 的步骤：

1. 配置 SRAM_L 数据反转寄存器 EIM0_INJ_DATA 为 0x80008000；
2. 使能 SRAM_L 通道的 EIM（EIM_CH_EN[EIM0_EN]写 1）；
3. 使能 EIM 全局使能位(EIM_GLB_EN[EIM_GLB_EN]写 1)；
4. 读取 SRAM_L 地址 0x1FFF6000，此时 ECC 捕获错误信息如下：
   - CH0_STATUS0[ERR_ST]值变为 1（指示 2-bit 错误状态）；

© 2013 - 2024 杰发科技有限公司

- CH0_STATUS1[ERR_STS1]值变为 1（2-bit 中断标志），（ECC_ERR_CTRL [ECC0_2BIT_IRQ_EN]=1 即 SRAM_L 通道 2-bit 错误中断使能的前提下）；
- CH0_STATUS1 [ERR_ADDR1]值变为 0x1800（0x6000<<2），即地址偏移值左移两位；
- 若开启了 2-bit 错误复位使能，读取地址 0x1FFF6000 将导致系统复位。

注意：

SRAM ECC 是以 word 为单位，即读取非 4 字节对齐的 SRAM 地址导致 ECC 错误的，ECC 捕获到的错误地址也是 4 字节对齐。

18.5.2 错误注入关闭流程

1. 关闭 EIM 全局使能位 EIM_GLB_EN[EIM_GLB_EN]；
2. 关闭进行错误注入的 SRAM 通道的使能位 EIM_CH_EN[EIMx_EN]；
3. 清除数据、地址、ECC 校验码反转寄存器。

18.6 寄存器定义

EIM 寄存器只能在特权模式下访问，用户模式下访问触发 hardfault。

EIM 寄存器只支持 32 位访问。

表 18-1 EIM 寄存器映射

EIM 基地址：0x40088000

地址	名称	宽度	描述
ECC_SRAM 基地址+0x0	EIM_GLB_EN	32	EIM 全局使能控制
ECC_SRAM 基地址+0x4	EIM_CH_EN	32	EIM 通道使能控制
ECC_SRAM 基地址+0x10	EIM0_INJ_DATA	32	SRAM_L 数据错误注入控制
ECC_SRAM 基地址+0x14	EIM0_INJ_ADDR	32	SRAM_L 地址错误注入控制
ECC_SRAM 基地址+0x18	EIM0_INJ_ECC	32	SRAM_L ECC 校验码错误注入控制
ECC_SRAM 基地址+0x80	EIM1_INJ_DATA	32	SRAM_U 数据错误注入控制
ECC_SRAM 基地址+0x84	EIM1_INJ_ADDR	32	SRAM_U 地址错误注入控制
ECC_SRAM 基地址+0x88	EIM0_INJ_ECC	32	SRAM_U ECC 校验码错误注入控制

© 2013 - 2024 杰发科技有限公司

18.6.1 EIM 全局使能控制寄存器（EIM_GLB_EN）

表 18-2 EIM 全局使能控制寄存器

EIM_GLB_EN **EIM 全局使能寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																EIM_GLB_EN
访问																RW
复位值																0

字段	说明
0 EIM_GLB_EN	**EIM 全局使能控制** 0: 禁止 1: 使能

18.6.2 EIM 通道使能控制寄存器（EIM_CH_EN）

表 18-3 EIM 通道使能控制寄存器

EIM_CH_EN **EIM 通道使能寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															EIM1_EN	EIM0_EN
访问															RW	RW
复位值															0	0

字段	说明
1 EIM1_EN	**SRAM_U 通道的 EIM 使能控制** 0: 禁止 1: 使能
0 EIM0_EN	**SRAM_L 通道的 EIM 使能控制**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 禁止 1: 使能

18.6.3 SRAM_L 数据错误注入寄存器（EIM0_INJ_DATA）

表 18-4 SRAM_L 数据错误注入寄存器

EIM0_INJ_DATA **SRAM_L 数据错误注入寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	EIM0_INJ_DATA[31:16]															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	EIM0_INJ_DATA[15:0]															
访问	RW															
复位值	0															

字段	说明
31:0 EIM0_INJ_DATA	**SRAM_L 数据位是否反转的控制** 0: 不反转 1: 反转

18.6.4 SRAM_L 地址错误注入寄存器（EIM0_INJ_ADDR）

表 18-5 SRAM_L 地址错误注入寄存器

EIM0_INJ_ADDR **SRAM_L 地址错误注入寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			EIM0_INJ_ADDR[13:0]													
访问			RW													
复位值			0													

© 2013 - 2024 杰发科技有限公司

字段	说明
13:0 EIM0_INJ_ADDR	**SRAM_L 地址位是否反转的控制** 0: 不反转 1: 反转

18.6.5 SRAM_L 校验码错误注入寄存器（EIM0_INJ_ECC）

表 18-6 SRAM_L 校验码错误注入寄存器

EIM0_INJ_ECC　　SRAM_L 校验码错误注入寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										EIM0_INJ_ECC[6:0]						
访问										RW						
复位值										0						

字段	说明
6:0 EIM0_INJ_ECC	**SRAM_L ECC 校验码是否反转的控制** 0: 不反转 1: 反转

18.6.6 SRAM_U 数据错误注入寄存器（EIM1_INJ_DATA）

表 18-7 SRAM_U 数据错误注入寄存器

EIM1_INJ_DATA　　SRAM_U 数据错误注入寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	EIM1_INJ_DATA[31:16]															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	EIM1_INJ_DATA[15:0]															
访问	RW															
复位值	0															

© 2013 - 2024 杰发科技有限公司

字段	说明
31:0 EIM1_INJ_DATA	**SRAM_U 数据位是否反转的控制** 0: 不反转 1: 反转

18.6.7 SRAM_U 地址错误注入寄存器（EIM1_INJ_ADDR）

表 18-8 SRAM_U 地址错误注入寄存器

EIM1_INJ_ADDR　　SRAM_U 地址错误注入寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			EIM1_INJ_ADDR[13:0]													
访问			RW													
复位值			0													

字段	说明
13:0 EIM1_INJ_ADDR	**SRAM_U 地址位是否反转的控制** 0: 不反转 1: 反转

18.6.8 SRAM_U 校验码错误注入寄存器（EIM1_INJ_ECC）

表 18-9 SRAM_U 校验码错误注入寄存器

EIM1_INJ_ECC　　SRAM_U 校验码错误注入寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										EIM1_INJ_ECC[6:0]						
访问										RW						
复位值										0						

© 2013 - 2024 杰发科技有限公司

字段	说明
6:0 EIM1_INJ_ECC	**SRAM_U ECC 校验码是否反转的控制** 0: 不反转 1: 反转

© 2013 - 2024 杰发科技有限公司

19 通用输入/输出（GPIO）

19.1 简介

通用输入输出（General-Purpose Input/Output, GPIO）模块包含两块逻辑：PORT 和 GPIO。其中 PORT 端口主要用于引脚功能配置，数字滤波和中断相关功能。GPIO 主要用于配置引脚的方向和电平逻辑。

当引脚配置为 GPIO 功能时，端口配置寄存器 GPIO_POER 控制每个引脚的方向。

端口输出数据寄存器 GPIO_PODR 控制每个引脚输出电平，也可以通过端口输出置位寄存器 GPIO_PSOR 置位引脚的电平，还可以通过端口输出复位寄存器 GPIO_PROR 复位引脚的电平。

当引脚配置用于输入功能时，GPIO 输入数据寄存器 GPIO_PIDR 显示每个引脚上的高低电平（1 代表高电平，0 代表低电平）。

同一个 PORT 的 32 个引脚，共享一个中断向量。引脚的中断支持上升沿，下降沿或双边沿触发，可通过 PORT_PCR[IRQC]寄存器进行配置。

MCU I/O 引脚通过多路复用器连接到外设，多路复用器一次只允许一个外设的复用功能连接到 I/O 引脚。这样，共享同一个 I/O 引脚的外设之间不会发生冲突。每个 I/O 引脚都有一个多路复用器，可通过 PORT_PCR[PINMUX]寄存器进行配置。当某一个外设的功能需要从当前 IO 转移到另一个 IO 时，除了新的 IO 需要将复用功能连接到该外设，原 IO 的复用配置也需要关闭，否则会导致外设在新 IO 管脚工作异常。

19.2 特性

GPIO 模块支持如下特性：

- 端口控制
  - 支持配置为高阻：上电默认为复用功能 0（高阻或模拟），除了上电为特殊复用功能的引脚，如 SWD。
  - 输入状态：浮空，上拉/下拉，模拟(和 ADC, ACMP 有关)
  - 输出状态：推挽或开漏 。除了 I2C 的复用功能为开漏，其它复用功能都是推挽。
  - 高灵活度的引脚复用，允许将 I/O 引脚用作 GPIO 或作为多种外设功能之一
  - 部分 I/O 的驱动能力可选：高/低驱动能力
  - 端口输入数据寄存器 GPIO_PIDR 对于所有的数字复用功能都有效
  - 端口输出数据寄存器 GPIO_PODR 也可以通过端口输出置位/复位/翻转寄存器来配置
  - 引脚的方向可以通过 GPIO_POER 和 GPIO_PIER 来可控制
- 引脚中断

© 2013 - 2024 杰发科技有限公司

- 每个 IO 都有单独的中断使能和状态标志位
- 每个 IO 支持边沿触发：上升沿，下降沿，双边沿
- 每个 IO 支持配置为中断或 DMA 请求
- 支持从低功耗模式唤醒

■ 数字输入滤波

- 每个 IO 的数字输入滤波功能支持单独配置
- 数字滤波的时钟源可以配置，滤波长度最大支持 31 个时钟周期
- 数字滤波配置对于引脚上所有的数字复用功能都有效

19.3 结构框图

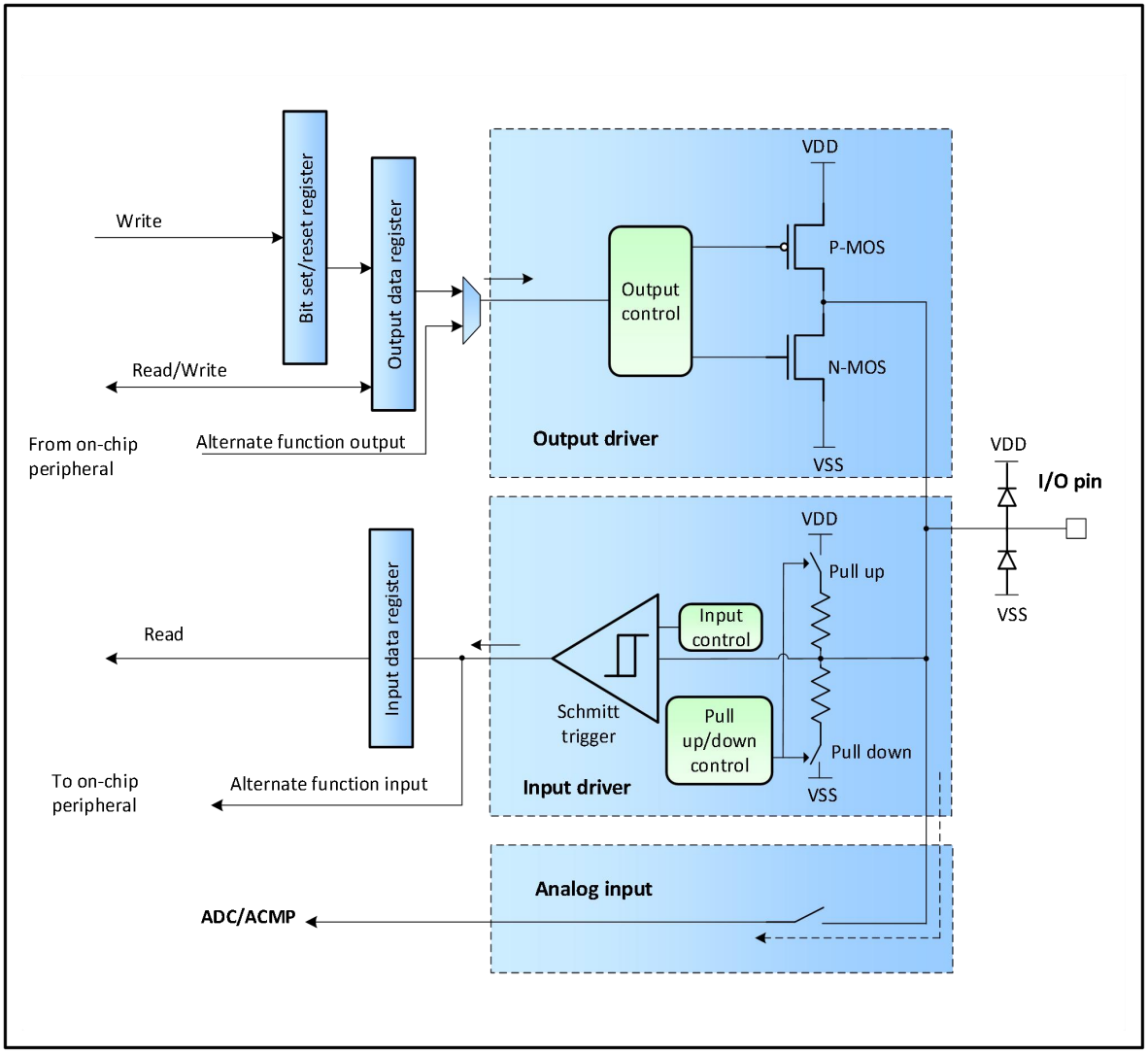


图 19-1 GPIO 结构框图

© 2013 - 2024 杰发科技有限公司

19.4 功能描述

19.4.1 外部中断

每个 IO 支持单独配置为下面的外部中断模式：

- 上升沿触发中断
- 下降沿触发中断
- 上升沿或下降沿触发中断

每个 PORT 都有一个独立的中断向量，如 PORTA, B, C, D, E。同一个 PORT 里面的 32 个 PIN 共享一个中断向量。每个 IO 的都有单独的中断配置和状态标志位，因此即使是同一个 PORT 的 32 个 PIN 中断是支持同时配置的。

当 IO 配置为外部中断时， 如果 IO 的数字输入滤波功能，脉冲的宽度必须大于滤波的总宽度，中断/DMA 请求才会响应。产生了有效的外部中断后，中断标志标志位将会被置位，并且中断会保持有效直到状态标志通过 PORT_ISFR 或 PORT_PCR 的 ISF bit 写“1”清除。

19.4.2 DMA 请求

每个 IO 支持单独配置为下面的 DMA 请求模式：

- 上升沿触发 DMA 请求
- 下降沿触发 DMA 请求
- 上升沿或下降沿触发 DMA 请求

DMA 请求可以配置为 PORTA, B, C, D, E，同一个 PORT 里面的任意一个 IO 都可以产生该 PORT 的 DMA 请求。DMA 传输完成当次的传输后，中断标志位会被自动清除。

19.4.3 数字滤波

每个 IO 的数字输入滤波功能支持单独配置，并且对于该引脚上所有的数字复用功能都有效。

同一个 PORT 的数字滤波功能时钟支持配置为 BUS clock 或 LSI clock。如果需要切换数字滤波功能时钟源，必须先禁用该 PORT 所有 IO 的数字滤波功能。

数字滤波的最大宽度为 31 个滤波功能时钟周期。如果需要修改数字滤波宽度，必须先禁用该 PORT 所有 IO 的数字滤波功能。

© 2013 - 2024 杰发科技有限公司

19.4.4 复用功能

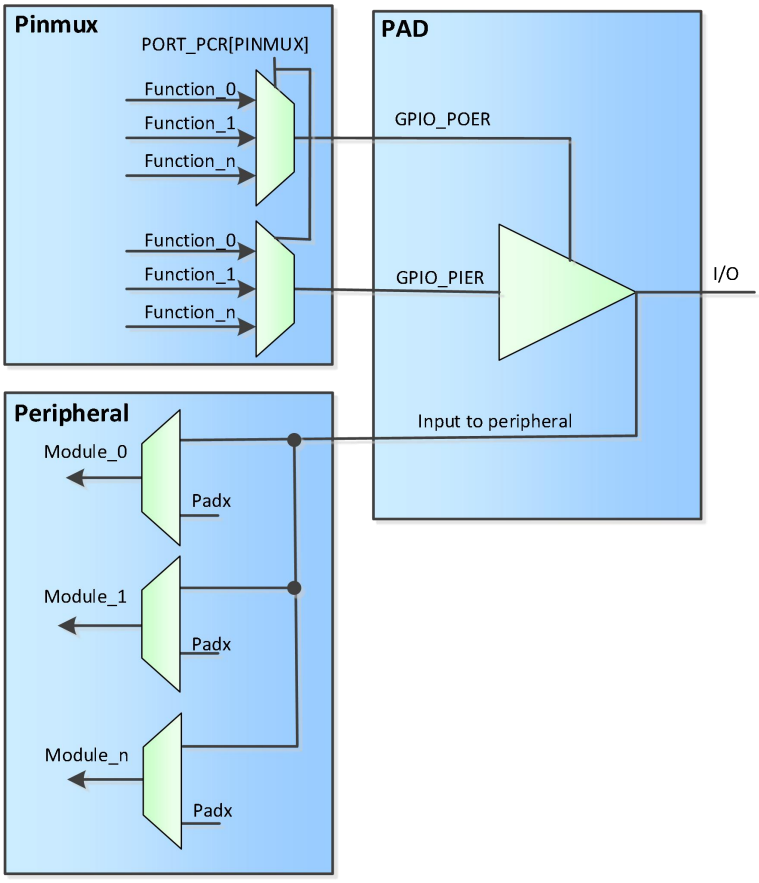


图 19-2 GPIO 复用功能

在上电后/复位期间/复位后，除了 PA4, PA5, PA10, PC4, PC5 引脚为复用功能 7， 其它所有的引脚都为复用功能 0。每个 IO 都有复用功能，如果要开始外设通信，应该先配置 GPIO 复用功能。每个 GPIO 的相应复用功能描述如下。

表 19-1 GPIO 复用功能描述

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
	1	1	PE16	HIGH_Z	GPIO	UART1_RTS	SPI2_SIN	PWM2_CH7	PWM4_FLT0	EIO_D3	TRGMUX_OUT7
	2	2	PE15	HIGH_Z	GPIO	UART1_CTS	SPI2_SCK	PWM2_CH6	PWM4_FLT1	EIO_D2	TRGMUX_OUT6
1	3	3	PD1	HIGH_Z	GPIO	PWM0_CH3	SPI1_SIN	PWM2_CH1	HIGH_Z	EIO_D1	TRGMUX_OUT2
2	4	4	PD0	HIGH_Z	GPIO	PWM0_CH2	SPI1_SCK	PWM2_CH0	HIGH_Z	EIO_D0	TRGMUX_O

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
											UT1
3	5	5	PE11	HIGH_Z	GPIO	SPI2_CS0	PCT0_ALT1	PWM2_CH5	HIGH_Z	EIO_D5	TRGMUX_OUT5
4	6	6	PE10	HIGH_Z	GPIO	CLKOUT	SPI2_CS1	PWM2_CH4	CAN0_STB	EIO_D4	TRGMUX_OUT4
	7	7	PE13	HIGH_Z	GPIO	PWM4_CH5	SPI2_CS2	PWM2_FLT0	HIGH_Z	HIGH_Z	HIGH_Z
5	8	8	PE5	HIGH_Z	GPIO	PWM_CLK2	PWM2_CH0	PWM2_CH3	CAN0_TX	EIO_D7	EWDG_IN
6	9	9	PE4	HIGH_Z	GPIO	HIGH_Z	PWM2_CH1	PWM2_CH2	CAN0_RX	EIO_D6	EWDG_OUT_b
		10	PA25	HIGH_Z	GPIO	PWM5_CH0	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
7	10	11	VDD								
		12	VSS								
8	11	13	VDDA								
9	12	14	VREFH								
	13	15	VREFL								
10	14	16	VSS								
11	15	17	PB7	XOSC_IN	GPIO	I2C0_SCL	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
12	16	18	PB6	XOSC_OUT	GPIO	I2C0_SDA	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		19	PA26	HIGH_Z	GPIO	PWM5_CH1	SPI1_CS0	SPI0_CS0	HIGH_Z	HIGH_Z	HIGH_Z
	17	20	PE14	HIGH_Z	GPIO	PWM0_FLT1	HIGH_Z	PWM2_FLT1	HIGH_Z	HIGH_Z	HIGH_Z
13	18	21	PE3	HIGH_Z	GPIO	PWM0_FLT0	UART2_RTS	PWM2_FLT0	CAN3_TX	TRGMUX_IN6	ACMP0_OUT
		22	PA27	HIGH_Z	GPIO	PWM5_CH2	SPI1_SOUT	UART0_TX	CAN0_TX	HIGH_Z	HIGH_Z
	19	23	PE12	HIGH_Z	GPIO	PWM0_FLT3	UART2_TX	PWM5_FLT0	CAN3_RX	HIGH_Z	HIGH_Z
		24	PA28	HIGH_Z	GPIO	PWM5_CH3	SPI1_SCK	UART0_RX	CAN0_RX	HIGH_Z	HIGH_Z
	20	25	PD17	HIGH_Z	GPIO	PWM0_FLT2	UART2_RX	PWM5_FLT1	CAN3_STB	HIGH_Z	HIGH_Z
		26	PA29	HIGH	GPIO	PWM5	HIGH	UART	SPI1_	HIGH	HIGH

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
				_Z		_CH4	_Z	2_TX	SIN	_Z	_Z
		27	PA30	HIGH_Z	GPIO	PWM5_CH5	UART2_RX	SPI0_SOUT	CAN0_STB	HIGH_Z	HIGH_Z
14	21	28	PD16	HIGH_Z	GPIO	PWM0_CH1	HIGH_Z	SPI0_SIN	ACMP0_RRT	HIGH_Z	HIGH_Z
15	22	29	PD15	HIGH_Z	GPIO	PWM0_CH0	HIGH_Z	SPI0_SCK	HIGH_Z	HIGH_Z	HIGH_Z
16	23	30	PE9	HIGH_Z	GPIO	PWM0_CH7	UART2_CTS	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		31	VSS								
		32	VDD								
		33	PA31	HIGH_Z	GPIO	PWM5_CH6	HIGH_Z	SPI0_CS1	HIGH_Z	HIGH_Z	HIGH_Z
	24	34	PD14	HIGH_Z	GPIO	PWM2_CH5	UART1_TX	HIGH_Z	HIGH_Z	HIGH_Z	CLKOUT
	25	35	PD13	HIGH_Z	GPIO	PWM2_CH4	UART1_RX	HIGH_Z	HIGH_Z	HIGH_Z	RTC_CLKOUT
		36	PB18	ADC0_IN16	GPIO	PWM5_CH7	HIGH_Z	SPI1_CS1	HIGH_Z	HIGH_Z	HIGH_Z
		37	PB20	ADC0_IN17	GPIO	HIGH_Z	HIGH_Z	UART3_TX	HIGH_Z	HIGH_Z	HIGH_Z
		38	PB21	ADC0_IN18	GPIO	HIGH_Z	HIGH_Z	UART3_RX	HIGH_Z	HIGH_Z	HIGH_Z
17	26	39	PE8	ACMP0_IN3	GPIO	PWM0_CH6	CAN0_STB	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
18	27	40	PB5	HIGH_Z	GPIO	PWM0_CH5	SPI0_CS1	SPI0_CS0	CLKOUT	TRGMUX_IN0	HIGH_Z
19	28	41	PB4	HIGH_Z	GPIO	PWM0_CH4	SPI0_SOUT	HIGH_Z	HIGH_Z	TRGMUX_IN1	HIGH_Z
20	29	42	PC3	ADC0_IN11/ACMP0_IN4	GPIO	PWM0_CH3	CAN0_TX	UART0_TX	HIGH_Z	HIGH_Z	HIGH_Z
21	30	43	PC2	ADC0_IN10/ACMP0_IN5	GPIO	PWM0_CH2	CAN0_RX	UART0_RX	HIGH_Z	HIGH_Z	HIGH_Z
22	31	44	PD7	ACMP0_IN6	GPIO	UART2_TX	HIGH_Z	PWM2_FLT3	HIGH_Z	HIGH_Z	HIGH_Z

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
23	32	45	PD6	ACMP0_IN7	GPIO	UART2_RX	HIGH_Z	PWM2_FLT2	HIGH_Z	HIGH_Z	HIGH_Z
24	33	46	PD5	HIGH_Z	GPIO	PWM2_CH3	PCT0_ALT2	PWM2_FLT1	HIGH_Z	TRGMUX_IN7	HIGH_Z
	34	47	PD12	HIGH_Z	GPIO	PWM2_CH2	HIGH_Z	HIGH_Z	HIGH_Z	UART2_RTS	HIGH_Z
	35	48	PD11	HIGH_Z	GPIO	PWM2_CH1	PWM2_CH0	HIGH_Z	HIGH_Z	UART2_CTS	HIGH_Z
	36	49	PD10	HIGH_Z	GPIO	PWM2_CH0	PWM2_CH1	HIGH_Z	HIGH_Z	CLKOUT	HIGH_Z
	37	50	VSS								
	38	51	VDD								
25	39	52	PC1	ADC0_IN9	GPIO	PWM0_CH1	SPI2_SOUT	HIGH_Z	HIGH_Z	PWM1_CH7	HIGH_Z
26	40	53	PC0	ADC0_IN8	GPIO	PWM0_CH0	SPI2_SIN	HIGH_Z	HIGH_Z	PWM1_CH6	HIGH_Z
	41	54	PD9	HIGH_Z	GPIO	HIGH_Z	EIO_D0	PWM2_FLT3	HIGH_Z	PWM1_CH5	HIGH_Z
	42	55	PD8	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	PWM2_FLT2	EIO_D1	PWM1_CH4	HIGH_Z
27	43	56	PC17	ADC0_IN15	GPIO	PWM1_FLT3	CAN2_TX	HIGH_Z	HIGH_Z	UART3_TX	HIGH_Z
28	44	57	PC16	ADC0_IN14	GPIO	PWM1_FLT2	CAN2_RX	HIGH_Z	HIGH_Z	UART3_RX	HIGH_Z
		58	PB22	ADC0_IN19	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	UART1_TX	HIGH_Z	HIGH_Z
29	45	59	PC15	ADC0_IN13	GPIO	PWM1_CH3	SPI2_SCK	HIGH_Z	CAN2_STB	TRGMUX_IN8	HIGH_Z
		60	PB23	ADC0_IN20	GPIO	HIGH_Z	UART1_RX	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
30	46	61	PC14	ADC0_IN12	GPIO	PWM1_CH2	SPI2_CS0	HIGH_Z	HIGH_Z	TRGMUX_IN9	HIGH_Z
		62	PB25	ADC0_IN21	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	SPI2_CS0	HIGH_Z	HIGH_Z
31	47	63	PB3	ADC0_IN7	GPIO	PWM1_CH1	SPI0_SIN	PWM1_CH0	PWM2_CH2	TRGMUX_IN2	HIGH_Z
		64	PB27	ADC0_IN22	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	SPI2_SOUT	HIGH_Z	HIGH_Z
		65	PB28	ADC0_IN23	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	SPI2_SIN	HIGH_Z	HIGH_Z

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
		66	VSS								
		67	VDD								
32	48	68	PB2	ADC0_IN6	GPIO	PWM1_CH0	SPI0_SCK	PWM1_CH1	HIGH_Z	TRGMUX_IN3	HIGH_Z
		69	PB29	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	SPI2_SCK	HIGH_Z	HIGH_Z
	49	70	PC13	HIGH_Z	GPIO	PWM3_CH7	PWM2_CH7	UART2_RTS	CAN3_TX	HIGH_Z	HIGH_Z
	50	71	PC12	HIGH_Z	GPIO	PWM3_CH6	PWM2_CH6	UART2_CTS	CAN3_RX	HIGH_Z	HIGH_Z
		72	PC19	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	SPI2_CS1	HIGH_Z	HIGH_Z
		73	PC23	HIGH_Z	GPIO	SPI0_SCK	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
	51	74	PC11	HIGH_Z	GPIO	PWM3_CH5	PWM4_CH2	HIGH_Z	CAN3_STB	TRGMUX_IN10	HIGH_Z
	52	75	PC10	HIGH_Z	GPIO	PWM3_CH4	HIGH_Z	HIGH_Z	HIGH_Z	TRGMUX_IN11	HIGH_Z
		76	PC27	HIGH_Z	GPIO	PWM4_CH4	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
33	53	77	PB1	ADC0_IN5/ ADC1_INTERLEAVE1	GPIO	UART0_TX	SPI0_SOUT	PWM_CLK0	CAN0_TX	PWM4_CH5	HIGH_Z
34	54	78	PB0	ADC0_IN4/ ADC1_INTERLEAVE0	GPIO	UART0_RX	SPI0_CS0	PCT0_ALT3	CAN0_RX	PWM4_CH6	HIGH_Z
		79	PC28	HIGH_Z	GPIO	PWM4_CH7	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
35	55	80	PC9	HIGH_Z	GPIO	UART1_TX	PWM1_FLT1	PWM5_CH0	CAN0_STB	UART0_RTS	HIGH_Z
36	56	81	PC8	HIGH_Z	GPIO	UART1_RX	PWM1_FLT0	PWM5_CH1	HIGH_Z	UART0_CTS	HIGH_Z
		82	PC29	HIGH_Z	GPIO	PWM5_CH2	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
37	57	83	PA7	ADC0_IN3	GPIO	PWM0_FLT2	PWM5_CH3	RTC_CLKI	HIGH_Z	UART1_RTS	HIGH_Z

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
								N			
		84	PC30	HIGH_Z	GPIO	PWM5_CH4	EIO_D0	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
38	58	85	PA6	ADC0_IN2	GPIO	PWM0_FLT1	SPI1_CS1	PWM5_CH5	HIGH_Z	UART1_CTS	HIGH_Z
		86	PC31	HIGH_Z	GPIO	PWM5_CH6	EIO_D1	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
39	59	87	PE7	HIGH_Z	GPIO	PWM0_CH7	PWM3_FLT0	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		88	PD18	ADC1_IN16	GPIO	PWM5_CH7	EIO_D2	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		89	PD19	ADC1_IN17	GPIO	HIGH_Z	EIO_D3	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
40	60	90	VSS								
41	61	91	VDD								
	62	92	PA17	HIGH_Z	GPIO	PWM0_CH6	PWM3_FLT0	EWDG_OUT_b	PWM5_FLT0	HIGH_Z	HIGH_Z
	63	93	PB17	HIGH_Z	GPIO	PWM0_CH5	SPI1_CS3	PWM5_FLT1	HIGH_Z	HIGH_Z	HIGH_Z
	64	94	PB16	ADC1_IN15	GPIO	PWM0_CH4	SPI1_SOUT	UART3_TX	HIGH_Z	HIGH_Z	HIGH_Z
	65	95	PB15	ADC1_IN14	GPIO	PWM0_CH3	SPI1_SIN	UART3_RX	HIGH_Z	HIGH_Z	HIGH_Z
	66	96	PB14	ADC1_IN9 / ADC0_INTERLEAVE1	GPIO	PWM0_CH2	SPI1_SCK	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
42	67	97	PB13	ADC1_IN8 / ADC0_INTERLEAVE0	GPIO	PWM0_CH1	PWM3_FLT1	CAN2_TX	HIGH_Z	HIGH_Z	HIGH_Z
43	68	98	PB12	ADC1_IN7	GPIO	PWM0_CH0	PWM3_FLT2	CAN2_RX	HIGH_Z	HIGH_Z	HIGH_Z
		99	PD22	ADC1_IN18	GPIO	HIGH_Z	HIGH_Z	CAN3_TX	HIGH_Z	HIGH_Z	HIGH_Z
44	69	100	PD4	ADC1_IN6	GPIO	PWM0_FLT3	PWM3_FLT3	HIGH_Z	CAN2_STB	HIGH_Z	HIGH_Z
45	70	101	PD3	ADC1_IN3	GPIO	PWM3_CH5	SPI1_CS0	EIO_D5	EIO_D7	TRGMUX_IN4	NMI_b

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
46	71	102	PD2	ADC1_IN2	GPIO	PWM3_CH4	SPI1_SOUT	EIO_D4	EIO_D6	TRGMUX_IN5	HIGH_Z
		103	PD23	ADC1_IN19	GPIO	HIGH_Z	HIGH_Z	CAN3_RX	HIGH_Z	HIGH_Z	HIGH_Z
47	72	104	PA3	ADC1_IN1	GPIO	PWM3_CH1	I2C0_SCL	EWDG_IN	EIO_D5	UART0_TX	HIGH_Z
48	73	105	PA2	ADC1_IN0	GPIO	PWM3_CH0	I2C0_SDA	EWDG_OUT_b	EIO_D4	UART0_RX	HIGH_Z
		106	PD24	ADC1_IN20	GPIO	HIGH_Z	HIGH_Z	CAN3_STB	HIGH_Z	HIGH_Z	HIGH_Z
	74	107	PB11	HIGH_Z	GPIO	PWM3_CH3	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
	75	108	PB10	HIGH_Z	GPIO	PWM3_CH2	I2C0_SDA	HIGH_Z	UART3_TX	HIGH_Z	HIGH_Z
	76	109	PB9	HIGH_Z	GPIO	PWM3_CH1	I2C0_SCL	HIGH_Z	UART3_RX	HIGH_Z	HIGH_Z
		110	PD27	ADC1_IN21	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
	77	111	PB8	HIGH_Z	GPIO	PWM3_CH0	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		112	PD28	ADC1_IN22	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
49	78	113	PA1	ADC0_IN1 / ACMP0_IN1	GPIO	PWM1_CH1	PWM2_CH2	EIO_D3	PWM1_CH0	UART0_RTS	TRGMUX_OUT0
		114	PD29	ADC1_IN23	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
50	79	115	PA0	ADC0_IN0 / ACMP0_IN0	GPIO	PWM2_CH1	HIGH_Z	EIO_D2	PWM2_CH0	UART0_CTS	TRGMUX_OUT3
		116	PD30	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
51	80	117	PC7	ADC1_IN5	GPIO	UART1_TX	CAN1_TX	PWM3_CH3	HIGH_Z	PWM1_CH0	HIGH_Z
52	81	118	PC6	ADC1_IN4	GPIO	UART1_RX	CAN1_RX	PWM3_CH2	HIGH_Z	PWM1_CH1	HIGH_Z
	82	119	PA16	ADC1_IN13	GPIO	PWM1_CH3	SPI1_CS2	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
	83	120	PA15	ADC1_IN12	GPIO	PWM1_CH2	SPI0_CS3	SPI2_CS3	HIGH_Z	HIGH_Z	HIGH_Z
53	84	121	PE6	ADC1	GPIO	SPI0_	HIGH	PWM3	CAN1	UART	HIGH

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
				_IN11		CS2	_Z	_CH7	_STB	1_RTS	_Z
54	85	122	PE2	ADC1_IN10	GPIO	SPI0_SOUT	PCT0_ALT3	PWM3_CH6	HIGH_Z	UART1_CTS	HIGH_Z
	86	123	VSS								
	87	124	VDD								
		125	PE19	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		126	PE20	HIGH_Z	GPIO	PWM4_CH0	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
	88	127	PA14	HIGH_Z	GPIO	PWM0_FLT0	PWM3_FLT1	EWDG_IN	HIGH_Z	PWM1_FLT0	HIGH_Z
		128	PE21	HIGH_Z	GPIO	PWM4_CH1	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		129	PE22	HIGH_Z	GPIO	PWM4_CH2	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
55	89	130	PA13	HIGH_Z	GPIO	PWM1_CH7	CAN1_TX	HIGH_Z	HIGH_Z	PWM2_CH0	HIGH_Z
		131	PE23	HIGH_Z	GPIO	PWM4_CH3	CAN2_STB	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		132	PE24	HIGH_Z	GPIO	PWM4_CH4	CAN2_TX	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
		133	PE25	HIGH_Z	GPIO	PWM4_CH5	CAN2_RX	HIGH_Z	HIGH_Z	HIGH_Z	HIGH_Z
56	90	134	PA12	HIGH_Z	GPIO	PWM1_CH6	CAN1_RX	HIGH_Z	HIGH_Z	PWM2_CH1	HIGH_Z
57	91	135	PA11	HIGH_Z	GPIO	PWM1_CH5	CAN1_STB	EIO_D1	ACMP0_RRT	HIGH_Z	HIGH_Z
58	92	136	PA10	HIGH_Z	GPIO	PWM1_CH4	HIGH_Z	EIO_D0	HIGH_Z	HIGH_Z	JTAG_TDO/noetm_TRACE_SWO
59	93	137	PE1	HIGH_Z	GPIO	SPI0_SIN	HIGH_Z	HIGH_Z	SPI1_CS0	PWM1_FLT1	HIGH_Z
60	94	138	PE0	HIGH_Z	GPIO	SPI0_SCK	PWM_CLK1	HIGH_Z	SPI1_SOUT	PWM1_FLT2	HIGH_Z
61	95	139	PC5	HIGH_Z	GPIO	PWM2_CH0	RTC_CLKOUT	HIGH_Z	HIGH_Z	PWM2_CH1	JTAG_TDI
62	96	140	PC4	ACMP0_IN2	GPIO	PWM1_CH0	RTC_CLKOUT	HIGH_Z	EWDG_IN	PWM1_CH1	JTAG_TCLK/SWD_CL

© 2013 - 2024 杰发科技有限公司

64LQFP	100LQFP	144LQFP	引脚名称	功能 0	功能 1	功能 2	功能 3	功能 4	功能 5	功能 6	功能 7
											K
63	97	141	PA5	HIGH_Z	GPIO	HIGH_Z	PWM_CLK1	HIGH_Z	HIGH_Z	HIGH_Z	RESET_b
64	98	142	PA4	HIGH_Z	GPIO	HIGH_Z	HIGH_Z	ACMP0_OUT	EWDG_OUT_b	HIGH_Z	JTAG_TMS/SWD_DIO
	99	143	PA9	HIGH_Z	GPIO	UART2_TX	SPI2_CS0	EIO_D7	PWM3_FLT2	PWM1_FLT3	PWM4_FLT0
	100	144	PA8	HIGH_Z	GPIO	UART2_RX	SPI2_SOUT	EIO_D6	PWM3_FLT3	PWM4_FLT1	HIGH_Z

注意：

- 除了底色标注为绿色(如 JTAG_TDO 等)，上电默认为复用功能 7。其它所有引脚上电默认为复用功能 0。
- NMI_b， RESET_b 的后缀_b 代表的是低电平有效。
- ADC0_INTERLEAVE0 、 ADC0_INTERLEAVE1 、 ADC1_INTERLEAVE0 、 ADC1_INTERLEAVE1 是 ADC0 和 ADC1 部分通道复用，实现 ADC0 输入通道和 ADC1 输入通道相关功能安全检测以及时钟交织等功能。

19.4.5 中断请求

如表 19-2 所示，GPIO 共支持 5 个中断向量，每个 PORT 的 32 pins 共享一个中断向量。

表 19-2 GPIO 模块中断汇总

中断源	标志	本地使能	全局使能
PORTA 任意一个 IO 产生中断	PORTA_ISFR	IRQC	
PORTB 任意一个 IO 产生中断	PORTB_ISFR	IRQC	
PORTC 任意一个 IO 产生中断	PORTC_ISFR	IRQC	
PORTD 任意一个 IO 产生中断	PORTD_ISFR	IRQC	
PORTE 任意一个 IO 产生中断	PORTE_ISFR	IRQC	

19.4.6 低功耗模式

GPIO 模块唤醒源，参见表 19-3。

© 2013 - 2024 杰发科技有限公司

表 19-3 GPIO 模块唤醒源

模式	唤醒源	备注
STOP1	GPIO 中断	如果使能了数字滤波功能，只能选择 LSI 作为功能时钟，BUS clock 会被关闭
STOP2	GPIO 中断	如果使能了数字滤波功能， LSI 或 BUS clock 都可以作为功能时钟
VLPS	GPIO 中断	如果使能了数字滤波功能，只能选择 LSI 作为功能时钟，BUS clock 会被关闭

19.4.7 调试模式

GPIO 模块不支持 Debug 位控制，不受调试状态的影响。

19.5 应用说明

19.5.1 开漏输出

GPIO PIN 没有单独配置为开漏输出的寄存器，但是可以通过寄存器的组合配置以达到开漏输出的功能。

- **模拟开漏输出高（需要外接上拉电阻）：**
  1. 配置该引脚为输入模式，即该 PIN 的 GPIO_POER 寄存器对应 BIT 设置为 0，GPIO_PIER 寄存器对应 BIT 设置为 1；
  2. 端口输出数据寄存器 GPIO_PODR 对应 BIT 设置为 0；
  3. 配置该引脚不带上拉，也不带下拉。即 PORT_PCR 寄存器 PU, PD BIT 配置为 0
- **模拟开漏输出低：**
  1. 配置该引脚为输出模式，即该 PIN 的 GPIO_POER 寄存器对应 BIT 设置为 1；
  2. 端口输出数据寄存器 GPIO_PODR 对应 BIT 设置为 0；
  3. 配置该引脚不带上拉，也不带下拉。即 PORT_PCR 寄存器 PU, PD BIT 配置为 0。

19.5.2 编程指南

- **GPIO 配置为输出：**
  1. 配置复用功能为 GPIO，即配置 PORT_PCR[PINMUX] 为 1。
  2. 配置引脚的输出电平，即 GPIO_PODR 寄存器对应的 BIT 为 0 或 1。
  3. 使能引脚的输出功能，即 GPIO_POER 寄存器对应的 BIT 设置为 1。
- **GPIO 配置为输入：**

© 2013 - 2024 杰发科技有限公司

1. 配置复用功能为 GPIO，即配置 PORT_PCR[PINMUX]为 1。
2. 配置上拉或下拉电阻，即配置 PORT_PCR[PU] 或 PORT_PCR[PD]。
3. 如果需要使能数字滤波，先配置 PORT_DFCR[CS]，选择滤波的时钟源。再配置 PORT_DFWRPORT_DFWR[DFW]，选择滤波的宽度。最后，配置 PORT_DFER 寄存器，使能对应引脚的数字滤波功能。
4. 如果需要使能中断功能，配置 PORT_PCR[IRQC]，选择中断上升沿，下降沿或双边沿触发。
5. 使能引脚的输入功能，即 GPIO_PIER 寄存器对应的 BIT 设置为 1。
6. 读取引脚的输入电平，即读取 GPIO_PIDR 寄存器对应的 BIT。输入数据寄存器 GPIO_PIDR 在每个 AHB 时钟周期捕获 I/O 引脚上的数据。

19.6 寄存器定义

通用输入输出 (GPIO) 模块寄存器支持 AHB 32/16/8 位访问。

表 19-4 GPIO 寄存器映射

PORTA 基地址： 0x40085200
PORTB 基地址： 0x40085300
PORTC 基地址： 0x40085400
PORTD 基地址： 0x40085500
PORTE 基地址： 0x40085600

GPIOA 基地址： 0x40085000
GPIOB 基地址： 0x40085040
GPIOC 基地址： 0x40085080
GPIOD 基地址： 0x400850C0
GPIOE 基地址： 0x40085100

地址	名称	宽度	描述
PORTx 基地址 + 0x00 PORTx 基地址 + 0x04 PORTx 基地址 + 0x08 … PORTx 基地址 + 0x7C	PORT_PCR	32	端口配置寄存器： 每个 PORT 都有 32 个 PCR 寄存器， 分别控制同一 PORT 里面的 32 个 PIN
PORTx 基地址 + 0xA0	PORT_ISFR	32	端口中断状态标志寄存器
PORTx 基地址 + 0xC0	PORT_DFER	32	端口数字滤波使能寄存器
PORTx 基地址 + 0xC4	PORT_DFCR	32	端口数字滤波时钟寄存器
PORTx 基地址 + 0xC8	PORT_DFWR	32	端口数字滤波宽度寄存器
GPIOx 基地址 + 0x00	GPIO_PODR	32	端口输出数据寄存器
GPIOx 基地址 + 0x04	GPIO_PSOR	32	端口输出置位寄存器
GPIOx 基地址 + 0x08	GPIO_PROR	32	端口输出复位寄存器
GPIOx 基地址 + 0x0C	GPIO_PIOR	32	端口输出翻转寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
GPIOx 基地址 + 0x10	GPIO_PIDR	32	端口输入数据寄存器
GPIOx 基地址 + 0x14	GPIO_POER	32	端口输出使能寄存器
GPIOx 基地址 + 0x18	GPIO_PIER	32	端口输入使能寄存器



说明：

上表中，x=A、B、C、D、E。

19.6.1 端口配置寄存器(PORT_PCR)

表 19-5 PORT_PCR 寄存器

PORT_PCR **端口配置寄存器** **复位值：**

0x00007808/0x00007F08

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称								ISF					IRQC			
访问								R/W1C					RW			
复位值								0					0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	LK					PINMUX				DSE					PD	PU
访问	RW					RW				RW					RW	RW
复位值	0					0	0	0		0					0	0

字段	说明
24 ISF	**中断状态标志位** 0: 没有产生中断 1: 已经产生中断。如果该引脚被配置为 DMA 请求，当前的 DMA 请求传输完成后，该状态标志位会被自动清除。 写“1”可清除该标志位。
19:16 IRQC	**中断配置** 0000b: 中断状态标志位检测不使能 0001b: 上升沿触发中断状态标志位和 DMA 请求 0010b: 下降沿触发中断状态标志位和 DMA 请求 0011b: 上升沿或下降沿触发中断状态标志位和 DMA 请求 1001b: 上升沿触发中断状态标志位和中断

© 2013 - 2024 杰发科技有限公司

字段	说明
	1010b: 下降沿触发中断状态标志位和中断 1011b: 上升沿或下降沿触发中断状态标志位和中断
15 LK	**锁寄存器** 0: PCR 寄存器 bit0~15 锁不使能 1: PCR 寄存器 bit0~15 锁使能，直到系统复位或模块复位才能重新配置
10:8 PINMUX	**复用功能控制** 000b: 复用功能 0(高阻或模拟功能) 001b: 复用功能 1(GPIO) 010b: 复用功能 2 011b: 复用功能 3 100b: 复用功能 4 101b: 复用功能 5 110b: 复用功能 6 111b: 复用功能 7 **说明：上电或复位后，PA4, PA5, PA10, PC4, PC5 引脚的复用功能默认为 7。其它引脚的复用功能默认值为 0。**
6 DSE	**驱动能力选择** 0： 低驱动能力(5 mA) 1： 高驱动能力(20 mA) 驱动能力的配置对于该引脚上所有的数字复用功能都有效。
1 PD	**下拉** 0： 内部下拉不使能 1： 内部下拉使能 下拉的配置对于该引脚上所有的数字复用功能都有效。
0 PU	**上拉** 0： 内部上拉不使能 1： 内部上拉使能 上拉的配置对于该引脚上所有的数字复用功能都有效。

© 2013 - 2024 杰发科技有限公司

19.6.2 端口中断状态标志寄存器 (PORT_ISFR)

表 19-6 PORT_ISFR 寄存器

PORT_ISFR 端口中断状态标志 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	ISF															
访问	R/W1C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	ISF															
访问	R/W1C															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31：0] ISF	**中断状态标志位** 0: 没有产生中断 1: 已经产生中断。如果该引脚被配置为 DMA 请求，当前的 DMA 请求传输完成后，该状态标志位会被自动清除。当该引脚配置为中断，状态标志位会一直保持直到写“1”清除。 寄存器的每个 32 个 bits，分别为同一 PORT 的 32 个 pin 的中断状态，如 bit0 为 PORTx_PIN0 的中断状态。

19.6.3 端口数字滤波使能寄存器(PORT_DFER)

表 19-7 PORT_DFER 寄存器

PORT_DFER 端口数字滤波使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DFE															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DFE															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31：0] DFE	**数字滤波使能** 0: 禁用数字滤波 1: 使能数字滤波

© 2013 - 2024 杰发科技有限公司

字段	说明
	寄存器的每个 bit 控制位，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的数字滤波使能。数字滤波配置对于该引脚上所有的数字复用功能都有效。

19.6.4 端口数字滤波时钟寄存器(PORT_DFCR)

表 19-8 PORT_DFCR 寄存器

PORT_DFCR 端口数字滤波时钟寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																CS
访问																RW
复位值																0

字段	说明
0 CS	**数字滤波时钟源选择** 0：BUS clock 1：LSI 128K clock 数字滤波时钟源必须在禁用所有引脚的数字滤波功能后，才能修改。数字滤波配置对于该引脚上所有的数字复用功能都有效。

19.6.5 端口数字滤波宽度寄存器(PORT_DFWR)

表 19-9 PORT_DFWR 寄存器

PORT_DFWR 端口数字滤波宽度寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												DFW				
访问												RW				
复位值												0	0	0	0	0

© 2013 - 2024 杰发科技有限公司

字段	说明
[4：0] DFW	**数字滤波宽度** 数字滤波总长度 = DFW * 数字滤波时钟源周期。当脉冲宽度小于或等于数字滤波总长度时，脉冲会被过滤。当脉冲宽度大于数字滤波总长度时，脉冲可以通过。 数字滤波时长度必须在禁用所有引脚的数字滤波功能后，才能修改。数字滤波配置对于引脚上所有的数字复用功能都有效。

19.6.6 端口输出数据寄存器(GPIO_PODR)

表 19-10 GPIO_PODR 寄存器

GPIO_PODR **端口输出数据寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	POD															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	POD															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31: 0] POD	端口数据输出 0：对应的引脚输出低电平 1：对应的引脚输出高电平 寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输出电平。对于没有引出来的引脚，读可能会返回不确定的值。

© 2013 - 2024 杰发科技有限公司

19.6.7 端口输出置位寄存器(GPIO_PSOR)

表 19-11 GPIO_PSOR 寄存器

GPIO_PSOR **端口输出置位寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PSO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PSO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31: 0] PSO	端口输出置位 0：对应的 PODRx 的 bit 没有改变 1：对应的 PODRx 的 bit 设置为逻辑“1” 寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输出电平是否设置为逻辑“1”

19.6.8 端口输出复位寄存器(GPIO_PROR)

表 19-12 GPIO_PROR 寄存器

GPIO_PROR **端口输出复位寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PRO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PRO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31: 0] PRO	端口输出复位 0：对应的 PODRx 的 bit 没有改变 1：对应的 PODRx 的 bit 设置为逻辑“0”

© 2013 - 2024 杰发科技有限公司

字段	说明
	寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输出电平是否设置为逻辑“0”

19.6.9 端口输出翻转寄存器(GPIO_PIOR)

表 19-13 GPIO_PIOR 寄存器

GPIO_PIOR **端口输出翻转寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PIO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PIO															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31: 0] PIO	端口输出翻转 0：对应的 PODRx 的 bit 没有改变 1：对应的 PODRx 的 bit 逻辑翻转 寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输出电平是否设置逻辑翻转

19.6.10 端口输入数据寄存器(GPIO_PIDR)

表 19-14 GPIO_PIDR 寄存器

GPIO_PIDR **端口输入数据寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PID															
访问	R															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PID															
访问	R															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
[31: 0] PID	端口输入数据 0：对应的引脚逻辑电平为逻辑“0”，或该引脚配置为非数字功能， 或该引脚没有引出来 1：对应的引脚逻辑电平为逻辑“1” 寄存器的每个 bit，分别对应同一 PORT 的 32 个 pin，如 bit0 对应 PORTx_PIN0 的引脚逻辑电平。

19.6.11 端口输出使能寄存器(GPIO_POER)

表 19-15 GPIO_POER 寄存器

GPIO_POER　　端口输出使能寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	POE															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	POE															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[31: 0] POE	端口输出使能 0：GPIO 复用功能时，对应的引脚设置为通用输入。如果 GPIO_PIER 对应 bit 也设置为 0, 引脚会被设置为 high-Z 1：GPIO 复用功能时，对应的引脚设置为通用输出。 寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输出使能。

© 2013 - 2024 杰发科技有限公司

19.6.12 端口输入使能寄存器(GPIO_PIER)

表 19-16 GPIO_PIER 寄存器

GPIO_PIER 端口输入使能寄存器 复位值：**0xFFFFFFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PIE															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PIE															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
[31: 0] PIE	端口输入使能 0：对应的引脚设置为非通用的输入，对应的 PIDR 寄存器 bit 读返回 0. 1：对应的引脚设置为通用的输入，并且对于任意的数字复用功能都有效。 寄存器的每个 bit，分别控制同一 PORT 的 32 个 pin，如 bit0 控制 PORTx_PIN0 的输入使能。

© 2013 - 2024 杰发科技有限公司

20 直接存储器访问（DMA）

20.1 简介

直接存储器访问(Direct Memory Access, DMA) 控制器用于在外设和存储器之间，或存储器与存储器之间的高速数据传输，无需 CPU 干预， 数据可以通过 DMA 快速地移动，提升了微处理器的性能。

DMA 共有 16 个通用通道用于不同类型的外设， 支持 UART, I2C, SPI, ADC, EIO, PWM, PORT, CAN_RX。 还有一个轮询仲裁器来协调通道间的优先级。

20.2 特性

- 16 个独立的可配置通道
- 每个 DMA 通道都支持 UART, I2C, SPI, ADC, EIO, PWM, PORT, CAN_RX 的 DMA 请求，每个通道上都同样支持软件触发(存储器到存储器)。 该配置可通过软件完成
- 在同一个 DMA 模块上，多个请求间的优先级可以通过软件编程设置 (4 个等级分别为：很高，高，中等和低)，优先级设置相等时由硬件决定 (通道 0 优先级最高，通道 15 优先级最低，依次类推)
- 独立数据源和目标数据区的传输大小 (字节，半字，字)，源/目标地址必须按数据传输宽度对齐
- 支持循环的缓冲器管理
- 每个通道都支持 DMA 传输完成和 DMA 传输错误等状态标志位和中断，这些状态标志逻辑或成为一个单独的中断请求
- 存储器和存储器间的传输， 外设到 CRC 的传输
- 外设和存储器 、存储器和外设间的传输
- SRAM 和 APB 外设均可作为访问的源和目标
- 可编程的数据传输字节长度：最大为 32767 bytes

© 2013 - 2024 杰发科技有限公司

20.3 结构框图

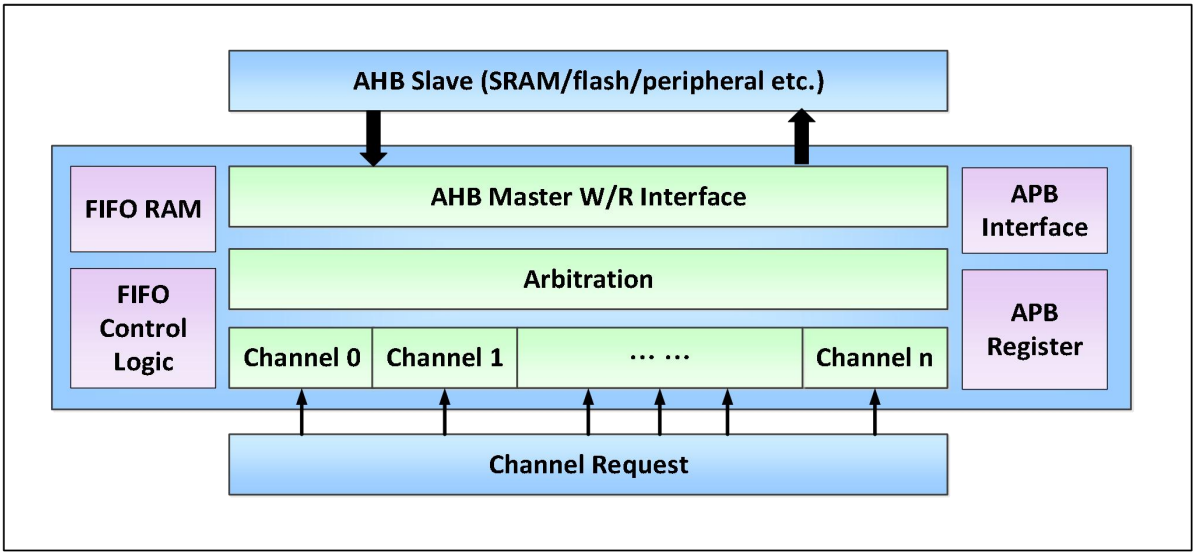


图 20-1 DMA 结构框图

20.4 功能描述

20.4.1 DMA 请求列表

表 20-1 DMA 请求列表

REQ_ID	外设	触发 DMA 搬运条件描述
0	保留	保留
1	UART0_RX	UART0 接收寄存器/FIFO 非空
2	UART0_TX	UART0 发送寄存器/FIFO 非满
3	UART1_RX	UART1 接收寄存器/FIFO 非空
4	UART1_TX	UART1 发送寄存器/FIFO 非满
5	UART2_RX	UART2 接收寄存器/FIFO 非空
6	UART2_TX	UART2 发送寄存器/FIFO 非满
7	UART3_RX	UART3 接收寄存器/FIFO 非空
8	UART3_TX	UART3 发送寄存器/FIFO 非满
13	EIO_SHIFTER0	EIO 移位寄存器 0 状态标志位置位
14	EIO_SHIFTER1	EIO 移位寄存器 1 状态标志位置位
15	EIO_SHIFTER2	EIO 移位寄存器 2 状态标志位置位
16	EIO_SHIFTER3	EIO 移位寄存器 3 状态标志位置位
17	PWM1_CH0	通道输入边沿捕获触发或输出计数达到阈值
18	PWM1_CH1	通道输入边沿捕获触发或输出计数达到阈值
19	PWM1_CH2	通道输入边沿捕获触发或输出计数达到阈值
20	PWM1_CH3	通道输入边沿捕获触发或输出计数达到阈值

© 2013 - 2024 杰发科技有限公司

REQ_ID	外设	触发 DMA 搬运条件描述
21	PWM1_CH4	通道输入边沿捕获触发或输出计数达到阈值
22	PWM1_CH5	通道输入边沿捕获触发或输出计数达到阈值
23	PWM1_CH6	通道输入边沿捕获触发或输出计数达到阈值
24	PWM1_CH7	通道输入边沿捕获触发或输出计数达到阈值
25	PWM1_UNDER_OR_OVER_FLOW	PWM1 任意通道发生上溢或下溢
26	PWM2_CH0	通道输入边沿捕获触发或输出计数达到阈值
27	PWM2_CH1	通道输入边沿捕获触发或输出计数达到阈值
28	PWM2_CH2	通道输入边沿捕获触发或输出计数达到阈值
29	PWM2_CH3	通道输入边沿捕获触发或输出计数达到阈值
30	PWM2_CH4	通道输入边沿捕获触发或输出计数达到阈值
31	PWM2_CH5	通道输入边沿捕获触发或输出计数达到阈值
32	PWM2_CH6	通道输入边沿捕获触发或输出计数达到阈值
33	PWM2_CH7	通道输入边沿捕获触发或输出计数达到阈值
34	PWM2_UNDER_OR_OVER_FLOW	PWM2 任意通道发生上溢或下溢
35	PWM0_OR_CH0_CH7	PWM0 任意通道输入边沿捕获触发或输出计数达到阈值
36	PWM0_UNDER_OR_OVER_FLOW	PWM0 任意通道发生上溢或下溢
37	PWM3_OR_CH0_CH7	PWM3 任意通道输入边沿捕获触发或输出计数达到阈值
38	PWM3_UNDER_OR_OVER_FLOW	PWM3 任意通道发生上溢或下溢
39	PWM4_OR_CH0_CH7	PWM4 任意通道输入边沿捕获触发或输出计数达到阈值
40	PWM4_UNDER_OR_OVER_FLOW	PWM4 任意通道发生上溢或下溢
41	PWM5_OR_CH0_CH7	PWM5 任意通道输入边沿捕获触发或输出计数达到阈值
42	PWM5_UNDER_OR_OVER_FLOW	PWM5 任意通道发生上溢或下溢
47	SPI0_RX	SPI0 接收寄存器/FIFO 非空
48	SPI0_TX	SPI0 发送寄存器/FIFO 非满
49	SPI1_RX	SPI1 接收寄存器/FIFO 非空
50	SPI1_TX	SPI1 发送寄存器/FIFO 非满
51	SPI2_RX	SPI2 接收寄存器/FIFO 非空
52	SPI2_TX	SPI2 发送寄存器/FIFO 非满
53	ADC0	ADC 转换完成
54	ADC1	ADC 转换完成
55	I2C0_RX	I2C0 接收寄存器/FIFO 非空
56	I2C0_TX	I2C0 发送寄存器/FIFO 非满
57	I2C1_RX	I2C1 接收寄存器/FIFO 非空
58	I2C1_TX	I2C1 发送寄存器/FIFO 非满
59	PORTA	PORTA 任意一个 IO 触发都有效
60	PORTB	PORTB 任意一个 IO 触发都有效
61	PORTC	PORTC 任意一个 IO 触发都有效
62	PORTD	PORTD 任意一个 IO 触发都有效

© 2013 - 2024 杰发科技有限公司

REQ_ID	外设	触发 DMA 搬运条件描述
63	PORTE	PORTE 任意一个 IO 触发都有效
64	CAN0_RX	CAN0 接收缓冲器非空
65	CAN1_RX	CAN1 接收缓冲器非空
66	CAN2_RX	CAN2 接收缓冲器非空
67	CAN3_RX	CAN3 接收缓冲器非空
72	ALWAYS_ENABLED	DMA 请求一直存在，直到 DMA 搬运完成。内存搬运到内存，需要配置为该 DMA 请求

20.4.2 仲裁器

仲裁器根据通道请求的优先级来管理通道请求，并启动外设/存储器访问序列。

每个通道的软件优先级可在通道配置寄存器 DMA_CONFIG[CHAN_PRIORITY]中配置，共有 4 个等级：

- 最高优先级
- 高优先级
- 中等优先级
- 低优先级

20.4.3 中断请求

DMA 共有 16 个 channel, 每个 channel 都有一个独立的中断向量。每个 DMA channel 都支持表 20-2 的中断请求源。

表 20-2 DMA 中断汇总

中断源	标志	本地使能	全局使能
完成中断	DMA_STATUS[FINISH]	DMA_INTEN[FINISH]	
目标总线错误	DMA_STATUS[DBE]	DMA_INTEN[TRANS_ERROR]	
源总线错误	DMA_STATUS[SBE]		
目标地址偏移配置错误	DMA_STATUS[DOE]		
源地址偏移配置错误	DMA_STATUS[SOE]		
目标地址配置错误	DMA_STATUS[DAE]		
源地址配置错误	DMA_STATUS[SAE]		

20.4.4 低功耗模式

DMA 模块不支持在 STOP1/STOP2/VLPS/Standby 下运行，因此在进入低功耗模式前，需要关闭 DMA（DMA_CHAN_ENABLE [CHAN_ENABLE]为 0）。DMA 模块不支持在低功耗模式下唤醒。

© 2013 - 2024 杰发科技有限公司

系统进入 STOP1/STOP2/VLPS/Standby 时，如果有一个 DMA channel 的 DMA_CHAN_ENABLE 寄存器的 CHAN_ENABLE 位为 1， 代表 DMA channel 还没完成工作，则不会回复休眠应答确认 ACK 信号，造成休眠超时。

20.4.5 调试模式

DMA_CHAN_ENABLE 寄存器中的 EDBG =1 时， 在 Debug Halt 状态，DMA channel 搬运暂停。若 EDBG =0，在 Debug Halt 状态，DMA 继续运行。

20.5 应用说明

说明：
用户使用 DMA 时，所有 DMA 搬运的 SRAM 数据都必须存放在 SRAM_U 区(0x20000000~0x2000EFFF)。

在用户配置通道使能寄存器 DMA_CHAN_ENABLE 之前，应编程除寄存器 DMA_CHAN_ENABLE 之外的所有寄存器。DMA 配置参数在 CHAN_ENABLE=1 生效。当 CHAN_ENABLE 为 1 时，请不要修改其它配置参数。

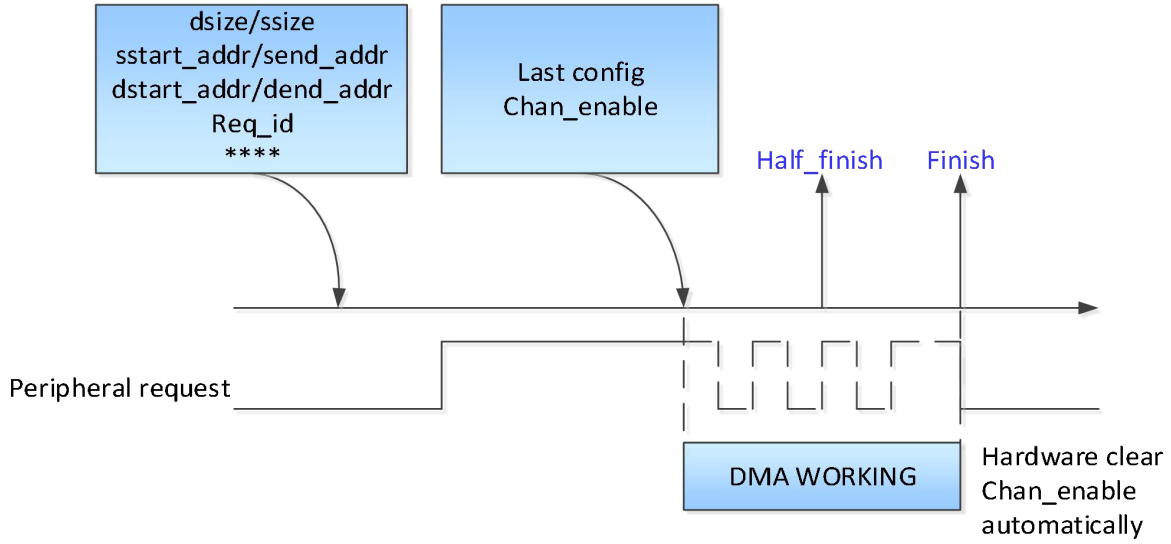


图 20-2 DMA 配置指南

© 2013 - 2024 杰发科技有限公司

20.5.1 软复位和硬复位

DMA 全局控制(DMA_TOP_RST)和每个 DMA 通道都支持设置软复位和硬复位。设置软复位后，在当前事务完成后 DMA 将复位，因此软复位不会导致任何总线挂起。相反，当设置硬复位时，DMA 将立即复位，因此总线可能由于未完成（并且永远不会完成）事务而停机。

全局软复位的机制为：当软件想要重启所有 DMA 通道或重新清除 DMA 的所有通道配置时，软件可以将通用 DMA 复位寄存器 DMA_TOP_RST[WARM_RST]设置为 1。然后 HW 将 WARM_RST 设置为 0 以完成全局软复位。

全局硬复位的机制为：当软件想要重启所有 DMA 通道或重新清除 DMA 的所有通道配置，且立即执行不需任何时间等待，用户可以将全局 DMA_TOP_RST[HARD_RST] 设置为 1 后，再软件设置为 0 以完成全局硬复位。请注意，这可能会破坏总线协议并导致系统挂起。

20.5.2 暂停和恢复

DMA 有暂停和恢复功能，机制如下：

1. 开始 DMA（程序根据需要设置后，然后设置 CHAN_ENABLE = 1）；
2. 暂停 DMA（设置 STOP = 1）；
3. 恢复 DMA（设置 STOP = 0）；
4. 等待 DMA 完成 （CHAN_ENABLE 将变为 0，中断标志将设置为 1）。

DMA 运行时，软件可以多次重复步骤 2 和 3。DMA 暂停不会立即生效，会等待当前事务全部完成后才会生效。

20.5.3 通道循环

循环模式用于处理循环缓冲区和连续的数据传输（如 ADC 扫描模式）。置位 DMA_CONFIG[CHAN_CIRCULAR]使能通道循环模式。当启动了循环模式，数据传输的数目变为 0 时，将会自动地被恢复成通道配置时设置的初值，DMA 操作将会继续进行。

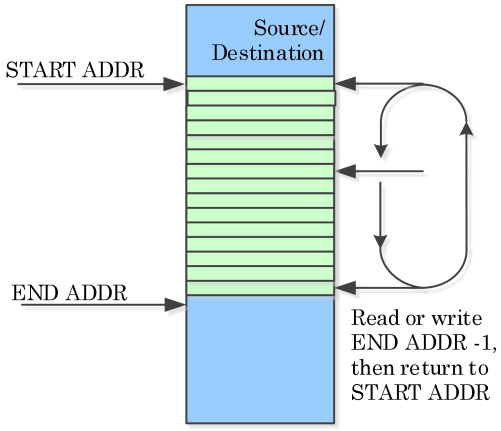


图 20-3 DMA 通道循环

© 2013 - 2024 杰发科技有限公司

20.5.4 I2C 使用 DMA

当达到为相应 DMA 编程的数据传输次数时，DMA 控制器将一个传输结束 EOT-1 信号发送到 I2C 模块，I2C 模块收到信号并决定是否发送 ACK/NACK 信号。

20.5.5 DMA 触发模式

DMA 通道 0~3 支持触发功能，触发源可以通过 CTU 模块进行配置。DMA 通道触发功能允许系统周期性地使用 DMA 传输，而不需要处理器参与，常常用于外设的发送。该触发功能通过抑制外设的 DMA 请求直到触发事件发生。详情参见图 20-4。

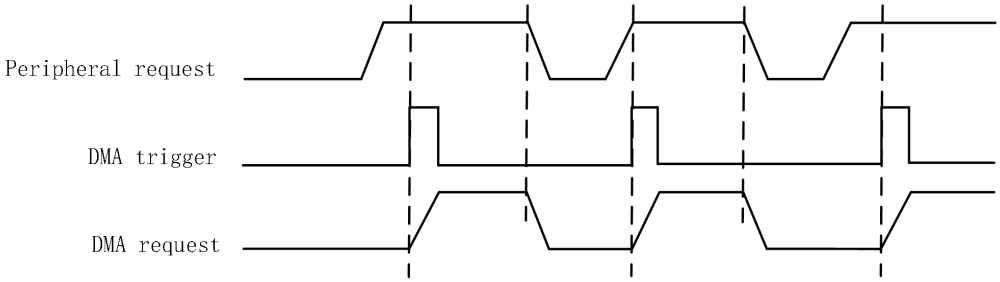


图 20-4 DMA 通道触发：正常操作

当 DMA 响应完请求后，需要等到下一次的外设请求和触发事件后才能再次响应。如果来自于 CTU 模块的触发事件已经产生，但外设的请求还没产生(如 UART_RX 的外设请求，由于 UART 此时没有接收到数据，没有产生请求)，该次触发会被忽略。详情参见图 20-5。

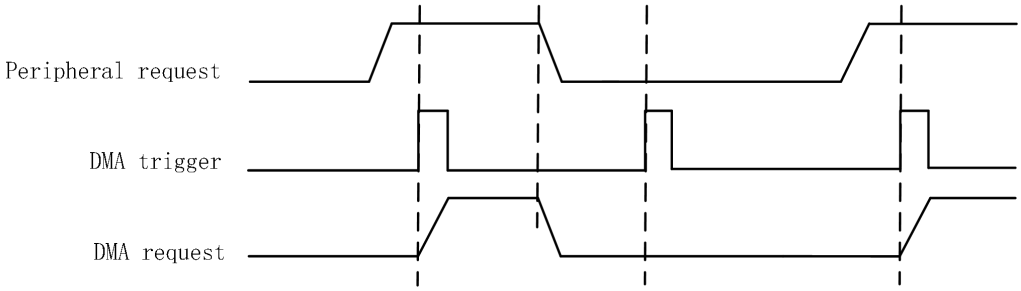


图 20-5 DMA 通道触发：忽略触发

20.5.6 配置指南

DMA 搬运时，方向都是从源地址搬运到目标地址的。

1. 配置该 DMA 通道源起始地址，结束地址：DMA_SSTART_ADDR，DMA_SEND_ADDR；
2. 配置该 DMA 通道目标起始地址，结束地址：DMA_DSTART_ADDR，DMA_DEND_ADDR；
3. 配置该 DMA 通道的中断使能：DMA_INTEN，可以配置为完成， 错误中断；

© 2013 - 2024 杰发科技有限公司

4. 配置该 DMA 通道的 DMA_CONFIG 寄存器和地址相关寄存器：配置源宽度的大小 SSIZE,目标宽度的大小 DSIZE, 通道 FIFO 控制，通道软件优先级 CHAN_PRIORITY

   ① **如果需要配置 DMA 为 MEM 到 PERIPH 传输：**

   配置源地址偏移 DMA_ADDR_OFFSET[SOFFSET]（必须为 SSIZE 的整数倍），目标地址偏移 DMA_ADDR_OFFSET[DOFFSET]=0（外设地址固定）

   ② **如果需要配置 DMA 为 PERIPH 到 MEM 传输：**

   配置源地址偏移 DMA_ADDR_OFFSET[SOFFSET]=0（外设地址固定），配置源地址偏移 DMA_ADDR_OFFSET[DOFFSET]（必须为 DSIZE 的整数倍）

   ③ **如果需要配置 DMA 为 MEM 到 MEM 传输：**

   配置源地址偏移 DMA_ADDR_OFFSET[SOFFSET]（必须为 SSIZE 的整数倍），配置源地址偏移 DMA_ADDR_OFFSET[DOFFSET]（必须为 DSIZE 的整数倍）。

5. 配置 DMA 通道 REQ_ID，即 DMA_MUX_CFG[REQ_ID]
6. 如果 DMA 通道需要配置为周期性触发模式，置位 DMA_MUX_CFG[TRIG_EN]。
7. 配置 DMA 通道需要搬运的总字节长度，即配置 DMA_CHAN_LENGTH 寄存器。
8. 使能该 DMA 通道，即置位 DMA_CHAN_ENABLE[CHAN_ENABLE]。

20.6 寄存器定义

表 20-3 DMA 寄存器映射

DMA0_BASE： 0x40012000

DMA0_Channelx 基地址： 0x40012040 + x*0x40

地址	名称	宽度	描述
DMA0_BASE +0x00	DMA_TOP_RST	32	通用 DMA 复位寄存器
DMA0_Channelx 基地址+0x00	DMA_STATUS	32	通道状态寄存器
DMA0_Channelx 基地址+0x04	DMA_INTEN	32	通道中断使能寄存器
DMA0_Channelx 基地址+0x08	DMA_RST	32	通道复位寄存器
DMA0_Channelx 基地址+0x0C	DMA_STOP	32	通道停止寄存器
DMA0_Channelx 基地址+0x10	DMA_CONFIG	32	通道配置寄存器
DMA0_Channelx 基地址+0x14	DMA_CHAN_LENGTH	32	通道长度寄存器
DMA0_Channelx 基地址+0x18	DMA_SSTART_ADDR	32	通道源起始地址寄存器
DMA0_Channelx 基地址+0x1C	DMA_SEND_ADDR	32	通道源结束地址寄存器
DMA0_Channelx 基地址+0x20	DMA_DSTART_ADDR	32	通道目标起始地址寄存器
DMA0_Channelx 基地址+0x24	DMA_CHAN_ENABLE	32	通道使能寄存器
DMA0_Channelx 基地址+0x28	DMA_DATA_TRANS_NUM	32	通道已传输数目寄存器
DMA0_Channelx 基地址+0x2C	DMA_INTER_FIFO_DATA_LEFT_NUM	32	通道 FIFO 数据剩余数目寄存器
DMA0_Channelx 基地址+0x30	DMA_DEND_ADDR	32	通道目标结束地址寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
DMA0_Channelx 基地址+0x34	DMA_ADDR_OFFSET	32	通道地址偏移寄存器
DMA0_Channelx 基地址+0x38	DMA_MUX_CFG	32	通道复用器配置寄存器

说明：
上表中(x 为 0~15， 代表 DMA 16 个通道)

20.6.1 通用 DMA 复位寄存器(DMA_TOP_RST)

表 20-4 DMA_TOP_RST 寄存器

DMA_TOP_RST **通用 DMA 复位寄存器** **复位值：0x00000004**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															HARD_RST	WARM_RST
访问															RW	RW
复位值															0	0

字段	说明
1 HARD_RST	**通用 DMA 硬复位(不论当前事务如何，都复位)** 0： 禁用 1： 使能 软件设置 HARD_RST 为 1，然后软件设置 HARD_RST 回 0，则该复位机制完成。HARD_RST 会使所有 DMA Channel 寄存器恢复默认值。
0 WARM_RST	**通用 DMA 软复位 (在当前事务完成后复位)** 0： 禁用 1： 使能 软件设置 WARM_RST 为 1 ，然后硬件清零 WARM_RST 位回 0，则该复位机制完成。WARM_RST 会使所有 DMA Channel 寄存器恢复默认值。

© 2013 - 2024 杰发科技有限公司

20.6.2 通道状态寄存器(DMA_STATUS)

表 20-5 DMA_STATUS 寄存器

DMA_STATUS 通道状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								CLE	SAE	DAE	SOE	DOE	SBE	DBE		FINISH
访问								R/W 0C	R/W 0C	R/W 0C	R/W 0C	R/W 0C	R/W 0C	R/W 0C		R/W 0C
复位值								0	0	0	0	0	0	0		0

字段	说明
8 CLE	**传输长度错误标志** 0： 传输长度错误没有发生 1： 传输长度错误发生，即 Length 没有按 SIZE 对齐 对该位置“0”将会清除此状态。
7 SAE	**源地址错误标志** 0： 源地址配置错误没有发生 1： 源地址配置错误发生，即 SADDR 没有按 SSIZE 对齐 对该位置“0”将会清除此状态。
6 DAE	**目标地址错误标志** 0： 目标地址配置错误没有发生 1： 目标地址配置错误发生，即 DADDR 没有按 DSIZE 对齐 对该位置“0”将会清除此状态。
5 SOE	**源地址偏移错误标志** 0： 源地址偏移配置错误没有发生 1： 源地址偏移配置错误发生，即 SOFF 不是 SSIZE 的整数倍 对该位置“0”将会清除此状态。
4 DOE	**目标地址偏移错误标志** 0： 目标地址偏移配置错误没有发生 1： 目标地址偏移配置错误发生，即 DOFF 不是 DSIZE 的整数倍

© 2013 - 2024 杰发科技有限公司

字段	说明
	对该位置“0”将会清除此状态。
3 SBE	**源总线错误标志** 0： 源总线错误没有发生 1： 源总线错误发生 指示当通道 n 读时，是否出现错误标志。 对该位置“0”将会清除此状态。
2 DBE	**目标总线错误标志** 0： 目标总线错误没有发生 1： 目标总线错误发生 指示当通道 n 写时，是否出现错误标志。 对该位置“0”将会清除此状态。
0 FINISH	**完成标志** 0： 数据传输未完成 1： 数据传输完成 指示当通道 n 读或写时，是否完成 DMA 通道数据传输。对该位置“0”将会清除此状态。

20.6.3 通道中断使能寄存器(DMA_INTEN)

表 20-6 DMA_INTEN 寄存器

DMA_INTEN **通道中断使能寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														TRANS_ERROR		FINISH
访问														RW		RW
复位值														0		0

字段	说明
2 TRANS_ERROR	**TRANS_ERROR 中断使能** 0： 中断禁用 1： 中断使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	此位由软件置位和清零
0 FINISH	**FINISH 中断使能** 0： 中断禁用 1： 中断使能 此位由软件置位和清零

20.6.4 通道复位寄存器(DMA_RST)

表 20-7 DMA_RST 寄存器

DMA_RST **通道复位寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														FLUSH	HARD_RST	WARM_RST
访问														RW	RW	RW
复位值														0	0	0

字段	说明
2 FLUSH	**DMA 通道刷新** 0： 禁用 1： 使能 设置 FLUSH = 1 将停止 DMA 并允许 DMA 将其内部缓冲区残留数据刷新至存储器。刷新完成后，DMA 将通道使能设置为 0 并停止 DMA。软件设置 FLUSH = 1 并等待 HW 清除为 0。
1 HARD_RST	**DMA 通道硬复位 (不论当前事务如何，都复位)** 0： 禁用 1： 使能 软件设置 ' HARD_RST ' 为 1，然后软件将 ' HARD_RST ' 设置回 0， 硬复位机制完成。
0 WARM_RST	**DMA 通道软复位(在当前事务后复位)** 0： 禁用

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 使能 软件设置 ' WARM_RST ' 为 1，然后硬件自动将 ' WARM_RST ' 设置回 0， 软复位机制完成

20.6.5 通道停止寄存器(DMA_STOP)

表 20- 8 DMA_STOP 寄存器

DMA_STOP **通道停止寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																STOP
访问																RW
复位值																0

字段	说明
0 STOP	**DMA 通道暂停 (在当前事务后暂停)** 0： 没有停止通道传输 1： 暂停通道传输 软件设置 STOP 为 1，当前事务后处理完成后，传输暂停。软件设置 STOP 为 0，通道继续数据传输，DMA 从暂停状态恢复为继续运行状态。 在暂停状态下，MEM 到 MEM 或 MEM 到外设是不会丢数据的。外设到 MEM 可能会丢数据，取决于暂停之后，外设接收数据 FIFO 是否有溢出。

© 2013 - 2024 杰发科技有限公司

20.6.6 通道配置寄存器(DMA_CONFIG)

表 20-9 DMA_CONFIG 寄存器

DMA_CONFIG **通道配置寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CHAN_FIFO	CHANE_CIRCULAR	DSIZE		SSZIE		CHAN_PRIORITY	
访问									RW	RW	RW		RW		RW	
复位值									0	0	0	0	0	0	0	0

字段	说明
7 CHAN_FIFO	**FIFO 加速使能** 0：不使能（M2P 或者 P2M 模式， 必须不使能） 1：使能（M2M 模式， 可选是否使能）
6 CHAN_CIRCULAR	**循环模式** 0： 循环模式禁用 1： 循环模式使能
5：4 DSIZE	**目标数据宽度** 00b： 8 位 01b：16 位 10b：32 位 11b： 保留
3：2 SSIZE	**源数据宽度** 00b： 8 位 01b：16 位 10b：32 位 11b： 保留
1：0 CHAN_PRIORITY	**通道优先级** 00b： 低 01b： 中等 10b： 高

© 2013 - 2024 杰发科技有限公司

字段	说明
	11b： 很高

20.6.7 通道长度寄存器(DMA_CHAN_LENGTH)

表 20-10 DMA_CHAN_LENGTH 寄存器

DMA_CHAN_LENGTH 通道长度寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CHAN_LENGTH															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15：0 CHAN_LENGTH	**DMA 通道传送长度** DMA 通道需要搬运的字节总长度： 1~32767 bytes， 其中 bit 15 应为 0 **注意：I2C 使用 DMA 传送时，需要配置 CHAN_LENGTH >= 2。**

20.6.8 通道源起始地址寄存器(DMA_SSTART_ADDR)

表 20-11 DMA_SSTART_ADDR 寄存器

DMA_SSTART_ADDR 通道源起始地址寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SSTART_ADDR[31：16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SSTART_ADDR [15：0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 SSTART_ADDR	**源起始地址**

© 2013 - 2024 杰发科技有限公司

字段	说明
	地址指向源数据的起始位置

20.6.9 通道源结束地址寄存器(DMA_SEND_ADDR)

表 20-12 DMA_SEND_ADDR 寄存器

DMA_SEND_ADDR 通道源结束地址寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SEND_ADDR[31：16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SEND_ADDR [15：0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 SEND_ADDR	**源结束地址** DMA 主机搬运完 SEND_ADDR-0x01 地址的数据后，回到 SSTART_ADDR 地址继续搬运。

20.6.10 通道目标起始地址寄存器(DMA_DSTART_ADDR)

表 20-13 DMA_DSTART_ADDR 寄存器

DMA_DSTART_ADDR 通道目标起始地址寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DSTART_ADDR[31：16]															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DSTART _ADDR [15：0]															
访问	RW															
复位值	0															

字段	说明
31：0 DSTART_ADDR	**目标起始地址** 地址指向目标数据的起始位置

© 2013 - 2024 杰发科技有限公司

20.6.11 通道使能寄存器(DMA_CHAN_ENABLE)

表 20-14 DMA_CHAN_ENABLE 寄存器

DMA_CHAN_ENABLE 通道使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															EDBG	CHAN_ENABLE
访问															RW	RW
复位值															0	0

字段	说明
1 EDBG	**调试模式使能** 0： 在调试模式下，DMA 通道继续运行 1： 在调试模式下，当 CPU 停止时，DMA 通道暂停
0 CHAN_ENABLE	**通道使能** 0： 禁用通道 1： 使能通道 在非通道循环模式下，软件设置 CHAN_ENABLE 为 1, 当传输完成时， 硬件将 CHAN_ENABLE 清零。 在通道循环模式下，软件设置 CHAN_ENABLE 为 1, 当传输完成时， 硬件使用相同的配置重新开始新的传输。

© 2013 - 2024 杰发科技有限公司

20.6.12 通道数据传输数目寄存器 (DMA_DATA_TRANS_NUM)

表 20-15 DMA_DATA_TRANS_NUM 寄存器

DMA_DATA_TRANS_NUM 通道数据传输数目寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA_TRANS_NUM															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15：0 DATA_TRANS_NUM	**DATA_TRANS_NUM** 标识已传输完成的字节长度

20.6.13 通道 FIFO 数据剩余数目寄存器 (DMA_INTER_FIFO_DATA_LEFT_NUM)

表 20-16 DMA_INTER_FIFO_DATA_LEFT_NUM 寄存器

DMA_INTER_FIFO_DATA_LEFT_NUM 通道 FIFO 数据剩余数目寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称											INTER_FIFO_DATA_LEFT_NUM					
访问											RO					
复位值											0	0	0	0	0	0

字段	说明
5：0 INTER_FIFO_DATA_LEFT_NUM	**INTER_FIFO_DATA_LEFT_NUM** 标识 DMA fifo 中剩余了的数据字节长度，通常与 FLUSH 一起使用。当超时且 INTER_FIFO_DATA_LEFT_NUM 不等于 0 时，软件可以启动刷新过程。

© 2013 - 2024 杰发科技有限公司

20.6.14 通道目标结束地址寄存器(DMA_DEND_ADDR)

表 20-17 DMA_DEND_ADDR 寄存器

DMA_DEND_ADDR 通道目标结束地址寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DEND_ADDR[31：16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DEND_ADDR [15：0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：0 DEND_ADDR	**目标结束地址** DMA 主机搬运完 DEND_ADDR-0x01 地址的数据后，回到 DSTART_ADDR 地址继续搬运。

20.6.15 通道地址偏移寄存器(DMA_ADDR_OFFSET)

表 20-18 DMA_ADDR_OFFSET 寄存器

DMA_ADDR_OFFSET 通道地址偏移寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DOFFSET															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SOFFSET															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31：16 DOFFSET	**目标地址偏移** 当前目标地址写完成后，下一个目标地址为：当前目标地址+目标地址偏移
15：0 SOFFSET	**源地址偏移** 当前源地址读完成后，下一个源地址为：当前源地址+源地址偏移

© 2013 - 2024 杰发科技有限公司

20.6.16 通道复用器配置寄存器(DMA_MUX_CFG)

表 20-19 DMA_MUX_CFG 寄存器

DMA_MUX_CFG 通道复用器配置寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									TRIG_EN	REQ_ID						
访问									RW	RW						
复位值									0	0	0	0	0	0	0	0

字段	说明
7 TRIG_EN	**触发使能** 0: 触发功能禁用。触发功能禁用，并且 CHAN_ENABLE 为 1， DMA 通道工作在正常的模式，只是由 DMA 请求源触发。 1: 触发功能使能。触发功能使能，并且 CHAN_ENABLE 为 1， DMA 通道工作在周期性的触发模式，只有当 DMA 外设请求和 DMA 触发源同时发生时，才会触发 DMA 搬运。
6：0 REQ_ID	**DMA 请求 ID** DMA 通道请求源设置， 具体请参考 20.4.1 DMA 请求列表 章节。

© 2013 - 2024 杰发科技有限公司

21 循环冗余校验模块（CRC）

21.1 简介

循环冗余校验模块（CRC）用于生成 16 或 32 位 CRC 校验码，对数据进行校验。CRC 模块提供 16 位或 32 位 CRC 标准所需的可编程多项式、写入种子和其他参数。

21.2 特性

- 16 / 32 位校验模式
- 可编程初始种子和多项式
- 输入按位 / 字节转置
- 输出按位 / 字节转置
- 结果按位取反
- DMA 支持

21.3 结构框图

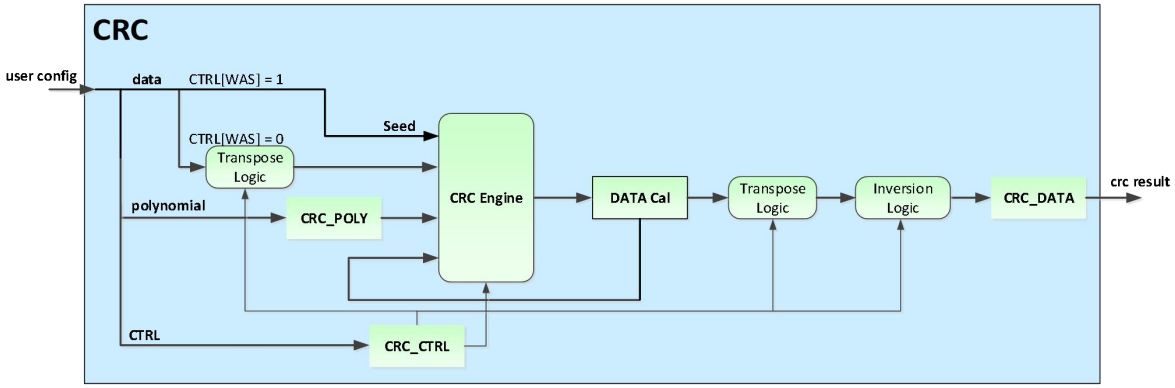


图 21-1 CRC 结构框图

21.4 功能描述

21.4.1 转置特性

转置功能默认不开启，CRC 按照正常的流程进行校验。

由于某些 CRC 标准要求对输入数据或最终校验结果进行转置，为了减少用户软件开销，CRC 模块具备硬件转置能力，用户软件可根据需要，选择配置转置功能。

© 2013 - 2024 杰发科技有限公司

输入数据转置使能后，写入的数据会先进行转置，后进行 CRC 计算。结果转置使能后，读出的数据是 CRC 校验结果转置后的数据。结果取反功能使能后，读出的数据是 CRC 校验结果转置并取反后的结果。

21.4.2 转置类型

CRC 模块提供了按位/字节翻转的多种转置功能类型，使用 CRC_CTRL[TOTW] 或 CRC_CTRL[TOTR]进行配置。

转置功能配置选项：

1. CRC_CTRL[TOTW] 或 CRC_CTRL[TOTR] 为 00

   不发生转置。

2. CRC_CTRL[TOTW] 或 CRC_CTRL[TOTR] 为 01

   字节中的位转置，字节不转置：reg[31：0] 变为 {reg[24：31], reg[16：23], reg[8：15], reg[0：7]}。

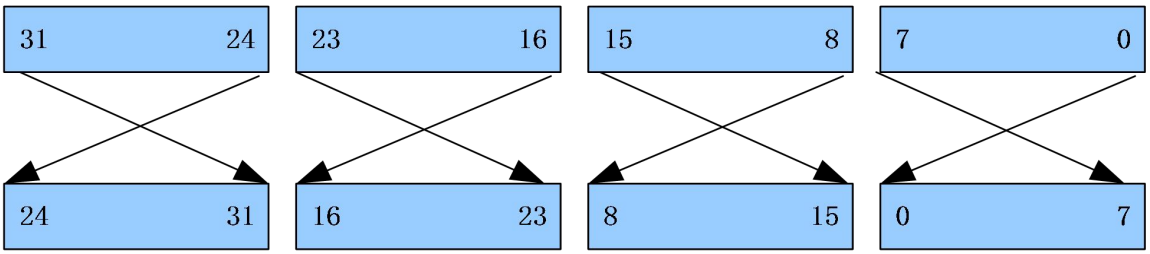


图 21-2 [TOTW] 或 [TOTR]为 01

3. CRC_CTRL[TOTW] 或 CRC_CTRL[TOTR] 为 10

   字节中的位和字节均转置：reg[31：0] 变为 {reg[0：7], reg[8：15],reg[16：23], reg[24：31]}.。

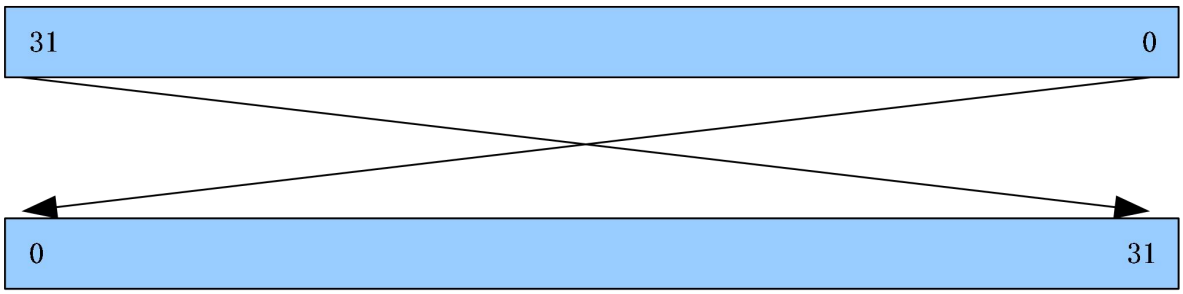


图 21-3 [TOTW] 或 [TOTR]为 10

4. CRC_CTRL[TOTW] 或 CRC_CTRL[TOTR] 为 11

   字节转置，但位不转置：reg[31：0] 变为 {reg[7：0], reg[15：8], reg[23：16], reg[31：24]}。

© 2013 - 2024 杰发科技有限公司

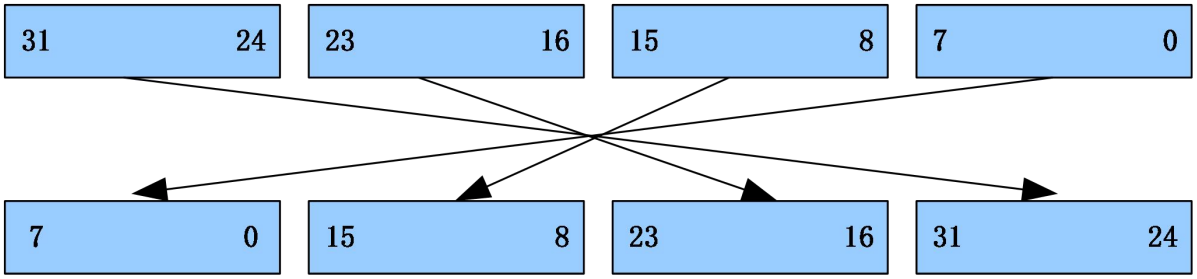


图 21-4 [TOTW] 或 [TOTR]为 11

说明：

当向 CRC_DATA 寄存器一次写入 1 个字节时，不会发生按字节转置；

当向 CRC_DATA 寄存器一次写入 2 个字节时，对写入的 2 个字节进行转置；

当向 CRC_DATA 寄存器一次写入 4 个字节时，对写入的 4 个字节进行转置。

21.4.3 结果反码

CRC_CTRL[FXOR]置位时，读取的校验结果会先进行取反。CRC_CTRL[FXOR]复位时，读取的校验结果不会取反，得到原值。

21.4.4 支持 DMA

其他外设寄存器或内存的值可以通过 DMA 搬运至 CRC_DATA 寄存器来计算校验值。

21.4.5 低功耗模式

在 STOP1/STOP2 模式下，CRC 模块不能工作，CRC 寄存器保持先前值，唤醒前后寄存器值保持不变。
在 VLPS 模式下，CRC 模块处于关电状态，系统唤醒后，CRC 寄存器恢复为默认值。

21.4.6 调试模式

CRC 模块没有调试模式配置，在进入调试模式时可正常运行。

© 2013 - 2024 杰发科技有限公司

21.5 应用说明

21.5.1 CRC 初始化

在对 CRC_POLY、CRC_DATA 寄存器进行操作前，首先必须对 CRC_CTRL 控制寄存器进行配置，选择使用的 CRC 模式 CRC16/32，CRC_CTRL[TCRC]选择 CRC 模式，CRC_CTRL[TOTW]配置写入数据转置类型，CRC_CTRL[TOTR]配置读取数据转置类型，CRC_CTRL[FXOR]使能读取数据取反，CRC_CTRL[WAS]选择对 CRC_DATA 寄存器写入的是种子还是校验数据。

21.5.2 CRC 多项式配置

此寄存器包含 CRC 校验时的多项式值。高半字段是 CRC 多项式的高 16 位，仅在 32 位 CRC 模式下有效，在 CRC16 模式下会忽略对高半字段的写操作。低半字段是 CRC 多项式的低 16 位，在 CRC16 / 32 模式下都会被使用。

21.5.3 CRC 校验

CRC_DATA 数据寄存器同时拥有种子、数据和校验值和这 3 种角色。

当 CRC_CTRL [WAS] 置位时，对数据寄存器进行的任何写操作都被视为种子值。

当 CRC_CTRL [WAS] 清零后， 对数据寄存器进行的任何写操作都被视为写入用于 CRC 计算的数据。在 16 位 CRC 模式下，不使用高半字段来编程种子值；在 32 位 CRC 模式下，所有字段都用于种子值的编程。可以一次写入字节、半字或字，以 MSB 在前的顺序进行校验。

在写入所有数据后， 可以从该数据寄存器中读取 CRC 结果。在 16 位 CRC 模式下，低半字段是 CRC 校验结果。在 32 位 CRC 模式下，所有字段为校验结果，对 CRC_DATA 寄存器的读操作都会返回当前的 CRC 计算结果。

21.5.4 编程指南

- 16 位 CRC：
  1. 清零 CRC_CTRL [TCRC] 使能 16 位 CRC 模式；
  2. 按 CRC 计算要求对转置相关功能进行编程；
  3. 将 16 位多项式写入 CRC_POLY 低半字段。CRC_POLY 高半字段在 16 位 CRC 模式下不可用；
  4. 置位 CRC_CTRL [WAS] 进行种子值的编程；
  5. 将 16 位种子写入 CRC_DATA[15：0]，CRC_DATA[31：16] 未使用；
  6. 复位 CRC_CTRL[WAS] 开始逐个写入校验数据；
  7. 将数据写入 CRC_DATA[31：0]，CRC 的中间计算结果都会保存在 CRC_DATA[15：0]；

© 2013 - 2024 杰发科技有限公司

8. 当所有数据都被写入后，从 CRC_DATA[15：0]中读取最终的 CRC 结果。

- 32 位 CRC：

1. 置位 CRC_CTRL [TCRC] 使能 32 位 CRC 模式；
2. 按 CRC 计算要求对转置相关功能进行编程；
3. 将 32 位多项式写入 CRC_POLY 多项式寄存器；
4. 置位 CRC_CTRL [WAS] 进行种子值的编程；
5. 将 32 位种子写入 CRC_DATA 寄存器；
6. 复位 CRC_CTRL [WAS] 开始逐个写入校验数据；
7. 将数据写入 CRC_DATA[31：0]，CRC 的中间计算结果都会保存在 CRC_DATA[31：0]；
8. 当所有数据都被写入后，从 CRC_DATA[31：0]中读取最终的 CRC 结果。

21.6 寄存器定义

CRC 寄存器只支持 32 位访问。

表 21-1 CRC 寄存器映射

CRC 基地址：0x40084000

地址	名称	宽度	描述
CRC 基地址+0x00	CRC_DATA	32	数据寄存器
CRC 基地址+0x04	CRC_POLY	32	多项式寄存器
CRC 基地址+0x08	CRC_CTRL	32	控制寄存器

21.6.1 数据寄存器(CRC_DATA)

表 21-2 CRC_DATA 寄存器

CRC_DATA **数据寄存器** **复位值：0xFFFFFFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DATA[31: 24]								DATA[23: 16]							
访问	RW								RW							
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA[15: 8]								DATA[7: 0]							
访问	RW								RW							
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
31: 24	**数据字节 3**

© 2013 - 2024 杰发科技有限公司

字段	说明
DATA	在 16 位 CRC 模式 (CRC_CTRL[TCRC] 为 0)下，不用于种子值编程。 在 32 位 CRC 模式 (CRC_CTRL[TCRC] 为 1)下，当 CRC_CTRL[WAS] 为 1 时，写入该字段的值是种子值的一部分。当 CRC_CTRL[WAS] 为 0 时，写入该字段的数据参与 16 或 32 位 CRC 校验，生成 CRC 校验和。
23: 16 DATA	**数据字节 2** 在 16 位 CRC 模式 (CRC_CTRL[TCRC] 为 0)下，不用于种子值编程。 在 32 位 CRC 模式 (CRC_CTRL[TCRC] 为 1)下，当 CRC_CTRL[WAS] 为 1 时，写入该字段的值是种子值的一部分。当 CRC_CTRL[WAS] 为 0 时，写入该字段的数据参与 16 或 32 位 CRC 校验，生成 CRC 校验和。
15: 8 DATA	**数据字节 1** 当 CRC_CTRL[WAS] 为 1 时，写入该字段的值是种子值的一部分。当 CRC_CTRL[WAS] 为 0 时，写入该字段的数据参与 16 或 32 位 CRC 校验，生成 CRC 校验和。
7: 0 DATA	**数据字节 0** 当 CRC_CTRL[WAS] 为 1 时，写入该字段的值是种子值的一部分。当 CRC_CTRL[WAS] 为 0 时，写入该字段的数据参与 16 或 32 位 CRC 校验，生成 CRC 校验和。

21.6.2 多项式寄存器(CRC_POLY)

表 21-3 CRC_POLY 寄存器

CRC_POLY **多项式寄存器** **复位值：0x00001021**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	POLY[31：16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	POLY[15：0]															
访问	RW															
复位值	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

字段	说明
31: 16 POLY	**多项式高半字** 该字段在 32 位 CRC 模式下可读写 (CRC_CTRL[TCRC] 为 1)。 该字段在 16 位 CRC 模式下不可写 (CRC_CTRL[TCRC] 为 0)。
15: 0 POLY	**多项式低半字** 该字段在 32 位与 16 位 CRC 模式下都可读写。

© 2013 - 2024 杰发科技有限公司

21.6.3 控制寄存器(CRC_CTRL)

表 21-4 CRC_CTRL 寄存器

CRC_CTRL 控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									TOTW		TOTR			FXOR	WAS	TCRC
访问									RW		RW			RW	RW	RW
复位值									0	0	0	0		0	0	0

字段	说明
7: 6 TOTW	**写入转置类型** 定义写入 CRC_DATA 寄存器的数据转置配置。 00b： 不转置 01b： 字节中的位转置，但字节不转置 10b： 字节中的位和字节均转置 11b： 仅字节转置，字节中的位不转置
5: 4 TOTR	**读取转置类型** 定义读取 CRC_DATA 寄存器时的数据转置配置。 00b： 不转置 01b： 字节中的位转置，但字节不转置 10b： 字节中的位和字节均转置 11b： 仅字节转置，字节中的位不转置
2 FXOR	**读取取反** 某些 CRC 协议要求最终校验和与 0xFFFFFFFF 或 0xFFFF 进行异或运算。 0： 读取时不执行 XOR 运算 1： 对读到的数据执行 XOR 运算
1 WAS	**写入 CRC_DATA 类型** 定义项 CRC_DATA 寄存器写入时的类型。 0：写入 CRC 数据寄存器的是数据 1：写入 CRC 数据寄存器的是种子值
0 TCRC	**CRC 校验类型**

© 2013 - 2024 杰发科技有限公司

字段	说明
	定义 CRC 校验的类型。 0：CRC16 1：CRC32

© 2013 - 2024 杰发科技有限公司

22 看门狗模块（WDG）

22.1 简介

看门狗（WDG）是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有被按时刷新，看门狗会产生系统复位。比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时得到执行，都会由于看门狗模块没有被刷新而产生系统复位，一般用于高安全性场合。

22.2 特性

- 4 种时钟源
- 可选固定 256 时钟预分频器
- 可编程 16-bit 超时时间
- 窗口模式
- 超时中断
- 调试模式
- 快速测试模式

22.3 结构框图

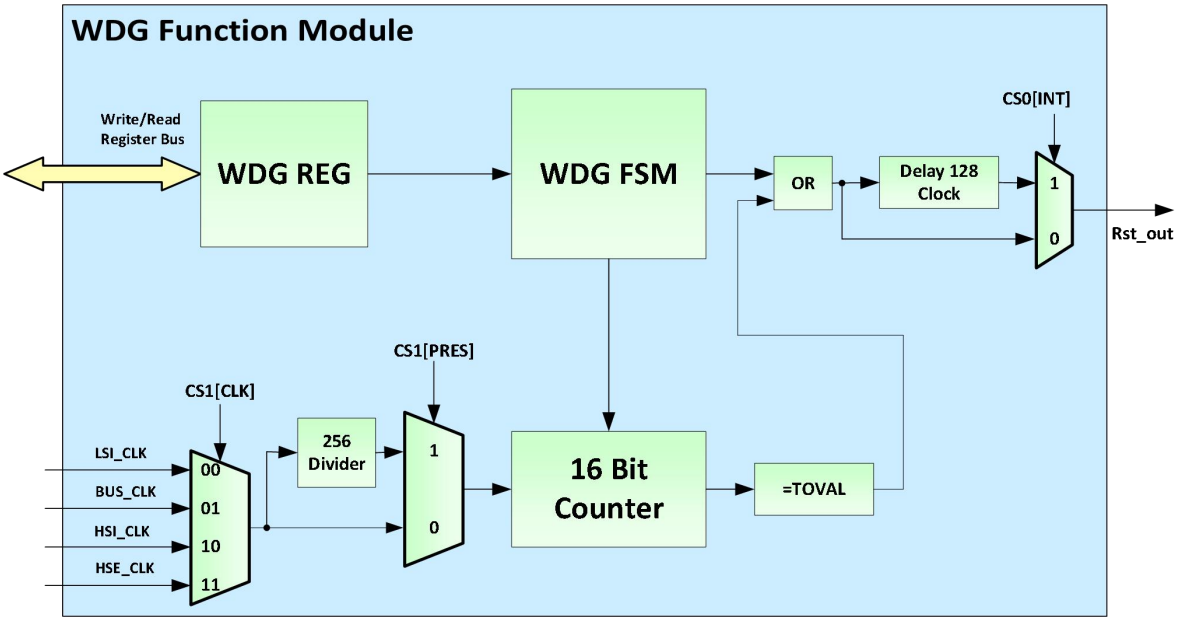


图 22-1 WDG 结构框图

© 2013 - 2024 杰发科技有限公司

22.4 功能描述

22.4.1 基本看门狗

看门狗具有四种时钟源：总线时钟(BUS_CLK)、LSI 时钟(LSI_CLK)、HSI 时钟(HSI_CLK)、HSE 时钟(HSE_CLK)。看门狗定时器使用一个 16 位的向上计数器以及一个可选的固定 256 分频器。

看门狗使能后，开始计数，若计数到 TOVAL 值，会产生系统复位，在计数达到 TOVAL 值前，刷新看门狗会使计数器复位并重新开始计数。

22.4.2 看门狗默认超时行为

在默认情况下，看门狗超时后会产生系统复位，用户也可通过配置 RCM 模块，屏蔽看门狗产生的系统复位，更多细节请参考 9 复位（RCM）章节。
在 MCU 上电后，看门狗计数器默认使用 LSI128K_CLK 时钟进行计数，并在 0x5000 个时钟周期后超时。这将导致 MCU 在上电约 160ms 后发生复位，为了避免这种情况的产生，用户需确保在 MCU 上电约 160ms 内配置或刷新看门狗。

22.4.3 窗口看门狗

看门狗具有窗口模式，在该模式下：

1 在计数器达到窗口值 WIN 前刷新看门狗，会产生系统复位；

2 计数器达到 TOVAL 值前未刷新看门狗，会产生系统复位；

3 在计数器大于等于 WIN 值，小于 TOVAL 值之间刷新看门狗会使计数器复位并重新开始计数。

22.4.4 备份复位

当选择非总线时钟源作为计数器的参考时钟时，备份复位才可用。备份复位是一种安全保护功能，在看门狗主逻辑丢失其时钟（总线时钟）而无法再监控计数器时，独立生成复位。
当看门狗中断使能且看门狗时钟来自非总线时钟时，如果看门狗计数连续两次溢出（没有强制复位），那么备份复位功能生效并生成复位，超时时间约为两个超时周期；
当看门狗中断禁能且看门狗时钟来自非总线时钟时，看门狗计数器溢出后，产生复位。

22.4.5 快速测试模式

看门狗可以通过配置 WDG_CS0 [TST]，使看门狗进入快速测试模式，在短时间内对看门狗计数器中每个位进行安全测试，以确保没有一个看门狗计数器位被卡在 0。

在快速测试模式下，16 位计数器的低字节[7:0]和高字节[15:8]将独立计数，当低字节[7:0]等于 TOVAL[7:0]，高字节[15:8]开始计数，当高字节[15:8]等于 TOVAL[15:8]，将发生复位。

© 2013 - 2024 杰发科技有限公司

注意：

在任何情况下，如果 16 位计数器[15:0]的计数值大于等于 TOVAL[15:0]，将发生复位。

WDG_CS0[TST]只能被 POR 复位清除，建议测试完成后配置关闭快速测试模式，否则复位后 WDG 会以快速测试模式计数，导致默认复位时间变为 0.625ms。

22.4.6 中断请求

WDG 模块共有 1 个中断源。

表 22-1 WDG 中断汇总

中断源	标志	本地使能	全局使能
看门狗计数器超时	WDG_CS1[FLG]	WDG_CS0[INT]	

22.4.7 低功耗模式

软件可以通过配置 WDG_CS0[STOP]，来控制看门狗在 STOP1、VLPS 模式下是否继续计数。

如果配置看门狗在低功耗模式下停止计数， 当 MCU 被唤醒后，看门狗计数器将继续计数。

MCU 在 STOP2 模式下，选择任意时钟源看门狗都可以正常计数。

MCU 在 STOP1 模式下，选择 LSI/HSI/HSE 作为时钟源，并且配置 WDG_CS0[STOP]为使能，看门狗将正常计数。

MCU 在 VLPS 模式下，必须选择 LSI 作为时钟源，并且配置 WDG_CS0[STOP]为使能，看门狗才可以正常计数。

表 22-2 低功耗模式 WDG 计数行为

模式 / 时钟源	STOP2	STOP1	VLPS
BUS	run	pause	pause
LSI	run	CS0[STOP] = 1, run CS0[STOP] = 0，pause	CS0[STOP] = 1, run CS0[STOP] = 0，pause
HSI	run	CS0[STOP] = 1, run CS0[STOP] = 0, pause	CS0[STOP] = 1, pause CS0[STOP] = 0，pause
HSE	run	CS0[STOP] = 1, run CS0[STOP] = 0，pause	CS0[STOP] = 1, pause CS0[STOP] = 0，pause

说明：

run 表示看门狗在低功耗模式下可以计数，pause 表示看门狗在低功耗模式下不能计数。

© 2013 - 2024 杰发科技有限公司

如果看门狗中断被使能，看门狗在低功耗模式下运行超时后，会产生中断将 MCU 唤醒，延时 128 个总线时钟后，产生系统复位。

表 22-3 看门狗低功耗模式下唤醒源

模式	唤醒源
STOP1	看门狗超时中断
STOP2	看门狗超时中断
VLPS	看门狗超时中断

22.4.8 调试模式

看门狗可以通过配置 WDG_CS0[DBG]来控制看门狗计数器在调试模式下的行为：

- 当 WDG_CS0[DBG] = 0 时，进入调试模式，看门狗会暂停计数，退出调试模式后看门狗会恢复原本的计数状态；
- 当 WDG_CS0[DBG] = 1 时，进入调试模式，单步或者断点都会暂停计数，全速运行状态下维持计数。

22.5 应用说明

22.5.1 配置看门狗

对看门狗的所有寄存器进行配置的条件是更新位 WDG_CS0[UPDATE]为 1 且看门狗解锁。解锁后，只能在 128 个总线时钟内对看门狗的任意寄存器进行配置，然后所有寄存器自动上锁，再次配置需要再次满足上述两个条件。

一旦用户将 WDG_CS0 [UPDATE]配置为 0 后，除非强制复位，否则无法修改看门狗寄存器配置。在 WDG_CS0 [UPDATE]为 0 的情况下解锁会引起系统复位。

解锁序列：

1. 向 WDG_CNT 寄存器先后写入 0xE064D987、0x868A8478，写入的值不正确或顺序颠倒会引起系统复位；
2. 写入两个解锁序列的间隔时间必须在 128 个总线时钟内，否则会引起系统复位。

22.5.2 刷新看门狗

在基本看门狗与窗口看门狗模式下，为了保证看门狗不复位系统，需要软件在规定时间内刷新看门狗。看门狗刷新后，看门狗计数器重新从 0 开始计数，软件需要再次刷新看门狗。这种机制使得软件必须定时刷新看门狗，在一定程度上反映了程序的正常运行，当程序意外跑飞后，看门狗会复位系统。

刷新序列：

© 2013 - 2024 杰发科技有限公司

1. 向 WDG_CNT 寄存器先后写入 0x7908AD15、0x5AD5A879，写入的值不对或顺序颠倒会引起系统复位；
2. 写入两个刷新序列的间隔时间必须在 128 个总线时钟内，否则会引起系统复位。

22.5.3 看门狗中断

看门狗具有中断功能，中断使能后，若看门狗计数器超时，看门狗不会立即复位系统，而是延迟 128 个总线时钟后复位系统。这 128 个总线时钟是留给中断服务程序响应的时间，用户软件可以在中断程序中进行简单的程序处理，但不建议在此时刷新看门狗。

22.5.4 编程指南

WDG 使用配置如下：

1. 向 WDG_CNT 寄存器先后写入解锁序列 0xE064D987、0x868A8478；
2. 配置 WDG_TOVAL 寄存器，设置超时值；
3. 配置 WDG_WIN 寄存器，设置窗口值；
4. 配置 WDG_CS1 寄存器；

注意：
寄存器 WDG_CS0[UPDATE]为 0 时，不允许对 WDG 寄存器进行配置，写入解锁序列后，必须保证在 128 个总线时钟内对寄存器进行配置。

5. 配置 WDG_CS0 寄存器，WDG 开始计数。

22.6 寄存器定义

WDG 寄存器只支持 32 位访问。
WDG 寄存器在写入前需要输入解锁序列 0xE064D987、0x868A8478。
当 WDG 寄存器 WDG_CS0[UPDATE]=0 时，WDG 寄存器被 Lock，不允许更新。

表 22-4 WDG 寄存器映射

WDG 基地址：0x4000B000

地址	名称	宽度	描述
WDG 基地址+0x00	WDG_CS0	32	配置寄存器 0
WDG 基地址+0x04	WDG_CS1	32	配置寄存器 1

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
WDG 基地址+0x08	WDG_CNT	32	计数器寄存器
WDG 基地址+0x0C	WDG_TOVAL	32	超时值寄存器
WDG 基地址+0x10	WDG_WIN	32	窗口值寄存器

22.6.1 配置寄存器 0(WDG_CS0)

表 22- 5 WDG_CS0 寄存器

WDG_CS0 **配置寄存器 0** **复位值： 0x000000A0**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									EN	INT	UPDATE	TST		ULK	DBG	STOP
访问									RW	RW	RW	RW		RO	RW	RW
复位值									1	0	1	0	0	0	0	0

字段	说明
7 EN	**看门狗使能** 0：看门狗禁用 1： 看门狗使能 使能看门狗计数器开始计数。（默认状态下，看门狗是使能状态）
6 INT	**看门狗中断** 0：看门狗中断禁用，看门狗复位系统无延迟 1：看门狗中断使能，看门狗复位系统延迟 128 个总线时钟 在看门狗超时或非法写入时，若使能了看门狗中断，会触发看门狗中断，并在中断响应 128 个总线时钟后再发生系统复位。
5 UPDATE	**看门狗配置更新位** 0：不允许更新。完成初始配置后，除非强制复位，否则无法修改看门狗寄存器配置 1：允许更新。执行解锁写入序列后，软件可以在 128 个总线时钟内修改看门狗配置寄存器
4:3 TST	**快速测试模式** 10b：看门狗工作在快速测试模式 其余任何值表示看门狗工作在正常模式。 TST 对应 bit 只能被 POR 复位清除。
2	**看门狗解锁状态**

© 2013 - 2024 杰发科技有限公司

字段	说明
ULK	0：当前看门狗寄存器处于被锁状态 1：当前看门狗寄存器处于解锁状态 该位指示看门狗寄存器是否处于解锁状态。
1 DBG	**调试使能** 0：调试模式禁用 1：调试模式使能 调试模式相关说明参见 22.4.8 调试模式 章节。
0 STOP	**低功耗模式控制位** 0：看门狗禁用，在低功耗模式下（STOP1/VLPS） 1：看门狗使能，在低功耗模式下（STOP1/VLPS）

22.6.2 配置寄存器 1(WDG_CS1)

表 22- 6 WDG_CS1 寄存器

WDG_CS1 **配置寄存器 1** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									WIN	FLG		PRES			CLK	
访问									RW	R/W1C		RW			RW	
复位值									0	0		0			0	0

字段	说明
7 WIN	**看门狗窗口** 0： 窗口模式禁用 1 ： 窗口模式使能 使能看门狗窗口模式
6 FLG	**看门狗中断标志** 0： 未发生中断

© 2013 - 2024 杰发科技有限公司

字段	说明
	1 ：发生中断 该位指示中断产生，写入 1 将其清零
4 PRES	**看门狗预分频器** 0 ：256 预分频器禁用 1 ：256 预分频器使能 使能看门狗计数器 256 倍预分频
1: 0 CLK	**看门狗时钟** 00b：LSI 时钟 01b：总线时钟 10b：HSI 时钟 11b：HSE 时钟 选择看门狗计数器时钟源

22.6.3 计数器寄存器(WDG_CNT)

表 22-7 WDG_CNT 寄存器

WDG_CNT **计数器寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CNT[31:16]															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CNT[15:0]															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31: 0 CNT	**看门狗计数值** 看门狗计数器寄存器（仅低16位）指示当前看门狗计数器的数值，软件可在任意时刻对计数器寄存器进行读操作。软件无法直接更改计数器的值。 但对这个寄存器的写操作具有特殊功能：刷新序列和解锁序列。

© 2013 - 2024 杰发科技有限公司

22.6.4 超时值寄存器(WDG_TOVAL)

表 22-8 WDG_TOVAL 寄存器

WDG_TOVAL 超时值寄存器 复位值：**0x00005000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TOVAL[15: 0]															
访问	RW															
复位值	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15: 0 TOVAL	**看门狗超时值寄存器** 看门狗计数器与超时数值连续作比较。若计数器达到超时值，则看门狗复位系统。计时长度等于 WDG_TOVAL+1，默认值为 0x5000。

22.6.5 窗口寄存器(WDG_WIN)

表 22-9 WDG_WIN 寄存器

WDG_WIN 窗口值寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	WIN[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15: 0 WIN	**看门狗窗口寄存器** 窗口模式使能后（WDG_CS1[WIN]置位），WIN 值决定看门狗允许刷新的最早时间，早于窗口值刷新会引起系统复位。

© 2013 - 2024 杰发科技有限公司

23 外部看门狗（EWDG）

23.1 简介

外部看门狗（EWDG），用于监控外部器件或电路。相比较内部看门狗（用来产生系统级别的复位，包括 CPU 和外设），EWDG 不会产生系统复位，仅产生一个低电平输出，这个低电平可以用来关断外部电路，起到保护外部电路的作用。相比编程 GPIO 关断外部电路，EWDG 不会存在因中断抢占问题而导致的关断不及时，适用于高安全性场合。

23.2 特性

- 独立 LSI 时钟
- 可编程 8-bit 计数器宽度
- 可编程 8-bit 预分频器
- 窗口模式
- 中断

23.3 结构框图

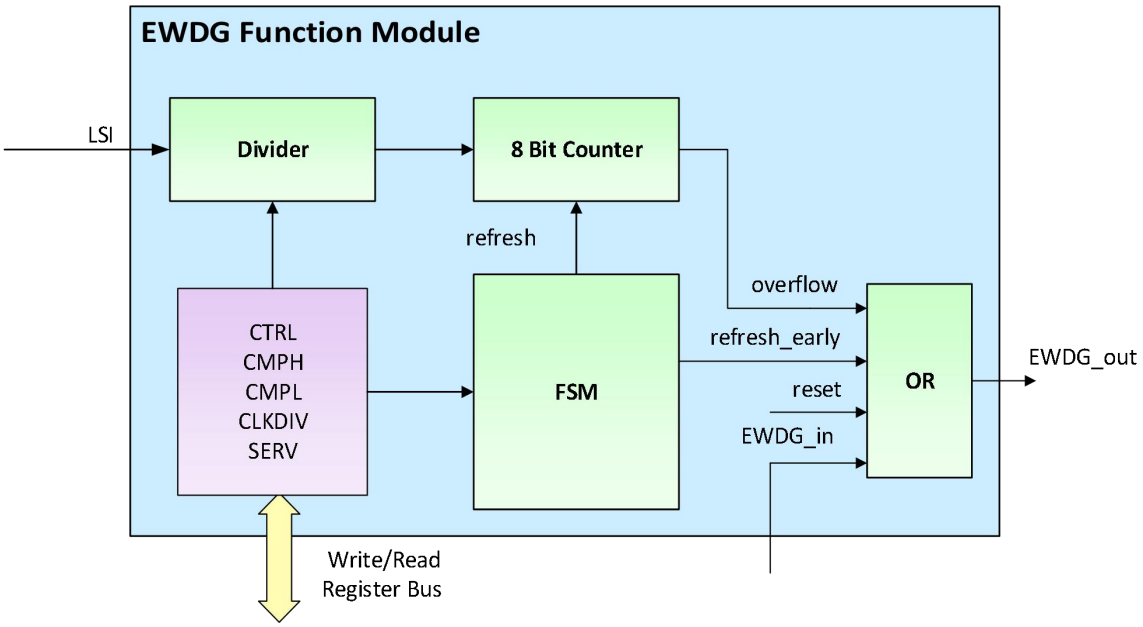


图 23-1 EWDG 结构框图

© 2013 - 2024 杰发科技有限公司

23.4 功能描述

23.4.1 基本功能

外部看门狗的时钟固定为芯片内部的 LSI 时钟，外部看门狗使能后，EWDG_out 引脚输出由低电平变为高电平，同时 8-bit 计数器从 0 开始计数，当满足以下三种情况之一时（其中的 CMPL 和 CMPH 即形成窗口区间），EWDG_out 引脚输出变为低电平，若中断使能，则产生中断：

1. 软件刷新出现在计数器小于 EWDG_CMPL 时

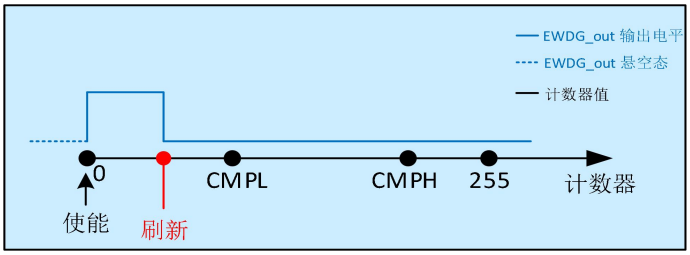


图 23-2 计数器小于 CMPL 时刷新

2. 计数器达到了 EWDG_CMPH

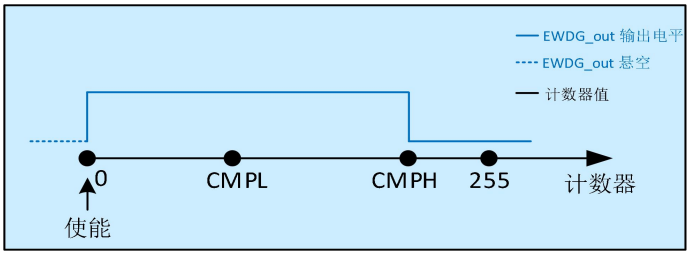


图 23-3 计数器达到了 CMPH

3. EWDG_in 满足电平条件时软件刷新

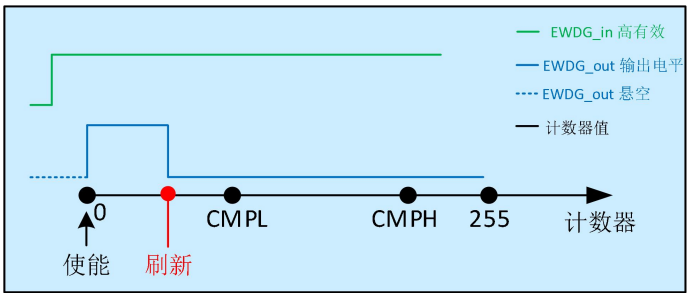


图 23-4 EWDG_in 有效时刷新

© 2013 - 2024 杰发科技有限公司

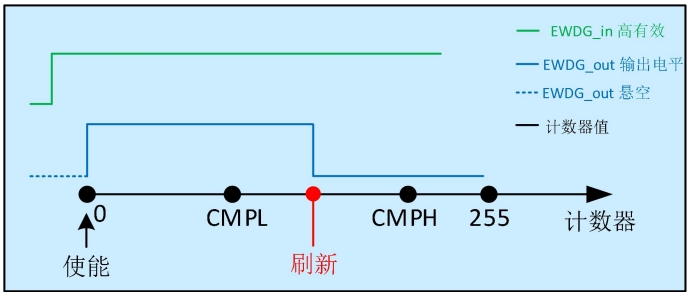


图 23-5 EWDG_in 有效时刷新

当系统复位时，EWDG_out 为悬空态，此时的电平由外部上下拉电阻决定，建议 EWDG_out 外接下拉电阻，当系统复位时，可以保持外部器件或电路处于关断状态：

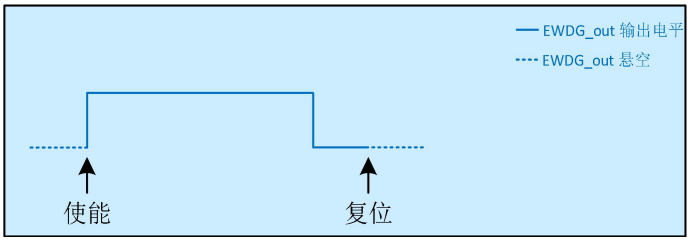


图 23-6 EWDG_out 复位状态

23.4.2 刷新操作

在 15 个外设总线时钟周期内向 EWDG_SERV 寄存器连续写入两个值：0xB4、0x2C 完成一次刷新操作，刷新使当前计数器清除为 0，并重新开始计数。

需要注意的是，刷新序列写入后，计数器清 0 必然会有延时，延时时间为 2 个 LSI 时钟周期，有可能在这 2 个延时时间内计数器达到了 EWDG_CMPH，导致拉低 EWDG_out。由于延时而导致的计数器增加值为：EWDG_CLKPRESCALER [CLK_DIV]=0 时，计数器延迟 3 个后清 0；EWDG_CLKPRESCALER [CLK_DIV]=1/2 时，计数器延迟 1 个后清 0；EWDG_CLKPRESCALER [CLK_DIV]>2，计数器清 0 无延迟。

当刷新的序列值不正确或者超时写入，刷新无效。

23.4.3 EWDG_in

EWDG_in 是一个可选择的功能，当 EWDG_in 有效时，并且刷新， EWDG_out 拉低。
EWDG_in 可配置为输入高有效或者低有效。

23.4.4 Write-Once

EWDG 的 EWDG_CTRL[EWDGEN] 、 EWDG_CTRL[ASSIN] 、 EWDG_CTRL[INEN] 、 EWDG_CMPL、EWDG_CMPH、EWDG_CLKPRESCALER[CLK_DIV]这些位是一次写入位，只能进行一次配置，重新的配置会触发 Hardfault 中断。

© 2013 - 2024 杰发科技有限公司

23.4.5 无效配置

当 EWDG_CMPH 为 255 或者 EWDG_CMPL 大于等于 EWDG_CMPH 时，使能 EWDG 不会使 EWDG 运行。

23.4.6 中断请求

表 23-1 EWDG 中断汇总

中断源	标志	本地使能	全局使能
EWDG_out 拉低		EWDG_CTRL [INTEN]	

说明：
当 EWDG_out 拉低事件发生时，如果使能了 EWDG 中断，则产生 EWDG 中断，EWDG 没有中断标志位，Disable CTRL[INTEN]关闭 EWDG 中断。

23.4.7 低功耗模式

芯片进入 STOP2 模式时，EWDG 保持全功能运行，中断使能条件下若 EWDG_out 拉低事件发生，则产生中断，系统唤醒。
芯片进入 STOP1/VLPS 模式时，EWDG 计数器暂停运行，唤醒后继续运行。

23.4.8 调试模式

EWDG 模块不受调试状态的影响。

23.5 应用说明

23.5.1 编程指南

EWDG 使用配置如下：

1. 编程 EWDG_CMPL 寄存器；
2. 编程 EWDG_CMPH 寄存器；
3. 编程 EWDG_CLKPRESCALER 寄存器；
4. 编程 EWDG_CTRL [INEN]，选择是否使用 EWDG_in 的功能；
5. 编程 EWDG_CTRL [ASSIN]，选择 EWDG_in 有效电平极性；
6. 编程 EWDG_CTRL [INTEN]，选择是否使能 EWDG 中断；

© 2013 - 2024 杰发科技有限公司

7. 编程 EWDG_CTRL [EWDGEN]，开始运行。

EWDG 模块一次编程后，便开始工作。当 EWDG_out 拉低事件发生后，复位模块或者系统复位重新使用 EWDG 模块。

23.6 寄存器定义

表 23-2 EWDG 寄存器映射

EWDG 基地址：0x4000B400

地址	名称	宽度	描述
EWDG 基地址+0x00	EWDG_CTRL	32	控制寄存器
EWDG 基地址+0x04	EWDG_SERV	32	计数器寄存器
EWDG 基地址+0x08	EWDG_CMPL	32	CMPL 寄存器
EWDG 基地址+0x0C	EWDG_CMPH	32	CMPH 寄存器
EWDG 基地址+0x10	EWDG_CLKPRESCALER	32	预分频器寄存器

23.6.1 控制寄存器（EWDG_CTRL）

表 23-3 EWDG_CTRL 寄存器

EWDG_CTRL **控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													INTEN	INEN	ASSIN	EWDGEN
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3 INTEN	**中断使能** 0：中断禁用 1： 中断使能 **当 EWDG_out 拉低时，此位决定是否同时产生中断** **EWDG 没有中断标志位，将 INTEN 写 0 来关闭中断**
2 INEN	**输入使能** 0：禁用 EWDG_in 1：启用 EWDG_in

© 2013 - 2024 杰发科技有限公司

字段	说明
	EWDG_in 启用后，ASSIN 决定 EWDG_in 的有效电平，EWDG_in 有效时，刷新会拉低 EWDG_out. **该位为一次写入位**
1 ASSIN	**输入极性选择位** 0：EWDG_in 的有效电平为低电平 1：EWDG_in 的有效电平为高电平 **该位为一次写入位**
0 EWDGEN	**模块使能** 0：EWDG 不工作 1：EWDG 工作 **当该位写 1 时，EWDG 按照预设的配置开始工作** **该位为一次写入位**

23.6.2 计数器寄存器（EWDG_SERV）

表 23- 4 EWDG_SERV 寄存器

EWDG_SERV **计数器寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SERV							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7: 0 SERV	**计数器** 向该寄存器写入 0xB4、0x2C，计数器将清 0 并重新开始计数，如果写入的值不对或者两次写入的间隔时间大于了 15 个 APB 总线时钟周期，刷新无效。 读取该寄存器返回当前的计数值。

© 2013 - 2024 杰发科技有限公司

23.6.3 CMPL 寄存器（EWDG_CMPL）

表 23-5 EWDG_CMPL 寄存器

EWDG_CMPL **CMPL 寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CMPL							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7: 0 CMPL	**CMPL** CMPL 决定 EWDG 窗口的下限值，当刷新时机出现在当前计数器的值小于 CMPL 时，EWDG_out 拉低。 **CMPL 为一次写入位**

23.6.4 CMPH 寄存器（EWDG_CMPH）

表 23-6 EWDG_CMPH 寄存器

EWDG_CMPH **CMPH 寄存器** **复位值：0x000000FF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CMPH							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7: 0 CMPH	**CMPH** CMPH 决定 EWDG 的超时溢出值，当计数器达到 CMPH 时， EWDG_out 拉低； **CMPH 为一次写入位**

© 2013 - 2024 杰发科技有限公司

23.6.5 预分频器寄存器（EWDG_CLKPRESCALER）

表 23-7 EWDG_CLKPRESCALER 寄存器

EWDG_CLKPRESCALER 预分频器寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CLK_DIV							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7: 0 CLK_DIV	**预分频寄存器** 对 LSI 时钟做分频，然后作为 EWDG 计数器的时钟。 计数频率 = LSI/(1+CLK_DIV)。 **CLK_DIV 为一次写入位**

© 2013 - 2024 杰发科技有限公司

24 连接单元（CTU）

24.1 简介

连接单元(Connection Unit，CTU)在不同的触发源和触发目标之间提供了灵活的连接机制，用于片上不同模块或引脚之间信号传递。 AC7840x 包含 1 个 CTU 模块。

24.2 特性

- ADC 硬件触发
- PWM 触发
  - PWM FAULT 触发
  - PWM 硬件触发
- PWM 同步
- ACMP 输出捕获
- RTC 时钟捕获

© 2013 - 2024 杰发科技有限公司

24.3 结构框图

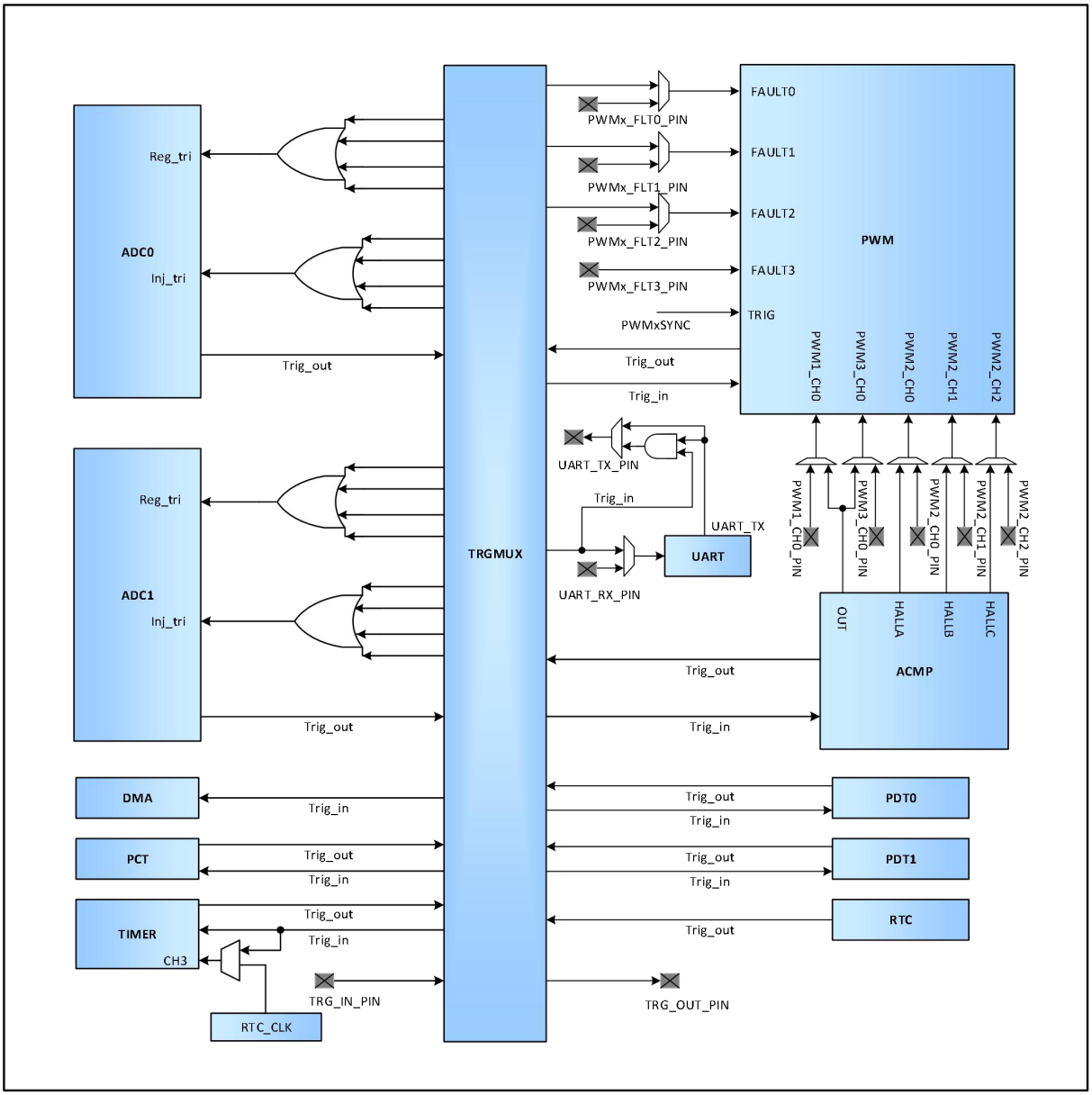


图 24-1 CTU 结构框图

24.4 功能描述

24.4.1 TRGMUX 触发源存储器映射表

TRGMUX 提供了各模块外设及硬件引脚间的互连的方式，该过程可以通过软件配置来实现。 支持的触发源如表 24-1 所示。

© 2013 - 2024 杰发科技有限公司

表 24-1 TRGMUX 触发源存储器映射表

序号	触发源	序号	触发源
0	TRIG_SOURCE_DISABLE	33	TRIG_SOURCE_PWM4_INIT_TRIG
1	TRIG_SOURCE_ENABLE	34	TRIG_SOURCE_PWM4_MATCH_TRIG
2	TRIG_SOURCE_EXT_IN0	35	TRIG_SOURCE_PWM4_MAX_TRIG
3	TRIG_SOURCE_EXT_IN1	36	TRIG_SOURCE_PWM5_INIT_TRIG
4	TRIG_SOURCE_EXT_IN2	37	TRIG_SOURCE_PWM5_MATCH_TRIG
5	TRIG_SOURCE_EXT_IN3	38	TRIG_SOURCE_PWM5_MAX_TRIG
6	TRIG_SOURCE_EXT_IN4	39	Reserved
7	TRIG_SOURCE_EXT_IN5	40	Reserved
8	TRIG_SOURCE_EXT_IN6	41	Reserved
9	TRIG_SOURCE_EXT_IN7	42	Reserved
10	TRIG_SOURCE_EXT_IN8	43	Reserved
11	TRIG_SOURCE_EXT_IN9	44	Reserved
12	TRIG_SOURCE_EXT_IN10	45	TRIG_SOURCE_ADC0_EOC
13	TRIG_SOURCE_EXT_IN11	46	TRIG_SOURCE_ADC0_IEOC
14	TRIG_SOURCE_ACMP0_OUT	47	TRIG_SOURCE_ADC0_AMO
15	Reserved	48	TRIG_SOURCE_ADC1_EOC
16	TRIG_SOURCE_TIMER_CH0	49	TRIG_SOURCE_ADC1_IEOC
17	TRIG_SOURCE_TIMER_CH1	50	TRIG_SOURCE_ADC1_AMO
18	TRIG_SOURCE_TIMER_CH2	51	TRIG_SOURCE_PDT0_TRIG
19	TRIG_SOURCE_TIMER_CH3	52	TRIG_SOURCE_PDT0_PULSE_OUT
20	TRIG_SOURCE_PCT0_TRIG	53	TRIG_SOURCE_PDT1_TRIG
21	TRIG_SOURCE_PWM0_INIT_TRIG	54	TRIG_SOURCE_PDT1_PULSE_OUT
22	TRIG_SOURCE_PWM0_MATCH_TRIG	55	TRIG_SOURCE_RTC_ALARM_TRIG
23	TRIG_SOURCE_PWM0_MAX_TRIG	56	TRIG_SOURCE_RTC_PRESCALER_TRIG
24	TRIG_SOURCE_PWM1_INIT_TRIG	57	TRIG_SOURCE_EIO_TRIG0
25	TRIG_SOURCE_PWM1_MATCH_TRIG	58	TRIG_SOURCE_EIO_TRIG1
26	TRIG_SOURCE_PWM1_MAX_TRIG	59	TRIG_SOURCE_EIO_TRIG2
27	TRIG_SOURCE_PWM2_INIT_TRIG	60	TRIG_SOURCE_EIO_TRIG3
28	TRIG_SOURCE_PWM2_MATCH_TRIG	61	TRIG_SOURCE_SW_TRIG0
29	TRIG_SOURCE_PWM2_MAX_TRIG	62	TRIG_SOURCE_SW_TRIG1
30	TRIG_SOURCE_PWM3_INIT_TRIG	63	TRIG_SOURCE_SW_TRIG2
31	TRIG_SOURCE_PWM3_MATCH_TRIG	64	TRIG_SOURCE_SW_TRIG3
32	TRIG_SOURCE_PWM3_MAX_TRIG	-	-

24.4.2 TRGMUX 触发目标选择

CTU 支持触发目标选择的模块如表 24-2 所示。各模块可通过 CTU 中的 TRGMUX 寄存器选择表 24-1 中的触发源，配置相应模块的触发功能。

© 2013 - 2024 杰发科技有限公司

表 24-2 CTU 支持触发目标选择的模块

支持触发目标选择的模块	CTU 寄存器
DMA	TRGMUX_DMA
EXTOUT	TRGMUX_EXTOUT0 TRGMUX_EXTOUT1
ADC	TRGMUX_ADC0_REG TRGMUX_ADC0_INJ TRGMUX_ADC1_REG TRGMUX_ADC1_INJ
ACMP	TRGMUX_ACMP
PWM	TRGMUX_PWM0 TRGMUX_PWM1 TRGMUX_PWM2 TRGMUX_PWM3 TRGMUX_PWM4 TRGMUX_PWM5
TIMER	TRGMUX_TIMER
PCT	TRGMUX_PCT
UART	TRGMUX_UART0 TRGMUX_UART1
PDT	TRGMUX_PDT0 TRGMUX_PDT1
EIO	TRGMUX_EIO

说明：

根据奈奎斯特采样定理，部分模块触发信号的脉宽为 1 个系统时钟，支持输入捕获功能的模块不能准确采样其触发信号。比如以下场景有此限制：

PWM 模块的触发信号，无法被 PCT, Timer 准确捕获；

PDT 模块的延迟触发信号，无法被 PCT，Timer，PWM 准确捕获；

Timer mode0，mode2，mode3 的触发信号，无法被 PCT，Timer 准确捕获；

24.4.3 ADC 硬件触发

每组 ADC 包含有 2 组触发输入：

- 规则组触发
- 注入组触发

© 2013 - 2024 杰发科技有限公司

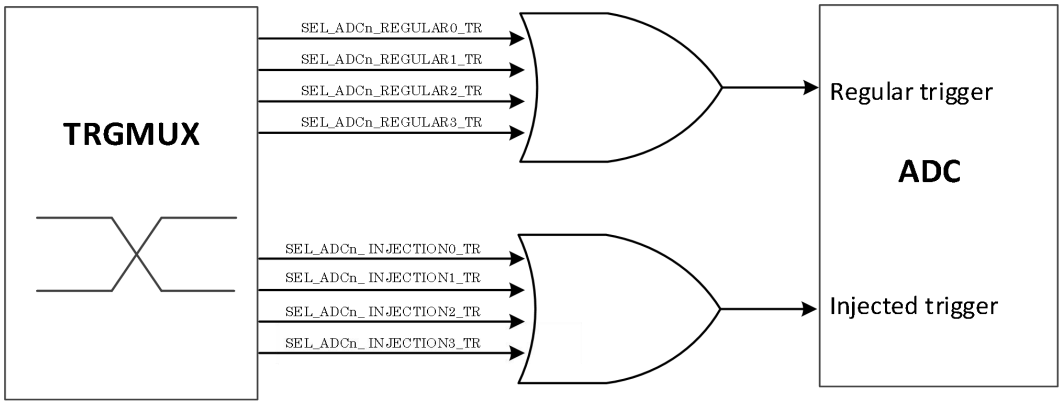


图 24- 2 ADC 硬件触发

TRGMUX 中可选择 ADC 规则组触发源/注入组触发源：

通过 TRGMUX_ADC0_REG，TRGMUX_ADC1_REG 寄存器可配置 ADCn 规则组的 4 路触发源，在配置了 4 路触发源信号后，当任意一路触发信号到达时，均会触发 ADCn 的规则组转换；

通过 TRGMUX_ADC0_INJ，TRGMUX_ADC1_INJ 寄存器可配置 ADCn 注入组的 4 路触发源，在配置了 4 路触发源信号后，当任意一路触发信号到达时，均会触发 ADCn 的注入组转换

24.4.4 PWM 触发

PWMn（n = 0, 1, 2, 3）模块支持 4 路触发输入(3 路 FAULT 触发，1 路 PWM 硬件触发)：

- PWM TRIG0
- PWM FAULT0
- PWM FAULT1
- PWM FAULT2

PWM4，PWM5 支持 1 路触发输入

- PWM TRIG0

24.4.4.1 PWM FAULT 触发

PWMn (n = 0, 1, 2, 3) FAULT 触发：

© 2013 - 2024 杰发科技有限公司

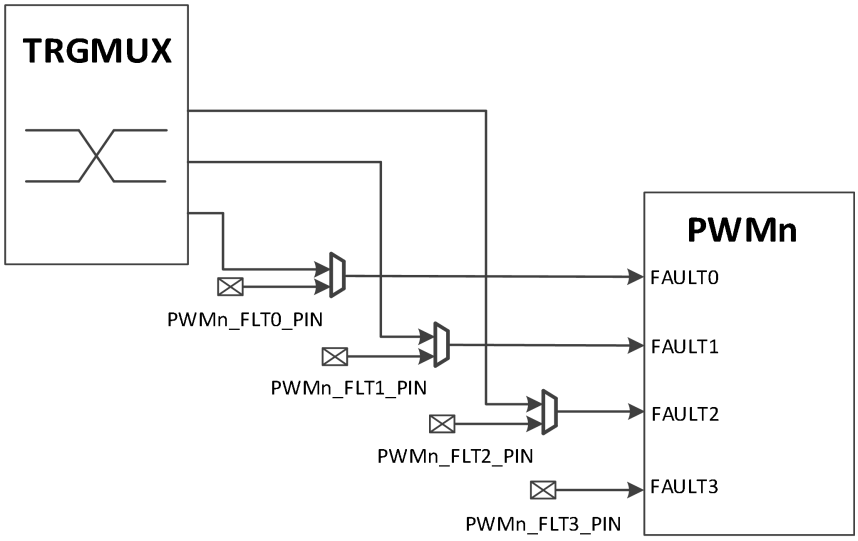


图 24-3 PWM_FAULT 触发输入

- CTU_CFG[SEL_PWM0_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM0_FAULT0，PWM0_FAULT0_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM0_FAULT1] = 1，选择 TRGMUX 输入信号连接至 PWM0_FAULT1，PWM0_FAULT1_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM0_FAULT2] = 1，选择 TRGMUX 输入信号连接至 PWM0_FAULT2，PWM0_FAULT2_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM1_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM1_FAULT0，PWM1_FAULT0_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM1_FAULT1] = 1，选择 TRGMUX 输入信号连接至 PWM1_FAULT1，PWM1_FAULT1_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM1_FAULT2] = 1，选择 TRGMUX 输入信号连接至 PWM1_FAULT2，PWM1_FAULT2_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM2_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM2_FAULT0，PWM2_FAULT0_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM2_FAULT1] = 1，选择 TRGMUX 输入信号连接至 PWM2_FAULT1，PWM2_FAULT1_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM2_FAULT2] = 1，选择 TRGMUX 输入信号连接至 PWM2_FAULT2，PWM2_FAULT2_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM3_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM3_FAULT0，PWM3_FAULT0_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM3_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM3_FAULT1，PWM3_FAULT1_PIN 引脚可用作其他功能；
- CTU_CFG[SEL_PWM3_FAULT0] = 1，选择 TRGMUX 输入信号连接至 PWM3_FAULT2，PWM3_FAULT2_PIN 引脚可用作其他功能；
- 其他通道的 PWMn_FAULT 可通过 TGRMUX 连接至 PWMn_FAULT_PIN。
- 配置 TRGMUX_PWM0，TRGMUX_PWM1，TRGMUX_PWM2，TRGMUX_PWM3 寄存器的硬件触发源选择位，可选择 PWMn FAULT (n=0，1，2，3)的触发输入源。

© 2013 - 2024 杰发科技有限公司

24.4.4.2 PWM 硬件触发

配置 TRGMUX_PWMn (n=0，1，…5) 寄存器的硬件触发源选择位，可选择 PWMn (n = 0，1，…5) 的硬件触发输入源。

24.4.5 PWM 同步

- CTU_CFG[SEL_PWM0SYNC] = 1，生成 PWM0 软件触发同步触发信号；
- CTU_CFG[SEL_PWM1SYNC] = 1，生成 PWM1 软件触发同步触发信号；
- CTU_CFG[SEL_PWM2SYNC] = 1，生成 PWM2 软件触发同步触发信号；
- CTU_CFG[SEL_PWM3SYNC] = 1，生成 PWM3 软件触发同步触发信号；
- CTU_CFG[SEL_PWM4SYNC] = 1，生成 PWM4 软件触发同步触发信号；
- CTU_CFG[SEL_PWM5SYNC] = 1，生成 PWM5 软件触发同步触发信号。

24.4.6 RTC 时钟捕获

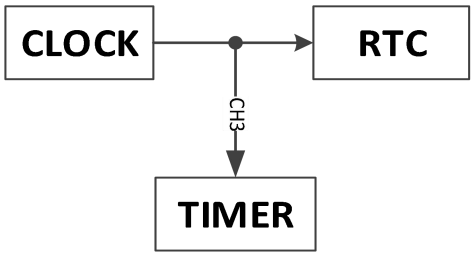


图 24-4 RTC 时钟捕获

CTU_CFG[RTC_CLK_CAP] = 1，使能 RTC_CLK 捕获，将 RTC_CLK 连接至 TIMER_CH3。

RTC_CLK 可通过 CKGEN_LP_CLK_MUX[RTC_CLK_MUX]配置，支持 4 种时钟选择。

© 2013 - 2024 杰发科技有限公司

24.4.7 ACMP 输出捕获

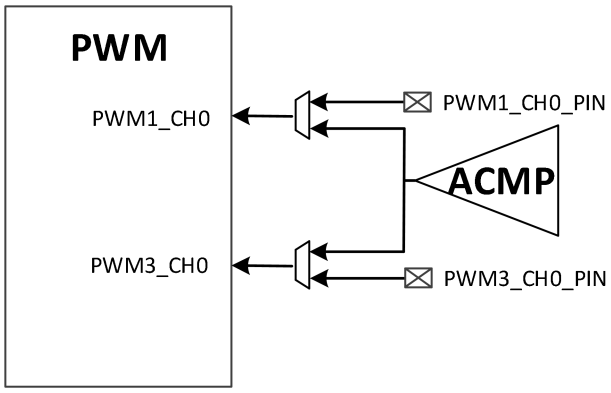


图 24-5 PWM 捕获 ACMP 输出

- CTU_CFG[SEL_PWM1_CH0] = 1，使能 ACMP_OUT 输出至 PWM1_CH0，PWM1_CH0_PIN 引脚引脚可用作其他功能；
- CTU_CFG[SEL_PWM3_CH0] = 1，使能 ACMP_OUT 输出至 PWM3_CH0，PWM3_CH0_PIN 引脚引脚可用作其他功能。

24.4.8 软件触发

配置 CTU_SW[SW_TRIGn_TS] = 1 (n = 0, 1, 2, 3)，生成软件触发信号。

24.4.9 寄存器锁定

TRGMUX 寄存器最高位 LKn 为寄存器锁定控制位。 LKn 位在系统复位(或模块复位)后可写，且仅可写一次。一旦设置了锁定（LKn = 1），则在下一次系统复位(或模块复位)清除锁定之前无法更改寄存器中触发源选择位的值。

24.4.10 中断请求

无。

© 2013 - 2024 杰发科技有限公司

24.4.11 低功耗模式

表 24-3 CTU 模块低功耗模式

模式	唤醒源	备注
STOP1	\	低功耗模式下模块正常工作
STOP2	\	
VLPS	\	

24.4.12 调试模式

调试模式下模块正常工作。

24.5 应用说明

24.5.1 UART 调制

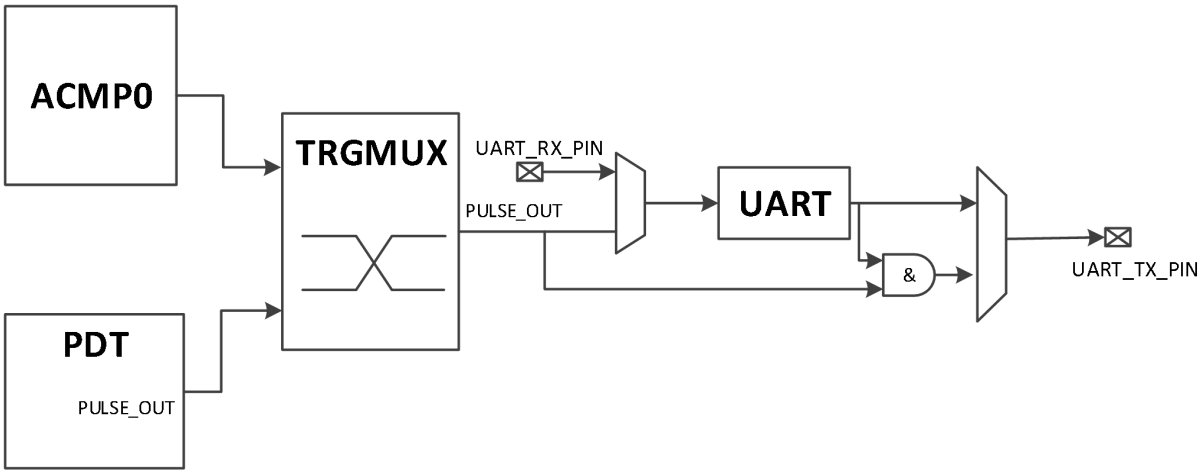


图 24-6 UART 调制

UART 调制配置过程：

1. 通过 CTU 配置 PDT 输出的脉宽信号作为 UART 的触发源；
2. 使能 UART 调制；
3. 将 UART 发送的信号与 PDT_PULSE_OUT 信号进行逻辑“与”后，通过 UART_TX_PIN 输出。

© 2013 - 2024 杰发科技有限公司

24.5.2 ACMP HALL 测量

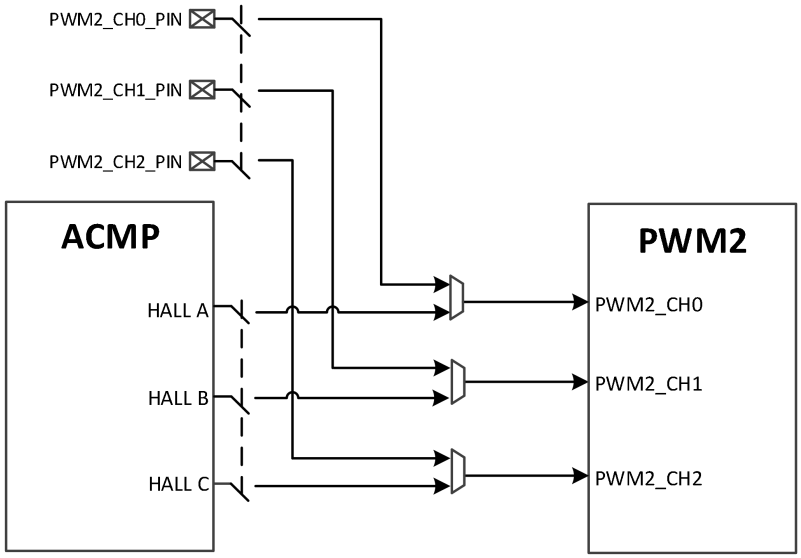


图 24- 7 ACMP HALL 输出至 PWM

ACMP HALL 输出至 PWM 配置过程：

1. 使能 PWM HALL 功能；
2. CTU_CFG[SEL_PWM2_HALL] = 1，将 HALLA / HALLB / HALLC 信号连接至 PWM2_CH0 / PWM2_CH1 / PWM2_CH2 通道；
3. PWM2_CH0_PIN / PWM2_CH1_PIN / PWM2_CH2_PIN 引脚可用作其他功能。

© 2013 - 2024 杰发科技有限公司

24.5.3 PWM 调制

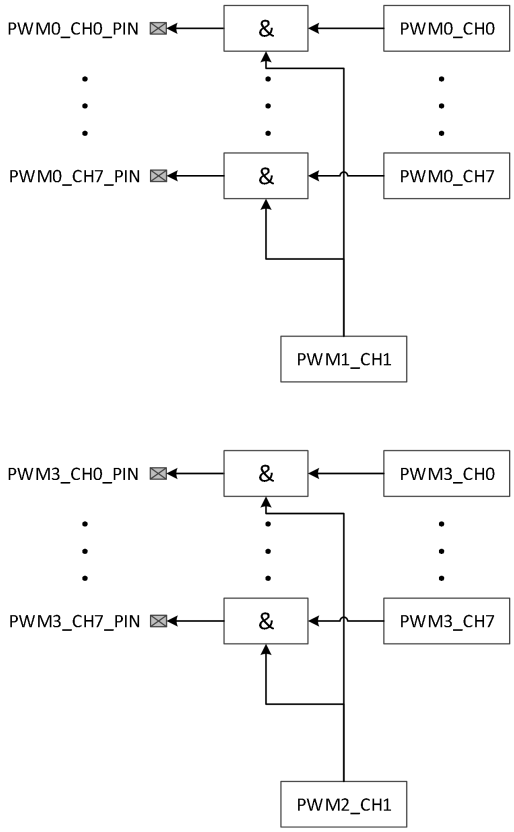


图 24-8 PWM 调制

PWM 调制配置过程：

1. PWM0 和 PWM3 模块支持 PWM 调制功能，PWM1_CH1 可调制 PWM0_CHn(n=0，1，…7)通道的输出，PWM2_CH1 可调制 PWM3_CHn(n=0，1，…7)通道的输出；
2. 调制后的输出信号：通道信号进行逻辑“与”操作后输出至引脚；
3. PWM_MODULATION 寄存器用于选择 PWM0 和 PWM3 的调制通道。

24.6 寄存器定义

表 24-4 CTU 寄存器映射

CTU 基地址：0x40006000

地址	名称	宽度	描述
CTU 基地址 + 0x00	TRGMUX_DMA	32	TRGMUX DMA 配置寄存器
CTU 基地址 + 0x04	TRGMUX_EXTOUT0	32	TRGMUX EXTOUT0 配置寄存器
CTU 基地址 + 0x08	TRGMUX_EXTOUT1	32	TRGMUX EXTOUT1 配置寄存器
CTU 基地址 + 0x0C	TRGMUX_ADC0_REG	32	TRGMUX ADC0 规则组配置寄存器
CTU 基地址 + 0x10	TRGMUX_ADC0_INJ	32	TRGMUX ADC0 注入组配置寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
CTU 基地址 + 0x14	TRGMUX_ADC1_REG	32	TRGMUX ADC1 规则组配置寄存器
CTU 基地址 + 0x18	TRGMUX_ADC1_INJ	32	TRGMUX ADC1 注入组配置寄存器
CTU 基地址 + 0x1C	TRGMUX_ACMP	32	TRGMUX ACMP 配置寄存器
CTU 基地址 + 0x20	TRGMUX_PWM0	32	TRGMUX PWM0 配置寄存器
CTU 基地址 + 0x24	TRGMUX_PWM1	32	TRGMUX PWM1 配置寄存器
CTU 基地址 + 0x28	TRGMUX_PWM2	32	TRGMUX PWM2 配置寄存器
CTU 基地址 + 0x2C	TRGMUX_PWM3	32	TRGMUX PWM3 配置寄存器
CTU 基地址 + 0x30	TRGMUX_PWM4	32	TRGMUX PWM4 配置寄存器
CTU 基地址 + 0x34	TRGMUX_PWM5	32	TRGMUX PWM5 配置寄存器
CTU 基地址 + 0x40	TRGMUX_TIMER	32	TRGMUX TIMER 配置寄存器
CTU 基地址 + 0x44	TRGMUX_PCT	32	TRGMUX PCT 配置寄存器
CTU 基地址 + 0x48	TRGMUX_UART0	32	TRGMUX UART0 配置寄存器
CTU 基地址 + 0x4C	TRGMUX_UART1	32	TRGMUX UART1 配置寄存器
CTU 基地址 + 0x50	TRGMUX_PDT0	32	TRGMUX PDT0 配置寄存器
CTU 基地址 + 0x54	TRGMUX_PDT1	32	TRGMUX PDT1 配置寄存器
CTU 基地址 + 0x58	TRGMUX_EIO	32	TRGMUX EIO 配置寄存器
CTU 基地址 + 0x5C	CTU_CFG	32	CTU 配置寄存器
CTU 基地址 + 0x60	CTU_SW	32	CTU 软件触发源控制寄存器
CTU 基地址 + 0x64	PWM_MODULATION	32	PWM 调制配置寄存器
CTU 基地址 + 0x68	ADC_SYNC	32	ADC 同步模式控制寄存器

24.6.1 TRGMUX DMA 配置寄存器(TRGMUX_DMA)

表 24-5 TRGMUX_DMA 寄存器

TRGMUX_DMA　　TRGMUX DMA 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK1	SEL_DMA_CH3								SEL_DMA_CH2						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_DMA_CH1								SEL_DMA_CH0						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK1	**TRGMUX 寄存器锁** 0：寄存器可写 1：寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK1 被设置为 1，

© 2013 - 2024 杰发科技有限公司

字段	说明
	TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK1。
30：24 SEL_DMA_CH3	**DMA_CH3 硬件触发源** 选择 DMA_CH3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_DMA_CH2	**DMA_CH2 硬件触发源** 选择 DMA_CH2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_DMA_CH1	**DMA_CH1 硬件触发源** 选择 DMA_CH1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_DMA_CH0	**DMA_CH0 硬件触发源** 选择 DMA_CH0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.2 TRGMUX EXTOUT0 配置寄存器(TRGMUX_EXTOUT0)

表 24- 6 TRGMUX_EXTOUT0 寄存器

TRGMUX_EXTOUT0　　TRGMUX EXTOUT0 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK2	SEL_EXT_OUT3_TR								SEL_EXT_OUT2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_EXT_OUT1_TR								SEL_EXT_OUT0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK2	**TRGMUX 寄存器锁** 0：寄存器可写 1：寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK2 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK2。
30：24 SEL_EXT_OUT3_TR	**EXT_OUT3 硬件触发源** 选择 EXT_CH3 的硬件触发源。

© 2013 - 2024 杰发科技有限公司

字段	说明
	相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_EXT_OUT2_TR	**EXT_OUT2 硬件触发源** 选择 EXT_CH2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_EXT_OUT1_TR	**EXT_OUT1 硬件触发源** 选择 EXT_CH1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_EXT_OUT1_TR	**EXT_OUT0 硬件触发源** 选择 EXT_CH0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.3 TRGMUX EXTOUT1 配置寄存器(TRGMUX_EXTOUT1)

表 24- 7 TRGMUX EXTOUT1 寄存器

TRGMUX_EXTOUT1　　TRGMUX EXTOUT1 配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK3	SEL_EXT_OUT7_TR								SEL_EXT_OUT6_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_EXT_OUT5_TR								SEL_EXT_OUT4_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK3	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK3 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK3。
30：24 SEL_EXT_OUT7_TR	**EXT_OUT7 硬件触发源** 选择 EXT_CH7 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_EXT_OUT6_TR	**EXT_OUT6 硬件触发源**

© 2013 - 2024 杰发科技有限公司

字段	说明
	选择 EXT_CH6 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_EXT_OUT5_TR	**EXT_OUT5 硬件触发源** 选择 EXT_CH5 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_EXT_OUT4_TR	**EXT_OUT4 硬件触发源** 选择 EXT_CH4 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.4 TRGMUX ADC0 规则组配置寄存器(TRGMUX_ADC0_REG)

表 24- 8 TRGMUX_ADC0_REG 寄存器

TRGMUX_ADC0_REG　　TRGMUX ADC0 规则组配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK4	SEL_ADC0_REGULAR3_TR								SEL_ADC0_REGULAR2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_ADC0_REGULAR1_TR								SEL_ADC0_REGULAR0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK4	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK4 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK4。
30：24 SEL_ADC0_REGULAR3_TR	**ADC0_REGULAR3 硬件触发源** 选择 ADC0_REGULAR3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_ADC0_REGULAR2_TR	**ADC0_REGULAR2 硬件触发源** 选择 ADC0_REGULAR2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

© 2013 - 2024 杰发科技有限公司

字段	说明
14：8 SEL_ADC0_REGULAR1_TR	**ADC0_REGULAR1 硬件触发源** 选择 ADC0_REGULAR1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_ADC0_REGULAR0_TR	**ADC0_REGULAR0 硬件触发源** 选择 ADC0_REGULAR0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.5 TRGMUX ADC0 注入组配置寄存器(TRGMUX_ADC0_INJ)

表 24- 9 TRGMUX ADC0_INJ 寄存器

TRGMUX_ADC0_INJ　　TRGMUX ADC0 注入组配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK5	SEL_ADC0_ INJECTION3_TR								SEL_ADC0_ INJECTION2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_ADC0_ INJECTION1_TR								SEL_ADC0_ INJECTION0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK5	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK5 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK5。
30：24 SEL_ADC0_INJECTION3_TR	**ADC0_INJECTION3 硬件触发源** 选择 ADC0_INJECTION3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_ADC0_INJECTION2_TR	**ADC0_INJECTION2 硬件触发源** 选择 ADC0_INJECTION2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_ADC0_INJECTION1_TR	**ADC0_INJECTION1 硬件触发源** 选择 ADC0_INJECTION1 的硬件触发源。

© 2013 - 2024 杰发科技有限公司

字段	说明
	相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_ADC0_INJECTION0_TR	**ADC0_INJECTION0 硬件触发源** 选择 ADC0_INJECTION0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.6 TRGMUX ADC1 规则组配置寄存器(TRGMUX_ADC1_REG)

表 24- 10 TRGMUX_ADC1_REG 寄存器

TRGMUX_ADC1_REG　　TRGMUX ADC1 规则组配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK6	SEL_ADC1_REGULAR3_TR								SEL_ADC1_REGULAR2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_ADC1_REGULAR1_TR								SEL_ADC1_REGULAR0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK6	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK6 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK6。
30：24 SEL_ADC1_REGULAR3_TR	**ADC1_REGULAR3 硬件触发源** 选择 ADC1_REGULAR3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_ADC1_REGULAR2_TR	**ADC1_REGULAR2 硬件触发源** 选择 ADC1_REGULAR2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_ADC1_REGULAR1_TR	**ADC1_REGULAR1 硬件触发源** 选择 ADC1_REGULAR1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_ADC1_REGULAR0_TR	**ADC1_REGULAR0 硬件触发源**

© 2013 - 2024 杰发科技有限公司

字段	说明
	选择 ADC1_REGULAR0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.7 TRGMUX ADC1 注入组配置寄存器(TRGMUX_ADC1_INJ)

表 24-11 TRGMUX_ADC1_INJ 寄存器

TRGMUX_ADC1_INJ **TRGMUX ADC1 注入组配置寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK7	SEL_ADC1_ INJECTION3_TR								SEL_ADC1_ INJECTION2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_ADC1_ INJECTION1_TR								SEL_ADC1_ INJECTION0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK7	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK7 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK7。
30：24 SEL_ADC1_INJECTION3_TR	**ADC1_INJECTION3 硬件触发源** 选择 ADC1_INJECTION3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_ADC1_INJECTION2_TR	**ADC1_INJECTION2 硬件触发源** 选择 ADC1_INJECTION2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_ADC1_INJECTION1_TR	**ADC1_INJECTION1 硬件触发源** 选择 ADC1_INJECTION1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_ADC1_INJECTION0_TR	**ADC1_INJECTION0 硬件触发源** 选择 ADC1_INJECTION0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

© 2013 - 2024 杰发科技有限公司

24.6.8 TRGMUX ACMP 配置寄存器(TRGMUX_ACMP)

表 24-12 TRGMUX_ACMP 寄存器

TRGMUX_ACMP **TRGMUX ACMP 配置寄存器** **复位值：0x 00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK8															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_ACMP0_TR						
访问										RW						
复位值										0						

字段	说明
31 LK8	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK8 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK8。
6：0 SEL_ACMP0_TR	**ACMP0 硬件触发源** 选择 ACMP0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.9 TRGMUX PWM0 配置寄存器(TRGMUX_PWM0)

表 24-13 TRGMUX_PWM0 寄存器

TRGMUX_PWM0 **TRGMUX PWM0 配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK9	SEL_PWM0_FAULT2_TR								SEL_PWM0_FAULT1_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_PWM0_FAULT0_TR								SEL_PWM0_TRIG0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
31 LK9	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK9 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK9。
30：24 SEL_PWM0_FAULT2_TR	**PWM0_FAULT2 硬件触发源** 选择 PWM0_FAULT2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_PWM0_FAULT1_TR	**PWM0_FAULT1 硬件触发源** 选择 PWM0_FAULT1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_PWM0_FAULT0_TR	**PWM0_FAULT0 硬件触发源** 选择 PWM0_FAULT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_PWM0_TRIG0_TR	**PWM0_TRIG0 硬件触发源** 选择 PWM0_TRIG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.10 TRGMUX PWM1 配置寄存器(TRGMUX_PWM1)

表 24-14 TRGMUX_PWM1 寄存器

TRGMUX_PWM1　　TRGMUX PWM1 配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK10	SEL_PWM1_FAULT2_TR								SEL_PWM1_FAULT1_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_PWM1_FAULT0_TR								SEL_PWM1_TRIG0 _TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK10	**TRGMUX 寄存器锁** 0： 寄存器可写

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK10 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK10。
30：24 SEL_PWM1_FAULT2_TR	**PWM1_FAULT2 硬件触发源** 选择 PWM1_FAULT2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_PWM1_FAULT1_TR	**PWM1_FAULT1 硬件触发源** 选择 PWM1_FAULT1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_PWM1_FAULT0_TR	**PWM1_FAULT0 硬件触发源** 选择 PWM1_FAULT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_PWM1_TRIG0_TR	**PWM1_TRIG0 硬件触发源** 选择 PWM1_TRIG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.11 TRGMUX PWM2 配置寄存器(TRGMUX_PWM2)

表 24-15 TRGMUX_PWM2 寄存器

TRGMUX_PWM2 **TRGMUX PWM2 配置寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK11	SEL_PWM2_FAULT2 _TR								SEL_PWM2_FAULT1_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_PWM2_FAULT0_TR								SEL_PWM2_TRIG0 _TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK11	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK11 被设置为

© 2013 - 2024 杰发科技有限公司

字段	说明
	1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK11。
30：24 SEL_PWM2_FAULT2_TR	**PWM2_FAULT2 硬件触发源** 选择 PWM2_FAULT2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_PWM2_FAULT1_TR	**PWM2_FAULT1 硬件触发源** 选择 PWM2_FAULT1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_PWM2_FAULT0_TR	**PWM2_FAULT0 硬件触发源** 选择 PWM2_FAULT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_PWM2_TRIG0_TR	**PWM2_TRIG0 硬件触发源** 选择 PWM2_TRIG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.12 TRGMUX PWM3 配置寄存器(TRGMUX_PWM3)

表 24-16 TRGMUX_PWM3 寄存器

TRGMUX_PWM3　　TRGMUX PWM3 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK12	SEL_PWM3_FAULT2 _TR								SEL_PWM3_FAULT1_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_PWM3_FAULT0_TR								SEL_PWM3_TRIG0 _TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK12	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK12 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK12。
30：24	**PWM3_FAULT2 硬件触发源**

© 2013 - 2024 杰发科技有限公司

SEL_PWM3_FAULT2_TR	选择 PWM3_FAULT2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_PWM3_FAULT1_TR	**PWM3_FAULT1 硬件触发源** 选择 PWM3_FAULT1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_PWM3_FAULT0_TR	**PWM3_FAULT0 硬件触发源** 选择 PWM3_FAULT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_PWM3_TRIG0_TR	**PWM3IG0 硬件触发源** 选择 PWM3IG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.13 TRGMUX PWM4 配置寄存器(TRGMUX_PWM4)

表 24-17 TRGMUX_PWM4 寄存器

TRGMUX_PWM4　　TRGMUX PWM4 配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK13															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_PWM4_TRIG0 _TR						
访问										RW						
复位值										0						

字段	说明
31 LK13	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK13 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK13。
6：0 SEL_PWM4_TRIG0_TR	**PWM4_TRIG0 硬件触发源** 选择 PWM4_TRIG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

© 2013 - 2024 杰发科技有限公司

24.6.14 TRGMUX PWM5 配置寄存器(TRGMUX_PWM5)

表 24-18 TRGMUX_PWM5 寄存器

TRGMUX_PWM5 **TRGMUX PWM5 配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK14															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_PWM5_TRIG0 _TR						
访问										RW						
复位值										0						

字段	说明
31 LK14	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK14 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK14。
6：0 SEL_PWM5_TRIG0_TR	**PWM5_TRIG0 硬件触发源** 选择 PWM5_TRIG0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.15 TRGMUX TIMER 配置寄存器(TRGMUX_TIMER)

表 24-19 TRGMUX_TIMER 寄存器

TRGMUX_TIMER **TRGMUX TIMER 配置寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK15	SEL_TIMER_CH3_TR								SEL_TIMER_CH2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_TIMER_CH1_TR								SEL_TIMER_CH0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
31 LK15	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK15 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK15。
30：24 SEL_TIMER_CH3_TR	**TIMER_CH3 硬件触发源** 选择 TIMER_CH3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_TIMER_CH2_TR	**TIMER_CH2 硬件触发源** 选择 TIMER_CH2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_TIMER_CH1_TR	**TIMER_CH1 硬件触发源** 选择 TIMER_CH1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_TIMER_CH0_TR	**TIMER_CH0 硬件触发源** 选择 TIMER_CH0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.16 TRGMUX PCT 配置寄存器(TRGMUX_PCT)

表 24-20 TRGMUX_PCT 寄存器

TRGMUX_PCT **TRGMUX PCT 配置寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK16															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_PCT0 _TR						
访问										RW						
复位值										0						

字段	说明
31 LK16	**TRGMUX 寄存器锁** 0： 寄存器可写

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK16 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK16。
6：0 SEL_PCT0_TR	**PCT0 硬件触发源** 选择 PCT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.17 TRGMUX UART0 配置寄存器(TRGMUX_UART0)

表 24-21 TRGMUX_UART0 寄存器

TRGMUX_UART0　　TRGMUX UART0 配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK17															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_UART0_TR						
访问										RW						
复位值										0						

字段	说明
31 LK17	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK17 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK17。
6：0 SEL_UART0_TR	**UART0 硬件触发源** 选择 UART0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.18 TRGMUX UART1 配置寄存器(TRGMUX_UART1)

表 24-22 TRGMUX_UART1 寄存器

TRGMUX_UART1　　TRGMUX UART1 配置寄存器　　复位值： 0x00000000

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK18															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_UART1_TR						
访问										RW						
复位值										0						

字段	说明
31 LK18	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK18 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK18。
6：0 SEL_UART1_TR	**UART1 硬件触发源** 选择 UART1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.19 TRGMUX PDT0 配置寄存器(TRGMUX_PDT0)

表 24- 23 TRGMUX_PDT0 寄存器

TRGMUX_PDT0 **TRGMUX PDT0 配置寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK19															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_PDT0_TR						
访问										RW						
复位值										0						

字段	说明
31 LK19	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写

© 2013 - 2024 杰发科技有限公司

字段	说明
	该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK19 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK19。
6：0 SEL_PDT0_TR	**PDT0 硬件触发源** 选择 PDT0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.20 TRGMUX PDT1 配置寄存器(TRGMUX_PDT1)

表 24-24 TRGMUX_PDT1 寄存器

TRGMUX_PDT1　　TRGMUX PDT1 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK20															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SEL_PDT1_TR						
访问										RW						
复位值										0						

字段	说明
31 LK20	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK20 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK20。
6：0 SEL_PDT1_TR	**PDT1 硬件触发源** 选择 PDT1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

24.6.21 TRGMUX EIO 配置寄存器(TRGMUX_EIO)

表 24-25 TRGMUX_EIO 寄存器

TRGMUX_EIO　　TRGMUX EIO 配置寄存器　　复位值： 0x00000000

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	LK21	SEL_EIO_TIMER3 _TR								SEL_EIO_TIMER2_TR						
访问	RW	RW								RW						
复位值	0	0								0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		SEL_EIO_TIMER1_TR								SEL_EIO_TIMER0_TR						
访问		RW								RW						
复位值		0								0						

字段	说明
31 LK21	**TRGMUX 寄存器锁** 0： 寄存器可写 1： 寄存器不可写 该 bit 位在系统复位(或模块复位)后可写，且仅可写一次。 当 LK21 被设置为 1，TRGMUX 寄存器中的触发源选择位不可被改变直到系统复位(或模块复位)清除 LK21。
30：24 SEL_EIO_TIMER3_TR	**EIO_TIMER3 硬件触发源** 选择 EIO_TIMER3 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
22：16 SEL_EIO_TIMER2_TR	**EIO_TIMER2 硬件触发源** 选择 EIO_TIMER2 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
14：8 SEL_EIO_TIMER1_TR	**EIO_TIMER1 硬件触发源** 选择 EIO_TIMER1 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。
6：0 SEL_EIO_TIMER0_TR	**EIO_TIMER0 硬件触发源** 选择 EIO_TIMER0 的硬件触发源。 相关字段的定义，可参考 TRGMUX 触发源存储器映射表。

© 2013 - 2024 杰发科技有限公司

24.6.22 CTU 配置寄存器(CTU_CFG)

表 24-26 CTU_CFG 寄存器

CTU_CFG　　CTU CFG 配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RTC_CLK_CAP		SEL_PWM3_FAULT2	SEL_PWM3_FAULT1	SEL_PWM3_FAULT0		SEL_PWM2_FAULT2	SEL_PWM2_FAULT1	SEL_PWM2_FAULT0		SEL_PWM1_FAULT2	SEL_PWM1_FAULT1	SEL_PWM1_FAULT0		SEL_PWM0_FAULT2	SEL_PWM0_FAULT1
访问	RW		RW	RW	RW		RW	RW	RW		RW	RW	RW		RW	RW
复位值	0		0	0	0		0	0	0		0	0	0		0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SEL_PWM0_FAULT0	SEL_PWM5SYNC	SEL_PWM4SYNC	SEL_PWM3SYNC	SEL_PWM2SYNC	SEL_PWM1SYNC	SEL_PWM0SYNC			SEL_PWM2_HALL	SEL_PWM3_CH0	SEL_PWM1_CH0	SEL_UART1_TX	SEL_UART0_TX	SEL_UART1	SEL_UART0
访问	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0			0	0	0	0	0	0	0

字段	说明
31 RTC_CLK_CAP	**RTC 时钟捕获** 0： TIMER_CH3 捕获 TRGMUX 的输入信号 1： TIMER_CH3 捕获 RTC 时钟信号
29 SEL_PWM3_FAULT2	**PWM3_FAULT2 输入信号选择** 0： PWM3_FAULT2_PIN 输入信号连接至 PWM3_FAULT2 1： TRGMUX 输入信号连接至 PWM3_FAULT2
28 SEL_PWM3_FAULT1	**PWM3_FAULT1 输入信号选择** 0： PWM3_FAULT1_PIN 输入信号连接至 PWM3_FAULT1 1： TRGMUX 输入信号连接至 PWM3_FAULT1
27 SEL_PWM3_FAULT0	**PWM3_FAULT0 输入信号选择** 0： PWM3_FAULT0_PIN 输入信号连接至 PWM3_FAULT0 1： TRGMUX 输入信号连接至 PWM3_FAULT0
25 SEL_PWM2_FAULT2	**PWM2_FAULT2 输入信号选择** 0： PWM2_FAULT2_PIN 输入信号连接至 PWM2_FAULT2

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： TRGMUX 输入信号连接至 PWM2_FAULT2
24 SEL_PWM2_FAULT1	**PWM2_FAULT1 输入信号选择** 0： PWM2_FAULT1_PIN 输入信号连接至 PWM2_FAULT1 1： TRGMUX 输入信号连接至 PWM2_FAULT1
23 SEL_PWM2_FAULT0	**PWM2_FAULT0 输入信号选择** 0： PWM2_FAULT0_PIN 输入信号连接至 PWM2_FAULT0 1： TRGMUX 输入信号连接至 PWM2_FAULT0
21 SEL_PWM1_FAULT2	**PWM1_FAULT2 输入信号选择** 0： PWM1_FAULT2_PIN 输入信号连接至 PWM1_FAULT2 1： TRGMUX 输入信号连接至 PWM1_FAULT2
20 SEL_PWM1_FAULT1	**PWM1_FAULT1 输入信号选择** 0： PWM1_FAULT1_PIN 输入信号连接至 PWM1_FAULT1 1： TRGMUX 输入信号连接至 PWM1_FAULT1
19 SEL_PWM1_FAULT0	**PWM1_FAULT0 输入信号选择** 0： PWM1_FAULT0_PIN 输入信号连接至 PWM1_FAULT0 1： TRGMUX 输入信号连接至 PWM1_FAULT0
17 SEL_PWM0_FAULT2	**PWM0_FAULT2 输入信号选择** 0： PWM0_FAULT2_PIN 输入信号连接至 PWM0_FAULT2 1： TRGMUX 输入信号连接至 PWM0_FAULT2
16 SEL_PWM0_FAULT1	**PWM0_FAULT1 输入信号选择** 0： PWM0_FAULT1_PIN 输入信号连接至 PWM0_FAULT1 1： TRGMUX 输入信号连接至 PWM0_FAULT1
15 SEL_PWM0_FAULT0	**PWM0_FAULT0 输入信号选择** 0： PWM0_FAULT0_PIN 输入信号连接至 PWM0_FAULT0 1： TRGMUX 输入信号连接至 PWM0_FAULT0
14 SEL_PWM5SYNC	**PWM5 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM5 同步触发信号至 PWM 模块
13 SEL_PWM4SYNC	**PWM4 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM4 同步触发信号至 PWM 模块
12 SEL_PWM3SYNC	**PWM3 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM3 同步触发信号至 PWM 模块

© 2013 - 2024 杰发科技有限公司

字段	说明
11 SEL_PWM2SYNC	**PWM2 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM2 同步触发信号至 PWM 模块
10 SEL_PWM1SYNC	**PWM1 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM1 同步触发信号至 PWM 模块
9 SEL_PWM0SYNC	**PWM0 同步信号选择** 0： 无同步触发信号产生 1： 生成 PWM0 同步触发信号至 PWM 模块
6 SEL_PWM2_HALL	**PWM2_CHx(x=0，1，2)输入信号选择** 0： 选择 PWM2_CHx_PIN 引脚(x=0，1，2) 1： 选择 ACMP_HALLA/ACMP_HALLB/ACMP_HALLC
5 SEL_PWM3_CH0	**PWM3_CH0 选择** 0： 选择 PWM3_CH0_PIN 连接至 PWM3_CH0 1： 选择 ACMP0_OUT 连接至 PWM3_CH0
4 SEL_PWM1_CH0	**PWM1_CH0 选择** 0： 选择 PWM1_CH0_PIN 连接至 PWM1_CH0 1： 选择 ACMP0_OUT 连接至 PWM1_CH0
3 SEL_UART1_TX	**UART1 调制** 0： UART1 Tx 触发输出无调制信号 1： UART1 Tx 触发输出有调制信号
2 SEL_UART0_TX	**UART0 调制** 0： UART0 Tx 触发输出无调制信号 1： UART0 Tx 触发输出有调制信号
1 SEL_UART1	**UART1 Rx 滤波选择** 0： RXD 输入信号直接连接到 UART1 模块 1： RXD 输入信号由 ACMP0 滤波，然后注入 UART1 当使能此功能后，且 TRGMUX_UART1 配置触发源为 TRIG_SOURCE_ACMP0_OUT 的情况下，任何标记有 ACMP0 输入的信号都可视为 UART1。
0 SEL_UART0	**UART0 Rx 滤波选择** 0： RXD 输入信号直接连接到 UART0 模块 1： RXD 输入信号由 ACMP0 滤波，然后注入 UART0

© 2013 - 2024 杰发科技有限公司

字段	说明
	当使能此功能后，且 TRGMUX_UART0 配置触发源为 TRIG_SOURCE_ACMP0_OUT 的情况下，任何标记有 ACMP0 输入的信号都可视为 UART0。

24.6.23 CTU 软件触发源控制寄存器(CTU_SW)

表 24-27 CTU_SW 寄存器

CTU_SW **CTU 软件触发源控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SW_TRIG4_TS	SW_TRIG3_TS	SW_TRIG2_TS	SW_TRIG1_TS
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3 SW_TRIG4_TS	**软件触发源 4** 0: 写 0 无触发。 1: 写 1 生成软件触发。
2 SW_TRIG3_TS	**软件触发源 3** 0: 写 0 无触发。 1: 写 1 生成软件触发。
1 SW_TRIG2_TS	**软件触发源 2** 0: 写 0 无触发。 1: 写 1 生成软件触发。
0 SW_TRIG1_TS	**软件触发源 1** 0: 写 0 无触发。 1: 写 1 生成软件触发。

© 2013 - 2024 杰发科技有限公司

24.6.24 PWM 调制配置寄存器(PWM_MODULATION)

表 24-28 PWM_MODULATION 寄存器

PWM_MODULATION　　PWM 调制配置寄存器　　复位值： 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PWM3_CH7_OUTSEL	PWM3_CH6_OUTSEL	PWM3_CH5_OUTSEL	PWM3_CH4_OUTSEL	PWM3_CH3_OUTSEL	PWM3_CH2_OUTSEL	PWM3_CH1_OUTSEL	PWM3_CH0_OUTSEL	PWM0_CH7_OUTSEL	PWM0_CH6_OUTSEL	PWM0_CH5_OUTSEL	PWM0_CH4_OUTSEL	PWM0_CH3_OUTSEL	PWM0_CH2_OUTSEL	PWM0_CH1_OUTSEL	PWM0_CH0_OUTSEL
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15 PWM3_CH7_OUTSEL	**PWM3_CH7 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
14 PWM3_CH6_OUTSEL	**PWM3_CH6 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
13 PWM3_CH5_OUTSEL	**PWM3_CH5 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
12 PWM3_CH4_OUTSEL	**PWM3_CH4 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
11 PWM3_CH3_OUTSEL	**PWM3_CH3 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
10 PWM3_CH2_OUTSEL	**PWM3_CH2 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制

© 2013 - 2024 杰发科技有限公司

字段	说明
9 PWM3_CH1_OUTSEL	**PWM3_CH1 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
8 PWM3_CH0_OUTSEL	**PWM3_CH0 通道与 PWM2_CH1 进行调制** 0： 无调制 1： 与 PWM2_CH1 进行调制
7 PWM0_CH7_OUTSEL	**PWM0_CH7 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
6 PWM0_CH6_OUTSEL	**PWM0_CH6 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
5 PWM0_CH5_OUTSEL	**PWM0_CH5 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
4 PWM0_CH4_OUTSEL	**PWM0_CH4 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
3 PWM0_CH3_OUTSEL	**PWM0_CH3 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
2 PWM0_CH2_OUTSEL	**PWM0_CH2 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
1 PWM0_CH1_OUTSEL	**PWM0_CH1 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制
0 0PWM0_CH0_OUTSEL	**PWM0_CH0 通道与 PWM1_CH1 进行调制** 0： 无调制 1： 与 PWM1_CH1 进行调制

© 2013 - 2024 杰发科技有限公司

24.6.25 ADC 同步模式控制寄存器 (ADC_SYNC)

表 24- 29 ADC_SYNC 寄存器

ADC_SYNC **ADC 同步模式控制寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	ADC1_INTER				ADC0_INTER										SIMU_INJ	SIMU_REG
访问	RW				RW										RW	RW
复位值	0	0	0	0	0	0	0	0							0	0

字段	说明
15：12 ADC1_INTER	**ADC1 时域交织功能配置** 0h: ADC1 时域交织功能关闭 1h: ADC1 时域交织功能开启，ADC0/1 同时采样 PB0 3h: ADC1 时域交织功能开启，ADC0/1 同时采样 PB1 其他值预留 **说明：ADC1 时域交织功能配置可参考 30.4.9 交织模式 章节。**
11：8 ADC0_INTER	**ADC0 时域交织功能配置** 0h: ADC0 时域交织功能关闭 1h: ADC0 时域交织功能开启，ADC0/1 同时采样 PB13 3h: ADC0 时域交织功能开启，ADC0/1 同时采样 PB14 其他值预留 **说明：ADC0 时域交织功能配置可参考 30.4.9 交织模式 章节。**
1 SIMU_INJ	**ADC0 与 ADC1 的注入组同步触发配置** 0：未使能注入组同步触发 1：使能注入组同步触发
0 SIMU_REG	**ADC0 与 ADC1 的规则组同步触发配置** 0：未使能规则组同步触发 1：使能规则组同步触发

© 2013 - 2024 杰发科技有限公司

25 通用异步收发器（UART）

25.1 简介

UART (Universal Asynchronous Receiver/Transmitter，通用异步收发器) 是一种基本的串行通信协议。它通过发射器和接收器来实现诸多功能，由寄存器的 4 个 bit：LINEN、RS485EN、IREN、ILEN 配置实现，如表 25-1 所示。

表 25-1 功能分类和配置

功能	LINEN	RS485EN	IREN	ILEN
BASIC UART	0	0	0	0
LIN	1	0	0	0
RS485	0	1	0	0
IRDA	0	0	1	0

说明：

1. 当处于 RS485 模式时，必须禁用硬件流控制（RTS, CTS）功能。
2. UART 支持 4 路软件 LIN。
3. LIN 模式仅支持 8 位数据格式及 16 倍过采样。与此同时，如果使能自动波特率功能 (LABAUDEN=1)，则同步字段数据 (0x55) 将被丢弃。
4. DMA 功能必须在 FIFO 使能时起作用。
5. IRDA 模式仅支持 8 位数据格式和 16 倍过采样，输出为正常波的 3/16 倍脉冲。
6. 在 ILEN 模式下，仅支持空闲检测功能。
7. 如果使用 DMA，UART 功能时钟需要与 BUS_CLK 同源且频率相同。
8. 如果使用 DMA，UART 使用 DMA 搬运数据的位置需要为：

   SRAM_U(0x20000000~0x2000EFFF)。

25.2 特性

- 全双工，标准不归零（NRZ）格式
- 可编程波特率（16 位分频器）
- 轮询或中断方式查询状态
  - 传输数据寄存器为空且传送完成

© 2013 - 2024 杰发科技有限公司

  - 传输数据寄存器非空
  - 接收数据寄存器已满
  - 接收溢出错误、帧错误、奇偶错误
  - 空闲线路检测
  - 支持 LIN 的同步间隔段检测，可选 10 或 11 位的同步间隔段检测
  - 通过有效的边沿检测将 MCU 从低功耗（STOP1、STOP2、VLPS）模式唤醒
- 可编程 7 或 8 或 9 位数据长度，1 或 2 位停止位，硬件自动生成奇偶校验位
- 可选择发送端输出和接收端输入极性
- 支持发送 13 到 28 位的 LIN 同步间隔段
- 支持时钟源可选
- 支持单线半双工工作模式
- 支持 loopback 模式
- 支持 DMA 传输
- 支持 2 bytes 数据大小的 FIFO 缓存
- 支持硬件流控制
- 支持 RS485 自动控制方向
- 支持 IrDA 编码、解码功能
- 支持数据匹配和地址匹配功能

© 2013 - 2024 杰发科技有限公司

25.3 结构框图

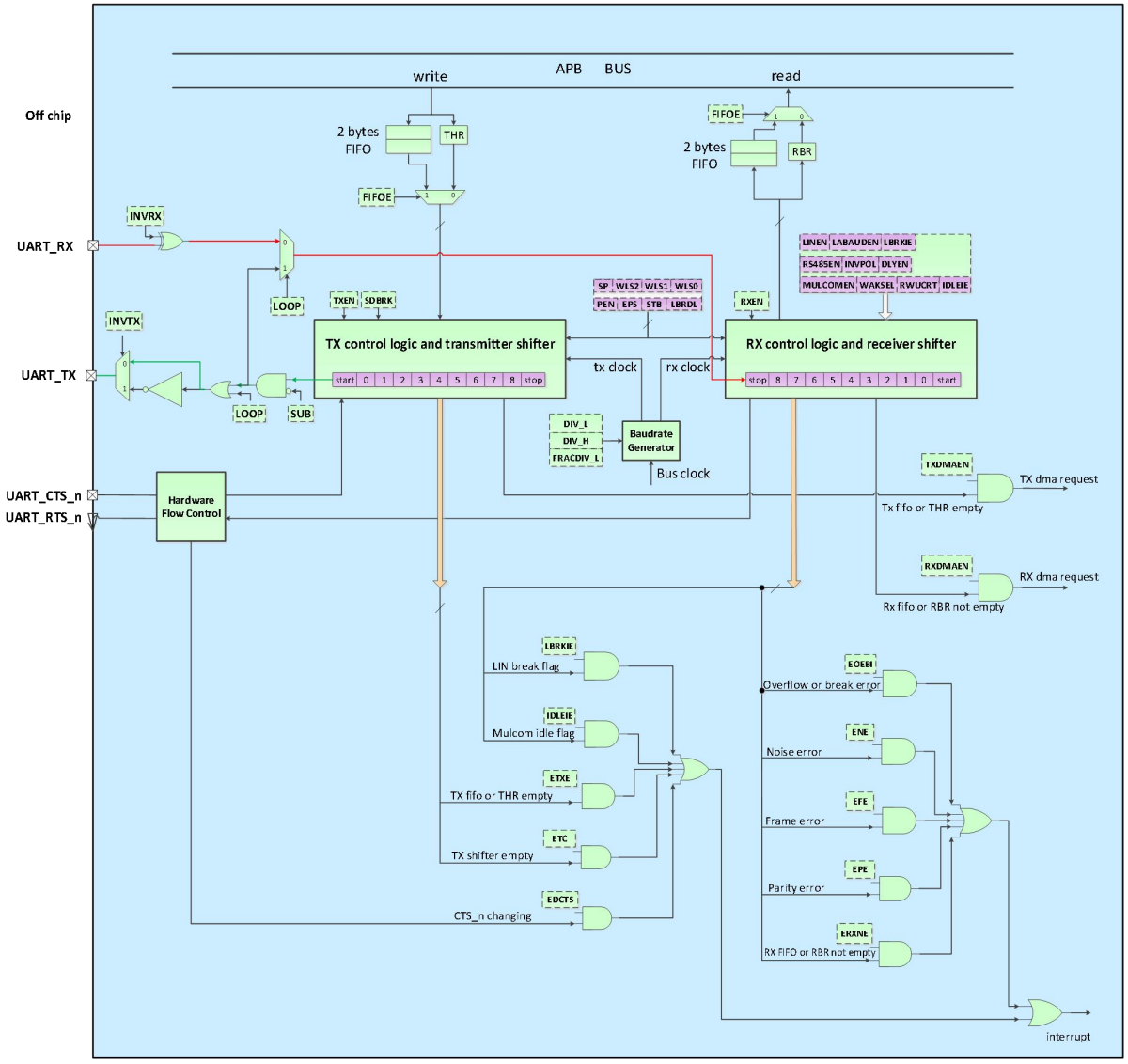


图 25-1 UART 结构框图

25.4 功能描述

UART 功能是逐位发送和接收串行数据。图 25-2 和图 25-3 描述了完整的数据位，包括起始位（ start bit）、数据位（data bits）、奇偶位（ parity bit）、停止位（ stop bits）和保护间隔（ guard time）。 其中 bit7， bit8， bit9，parity，stop1，stop2 位和 guard time 位可由用户配置，详细配置信息请参考 UART_LCR0 和 UART_LCR1 寄存器。通讯时每一位数据，对应于由波特率控制的一个位时间。

UART 发送和接收时有多种状态。知道在发送或接收过程中何时产生这些状态，用户可以更好地使用 UART 功能。

© 2013 - 2024 杰发科技有限公司

如图 25-2 所示，在全局复位或上电后的初始化状态期间，TDNF 和 TC 在 TXEN 设置为 1 后会立即变为 1。但在传输过程中，TDNF 会在起始位后立即变为 1，同时 TC 在最后一位后立即变为 1，比如，若 GUARDEN=1，保护时间位也会如此。

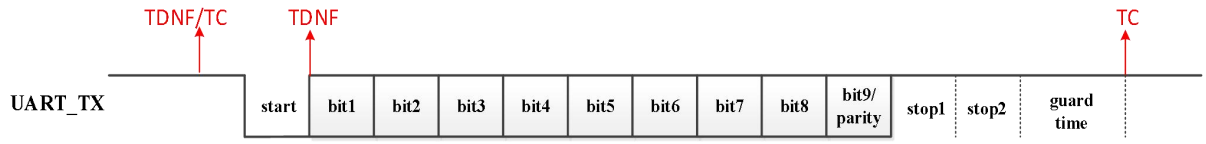


图 25-2 UART 传送器流程

如果在接收过程中发生相关事件，如图 25-3 所示，则状态位 DR， OE， PE， FE， BI 和 NE 会在 stop1 位后立即置为 1。

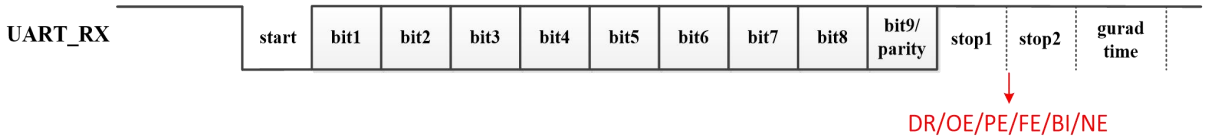


图 25-3 UART 接收器流程

需要指出的是，PE，FE 和 NE 状态仅针对当前接收数据，并会在下一个数据接收完成时被自动清除。对于其他状态，如果未通过读取或写入 1 来清除它们，则它们始终保持该值。

25.4.1 输入输出定时

表 25-2 UART 输入输出定时

引脚名	相应信号	宽度	输入/输出	是否上拉	定时限制
UARTx_TX	uart_tx	1	O（输出）	否	无
UARTx_RX	uart_rx	1	I（输入）	否	无
UARTx_RTS	uart_rts_n	1	O（输出）	否	无
UARTx_CTS	uart_cts_n	1	1（输入）	否	无

说明：

x= 0~3。仅 UART0 ~ UART2 有完整的硬件流控制，其四个信号（UART_TX、UART_RX、UART_RTS、UART_CTS）都可以在引脚中找到。

25.4.2 噪声检测（Noise Detection）

对于 NE 状态，当 UART_RX 信号中存在噪声时就会产生此信号。为了检测噪声， UART_RX 在中间位置被采样三次，如图 25-4 所示。

© 2013 - 2024 杰发科技有限公司

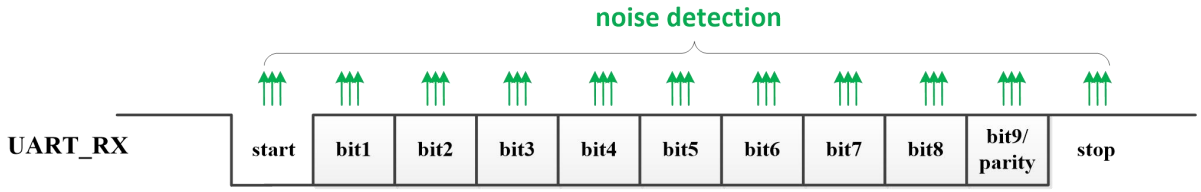


图 25-4 UART 噪声检测

为了更清楚方便地解释，三次采样的值分别叫做 SM1，SM2 和 SM3。 如果 SM1，SM2 和 SM3 中有两个以上的“1”作为起始位，则起始位无效且接收器复位成再次接收。除了起始位的这种情况之外，如果 SM1， SM2 和 SM3 的值互不相同，则检测到噪声， NE 状态变为 1。

25.4.3 波特率描述

UART 波特率精度由多个方面确定，包括 UART 时钟、过采样时间等。因此，一些特定的、过高的波特率没有实现或实现起来误差较大。

表 25-3、表 25-4、表 25-5 分别描述了典型波特率在不同系统时钟下配置及相应的误差率。

表 25-3 UART 的典型波特率及误差率（总线时钟为 24MHz）

序号	理论值 (Kbps)	实际值 (bps)	DIV_MANTI[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
1	2.4	2400	625	0	16	0
2	9.6	9600	156	8	16	0
3	19.2	19200	78	4	16	0
4	57.6	57600	26	1	16	0
5	115.2	115200	13	1	16	0
6	230.4	230769.23	6	16	16	0.16%
7	460.8	461538.47	3	8	16	0.16%
8	921.6	923076.94	1	20	16	0.16%
9	1843.2	-	-	-	-	-
10	4200.0	-	-	-	-	-

表 25-4 UART 的典型波特率及误差率（总线时钟为 48MHz）

序号	理 论 值 (Kbps)	实际值 (bps)	DIV_MANTI[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
1	2.4	2399.98	1250	0	16	-0.001%
2	9.6	9599.69	312	16	16	-0.003%
3	19.2	19202.22	156	8	16	0.006%
4	57.6	57603.69	52	3	16	0.006%
5	115.2	115207.38	26	1	16	0.006%
6	230.4	230414.75	13	1	16	0.006%
7	460.8	460829.50	6	16	16	0.006%
8	921.6	917431.19	3	8	16	-0.452%

© 2013 - 2024 杰发科技有限公司

序号	理 论 值 (Kbps)	实际值 (bps)	DIV_MANTI[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
9	1843.2	1834862.38	3	8	8	-0.452%
10	4200	4166666.75	1	14	8	-0.794%

表 25-5 UART 的典型波特率及误差率（总线时钟为 60MHz）

序号	理 论 值 (Kbps)	实际值 (bps)	DIV_MANTI[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
1	2.4	2400	1562	16	16	-0.001%
2	9.6	9600	390	20	16	-0.003%
3	19.2	19200	195	10	16	0.006%
4	57.6	57609.21747	65	3	16	0.016%
5	115.2	115163.1478	32	18	16	-0.032%
6	230.4	230326.2956	16	9	16	-0.032%
7	460.8	461538.4615	8	4	16	0.160%
8	921.6	923076.9231	4	2	16	0.160%
9	1843.2	1846153.846	2	1	16	0.160%
10	4200	4210526.316	1	25	8	0.251%

25.4.4 单线半双工功能

UART 通讯时，支持一个 UART 节点既用作发送端，也用作接收端，此时 UART 不仅发送数据，也会接收数据，实际应用硬件连接如图 25-5 所示。

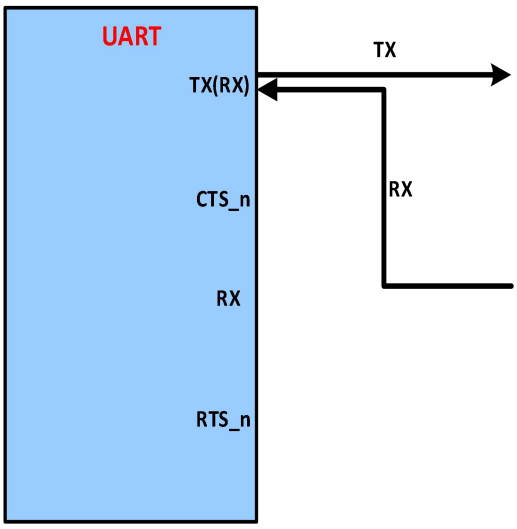


图 25-5 单向半双工连接

说明：

UART 工作于单向半双工模式时，UART TX 端口既可用于发送数据，也可用于接收数据，应用于此工作模式时，一般需要接外部电路控制总线数据传输方向，使数据线在同一时刻只处于输入或输出状态。

25.4.5 硬件流控制功能

UART 硬件流控制两个 UART 间的通讯，并通过 RTS_n 和 CTS_n 以减少 CPU 负荷，这样从容地逐一处理数据，实际应用连接如图 25-6 所示。

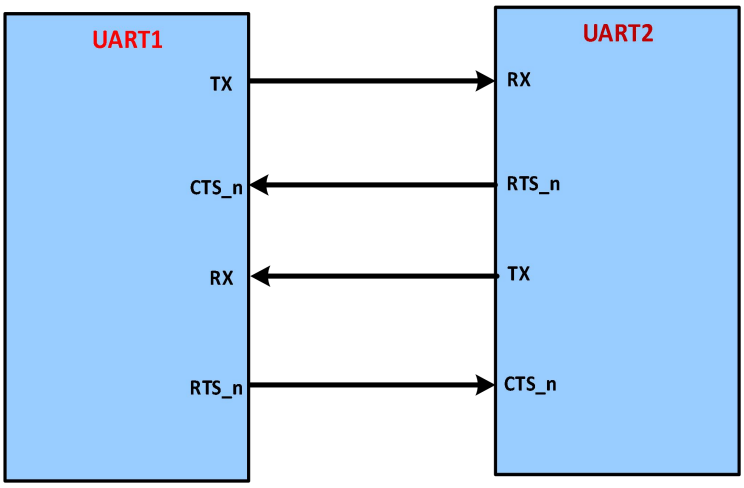


图 25-6 硬件流控制连接

在图 25-7 中，当 UART2 RX 数据寄存器或 FIFO 已满，UART2 的信号 RTS_n 会通知 UART1 不发送数据。当 UART2 RX 数据寄存器或 FIFO 通过读操作变为未满时， UART2 RTS_n 自动变为低电平。然后 UART1 可以通过检测 UART1 CTS_n 为低电平来传输数据。以类似的方式，UART1 RTS_n 和 UART2 CTS_n 用于控制 UART2 传输。

特别地，如果在 UART1 传输数据期间，UART1 CTS_n 变为高电平，则将完整发送当前数据。因此，用户通常应检查 RTS_n 或 CTS_n 状态以使用硬件流控制功能和其他状态位。

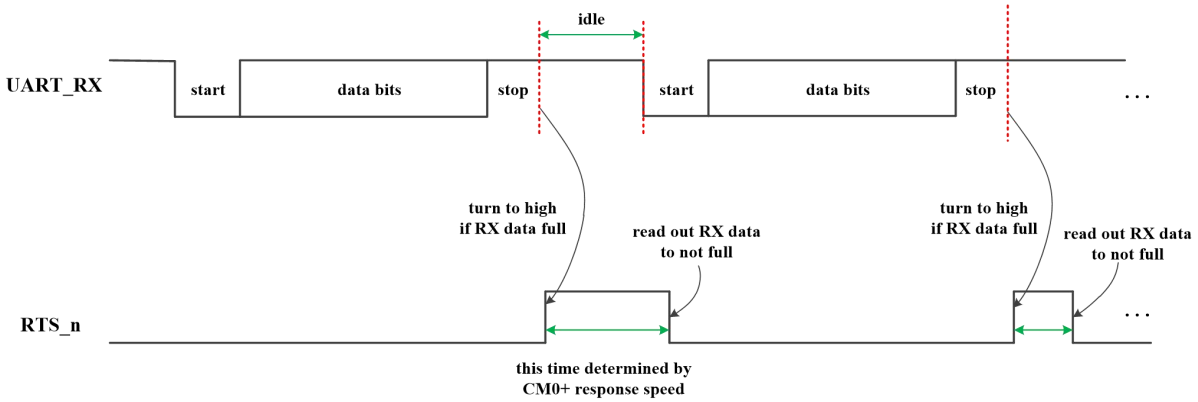


图 25-7 硬件流控制原理

25.4.6 RS485 功能

和 UART 功能相比，RS485 功能有一个自动方向控制信号 UART_RTS，如图 25-8 和图 25-9 所示。其在接收数据时默认为低电平，发送数据时默认为高电平。由于采用半双工，RS485 在同一时刻只可

© 2013 - 2024 杰发科技有限公司

以实现发送或接收操作中的一个。在图 25-8 中，有两个延迟（delay），delay1 用于在实际传输数据之前上拉 UART_RTS 信号，而 guard time 用于在数据位传输已经全部完成后下拉 UART_RTS 信号。实际的 PCB 布线延迟可能导致 UART_RTS 信号在 UART_TX 之后变为高电平，从而可能损坏第一个数据位。因此， 延迟有助于保证整个传输过程中 UART_RTS 为高电平。传输完成后，UART_RTS 将自动变为低电平，使 UART 处于接收状态。

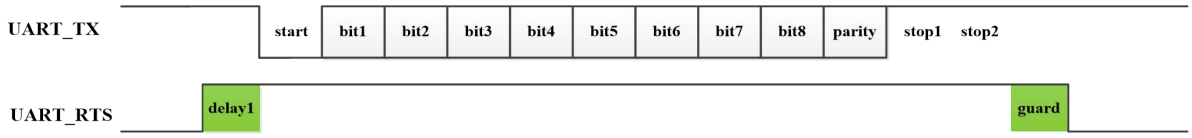


图 25-8 单字节数据传输

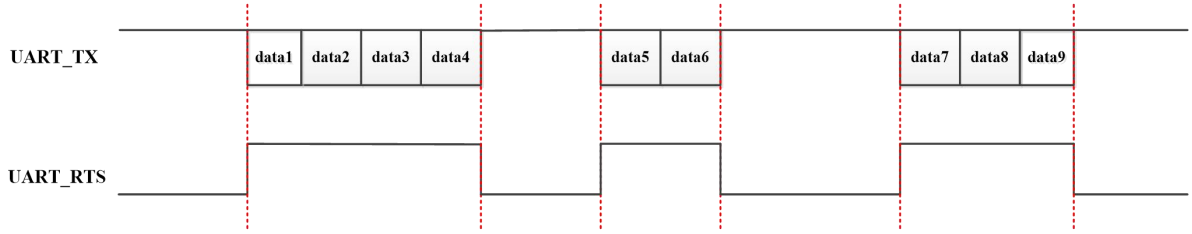


图 25-9 多字节数据传输

图 25-9 描述了用户应用连接的一个典型实例。UART_RTS 充当 MAX485 的方向控制信号。使用此连接方法，UART_RTS 默认为低电平，因此 MAX485 处于默认接收条件下。当 UART 想要传输数据时，UART_RTS 信号由硬件设置为高电平。

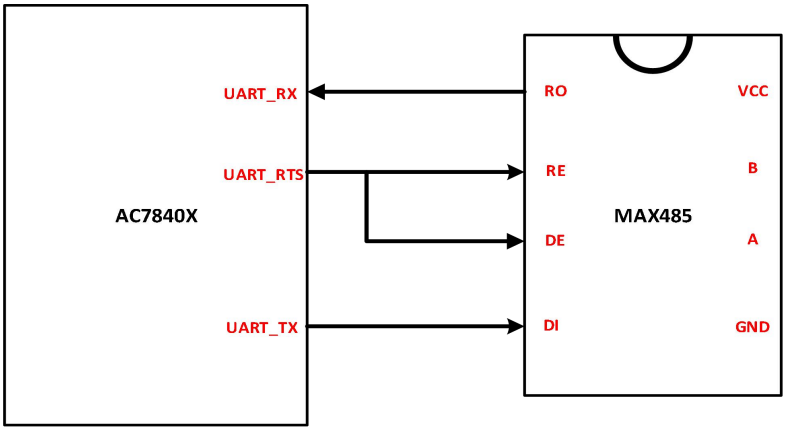


图 25-10 实际电路连接

25.4.7 LIN 功能

UART LIN 只是一个软件 LIN，具有发送同步间隔段（break field），同步段（synchronous field）和数据的功能。如图 25-11 所示。用户可以在最新的 LIN 协议中了解更多细节，该图仅描述了一个基本的帧。除基本 UART 寄存器外，用户还需关注 UART_LINCR 等寄存器。在 UART 模块中，存在一个硬件逻辑单元，用于 LIN 检测，并且当 LINEN 配置为 1 时，就启用 LIN 功能。

© 2013 - 2024 杰发科技有限公司

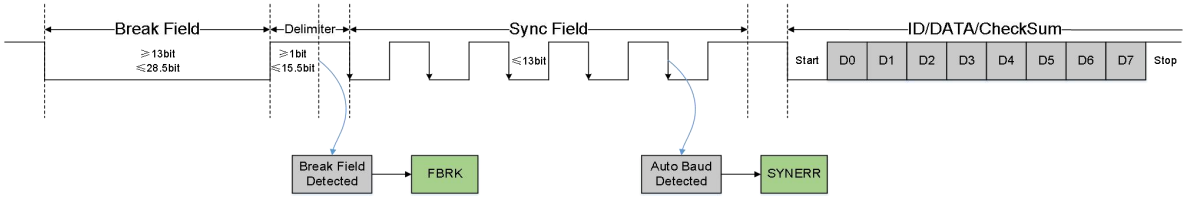


图 25- 11 LIN 帧流程

UART-LIN 帧流程如图 25- 11 所示，从机接收过程参考如下步骤：

1. 当 UART RX 接收到 10（LBRKDL=0）或 11 （LBRKDL=1）个位的低电平时，LIN 检测逻辑单元将其视为有效的 LIN 帧同步间隔段，LIN 同步间隔段标志 FBRK 由硬件设置为 1。需要指出的是，LIN 同步间隔段不是数据，也不存储在 UART 接收数据寄存器或 FIFO 中。
2. 在同步间隔段之后，是同步段即数据为 0x55。如果 LABAUDEN 配置为 1，则在同步段期间自动波特率同步检测开始运行，通过检测连续 5 个下降沿来完成如图 25- 11 所示。但数据 0x55 不会存储到 RX 数据寄存器或 FIFO 中。若 LABAUDEN 配置为 0，则不执行自动波特率检测，数据 0x55 存储在 RX 数据寄存器或 FIFO 中。波特率同步功能支持±14%的主机波特率偏差，否则可能会出现同步段错误 SYNERR。
3. 在同步段之后，数据都是由 UART 接收。

为避免在发生异常情况时模块暂停，引入超时机制，如图 25- 11 所示：

1. LIN 从机接收到的同步间隔段最大为 28.5 bits 时间，超过该值则认为是无效的同步间隔段。
2. 对于同步间隔段分隔符（Break delimiter）超过 15.5 bits 时间会导致模块重置接收器和 LIN 检测逻辑。
3. 如果同步段出错，硬件置位 UART_LSR1 的 SYNERR 位，如果使能 UART_LINCR 的 SYNERRIE 位，当同步出错时候，硬件产生同步段错误 SYNERR 中断。

UART-LIN 主机发送过程参考如下步骤：

1. 作为软件 LIN，如何发送同步间隔段是关键步骤。当用户想要发送同步间隔段时，用户需检查 UART_LSR0[TDNF]的状态。发送的同步间隔段长度取决于寄存器 UART_BRKLGH 的配置。

   寄存器 UART_BRKLH 的取值为 0-15 分别对应 13-28 bits 的同步间隔段。当 UART 处于空闲（IDLE）状态时，将 UART_LINCR[SDBRK]写入 1，立即发送同步间隔段。发送完成 SDBRK 将由硬件自动清 0。
2. 将同步段（0x55）和其他后续数据写入 UART_RBR/THR 数据寄存器并作为正常数据发送。

UART-LIN 休眠唤醒功能：

1. 用户可设置 UART_LINCR[LINSLP]为 1，使 LIN 处于休眠状态，该状态下不能接收数据；
2. 当 LIN 处于休眠状态下，UART RX 上收到大于 150μs 的低电平唤醒信号，LIN 将退出休眠状态，如果使能 UART_LINCR[LINWAKIE]，则会触发 UART_LSR1[LINWAK]中断，然后可以正常接收数据；用户也可以设置 LINSLP 为 0，使其退出休眠状态。

© 2013 - 2024 杰发科技有限公司

25.4.8 IRDA 功能

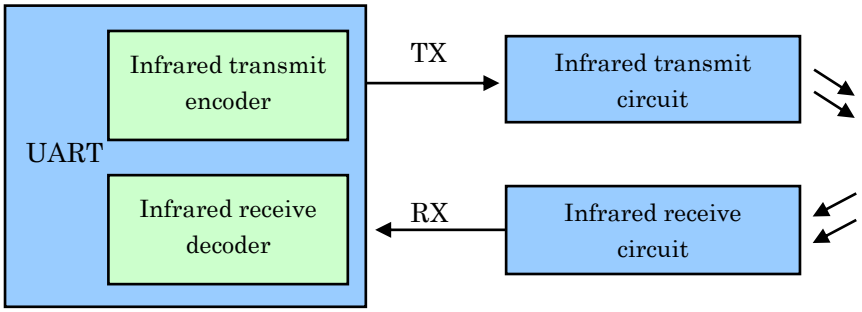


图 25-12 IRDA 发送和接收

UART 集成了红外发射编码和接收解码功能，然后通过外部红外传输器件发射 IRDA 格式的无线数据，以及将接收到的无线信号转换为 IRDA 格式。对于红外传输编码器，在 IrDA 模式下，假定数据模式为 8 位无奇偶校验一个停止位。数据通过信号 IrDA_out 输出，且 IrDA 工作于发送模式时，脉冲宽度固定为 3/16。“0”默认为该信号在中心为 3/16 位宽脉冲的低电平信号。“1”表示为该信号在整个周期持续为低的电平。因此，应在外部使用 IrDA_out 高来传输红外信号，图 25-13 为典型的红外传输波形。

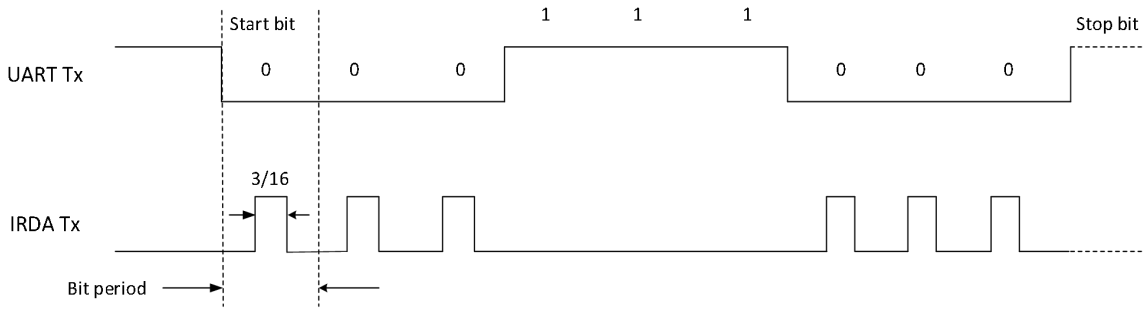


图 25-13 IRDA 发送编码

对于红外传输解码器，数据是通过信号输入的。“0”默认表示为该信号在中心为 3/16 位周期宽脉冲的（由 UART_PWDTH[PWDTH]控制，默认且只可配置 0，对应为 3/16）的高电平。“1”表示为整个周期持续为高的电平信号。因此使用外部红外探测器应指示无光为高电平，有光为低电平。脉冲的宽度是用 bclk 来测量的。最小有效值“0”可以编程到 UART_PMIN0[7:0]、UART_PMIN1[7:0]寄存器中。脉冲宽度值等于（SMP x DIV）的相应分数（接收端的 3/16 倍），图 25-14 为典型的红外接收波形。

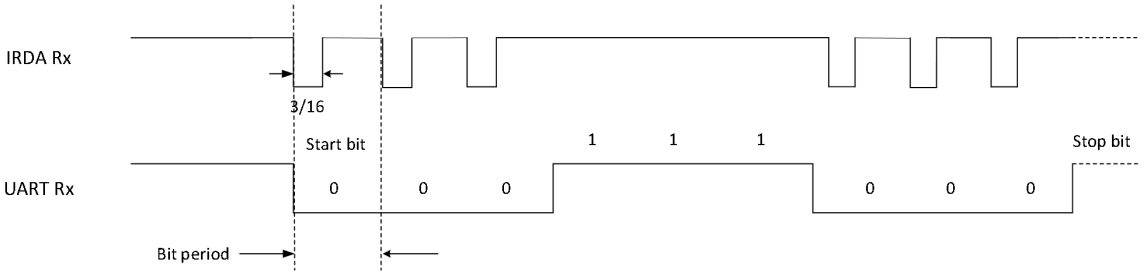


图 25-14 IRDA 接收解码

© 2013 - 2024 杰发科技有限公司

⚠注意：

- 打开 UART_LCR1[IREN]进入 IRDA 模式，即关闭奇偶校验功能、lin 等附加功能，只保留基本的 UART 功能。
- 为了避免编码器和解码器的干扰，IrDA 物理层只支持半双工红外通信链路的数据交换，因此在 IrDA 解码器或编码器工作时，应关闭其中对应 UART TX 或 RX enable 功能。

25.4.9 数据、地址匹配功能

数据匹配功能一般用于唤醒。当 DTMTCH_EN 设置时，UART RX 将检查接收到的数据是否与 UART_DATA[8:0]一致，如果 UART_MTCHCR[0]设置，将发生中断，写 UART_LSR1[DTMATCH]会清除中断。

地址匹配功能一般用于滤波。当 ADDTCH_EN 设置时，接收到的数据与 UART_ADDR[8:0]匹配，如果 UART_MTCHCR[1]设置，则匹配设置地址，随后的数据将被接收，直到下一个地址匹配失败。

0x85	0x01	0x02	0x03	0x86	0x04	0x05	0x06	0x85	0x07	0x08	0x09

在如上的 UART 数据流中，默认 UART_ADDR 为 0x85，数据宽度为 8。因为地址 0x85 匹配，UART 会接收数据 0x01、0x02、0x03。然后地址 0x86 不匹配，则数据 0x04、0x05、0x06 被过滤掉。当地址 0x85 再次匹配，UART 会接收数据 0x07、0x08、0x09。

25.4.10 中断请求

如表 25-6 所示，UART 模块共有 13 个中断源。

表 25-6 UART 中断汇总

中断源	标志	本地使能	全局使能
接收非空中断	DR	ERXNE	
发送非空中断	TDNF	ETXNF	
发送完成中断	TC	ETC	
奇偶校验错误中断	PE	EPE	
帧错误中断	FE	EFE	
噪声中断	NE	ENE	
溢出或分隔符错误中断	OE/BI	EOEBI	
CTS_n 状态切换中断	DCTS	EDCTS	

© 2013 - 2024 杰发科技有限公司

发送数据寄存器或发送缓存满中断	TXDF	ETXDF	
空闲线路中断	IDLE	IDLEIE	
LIN 唤醒中断	LINWAK	LINWAKIE	
LIN 同步段错误检测中断	SYNERR	SYNERRIE	
LIN 同步间隔段检测中断	FBRK	LBRKIE	

25.4.11 低功耗模式

表 25-7 UART 唤醒源

模式	唤醒源
STOP1	UART 接收中断、数据匹配中断
STOP2	UART 接收中断、数据匹配中断
VLPS	UART 接收中断、数据匹配中断

当 UART 处于接收中断或数据匹配模式时，进入 STOP1/STOP2/VLPS 模式的 MCU，接收到数据会唤醒 MCU 至正常模式。由于唤醒流程需要时间，UART 通常可以在 5ms 的唤醒总时间和 UART 重新配置的时间之后立即接收数据。此时 TX1 可以发送 0xFF，或者其他数据也可以，然后等待下降沿进入正常模式。在低功耗模式下，如果 UART 的功能时钟源是 HSI，它仍然是工作的。此时，UART 可以接收数据，且数据不会丢失。在某些情况下，这些数据非常重要，不能丢失。

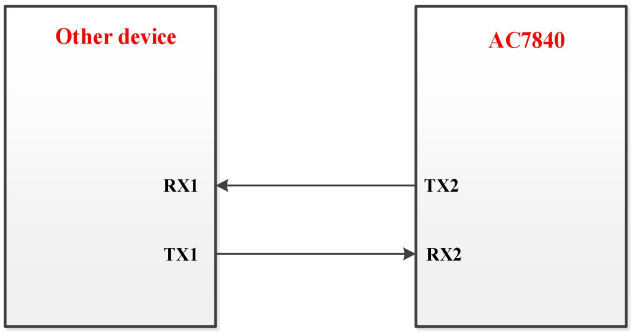


图 25-15 UART 通讯信号连接图

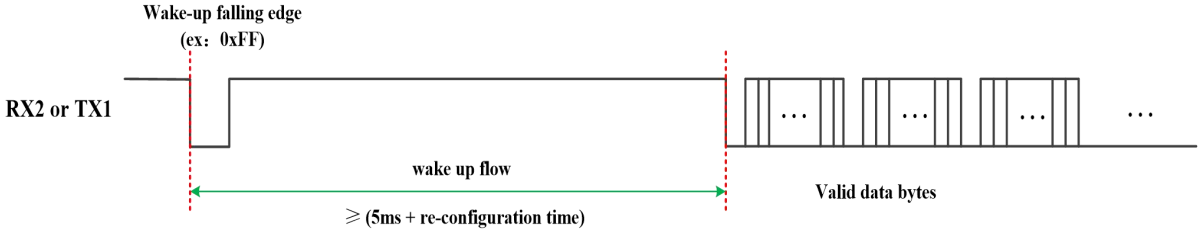


图 25-16 UART 唤醒芯片的典型流程

© 2013 - 2024 杰发科技有限公司

25.4.12 调试模式

支持 debug 模式控制位设置，debug 模式对 UART 模块运行无影响。

25.5 应用说明

25.5.1 波特率配置说明

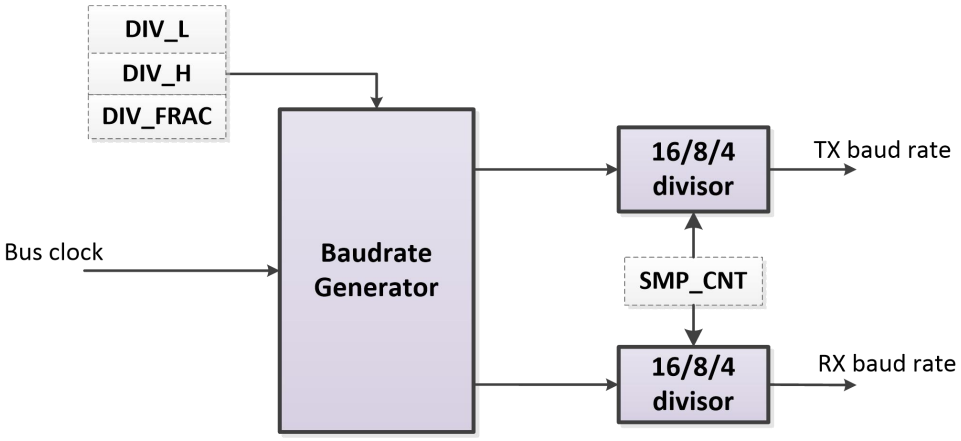


图 25-17 波特率发生器框图

如图 25-17 所示，波特率相关的配置寄存器如下所示：UART_SMP_CNT/ UART_DIV_H/ UART_DIV_L/ UART_DIV_FRAC。在寄存器映射表中描述了详细的信息。对于配置，如下公式用于波特率配置。

$$Baudrate = \frac{f_{clk}}{DIV * SMP_CNT}$$

在上面公式中， DIV = {UART_DIV_H, UART_DIV_L} ; SMP_CNT = UART_SMP_CNT。

例如：如果用户希望在 8 倍采样模式下以 50MHz 功能时钟频率获得波特率 230400 bps，则可以按如下方式获得 DIV 配置。因此， UART_DIV_L=13, UART_DIV_H=0, UART_DIV_FRAC=32 * 0.5634=18

$$DIV = \frac{50000000}{Baudrate*SMP\ CNT} = \frac{50000000}{230400*8} = 13.5634$$

结合上面给出的例子，可以清晰地解释 UART_DIV_FRAC[4：0] 。在上面的表达式中， DIV 不是整数。如果用户删除小数部分，波特率的准确度将降低。特别是在大波特率条件下，丢弃 DIV 小数部分可能会将精度降低到较低水平，这样正常的数据传输可能会出错。

© 2013 - 2024 杰发科技有限公司

为了保证高精度，UART_DIV_FRAC[4：0]配置为 DIV 小数部分。由于宽度为 5 位，UART n_DIV_FRAC 取值范围是 0 至 31。因此，32 乘以 DIV 分数生成 UART_DIV_FRAC [4：0]的配置值。

25.5.2 UART 配置说明

配置步骤：

1. TX/RX 数据存储模式： UART_FCR
2. 波特率： UART_DIV_L /UART_DIV_H/ UART_DIV_FRAC / UART_SMP_CNT

⚠**注意：**

1. DMA 功能必须在 FIFO 模式下才能起作用。
2. DMA 接收溢出时：
   如果认为溢出数据有效，使用读接收缓存接口可读取到溢出数据，重新初始化 DMA 并使能 DMA，DMA 只会搬运新接收数据；
   如果认为溢出数据无效，不使用读接收缓存接口会丢弃溢出数据，重新初始化 DMA 并使能 DMA，DMA 只会搬运新接收数据；

3. DMA： UART_DMA_EN

⚠**注意：**

务必注意 SB 位。

4. 数据格式： UART_LCR0/UART_LCR1

5. UART 软件 LIN 功能： UART_LINCR[LINEN]
6. RS485 通讯功能：UART_RS485CR[RS485EN]
7. 红外协议功能：UART_LCR1[IREN]

⚠**注意：**

这个步骤对不同功能来说是可选的。

8. 空闲模式检测功能：UART_IDLE

9. 中断使能： UART_IER
10. 匹配中断： UART_MTCHCR

© 2013 - 2024 杰发科技有限公司

11. 收发器使能：UART_LCR1[TXEN] / UART_LCR1[RXEN]

注意：

此步骤实际上在正常的发送或接收数据过程中。

12. 发送或接收数据：UART_RBR/THR[THR] /UART_RBR/THR[RBR]

说明：

1. 对于 LIN 功能，数据格式必须配置为 8 位，没有奇偶校验，16 次过采样。
2. 对于 LIN 功能，当 LABAUDEN = 0 时，将接收 sync field 数据（0x55）并将其存储到 FIFO 或 RX 寄存器中，当 LABAUDEN = 1 时，将接收 sync field 数据（0x55），且不存储到 FIFO 或 RX 寄存器中。
3. 对于 RS485 功能， RTS_n PIN 用做发送或接收方向控制信号。

25.6 寄存器定义

表 25- 8 UART 寄存器映射

UART0 基地址：0x40018000
UART1 基地址：0x40018200
UART2 基地址：0x40018400
UART3 基地址：0x40018600

地址	名称	宽度	描述
UARTx 基地址+0x00	UART_RBR/THR	32	TX 保持寄存器 /RX 缓冲区寄存器
UARTx 基地址+0x04	UART_DIV_L	32	分频器低 8 位
UARTx 基地址+0x08	UART_DIV_H	32	分频器高 8 位
UARTx 基地址+0x0C	UART_LCR0	32	UART 辅助控制寄存器 0
UARTx 基地址+0x10	UART_LCR1	32	UART 辅助控制寄存器 1
UARTx 基地址+0x14	UART_FCR	32	FIFO 控制寄存器
UARTx 基地址+0x18	UART_EFR	32	硬件流使能寄存器
UARTx 基地址+0x1C	UART_IER	32	中断使能寄存器
UARTx 基地址+0x20	UART_LSR0	32	状态寄存器 0
UARTx 基地址+0x24	UART_LSR1	32	状态寄存器 1
UARTx 基地址+0x28	UART_SMP_CNT	32	UART 采样计数寄存器
UARTx 基地址+0x2C	UART_ADDR	32	地址匹配寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
UARTx 基地址+0x30	UART_DATA	32	数据匹配寄存器
UARTx 基地址+0x34	UART_RS485_GUARD	32	保护时间（Guard time） 添加寄存器
UARTx 基地址+0x3C	UART_SLEEP_EN	32	休眠使能寄存器
UARTx 基地址+0x40	UART_DMA_EN	32	DMA 使能寄存器
UARTx 基地址+0x44	UART_DIV_FRAC	32	小数分频器寄存器
UARTx 基地址+0x48	UART_MTCHCR	32	匹配控制寄存器
UARTx 基地址+0x4C	UART_RS485CR	32	RS485 控制寄存器
UARTx 基地址+0x54	UART_CNTR	32	RS485 延迟时间
UARTx 基地址+0x58	UART_IDLE	32	空闲中断使能寄存器
UARTx 基地址+0x5C	UART_LINCR	32	软件 LIN 控制寄存器
UARTx 基地址+0x60	UART_BRKLGH	32	软件 LIN 同步间隔段控制寄存器
UARTx 基地址+0x64	UART_PMIN0	32	IrDA 模式接收端最小脉宽[7:0]
UARTx 基地址+0x68	UART_PMIN1	32	IrDA 模式发送端最小脉宽[15:8]
UARTx 基地址+0x6C	UART_PWDTH	32	IrDA 模式发送脉冲宽度

25.6.1 RX/TX 数据寄存器(UART_RBR/THR)

表 25-9 UART_RBR/THR 寄存器

UART_RBR/THR　　RX/TX 数据寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								RBR/THR[8:0]								
访问								RW								
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8：0 RBR/THR	**RX/TX 数据寄存器** 通过访问该寄存器可以读取接收到的数据，且发送数据可以写入该寄存器。数据长度不超过 9 位。

© 2013 - 2024 杰发科技有限公司

25.6.2 分频器低 8 位寄存器(UART_DIV_L)

表 25-10 UART_DIV_L 寄存器

UART_DIV_L 分频器低 8 位寄存器 复位值：0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DIV_L[7:0]							
访问									RW							
复位值									0	0	0	0	0	0	0	1

字段	说明
7：0 DIV_L	**波特率分频器** 分频器低 8 位

25.6.3 分频器高 8 位寄存器(UART_DIV_H)

表 25-11 UART_DIV_H 寄存器

UART_DIV_H 分频器高 8 位寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DIV_H[7:0]							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7：0 DIV_H	**波特率分频器** 分频器高 8 位

© 2013 - 2024 杰发科技有限公司

25.6.4 控制寄存器 0(UART_LCR0)

表 25- 12 UART_LCR0 寄存器

UART_LCR0 **控制寄存器 0** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SB	SP	EPS	PEN	STB	WLS1_WLS0[1:0]	
访问										RW	RW	RW	RW	RW	RW	
复位值										0	0	0	0	0	0	0

注意：

一定要将 SB 位配置为 0，否则 tx 在任何时候都发送 ‘0’。

字段	说明
6 SB	**设置 Break** 0：没有效果 1：SOUT 信号被强制进入 "0" 状态
5 SP	**奇偶校验位** 0：没有效果 1：根据 EPS 和 PEN 的状态，奇偶校验位强制进入自定义状态 如果 EPS = 1 & PEN = 1，奇偶校验位的设置(发送)值和检查值为 0. 如果 EPS = 0 & PEN = 1，奇偶校验位的设置(发送)值和检查值为 1.
4 EPS	**选择偶校验** 0：发送并检查奇校验 1：发送并检查偶校验
3 PEN	**使能奇偶校验** 0：不传输和检查奇偶性 1：传输和检查奇偶性
2 STB	**STOP 位个数** 0：始终添加一个 STOP 位 1：每个字符发送后添加两个 STOP 位；除非字符长度等于 7 时，添加 1 个 STOP 位
1：0	**选择字长**

© 2013 - 2024 杰发科技有限公司

字段	说明
WLS1_WLS0	
	00： 7 位
	01： 8 位
	10： 9 位
	11： 8 位（保留）

25.6.5 控制寄存器 1(UART_LCR1)

表 25- 13 UART_LCR1 寄存器

UART_LCR1 控制寄存器 1 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									INVTX	INVRX	IREN	LOOP	TXDIR	RSRC	TXEN	RXEN
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 INVTX	**反转 tx 输出，包括 idle， break， data 位，start 位， stop 位** 0：不反转 tx 输出 1：反转 tx 输出
6 INVRX	**反转 RX 输入，包括 idle， break， data 位，start 位， stop 位** 0： 不反转 rx 输入 1： 反转 rx 输入
5 IREN	**确定 是否适用 IrDA 编码合解码功能** 0： IrDA 模式关闭 1： IrDA 模式开启，如果 uart_tx 开启，UART 可以发送 IrDA 格式的数据；如果 uart_rx 开启，UART 可以接收 IrDA 格式的数据
4 LOOP	**循环** 0： 供用户正常使用 1： 控制 uart 进入循环模式(可以用来测试 uart 自身)
3 TXDIR	**TX 方向** 0： TXD 是单线输入模式 1： TXD 是单线输出模式

© 2013 - 2024 杰发科技有限公司

字段	说明
2 RSRC	**输入路径选择** 0：如果设置了 LOOPS, 选择内部环路模式，并且 LPUART 不使用 RXD 管脚 1：单线 LPUART 模式，TXD 管脚连接到发送端输出端和接收输入端
1 TXEN	**UART 发射器使能** 0： 禁用 1： 使能
0 RXEN	**UART 接收器使能** 0： 禁用 1： 使能

25.6.6 FIFO 控制寄存器(UART_FCR)

表 25-14 UART_FCR 寄存器

UART_FCR **FIFO 控制寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																FIFOE
访问																RW
复位值																0

字段	说明
0 FIFOE	**使能 FIFO** 0： 禁用 RX 和 TX FIFO 1： 使能 RX 和 TX FIFO

© 2013 - 2024 杰发科技有限公司

25.6.7 硬件流使能寄存器(UART_EFR)

表 25-15 UART_EFR 寄存器

UART_EFR 硬件流使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									AUTO_CTS	AUTO_RTS						
访问									RW	RW						
复位值									0	0						

说明：

AUTOCTS=1 表示使能 CTS_n 引脚的硬件流功能，因此如果 AUTOCTS=1，用户必须将 n_CTS 引脚连接到固定的电平，比如其它 MCU 或设备的引脚。如果 AUTOCTS=0，用户不需要关注 CTS_n 引脚。

字段	说明
7 AUTO_CTS	**使能硬件发送流程控制** 0：禁用 1：使能
6 AUTO_RTS	**使能硬件接收流程控制** 0： 禁用 1： 使能

© 2013 - 2024 杰发科技有限公司

25.6.8 中断使能寄存器(UART_IER)

表 25-16 UART_IER 寄存器

UART_IER 中断使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								ETXDF	EDCTS	EOEBI	ENE	EFE	EPE	ETC	ETXNF	ERXNE
访问								RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8 ETXDF	**发送寄存器或发送 FIFO 满中断使能** 0： 禁用 1： 使能
7 EDCTS	**CTS_n 变化中断使能** 0： 禁用 1： 使能
6 EOEBI	**溢出错误或分隔符错误中断使能** 0： 禁用 1： 使能
5 ENE	**噪声错误中断使能** 0： 禁用 1： 使能
4 EFE	**帧错误中断使能** 0： 禁用 1： 使能
3 EPE	**奇偶校验错误中断使能** 0： 禁用 1： 使能
2 ETC	**发送完成中断使能** 0： 禁用 1： 使能
1	**发送数据寄存器空中断使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
ETXNF	0： 禁用 1： 使能 **注意： fifoe=1 表示 fifo 为非满** **fifoe=0 表示数据寄存器为空**
0 ERXNE	**接收数据寄存器非空中断使能** 0： 禁用 1： 使能 **注意：fifoe=1 表示 fifo 非空** **fifoe=0 表示数据寄存器非空**

25.6.9 线路状态寄存器 0(UART_LSR0)

表 25- 17 UART_LSR0 寄存器

UART_LSR0 **线路状态寄存器 0** **复位值：0x00000020**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								TXDF	NE	TC	TDNF	BI	FE	PE	OE	DR
访问								RO	R/W 1C	RO	RO	R/W 1C	R/W 1C	R/W 1C	R/W 1C	R/W 1C
复位值								0	0	0	1	0	0	0	0	0

注意：
SUB NE/PE/FE 错误仅针对当前的数据，与此同时， OE/BI 将一直存在，直到其被清除。

字段	说明
8 TXDF	**发送数据寄存器或者发送 FIFO 满标志** 0：发送数据寄存器(fifoe=0)或者发送 FIFO (fifoe=1)不满 1：发送数据寄存器(fifoe=0)或者发送 FIFO (fifoe=1)满 **注意：该标志反映了数据和发送状态。**
7	**噪声错误标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
NE	0：不存在噪声错误 1： 存在噪声错误 **注意：写 1 将清除此标志为 0。**
6 TC	**传输完成标志** 0： TX FIFO(fifoe=1) 或 TX 寄存器(fifoe=0) 非空，或发送端还没有完成数据移位。 1： TX FIFO(fifoe=1) 或 TX 寄存器(fifoe=0)为空，发送端完成数据移位。 **注意：将数据写入 TX FIFO(fifoe=1)/TX 寄存器(fifoe=0) 以将清除此标志为 0。对于 LIN 功能，设置 SBRK 位也可以将此标志清除为 0。**
5 TDNF	**TX 数据寄存器或 TX FIFO 非空标志** 0： TX 数据寄存器不为空或 TX FIFO 内容满 1： TX 数据寄存器为空或 TX FIFO 内容为非满 **注意： 将数据写入 TX FIFO(fifoe=1)/TX 寄存器(fifoe=0)将此标志清除为 0。**
4 BI	**分隔符错误标志** 0：无分隔符错误 1：如果禁用 FIFO，只要 SIN 保持在 0 状态超过一个帧传输时间(START 位 + DATA 位 + PARITY 位+ STOP 位)时，该位被置位。 当中断发生时，只有一个零字符被加载到 FIFO 或 TX 保持寄存器中。 **注意：写 1 将此标志清除为 0.**
3 FE	**帧错误标志** 0：无帧错误 1：如果接收数据没有一个有效的 STOP 位， 该位被置位。 **注意：写 1 将此标志清除为 0.**
2 PE	**奇偶错误标志** 0： 无奇偶错误 1： 产生奇偶错误。没有接收到有效校验位，该位被置位。 **注意：写 1 将此标志清除为 0.**
1 OE	**溢出错误标志** 0：无接收溢出错误 1：如果禁用 FIFO 时，在 RX 移位寄存器的新数据覆盖先前数据之前，CPU 未读取 RX 缓冲区，则该位将置 1。如果使能 FIFO 时，在 RX FIFO 已满且 RX 移位寄存器变满时，会发生溢出错误。这个时候会置起 OE ，然后移位寄存器中的字符被覆盖，但不会传输到 FIFO。

© 2013 - 2024 杰发科技有限公司

字段	说明
	注意：写 1 将此标志清除为 0.
0 DR	**数据就绪标志** 0：由 CPU 读取 RX 缓冲区或读取所有 FIFO 字节来进行清除 1：由 RX 缓冲区变满或 RX FIFO 非空进行置位(至少有一个字节被传输到 FIFO 中)。

25.6.10 线路状态寄存器 1(UART_LSR1)

表 25- 18 UART_LSR1 寄存器

UART_LSR1 **线路状态寄存器 1** **复位值：0x000000E0**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							ADDMTCH	DDTMTCH	RTS	CTS	UART_IDLE	LINWAK	DCTS	FBRK	SYNERR	IDLE
访问							R/W 1C	R/W 1C	RO	RO	RO	R/W 1C	R/W 1C	R/W 1C	R/W 1C	R/W 1C
复位值							0	0	1	1	1	0	0	0	0	0

字段	说明
9 ADDMTCH	**地址匹配状态 – ADDMTCH** 0：地址匹配标志位为 0 1：地址匹配标志位为 1 **注意：写 1 将此标志清除为 0。**
8 DTMTCH	**数据匹配状态 – DTMTCH** 0：数据匹配标志位为 0 1：数据匹配标志位为 1 **注意：写 1 将此标志清除为 0。**
7 RTS	**硬件流程状态 – RTS** 0：在硬件流程控制功能下，它表示 RX FIFO 或 RX 寄存器已满。该信号可以通知其他设备不向 MCU 发送数据。 1：在硬件流程控制功能下，它表示 RX FIFO 或 RX 寄存器未满。

© 2013 - 2024 杰发科技有限公司

字段	说明
	注意： RTS 和 RTS_n 引脚信号相反，它不是一个中断源。
6 CTS	**硬件流程状态 – CTS** 0：在硬件流控制功能下，它表示 RX FIFO 或 RX 寄存器已满。该信号会通知 MCU 不发送下一个数据。 1：在硬件流程控制功能下，它表示其他设备的 RX FIFO 或 RX 寄存器未满。 **注意： CTS 和 CTS_n 引脚信号相反，它不是一个中断源。**
5 UART_IDLE	**UART_IDLE** 0： UART 正在工作中 1： UART 未工作，也就是说，发射器和接收器不工作、已完成数据传输或接收
4 LINWAK	**LIN 唤醒标志** 0： 还没接收到唤醒事件 1： LIN 唤醒事件已经发生，LIN 唤醒标志置位
3 DCTS	**CTS_n 引脚信号变化标志** 0： 未变化 1： 表示 CTS_n 引脚信号从 1 变为 0 或 0 变为 1 **注意：写 1 将此标志清除为 0.**
2 FBRK	**LIN 同步间隔段发生标志** 0：在 LIN 功能中没有检测到 LIN 帧中的 break 字段 1：在 LIN 功能中检测到 LIN 帧中的 break 字段 **注意：写 1 将此标志清除为 0.**
1 SYNERR	**LIN 同步段错误标志** 0： 没有错误 1： 存在错误 **注意：写 1 将此标志清除为 0**
0 IDLE	**IDLE 标志** 0：尚未检测到空闲线路 1：检测到空闲线路 接收器已经接收到数据，该数据后面为一个至少保持一个字节数据时间的高电平。IDLE 状态标志必须先使能总线空闲检测 ILEN 才起作用。 **注意：写 1 将此标志清除为 0.**

© 2013 - 2024 杰发科技有限公司

25.6.11 采样计数器寄存器(UART_SMP_CNT)

表 25-19 UART_SMP_CNT 寄存器

UART_SMP_CNT 采样计数器寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															SMP_CNT[1:0]	
访问															RW	
复位值															0	0

字段	说明
1：0 SMP_CNT	**UART 采样计数器** 00： 基于 16*baud_pulse, baud_rate = 系统时钟频率/16/{DIV_H, DIV_L} 01： 基于 8*baud_pulse, baud_rate =系统时钟频率/8/{ DIV_H, DIV_L } 10： 基于 4*baud_pulse, baud_rate =系统时钟频率/4/{ DIV_H, DIV_L } 11： 保留

25.6.12 UART 地址匹配寄存器(UART_ADDR)

表 25-20 UART_ADDR 寄存器

UART_ADDR 地址匹配寄存器 复位值：0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								UART_ADDR[8:0]								
访问								RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值								0	0	0	0	0	0	0	0	1

字段	说明
8：0 UART_ADDR	**地址匹配寄存器 – UART_ADDR** 地址匹配寄存器，用于地址匹配 **注意：如果实际数据长度较短，则只检查实际长度的 msb。例如，7 位数据 0x5c 输入（实**

© 2013 - 2024 杰发科技有限公司

字段	说明
	际数据为 0x1c，msb=1），UART_ADDR 为 0x9c，则进行地址匹配；当 8 位数据 0x9c 输入（实数据是 0x1c，msb=1），但 UART_ADDR 是 0x9c 时，将不会进行地址匹配。

25.6.13 UART 数据匹配寄存器(UART_DATA)

表 25- 21 UART_DATA 寄存器

UART_DATA　　数据匹配寄存器　　复位值：0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								UART_DATA[8:0]								
访问								RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值								0	0	0	0	0	0	0	0	1

字段	说明
8：0 UART_DATA	**匹配寄存器 – UART_DATA** 数据匹配寄存器，用于数据匹配 **注意：如果实际数据长度较短，则只检查实际长度位。例如，7 位数据 0x1c 输入，但 UART_DATA 为 0x9c，会进行数据匹配；当 9 位数据 0x11c 输入，但 UART_DATA 为 0x1c 时，将不会进行数据匹配。**

© 2013 - 2024 杰发科技有限公司

25.6.14 保护时间寄存器(UART_RS485_GUARD)

表 25-22 UART_RS485_GUARD 寄存器

UART_RS485_GUARD **保护时间寄存器** **复位值：0x0000000F**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												GUARD_EN	GUARD_CNT[3:0]			
访问												RW	RW			
复位值												0	1	1	1	1

注意：

添加保护时间有助于消除每个字节的累积误差， 因此，通过使用具有保护时间的小数分频器来提高波特率的准确性是很重要的。

字段	说明
4 GUARD_EN	**保护间隔时间添加使能信号** 0： 禁用 1： 使能
3：0 GUARD_CNT	**保护间隔计数值** 0000b：0 位保护时间 0001b：1 位保护时间 …… 1111b：16 位保护时间

© 2013 - 2024 杰发科技有限公司

25.6.15 休眠使能寄存器(UART_SLEEP_EN)

表 25-23 UART_SLEEP_EN 寄存器

UART_SLEEP_EN 休眠使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																SLEEP_EN
访问																RW
复位值																0

字段	说明
0 SLEEP_EN	**休眠功能使能** 0：不处理睡眠模式指示信号 1：当芯片进入休眠模式时，根据软件初始设置，激活硬件流程控制。当芯片唤醒时释放硬件流程

25.6.16 DMA 使能寄存器(UART_DMA_EN)

表 25-24 UART_DMA_EN 寄存器

UART_DMA_EN DMA 使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															TX_DMA_EN	RX_DMA_EN
访问															RW	RW
复位值															0	0

字段	说明
1 TX_DMA_EN	**TX_DMA 机制使能信号** 0： 在 TX 不使用 DMA 1： 当该寄存器使能时，在 TX 使用 DMA
0 RX_DMA_EN	**RX_DMA 机制使能信号**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：在 RX 不使用 DMA 1：当该寄存器使能时，在 RX 使用 DMA

25.6.17 小数分频器寄存器(UART_DIV_FRAC)

表 25-25 UART_DIV_FRAC 寄存器

UART_DIV_FRAC 小数分频器寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												DIV_FRAC[4:0]				
访问												RW				
复位值												0	0	0	0	0

字段	说明
4：0 DIV_FRAC	小数分频器 如果实际的分频器为 135.65，则 DIV_FRAC 为 0.65*32 = [20.8]=21 ，并且 DIV_L=135

25.6.18 UART 匹配控制寄存器(UART_MTCHCR)

表 25-26 UART_MTCHCR 寄存器

UART_MTCHCR_ UART 匹配控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														MTCHIE	ADDMTCH_EN	DTMTCH_EN
访问														RW	RW	RW
复位值														0	0	0

字段	说明
2 MTCHIE	匹配中断寄存器 – MTCHIE

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 禁用 1： 使能
1 ADDMTCH_EN	**地址匹配寄存器 – ADDMTCH_EN** 0： 没有影响 1： 打开地址匹配功能，仅当 MSB 为 1 且数据捕获时，地址才发生匹配
0 DTMTCH_EN	**数据匹配寄存器 – DTMTCH_EN** 0： 没有影响 1： 打开数据匹配功能，仅当数据捕获时就发生数据匹配

25.6.19 RS485 控制寄存器(UART_RS485CR)

表 25- 27 UART_RS485CR 寄存器

UART_RS485CR **RS485 控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									RS485EN		INVPOL	DLYEN				
访问									RW		RW	RW				
复位值									0		0	0				

注意：

- 发送前的延迟对应于 UART_CNTR ，发送后的延迟可以使用 UART_GUARD。
- RS485 功能使用 PIN RTS_n 作为发送或接收方向控制引脚。

字段	说明
7 RS485EN	0：禁用 rs485 模式 1：使能 rs485 模式
5 INVPOL	0： 不反转 rts_n 的极性 1： 反转 rts_n 的极性
4 DLYEN	在最后一个停止位和 **rts_n** 或 **dtr_n de-assertion DELAY** 之间插入 **DELAY**，对应于延迟计数器寄存器。

© 2013 - 2024 杰发科技有限公司

25.6.20 RS485 延迟时间寄存器(UART_ CNTR)

表 25- 28 UART_ CNTR 寄存器

UART_CNTR RS485 延迟时间寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CNTR[7:0]							
访问									RW							
复位值									0	0	0	0	0	0	0	0

位	说明
7：0 CNTR	**计数器** 0~255 位时间作为 RS485 模式下的时间延迟

25.6.21 空闲中断使能寄存器(UART_IDLE)

表 25- 29 UART_IDLE 寄存器

UART_IDLE 空闲中断使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									ILEN			IDLEIE				
访问									RW			RW				
复位值									0			0				

字段	说明
7 ILEN	**IDLE 使能** 0: 禁用 1: 使能
4 IDLEIE	**IDLE 中断使能** 0：禁用 1：使能

© 2013 - 2024 杰发科技有限公司

25.6.22 LIN 控制寄存器 (UART_LINCR)

表 25-30 UART_LINCR 寄存器

UART_LINCR **LIN 控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									LINEN	LBRKIE	LBRKDL	SDBRK	LABAUDEN	SYNERRIE	LINWAKIE	LINSLP
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 LINEN	**LIN 模式使能** 0： 禁用 1： 使能
6 LBRKIE	**LIN 同步间隔段检测中断使能** 0： 禁用 1： 使能
5 LBRKDL	**LIN 同步间隔段检测阀值** 0： 10 位 1： 11 位
4 SDBRK	**LIN 发送同步间隔段使能** 0： 禁用 1： 使能发送 Break。Break 长度由寄存器 BRKLGH 决定 由软件置位，并在 break 发送完成后由控制器内部硬件清 0。
3 LABAUDEN	**LIN 自动波特率同步使能** 0：0x55 不用于自动波特率检测 1：0x55 用于自动波特率检测
2 SYNERRIE	**LIN 同步段错误检测中断使能** 0：禁用同步段错误中断 1：使用同步段错误中断
1 LINWAKIE	**LIN 唤醒中断使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：禁用唤醒中断 1：使用唤醒中断
0 LINSP	**LIN 休眠** 0：（读 0）LIN 不在休眠状态或（写 0）退出休眠状态 1：（读 1）LIN 在休眠状态或（写 1）进入休眠状态 读此位表示 LIN 休眠状态，写此位可更改 LIN 休眠状态。

25.6.23 LIN 同步间隔段控制寄存器(UART_BRKLGH)

表 25-31 UART_BRKLGH 寄存器

UART_BRKLGH　　LIN 同步间隔段控制寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													BRKLGH			
访问													RW			
复位值													0	0	0	0

字段	说明
3：0 BRKLGH	**发送同步间隔段长度** 0000b：13bits 同步间隔段 0001b：14bits 同步间隔段 …… 1111b： 28bits 同步间隔段

© 2013 - 2024 杰发科技有限公司

25.6.24 UART 接收最小脉宽(UART_PMIN0)

表 25-32 UART_PMIN0 寄存器

UART_PMIN0 **UART 接收最小脉宽** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									PMIN0[7:0]							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明
7：0 PMIN0	**IRDA 模式最小接收脉冲宽度** 00000000b：PMIN 低 8 位 0bits 脉冲宽度 00000001b：PMIN 低 8 位 1bits 脉冲宽度 …… 11111111b：PMIN 低 8 位 255bits 脉冲宽度 ir_rxd 用于检测逻辑 0 的 uart_clk 周期数，必须为低的 uart\U clk 周期数，这个值应等于（（BDIV+1）x CD）的分数（默认接收端的 3/16）： 0-65535，最小脉冲宽度 = uart_clk 周期 x PMIN

25.6.25 UART 接收最小脉宽(UART_PMIN1)

表 25-33 UART_PMIN1 寄存器

UART_PMIN1 **UART 接收最小脉宽** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									PMIN1[7:0]							
访问									RW							
复位值									0	0	0	0	0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
7：0 PMIN1	**IRDA 模式接收最小脉冲宽度[7:0]** 00000000b：PMIN 高 8 位 0bits 脉冲宽度 00000001b：PMIN 高 8 位 1bits 脉冲宽度 …… 11111111b：PMIN 高 8 位 255bits 脉冲宽度 ir_rxd 用于检测逻辑 0 的 uart_clk 周期数，必须为低的 uart\U clk 周期数，这个值应等于（（BDIV+1）x CD）的分数（默认接收端的 3/16）： 0-65535，最小脉冲宽度 = uart_clk 周期 x PMIN

25.6.26 UART 传输最小脉宽(UART_PWDTH)

表 25-34 UART_PWDTH 寄存器

UART_PWDTH **UART 传输最小脉宽** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																PWDTH
访问																RW
复位值																0

字段	说明
0 PWDTH	**IRDA 模式传输最小脉冲宽度** 0b：(3/16) *UART 传输 bit 时间

© 2013 - 2024 杰发科技有限公司

26 控制器局域网（CAN）

26.1 简介

26.1.1 CAN 控制器

CAN-CTRL 控制器是一个基于 CAN 协议执行串行通信的串行通信控制器，符合 CAN 2.0B 规范和可变数据速率的 CAN FD 规范。

26.1.2 CAN 协议

CAN 通信按帧组织进行分类，共有两种类型的帧结构：标准帧和扩展帧。对于 CAN 2.0B，一帧最大传输数据为 8 个字节，而 CAN FD 一帧最多传输 64 个字节。

使用消息标识符完成数据寻址。在 CAN 网络中，只有一个节点必须传输具有特定标识符的消息。所有节点都接收所有消息，并且节点主控制器必须确定它是否有需要的消息标识符寻址。为了减少主控制器的负载，CAN 控制器可以使用接收过滤器。这些过滤器将所有接收的消息标识符与用户可选择的位配置进行比较。仅当消息通过接收过滤器时，它才会存储在接收缓冲区中并通知主控制器。

CAN 帧标识符也用于总线仲裁。当具有较高优先级标识符的消息由另一个 CAN 节点发送时，CAN 控制器停止发送具有低优先级标识符的消息。CAN 控制器自动尝试在下一个可能的发送位置重新发送已停止的消息。

CAN 2.0B 定义了高达 1Mbit/s 的数据比特率。CAN FD 没有固定限制，CAN FD 标准定义了比特率的切换。如果启用，帧有效载荷可以较高的速度传输，而帧头以较低的速度发送。

26.1.3 型号选择

不同的芯片系列支持不同的 CAN 模块数量和功能，如表 26-1 所示：

表 26-1 CAN 型号选择

芯片型号	序号	支持 CANFD
AC78406HGLA/ AC78406HFLA/	CAN0	√
	CAN1	√
AC78407HGLA/ AC78407HFLA/ AC78408HGLA/ AC78408HFLA/ AC78409HGLA/ AC78409HFLA	CAN2	√
AC78406YGLA/ AC78406LGLA/ AC78406YFLA/	CAN0	√
	CAN1	√
	CAN2	√

© 2013 - 2024 杰发科技有限公司

芯片型号	序号	支持 CANFD
AC78406LFLA/ AC78407YGLA/ AC78407LGLA/ AC78407LFLA/ AC78408YGLA/ AC78408LGLA/ AC78408YFLA/ AC78408LFLA/ AC78409YGLA/ AC78409LGLA/ AC78409LFLA	CAN3	√

26.2 特性

- 支持 CAN 规范
  - CAN 2.0B （最多 8 个字节的有效载荷，经 Bosch 参考模型验证）
  - CAN FD（最多 64 字节的有效载荷，ISO 11898-1:2015 或者非 ISO Bosch）
- 可编程的比特率
  - CAN 2.0B 支持最高 1Mbit/s
  - CAN FD 支持最高 8Mbit/s（受收发器和所选择的时钟源频率的限制）
- 可选择外部晶振时钟 HSEDIV2，系统时钟 SYS_CLK 或者 SPLLDIV2 时钟作为时钟源
- 可编程波特率预分频器（1 至 1/256）
- 1 个接收缓冲区，FIFO 深度为 13，支持 DMA 接收数据
- 两个发送缓冲区：
  - 主发送缓冲区（PTB）深度为 1
  - 次发送缓冲区（STB）深度为 6，可按 FIFO 或优先级模式决定数据帧发送的先后顺序
- 60 个可编程的接收过滤器，每个过滤器可单独设置 ID 和掩码
- 中止发送功能
- 发送 CAN FD 数据时，可配置收发器延迟补偿
- 扩展模式：
  - 单次发送模式（PTB 和 STB 都支持）
  - 监听模式
  - 环回模式（内部、外部）
  - 收发器待机模式

© 2013 - 2024 杰发科技有限公司

- 错误状态管理：
  - 捕获最后发生错误的类型和仲裁失败的位置
  - 可编程错误警告限制
  - 发送和接收错误计数器
  - 主动错误状态，被动错误状态，总线关闭状态，总线自动恢复
- 可配置的中断源
- 32 位的发送和接收时间戳，可选择外部时钟或位时钟
- 带集成式低通滤波器的唤醒功能
- CAN 数据存储带 ECC 保护

26.3 结构框图

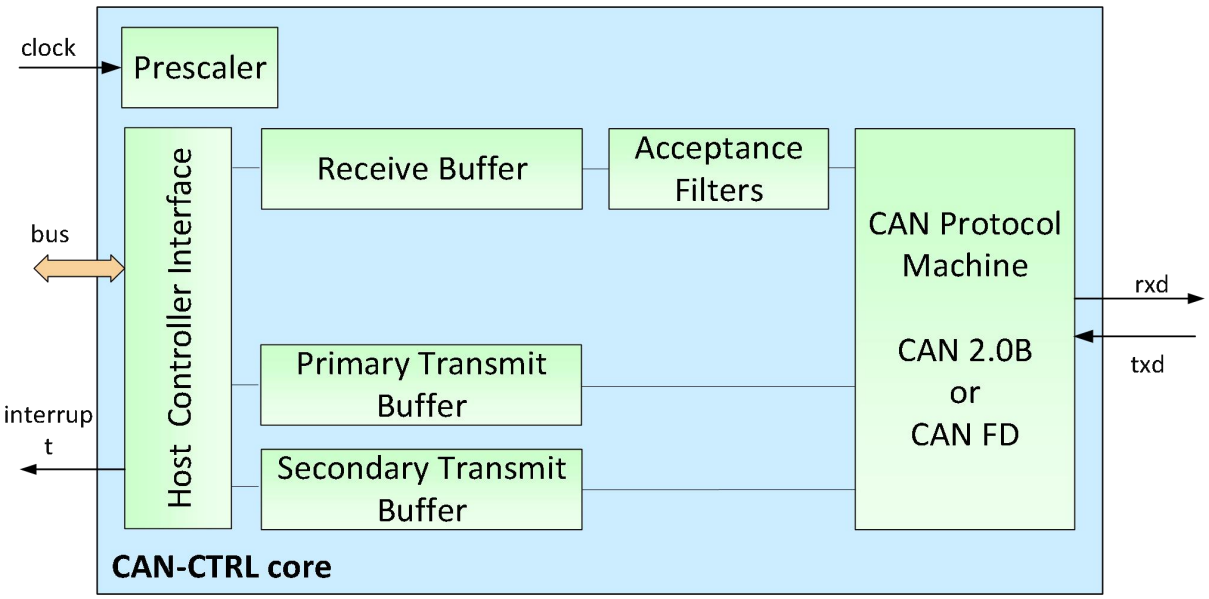


图 26- 1 CAN 控制器结构框图

© 2013 - 2024 杰发科技有限公司

26.4 功能描述

26.4.1 消息缓冲区

26.4.1.1 接收缓冲区

为了减少主控制器接收帧的负载，CAN 控制器使用接收过滤器。CAN 控制器接收数据时，接收过滤器会检查消息标识符。如果接收到的帧与其中一个接收过滤器的过滤条件相匹配，则它将被存储在接收缓冲区（RB）中，该接收缓冲区具有类似 FIFO 的行为。

根据可用消息缓冲区的数量，主控制器不需要立即读取传入消息。CAN 控制器能够在每个接收信息上生成中断。如果用户使能了对应的“满”中断使能寄存器 RFIF 或者“几乎满”寄存器 RAFIE，当接收缓冲区（RB）“已满”或“几乎满”时，CAN 控制器也会产生对应的中断。由于类似 FIFO 的行为，主控制器始终从 RB 读取最旧的消息。

表 26-2 接收缓冲区寄存器 RBUF – 标准格式（RO）

地址	位								功能
	7	6	5	4	3	2	1	0	
RBUF	ID(7:0)								标识符
RBUF+1						ID(10:8)			标识符
RBUF+2									标识符
RBUF+3	ESI								标识符
RBUF+4	IDE=0	RTR	FDF	BRS	DLC(3:0)				控制位
RBUF+5	KOER			TX					状态位
…									保留
RBUF+8	d1(7:0)								数据字节 1
RBUF+9	d2(7:0)								数据字节 2
…	…								…
RBUF+71	d64(7:0)								数据字节 64
RBUF+72	RTS(7:0)								接收时间戳
…	…								…
RBUF+75	RTS(31:24)								接收时间戳

表 26-3 接收缓冲区寄存器 RBUF – 扩展格式（RO）

地址	位								功能
	7	6	5	4	3	2	1	0	
RBUF	ID(7:0)								标识符
RBUF+1	ID(15:8)								标识符
RBUF+2	ID(23:16)								标识符
RBUF+3	ESI			ID(28:24)					标识符

© 2013 - 2024 杰发科技有限公司

地址	位								功能
RBUF+4	IDE=1	RTR	FDF	BRS	DLC(3:0)				控制位
RBUF+5	KOER			TX					状态位
…									保留
RBUF+8	d1(7:0)								数据字节 1
RBUF+9	d2(7:0)								数据字节 2
…	…								…
RBUF+71	d64(7:0)								数据字节 64
RBUF+72	RTS(7:0)								接收时间戳
…	…								…
RBUF+75	RTS(31:24)								接收时间戳

RBUF 寄存器（0x00 至 0x4f）指向的是接收缓存区 RB 中最早收到的消息。所有 RBUF 寄存器可以以任意顺序读取。

在 RBUF 中的 KOER 位和寄存器 CAN_ERRINFO 中的 KOER 意义相同。如果 RBALL = 1，RBUF 中的 KOER 才有意义。

如果使能了环回模式，且 CAN 控制器已经接收到了它自己发送的数据帧，则状态位 TX 会被置位为 1。

接收时间戳 RTS 被存储在消息的后面，因此和 TTS 相比较，RTS 存储位置和 RBUF slot 有关。

26.4.1.2 发送缓冲区

出于帧发送的需要，提供了两个发送缓冲区（TB）。主发送缓冲区（PTB）优先级较高，但只能缓冲一帧。次发送缓冲区（STB）优先级较低，它可以在 FIFO 或优先级模式下运行。PTB 和 STB 间的优先级是固定的，完全独立于 CAN 总线仲裁。总线仲裁是基于帧标识符的优先级来决定的。

可以请求 STB 发送一帧或所有帧。在 FIFO 模式下，首先发送此缓冲区内最旧的帧。在优先级模式中，首先发送在该缓冲区内具有最高优先级的帧（基于帧标识符）。

无论帧标识符如何，对于 CAN 控制器来说，位于 PTB 中的帧始终比 STB 中的帧具有更高的优先级。PTB 发送会停止并延迟 STB 发送。在成功发送 PTB 帧之后，STB 发送会自动重新启动。

如果 STB 已经启动发送且发送未完成，则 PTB 会在当前 STB 本帧发送完成后停止或者延迟 STB 发送。

在以下情况下，可以使用 PTB 发送打断 STB 发送：

1. 请求 STB 发送所有存储的帧，并且在完成 STB 所有帧发送前，主控制器请求 PTB 发送；
2. 请求 STB 发送单个帧，并且在 STB 使能发送前，主控制器请求 PTB 发送。

如果主控制器等待直到每个命令传输完成，那么它可以很容易地决定哪个缓冲区将传输下一帧。

表 26-4 发送缓冲区寄存器 TBUF – 标准格式（RW）

地址	位								功能
	7	6	5	4	3	2	1	0	

© 2013 - 2024 杰发科技有限公司

地址	位								功能
TBUF	ID(7:0)								标识符
TBUF+1						ID(10:8)			标识符
TBUF+2									标识符
TBUF+3	ESI	TTSEN							标识符
TBUF+4	IDE=0	RTR	FDF	BRS	DLC(3:0)				控制位
…									保留
TBUF+8	d1(7:0)								数据字节 1
TBUF+9	d2(7:0)								数据字节 2
…	…								…
TBUF+71	d64(7:0)								数据字节 64
TBUF+72	TTS(7:0) (RO)								发送时间戳
…	…								…
TBUF+75	TTS(31:24) (RO)								发送时间戳

表 26-5 发缓冲区寄存器 TBUF – 扩展格式（RW）

地址	位								功能
	7	6	5	4	3	2	1	0	
TBUF	ID(7:0)								标识符
TBUF+1	ID(15:8)								标识符
TBUF+2	ID(23:15)								标识符
TBUF+3	ESI	TTSEN		ID(28:24)					标识符
TBUF+4	IDE=1	RTR	FDF	BRS	DLC(3:0)				控制位
…									保留
TBUF+8	d1(7:0)								数据字节 1
TBUF+9	d2(7:0)								数据字节 2
…	…								…
TBUF+71	d64(7:0)								数据字节 64
TBUF+72	TTS(7:0) (RO)								发送时间戳
…	…								…
TBUF+75	TTS(31:24) (RO)								发送时间戳

如果 TBSEL = 1，TBUF 寄存器（0x50 至 0x97）指向 STB 中的下一个空消息缓冲区，否则指向 PTB。所有 TBUF 寄存器可以以任意顺序写入。对于 STB，需要设置 TSNEXT 以标记填充的缓冲区并跳转到下一个消息缓冲区。

TBUF 使用 32 位的存储器，因此写访问需要以 32bit 写入的方式执行。

RBUF 和 TBUF 都包含一些独立帧控制位（如表 26-6 所示）。对于 RBUF，这些位表示接收到的 CAN 帧相应 CAN 控制字段位的状态，而对于 TBUF，这些位为必须发送的帧选择适当的 CAN 控制字段位。

与存储每个接收帧的 RTS 相比，TTS 仅存储最后一次发送的帧。TTS 与实际选择的 TBUF slot 无关。

© 2013 - 2024 杰发科技有限公司

表 26-6 RBUF 和 TBUF 的控制位

名称	说明
IDE	**标识符扩展** 0：标准帧：ID(10：0) 1：扩展帧：ID(28：0)
RTR	**远程传输请求** 0：数据帧 1：远程帧 只有 CAN 2.0 帧可以是远程帧。 CAN FD 没有远程帧。 因此如果 TBUF 和 RBUF 中的 FDF = 1，则 RRS 位（对应的 CAN2.0 帧的 RTR 位）强制为 0。 当 FDF = 1 时即使 RRS = 1，接收节点仍可以正确接收到 CAN FD 帧。
FDF	**CAN FD 帧** 0：CAN 2.0 帧（每帧最多 8 个字节） 1：CAN FD 帧（每帧最多 64 个字节）
BRS	**位速率切换** 0：整个帧为正常/低速比特率 1：切换到数据/快速比特率，因此如果 FDF = 0，BRS 被强制转变为 0
DLC	**数据长度代码** 请参考表 26-7。
ESI	**错误状态指示器** 0：CAN 总线其它节点没有被动错误 1：CAN 总线其它节点有被动错误 在 RBUF 中为只读状态位，在 TBUF 中可读写； ESI 仅包含在 CAN FD 帧中，不存在于 CAN 2.0 帧中，ESI 位始终保持低； CAN FD 是否发送错误状态位，由被动错误状态 CAN_CTRL1[EPASS]和 TBUF 中 ESI 位共同决定，只要其中有一个为 1，则发送该状态位。
TTSEN	**发送时间戳使能** 0：不获取该帧的发送时间戳（TTS 值无效） 1：使能发送时间戳 在 TBUF 中，对于时间戳 TTS 可选择使能更新，并且需要先使能 TIMEEN。

RBUF 和 TBUF 中的数据长度代码（DLC）定义了有效载荷的长度即帧中有效载荷字节的数量。

远程帧（仅用于 FDF = 0 的 CAN 2.0 帧）总是以 0 个有效载荷字节发送，而 DLC 的内容在控制段中发送。因此，可以将一些信息编码到远程帧的 DLC 位中。但是，如果允许不同的 CAN 节点发送具有

© 2013 - 2024 杰发科技有限公司

相同 ID 的远程帧，则需要注意。在这种情况下，所有发射器都需要使用相同的 DLC，否则会导致无法解决的总线冲突。

表 26- 7 DLC 定义（基于 CAN 2.0 和 CAN FD 规范）

DLC（二进制）	帧类型	有效字节数
0000 - 1000b	CAN 2.0 和 CAN FD	0 - 8
1001 - 1111b	CAN 2.0	8
1001b	CAN FD	12
1010b	CAN FD	16
1011b	CAN FD	20
1100b	CAN FD	24
1101b	CAN FD	32
1110b	CAN FD	48
1111b	CAN FD	64

TBUF 寄存器可读写。 因此如果有需要时主控制器可以使用 TBUF 依次逐位地准备消息。

26.4.2 总线关闭状态

CAN_CTRL0 寄存器中的状态位 BUSOFF 用来标识“总线关闭”状态。如果 CAN 节点的发送错误计数器 CAN_ERRINFO[TECNT]超过 255，则该节点自动进入“总线关闭”状态。在恢复到主动错误状态之前，它不会参与进一步的通信。如果 CAN 节点接收到 128 组连续 11 个隐性位（恢复序列），则 CAN 节点自动恢复到主动错误状态。

在“总线关闭”状态下，CAN_ERRINFO[RECNT]用于计数恢复序列，而 TECNT 保持不变。请注意，在进入“总线关闭”状态时，TECNT 会溢出，可能变为一个较小的值 0 - 7。因此，建议将 BUSOFF 作为 TECNT 的第 8 位使用或者在节点进入“总线关闭”状态之前使用 TECNT。

进入总线关闭状态后，如需手动退出总线关闭状态，可通过模块复位（SRST_CANx）或者写 BUSOFF 为 0。如果节点从“总线关闭”恢复，则 RECNT 和 TECNT 将自动重置为 0。

如果一个帧正在等待传输，但 CAN 节点已进入总线断开状态，则该帧保持挂起状态。如果一个节点在总线关闭后返回主动错误状态，那么将尝试传输挂起的帧。如果不需要，则主控制器应中止该帧。

用户可配置寄存器 CAN_CTRL0[BOREC]为 1，来关闭自动恢复功能。在配置 BOREC 为 1 并进入总线关闭状态后，等待需要的时间，可再通过配置 BOREC 为 0 或 BUSOFF 为 0 来恢复到主动错误状态。注意，如使用配置 BOREC 为 0 来恢复，则还需要等待 128 组连续 11 个隐性位才能恢复；而配置 BUSOFF 为 0，则立即退出总线关闭状态。

26.4.3 接收过滤器

为了减少主控制器接收帧的负载，CAN 控制器使用接收过滤器过滤消息。因此，每个接收过滤器的长度是 29 位。

© 2013 - 2024 杰发科技有限公司

如果消息通过其中一个过滤器，则该消息会被接收且将其存储到接收缓冲区（RB）中，如果用户使能接收中断使能寄存器 CAN_CTRL1[RIE]，则此时 CAN_CTRL1[RIF]会被置位。如果该消息未被接收，则 RIF 不会被置位且 RB FIFO 指针不会增加。未被接收的消息将被丢弃并被下一条消息覆盖。任何未被接收的消息都不会覆盖已存储且有效的消息。

不论消息是否通过接收过滤器，CAN 控制器检查总线上的每条消息，并向总线发送消息应答或错误帧。

接收掩码定义了是否要进行比较的位，而接收代码定义了需要匹配的值。将掩码配置为 0 可以将所选择的接收码位与相应的消息标识符位进行比较。设置为 1 的掩码位不进行相应位的接收检查。

标识符位将与相应的接收代码位 ACODE 进行比较，如下所示：

- 标准：ID(10：0) 和 ACODE(10：0)
- 扩展：ID(28：0) 和 ACODE(28：0)

例如： 如果 AMASK_x(0) = 0 且所有其他 AMASK_x 位都为 1，那么最后一个 ID 位的值必须等于接收消息的 ACODE(0)的值，所有其他 ID 位会被过滤器忽略。表 26-8 为过滤器设置示例。

表 26-8 过滤器设置示例

CAN_ACF(ACODE)	CAN_ACF(AMASK)	通过 ID
0x0000 0000	0x1FFF FFFF	任意 ID
0x0000 0001	0x0000 0000	标准帧 0x001 或扩展帧 0x0000 0001
0x0000 0020	0x1FFF FF0F	标准帧 0xX2X 或扩展帧 0xXXXX XX2X
0x0000 0300	0x5FFF F0FF	标准帧 0x3XX
0x0004 0000	0x7FF0 FFFF	扩展帧 0xXXX4 XXXX

注意：

通过设置 AE_x = 0 来禁用滤波器会阻止接收消息。与此相反，禁用 AMASK_x 中的掩码位会禁用对此位的检查，这会导致接收消息。

仅 AMASK 和 ACODE 的定义不区分标准帧或扩展帧。如果位 AIDEE = 1，则接收 AIDE 定义的帧类型值。否则，如果 AIDEE = 0，则两种类型都会被接受。

上电复位或模块复位后，默认配置 CAN 控制器会接收所有消息（默认会设置 AE_0 = 1 使能过滤器 0，AMASK_0 的所有位都被置为 1 同时 AIDEE 置为 0。除过滤器 0 外所有其他过滤器都被禁用。过滤器 0 是唯一有定义 AMASK/ACODE 复位值的过滤器，而所有其他过滤器复位值则未定义）。

© 2013 - 2024 杰发科技有限公司

26.4.4 消息接收

接收数据被存储在接收缓冲区 RB 中，具有类似 FIFO 的行为。如果使能了 CAN_CTRL1[RIE]，则每个收到有效、可被接收的消息都会置位 CAN_CTRL1[RIF]，并根据填充状态设置 CAN_CTRL0[RSTAT]。当填充缓冲区的数量等于可编程值 CAN_CTRL1[AFWL]时，如果使能了 RAFIE，则置位 RAFIF。当所有缓冲区都已满，如果使能了 RFIE，则置位 RFIF。

RB 始终将包含最旧消息的消息 slot 映射到 RBUF 寄存器。RB 为每条消息提供 slot，并且主控制器在读取完成当前消息后，设置 CAN_CTRL0[RREL]以跳转到下一个 RB slot。可以按任意顺序读取实际 slot 的所有 RBUF 字节。

如果 RB 已满，则下一个传入的消息将被临时存储，直到它接收完成（第 6 个 EOF 位）。如果 CAN_CTRL0[ROM] 为 0，最旧的消息将被最新的消息所覆盖，或者如 ROM 为 1，则最新消息将被丢弃。在这两种情况下，如果使能了 ROIE，则 ROIF 会被置位。如果主控制器读取最旧的消息并在新传入的消息变为有效之前设置 RREL，则不会丢失任何消息。

26.4.5 处理消息接收

如果没有接收过滤器，CAN 控制器对接收到的每帧发出接收信号，并且主控制器会被要求决定它是否进行寻址，这会给主控制器带来很大的负担。

可以禁用中断并使用接收滤波器来减少主控制器的负载。一个基本操作为：如果接收中断寄存器 RIE 被使能且 CAN 控制器已收到有效消息，则 RIF 会被置位。为了减少接收中断的数量，可以使用 CAN_CTRL1[RAFIE/RAFIF]（RB 几乎满中断）或 RFIE/RFIF（RB 满中断）而不是 RIE/RIF（接收中断）。“几乎满限制”可使用 AFWL 进行编程。

RB 包含多个 RB slot，读取 RB 应按如下方式进行：

1. 使用 RBUF 寄存器从 RB FIFO 中读取最旧的消息；
2. 用 RREL = 1 释放 RB slot。这将选择下一条消息（下一个 FIFO slot），RBUF 会被自动更新；
3. 重复以上动作直到 CAN_CTRL0[RSTAT] 发出空 RB 信号。

如果 RB FIFO 已满并且新接收的消息被识别为有效（第 6 个 EOF 位），则将丢失一条消息（参见 ROM 位）。在此之前，不会丢失任何消息。应该给出足够的时间让主控制器在 RB FIFO 已满并且发生所选中断之后从 RB 读取至少一帧。为了实现这种行为，RB 包括比最大 slot 数量多一个隐藏 slot。此隐藏 slot 用于接收消息，验证该消息并在溢出发生之前检查它是否与接收过滤器相匹配。

26.4.6 DMA 接收

为减少收到数据后，从 RBUF 中读取数据到 SRAM 中的时间，CAN 模块支持将填充到 RBUF 中数据自动通过 DMA，搬运到 SRAM 中。初始化完成 DMA 接收后，当 RBUF 中接收到数据时，会自动触发 DMA 搬运数据。用户可根据需要，设置 DMA 传输的目标地址和数据长度，DMA 搬运完成设置的数据长度后，将自动释放该缓冲区。使能 DMA 后，CAN 消息接收完成后，也会置位 RIF，并触发相

© 2013 - 2024 杰发科技有限公司

应中断，需要注意，收到该中断后，软件不能释放缓冲区 CAN_CTRL0[RREL]，否则可能会丢失数据。当 DMA 接收使能时，软件驱动中默认已关闭 RIE 的中断使能。

26.4.7 消息发送

开始发送之前，必须向至少一个发送缓冲区（PTB 或 STB）加载消息。TBUF 寄存器提供对 PTB 和 STB 的访问。

推荐的编程流程如下：

1. 将 CAN_CTRL0[TBSEL]设置为需要的值以选择 PTB 或 STB；
2. 将帧写入 TBUF 寄存器；
3. 对于 STB，设置 CAN_CTRL0[TSNEXT] = 1 以完成该 STB slot 的加载。

CAN 2.0 消息的最大有效载荷长度为 8 个字节，CAN FD 的消息最大为 64 个字节。每条消息的长度由 DLC 定义。对于远程帧（RTR 位），DLC 的值没有作用，因为远程帧的数据长度始终为 0 字节。主控制器需要设置 TSNEXT 以跳转到下一个 STB slot。所有 TBUF 字节都可以按任意顺序写入。

如果 TBSEL = 0 时选择 PTB，则设置 TSNEXT = 1 没有作用。在这种情况下，TSNEXT 会被自动清除并且不会有任何影响。

使能 CAN_CTRL0[TPE]将发送 PTB 中的消息。而要发送 STB 中的消息，需使能 CAN_CTRL0[TSONE]来发送单个消息或 CAN_CTRL0[TSALL]来发送所有消息。

PTB 的优先级始终比 STB 高。如果 PTB 和 STB 同时使能发送，则无论帧标识符如何，都将始终先发送 PTB 消息。如果来自 STB 的发送已经激活，则在来自 PTB 的消息在下一个可能的发送位置（下一个帧 slot）发送之前完成。在 PTB 发送完成或中止之后，CAN 控制器返回以处理来自 STB 其他暂停的消息。

当发送完成后，会设置如下发送中断：

- 对于 PTB，如果使能了 CAN_CTRL1[TPIE]，则会置位 CAN_CTRL1[TPIF]；
- 对于使用 TSONE 的 STB，如果一条消息已完成且已使能 CAN_CTRL1[TSIE]，则会置位 CAN_CTRL1[TSIF]；
- 对于使用 TSALL 的 STB，如果所有消息已完成发送且已使能 TSIE，则会置位 TSIF。换言之，如果 STB 为空，则 TSIE 会被置位。因此，如果主控制器在 TSALL 发送开始后向 STB 写入附加消息，则此附加消息在 TSIF 置位之前也会被发送。

当 STB 为空时，设置 TSONE 或 TSALL 是无作用的。在这种情形下，TSONE 和 TSALL 都将会自动复位。不会置位中断标志，也不会发送帧。

26.4.8 消息发送中止

由于其低优先级而无法发送缓冲区中的消息，这将长时间阻塞缓冲区。为了避免这种情况，如果尚未开始发送或仲裁失利后，则主控制器可以通过分别设置 CAN_CTRL0[TPA]或 CAN_CTRL0[TSA]来中止

© 2013 - 2024 杰发科技有限公司

发送请求。当中止请求成功后，TPA 和 TSA 只提供单个中断标志 CAN_CTRL1[AIF]。CAN 控制器只有在不向 CAN 总线发送任何内容时才执行中止。因此，以下行为是有效的：

- 总线仲裁期间不能中止
  - 如果节点仲裁失利，则在此之后将执行中止；
  - 如果节点赢得仲裁，则发送该帧。
- 发送帧时不能中止
  - 如果成功发送帧，则通知主控制器发送成功。在这种情况下，不会发出中止信号；
  - 在 CAN 节点未接收到应答的发送失败之后，错误计数器递增并且将执行中止；
  - 当主控制器使能发送所有的帧时（TSALL = 1），如果 STB 中剩余至少一帧，则完成发送的帧以及中止的帧都会通知主控制器。

基于以上，中止此发送可能需要一些时间，具体取决于 CAN 通信速度和帧长度。如果执行中止，则会导致以下操作：

- TPA 释放 PTB，导致 TPE = 0；

  在释放 PTB 之后，帧数据仍存储在 PTB 中；
- TSA 释放 STB 的单个或所有消息 slot，这取决于是使用 TSONE 还是 TSALL 来启动发送。TSSTAT 会相应地进行更新。释放 STB 中的帧会导致丢弃帧数据，因为主控制器无法访问它。

不建议同时设置 TPA 和 TSA 。如果主控制器仍然决定这样做，则将置位 AIF，并且如果可能的话，将中止来自 PTB 和 STB 的所有发送。如前所述，如果在可以执行中止之前完成一次发送，则这将导致信号发送成功。因此，如果使能，可以设置以下中断标志：

- AIF (设置一次，用于 PTB 和 STB 发送中止)
- TPIF + AIF
- TSIF + AIF
- TPIF + TSIF (很少，只有在主控制器没有立即处理 TPIF 时才会发生)
- TPIF + TSIF + AIF (很少，只有在主控制器没有立即处理 TPIF 和 TSIF 时才会发生)

要清除整个 STB，需要设置 TSALL 和 TSA。为了检测由于失去仲裁而无法长时间发送消息，主控制器可以使用 ALIF/ALIE。

26.4.9 STB 满

在向 STB 写入一条消息之后，CAN_CTRL0[TSNEXT] = 1 标记填充的缓冲 slot 并跳转到下一个空闲消息 slot。此操作后，CAN 控制器会自动将 TSNEXT 复位为 0。如果最后一个消息 slot 已被填满，因此所有消息 slot 都被占用，则 TSNEXT 保持置位状态，直到新的消息 slot 空闲为止。当 TSNEXT = 1 时，CAN 控制器会阻止写至 TBUF。

© 2013 - 2024 杰发科技有限公司

当 slot 空闲时，CAN 控制器会自动将 TSNEXT 重置为 0。如果来自 STB 的帧成功发送或者主控制器请求中止（TSA = 1），则 slot 会变为空闲。如果 TSALL 发送中止，则 TSNEXT 也会复位，但整个 STB 会被标记为空。

26.4.10 错误处理

在 CAN 总线通信时，会发生通讯错误。如下功能支持检测和分析错误。

26.4.10.1 可编程错误警告限制

接收或发送期间的错误由 CAN_ERRINFO[RECNT] 和 CAN_ERRINFO[TECNT] 来计数。CAN_CTRL1 寄存器中的可编程错误警告限制 EWL 可由主控制器灵活配置以用于响应接收/发送错误警告事件。可以从 8 到 128 中每 8 个错误一步选择限制值：

错误计数限制 = (EWL + 1) * 8

如果在以下条件下有 CAN_CTRL1[EIE]使能，则会触发错误中断 CAN_CTRL1[EIF]：

- RECNT 或 TECNT 在错误警告限制的边界发生任一方向上的变化（即超越警告限制或减到警告限制以下）；
- BUSOFF 位发生变化（即进入或退出 BUSOFF 状态）。

26.4.10.2 仲裁失利

CAN 控制器能够检测仲裁段中仲裁失利事件。如果使能了 CAN_CTRL1[ALIE]中断，则可以通过 CAN_CTRL1[ALIF]中断检测仲裁失利事件，并且 CAN_ERRINFO[ALC]可以指示失利在仲裁段中的位置。如果此节点能够赢得此仲裁，则 ALC 的值会保持不变。ALC 保持最后一次仲裁失利的值。

ALC 的值定义如下：一帧以 SOF 位开始，发送 ID 的第 1 位。第一个 ID 位的 ALC 值为 0，第二个 ID 位的 ALC 值为 1，依此类推。仲裁仅允许在仲裁域中进行。因此，ALC 的最大值为 31，是扩展帧的 RTR 位。

如果标准远程帧与扩展帧进行仲裁，则扩展帧会在 IDE 位失去仲裁，ALC 将为 12。发送标准远程帧的节点将不会注意到已经发生了的仲裁，因为该节点已经获胜。在仲裁域之外不可能获得仲裁失利，这样的事件为位错误。

26.4.10.3 错误类型

CAN 控制器能检测 CAN 总线上的错误，并将最后一个错误事件存储在 CAN_ERRINFO[KOER]位中。如果使能 CAN_CTRL1[BEIE]中断，则可以通过 CAN_CTRL1[BEIF]中断检测 CAN 总线错误信号。每个新的错误事件都会覆盖之前存储的 KOER 值。因此，主控制器必须对错误事件做出快速反应。

© 2013 - 2024 杰发科技有限公司

26.4.10.4 接收所有的数据帧

如果 CAN_CTRL0[RBALL] = 1，则所有接收到的数据帧都将被存储，即使有错误的数据帧也会被存储，该功能一般用来做测试。这也适用于环回模式。仅数据帧存储在 RBUF 中，错误帧或过载帧不会被存储。

如果启用了时间戳记（CAN_ACFCTRL0[TIMEEN] = 1）并且为 EOF 配置了时间戳记（TIMEPOS = 1），则在出现错误的情况下，将在错误帧的开始处获取时间戳。

仅当节点是帧的发送方时，大多数可能的错误才会发生。在这种情况下如果激活了环回模式，则仅将其存储在 RBUF 中。根据错误部分的类型，当其他部分未知时，存储在 RBUF slot 中的帧可能有效。如在接收数据中，帧错误的帧数据可能是有效的。

26.4.11 扩展功能

26.4.11.1 单次发送

有时候发送失败或者仲裁失利后，不需要自动重发。因此，可以通过 CAN_CTRL0[TPSS]位对发送缓冲区 PTB、通过 CAN_CTRL0[TSSS]位对发送缓冲区 STB 分别设置发送一次消息的命令。在这种情况下，如果所选发送有效，则在发生错误或仲裁失利的情况下不会执行重发。

在立即发送成功的情况下，与正常发送没有区别。但是在发送失败的情况下，会出现如下问题：

- 如果使能，则 TPIF 会被置位，相应的发送缓冲区 slot 会被清除；
- 如果出现错误，会更新 KOER 和错误计数器。如果使能 BEIE，BEIF 会被置位，其他错误中断标志将相应地起作用；
- 如果仲裁失利，当使能 ALIE 时，会置位 ALIF。

因此，对于单次发送，TPIF 自身并不指示帧是否已成功发送。因此，单次发送应仅与 BEIF 和 ALIF 一起使用。

如果单次发送使用 TSALL，并且 STB 中存在多个帧，则对于每个帧，进行单次发送。不管是否未成功发送任何帧（例如：由于应答错误），CAN 控制器跳转到下一帧，直到 STB 为空则停止。

因此，在这种场景下，只有错误计数器指示发生了什么。如果两个帧中的一个出现错误，则主控制器无法检测哪个是成功的帧。

如果总线被另一个帧占用，如果单次发送开始，则 CAN 控制器会一直等待直到总线空闲，然后尝试发送单次帧。

26.4.11.2 监听模式

监听模式提供监控 CAN 总线的能力，而不会对总线产生任何影响，通过设置 CAN_CTRL0[LOM] = 1 开启。内部连接如图 26- 2 所示。

© 2013 - 2024 杰发科技有限公司

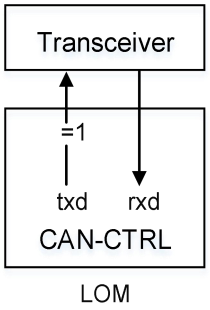


图 26- 2 监听模式

另一个应用是自动比特率检测，其中主控制器尝试不同的定时设置，直到没有错误发生。

在 LOM 中监视错误 （KOER，BEIF）。

在 LOM 中，控制器无法将显性位写入总线（没有主动错误标志或过载标志，没有应答）。需要使用如下规则完成：

- 在 LOM 中，控制器就好像处于错误被动模式，其中仅生成隐性错误标志。只有控制器就像处于错误被动模式一样。不触及包括状态寄存器在内的所有其他组件；
- 在 LOM 中，控制器不生成显性应答；
- 不管什么错误，错误计数器保持不变。

关于 LOM 的应答：

- 如果帧由一个节点发送，则仅当至少一个不在 LOM 状态的附加节点连接到总线时，才会生成在总线上可见的应答。这样就不会出现错误，所有节点（也包括 LOM 中的节点）都会收到该帧；
- 如果在应答错误之后存在主动或被动错误标志，则 LOM 中的节点能够将其检测为应答错误。

需注意当发送处于活动状态时，不应使能监听模式。如果使能 LOM ，则无法开始发送。

26.4.11.3 环回模式 （LBMI 和 LBME）

CAN 控制器支持两种环回模式：内部环回（CAN_CTRL0[LBMI] ）和外部环回（CAN_CTRL0[LBME]）。两种模式都可以接收自己发送的帧，便于自测。内部连接如图 26- 3 所示。

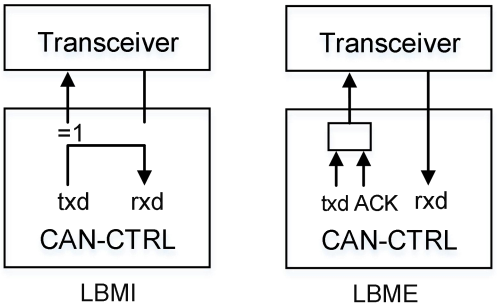


图 26- 3 环回模式

© 2013 - 2024 杰发科技有限公司

在 LBMI 模式下，CAN 控制器的 txd 和 rxd 与 CAN 收发器断开连接，CAN_TX 引脚输出为隐性电平。输出数据流在内部反馈到输入。在 LBMI 模式下，节点生成自应答信息以避免应答错误。

在 LBME 模式下，CAN 控制器的 txd 和 rxd 保持与收发器的连接，并且在总线上可以看到发送的帧。在收发器的帮助下，CAN 控制器接收自己发送的帧。在 LBME 模式下，当 CAN_CTRL0[SACK] = 0 时节点不产生自应答信号。如果 SACK = 1 时节点产生应答信号。因此，在 LBME 模式下，帧发送有两种可能的结果：

1. 另外一个节点也接收该帧，并产生应答信号。这会导致一次成功的发送和接收；
2. 没有其他节点连接到总线，这会导致应答错误。为了避免重传并增加错误计数，如果不知道其他节点是否连接到总线，可以使能 SACK 或使能 TPSS 或 TSSS。

在环回模式下，控制器接收其自身的消息，将其存储在 RBUF 中，并在使能时置位适当的发送和接收中断标志。

LBMI 可用于芯片内部和软件测试，而 LBME 可以测试收发器及其连接。

需注意当发送处于活动状态时，两种环回模式均不能使能。

LBMI 模式切换回正常操作不能通过简单地将 LBMI 设置为 0 来完成，应将 RESET 位设置为 1 来切换回正常操作，这时 LBMI 自动清零。然后再禁用 RESET 来使控制器返回至正常操作。与此相反，LBME 可以设置为 0 来退出该模式。

26.4.11.4 总线连接测试

要检测节点是否连接至总线，可以采用如下测试步骤：

- 发送一帧。如果节点连接到总线，则其 TX 位在其 RX 输入上是可见的；
- 如果有其他节点连接到 CAN 总线，则预期会发送成功(包括来自其他节点的应答)，不会发出任何错误信号；
- 如果该节点是唯一一个连接到 CAN 总线的节点（但总线、收发器和 CAN 控制器之间的连接正常）。由于没有来自其他节点的应答信息，第一个常规错误发生在应答段中。如果使能且 KOER = 100b（应答错误），则会产生 BEIF 错误中断；
- 如果与收发器或总线间的连接断开，则在 SOF 位之后立即置位 BEIF 错误中断并且 KOER = 001b（位错误）。

26.4.11.5 收发器待机模式

使用寄存器 CAN_CTRL0[STBY]，可以驱动输出待机信号。它可用于激活收发器的待机模式。

一旦待机模式被激活，无法进行发送，因此无法设置 TPE，TSONE 和 TSALL。另一方面，如果发送已经激活（设置了 TPE，TSONE 或 TSALL），CAN 控制器不允许设置 STBY。

如果已经置位 STBY， 收发器进入低功耗模式。在此模式下，无法以全速接收帧，但会监视 CAN 总线的显性状态。如果显性状态在收发器数据手册中定义的时间内有效，则收发器会将 rxd 信号拉低。如果

© 2013 - 2024 杰发科技有限公司

rxd 为低电平，CAN 控制器会自动将 STBY 清零，从而禁用收发器的待机模式。这是在没有中断到主控制器的情况下完成的。

从待机模式切换到活动模式对收发器需要一些时间，因此无法成功接收初始唤醒帧。因此，最近处于待机状态的节点不会发送应答信号。如果总线上的 CAN 节点没有应答唤醒帧，这会导致唤醒帧的发送器应答错误。然后发射器将自动重复该帧。在此重复期间，收发器将返回活动模式，CAN 控制器将接收此帧并将做应答。

总之，一个节点发送用于唤醒的帧。如果所有其他节点处于待机模式，发送器会收到应答错误并自动重复该帧。在重复的过程中，节点返回活动模式并将做应答。

STBY 不受 CAN_CTRL0[RESET]位的影响。

26.4.12 时间戳

CAN 控制器中的时间戳（time stamp）包括发送帧时间戳和接收帧时间戳（32 位）。当接收数据或者发送数据时，CAN 控制器复制计时器值时间戳存储在 RTS 和 TTS 中，用户可以读取寄存器 RTS 或者 TTS 获取时间戳值。

CAN 控制器可以在有效帧的 SOF 或 EOF 位的采样点获取时间戳。可以通过配置位 CAN_ACFCTRL0[TIMEPOS]进行选择采样点位置。 ACK 界定符之后的 7 个隐性位形成 CAN/CAN FD 帧的 EOF。在许多系统中广泛使用的基于软件的时间戳依赖于接收和传输中断。因此，建议时间戳采样点配置在 EOF。

CAN 控制器仅支持一个发送帧（TTS）的时间戳，但是对接收帧（RTS）有单独的时间戳。生成发送帧的时间戳可以使用 TBUF slot 内的 TTSEN 位分别为每个帧启用或禁用。

CAN 控制器内部包含计时器，时间戳机制，存储 TTS 的寄存器以及每帧存储 RTS 的缓存。寄存器 CAN_ACFCTRL0[TIMEEN]位启用或禁用时间戳。如果禁用，则 TTS 和 RTS 无效。

CAN 时间戳可配置为外部时钟和内部位时钟，具体框图如图 26-4 所示。可通过寄存器 CAN_ACFCTRL0[TCSS]选择外部时钟并进行 1 到 8 分频作为时间戳的时钟，也可以选择配置标称比特率的位时钟来作为时钟。

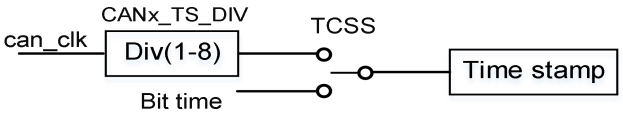


图 26-4 时间戳时钟配置

下面举例说明 CAN 控制器发送时间戳功能使用步骤：

1. 初始化 CAN 控制器，使能发送中断，配置其时钟源为 8MHz；
2. 设置时钟源为外部时钟，时钟分频 8，则 TTS 中每个计数值就是 1μs；
3. 通过配置 TIMEPOS 寄存器设置采样点位置为 EOF；
4. 使能 TIMEEN 和 TCEN 寄存器以及发送消息中的 TTSEN 位；

© 2013 - 2024 杰发科技有限公司

5. 在 CAN 控制器发送完成后读取 TTS 值。

下面举例说明 CAN 控制器接收时间戳功能使用步骤：

1. 初始化 CAN 控制器，使能接收中断，配置其通信波特率为 500kbps；
2. 设置时钟源为内部位时间，则 RTS 每个计数值就是 2μs；
3. 通过配置 TIMEPOS 寄存器设置采样点位置为 EOF；
4. 使能 TIMEEN 和 TCEN 寄存器；
5. 在 CAN 控制器接收完成后读取 RTS 值。

26.4.13 ECC 保护

CAN 的数据存储 SRAM 内存放了发送和接收缓冲区以及过滤器的配置，为增强存储数据的可靠性，增加了存储 SRAM 的 ECC 的保护。在需要的时候，可在初始化状态 RESET = 1 下，配置 CAN_VERMEM[MEEN] = 1 来使能 ECC 的保护功能。

存储 ECC 提供了每 32 位数据 1 个位数据警告并自动纠错，和 2 位数据错误检测并报告出错位置的功能。

当存储数据后，读取数据过程中检测到 1 个位的数据错误，如果使能了 CAN_VERMEM[MDWIE]，会触发数据警告 CAN_VERMEM[MDWIF]中断，同时自动将该位数据纠正。

当存储数据后，读取数据过程中检测到 2 个位的数据错误，如果使能了 CAN_VERMEM[MDEIE]，会触发数据错误 CAN_VERMEM[MDEIF]中断，同时通过 CAN_VERMEM[ACFA]，TXS 和 HELOC 报告错误位置。当接收滤波器工作时，从存储 SRAM 中读取 ACODE 和 AMASK，如出现 2 位数据错误，则产生数据错误中断并报告 ACFA 状态，这种情况下过滤器失效，导致接收到所有的消息。当 CAN 控制器发送数据时，从存储 SRAM 中的 TBUF 读取帧数据，出现 2 位数据错误，则产生数据错误中断并报告 TXS 状态，这种情况下发送数据停止发送，并将 TBUF 中数据丢弃。当主控制器读取 TBUF，RBUF 和 ACF 中的数据时，出现 2 位数据错误，则产生数据错误中断并报告相应的 HELOC 状态。

存储错误注入寄存器 CAN_MEMES 仅用于内部测试。

26.4.14 中断请求

每路 CAN 模块共有 2 个中断向量，对应中断源如表 26-9 和表 26-10 所示。

表 26-9 CAN 中断汇总

中断源	标志	本地使能	全局使能
接收中断	RIF	RIE	
接收缓冲区溢出中断	ROIF	ROIE	
接收缓冲区满中断	RFIF	RFIE	
接收缓冲区几乎满中断	RAFIF	RAFIE	

© 2013 - 2024 杰发科技有限公司

主发送完成中断	TPIF	TPIE	
次发送完成中断	TSIF	TSIE	
错误中断	EIF	EIE	
中止中断	AIF	-	
错误被动中断	EPIF	EPIE	
仲裁失利中断	ALIF	ALIE	
总线错误中断	BEIF	BEIE	
存储数据警告中断	MDWIF	MDWIE	
存储数据错误中断	MDEIF	MDEIE	

表 26-10 CAN 唤醒中断

中断源	标志	本地使能	全局使能
唤醒中断（低功耗模式下）	-	WUEN	

26.4.15 低功耗模式

当 MCU 进入低功耗 STOP1/STOP2/VLPS 模式前，可使能 CAN 唤醒中断 WUEN，当检测到总线活动时（显性电平）会产生唤醒中断并唤醒 MCU。如需滤除毛刺信号（如低于 2μs，请参考《ATC_AC7840x_Datasheet_CH》），可设置寄存器 LPFEN 等于 1 来使能低通滤波器。

CAN 在 STOP1/STOP2/VLPS 模式下存在不同的唤醒源，具体如表 26-11 所示：

表 26-11 CAN 模块唤醒源

模式	唤醒源
STOP1	低电平唤醒中断
STOP2	低电平唤醒中断
	接收中断唤醒
VLPS	低电平唤醒中断

系统要进入 STOP1/STOP2/VLPS 模式时，如果 CAN 模块正处于忙状态（正在接收发送数据），则不会回复休眠应答确认 ACK 信号，造成休眠超时。

在 VLPR 模式下，CAN 模块由于受到总线时钟和功能时钟的最大频率影响，部分功能可能会受到限制，如接收过滤器个数和最大通信比特率。

26.4.16 调试模式

CAN 模块没有调试控制位，调试模式对 CAN 模块运行无影响。

© 2013 - 2024 杰发科技有限公司

26.5 应用说明

26.5.1 CAN 位时间

CAN 2.0B 定义了高达 1Mbit/s 的数据比特率。对于 CAN FD 没有固定的限制，对于实际的系统，数据速率受所使用的收发器和 CAN 控制器选择的时钟源频率的限制，这取决于实际的选择与配置。

CAN 控制器可以配置为任意的数据速率，仅受相应位定时和预分频器寄存器中位设置范围的限制。

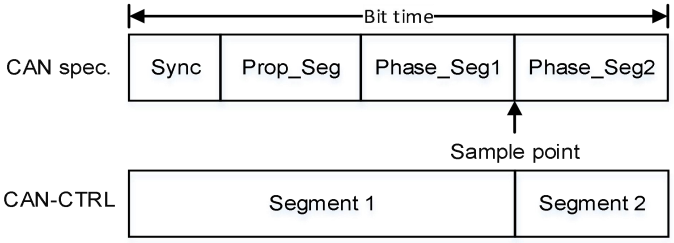


图 26-5 CAN 位定时

CAN 位时间 BT 由若干段（segment）组成，如图 26-5 所示。每个段由 n_{TQ} 个时间份额 TQ（time quantum）组成。

时间份额 TQ 的时长为：

$$TQ = n_{prescaler} / f_{CLOCK}$$

位时间 BT 的时长为：

$$BT = n_{prescaler} * n_{TQ} / f_{CLOCK} = t_{Seg_1} + t_{Seg_2}$$

CAN 规范要求各段长度之间存在相应关系如表 26-12 所示，该表列举了 CAN 规范定义的各段最小配置范围。

表 26-12 CAN 定时段

段	说明
SYNC_SEG	**同步段** = 1 TQ
PROP_SEG	**传播段** [1...8] TQ　CAN 2.0 比特率　CAN FD 未启用 [1...48] TQ　CAN 2.0 比特率　CAN FD 已启用 [1...48] TQ　CAN FD 标称比特率 [0...8] TQ　CAN FD 数据比特率
PHASE_SEG1	**相位缓冲段 1** [1...8] TQ　CAN 2.0 比特率　CAN FD 未启用 [1...16] TQ　CAN 2.0 比特率　CAN FD 已启用 [1...16] TQ　CAN FD 标称比特率 [1...8] TQ　CAN FD 数据比特率
PHASE_SEG2	**相位缓冲段 2** [2...8] TQ　CAN 2.0 比特率　CAN FD 未启用

© 2013 - 2024 杰发科技有限公司

段	说明
	[2...16] TQ CAN 2.0 比特率 CAN FD 已启用 [2...16] TQ CAN FD 标称比特率 [2...8] TQ CAN FD 数据比特率
SJW	**同步跳转宽度** [1...4] TQ CAN 2.0 比特率 CAN FD 未启用 [1...16] TQ CAN 2.0 比特率 CAN FD 已启用 [1...16] TQ CAN FD 标称比特率 [1...8] TQ CAN FD 数据比特率
IPT	**信息处理时间** = [0…2] TQ PHASE_SEG2 ⩾ IPT

CAN 控制器将 SYNC_SEG，PROP_SEG 和 PHASE_SEG1 形成一段，该段的长度由 t_{Seg_1} 进行配置，PHASE_SEG2 单独一段，长度由 t_{Seg_2} 进行配置。表 26-13 列出了这些段可用的配置范围。请注意，CAN 控制器不会检查是否满足所有规则，并提供比 CAN 2.0/CAN FD 规范定义的更宽的配置范围。

表 26-13 CAN 控制器定时配置

配置项	需求			
t_{Seg_1}	[2…65] TQ		CAN 2.0 比特率	(慢速)
	[2…65] TQ		CAN FD 标称比特率	(慢速)
	[2…17] TQ		CAN FD 数据比特率	(快速)
t_{Seg_2}	[2…16] TQ	$t_{Seg_1} \geq t_{Seg_2} + 2$	CAN 2.0 比特率	(慢速)
	[2…32] TQ	$t_{Seg_1} \geq t_{Seg_2} + 2$	CAN FD 标称比特率	(慢速)
	[2…8] TQ	$t_{Seg_1} \geq t_{Seg_2} + 1$	CAN FD 数据比特率	(快速)
t_{SJW}	[1…16] TQ	$t_{Seg_2} \geq t_{SJW}$	CAN 2.0 比特率	(慢速)
	[1…16] TQ	$t_{Seg_2} \geq t_{SJW}$	CAN FD 标称比特率	(慢速)
	[1…8] TQ	$t_{Seg_2} \geq t_{SJW}$	CAN FD 数据比特率	(快速)

使用寄存器 S_SEG_1，S_SEG_2，S_SJW 和 S_PRESC 来配置 CAN 2.0 比特率以及 CAN FD（慢速）标称比特率。使用寄存器 F_SEG_1，F_SEG_2，F_SJW 和 F_PRESC 来配置 CAN FD（快速）数据比特率。

$t_{Seg_1} = (S_SEG_1 + 2) * TQ$　　$t_{Seg_1} = (F_SEG_1 + 2) * TQ$

$t_{Seg_2} = (S_SEG_2 + 1) * TQ$　　$t_{Seg_2} = (F_SEG_2 + 1) * TQ$

$t_{SJW} = (S_SJW + 1) * TQ$　　$t_{SJW} = (F_SJW + 1) * TQ$

$n_{prescaler} = S_PRESC + 1$　　$n_{prescaler} = F_PRESC + 1$

采样点 $SP = (S_SEG_1 + 2) / (S_SEG_1 + 2 + S_SEG_2 + 1)$

CAN FD 控制器通过帧设置 BRS = 1 就可以从低速标称比特率切换到快速数据比特率，并在 CRC 分隔符位的采样点切回。

© 2013 - 2024 杰发科技有限公司

对于 CAN FD 节点，由于收发器的延迟导致 CAN FD 的通信可能会延迟超过一个快速数据位的时间，因此，无法使用原始采样点 SP（Sample point）对正确的位值进行采样，CAN FD 规范定义了一个额外的二次采样点（SSP），此时可以选择启用发射机延迟补偿（TDC）即 CAN_ERRINFO[TDCEN] = 1 且配置二次采样点偏移 CAN_ERRINFO[SSPOFF]寄存器，建议 SSPOFF 配置等于 t_{Seg_1}，用户可根据实际情况配置该寄存器。如果不启用 TDC 发送节点可能就不能正确采样到它自己所发出的数据，此时发送节点就会检测到一个位错误。

初始化 CAN 控制器的步骤如下：

1. 设置位 RESET = 1；
2. 设置寄存器 S_SEG_1 和 S_SEG_2：

   在该示例中，总线数据速率为 1M 波特率，系统时钟为 48MHz；

   所选 n_{TQ} 和 $n_{prescaler}$ 的值需适配 BT，在该示例中，所选 $n_{prescaler} = 2$，$n_{TQ} = 24$，这样可以实现完全匹配：$BT = 24TQ$；

   在时间段定义中，可以选择 $t_{Seg_1} = 18TQ$，$t_{Seg_2} = 6TQ$，对应寄存器 S_SEG_1 = 16，S_SEG_2 = 5；
3. 加载过滤器验证码和掩码寄存器（可选）；
4. 设置 S_SJW 寄存器；

   当满足 $t_{Seg_2} \geq t_{SJW}$，可以自由选择 $t_{SJW} = 4$，对应寄存器 S_SJW = 3；
5. 加载时钟预分频寄存器 S_PRESC：$n_{prescaler}$ = PRESC + 1，对应 S_PRESC = 1；
6. 设置位 RESET = 0；

   如上给出的顺序不是强制的。只需要在开始时设置 RESET = 1，否则无法加载位定时，ACODE 和 AMASK 寄存器。配置完成后，需要将 RESET = 0；
7. 继续配置中断其他配置位，并执行指令。

下面是一些适用于 CAN 网络所有节点位定时设置的示例表。

表 26-14 48MHz can_clk 的示例设置

比特率[Mbit/s]	SP[%]	预分频器	位时间[TQ]	Seg1 [TQ]	Seg2 [TQ]	SJW [TQ]
1	75	1	48	36	12	12
0.8	80	2	30	24	6	6
0.5	75	2	48	36	12	12
0.25	75	4	48	36	12	12
0.125	75	8	48	36	12	12
0.1	75	10	48	36	12	12

表 26-15 8MHz can_clk 的示例设置

比特率[Mbit/s]	SP[%]	预分频器	位时间[TQ]	Seg1 [TQ]	Seg2 [TQ]	SJW [TQ]

© 2013 - 2024 杰发科技有限公司

比特率[Mbit/s]	SP[%]	预分频器	位时间[TQ]	Seg1 [TQ]	Seg2 [TQ]	SJW [TQ]
1	75	1	8	6	2	2
0.8	80	1	10	8	2	2
0.5	75	1	16	12	4	4
0.25	75	2	16	12	4	4
0.125	75	4	16	12	4	4
0.1	75	4	20	15	5	4

表 26-16 48MHz can fd clk 的示例设置

比特率[Mbit/s]	SP[%]	预分频器	位时间[TQ]	Seg1[TQ]	Seg2[TQ]	SJW[TQ]	TDC[clk]
6	75	1	8	6	2	2	6
4	75	1	12	9	3	3	9
2	75	1	24	18	6	6	18
1	75	2	24	18	6	6	36
0.1	75	20	24	18	6	6	-
0.05	75	40	24	18	6	6	-

26.5.2 软件复位

如果 CAN_CTRL0 寄存器中的 RESET 位被置为 1，则软件复位处于激活状态。如果 RESET = 1，则控制器内部的几个组件被强制置为复位状态，但 RESET 不会触及所有组件。表 26-17 列举了软件复位是否会复位的 CAN 模块寄存器。而上电复位或模块复位 SRST_CANx 则会复位所有寄存器。

表 26-17 软件复位

寄存器	复位(RESET)	说明
ACFADR	否	-
ACODE_x	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
ACFA	是	-
AE_x	否	-
AFWL	否	-
AIF	是	-
ALC	是	-
ALIE	否	-
ALIF	是	-
AMASK_x	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
BEIE	否	-
BEIF	是	-
BOREC	否	-
BUSOFF	否	通过设置BUSOFF = 0可以复位BUSOFF及错误计数器
DREN	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
EIE	否	-
EIF	否	-

© 2013 - 2024 杰发科技有限公司

寄存器	复位(RESET)	说明
EPASS	否	-
EPIE	否	-
EPIF	是	-
EWARN	否	-
EWL	是	-
FDISO	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
F_PRESC	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
F_SEG_1	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
F_SEG_2	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
F_SJW	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
HELOC	是	-
IDLE	是	-
KOER	是	-
LBME	是	-
LBMI	是	-
LOM	否	-
LPFEN	否	-
MDEIE	否	-
MDEIF	是	-
MDWIE	否	-
MDWIF	是	-
MEL	否	-
MEID	否	-
MEBP1	否	-
MEBP2	否	-
ME1EE	否	-
ME2EE	否	-
MEEEC	否	-
MENEC	否	-
MES	否	-
MEEN	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
RACTIVE	是	即使接收处于活动状态，接收也会立即取消，不会生成 ACK 信息
RAFIE	否	-
RAFIF	是	-
RBALL	是	-
RBUF	(是)	所有 RB slot 被标记为空，RBUF 包含未知数据
RECNT	否	-
RFIE	否	-
RFIF	是	-
RIE	否	-
RIF	是	-
ROIE	否	-

© 2013 - 2024 杰发科技有限公司

寄存器	复位(RESET)	说明
ROIF	是	-
ROM	否	-
ROV	是	所有 RB slot被标记为空
RREL	是	-
RSTAT	是	所有RB slot被标记为空
SACK	是	-
SELMASK	否	-
STBY	否	-
S_PRESC	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
S_Seg_1	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
S_Seg_2	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
S_SJW	否	如果RESET = 1则寄存器可写，为0则寄存器写锁定
TACTIVE	是	在 RESET 时，所有发送都立即停止。如果发送有效，则会产生不完整的帧，其他节点会产生错误帧。
TBSEL	是	TBUF 固定指向 PTB
TBUF	(是)	所有 STB slot 被标记为空，因为 TBSEL TBUF 指向 PTB
TCEN	否	-
TCSS	否	-
TECNT	否	-
TIMEEN	否	-
TIMEPOS	否	-
TPA	是	-
TPE	是	-
TSA	是	-
TSALL	是	-
TSMODE	否	-
TSNEXT	是	-
TSONE	是	-
TPIE	否	-
TPIF	是	-
TPSS	是	-
TSFF	是	所有STB slot 被标记为空
TSIE	否	-
TSIF	是	-
TSSS	是	-
TSSTAT	是	所有STB slot被标记为空
TTS	否	-
TXS	是	-
WUEN	否	-

© 2013 - 2024 杰发科技有限公司

26.6 寄存器定义

CAN 寄存器只支持 32 位访问，如表 26-18 所示。

表 26-18 CAN-CRTL 寄存器映射

CAN0 基地址：0x40007000

CAN1 基地址：0x40007200

CAN2 基地址：0x40007400

CAN3 基地址：0x40007600

地址	名称	宽度	描述
CANx 基地址 ~ CANx 基地址+0x4F	CAN_RBUF	32*20	接收缓冲区寄存器 和接收时间戳
CANx 基地址+0x50 ~ CANx 基地址+0x9F	CAN_TBUF	32*20	发送缓冲区寄存器 和发送时间戳
CANx 基地址+0xA0	CAN_CTRL0	32	控制寄存器 0
CANx 基地址+0xA4	CAN_CTRL1	32	控制寄存器 1
CANx 基地址+0xA8	CAN_SBITRATE	32	低速波特率配置寄存器
CANx 基地址+0xAC	CAN_FBITRATE	32	快速波特率配置寄存器
CANx 基地址+0xB0	CAN_ERRINFO	32	错误状态寄存器
CANx 基地址+0xB4	CAN_ACFCTRL0	32	接收过滤器控制寄存器 0
CANx 基地址+0xB8	CAN_ACFCTRL1	32	接收过滤器控制寄存器 1
CANx 基地址+0xBC	CAN_ACFCTRL2	32	接收过滤器控制寄存器 2
CANx 基地址+0xC0	CAN_ACF	32	过滤器配置寄存器（ACODE 与 AMASK 共用）
CANx 基地址+0xC4	CAN_VERMEM	32	版本及存储保护寄存器
CANx 基地址+0xC8	CAN_MEMES	32	存储错误注入寄存器
CANx 基地址+0xCC	CAN_WAKEUP	32	唤醒配置寄存器

说明：

CANx 中 x 最大范围为 0~3。

© 2013 - 2024 杰发科技有限公司

26.6.1 控制寄存器 0(CAN_CTRL0)

表 26-19 CAN_CTRL0 寄存器

CAN_CTRL0 控制寄存器 0 复位值：0x00840080

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SACK	ROM	ROV	RREL	RBALL	DREN	RSTAT		FDISO	TSNEXT	TSMODE		BOREC	IDLE	TSSTAT	
访问	RW	RW	RO	RW	RW	RW	RO		RW	RW	RW		RW	RO	RO	
复位值	0	0	0	0	0	0	0		1	0	0		0	1	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TBSEL	LOM	STBY	TPE	TPA	TSONE	TSALL	TSA	RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	R/W0C
复位值	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

字段	说明
31 SACK	**自应答** 0：不自应答 1：自应答（LBME 模式）
30 ROM	**接收缓冲区溢出模式** 0：最旧的消息将被覆盖 1：新消息将不会被存储 如果接收到新消息时 RBUF 已满，则根据 ROM 选择做不同的动作。
29 ROV	**接收缓冲区溢出** 0：没有溢出 1：溢出，至少一个消息丢失 ROV 会被 RREL = 1 清除。
28 RREL	**接收缓冲区释放** 0：不释放 1：接收缓冲区（最早的）释放，消息已经被读取 主控制器已读取接收缓冲区并将其释放。然后，CAN 控制器指向下一个接收缓冲区， RSTAT 将会被更新。
27	**接收缓冲区存储所有的数据帧**

© 2013 - 2024 杰发科技有限公司

字段	说明
RBALL	
	0：正常操作 1：接收缓冲区存储所有接收的数据帧，即使数据帧有误也会被存储，一般用来做测试。
26 DREN	**DMA 接收使能** 0：禁用 1：使能 该位只有在 RESET = 1 时才可写。
25：24 RSTAT	**接收缓冲区状态** 00b：空 01b：非空，小于几乎满（AFWL） 10b：几乎满（AFWL 可编程阀值），但不满不溢出 11b：满 （在溢出的情况下保持状态）
23 FDISO	**CAN FD ISO 模式** 0：Bosch CAN FD（non-ISO）模式 1：ISO CAN FD 模式（ISO 11898-1:2015） ISO CAN FD 模式有不同的 CRC 校验格式，在同一网络中不能同时混合这两种模式。 该位对 CAN2.0B 无作用。 该位只有在 RESET = 1 时才可写。
22 TSNEXT	**选择下一个次发送缓冲区** 0：没有动作 1：填充 STB 缓冲区后，选择下一个缓冲区 在将所有帧字节写入 TBUF 寄存器后，主控制器必须设置 TSNEXT 表示此缓冲区已填充。然后 CAN 控制器将 TBUF 寄存器连接到下一个缓冲区。 一旦缓冲区标记为已填充，就可以使用 TSONE 或 TSALL 启动发送。 可以在一次写访问中，将 TSNEXT，TSONE 或 TSALL 一起设置。 TSNEXT 必须由主控制器设置，并在设置后立即由 CAN 控制器自动复位。 如果 TBSEL = 0，则设置 TSNEXT 是没有意义的。在这种情况下，TSNEXT 将被忽略并自动清除且没有任何影响。如果 STB 的所有缓冲区都已填满，则 TSNEXT 将保持设置状态，直到缓冲区空闲为止。
21 TSMODE	**次发送缓冲区模式** 0：FIFO 模式 1：优先级决定模式 在 FIFO 模式下，帧按照它们写入 STB 的顺序发送。在优先级决策模式中，首先自动发送 STB 中具有最高优先级的帧。帧的 ID 用于优先级决定。较低的 ID 表示一个具有较高优先级的帧。无论 ID 如何，PTB 中的帧始终具有最高优先级。只有当 STB 为空时，才能切换 TSMODE。

© 2013 - 2024 杰发科技有限公司

字段	说明
19 BOREC	**自动总线关闭恢复** 0：自动总线关闭恢复使能 1：自动总线关闭恢复禁用 进入总线关闭状态后，如 BOREC 为 0，则按照 ISO 11898-1 规范，在 128 个连续 11 个位的隐性电平后，自动恢复总线。 进入总线关闭状态后，如 BOREC 为 1，则总线关闭状态保持；如这时设置 BOREC 为 0，则同样在 128 个连续 11 个位的隐性电平后，自动恢复总线。
18 IDLE	**总线空闲状态** 0：总线忙状态 1：总线空闲状态（没有处在接收状态，也没有任何发送任务）
17：16 TSSTAT	**次发送缓冲区状态** 00b：STB 为空 01b：STB 小于或等于半满 10b：STB 多于半满 11b：STB 满
15 TBSEL	**发送缓冲区选择** 0：PTB（高优先级缓冲区） 1：STB 选择要填充消息的发送缓冲区，使用 TBUF 寄存器进行访问。在写入 TBUF 寄存器和设置 TSNEXT 时，TBSEL 需要始终保持状态。
14 LOM	**监听模式** 0：禁用 1：使能 当处于发送状态时，不能开启监听模式。如果开启监听模式，则无法启动发送。
13 STBY	**收发器待机模式** 0：禁用 1：使能 该寄存器位连接到输出引脚 standby，可用于控制收发器的待机模式。 如果 TPE = 1，TSONE = 1 或 TSALL = 1，STBY 不能被设置成 1。 如设置 STBY 从待机模式退出，则收发器在新的数据发送之前，需要等待一定的时间。
12 TPE	**主发送使能** 0：不发送 PTB 1：发送高优先级 PTB 中的消息

© 2013 - 2024 杰发科技有限公司

字段	说明
	如果设置了 TPE，则来自 PTB 的消息将在下一个可能的发送位置发送。来自 STB 的开始发送将在之前完成，但是等待新消息被延迟直到 PTB 消息已被发送。 TPE 保持设置，直到消息成功发送或使用 TPA 中止。 主控制器可以将 TPE 设置为 1 但不能将其重置为 0，这将只能使用 TPA 并中止消息。 在 CAN 控制器重置该位的短时间内，主控制器无法设置它。 如果 RESET = 1，STBY = 1，（LOM = 1 且 LBME = 0），该位将复位为硬件复位值。
11 TPA	**主发送中止** 0：不中止 1：中止 PTB 的发送，该发送已被 TPE = 1 请求但尚未启动。（消息的数据字节保留在 PTB 中） 该位必须由主控制器设置，并由 CAN 控制器复位。 设置 TPA 会自动取消激活 TPE。 主控制器可以将 TPA 设置为 1，但不能将其重置为 0。 在 CAN 控制器重置该位的短时间内，主控制器无法设置它。 如果 RESET = 1，该位将复位为硬件复位值。 TPA 不应与 TPE 同时设置。
10 TSONE	**次发送一帧使能** 0：不发送 STB 1：发送 STB 中的一个消息 在 FIFO 模式下，发送最早的消息。在优先级模式下，发送最高优先级 ID 的消息。 在优先级模式下，TSONE 难以处理，因为如果同时将新消息写入 STB，则不总是清楚哪个消息将被发送。 一旦总线空闲并且没有 PTB（TPE 位）的请求，控制器就开始发送。 TSONE 保持置位状态，直到消息成功发送或被 TSA 中止。 主控制器可以将 TSONE 设置为 1，但不能将其重置为 0，这只能使用 TSA 并中止消息。 在 CAN 控制器重置该位的短时间内，主控制器无法设置它。 如果 RESET = 1，STBY = 1，（LOM = 1 且 LBME = 0），该位将被复位。
9 TSALL	**次发送所有帧使能** 0：不发送 STB 1：发送 STB 中的所有消息 一旦总线空闲并且没有 PTB（TPE 位）的请求，控制器就开始发送。 TSALL 保持设置状态，直到所有消息都已成功发送或被 TSA 中止。 主控制器可以将 TSALL 设置为 1，但不能将其重置为 0，这只能使用 TSA 并中止消息。 在 CAN 控制器重置该位的短时间内，主控制器无法设置它。 如果 RESET = 1，STBY = 1，（LOM = 1 且 LBME = 0），该位将被复位。 如果在发送期间STB 加载了新帧，则新帧也将被发送。换言之，当STB 变空时，由TSALL 启动的发送才结束。
8 TSA	**次发送中止** 0：不中止

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：从 STB 中止已经请求但尚未启动的发送 对于 TSONE 发送，只有一帧被中止，而对于 TSALL 发送，所有帧都被中止。将释放一个或所有消息缓冲区，用于更新 TSSTAT。 所有中止的消息都将丢失，因为它们不再可访问。 在优先级模式下，如果 TSONE 发送中止，此时如果同时向 STB 写入新帧，则不清楚哪个帧将被中止。 该位必须由主控制器设置，并由 CAN 控制器复位。 设置 TSA，分别自动取消 TSONE 或 TSALL。主控制器可以将 TSA 设置为 1，但不能将其重置为 0。 如果 RESET = 1，该位将被复位。 TSA 不应与 TSONE 或 TSALL 同时设置。
7 RESET	**复位请求** 0：CAN 控制器不复位 1：执行 CAN 控制器的本地复位 若 RESET = 1，则只能修改某些寄存器（例如节点配置）。 RESET = 1 会迫使多个组件进入复位状态。 RESET 切换为 0，在 11 个连续的 CAN 隐性位时间后，CAN 节点将参与 CAN 通信。 CAN 标准（总线空闲时间）需要此延迟。 若 RESET 设置为 1 并立即设置为 0，则需要一些时间才能将 RESET 读取为 0 并变为非活动状态。
6 LBME	**环回模式，外部** 0：禁用 1：使能 当发送处于活动状态时，不应启用 LBME。
5 LBMI	**环回模式，内部** 0：禁用 1：使能 当发送处于活动状态时，不应启用 LBMI。
4 TPSS	**PTB 单次发送模式** 0：禁用 1：使能（禁用发送帧出错或仲裁失利后重发功能）
3 TSSS	**STB 单次发送模式** 0：禁用 1：使能（禁用发送帧出错或仲裁失利后重发功能）
2 RACTIVE	**接收状态** 0：没有接收活动 1：控制器当前正在接收帧

© 2013 - 2024 杰发科技有限公司

字段	说明
1 TACTIVE	**发送状态** 0：没有发送活动 1：控制器当前正在发送帧
0 BUSOFF	**总线关闭状态** 0：控制器状态为“总线打开（bus on）” 1：控制器状态为“总线关闭（bus off）” 在总线关闭状态下，写入 0 将直接退出总线关闭状态，并重置 TECNT 和 RECNT（不会产生退出总线关闭中断）。

26.6.2 控制寄存器 1(CAN_CTRL1)

表 26-20 CAN_CTRL1 寄存器

CAN_CTRL1　　控制寄存器 1　　复位值：0x1B0000FE

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	AFWL				EWL				EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF
访问	RW				RW				RO	RO	RW	R/W1C	RW	R/W1C	RW	R/W1C
复位值	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF	RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF
访问	R/W1C								RW	RW	RW	RW	RW	RW	RW	RO
复位值	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

字段	说明
31：28 AFWL	**接收缓冲区几乎满告警限制** AFWL 定义了接收缓冲区警告限制值，其值需小于等于最大接收缓冲区 slots 数量。 将 AFWL 与填充的接收缓冲区中的消息帧数进行比较，并且如果相等则触发 RAFIF。 AFWL = 0 自动等同于 1； AFWL> 最大缓冲区 slots 数量，等同于最大接收缓冲区数量，当消息帧 达到数量时，RFIF 也会触发。
27：24 EWL	**可编程错误警告限制值** 错误警告限制值 = (EWL + 1) * 8，可设置限制值为 8, 16, …128。 EWL 的值控制 EIF 产生的条件。
23 EWARN	**达到错误警告限制**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：两个计数器的值都小于 EWL 1：错误计数器 RECNT 或 TECNT 的一个等于或大于 EWL
22 EPASS	**错误被动模式激活** 0：主动错误状态 1：被动错误状态
21 EPIE	**错误被动中断使能** 0：禁用 1：使能
20 EPIF	**错误被动中断标志** 0：没有状态改变 1：错误状态从主动错误状态变为被动错误状态，或反之； 写 1 可以清除该标志。
19 ALIE	**仲裁失利中断使能** 0：禁用 1：使能
18 ALIF	**仲裁失利中断标志** 0：没有仲裁失利 1：仲裁失利 写 1 可以清除该标志。
17 BEIE	**总线错误中断使能** 0：禁用 1：使能
16 BEIF	**总线错误中断标志** 0：没有总线错误 1：总线错误，错误类型对应 KOER 写 1 可以清除该标志。
15 RIF	**接收中断标志** 0：没有帧被接收 1：数据或远程帧已接收，并且可在接收缓冲区中使用 写 1 可以清除该标志。
14 ROIF	**接收缓冲区溢出中断标志** 0：没有 RB 被覆盖

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：RB 中至少有一条接收消息被覆盖 如果发生溢出，ROIF 和 RFIF 都将被置位。 写 1 可以清除该标志。
13 RFIF	**接收缓冲区满中断标志** 0：RB FIFO 未满 1：所有 RB 都已满。 写 1 可以清除该标志。
12 RAFIF	**接收缓冲区几乎满中断标志** 0：填充的 RB 缓冲区数 < AFWL 1：填充的 RB 缓冲区数 ≥ AFWL 写 1 可以清除该标志。
11 TPIF	**主发送完成中断标志** 0：没有完成 PTB 的发送 1：已成功完成所请求的 PTB 发送 写 1 可以清除该标志。
10 TSIF	**次发送完成中断标志** 0：没有完成 STB 的发送 1：已成功完成所请求的 STB 发送 写 1 可以清除该标志。
9 EIF	**错误中断标志** 0：没有状态改变 1：错误警告限制的边界已在任一方向上穿越，或者 BUSOFF 位已在任一方向上改变 写 1 可以清除该标志。
8 AIF	**中止中断标志** 0：没有执行中止 1：在设置 TPA 或 TSA 之后，已经中止了相应的消息 建议不要同时设置 TPA 和 TSA，因为两者都是源 AIF。 AIF 没有关联的使能寄存器。 写 1 可以清除该标志。
7 RIE	**接收中断使能** 0：禁用

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：使能
6 ROIE	**接收缓冲区溢出中断使能** 0：禁用 1：使能
5 RFIE	**接收缓冲区满中断使能** 0：禁用 1：使能
4 RAFIE	**接收缓冲区几乎满中断使能** 0：禁用 1：使能
3 TPIE	**主发送（Transmission Primary）完成中断使能** 0：禁用 1：使能
2 TSIE	**次发送（Transmission Secondary）完成中断使能** 0：禁用 1：使能
1 EIE	**错误中断使能** 0：禁用 1：使能，中断标志对应 EIF
0 TSFF	**次发送缓冲区满标志** 0：STB 没有接收到最大数量的消息 1：STB 接收到最大数量的消息

26.6.3 低速波特率配置寄存器(CAN_SBITRATE)

表 26-21 CAN_SBITRATE 寄存器

CAN_SBITRATE **低速波特率配置寄存器** **复位值：0x01020203**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	S_PRESC									S_SJW						
访问	RW									RW						
复位值	0x01									0x02						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		S_SEG_2							S_SEG_1							
访问		RW							RW							
复位值		0x02							0x03							

© 2013 - 2024 杰发科技有限公司

字段	说明
31：24 S_PRESC	**预分频器** 预分频器将 CAN 时钟分频以获得时间份额时钟 tq_clk。有效范围 S_PRESC=[0x00, 0xff]，对应分频器值为 1 ~ 256。 该位只有在 RESET = 1 时才可写。
22：16 S_SJW	**同步补偿宽度** 同步补偿宽度 $t_{SJW} = (S_SJW + 1) * TQ$ 是缩短或延长重新同步的位时间的最长时间，其中 TQ 是时间份额。 该位只有在 RESET = 1 时才可写。
14：8 S_SEG_2	**位定时段 2** 在采样点后定时 $t_{Seg_2} = (S_SEG_2 + 1) * TQ$。 该位只有在 RESET = 1 时才可写。
7：0 S_SEG_1	**位定时段 1** 位定时开始后，采样点被设置为 $t_{Seg_1} = (S_SEG_1 + 2) * TQ$。 该位只有在 RESET = 1 时才可写。

26.6.4 快速波特率配置寄存器 (CAN_FBITRATE)

表 26-22 CAN_FBITRATE 寄存器

CAN_FBITRATE **快速波特率配置寄存器** **复位值：0x01020203**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	F_PRESC												F_SJW			
访问	RW												RW			
复位值	0x01												0x02			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					F_SEG_2							F_SEG_1				
访问					RW							RW				
复位值					0x02							0x03				

字段	说明
31: 24 F_PRESC	**预分频器（用于 CAN FD 数据比特率）** 预分频器将 CAN 时钟分频以获得时间份额时钟 tq_clk。有效范围 F_PRESC = [0x00, 0xff] 对应分频器值为 1 ~ 256。 该位只有在 RESET = 1 时才可写。
19: 16 F_SJW	**同步补偿宽度（用于 CAN FD 数据比特率）** 同步补偿宽度 $t_{SJW} = (F_SJW + 1) * TQ$ 是缩短或延长重新同步的位时间的最长时间，其中 TQ 是时间份额。

© 2013 - 2024 杰发科技有限公司

字段	说明
	该位只有在 RESET = 1 时才可写。
11: 8 F_SEG_2	**位定时段 2 （用于 CAN FD 数据比特率）** 在采样点后定时 t_{Seg_2} = (*F_SEG_2* + 1) * *TQ*。 该位只有在 RESET = 1 时才可写。
4: 0 F_SEG_1	**位定时段 1 （用于 CAN FD 数据比特率）** 位定时开始后，采样点被设置为 t_{Seg_1} = (*F_SEG_1* + 2) * *TQ*。 该位只有在 RESET = 1 时才可写。

26.6.5 错误状态寄存器(CAN_ERRINFO)

表 26- 23 CAN_ERRINFO 寄存器

CAN_ERRINFO **错误状态寄存器** **复位值：**
0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	TECNT								RECNT							
访问	RO								RO							
复位值	0								0							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TDC EN	SSPOFF							KOER			ALC				
访问	RW	RW							RO			RO				
复位值	0	0							0			0				

字段	说明
31：24 TECNT	**发送错误计数器** TECNT 按照 CAN 规范中的定义递增和减少，TECNT 会溢出 有关 TECNT 和“总线关闭”状态的更多详细信息，请参考 26.4.2。
23：16 RECNT	**接收错误计数器** RECNT 按照 CAN 规范中的定义递增和减少 RECNT 不溢出， RECNT 将 0xff = 255 作为最大值 有关 RECNT 和“总线关闭”状态的更多详细信息，请参考 26.4.2。
15 TDCEN	**发送器延迟补偿使能** 0：禁用 1：使能 如果 TDCEN = 1，则在 BRS 处于活动状态时，将在 CAN FD 帧的数据阶段激活 TDC。
14：8 SSPOFF	**二次采样点偏移**

© 2013 - 2024 杰发科技有限公司

字段	说明
	传输延迟 SSPOFF 为 TDC 定义了二次采样点的时间，SSPOFF 是 TQ 的个数；当使能延迟补偿时，SSPOFF 不能设置为 0。
7：5 KOER	**错误类型** 000b：没有错误（No error） 001b：位错误（BIT ERROR） 010b：格式错误（FORM ERROR） 011b：填充错误（STUFF ERROR） 100b：应答错误（ACKNOWLEDGEMENT ERROR） 101b：校验错误（CRC ERROR） 110b：其他错误（OTHER ERROR） (错误标志后收到显性电平，接收到主动错误标志太长，ACK 错误后的被动错误标志收到显性电平)。 111b：保留 KOER 会随着每个新错误而更新。因此当帧被成功地发送或接收时，它保持不变。
4：0 ALC	**仲裁失利位置** 指向仲裁失利的位在帧中所处的位置。

26.6.6 接收过滤器控制寄存器 0(CAN_ACFCTRL0)

表 26-24 CAN_ACFCTRL0 寄存器

CAN_ACFCTRL0　　接收过滤器控制寄存器 0　　复位值：0x00010200

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	ACFENx															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					TCSS	TCEN	TIMEPOS	TIMEN		SELMASK	ACFADR					
访问					RW	RW	RW	RW		RW	RW					
复位值					0	0	1	0		0	0					

字段	说明
31：16 ACFENx	**接收过滤器使能（x = 0-15）** 0：接收过滤器 x 禁用 1：接收过滤器 x 使能 每个接收过滤器 （AMASK/ACODE）可以单独启用或禁用。复位后，默认情况下仅使能 0

© 2013 - 2024 杰发科技有限公司

字段	说明
	号过滤器。 禁用所有过滤器将接收不到任何消息。只有接收到的消息与相应的 AMASK/ACODE 配置匹配，使能的过滤器才可以接收消息。
11 TCSS	**时间戳时钟源选择** 0：外部时钟（可 1-8 分频） 1：内部位时钟（bit timing） 仅当 TIMEEN = 0 时才能更改 TCSS。
10 TCEN	**时间戳计数器使能** 0：禁用 1：使能
9 TIMEPOS	**时间戳标记位置** 0：帧起始位 SOF 1：帧结束位 EOF 仅当 TIMEEN = 0 时才能更改 TIEMPOS，也可同时修改 TIMEPOS 和设置 TIMEEN = 1。
8 TIMEEN	**时间戳标记使能** 0：禁用 1：使能
6 SELMASK	**选择接受掩码（Select acceptance MASK）** 0：ACF_x 寄存器指向验证码 1：ACF_x 寄存器指向接收掩码 ACFADR 选择一个特定的接收滤波器。
5：0 ACFADR	**接收滤波器地址** ACFADR 指向某一个接收滤波器 0-59。选择后才能使用寄存器 ACF_x 设置对应滤波器。 SELMASK 位在所选接受过滤器的验证码和掩码之间进行选择。

26.6.7 接收过滤器控制寄存器 1(CAN_ACFCTRL1)

表 26- 25 CAN_ACFCTRL1 寄存器

CAN_ACFCTRL1	接收过滤器控制寄存器 1	复位值：0x00000000

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	ACFENx															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	ACFENx															
访问	RW															
复位值	0															

字段	说明
31：0 ACFENx	**接收过滤器使能（x = 16-47）** 0：接收过滤器 x 禁用 1：接收过滤器 x 使能 每个接收过滤器（AMASK/ACODE） 可以单独启用或禁用。

26.6.8 接收过滤器控制寄存器 2(CAN_ACFCTRL2)

表 26- 26 CAN_ACFCTRL2 寄存器

CAN_ACFCTRL2　　接收过滤器控制寄存器 2　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					ACFENx											
访问					RW											
复位值					0											

字段	说明
11：0 ACFENx	**接收过滤器使能（x = 48-59）** 0：接收过滤器 x 禁用 1：接收过滤器 x 使能 每个接收过滤器（AMASK/ACODE） 可以单独启用或禁用。

© 2013 - 2024 杰发科技有限公司

26.6.9 过滤器代码 ACODE 寄存器(CAN_ACF)

表 26-27 CAN_ACF 寄存器

CAN_ACF **过滤器代码 ACODE 寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				ACODE												
访问				RW												
复位值				0												
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	ACODE															
访问	RW															
复位值	0															

字段	说明
28：0 ACODE	**接收代码（Acceptance CODE），需设置 SELMASK 等于 0** 0/1：每个位与接收消息的 ID 位进行比较 ACODE_x(10：0)用于标准帧，ACODE_x(28：0)用于扩展帧 只有 0 号过滤器受上电复位的影响，所有其他过滤器保持未初始化状态。 该位只有在 RESET = 1 时才可写。

26.6.10 过滤器掩码 AMASK 寄存器(CAN_ACF)

表 26-28 CAN_ACF 寄存器

CAN_ACF **过滤器掩码 AMASK 寄存器** **复位值：0x1FFFFFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		AIDEE	AIDE	AMASK												
访问		RW	RW	RW												
复位值		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	AMASK															
访问	RW															
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
30 AIDEE	**接收掩码 IDE 位检查使能，需设置 SELMASK 等于 1** 0：接收过滤器接收标准或扩展帧

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：接收过滤器接收 AIDE 定义的标准或扩展帧 只有 0 号过滤器受上电复位的影响，所有其他过滤器保持未初始化状态。 该位只有在 RESET = 1 时才可写。
29 AIDE	**接收掩码 IDE 位的值，需设置 SELMASK 等于 1** 0：接收过滤器仅接收标准帧（AIDEE = 1 时） 1：接收过滤器仅接收扩展帧（AIDEE = 1 时） 该位只有在 RESET = 1 时才可写。
28：0 AMASK	**接收掩码 MASK，需设置 SELMASK 等于 1** 0： 匹配对应的接收过滤位 1： 屏蔽对应的接收过滤位 AMASK_x(10：0)用于标准帧，AMASK_x(28：0) 用于扩展帧。 屏蔽位接收该消息位，不做匹配。因此，过滤器 0 重置后的默认配置接受所有消息。 只有 0 号过滤器受上电复位或模块复位的影响，所有其他过滤器保持未初始化禁用状态。 该位只有在 RESET = 1 时才可写。

26.6.11 版本及存储保护寄存器(CAN_VERMEM)

表 26-29 CAN_VERMEM 寄存器

CAN_VERMEM **版本及存储保护寄存器** **复位值：**
0x0080070B

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				HELOC			TXS	ACFA	MEID			MDEIF	MDEIE	MDWIF	MDWIE	MEEN
访问				RO			RO	RO	RO			R/W 1C	RW	R/W 1C	RW	RW
复位值				0			0	0	1			0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	VERSION															
访问	RO															
复位值	0x070B															

字段	说明
28：27 HELOC	**主控制器端数据错误位置** 00b：没有错误 01b：主控制器读 TBUF 时出错 10b：主控制器读 RBUF 时出错 11b：主控制器读 ACF 时出错

© 2013 - 2024 杰发科技有限公司

字段	说明
	该位可被新的错误更新，或者 RESET = 1 时复位。
25 TXS	**发送停止** 0：正常工作 1：发送停止 发送数据的过程中，从填充的 TBUF 中读取到的数据错误，则本次发送将被停止。 该位可被新的错误更新，或者 RESET = 1 时复位。
24 ACFA	**过滤器接收** 0：正常通过过滤器接收数据 1：过滤器无效，所有数据都被接收 接收过滤器工作过程中，从设置的过滤器中读取到过滤器 CODE 和 MASK 数据错误，则本次接收数据过滤器无效，所有数据都被接收。 该位可被新的错误更新，或者 RESET = 1 时复位。
23 MEID	**存储数据初始化完成标志** 0：存储数据正在初始化 1：存储数据完成初始化
20 MDEIF	**存储数据错误中断标志** 0：没有数据错误 1：数据错误，检测到有 2bit 的数据出错，并不能纠正
19 MDEIE	**存储数据错误中断使能** 0：禁用 1：使能
18 MDWIF	**存储数据警告中断标志** 0：没有数据警告 1：数据警告，检测到 1bit 的数据出错，并已自动纠正
17 MDWIE	**存储数据警告中断使能** 0：禁用 1：使能
16 MEEN	**存储 ECC 使能** 0：禁用 1：使能 存储 ECC 使能后，会自动初始化将存储区域的数据为 0；存储区域重新上电后，如需开启 ECC 功能，需要将 MEEN 重新先写 0 再写 1。 该位只有在 RESET = 1 时才可写。

© 2013 - 2024 杰发科技有限公司

字段	说明
15：0 VERSION	**CAN 控制器版本**

26.6.12 存储错误注入寄存器(CAN_MEMES)

表 26- 30 CAN_MEMES 寄存器

CAN_MEMES **存储错误注入寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称						MES	MEL		MENEC				MEEEC			
访问						RW	RW		RW				RW			
复位值						0	0		0				0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		ME2EE	MEBP2							ME1EE	MEBP1					
访问		RW	RW							RW	RW					
复位值		0	0							0	0					

字段	说明
26 MES	**错误注入端** 0：主控制器端（MCU 访问寄存器） 1：CAN 控制器端
25：24 MEL	**错误注入位置** 00b：ACF 01b：TBUF 10b：RBUF 11b：保留
23：20 MENEC	**不发生错误注入的访问次数** MENEC 每次访问（在 MEL 和 MES 定义的位置）自动减 1，直到减成 0。
19：16 MEEEC	**错误注入的访问次数** 当 MENEC 为 0 时，每次访问（在 MEL 和 MES 定义的位置）注入错误，并且 MEEEC 自动减 1，直到减成 0。
14 ME2EE	**第 2 个错误注入位置使能** 0：禁用 1：使能
13：8 MEBP2	**第 2 个错误注入位置**

© 2013 - 2024 杰发科技有限公司

字段	说明
	在访问存储时，在第 2 个位置注入 1 个位的错误（可设置 0 到 38，需与第 1 个位置值不同）。
6 ME1EE	**第 1 个错误注入位置使能** 0：禁用 1：使能
5：0 MEBP1	**第 1 个错误注入位置** 在访问存储时，在第 1 个位置注入 1 个位的错误（可设置 0 到 38，大于 38 无作用）。

26.6.13 唤醒配置寄存器(CAN_WAKEUP)

表 26-31 CAN_WAKEUP 寄存器

CAN_WAKEUP **唤醒配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															WUEN	LPFEN
访问															RW	RW
复位值															0	0

字段	说明
1 WUEN	**低电平唤醒使能** 0：禁用 1：使能
0 LPFEN	**低通滤波器使能** 0：禁用 1：使能

© 2013 - 2024 杰发科技有限公司

27 集成电路总线（I2C）

27.1 简介

I2C（Inter-Integrated Circuit） 总线是一种简单、双向二线制同步串行总线，通过时钟线（SCL）与数据线（SDA）进行数据传输。SCL 是由主机驱动产生的时钟信号，SDA 是双向数据信号，既可由主设备（主机）产生，也可由从设备（从机）产生。

27.2 特性

- 主从一体模式
- 标准、快速、快速+ 模式
- 支持 7bit 范围地址
- 支持 10bit 扩展地址
- 支持从机 Stretch
- 支持从机低功耗模式唤醒
- 支持从机监测功能
- 支持多主机仲裁
- 主机仲裁丢失切换为从机
- 可编程毛刺滤波器
- 总线开始/停止信号检测
- 软件控制应答信号
- DMA 发送/接收
- 地址匹配中断
- 字节传输中断
- 无应答中断
- 发送/接收溢出中断
- SDA/SCL 低超时中断
- 支持时钟同步
- 支持 SMBus 2.0 – 警告地址
- 支持 SDA/SCL 低超时检测

© 2013 - 2024 杰发科技有限公司

27.3 结构框图

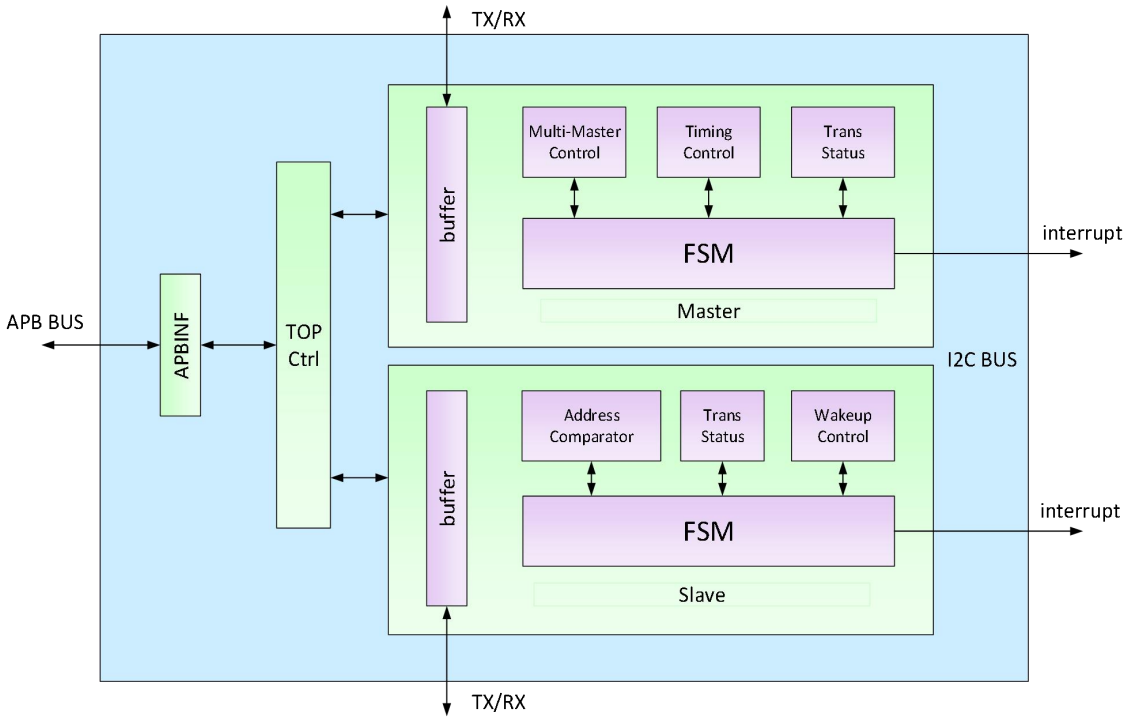


图 27-1 I2C 结构框图

27.3.1 I2C 信号组成

I2C 通讯以起始信号开始，以停止信号结束。当 SCL 为高电平时，SDA 线上高电平到低电平的跳变定义了 START 条件。当 SCL 为高电平时，SDA 线上低电平到高电平的跳变定义了 STOP 条件。

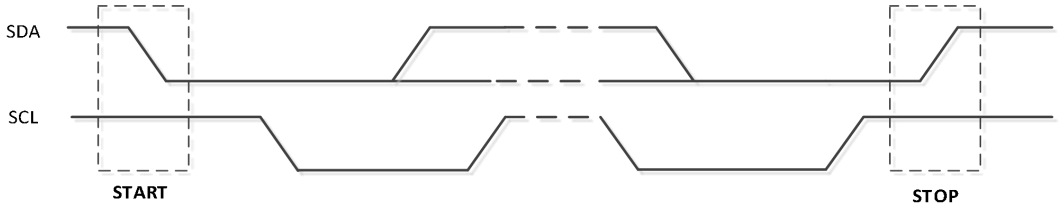


图 27-2 START 和 STOP 条件

每帧数据由 9 位组成， 8 位数据（MSB 在前） 和 1 位应答信号，传输次数不受限制。

© 2013 - 2024 杰发科技有限公司

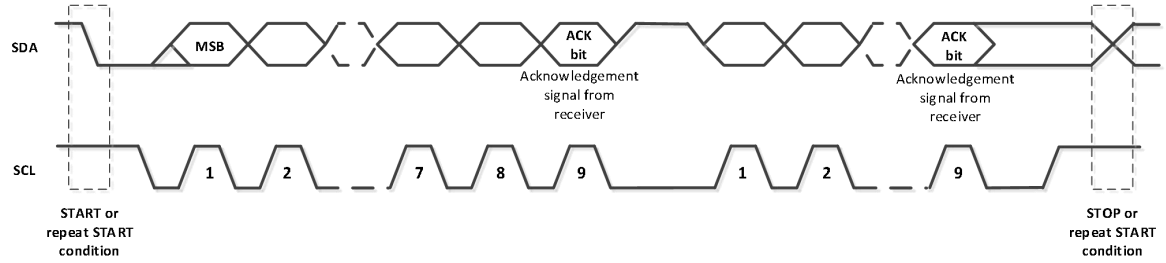


图 27-3 数据传输格式

起始信号后的第一个数据是地址字节，地址字节后的继续传输的数据都是数据字节。

7bit 地址模式下，I2C 协议规定地址字节的高 7 位是从机的地址，最低位为 0，表示写入从机，最低位为 1，表示读取从机。

10bit 地址模式下，地址由两个字节组成，写从机时，I2C 协议规定：第一个字节发送 11110XX0，高 5 位固定为 11110，bit2，bit1 是 10bit 地址中的高 2 位，bit0 是方向位，值为 0，表示方向为写；第二个字节是 10bit 地址的低 8 位。读从机时，I2C 规定：先发写的地址（两个字节），再发读的地址（读的地址只需要发送一个字节）。具体流程如下：先发写的地址，第一个字节发送 11110XX0，高 5 位固定为 11110，bit2、bit1 是 10bit 地址的高 2 位，bit0 是方向位，值为 0，表示方向为写，第二个字节是 10bit 地址的低 8 位；再发读的地址，发送 11110XX1，bit2、bit1 是 10bit 地址中的高 2 位，bit0 是方向位，值为 1，表示方向为读。

27.3.2 波特率组成

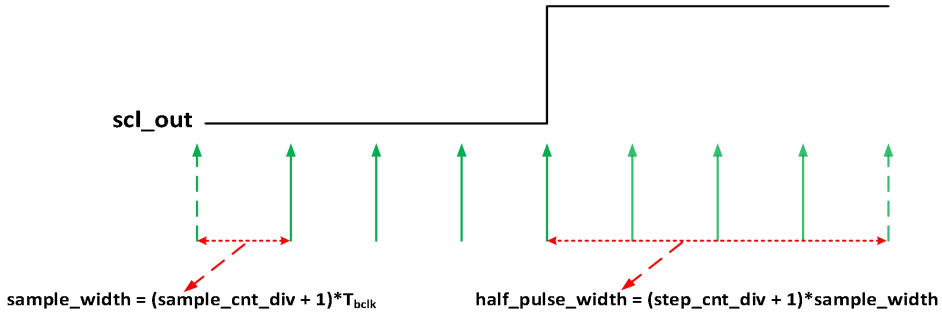


图 27-4 波特率生成

波特率： $f_{SCL}=f_{fclk}/(((SAMPLE_CNT_DIV+1)*(STEP_CNT_DIV+1))*2)$

说明：

I2C 配置为 Slave，I2C 功能时钟 Fclk 需要大于等于 20*波特率的关系。

f_{fclk} 是 I2C 功能时钟频率。

© 2013 - 2024 杰发科技有限公司

27.3.3 数据流程

作为发送器，数据被写入一个 8 位的发送缓冲区。然后由波特率控制器加载到发送移位寄存器。MSB 在前，发送器会在每个字节的第 9 个 SCL 采样应答位。

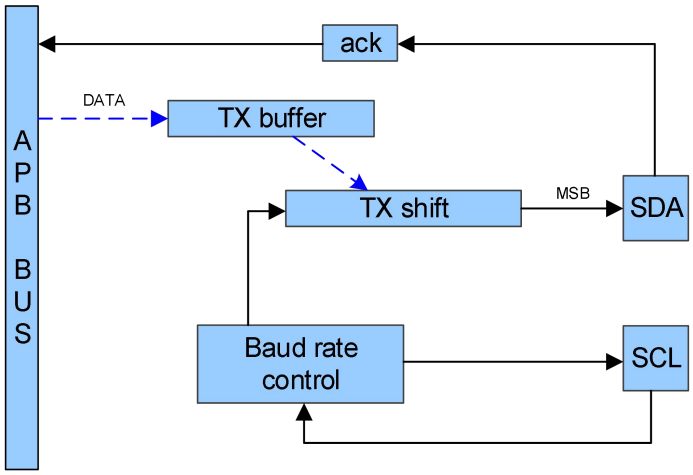


图 27-5 发送器数据流程

对于接收器，接收移位寄存接收每个比特位并依次传入移位寄存器，数据在每第 8 个 SCL 之后将被存储到 8 位接收缓冲器中，MSB 在前。应答位在第 9 个 SCL 驱动到 SDA 线上。

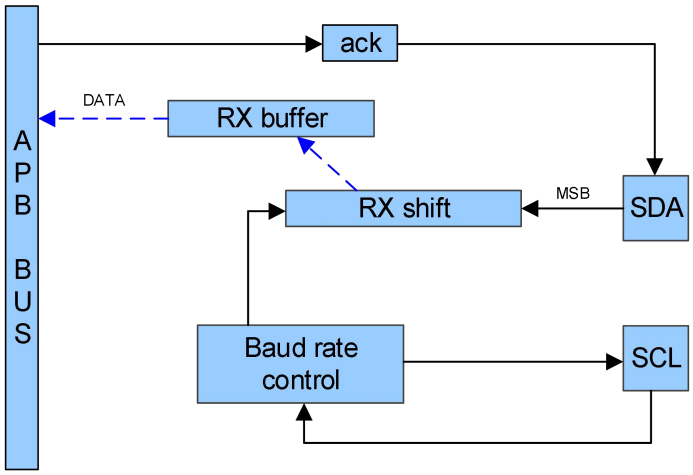


图 27-6 接收器数据流程

27.4 功能描述

I2C 模块支持标准、快速与快速+通信模式，I2C 外设同一时刻只能当做主机或从机模式使用。一旦配置为从机模式，除非本模块复位，否则会一直处于从机模式下。

27.4.1 主机模式

在主机模式下，主机发送起始信号后，每发送完一字节数据，I2C_STATUS0[BND]标志位都会置位。

© 2013 - 2024 杰发科技有限公司

主机可发送重复起始信号，在不发送 STOP 信号前提下，重新对从机寻址并写入或读取数据，如图 27-7 所示。

主机支持时钟同步，在多主从机通讯或者有从机 stretch 总线时，能与其它主从机的时钟进行同步：当主机 SCL 输出高时，不会开始它的高电平时间计数，而是等待总线的 SCL 被所有主机拉高后开始计数，这样实现了多主机之间的时钟同步，此时实际的 SCL 时钟频率会变慢，因为 SCL 拉高是通过外部上拉电阻充电拉高，此时的频率取决于外部总线电容和上拉电阻的取值。

支持主机仲裁，在多主机通讯情况下，I2C 模块对总线上传输的数据进行逐 bit 比较。当主机发送 1 而检测到总线数据为 0 时，仲裁丢失，置位仲裁丢失标志，可产生中断请求，并将自己切换为从机模式，因为此时其它主机可能在寻址自己。注意，该功能的前提条件是时钟同步功能先打开，同时在主机仲裁失败切换为从机模式后将会一直保持从机状态，直到被重新初始化。

在 DMA 模式下，发送或者接收时，DMA 的使能信号会屏蔽字节结束中断产生，若使能 DMA 传输完成字节结束中断即（I2C_CTRL2[BND_DMA_IE] = 1）且 I2C_CTRL3[DMATXEN] = 1，在最后一个字节传输完成时会产生字节结束中断。具体是：DMA 发送过程中，字节结束标志都不会置位，不会产生字节结束中断，当最后一个字节发送完成时，字节结束标志置起，产生字节结束中断；DMA 接收过程中，字节结束标志都不会置位，不会产生字节结束中断，当最后一字节接收完后，产生字节结束中断，字节结束标志被自动清除。

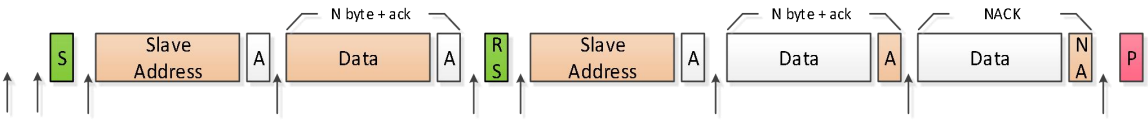


图 27-7 主机组合模式

27.4.2 从机模式

在从机模式下，从机支持 7bit 地址、10bit 地址、7bit 范围地址、广播地址四种地址匹配模式。

从机地址匹配后，I2C_STATUS0[SAMF]标志位置位，I2C_STATUS0[SRW]标志位根据主机的写/读信号置 0/1。地址匹配成功后，主机后续传输的每一个字节数据，从机的字节结束标志位都会置位。如果主机在读取从机时，发送了 NACK 信号，后续主机继续传输，从机不会产生字节结束标志，需要重新发起寻址，才能再次与从机建立通讯。

从机具备 stretch 功能，使能后，从机收到一字节数据但软件没有读取，或者主机读取时从机还没有将数据准备好，从机将主动拉低 SCL 总线，防止主机发出下一帧数据的时钟信号，当然主机需要使能时钟同步功能，以等待从机释放 SCL 总线。

从机具有以下状态标志：

- **发送缓存区空标志 TXEF**，当从机发送寄存器未载入发送数据或一字节发送完成时，此标志置位，可以产生中断请求。
- **发送缓存区溢出标志 TXUF**，当从机发送完一字节数据，但没有写入下一字节数据，而此时主机再次发完一字节时钟信号后，此标志置位，可以产生中断请求。

© 2013 - 2024 杰发科技有限公司

- **接收缓存区满标志 RXFF**，当从机数据寄存器收到一字节数据后，此标志置位，可以产生中断请求。
- **接收缓存区溢出标志 RXOF**，当从机接收完一字节数据后，若软件没有及时读取数据，而此时主机再次发完一字节时钟信号后，此标志置位，可以产生中断请求。

说明：
I2C 配置为 Slave， 当总线上 master 性能优于 slave 时，不推荐 master 使用 DMA。

27.4.3 DMA 发送/接收

I2C 模块支持逐字节响应 DMA 发送/接收请求。DMA 仅用于数据传输阶段， START、STOP、RESTART 信号以及发地址信号仍需由软件操作完成。

27.4.3.1 主机发送器

在主机 DMA 发送模式下，地址匹配后，DMA 会自动搬运内存数据至 I2C 数据寄存器。后续每传输完成一个字节数据时检测 ACK 信号，当检测为 ACK 时，产生 DMA 请求，搬运下一字节数据，当检测为 NACK 时，DMA 不会进行后续的传输，如果 I2C_CTRL2[NACKIE] = 1，将产生 NACK 中断请求。在最后一字节传输完成后， 如果 I2C_CTRL2[BND_DMA_IE] = 1，将产生字节结束中断请求。

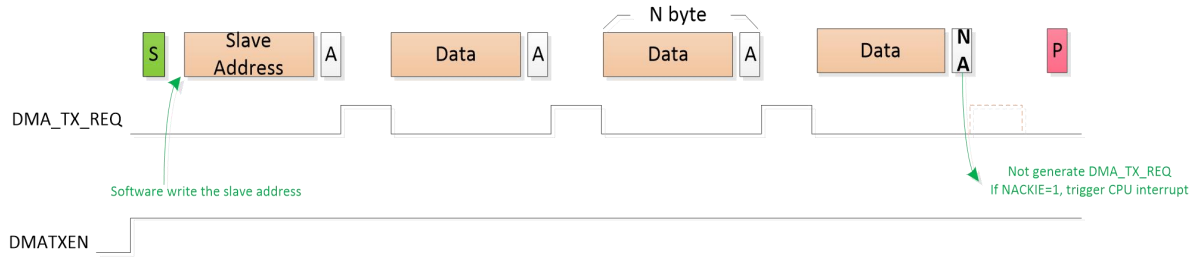


图 27-8 主机发送器情形 1

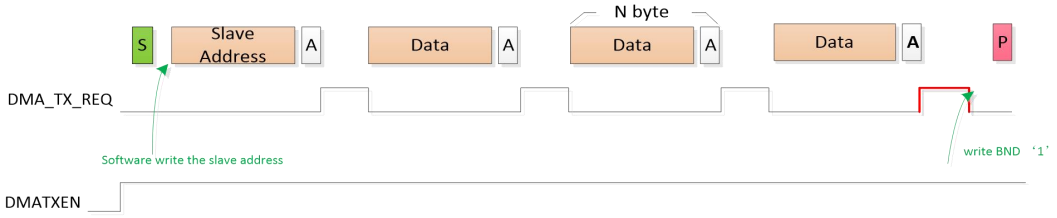


图 27-9 主机发送器情形 2

© 2013 - 2024 杰发科技有限公司

27.4.3.2 主机接收器

在主机 DMA 接收模式下，**地址匹配后，必须先手动清除字节结束标志**，然后将数据传输方向切换为读取，并执行空读操作产生时钟信号，一字节传输完毕后字节结束标志置起会请求 DMA 将数据搬运到内存。DMA 会根据设定的传输长度自动在最后一字节回 NACK。在最后一字节传输完成后， 如果 I2C_CTRL2[BND_DMA_IE] = 1 且 I2C_CTRL3[DMARXEN] = 1，将产生字节结束中断请求，字节结束标志自动清零。

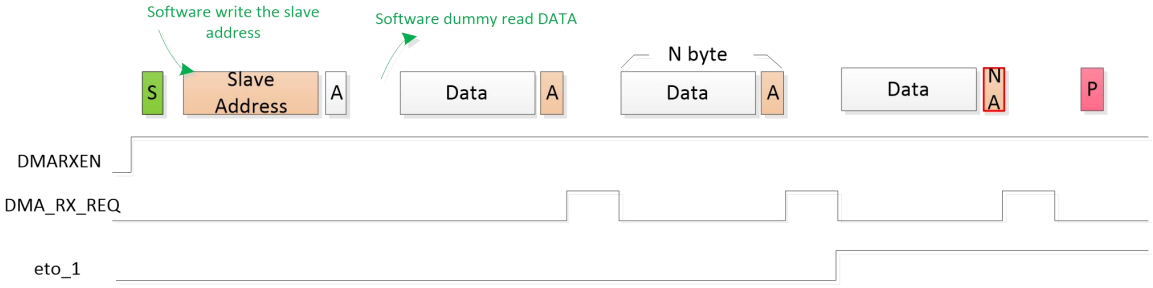


图 27-10 主机接收器

27.4.3.3 从机发送器

在从机 DMA 发送模式下，如果地址匹配且为读信号，则触发 DMA 将数据搬运至数据寄存器。并检测后续每一字节的 ACK 信号，收到 ACK 信号时会产生 DMA 搬运请求，收到 NACK 信号时，DMA 会停止搬运，若 I2C_CTRL2[NACKIE] = 1，产生 NACK 中断请求。在最后一字节传输完成后， 如果 I2C_CTRL2[BND_DMA_IE] = 1，将产生字节结束中断请求。

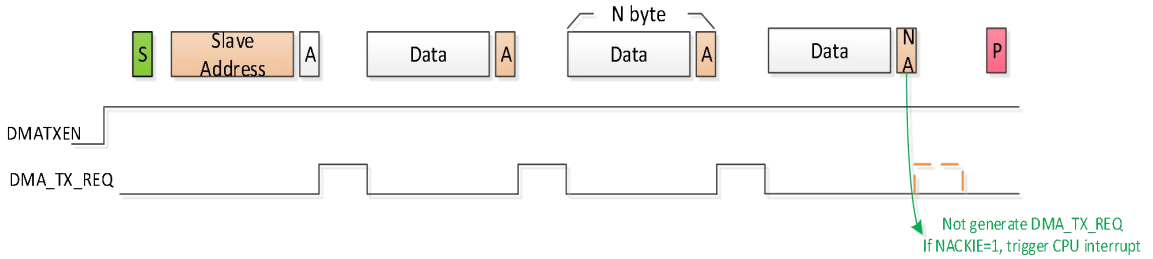


图 27-11 从机发送器

27.4.3.4 从机接收器

在 Slave DMA 接收模式下，**地址匹配后，必须先手动清除地址匹配标志**，然后每当接收到一字节数据产生的 I2C_STATUS0[BND]标志都将触发 DMA 将数据搬运到内存。DMA 会根据设定的传输长度自动在最后一字节回 NACK。在最后一字节传输完成后， 如果 I2C_CTRL2[BND_DMA_IE] = 1 且 I2C_CTRL3[DMARXEN] = 1，将产生字节结束中断请求，字节结束标志位自动清零。

© 2013 - 2024 杰发科技有限公司

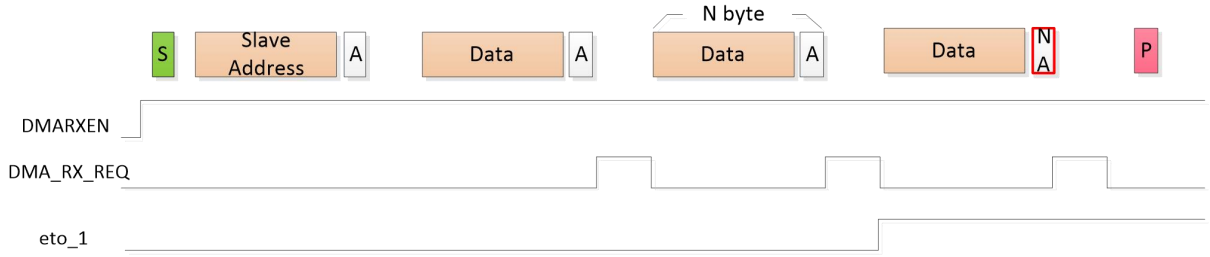


图 27-12 从机接收器

27.4.4 SMBus

I2C 模块支持系统管理总线（SMBus）2.0 规范。SMBus 是一种简单的单端两线总线，通常用于低带宽通信。

27.4.4.1 SMBus – SCL/SDA 低超时检测

I2C 用作主机或从机时，支持 SCL/SDA 低超时检测。超时时间可软件配置。默认检测 SCL，可选择是否检测 SDA。若 I2C_CTRL2[PLTIE] = 1，PLTF 置位时将产生中断请求。

27.4.4.2 SMBus –警告地址响应

I2C 作为从机可响应 SMBus 的警告地址 0001100X。从机在地址得到匹配后，I2C_STATUS0[SAMF]标志置位时 I2C_STATUS1[SARF]标志同时置位。

27.4.5 中断请求

I2C 模块共有 11 个中断源。

表 27-1 I2C 中断汇总

中断源	标志	本地使能	全局使能
完成 1 个字节的传输	BND	BND_DMA_IE[1]	IICIE
从机地址匹配	SAMF		IICIE
仲裁丢失	ARBLOST		IICIE
总线起始信号检测	START	SSIE	IICIE
总线停止信号检测	STOP	SSIE	IICIE
RX 缓冲区溢出	RXOF	RXOFIE	IICIE
TX 缓冲区溢出	TXUF	TXUFIE	IICIE
RX 缓冲区满	RXFF	RXFIE	IICIE
TX 缓冲区空	TXEF	TXEIE	IICIE
获得应答	RACK	NACKIE	IICIE
SDA/SCL 低超时	PLTF	PLTIE	IICIE

© 2013 - 2024 杰发科技有限公司

1：只有在 DMA 传输方式下，I2C_CTRL2[BND_DMA_IE]有效，且在 DMA 传输过程中不会响应 I2C_STATUS0[BND]的中断请求，只有在传输完成最后一字节数据才会响应字节结束中断请求。

在 DMA 传输方式下，不会响应 RX 缓冲区满及 TX 缓冲区空的中断请求。

27.4.6 低功耗模式

MCU 在 STOP1/STOP2/VLPS 模式下，I2C 用作从机，若使能唤醒功能（设置 I2C_CTRL1[WUEN] = 1）、I2C 全局中断（I2C_CTRL0[IICIE] = 1），则地址匹配时，将在第 9 个 SCL 处产生 ACK 低电平信号，并产生唤醒信号唤醒 MCU，唤醒后可开始传输数据。若要失能唤醒功能，则需设置 I2C_CTRL0[WUEN] = 0，I2C_CTRL0[IICIE] = 0。

表 27-2 I2C 唤醒源

模式	唤醒源
STOP1	地址匹配中断
STOP2	地址匹配中断
VLPS	地址匹配中断

说明：

1.系统进入 STOP1/STOP2/VLPS 时，如果 I2C 正处于忙状态(master 正在接收/发送数据，或 slave 正在接收/发送数据，或 slave 地址匹配)，则不会回复休眠应答确认 ACK 信号，造成休眠超时。

2.如果使用 I2C 唤醒 VLPS 模式，I2C 功能时钟和系统时钟必须是同一个时钟源。

3.使用 I2C 唤醒并传输数据时，主机发送地址和第一个数据的间隔时间需要大于从机芯片的唤醒时间。

27.4.7 调试模式

当 I2C_CTRL0[DBGEN] = 1 时，调试模式下 I2C 传输遇到 Debug Halt 状态会自动产生停止信号。否则不会自动产生停止信号。

27.5 应用说明

27.5.1 数据传输

写 I2C_STARTSTOP[START]为 '1' 发送 START 信号到 I2C 总线。传输结束写 STARTSTOP[STOP]为 '1' 发送 STOP 信号到 I2C 总线。

© 2013 - 2024 杰发科技有限公司

主机发送时，I2C_CTRL0[TX]控制置位 1，对数据寄存器写入后，波特率控制器自动开始向 I2C 总线传输数据，一个字节传输完成后，I2C_STATUS0[BND]标志置位，表明可以再次写入数据，写入数据或向 I2C_STATUS0[BND]位写 1 可清除字节结束标志。

主机接收时，I2C_CTRL0[TX]控制位置 0，对数据寄存器的读操作，会触发波特率控制器自动向 I2C 总线发出时钟信号，并根据 I2C_CTRL0[TACK]控制应答，收到的数据载入数据寄存器。字节结束标志位置位，对数据寄存器进行读取或向 I2C_STATUS0[BND]位写 1 可清除字节结束标志。

从机发送时，先将待发送数据写入数据寄存器，当收到主机读取的完整时钟信号后，数据被传出，I2C_STATUS1[TXEF]标志位置位，字节结束标志位置位，对数据寄存器的写入操作可清除 TXEF 和字节结束标志位，或向 I2C_STATUS0[BND]位写 1 也可清除字节结束标志。

从机接收时，主机发送一字节完整数据后，数据被载入从机数据寄存器，I2C_STATUS1[RXFF]标志位置位，字节结束标志位置位，读取数据寄存器可清除 RXFF 和字节结束标志位，或向 I2C_STATUS0[BND]位写 1 也可清除字节结束标志。

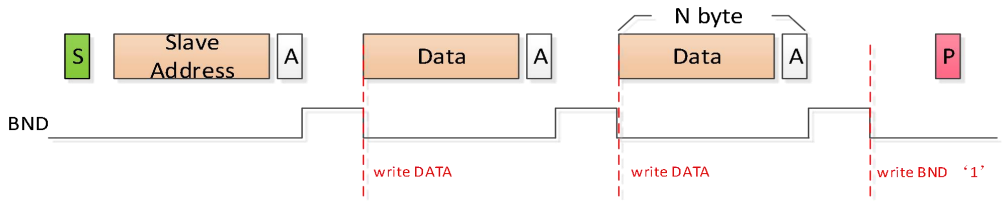


图 27-13 主机写从机模式的 BND 序列

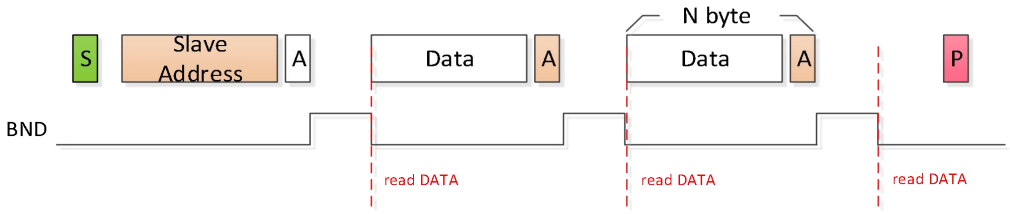


图 27-14 主机读从机模式的 BND 序列

© 2013 - 2024 杰发科技有限公司

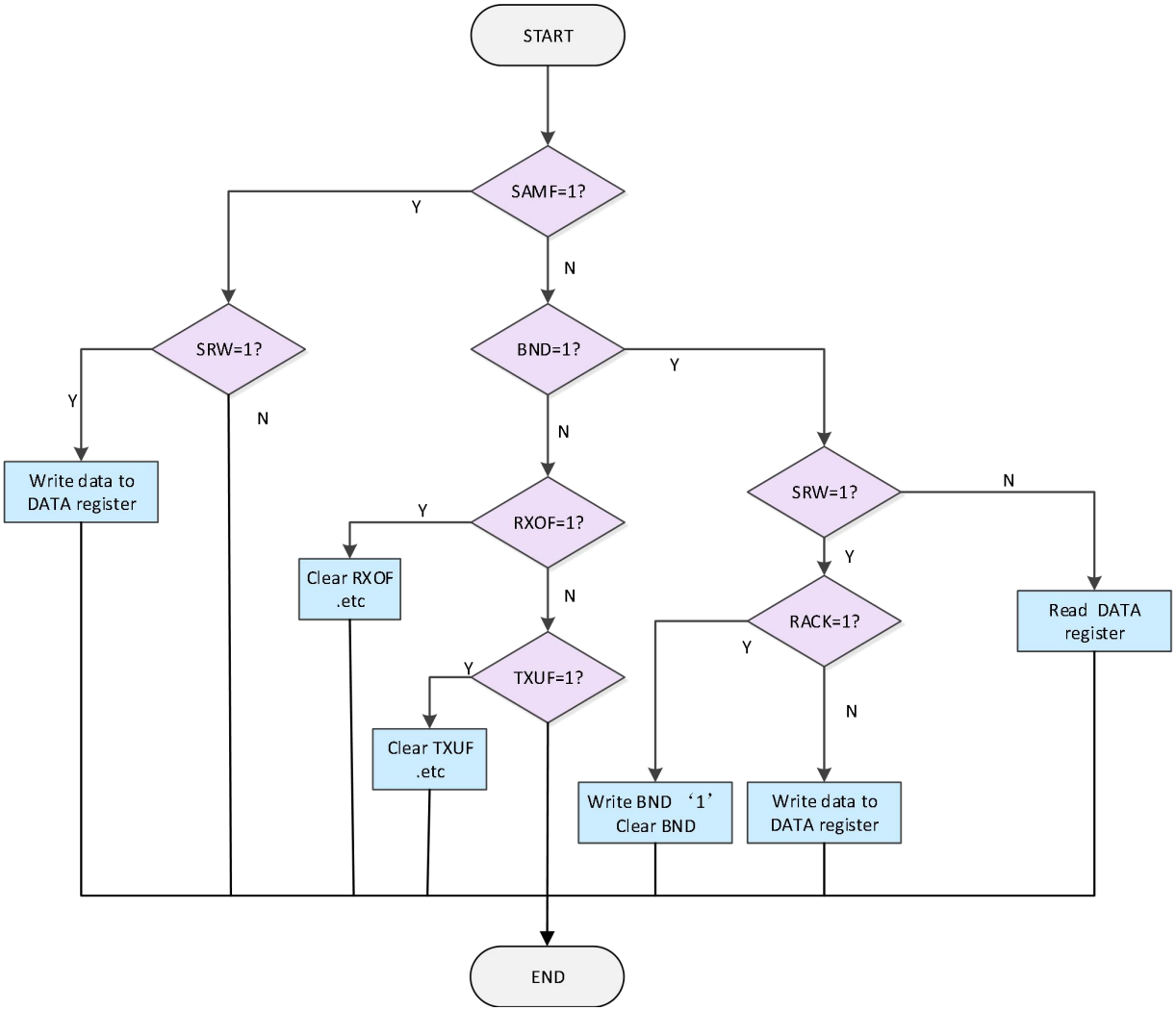


图 27- 15 典型 I2C 从机中断程序

27.5.2 应答控制

I2C 模块通过软件方式控制应答信号。

在主机接收数据时，I2C_CTRL0[TACK]位的值决定下一字节读取完后，在第 9 个时钟信号上的应答值。

在从机接收数据时，I2C_CTRL0[TACK]位的值决定下一字节读取完后，在第 9 个时钟信号上的应答值。

27.5.3 编程指南

主机：

1. 写 I2C_SAMPLE_CNT、I2C_STEP_CNT 寄存器，设置波特率；
2. 写 I2C 控制寄存器，根据要求设置工作模式、中断、DMA、数据传输方向等；

© 2013 - 2024 杰发科技有限公司

3. 置位 I2C_STARTSTOP[START]，触发开始信号；
4. 等待 I2C_STATUS0[BUSY] = 1，I2C_STATUS0[READY] = 1 后向 I2C_DATA 寄存器写入从机读/写地址。若为读地址，在地址发送完成后，空读 I2C_DATA 寄存器；
5. 若传输方向为主机写，接收到从机 ACK，且当前数据已传输完成 I2C_STATUS0[RACK] = 0，I2C_STATUS0[BND] = 1，写数据到 I2C_DATA；若传输方向为主机读，当 I2C_STATUS0[BND] = 1，读 I2C_DATA 寄存器并写 I2C_CTRL0[TACK]向从机回复 ACK 或 NACK；
6. 重复步骤 5 继续数据传输；
7. 数据传输完成后，置位 I2C_STARTSTOP[STOP]，触发停止信号。

从机：

1. 写 I2C_ADDR0，I2C_ADDR1 寄存器，设置从机地址；
2. 写 I2C 控制寄存器，设置为工作模式、中断等；
3. 等待 I2C_STATUS0[SAMF] = 1 地址匹配后，读 I2C_STATUS0[SRW]位确认当前的传输方向；
4. 若 I2C_STATUS0[SRW] = 1，则写 I2C_DATA 寄存器，否则读 I2C_DATA 寄存器并写 I2C_CTRL0[TACK]向主机回复 ACK 或 NACK；
5. 等待 I2C_STATUS0[BND] = 1，重复步骤 4 继续数据传输直到检测到总线上有停止信号。

27.6 寄存器定义

I2C 寄存器只支持 32 位访问。

表 27-3 I2C 寄存器映射

I2C0 基地址：0x4000E000

地址	名称	宽度	描述
I2Cx 基地址+0x00	I2C_ADDR0	32	地址寄存器 0
I2Cx 基地址+0x04	I2C_ADDR1	32	地址寄存器 1
I2Cx 基地址+0x08	I2C_SAMPLE_CNT	32	波特率配置寄存器 0
I2Cx 基地址+0x0C	I2C_STEP_CNT	32	波特率配置寄存器 1
I2Cx 基地址+0x10	I2C_CTRL0	32	控制寄存器 0
I2Cx 基地址+0x14	I2C_CTRL1	32	控制寄存器 1
I2Cx 基地址+0x18	I2C_CTRL2	32	控制寄存器 2
I2Cx 基地址+0x1C	I2C_CTRL3	32	控制寄存器 3
I2Cx 基地址+0x20	I2C_STATUS0	32	状态寄存器 0
I2Cx 基地址+0x24	I2C_STATUS1	32	状态寄存器 1
I2Cx 基地址+0x28	I2C_DGLCFG	32	毛刺滤波配置寄存器
I2Cx 基地址+0x2C	I2C_DATA	32	数据寄存器

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
I2Cx 基地址+0x30	I2C_STARTSTOP	32	起始与停止信号控制寄存器

说明：
上表中，x = 0。

27.6.1 地址寄存器 0（I2C_ADDR0）

表 27-4 I2C_ADDR0 寄存器

I2C_ADDR0　　地址寄存器 0　　复位值：0x000000FE

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									AD [6: 0]							
访问									RW							
复位值									1	1	1	1	1	1	1	

字段	说明
7: 1 AD[6: 0]	**I2C 从机 7Bit 地址** 在从机模式下设置 7 位地址或 10 位地址中的低 7 位。

27.6.2 地址寄存器 1（I2C_ADDR1）

表 27-5 I2C_ADDR1 寄存器

I2C_ADDR1　　地址寄存器 1　　复位值：0x000007F7

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				RMEN		RAD								AD[9: 7]		
访问				RW		RW								RW		
复位值				0		1	1	1	1	1	1	1		1	1	1

字段	说明
12 RMEN	**从机模式时，范围地址使能位** 0： 禁用 1： 使能 范围地址使能控制位。
10: 4 RAD [6: 0]	**从机模式时，范围地址值** 当范围地址使能时，从机接收到的地址大于 AD[6: 0]且小于等于 RAD[6:0]时，将得到匹配。
2: 0 AD [9: 7]	**I2C 从机 10Bit 地址** 在从机模式下设置 10 位地址的高 3 位。

27.6.3 波特率配置寄存器 0（I2C_SAMPLE_CNT）

表 27-6 I2C_SAMPLE_CNT 寄存器

I2C_SAMPLE_CNT **波特率配置寄存器 0** **复位值：0x00000004**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									SAMPLE_CNT							
访问									RW							
复位值									0	0	0	0	0	1	0	0

字段	说明
7: 0 SAMPLE_CNT	**调整每个采样点的宽度** 采样周期 = (SAMPLE_CNT +1) *功能时钟周期

© 2013 - 2024 杰发科技有限公司

27.6.4 波特率配置寄存器 1（I2C_STEP_CNT）

表 27-7 I2C_STEP_CNT 寄存器

I2C_STEP_CNT 波特率配置寄存器 1 复位值：0x00000004

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									STEP_CNT							
访问									RW							
复位值									0	0	0	0	0	1	0	0

字段	说明
7: 0 STEP_CNT	**调整半脉冲宽度** 半波特率周期= (STEP_CNT +1)*采样宽度 **注意：** **STEP_CNT 最小值为 3。**

27.6.5 控制寄存器 0（I2C_CTRL0）

表 27-8 I2C_CTRL0 寄存器

I2C_CTRL0 控制寄存器 0 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									IICEN	IICIE	MSTR	TX	TACK	WUEN	DBGEN	SRST
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 IICEN	**I2C 模块使能** 0： 禁用

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 使能
6 IICIE	**I2C 全局中断使能** 0： 禁用 1： 使能 I2C 模块全局中断
5 MSTR	**I2C 模式选择** 0： 从机模式 1： 主机模式 **注意：** **如果 ARB_LOST 置位时，该位会自动清零。**
4 TX	**传输方向选择** 0： 接收(RX) 1： 发送(TX) 主要作用在主机模式下的读取：TX=0，读取 DATA 寄存器会触发读时钟； TX=1，读取 DATA 寄存器不会触发读时钟。
3 TACK	**应答控制** 0： 向接下来的接收字节上的总线发送 ACK 信号 1： 向接下来的接收字节上的总线发送 NACK 信号 指定在主机或从机的数据应答周期过程中驱动到 SDA 的值。
2 WUEN	**唤醒功能使能** 0： 禁用唤醒功能 1： 在低功耗模式下使能唤醒功能 当出现从机地址匹配时，I2C 模块可将 MCU 从低功耗模式唤醒。 **注意：** **使能唤醒功能时，需同步使能 I2C 中断。当 I2C 模块处于从机模式并进入低功耗模式时，发生 7bit、10bit（ADEXT=1），或通用调用广播地址（GCAEN=1）匹配，或范围地址匹配（RMEN=1）时，或警告地址匹配（SAEN = 1） I2C 模块将唤醒 MCU。** **禁用唤醒功能时，需同步失能 I2C 中断（IICIE = 0）。**
1 DBGEN	**Debug 功能使能** 0： 调试模式下 I2C 传输遇到 Debug Halt 状态会自动产生停止信号 1： 调试模式下 I2C 传输遇到 Debug Halt 状态不会自动产生停止信号
0 SRST	**I2C 软件复位功能使能** 0： 不复位主机/从机传输逻辑

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 复位主机/从机传输逻辑 复位内部主、从逻辑和状态寄存器。该位只可被 POR 或软件清除。

27.6.6 控制寄存器 1（I2C_CTRL1）

表 27-9 I2C_CTRL1 寄存器

I2C_CTRL1 **控制寄存器 1** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									GCAEN	ADEXT		SYNCEN	ARBEN		SAEN	STREN
访问									RW	RW		RW	RW		RW	RW
复位值									0	0		0	0		0	0

字段	说明
7 GCAEN	**从机通用广播地址使能** 0： 禁用 1： 使能
6 ADEXT	**从机地址扩展使能** 0： 7 位地址模式 1： 10 位地址模式
4 SYNCEN	**主机 SCL 同步使能** 0： 禁用 1： 使能 使能主机的 SCL 同步功能。
3 ARBEN	**主机仲裁使能** 0： 禁用 1： 使能 使能主机仲裁功能。 **注意：**

© 2013 - 2024 杰发科技有限公司

字段	说明
	如果在多主机系统中使用 I2C，则多主机功能必须同时设置 SYNC_EN 和 ARB_EN。 如果 I2C 在单主机系统中使用，且从机具有 SCL 拉伸功能，则可以只设置 SYNC_EN。
1 SAEN	从机 SMBus 警告地址使能 0： 禁用 1： 使能
0 STREN	从机 SCL 拉伸使能 0： 禁用 1： 使能 使能此位，从机将在每个字节第 9 个 SCL 下降沿后将 SCL 拉低。 注意： 在 SAMF=1 后，如果 SRW=1， TXEF=1 将产生 scl stretch，如果 SRW=0, RXFF=1 将产生 scl stretch。因此当使能 STREN 且从机为发送器时(TX)，在每个字节（包括地址字节）的第 9 个 SCL 下降沿后，从机会拉低 SCL，直到写 DATA 寄存器。类似地，当从机为接收器时(RX),在每个字节的第 9 个 SCL 下降沿后，从机会拉低 SCL，直到读取数据寄存器。 在从机模式下，当使能 MNTEN 时，从机不会拉低 SCL。

27.6.7 控制寄存器 2（I2C_CTRL2）

表 27-10 I2C_CTRL2 寄存器

I2C_CTRL2 控制寄存器 2 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									RXOFIE	TXUFIE	RXFIE	TXEMIE	BND_DMA_IE	PLTIE	NACKIE	MNTEN
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 RXOFIE	从机接收缓存区溢出中断使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 禁用 1： 使能
6 TXUFIE	**从机发送缓存溢出中断使能** 0： 禁用 1： 使能
5 RXFIE	**从机接收缓存满中断使能** 0： 禁用 1： 使能
4 TXEMIE	**从机发送缓存空中断使能** 0： 禁用 1： 使能
3 BND_DMA_IE	**DMA 传输完成字节结束中断使能** 0： 禁用 1： 使能
2 PLTIE	**SDA/SCL 低超时中断使能** 0： 禁用 1： 使能
1 NACKIE	**NACK 中断使能** 0： 禁用 1： 使能
0 MNTEN	**从机监测功能使能** 0： 禁用 1： 使能

© 2013 - 2024 杰发科技有限公司

27.6.8 控制寄存器 3（I2C_CTRL3）

表 27-11 I2C_CTRL3 寄存器

I2C_CTRL3 控制寄存器 3 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PINLOW													TIMECFG	DMARXEN	DMATXEN
访问	RW													RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15:3 PINLOW	**SDA/SCL 低超时时间** 设置 SDA 或 SCL 低电平检测超时时间。当 SCL 或 SDA 低电平时间超过 PINLOW * 256 个时钟周期，会置位 PLTF。当 PINLOW 值为 0 时，则禁用 SDA/SCL 低超时检测功能。
2 TIMECFG	**超时检测类型** 0：SCL 1：SDA/SCL
1 DMARXEN	**DMA 接收使能** 0： 禁用 1： 使能
0 DMATXEN	**DMA 发送使能** 0： 禁用 1： 使能

© 2013 - 2024 杰发科技有限公司

27.6.9 状态寄存器 0（I2C_STATUS0）

表 27-12 I2C_STATUS0 寄存器

I2C_STATUS0 **状态寄存器 0** **复位值：0x00000008**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									BND	SAMF	BUSY	ARBLOST	READY	SRW	A10MF	RACK
访问									R/W 1C	R/W 1C	RO	R/W1C	RO	RO	RO	R/W 1C
复位值									0	0	0	0	1	0	0	0

字段	说明
7 BND	**字节结束标志** 0： 传输进行中，一个字节传输未结束 1： 一个字节传输完成(包括 ACK 位，共 9 个 SCL) 复位后，*BND* 为‘0’。*BND* 仅在总线上 START 和 STOP 信号之间的数据传输期间设置。BND 会在每 9 个 SCL 下降沿之后设置。 在 Master 模式下，发送数据时，软件写 DATA 寄存器会清零此位，并发出 DATA；读取数据时，一字节传输完成，软件读取 DATA 寄存器会清零此位，并发出下一字节数据时钟。 在 Slave 模式下，地址匹配后置位 SAMF，此时 BND 标志不会置位，BND 在地址匹配后的每次数据传输置位。具体为：地址匹配后，主机写数据，从机置位 BND，当从机主动 NACK 后，主机若继续写数据，从机不再置位 BND；主机读取数据，从机置位 BND，若收到主机的 NACK，主机继续读数据，从机不再置位 BND。 **注意：** **当 I2C 为 从机模式且 MNTEN=1，当 BND 为 “1”时，读取 DATA 寄存器将清零此位。** **写 ’1’ 也可以清零此位。**
6 SAMF	**从机地址匹配标志** 0： 地址不匹配 1： 地址匹配 **注意：** **该位在满足如下条件之一时置位 ：** **1. 7 位地址匹配** **2. 10 位地址，第 1st 字节和第 2nd 字节匹配。并且在第 2nd 字节匹配后置位。** **3. 通用广播地址匹配。** **4. 范围地址匹配**

© 2013 - 2024 杰发科技有限公司

字段	说明
	写 “1” 清零此位。
5 BUSY	**总线忙** 0： 总线空闲 1： 总线忙 指示总线的状态，对从机模式和主机模式都有效。当硬件在总线上检测到 START 信号时，该位置位。检测到 STOP 信号时，该位清零。
4 ARBLOST	**仲裁丢失标志** 0： 仲裁未丢失 1： 仲裁丢失 **注意：** **写 “1”清零此位。** **当仲裁丢失时， I2C 主机将切换至从机模式，MSTR 位由硬件清零。**
3 READY	**准备好接收新的命令** 0： 内部硬件未准备好 1： 内部硬件准备好接收软件的新命令 该位指示内部硬件状态，仅对主机模式有效。 **注意：** **此位对主机模式有效，在主机产生 START/STOP 信号时使用。当 I2C 模块处于主机模式时，写 STARTSTOP 寄存器产生 START/STOP 信号后必须等待 READY 位为 1。**
2 SRW	**从机读取/写入方向** 0： 从机接收(RX)，主机向从机写入 1： 从机发送(TX)，主机从从机读取
1 A10MF	**从机 10 位地址匹配标志位** 0：10 位地址不匹配 1：10 位地址匹配
0 RACK	**接收应答** 该字段只在发送方向有效，主机发送或从机发送。 0： 在总线上完成一个字节的数据传送后，接收到应答信号 1： 未检测到应答信号 **注意：若 NACKIE=1, RACK=1 将会产生中断请求，写 1 清零此位 。**

© 2013 - 2024 杰发科技有限公司

27.6.10 状态寄存器 1（I2C_STATUS1）

表 27-13 I2C_STATUS1 寄存器

I2C_STATUS1 状态寄存器 1 复位值：0x00000081

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										SARF	PLTF		RXOF	TXUF	RXFF	TXEF
访问										R/W1C	R/W1C		R/W1C	R/W1C	R/W1C	R/W1C
复位值										0	0		0	0	0	1

字段	说明
6 SARF	**从机 SMBus 警告地址匹配标志** 0：禁用警告地址或未匹配 1：使能警告地址并匹配 **写 “1” 清零此位。**
5 PLTF	**SDA/SCL 低超时标志** 0：未超时 1：SDA/SCL 低超时 **注意：** **当 SDA/SCL 低电平维持时间超过设置的 PINLOW 将置位该标志。** **只要 SDA/SCL 一直为低电平该标志不能被清除。** **在 I2C 发送起始信号前，必须清除该标志。** **写 “1” 清零此位。**
3 RXOF	**从机 RX 缓冲区溢出标志** 0： 未溢出 1： RX 缓冲区溢出 **注意：** **当 RX 缓冲区溢出时，新接收的数据不会存储在 RX 缓冲区中。** **写 “1” 清零此位。**
2 TXUF	**从机 TX 缓冲区溢出标志** 0： 未溢出 1： TX 缓冲区溢出

© 2013 - 2024 杰发科技有限公司

字段	说明
	注意： **当 TX 缓冲区溢出时，再次发送时发送缓冲区的最后一个 DATA。** **写”1” 清零此位。**
1 RXFF	**从机 RX 缓冲区满标志** 0： 未满 1： RX 缓冲区满 **注意：** **读取 *DATA* 寄存器将清零此位，写”1” 也可清零此位。**
0 TXEF	**从机 TX 缓冲区空标志** 0： 非空 1： TX 缓冲区为空 **注意：** **写 DATA 寄存器会清零此位，写”1” 也可清零此位。**

27.6.11 毛刺滤波配置寄存器（I2C_DGLCFG）

表 27- 13 I2C_DGLCFG 寄存器

I2C_DGLCFG **毛刺滤波配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DGLEN	STOPF	SSIE	STARTF	DGL_CNT			
访问									RW	R/W1C	RW	R/W1C	RW			
复位值									0	0	0	0	0	0	0	0

字段	说明
7 DGLEN	**毛刺滤波器使能** 0： 禁用 1： 使能
6 STOPF	**总线 STOP 标志** 0： 在 I2C 总线上未检测到 STOP 信号 1： 在 I2C 总线上检测到 STOP 信号

© 2013 - 2024 杰发科技有限公司

字段	说明
	当 I2C 总线上检测到 STOP 信号时，由硬件置位此位。 **注意：** **写 “1” 清零此位。**
5 SSIE	**总线 STOP 或 START 中断使能** 0： 禁用 1： 使能 STRAT 或 STOP 检测中断 **注意：** **写 ‘1 ’ 清除 STARTF 或 STOPF 标志。**
4 STARTF	**总线 START 标志** 0： 未检测到 START 1： 在 I2C 总线上检测到 START 当 I2C 总线上检测到 START 信号时，由硬件置位此位。 **注意：** **写 “1” 清零此位**
3: 0 DGL_CNT	**毛刺滤波器计数器。** 0h： 无滤波器 / 旁路 1-Fh：对宽度最多为 1-15 个总线时钟周期的毛刺进行过滤 选择毛刺滤波宽度。对于宽度小于或等于所设置滤波宽度的任意毛刺，滤波器不会允许该毛刺通过。

27.6.12 数据寄存器（I2C_DATA）

表 27-14 I2C_DATA 寄存器

I2C_DATA **数据寄存器** **复位值：0x000001FF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								MAK	DATA							
访问								RO	RW							
复位值								1	1	1	1	1	1	1	1	1

© 2013 - 2024 杰发科技有限公司

字段	说明
8 MAK	**从机监控功能 ACK 位** 对于从机监测器，该字段为 I2C 总线的 ACK 位。 **注意：** **MAK='1', NACK** **MAK='0', ACK** **对于监测器，DATA 是 I2C 总线上传送的数据， 第 8 位 为 ACK 位。**
7: 0 DATA	**数据** 在主机传送(TX)模式下，当向该寄存器中写入数据时，会启动数据传输。首先发送最高有效位。在主机接收(RX) 模式下，读该寄存器会触发主机发出读时钟。 **注意：** **当从主机接收模式转换时，在读取 DATA 寄存器之前需要先切换 I2C 模式，以防意外启动主机数据接收传输流程。**

27.6.13 起始与停止信号控制寄存器（I2C_STARTSTOP）

表 27- 15 I2C_STARTSTOP 寄存器

I2C_STARTSTOP **起始停止信号控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															STOP	START
访问															RW	RW
复位值															0	0

字段	说明
1 STOP	**主机发送停止信号** 写"1"，主机会发送 STOP 信号 读取此位会始终返回 "0"
0 START	**主机发送起始信号** 写"1"，主机将会发送 START 或 RESTART 信号 读取此位会始终返回 "0"

© 2013 - 2024 杰发科技有限公司

28 串行外设接口（SPI）

28.1 简介

SPI (Serial Peripheral Interface，串行外设接口)是一种全双工协议的外设总线，通过片选信号(CS0、CS1、CS2、CS3)、时钟线(SCK)、数据线(SIN、SOUT)进行数据传输。SPI 模块支持单主机-单从机通讯，也支持单主机-多从机通讯，通讯总线通过片选信号最多可连接 4 个从机设备。

SPI 主机和从机设备之间的连接示例，参见图 28- 1。

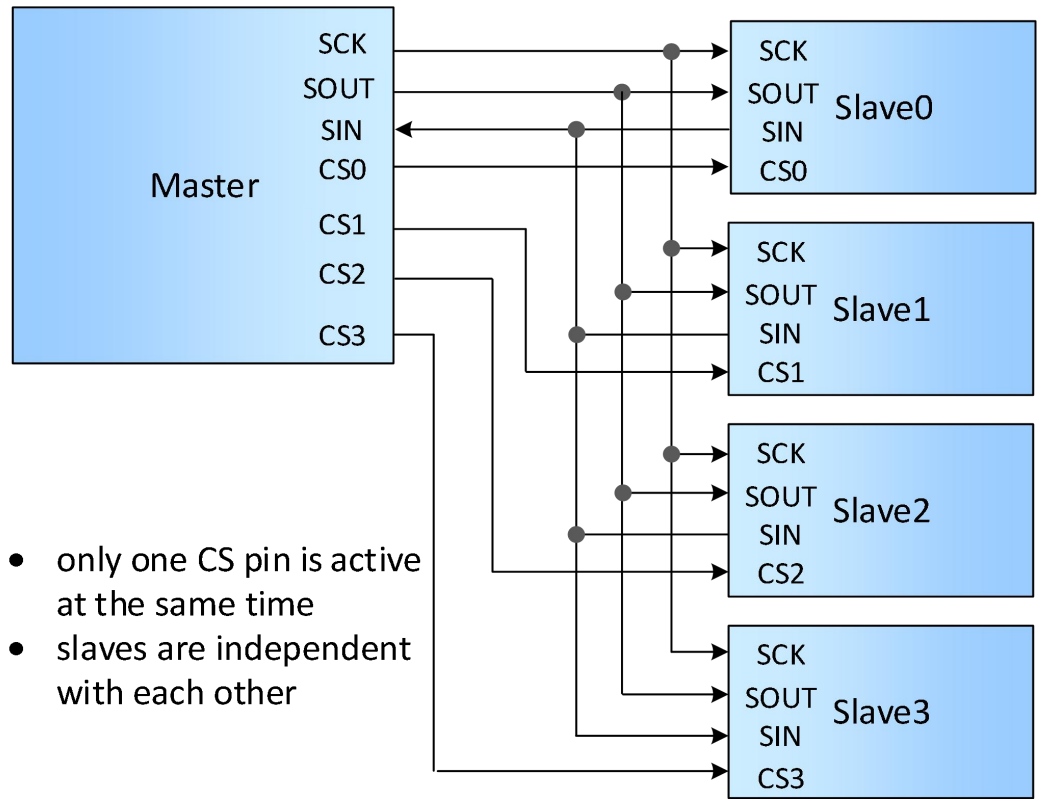


图 28- 1 SPI 系统连接

28.2 特性

- 主机模式或从机模式
- 全双工模式
- 主机可编程传送比特率
- 串行时钟相位和极性选择
- 片选信号极性选择
- 最多可支持 4 路从机设备

© 2013 - 2024 杰发科技有限公司

- 连续或不连续片选 CS 输出
- 可配置 CS 建立时间、保持时间、空闲时间
- 可配置 SCK 高、低周期
- 带 CPU 中断功能的模式错误标志位
- 可供选择的最高有效位（MSB）优先或最低有效位（LSB）优先移位
- 4-32 位传输帧格式选择
- 支持 DMA 模式
- 支持传输 1-bit、2-bit、4-bit 每一个时钟信号
- SPI 主机请求（HREQ）功能
- 支持数据匹配中断
- 带中断功能的 TX 缓冲区下溢及 RX 缓冲区溢出标志位
- 从机支持低功耗模式（STOP1、STOP2、VLPS）唤醒功能

28.3 结构框图

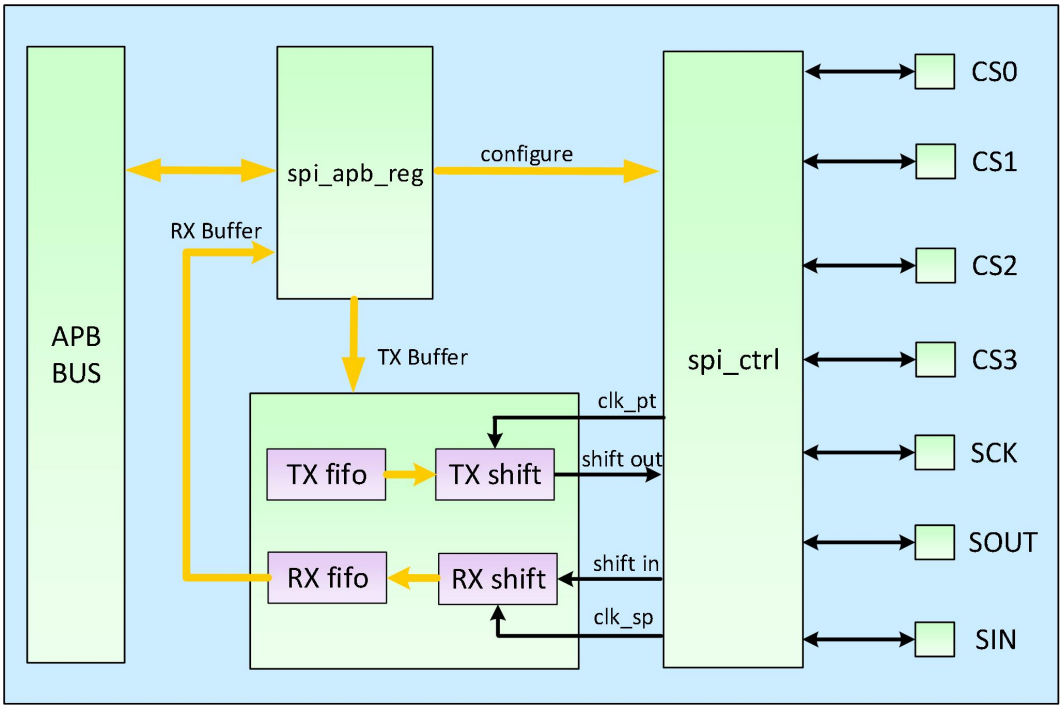


图 28-2 SPI 结构框图

© 2013 - 2024 杰发科技有限公司

28.4 功能描述

28.4.1 数据流 & 算法

28.4.1.1 主机 1-bit 传输

对于主机模式，数据被写入一个 32 位 TX 缓冲区，然后参考波特率控制逻辑，加载到移位寄存器。由 TMSBF 控制输出数据是高位先发还是低位先发。在 FRMSIZE 指定的 SCK 周期数之后，移位寄存器从 SIN 引脚移入数据。接收的数据存储在 32 位 RX 缓冲区中。默认情况下，CS0 为片选信号，可通过 PCS 选择其它 CS 管脚为片选信号，但同一时刻只有一个硬件 CS 管脚有效。同时还可通过 PCSPOL 选择每一个 CS 管脚的极性。

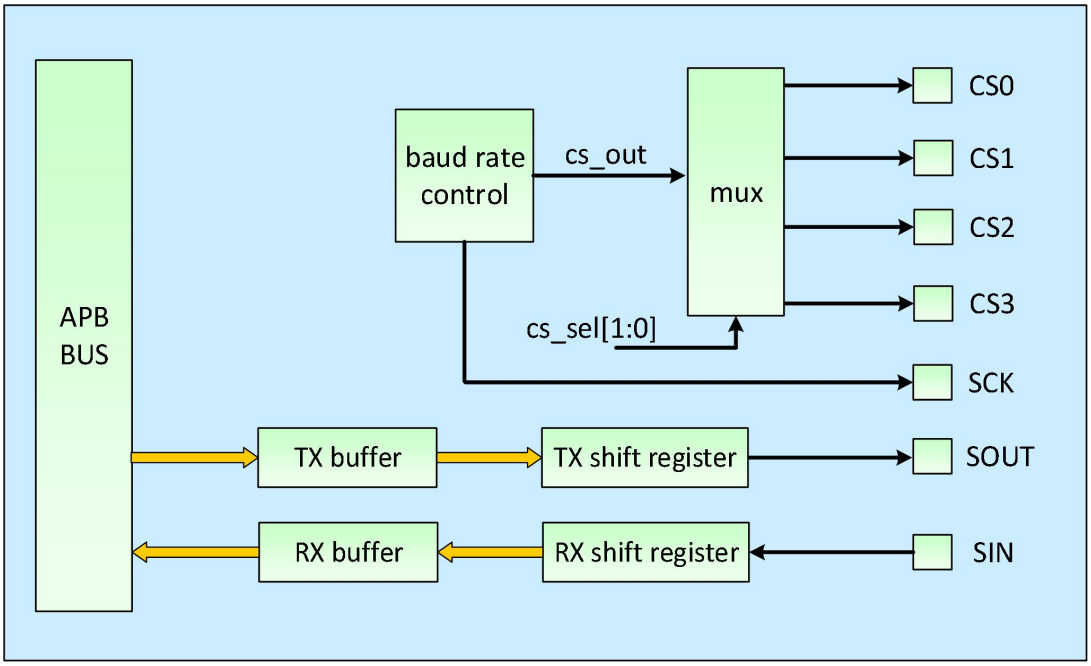


图 28-3 主机 1-bit 模式数据流

28.4.1.2 从机

对于从机模式，数据流类似于主机模式。但 CS 引脚是从机选择输入，SCK 是来自主机的 SPI 时钟输入。在发生数据传输之前，从机 SPI 的 CS 引脚必须为低电平。SOUT 是从机数据输入引脚，SIN 是数据输出引脚。默认情况下，CS0 为片选信号，可通过 PCS 选择其它 CS 管脚为片选信号。

© 2013 - 2024 杰发科技有限公司

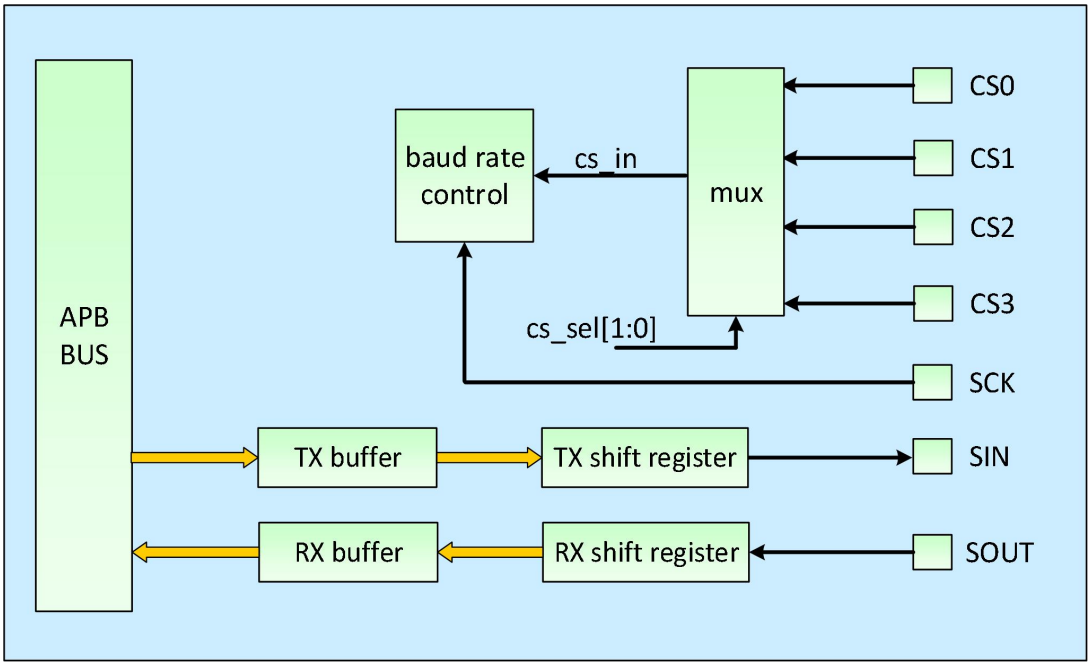


图 28-4 从机数据流

28.4.1.3 主机 2-bit 传输

对于主机模式，可以通过配置 SPI_CFG2[WIDTH]选择传输数据宽度。当配置为‘01b’时，SPI 发送和接收数据将采用双线制模式，即 SPI 在每一个时钟周期发送和接收 2bit 宽度的数据。在传输数据前 TOEN 或 ROEN 必须使能。当使能 TOEN（ROEN 必须置为 0），发送数据流激活，SOUT(DATA0)和 SIN(DATA1)管脚都用于数据输出。当使能 ROEN（TOEN 必须置为 0），接收数据流激活，SOUT(DATA0) 和 SIN(DATA1)管脚都用于数据输入。

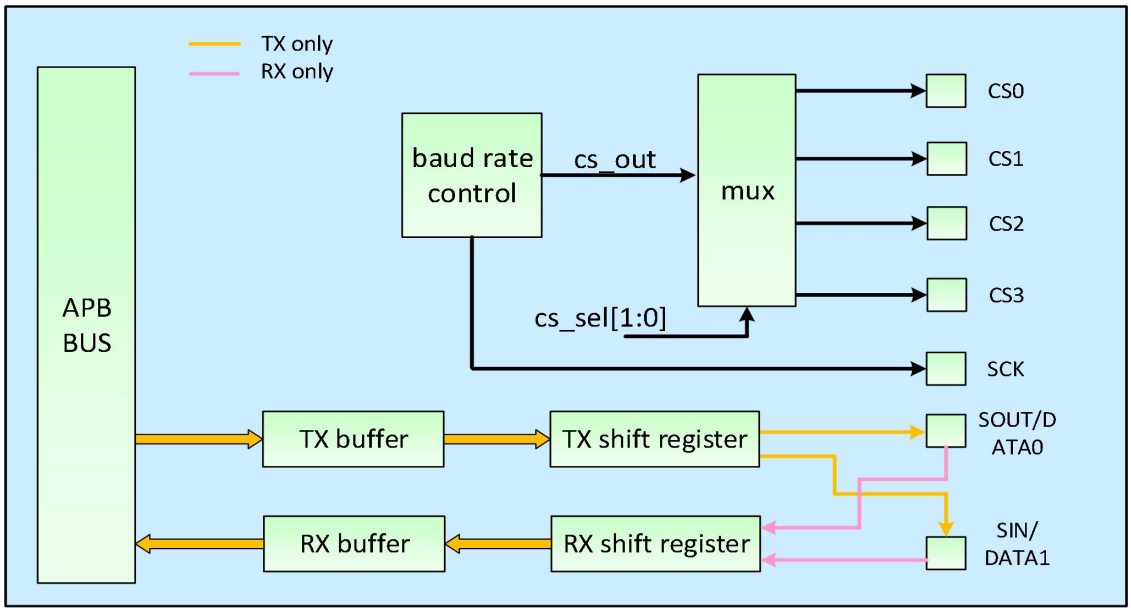


图 28-5 主机 2-bit 模式数据流

© 2013 - 2024 杰发科技有限公司

28.4.1.4 主机 4-bit 传输

对于主机模式，可以通过配置 SPI_CFG2[WIDTH]选择传输数据宽度。当配置为‘10b’时，SPI 发送和接收数据将采用四线制模式，即 SPI 在每一个时钟周期发送和接收 4bit 宽度的数据。在传输数据前 TOEN 或 ROEN 必须使能。当使能 TOEN（ROEN 必须置为 0），发送数据流激活，SOUT(DATA0)、SIN(DATA1)、CS2(DATA2)、CS3(DATA3)管脚都用于数据输出。当使能 ROEN（TOEN 必须置为 0），接收数据流激活，SOUT(DATA0)、SIN(DATA1)、CS2(DATA2)、CS3(DATA3)管脚都用于数据输入。

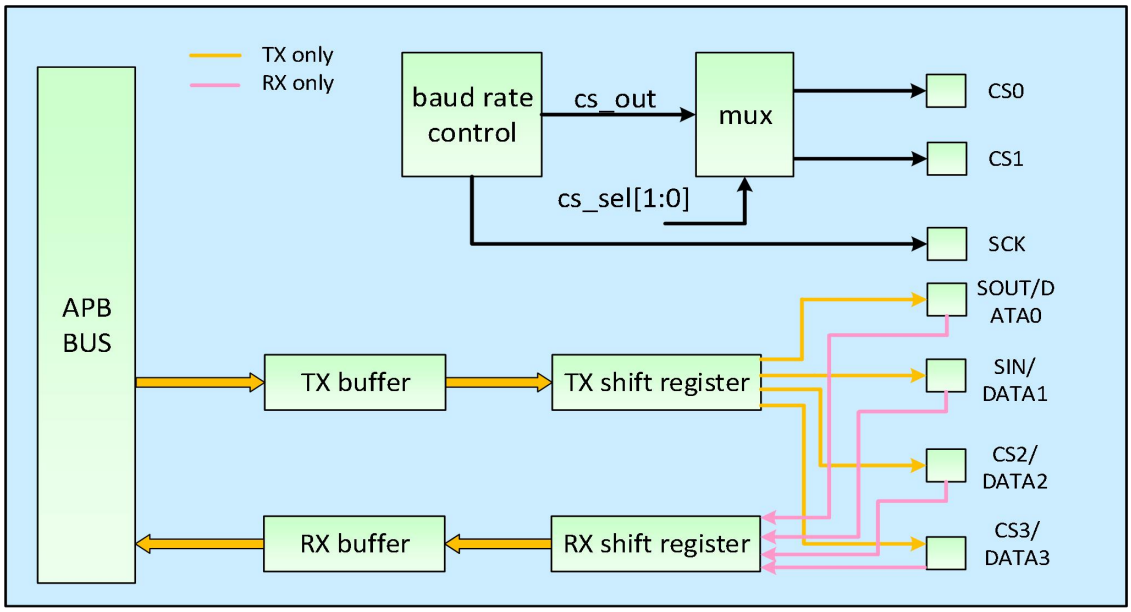


图 28-6 主机 4-bit 模式数据流

28.4.2 输入输出时序

28.4.2.1 CPHA = 0 传输格式

SCK 线上的第一个边沿用于将从机的第一个数据位计时到主机，将主机的第一个数据位计时到从机。在某些外设中，只要选择了从机，从机的数据输出引脚就会提供从机数据的第一位。在这种格式中，在 CS 变低之后，第一个 SCK 边沿发出半个周期。

半个 SCK 周期后，第二个边沿出现在 SCK 线上。当第二个边沿出现时，先前从串行数据输入引脚锁存的值被移入移位寄存器。

在第二个边沿之后，SPI 主机的下一位从主机的串行数据输出引脚传输到从机的串行输入引脚。该过程在 SCK 线上总共持续 16 个边沿，数据被锁存在奇数边沿上并在偶数边缘上移位。

© 2013 - 2024 杰发科技有限公司

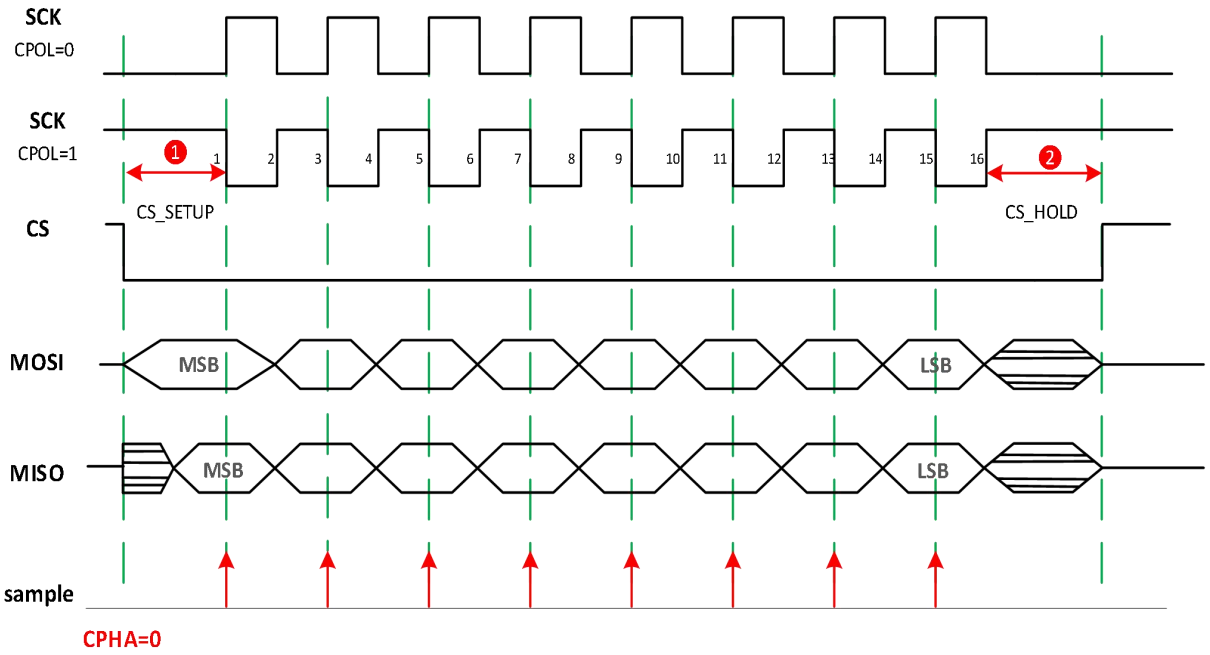


图 28-7 CPHA=0 传输格式

28.4.2.2 CPHA = 1 传输格式

某些外设在第一个数据位在数据输出引脚变得可用之前需要第一个 SCK 边沿，第二个边沿将数据计入系统。

SCK 的第一个边沿在半个 SCK 时钟周期同步延迟之后立即发生。第一个边沿命令从机将其第一个数据位传输到主机的串行数据输入引脚。半个 SCK 周期后，第二个边沿出现在 SCK 引脚上。这是主机和从机的锁存边沿。

当第三个边沿出现时，先前从串行数据输入引脚锁存的值被移入移位寄存器。在此边沿之后，主机数据的下一位从主机的串行数据输出引脚耦合到从机上的串行输入引脚。

该过程在 SCK 线上总共持续 16 个边沿，数据被锁存在偶数边沿上并且移位发生在奇数边沿上。

© 2013 - 2024 杰发科技有限公司

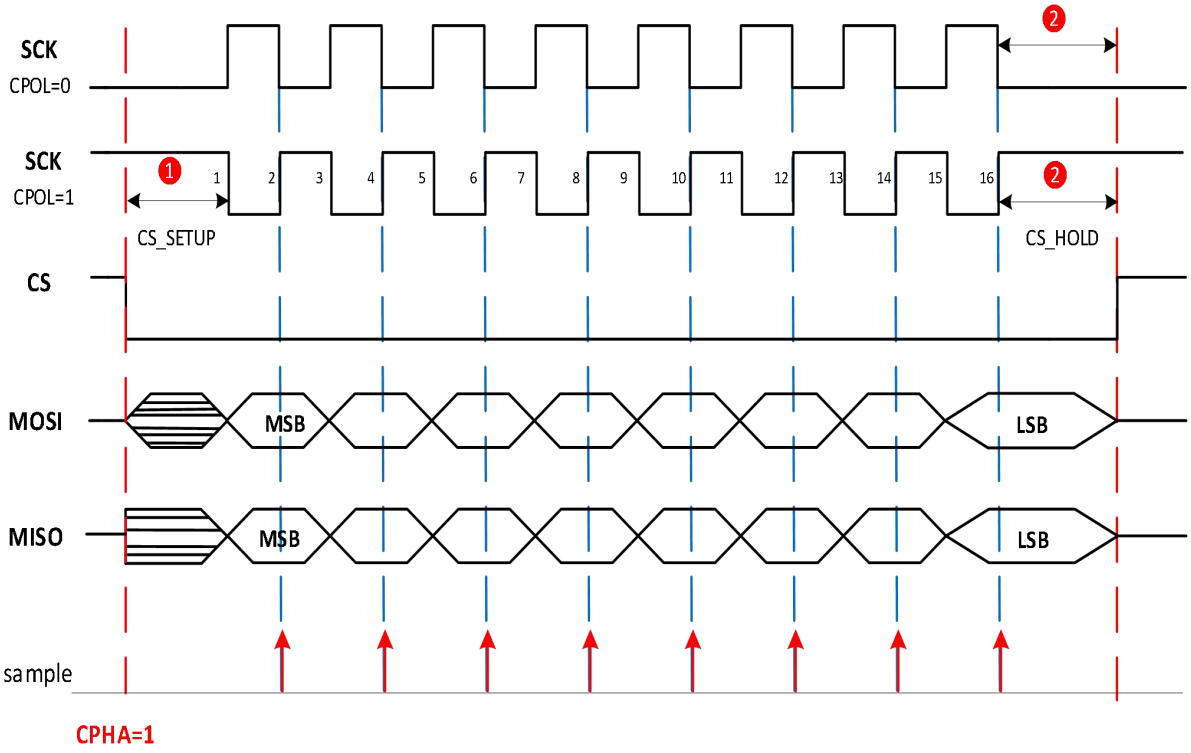


图 28-8 CPHA=1 传输格式

28.4.3 主机 SCK 输出时序设置

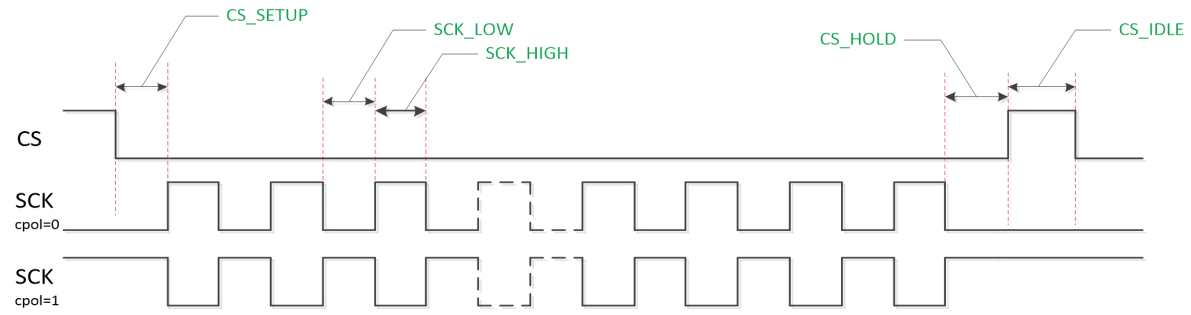


图 28-9 波特率生成

波特率： $f_{SCK}=f_{fclk}/(SCK_LOW+1+SCK_HIGH+1)$, f_{fclk} 是功能时钟的频率。

28.4.4 主机模式故障检测

如果在 SPI 主机初始化传输之前 CS 引脚已被驱动为低电平，则置位 MODEF。主模式故障检测功能仅在 MSTR=1, MODFEN=1, CSOE=1 时有效。

图 28-10 的红色部分为主机检测故障的时间段。

© 2013 - 2024 杰发科技有限公司

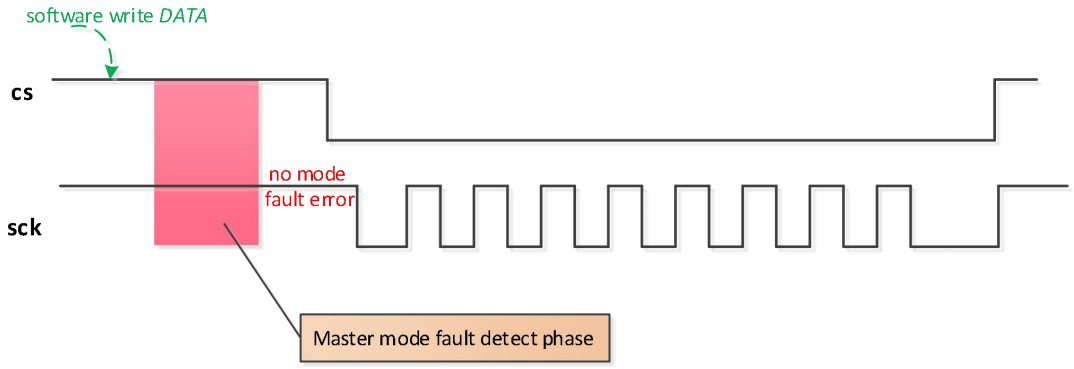


图 28-10 在模式故障检测使能时的 SCK 输出时序

在某些特殊情况下，此模式故障检测功能可能无法检测到模式故障。如果在图 28-11 中的（1）周期内另一个主机驱动 CS 为低电平，则 SPI 主控制器不会设置 MODEF。只有另一个主机在（1）周期之前驱动 CS 为低电平，SPI 主机才可以正常设置 MODEF。

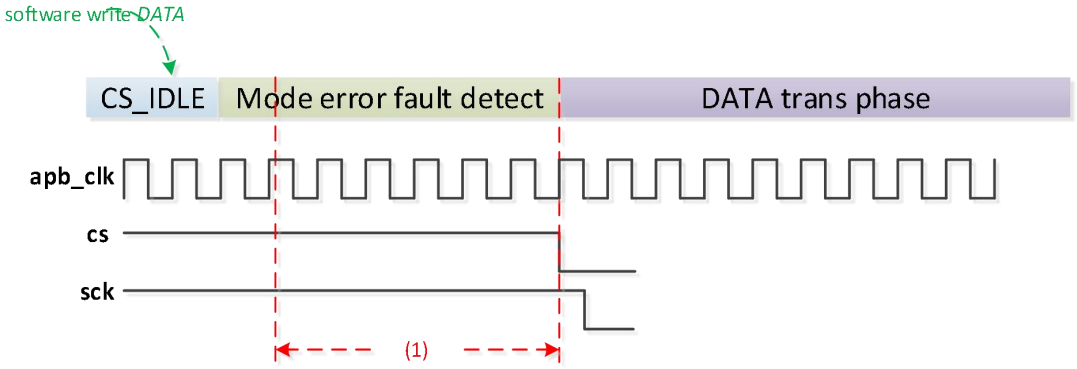


图 28-11 模式故障检测限制

28.4.5 管脚数据方向配置

SPI 1-bit 模式传输数据时，数据管脚 SOUT、SIN 可以根据需要配置 SPI_CFG1[PINCFG]控制位为不同方向。

PINCFG = 0：SOUT 用作 MOSI 功能，SIN 用作 MISO 功能

PINCFG = 1：SOUT 用作 MISO 功能，SIN 用作 MOSI 功能

28.4.6 SPI 接收数据匹配

SPI 接收数据时支持数据匹配功能。如果 DMIE 置 1，发生数据匹配时将会产生数据匹配中断。在接收数据前，用户应该写匹配数据至 DMV 寄存器，并根据需要置 DMIE 为 1。在接收数据的时候，一旦检测到 rxbuf1_ful_set 信号置高，DATA 寄存器会立刻更新数据。在下一个 APB 时钟 clock，如果当前

© 2013 - 2024 杰发科技有限公司

数据匹配，RDMF 会立即拉起，并在 DMIE 有效的时候产生中断。读取缓存数据不会清除 RDMF 位，只能通过用户写 1 的方式清除。

28.4.7 中断请求

SPI 共有 6 个中断信号，如下。

表 28-1 中断汇总

中断源	标志	本地使能	全局使能
发送缓冲区空中断	TXEF	TXEIE	
接收缓冲区非空中断	RXFF	RXFIE	
发送缓冲区下溢中断	TXUF	TXUIE	
接收缓冲区溢出中断	RXOF	RXOIE	
主机模式失效中断	MODEF	MODFIE	
接收数据匹配中断	RDMF	DMIE	

28.4.8 低功耗模式

表 28-2 SPI 唤醒源

模式	唤醒源
STOP1	SPI 接收中断、数据匹配中断
STOP2	SPI 接收中断、数据匹配中断
VLPS	SPI 接收中断、数据匹配中断

当 SPI 处于接收中断或数据匹配时，进入 STOP1/STOP2/VLPS 后的 MCU，接收到数据时会唤醒 MCU 至正常模式。

在 STOP1/STOP2/VLPS 模式的 SPI 从机可以产生中断，以在接收到数据时将 CPU 从低功耗模式唤醒。为确保 SPI 模块低功耗唤醒功能正确，系统必须遵循一些规定。在 CPU 进入低功耗模式之前，系统必须确认 SPI 模块处于空闲状态。软件可以检查状态寄存器 SPI_STATUS[8] IDLEF 位的状态。对于主模式，tx 缓冲区为空，rx 缓冲区为空，内部硬件空闲，IDLEF 为“1”。对于从机模式，tx 缓冲区为空，rx 缓冲区为空，CS 为无效（CS 为高电平），IDLEF 为“1”。如果当 SPI 模块忙时，CPU 进入低功耗模式，则 SPI 模块无法确保数据有效或唤醒功能正确。

从机仅在如下条件全部得到满足时生成异步唤醒中断：

1. SPI 模块处于从机模式；
2. SPI 从机处于空闲状态；
3. 单字节唤醒序列传输结束。

© 2013 - 2024 杰发科技有限公司

CS 从高到低会初始化唤醒阶段，从机在 FRMSIZE 指定的 SCK 周期后生成异步唤醒请求。SPI 从机可以接收唤醒相位字节的数据。芯片唤醒结束（时钟恢复）后，RXFF 标志位会置位，读取数据寄存器将返回主机发送的唤醒相位字节的数据。唤醒阶段只能传输一个字节。在唤醒阶段，来自主机的连续传输会破坏接收到的数据，并可能导致无法正确设置 RXFF 标志。

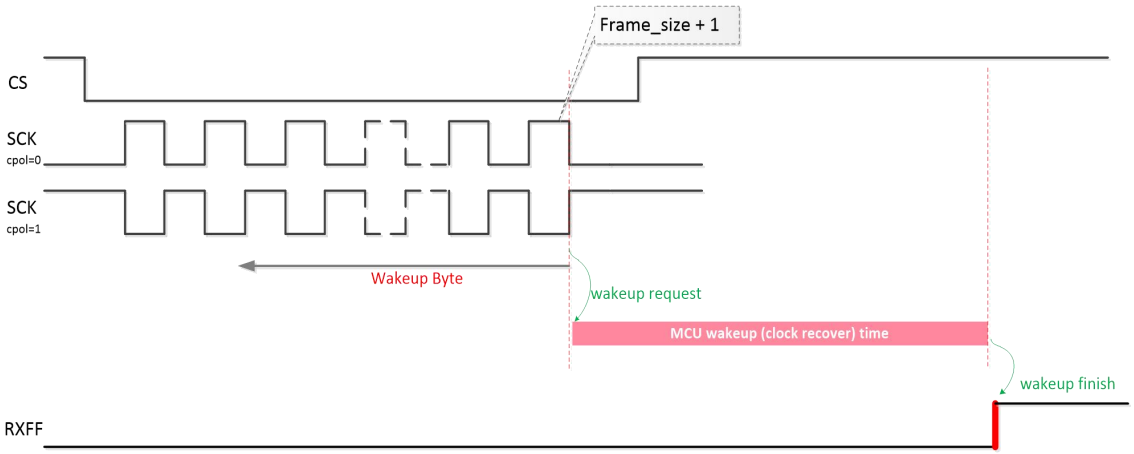


图 28-12 唤醒序列

28.4.9 调试模式

支持 debug 模式控制位设置，debug 模式对 SPI 模块运行无影响.

28.5 应用说明

28.5.1 主机 CS 连续模式

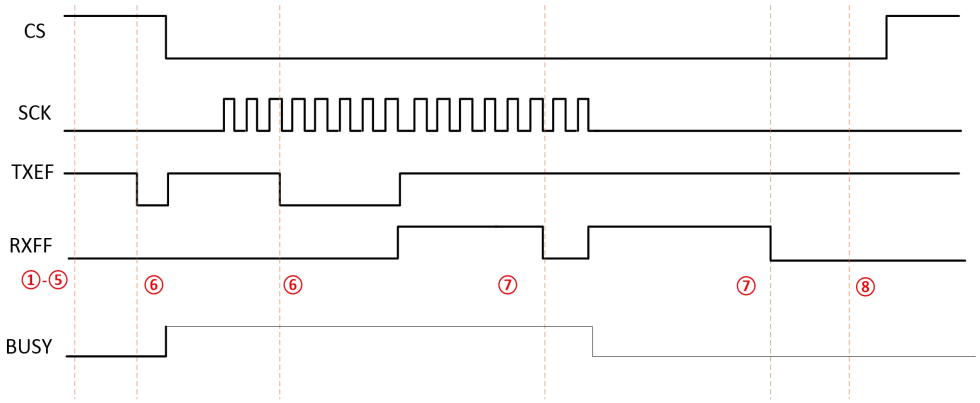


图 28-13 CS 连续模式

CS 连续输出

1. 配置寄存器 SPI_CFG0：CS_SETUP，CS_HOLD，SCK_LOW，SCK_HIGH；
2. 配置寄存器 SPI_CFG1：CS_IDLE；

© 2013 - 2024 杰发科技有限公司

3. 配置 FRMSIZE, CPHA，CPOL，RMSBF，TMSBF 等；
4. 配置 CSOE，CONT_CS，MSTR；
5. SPIEN=1；
6. TXEF=1，写数据至 DATA；
7. RXFF=1，从 DATA 中读数据；
8. 写 CSRLS 为“1”，释放 CS，然后硬件进入空闲状态。即 CSOE = 1, CONT_CS = 1 时， CS 由硬件自动拉低，但是，数据发送完成后，需要用户软件写 CSRLS 为“1”，拉高 CS。

28.5.2 主机 CS 非连续输出模式

1. 配置寄存器 SPI_CFG0：CS_SETUP， CS_HOLD， SCK_LOW， SCK_HIGH；
2. 配置寄存器 SPI_CFG1：CS_IDLE；
3. 配置 FRMSIZE， CPHA， CPOL， RMSBF， TMSBF 等；
4. 配置 CSOE， CONT_CS， MSTR；
5. SPIEN=1；
6. TXEF=1，写数据至 DATA；
7. RXFF=1，从 DATA 读取数据。

当 CS 处于不连续模式时，CS 通过硬件变为低或高电平，所以软件不需要配置 CSRLS。

28.5.3 主机 2/4 bit 传输模式

1. 配置寄存器 SPI_CFG0：CS_SETUP， CS_HOLD， SCK_LOW， SCK_HIGH；
2. 配置寄存器 SPI_CFG1：CS_IDLE；
3. 配置寄存器 SPI_CFG2：TOEN 或 ROEN， WIDTH；
4. 配置 FRMSIZE， CPHA， CPOL， RMSBF， TMSBF， PCSCFG 等；
5. 配置 CSOE， CONT_CS， MSTR；
6. SPIEN=1；
7. TXEF=1， 写数据至 DATA(txonly 模式)；
8. RXFF=1， 从 DATA 中读数据(rxonly 模式)。

大多数时候，SPI 主机工作于正常模式。当传输触发从 bus 时钟至模块功能时钟同步(一个 fclk 时钟后，主机检测新的传输触发信号)，一个错误将会被触发。所以用户必须配置 SDEN and DBT 去延长这个检测时间。第一个发送信号边沿由 CPHA 决定，Tsync 是同步状态的 SYNC 时间。当使能 SDEN，在进入 PKT_CS_WAIT 状态前 FSM 会切换至 SYNC 状态。DBT 位可以调整 SYNC 状态的周期。

© 2013 - 2024 杰发科技有限公司

28.5.4 从机模式

1. 配置 FRMSIZE，CPHA，CPOL.RMSBF，TMSBF 等；
2. 配置 MSTR；
3. SPIEN=1；
4. TXEF=1， 写数据至 DATA；
5. RXFF=1，从 DATA 读取数据。

28.5.5 DMA 模式

当 TXEF=1 时，会产生 DMA TX 请求。

当 RXFF=1 时，会产生 DMA RX 请求。

1. 初始化 DMA；
2. 初始化 SPI 模块，并使能 DMATXEN，DMARXEN；
3. 等待 DMA 完成；
4. 如果 CS 信号为连续、DMA 接收模式，需要在 DMA 接收完成中断中释放 CS。

注意：

1. DMA 接收溢出时：
   如果认为溢出数据有效，不使用读接收缓存接口，重新初始化 DMA 并使能 DMA，DMA 会先搬运溢出数据，再搬运新数据；
   如果认为溢出数据无效，使用读接收缓存接口，重新初始化 DMA 并使能 DMA，DMA 只会搬运新接收数据；

28.6 寄存器定义

表 28-3 SPI 寄存器映射

SPI0 基地址： 0x4000C000
SPI1 基地址： 0x4000C800
SPI2 基地址： 0x4000D000

地址	名称	宽度	描述
SPIx 基地址+0x00	SPI_CFG0	32	SPI 配置寄存器 0
SPIx 基地址+0x04	SPI_CFG1	32	SPI 配置寄存器 1

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
SPIx 基地址+0x08	SPI_CMD	32	SPI 命令寄存器
SPIx 基地址+0x0c	SPI_STATUS	32	SPI 状态寄存器
SPIx 基地址+0x10	SPI_DATA	32	SPI 数据寄存器
SPIx 基地址+0x14	SPI_CFG2	32	SPI 配置寄存器 2
SPIx 基地址+0x20	SPI_DMV	32	SPI 数据匹配寄存器



说明：

上表中，x=0~2。

28.6.1 配置寄存器 0(SPI_CFG0)

表 28-4 SPI_CFG0 寄存器

SPI_CFG0 **配置寄存器 0** **复位值：0x05050505**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CS_SETUP[31:24]								CS_HOLD[23:16]							
访问	RW								RW							
复位值	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SCK_LOW[15:8]								SCK_HIGH[7:0]							
访问	RW								RW							
复位值	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

字段	说明
31: 24 CS_SETUP	**CS 建立时间** 片选建立时间 = (CS_SETUP_COUNT+1)*CLK_PERIOD，其中 CLK_PERIOD 是 SPI 引擎采用的时钟周期时间
23: 16 CS_HOLD	**CS 保持时间** 片选保持时间 = (CS_HOLD_COUNT+1)*CLK_PERIOD.
15:8 SCK_LOW	**SCK 低电平时间** SCK 时钟低电平时间 = (SCK_LOW_COUNT+1)*CLK_PERIOD. **注意：CPOL 为 0 时，该位配置的是 SCK_LOW 的时间。CPOL 为 1 时，配置的是 SCK_HIGH 的时间。**
7：0 SCK_HIGH	**SCK 高电平时间**

© 2013 - 2024 杰发科技有限公司

字段	说明
	SCK 时钟高电平时间 = (SCK_HIGH_COUNT+1)*CLK_PERIOD. **注意：CPOL 为 0 时，该位配置的是 SCK_HIGH 的时间。CPOL 为 1 时，配置的是 SCK_LOW 的时间。**

28.6.2 配置寄存器 1(SPI_CFG1)

表 28- 5 SPI_CFG1 寄存器

SPI_CFG1　　配置寄存器 1　　复位值：0x027C0005

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DMIE			CONT_CS	PINCFG	MODFEN	CSOE	FRMSIZE[24:20]					RMSBF	TMSBF	CPHA	CPOL
访问	RW			RW	RW	RW	RW	RW					RW	RW	RW	RW
复位值	0			0	0	0	1	0	0	1	1	1	1	1	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DMARXEN	DMATXEN	MODFIE	MSTR	RXOIE	TXUIE	RXFIE	TXEIE	CS_IDLE[7:0]							
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

字段	说明
31 DMIE	**数据匹配中断使能** 0： 禁用 1： 使能
28 CONT_CS	**CS 连续输出使能** 0： CS 输出不连续 1： CS 输出连续
27 PINCFG	**PIN 管脚配置** 0： SOUT-MOSI, SIN-MISO 1： SOUT-MISO, SIN-MOSI
26 MODFEN	**主机模式故障检测使能** 0： 禁用主机模式故障检测功能 1： 使能主机模式故障检测功能
25 CSOE	**CS 硬件输出使能** 0： 禁用 CS 硬件输出

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 使能 CS 硬件输出
23：20 FRMSIZE	**帧大小** 00000： 4bit 00001： 4bit 00010： 4bit 00011： 4bit 00100： 5bit 00101： 6bit 00110： 7bit 00111： 8bit ``` 11110：31bit 11111：32bit
19 RMSBF	**RX 最高有效位优先** 0：移位器移位的第一位是输入数据的 LSB 1：移位器移位的第一位是输入数据的 MSB
18 TMSBF	**TX 最高有效位优先** 0： TX LSB 优先(LSB 位首先移出) 1： TX MSB 优先(MSB 位首先移出)
17 CPHA	**时钟相位** 0： 第一个 SCK 过渡边沿为数据移出边沿 1： 第一个 SCK 过渡边沿为数据捕获边沿
16 CPOL	**时钟极性** 0： 空闲时，SCK 为 0 1： 空闲时，SCK 为 1
15 DMARXEN	**DMA RX 请求使能** 0： 禁用 DMA RX 请求 1： 使能 DMA RX 请求
14 DMATXEN	**DMA TX 通道使能** 0：禁用 DMA TX 请求 1：使能 DMA TX 请求
13 MODFIE	**模式故障中断使能** 0： 禁用 1： 使能
12 MSTR	**主机或从机模式选择** 0： 从机模式

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 主机模式
11 RXOIE	**RX 缓冲区溢出中断使能** 0： 禁用 1： 使能
10 TXUIE	**TX 缓冲区下溢中断使能** 0： 禁用 1： 使能
9 RXFIE	**RX 缓冲区满中断使能** 0： 禁用 1： 使能
8 TXEIE	**TX 缓冲区空中断使能** 0： 禁用 1： 使能
7：0 CS_IDLE	**CS 空闲时间** CS 空闲时间 = (CS_IDLEA_COUNT+1)*CLK_PERIOD

28.6.3 命令寄存器 (SPI_CMD)

表 28-6 SPI_CMD 寄存器

SPI_CMD **命令寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										ROTRIG	CSRLS	SWRST				SPIEN
访问										RW	RW	RW				RW
复位值										0	0	0				0

字段	说明
6 ROTRIG	**主机只读模式触发** **注意： 当 CFG2 中的 ROEN=1 时，将此位写入 “1”会触发读序列。读取此位将始终返回 ‘0’ 。**
5 CSRLS	**CS 释放**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 不起作用 1： 释放 CS **注意：软件将此位写入”1”，然后 CS 将变为高电平。读取此位始终返回 “1”，该位对 CS 连续输出有效(CONT_CS=1, CSOE=1)。**
4 SWRST	**软件复位** 0： 不起作用 1： 复位 **注意：本复位只能复位主机引擎/缓冲区/标志位逻辑，从机缓冲区 /标志位逻辑。CFG0/CFG1/CFG2/CMD 控制位不会复位。**
0 SPIEN	**SPI 使能** 0： 禁用 1： 使能

28.6.4 状态寄存器(SPI_STATUS)

表 28- 7 SPI_STATUS 寄存器

SPI_STATUS **状态寄存器** **复位值：0x00000101**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								IDLEF	MEBY		RDMF	MODEF	RXOF	TXUF	RXFF	TXEF
访问								RO	RO		R/W1C	RO	R/W1C	R/W1C	RO	RO
复位值								1	0		0	0	0	0	0	1

字段	说明
8 IDLEF	**SPI 空闲标志** 0： SPI 模块硬件非空闲 1： SPI 模块硬件空闲 **注意：对于主机，TX 缓冲区为空，RX 缓冲区为空，内部硬件空闲，该位可以为 “1”。** **对于从机， TX 缓冲区为空，RX 缓冲区为空， CS 无效(没有没选中) ，该位可以为” 1” 。**
7	**SPI 主机引擎忙标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
MEBY	0： SPI 主机硬件一个字节传输完成 1： SPI 主机硬件一个字节传输未完成
5 RDMF	**接收数据匹配标志** 0： 接收数据不匹配 data match 寄存器 1：接收数据匹配 data match 寄存器 **注意：写 “1” 清零此位。产生下溢时，需要及时清除。**
4 MODEF	**主机模式故障检测标志** 0： 未检测到主机模式故障 1： 检测到主机模式故障 **注意：当 SPI 配置为主机时， 它会在驱动 CS 低电平之前检测 CS 线路状态。如果 CS 已经为低电平，表明另一个主机已经发起一个传输，置起模式故障检测错误标志。仅当 CSOE 置 1、MODEEN 置 1 的时候，CS 管脚用于主机模式故障检测。**
3 RXOF	**RX 缓冲区溢出标志** 0： 没有溢出 1： RX 缓冲区溢出 接收缓冲区有两个 FIFO，如果接收到两个字节后，RX 缓冲区数据都没有被及时读走，下一个接收的数据过来，会产生 RX 溢出 **注意：写 “1” 清零此位。产生溢出时，需要及时清除，以免影响 RXFF 标志和 DMA 接收。如果溢出，则 DATA 寄存器中仅能读出一个字节有效数据。**
2 TXUF	**TX 缓冲区下溢标志** 0： 没有下溢 1： TX 缓冲区下溢 当 SPI 配置为从机时，如果没有往 TX 数据寄存器写数据，master 端就开始通讯，会产生 TX 下溢 **注意：写 “1” 清零此位。产生下溢时，需要及时清除。**
1 RXFF	**RX 缓冲区满标志** 0： 不满 1： 满 **注意：只要 rx 缓冲区中有有效的接收数据 (1 字节或 2 字节)，此位将为”1”。因此该位 “1” 不表示在 rx 缓冲区中接收到 2 个字节的数据。读取 DATA 寄存器将会自动清零该位 。此位称为 RX 缓冲区非空更为合理。**
0	**TX 缓冲区空标志**

© 2013 - 2024 杰发科技有限公司

字段	说明
TXEF	
	0： 非空
	1： 空
	注意：只要 tx 缓冲区不满(2 字节数据)，该位将为"1"，写 DATA 寄存器将自动清零此硬件位 。此位称为 TX 缓冲区非满更为合理。

28.6.5 数据寄存器(SPI_DATA)

表 28-8 SPI_DATA 寄存器

SPI_DATA **数据寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DATA[31:16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA[15:0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0	**SPI 数据端口寄存器**
DATA	
	读： 读操作将返回接收的 DATA 值
	写： 写入要发送的 DATA

28.6.6 配置寄存器 2(SPI_CFG2)

表 28-9 SPI_CFG2 寄存器

SPI_CFG2 **配置寄存器 2** **复位值：0x00000000**

© 2013 - 2024 杰发科技有限公司

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CS_CHECK								CSDETEN[23:20]							DBGEN
访问	RW								RW							RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0				0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			PCSPOL[13:10]				PCSCFG[9:8]		HREN	HRPOL	WIDTH[5:4]		ROEN	TOEN	MNOV	
访问			RW				RW		RW	RW	RW		RW	RW	RW	
复位值			0	0	0	0	0	0	0	0	0	0	0	0	0	

字段	说明
31:24 CS_CHECK	**CS 检测记数周期** 0：1 个功能时钟检测周期 1：2 个功能时钟检测周期 ``` 255:256 个功能时钟检测周期
23 CSDETEN(CS3)	**CS3 检测使能** 0：SPI 关闭 CS3 错误检测 1：SPI 打开 CS3 错误检测
22 CSDETEN(CS2)	**CS2 检测使能** 0：SPI 关闭 CS2 错误检测 1：SPI 打开 CS2 错误检测
21 CSDETEN(CS1)	**CS1 检测使能** 0：SPI 关闭 CS1 错误检测 1：SPI 打开 CS1 错误检测
20 CSDETEN(CS0)	**CS0 检测使能** 0：SPI 关闭 CS0 错误检测 1：SPI 打开 CS0 错误检测
16 DBGEN	**Debug 使能** 0：SPI 在 debug 模式下失效 1：SPI 在 debug 模式下有效
13 PCSPOL(PCS3)	**PCS3 信号极性** 0：PCS3 低电平有效 1：PCS3 高电平有效
12 PCSPOL(PCS2)	**PCS2 信号极性**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：PCS2 低电平有效 1：PCS2 高电平有效
11 PCSPOL(PCS1)	**PCS1 信号极性** 0：PCS1 低电平有效 1：PCS1 高电平有效
10 PCSPOL(PCS0)	**PCS0 信号极性** 0：PCS0 低电平有效 1：PCS0 高电平有效
9:8 PCSCFG	**CS 信号选择配置** 00：PCS0 使能 01：PCS1 使能 02：PCS2 使能 03：PCS3 使能
7 HREN	**HREN 使能** 0：HREQ 功能失效 1：HREQ 功能有效 **注意：HREQ 有效时 SPI CS1 管脚用于数据监控(HREQ)功能，HREQ 无效时 SPI CS1 管脚用于片选功能。**
6 HRPOL	**HREO 极性** 0：HREQ 管脚高时有效 1：HREQ 管脚低时有效
5:4 WIDTH	**传输数据宽度** 00：1bit 传输 01：2bit 传输 10：4bit 传输 **注意： 2/4bit 数据传输仅用于 TX Only 或 RX Only 模式，并仅支持主机模式。**
3 ROEN	**仅 RX 模式使能** 0：禁用 1：使能 **注意： 仅 RX 模式，TXEF 将保持 0。即使 TXEIE=1 也不触发 TXEF IRQ。**
2 TOEN	**仅 TX 模式使能** 0： 禁用 1： 使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	注意： 仅 TX 模式，在一个字节传输完成后，RXFF 不置位。即使 RXFIE=1，也不触发 RXFF IRQ。
1 MNOV	**主机非溢出模式** 0： 禁用 1： 使能 **注意：如果 rxbuff 为满，则写入 txbuff 不会触发新的发送操作。**

28.6.7 数据匹配寄存器(SPI_DMV)

表 28-10 SPI DMV 寄存器

SPI_DMV **数据匹配寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DATA MATCH VALUE[31:16]															
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DATA MATCH VALUE[15:0]															
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 DMV	**数据匹配值** 读： 读会返回数据匹配值 写： 写会配置数据匹配值

© 2013 - 2024 杰发科技有限公司

29 增强型 IO(EIO)

29.1 简介

EIO (Enhanced Input and Output) 是一个高度灵活可配置的模块，集成了 32-bit 串行移位寄存器 (shifter)、 16-bit 定时器 (timer) 并与 I/O 连接，从而可以实现软件灵活配置模拟不同的外设功能。输出复杂的波形，自动产生通信时序，无需 CPU 干预，从而减小 CPU Loading ， EIO 提供资源参见表 29-1。

表 29-1 EIO 提供资源

定时器 (Timers)	移位寄存器 (Shifters)	引脚 (Pins)
4 个	4 个	8 个

29.2 特性

- **EIO 模块能够支持多种协议，包括但不仅限于：**
  - UART
  - I2C
  - SPI
  - I2S
  - PWM
- **EIO 提供以下主要功能：**
  - 具有发送、接收和数据匹配模式的 32 位移位寄存器
  - 高度灵活的 16 位定时器，支持各种内部或外部触发，复位、使能和失能配置
  - 移位寄存器支持字节，位交换，用来模拟串行数据传输的 LSB/MSB
  - 开始/停止位自动生成与检测
  - 中断、DMA 或轮询发送/接收操作
  - 可编程波特率

© 2013 - 2024 杰发科技有限公司

29.3 结构框图

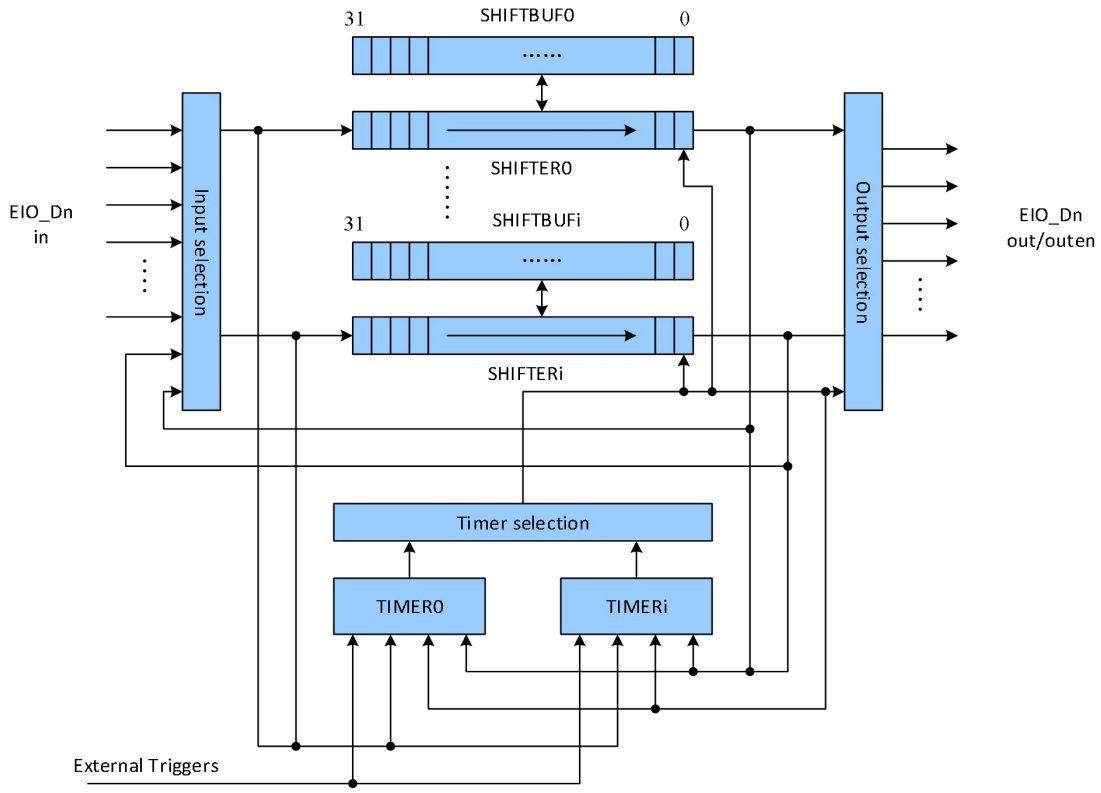


图 29-1 EIO 结构框图

29.4 功能描述

29.4.1 功能时钟

EIO 功能时钟与总线时钟异步，并且可以在低功耗模式下保持启用状态。在访问任何 EIO 寄存器之前，必须启用 EIO 的总线时钟，否则触发 hardfault。

29.4.2 软件复位

EIO 在其控制寄存器中实现了一个软件复位控制位 EIO_CTRL[SWRST]，EIO_CTRL[SWRST]置 1 将会把所有逻辑和寄存器重置为其默认状态，除了 EIO_CTRL 寄存器本身。

© 2013 - 2024 杰发科技有限公司

29.4.3 移位操作

移位寄存器负责将缓冲寄存器 EIO_SHIFTBUFn 里的数据移入或移出 EIO。移位、加载和存储事件的时序由通过 EIO_SHIFTCTLn[TIMSEL] 寄存器分配给移位寄存器的定时器控制。移位寄存器支持 DMA、中断或轮询操作。

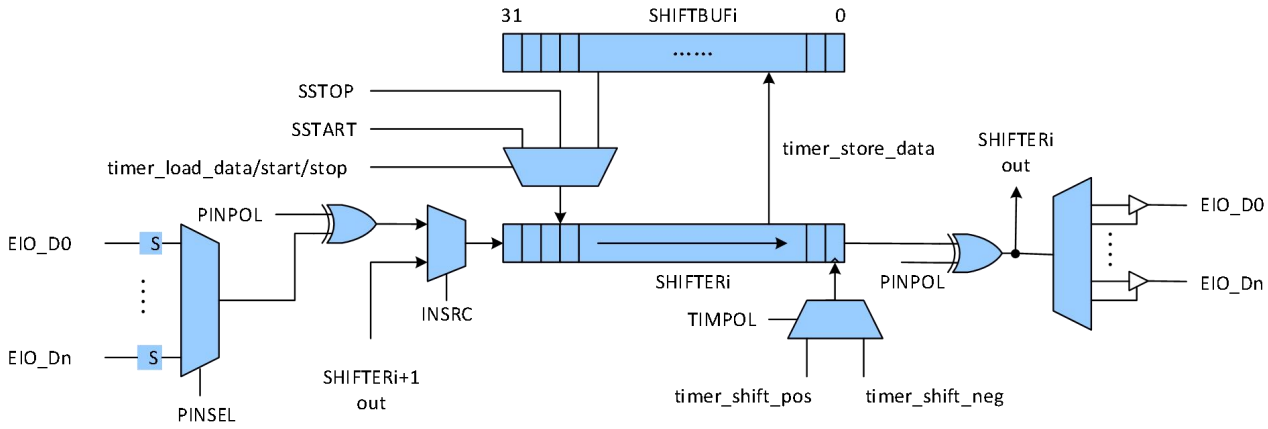


图 29-2 移位器结构框图

29.4.3.1 发送模式

当配置为发送模式(EIO_SHIFTCTLn[SMOD] = 010b)时，移位寄存器将从 缓冲寄存器加载数据，并在分配的定时器发出加载事件信号时将数据移出。通过配置移位寄存器和定时器中的

EIO_SHIFTCFGn[SSTART]、EIO_TIMCFGn[TSTART]或 EIO_SHIFTCFGn[SSTOP]、EIO_TIMCFGn[TSTOP]寄存器，也可以在缓冲寄存器(EIO_SHIFTBUFn) 数据之前/之后自动加载一个可选的开始/停止位。

当数据从缓冲寄存器(EIO_SHIFTBUFn) 加载到移位寄存器中或移位寄存器初始配置为发送模式时，移位器状态标志 EIO_SHIFTSTAT[SSF]都将置位。如果使能了 DMA 和中断，将会发生 DMA 请求和 EIO 中断。当新数据写入缓冲寄存器时，该标志位将清零。

在新数据写入缓冲寄存器(EIO_SHIFTBUFn)之前，移位寄存器就已经开始从缓冲寄存器加载新数据。

这样会由于缓冲寄存器未及时加载新数据(Underrun 下溢)而产生错误标志(EIO_SHIFTERR[SEF])，

如果使能了中断，将会发生 EIO 中断。该标志位可以通过写入逻辑 1 来清除。

29.4.3.2 接收模式

当配置为接收模式(EIO_SHIFTCTLn[SMOD] = 001b)时，当分配的定时器发出存储事件信号时，移位器会将数据移入并将数据存储到缓冲寄存器(EIO_SHIFTBUFn) 中。通过配置移位器和定时器中的

© 2013 - 2024 杰发科技有限公司

EIO_SHIFTCFGn[SSTART]、EIO_TIMCFGn[TSTART]或 EIO_SHIFTCFGn[SSTOP]、EIO_TIMCFGn[TSTOP]，可以在采样移位寄存器数据之前/之后启用对开始/停止位的检查。

当数据从移位器存储到缓冲寄存器(EIO_SHIFTBUFn) 寄存器时，移位器状态标志(EIO_SHIFTSTAT[SSF])会置 1，如果使能了 DMA 和中断，将会发生 DMA 请求和 EIO 中断。当从缓冲寄存器中读取数据时，该标志位将被清除。

如果试图将数据存储到已经满了的缓冲寄存器(EIO_SHIFTBUFn)中(Overrun 上溢)或在开始/停止位检查中发生不匹配时，移位器错误标志 (EIO_SHIFTERR[SEF])将置位 1。如果使能了中断，将会发生 EIO 中断。该标志位可以通过写入逻辑 1 来清除。

29.4.3.3 匹配存储模式

当配置为匹配存储模式(EIO_SHIFTCTLn[SMOD] = 100b)时，当分配的定时器发出存储事件信号时，检查匹配结果，如果匹配正确，则将匹配数据存储到缓冲寄存器(EIO_SHIFTBUFn)中。通过配置移位寄存器和定时器中的 EIO_SHIFTCFGn[SSTART]、EIO_TIMCFGn[TSTART]或 EIO_SHIFTCFGn[SSTOP]、EIO_TIMCFGn[TSTOP]，可以在采样移位寄存器数据之前/之后启用对开始/停止位的检查。使用缓冲寄存器 EIO_SHIFTBUFn [15:0] 配置要匹配的数据和(EIO_SHIFTBUFn)[31:16] 来屏蔽匹配结果，最多可以比较 16 位的数据。

当匹配发生且匹配数据已从移位寄存器存储到缓冲寄存器(EIO_SHIFTBUFn)中时，移位寄存器状态标志 (EIO_SHIFTSTAT[SSF]) 会置 1，如果使能了 DMA 和中断，将会发生 DMA 请求和 EIO 中断。当从 缓冲寄存器中读取数据时，该标志位将被清除。

如果试图将匹配的数据存储到已经满了的缓冲寄存器(EIO_SHIFTBUFn)中(Overrun 上溢)或在开始/停止位检查中发生不匹配时，移位器错误标志 (EIO_SHIFTERR[SEF])将置位 1。如果使能了中断，将会发生 EIO 中断。该标志位可以通过写入逻辑 1 来清除。

29.4.4 定时器操作

16 位的递减定时器控制移位寄存器的加载、移位和存储。 它们可以执行通用定时器功能，例如输出 PWM 波形。 定时器的启动/停止/复位条件配置多样，具体见定时器配置寄存器 EIO_TIMCFGn。 定时器可以选择性地使能开始条件和停止条件。

注意：
应在设置定时器模式 TIMOD 之前配置定时器配置寄存器 (EIO_TIMCFGn)。

© 2013 - 2024 杰发科技有限公司

每个定时器独立运行，尽管定时器可以配置为与前一个定时器同时启用或禁用(例如：定时器 1 可以与定时器 0 同时启用或禁用)。

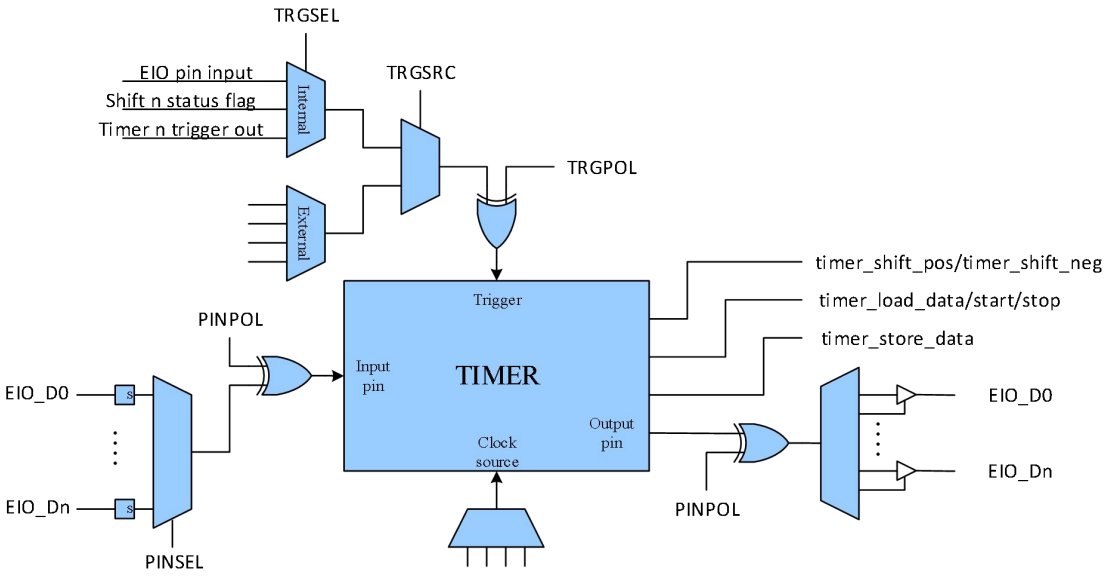


图 29-3 定时器结构框图

29.4.4.1 定时器双 8 位波特率计数模式

在 8 位波特率计数器模式下，16 位计数器分为两个 8 位计数器。低 8 位用于配置移位时钟的波特率，高 8 位用于配置传输中的移位时钟边沿数。当低 8 位递减为零时，定时器输出翻转，低 8 位从比较寄存器重新加载。 当低 8 位递减到 0 时，高 8 位递减。

注意：
8 位波特率计数器模式下的定时器复位事件只会复位低 8 位计数器，高 8 位计数器不受影响并且可以递减。

当高 8 位递减到 0 时，会发生定时器比较事件。 定时器状态标志在定时器比较事件上设置，即定时器状态寄存器(EIO_TIMSTAT)相应位置 1。

29.4.4.2 定时器双 8 位 PWM 模式

在 8 位 PWM 模式下，16 位计数器被分成两个 8 位计数器。 低 8 位用于配置定时器输出高电平周期，高 8 位用于配置定时器输出低电平周期。 当输出为高电平时，低 8 位递减。 当低 8 位递减到 0 时，定

© 2013 - 2024 杰发科技有限公司

时器输出清零，低 8 位从比较寄存器重新加载。 当输出为低时，高 8 位递减。 当高 8 位递减到 0 时，定时器输出置位，高 8 位从比较寄存器重新加载。

当高 8 位递减到 0 时，会发生定时器比较事件。 定时器状态标志在定时器比较事件上设置。即定时器状态寄存器(EIO_TIMSTAT)相应位置 1。

29.4.4.3 定时器单 16 位计数模式

在 16 位计数器模式下，16 位计数器可用于配置移位时钟的波特率(例如：TIMDEC[1:0] != 10 或 11)或传输中的移位时钟边沿数(例如：TIMDEC[1:0] = 10 或 11)。 当 16 位计数器递减到 0 时，定时器输出翻转，计数器从比较寄存器重新加载。

当 16 位计数器递减到 0 时，会发生定时器比较事件。 定时器状态标志在定时器比较事件上设置。即定时器状态寄存器(EIO_TIMSTAT)相应位置 1。

29.4.5 引脚操作

每个定时器和移位器的引脚配置都可以是任何一个 EIO 引脚，EIO 引脚可以配置为输入、输出、开漏输出。通过 EIO_PIN 寄存器可以实时获取 EIO 8 个引脚的电平数据。

29.4.6 支持中断与 DMA

表 29-2 说明了可以生成 EIO 中断和 DMA 请求的状态标志

表 29-2 EIO 标志位说明

标志位	描述	中断	DMA 请求
SSF	移位寄存器状态标志位	Y	Y
SEF	移位寄存器错误状态标志位	Y	N
TSF	定时器状态标志位	Y	N

29.4.7 外设触发

EIO 定时器可以由其他外设通过 CTU 触发启动。

29.4.8 输出触发信号

每个 EIO 定时器都会产生一个与定时器输出相等的输出触发信号。 输出的触发信号不受定时器引脚极性配置的影响。

29.4.9 中断请求

EIO 模块共有 3 种中断类型，共同触发 EIO_IRQHandler 中断

© 2013 - 2024 杰发科技有限公司

表 29-3 EIO 中断汇总

中断源	标志	本地使能	全局使能
移位寄存器状态(接收到数据/发送为空)	SSF	SSIE	无
移位寄存器错误状态(上溢/下溢/起始位错误/停止位错误)	SEF	SEIE	无
定时器状态(产生比较事件)	TSF	TSIE	无

29.4.10 低功耗模式

EIO 模块唤醒源：

表 29-4 EIO 模块唤醒源

模式	唤醒源
STOP1	移位寄存器数据接收中断
	定时器比较事件中断
	起始位停止位错误中断
STOP2	所有中断
VLPS	移位寄存器数据接收中断
	定时器比较事件中断
	起始位停止位错误中断

说明：
系统进入 STOP1/STOP2/VLPS 时，如果 EIO 正处于忙状态(EIO 定时器处于运行状态)，则不会回复休眠应答确认 ACK 信号，造成休眠超时。

29.4.11 调试模式

EIO_CTRL 寄存器中的 DBG_EN=1 时，在 Debug Halt 状态，EIO 继续运行。若 DBG_EN=0，在 Debug Halt 状态，EIO 停止运行。

29.5 应用说明

29.5.1 UART 发送

EIO 模拟 UART 发送需要使用一个定时器、一个移位寄存器和一个引脚(若要支持 CTS，则需要两个引脚)。自动插入起始位和停止位，并且可以使用 DMA 进行多次数据传输。对缓冲寄存器 SHIFTBUFn(或缓冲器 SHIFTBUFBISn 用于先发送 MSB)执行字节写入。发送时钟由 EIO 功能时钟

© 2013 - 2024 杰发科技有限公司

分频，用户可以根据波特率和功能时钟计算最佳分频比。

注意：
EIO 不支持奇偶校验位、中断符、空闲符的自动插入，若要支持，则需要额外的软件干预。

表 29-5 UART 发送配置示例

寄存器	值	说明
SHIFTCFGn	0x0000_0032	配置起始位为 0 和停止位为 1
SHIFTCTLn	0x0003_0002	移位寄存器模式配置为输出模式；选择由定时器 0 控制；使用引脚 0 输出数据；在时钟的上升沿进行移位；可以通过设置 PINPOL 来翻转输出数据；可以通过设置 PINCFG=0x1 来支持引脚开漏
TIMCMPn	0x0000_0F01	配置 为 8 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频/2) - 1
TIMCFGn	0x0000_2222	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在触发为高时启动(可以通过配置 TIMEN=0x3 来支持 CTS)
TIMCTLn	0x01C0_0001	配置为双 8 位计数器波特率模式 ；使用移位寄存器 0 状态标志位，取反后做为内部触发源触发定时器 0 启动；可以通过配置 PINSEL = 0x1(用于 Pin 1)和 PINPOL = 0x1 来支持 CTS
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUF[7:0]来启动 8 位数据传输；使用移位寄存器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUFBBS[7:0] 支持 MSB 优先传输

说明：
上表配置示例中：n=0，Pin0=TX。

EIO_UART 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.2 UART 接收

EIO 模拟 UART 接收需要使用一个定时器、一个移位器和一个引脚 (如果支持 RTS，则为两个定时器和两个引脚)。 自动检测起始位和停止位，并且可以使用 DMA 进行多次数据接收。 UART 接收波特率配置与 UART 发送相同。 EIO 模块不支持小数波特率，因此需要配置 TIMRST 来支持接收同步。 EIO 不支持奇偶校验位的自动验证。

© 2013 - 2024 杰发科技有限公司

表 29-6 UART 接收配置示例

寄存器	值	说明
SHIFTCFGn	0x0000_0032	配置起始位为 0 和停止位为 1
SHIFTCTLn	0x0080_0001	移位寄存器模式配置为输入模式；选择由定时器 0 控制；使用引脚 0 输入数据；在时钟的下降沿进行移位；可以通过设置 PINPOL 来翻转输入数据
TIMCMPn	0x0000_0F01	配置 为 8 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频/2) - 1
TIMCFGn	0x0204_2422	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在定时器引脚输入上升沿时启动；可以通过配置 TIMOUT=0x2 and TIMRST=0x4 来支持接收数据同步
TIMCTLn	0x0000_0081	配置为双 8 位计数器波特率模式 ；引脚 0 取反后输入
SHIFTBUFn	接收的数据	接收到的 8 位数据可以从 SHIFTBUFBYS[7:0] 读取使用；移位器状态标志指示何时可以读取数据；可以通过读取 SHIFTBUFBIS[7:0] 支持 MSB 优先传输

说明：
上表配置示例中：n=0，Pin0=RX。

带有 RTS 配置的 UART 接收需使用第二个定时器来生成 RTS 输出。 RTS 引脚将在检测到起始位时置高，并在从缓冲寄存器 SHIFTBUFn 读取数据后置为低。 当 RTS 被置高时，不会检测起始位，接收的数据将被忽略。

表 29-7 带有 RTS 的 UART 接收配置

寄存器	值	说明
SHIFTCFGn	0x0000_0032	配置起始位为 0 和停止位为 1
SHIFTCTLn	0x0080_0001	移位寄存器模式配置为输入模式；选择由定时器 0 控制；使用引脚 0 输入数据；在时钟的下降沿进行移位；可以通过设置 PINPOL 来翻转输入数据
TIMCMPn	0x0000_0F01	配置 为 8 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频/2) - 1
TIMCFGn	0x0204_2522	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在触发为高和定时器引脚输入上升沿时启动；可以通过配置 TIMOUT=0x2 and TIMRST=0x4 来支持接收数据同步
TIMCTLn	0x02C0_0081	配置为双 8 位计数器波特率模式 ；引脚 0 取反后输入；引脚 1 取反后作为内部触发
TIMCMP(n+1)	0x0000_FFFF	不比较
TIMCFG(n+1)	0x0030_6100	定时器 1 在定时器 0 启动时启动；定时器 1 在下降沿触发时停止；定时器 1 输出逻辑高
TIMCTL(n+1)	0x0143_0003	配置 16 位计数器并在引脚 1 上输出。使用移位器 0 标志位做为内部触

© 2013 - 2024 杰发科技有限公司

寄存器	值	说明
		发
SHIFTBUFn	接收的数据	接收到的 8 位数据可以从 SHIFTBUFBYS[7:0] 读取使用；移位器状态标志指示何时可以读取数据；可以通过读取 SHIFTBUFBIS[7:0] 支持 MSB 优先传输

说明：

上表配置示例中：n=0，Pin0=RX，Pin1=RTS。

EIO_UART 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.3 SPI 主模式

EIO 模拟 SPI Master 需要使用两个定时器、两个移位器和四个引脚来支持 SPI 主模式。可以支持时钟相位 CPHA=0/1 和 时钟极性 CPOL=0/1 ，并且可以使用 DMA 进行多次数据传输。

表 29- 8 SPI 主模式(CPHA=0)配置

寄存器	值	说明
SHIFTCFGn	0x0000_0000	起始位和停止位不使能
SHIFTCTLn	0x0083_0002	移位寄存器模式配置为输出模式；选择由定时器 0 控制；在时钟的下降沿进行移位；使用引脚 0 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0000_0101	移位寄存器模式配置为输入模式；选择由定时器 0 控制；在时钟的上升沿进行移位；使用引脚 1 输入数据；
TIMCMPn	0x0000_3F01	配置 为 32 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频 /2) - 1
TIMCFGn	0x0100_2222	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在触发为高时启动；定时器输出逻辑 0
TIMCTLn	0x01C3_0201	定时器 0 配置为双 8 位计数器波特率模式 ；引脚 2 输出(移位时钟 SCK)；移位寄存器 0 的状态标志位取反后做为内部触发；当时钟极性 CPOL=0 时设置 PINPOL =0 ；当时钟极性 CPOL=1 时设置 PINPOL =1 翻转 SCK 时钟的输出
TIMCMP(n+1)	0x0000_FFFF	不比较
TIMCFG(n+1)	0x0000_1100	定时器 1 在定时器 0 启动时启动；定时器 1 在定时器 0 停止时停止
TIMCTL(n+1)	0x0003_0383	配置 16 位计数器输出；使用引脚 3 取反后输出(作为从选择)
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUF 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUFBBS 支持 MSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUF 读取使用；移位器状态标志指示何时可以读取数据；可以

© 2013 - 2024 杰发科技有限公司

寄存器	值	说明
		通过读取 SHIFTBUFBIS 支持 MSB 优先传输

说明：

上表配置示例中：n=0，Pin0=MOSI，Pin1=MISO，Pin2=SCK，Pin3=SS，从选择为非连续模式。

表 29- 9 SPI 主模式(CPHA=1)配置

寄存器	值	说明
SHIFTCFGn	0x0000_0021	起始位在第一次移位被加载
SHIFTCTLn	0x0003_0002	移位寄存器模式配置为输出模式；选择由定时器 0 控制；在时钟的上升沿进行移位；使用引脚 0 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0080_0101	移位寄存器模式配置为输入模式；选择由定时器 0 控制；在时钟的下降沿进行移位；使用引脚 1 输入数据；
TIMCMPn	0x0000_3F01	配置为 32 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频 /2) - 1
TIMCFGn	0x0100_2222	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在触发为高时启动；定时器输出逻辑 0
TIMCTLn	0x01C3_0201	定时器 0 配置为双 8 位计数器波特率模式；引脚 2 输出(移位时钟 SCK)；移位寄存器 0 的状态标志位取反后做为内部触发；当时钟极性 CPOL=0 时设置 PINPOL =0；当时钟极性 CPOL=1 时设置 PINPOL =1 翻转 SCK 时钟的输出
TIMCMP(n+1)	0x0000_FFFF	不比较
TIMCFG(n+1)	0x0000_1100	定时器 1 在定时器 0 启动时启动；定时器 1 在定时器 0 停止时停止
TIMCTL(n+1)	0x0003_0383	配置 16 位计数器输出；使用引脚 3 取反后输出(作为从选择)
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUF 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUFBBS 支持 MSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUF 读取使用；移位器状态标志指示何时可以读取数据；可以 通过读取 SHIFTBUFBIS 支持 MSB 优先传输

© 2013 - 2024 杰发科技有限公司

说明：

上表配置示例中：n=0，Pin0=MOSI，Pin1=MISO，Pin2=SCK，Pin3=SS，从选择为非连续模式。

EIO_SPI 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.4 SPI 从模式

EIO 模拟 SPI Slave 需要使用一个定时器、两个移位器和四个引脚来支持 SPI 从模式。可以支持时钟相位 CPHA=0/1 和 时钟极性 CPOL=0/1 ，并且可以使用 DMA 进行多次数据传输。

表 29- 10 SPI 从模式(CPHA=0)配置

寄存器	值	说明
SHIFTCFGn	0x0000_0000	起始位和停止位不使能
SHIFTCTLn	0x0083_0102	移位寄存器模式配置为输出模式；选择由定时器 0 控制；在时钟的下降沿进行移位；使用引脚 1 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0000_0001	移位寄存器模式配置为输入模式；选择由定时器 0 控制；在时钟的上升沿进行移位；使用引脚 0 输入数据；
TIMCMPn	0x0000_003F	配置 为 32 位传输，设置 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFGn	0x0120_0600	定时器 0 在上升沿触发时启动；定时器 0 在下降沿触发时停止
TIMCTLn	0x06C0_0203	配置为 16 位计数模式；引脚 2 输入(移位时钟 SCK)取反后做为内部触发；当时钟极性 CPOL=0 时设置 PINPOL =0；当时钟极性 CPOL=1 时设置 PINPOL =1 翻转 SCK 时钟的输入
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUF 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUFBBS 支持 MSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUF 读取使用；移位器状态标志指示何时可以读取数据；可以 通过读取 SHIFTBUFBIS 支持 MSB 优先传输

说明：

上表配置示例中：n=0，Pin0=MOSI，Pin1=MISO，Pin2=SCK，Pin3=SS，从选择为非连续模式。

© 2013 - 2024 杰发科技有限公司

表 29- 11 SPI 从模式(CPHA=1)配置

寄存器	值	说明
SHIFTCFGn	0x0000_0001	移位器配置为在第一次移位时加载并禁用停止位
SHIFTCTLn	0x0003_0102	移位寄存器模式配置为输出模式；选择由定时器 0 控制；在时钟的上升沿进行移位；使用引脚 1 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0080_0001	移位寄存器模式配置为输入模式；选择由定时器 0 控制；在时钟的下降沿进行移位；使用引脚 0 输入数据；
TIMCMPn	0x0000_003F	配置 为 32 位传输，设置 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFGn	0x0120_6602	定时器 0 在上升沿触发时启动；定时器 0 在下降沿触发时停止
TIMCTLn	0x06C0_0203	配置为 16 位计数模式；引脚 2 输入(移位时钟 SCK)取反后做为内部触发；当时钟极性 CPOL=0 时设置 PINPOL =0；当时钟极性 CPOL=1 时设置 PINPOL =1 翻转 SCK 时钟的输入
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUF 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUFBBS 支持 MSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUF 读取使用；移位器状态标志指示何时可以读取数据；可以 通过读取 SHIFTBUFBIS 支持 MSB 优先传输

说明：

上表配置示例中：n=0，Pin0=MOSI，Pin1=MISO，Pin2=SCK，Pin3=SS，从选择为非连续模式。

EIO_SPI 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.5 I2C 主模式

EIO 模拟 I2C Master 需要使用两个定时器、两个移位寄存器和两个引脚来支持 I2C 主模式。 一个定时器用于产生 SCL 输出，一个定时器用于控制移位寄存器。 两个移位寄存器用于每个字的发送和接收，

说明：

EIO 不支持模拟 I2C 的从模式。

接收时发送器必须发送 0xFF。在每个字后插入一个停止位以生成或者验证 ACK/NACK。

© 2013 - 2024 杰发科技有限公司

表 29-12 I2C 主模式配置

寄存器	值	说明
SHIFTCFGn	0x0000_00032	起始位使能(逻辑 0)和停止位使能(逻辑 1)
SHIFTCTLn	0x0101_0082	移位寄存器模式配置为输出模式；选择由定时器 1 控制；在时钟的上升沿进行移位；使用引脚 0 开漏输出数据
SHIFTCFG(n+1)	0x0000_0020	起始位不使能和停止位使能(逻辑 0)用于检测 ACK/NACK
SHIFTCTL(n+1)	0x0180_0001	移位寄存器模式配置为输入模式；选择由定时器 1 控制；在时钟的下降沿进行移位；使用引脚 0 输入数据
TIMCMPn	0x0000_2501	配置为 2 个字节传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (字节数 x 18) + 1。设置 TIMCMP[7:0] = (波特率分频/2) - 1
TIMCFGn	0x0102_2222	配置起始位在定时器启动时使能；配置停止位在定时器比较事件停止时使能；定时器在触发为高时启动；定时器输出逻辑 0
TIMCTLn	0x01C1_0101	配置为双 8 位计数器波特率模式；使用移位寄存器 0 状态标志位，取反后做为内部触发源触发定时器 0 启动；使用引脚 1 开漏输出 SCL 时钟
TIMCMP(n+1)	0x0000_000F	配置为 8 位传输，设置 TIMCMP[15:0] = (位数 x 2) - 1
TIMCFG(n+1)	0x0020_1112	配置起始位在定时器启动时使能；配置停止位在定时器比较事件时使能；定时器 1 在定时器 0 启动时启动；定时器 1 在定时器 0 停止时停止
TIMCTL(n+1)	0x01C0_0183	配置为 16 位计数模式；使用引脚 1 输入 SCL 时钟
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUFBBS[7:0]来启动传输；使用移位器状态标志来指示何时可以写入数据；
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUFBIS[7:0]读取使用；移位器状态标志指示何时可以读取数据；

说明：

上表配置示例中：n=0，Pin0=SCL，Pin1=SDA。

EIO_I2C 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.6 I2S 主模式

EIO 模拟 I2S Master 需要使用两个定时器、两个移位寄存器和四个引脚来支持 I2S 主模式。 一个定时器用于产生位时钟和控制移位寄存器，一个定时器用于产生帧同步。

表 29-13 I2S 主模式配置

寄存器	值	说明
SHIFTCFGn	0x0000_0001	移位器配置为在第一次移位时加载并禁用停止位
SHIFTCTLn	0x0003_0002	移位寄存器模式配置为输出模式；选择由定时器 0 控制；在时钟的上升

© 2013 - 2024 杰发科技有限公司

寄存器	值	说明
		沿进行移位；使用引脚 0 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0080_0101	移位寄存器模式配置为输入模式；选择由定时器 0 控制；在时钟的下降沿进行移位；使用引脚 1 输入数据；
TIMCMPn	0x0000_3F01	配置为 32 位传输，波特率为 EIO 时钟的 4 分频。设置 TIMCMP[15:8] = (位数 x 2) - 1。设置 TIMCMP[7:0] = (波特率分频/2) - 1
TIMCFGn	0x0000_0202	配置起始位在定时器启动时使能；定时器在触发为高时启动；定时器输出逻辑 1
TIMCTLn	0x01C3_0281	定时器 0 配置为双 8 位计数器波特率模式；引脚 2 输出(移位时钟 SCK)；移位寄存器 0 的状态标志位取反后做为内部触发；可以配置 PINPOL 翻转 SCK 时钟的输出
TIMCMP(n+1)	0x0000_007F	配置为 32 位传输，设置 TIMCMP[15:0] = (位数 x 波特率分频) - 1。
TIMCFG(n+1)	0x0000_0100	定时器 1 在定时器 0 启动时启动；定时器 1 不会停止
TIMCTL(n+1)	0x0003_0383	配置 16 位计数器输出；使用引脚 3 取反后输出(作为帧同步 WS)
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUFBIS 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUF 支持 LSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUFBIS 读取使用；移位器状态标志指示何时可以读取数据；可以 通过读取 SHIFTBUF 支持 LSB 优先传输

说明：

上表配置示例中：n=0，Pin0=TX，Pin1=RX，Pin2=SCK，Pin3=WS。

EIO_I2S 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

29.5.7 I2S 从模式

EIO 模拟 I2S Slave 需要使用三个定时器、两个移位寄存器和四个引脚来支持 I2S 从模式。

© 2013 - 2024 杰发科技有限公司

表 29-14 I2S 从模式配置

寄存器	值	说明
SHIFTCFGn	0x0000_0000	起始位和停止位不使能
SHIFTCTLn	0x0103_0002	移位寄存器模式配置为输出模式；选择由定时器 1 控制；在时钟的上升沿进行移位；使用引脚 0 输出数据；
SHIFTCFG(n+1)	0x0000_0000	起始位和停止位不使能
SHIFTCTL(n+1)	0x0180_0101	移位寄存器模式配置为输入模式；选择由定时器 1 控制；在时钟的下降沿进行移位；使用引脚 1 输入数据；
TIMCMPn	0x0000_007F	配置每帧两个 32 位传输。设置 TIMCMP[15:0] = (位数 x 4)
TIMCFGn	0x0020_2500	定时器在触发为高和引脚上升沿触发时启动；定时器输出逻辑 1
TIMCTLn	0x0B40_0283	配置为 16 位计数模式 ；引脚 2 取反后输入(移位时钟 SCK)；定时器 2 输出做为触发源；
TIMCMP(n+1)	0x0000_003F	配置 为 32 位传输，设置 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFG(n+1)	0x0020_2500	定时器在触发为高和引脚上升沿触发时启动；定时器输出逻辑 1
TIMCTL(n+1)	0x0340_0283	配置为 16 位计数模式 ；引脚 2 取反后输入(移位时钟 SCK)；定时器 0 输出做为触发源；
TIMCMP(n+2)	0x0000_0000	与 0 比较(第一个边沿)
TIMCFG(n+2)	0x0020_6400	定时器 2 在引脚上升沿启动；定时器 2 在下降沿触发时停止；定时器输出逻辑 1
TIMCTL(n+2)	0x0440_0383	配置为 16 位计数模式 ；引脚 3 取反后输入(帧同步 WS)；Pin2(移位时钟 SCK)做为触发源；
SHIFTBUFn	发送的数据	发送数据可以写入 SHIFTBUFBIS 来启动传输；使用移位器状态标志来指示何时可以写入数据；可以 通过写入 SHIFTBUF 支持 LSB 优先传输
SHIFTBUF(n+1)	接收的数据	接收到的数据可以从 SHIFTBUFBIS 读取使用；移位器状态标志指示何时可以读取数据；可以 通过读取 SHIFTBUF 支持 LSB 优先传输

说明：

上表配置示例中：n=0，Pin0=TX，Pin1=RX，Pin2=SCK，Pin3=WS。

EIO_I2S 使用 DMA 读写搬运数据的位置需要为：SRAM_U (0x20000000~0x2000EFFF)。

© 2013 - 2024 杰发科技有限公司

29.6 寄存器定义

表 29-15 EIO 寄存器映射

EIO 基地址：0x4000A000

地址	名称	宽度	描述
EIO 基地址+0x000	EIO_CTRL	32	控制寄存器
EIO 基地址+0x004	EIO_PIN	32	引脚寄存器
EIO 基地址+0x008	EIO_SHIFTSTAT	32	移位器状态寄存器
EIO 基地址+0x00C	EIO_SHIFTERR	32	移位器错误状态寄存器
EIO 基地址+0x010	EIO_TIMSTAT	32	定时器状态寄存器
EIO 基地址+0x014	EIO_SHIFTSIEN	32	移位器状态中断使能寄存器
EIO 基地址+0x018	EIO_SHIFTEIEN	32	移位器错误中断使能寄存器
EIO 基地址+0x01C	EIO_TIMIEN	32	定时器中断使能寄存器
EIO 基地址+0x020	EIO_SHIFTSDEN	32	移位器状态 DMA 使能寄存器
EIO 基地址+0x030~0x03C	EIO_SHIFTCTLn	32	移位器控制寄存器 n
EIO 基地址+0x040~0x04C	EIO_SHIFTCFGn	32	移位器配置寄存器 n
EIO 基地址+0x050~0x05C	EIO_SHIFTBUFn	32	移位器缓冲寄存器 n
EIO 基地址+0x060~0x06C	EIO_SHIFTBUFBISn	32	移位器缓冲位交换寄存器 n
EIO 基地址+0x070~0x07C	EIO_SHIFTBUFBYSn	32	移位器缓冲字节交换寄存器 n
EIO 基地址+0x080~0x08C	EIO_SHIFTBUFBBSn	32	移位器缓冲位字节交换寄存器 n
EIO 基地址+0x090~0x09C	EIO_TIMCTLn	32	定时器控制寄存器 n
EIO 基地址+0x0A0~0x0AC	EIO_TIMCFGn	32	定时器配置寄存器 n
EIO 基地址+0x0B0~0x0BC	EIO_TIMCMPn	32	定时器比较值寄存器 n

说明：

表 29-15 中：n=1、2、3

© 2013 - 2024 杰发科技有限公司

29.6.1 控制寄存器(EIO_CTRL)

表 29-16 EIO_CTRL 寄存器

EIO_CTRL **控制寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														DBG_EN	SWRST	EIO_EN
访问														RW	RW	RW
复位值														0	0	0

字段	说明
2 DBG_EN	**Debug 模式使能** 0: 在 Debug 模式下，EIO 不使能 1: 在 Debug 模式下，EIO 使能
1 SWRST	**软件复位** 0: 软件复位不使能 1: 软件复位使能，除控制寄存器(EIO_CTRL)外的所有 EIO 寄存器都将被复位 控制寄存器 EIO_CTRL 不受 SWRST 影响。 SWRST = 1 时，对除 EIO_CTRL 外的所有 EIO 寄存器访问无效
0 EIO_EN	**EIO 全局使能** 0: 不使能 1: 使能

© 2013 - 2024 杰发科技有限公司

29.6.2 引脚寄存器(EIO_PIN)

表 29-17 EIO_PIN 寄存器

EIO_PIN 引脚寄存器 复位值：
0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								PDI								
访问								RO								
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8:0 PDI	**EIO 引脚输入** 0：对应的引脚输入低电平 1：对应的引脚输入高电平 返回每个 EIO 引脚上的输入数据

29.6.3 移位状态寄存器(EIO_SHIFTSTAT)

表 29-18 EIO_SHIFTSTAT 寄存器

EIO_SHIFTSTAT 移位器状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SSF			
访问													R/W1C			
复位值													0	0	0	0

字段	说明
3:0 SSF	**移位寄存器状态标志** 0：标志位清零 1：标志位置位

© 2013 - 2024 杰发科技有限公司

字段	说明
	当发生以下事件之一时，会更新移位寄存器状态标志位： **当为接收模式时**(SMOD = 001b)，当缓冲器(SHIFTBUF) 已经加载了移位寄存器的数据时(缓冲器(SHIFTBUF)非空)，该位置 1。读取缓冲器(SHIFTBUF)寄存器时硬件自动清除该状态标志位。 **当为发送模式时**(SMOD = 010b)， 当缓冲器(SHIFTBUF) 的数据已传输到了移位寄存器时(缓冲器(SHIFTBUF)为空)，该位置 1。写入缓冲器(SHIFTBUF)寄存器时硬件自动清除该状态标志位。 **当为匹配接收模式时**(SMOD = 100b)，当缓冲器(SHIFTBUF)和移位寄存器之间发生数据匹配，并将移位寄存器的数据加载到缓冲器(SHIFTBUF)时(缓冲器(SHIFTBUF)非空)，该位置 1。读取缓冲器(SHIFTBUF)寄存器时硬件自动清除该状态标志位。 SSF bit3~bit0 分别对应 EIO 的四个移位寄存器 shifter3~shifter0 写“1”清除该标志位

29.6.4 移位错误状态寄存器(EIO_SHIFTERR)

表 29- 19 EIO_SHIFTERR 寄存器

EIO_SHIFTERR **移位器错误状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SEF			
访问													R/W1C			
复位值													0	0	0	0

字段	说明
3:0 SEF	**移位寄存器错误状态标志** 0：错误状态标志位清零 1：错误状态标志位置位 当发生以下事件之一时，会更新移位寄存器错误状态标志位： **当为接收模式时**(SMOD = 001b)，如果在移位寄存器的新数据覆盖先前内容之前，未读取缓冲器(SHIFTBUF)(缓冲器(SHIFTBUF)上溢)，则该位置 1。或者当接收到的开始或停止位与预期值不匹配时，该位也会置 1。

© 2013 - 2024 杰发科技有限公司

字段	说明
	当为发送模式时(SMOD = 010b)， 如果当移位器已准备好从缓冲器(SHIFTBUF)加载新数据到移位寄存器时，但未往缓冲器(SHIFTBUF)里面写入新数据(缓冲器(SHIFTBUF)下溢)，则该位置 1。 **当为匹配接收模式时**(SMOD = 100b)，如果上次匹配的数据未读取，又发生新的数据匹配(缓冲器(SHIFTBUF)上溢)，则该位置 1。读取缓冲器(SHIFTBUF)寄存器时硬件自动清除该状态标志位。 SEF bit3~bit0 分别对应 EIO 的四个移位器 shifter3~shifter0 写“1”清除该标志位

29.6.5 定时器状态寄存器(EIO_TIMSTAT)

表 29- 20 EIO_TIMSTAT 寄存器

EIO_TIMSTAT **定时器状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													TSF			
访问													R/W1C			
复位值													0	0	0	0

字段	说明
3:0 TSF	**定时器状态标志** 0：状态标志位清零 1：状态标志位置位 定时器的状态标志位是根据定时器的模式来置位： **当为 8 位波特率计数器模式时**(TMOD = 01b)，当高 8 位计数器为零并递减时，则该位置 1。 **当为 8 位高电平 PWM 模式时**(TMOD = 10b)，当高 8 位计数器为零并递减时，则该位置 1。 **当为 16 位计数器模式时**(TMOD = 11b)，当 16 位计数器为零并递减时，则该位置 1。 TSF bit3~bit0 分别对应 EIO 的四个定时器 timer3~timer0 写“1”清除该标志位

© 2013 - 2024 杰发科技有限公司

29.6.6 移位状态中断使能寄存器(EIO_SHIFTSIEN)

表 29-21 EIO_SHIFTSIEN 寄存器

EIO_SHIFTSIEN 移位状态中断使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SSIE			
访问													RW			
复位值													0	0	0	0

字段	说明
3:0 SSIE	**移位状态标志中断使能** 0：移位状态标志中断不使能 1：移位状态标志中断使能 如果标志位中断使能，当 SSF 标志位置 1 时，生成 EIO 中断

29.6.7 移位错误状态中断使能寄存器(EIO_SHIFTEIEN)

表 29-22 EIO_SHIFTEIEN 寄存器

EIO_SHIFTEIEN 移位错误状态中断使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SEIE			
访问													RW			
复位值													0	0	0	0

字段	说明
3:0 SEIE	**移位错误状态标志中断使能** 0：移位错误状态标志中断不使能 1：移位错误状态标志中断使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	如果标志位中断使能，当 SEF 标志位置 1 时，生成 EIO 中断

29.6.8 定时器中断使能寄存器(EIO_TIMIEN)

表 29-23 EIO_TIMIEN 寄存器

EIO_TIMIEN 定时器中断使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													TSIE			
访问													RW			
复位值													0	0	0	0

字段	说明
3:0 TSIE	**定时器状态标志中断使能** 0：定时器状态标志中断不使能 1：定时器状态标志中断使能 如果标志位中断使能，当 TSF 标志位置 1 时，生成 EIO 中断

29.6.9 移位状态 DMA 使能寄存器(EIO_SHIFTSDEN)

表 29-24 EIO_SHIFTSDEN 寄存器

EIO_SHIFTSDEN 移位状态 DMA 使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													SHIFTSDEN			
访问													RW			
复位值													0	0	0	0

字段	说明
3:0 SHIFTSDEN	**移位状态 DMA 使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：移位状态 DMA 不使能 1：移位状态 DMA 使能 如果移位状态 DMA 使能，当 SSF 标志位置 1 时，生成 DMA 请求

29.6.10 移位控制寄存器 n(EIO_SHIFTCTL0 – EIO_SHIFTCTL3)

表 29- 25 EIO_SHIFTCTLn 寄存器

EIO_SHIFTCTLn **移位控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称							TIMSEL		TIMPOL						PINCFG	
访问							RW		RW						RW	
复位值							0	0	0						0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						PINSEL			PINPOL					SMOD		
访问						RW			RW					RW		
复位值						0	0	0	0					0	0	0

字段	说明
25:24 TIMSEL	**定时器选择** 0-3：timer0-timer3 选择用于控制移位寄存器和生成移位寄存器时钟的定时器
23 TIMPOL	**定时器极性** 0：在移位寄存器时钟的上升沿移位 1：在移位寄存器时钟的下降沿移位
17:16 PINCFG	**移位寄存器引脚配置** 00b：移位寄存器引脚输出不使能，即为输入 01b：移位寄存器引脚开漏输出 10b：保留 11b：移位寄存器引脚输出
10:8 PINSEL	**移位寄存器引脚选择** 0-7：EIO_Pin0 - EIO_Pin7 选择移位寄存器控制的引脚

© 2013 - 2024 杰发科技有限公司

字段	说明
7 PINPOL	**移位寄存器引脚极性** 0：高电平有效 1：低电平有效
2:0 SMOD	**移位寄存器模式** 000b：不使能 001b：接收模式，在定时器计时到时将移位寄存器的内容加载到缓冲器(SHIFTBUF)中。 010b：发送模式，在定时器计时到时将缓冲器(SHIFTBUF)的内容传输到移位寄存器中。 100b：匹配接收模式，在定时器计时到时将移位寄存器的数据与缓冲器(SHIFTBUF)进行匹配，匹配成功时，将数据加载到缓冲器(SHIFTBUF)里面。

29.6.11 移位配置寄存器 n(EIO_SHIFTCFG0 – EIO_SHIFTCFG3)

表 29-26 EIO_SHIFTCFGn 寄存器

EIO_SHIFTCFGn　　移位配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								INSRC			SSTOP				SSTART	
访问								RW			RW				RW	
复位值								0			0	0			0	0

字段	说明
8 INSRC	**移位寄存器输入源选择** 0：输入源选择引脚 1：输入源选择前一个移位寄存器(N+1)的输出 最后一个移位器不支持配置 INSRC=1。
5:4 SSTOP	**移位寄存器停止位** 00b：禁用停止位 01b：保留 10b：发送器在存储时输出停止位值 0，如果停止位不为 0，接收器/匹配存储设置错误标志 11b：发送器在存储时输出停止位值 1，如果停止位不是 1，接收器/匹配存储设置错误标志 **当为发送模式时**(SMOD = 010b)，如果所选的定时器启用了停止位，则允许自动插入停

© 2013 - 2024 杰发科技有限公司

字段	说明
	止位。 **当为接收模式 / 匹配接收模式时**(SMOD = 001b / SMOD = 100b)，如果所选的定时器启用了停止位，则允许自动检测停止位。
1:0 SSTART	**移位寄存器起始位** 00b：禁用起始位，发送寄存器在定时器使能时加载数据 01b：禁用起始位，发送寄存器在第一次移位时加载数据 10b：在第一次移位加载数据之前，发送器输出起始位0，如果起始位不为0，则接收器/匹配存储设置错误标志 11b：在第一次移位加载数据之前，发送器输出起始位1，如果起始位不为1，则接收器/匹配存储设置错误标志 **当为发送模式时**(SMOD = 010b)，如果所选的定时器启用了起始位，则允许自动插入起始位。 **当为接收模式 / 匹配接收模式时**(SMOD = 001b / SMOD = 100b)，如果所选的定时器启用了起始位，则允许自动检测起始位。

29.6.12 移位缓冲寄存器 n(EIO_SHIFTBUF0 – EIO_SHIFTBUF3)

表 29- 27 EIO_SHIFTBUFn 寄存器

EIO_SHIFTBUFn **移位缓冲寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SHIFTBUF															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SHIFTBUF															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 SHIFTBUF	**移位缓冲寄存器** **当为接收模式时**(SMOD = 001b)，在定时器计时到时，移位寄存器的数据被加载到缓冲器(SHIFTBUF)中，并置位相应的移位状态寄存器 SHIFSTAT。 **当为发送模式时**(SMOD = 010b)， 在定时器开始计时之前，缓冲器(SHIFTBUF)的数据被传输到移位寄存器中。 **当为匹配接收模式时**(SMOD = 100b)， 缓冲器 SHIFTBUF[15:0] 包含要匹配的数据，

© 2013 - 2024 杰发科技有限公司

字段	说明
	SHIFTBUF[31:16] 为 Mask 位，用于屏蔽匹配位(1 无需匹配，0 需匹配）。当检测到数据匹配时，移位器寄存器数据 [15:0] 被写入 SHIFTBUF[15:0]，并且相应的状态寄存器置位。 **说明：接收模式是低位优先，也就是先进来的数据放在移位寄存器低位。匹配模式是高位优先，也就是先进来的数据放在移位寄存器高位。**

29.6.13 移位缓冲位交换寄存器 n(EIO_SHIFTBUFBIS0 – EIO_SHIFTBUFBIS3)

表 29-28 EIO_SHIFTBUFBISn 寄存器

EIO_SHIFTBUFBISn　　移位缓冲位交换寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SHIFTBUFBIS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SHIFTBUFBIS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 SHIFTBUFBIS	**移位缓冲位交换寄存器** 访问 SHIFTBUFBIS 本质上就是访问 SHIFTBUF，但对 SHIFTBUFBIS 的访问是位交换的 例：读取 SHIFTBUFBIS[31:0]，返回的是 SHIFTBUF[0:31]的结果

© 2013 - 2024 杰发科技有限公司

29.6.14 移位缓冲字节交换寄存器 n(EIO_SHIFTBUFBYS0 – EIO_SHIFTBUFBYS3)

表 29- 29 EIO_SHIFTBUFBYSn 寄存器

EIO_SHIFTBUFBYSn 移位缓冲字节交换寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SHIFTBUFBYS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SHIFTBUFBYS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 SHIFTBUFBYS	**移位缓冲字节交换寄存器** 访问 SHIFTBUFBYS 本质上就是访问 SHIFTBUF，但对 SHIFTBUFBYS 的访问是字节交换的 例：读取 SHIFTBUFBYS[31:0]，返回的是 SHIFTBUF[7:0]、SHIFTBUF[15:8]、SHIFTBUF[23:16]、SHIFTBUF[31:24]的结果

29.6.15 移位缓冲位字节交换寄存器 n(EIO_SHIFTBUFBBS0 – EIO_SHIFTBUFBBS3)

表 29- 30 EIO_SHIFTBUFBBSn 寄存器

EIO_SHIFTBUFBBSn 移位缓冲位字节交换寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SHIFTBUFBBS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SHIFTBUFBBS															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 SHIFTBUFBBS	**移位缓冲位字节交换寄存器**

© 2013 - 2024 杰发科技有限公司

字段	说明
	访问 SHIFTBUFBYS 本质上就是访问 SHIFTBUF，但对 SHIFTBUFBYS 的访问是位字节交换的 例：读取 SHIFTBUFBBS[31:0]，返回的是 SHIFTBUF[24:31], SHIFTBUF[16:23], SHIFTBUF[8:15], SHIFTBUF[0:7]的结果

29.6.16 定时控制寄存器 n(EIO_TIMCTL0 – EIO_TIMCTL3)

表 29- 31 EIO_TIMCTLn 寄存器

EIO_TIMCTLn **定时控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					TRGSEL				TRGPOL	TRGSRC					PINCFG	
访问					RW				RW	RW					RW	
复位值					0	0	0	0	0	0					0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						PINSEL			PINPOL						TMOD	
访问						RW			RW						RW	
复位值						0	0	0	0						0	0

字段	说明
27:24 TRGSEL	触发选择 对于引脚触发，N=0 到 7 对于移位寄存器触发，N=0 到 3 对于定时器触发，N=0 到 3 **说明：CTU 触发为外部触发，其它的触发都为内部触发** **当配置为外部触发时(TRGSRC = 0)** • N：External trigger N input **当配置为内部触发时(TRGSRC = 1)** • 2*N：引脚 N 的触发 • 4*N+1 ：移位寄存器 N 的状态标志位 • 4*N+3 ：定时器 N 的触发 换种方式描述，内部触发选择为： • 0000 = 引脚 0 • 0001 = 移位寄存器 0 标志位 • 0010 = 引脚 1 • 0011 = 定时器 0 触发 • 0100 = 引脚 2 • 0101 = 移位寄存器 1 标志位

© 2013 - 2024 杰发科技有限公司

字段	说明
	• 0110 = 引脚 3 • 0111 = 定时器 1 触发 • ... • 直到引脚 7，移位寄存器 3，定时器 3
23 TRGPOL	**触发极性** 0：高电平有效 1：低电平有效
22 TRGSRC	**触发源选择** 0：外部触发 1：内部触发
17:16 PINCFG	**定时器引脚配置** 00b：定时器引脚输出不使能，即输入 01b：定时器引脚开漏输出 10b：保留 11b：定时器引脚输出
10:8 PINSEL	**定时器引脚选择** 0-7：EIO_Pin0 - EIO_Pin7 选择定时器控制的引脚
7 PINPOL	**定时器引脚极性** 0：高电平有效 1：低电平有效
1:0 TMOD	**定时器模式** 00b：定时器禁用 01b ：双 8 位计数器波特率模式 10b ：双 8 位计数器 PWM 模式 11b ：单 16 位计数器模式 1. 在 8 位波特计数器模式下，计数器和比较寄存器的低 8 位用于配置定时器移位时钟的波特率，高 8 位用于配置移位寄存器的位数。 2. 在8位PWM高电平模式下，计数器和比较寄存器的低8位用于配置定时器移位时钟的高电平周期，高8位用于配置定时器移位时钟的低电平周期。 3. 在 16 位计数器模式下，计数器和比较寄存器的全部 16 位用于配置移位时钟的波特率或移位寄存器位计数。

© 2013 - 2024 杰发科技有限公司

29.6.17 定时器配置寄存器 n(EIO_TIMCFG0 – EIO_TIMCFG3)

表 29-32 EIO_TIMCFGn 寄存器

EIO_TIMCFGn 定时器配置寄存器 复位值：**0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称							TIMOUT				TIMDEC			TIMRST		
访问							RW				RW			RW		
复位值							0	0			0	0		0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		TIMDIS				TIMENA					TSTOP				TS TA RT	
访问		RW				RW					RW				RW	
复位值		0	0	0		0	0	0			0	0			0	

字段	说明
25:24 TIMOUT	**定时器输出** 00b：定时器输出在启用时为逻辑 1，不受定时器复位影响 01b：定时器输出在启用时为逻辑 0，不受定时器复位影响 10b：定时器输出在启用和定时器复位时为逻辑 1 11b：定时器输出在启用和定时器复位时为逻辑 0
21:20 TIMDEC	**定时器递减** 00b：定时器在EIO时钟上递减计数，移位时钟等于定时器输出 01b：定时器在触发输入(双边沿)上递减计数，移位时钟等于定时器输出 10b：定时器在引脚输入(双边沿)上递减计数，移位时钟等于引脚输入 11b：定时器在触发输入(双边沿)上递减计数，移位时钟等于触发输入
18:16 TIMRST	**定时器复位** 000b：定时器永不复位 001b：保留 010b：当定时器引脚等于定时器输出时，定时器复位 011b：当定时器触发等于定时器输出时，定时器复位 100b：当定时器引脚输入上升沿时，定时器复位 101b：保留 110b：当定时器被上升沿触发时，定时器复位 **说明：在 8 位计数器模式下，定时器复位只会复位配置波特率的低 8 位。在所有其他模式下，定时器复位将重置计数器的全部 16 位**
14:12 TIMDIS	**定时器引脚配置** 000b：定时器从不停止

© 2013 - 2024 杰发科技有限公司

字段	说明
	001b：在定时器 N-1 停止时停止定时器 010b：定时器产生比较事件时停止定时器 011b：定时器产生比较事件和触发为低时停止定时器 100b：在定时器引脚输入上升沿或者下降沿时停止定时器 101b：触发为高电平，引脚输入上升沿或者下降沿时停止定时器 110b：触发为下降沿时停止定时器 111b：保留
10:8 TIMEN	**定时器使能** 000b：定时器始终启动 001b：在定时器 N-1 启动时启动定时器 010b：触发为高电平时启动定时器 011b：触发为高电平和引脚输入高电平时启动定时器 100b：在引脚输入上升沿时启动定时器 101b：在引脚输入上升沿和触发为高电平时启动定时器 110b：在触发为上升沿时启动定时器 111b：在触发为上升沿或下降沿时启动定时器
5:4 TSTOP	**定时器停止位** 00b：停止位禁用 01b：在定时器产生比较事件时启用停止位 10b：停止位在定时器禁用时启用 11b：在定时器产生比较事件时和定时器禁用时启用停止位
1 TSTART	**定时器起始位** 0：禁用起始位 1：启用起始位

29.6.18 定时器比较值寄存器 n(EIO_TIMCMP0 – EIO_TIMCMP3)

表 29-33 EIO_TIMCMPn 寄存器

EIO_TIMCMPn **定时器比较值寄存器** **复位值：**

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMP															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2013 - 2024 杰发科技有限公司

字段	说明
15:0 CMP	**定时器比较值** **在 双 8 位波特计数器模式下(TMOD = 01b)**，低 8 位配置波特率分频器等于(CMP[7:0] + 1) * 2。高 8 位配置每个字的位数等于 (CMP [15:8] + 1) / 2。 **在 双 8 位 PWM 模式下(TMOD = 10b)**，低 8 位将输出的高电平周期配置为 (CMP[7:0] + 1)，高 8 位将输出的低电平周期配置为 (CMP[15: 8] + 1)。 **在 单 16 位计数器模式下(TMOD = 11b)**，比较值可以用来产生波特率分频器（如果移位时钟源是定时器输出）等于 (CMP[15:0] + 1) * 2。当移位时钟源是一个引脚或触发输入，比较寄存器用于设置每个字的位数等于（CMP[15:0] + 1）/2。

© 2013 - 2024 杰发科技有限公司

30 模数转换器（ADC）

30.1 简介

AC7840x 中有 2 个相同的 ADC 模块。其采用的 ADC 是逐次逼近型模拟数字转换器，分辨率为 12bit，拥有最多 24 路外部通道，支持单次、连续、扫描或间断转换等多种模式。

该 ADC 支持模拟监控器功能，可以监测输入电压是否超出设定的电压范围，并可在超出范围时发出中断通知应用程序。

30.2 特性

- 2 个独立的 ADC 模块
- 支持 12/10/8 位分辨率
- 通道输入电压范围：AVSS < Vin < AVDD
- 支持软件选择 ADC 参考电压：VDDA 或 VREF
- 12bit 分辨率最大转换率为 1Msps，10bit 分辨率最大转换率为 1.33Msps，8bit 分辨率最大转换率为 1.46Msps
- 28 路通道：24 路单端外部通道， 1 路内部温度传感器（T-Sensor），1 路内部带隙基准电压（Bandgap），1 路内部供电（Internal Supply），1 路内部校准信号通路，每路通道可单独配置采样时间
- 转换序列分为 规则组（regular group）和注入组（injection group）
  - 规则组：一个或多个通道可配置为规则组，最多可配置 24 个通道，可选择任意通道
  - 注入组：一个或多个通道可配置为注入组，最多可配置 4 个通道，可选择任意通道
- 8 种操作模式 (方便起见，称为 mode x，x=1~8)
  - 规则组单通道单次转换(mode1)
  - 规则组单通道连续转换(mode2)
  - 规则组扫描+注入组扫描模式多通道单次转换(mode3 注入组扫描模式)
  - 规则组扫描+注入组间隔模式多通道单次转换(mode3 注入组间隔模式)
  - 规则组扫描+自动触发注入组扫描模式多通道单次转换(mode4)
  - 规则组扫描+注入组扫描模式多通道连续转换(mode5 注入组扫描模式)
  - 规则组扫描+注入组间隔模式多通道连续转换(mode5 注入组间隔模式)
  - 规则组扫描+自动触发注入组扫描模式多通道连续转换(mode6)
  - 规则组子组扫描模式转换(mode7)

© 2013 - 2024 杰发科技有限公司

  - 注入组子组扫描模式转换(mode8)
- 通过内部软件触发或外部硬件触发启动 ADC
- 模拟监控器功能：
  - 配置为单个或所有通道电压检查
  - 监控通道电压是否低于低阈值或高于高阈值
- 中断：
  - 规则组和注入组中每一个通道转换结束标志(EOC，End Of Conversion)
  - 模拟监控器事件(AMO)
- DMA 访问，仅用于规则组通道
- 低功耗模式下支持使用 ADC 模拟信号监控功能进行唤醒
- 支持使用内部信号进行自校验和自定义 OE/GE 配置
- 支持 4/8/16/32 次均值功能

30.3 结构框图

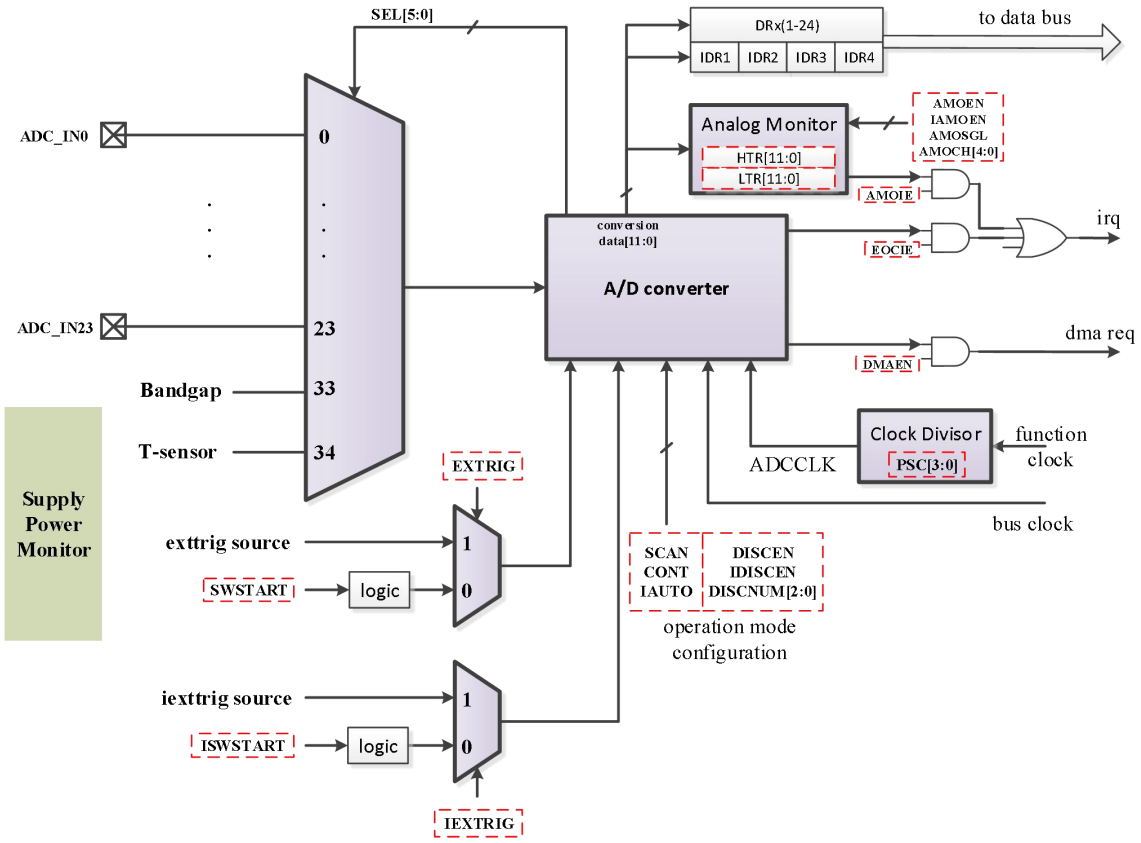


图 30-1 ADC 结构框图

© 2013 - 2024 杰发科技有限公司

30.4 功能描述

每一个 ADC 主要由 A/D 转换器单元（AD converter unit），输入通道选择器（input channel selector），时钟分频器（clock divisor）和模拟监控器（analog monitor）等组成。如图 30-1 所示，A/D 转换器单元转换使用的转换时钟是通过外部 function clock，简称 FCLK，经过 ADC 内部分频器后得到的时钟，简称为 ADCCLK。ADC 的 funciton clock 可通过 CKGEN 模块进行时钟源的选择及使能，具体参考 CKGEN_PERI_CLK_MUX0 描述。function clock 旁还有一个外部输入时钟，名字是 bus clock，简称 BCLK，该时钟用于 ADC 的寄存器读写访问，该时钟是来自 CKGEN 输出的 BUS_CLOCK。

下面介绍一个典型的操作流程。

1. ADC 首先通过 ADC_CTRL1[ADON]上电，然后可以通过模块内部 ADC_CTRL0[SWSTART]或外部触发源触发 ADC，其中外部触发来源于其它模块。
2. 触发后 ADC 转换器单元开始工作，并将通道选择信号发送至输入通道选择器，根据规则或注入组通道序列逐个选择所需的通道。
3. 在一个通道完成转换后，转换结果将根据当前转换通道所属的组存储到 ADC_RDRx 或 ADC_IDRx 中，并且产生相应的 EOC 标志置位。
4. 模拟监控器工作时，如果发生相应的事件则会出现相关的状态标志。

需要指出的是，不同的工作模式在工作流程上存在一些差异，详细信息请参考 30.4.2。

30.4.1 上电时序

在开始所有功能之前， ADC 首先上电，然后有效的触发信号可以启动 ADC 以配置的模式工作。上电时序如图 30-2 所示，将 ADC_CTRL1 [ADON]位置为 1 以控制上电过程。在 ADON 置位后，A/D 转换器单元上电等待时间不应低于 100μs。

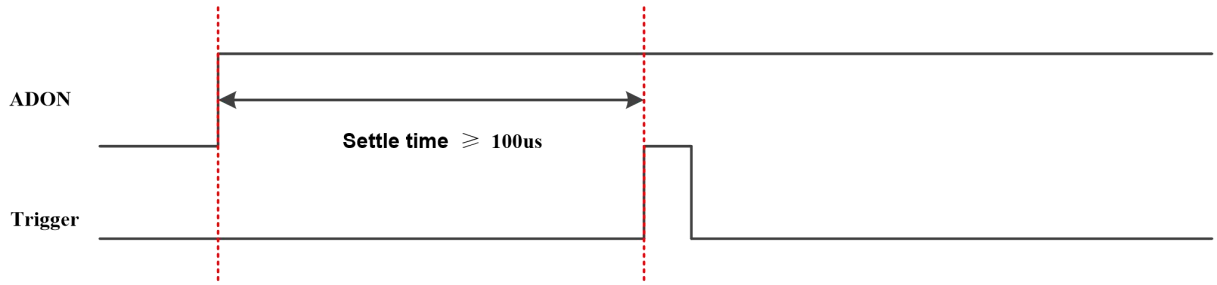


图 30-2 ADC 上电时序

30.4.2 工作模式

根据实际应用可以灵活使用不同的模式，上电和有效触发后 ADC 工作于以下模式之一。

© 2013 - 2024 杰发科技有限公司

表 30-1 工作模式配置表

工作模式	MODE_BITS	触发源	转换序列
mode1	5'b0000x	规则触发	规则组单通道单次转换
mode2	5'b0100x	规则触发	规则组单通道连续转换
mode3 (注入组扫描模式)	5'b10000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式多通道单次转换
mode3 (注入组间隔模式)	5'b10000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式多通道单次转换
mode4	5'b10001	规则触发+自动注入触发	规则组扫描+注入组扫描模式多通道单次转换
mode5 (注入组扫描模式)	5'b11000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式多通道连续转换
mode5 (注入组间隔模式)	5'b11000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式多通道连续转换
mode6	5'b11001	规则触发+自动注入触发	规则组扫描+注入组扫描模式多通道连续转换
mode7	5'b1x10x	规则触发	规则组子组扫描模式转换
mode8	5'b1x01x	注入触发	注入组子组扫描模式转换



说明：

MODE_BITS = {SCAN, CONT, DISCEN, IDISEN, IAUTO}，依次对应 ADC_CTRL0 [16:12]

在描述每个模式操作流程之前，有必要介绍一些术语，例如规则组，注入组等。

ADC 的输入通道中，编号 0-23 是外部输入通道，本文中称为 ch0-ch23，其他编号为内部通道，请参考 ADC_RSQRx 或 ADC_ISQRx 中 SEQ 的编号定义。

AC7840x 的 2 个 ADC 模块各包含一个规则组和一个注入组可配置。规则组和注入组均可配置转换零个或多个 ADC 输入通道。同一组或不同组中的输入通道配置，除了最大数量外，没有额外限制，即任意一个输入通道，如 ch0，可以配置到规则组或注入组，或两个组都配置，甚至一个组中配置多个 ch0 都可以。

规则组和注入组的区别，除了最大通道数，并没有区别，只是在不同模式下能够转换的组，以及转换方式有区别。详细内容可参考本章节后续各个工作模式的描述。

当 ADC 配置为可使用规则组和注入组进行采样的模式时，如果某个组被触发，会根据对应的序列寄存器，按顺序进行转换。如果是规则组，则会按照 ADC_RSQR0，ADC_RSQR1 和 ADC_RSQR2 寄存器的顺序，依次转换寄存器中配置的通道。与规则组相应，注入组则是按照 ADC_ISRQ0, ADC_ISRQ1 等寄存器的顺序，依次转换寄存器中配置的通道。

© 2013 - 2024 杰发科技有限公司

例如，如果 RSQ0~RSQ11 分别设置为 9,8,12,1,5,4,7,3,13,2,0,0，且规则组长度为 12，则规则组将按图 30-3 所示顺序依次进行转换。

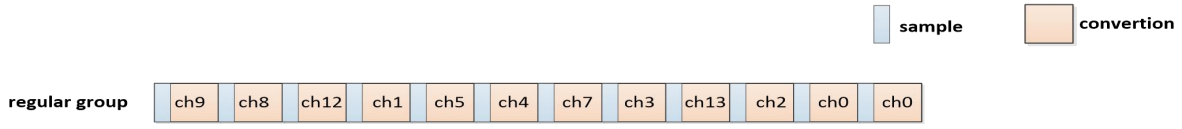


图 30-3 规则组序列

如果 RSQL 设置为 8 (Length=9)，则最后 3 个通道将无效且无法转换。因此，有效的规则组序列如图 30-4 所示。

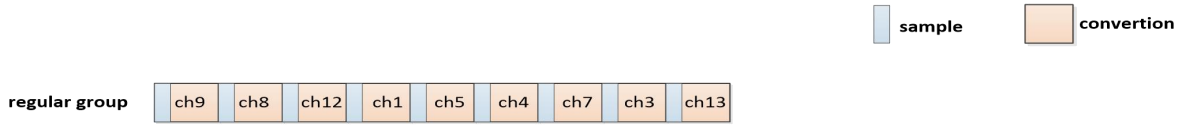


图 30-4 有效规则组序列

以同样的方式，注入组由最多 4 个通道组成，顺序依次为 ISQ0 至 ISQ3。
例如，如果 ISQ0~ISQ3 分别设置为 12,7,13,2，则将注入组按图 30-5 所示进行排列。

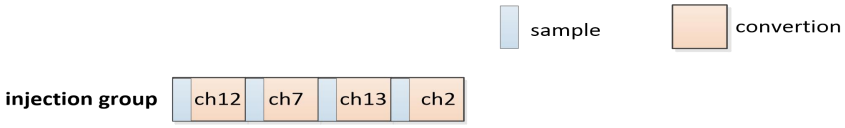


图 30-5 注入组序列

如果 ISQL 设置为 2 (Length=3)，则最后 1 个通道将无效并且不会被转换。因此，有效注入组序列如图 30-6 所示。

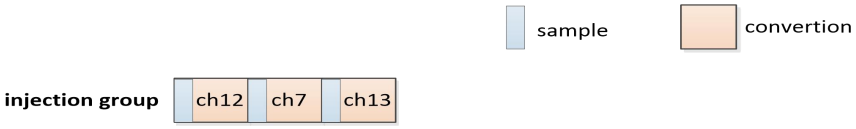


图 30-6 有效注入组序列

显然，规则组触发和注入组触发是开始转换规则组和注入组序列的相应信号。该触发源自 ADC 内部 ADC_CTRL0 的 SWSTART、ISWSTART 或外部触发源。当 ADC 处于规则组通道转换过程中时，规则触发无效。基于此基本介绍，每种模式的详细描述如本节后续所述。

30.4.2.1 Mode 1

此模式下，无论 ADC_SQL[RSQL]为何值，ADC 仅转换规则组的第一个通道。模式按表 30-1 进行配置后，有效触发可使 ADC 工作在此模式。

© 2013 - 2024 杰发科技有限公司

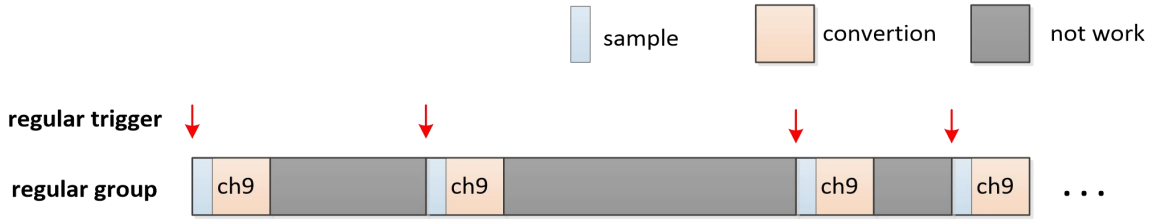


图 30-7 Mode 1 工作流程

如图 30-7 所示，规则组中的第一个通道在有效的规则触发后转换一次。然后 ADC 进入空闲状态，直到下一次有效规则触发带来的下一次转换。

30.4.2.2 Mode 2

此模式下，无论 RSQL 为何值，只会连续转换规则组中的第一个通道。模式按表 30-1 进行配置后，有效触发可以使 ADC 在此模式下工作。

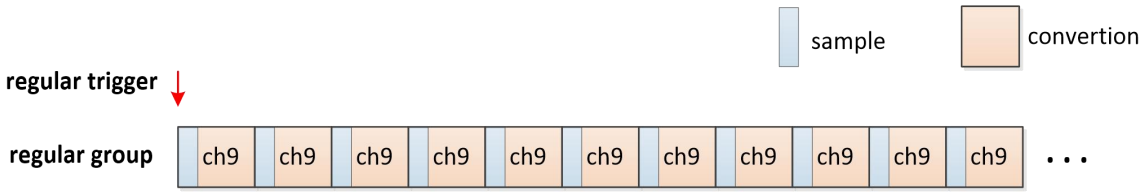


图 30-8 Mode 2 工作流程

如图 30-8 所示，在有效的规则触发后，规则组第一个通道将不断转换，除非，断电/复位或者更改 ADC 工作模式。

30.4.2.3 Mode 3

30.4.2.3.1 interval bit=0，注入组为扫描模式

此模式转换规则组通道和注入组通道。有效的规则和注入组通道长度分别由 ADC_SQL 中的 RSQL 和 ISQL 决定。使用表 30-1 中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6 (Length=7)，ISQL 设置为 2 (Length=3)，一个典型操作如图 30-9 所示。

第一次规则触发会触发 ADC 转换规则组中的 7 个通道。当 ADC 正在转换规则组中的 ch1 时，此时产生注入触发，会不等待 ch1 的转换完成，而立刻（延时不超过 1 个转换周期）切换至转换 3 个注入组通道，在所有注入组通道转换完成后，自动切换回规则组通道，重新开始 ch1 的转换，完成全部的规则通道转换后，ADC 将运行至空闲状态，直至下一次触发到来。

© 2013 - 2024 杰发科技有限公司

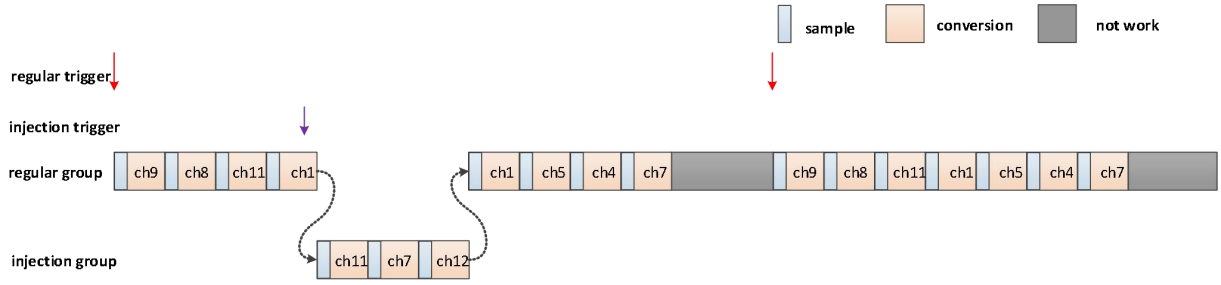


图 30-9 Mode 3 注入组扫描模式工作流程

如果在 ADC 空闲时发生注入触发，ADC 将完成有效注入组通道的转换，如图 30-10 所示。

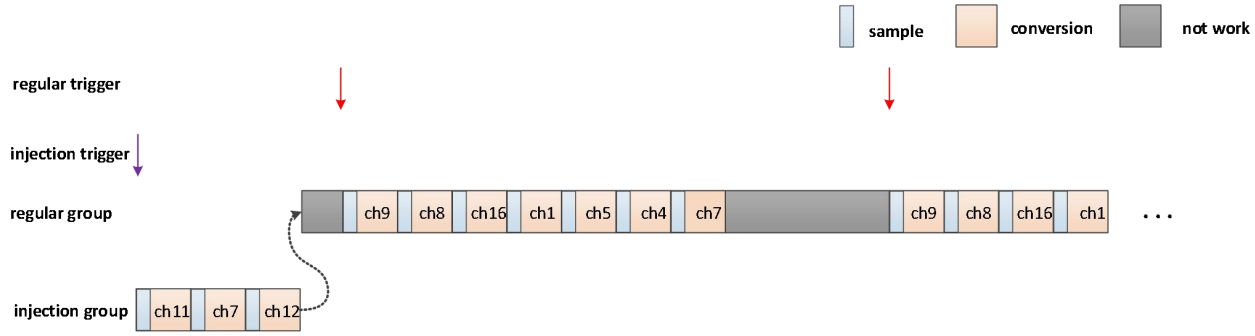


图 30-10 Mode 3 在 ADC 空闲状态下具有注入触发的工作流程

30.4.2.3.2 interval bit=1，注入组为间隔模式

与图 30-9 的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换。

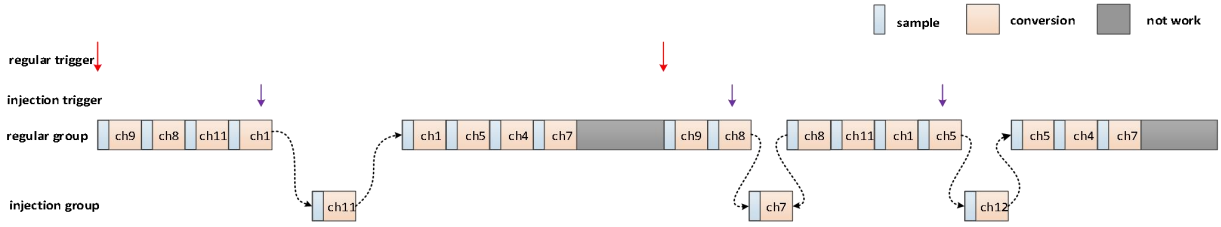


图 30-11 Mode 3 注入组间隔模式工作流程

如果该模式下在 ADC 空闲时发生注入触发，ADC 将完成一个有效注入组通道的转换，如图 30-12 所示：

© 2013 - 2024 杰发科技有限公司

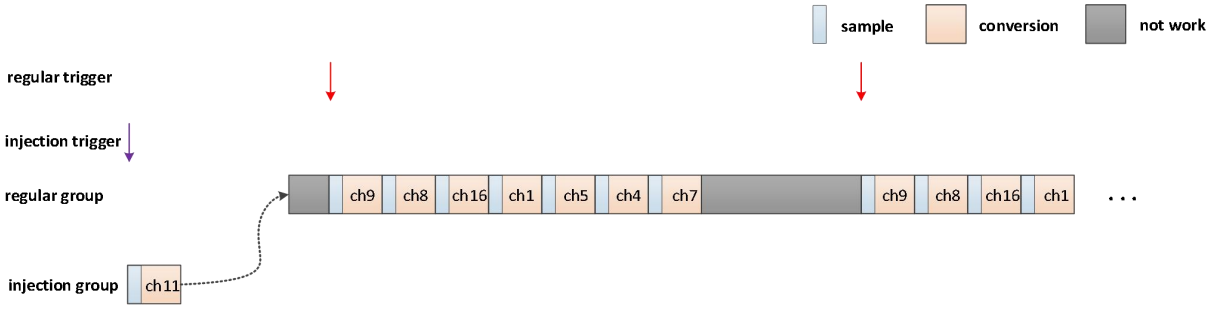


图 30-12 Mode 3 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程

30.4.2.4 Mode 4

此模式下收到规则触发后将自动按照先转换规则组通道，后转换注入组通道的顺序进行转换。有效的规则组通道和注入组通道分别由 RSQL 和 ISQL 决定。使用表 30-1 中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6(Length=7)，ISQL 设置为 2(Length=3)。典型操作如图 30-13 所示。规则触发器开始转换前 7 个规则组通道，然后自动开始转换 3 个注入组通道。在总共 10 个通道均完成完全转换后，ADC 将运行至空闲状态，直到下一个有效的规则触发。

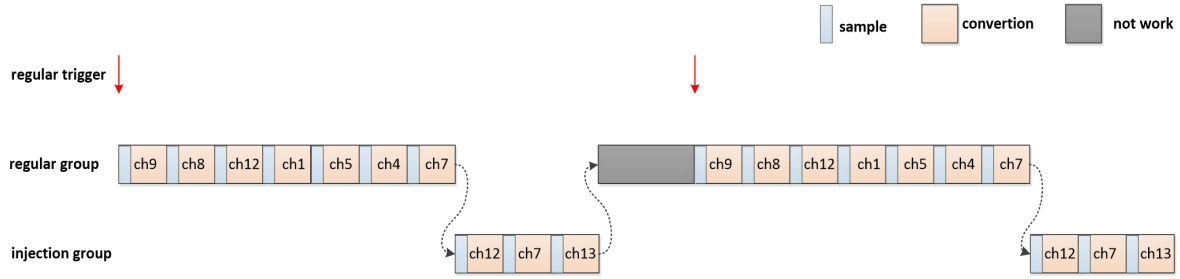


图 30-13 Mode 4 工作流程

30.4.2.5 Mode 5

30.4.2.5.1 interval bit=0，注入组为扫描模式

与 Mode3 区别在于此模式为规则组连续转换。使用表 30-1 中的模式配置，有效触发可使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电、复位或模式更改。例如，RSQL 设置为 6(Length=7)，ISQL 设置为 2(Length=3)。一个典型操作如图 30-14 所示。在规则触发后，ADC 按规则组通道顺序工作，如果发生注入触发，则在注入组通道上工作。

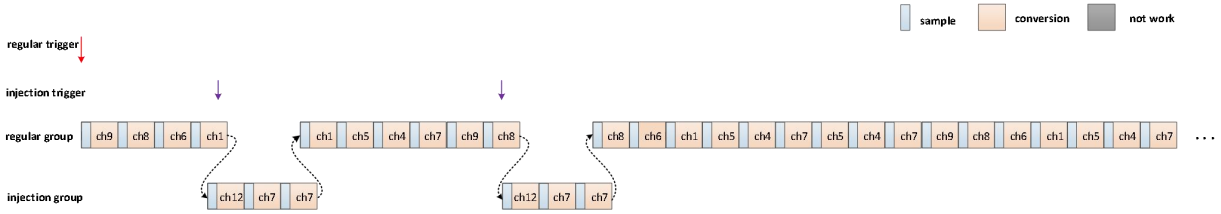


图 30-14 Mode 5 注入组扫描模式工作流程

© 2013 - 2024 杰发科技有限公司

特别地，如果在 ADC 空闲时发生注入触发，ADC 将首先完成有效注入组通道的转换，如图 30-15 所示。

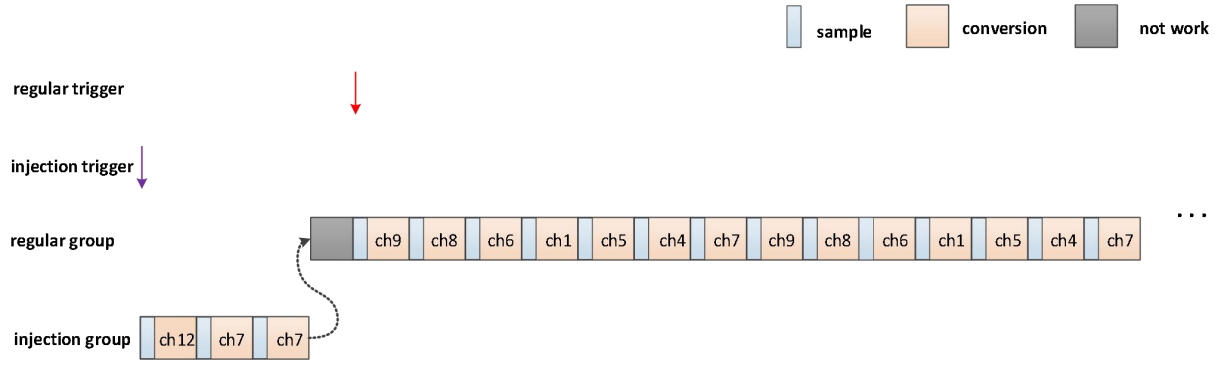


图 30-15 Mode 5 在 ADC 空闲状态下具有注入触发的工作流程

30.4.2.5.2 interval bit=1，注入组为间隔模式

与图 30-14 的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换。工作流程如图 30-16 所示。

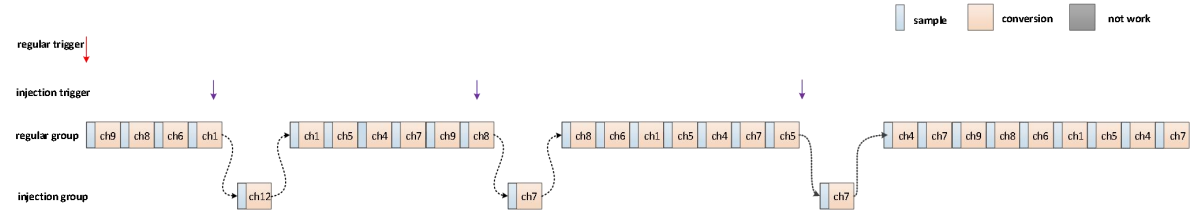


图 30-16 Mode 5 注入组间隔模式工作流程

如果该模式下在 ADC 空闲时发生注入触发，ADC 将完成一个有效注入组通道的转换，如图 30-17 所示：

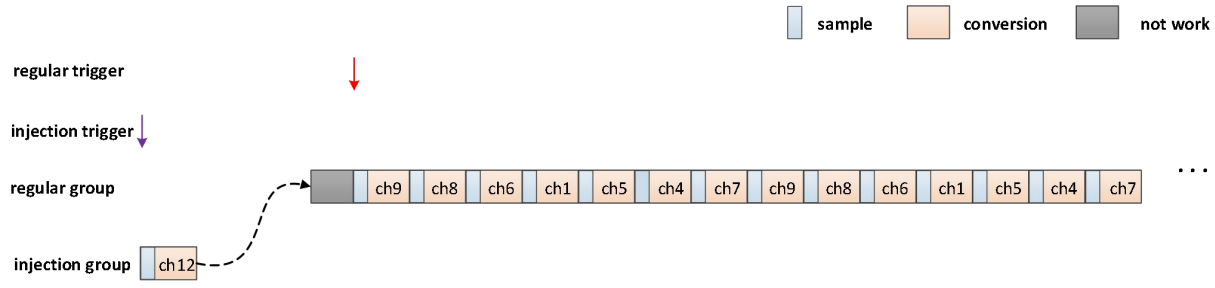


图 30-17 Mode 5 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程

30.4.2.6 Mode 6

与 Mode4 区别在于此模式为连续转换。使用表 30-1 中的模式配置，有效的规则触发可以使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电，复位或模式更

© 2013 - 2024 杰发科技有限公司

改。例如，RSQL 设置为 6(Length=7)，ISQL 设置为 2(Length=3)，操作流程如图 30-18 所示。ADC 在规则组通道上按顺序工作，然后在规则组转换完成后转换注入组通道。

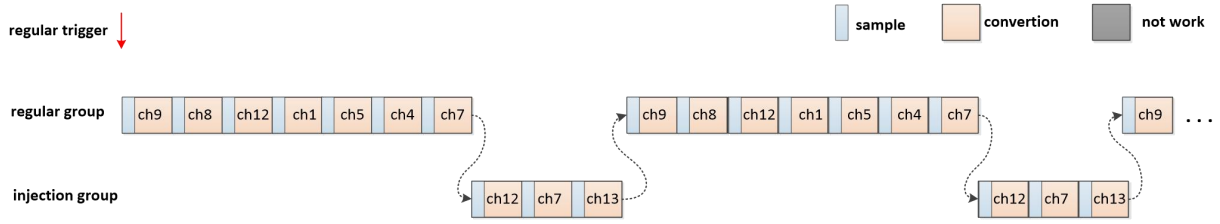


图 30-18 Mode 6 操作流程

30.4.2.7 Mode 7

此模式仅转换规则组通道。有效的规则组通道由 RSQL 决定。使用表 30-1 中的模式配置，ADC 可以在此模式下工作。依据 ADC_CTRL0[DISCNUM]将有效的规则通道分成若干子组。

例如，RSQL 设置为 6(Length=7)，DISCNUM 设置为 1（Length=2）。

第一次规则触发： ch9, ch8；

第二次规则触发： ch12, ch1；

第三次规则触发： ch5, ch4；

第四次规则触发： ch7；

因此，实际的转换流程如图 30-19 所示。

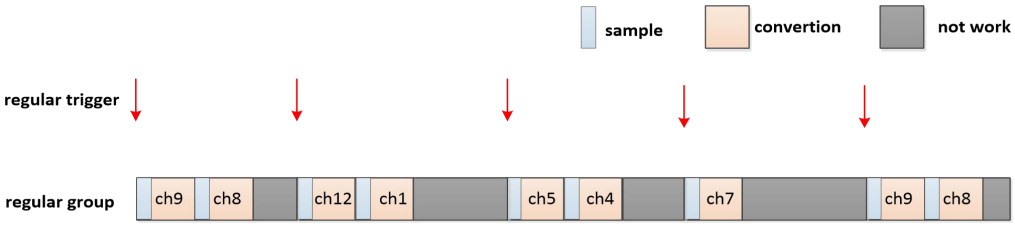


图 30-19 Mode 7 操作流程

30.4.2.8 Mode 8

该模式仅转换注入组通道。有效的注入通道组通道由 ISQL 决定。使用表 30-1 中的模式配置，ADC 可以在此模式下工作。每次触发只转换一个通道。例如，ISQL 设置为 2(Length=3)。

第一次注入触发：ch12；

第二次注入触发：ch7；

第三次注入触发：ch13；

第四次注入触发：ch12；

…

因此，实际的转换流程如图 30-20 所示。

© 2013 - 2024 杰发科技有限公司

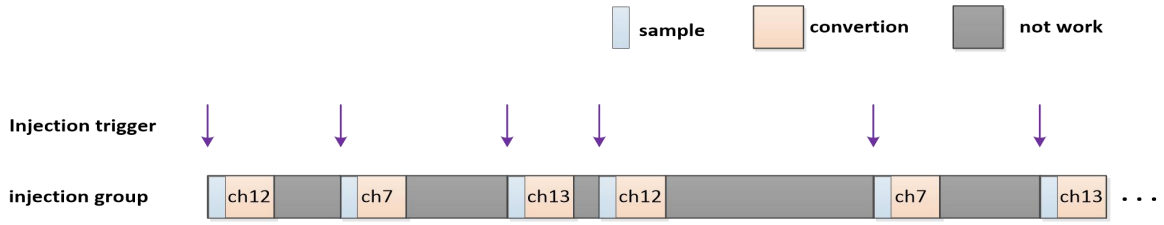


图 30-20 Mode 8 操作流程

30.4.3 触发方式

根据不同的触发方式可以组合出以下 7 种情景。

表 30-2 不同触发方式下的响应行为

触发方式	响应行为
触发规则组	规则组转换
触发注入组	注入组转换
在规则组转换期间产生规则触发	规则组持续转换，第二次触发事件不响应
在规则组转换期间产生注入触发	不等待当前转换通道完成转换，在小于一个转换周期的时间内立刻切换到注入组，注入组转换完后切换到原来的规则组中被打断的通道继续执行（规则组序列未转换完的情况下）
在注入组转换期间产生注入触发	注入组持续转换，第二次注入触发事件不响应
在注入组转换期间产生规则触发	在注入转换期间产生一次规则事件，注入转换不会被打断，但是规则序列将在注入序列结束后被执行
规则触发和注入触发同一时刻产生	注入组先转换，完成后再转换规则组 当规则组正在进行转换时，此时的行为可认为是收到一次注入触发和一次冗余规则触发。 当注入组正在进行转换时，此时的行为可认为是收到一次冗余的注入触发和一次规则触发。

30.4.4 模拟监控器

模拟监控器支持电平触发监控事件和边沿触发监控事件模式，当监控通道的电压值超出阈值范围时，模拟监控器将产生监控事件。

模拟监控器需要监测的通道可通过 ADC_CTRL0 的 AMOEN，IAMOEN，AMOSGL 和 AMOCH 位来配置。其中：

1. AMOEN 为 1 表示模拟监控器会对规则组通道进行监控，为 0 则不监控规则组；
2. IAMOEN 为 1 表示模拟监控器会对注入组通道进行监控，为 0 则不监控注入组；
3. AMOSGL 为 1 表示模拟监控器会对指定通道进行监控，此时通过 AMOCH 来指定监控的通道。为 0 则不对单个通道进行监控。

由于模拟监控器需要指定被监控的通道，如果当前 ADC 未对监控通道进行转换，则模拟监控器无法发挥作用。关于 ADC 工作模式及模拟监控器监控配置的关系可参考表 30-3。

© 2013 - 2024 杰发科技有限公司

表 30-3 模拟监控通道配置

模拟监控通道	{AMOEN,IAMOEN,AMOSGL}	工作模式	注释
无	3'b00x	所有模式	-
所有注入组通道	3'b010	mode3/4/5/6/8	-
所有规则组通道	3'b100	除了 mode8	-
所有通道	3'b110	所有模式	-
单注入组通道	3'b011	mode3/4/5/6/8	转换序列必须包含由 AMOCH [5：0]指定的注入通道
单规则组通道	3'b101	mode1 ~ 7	转换序列必须包含由 AMOCH [5：0] 指定的规则通道
单规则组或注入组通道	3'b111	所有通道	转换序列必须包含由 AMOCH [5：0] 指定的规则或注入通道

阈值通过 ADC_AMOHR 和 ADC_AMOLR 寄存器进行配置。这两个寄存器各个域，即 AMOHT/ AMOLT/ AMOHO/ AMOLO 配置的单位是 LSB，即一个 ADC code 对应的电压。注意，阈值比较使用的是原始数据进行比较，设置数据对齐或注入组偏移不影响比较结果。

30.4.4.1 电平触发模式

设置 ADC_CTRL0[AMOMODE]=0，模拟监控器工作于电平触发模式。
如果被监控通道当前的转换结果大于高阈值 ADC_AMOHR[AMOHT]或小于低阈值 ADC_AMOLR [AMOLT]，则模拟监控器将 AMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生中断。

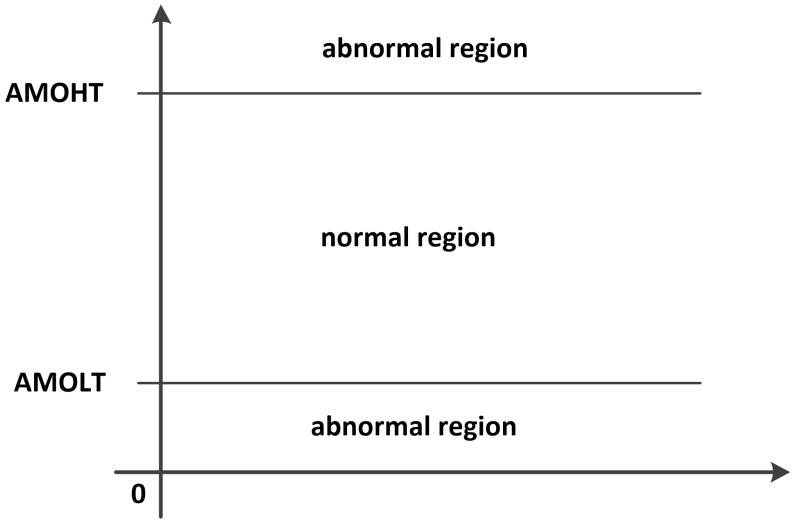


图 30-21 电平触发模式下监控区域

30.4.4.2 边沿触发模式

设置 AMOMODE=1，模拟监控器工作于边沿触发模式。

© 2013 - 2024 杰发科技有限公司

当监控通道电压从正常区域（AMOLO 和 AMOHO 之间，包括 AMOLO 和 AMOHO）到异常区域（低于低阈值 AMOLO 或高于高阈值 AMOHO）时则产生一次监控异常事件，模拟监控器将 AAMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生监控事件中断。

当监控通道电压从异常区域到正常区域，则产生一次监控恢复事件，模拟监控器将 NAMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生监控事件中断。异常区域是低于低阈值 AMOLO 或高于高阈值 AMOHO，正常区域边界值 = [高阈值-高偏移值，低阈值+低偏移值]，即[AMOHT-AMOHO, AMOLT+AMOLO]，如图 30-22 所示。

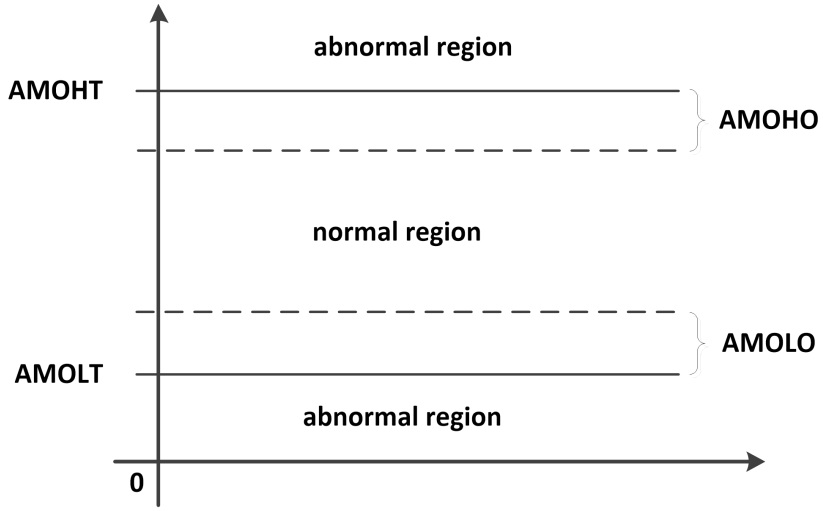


图 30-22 边沿触发模式下监控区域

说明：
边沿触发模式建议只在监控单通道时使用，因为监控多通道无法区分不同通道触发的异常和恢复中断。

30.4.5 校准

AC7840x 的 ADC 模块支持软硬件结合的校准功能。该方法主要应用于中等精度 SAR ADC，由于通道数目多，开关个数多，导致实际应用时会产生非理想因素带来的性能下降，主要表现在静态性能的恶化。码值校正的目的是将非理想的 ADC 传递函数曲线通过测量、换算，优化成接近理想传递函数的曲线，进而提高静态性能。该过程如图 30-23 所示。

© 2013 - 2024 杰发科技有限公司

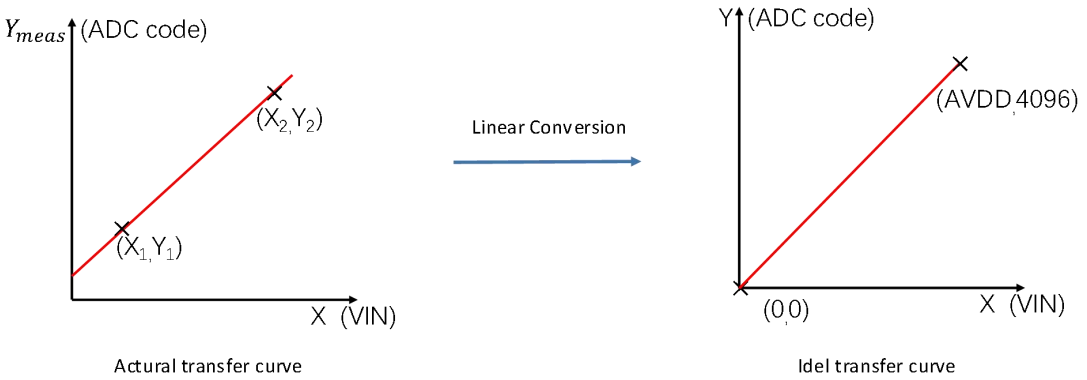


图 30-23 ADC GEOE 校准转换示意

需要注意的是，该校正无法改变 ADC 自身传递曲线，只能对传递曲线进行移位优化，故无法优化动态性能。

硬件部分提供了多个用于 GE OE 校准的内部电压信号，可通过配置 ADC_RSQRx 或 ADC_ISQRx 选择 GEOE 校准通道，然后配置 AUXADC_CFG1 或 AUXADC_CFG5 中的 RG_ADCx_GEOECAL_VIN_SEL 选择校准信号。

软件部分需要选择 2 个校准信号，并得到对应信号的 ADC 转换结果，然后根据如下公式计算 GE OE 的值，并存储到 ADC_CALI0 和 ADC_CALI1 寄存器。

$$\mathrm{GE} = \left(\frac{(2^N - 1)(x2 - x1)}{Y2 - Y1} - 1\right) \times 2^{12}$$

$$\mathrm{OE} = \left(\frac{Y2 * x1 - Y1 * x2}{Y2 - Y1}\right) \times 2^{12}$$

其中$x1$ 和$x2$ 分别为校准信号 1 和 2，比如校准信号 1 选择参考电压的 4/16，则公式中的$x1$ 则为 4/16。$Y1$ 和$Y2$ 分别对应校准信号 1 和 2 的 ADC 转换结果。N 是 ADC 的分辨率位数。

该 GE OE 值填入相应寄存器后，ADC 会自动将转换的数据根据内置公式转换成校准之后的值，填入 ADC_RDRx 或 ADC_IDRx。

该校准流程一般在以下两种情况下需要执行：

1. 在 MCU 上电启动后执行一次；
2. 外部供电电压或 ADC 参考电压发生变化时执行一次。

© 2013 - 2024 杰发科技有限公司

说明：

由于实际进行校准时的电压信号会带有白噪声，因此需要多次获取校准信号的 ADC 转换结果，并进行平均来得到 Y1 和 Y2。一般建议均值次数为 8 次或更多。

一般情况下，内部通道和外部通道可使用相同 GE OE 值。

在使用内部电压信号进行校准时，建议使用 4/16 和 12/16 这两个档位进行校准。

更新校准电压只能用 VREF 做参考，且 VREF 须接稳定电源，否则不建议调用自校准相关接口，使用寄存器 GE OE 的值。

30.4.6 数据对齐方式

数据对齐功能用于控制 ADC 转换结果数据在数据寄存器中的排布，可选择左对齐或右对齐方式。该排布方式会受以下两个功能的影响：

1. 注入组有数据偏置功能，因此注入组的数据寄存器包含符号位；
2. ADC 可选择不同分辨率，因此在不同分辨率配置下，数据的位置会有不同。

各个配置下的数据对齐方式如图 30-24、图 30-25、图 30-26 所示：



图 30-24 12-bit 配置下的 ADC 数据排布

© 2013 - 2024 杰发科技有限公司



图 30-25 10- bit 配置下的 ADC 数据排布



图 30-26 8- bit 配置下的 ADC 数据排布

30.4.7 采样转换时间

ADC 使用若干个 ADCCLK 周期对输入电压采样，即对 ADC 内部电路进行充电，使其达到外部输入信号的电平，完成采样之后才能进行模拟到数字的转换。采样周期个数可通过 ADC_SPTx (x=0~3) 寄存器中的 SPT[2:0]位配置。每个通道可以分别用不同的时间进行采样。 完成采样后，需根据 ADC 分辨率位数来进行模数转换，ADC 分辨率通过 ADC_CTRL1 [RES]配置。

总转换时间公式： (SPT+ resolution（12/10/8）)×ADCCLK 周期+5 个 FCLK

© 2013 - 2024 杰发科技有限公司

例：当 FCLK=60MHz，ADCCLK=30MHz，SPT=15 ADCCLK，采用 12bit 模式，总转换时间=(15+12)/30+(5/60) ≈ 1μs。

注意：
ADCCLK 时钟最高支持 30MHz，FCLK 最高支持 60MHz 时钟，需保证 CKGEN 提供的 ADC FCLK 及 ADC_CTRL1 [PSC]能够满足该条件。
ADC 的最大采样率是 1M，需保证配置的采样率不能超过该值太多。

以 FCLK=60MHz，ADCCLK=30MHz 举例，各个 SPT 配置下的 ADC 采样时间及采样率如下表所示：

表 30-4 ADC 采样率示例

SPT	采样周期(30M)	采样时间(μs)	转换周期(60M)	总转换周期(us)	采样率(kHz)
0	5	0.167	12	0.65	~~1538.46~~
1	10	0.333	12	0.817	~~1224.49~~
2	15	0.5	12	0.983	1016.95
3	23	0.767	12	1.25	800
4	35	1.167	12	1.65	606.06
5	45	1.5	12	1.983	504.2
6	85	2.833	12	3.317	301.51
7	185	6.167	12	6.65	150.38

30.4.8 同步触发模式

同步触发模式可用于使用同一个触发信号同时触发 ADC0 和 ADC1 的规则组或注入组。具体寄存器配置可参考 CTU 模块中的 ADC_SYNC 寄存器。

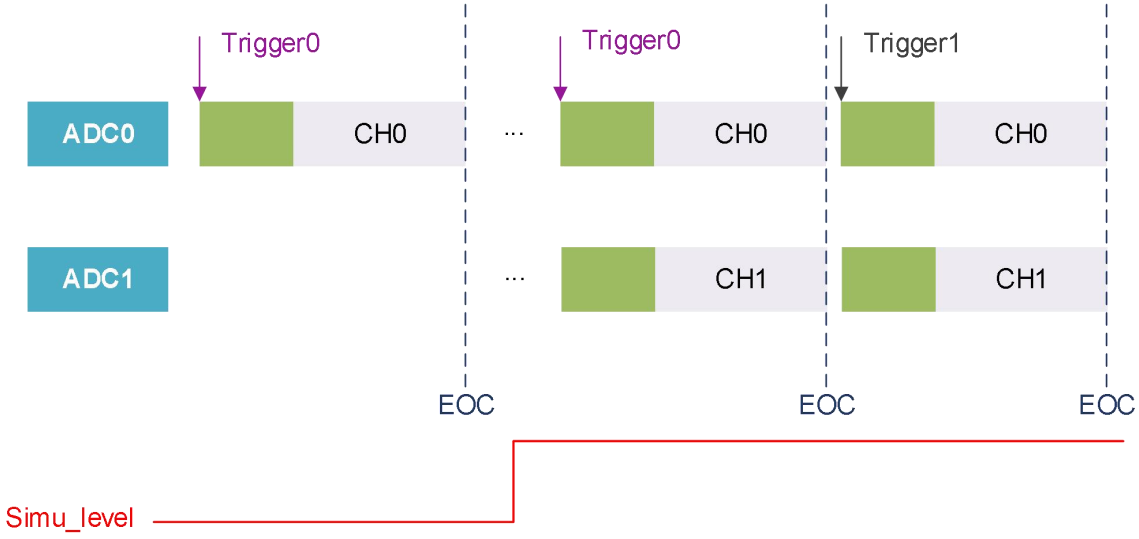


图 30-27 ADC 同步触发时序图

© 2013 - 2024 杰发科技有限公司

图 30-27 中的 Simu_level 对应 ADC_SYNC 的 SIMU_REG 或 SIMU_INJ，可配置是否使能 ADC0 和 ADC1 的规则组或注入组同时触发功能。开启该功能后，触发 ADC0 或 ADC1 都会同时触发两个 ADC 开始转换。

30.4.9 交织模式

交织模式用于让 ADC0 和 ADC1 测量同一个引脚的信号。该功能需要用到 CTU 中的 ADC_SYNC 寄存器。

ADC0 和 ADC1 的所有引脚中，可使用交织模式的引脚连接图可参考图 30-28。

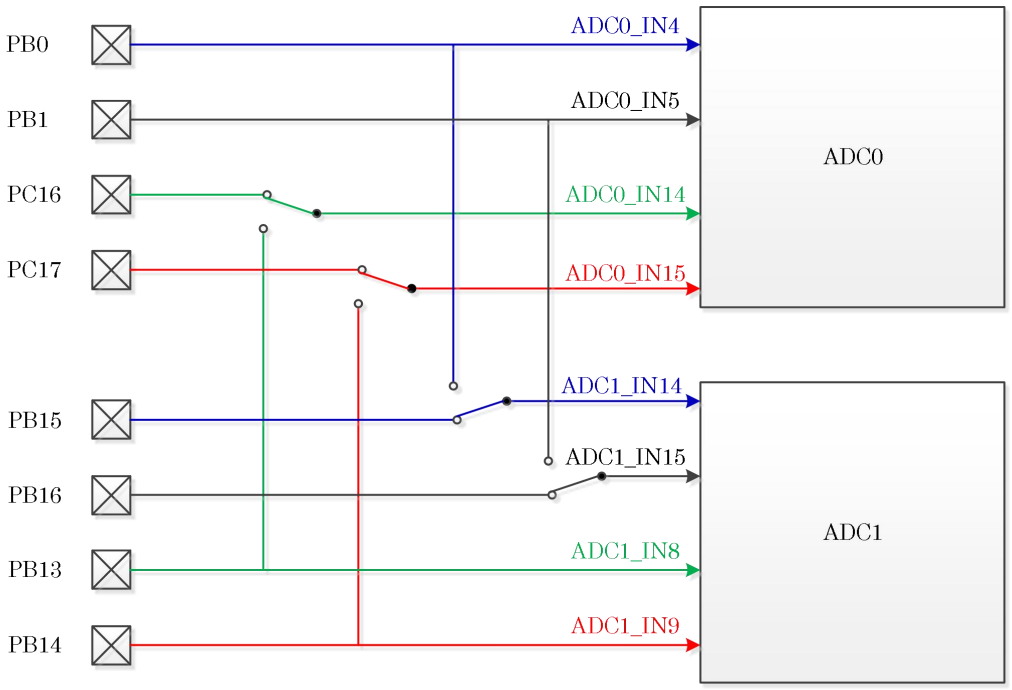


图 30-28 硬件交织

以 ADC0_IN4 和 ADC1_IN14 通道为例，默认状态下这两个外部通道可分别连到两个不同的引脚独立工作，即 ADC0_IN4 连接到 PB0 引脚，ADC1_IN14 连接到 PB15 引脚，如果打开这两个 ADC 通道的交织功能，则 ADC0_IN4 和 ADC1_IN14 同时连接到同一个外部引脚 PB0。

在使用该功能时，有以下几点需注意：

1. 使用该功能需要将默认连接的 ADC 配置为 RG 模式，即如果需要 ADC0 和 ADC1 同时量 PB0 的信号，需要将 ADC0 开启 RG 模式，即配置 AUXADC_CFG0[RG_ADC0_RGMODE_EN]，如果需要 ADC0 和 ADC1 同时量 PB13 的信号，则需要将 ADC1 开启 RG 模式。
2. 开启 RG 模式后，需要使能对应 ADC 的 RG_ADCx_LDO_EN, RG_ADCx_VREFGEN_EN, RG_ADCx_EN。
3. 开启 RG 模式后，需要通过 RG_ADCx_CHANNEL_SEL 寄存器，配置转换的通道，比如要 ADC0 和 ADC1 同时量 PB0 的信号，则需要配置 AUXADC_CFG0[RG_ADC0_CHANNEL_SEL] 为 4。注意，使能交织模式后，两个 ADC 不能选择交织通道之外的通道，即如果 PB0 使用 ADC 交织模式时，则 ADC0 只能转换通道 4，且 ADC1 只能转换通道 14。
4. 对应通道的采样时间配置依然在对应的 ADC_SPT 寄存器。

© 2013 - 2024 杰发科技有限公司

30.4.10 硬件均值功能

硬件均值功能是对同一个信号进行多次采样并进行均值计算的功能。具体采样均值次数配置可参考 ADC_CTRL1 中的 AVGS 的说明。打开硬件均值功能之后，平均后的数据依然存放在对应的数据寄存器中。

由于硬件均值功能开启后，每一个 ADC 数据需要进行多次采样，因此 ADC 的采样时序会发生变化，以 4 次均值为例，使能硬件均值功能之后的时序图如图 30-29 所示。

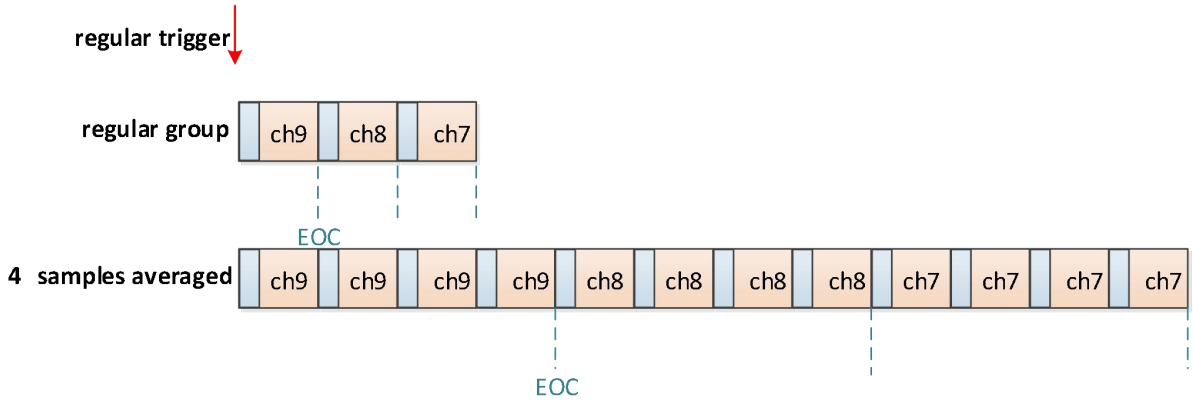


图 30-29 硬件均值时序图

30.4.11 注入组数据偏置功能

注入组数据偏置功能可将注入组转换的结果减去一个固定的偏置数据。由于最多有 4 个注入组通道进行转换，因此该偏置数据寄存器也相应有 4 个。

由于注入组数据减去偏置后可能为负数，因此需要符号位表示注入组数据当前的符号，具体符号位的位置可参考 30.4.6 数据对齐方式 章节中的描述。

30.4.12 温度传感器

温度传感器可以用来测量器件周围的温度(T_A)，与 ADC 内部直接连接，通过 ADC 把传感器输出电压转换成数字量。

内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。内部温度电压值只随温度变化不随外部电源变化。

温度计算公式：温度(℃) = {(V_{SENSE} - V_{TEMP25}) / Slope} + 25

V_{TEMP25}：25℃时的电压数值

V_{SENSE}：当前温度电压数值

Slope：温度传感器的平均斜率(单位为 mV/℃)

具体可参考数据手册的电气特性章节中 **V_{TEMP25}** 和 Slope 的实际值。

© 2013 - 2024 杰发科技有限公司

30.4.13 中断请求

ADC 有两种转换状态标志位：EOC 和 AMO（边沿触发模式使用 AAMO 和 NAMO）。这两种状态标志均有单独的中断使能配置位，当使能相应的中断请求使能时，在标志置起的同时，会触发系统中断。

AC7840x 系列中 ADC 的每个规则组和注入组通道均有单独的 EOC 中断标志。EOC 标志表示规则组或注入组中某个通道的转换结束，产生的 EOC 状态位会在当前完成转换的序列寄存器中置起。

AMO 标志表示是否发生模拟监控器事件。

30.4.14 低功耗模式

ADC 可配置为在系统进入低功耗模式后仍可以工作，此时需要将 ADC 的 FCLK 的时钟源选为 HSI。如果 ADC 在休眠前配置为非连续转换模式（mode 1/3/4/5/7/8），此时需要保证作为 ADC 触发源的模块以及 CTU 处于工作状态，并触发 ADC 转换。如果 ADC 在休眠前配置为连续转换模式，且已开始转换，则系统进入休眠后继续连续转换，若没有开始转换，则和非连续转换模式一样，需要触发源模块及 CTU 处于工作状态，并触发 ADC 转换。

ADC 也可作为唤醒源，ADC 唤醒源可参考表 30-5。

表 30-5 ADC 模块唤醒源

模式	唤醒源
STOP1	AMO 中断，包括 AMO/AMO_NROMAL/AMO_ABNORMAL
STOP2	AMO 中断，包括 AMO/AMO_NROMAL/AMO_ABNORMAL 以及 EOC 中断
VLPS	AMO 中断，包括 AMO/AMO_NROMAL/AMO_ABNORMAL

说明：
系统进入 STOP1/STOP2/VLPS 时，ADC 永远回复 ACK。

30.4.15 调试模式

ADC 无调试模式相关配置。在调试模式下，ADC 可正常工作。

30.5 应用说明

30.5.1 重置和使能

本节介绍了 ADC 的 3 种重置范围，为了更好的使用 ADC，需要了解这 3 种重置范围。

© 2013 - 2024 杰发科技有限公司

表 30-6 ADC 重置范围

编号	重置信号	ADC 重置效果	描述
1	将位于 CKGEN 中的 ADC 相应 reset bit 置位	重置整个 ADC，包括所有寄存器	ADC 所有寄存器值均不会保留。 如果需要重新使用 ADC，需要进行重新配置，也需要在 ADON 置为 1 后等待超过 100μs，再触发转换。
2	将 ADON 置为 0	重置 ADC 内部状态，并重置 RDR/IDR 寄存器	ADC 配置相关寄存器会保留。 如果需要重新使用 ADC，只需要将 ADON 置为 1，并发出触发信号。
3	修改 ADC 工作模式或修改 RSQR、ISQR、RSQL、ISQL 中任意一个	仅重置 ADC 内部状态，不会影响配置寄存器	ADC 配置相关寄存器会保留。 如果需要重新使用 ADC，只需要发出触发信号。

关于使能 ADC 模块，需要在 CKGEN 模块中进行以下操作：

1. 使能 ADC 时钟
2. 关闭 ADC 重置信号
3. 选择 ADC 工作时钟

相关寄存器配置请参考 10.3.2 外设复位功能 和 10.3.3 外设时钟 。

30.5.2 ADC 上电延时

如 30.4.1 上电时序 章节所述，在 ADC 模块第一次上电时，需要有约 100μs 的时间用于模块内部信号的稳定，在系统处于低功耗模式（STOP1/STOP2/VLPS）时，当 ADC 转换结束，则 ADC 会自动断电，如果配置 ADC 为此模式下工作，则当 ADC 收到触发转换的信号时，ADC 存在硬件自动上电的过程，此时 ADC 也存在 100μs 的上电延时。

30.5.3 DMA 访问

当进行规则组的转换时，可通过 DMA 将 ADC 数据保存到内存中。需要注意，在配置 DMA 时，需根据当前 ADC 的配置，设置 DMA 的源地址和结束地址，比如当前使用 mode3，RSQL 设置为 3

说明：

在使用 DMA 功能时，由于 DMA 读取 ADC RDR 的速度较快，EOC 状态和中断请求会被迅速清除， 导致软件和中断控制器无法正常读到 EOC 状态和检测到中断。

(Length=4)，则 DMA 的源地址为 ADC_RDR0，结束地址为 ADC_RDR4 的地址。

© 2013 - 2024 杰发科技有限公司

30.6 寄存器定义

表 30-7 ADC 寄存器映射

ADC0 基地址：0x40003000

ADC1 基地址：0x40004000

ANALOG 基地址：0x40008800

地址	名称	宽度	描述
ADCx 基地址+0x0	ADC_STR	32	状态寄存器
ADCx 基地址+0x4	ADC_CTRL0	32	控制寄存器 0
ADCx 基地址+0x8	ADC_CTRL1	32	控制寄存器 1
ADCx 基地址+0xC ~ ADCx 基地址+0x18	ADC_IOFRx	32	注入组偏移寄存器 0
ADCx 基地址+0x1C	ADC_AMOHR	32	模拟监控器高阈值寄存器
ADCx 基地址+0x20	ADC_AMOLR	32	模拟监控器低阈值寄存器
ADCx 基地址+0x24	ADC_SPT0	32	采样时间寄存器 0
ADCx 基地址+0x28	ADC_SPT1	32	采样时间寄存器 1
ADCx 基地址+0x2C	ADC_SPT2	32	采样时间寄存器 2
ADCx 基地址+0x30	ADC_SPT3	32	采样时间寄存器 3
ADCx 基地址+0x34 ~ ADCx 基地址+0x90	ADC_RSQRx	32	规则组序列配置寄存器
ADCx 基地址+0xB4 ~ ADCx 基地址+0xC0	ADC_ISQRx	32	注入组序列配置寄存器
ADCx 基地址+0xC4	ADC_SQL	32	序列长度寄存器
ADCx 基地址+0xC8	ADC_CALI0	32	校准寄存器 0
ADCx 基地址+0xCC	ADC_CALI1	32	校准寄存器 1
ADCx 基地址+0xE8 ~ ADCx 基地址+0x144	ADC_RDRx	32	规则组数据寄存器
ADCx 基地址+0x168 ~ ADCx 基地址+0x174	ADC_IDRx	32	注入组数据寄存器
ANALOG 基地址+0x60	AUXADC_CFG0	32	ADC analog 配置寄存器 0
ANALOG 基地址+0x64	AUXADC_CFG1	32	ADC analog 配置寄存器 1
ANALOG 基地址+0x70	AUXADC_CFG4	32	ADC analog 配置寄存器 4
ANALOG 基地址+0x74	AUXADC_CFG5	32	ADC analog 配置寄存器 5

© 2013 - 2024 杰发科技有限公司

30.6.1 状态寄存器(ADC_STR)

表 30-8 ADC_STR 寄存器

ADC_STR **ADC 状态寄存器** **复位值：0x00000010**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							COVCFT			AAMO	NAMO	IDLE				AMO
访问							R/W0C			R/W0C	R/W0C	RO				R/W0C
复位值							0			0	0	1				0

字段	说明
9 COVCFT	**触发冲突标志** 0: 没有触发冲突 1: 发生触发冲突，写 0 清除 触发冲突有 2 种情况会产生： 1. 在规则组转换过程中收到规则组触发 2. 在注入组转换过程中收到规则组触发或注入组触发
6 AAMO	**模拟监控器事件发生(边沿触发模式使用)** 0：没有异常事件 1：发生异常事件，写 0 清除
5 NAMO	**模拟监控恢复事件发生标志(边沿触发模式使用)** 0：没有恢复事件 1：发生恢复事件，写 0 清除
4 IDLE	**ADC 空闲状态标志** 0： ADC 处于非空闲状态 1： ADC 处于空闲状态
0 AMO	**模拟监控异常事件发生标志(电平触发模式使用)** 0：没有异常事件 1：发生异常事件，写 0 清除

© 2013 - 2024 杰发科技有限公司

30.6.2 控制寄存器 0(ADC_CTRL0)

表 30- 9 ADC_CTRL0 寄存器

ADC_CTRL0 **ADC 控制寄存器 0** **复位值：0x10000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	SWSTART	ISWSTART					INTERVAL	AMOMODE	ALIGN	IEXTTRIG	EXTTRIG	DMAEN	AMOIE			SCAN
访问	RW	RW					RW	RW	RW	RW	RW	RW	RW			RW
复位值	0	0					0	0	0	0	0	0	0			0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CONT	DISCEN	IDISCEN	IAUTO	DISCNUM			AMOEN	IAMOEN	AMOSGL	AMOCH					
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW					
复位值	0	0	0	0	0	0	0	0	0	0	0					

字段	说明
31 SWSTART	**规则通道软件触发** 写 1 触发，读为 0
30 ISWSTART	**注入通道软件触发** 写 1 触发，读为 0
25 INTERVAL	**间隔模式(仅在 Mode3/5 使用)** 0：注入组为扫描模式 1：注入组为间隔模式
24 AMOMODE	**模拟监控器触发模式** 0：电平触发模式 1：边沿触发模式
23 ALIGN	**数据对齐** 0：右对齐 1：左对齐
22 IEXTTRIG	**注入组触发源选择** 0：内部（软件触发） 1：外部
21 EXTTRIG	**规则组触发源选择** 0：内部（软件触发）

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：外部
20 DMAEN	**DMA 功能使能** 0：禁用 1：使能
19 AMOIE	**模拟监控器中断功能使能** 0：禁用 1：使能
16：12 Modes control bits: SCAN, CONT, DISCEN, IDISCEN, IAUTO	**ADC 工作模式** 详细配置参考 30.4.2 工作模式
11：9 DISCNUM	**通道的不连续转换长度** 0 ~ 7： mode 7 规则组子组长度为 1~8，即 DISCNUM+1
8： 6 Analog monitor control bits: AMOEN, IAMOEN, AMOSGL	**模拟监控通道配置** 详细配置参考 30.4.4 模拟监控器
5：0 AMOCH	**模拟监控通道** 当模拟监控器配置为仅检测单个通道时，指定被监测的通道。该寄存器域的通道编码与 ADC_RSQRx 中 SEQ 域的通道编码相同。

30.6.3 控制寄存器 1(ADC_CTRL1)

表 30-10 ADC_CTRL1 寄存器

ADC_CTRL1　　ADC 控制寄存器 1　　复位值：0x0000F802

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															RESL	
访问															RW	
复位值															0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PSC					AVGE	AVGS								CALEN	ADON
访问	RW	RW	RW	RW			RW	RW							RW	RW
复位值	1	1	1	1		0	0	0							1	0

字段	说明
17：16 RESL	**分辨率选择** 0h: 12-bit 1h: 10-bit 2h: 8-bit

© 2013 - 2024 杰发科技有限公司

字段	说明
	其他值未使用。
15：12 PSC	**总线时钟预分频为 ADC 工作时钟** 0 ~ 15 ： 1 ~ 16 倍分频系数
10 AVGE	**均值功能使能** 0：不使用均值功能 1：使用均值功能
9: 8 AVGS	**转换均值长度选择** 0h：4 次平均 1h：8 次平均 2h：16 次平均 3h：32 次平均
1 CALEN	**校准使能** 0：禁用校准 1：使能校准 注意，该功能一般需要保持使能，以保证 ADC 结果的准确性。
0 ADON	**ADC 上电** 0： ADC 断电，复位 ADC (配置寄存器不会被重置) 1： ADC 上电

30.6.4 注入组偏移寄存器(ADC_IOFRx)

表 30-11 ADC_IOFRx（x= 0 ~ 3）寄存器

ADC_IOFRx（x= 0 ~ 3）　　ADC 注入组偏移寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					IOFR											
访问					RW											
复位值					0											

字段	说明
11：0 IOFR	**注入组偏移值** 注入组通道转换的数据值 IDR 已经减去了在 ADC_IOFR 寄存器中定义的偏移量。

© 2013 - 2024 杰发科技有限公司

30.6.5 高阈值寄存器(ADC_AMOHR)

表 30-12 ADC_AMOHR 寄存器

ADC_AMOHR **ADC 高阈值寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					AMOHO											
访问					RW											
复位值					0											
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					AMOHT											
访问					RW											
复位值					0											

字段	说明
27：16 AMOHO	**模拟监控器高阀值对应的迟滞值** 定义高阀值的迟滞
11：0 AMOHT	**模拟监控器的高阀值** 定义高阀值

30.6.6 低阈值寄存器(ADC_AMOLR)

表 30-13 ADC_AMOLR 寄存器

ADC_AMOLR **ADC 低阈值寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					AMOLO											
访问					RW											
复位值					0											
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					AMOLT											
访问					RW											
复位值					0											

字段	说明
27：16 AMOLO	**模拟监控器低阀值对应的迟滞** 定义低阀值的迟滞
11：0 AMOLT	**模拟监控器的低阀值** 定义低阀值

© 2013 - 2024 杰发科技有限公司

30.6.7 采样时间寄存器 0(ADC_SPT0)

表 30-14 ADC_SPT0 寄存器

ADC_SPT0 **ADC 采样时间寄存器 0** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			SPT9			SPT8			SPT7			SPT6			SPT5[2:1]	
访问			RW			RW			RW			RW			RW	
复位值			0			0			0			0			0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	STP5[0]	SPT4			SPT3			SPT2			SPT1			SPT0		
访问	RW	RW			RW			RW			RW			RW		
复位值	0	0			0			0			0			0		

字段	说明
29：0 SPTx(x=0 ~ 9)	**通道采样时间** 0h：5 ADCCLK 1h：10 ADCCLK 2h：15 ADCCLK 3h：23 ADCCLK 4h：35 ADCCLK 5h：45 ADCCLK 6h：85 ADCCLK 7h：185 ADCCLK SPTx 的编号代表以下通道： 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值对应的 SPT 寄存器位为保留位

© 2013 - 2024 杰发科技有限公司

30.6.8 采样时间寄存器 1(ADC_SPT1)

表 30-15 ADC_SPT1 寄存器

ADC_SPT1 **ADC 采样时间寄存器 1** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			SPT19			SPT18			SPT17			SPT16			SPT15[2:1]	
访问			RW			RW			RW			RW			RW	
复位值			0			0			0			0			0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	STP15[0]	SPT14			SPT13			SPT12			SPT11			SPT10		
访问	RW	RW			RW			RW			RW			RW		
复位值	0	0			0			0			0			0		

字段	说明
29：0 SPTx (x=10 ~ 19)	**通道采样时间** 0h：5 ADCCLK 1h：10 ADCCLK 2h：15 ADCCLK 3h：23 ADCCLK 4h：35 ADCCLK 5h：45 ADCCLK 6h：85 ADCCLK 7h：185 ADCCLK SPTx 的编号代表以下通道： 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值对应的 SPT 寄存器位为保留位

© 2013 - 2024 杰发科技有限公司

30.6.9 采样时间寄存器 2(ADC_SPT2)

表 30-16 ADC_SPT2 寄存器

ADC_SPT2 **ADC 采样时间寄存器 2** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					SPT23			SPT22			SPT21			SPT20		
访问					RW			RW			RW			RW		
复位值					0			0			0			0		

字段	说明
11：0 SPTx(x=20 ~ 23)	**通道采样时间** 0h：5 ADCCLK 1h：10 ADCCLK 2h：15 ADCCLK 3h：23 ADCCLK 4h：35 ADCCLK 5h：45 ADCCLK 6h：85 ADCCLK 7h：185 ADCCLK SPTx 的编号代表以下通道： 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值对应的 SPT 寄存器位为保留位

© 2013 - 2024 杰发科技有限公司

30.6.10 采样时间寄存器 3(ADC_SPT3)

表 30-17 ADC_SPT3 寄存器

ADC_SPT3 **ADC 采样时间寄存器 3** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									SPT37						SPT35[2:1]	
访问									RW						RW	
复位值									0						0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SPT35[0]	SPT34			SPT33											
访问	RW	RW			RW											
复位值	0	0			0											

字段	说明
17：9 23：21 SPTx(x=33/34/35/37)	**通道采样时间** 0h：5 ADCCLK 1h：10 ADCCLK 2h：15 ADCCLK 3h：23 ADCCLK 4h：35 ADCCLK 5h：45 ADCCLK 6h：85 ADCCLK 7h：185 ADCCLK SPTx 的编号代表以下通道： 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值对应的 SPT 寄存器位为保留位

© 2013 - 2024 杰发科技有限公司

30.6.11 规则组序列配置寄存器 (ADC_RSQRx)

表 30-18 ADC_RSQRx 寄存器

ADC_RSQRx（x=0~23） ADC 规则组序列配置寄存器 复位值：0x0000003F

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									EOC	IE	SEQ					
访问									R/W1C	RW	RW					
复位值									0	0	1	1	1	1	1	1

字段	说明
7 EOC	**转换结束标志** 0：转换未结束 1：转换结束 **说明：** **该位除了可通过写 1 清除外，还可以通过读取相应规则组数据寄存器来清除。比如序列号为 0 的 ADC_RSQR0 的 EOC 位可通过读取 ADC_RDR0 来清除，ADC_RSQR1 的 EOC 位可通过读取 ADC_RDR1 来清除。**
6 IE	**EOC 中断使能** 0：不使用 EOC 中断 1：使用 EOC 中断 **说明：** **该位仅用于配置该通道转换结束后是否触发 EOC 中断。**
5：0 SEQ	**规则组的 x 次转换的通道编号** 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值为未使用设置。

© 2013 - 2024 杰发科技有限公司

30.6.12 注入组序列配置寄存器(ADC_ISQRx)

表 30-19 ADC_ISQRx 寄存器

ADC_ISQRx (x = 0~3)　　ADC 注入组序列配置寄存器　　复位值：0x0000003F

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									EOC	IE	SEQ					
访问									RW1C	RW	RW					
复位值									0	0	1	1	1	1	1	1

字段	说明
7 EOC	**转换结束标志** 0：转换未结束 1：转换结束 说明： 该位除了可通过写 1 清除外，还可以通过读取相应规则组数据寄存器来清除。比如序列号为 0 的 ADC_ISQR0 的 EOC 位可通过读取 ADC_IDR0 来清除，ADC_ISQR1 的 EOC 位可通过读取 ADC_IDR1 来清除。
6 IE	**中断使能** 0：不使用 EOC 中断 1：使用 EOC 中断 说明： 该位仅用于配置该通道转换结束后是否触发中断。
5：0 SEQ	**注入组的 x 次转换的通道编号** 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值为未使用设置。

© 2013 - 2024 杰发科技有限公司

30.6.13 序列长度寄存器(ADC_SQL)

表 30-20 ADC_SQL 寄存器

ADC_SQL **ADC 转换组长度寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称								REFSEL						SUPPLY		
访问								RW						RW		
复位值								0						0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							ISQL				RSQL					
访问							RW				RW					
复位值							0	0			0	0	0	0	0	0

字段	说明
24 REFSEL	**ADC 参考电压选择** 0：使用 VREF 1：使用 VDDA
18~16 SUPPLY	**内部供电通道源选择** 0h：VDD，外部输入 1h：VDDA，外部输入 2h：VREFH，外部输入 3h：XOSC 供电，2.4v 4h：Flash 供电 ，2.5v 5h：数字部分供电，1.1v 6h~7h：未使用
9~8 ISQL	**注入组长度** 0~3：注入组长度为 1~4
5~0 RSQL	**规则组长度** 0~23：规则组长度为 1~24

© 2013 - 2024 杰发科技有限公司

30.6.14 校准寄存器 0(ADC_CALI0)

表 30-21 ADC_CALI0 寄存器

ADC_CALI0 **ADC 校准寄存器 0** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				GE1												
访问				RW												
复位值				0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				GE0												
访问				RW												
复位值				0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
28：16 GE1	**内部通道增益误差（Gain Error）** 该域为有符号数，最高位为符号位。
12：0 GE0	**外部通道增益误差（Gain Error）** 该域为有符号数，最高位为符号位。

30.6.15 校准寄存器 1(ADC_CALI1)

表 30-22 ADC_CALI1 寄存器

ADC_CALI1 **ADC 校准寄存器 1** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称						OE1										
访问						RW										
复位值						0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						OE0										
访问						RW										
复位值						0	0	0	0	0	0	0	0	0	0	0

字段	说明
26：16 OE1	**内部通道偏置误差（Offset Error）** 该域为有符号数，最高位为符号位。
10：0 OE0	**外部通道偏置误差（Offset Error）** 该域为有符号数，最高位为符号位。

© 2013 - 2024 杰发科技有限公司

30.6.16 规则组数据寄存器(ADC_RDRx)

表 30-23 ADC_RDRx(x=0~23)寄存器

ADC_RDRx （x=0~23） ADC 规则组数据寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称										PARITY	CH					
访问										RO	RO					
复位值										0	0					
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RDR															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
22 PARITY	**转换结果的偶校验值** 软件可将 [15:0]的数据寄存器结果进行偶校验计算，结果可与该 bit 值进行比较。若结果相同，说明软件读到的 ADC 数据与硬件提供的值相同。
21:16 CH	**转换结果对应的通道** 该值与对应 ADC_RSQRx 中的 SEQ 的值相同，可供软件快速获取该寄存器 RDR 值对应的通道编号。
15：0 RDR	**规则组数据寄存器** **注意：当 ADC 配置为右对齐时，高 4 位无效；当 ADC 配置为左对齐时，低 4 位无效。具体可参考 30.4.6 数据对齐方式。**

30.6.17 注入组数据寄存器(ADC_IDRx)

表 30-24 ADC_IDRx（x=0 ~ 3)寄存器

ADC_IDRx （x=0~3） ADC 注入组数据寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称										PARITY	CH					
访问										RO	RO					
复位值										0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	IDR															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
22 PARITY	**转换结果的偶校验值** 软件可将 [15:0]的数据寄存器结果进行偶校验计算，结果可与该 bit 值进行比较。若结果相同，说明软件读到的 ADC 数据与硬件提供的值相同。
21:16 CH	**转换结果对应的通道** 该值与对应 ADC_ISQRx 中的 SEQ 的值相同，可供软件快速获取该寄存器 IDR 值对应的通道编号。
15：0 IDR	**注入组数据寄存器** **注意：ADC_IDR 已经减去了在 ADC_IOFR 寄存器中定义的偏移量。当 ADC 配置为右对齐时，高 3 位无效；当 ADC 配置为左对齐时，低 3 位无效。**

30.6.18 ADC analog 配置寄存器 0 (AUXADC_CFG0)

表 30-25 AUXADC_CFG0 寄存器

AUXADC_CFG0 **ADC analog 配置寄存器 0** **复位值：0x440027F0**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称											RG_ADC0_LDO_EN	RG_ADC0_VREFGEN_EN	RG_ADC0_RGMODE_EN			
访问											RW	RW	RW			
复位值											0	0	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		RG_ADC0_EN					RG_ADC0_CHANNEL_SEL									
访问		RW					RW	RW	RW	RW	RW	RW				
复位值		0					1	1	1	1	1	1				

字段	说明
21 RG_ADC0_LDO_EN	**ADC0 RG 模式 LDO 使能** 0: 不使能 LDO 1: 使能 LDO
20	**ADC0 RG 模式 VREFGEN 使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
RG_ADC0_VREFGEN_EN	0: 不使能 VREFGEN 1: 使能 VREFGEN
19 RG_ADC0_RGMODE_EN	**ADC0 RG 模式使能** 0: 不使能 RG 模式 1: 使能 RG 模式
14 RG_ADC0_EN	**ADC0 RG 模式 ADC 使能** 0: 不使能 ADC0 1: 使能 ADC0
9:4 RG_ADC0_CHANNEL_SEL	**ADC0 RG 模式通道选择** 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值为未使用设置。

30.6.19 ADC analog 配置寄存器 1 (AUXADC_CFG1)

表 30-26 AUXADC_CFG1 寄存器

AUXADC_CFG1 **ADC analog 配置寄存器 1** **复位值：0x00000082**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								RG_ADC0_GEOECAL_VIN_EN	RG_ADC0_GEOECAL_VIN_SEL							
访问								RW	RW							
复位值								0	1	0	0					

字段	说明
8 RG_ADC0_GEOECAL_VIN_EN	**GEOE 校准信号使能** 0：关闭 GEOE 校准信号 1：开启 GEOE 校准信号
7:5 RG_ADC0_GEOECAL_VIN_SEL	**GEOE 校准信号选择**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0h: 悬空 1h: 参考电压的 4/16 2h: 参考电压的 5/16 3h: 参考电压的 6/16 4h: 参考电压的 8/16 5h: 参考电压的 10/16 6h: 参考电压的 11/16 7h: 参考电压的 12/16 这里的参考电压信号与 ADC_SQL 中的 REFSEL 配置相对应。

30.6.20 ADC analog 配置寄存器 4 (AUXADC_CFG4)

表 30-27 AUXADC_CFG4 寄存器

AUXADC_CFG4　　ADC analog 配置寄存器 4　　复位值：0x009FC000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RG_ADC1_LDO_EN	RG_ADC1_VREFGEN_EN	RG_ADC1_RGMODE_EN					RG_ADC1_EN					RG_ADC1_CHANNEL_SEL[5:2]			
访问	RW	RW	RW					RW					RW	RW	RW	RW
复位值	0	0	0					0					1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RG_ADC1_CHANNEL_SEL[1:0]															
访问	RW	RW														
复位值	1	1														

字段	说明
31 RG_ADC1_LDO_EN	**ADC1 RG 模式 LDO 使能** 0: 不使能 LDO 1: 使能 LDO
30 RG_ADC1_VREFGEN_EN	**ADC1 RG 模式 VREFGEN 使能** 0: 不使能 VREFGEN 1: 使能 VREFGEN
29 RG_ADC1_RGMODE_EN	**ADC1 RG 模式使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 不使能 RG 模式 1: 使能 RG 模式
24 RG_ADC1_EN	**ADC1 RG 模式 ADC 使能** 0: 不使能 ADC1 1: 使能 ADC1
19:14 RG_ADC1_CHANNEL_SEL	**ADC1 RG 模式通道选择** 0~23：外部输入通道 0~23 33：BG 通道 34：T-Sensor 通道 35：OE GE 校准信号通道 37：内部供电通道 其他值为未使用设置。

30.6.21 ADC analog 配置寄存器 5(AUXADC_CFG5)

表 30-28 AUXADC_CFG5 寄存器

AUXADC_CFG5 **ADC analog 配置寄存器 5** **复位值：0x00000082**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								RG_ADC1_GEOECAL_VIN_EN	RG_ADC1_GEOECAL_VIN_SEL							
访问								RW	RW							
复位值								0	1	0	0					

字段	说明
8 RG_ADC1_GEOECAL_VIN_EN	**GEOE 校准信号使能** 0：关闭 GEOE 校准信号 1：开启 GEOE 校准信号
7:5 RG_ADC1_GEOECAL_VIN_SEL	**GEOE 校准信号选择** 0h: 悬空 1h: 参考电压的 4/16 2h: 参考电压的 5/16

© 2013 - 2024 杰发科技有限公司

字段	说明
	3h: 参考电压的 6/16 4h: 参考电压的 8/16 5h: 参考电压的 10/16 6h: 参考电压的 11/16 7h: 参考电压的 12/16 这里的参考电压信号与 ADC_SQL 中的 REFSEL 配置相对应。

© 2013 - 2024 杰发科技有限公司

31 模拟比较器（ACMP）

31.1 简介

AC7840x 中包含一个模拟比较器模块。该模块可提供一个用于比较两个输入电压大小的电路。其直接输入信号来自内部模拟多路复用器，该模拟多路复用器提供一个用于从 9 路通道中选择模拟输入信号的电路，其中 1 路通道由 8 位数字模拟转换器（DAC）提供，其他通道由外部输入提供。ACMP 的轮询模式和霍尔输出（Hall）功能专为电机应用而设计。

31.2 特性

- 1 个 ACMP 模块
- 2 个比较信号可以来自 8 个外部通道和 1 个内部 DAC 输出信号
- 片上 8 位数字模拟转换器（DAC），可从 VDD 或内部带隙基准电压（Bangap）中选择基准电压
- 支持一个 ACMP 输出引脚用于输出比较结果
- 可配置迟滞，支持 10mV/20mV/40mV 和两种迟滞模式
- 可在比较器输出上升沿、下降沿或上升/下降沿时产生中断.
- 支持窗口控制模式
- 支持低功耗模式（STOP1/STOP2/VLPS）唤醒
- 支持轮询模式
- 支持霍尔(Hall) 输出
- 支持数字滤波器

© 2013 - 2024 杰发科技有限公司

31.3 结构框图

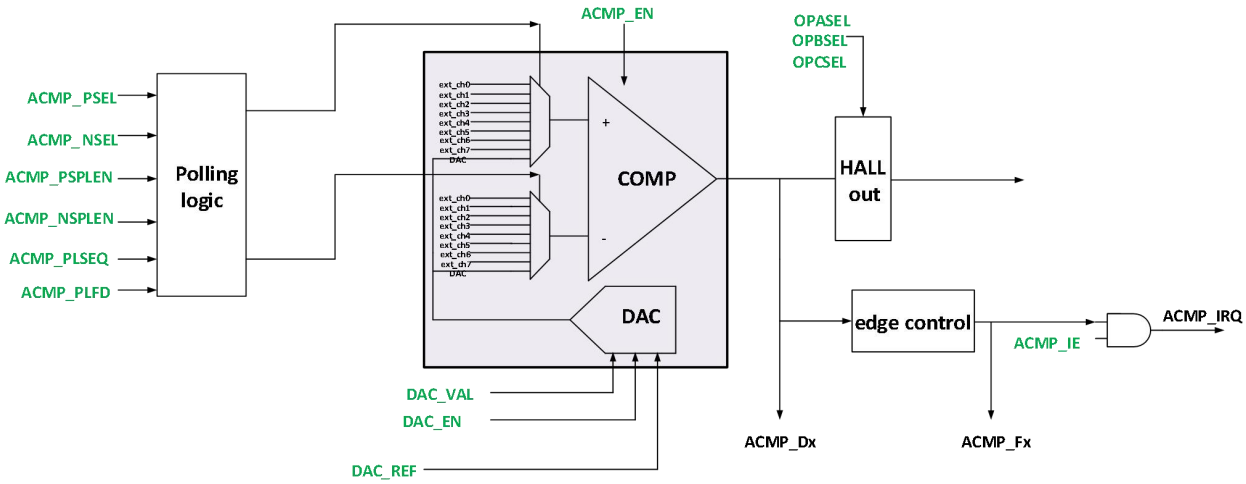


图 31-1 ACMP 结构框图(包含 DAC 和 CMP)

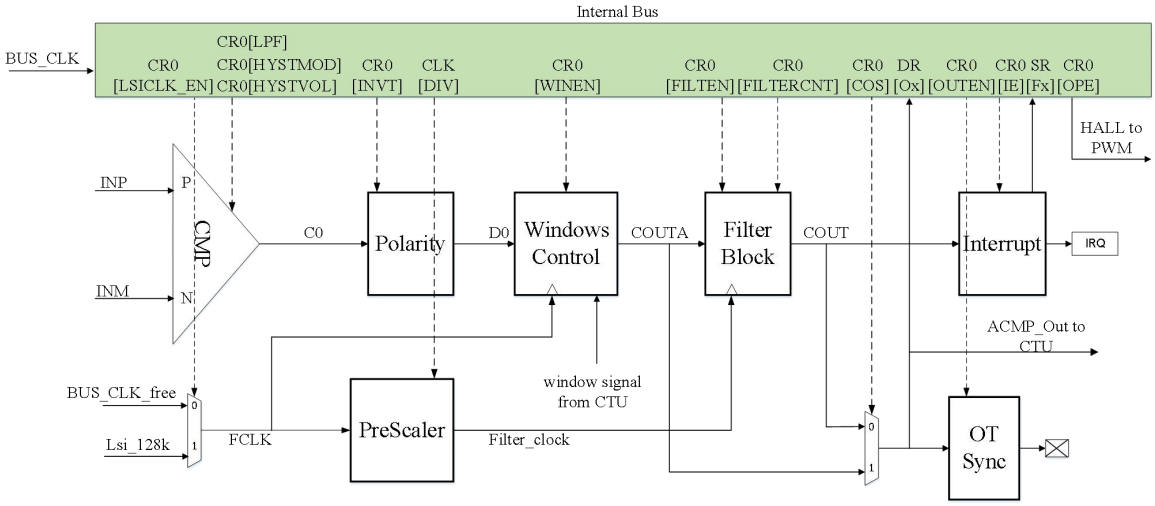


图 31-2 ACMP 内部比较器结构框图

上图中，ACMP 的时钟包含 2 个，一个用于寄存器访问的时钟，使用系统提供的 BUS_CLK，另一个用于内部模块工作的时钟，使用系统提供的和 BUS_CLK 同源的一个时钟，即 BUS_CLK_free。该时钟和 BUS_CLK 不同，在 STOP1 和 VLPS 模式下不会自动关闭。

31.4 功能描述

ACMP 模块就功能而言由两部分组成：数模转换器 (DAC)和比较器 (CMP)。

DAC 包含一个 8bit DAC （数模转换器）和相关的控制逻辑。通过配置 DACREF，DAC 可选择 Vdd 或片上带隙基准源作为 DAC 输入源。在 DAC 使能后，将 DACVAL 中设置的数据转换为步进式模拟输出，馈入输入端进行比较。

© 2013 - 2024 杰发科技有限公司

ACMP 可以实现正输入和负输入的模拟比较，然后提供一个数字输出和相关的中断。在未开启 ACMP_CR0[INVT]反向功能时，如果正输入信号电压大于负输入信号电压，则模拟比较器输出 1，否则输出 0。如果开启 INVT，则输出结果相反。模拟比较器的正负输入均可从 9 个通用输入中选择一个。

31.4.1 普通模式

普通模式下，正输入和负输入固定所选中通道进行比较，比较结果以数字输出呈现。只要输出出现设置的有效边沿， SR 的状态位就会变为有效值。如果 IE 置位，则发生中断。ACMP 输出由总线时钟同步生成比较结果，以便 CPU 能读出比较结果。数据寄存器根据比较结果而改变，因此它可以用做一个跟踪标志，连续指示输入的电压变化。

31.4.2 轮询模式

轮询模式下，可通过相关逻辑动态切换比较器正输入或负输入的输入通道。轮询序列在 ACMP_CR4[PLSEQ]中定义，切换频率由 ACMP_FD[PLFD]控制。PSPLEN 和 NSPLEN 是轮询模式的使能位，PSPLEN 和 NSPLEN 无法同时使能。PSPLEN 和 NSPLEN 都使能不会触发轮询模式。因此，软件必须确保只使能上述两个字段其中之一。

这里提供一个有关轮询模式的示例。正输入轮询，轮询频率为 source_clk/101，对外部通道 1~6 和 DAC 输出作轮询，负输入选择外部通道 0，下降沿触发中断。

其中寄存器 ACMP_Ox 和 ACMP_Fx 分别对应通道 x 的数据输出和状态寄存器标志。ACMP0_RRT 为 ACMP 通道切换信号，该信号在 ACMP 内部切换输入通道时，会输出 1 个 BUS_CLK 周期宽度的脉冲信号，具体引脚可参考 19.4.4 复用功能。

- **步骤 1**：IE = 1b,MOD = 00b；
- **步骤 2**：DACEN = 1b,DACVAL=value；
- **步骤 3**：PSPLEN = 1b, NSPLEN = 0b；
- **步骤 4**：PLFD = 01b, PLSEQ = 101111110b,；
- **步骤 5**：NSEL = 000b
- **步骤 6**：EN = 1b。

© 2013 - 2024 杰发科技有限公司

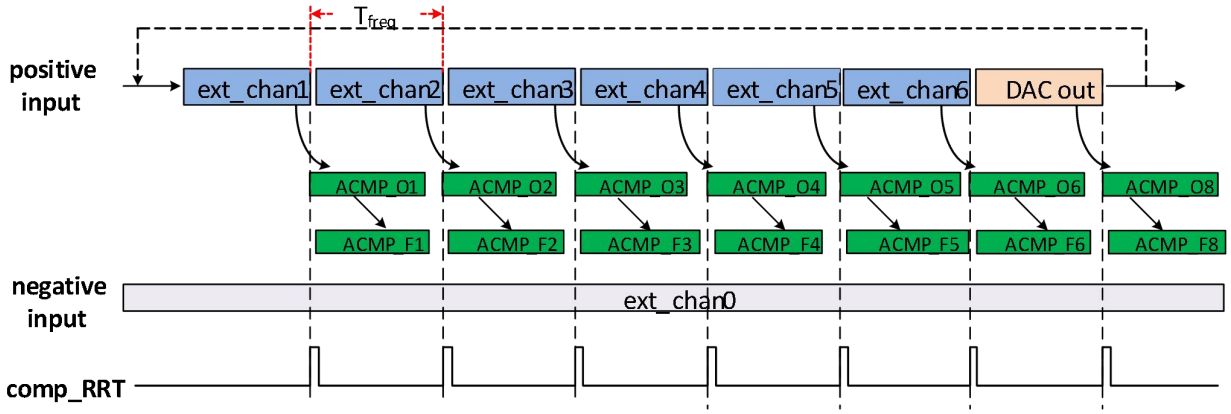


图 31-3 轮询模式工作流程图

31.4.3 霍尔输出

ACMP 支持 3 个内部输出信号 HALL A/B/C 可用于霍尔输出功能。这三个输出信号被连接到内部的 PWM 模块。

该功能需配合轮询模式一起使用。当配置轮询模式后，可分别配置 HALL A/B/C 的输出为被轮询通道的比较结果之一。

31.4.4 迟滞

ACMP 支持两种迟滞模式，分别如图 31-4 和图 31-5 所示。

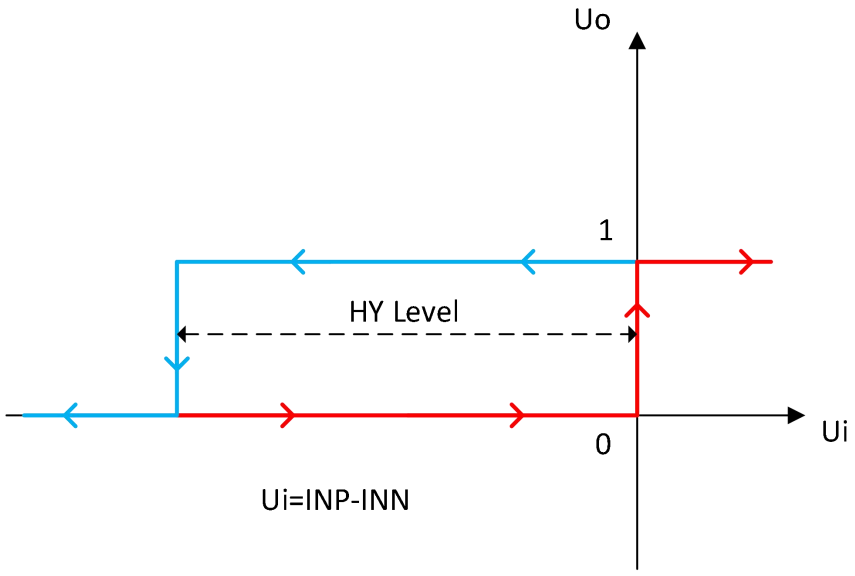


图 31-4 下降沿迟滞模式

© 2013 - 2024 杰发科技有限公司

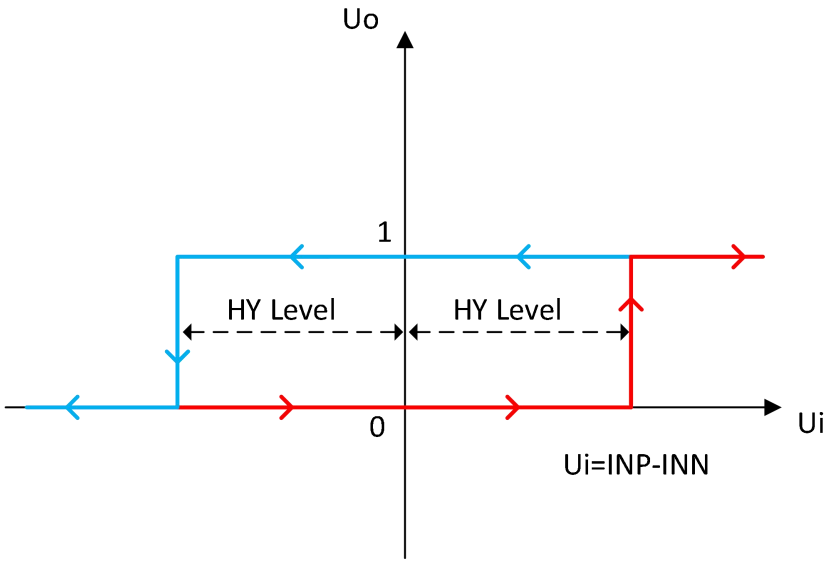


图 31-5 双边沿迟滞模式

其中 INP 和 INN 分别代表正输入和负输入电压，HY Level 代表 ACMP_CR0[HYSTVOL]所选迟滞电压值。

31.4.5 窗口控制模式

使能窗口控制模式后，ACMP 只有在窗口信号为高电平时进行模拟信号比较，当窗口信号为低电平时，ACMP 保持上一次的输出。该模式下的 ACMP 输出如图 31-6 所示。

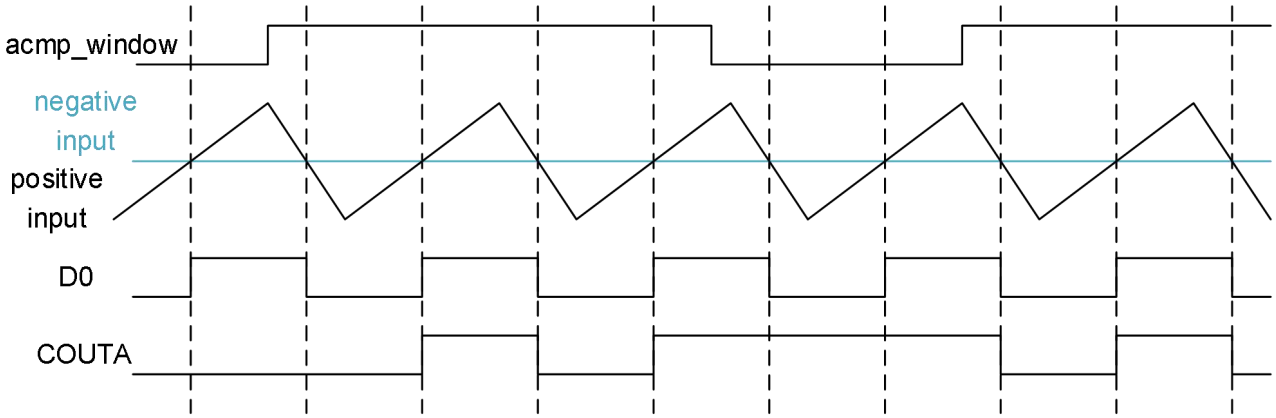


图 31-6 窗口控制模式示意图

图 31-6 中的正输入和负输入的比较结果 D0 是 INVT 为 0 时的结果。D0 为 ACMP 模块的内部信号，可参考图 31-2。窗口模式的输入信号，即 acmp_window，来自 CTU 的 ACMP 触发输入，具体可参考 24.4.1 TRGMUX 触发源存储器映射表 和 24.4.2 TRGMUX 触发目标选择。

在图 31-6 中，只有在虚线代表的时刻才会进行 acmp_window 信号的确认以及 D0、COUTA 结果的更新，在虚线之间，acmp_window 信号的变化不会影响 D0 和 COUTA 的结果。

© 2013 - 2024 杰发科技有限公司

注意，当 ACMP 第一次使能窗口控制模式，且当前窗口信号为低电平时，ACMP 默认输出为 0。

31.4.6 中断请求

ACMP 共有 10 个中断，中断请求可通过寄存器 ACMP_CR0 中的 IE 和 MOD 来控制。在使能中断后，当 ACMP 普通模式或轮询模式检测到某一通道发生了 MOD 对应的事件时，就会触发中断。

表 31-1 ACMP 中断汇总

中断源	标志	本地使能	全局使能
普通模式中断标志	F	IE	无
轮询模式通道 DAC 中断标志	F8	IE	无
轮询模式通道 7 中断标志	F7	IE	无
轮询模式通道 6 中断标志	F6	IE	无
轮询模式通道 5 中断标志	F5	IE	无
轮询模式通道 4 中断标志	F4	IE	无
轮询模式通道 3 中断标志	F3	IE	无
轮询模式通道 2 中断标志	F2	IE	无
轮询模式通道 1 中断标志	F1	IE	无
轮询模式通道 0 中断标志	F0	IE	无

31.4.7 低功耗模式

ACMP 可配置为在系统进入低功耗模式后仍可以工作，包括除寄存器访问外的所有功能。
这里需要注意，如果系统进入 VLPS，且 SYS_CLK 的时钟源被切换为该模式下不工作的时钟源时，为了让 ACMP 继续工作，需要进行以下操作：

1. 将 ACMP 的 function clock 配置为 LSI，可通过 ACMP_CR0 的 LSICLK_EN 位进行设置；
2. 需要将 ACMP 配置为低功耗模式，可通过 ACMPDAC_CFG 的 RG_MCU_ACMPDAC_LP_EN 位进行设置。

注意，ACMP 的工作速度与 function clock 频率有关，在切换 ACMP 的 function clock 的时钟源后，ACMP 轮询模式的切换频率和数字滤波器的滤波周期等也会发生相应的变化。
ACMP 的中断信号也可作为唤醒源，ACMP 唤醒源可参考表 31-2。

表 31-2 ACMP 模块唤醒源

模式	唤醒源
STOP1	10 个中断中任意一个均可唤醒
STOP2	10 个中断中任意一个均可唤醒
VLPS	10 个中断中任意一个均可唤醒

31.4.8 调试模式

调试模式对 ACMP 的工作无影响。

© 2013 - 2024 杰发科技有限公司

31.5 应用说明

31.5.1 轮询模式下霍尔输出

ACMP 有三个 Hall 输出：Hall A Output， Hall B Output 和 Hall C Output， 这些信号连接到芯片内部的 PWM 模块。Hall 输出需与轮询功能配合，当 ACMP 配置为轮询模式时，霍尔输出的三个通道中任意一个通道可以选择当前被轮询的任意一个通道。

使用该功能可获取到无传感电机的 Hall 位置（通过电机反电动势检测电机转子所在位置）。每个 Hall 输出都可以通过轮询模式选择 8 个外部输入通道其中之一。

例如，若轮询模式 PLSEQ = 00001110b，则轮询顺序为：外部输入 1- >外部输入 2 -> 外部输入 3。

设置 ACMP_OPA[OPASEL] = 010b，则 Hall A Output 为 ACMP_DR[O2]。

31.5.2 数字滤波器

在实际应用场景中，由于输入信号带有噪声，可能会造成输出结果产生抖动，这时除了使用迟滞功能进行滤波，还可以使用数字滤波器功能。该功能直接作用于 ACMP 的输出，只有在输出信号保持相同值的时间超过滤波器配置值，才会输出相应的结果。该过程如图 31- 7 所示，图中信号标志可参考图 31- 2。

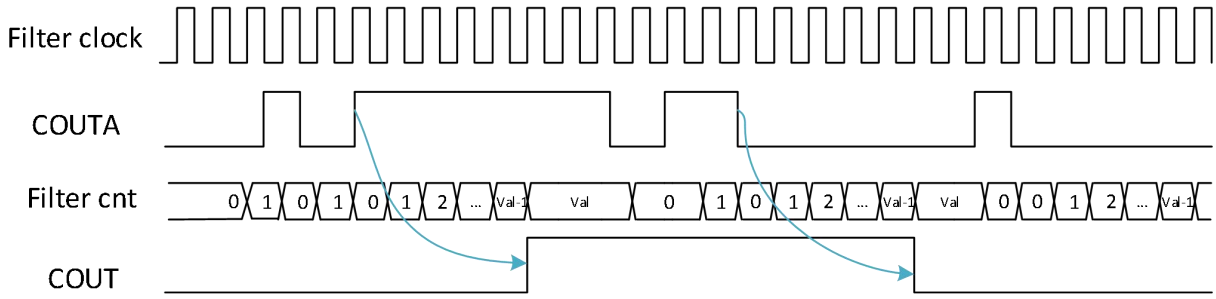


图 31- 7 ACMP 数字滤波器

31.6 寄存器定义

表 31- 3 ACMP 寄存器映射

ACMP0 基地址：0x40005000

ANALOG 基地址：0x40008800

地址	名称	宽度	描述
ACMPx 基地址+0x0	ACMP_CR0	32	配置寄存器 0
ACMPx 基地址+0x4	ACMP_CR1	32	配置寄存器 1
ACMPx 基地址+0x8	ACMP_CR2	32	配置寄存器 2
ACMPx 基地址+0xC	ACMP_CR3	32	配置寄存器 3

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
ACMPx 基地址+0x10	ACMP_CR4	32	配置寄存器 4
ACMPx 基地址+0x14	ACMP_DR	32	数据输出寄存器
ACMPx 基地址+0x18	ACMP_SR	32	状态寄存器
ACMPx 基地址+0x1C	ACMP_FD	32	轮询分频器寄存器
ACMPx 基地址+0x20	ACMP_OPA	32	Hall A 输出设置寄存器
ACMPx 基地址+0x24	ACMP_OPB	32	Hall B 输出设置寄存器
ACMPx 基地址+0x28	ACMP_OPC	32	Hall C 输出 设置寄存器
ACMPx 基地址+0x48	ACMP_CLK	32	数字滤波器时钟配置寄存器
ANA 基地址+0x50	ACMPDAC_CFG	32	ACMP analog 配置寄存器

31.6.1 配置寄存器 0(ACMP_CR0)

表 31-4 ACMP_CR0 寄存器

ACMP_CR0 **ACMP 配置寄存器 0** **复位值：0x00000040**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					COS	FILTERCNT								INVT	WINEN	FILTEN
访问					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值					0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				LSICLK_EN	LPF		EN	HYSTMOD	HYSTVOL			IE	OUTEN	OPE	MOD	
访问				RW	RW		RW	RW	RW			RW	RW	RW	RW	RW
复位值				0	0	0	0	0	0	1	0	0	0	0	0	0

字段	说明
COS 27	**ACMP 输出信号选择** 0：使 ACMP 的输出等于 COUT(经过数字滤波器) 1：使 ACMP 的输出等于 COUTA(未经过数字滤波器)
26:19 FILTERCNT	**滤波器值** **0~255: 1~256 个 ACMP 功能时钟周期**
18 INVT	**输出结果反转** 0： 禁用输出结果反转 1： 使能输出结果反转 具体反转的位置参考图 31-2
17 WINEN	**窗口控制模式使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 禁用 1： 使能
16 FLTEN	**数字滤波使能** 0： 禁用 1： 使能
12 LSICLK_EN	**ACMP 功能时钟选择** 0: 使用 BUS_CLK_free 1: 使用 LSI_CLK
11:10 LPF	**低通滤波器配置** 0h: 500KHz 1h: 1MHz 2h: 2MHz 3h: 跳过低通滤波器 该滤波器直接作用于外部输入模拟信号，与数字滤波器相独立。
9 EN	**ACMP 使能** 0： 禁用 1： 使能
8 HYSTMOD	**迟滞模式** 0: 下降沿模式 1: 双边沿模式
7:6 HYSTVOL	**迟滞等级** 00b: 关闭迟滞功能 01b: 10mV 10b: 20mV 11b: 40mV
4 IE	**中断使能** 0： 禁用 1： 使能
3 OUTEN	**比较结果输出到外部 PIN** 0： 禁用 1： 使能
2 OPE	**Hall 输出使能** 0： 禁用 1： 使能
1：0 MOD	**中断触发模式**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0h： 输出下降沿中断 1b： 输出上升沿中断 2b： 输出下降沿中断（**说明：0h 和 2h 是相同的配置**） 3b： 输出下降沿或上升沿中断

31.6.2 配置寄存器 1(ACMP_CR1)

表 31-5 ACMP_CR1 寄存器

ACMP_CR1 **ACMP 配置寄存器 1** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									PSEL				NSEL			
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7：4 PSEL	**正输入选择** 0h：外部输入 0 1h：外部输入 1 2h：外部输入 2 3h：外部输入 3 4h：外部输入 4 5h：外部输入 5 6h：外部输入 6 7h：外部输入 7 8h：DAC 输出 其他值未使用
3：0 NSEL	**负输入选择** 0h：外部输入 0 1h：外部输入 1 2h：外部输入 2 3h：外部输入 3 4h：外部输入 4 5h：外部输入 5 6h：外部输入 6 7h：外部输入 7 8h：DAC 输出

© 2013 - 2024 杰发科技有限公司

字段	说明
	其他值未使用

31.6.3 配置寄存器 2(ACMP_CR2)

表 31-6 ACMP_CR2 寄存器

ACMP_CR2 ACMP 配置寄存器 2 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							DACREF	DACEN	DACVAL							
访问							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值							0	0	0	0	0	0	0	0	0	0

字段	说明
9 DACREF	**DAC 参考电压选择** 0：使用 BG 作为 DAC 参考电压 1：使用 VDD 作为 DAC 参考电压
8 DACEN	**DAC 使能** 0： 禁用 DAC 功能 1： 使能 DAC 功能
7：0 DACVAL	**DAC 输出电压值** DAC 输出电压 DACO = $(V_{ref}/256) \times DACVAL$ DACVAL 取值范围：0~255 DAC 输出电压范围：0~$(V_{ref}/256) \times 255$ 其中V_{ref}与 DACREF 的选择有关

© 2013 - 2024 杰发科技有限公司

31.6.4 配置寄存器 3(ACMP_CR3)

表 31-7 ACMP_CR3 寄存器

ACMP_CR3 **ACMP 配置寄存器 3** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									PSPLEN				NSPLEN			
访问									RW				RW			
复位值									0				0			

字段	说明
7 PSPLEN	**使能正输入为轮询模式** 0： 禁用正输入轮询功能 1： 使能正输入轮询功能 **注意：**PSPLEN 和 NSPLEN 无法同时使能，PSPLEN 和 NSPLEN 都使能不会触发轮询模式。
3 NSPLEN	**使能负输入为轮询模式** 0： 禁用负输入轮询功能 1： 使能负输入轮询功能 **注意：**PSPLEN 和 NSPLEN 无法同时使能，PSPLEN 和 NSPLEN 都使能不会触发轮询模式。

© 2013 - 2024 杰发科技有限公司

31.6.5 配置寄存器 4(ACMP_CR4)

表 31-8 ACMP_CR4 寄存器

ACMP_CR4 ACMP 配置寄存器 4 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								PLSEQ								
访问								RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8：0 PLSEQ	**轮询通道序列设置** 0： 禁用相应的通道 1： 使能相应的通道 其中各个 bit 对应通道与 ACMP_DR 相同

31.6.6 数据输出寄存器(ACMP_DR)

表 31-9 ACMP_DR 寄存器

ACMP_DR ACMP 数据寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							O	O8	O7	O6	O5	O4	O3	O2	O1	O0
访问							RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值							0	0	0	0	0	0	0	0	0	0

字段	说明
9 O	**正常模式输出**
8 O8	**DAC 通道输出**
7 O7	**轮询模式通道 7 输出**
6 O6	**轮询模式通道 6 输出**

© 2013 - 2024 杰发科技有限公司

字段	说明
5 O5	**轮询模式通道 5 输出**
4 O4	**轮询模式通道 4 输出**
3 O3	**轮询模式通道 3 输出**
2 O2	**轮询模式通道 2 输出**
1 O1	**轮询模式通道 1 输出**
0 O0	**轮询模式通道 0 输出**

31.6.7 状态寄存器(ACMP_SR)

表 31-10 ACMP_SR 寄存器

ACMP_SR **ACMP 状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							F	F8	F7	F6	F5	F4	F3	F2	F1	F0
访问							R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
复位值							0	0	0	0	0	0	0	0	0	0

字段	说明
9 F	**正常模式中断标志** 写 1 清除该标志
8 F8	**轮询模式 DAC 中断标志** 写 1 清除该标志
7 F7	**轮询模式通道 7 中断标志** 写 1 清除该标志
6 F6	**轮询模式通道 6 中断标志** 写 1 清除该标志
5 F5	**轮询模式通道 5 中断标志** 写 1 清除该标志

© 2013 - 2024 杰发科技有限公司

字段	说明
4 F4	**轮询模式通道 4 中断标志** 写 1 清除该标志
3 F3	**轮询模式通道 3 中断标志** 写 1 清除该标志
2 F2	**轮询模式通道 2 中断标志** 写 1 清除该标志
1 F1	**轮询模式通道 1 中断标志** 写 1 清除该标志
0 F0	**轮询模式通道 0 中断标志** 写 1 清除该标志

31.6.8 轮询分频器寄存器(ACMP_FD)

表 31-11 ACMP_FD 寄存器

ACMP_FD **ACMP 轮询分频寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															PLFD	
访问															RW	RW
复位值															0	0

字段	说明
1：0 PLFD	**轮询模式分频器** 此分频器控制轮询通道的切换频率 0h： source_clk/256 1h： source_clk/101 2h： source_clk/71 3h： source_clk/51 source_clk 为图 31-2 中的 FCLK。

31.6.9 霍尔输出 A 设置寄存器(ACMP_OPA)

表 31-12 ACMP_OPA 寄存器

© 2013 - 2024 杰发科技有限公司

ACMP_OPA **ACMP 霍尔输出 A 设置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													OPASEL			
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3：0 OPASEL	**Hall A 输出 设置** 0h：轮询通道 0 1h：轮询通道 1 2h：轮询通道 2 3h：轮询通道 3 4h：轮询通道 4 5h：轮询通道 5 6h：轮询通道 6 7h：轮询通道 7 8h：轮询 DAC

31.6.10 霍尔输出 B 设置寄存器(ACMP_OPB)

表 31-13 ACMP_OPB 寄存器

ACMP_OPB **ACMP 霍尔输出 B 设置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													OPBSEL			
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3：0 OPBSEL	**Hall B 输出设置** 0h：轮询通道 0 1h：轮询通道 1 2h：轮询通道 2 3h：轮询通道 3

© 2013 - 2024 杰发科技有限公司

字段	说明
	4h：轮询通道 4 5h：轮询通道 5 6h：轮询通道 6 7h：轮询通道 7 8h：轮询 DAC

31.6.11 霍尔输出 C 设置寄存器(ACMP_OPC)

表 31-14 ACMP_OPC 寄存器

ACMP_OPC　　ACMP 霍尔输出 C 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													OPCSEL			
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3：0 OPCSEL	**Hall C 输出设置** 0h：轮询通道 0 1h：轮询通道 1 2h：轮询通道 2 3h：轮询通道 3 4h：轮询通道 4 5h：轮询通道 5 6h：轮询通道 6 7h：轮询通道 7 8h：轮询 DAC

© 2013 - 2024 杰发科技有限公司

31.6.12 数字滤波器时钟配置寄存器(ACMP_CLK)

表 31-15 ACMP_CLK 寄存器

ACMP_CLK　　ACMP 数字滤波器时钟配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														DIV		
访问														RW	RW	RW
复位值														0	0	0

字段	说明
2：0 DIV	**数字滤波器时钟分频选择** 0~7: 1~8 倍分频 被分频时钟的时钟源参考图 31-2 中的 FCLK

31.6.13 ACMP analog 配置寄存器(ACMPDAC_CFG)

表 31-16 ACMPDAC_CFG 寄存器

ACMPDAC_CFG　　ACMP analog 配置寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																RG_MCU_ACMPDAC_LP_EN
访问																RW
复位值																0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																
访问																
复位值																

字段	说明
16 RG_MCU_ACMPDA	**ACMPDAC 低功耗模式使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
C_LP_EN	0: 关闭低功耗模式 1: 开启低功耗模式

© 2013 - 2024 杰发科技有限公司

32 脉宽调制（PWM）

32.1 简介

PWM(Pulse Width Modulation)模块是一个多功能的定时器，支持输入捕获、输出比较、正交解码和PWM 信号生成。PWM 的计数功能是通过一个 16 位的计数器产生的。每个 PWM 模块支持 8 通道。

32.2 特性

PWM 特性包括：

- PWM 时钟源可选。
- 时钟源可以选择系统时钟，固定频率时钟或者外部时钟。
- 可选择外部时钟将 PWM 时钟连到芯片输入引脚，允许 PWM 计数器与外部时钟源同步。
- 16 位预分频器支持 1 至 65536 分频。
- 16 位计数器
  - 它可以为一个自由运行、没有限制的计数器，或一个有初值和终值的计数器。
  - 支持向上、向上-向下两种计数方式。
- 每个通道都可以配置为输入捕获、输出比较或边沿对齐 PWM 模式、中心对齐 PWM 模式。
- 在输入捕获模式下，捕获可以发生在上升沿、下降沿或上升沿/下降沿。
- 输入捕获模式下，可以为某些通道选择输入滤波器。
- 支持 Hall 输入信号测量。
- 在输出比较模式下， 可以在匹配时设置、清除或者翻转输出。
- 每对通道都可以组合起来生成一个 PWM 信号，并且能够独立控制 PWM 信号的上升沿和下降沿。
- PWM 通道可以采用具有同等输出或者互补输出的成对工作方式。
- 死区插入可用于每一对组合通道。
- 可生成匹配触发。
- 软件控制 PWM 输出。
- 输出屏蔽可设置通道为无效状态。
- 对于故障控制最多有 4 个故障输入。
- 每个通道的极性可配置。
- 每个通道都可以产生中断。
- 计数器上溢或下溢时可产生中断。

© 2013 - 2024 杰发科技有限公司

- 当检测到故障条件时，产生中断。
- 同步加载写缓冲 PWM 寄存器。
- 关键寄存器支持写保护。
- 用于脉冲和周期宽度测量的双边沿捕获。
- 支持正交解码（ABZ 相输入引脚映射到每个 PWM 模块的 CH0、CH1 和 CH2）。
- 支持全局时基。
- 支持 PWM 输出波形相位偏移。
- 支持 DMA 传输。
- 支持 Dither 功能。

© 2013 - 2024 杰发科技有限公司

32.3 结构框图

PWM 每个通道使用一个输入/输出（I / O）引脚、CHn(PWM 通道(n))，其中 n 是通道编号（0-7）。图 32- 1 为 PWM 结构图。PWM 的核心部分为 16 位计数器，具有可编程的初始值和最终值，其计数可以是向上或向上-向下。

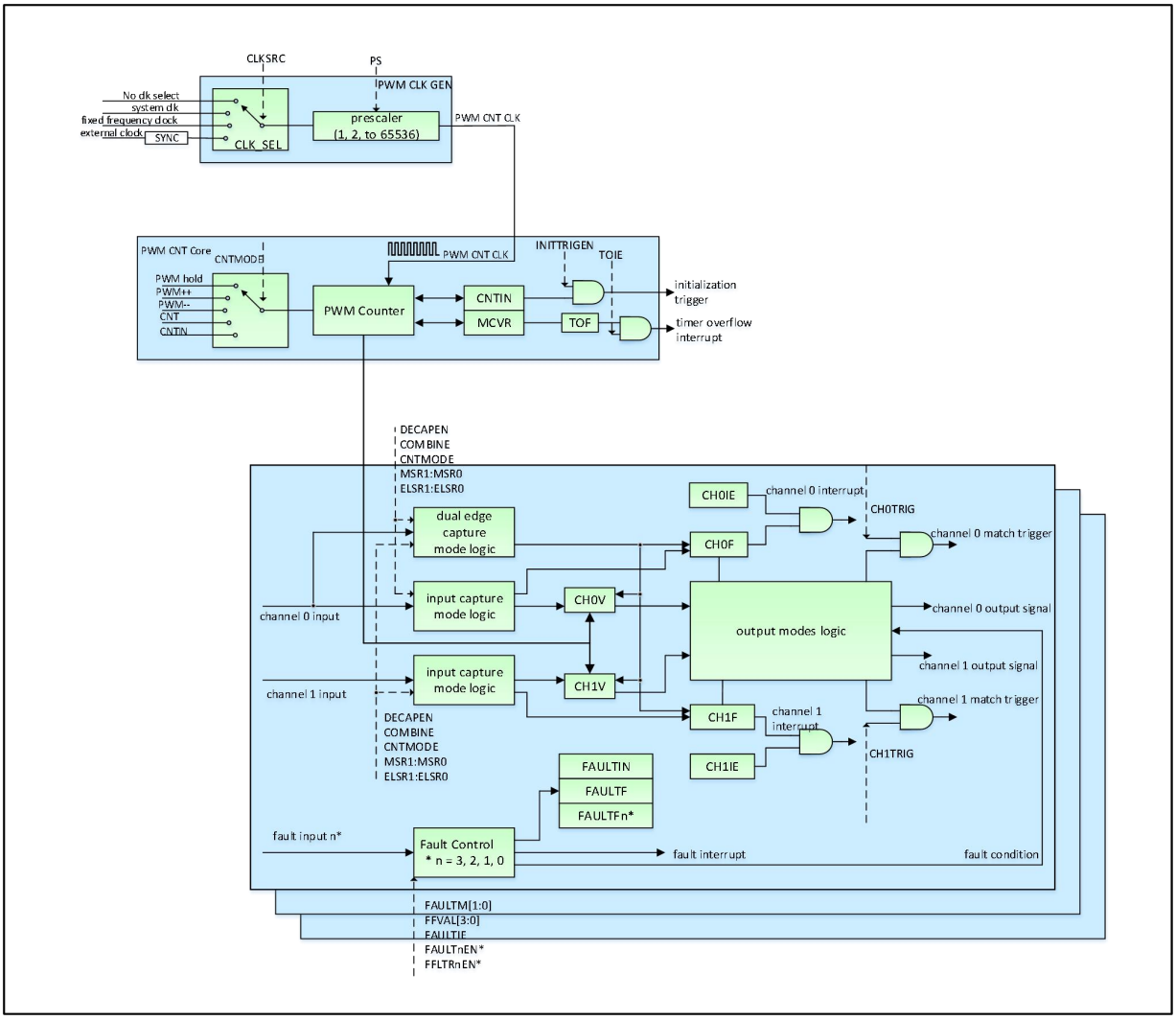


图 32- 1 PWM 结构框图

© 2013 - 2024 杰发科技有限公司

32.4 功能描述

32.4.1 时钟源

图 32-2 为 PWM 时钟源框图。

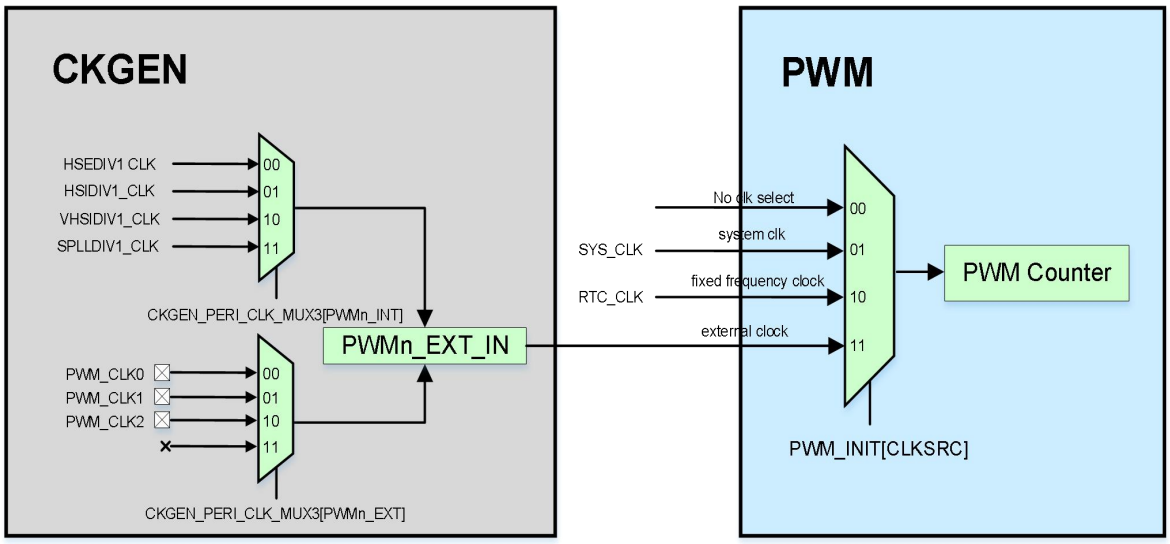


图 32-2 PWM 时钟框图

INIT[CLKSRC]位选择 PWM 计数器的三个可能时钟源之一或禁用 PWM 计数器。MCU 复位后，INIT[CLKSRC] = 0:0，因此没有选择时钟源。

在写保护功能未使能时，INIT[CLKSRC]时钟源选择位在任意时刻均可读写。通过将 INIT[CLKSRC]位写入 0:0，禁用 PWM 计数器不会影响 PWM 计数器值或其他寄存器；将 INIT[CLKSRC]位写入 0:1，将使用系统时钟 SYS_CLK 作为 PWM 计数器的时钟源；将 INIT[CLKSRC]位写入 1:0，将使用固定频率时钟 RTC_CLK 作为 PWM 计数器的时钟源，该固定频率时钟可通过 CKGEN_LP_CLK_[RTC_CLK_MUX]选择相应的时钟源。将 INIT[CLKSRC]位写入 1:1，将使用外部时钟作为 PWM 计数器的时钟源。固定频率时钟/外部时钟由系统时钟进行同步，来确保计数器转换与系统时钟对齐。为了满足奈奎斯特采样定理，固定频率时钟/外部时钟源的频率不得超过系统时钟频率的 1/4。

32.4.2 计数器

PWM 包含一个 16 位计数器，用于通道输入或输出模式。PWM 计数器时钟是由预分频器分频的选定时钟。PWM 计数器具有以下工作模式：

- 向上计数
- 向上-向下计数

注意: 计时器配置过程中，计数的终值(MCVR)需大于计数的起始值(CNTIN)。

© 2013 - 2024 杰发科技有限公司

32.4.2.1 向上计数

当 QDI[QDIEN] = 0 且 INIT[CNTMODE] = 0 时，为向上计数模式。CNTIN 定义计数的起始值，MCVR 定义计数的终值，如图 32-3 所示。CNTIN 的值加载到 PWM 计数器中，计数器递增，直到达到 MCVR 的值，此时计数器重新加载 CNTIN 的值。向上计数模式 PWM 周期为(MCVR - CNTIN + 0x0001) × PWM 计数器时钟周期。当 PWM 计数器从 MCVR 变为 CNTIN 时，CNTOF 位置 1。

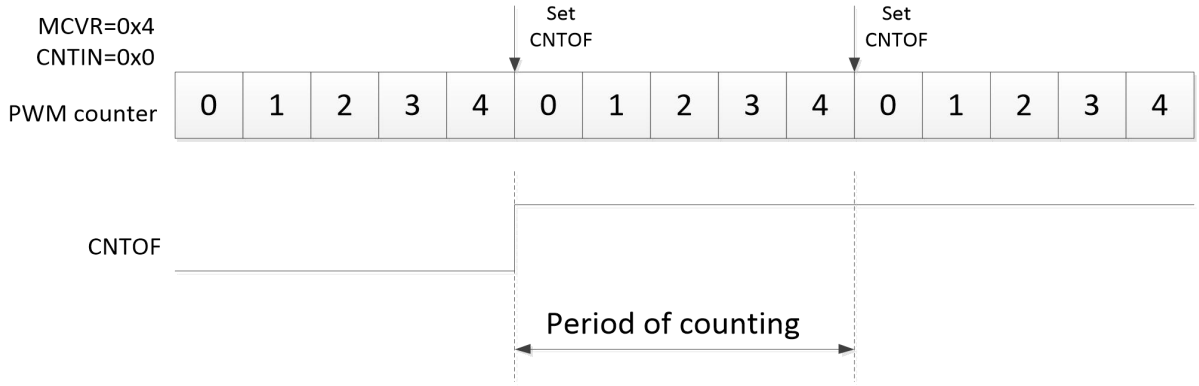


图 32-3 向上计数

32.4.2.2 向上-向下计数

当 QDI [QDIEN] = 0 且 INIT[CNTMODE] = 1 时，选择向上-向下计数。CNTIN 定义计数的起始值，MCVR 定义计数的终值。CNTIN 的值被加载至 PWM 计数器中，并且计数器递增直到达到 MCVR 的值，此时计数器递减，直到它返回到 CNTIN 的值，然后重新开始上下计数。

向上-向下计数时的 PWM 周期为 2 × (MCVR – CNTIN) × PWM 计数器时钟周期。当 PWM 计数器从 MCVR 变为 MCVR - 1 时，CNTOF 位置 1，如图 32-4 所示

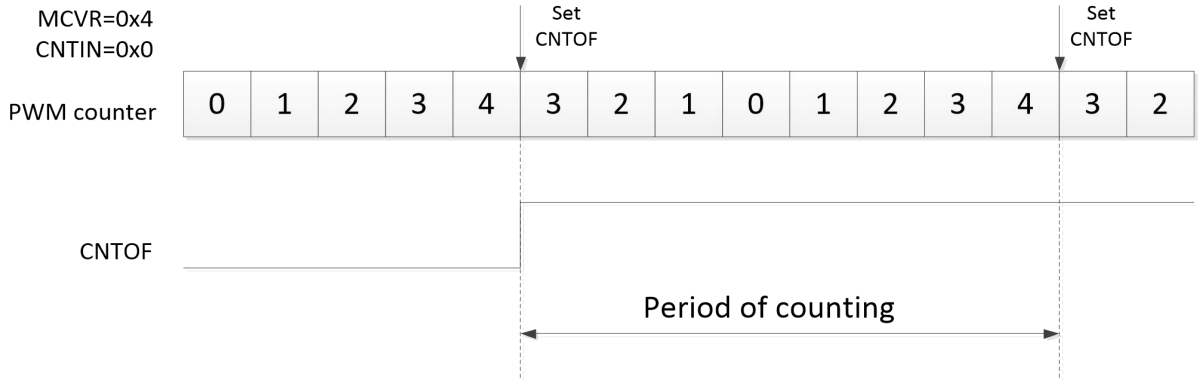


图 32-4 向上-向下计数

32.4.3 工作模式

© 2013 - 2024 杰发科技有限公司

PWM 可配置为输入捕获、输出比较或边沿对齐 PWM 模式、中心对齐 PWM 模式、组合模式、正交解码模式，详细配置参考图 32-5。

QDIEN	DECAPEN	HALLEN	COMBINE	CNTMODE	MSR1：MSR0		ELSR1：ELSR0		工作模式	配置
0	0		0	0	0	0	0	1	输入捕获	上升沿捕获
0	0		0	0	0	0	1	0		下降沿捕获
0	0		0	0	0	0	1	1		上升或下降沿捕获
0	0		0	0	0	1	0	1	输出比较	匹配时切换输出
0	0		0	0	0	1	1	0		匹配时清除输出
0	0		0	0	0	1	1	1		匹配时设置输出
0	0		0	0	1	X	1	0	边沿对齐 PWM	High-true 脉冲(匹配时清除输出)
0	0		0	0	1	X	X	1		Low-true 脉冲(匹配时设置输出)
0	0		0	1	X	X	1	0	中心对齐 PWM	High-true 脉冲(匹配时清除输出)
0	0		0	1	X	X	X	1		Low-true 脉冲(匹配时设置输出)
0	0		1	0	X	X	1	0	向上计数组合模式	Channel(n) 匹配时设置输出 Channel(n+1) 匹配时清除输出
0	0		1	0	X	X	X	1		Channel (n)匹配时清除输出，Channel (n+1) 匹配时设置输出
0	0		1	1	X	X	1	0	向上-向下计数组合模式	Channel(n) 匹配时设置输出 Channel(n+1) 匹配时清除输出
0	0		1	1	X	X	X	1		Channel (n)匹配时清除输出，Channel (n+1) 匹配时设置输出
0	1	0	0	0	X	0	0	1	双边沿单次捕获	上升沿
0	1	0	0	0	X	0	1	0		下降沿
0	1	0	0	0	X	0	1	1		上升沿或下降沿
0	1	0	0	0	X	1	0	1	双边沿持续捕获	上升沿
0	1	0	0	0	X	1	1	0		下降沿
0	1	0	0	0	X	1	1	1		上升沿或下降沿
0	1	1	0	0	X	1	0	1	Hall 功能	作为 Hall 功能时：Channel 0~2 作为 Hall A/B/C 的输入，Channel 3~7 禁止使用。
0	1	1	0	0	X	1	1	0		
0	1	1	0	0	X	1	1	1		
1	X		X	X	X	X	X	X	正交解码	QDIEN=1使能正交解码，正交解码器模式优先于其他模式

图 32-5 工作模式配置

32.4.4 输入捕获模式

当通道输入出现选定的边沿时，PWM 计数器的当前值会被捕获到 CHnV 寄存器中。同时，CHnSCR[CHIF]位置 1。如果 CHnSCR[CHIE] = 1，则产生通道中断。当通道配置为输入捕获时，PWMx_CHn 引脚为边沿敏感输入。ELSnR1：ELSnR0 控制位决定何种边沿（下降沿或上升沿）触发输入捕获事件。注意，可以被正确检测到的输入信号最大频率为 1/4 总线时钟，这是满足信号采样的奈奎斯特采样定理所必需的条件。在输入捕获模式下，忽略对 CHnV 寄存器的写入操作。

如图 32-6 所示，通道配置检测上升沿捕获。

© 2013 - 2024 杰发科技有限公司

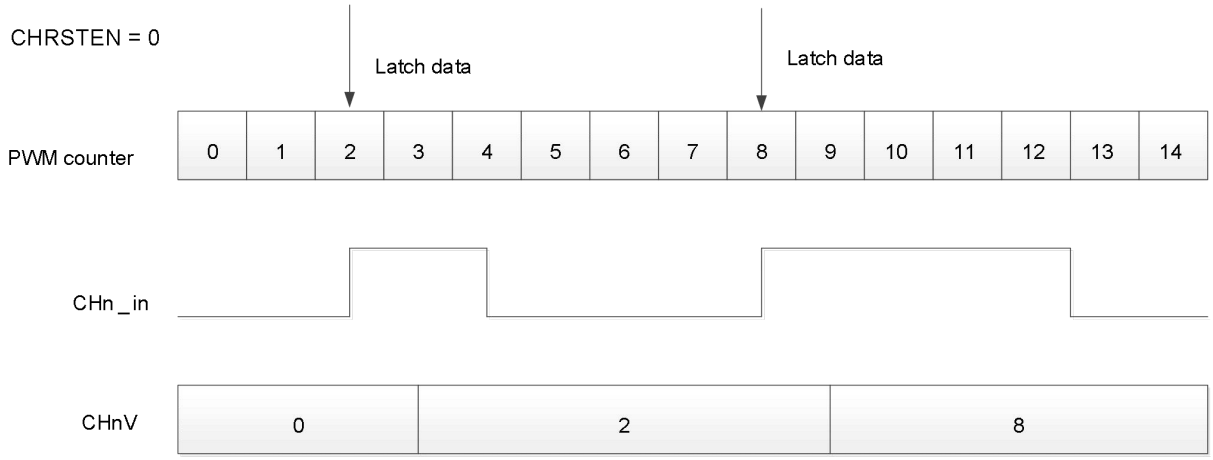


图 32- 6 输入捕获模式(CHRSTEN = 0)

如图 32- 7 所示，在输入捕获模式下若使能 CHnSCR[CHRSTEN]位，当检测到通道(n)选择的边沿时，PWM 计数器的的值会被复位为 CNTIN 的值重新开始计数。

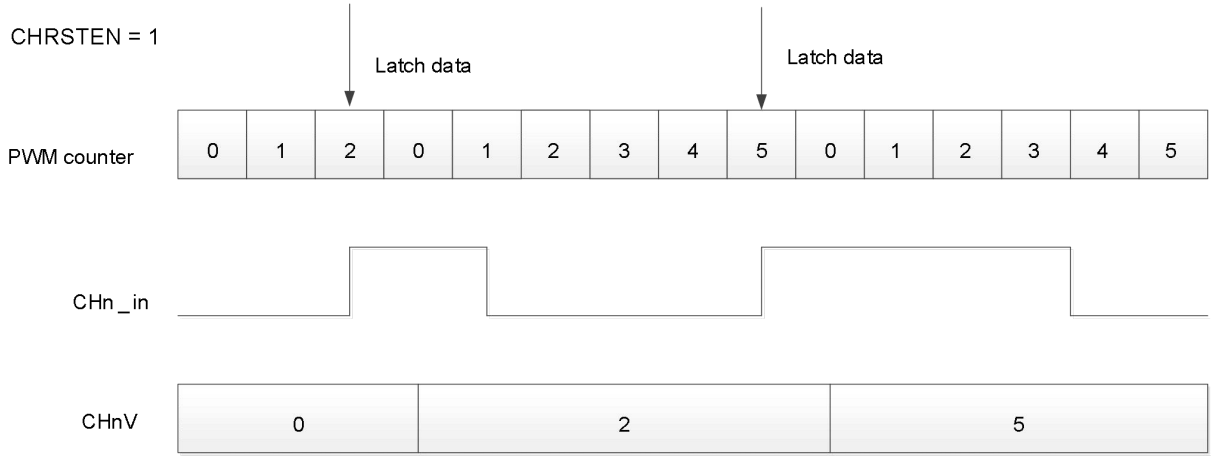


图 32- 7 输入捕获模式(CHRSTEN = 1)

对于 PWM CH0~CH3，存在额外的捕获滤波器来过滤输入信号。滤波器为 4 位计数器，可通过寄存器 CAPFILTER[CHnCAPFVAL]（n = 0,1,2,3）进行配置，滤波器分频为 CAPFILTER [CAPFPSC]，当 CAPFILTER [CHnCAPFVAL] = 0 时，滤波器功能被禁用，如果 CAPFILTER [CHnCAPFVAL] ≠ 00000，输入信号将被延迟(CHnCAPFVAL[3：0] × CAPFPSC[3：0])个总线时钟，然后才传输到边沿检测器。

32.4.5 输出比较模式

在输出比较模式下，PWM 可以生成具有可编程位置、极性、持续时间和频率的定时脉冲。当计数器与输出比较通道的 CHnV 值匹配时，可以设置、清除或翻转通道 n 输出。当通道最初配置为翻转

© 2013 - 2024 杰发科技有限公司

（Toggle）模式时，通道保持输出先前的值，直到发生第一个输出比较事件。在通道 n 匹配时（PWM 计数器 = CHnV）CHnSCR[CHIF]位置 1，如果此时 CHnSCR [CHIE] = 1，则产生通道 n 中断。

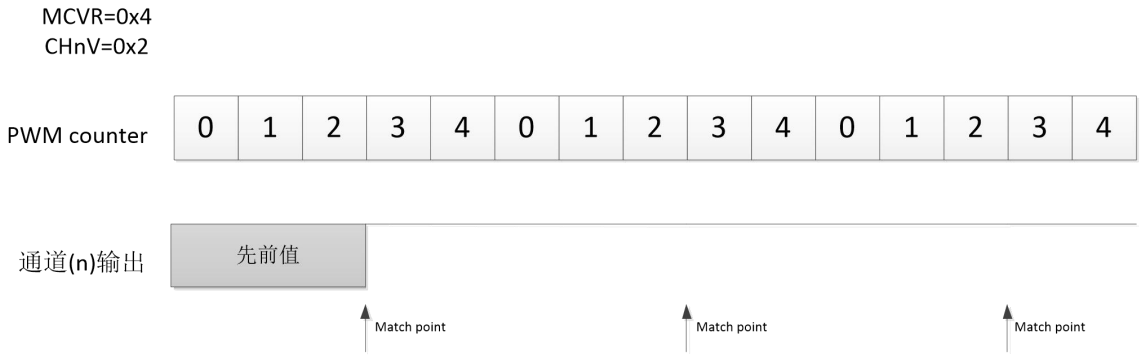


图 32-8 匹配设置输出比较模式

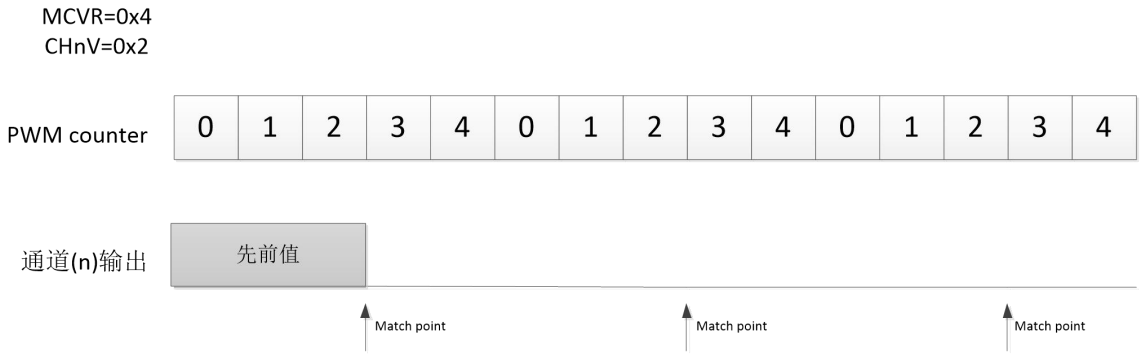


图 32-9 匹配清除输出比较模式

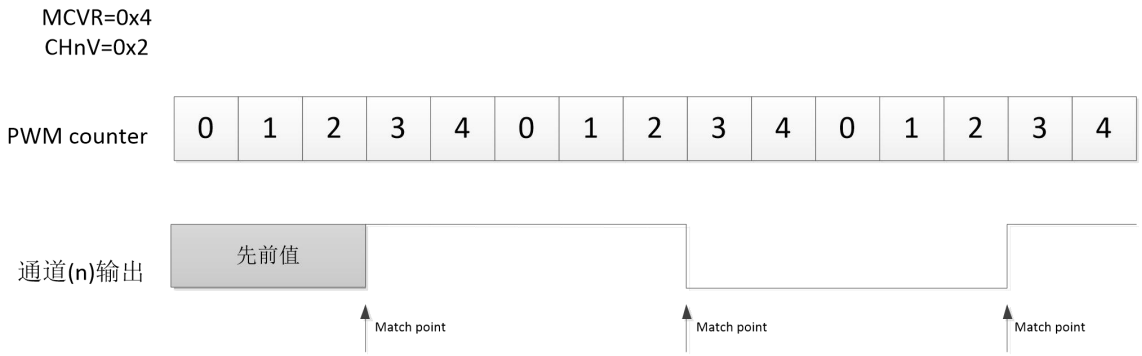


图 32-10 匹配翻转输出比较模式

32.4.6 边沿对齐 PWM 模式(EPWM)

周期 = (MCVR - CNTIN + 0x0001) × PWM 计数器时钟周期

脉冲宽度(占空比) = (CHnV + 0x0001 – CNTIN) × PWM 计数器时钟周期

© 2013 - 2024 杰发科技有限公司

ELSnR1：ELSnR0 = 1：0，通道(n)输出在 CNTIN 值加载到 PWM 计数器时为高电平，在通道(n)匹配（PWM 计数器 = CHnV）时为低电平。

ELSnR1：ELSnR0 = X：1，通道(n)输出在 CNTIN 值加载到 PWM 计数器时为低电平，在通道(n)匹配（PWM 计数器 = CHnV）时为高电平。

CHnSCR [CHIE] = 1，通道(n)匹配（PWM 计数器 = CHnV）时，CHnSCR [CHIF]位置 1 且产生通道(n)中断。

这种类型的 PWM 信号称为边沿对齐，是因为所有 PWM 信号的前沿与周期的开端对齐，这对于 PWM 内的所有通道都是相同的。

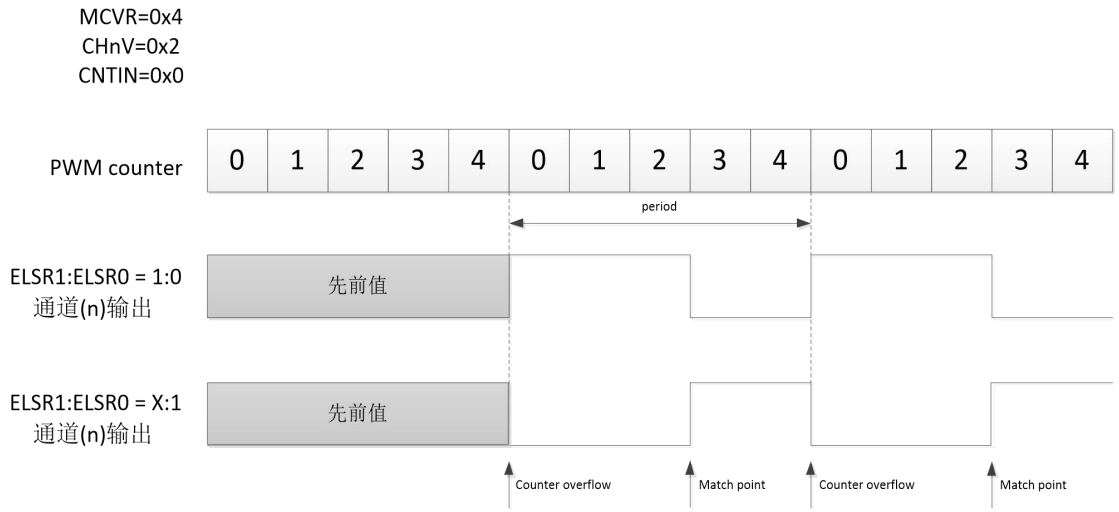


图 32-11 EPWM 波形

ELSnR1：ELSnR0 = 1：0，如果(CHnV = 0x0000)，则通道(n)输出为 0%占空比 EPWM 信号；如果(CHnV >= MCVR)，则通道(n)输出为 100%占空比 EPWM 信号，ELSnR1：ELSnR0 = X：1 则反之。

32.4.7 中心对齐 PWM 模式(CPWM)

周期 = 2 × (MCVR - CNTIN) ×PWM 计数器时钟周期

脉冲宽度(占空比) = 2 × (CHnV - CNTIN) × PWM 计数器时钟周期

在 CPWM 模式下，PWM 计数器向上计数直到达到 MCVR，然后向下计数直至达到 CNTIN。该模式下 CHnV 必须保持在 0x0001 到 0x7FFF 的范围内，超出此范围在 high-true 模式下通道(n)会输出低电平，low-true 模式下通道(n)会输出高电平。

ELSnR1：ELSnR0 = 1：0，通道(n)输出在向下计数时与通道(n)匹配（PWM 计数器 = CHnV）下为高电平，在向上计数时与通道(n)匹配（PWM 计数器 = CHnV）的情况下为低电平。

ELSnR1：ELSnR0 = X：1，通道(n)输出在向下计数时与通道(n)匹配（PWM 计数器 = CHnV）时为低电平，在向上计数时与通道(n)匹配（PWM 计数器 = CHnV）的情况下为高电平。

© 2013 - 2024 杰发科技有限公司

CHnSCR [CHIE] = 1，当 PWM 计数减少或 PWM 计数增加，在通道(n)匹配（PWM 计数器 = CHnV）时，CHnSCR [CHIF]位置 1 且通道(n)中断产生。

这种类型的 PWM 信号被称为中心对齐，因为所有通道的脉冲宽度中心与 CNTIN 的值对齐。

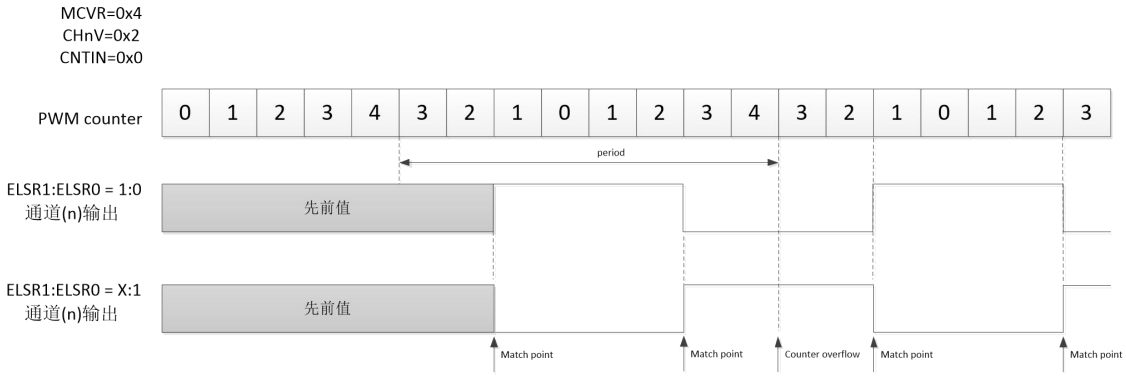


图 32-12 CPWM 波形

ELSnR1：ELSnR0 = 1：0，如果(CHnV = 0x0000)，则通道(n)输出为 0%占空比 CPWM 信号；如果(CHnV >= MCVR)，则通道(n)输出为 100%占空比 CPWM 信号，ELSnR1：ELSnR0 = X：1 则反之。

32.4.8 组合模式

组合模式下，将偶数通道(n)和相邻的奇数通道(n + 1)组合以在通道(n)输出中产生 PWM 信号。

根据计数方式的不同，可分为向上计数组合模式和向上-向下计数组合模式。

ELSnR1：ELSnR0 = 1：0，在通道(n)匹配时(PWM 计数器 = CH(n)V)为高电平；在通道(n + 1)匹配时为低电平。

ELSnR1：ELSnR0 = X：1，在通道(n)匹配时(PWM 计数器 = CH(n)V)为低电平；在通道(n + 1)匹配时为高电平。

CHnSCR [CHIE] = 1，通道(n)匹配时(PWM 计数器 = CH(n)V)，CHnSCR [CHIF]位置 1 且产生通道(n)中断。

CHn+1SCR[CHIE] = 1，通道(n + 1) 匹配时(PWM 计数器 = CH(n+1)V)，CHn+1SCR[CHIF]位置 1 且产生通道(n + 1)中断。

说明：

ELS(n + 1)R1 和 ELS(n + 1)R0 位不用于控制 channels (n) 和 (n + 1)输出。

© 2013 - 2024 杰发科技有限公司

32.4.8.1 向上计数组合模式

周期 = (MCVR - CNTIN + 0x0001) × PWM 计数器时钟周期

脉冲宽度 (占空比) = |CH(n + 1)V - CH(n)V| × PWM 计数器时钟周期

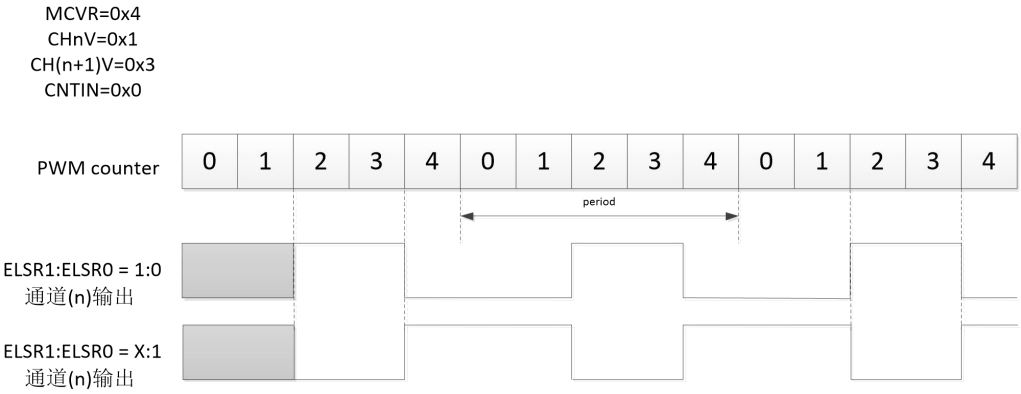


图 32- 13 向上计数组合模式输出波形

以下展示了向上计数组合模式的各种条件下 PWM 信号输出波形：

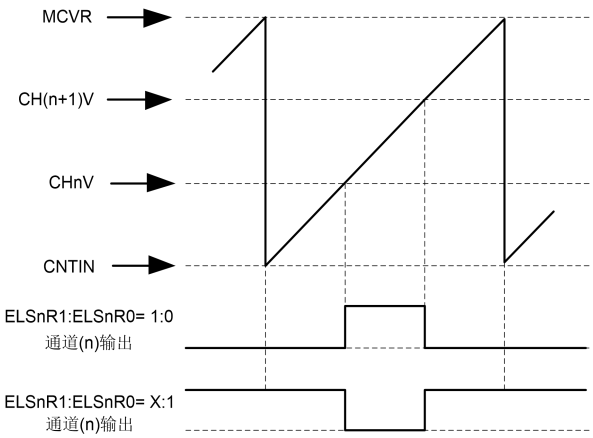


图 32- 14 (CNTIN < CHnV / CH(n + 1)V < MCVR) & (CHnV < CH(n + 1)V)条件下输出波形

© 2013 - 2024 杰发科技有限公司

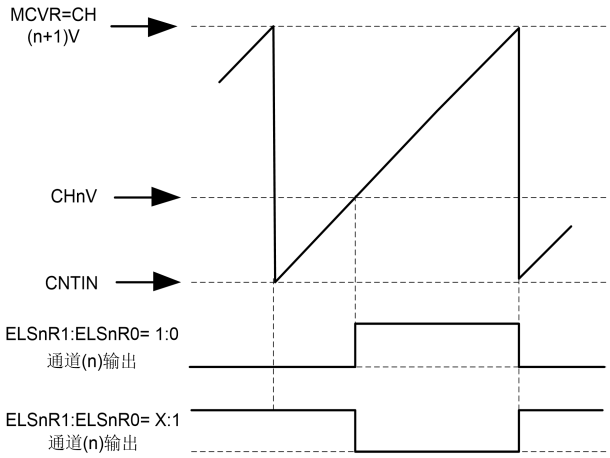


图 32- 15 (CNTIN < CHnV < MCVR) & (CH(n + 1)V = MCVR)条件下输出波形

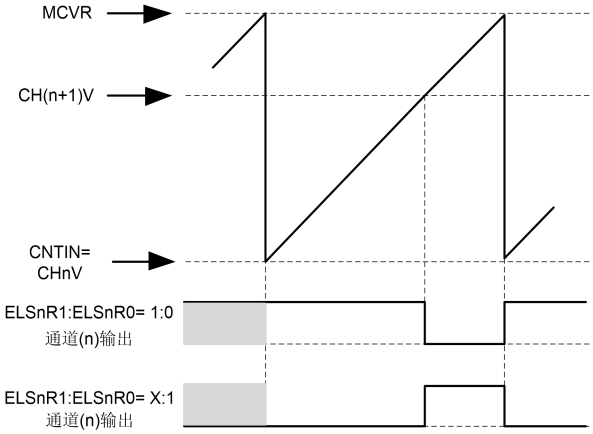


图 32- 16 (CHnV = CNTIN) & (CNTIN < CH(n + 1)V < MCVR)条件下输出波形

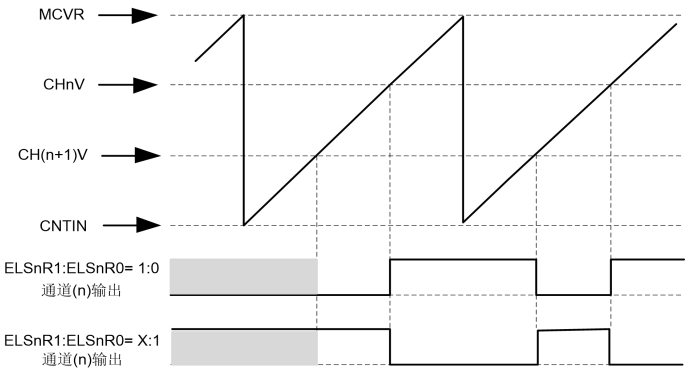


图 32- 17 (CNTIN < CHnV / CH(n+1)V < MCVR) & (CHnV > CH(n + 1)V)条件下输出波形

© 2013 - 2024 杰发科技有限公司

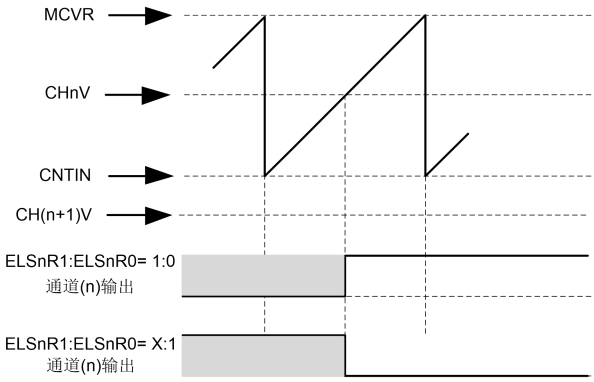


图 32- 18 (CH(n + 1)V < CNTIN) & (CNTIN < CHnV < MCVR)条件下输出波形

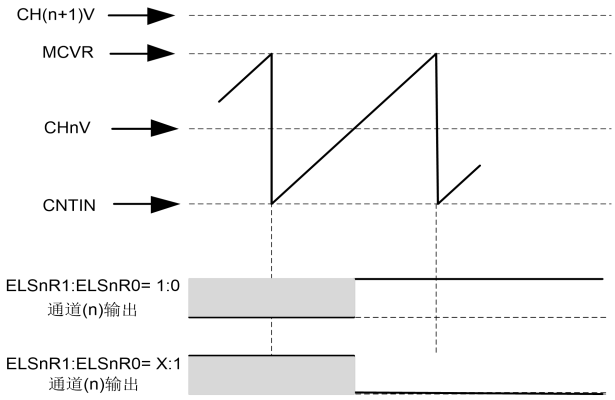


图 32- 19 (CH(n + 1)V > MCVR) & (CNTIN < CHnV < MCVR)条件下输出波形

32.4.8.2 向上-向下计数组合模式

周期 = 2 × (MCVR - CNTIN) × PWM 计数器时钟周期

在向上-向下计数过程中，通道在向上计数产生一次匹配，向下计数也会产生一次匹配。为了便于控制通道输出，提供了匹配生效点设置功能，可通过 CHnSCR [DIR]设置匹配生效点在向上计数或向下计数过程中作用。因为匹配生效点依赖于计数方向，需要清晰定义向上计数和向下计数区间范围，如图 32- 20 所示。

© 2013 - 2024 杰发科技有限公司

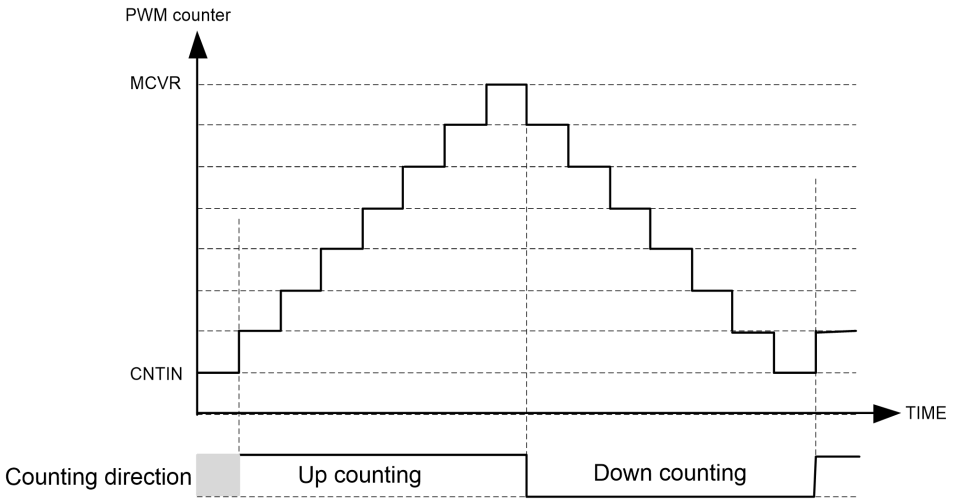


图 32- 20 向上-向下计数区间范围

说明：

首次计数周期 CNTIN 值为向上计数区间，后续周期 CNTIN 值都为向下计数区间；MCVR 值都为向上计数区间。

2 个匹配点可以组合出 4 种情况：

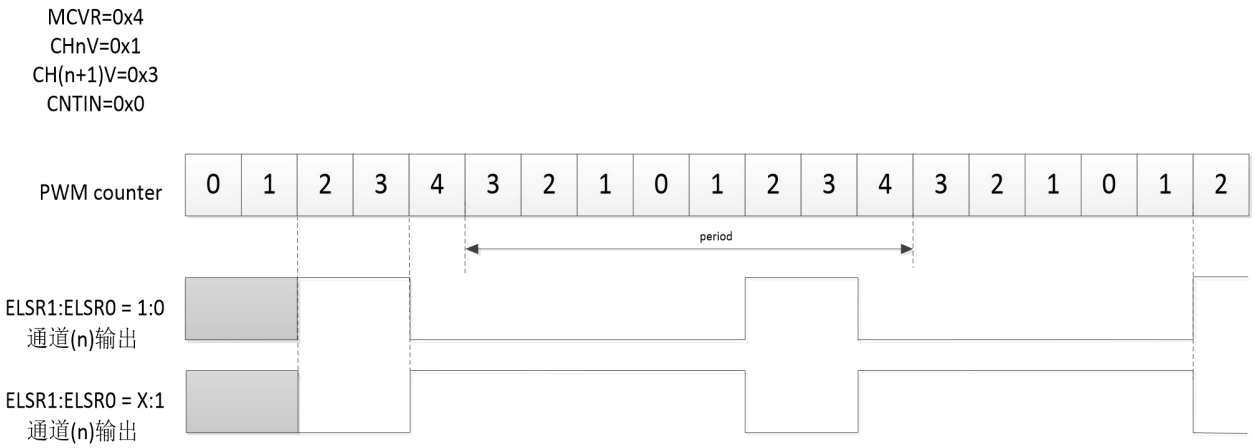


图 32- 21 CHn 匹配点 DIR = 1(Up)，CH(n + 1)匹配点 DIR = 1(Up)

© 2013 - 2024 杰发科技有限公司

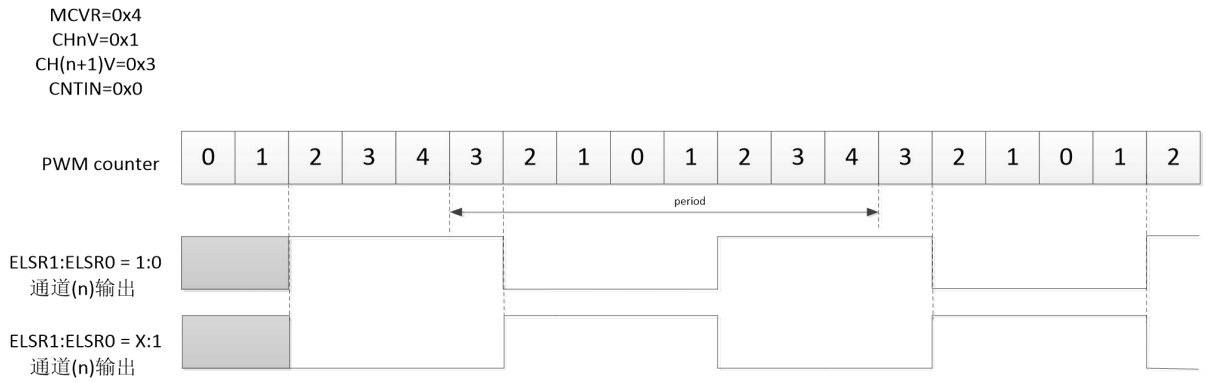


图 32- 22 CHn 匹配点 DIR = 1(Up)，CH(n + 1)匹配点 DIR = 0(Down)

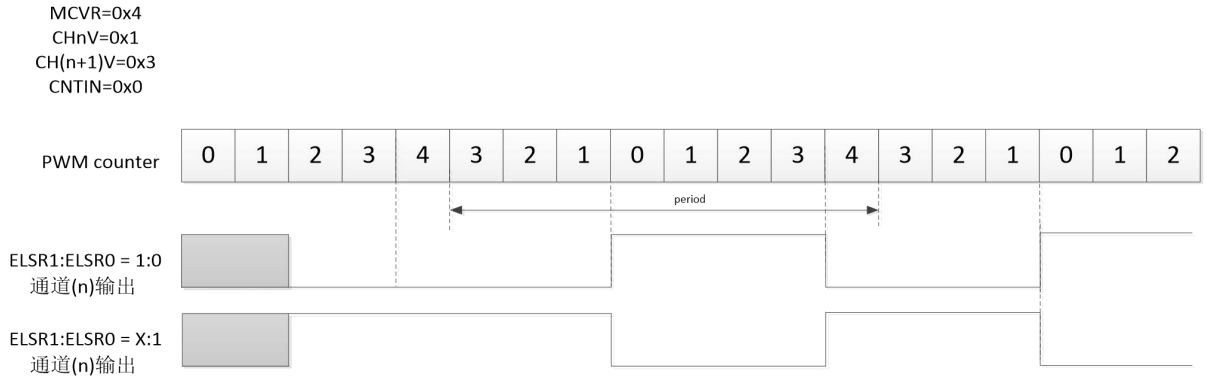


图 32- 23 CHn 匹配点 DIR = 0(Down)，CH(n + 1)匹配点 DIR = 1(Up)

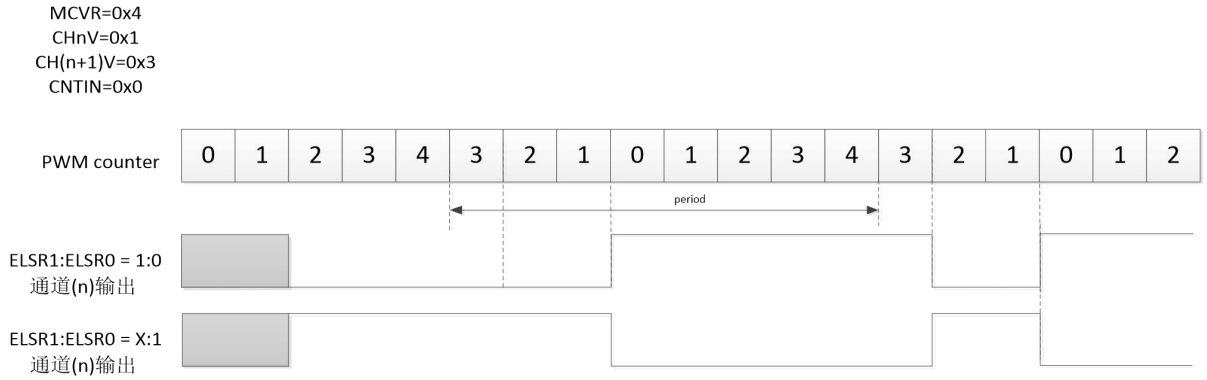


图 32- 24 CHnV 匹配点 DIR = 0(Down)，CH(n + 1)V 匹配点 DIR = 0(Down)

32.4.8.3 互补功能

组合模式下支持互补功能。使能互补 MODESEL[PAIRnCOMPEN]=1， 通道(n + 1) 输出和通道(n)输出电平相反。否则通道(n+1)输出与通道(n)相同。

© 2013 - 2024 杰发科技有限公司

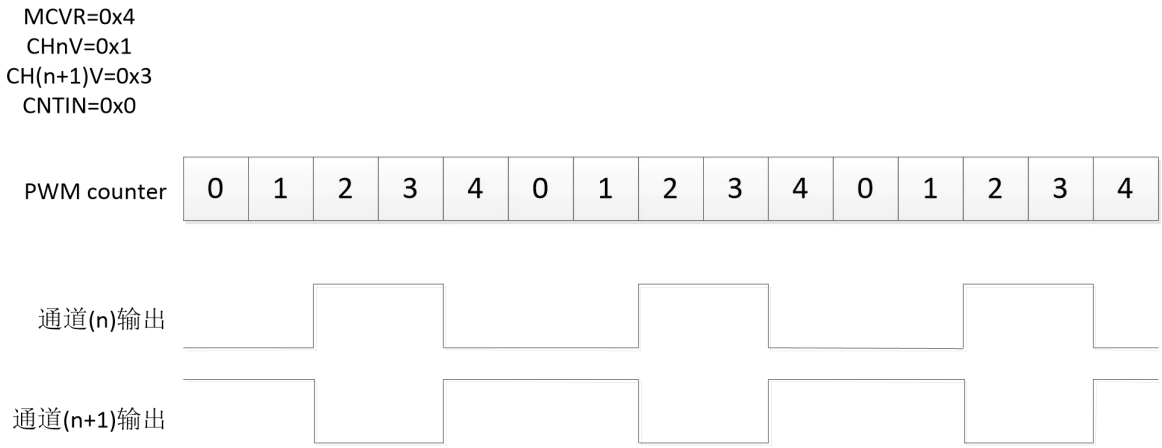


图 32- 25 互补模式输出

32.4.8.4 死区时间插入

当 MODESEL[PAIRnDTEN] = 1，且 DTSETn[DTVAL]为非零时，使能死区插入。DTSETn 寄存器定义了可用于所有 PWM 通道的死区延迟。DTSETn[DTPSC]位定义总线时钟的预分频器，DTSETn[DTVAL]位定义死区模数，即死区预分频器时钟数。死区延迟插入确保没有两个信号（通道 (n)和 通道(n+1)）同时驱动活动状态。

如果 CH(n)POL = 0，CH(n + 1)POL = 0，并且使能死区，那么当出现通道(n)匹配(PWM 计数器 = CH(n)V)时，通道(n)输出保持低电平状态，直到死区延迟结束，通道(n)输出置位。类似地，当发生通道(n + 1)匹配(PWM 计数器 = CH(n+1)V)时，通道(n + 1)输出保持低电平状态，直到死区延迟结束，通道(n + 1)输出置位。

如果 CH(n)POL = 1， CH(n + 1)POL = 1，并且使能死区，则当出现通道(n)匹配(PWM 计数器 = CH(n)V)时，通道 (n)输出保持高电平状态，直到死区延迟结束，通道 (n)输出清零。类似地，当发生通道(n + 1)匹配(PWM 计数器 = CH(n + 1)V)时，通道(n + 1)输出保持高电平状态，直到死区延迟结束，通道(n + 1)输出清零。

© 2013 - 2024 杰发科技有限公司

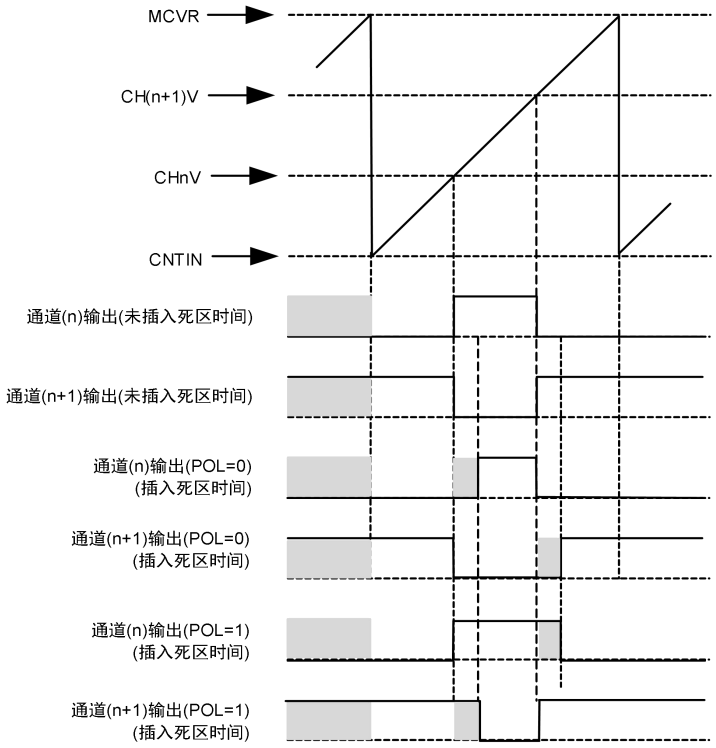


图 32-26 互补死区时间插入

32.4.8.5 反相

反相功能用于将通道(n)和通道(n + 1)输出信号进行交换。在如下情况选择反相操作：

INVCR[PAIRnINVEN] = 1, n=0,1,2,3。

32.4.8.6 移相

在某些情况下，通道(n+1)匹配会发生在下一个 PWM 周期。 当(CNTIN < CHnV < MCVR) & (CNTIN < CH(n + 1)V < MCVR)时， CHnV > CH(n + 1)V 时会发生此种情景。

© 2013 - 2024 杰发科技有限公司

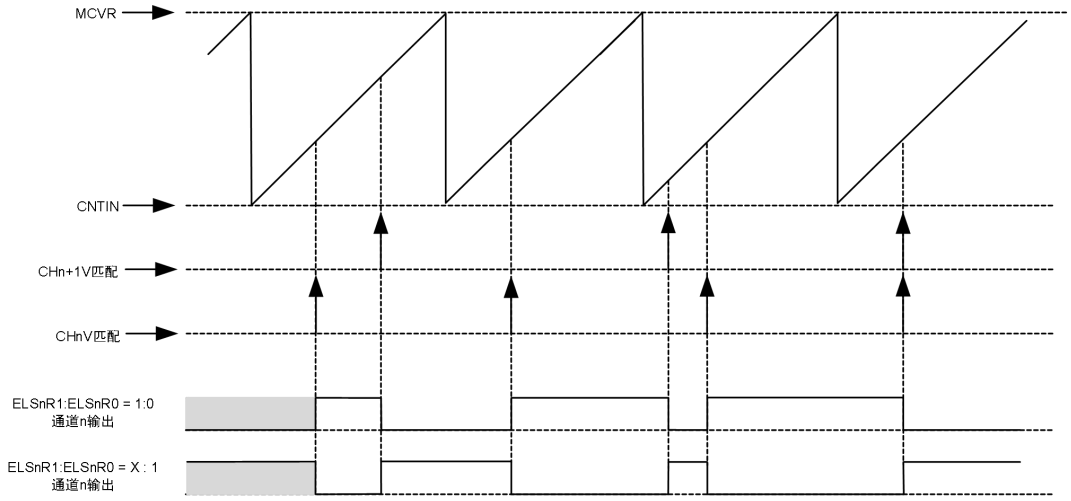


图 32-27 CH(n + 1)V 在下一周期匹配的输出波形

基于以上特性，当 PWM 模块多个通道以组合模式输出时，可设定同一个 PWM 模块不同通道对之间的相位偏移量。此种行为有利于产生合适的车灯控制信号，通道输出边沿不重合的信号便于消除噪声。

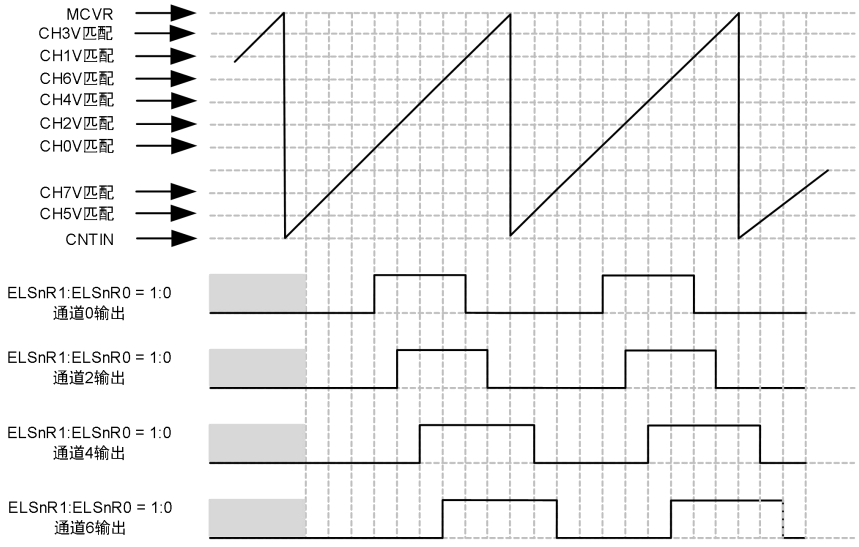


图 32-28 多通道之间相位偏移输出波形

32.4.9 双边沿捕获模式

MODESEL[PAIRnDECAPEN]= 1，则选择双边沿捕获模式。该模式可用于测量通道(n)输入端脉冲宽度或信号周期。在此模式下，仅使用通道(n)输入，忽略通道(n + 1)输入。当 n 为 0 或 2 时，通道(n)滤波器在此模式下有效。ELSnR1：ELSnR0 位选择通道(n)捕获的边沿，ELS(n + 1)R1：ELS(n + 1)R0 位选择通道(n + 1)捕获的边沿。如果 ELSnR1：ELSnR0 和 ELS(n + 1)R1：ELS(n + 1)R0_位都选择相同的边沿，则为周期测量；如果这些位选择不同的边沿，则为脉冲宽度测量。

如果在通道(n)输入处检测到通道(n)位选择的边沿，则 CHnSCR [CHIF]置位并生成通道(n)中断(如果 CHnSCR [CHIE] = 1)。如果在通道(n) 输入且(CHnSCR [CHIF] = 1)时检测到通道(n + 1)位选择的边

© 2013 - 2024 杰发科技有限公司

沿，则 CHn+1SCR[CHIF]置位且生成通道(n + 1) 中断(如果 CHn+1SCR[CHIE] = 1)。普通的双边沿模式不支持同时使能通道(n)和通道(n + 1)中断。

当在通道(n)输入处检测到通道(n)选择的边沿时，PWM_CH(n)V 寄存器存储 PWM 计数器的值。在通道(n)输入处检测到通道(n + 1)选择的边沿时，PWM_CH(n + 1)V 寄存器存储 PWM 计数器的值。在此模式下，当读取 PWM_CH(n)V 和 PWM_CH(n + 1)V 寄存器时，唯一的要求是必须在 CH(n + 1)V 之前读取 CH(n)V。

在双边沿捕获模式下若使能 CHnSCR[CHRSTEN]位，当检测到通道(n)位选择的边沿时，PWM 计数器的的值会被复位为 CNTIN 的值重新开始计数。通过设置相关的敏感边沿和 CHnSCR [CHRSTEN]位，可以直接得到输入信号的周期或脉宽。且当通过 FUNSEL[CHPnPNWEN]使能脉宽测量模式时，配合 CHnSCR [CHRSTEN]的设置可同时使能通道(n)和通道(n + 1)的中断，此时 CH(n)V 和 CH(n+1)V 也没有读取顺序的限制。

如图 32-29 和 图 32-30 所示，通道(n)选择上升沿捕获，通道(n + 1)选择下降沿捕获，用于测量脉宽。

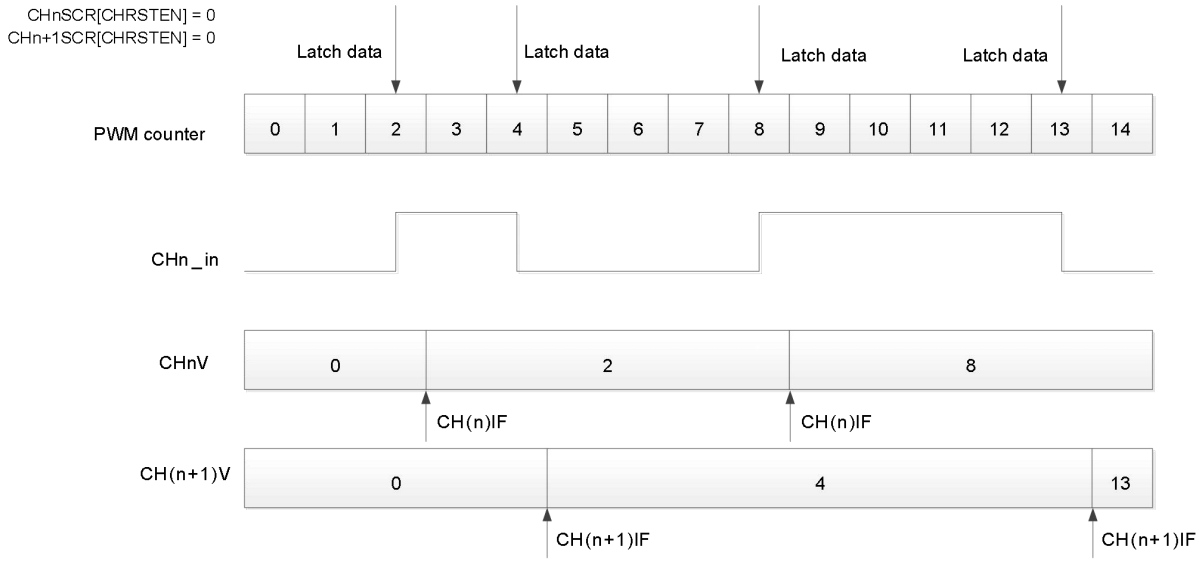


图 32-29 双边沿捕获模式(CHnSCR[CHRSTEN] = 0 & CHn+1SCR[CHRSTEN] = 0)

© 2013 - 2024 杰发科技有限公司

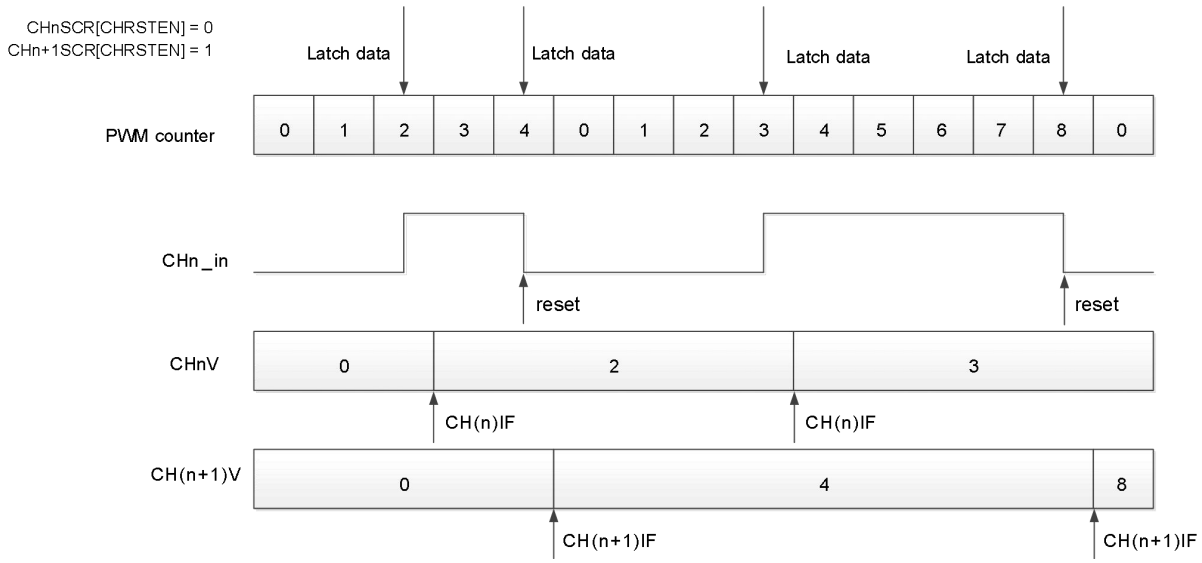


图 32-30 双边沿捕获模式(CHnSCR[CHRSTEN] = 0 & CHn+1SCR[CHRSTEN] = 1)

32.4.10 HALL 模式

MODESEL[PAIRnDECAPEN]= 1，FUNSEL[HALLEN] = 1 则选择 Hall 模式。作为 Hall 功能时，通道 0~2 作为 Hall A/B/C 的输入，而通道 3~7 无法正常使用，因为 Hall 模式使能后，每次通道中断触发后会自动复位 CNT 计数器。使用 Hall 功能时需要将通道 0 配置为双边沿捕获模式，模块测量从 3 个通道输入的异或(XOR)得到的脉冲信号，参见图 32-31。为正确的捕获 Hall 信号，需要配置 CH0SCR / CH1SCR 为通道 0 和 1 选择不同的边沿。如果 CH0SCR 选择上升沿捕获，CH1SCR 应该选择下降沿捕获。如果 CH1SCR 选择上升沿捕获，CH0SCR 应该选择下降沿捕获。当 CH0SCR[CHIE]和 CH1SCR[CHIE]使能时，每次捕获到 Hall 状态变化，对应的 CHnSCR[CHIF]位也会置 1，可以通过读取 STR[HALL_STATUS]得到此时的 Hall 状态，使能 CH0SCR[CHRSTEN]和 CH1SCR[CHRSTEN]位（Hall 模式下硬件会自动使能它们，用户无需配置这两位）以及脉宽测量使能位 FUNSEL[CHPnPNWEN]，可以在每次中断到来时通过 CH0V / CH1V 获得 Hall 信号的周期和占空比。

© 2013 - 2024 杰发科技有限公司

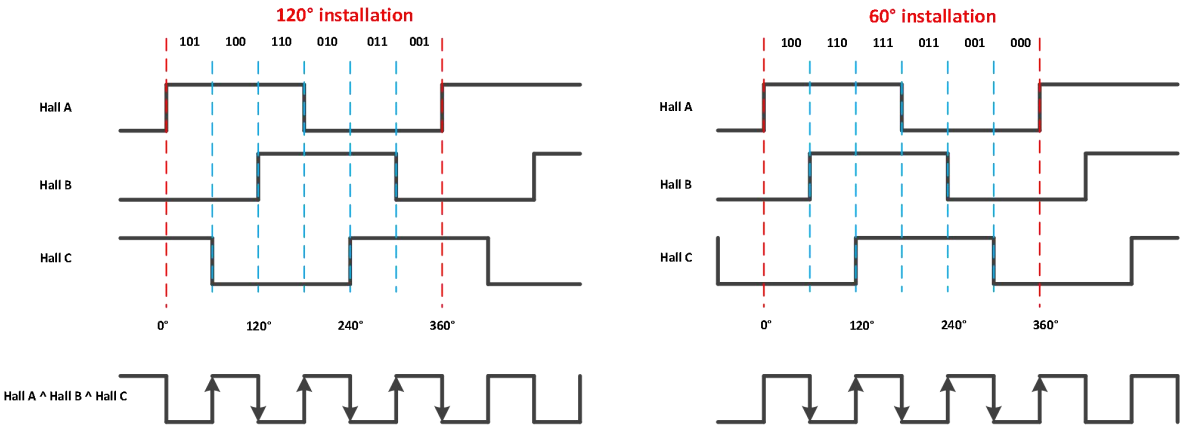


图 32-31 两种常见的 Hall 安装方式

32.4.11 正交解码模式

QDI [QDIEN] = 1 使能正交解码器模式。正交解码器模式采用输入信号相位 A 和相位 B 控制 PWM 计数器递增和递减，相位 Z 主要是用来对计数器进行复位，出现对应信号后，计数复位为起始值。每个输入信号相位 A 和相位 B 以及 Z 信号都有一个滤波器，该滤波器和通道输入中使用的滤波器是相同的，相位 A 的滤波值由 CAPFILTER [CH0CAPFVAL] 位定义，相位 B 的滤波值由 CAPFILTER [CH1CAPFVAL]位定义，相位 Z 的滤波值由 CAPFILTER [CH2CAPFVAL]位定义，值为 0 时禁用滤波器。

QDI [PHAPOL]位选择 A 相输入的极性，QDI [PHBPOL]位选择 B 相输入的极性。QDI [PHZPOL]位选择 Z 相输入的极性。QDI [QUADMODE] 选择正交解码器使用的编码模式。如果 QDI [QUADMODE] = 1，则使能计数和方向编码模式，参见图 32-32。在该模式下，相位 B 的输入表示计数方向，相位 A 的输入定义计数频率。当相位 A 输入信号有上升沿时，PWM 计数器将被更新。

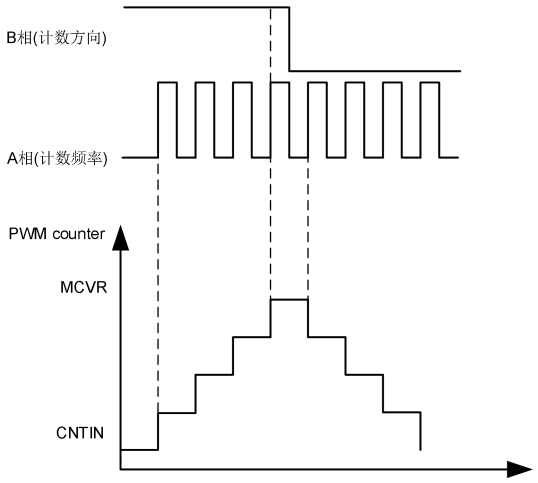


图 32-32 计数和方向编码模式

© 2013 - 2024 杰发科技有限公司

如果 QDI [QUADMODE] = 0，则启用相位 A 和相位 B 编码模式，参见图 32-33。在这种模式下，相位 A 和 B 信号之间的关系表示计数方向，相位 A 和 B 信号定义计数频率。当相位 A 或相位 B 信号有边沿时，PWM 计数器将被更新。

如果 PHAPOL = 0 & PHBPOL = 0，则 PWM 计数器递增发生在以下情况：

- A 相信号上升沿时，B 相信号为低电平；
- B 相信号上升沿时，A 相信号为高电平；
- B 相信号下降沿时，A 相信号为低电平；
- A 相信号下降沿时，B 相信号为高电平。

PWM 计数器递减发生在以下情况：

- A 相信号下降沿时，B 相信号为低电平；
- B 相信号下降沿时，A 相信号为高电平；
- B 相信号上升沿时，A 相信号为低电平；
- A 相信号上升沿时，B 相信号为高电平。

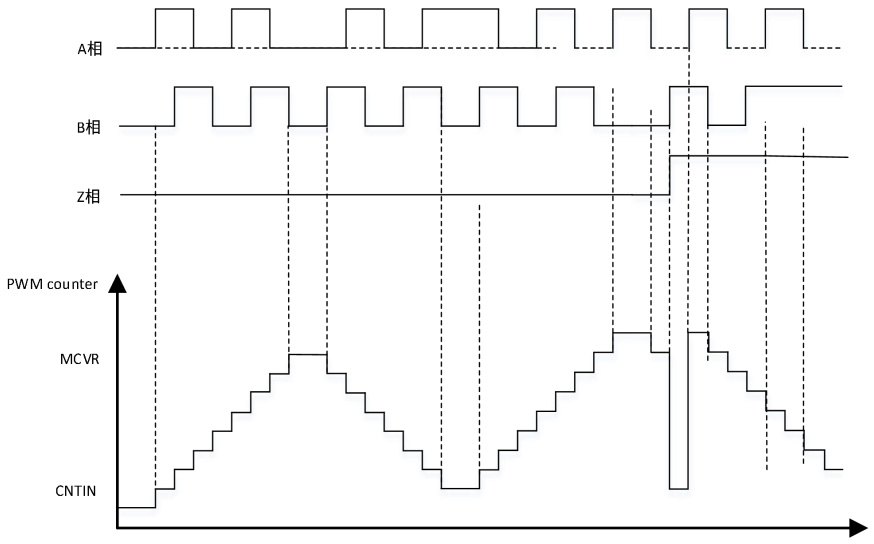


图 32-33 ABZ 编码模式

图 32-34 显示了向上计数时 PWM 计数器溢出。当 PWM 计数器从 MCVR 更改为 CNTIN 时，设置 INIT[CNTOF]和 INIT[OFDIR]位。INIT[CNTOF]位表示发生了 PWM 计数器溢出，INIT[OFDIR]指示 PWM 计数器在向上计数时发生溢出。

© 2013 - 2024 杰发科技有限公司

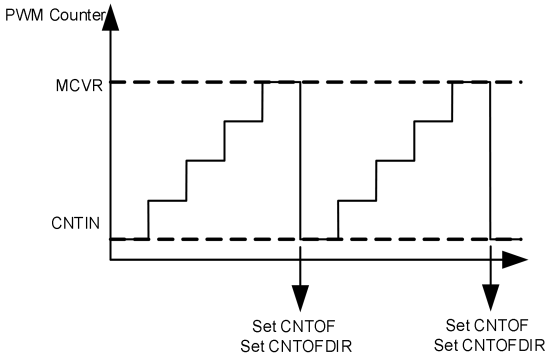


图 32-34 向上计数 PWM counter 溢出

图 32-35 显示了向下计数时 PWM 计数器溢出。当 PWM 计数器从 CNTIN 更改为 MCVR 时，将设置 INIT [CNTOF] 位并清除 INIT[OFDIR] 位。INIT[CNTOF] 位表示发生了 PWM 计数器溢出，INIT[OFDIR]指示 PWM 计数器在向下计数时发生溢出。

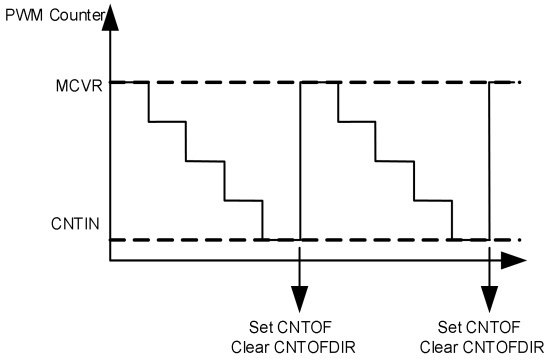


图 32-35 向下计数 PWM counter 溢出

32.4.12 写保护

在一些安全等级要求高的应用场景(如电机控制)，错误修改寄存器配置会引起系统异常，容易导致电机损坏。写保护功能可提供对某些关键寄存器位写入保护。用户在完成初始化配置后，可通过设置 FDSR[WPEN] = 1 激活写保护功能，避免人为或其它异常引起的对关键寄存器位误操作。设置 FUNCSEL[WPDIS] = 1 关闭写保护功能，支持写保护功能的位可重新写入。支持写保护功能的位在表 32-5 中进行说明。

32.4.13 初始化

向 PWM_FUNCSEL[INIT]位写入 1 时，初始化将强制通道(n)输出为 OUTINIT[CHnOIV]位的值。初始化特性只能在禁用 PWM 计数器时设置使用，当 PWM 计数器开始工作时 PWM_FUNCSEL[INIT]位自动清除，初始化输出一直保持直至 PWM 通道开始接管控制。

32.4.14 极性控制

© 2013 - 2024 杰发科技有限公司

CHPOLCR[CHnPOL]位选择通道(n) 输出极性：

- CHPOLCR[CHnPOL] = 0，通道(n) 输出极性为高，逻辑 1 为有效状态，逻辑 0 为无效状态
- CHPOLCR[CHnPOL] = 1，通道(n) 输出极性为低，逻辑 0 为有效状态，逻辑 1 为无效状态

32.4.15 输出屏蔽

输出屏蔽可用于通过软件强制通道输出为各自的无效状态。如果 OMCR[CHnOMEN] = 1，则通道(n) 输出强制为通道的无效状态（CHPOLCR[CHnPOL]位值）。

32.4.16 软件输出控制

软件输出控制可在 PWM 输出过程中强制通道输出软件定义值。CHOSWCR[CH(n)SWEN]位使能软件输出控制，CHOSWCR[CH(n)SWCV]设置通道强制输出值。非组合模式，软件输出功能每个通道可单独设置，组合模式软件输出功能如表 32-1 所示。

表 32-1 组合模式软件输出控制行为

CH(n)SWEN	CH(n+1)SWEN	CH(n)SWCV	CH(n+1)SWCV	Channel(n) 输出	Channel(n+1) 输出
0	0	X(1)	X(1)	关闭软件控制功能	关闭软件控制功能
1	0	0	X(1)	0	关闭软件控制功能
1	0	1	X(1)	1	关闭软件控制功能 or 0(2)
0	1	X(1)	0	关闭软件控制功能	0
0	1	X(1)	1	关闭软件控制功能 or 0(2)	1
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1 or 0(2)

说明：

- 注(1): X 为 0 或 1 中的任意值。
- 注(2): 当使能互补功能时，输出低电平。

32.4.17 初始化触发器

© 2013 - 2024 杰发科技有限公司

如果 EXTTRIG[INITTRIGEN] = 1，则在以下情况中 PWM 计数器更新为 CNTIN 寄存器值时，生成初始化触发。初始化触发输出提供用于片上模块的触发信号。

- PWM 计数器通过所选计数模式自动更新为 PWM_CNTIN 寄存器值
- 对 CNT 寄存器执行写操作
- PWM 计数器同步

32.4.18 通道匹配触发器

如果 EXTTRIG[CHnTRIG] = 1，在发生通道(n) 匹配(PWM 计数器 = CH(n)V)时，PWM 产生通道触发。通道触发输出提供用于片上模块的触发信号。

在向上-向下组合模式，通道值在向上计数产生一次匹配，向下计数也会产生一次匹配。为了便于控制通道匹配触发，可通过设置匹配生效点 CHnSCR[DIR]是在向上计数还是向下计数时作用。

32.4.19 最大值触发器

如果 EXTTRIG[MAXTRIGEN]= 1，当 PWM 计数值等于最大值时，产生最大值触发信号，最大值触发输出提供用于片上模块的触发信号。

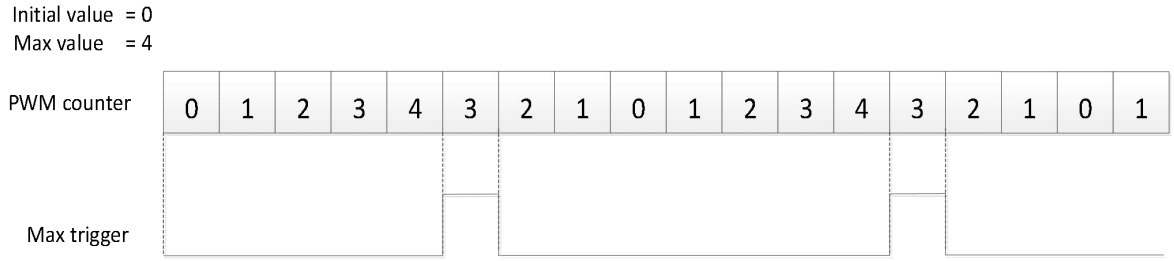


图 32-36 最大值触发器

32.4.20 故障控制

PWM 有以下故障源可供选择

表 32-2 PWM 故障源

	PWMn_Fault0	PWMn_Fault1	PWMn_Fault2	PWMn_Fault3
PWM0	PWM0_FLT0 pin or CTU output	PWM0_FLT1 pin or CTU output	PWM0_FLT2 pin or CTU output	PWM0_FLT3 pin
PWM1	PWM1_FLT0 pin or CTU output	PWM1_FLT1 pin or CTU output	PWM1_FLT2 pin or CTU output	PWM1_FLT3 pin
PWM2	PWM2_FLT0 pin or CTU output	PWM2_FLT1 pin or CTU output	PWM2_FLT2 pin or CTU output	PWM2_FLT3 pin
PWM3	PWM3_FLT0 pin or CTU output	PWM3_FLT1 pin or CTU output	PWM3_FLT2 pin or CTU output	PWM3_FLT3 pin

© 2013 - 2024 杰发科技有限公司

	PWMn_Fault0	PWMn_Fault1	PWMn_Fault2	PWMn_Fault3
PWM4	PWM4_FLT0 pin	PWM4_FLT1 pin	N/A	N/A
PWM5	PWM5_FLT0 pin	PWM5_FLT1 pin	N/A	N/A

FFAFER[FERnEN] 位使能故障输入 n，FFAFER[FFnEN] 位使能故障输入 n 滤波器。FFAFER[FFVAL]选择已使能的每个故障输入滤波器的值。

如果故障控制和故障输入 n 已使能，且在故障输入 n 信号上检测到有效边沿，则表明已出现故障状况且 FDSR[FAULTDFn]位已置位。FDSR[FAULTDF]是 FAULTDFn[3：0]位的逻辑或(OR)。

如果已使能故障控制(PWM_FUNCSEL[FAULTMODE] ≠ 0：0)、使能故障输入且已检测到故障，则强制 PWM 通道输出为各自安全值：

- 通道(n)输出采用 CHPOLCR[CHnPOL]的值
- 通道(n + 1)采用 CHPOLCR[CH(n+1)POL]的值

当(FDSR[FAULTDF]= 1)且(PWM_FUNCSEL[FAULTIE] = 1)时，生成故障中断。

32.4.20.1 自动故障清除

如果选择自动故障清除模式（PWM_FUNCSEL[FAULTMODE] = 1：1），则当故障输入信号(FDSR[FAULTIN])为零且新的 PWM 周期开始时，被故障控制禁用的通道输出将恢复正常输出。

FDSR[FAULTIN]的状态只与实际的 FAULT 输入和极性有关。

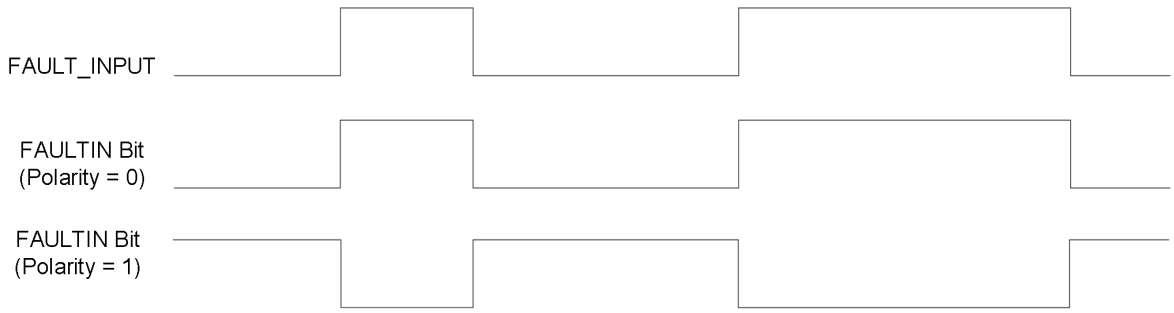


图 32-37 FAULTIN

32.4.20.2 手动故障清除

如果选择手动故障清除 (PWM_FUNCSEL[FAULTMODE] = 0：1 或 1：0)，则当 FDSR[FAULTDF]被清除且新的 PWM 周期开始时，被故障控制禁用的通道输出将恢复正常输出。

32.4.20.3 故障输入极性

FLTPOL[FLTnPOL]位选择故障输入 n 的极性：

- FLTPOL[FLTnPOL] = 0，故障 n 输入极性为高，故障输入 n 处的逻辑 1 代表一个故障
- FLTPOL[FLTnPOL] = 1，故障 n 输入极性为低，故障输入 n 处的逻辑 0 代表一个故障

© 2013 - 2024 杰发科技有限公司

32.4.21 PWM 同步

PWM 同步功能提供一个机制可将 MCVR，CNTIN， CHnV， OMCR， INVCR， CHOSWCR，DITHER0， DITHER1 和 DITHER2 寄存器对应的缓存值更新到寄存器，并设置 PWM 计数器为 CNTIN 寄存器值。

32.4.21.1 硬件触发器

当 SYNC[TRIGn] = 1 时，使能 PWM 模块的三个硬件触发信号输入，其中 n(0,1 或 2)分别对应于每个输入信号。硬件触发器输入 n 由总线时钟同步。

- 硬件触发器 0: CTU 触发输出
- 硬件触发器 1: CTU 中的 SEL_PWMnSYNC 位
- 硬件触发器 2: PWMn_FLT0 pin

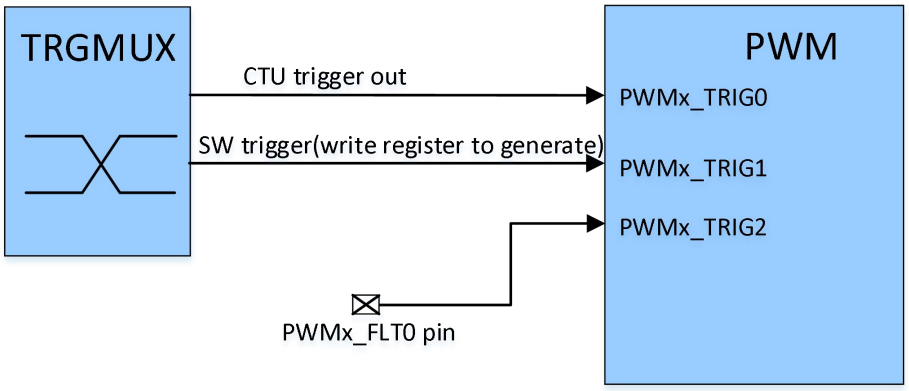


图 32-38 PWM 同步信号选择

如果 SYNCONF [HWTRIGMODE] = 0，在使能的硬件触发输入处检测到上升沿时，将启动与硬件触发的 PWM 同步。当 PWM_SYNC[TRIGn] 写入 0 或检测到触发 n 事件时，对应的 PWM_SYNC[TRIGn]位被清零。

在这种情况下，如果启用了两个或更多硬件触发器（例如，SYNC [TRIG0] = 1 且 SYNC[TRIG1] = 1），但仅发生触发 SYNC [TRIG1]的事件，则仅清零 SYNC [TRIG1]位，SYNC [TRIG0]位保持置 1。如果触发事件与写 SYNC [TRIGn]位一起发生，则启动同步，但由于写操作，SYNC [TRIGn]位保持置 1。

© 2013 - 2024 杰发科技有限公司

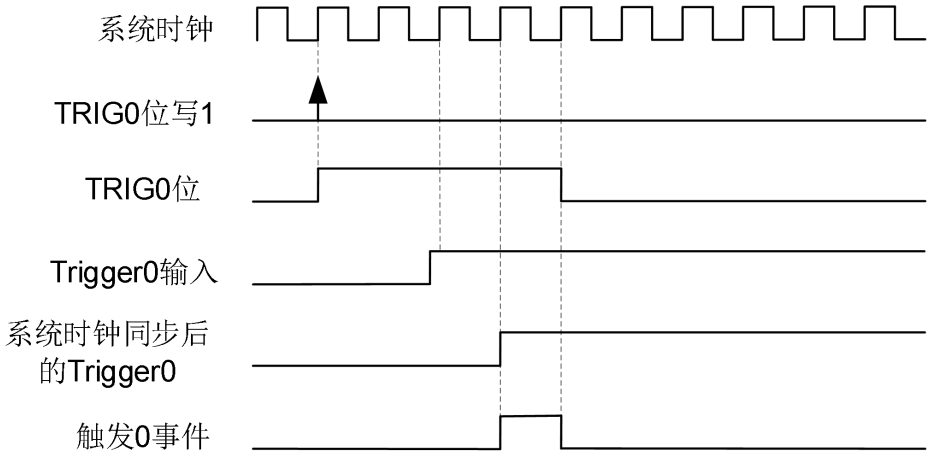


图 32-39 HWTRIGMODE = 0 硬件触发事件

32.4.21.2 软件触发器

当 SYNC [SWSYNC]位写入 1 时，会发生软件触发事件。当 SYNC [SWSYNC]位写入 0 时，或当由软件事件启动的 PWM 同步完成时，该位被清零。

如果同时发生另一个软件触发事件（通过将另一个 1 写入 SYNC [SWSYNC]位），则由先前软件触发事件启动的 PWM 同步结束，新的 PWM 同步开始，SYNC [SWSYNC]位保持为 1。

SYNCONF [SYNCMODE] = 1，则根据 SYNCONF [CNTVSWSYNC]位，SYNC [SWSYNC]位也会被 PWM 清零。如果 SYNCONF [CNTVSWSYNC] = 0，则软件触发事件发生后，在下一个选定的加载点清除 SYNC [SWSYNC]位。如果 SYNCONF [CNTVSWSYNC] = 1，然后在软件触发事件发生时清除 SYNC [SWSYNC]位。

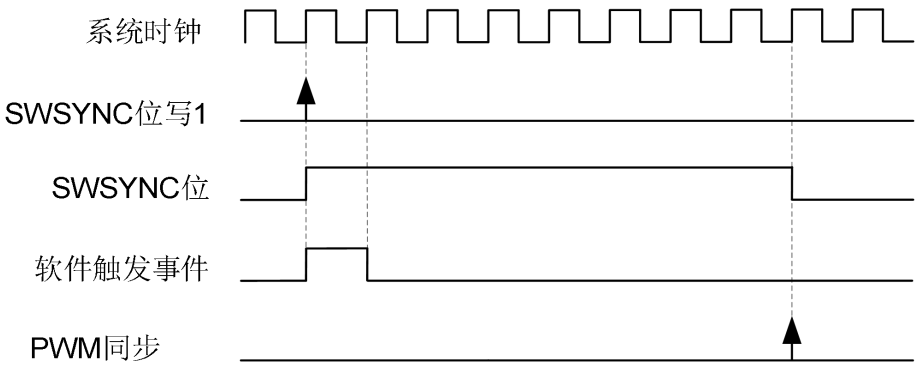


图 32-40 软件触发事件

32.4.21.3 边界周期和加载点

向上计数模式下，边界周期定义为计数器变为其初始值(CNTIN)的时候。向上-向下计数模式下，边界周期则定义为计数器从向下计数变为向上计数的时候以及从向上计数变为向下计数的时候。图 32-41 显示了寄存器的边界周期和加载点：

© 2013 - 2024 杰发科技有限公司

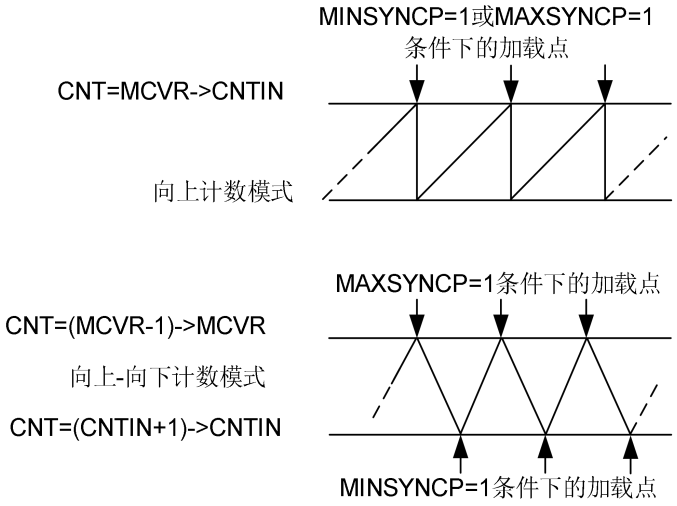


图 32- 41 边界周期与加载点

向上计数模式中，SYNC [MINSYNCP]或 SYNC [MAXSYNCP]其中一位为 1，则使能加载点。向上-向下计数模式下，由 SYNC [MINSYNCP]和 SYNC [MAXSYNCP]位选择加载点。在这两种计数模式中，如果 SYNC [MINSYNCP]和 SYNC [MAXSYNCP]都不是 1，则边界周期不用作寄存器更新的加载点。有关详细信息，请参见以下各节中的寄存器同步说明。

32.4.21.4 PWM_MCVR 寄存器同步

FUNSEL[PWMSYNCEN] = 1 且 SYNCONF[SYNCMODE] = 1 时，若 SYNCONF [PWMSVHWSYNC] = 1 或者 SYNCONF[PWMSVSWSYNC] = 1，则使能 MCVR 寄存器同步功能，同步时将其缓存值更新至 MCVR 寄存器。

MCVR 寄存器同步流程图如下：

© 2013 - 2024 杰发科技有限公司

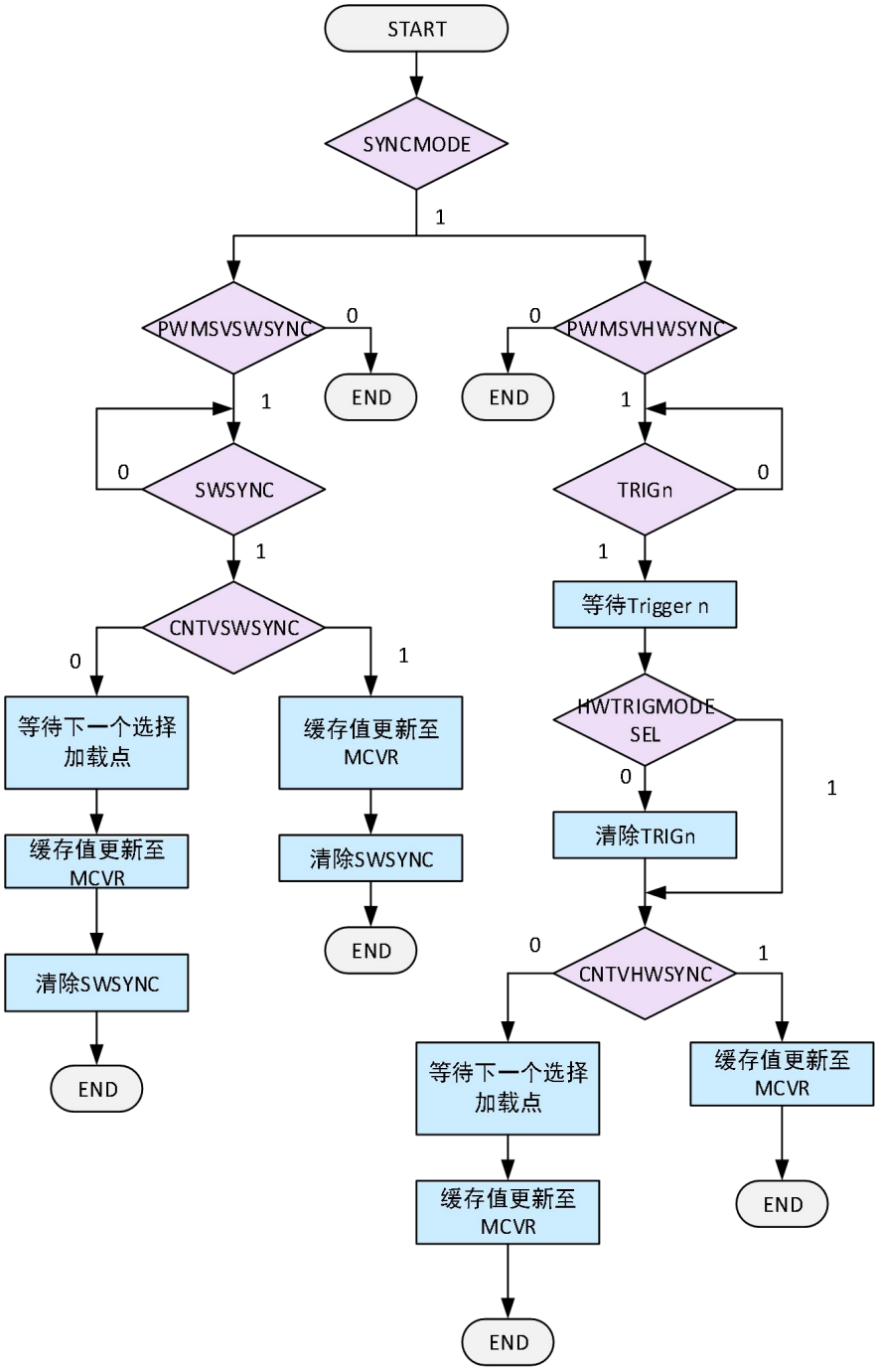


图 32-42 PWM_MCVR 寄存器同步流程

© 2013 - 2024 杰发科技有限公司

32.4.21.5 PWM_CNT 寄存器同步

CNT 寄存器同步功能可以在 PWM 周期中的特定点重新开始生成 PWM。通道输出强制为各自的初始值，CNT 寄存器强制为 CNTIN 寄存器定义的初始计数值。CNT 寄存器同步流程图如下：

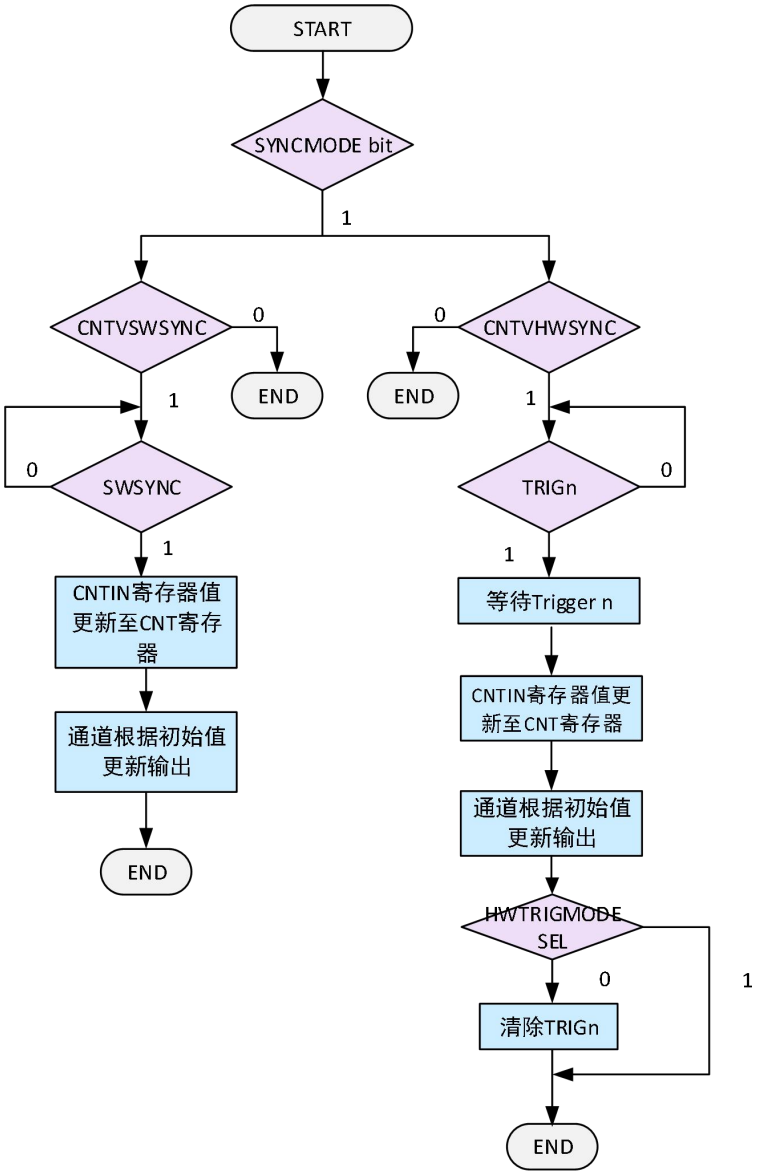


图 32- 43 PWM_CNT 寄存器同步流程

32.4.21.6 PWM_CNTIN 寄存器同步

FUNSEL[PWMSYNCEN] = 1、 SYNCONF [SYNCMODE] = 1 且 SYNCONF [CNTINC] = 1，将使能这种同步。同步机制与 MCVR 寄存器同步相同，参照图 32- 42。

© 2013 - 2024 杰发科技有限公司

32.4.21.7 PWM_CH(n)V 和 PWM_CH(n+1)V 寄存器同步

FUNSEL[PWMSYNCEN] = 1 且 MODESEL[PAIRnSYNCEN] = 1，则会使能该同步。同步机制与 MCVR 寄存器同步流程相同。

32.4.21.8 PWM_OMCR 寄存器同步

OMCR 寄存器同步将其缓存值更新至 OMCR 寄存器。参照以下流程图：

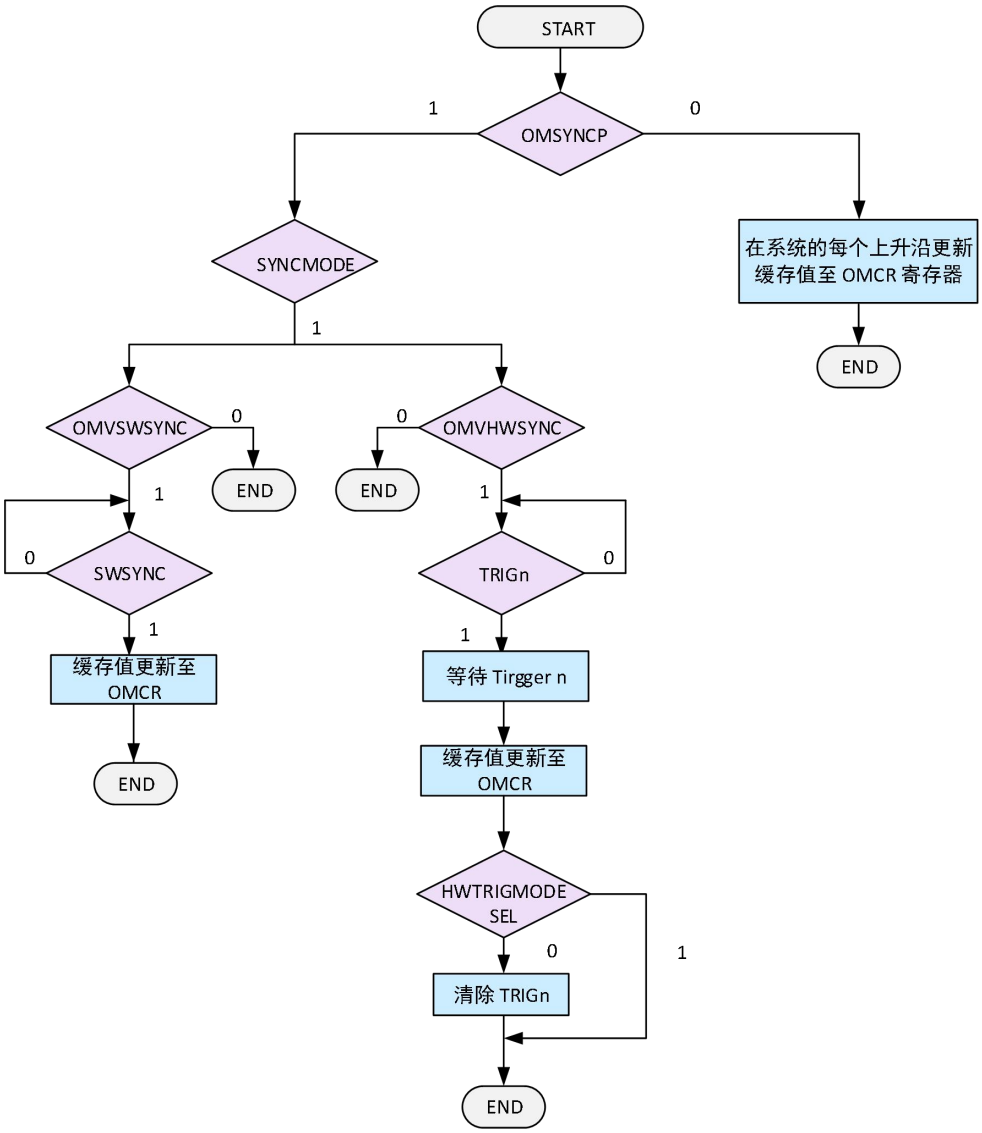


图 32-44 PWM_OMCR 寄存器同步流程

© 2013 - 2024 杰发科技有限公司

32.4.21.9 PWM_INVCR 寄存器同步

INVCR 寄存器同步将其缓存值更新至 INVCR 寄存器。参照以下流程图：

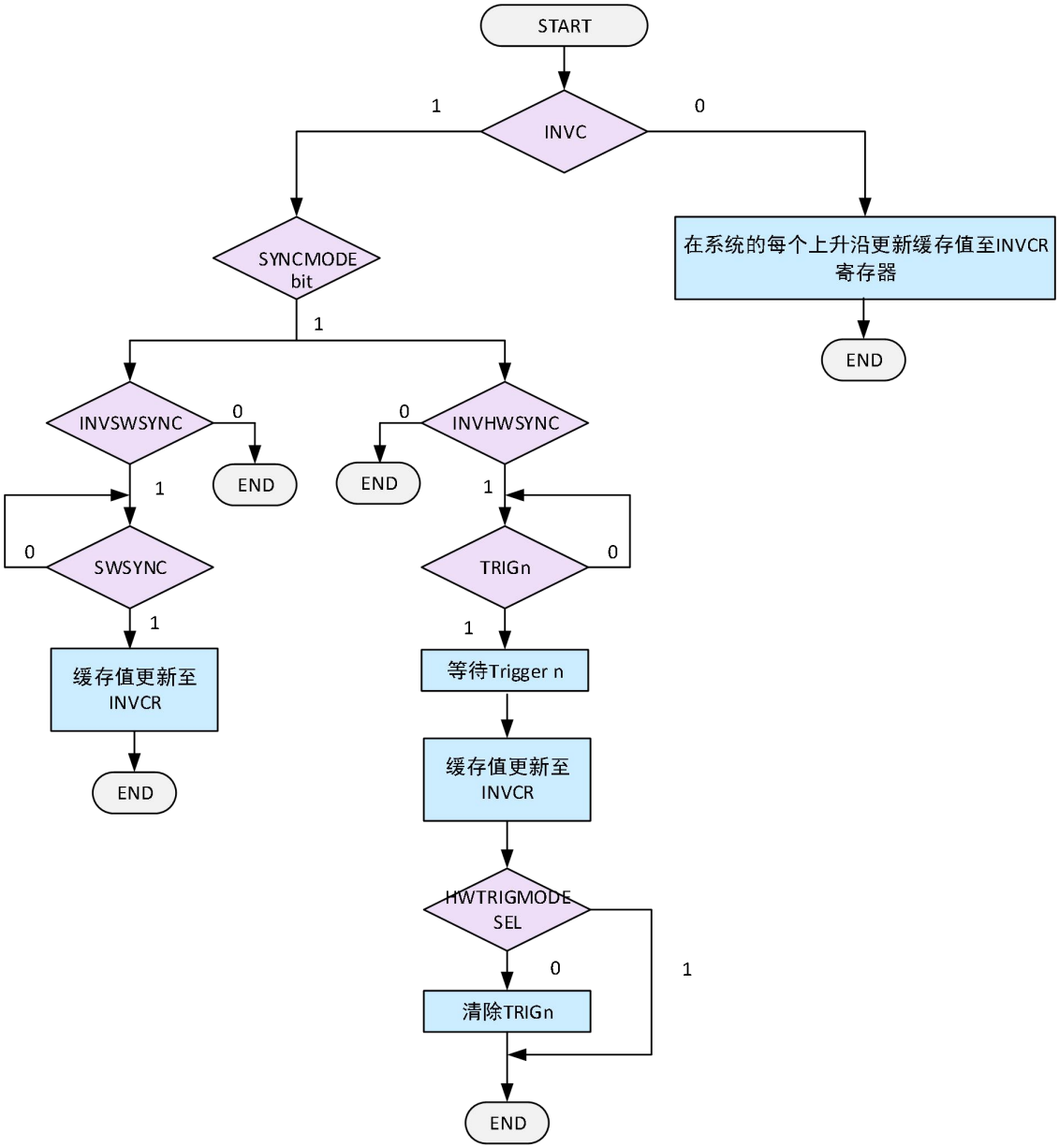


图 32-45 PWM_INVCR 寄存器同步流程

© 2013 - 2024 杰发科技有限公司

32.4.21.10 PWM_CHOSWCR 寄存器同步

CHOSWCR 寄存器同步将其缓存值更新至 CHOSWCR 寄存器。参照以下流程图：

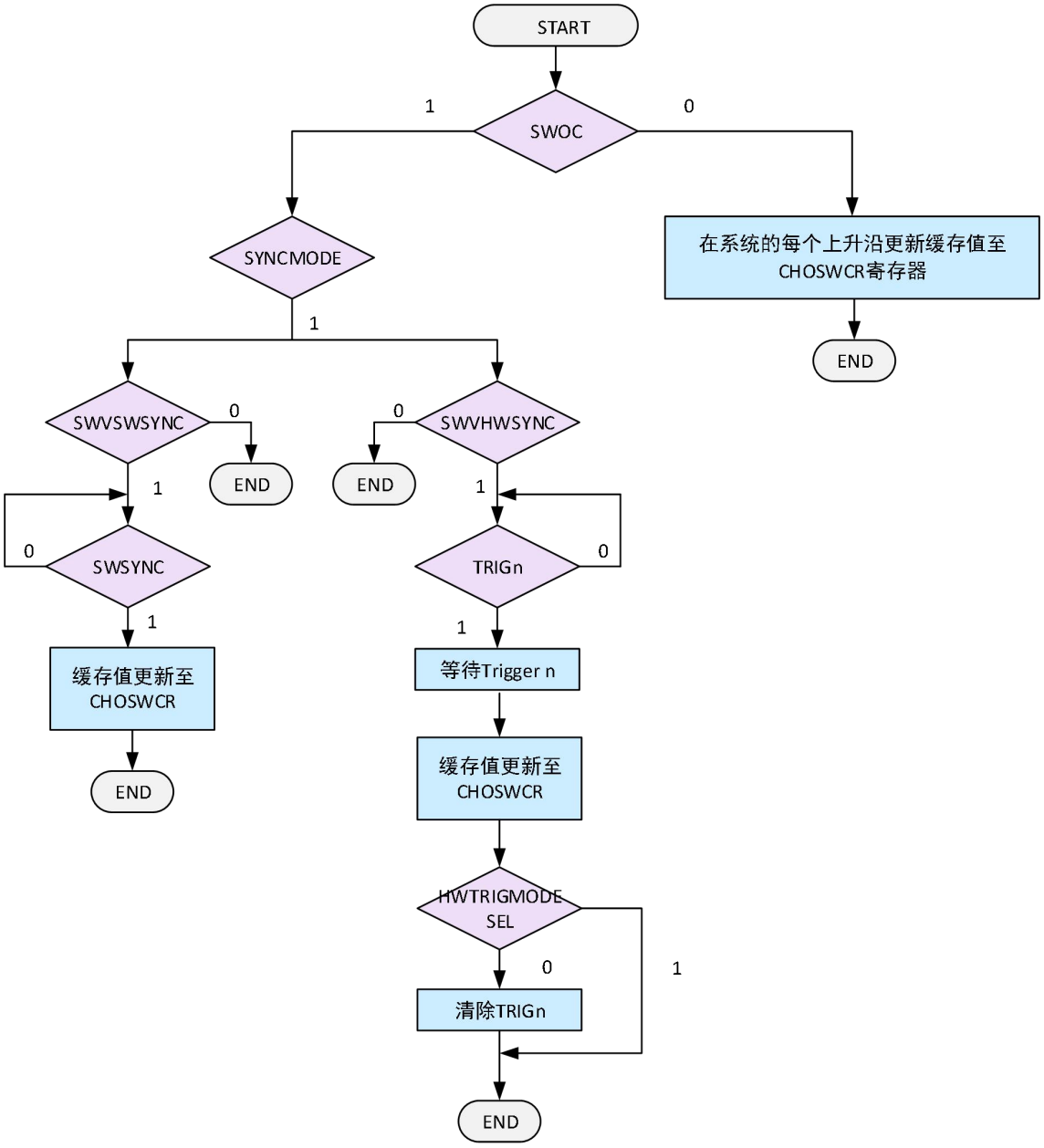


图 32-46 PWM_CHOSWCR 寄存器同步流程

© 2013 - 2024 杰发科技有限公司

32.4.21.11 PWM_CHOPOLCR 寄存器同步

CHOPOLCR 寄存器同步将其缓存值更新至 CHOPOLCR 寄存器。参照以下流程图：

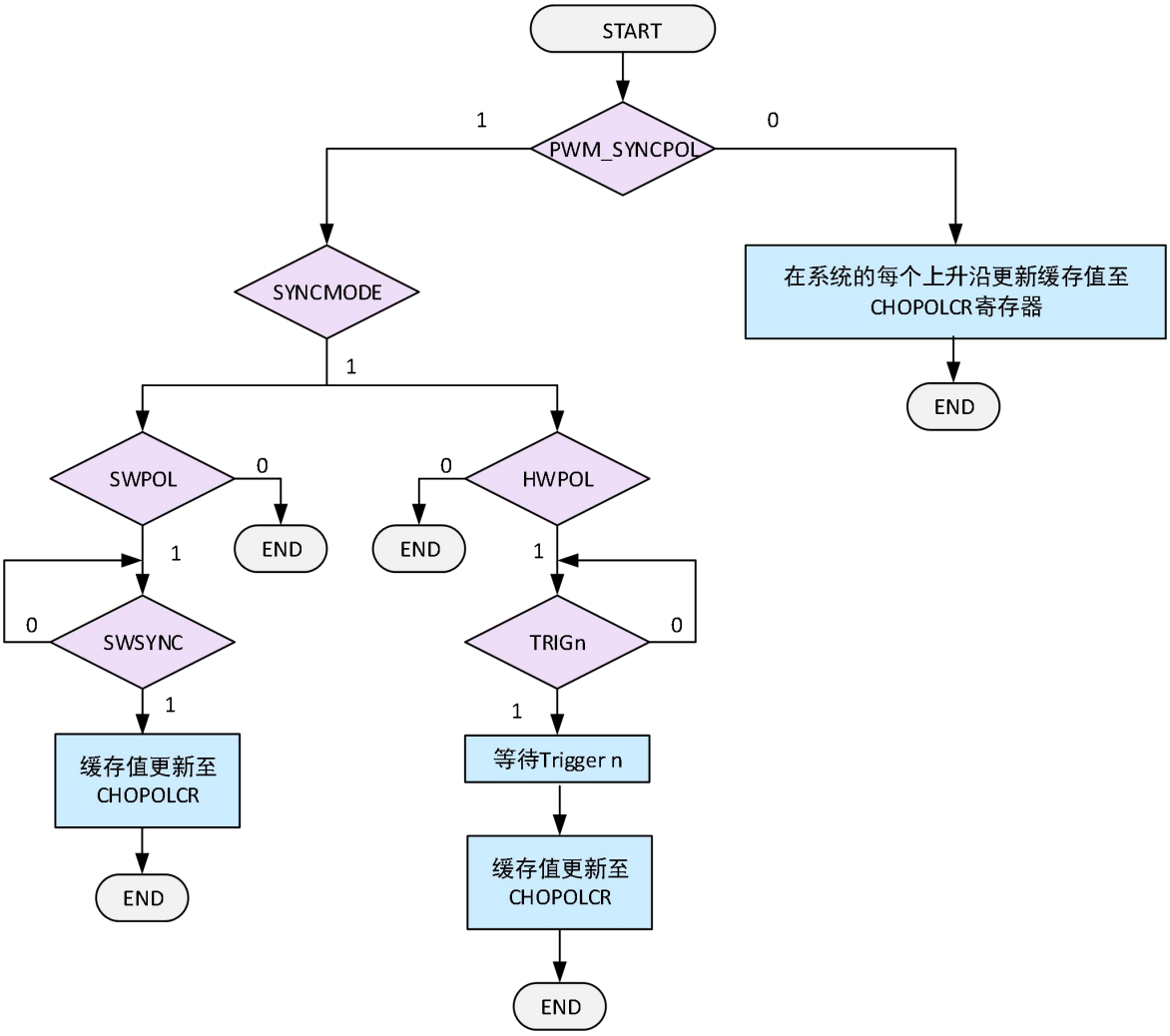


图 32- 47 PWM_CHOPOLCR 寄存器同步流程

32.4.21.12 PWM_DITHERn，PWM_PHDR 寄存器同步

FUNSEL[PWMSYNCEN] = 1 且 MODESEL[PAIRnSYNCEN] = 1，则会使能该同步。同步机制与 MCVR 寄存器同步流程相同。

32.4.22 特性优先级

下图为通道(n)和(n+1)输出信号时所使用的特性优先级。

© 2013 - 2024 杰发科技有限公司

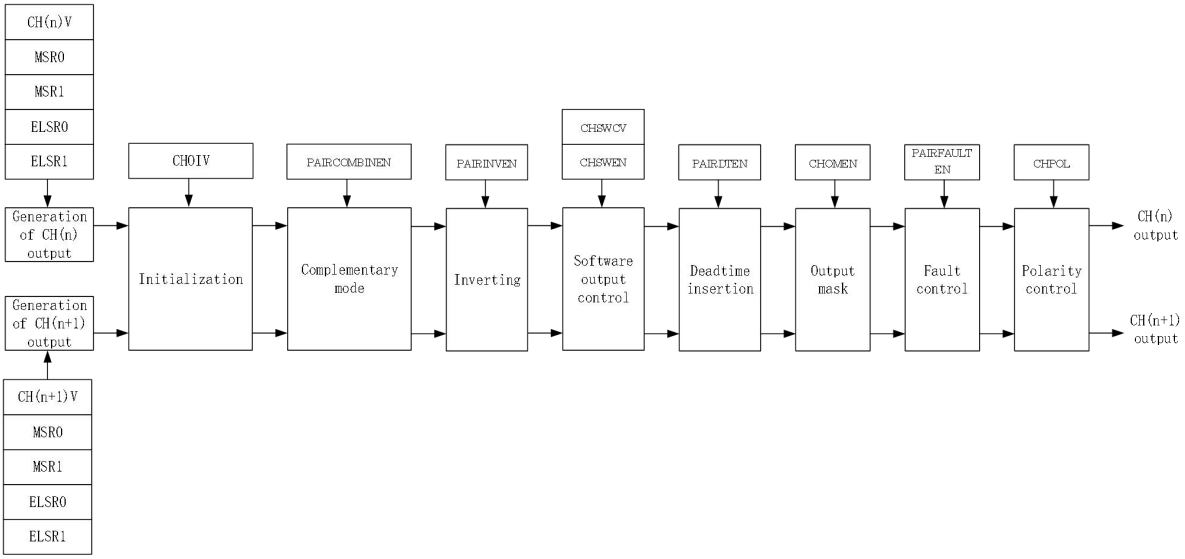


图 32-48 PWM 特性优先级

32.4.23 PWM 抖动

PWM 抖动主要是用于实现部分延迟，以实现对生成的 PWM 信号的精细分辨率。

32.4.23.1 PWM 边沿抖动

配置 DITHERn[CnDHR]寄存器可以配置各通道对应的边沿抖动值，当计数器值等于 PWM MCVR 值时，会将 DITHERn[CnDHR]的值累加到内部一个宽度为 5 的累加器上，当累加器的值大于或者等于 0x20 时，会在下个周期的 CHnV 处增加 1 个计数器的值。

在组合模式下，抖动插入的时机与独立模式一致，每个通道插入的时机与个各自通道抖动值有关。

© 2013 - 2024 杰发科技有限公司

32.4.23.1.1 向上计数

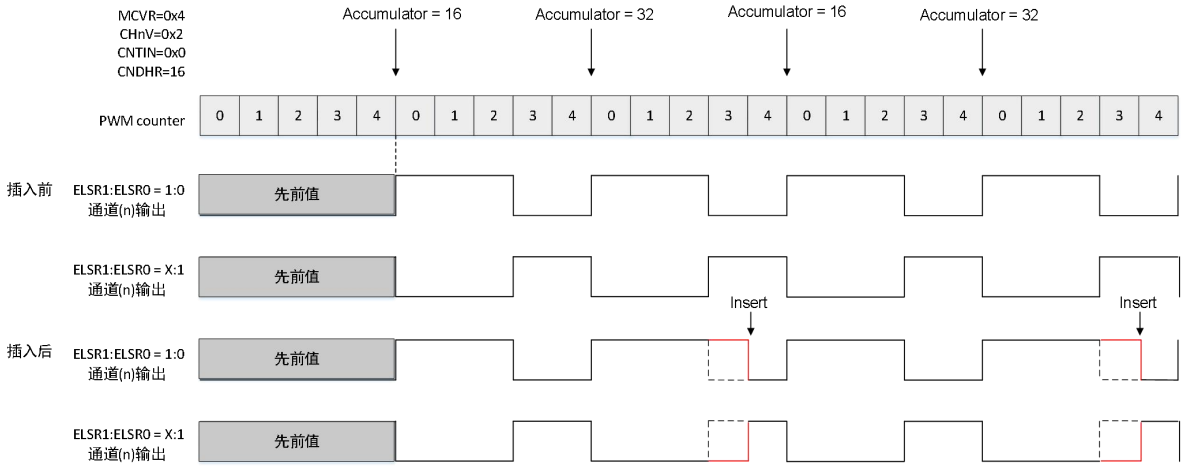


图 32-49 PWM 向上计数_边沿抖动波形

32.4.23.1.2 向上-向下计数

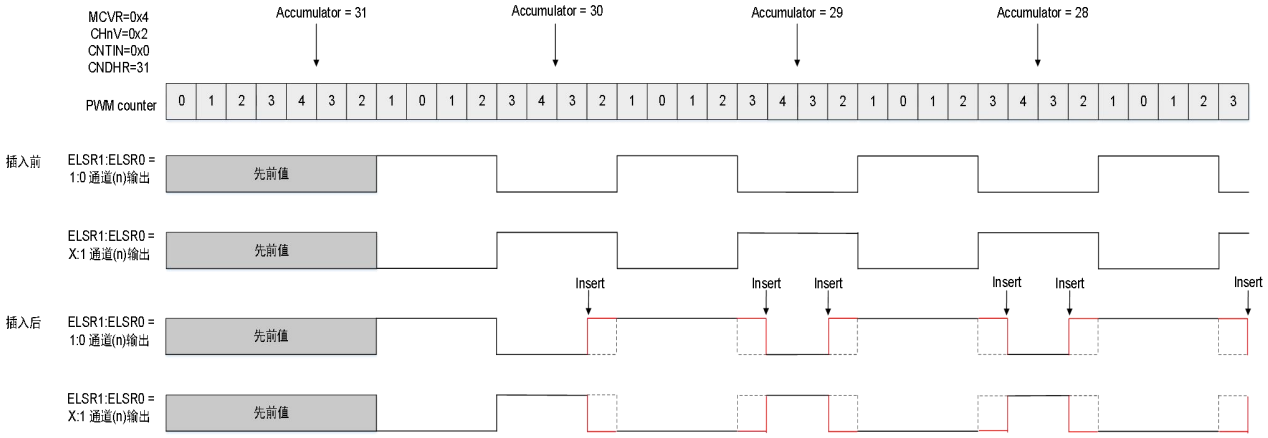


图 32-50 PWM 向上-向下计数_边沿抖动波形

© 2013 - 2024 杰发科技有限公司

32.4.23.2 PWM 周期抖动

配置 DITHER2 [PDHR]寄存器可以配置各通道对应的周期抖动值，当计数器值等于 MCVR 值时，会将 DITHER2[PDHR]的值累加到内部一个宽度为 5 的累加器上，当累加器大于或者等于 0x20 时，会在下个周期的 MCVR 处增加 1 个计数器的值。

32.4.23.2.1 向上计数

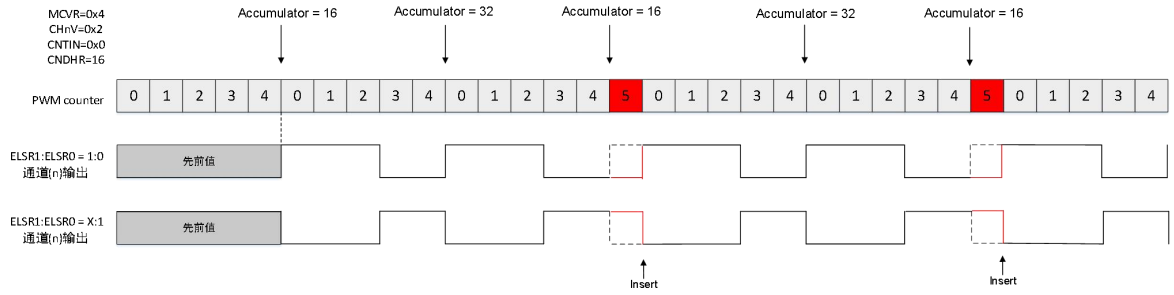


图 32-51 PWM 向上计数_周期抖动波形

32.4.23.2.2 向上-向下计数

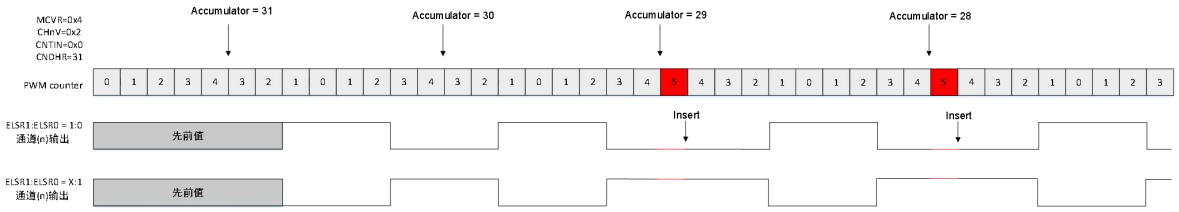


图 32-52 PWM 向上-向下计数_周期抖动波形

32.4.24 DMA 功能

每个 PWM 中存在三种 DMA 请求，包括单个信道请求（CHn_dma_req，n = 0~7）、或信道请求（CHor_ dma_ req）和上溢/下溢请求（OFUF_dma_req）。

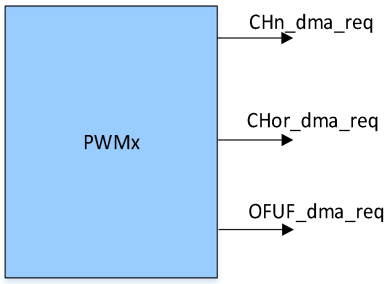


图 32-53 PWM DMA 信号

用户可以根据 DMA 中配置的源/目标地址和传输长度，将数据从 PWM 寄存器移到 RAM 存储器或从 RAM 存储器移到 PWM 寄存器。在 PWM 中，用户应在 PWMx_INIT/PWMx_CHnSCR 中配置相应

© 2013 - 2024 杰发科技有限公司

的使能位。例如，图 32-54 描述了两种方向转移的情况。一个方向是将 PWM 寄存器数据移动到 RAM，另一个方向是将 RAM 数据移动到 PWM 寄存器。同时，详细地址和传输号等的详细配置由 DMA 配置决定。DMA 请求信号的持续时间由 DMA 传输速度确定。

当一次传输的长度超过 32bit 时，需要使能 DMACTRL[DMA_TRANS_EN] 位，并通过 DMACTRL[DMA_TRANS_LEN]配置一次传输的长度。可参考图 32-55。

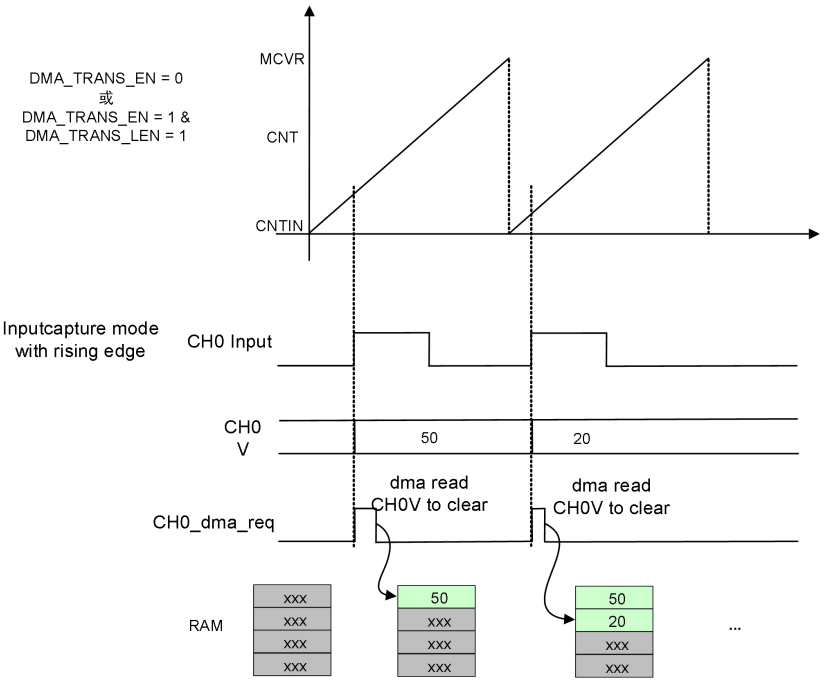


图 32-54 PWM DMA Read

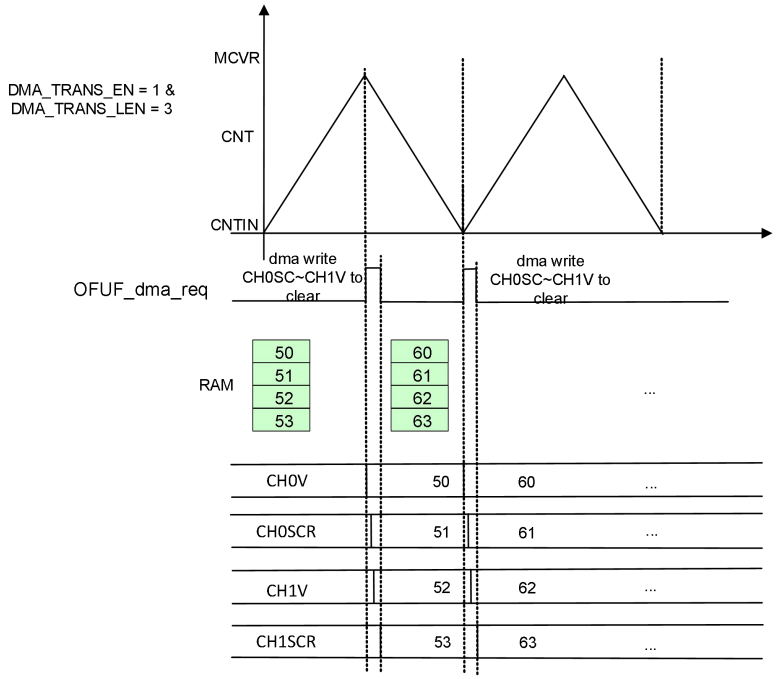


图 32-55 PWM DMA Write

© 2013 - 2024 杰发科技有限公司

32.4.25 全局时基

全局时基(GTB)允许在同一时刻启动芯片上的多个 PWM 模块同时工作，达到多个 PWM 模块同步开始计数的目的。

GTB 功能可通过 CONF 寄存器中的 GTBEEN 和 GTBEOUT 位实现。

要使能 GTB 特性，请对每个参与的 PWM 模块执行以下步骤：

1. 复位 PWM 模块；
2. 配置 PWM 模块，但不开启 PWM 时钟，向 INIT[CLKSRC]写入 00b；
3. 向 CONF[GTBEEN]写入 1，同时向 CONF[GTBEOUT]写入 0；
4. 在 PWM_INIT[CLKSRC]中选择 PWM 计数器时钟源；
5. 向 PWM_CNT 寄存器中写入任意值复位 PWM 计数器；
6. CONF[GTBEOUT]写入 1，启动多个 PWM 模块同时工作。

32.4.26 中断请求

表 32-3 PWM 中断汇总

中断源	标志	本地使能	全局使能
PWM 计数器上溢/下溢	CNTOF/OFDIR	COFE/ CUFE	无
PWM 通道事件	CHIF	CHIE	无
PWM 故障事件	FAULTDF1/ FAULTDF2/ FAULTDF3/ FAULTDF	FAULTIE	无

32.4.27 低功耗模式

表 32-4 PWM 低功耗模式

模式	唤醒源	备注
STOP1	\	关闭状态，寄存器值保持不变
STOP2	\	关闭状态，寄存器值保持不变
VLPS	\	掉电状态，寄存器值被复位

32.4.28 调试模式

可以通过配置 DITHER2[DEBUGCTRL]来配置 PWM 的调试模式。在不同的调试模式下，debug halt 后 PWM 的计数器和输出电平状态不一样。

© 2013 - 2024 杰发科技有限公司

32.5 寄存器定义

表 32-5 PWM 寄存器映射

PWM0 基地址 = 0x40080000
PWM1 基地址 = 0x40080800
PWM2 基地址 = 0x40081000
PWM3 基地址 = 0x40081800
PWM4 基地址 = 0x40082000
PWM5 基地址 = 0x40082800

地址	名称	宽度	描述
PWMx 基地址+0x00	PWM_INIT	32	初始化寄存器
PWMx 基地址+0x04	PWM_CNT	32	计数器寄存器
PWMx 基地址+0x08	PWM_MCVR	32	最大计数值寄存器
PWMx 基地址+0x0C	PWM_CH0SCR	32	Channel (0) 状态和控制寄存器
PWMx 基地址+0x10	PWM_CH0V	32	Channel (0)值
PWMx 基地址+0x14	PWM_CH1SCR	32	Channel (1) 状态和控制寄存器
PWMx 基地址+0x18	PWM_CH1V	32	Channel (1) 值
PWMx 基地址+0x1C	PWM_CH2SCR	32	Channel (2) 状态和控制寄存器
PWMx 基地址+0x20	PWM_CH2V	32	Channel (2) 值
PWMx 基地址+0x24	PWM_CH3SCR	32	Channel (3) 状态和控制寄存器
PWMx 基地址+0x28	PWM_CH3V	32	Channel (3) 值
PWMx 基地址+0x2C	PWM_CH4SCR	32	Channel (4) 状态和控制寄存器
PWMx 基地址+0x30	PWM_CH4V	32	Channel (4) 值
PWMx 基地址+0x34	PWM_CH5SCR	32	Channel (5) 状态和控制寄存器
PWMx 基地址+0x38	PWM_CH5V	32	Channel (5) 值
PWMx 基地址+0x3C	PWM_CH6SCR	32	Channel (6) 状态和控制寄存器
PWMx 基地址+0x40	PWM_CH6V	32	Channel (6) 值
PWMx 基地址+0x44	PWM_CH7SCR	32	Channel (7) 状态和控制寄存器
PWMx 基地址+0x48	PWM_CH7V	32	Channel (7) 值
PWMx 基地址+0x4C	PWM_CNTIN	32	计数器初始值寄存器
PWMx 基地址+0x50	PWM_STR	32	捕获和比较状态寄存器
PWMx 基地址+0x54	PWM_FUNCSEL	32	功能选择寄存器
PWMx 基地址+0x58	PWM_SYNC	32	同步寄存器
PWMx 基地址+0x5C	PWM_OUTINIT	32	通道输出的初始状态寄存器
PWMx 基地址+0x60	PWM_OMCR	32	输出屏蔽控制寄存器
PWMx 基地址+0x64	PWM_MODESEL	32	模式选择寄存器
PWMx 基地址+0x68	PWM_DTSET0	32	死区设置寄存器 0
PWMx 基地址+0x6C	PWM_DTSET1	32	死区设置寄存器 1
PWMx 基地址+0x70	PWM_EXTTRIG	32	外部触发器
PWMx 基地址+0x74	PWM_CHOPOLCR	32	通道输出极性控制寄存器
PWMx 基地址+0x78	PWM_FDSR	32	故障检测状态寄存器
PWMx 基地址+0x7C	PWM_CAPFILTER	32	输入捕获滤波器控制

© 2013 - 2024 杰发科技有限公司

地址	名称	宽度	描述
PWMx 基地址+0x80	PWM_FFAFER	32	故障滤波和使能寄存器
PWMx 基地址+0x84	PWM_QDI	32	正交解码控制和状态寄存器
PWMx 基地址+0x88	PWM_CONF	32	配置寄存器
PWMx 基地址+0x8C	PWM_FLTPOL	32	故障输入极性寄存器
PWMx 基地址+0x90	PWM_SYNCONF	32	同步配置寄存器
PWMx 基地址+0x94	PWM_INVCR	32	反相控制寄存器
PWMx 基地址+0x98	PWM_CHOSWCR	32	通道软件输出控制寄存器
PWMx 基地址+0x9C	PWM_DITHER0	32	抖动插入配置寄存器 0
PWMx 基地址+0xA0	PWM_DITHER1	32	抖动插入配置寄存器 1
PWMx 基地址+0xA4	PWM_DITHER2	32	抖动插入配置寄存器 2
PWMx 基地址+0xA8	PWM_DMACTRL	32	DMA 传输长度配置寄存器

说明：

上表中，x=0,1,2,3,4,5。

32.5.1 初始化寄存器(PWM_INIT)

表 32-6 PWM_INIT 寄存器

PWM_INIT 初始化寄存器 **复位值：0x00000001**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			PWM_RB_EN	OFUFDMAEN					CLKPSC[15:8]							
访问			RW	RW					RW							
复位值			0	0					0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CLKPSC[7:0]								CNTOF	CNTOIE	CNTMODE	CLKSRC		OFDIR	CUFE	COFE
访问	RW								R/W0C	RW	RW	RW		R/W0C	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

字段	说明
29 PWM_RB_EN	**内部回读功能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0: 禁止 1: 使能 该位仅应用于功能安全场景。使能该位后 PWM 内部会将 PWMn 各通道在内部连接到对应的 PWMn+1 各通道上，此时 PWMn 各通道可作为输出，PWMn+1 各通道可作为输入，可以在不进行外部连接的情况下测试 PWM 是否可以正常工作。
28 OFUFDMAEN	**上溢/下溢事件的 DMA 请求使能** 0：禁止 1：使能
23：8 CLKPSC	**PWM CLK 预分频器** 在将新值更新为寄存器位后，新的预分频因子会影响下一个总线时钟周期的时钟源。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时写入。 Clk Prescaler = CLKPSC + 1。
7 CNTOF	**定时器上溢/下溢标志** 0： PWM 计数器未溢出 1： PWM 计数器溢出 当 PWM 计数器值达到 MCVR 寄存器中值时或者当 PWM 计数器值达到 CNTIN 寄存器中值时置位。当 CNTOF 置 1 时，通过读取 INIT 寄存器，然后将 0 写入 CNTOF 位来清除 CNTOF 位。如果在读和写操作之间发生另一次 PWM 溢出， 则清除操作无效，CNTOF 仍保持置位。
6 CNTOIE	**定时器溢出中断使能** 0： 禁用 CNTOF 中断，使用软件轮询 1： 使能 CNTOF 中断，当 CNTOF 置位时，产生中断 使能 PWM 溢出中断
5 CNTMODE	**计数器工作模式** 0： PWM 计数器以向上计数模式工作 1： PWM 计数器以向上-向下计数模式工作 选择计数器工作模式。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时写入。
4：3 CLKSRC	**时钟源选择** 00：没有选择任何时钟。禁用 PWM 计数器 01：系统时钟 10：固定频率时钟 11：外部时钟 选择 PWM 计数器时钟源。该字段为写保护，它仅在 FUNCSEL[WPDIS] = 1 时可写。
2	**计数器上溢 / 下溢方向**

© 2013 - 2024 杰发科技有限公司

字段	说明
OFDIR	0：当计数到起始值时产生下溢标志 1：当计数达到最大值时产生上溢标志 该状态位写 0 清除
1 CUFE	**计数器下溢事件使能** 0：禁止 1：使能
0 COFE	**计数器上溢事件使能** 0：禁止 1：使能

32.5.2 计数器寄存器(PWM_CNT)

表 32- 7 PWM_CNT 寄存器

PWM_CNT **计数器寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	COUNT															
访问	RW															
复位值	0															

字段	说明
15：0 COUNT	**PWM 计数器的值** CNT 寄存器包含 PWM 计数器值。 Reset 清除 CNT 寄存器。将任何值写入 COUNT 都会使用其初始值 CNTIN 更新计数器。

© 2013 - 2024 杰发科技有限公司

32.5.3 最大计数值寄存器(PWM_MCVR)

表 32-8 PWM_MCVR 寄存器

PWM_MCVR 最大计数值寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	MCVR															
访问	RW															
复位值	0															

字段	说明
15：0 MCVR	**最大计数值寄存器** MCVR 寄存器包含 PWM 计数器的模数值。当 PWM 计数值达到 MCVR 值后，溢出标志 INIT [CNTOF] 在下一个时钟置起，计数器的下一个值取决于所选的计数方法。写入 MCVR 寄存器会将值锁存到缓冲区中。根据从写缓冲区更新的寄存器， MCVR 寄存器使用其写缓冲区的值进行更新。在写入 MCVR 寄存器之前，通过写入 CNT 来初始化 PWM 计数器。

32.5.4 通道状态和控制寄存器(PWM_CHnSCR)

表 32-9 PWM_CHnSCR 寄存器

PWM_CHnSCR Channel (n) 状态和控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								DMAEN	CHIF	CHIE	MSR1	MSR0	ELSR1	ELSR0	DIR	CHRSTEN
访问								RW	R/W0C	RW	RW	RW	RW	RW	RW	RW
复位值								0	0	0	0	0	0	0	o	0

字段	说明
8 DMAEN	**通道事件的 DMA 请求使能** 0：禁止

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：使能
7 CHIF	**通道中断标志** 0： 没有发生通道事件 1： 通道事件已经发生 在通道上发生事件时由硬件置位。通过读取 CHSCR 寄存器，然后将 CHIF 位写 0 来清除 CHIF 位。向 CHIF 中写 1 不起作用。如果在读取和写入操作间发生另一个事件，则清除操作无效，CHIF 仍保持置位。
6 CHIE	**通道中断使能** 0： 禁用通道中断 1： 使能通道中断 使能通道中断
5 MSR1	**通道模式选择寄存器 1** 用于通道逻辑的进一步选择，其功能取决于通道模式。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
4 MSR0	**通道模式选择寄存器 0** 用于进一步选择通道逻辑，其功能取决于通道模式。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
3 ELSR1	**边沿或电平选择寄存器 1** ELSR1 和 ELSR0 的功能依赖于通道模式。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
2 ELSR0	**边沿或电平选择寄存器 0** ELSR1 和 ELSR0 的功能依赖于通道模式。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
1 DIR	**匹配点方向** 0：在向下计数过程中匹配点生效 1：在向上计数过程中匹配点生效 仅用于向上-向下计数组合模式，用于选择匹配生效点方向。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
0 CHRSTEN	**通道事件对计数器的复位使能** 0：禁止 1：使能

© 2013 - 2024 杰发科技有限公司

32.5.5 通道值寄存器(PWM_CHnV)

表 32-10 PWM_CHnV 寄存器

PWM_CHnV **Channel (n) 值** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CHCVAL															
访问	RW															
复位值	0															

字段	说明
15：0 CHCVAL	**通道计数值** 这些寄存器包含输入模式的捕获 PWM 计数器值或输出模式的匹配值。在输入捕获和双边沿捕获模式下，忽略对 CHnV 寄存器的任何写入操作。在输出模式下，写入 CHnV 寄存器会将值锁存在缓冲区中。根据从写缓冲区更新的寄存器，使用其写缓冲区的值更新 CHnV 寄存器。

32.5.6 计数器初始值寄存器(PWM_CNTIN)

表 32-11 PWM_CNTIN 寄存器

PWM_CNTIN **计数器初始值寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CNTINIT															
访问	RW															
复位值	0															

字段	说明
15：0 CNTINIT	**计数器初始值** 计数器初始值寄存器包含 PWM 计数器的初始值。写入 CNTIN 寄存器会将值锁存至缓冲区中。根据从写缓冲区更新的寄存器，使用其写缓冲区的值更新 CNTIN 寄存器。当最初选择 PWM 时钟时，通过向 INIT[CLKSRC]位写入非 0 值，PWM 计数器以值 0x0000 开始。为避免这种行为，在第一次写入选择 PWM 时钟前，将新值写入 CNTIN 寄存器，然后通过在 CNT 寄存器中写入任意值，来初始化 PWM 计数器。

© 2013 - 2024 杰发科技有限公司

32.5.7 捕获和比较状态寄存器(PWM_STR)

表 32-12 PWM_STR 寄存器

PWM_STR 捕获和比较状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称						HALLA	HALLB	HALLC	CH7STS	CH6STS	CH5STS	CH4STS	CH3STS	CH2STS	CH1STS	CH0STS
访问						RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值						0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CH7SF	CH6SF	CH5SF	CH4SF	CH3SF	CH2SF	CH1SF	CH0SF
访问									R/W0C	R/W0C	R/W0C	R/W0C	R/W0C	R/W0C	R/W0C	R/W0C
复位值									0	0	0	0	0	0	0	0

字段	说明
26:24 HALL_STATUS (HALLC-HALLA)	**HALL 状态** 这些状态位表示了当前实际的 Hall 状态，主要是用于电机控制
23:16 CH7STS~CH0STS	**通道 0~7 状态** 对于通道输入，如果相应的滤波器使能，这些状态表示经过滤波器后的通道输入信号状态 对于通道输出，这些状态表示 PWM 通道最终输出信号状态
7 CH7SF	**通道 7 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
6 CH6SF	**通道 6 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
5 CH5SF	**通道 5 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
4 CH4SF	**通道 4 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
3 CH3SF	**通道 3 状态标志** 0： 没有发生通道事件

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： 已发生通道事件
2 CH2SF	**通道 2 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
1 CH1SF	**通道 1 状态标志** 0： 没有发生通道事件 1： 已发生通道事件
0 CH0SF	**通道 0 状态标志** 0： 没有发生通道事件 1： 已发生通道事件

32.5.8 功能选择寄存器(PWM_FUNCSEL)

表 32-13 PWM_FUNCSEL 寄存器

PWM_FUNCSEL **功能选择寄存器** **复位值：0x01000004**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称								PDYTPEN	CHxHIZEN							
访问								RW	RW							
复位值								1	0							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CHP3PNWEN	CHP2PNWEN	CHP1PNWEN	CHP0PNWEN			HALLEN	FAULTHIZEN	FAULTIE	FAULTMODE			PWMSYNC	WPDIS	INIT	PWMSYNCEN
访问	RW	RW	RW	RW			RW	RW	RW	RW			RW	RW	RW	RW
复位值	0	0	0	0			0	0	0	0			0	1	0	0

字段	说明
24 PDYTPEN	**组合中心对齐模式下的 PWM 占空比类型选择** 0：CHnV=CHn+1V=PWM_CNTIN, 输出 0%占空比; CHnV=CHn+1V=PWM_MCVR, 输出 100%占空比。 1：CHnV=CHn+1V=PWM_CNTIN, 输出 100%占空比; CHnV=CHn+1V=PWM_MCVR, 输出 0%占空比。
23:16 CHxHIZEN	**通道高阻态输出使能** 0：禁止高阻态输出

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：使能高阻态输出 当 FAULTHIZEN=1 时，对应的 GPIO 高阻态输出由 CHxHIZEN 或故障事件确定，CHxHIZEN=1 只控制一个通道输出高阻态；但是 FAULTHIZEN=1 的故障事件可以控制 8 个通道输出高阻态 该字段为写保护，它仅在 FUNCSEL[WPDIS] = 1 时可写。
15 CHP3PNWEN	**通道 6 / 通道 7 正负脉宽测量使能** 0：禁止 1：使能 该功能仅在双边沿捕获模式下使用，且此时要根据需要配置适合的 ELSR1:ELSR0 以及对应的 CHRSTEN 功能
14 CHP2PNWEN	**通道 4 / 通道 5 正负脉宽测量使能** 0：禁止 1：使能 该功能仅在双边沿捕获模式下使用，且此时要根据需要配置适合的 ELSR1:ELSR0 以及对应的 CHRSTEN 功能
13 CHP1PNWEN	**通道 2 / 通道 3 正负脉宽测量使能** 0：禁止 1：使能 该功能仅在双边沿捕获模式下使用，且此时要根据需要配置适合的 ELSR1:ELSR0 以及对应的 CHRSTEN 功能
12 CHP0PNWEN	**通道 0 / 通道 1 正负脉宽测量使能** 0：禁止 1：使能 该功能仅在双边沿捕获模式下使用，且此时要根据需要配置适合的 ELSR1:ELSR0 以及对应的 CHRSTEN 功能
9 HALLEN	**双边沿捕获模式下的 Hall 功能使能** 0：禁止 Hall 输入捕获功能 1：使能 Hall 输入捕获功能 为了正确的捕获 Hall 信号，用户应该通过 CH0SCR/CH1SCR 寄存器选择不同的边沿。例如：如果 CH0SCR 选择上升沿捕获，CH1SCR 应该选择下降沿捕获。如果 CH1SCR 选择上升沿捕获，CH0SCR 应该选择下降沿捕获。在选择的了正确的边沿后，捕获到的值会保存到对应的 CH0V/CH1V 寄存器中。
8 FAULTHIZEN	**使能故障事件产生时所有 8 个通道的高阻态输出。**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0：禁止 1：使能 该字段为写保护，它仅在 FUNCSEL[WPDIS] = 1 时可写。
7 FAULTIE	**错误中断使能** 0： 禁用故障控制中断 1： 使能故障控制中断 当 PWM 检测到故障，并使能 PWM 故障控制时，允许产生中断。
6：5 FAULTMODE	**故障控制模式** 00： 所有通道禁用故障控制. 01： 仅对偶数通道使能故障控制 (通道： 0, 2, 4 和 6)，所选模式为手动故障清除 10： 所有通道均支持故障控制，所选模式为手动故障清除 11： 所有通道均支持故障控制，所选模式为自动故障清除 定义 PWM 故障控制模式。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
3 PWMSYNC	**PWM 同步模式** 0： 没有限制，MCVR, CHnV, OMCR 和 PWM 计数器同步可以使用软件和硬件触发器 1： 软件触发器只能由 MCVR 和 CHnV 同步使用，硬件触发只能由 OMCR 和 PWM 计时器同步使用。 选择 MCVR，CHnV，OMCR 和 PWM 计数器同步可以使用的触发器。
2 WPDIS	**禁用写保护** 0：使能写保护 1：禁用写保护 只允许写 1，写 0 无效。当写保护使能时，具有写保护功能的位不能被写入。当写保护禁用时后，具有写保护功能的位可以写入。WPDIS 位是 WPEN 位取反。当 WPEN 位写入 1 时，可清除 WPDIS，即使能写保护。当 WPDIS 位写入 1 时，可清除 WPEN，即禁用写保护。
1 INIT	**初始化通道输出** 当 INIT 位写入 1 时，根据 OUTINIT 寄存器中相应位的状态，初始化通道输出。将 INIT 位置 0 不起作用。INIT 位始终读为 0。
0 PWMSYNCEN	**PWM 同步功能使能** 0：禁用寄存器同步功能 1：使能寄存器同步功能 该字段写保护，它仅能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.9 同步寄存器(PWM_SYNC)

© 2013 - 2024 杰发科技有限公司

表 32-14 PWM_SYNC 寄存器

PWM_SYNC 同步寄存器 复位值：0x00020000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															BFRGCHSEN	SYNC_BYPEN
访问															RW	RW
复位值															1	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					PWM_SYNCPOL				SWSYNC	TRIG2	TRIG1	TRIG0	OMSYNCP		MAXSYNCP	MINSYNCP
访问					RW				RW	RW	RW	RW	RW		RW	RW
复位值					0				0	0	0	0	0		0	0

字段	说明
17 BFRGCHSEN	CHnV/CNTIN/MCVR 缓存区和寄存器读取选择位 0：用户通过地址读到的是缓存区的值 1：用户通过地址读到的是寄存器的值
16 SYNC_BYPEN	**同步旁路功能** 0：写到 CHnV/CNTIN/MCVR 缓存区的值到寄存器的更新受同步机制的控制 1：写到 CHnV/CNTIN/MCVR 缓存区的值立即更新到寄存器，不受同步机制的控制。 CHnV/CNTIN/MCVR 同步旁路使能位
11 PWM_SYNCPOL	**PWM_SYNCPOL** 0： POL 寄存器在总线时钟的所有上升沿使用其缓冲区的值进行更新 1： POL 寄存器仅通过 PWM 同步单元更新其缓冲区的值 选择何时使用缓冲区的值更新 CHOPOLCR 寄存器。
7 SWSYNC	**PWM 同步软件触发器** 0 ： 未选择软件触发 1 ： 选择软件触发 选择软件触发器作为 PWM 同步触发器 。将 1 写入 SWSYNC 位时，会发生软件触发。
6 TRIG2	**PWM 同步硬件触发器 2** 0： 触发禁用 1： 触发使能 PWMn_FLT0 pin 输出作为 PWM 同步硬件触发源，输入信号检测到上升沿时，产生硬件触发。

© 2013 - 2024 杰发科技有限公司

字段	说明
5 TRIG1	**PWM 同步硬件触发器 1** 0： 触发禁用 1： 触发使能 CTU 模块的 SEL_PWMnSYNC 控制位作为 PWM 同步硬件触发源，输入信号中检测到上升沿时，产生硬件触发。
4 TRIG0	**PWM 同步硬件触发器 0** 0： 触发禁用 1： 触发使能 CTU 的 trigger output 作为同步的硬件触发源，输入信号中检测到上升沿时，产生硬件触发。
3 OMSYNCP	**输出掩码同步** 0：OMCR 寄存器在总线时钟的所有上升沿使用其缓冲区的值进行更新 1：OMCR 寄存器仅通过 PWM 同步单元更新其缓冲区的值 选择何时使用缓冲区的值更新 OMCR 寄存器。
1 MAXSYNCP	**使能最大加载点** 0： 禁用最大加载点 1： 使能最大加载点 选择 PWM 同步的最大加载点。如果 MAXSYNCP 为 1，则所选加载点为 PWM 计数器达到其最大值（MCVR 寄存器）时。
0 MINSYNCP	**使能最小加载点** 0: 禁用最小加载点 1: 使能最小加载点 选择 PWM 同步的最小加载点。如果 MINSYNCP 为 1，则所选加载点为 PWM 计数器达到其最小值（CNTIN 寄存器）时。

32.5.10 通道输出初始状态寄存器(PWM_OUTINIT)

© 2013 - 2024 杰发科技有限公司

表 32-15 PWM_OUTINIT 寄存器

PWM_OUTINIT 通道输出的初始状态寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	16	15	14	13	12	11	10	9	7	6	5	4	3	2	1	0
名称									CH7OIV	CH6OIV	CH5OIV	CH4OIV	CH3OIV	CH2OIV	CH1OIV	CH0OIV
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 CH7OIV	**通道 7 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
6 CH6OIV	**通道 6 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
5 CH5OIV	**通道 5 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
4 CH4OIV	**通道 4 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
3 CH3OIV	**通道 3 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
2	**通道 2 输出初始值**

© 2013 - 2024 杰发科技有限公司

字段	说明
CH2OIV	0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
1 CH1OIV	**通道 1 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值
0 CH0OIV	**通道 0 输出初始值** 0： 初始值为 0 1： 初始值为 1 进行初始化时，强制进入通道输出的值

32.5.11 输出屏蔽控制寄存器(PWM_OMCR)

表 32-16 PWM_OMCR 寄存器

PWM_OMCR **输出屏蔽控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CH7OMEN	CH6OMEN	CH5OMEN	CH4OMEN	CH3OMEN	CH2OMEN	CH1OMEN	CH0OMEN
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 CH7OMEN	**通道 7 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
6 CH6OMEN	**通道 6 输出屏蔽**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
5 CH5OMEN	**通道 5 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
4 CH4OMEN	**通道 4 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
3 CH3OMEN	**通道 3 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
2 CH2OMEN	**通道 2 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
1 CH1OMEN	**通道 1 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
0 CH0OMEN	**通道 0 输出屏蔽** 0： 通道输出没有被屏蔽，继续正常运行 1： 通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽

© 2013 - 2024 杰发科技有限公司

32.5.12 模式选择寄存器(PWM_MODESEL)

表 32-17 PWM_MODESEL 寄存器

PWM_MODESEL 模式选择寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PAIR3SYMEN	PAIR3FAULTEN	PAIR3SYNCEN	PAIR3DTEN	PAIR3DECAP	PAIR3DECAPEN	PAIR3COMPEN	PAIR3COMBINEN	PAIR2SYMEN	PAIR2FAULTEN	PAIR2SYNCEN	PAIR2DTEN	PAIR2DECAP	PAIR2DECAPEN	PAIR2COMPEN	PAIR2COMBINEN
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PAIR1SYMEN	PAIR1FAULTEN	PAIR1SYNCEN	PAIR1DTEN	PAIR1DECAP	PAIR1DECAPEN	PAIR1COMPEN	PAIR1COMBINEN	PAIR0SYMEN	PAIR0FAULTEN	PAIR0SYNCEN	PAIR0DTEN	PAIR0DECAP	PAIR0DECAPEN	PAIR0COMPEN	PAIR0COMBINEN
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31 PAIR3SYMEN	**通道对称使能(n = 6)** 0：禁止通道对称功能，此时用户应该分别配置 CHnV 和 CHn+1V 的值。 1：使能通道对称功能，此时用户只配置 CHnV,则 CHn+1V 会自动同步为 CHnV 的值
30 PAIR3FAULTEN	**故障控制使能（n = 6）** 0： 禁用此通道对中的故障控制 1： 使能此通道对中的故障控制 在通道 (n) 和 (n+1)间使能故障控制，该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
29 PAIR3SYNCEN	**同步使能（n = 6）** 0 ：禁用此通道对中的 PWM 同步 1： 使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
28 PAIR3DTEN	**死区使能（n = 6）**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 禁用此通道对中的死区插入 1： 使能此通道对中的死区插入 在 通道 (n) 和 (n+1)之间支持死区插入。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
27 PAIR3DECAP	**双边沿捕获模式捕获（n = 6）** 0： 双边沿捕获无效 1： 双边沿捕获有效 根据通道 (n) 输入事件和双边沿捕获位的配置，使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式，且发生捕获通道(n+1)的事件，则硬件将自动清除 DECAP 位。
26 PAIR3DECAPEN	**双边沿捕获模式使能（ n = 6）** 0：该通道对中的双沿捕捉模式禁用 1 ：该通道对中的双沿捕捉模式使能。 在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0，ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
25 PAIR3COMPEN	**通道互补（n = 6）** 0 ： 通道(n+1) 输出和通道(n)输出相同 1 ： 通道(n+1) 输出是通道(n)输出互补 使能组合通道的互补模式。在互补模式下，通道 (n+1) 输出是通道(n) 输出取反。该字节为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
24 PAIR3COMBINEN	**组合通道 （ n = 6）** 0 ：通道(n)和(n+1) 是独立的 1 ：通道(n)和(n+1) 是组合的 使能 通道 (n) 和 (n+1)的组合功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
23 PAIR2SYMEN	**通道对称使能(n = 4)** 0：禁止通道对称功能，此时用户应该分别配置 CHnV 和 CHn+1V 的值。 1：使能通道对称功能，此时用户只配置 CHnV,则 CHn+1V 会自动同步为 CHnV 的值
22 PAIR2FAULTEN	**故障控制使能（n = 4）** 0： 禁用此通道对中的故障控制 1： 使能此通道对中的故障控制 在通道 (n) 和 (n+1)间使能故障控制，该字段写保护，它只能在 FUNCSEL[WPDIS] = 1

© 2013 - 2024 杰发科技有限公司

字段	说明
	时可写。
21 PAIR2SYNCEN	**同步使能（n = 4）** 0：禁用此通道对中的 PWM 同步 1： 使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
20 PAIR2DTEN	**死区使能（n = 4）** 0： 禁用此通道对中的死区插入 1： 使能此通道对中的死区插入 在 通道 (n) 和 (n+1)之间支持死区插入。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
19 PAIR2DECAP	**双边沿捕获模式捕获（n = 4）** 0： 双边沿捕获无效 1： 双边沿捕获有效 根据通道 (n) 输入事件和双边沿捕获位的配置，使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式，且发生捕获通道(n+1)的事件，则硬件将自动清除 DECAP 位。
18 PAIR2DECAPEN	**双边沿捕获模式使能（n = 4）** 0：该通道对中的双沿捕捉模式禁用 1：该通道对中的双沿捕捉模式使能。 在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0，ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
17 PAIR2COMPEN	**通道互补（n = 4）** 0： 通道(n+1) 输出和通道(n)输出相同 1： 通道(n+1) 输出是通道(n)输出互补 使能组合通道的互补模式。在互补模式下，通道 (n+1) 输出是通道(n) 输出取反。该字节为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
16 PAIR2COMBINEN	**组合通道 （n = 4）** 0：通道(n)和(n+1) 是独立的 1：通道(n)和(n+1) 是组合的 使能 通道 (n) 和 (n+1)的组合功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
15	**通道对称使能(n = 2)**

© 2013 - 2024 杰发科技有限公司

字段	说明
PAIR1SYMEN	0：禁止通道对称功能，此时用户应该分别配置 CHnV 和 CHn+1V 的值。 1：使能通道对称功能，此时用户只配置 CHnV,则 CHn+1V 会自动同步为 CHnV 的值
14 PAIR1FAULTEN	**故障控制使能（n = 2）** 0：禁用此通道对中的故障控制 1：使能此通道对中的故障控制 使能 通道 (n) 和 (n+1)的故障控制。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
13 PAIR1SYNCEN	**同步使能（n = 2）** 0：禁用此通道对中的 PWM 同步 1：使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
12 PAIR1DTEN	**死区使能（n = 2）** 0：禁用此通道对中的死区插入 1：使能此通道对中的死区插入 使能通道 (n) 和 (n+1)的死区插入，该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
11 PAIR1DECAP	**双边沿捕获模式捕获（n = 2）** 0： 双边沿捕获无效 1： 双边沿捕获有效 根据通道 (n) 输入事件和双边沿捕获位的配置，使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式，且发生捕获通道(n+1)的事件，则硬件将自动清除 DECAP 位。
10 PAIR1DECAPEN	**双边沿捕获模式使能 （n = 2）** 0：该通道对中的双沿捕捉模式禁用 1：该通道对中的双沿捕捉模式使能。 在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0，ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
9 PAIR1COMPEN	**Channel (n) 互补（n = 2）** 0： 通道(n+1) 输出和通道(n)输出相同 1： 通道(n+1) 输出是通道(n)输出互补 使能组合通道的互补模式。在互补模式下，通道 (n+1) 输出是通道(n) 输出取反。该字节

© 2013 - 2024 杰发科技有限公司

字段	说明
	为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
8 PAIR1COMBINEN	**组合通道 （n = 2）** 0 ：通道(n)和(n+1) 是独立的 1 ：通道(n)和(n+1) 是组合的 使能通道 (n) 和 (n+1)的组合功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
7 PAIR0SYMEN	**通道对称使能(n = 0)** 0：禁止通道对称功能，此时用户应该分别配置 CHnV 和 CHn+1V 的值。 1：使能通道对称功能，此时用户只配置 CHnV,则 CHn+1V 会自动同步为 CHnV 的值
6 PAIR0FAULTEN	**故障控制使能（n = 0）** 0 ：禁用此通道对中的故障控制 1 ：使能此通道对中的故障控制 使能通道 (n) 和 (n+1)的故障控制。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
5 PAIR0SYNCEN	**同步使能 （n = 0）** 0 ：禁用此通道对中的 PWM 同步 1： 使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
4 PAIR0DTEN	**死区使能 （n = 0）** 0：禁用此通道对中的死区插入 1 ：使能此通道对中的死区插入 使能通道 (n)和(n+1)的死区插入，该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
3 PAIR0DECAP	**双边沿捕获模式捕获 （n = 0）** 0： 双边沿捕获无效 1： 双边沿捕获有效 根据通道 (n) 输入事件和双边沿捕获位的配置，使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式，且发生捕获通道(n+1)的事件，则硬件将自动清除 DECAP 位。
2 PAIR0DECAPEN	**双边沿捕获模式使能 （n = 0）** 0：该通道对中的双沿捕捉模式禁用 1 ：该通道对中的双沿捕捉模式使能

© 2013 - 2024 杰发科技有限公司

字段	说明
	在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0，ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。
1 PAIR0COMPEN	**Channel (n)互补（n = 0）** 0： 通道(n+1) 输出和通道(n)输出相同 1： 通道(n+1) 输出是通道(n)输出互补 使能组合通道的互补模式。在互补模式下，通道 (n+1) 输出是通道(n) 输出取反。该字节为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
0 PAIR0COMBINEN	**组合通道（n = 0）** 0： 通道(n)和(n+1) 是独立的 1： 通道(n)和(n+1) 是组合的 使能通道 (n) 和 (n+1)的组合功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.13 死区插入控制寄存器(PWM_DTSET0)

表 32- 18 PWM_DTSET 寄存器 0

PWM_DTSET0 **死区插入控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					DTPSC1		DTVAL1[9:0]									
访问					RW		RW									
复位值					0		0									
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					DTPSC0		DTVAL0[9:0]									
访问					RW		RW									
复位值					0		0									

字段	说明
27:26 DTPSC1	**死区预分频值(n = 2)** 0x： 总线时钟 1 分频 10： 总线时钟 4 分频 11： 总线时钟 16 分频 选择总线时钟分频因子。通道(n)和通道(n + 1)死区计数器使用此预分频时钟，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
25:16 DTVAL1	**死区值(n = 2)** 选择通道(n)和通道(n + 1)死区计数器的死区插入值 。

© 2013 - 2024 杰发科技有限公司

字段	说明
	死区插入值 = (DTPSC x DTVAL)。 DTVAL 选择插入的死区计数值，如下所示： 当 DTVAL 为 0 时，没有插入任何计数值 当 DTVAL 为 1 时，插入 1 个计数值 当 DTVAL 为 2 时，插入 2 个计数值 这种模式可持续到 1023 个计数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
11：10 DTPSC0	**死区预分频值(n = 0)** 0x： 总线时钟 1 分频 10： 总线时钟 4 分频 11： 总线时钟 16 分频 选择总线时钟分频因子。通道(n)和通道(n + 1)死区计数器使用此预分频时钟，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
9：0 DTVAL0	**死区值(n = 0)** 选择通道(n)和通道(n + 1)死区计数器的死区插入值 。 死区插入值 = (DTPSC x DTVAL)。 DTVAL 选择插入的死区计数值，如下所示： 当 DTVAL 为 0 时，没有插入任何计数值 当 DTVAL 为 1 时，插入 1 个计数值 当 DTVAL 为 2 时，插入 2 个计数值 这种模式可持续到 1023 个计数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.14 死区插入控制寄存器(PWM_DTSET1)

表 32-19 PWM_DTSET 寄存器 1

PWM_DTSET1 **死区插入控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					DTPSC3		DTVAL3[9:0]									
访问					RW		RW									
复位值					0		0									
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					DTPSC2		DTVAL2[9:0]									
访问					RW		RW									
复位值					0		0									

字段	说明
27:26 DTPSC3	**死区预分频值(n = 6)** 0x： 总线时钟 1 分频 10： 总线时钟 4 分频 11： 总线时钟 16 分频

© 2013 - 2024 杰发科技有限公司

字段	说明
	选择总线时钟分频因子。通道(n)和通道(n+1)死区计数器使用此预分频时钟，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
25:16 DTVAL3	**死区值(n = 6)** 选择通道(n)和通道(n+1)死区计数器的死区插入值 。 死区插入值 = (DTPSC x DTVAL)。 DTVAL 选择插入的死区计数值，如下所示： 当 DTVAL 为 0 时，没有插入任何计数值 当 DTVAL 为 1 时，插入 1 个计数值 当 DTVAL 为 2 时，插入 2 个计数值 这种模式可持续到 1023 个计数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
11：10 DTPSC2	**死区预分频值(n = 4)** 0x： 总线时钟 1 分频 10： 总线时钟 4 分频 11： 总线时钟 16 分频 选择总线时钟分频因子。通道(n)和通道(n+1)死区计数器使用此预分频时钟，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
9：0 DTVAL2	**死区值(n = 4)** 选择通道(n)和通道(n+1)死区计数器的死区插入值 。 死区插入值 = (DTPSC x DTVAL)。 DTVAL 选择插入的死区计数值，如下所示： 当 DTVAL 为 0 时，没有插入任何计数值 当 DTVAL 为 1 时，插入 1 个计数值 当 DTVAL 为 2 时，插入 2 个计数值 这种模式可持续到 1023 个计数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

© 2013 - 2024 杰发科技有限公司

32.5.15 外部触发器寄存器(PWM_EXTTRIG)

表 32-20 PWM_EXTTRIG 寄存器

PWM_EXTTRIG 外部触发器寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		TRIGRATIO				MAXTRIGEN	TRIGF	INITTRIGEN	CH7TRIG	CH6TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG	CH1TRIG	CH0TRIG
访问		RW				RW	R/W0C	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值		0				0	0	0	0	0	0	0	0	0	0	0

字段	说明
14:12 TRIGRATIO	**触发比例** 0~7：每 1~8 个 PWM 周期产生一次起始值/最大值/匹配触发输出
10 MAXTRIGEN	**使能最大值触发器** 0：禁止 1：使能
9 TRIGF	**通道触发标志** 0： 没有生成通道触发 1： 产生通道触发 任意某个通道触发时置 1。 通过读取 EXTTRIG，然后将 0 写入 TRIGF 可清除 。如果在清除序列完成之前生成另一个通道触发，则清除操作无效，TRIGF 仍保持置位。
8 INITTRIGEN	**使能初始化触发器** 0： 禁用初始化触发 1： 使能初始化触发 当 PWM 计数器等于 CNTIN 寄存器时，允许产生触发。
7 CH7TRIG	**通道 7 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。

© 2013 - 2024 杰发科技有限公司

字段	说明
6 CH6TRIG	**通道 6 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
5 CH5TRIG	**通道 5 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
4 CH4TRIG	**通道 4 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
3 CH3TRIG	**通道 3 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
2 CH2TRIG	**通道 2 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
1 CH1TRIG	**通道 1 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。
0 CH0TRIG	**通道 0 触发使能** 0：禁用通道触发 1：使能通道触发 当 PWM 计数器等于 CHnV 寄存器时， 允许产生通道触发。

32.5.16 通道输出极性控制寄存器(PWM_CHOPOLCR)

© 2013 - 2024 杰发科技有限公司

表 32-21 PWM_CHOPOLCR 寄存器

PWM_CHOPOLCR 通道输出极性控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CH7 POL	CH6 POL	CH5 POL	CH4 POL	CH3 POL	CH2 POL	CH1 POL	CH0 POL
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 CH7POL	**通道 7 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
6 CH6POL	**通道 6 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
5 CH5POL	**通道 5 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
4 CH4POL	**通道 4 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
3 CH3POL	**通道 3 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
2	**通道 2 极性**

© 2013 - 2024 杰发科技有限公司

字段	说明
CH2POL	
	0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
1 CH1POL	**通道 1 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
0 CH0POL	**通道 0 极性** 0： 通道极性为高电平有效 1： 通道极性为低电平有效 定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.17 故障检测状态寄存器(PWM_FDSR)

表 32-22 PWM_FDSR 寄存器

PWM_FDSR **故障检测状态寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									FAULTDF	WPEN	FAULTIN		FAULTDF3	FAULTDF2	FAULTDF1	FAULTDF0
访问									R/W0C	RW	RO		R/W0C	R/W0C	R/W0C	R/W0C
复位值									0	0	0		0	0	0	0

字段	说明
7 FAULTDF	**故障检测标志** 0： 未检测到故障 1： 检测到故障 FAULTDF j 位的逻辑或结果，其中 j = 3，2, 1, 0。在 FAULTDF 置位的情况下，当故障输入端不存在故障条件时，读取故障状态寄存器，向 FAULTDF 写入 0 可清零 FAULTDF。如果在清

© 2013 - 2024 杰发科技有限公司

字段	说明
	除序列完成前在一个已使能的故障输入中检测到另一个故障，则清除操作无效，FAULTDF 仍保持置位状态。当 FAULTDF j 位全部清除后，FAULTDF 也会被清 0。
6 WPEN	**写保护使能** 0： 禁用写保护， 写保护位可写 1： 使能写保护，写保护位不可写 只允许写 1，写 0 无效。当写保护使能时，具有写保护功能的位不能被写入。当写保护禁用时后，具有写保护功能的位可以写入。WPEN 位是 WPDIS 位取反。当 WPDIS 位写入 1 时，可清除 WPEN，即禁用写保护。当 WPEN 位写入 1 时，可清除 WPDIS，即使能写保护。
5 FAULTIN	**故障输入** 0： 使能故障输入的逻辑或为 0 1： 使能故障输入的逻辑或为 1 表示当使能故障控制后，在其过滤器（若已启用过滤器）后使能的故障输入的逻辑或。
3 FAULTDF3	**故障检测标志 3** 0：未检测到故障 1： 检测到故障 在 FAULTDF3 置 1 时，读取 FDSR 寄存器然后往 FAULTDF3 写 0 可清除。当 FAULTDF 位被清 0 时，FAULTDF3 位也会被清 0 。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除操作无效， FAULTDF3 仍保持置位状态。
2 FAULTDF2	**故障检测标志 2** 0：未检测到故障 1： 检测到故障 在 FAULTDF2 置 1 时，读取 FDSR 寄存器然后往 FAULTDF2 写 0 可清除。当 FAULTDF 位被清 0 时，FAULTDF2 位也会被清 0 。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除操作无效， FAULTDF2 仍保持置位状态。
1 FAULTDF1	**故障检测标志 1** 0： 未检测到故障 1： 检测到故障 在 FAULTDF1 置 1 时，读取 FDSR 寄存器然后往 FAULTDF1 写 0 可清除。当 FAULTDF 位被清 0 时，FAULTDF1 位也会被清 0 。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除操作无效， FAULTDF1 仍保持置位状态。
0 FAULTDF0	**故障检测标志 0** 0： 未检测到故障 1： 检测到故障

© 2013 - 2024 杰发科技有限公司

字段	说明
	在 FAULTDF0 置 1 时，读取 FDSR 寄存器然后往 FAULTDF0 写 0 可清除。当 FAULTDF 位被清 0 时，FAULTDF0 位也会被清 0 。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除无效， FAULTDF0 将保持置位状态。

32.5.18 输入捕获滤波器控制(PWM_CAPFILTER)

表 32- 23 PWM_CAPFILTER 寄存器

PWM_CAPFILTER 输入捕获滤波器控制 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					CAPFPSC[3:0]									CH3CAPFVAL [3:1]		
访问					RW									RW		
复位值					0									0		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CH3CAPFVAL [0:0]		CH2CAPFVAL [3:0]					CH1CAPFVAL [3:0]					CH0CAPFVAL [3:0]			
访问	RW		RW					RW					RW			
复位值	0		0					0					0			

字段	说明
27:24 CAPFPSC	**通道输入滤波器以及故障输入滤波器分频** 0000b: 1 分频 0001b: 2 分频 0010b: 4 分频 0011b: 8 分频 0100b: 16 分频 0101b: 32 分频 0110b: 64 分频 0111b: 128 分频 1000b: 256 分频 1001b: 1024 分频 1010b: 2048 分频 1011b: 4096 分频 其他：默认 1 分频
18：15 CH3CAPFVAL	**通道 3 输入滤波器** 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
13：10 CH2CAPFVAL	**通道 2 输入滤波器**

© 2013 - 2024 杰发科技有限公司

字段	说明
	选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
8：5 CH1CAPFVAL	**通道 1 输入滤波器** 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
3：0 CH0CAPFVAL	**通道 0 输入滤波器** 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。

32.5.19 故障滤波器和故障使能寄存器(PWM_FFAFER)

表 32- 24 PWM_FFAFER 寄存器

PWM_FFAFER **故障滤波器和故障使能寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					FFVAL[3:0]				FF3EN	FF2EN	FF1EN	FF0EN	FER3EN	FER2EN	FER1EN	FER0EN
访问					RW				RW	RW	RW	RW	RW	RW	RW	RW
复位值					0				0	0	0	0	0	0	0	0

字段	说明
11：8 FFVAL	**故障输入滤波器** 0: 禁用故障滤波器 1~15: 滤波值为 1~15 选择用于故障输入的滤波器值，当值为 0 时禁用故障滤波器。
7 FF3EN	**故障输入 3 滤波器使能** 0： 禁用故障输入滤波器 1： 使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
6 FF2EN	**故障输入 2 滤波器使能** 0： 禁用故障输入滤波器 1： 使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
5	**故障输入 1 滤波器使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
FF1EN	0： 禁用故障输入滤波器 1： 使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
4 FF0EN	**故障输入 0 滤波器使能** 0： 禁用故障输入滤波器 1： 使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
3 FER3EN	**故障输入 3 使能** 0：禁用故障输入 1：使能故障输入 使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
2 FER2EN	**故障输入 2 使能** 0：禁用故障输入 1：使能故障输入 使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
1 FER1EN	**故障输入 1 使能** 0：禁用故障输入 1：使能故障输入 使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
0 FER0EN	**故障输入 0 使能** 0：禁用故障输入 1：使能故障输入 使能故障输入。该字段为写保护，它只能在 FUNCSELWPDIS] = 1 时可写。

© 2013 - 2024 杰发科技有限公司

32.5.20 正交解码器接口配置寄存器(PWM_QDI)

表 32-25 PWM_QDI 寄存器

PWM_QDI 正交解码器接口配置寄存器 复位值：0x00000004

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					PHZSTS		PHZPOL	PHZRSTEN			PHAPOL	PHBPOL	QUADMODE	QUADIR	CNTOFDIR	QDIEN
访问					R/W 0C		RW	RW			RW	RW	RW	RO	RO	RW
复位值					0		0	0			0	0	0	1	0	0

字段	说明
11 PHZSTS	**Z 相位状态** 0：Z 相位事件没有产生 1：Z 相位事件已经产生，写 0 清除
9 PHZPOL	**Z 相位的输入极性** 0：正常极性。在识别 Z 相输入信号的上升沿和下降沿之前，该信号未反相 1：反转极性。在识别 Z 相输入信号的上升沿和下降沿之前，该信号反相 选择正交解码器 Z 相输入的极性。
8 PHZRSTEN	**Z 相位事件对 PWM 计数器的复位功能** 0：禁止 1：使能
5 PHAPOL	**A 相 输入极性** 0：正常极性。在识别 A 相输入信号的上升沿和下降沿之前，该信号未反相 1：反转极性。在识别 A 相输入信号的上升沿和下降沿之前，该信号反相 选择正交解码器 A 相输入的极性。
4 PHBPOL	**B 相 输入极性** 0：正常极性。在识别 B 相输入信号的上升沿和下降沿之前，该信号未反相 1：反转极性。在识别 B 相输入信号的上升沿和下降沿之前，该信号反相 选择正交解码器 B 相输入的极性
3	**正交解码器模式**

© 2013 - 2024 杰发科技有限公司

字段	说明
QUADMODE	0： A 相和 B 相编码模式 1： 计数和方向编码模式 选择正交解码器模式下使用的编码模式。
2 QUADIR	**正交解码器模式下的 PWM 计数器方向** 0： 计数方向为降低 (PWM 计数器递减). 1： 计数方向为增加 (PWM 计数器递增) 表示计数方向。
1 CNTOFDIR	**正交解码器模式下的定时器溢出方向** 0： CNTOF 位设置在计数的底部 。 PWM 计数器递减，PWM 计数器从其最小值（CNTIN 寄存器）变为最大值（MCVR 寄存器） 1： CNTOF 位设置在计数的顶部 。 PWM 计数器递增，PWM 计数器从其最大值（MCVR 寄存器）变为最小值（CNTIN 寄存器） 表示 CNTOF 位是否设置在计数的顶部或底部 。
0 QDIEN	**正交解码器模式使能** 0：禁用正交解码器模式 1：使能正交解码器模式 使能正交解码器模式。在此模式下，A 相和 B 相输入信号控制 PWM 计数器方向。正交解码器模式优先于其他模式。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.21 配置寄存器(PWM_CONF)

表 32- 26 PWM_CONF 寄存器

PWM_CONF **配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	EVENT7PSC		EVENT6PSC		EVENT5PSC		EVENT4PSC		EVENT3PSC		EVENT2PSC		EVENT1PSC		EVENT0PSC	
访问	RW		RW		RW		RW		RW		RW		RW		RW	
复位值	0		0		0		0		0		0		0		0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称						GTBEOUT	GTBEEN					CNTOFNUM				
访问						RW	RW					RW				
复位值						0	0					0				

© 2013 - 2024 杰发科技有限公司

字段	说明
31：30 EVENT7PSC	**通道 7 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
29：28 EVENT6PSC	**通道 6 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
27：26 EVENT5PSC	**通道 5 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
25：24 EVENT4PSC	**通道 4 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
23：22 EVENT3PSC	**通道 3 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
21：20 EVENT2PSC	**通道 2 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
19：18 EVENT1PSC	**通道 1 输入事件预分频** 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
17：16 EVENT0PSC	**通道 0 输入事件预分频** 00: 1 个输入事件触发一次捕获

© 2013 - 2024 杰发科技有限公司

字段	说明
	01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
10 GTBEOUT	**全局时基输出** 0：禁用全局时基信号生成 1：使能全局时基信号生成 使能到其它 PWM 的全局时基信号生成。
9 GTBEEN	**全局时基使能** 0 ：禁用外部全局时基 1：使能外部全局时基
4：0 CNTOFNUM	**CNTOF 频率** 选择计数器溢出次数与 CNTOF 位置 1 次数之间的比率 CNTOFNUM = 0： 每个计数器溢出都设置 CNTOF 位 CNTOFNUM = 1：CNTOF 位设置为第一次计数器溢出，但不是下一次溢出 CNTOFNUM = 2：CNTOF 位设置为第一次计数器溢出，但不会设置为接下来的两次溢出 CNTOFNUM = 3：CNTOF 位设置为第一次计数器溢出，但不会设置为接下来的三次溢出

32.5.22 故障输入极性寄存器(PWM_FLTPOL)

表 32-27 PWM_FLTPOL 寄存器

PWM_FLTPOL **故障输入极性控制器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													FLT3POL	FLT2POL	FLT1POL	FLT0POL
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3 FLT3POL	**故障输入 3 极性** 0： 故障输入极性为高电平有效 。故障输入处的 1 表示一个故障。 1： 故障输入极性为低电平有效 。故障输入处的 0 表示一个故障。

© 2013 - 2024 杰发科技有限公司

字段	说明
	定义故障输入极性。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
2 FLT2POL	**故障输入 2 极性** 0： 故障输入极性为高电平有效 。故障输入处的 1 表示一个故障。 1： 故障输入极性为低电平有效 。故障输入处的 0 表示一个故障。 定义故障输入极性。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
1 FLT1POL	**故障输入 1 极性** 0： 故障输入极性为高电平有效 。故障输入处的 1 表示一个故障。 1： 故障输入极性为低电平有效 。故障输入处的 0 表示一个故障。 定义故障输入极性。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
0 FLT0POL	**故障输入 0 极性** 0： 故障输入极性为高电平有效 。故障输入处的 1 表示一个故障。 1： 故障输入极性为低电平有效 。故障输入处的 0 表示一个故障。 定义故障输入极性。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

32.5.23 同步配置寄存器(PWM_SYNCONF)

表 32-28 PWM_SYNCONF 寄存器

PWM_SYNCONF **同步配置寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称										HWPOL	SWPOL	SWVHWSYNC	INVHWSYNC	OMVHWSYNC	PWMSVHWSYNC	CNTVHWSYNC
访问										RW	RW	RW	RW	RW	RW	RW
复位值										0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				SWVSWSYNC	INVSWSYNC	OMVSWSYNC	PWMSVSWSYNC	CNTVSWSYNC	SYNCMODE		SWOC	INVC		CNTINC		HWTRIGMODE
访问				RW	RW	RW	RW	RW	RW		RW	RW		RW		RW
复位值				0	0	0	0	0	0		0	0		0		0

© 2013 - 2024 杰发科技有限公司

字段	说明
22 HWPOL	**通道 CHPOLCR 同步由硬件触发器激活** 0：硬件触发器不会激活 CHPOLCR 寄存器同步 1：硬件触发器激活 CHPOLCR 寄存器同步
21 SWPOL	**通道 CHPOLCR 同步由软件触发器激活** 0： 软件触发器不会激活 CHPOLCR 寄存器同步 1： 软件触发器激活 CHPOLCR 寄存器同步
20 SWVHWSYNC	**软件输出控制同步由硬件触发器激活** 0： 硬件触发器不会激活 CHOSWCR 寄存器同步 1 ：硬件触发器激活 CHOSWCR 寄存器同步
19 INVHWSYNC	**通过硬件触发器激活反相控制同步** 0： 硬件触发器不会激活 INVCR 寄存器同步 1： 硬件触发器激活 INVCR 寄存器同步
18 OMVHWSYNC	**通过硬件触发器激活输出屏蔽同步** 0 ： 硬件触发器不会激活 OMCR 寄存器同步 1： 硬件触发器激活 OMCR 寄存器同步
17 PWMSVHWSYNC	**通过硬件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步** 0 ： 硬件寄存器不会激活 MCVR, CNTIN 和 CHV 寄存器同步 1： 硬件寄存器激活 MCVR, CNTIN 和 CHV 寄存器同步
16 CNTVHWSYNC	**通过硬件触发器激活 PWM 计数器同步** 0 ： 硬件触发器不会激活 PWM 计数器同步 1 ： 硬件触发器激活 PWM 计数器同步
12 SWVSWSYNC	**通过软件触发器激活软件输出控制同步** 0： 软件触发器不会激活 CHOSWCR 寄存器同步. 1： 软件触发器激活 CHOSWCR 寄存器同步
11 INVSWSYNC	**通过软件触发器激活反相控制同步** 0 ： 软件触发器不会激活 INVCR 寄存器同步. 1 ： 软件触发器激活 INVCR 寄存器同步
10 OMVSWSYNC	**通过软件触发器激活输出掩码同步** 0 ： 软件触发器不会激活 OMCR 寄存器同步 1： 软件触发器激活 OMCR 寄存器同步
9 PWMSVSWSYNC	**通过软件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步** 0 : 软件触发器不会激活 MCVR, CNTIN 和 CHV 寄存器同步 1：软件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步.
8	**通过软件触发器激活 PWM 寄存器同步**

© 2013 - 2024 杰发科技有限公司

字段	说明
CNTVSWSYNC	0： 软件触发器不会激活 PWM 计数器同步 1： 软件触发器激活 PWM 计数器同步
7 SYNCMODE	**同步模式** 0： 选择传统 PWM 同步 1： 选择增强 PWM 同步 选择 PWM 同步模式
5 SWOC	**CHOSWCR 寄存器同步** 0： 在总线时钟的所有上升沿，使用其缓冲区的值更新 CHOSWCR 寄存器 1： 通过 PWM 同步，使用其缓冲区的值更新 CHOSWCR 寄存器
4 INVC	**INVCR 寄存器同步** 0：在总线时钟的所有上升沿，使用其缓冲区的值更新 INVCR 寄存器 1：通过 PWM 同步，使用其缓冲区的值更新 INVCR 寄存器
2 CNTINC	**CNTIN 寄存器同步** 0：在总线时钟的所有上升沿，使用其缓冲区的值更新 CNTIN 寄存器 1： 通过 PWM 同步，使用其缓冲区的值更新 CNTIN 寄存器
0 HWTRIGMODE	**硬件触发模式** 0： 当检测到硬件触发器 j 时，PWM 清零 TRIGj 位 ， 其中 j = 0, 1,2. 1： 当检测到硬件触发器 j 时，PWM 不会清零 TRIGj 位 ， 其中 j = 0, 1,2.

32.5.24 反相控制寄存器(PWM_INVCR)

表 32- 29 PWM_INVCR 寄存器

PWM_INVCR **反相控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													PAIR 3INV EN	PAIR 2INV EN	PAIR 1INV EN	PAIR 0INV EN
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3	**配对通道 3 反相使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
PAIR3INVEN	0： 禁用反相 1： 使能反相
2 PAIR2INVEN	**配对通道 2 反相使能** 0： 禁用反相 1： 使能反相
1 PAIR1INVEN	**配对通道 1 反相使能** 0： 禁用反相 1： 使能反相
0 PAIR0INVEN	**配对通道 0 反相使能** 0： 禁用反相 1： 使能反相

32.5.25 通道软件输出控制寄存器(PWM_CHOSWCR)

表 32- 30 PWM_CHOSWCR 寄存器

PWM_CHOSWCR **通道 软件输出控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CH7SWCV	CH6SWCV	CH5SWCV	CH4SWCV	CH3SWCV	CH2SWCV	CH1SWCV	CH0SWCV	CH7SWEN	CH6SWEN	CH5SWEN	CH4SWEN	CH3SWEN	CH2SWEN	CH1SWEN	CH0SWEN
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15 CH7SWCV	**通道 7 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
14 CH6SWCV	**通道 6 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
13 CH5SWCV	**通道 5 软件输出控制值**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
12 CH4SWCV	**通道 4 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
11 CH3SWCV	**通道 3 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
10 CH2SWCV	**通道 2 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
9 CH1SWCV	**通道 1 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
8 CH0SWCV	**通道 0 软件输出控制值** 0： 软件输出控制将通道输出强制置为 0 1： 软件输出控制将通道输出强制置为 1
7 CH7SWEN	**通道 7 软件输出控制使能** 0： 通道输出不受软件输出控制的影响 1 ：通道输出受软件输出控制的影响
6 CH6SWEN	**通道 6 软件输出控制使能** 0： 通道输出不受软件输出控制的影响 1 ：通道输出受软件输出控制的影响
5 CH5SWEN	**通道 5 软件输出控制使能** 0： 通道输出不受软件输出控制的影响 1 ：通道输出受软件输出控制的影响
4 CH4SWEN	**通道 4 软件输出控制使能** 0： 通道输出不受软件输出控制的影响 1 ：通道输出受软件输出控制的影响
3 CH3SWEN	**通道 3 软件输出控制使能** 0： 通道输出不受软件输出控制的影响 1 ：通道输出受软件输出控制的影响
2 CH2SWEN	**通道 2 软件输出控制使能** 0： 通道输出不受软件输出控制的影响

© 2013 - 2024 杰发科技有限公司

字段	说明
	1：通道输出受软件输出控制的影响
1 CH1SWEN	**通道 1 软件输出控制使能** 0：通道输出不受软件输出控制的影响 1：通道输出受软件输出控制的影响
0 CH0SWEN	**通道 0 软件输出控制使能** 0：通道输出不受软件输出控制的影响 1：通道输出受软件输出控制的影响

32.5.26 通道抖动控制寄存器 0(PWM_DITHER0)

表 32- 31 PWM_DITHER0 寄存器

PWM_DITHER0 **通道抖动控制寄存器 0** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				C3DHR								C2DHR				
访问				RW								RW				
复位值				0								0				
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				C1DHR								C0DHR				
访问				RW								RW				
复位值				0								0				

字段	说明
28:24 C3DHR	**通道 3 抖动值或分数匹配值** 这些位实际上表示的是 CH3V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C3DHR 从缓存区到寄存器的更新条件与 CH3V 寄存器是一样的。 例如：如果用户想要配置匹配值为 15.3，那么 CH3V 为 15， C3DHR = round (32 * 0.3)-1 =9，该功能仅用于所有的输出模式。
20:16 C2DHR	**通道 2 抖动值或分数匹配值** 这些位实际上表示的是 CH2V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C2DHR 从缓存区到寄存器的更新条件与 CH2V 寄存器是一样的。
12:8 C1DHR	**通道 1 抖动值或分数匹配值** 这些位实际上表示的是 CH1V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C1DHR 从缓存区到寄存器的更新条件与 CH1V 寄存器是一样的。
4:0	**通道 0 抖动值或分数匹配值**

© 2013 - 2024 杰发科技有限公司

字段	说明
C0DHR	这些位实际上表示的是 CH0V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C0DHR 从缓存区到寄存器的更新条件与 CH0V 寄存器是一样的。

32.5.27 通道抖动控制寄存器 1(PWM_DITHER1)

表 32- 32 PWM_DITHER1 寄存器

PWM_DITHER1 通道抖动控制寄存器 1 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				C7DHR								C6DHR				
访问				RW								RW				
复位值				0								0				
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				C5DHR								C4DHR				
访问				RW								RW				
复位值				0								0				

字段	说明
28:24 C7DHR	**通道 7 抖动值或分数匹配值** 这些位实际上表示的是 CH7V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C7DHR 从缓存区到寄存器的更新条件与 CH7V 寄存器是一样的。
20:16 C6DHR	**通道 6 抖动值或分数匹配值** 这些位实际上表示的是 CH6V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C6DHR 从缓存区到寄存器的更新条件与 CH6V 寄存器是一样的。
12:8 C5DHR	**通道 5 抖动值或分数匹配值** 这些位实际上表示的是 CH5V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C5DHR 从缓存区到寄存器的更新条件与 CH5V 寄存器是一样的。
4:0 C4DHR	**通道 4 抖动值或分数匹配值** 这些位实际上表示的是 CH4V 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 C4DHR 从缓存区到寄存器的更新条件与 CH4V 寄存器是一样的。

© 2013 - 2024 杰发科技有限公司

32.5.28 通道抖动控制寄存器 2(PWM_DITHER2)

表 32-33 PWM_DITHER2 寄存器

PWM_DITHER2 通道抖动控制寄存器 2 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DEBUGCTRL															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												PDHR				
访问												RW				
复位值												0				

字段	说明
31:30 DEBUGCTRL	**调试模式** 00：对于计数，通道输出，CNTIN/MCVR/CHnV/CHn+1V 缓存区和寄存器都没有影响。 01：计数停止，通道输出保持进入调试模式前的电平状态，写 CNTIN/MCVR/CHnV/CHn+1V 缓存区会立即更新到寄存器中。 1x：计数停止，通道输出安全电平，写 CNTIN/MCVR/CHnV/CHn+1V 缓存区会立即更新到寄存器中。
4:0 PDHR	**周期抖动值或分数周期值** 这些位实际上表示的是 MCVR 的分数值，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。 PDHR 从缓存区到寄存器的更新条件与 MCVR 寄存器是一样的。

© 2013 - 2024 杰发科技有限公司

32.5.29 PWM DMA 控制寄存器(PWM_DMACTRL)

表 32-34 PWM_DMACTRL 寄存器

PWM_DMACTRL　　DMA 控制寄存器　　复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DMA_TRANS_EN			DMA_TRANS_LEN				
访问									RW			RW				
复位值									0			0				

字段	说明
7 DMA_TRANS_EN	**PWM DMA_TRANS_LEN 使能** 0：禁止 1：使能
4:0 DMA_TRANS_LEN	**DMA 传输长度** 0~31：1~32 传输长度。 该寄存器主要是用于配置一次 PWM DMA Req 请求清除的 DMA 传输长度 通常该长度会少于 DMA 传输的总长度。该配置仅当 DMA_TRANS_EN 使能时有效。

© 2013 - 2024 杰发科技有限公司

33 可编程延时定时器(PDT)

33.1 简介

可编程延迟定时器(Programmable Delay Timer，PDT)提供了从触发信号输入到触发信号输出间的可控延迟。 AC7840x 包含 2 个 PDT 模块。

33.2 特性

- 计数模式为向上计数的 16 位计数器
- 支持硬件触发或软件触发
- 支持一个触发输入
- 8 个可独立配置的延时值作用于一个触发输出
- 支持两种工作模式： 单次模式或者连续模式
- 可编程的脉宽输出
- 1 个可编程的延时中断

© 2013 - 2024 杰发科技有限公司

33.3 结构框图

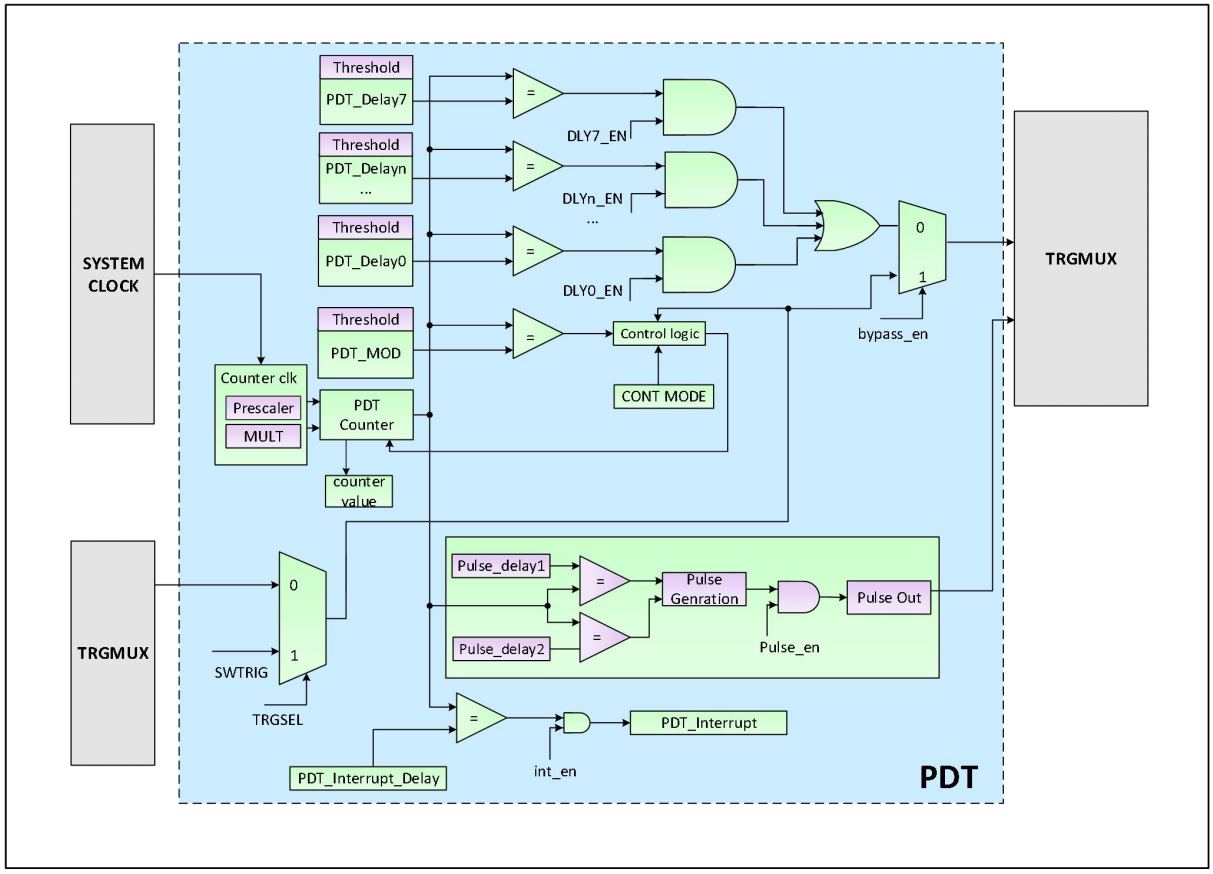


图 33- 1 PDT 结构框图

33.4 功能描述

33.4.1 计数器时钟

PDT 计数时钟等于外设功能时钟除以预分频因子 PDT_SC[PRE]与乘法因子 PDT_SC[MULT]的乘积，即： PDT 计数器时钟 = (外设功能时钟) / (预分频器预分频因子 * 预分频器的乘法因子)。

- 3-bit 预分频器 PDT_SC[PRE]的预分频因子：
  - 0h：预分频因子为 1
  - 1h：预分频因子为 2
  - 2h：预分频因子为 4
  - 3h：预分频因子为 8
  - 4h：预分频因子为 16

© 2013 - 2024 杰发科技有限公司

- 5h：预分频因子为 32
- 6h：预分频因子为 64
- 7h：预分频因子为 128

- 2-bit 预分频器的乘法因子 PDT_SC[MULT]：
  - 0h： MULT 为 1
  - 1h： MULT 为 10
  - 2h： MULT 为 20
  - 3h： MULT 为 40

33.4.2 计数器

- PDT 计数器是一个 16 位、计数模式为向上计数的计数器
- PDT 计数器可通过 PDT_SC[PDTEN]控制位来启用或者禁用
- PDT 模块在接收到所选的触发信号或者软件触发信号的上升沿时，计数器 PDT_CNT[CNT]从 0 开始计数
- 计数器值 PDT_CNT[CNT]达到计数的最大值时(PDT_MOD[MOD] + 1)，计数值 PDT_CNT[CNT]复位为 0

33.4.3 模数值

- 16-bit 的模数值寄存器
- 当计数值 PDT_CNT[CNT]等于(PDT_MOD[MOD] + 1)时，计数值复位为 0
- 周期 = (PDT_MOD[MOD] + 1) / 计数频率
- 模数值 PDT_MOD[MOD]的默认值为 0xFFFF

33.4.4 可编程延时

PDT_SC[BYPASS_EN] = 0 时，PDT 模块处于延时功能使能模式，当 PDT 被触发时计数器从 0 开始计数，当计数值 PDT_CNT[CNT]等于 PDT_DLYn[DLYn](n=0，1，…7)时产生触发输出信号，其中 PDT_DLYn[DLYn] (n=0，1，…7)为 16 位的可编程延时寄存器，不同的延时通道可通过配置 PDT_DLYn_EN 寄存器单独使能或者禁用。

PDT_SC[BYPASS_EN] = 1 时，PDT 模块处于延时功能禁用模式，当 PDT 被触发时将立即输出触发信号。

© 2013 - 2024 杰发科技有限公司

33.4.5 触发模式

PDT_SC[TRGSEL] = 1 时，PDT 为软件触发模式，当 PDT_SC[SWTRIG]控制位写 1，则会产生软件触发信号，PDT_SC[SWTRIG]位会自动清除，写 0 不会有影响。

PDT_SC[TRGSEL] = 0 时，PDT 为硬件触发模式，PDT 的触发信号来源于选择的触发源。

33.4.6 工作模式

PDT_SC[BYPASS_EN] = 0 ，PDT 模块延时功能使能时，PDT 的工作模式可分为单次模式和连续模式：

- PDT_SC[CONT] = 0 时，PDT 模块工作在单次模式。

  当计数值 PDT_CNT[CNT]等于模数值 PDT_MOD[MOD] + 1，计数值 PDT_CNT[CNT]复位为 0 并停止计数。

- PDT_SC[CONT] = 1 时，PDT 模块工作在连续模式。

  当计数值 PDT_CNT[CNT]等于模数值 PDT_MOD[MOD] + 1，计数值 PDT_CNT[CNT]复位为 0 并重新开始计数。在连续模式下，PDT_SC[BYPASS_EN] = 0 延时功能使能时，每当计数值 PDT_CNT[CNT]等于 PDT_DLYn[DLYn](n=0，1，…7)时，会产生触发输出信号。

PDT_SC[BYPASS_EN] = 1 ，PDT 模块延时功能禁用时，PDT 被触发将立即输出触发信号且不受工作模式的影响。

33.4.7 脉冲输出

PDT 模块可生成脉宽可调的方波，通过配置 TRGMUX 可输出至其他模块或者引脚，PDT_POEN[POEN] = 1 脉冲输出使能的条件下：

- 当 PDT 计数值等于 PDT_PODLY[PODLY1]时，脉冲输出为高电平。
- 当 PDT 计数值等于 PDT_PODLY[PODLY2]时，脉冲输出为低电平。

PDT_PODLY[PODLY2]可大于或者小于 PDT_PODLY[PODLY1]，当 PDT_PODLY[PODLY2] = PDT_PODLY[PODLY1]时，PDT 脉冲输出低电平。

© 2013 - 2024 杰发科技有限公司

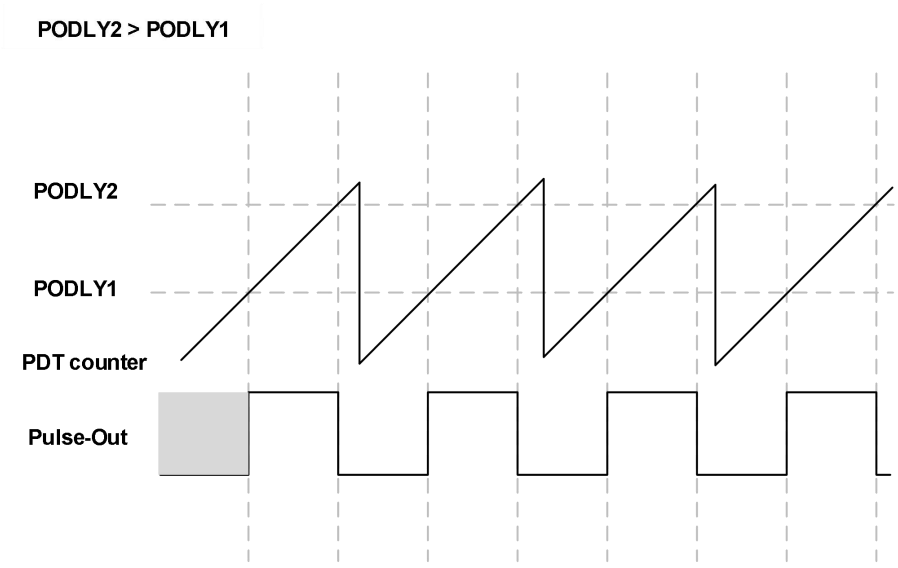


图 33-2 PDT 脉冲输出(PODLY2 > PODLY1)

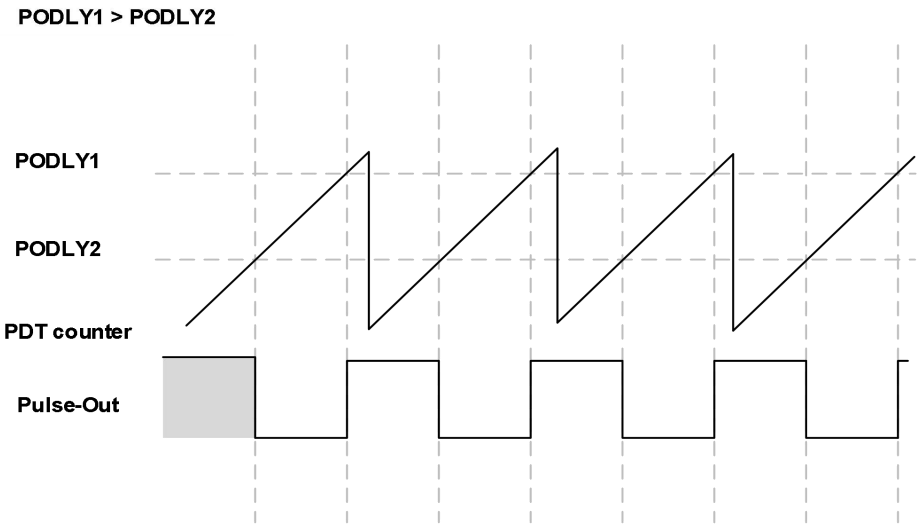


图 33-3 PDT 脉冲输出(PODLY1 > PODLY2)

33.4.8 寄存器更新

33.4.8.1 寄存器缓存

如下为带有缓存功能的寄存器，对相关寄存器进行写入会首先写到对应的缓存区。

- PDT 模数值寄存器(PDT_MOD)
- PDT 中断触发延时寄存器(PDT_IDLY)
- PDT 触发延时寄存器(PDT_DLYn)

© 2013 - 2024 杰发科技有限公司

- PDT 脉冲输出延时寄存器(PDT_PODLY)

33.4.8.2 寄存器加载

当 PDT_SC[LDOK]写 1 时，会使用缓存中的数据更新寄存器，寄存器新值生效是依据 PDT_SC[LDMOD]配置的加载模式。

PDT_SC[LDOK]寄存器有如下特点：

- 在 PDT_SC[LDOK]写 1 之前，寄存器缓存中的数据均为无效。
- 缓存中的数据被写入但未加载至寄存器之前，缓存中的数据为更新前的最后一次写入值
- 当缓存中的数据加载到寄存器中时，PDT_SC[LDOK]会自动清除。
- 对 PDT_SC[LDOK]写 0 为无效操作。

33.4.8.3 加载模式

PDT_SC[LDMOD]与加载模式的说明，参见表 33-1。

表 33-1 加载模式

LDMOD	加载模式说明
0h	对 LDOK 寄存器写 1 后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中
1h	对 LDOK 寄存器写 1 后，当 PDT 计数值 CNT 等于 MOD+1 时，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中
2h	对 LDOK 寄存器写 1 后，当检测到输入 触发事件后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中
3h	对 LDOK 寄存器写 1 后，当 PDT 计数值 CNT 等于 MOD+1 时或者检测到输入 触发事件后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中

图 33-4、图 33-5、图 33-6、图 33-7 分别为 LDMOD = 0h，LDMOD = 1h，LDMOD = 2h，LDMOD = 3h 时的缓存值更新至寄存器的时间点。

© 2013 - 2024 杰发科技有限公司

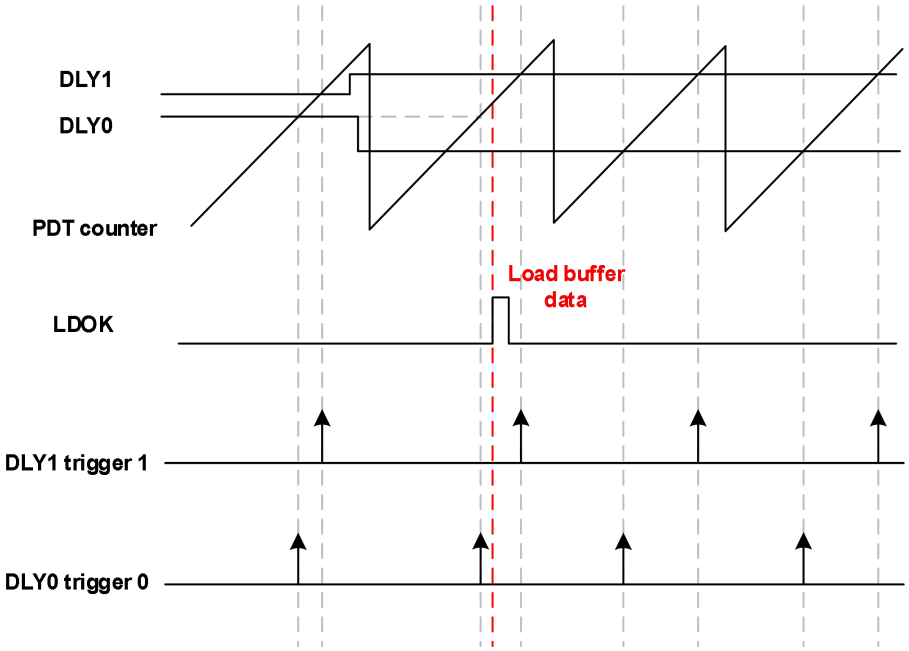


图 33-4 PDT_SC[LDMOD] = 0h 条件下寄存器缓存更新

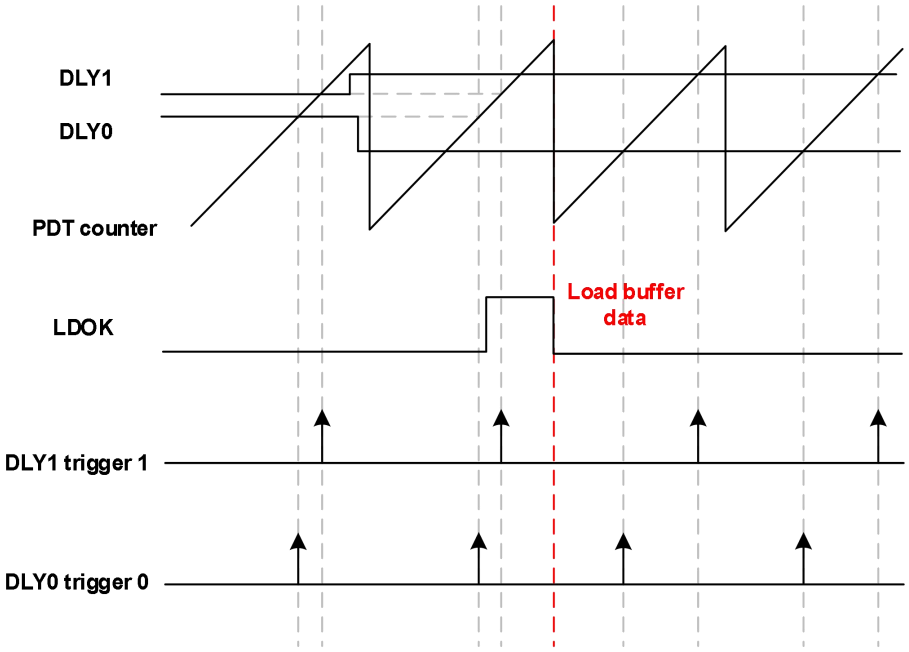


图 33-5 PDT_SC[LDMOD] = 1h 条件下寄存器缓存更新

© 2013 - 2024 杰发科技有限公司

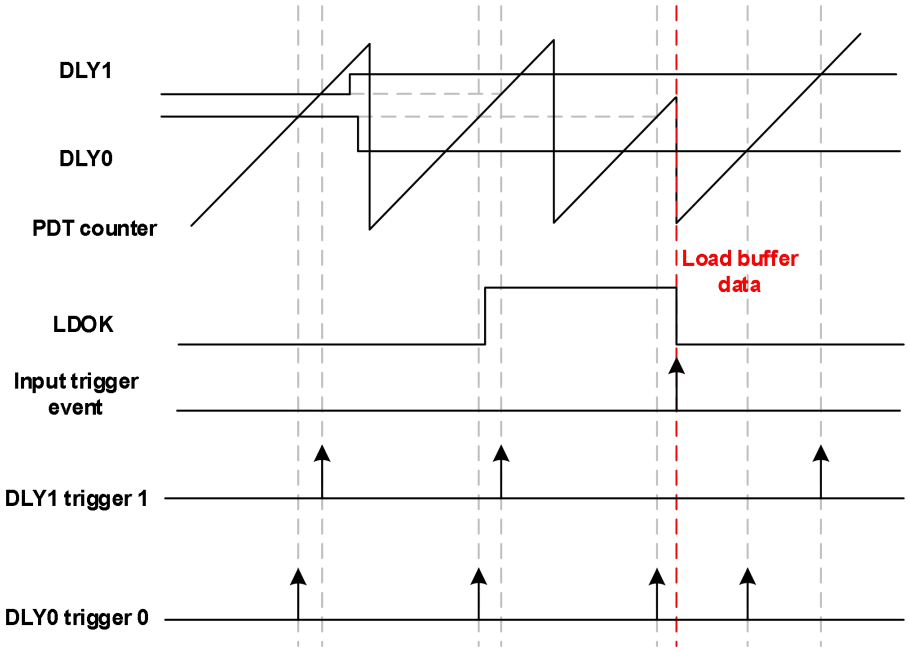


图 33- 6 PDT_SC[LDMOD] = 2h 条件下寄存器缓存更新

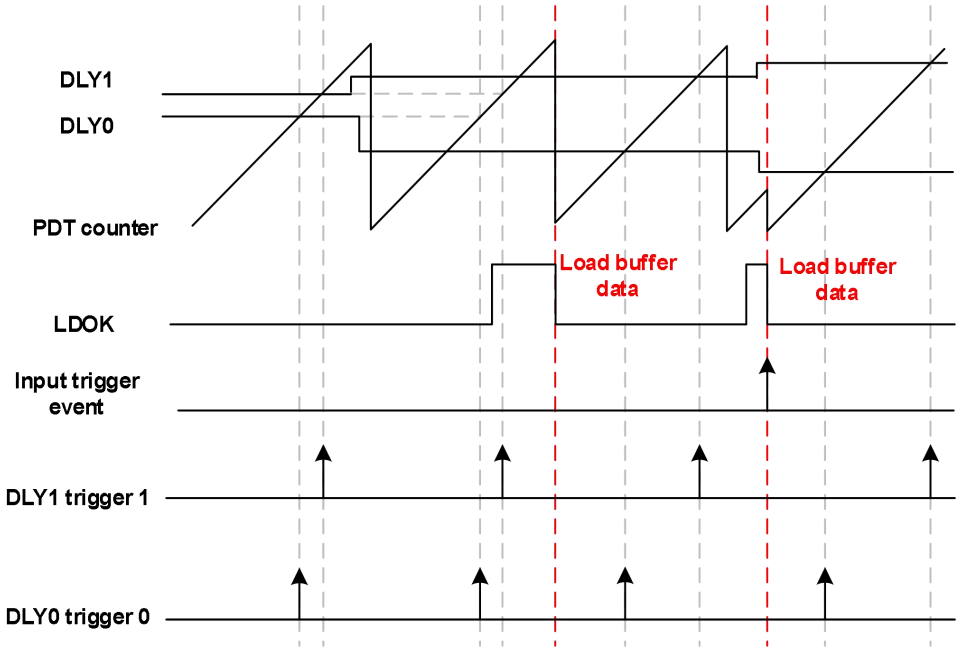


图 33- 7 PDT_SC[LDMOD] = 3h 条件下寄存器缓存更新

33.4.9 中断请求

PDT 模块支持 1 个中断，PDT_SC[PDTIE] = 1 时，PDT 中断使能。当计数器计数值等于 IDLY，则会产生对应的中断。 通过配置 PDT_IDLY[IDLY]寄存器可配置 PDT 中断的延时时间，当计数器计数值 PDT_CNT[CNT]等于 PDT_IDLY[IDLY]，则会产生 PDT 中断。

© 2013 - 2024 杰发科技有限公司

表 33-2 PDT 中断汇总

中断源	标志	本地使能	全局使能
计数器计数值等于 IDLY	PDTIF	PDTIE	无

33.4.10 低功耗模式

表 33-3 PDT 模块低功耗模式

模式	唤醒源	备注
STOP1	\	低功耗模式下模块停止计数
STOP2	\	
VLPS	\	

33.4.11 调试模式

调试模式下模块正常工作。

© 2013 - 2024 杰发科技有限公司

33.5 应用说明

33.5.1 ADC 定时采样控制功能编程指南

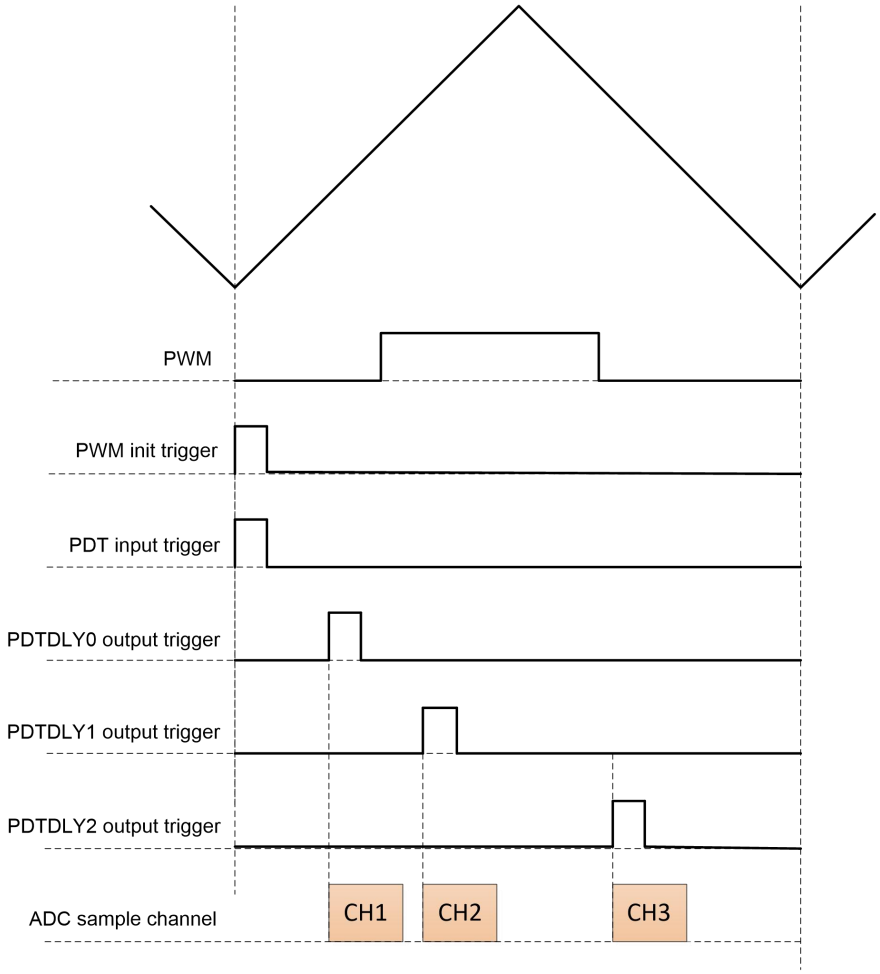


图 33-8 PDT 延时采样

当 ADC 设置为间隔模式，选择 PDT 延时触发输出作为硬件触发源时，可以通过配置 PDT 不同的延迟值来实现精确的采样定时控制。

配置流程：

1. 初始化 PWM，ADC，PDT，CTU 模块
2. 配置 TRGMUX 使 PWM 初始化触发作为 PDT 输入触发源
3. 配置 TRGMUX 使 PDT 延时触发作为 ADC 触发输入源
4. 配置 PDT 触发延时寄存器 PDT_DLYn
5. PDT 周期与 PWM 周期相等

© 2013 - 2024 杰发科技有限公司

33.6 寄存器定义

表 33-4 PDT 寄存器映射

PDT0 基地址：0x40086000

PDT1 基地址：0x40086800

地址	名称	宽度	描述
PDTx 基地址 + 0x00	PDT_SC	32	状态控制寄存器
PDTx 基地址 + 0x04	PDT_MOD	32	模数值寄存器
PDTx 基地址 + 0x08	PDT_DLY0	32	触发延时寄存器 0
PDTx 基地址 + 0x0C	PDT_DLY1	32	触发延时寄存器 1
PDTx 基地址 + 0x10	PDT_DLY2	32	触发延时寄存器 2
PDTx 基地址 + 0x14	PDT_DLY3	32	触发延时寄存器 3
PDTx 基地址 + 0x18	PDT_DLY4	32	触发延时寄存器 4
PDTx 基地址 + 0x1C	PDT_DLY5	32	触发延时寄存器 5
PDTx 基地址 + 0x20	PDT_DLY6	32	触发延时寄存器 6
PDTx 基地址 + 0x24	PDT_DLY7	32	触发延时寄存器 7
PDTx 基地址 + 0x28	PDT_IDLY	32	中断触发延时寄存器
PDTx 基地址 + 0x2C	PDT_PODLY	32	脉冲输出延时寄存器
PDTx 基地址 + 0x30	PDT_DLYn_EN	32	触发使能寄存器
PDTx 基地址 + 0x34	PDT_POEN	32	脉冲输出使能寄存器
PDTx 基地址 + 0x38	PDT_CNT	32	计数器寄存器

说明：

上表中，x = 0, 1。

© 2013 - 2024 杰发科技有限公司

33.6.1 状态控制寄存器(PDT_SC)

表 33-5 PDT_SC 寄存器

PDT_SC **PDT 状态控制寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		BYPASS_EN	TRGSEL	SWTRIG	PDTEN	PDTIF	PDTIE	LDMOD		LDOK	CONT	PRE			MULT	
访问		RW	RW	RW	RW	R/W0C	RW	RW		RW	RW	RW			RW	
复位值		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
14 BYPASS_EN	**BYPASS 功能使能** 0：延时模式使能 1：延时模式禁用，忽略所有通道的延时，在收到触发时立即输出触发
13 TRGSEL	**触发方式选择** 0： 硬件触发 1： 软件触发
12 SWTRIG	**软件触发** 0：写 0 无影响 1：写 1 生成软件触发 **说明：当 PDTEN = 1 且 TRGSEL = 1 时，SWTRIG bit 位写 1 生成软件触发信号，SWTRIG bit 位会自动清除为 0。读取 SWTRIG bit 位始终为 0。**
11 PDTEN	**PDT 使能** 0： PDT 禁用，计数器处于关闭状态 1： PDT 使能
10 PDTIF	**PDT 中断标志** 0： 写 0 清除 PDTIF 标志 1： 写 1 无影响 **说明：当计数器值等于 IDLY +1 时，PDTIF 会被置位（PDTIF = 1）。**

© 2013 - 2024 杰发科技有限公司

字段	说明
9 PDTIE	**PDT 中断使能** 0： PDT 中断禁用 1： PDT 中断使能
8：7 LDMOD	**数据加载模式选择** 0h： 对 LDOK 寄存器写 1 后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中 1h： 对 LDOK 寄存器写 1 后，当 PDT 计数值 CNT 等于 MOD+1 时，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中 2h：对 LDOK 寄存器写 1 后，当检测到输入 触发事件后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中 3h： 对 LDOK 寄存器写 1 后，当 PDT 计数值 CNT 等于 MOD+1 时或者检测到输入 触发事件后，立即将 MOD/IDLY/DLYn/POLDY 缓存区的数据更新到寄存器中 **说明：加载模式的使用配置可参考 33.4.8.3 加载模式 章节。**
6 LDOK	**载入成功** 0：写 0 无影响 1：对 LDOK bit 位写 1 后，会使用寄存器缓存值更新 MOD/IDLY/DLYn/POLDY 寄存器。寄存器缓存值依据 LDMOD 配置的模式进行更新，在 LDMOD 写 1 之前，缓存值不会更新至寄存器。 **说明： LDOK 会在内部缓存数据载入寄存器时自动清 0。寄存器缓存值更新的使用配置可参考 33.4.8 寄存器更新 章节。**
5 CONT	**工作模式** 0： PDT 工作在单次模式 1： PDT 工作连续模式
4：2 PRE	**时钟预分频器** 0h：预分频因子为 1 1h：预分频因子为 2 2h：预分频因子为 4 3h：预分频因子为 8 4h：预分频因子为 16 5h：预分频因子为 32 6h：预分频因子为 64 7h：预分频因子为 128
1：0 MULT	**预分频器的乘法因子** 0h：乘法因子为 1 1h：乘法因子为 10 2h：乘法因子为 20

© 2013 - 2024 杰发科技有限公司

字段	说明
	3h ：乘法因子为 40

33.6.2 模数值寄存器(PDT_MOD)

表 33-6 PDT_MOD 寄存器

PDT_MOD **PDT 计数阈值寄存器** **复位值：0x0000FFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	MOD															
访问	RW															
复位值	FFFFh															

字段	说明
15：0 MOD	**模数值** 计数器的模数值。当计数值 CNT 等于(MOD+1)时，计数值将复位为 0(CNT = 0)。若 PDT 模块工作在连续模式(CONT = 1)，当计数值 CNT 等于(MOD+1)，计数值 CNT 复位为 0 并重新开始计数。读取 MOD 寄存器将返回当前周期实际生效的 MOD 值。

33.6.3 触发延时寄存器 0(PDT_DLY0)

表 33-7 PDT_DLY0 寄存器

PDT_DLY0 **PDT 触发延时寄存器 0** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY0															
访问	RW															
复位值	0															

字段	说明
15：0 DLY0	**DLY0 延时值** DLY0 定义了对应通道触发器的延迟值，当计数器计数值等于 DLY0 时生成触发信号。读取 DLY0 寄存器将返回当前 PDT 周期实际生效的 DLY0 值。

© 2013 - 2024 杰发科技有限公司

33.6.4 触发延时寄存器 1(PDT_DLY1)

表 33-8 PDT_DLY1 寄存器

PDT_DLY1 **PDT 触发延时寄存器 1** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY1															
访问	RW															
复位值	0															

字段	说明
15：0 DLY1	**DLY1 延时值** DLY1 定义了对应通道的预触发器的延迟值， 当计数器计数值等于 DLY1 时生成触发信号。读取 DLY1 寄存器将返回当前 PDT 周期实际生效的 DLY1 值。

33.6.5 触发延时寄存器 2(PDT_DLY2)

表 33-9 PDT_DLY2 寄存器

PDT_DLY2 **PDT 触发延时寄存器 2** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY2															
访问	RW															
复位值	0															

字段	说明
15：0 DLY2	**DLY2 延时值** DLY2 定义了对应通道的预触发器的延迟值，当计数器计数值等于 DLY2 时生成触发信号。读取 DLY2 寄存器将返回当前 PDT 周期实际生效的 DLY2 值。

© 2013 - 2024 杰发科技有限公司

33.6.6 触发延时寄存器 3(PDT_DLY3)

表 33-10 PDT_DLY3 寄存器

PDT_DLY3 **PDT 触发延时寄存器 3** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY3															
访问	RW															
复位值	0															

字段	说明
15：0 DLY3	**DLY3 延时值** DLY3 定义了对应通道的预触发器的延迟值， 当计数器计数值等于 DLY3 时生成触发信号。读取 DLY3 寄存器将返回当前 PDT 周期实际生效的 DLY3 值。

33.6.7 触发延时寄存器 4(PDT_DLY4)

表 33-11 PDT_DLY4 寄存器

PDT_DLY4 **PDT 触发延时寄存器 4** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY4															
访问	RW															
复位值	0															

字段	说明
15：0 DLY4	**DLY4 延时值** DLY4 定义了对应通道的预触发器的延迟值，当计数器计数值等于 DLY4 时生成触发信号。读取 DLY4 寄存器将返回当前 PDT 周期实际生效的 DLY4 值。

© 2013 - 2024 杰发科技有限公司

33.6.8 触发延时寄存器 5(PDT_DLY5)

表 33-12 PDT_DLY5 寄存器

PDT_DLY5 **PDT 触发延时寄存器 5** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY5															
访问	RW															
复位值	0															

字段	说明
15：0 DLY5	**DLY5 延时值** DLY5 定义了对应通道的预触发器的延迟值，当计数器计数值等于 DLY5 时生成触发信号。读取 DLY5 寄存器将返回当前 PDT 周期实际生效的 DLY5 值。

33.6.9 触发延时寄存器 6(PDT_DLY6)

表 33-13 PDT_DLY6 寄存器

PDT_DLY6 **PDT 触发延时寄存器 6** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY6															
访问	RW															
复位值	0															

字段	说明
15：0 DLY6	**DLY6 延时值** DLY6 定义了对应通道的预触发器的延迟值，当计数器计数值等于 DLY6 时生成触发信号。读取 DLY6 寄存器将返回当前 PDT 周期实际生效的 DLY6 值。

© 2013 - 2024 杰发科技有限公司

33.6.10 触发延时寄存器 7(PDT_DLY7)

表 33-14 PDT_DLY7 寄存器

PDT_DLY7 **PDT 触发延时寄存器 7** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	DLY7															
访问	RW															
复位值	0															

字段	说明
15：0 DLY7	**DLY7 延时值** DLY7 定义了对应通道的预触发器的延迟值，当计数器计数值等于 DLY7 时生成触发信号。读取 DLY7 寄存器将返回当前 PDT 周期实际生效的 DLY7 值。

33.6.11 中断触发延时寄存器(PDT_IDLY)

表 33-15 PDT_IDLY 寄存器

PDT_IDLY **PDT 中断触发延时寄存器** **复位值：0x0000FFFF**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	IDLY															
访问	RW															
复位值	FFFFh															

字段	说明
15：0 IDLY	**中断延时值** 当 PDTIE = 1，计数器计数值 CNT 等于 IDLY 时，产生对应的中断。读取 PDT_IDLY 寄存器将返回当前 PDT 周期实际生效的 IDLY 值。

33.6.12 脉冲输出延时寄存器(PDT_PODLY)

表 33-16 PDT_PODLY 寄存器

© 2013 - 2024 杰发科技有限公司

PDT_PODLY **PDT 脉冲输出延时寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	PODLY2															
访问	RW															
复位值	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PODLY1															
访问	RW															
复位值	0															

字段	说明
31：16 PODLY2	**脉冲输出延时值 PODLY2** 当 PDT 计数值等于 PODLY2 时，输出低电平。读取 PODLY2 寄存器将返回当前 PDT 周期实际生效的 PODLY2 值。
15：0 PODLY1	**脉冲输出延时值 PODLY1** 当 PDT 计数值等于 PODLY1 时，输出高电平。读取 PODLY1 寄存器将返回当前 PDT 周期实际生效的 PODLY1 值。

33.6.13 触发使能寄存器(PDT_DLYn_EN)

表 33-17 PDT_DLYn_EN 寄存器

PDT_DLYn_EN **PDT 触发使能寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									DLY7_EN	DLY6_EN	DLY5_EN	DLY4_EN	DLY3_EN	DLY2_EN	DLY1_EN	DLY0_EN
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
7 DLY7_EN	**DLY7 触发使能** 0： DLY7 禁用 1： DLY7 使能
6 DLY6_EN	**DLY6 触发使能** 0： DLY6 禁用

© 2013 - 2024 杰发科技有限公司

字段	说明
	1： DLY6 使能
5 DLY5_EN	**DLY4 触发使能** 0： DLY5 禁用 1： DLY5 使能
4 DLY4_EN	**DLY4 触发使能** 0： DLY4 禁用 1： DLY4 使能
3 DLY3_EN	**DLY3 触发使能** 0： DLY3 禁用 1： DLY3 使能
2 DLY2_EN	**DLY2 触发使能** 0： DLY2 禁用 1： DLY2 使能
1 DLY1_EN	**DLY1 触发使能** 0： DLY1 禁用 1： DLY1 使能
0 DLY0_EN	**DLY0 触发使能** 0： DLY0 禁用 1： DLY0 使能

33.6.14 脉冲输出使能寄存器(PDT_POEN)

表 33-18 PDT_POEN 寄存器

PDT_POEN **PDT 脉冲输出使能寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																POEN
访问																RW
复位值																0

字段	说明
0 POEN	**脉冲输出使能**

© 2013 - 2024 杰发科技有限公司

字段	说明
	0： 禁用 PDT 脉冲输出 1： 使能 PDT 脉冲输出

33.6.15 计数器寄存器(PDT_CNT)

表 33- 19 PDT_CNT 寄存器

PDT_CNT **PDT 计数器寄存器** **复位值： 0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CNT															
访问	RO															
复位值	0															

字段	说明
0 CNT	**计数器值** 只读寄存器，读取当前 PDT 计数器值。计数器值 PDT_CNT[CNT]达到 计数的最大值时(PDT_MOD[MOD] + 1)，计数值 PDT_CNT[CNT]复位为 0。

© 2013 - 2024 杰发科技有限公司

34 周期性中断定时器（Timer）

34.1 简介

Timer 模块是一个 4 通道 32-bit 定时器，用于定时发起中断和触发。

34.2 结构框图

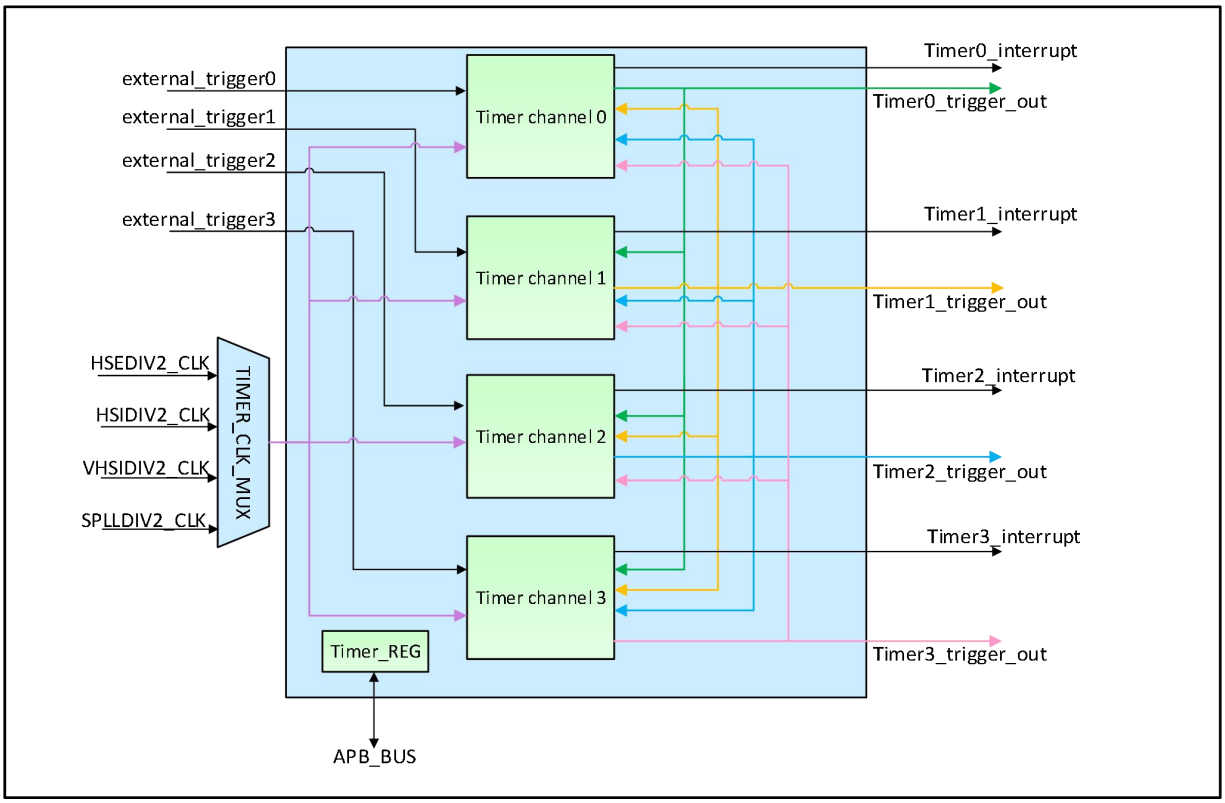


图 34-1 Timer 结构框图

34.3 特性

- 定时器能够生成中断
- 定时器能够生成触发脉冲
- 每个定时器都具有独立的超时周期，生效时刻可配置
- 包含 4 通道 32-bit 向上计数器
- 支持 4 种工作模式
- 支持 3 种控制位

© 2013 - 2024 杰发科技有限公司

- 实计数值可获取、可编程清零
- 支持链（ Link）模式
- 低功耗模式下可选择异步时钟源继续工作

34.4 功能描述

34.4.1 时钟源

系统提供 BUS_CLK 和功能时钟给 Timer 模块，这些时钟源在 clock 模块都可以被使能/禁止；

其中功能时钟有下面四种（时钟源来源参考 10.2 结构框图 章节的“时钟控制结构图”）：

- SPLLDIV2_CLK,
- VHSIDIV2_CLK,
- HSIDIV2_CLK,
- HSEDIV2_CLK.

Timer 的功能时钟在 clock 模块寄存器 CKGEN_PERI_CLK_MUX0[TIMER_MUX]选择，且是 4 通道 Timer 共用的。在不同功耗模式下可以选择不同的时钟源。

说明：
BUS_CLK 用于寄存器访问，功能时钟作为 Mode0 和 Mode1 的计数时钟。

34.4.2 工作模式

Timer 有 4 种工作模式，每一通道可以通过寄存器 TIMER_CHx_CTRL[MODE]独立设置：

- Mode0：32-bit 周期计数器
- Mode1：双 16-bit 周期计数器
- Mode2：32-bit 脉冲累加器
- Mode3：32-bit 脉冲捕获模式

© 2013 - 2024 杰发科技有限公司

其中，Mode0~2 都属于比较模式，Mode3 属于捕获模式；每种模式的详细时序图见 34.5.2 详细时序图。

处于链模式的 Timer 通道，其工作模式设置值将被忽略。

34.4.2.1 32-bit 周期计数器

Timer 对所选的时钟源进行计数，到达超时值置位超时标志、产生触发输出和中断。

此种工作模式下，3 个控制位（TSOT、TROT、TSOI）都有效。

34.4.2.2 双 16-bit 周期计数器

Timer 对所选的功能时钟进行计数，到达低 16-bit 超时值对应值时，拉高触发输出；然后再计数至高 16-bit 超时值时，拉低触发输出，并置位超时标志。

此种工作模式下，3 个控制位（TSOT、TROT、TSOI）都有效。

34.4.2.3 32-bit 脉冲累加器

Timer 对所选的片内触发输入进行计数，在每一个触发输入的上升沿 Timer CVAL 递增一次，到达超时值置位超时标志、产生触发输出和中断。

此种工作模式下，只有控制位 TSOI 有效。

34.4.2.4 32-bit 脉冲捕获模式

Timer 对所选的片内触发输入进行捕获，实时计数值 CVAL 在全范围（即 32-bit 0xFFFFFFFF）内计数，当捕获到触发输入时，会将此时的实时 CVAL 值加载到超时值寄存器里；并产生触发输出和中断；

此种工作模式下，控制位（TROT、TSOI）有效。

34.4.3 触发输入选择

可以为每路 Timer 选择不同的触发输入：

- 选择 Timer 内部触发源：将 TIMER_CHx_CTRL[TRG_SRC] 选择内部触发，然后再在 TIMER_CHx_CTRL [TRG_SEL]选择一路该通道以外的 Timer 通道的触发输出；
- 选择片内其他模块的触发输出作为 Timer 的触发输入：将[TRG_SRC]选择外部触发源，然后再在 TRGMUX_TIMER[SEL_TIMER_CHx]设置该通道 Timer 的触发输入来自哪个片内模块。

© 2013 - 2024 杰发科技有限公司

34.4.4 超时值设置

可以为每路 Timer 设置独立的超时值 TIMER_CHx_TVAL[TVAL]，向上计数到达超时值会置位超时标志，产生触发输出，若同时使能了该通道 Timer 的中断、则还会产生相应的中断。

计数周期计算公式如下：

the timing period(Unit: second) = (TVAL + 1) / timing clock frequency(Unit: HZ)

超时值有两种生效模式（可以通过寄存器 TIMER_CHx_CTRL[TVAL_UP]配置）：

1. [TVAL_UP]=0：下一超时周期生效
2. [TVAL_UP]=1：立即生效

34.4.5 实时计数值可获取可清零

通过 TIMER_CHx_CVAL[CVAL]可以随时获取该 Timer 的实时计数值，也可以通过向 CVAL 写任意值将其清为零。

读取 CVAL 值时，需要通过 TIMER_CHx_CTRL [CVAL_SEL]设置读取的分频 SEL 值，不同的 BUS_CLK 跟功能时钟频率，需要按照寄存器描述配置不同的 SEL 值。

说明：脉冲累加模式，SEL 值必须设置为 00.

34.4.6 触发输出

1. 产生时刻：

   Mode0~2 计数到达超时值、Mode3 捕获到触发输入上升沿时候，该通道 Timer 会输出触发输出。

2. 脉宽：

   Mode0、2、3 触发输出的脉宽是一个时钟源周期（比如 Timer 功能时钟是 48MHZ，那么触发输出在一个周期内的高电平就是一个 48MHZ 时钟周期）；

   Mode1 触发输出的脉宽是 TVAL_H(超时值高 16-bit 对应值 TIMER_CHx_TVAL[TVAL][31:16])个时钟源周期。

3. 周期：

   触发输出信号的周期就是达到超时或发生捕获的周期。

4. 用途：

   Timer 的触发输出可以通过 CTU 模块再作为其他模块的触发输入。

© 2013 - 2024 杰发科技有限公司

34.4.7 控制位

关于计数开始时刻、单次计数与否、检测到触发输入时是否重新加载超时值，有三个相应的控制位 TIMER_CHx_CTRL[TROT][TSOI][TSOT]。

每路 Timer 都可以独立设置不同的控制位。

34.4.7.1 检测到触发输入时重新加载超时值 （TROT）

当 TROT=1，每检测到一次触发输入，该通道 Timer 的实时计数值就被重新清零，从初始值 0 开始计数；

当 TROT=0，Timer 运行起来后的触发输入不会对实时计数值产生影响。

34.4.7.2 单次计数 （TSOI）

当 TSOI=1，该通道 Timer 在完成一次超时或捕获后即停止计数；

当 TSOI=0，该通道 Timer 在完成一次超时或捕获后不会停止计数，将持续产生触发输出和中断。

34.4.7.3 检测到触发输入时才开始计数（TSOT）

当 TSOT=1，即使置位 TIMER_ENR[TENx]=1 也不会触发 Timer 开始计数，必须等检测到一次触发输入才会开始计数，一旦开始计数，后面来的触发输入会被忽略；

当 TSOT=0，置位[TENx]=1 即会触发 Timer 开始计数。

34.4.8 链模式

当某个定时器的链模式 TIMER_CHx_CTRL[CHN_EN]处于使能状态，那么只有在上一个定时器溢出后，它才会开始计时。因此，如果定时器 n-1 计数至超时值，定时器 n 的值将递增 1。这样就能将某些

说明：
第一个定时器 (Timer 0) 不能连接至其他定时器。

定时器连接起来形成更长的定时器。

当某一 Timer 处于链模式，其工作模式的设置值将不起作用；但其控制位的设置值还是可以起作用。

© 2013 - 2024 杰发科技有限公司

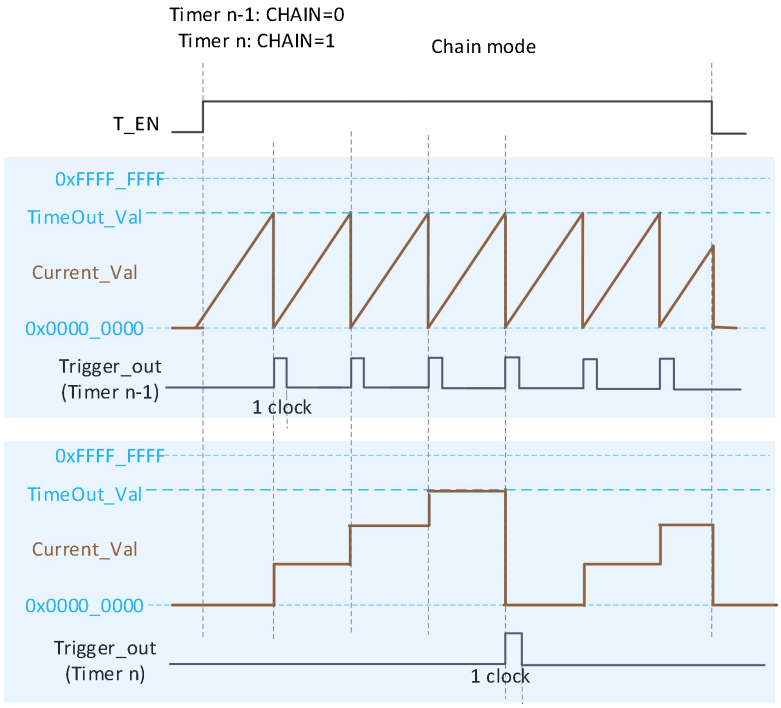


图 34-2 链模式

34.4.9 中断请求

定时器中断可通过置位 TIMER_IER[TIEx]来使能。当相关定时器发生超时， TIMER_SR[TIFx]超时标志置位为 1，写入 1 可清零。在使用链模式时，一般只使能定时器 n 中断，与 n 链接的定时器中断都是关闭状态。

表 34-1 Timer 中断汇总

中断源	标志	使能	全局使能
Timer 通道 0 超时	TIF0	TIE0	
Timer 通道 1 超时	TIF1	TIE1	
Timer 通道 2 超时	TIF2	TIE2	
Timer 通道 3 超时	TIF3	TIE3	

34.4.10 低功耗模式

VLPS 模式只有 HSIDIV2_CLK 时钟源可以选择。三种停止模式 Timer 唤醒源如下：

表 34-2 Timer 唤醒源

模式	唤醒源
STOP1	Timer0~3 超时中断
STOP2	Timer0~3 超时中断

© 2013 - 2024 杰发科技有限公司

模式	唤醒源
VLPS	Timer0~3 超时中断

34.4.11 调试模式

TIMER_CR[DBG_EN]=1 时，在 Debug Halt 状态，Timer 继续运行；

若[DBG_EN]=0，在 Debug Halt 状态，Timer 停止运行。

34.5 应用说明

34.5.1 编程指南

1. 在 clock 模块寄存器 CKGEN_PERI_CLK_MUX0[TIMER_MUX]选择 Timer 的功能时钟；
2. 配置 TIMER_CR[MC_EN]使能 Timer 模块 clock；
3. 配置 TIMER_CHx_TVAL[TVAL]为每一路 Timer 通道设置工作模式、超时值；
4. 配置 TIMER_CHx_CTRL 设置触发输入、控制位、以及是否使能链模式等；
5. 配置 TIMER_IER[TIEx]设置 Timer 中断是否使能；
6. 配置 TIMER_ENR[TENx]使能 Timer。

其中，上述 x=0~3。

34.5.2 详细时序图

下述工作模式与控制位的组合中，“X”表示该控制位对该模式 Timer 无影响，时序图中“1 clock”标识指的是触发输出脉宽为一个计数时钟源周期。

32-bit 周期计数器和双 16-bit 周期计数器模式下，控制位组合对计数的影响一样，所以只给出了 32-bit 周期计数器模式下的时序图。

表 34-3 Timer 控制位和工作模式组合

工作模式	影响 Timer 工作的控制位			
	时序图	TSOT	TROT	TSOI
32-bit 周期计数器（Mode0）	图 34-3	0	0	0
	图 34-4	0	0	1
	图 34-5	0	1	0
	图 34-6	0	1	1
	图 34-7	1	0	0

© 2013 - 2024 杰发科技有限公司

工作模式	影响 Timer 工作的控制位			
	图 34-8	1	0	1
	图 34-9	1	1	0
	图 34-10	1	1	1
双 16-bit 周期计数器（Mode1）	图 34-11	0	0	0
		0	0	1
		0	1	0
		0	1	1
		1	0	0
		1	0	1
		1	1	0
		1	1	1
32-bit 脉冲累加器（Mode2）	图 34-12	X	X	0
	图 34-13	X	X	1
32-bit 脉冲捕获模式（Mode3）	图 34-14	X	0	0
	图 34-15	X	1	0
	图 34-16	X	0	1

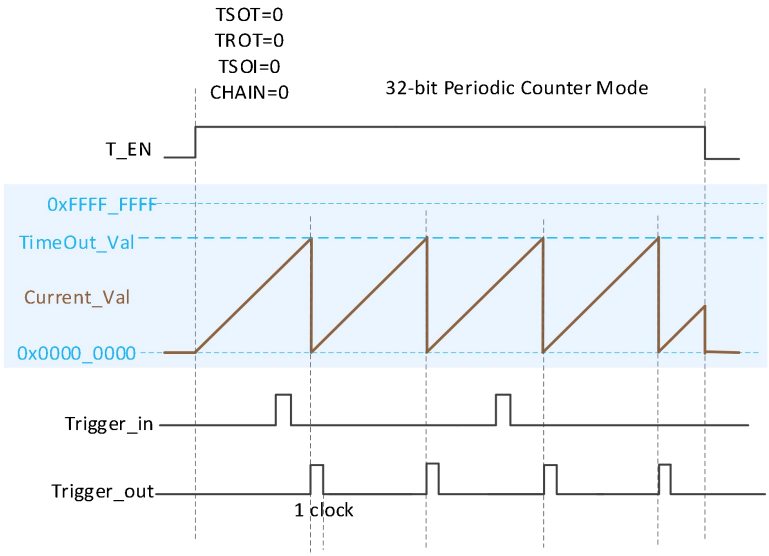


图 34-3 Mode0：TSOT=0,TROT=0,TSOI=0

© 2013 - 2024 杰发科技有限公司

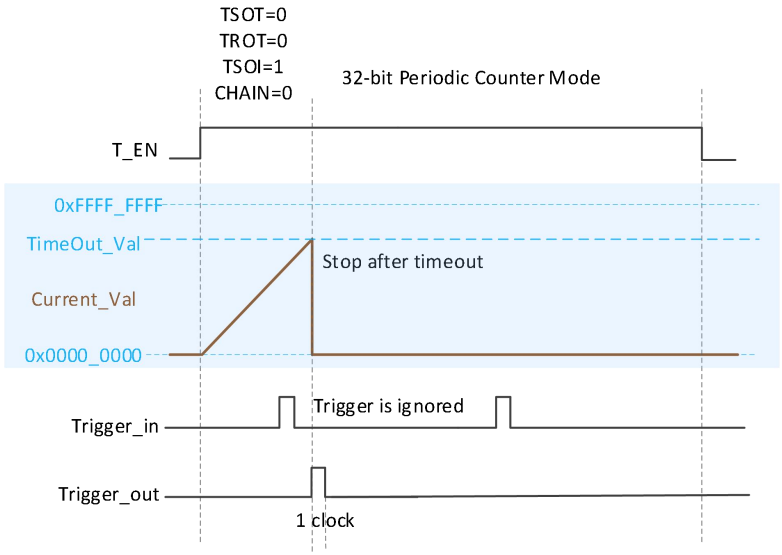


图 34-4 Mode0：TSOT=0,TROT=0,TSOI=1

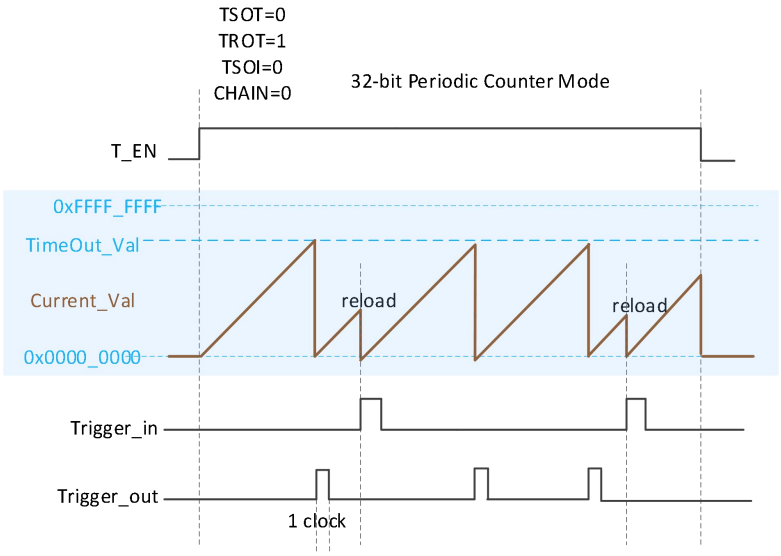


图 34-5 Mode0：TSOT=0,TROT=1,TSOI=0

© 2013 - 2024 杰发科技有限公司

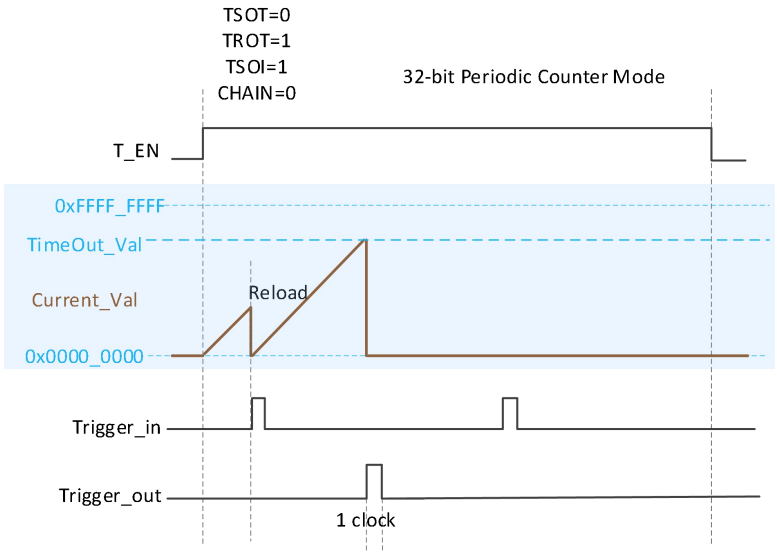


图 34-6 Mode0：TSOT=0,TROT=1,TSOI=1

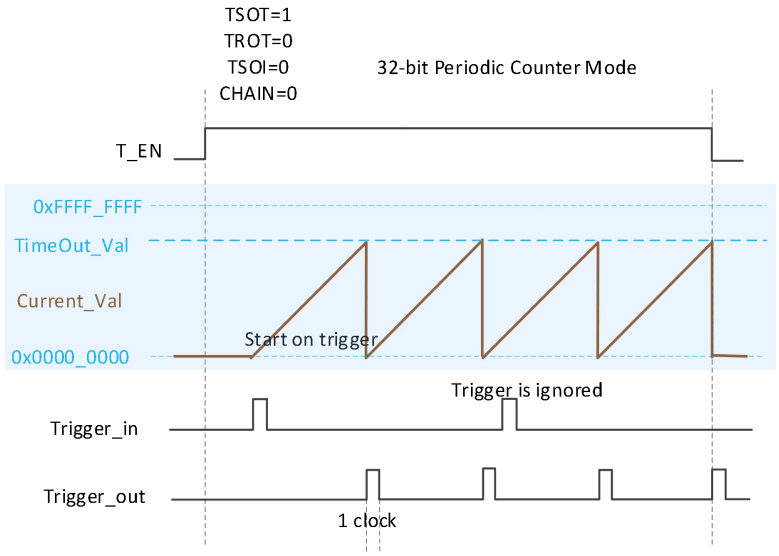


图 34-7 Mode0：TSOT=1,TROT=0,TSOI=0

© 2013 - 2024 杰发科技有限公司

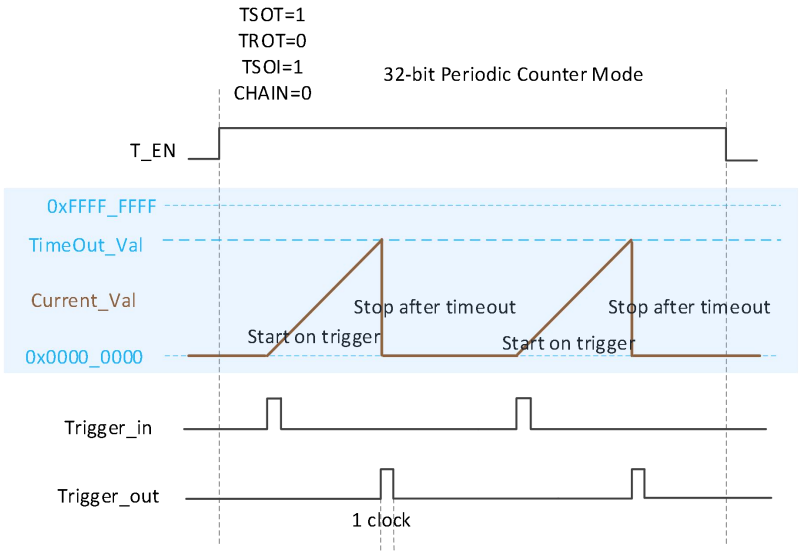


图 34- 8 Mode0：TSOT=1,TROT=0,TSOI=1

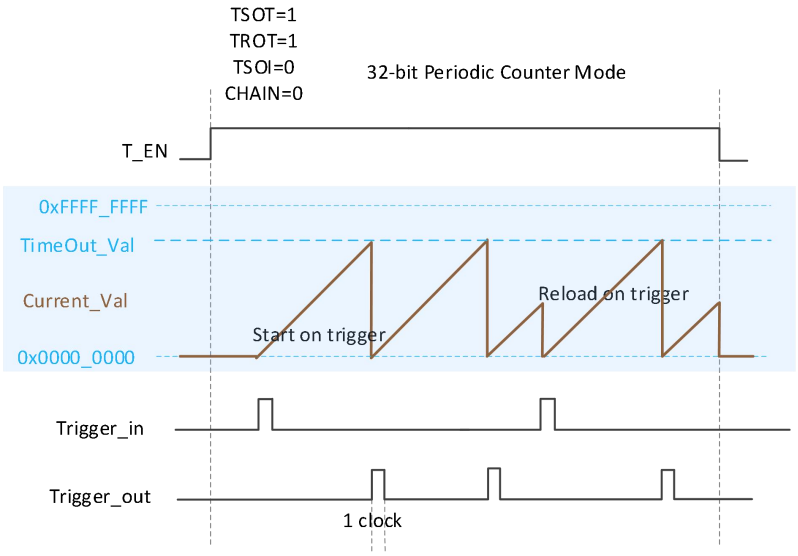


图 34- 9 Mode0：TSOT=1,TROT=1,TSOI=0

© 2013 - 2024 杰发科技有限公司

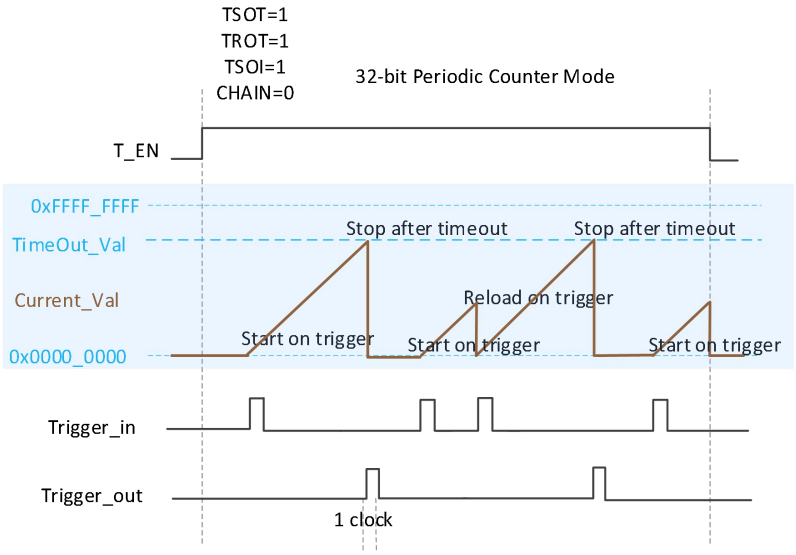


图 34-10 Mode0：TSOT=1,TROT=1,TSOI=1

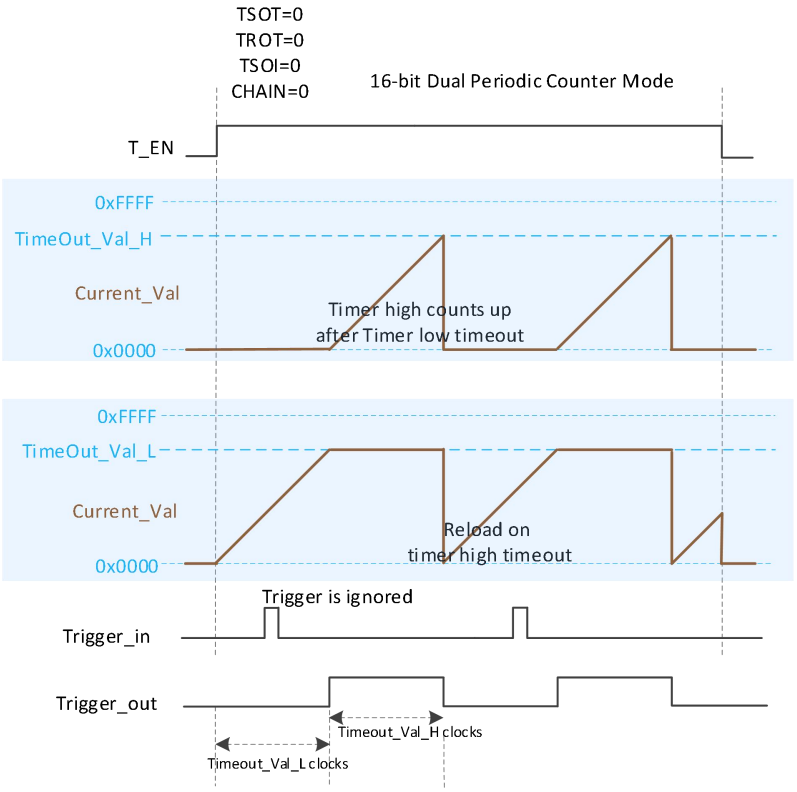


图 34-11 Mode1：TSOT=0,TROT=0,TSOI=0

© 2013 - 2024 杰发科技有限公司

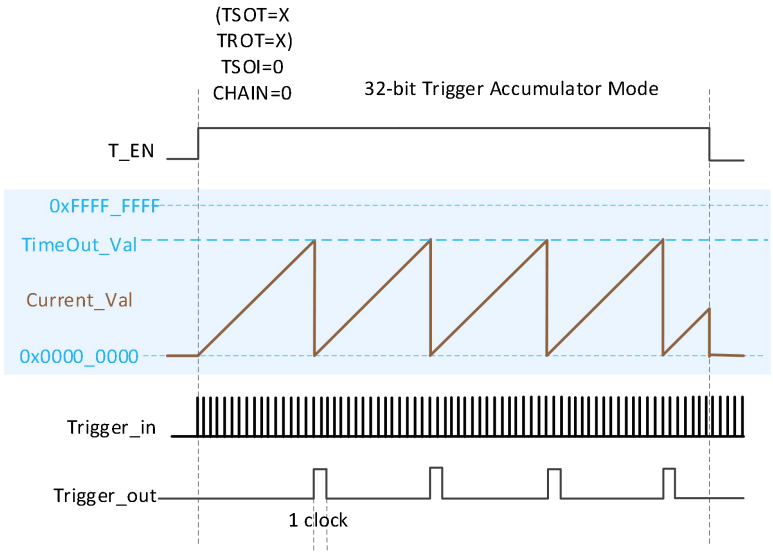


图 34- 12 Mode2：TSOT=X,TROT=X,TSOI=0

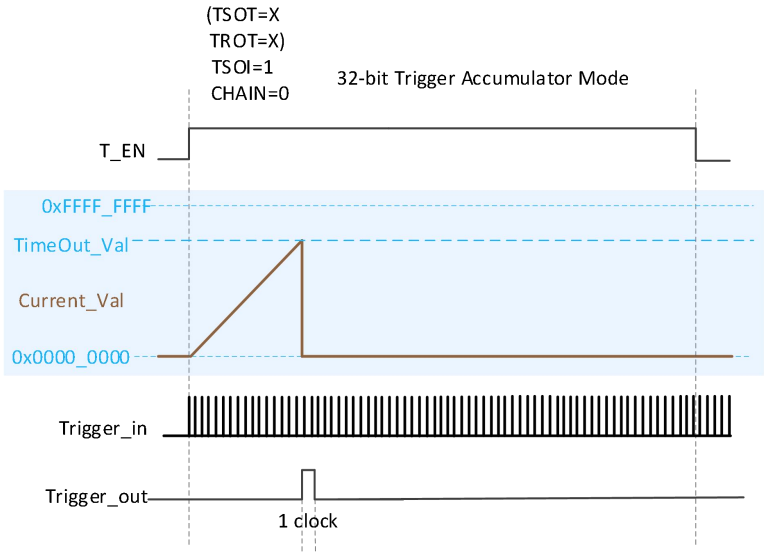


图 34- 13 Mode2：TSOT=X,TROT=X,TSOI=1

© 2013 - 2024 杰发科技有限公司

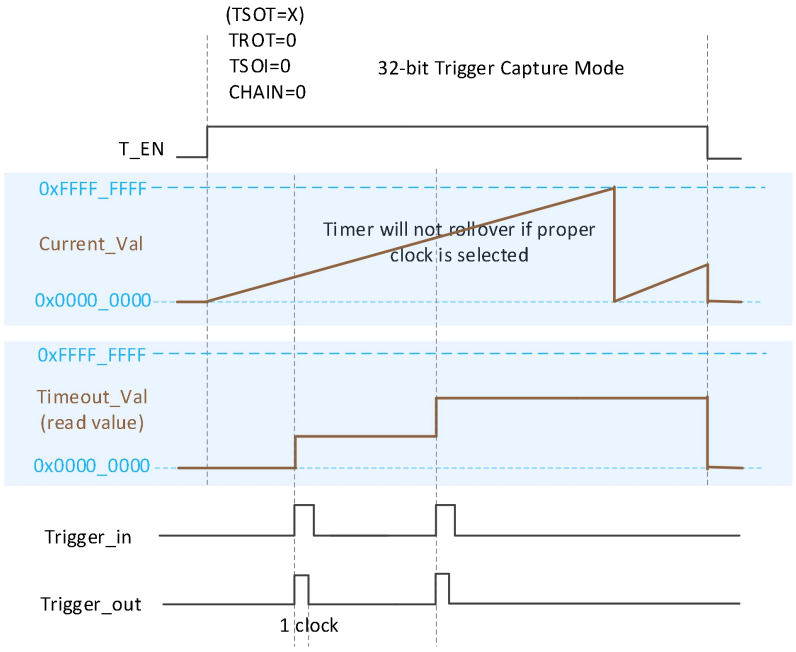


图 34-14 Mode3：TSOT=X,TROT=0,TSOI=0

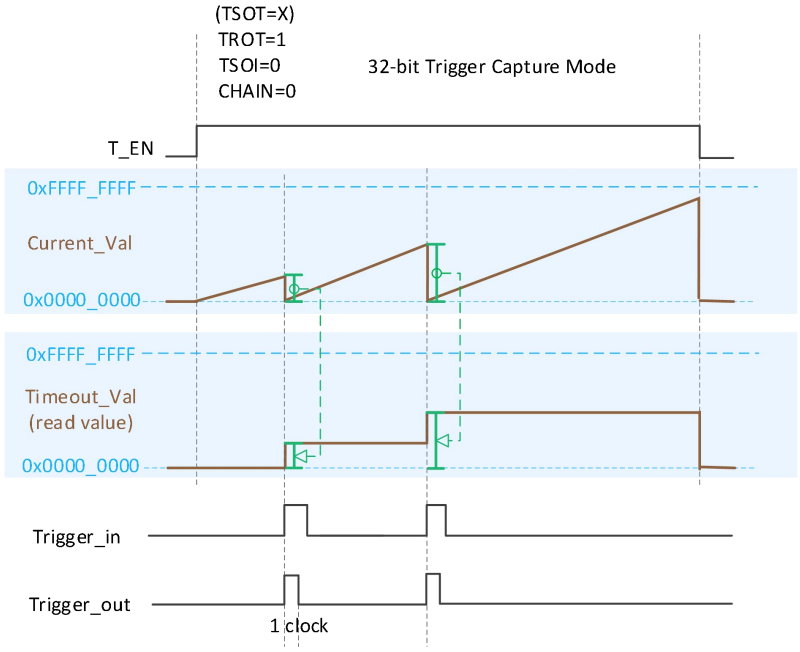


图 34-15 Mode3：TSOT=X,TROT=1,TSOI=0

© 2013 - 2024 杰发科技有限公司

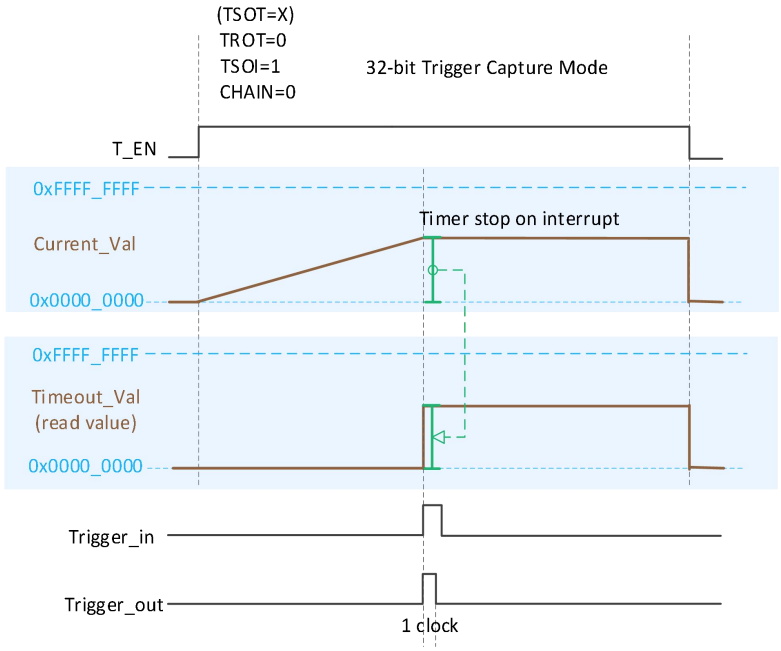


图 34-16 Mode3：TSOT=X,TROT=0,TSOI=1

34.6 寄存器定义

Timer 寄存器只支持 32 位访问。

表 34-4 Timer 寄存器映射

TIMER 基地址 = 0x40011000
TIMER_CH0 基地址=0x40011010
TIMER_CH1 基地址=0x40011020
TIMER_CH2 基地址=0x40011030
TIMER_CH3 基地址=0x40011040

地址	名称	宽度	描述
TIMER 基地址+0x00	TIMER_CR	32	模块控制器寄存器
TIMER 基地址+0x04	TIMER_SR	32	中断标志位寄存器
TIMER 基地址+0x08	TIMER_IER	32	中断使能寄存器
TIMER 基地址+0x0C	TIMER_ENR	32	通道使能寄存器
TIMER_CHx 基地址+0x00	TIMER_CHx_TVAL	32	装载值寄存器
TIMER_CHx 基地址+0x04	TIMER_CHx_CVAL	32	当前值寄存器
TIMER_CHx 基地址+0x08	TIMER_CHx_CTRL	32	通道控制寄存器

© 2013 - 2024 杰发科技有限公司

说明：
上表中，x=0~3。

34.6.1 模块控制寄存器(TIMER_CR)

表 34-5 TIMER_CR 寄存器

TIMER_CR **模块控制寄存器** **复位值：**
0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															DBG_EN	MC_EN
访问															RW	RW
复位值															0	0

字段	说明
1 DBG_EN	**调试模式 Timer 使能控制** 0：调试模式下停止 Timer 1：调试模式下 Timer 继续运行 调试模式意味着 CPU 进入 Halt 模式，DBG_EN 控制该模式下 Timer 是否运行。
0 MC_EN	Timer Module Clock 使能控制 0：禁止 Timer Module clock 1：使能 Timer Module clock

© 2013 - 2024 杰发科技有限公司

34.6.2 中断标志寄存器 (TIMER_SR)

表 34- 6 TIMER_SR 寄存器

TIMER_SR 中断标志寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													TIF3	TIF2	TIF1	TIF0
访问													R/W1C	R/W1C	R/W1C	R/W1C
复位值													0	0	0	0

字段	说明
3 TIF3	**Timer3 中断标志位** 0: 超时/捕获尚未发生 1: 超时/捕获已经发生 **说明：写 1 可以清除该标志位** 比较模式：当 Timer3 计数到达 TVAL 值，该中断标志会被置为 1 捕获模式：当 Timer3 捕获到触发输入时，该中断标志会被置为 1
2 TIF2	**Timer2 中断标志位** 0: 超时/捕获尚未发生 1: 超时/捕获已经发生 **说明：写 1 可以清除该标志位** 比较模式：当 Timer2 计数到达 TVAL 值，该中断标志会被置为 1 捕获模式：当 Timer2 捕获到触发输入时，该中断标志会被置为 1
1 TIF1	**Timer1 中断标志位** 0: 超时/捕获尚未发生 1: 超时/捕获已经发生 **说明：写 1 可以清除该标志位** 比较模式：当 Timer1 计数到达 TVAL 值，该中断标志会被置为 1 捕获模式：当 Timer1 捕获到触发输入时，该中断标志会被置为 1
0 TIF0	**Timer0 中断标志位** 0: 超时/捕获尚未发生 1: 超时/捕获已经发生

© 2013 - 2024 杰发科技有限公司

字段	说明
	说明：写 1 可以清除该标志位 比较模式：当 Timer0 计数到达 TVAL 值，该中断标志会被置为 1 捕获模式：当 Timer0 捕获到触发输入时，该中断标志会被置为 1

34.6.3 中断使能寄存器(TIMER_IER)

表 34- 7 TIMER_IER 寄存器

TIMER_IER **中断使能寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													TIE3	TIE2	TIE1	TIE0
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3 TIE3	**Timer3 中断使能控制** 0：禁止 1：使能
2 TIE2	**Timer2 中断使能控制** 0：禁止 1：使能
1 TIE1	**Timer1 中断使能控制** 0：禁止 1：使能
0 TIE0	**Timer0 中断使能控制** 0：禁止 1：使能

© 2013 - 2024 杰发科技有限公司

34.6.4 通道使能寄存器(TIMER_ENR)

表 34-8 TIMER_ENR 寄存器

TIMER_ENR 通道使能寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													TEN3	TEN2	TEN1	TEN0
访问													RW	RW	RW	RW
复位值													0	0	0	0

字段	说明
3 TEN3	**Timer3 通道使能控制** 0：禁止 1：使能
2 TEN2	**Timer2 通道使能控制** 0：禁止 1：使能
1 TEN1	**Timer1 通道使能控制** 0：禁止 1：使能
0 TEN0	**Timer0 通道使能控制** 0：禁止 1：使能

© 2013 - 2024 杰发科技有限公司

34.6.5 装载值寄存器(TIMER_CHx_TVAL)

表 34- 9 TIMER_CHx_TVAL 寄存器

TIMER_CHx_TVAL 装载值寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	TVAL[31: 16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TVAL[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 TVAL	**Timer_CHx 装载值** 比较模式：TVAL 是 Timer_CHx 超时值，Timer_CHx 的 CVAL 会从 0 计数到 TVAL 然后发生超时 捕获模式：TVAL 将会记录储存发生捕获时的 CVAL 值

34.6.6 当前值寄存器(TIMER_CHx_CVAL)

表 34- 10 TIMER_CHx_CVAL 寄存器

TIMER_CHx_CVAL 当前值寄存器 复位值：
0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CVAL[31: 16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CVAL[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 CVAL	**Timer_CHx 当前计数值** 代表 Timer_CHx 实时计数值，Timer_CHx 被使能时，向 CVAL 写入任意值都可以将其清为 0

© 2013 - 2024 杰发科技有限公司

34.6.7 通道控制寄存器(TIMER_CHx_CTRL)

表 34- 11 TIMER_CHx_CTRL 寄存器

TIMER_CHx_CTRL 通道控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																TRG_SRC
访问																RW
复位值																0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TRG_SEL					TROT	TSOI	TSOT			MODE		CVAL_SEL		TVAL_UP	CHN_EN
访问	RW					RW	RW	RW			RW		RW		RW	RW
复位值	0	0	0	0		0	0	0			0	0	0	0	0	0

字段	说明
16 TRG_SRC	**Timer_CHx 内部/外部触发源选择** 0: 选择 Timer 以外模块的触发输出作为触发输入 1: 选择 Timer 模块内其它通道作为触发输入 **说明：当选择外部触发源时候，需要 CTU 模块配置来自哪个模块**
15:12 TRG_SEL	**触发源通道选择** 0000b：选择 Timer0 的触发输出作为触发输入 0001b：选择 Timer1 的触发输出作为触发输入 0010b：选择 Timer2 的触发输出作为触发输入 0011b：选择 Timer3 的触发输出作为触发输入 0100b-1111b：保留
10 TROT	**触发源来时是否重新加载 TVAL 的控制** 0: 不重新加载 1: 重新加载
9 TSOI	**发生一次中断后是否停止 Timer 的控制** 0：不停止 1：停止 **说明：** **TSOI=1 发生一次中断后 Timer 停止， Timer 下一次开始计数的时刻取决于 TSOT 的设置。** ■ **TSOT=0 时候，Timer 将会在检测到 TEN 的上升沿后开始下一次计数；** ■ **TSOT=1 时候，Timer 将会在所选触发源出现上升沿时候开始下一次计数。**

© 2013 - 2024 杰发科技有限公司

字段	说明
8 TSOT	**触发源发生时 Timer 才开始计数的控制** 0：配置 MC_EN=1，TENx=1 即会使 Timer 开始计数； 1：MC_EN=1，TENx=1 且所选触发源发生上升沿时 Timer 才开始计数。
5:4 MODE	**Timer 工作模式** 00b: 32-bit 周期计数器 01b: 双 16-bit 周期计数器 10b: 32-bit 脉冲累加器 11b: 32-bit 脉冲捕获模式
3:2 CVAL_SEL	**Timer 读取 CVAL 值 SEL 设置** 00b: BUSCLK/Function clk >= 4 01b: BUSCLK/Function clk >= 3 10b: BUSCLK/Function clk >= 1 **说明： Timer 配置为脉冲累加功能时，CVAL_SEL 必须设置为 00。**
1 TVAL_UP	**TVAL 生效时刻控制** 0: 当 TVAL 被写一个新值，它会再该计数周期发生超时后开始生效 1: 当 TVAL 被写一个新值，它会立即生效
0 CHN_EN	**链模式使能控制** 0：禁止 1：使能 **说明：当 Timer_CH $_{(n+1)}$ 链模式使能时，Timer_CH $_{(n+1)}$ 的 CVAL 在 Timer_CH$_n$ 发生超时后递增一次**

© 2013 - 2024 杰发科技有限公司

35 脉冲计数定时器（PCT）

35.1 简介

PCT（Pulse Count Timer）可作为一路 16-bit 时钟计数器，也可作为 16-bit 脉冲计数器，可以产生周期性中断和脉冲输出。

35.2 特性

- 定时器能够生成中断
- 定时器能够生成触发输出
- 超时周期可配置
- 带可选分频功能的 16-bit 向上时钟计数器
- 带可选滤波功能的 16-bit 向上脉冲计数器：可选片内触发源或片外脉冲输入、检测极性可配置
- 支持 free-running 模式
- 只能被 POR 复位，可以作为时间计数器
- 低功耗模式下可选择异步时钟源继续工作，作为唤醒源

35.3 结构框图

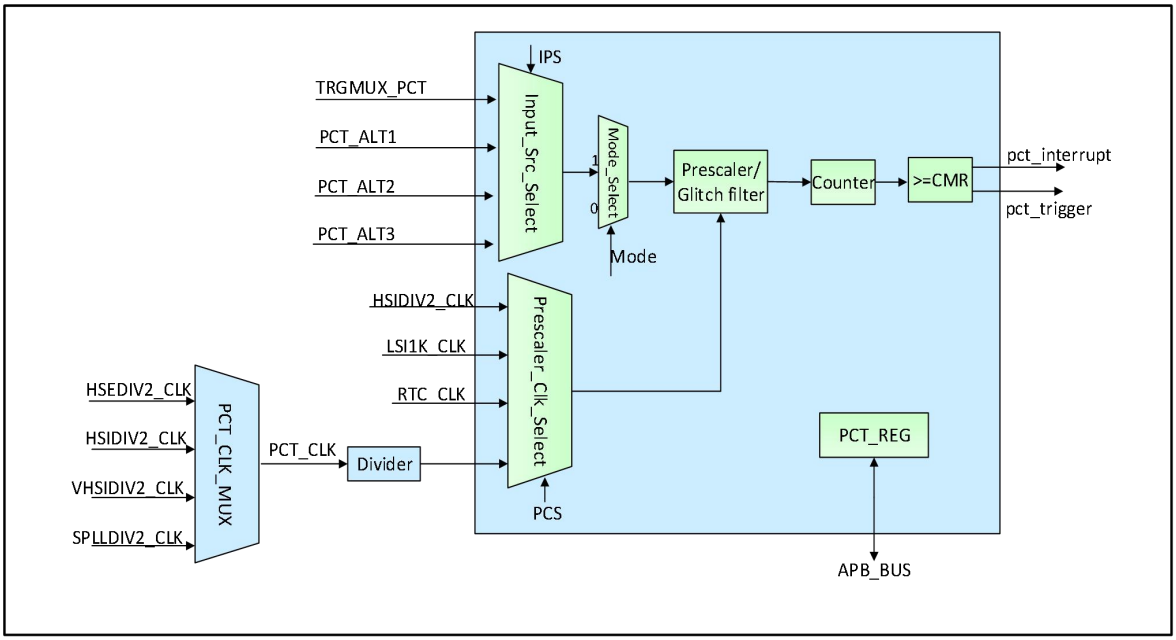


图 35-1 PCT 结构框图

© 2013 - 2024 杰发科技有限公司

35.4 功能描述

35.4.1 时钟源

除了用于寄存器访问的 BUS_CLK，系统还给 PCT 提供下面 6 种功能时钟源（时钟源来源参考 10.2 结构框图 章节的“时钟控制结构图”）可以用于计数：

- HSEDIV2_CLK,
- HSIDIV2_CLK,
- VHSIDIV2_CLK,
- SPLLDIV2_CLK,
- LSI1K_CLK,
- RTC_CLK

这些功能时钟源通过 PCT 模块内部寄存器 PCT_PSR[PCS] 和 clock 模块寄存器 CKGEN_PERI_CLK_MUX1[PCT_MUX] 做两级选择，对后者选择的时钟源还可以通过 CKGEN_PERI_CLK_DIV[PCT_DIV]进行 2~16 分频。

其中 HSIDIV2_CLK 的配置有两种实现方式，既可以通过 PCT_PSR[PCS]直接配置得到，也可以通过 clock 模块寄存器配置得到，在 35.3 结构框图 章节中可以看到。

说明：
BUS_CLK 用于模块寄存器访问，上面 6 种功能时钟源可以作为 PCT 时钟计时器模式、以及脉冲计数器模式的滤波子模块的时钟源。

在不同功耗模式下可以选择不同的时钟源。

35.4.2 工作模式

PCT 有两种工作模式：

- 带可选分频功能的 16-bit 向上时钟计数器
- 带可选滤波功能的 16-bit 向上脉冲计数器

其中分频和滤波功能共用寄存器 PCT_PSR。

© 2013 - 2024 杰发科技有限公司

35.4.2.1 时钟计数器

PCT 对所选的时钟源进行计数，或者时钟源分频后的源周期进行计数；

到达 PCT_CMR[CMR]所设定的超时值时，置位超时标志 PCT_CSR[CF]，产生触发输出。如果使能了超时中断 PCT_CSR[CIE]，将产生超时中断。

35.4.2.2 脉冲计数器

PCT 对所选的片内触发输入、或者片外脉冲进行计数，到达 PCT_CMR[CMR]所设定的超时值时，置位超时标志 PCT_CSR[CF]，产生触发输出。如果使能了超时中断 PCT_CSR[CIE]，将产生超时中断。

脉冲检测极性可以通过 PCT_PSR[PPS]进行设置。

35.4.3 触发/脉冲源选择

可以为 PCT 选择不同的触发输入或脉冲输入：

- 选择内部触发输入：配置 PCT_CSR[IPS]选择 pulse input 0，再在 TRGMUX_PCT[SEL_PCT0]选择 PCT 触发输入来自哪个片内模块；
- 选择片外脉冲：配置[IPS]选择片外输入 pulse input 1~3。

35.4.4 触发/脉冲极性配置

当 PCT 工作在脉冲计数模式时，可以通过 PCT_CSR[PPS]对输入源检测极性进行配置：

[PPS]=0：输入源高有效，输入源出现上升沿且高电平持续超过滤波宽度时，CNR 递增一次；

[PPS]=1：输入源低有效，输入源出现下降沿且低电平持续超过滤波宽度时，CNR 递增一次。

35.4.5 超时值设置

可以通过 PCT_CMR[CMR]为 PCT 设置超时值，向上计数到达超时值会置位超时标志 PCT_CSR[CF]，产生触发输出，若同时使能了 PCT 的中断 PCT_CSR[CIE]、则还会产生相应的中断；

CNR 达到 CMR 超时值后是否清零取决于 PCT_CSR[FRE]的设置值。

超时值更新后将立即生效。

35.4.6 Free-running 模式

1. 当 PCT_CSR[FRE]=1 即 free-running 模式使能时候，PCT CNR 在 16-bit 全范围（即 0xFFFF）内计数，CNR 到达 CMR 值之后继续递增直到 0xFFFF 才会清零。此时计数周期为：

© 2013 - 2024 杰发科技有限公司

the timing period(Unit: second) = (0xFFFF + 1) / timing clock frequency(Unit: Hz)

但 free-running 模式下中断的产生时刻还是受 CNR 是否等于 CMR 值、进而 PCT_CSR[CF]是否被置位控制。使能 free-running 模式时需要设置[BYP]=1 关闭分频功能。

2. 当 PCT_CSR[FRE]=0 即 free-running 模式禁用时候，PCT CNR 在 CMR 设置值范围内计数，CNR 到达 CMR 值之后即会清零。此时计数周期为：

the timing period(Unit: second) = (CMR + 1) / timing clock frequency(Unit: Hz)

35.4.7 分频/滤波功能

35.4.7.1 分频功能

仅对时钟计数器模式有效。通过 PCT_PSR[BYS]对分频功能进行使能/关闭：

- 当 [BYP]=0 使能分频功能时候，会对时钟源做 $2^{(N+1)}$倍数的分频，即实时计数值 CNR 会在每 $2^{(N+1)}$个时钟源周期递增一次；
- 当 [BYP]=1 关闭分频功能时候，直接对所选时钟源进行计数，即实时计数值 CNR 会在每个时钟源周期都会递增一次。

其中 N 值在 PCT_PSR[PS_VAL]设置，分频功能中 N 取值范围 0~15。

35.4.7.2 滤波功能

仅对脉冲计数器模式有效。通过 PCT_PSR[BYS]对滤波功能进行使能/关闭：

- 当 [BYP]=0 使能滤波功能时候：
  - 若配置为检测上升沿，高电平小于 2^N 个时钟源周期的脉冲将会被过滤掉，不会使实时计数值 CNR 递增；
  - 若配置为检测下降沿，低电平小于 2^N 个时钟源周期的脉冲将会被过滤掉，不会使实时计数值 CNR 递增。
- 当 [BYP]=1 关闭滤波功能时候，直接对所选脉冲源进行计数，即实时计数值 CNR 在每个脉冲输入时都会递增一次。

其中 N 值在 PCT_PSR[PS_VAL]设置，滤波功能中 N 取值范围 1~15。

35.4.8 实时计数值可获取

通过 PCT_CNR[CNR]可以随时获取 PCT 的实时计数值。

© 2013 - 2024 杰发科技有限公司

读取 CVAL 值时，需要通过 PCT_PSR [CVAL_SEL]设置读取的分频 SEL 值，不同的 BUS_CLK 跟功能时钟频率，需要按照寄存器描述配置不同的 SEL 值。

说明：脉冲计数器模式，SEL 值必须设置为 00.

35.4.9 触发输出

1. 脉宽：

   时钟计数模式时，触发输出的脉宽是经过分频器后的计数源周期；

   脉冲计数模式时，触发输出的脉宽是输入的触发输入或脉冲输入信号的周期。

2. 周期：

   触发输出的周期就是 CNR 清零的周期。

3. 用途：

   PCT 的触发输出可以通过 CTU 模块再配置为其他模块的触发输入。

35.4.10 中断请求

定时器中断可通过置位 PCT_CSR[CIE]来使能。当 PCT 发生超时， PCT_CSR[CF]超时标志置位为 1，写入 1 可清零。

表 35-1 PCT 中断汇总

中断源	标志	使能	全局使能
PCT 超时	CF	CIE	

35.4.11 低功耗模式

VLPR 模式只有 HSIDIV2_CLK 和 LSI1K_CLK 两种时钟源可以选择。三种停止模式 PCT 唤醒源如下：

表 35-2 PCT 唤醒源

模式	唤醒源
STOP1	PCT 超时中断
STOP2	PCT 超时中断
VLPS	PCT 超时中断

© 2013 - 2024 杰发科技有限公司

35.4.12 调试模式

若 PCT 被配置为时钟计数模式，当进入 Debug Halt 状态时候，PCT 将停止运行；

若 PCT 被配置为脉冲计数模式，当进入 Debug Halt 状态时候，PCT 将继续运行。

35.5 应用说明

35.5.1 时钟计数器配置流程

以功能时钟选择 VHSIDIV2_CLK（48MHZ）为例，要得到 128 毫秒的计时周期，可以按照下面步骤配置：

1. clock 模块寄存器 CKGEN_PERI_CLK_MUX1[PCT_MUX] 选择 110b: VHSI DIV2 CLK，PCT_PSR[PCS]选择 11b: PCT_CLK；
2. 配置 PCT_CSR[MS]=0：时钟计数器模式；
3. 配置 PCT_CMR[CMR]=48000-1；
4. 配置 PCT_PSR[BYP]=0 使能分频功能，PCT_PSR[PS_VAL]=6；
5. 配置 PCT_CSR[FRE]=0free-running 模式禁用；
6. 配置 PCT_PSR[CIE]=1 使能中断；
7. 配置 PCT_CSR[PCT_EN]=1 使能 PCT 模块，开始计数

至此，得到计数周期为 $48000*2^{(6+1)}/48M=128000\mu s=128ms$

35.5.2 脉冲计数器配置流程

以 PCT 功能时钟选择 SPLLDIV2_CLK（24MHZ）为例，触发源选择 Timer0 通道，可以按照下面步骤配置：

1. 配置 timer0 通道为双 16-bit 周期计数器，超时值 TIMER_CH0_TVAL[TVAL]=0x5DC0BB80（其中，0x5DC0=24000，0xBB80=48000）,Timer0 输出高低电平分别为 1 毫秒和 2 毫秒的触发输出，周期 3 毫秒；
2. clock 模块寄存器 CKGEN_PERI_CLK_MUX1[PCT_MUX] 选择 111b: SPLL DIV2 CLK，PCT_PSR[PCS]选择 11b: PCT_CLK；
3. 配置 PCT_CSR[MS]=1 选择脉冲计数器模式；
4. 配置 PCT_CMR[CMR]=9；
5. 配置 PCT_CSR[IPS]选择 pulse input 0 即片内触发源，再配置 CTU 模块寄存器 TRGMUX_PCT [SEL_PCT0]=16，即 PCT 的触发源来自 Timer0；

© 2013 - 2024 杰发科技有限公司

6. 配置 PCT_PSR[BYP]=0 使能滤波功能，PCT_PSR[PS_VAL]=15，此时滤波宽度是(1/24)*2^{15}μs，约等于 1.365 毫秒；
7. 配置 PCT_CSR[PPS]=1，触发源下降沿来时实时计数值 PCT_CNR[CNR]递增一次

   （若极性设置为上升沿，则由于输入触发源高电平只有 1 毫秒，会被滤波掉，所以 PCT_CNR[CNR]不会递增）；
8. 配置 PCT_CSR[FRE]=0 禁用 free-running 模式；
9. 配置 PCT_PSR[CIE]=1 使能中断；
10. 配置 PCT_CSR[PCT_EN]=1 使能 PCT 模块，开始计数

至此，得到计数周期为 3*(9+1)=30ms，每 30 毫秒 PCT 输出脉宽为 3 毫秒的触发信号。

35.6 寄存器定义

PCT 寄存器只支持 32 位访问。

表 35-3 PCT 寄存器映射

PCT 基地址 = 0x40019000

地址	名称	宽度	描述
PCT 基地址+0x00	PCT_CSR	32	控制器寄存器
PCT 基地址+0x04	PCT_PSR	32	分频滤波设置寄存器
PCT 基地址+0x08	PCT_CMR	32	超时值寄存器
PCT 基地址+0x0C	PCT_CNR	32	当前值寄存器

35.6.1 控制寄存器(PCT_CSR)

表 35-4 PCT_CSR 寄存器

PCT_CSR **控制寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								CF		CIE	IPS		PPS	FRE	MS	PCT_EN
访问								R/W1C		RW	RW		RW	RW	RW	RW
复位值								0		0	0	0	0	0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
8 CF	**超时标志位** 0：实时计数值不等于 CMR 并递增 1：实时计数值等于 CMR 并递增 当实时计数值递增达到 CMR 设置值时并继续递增时，CF 被置位；写 1 可以清除该标志。
6 CIE	**PCT 中断使能控制** 0：禁止 1：使能 **说明：当 CIE=1，CF 被置位 1 时候将产生中断。**
5:4 IPS	**输入源选择** 00b：片内其他模块的触发输出 01b：片外引脚脉冲输入 1 10b：片外引脚脉冲输入 2 11b：片外引脚脉冲输入 3 **说明：只能在 PCT 未使能时候修改 IPS 配置**
3 PPS	**输入源极性选择** 0：输入源高有效，CNR 在输入源上升沿递增 1：输入源低有效，CNR 在输入源下降沿递增 **说明：只能在 PCT 模块未使能时候修改 PPS 配置**
2 FRE	**Free-running 模式使能控制** 0：free-running 模式不使能，在 CF 置位时候，CNR 将复位到 0 1：free-running 模式使能，当 CNR 计数到全范围（即 0xFFFF）时，CNR 将复位到 0
1 MS	**PCT 工作模式选择** 0：时钟计数模式 1：脉冲计数模式
0 PCT_EN	**PCT 模块使能控制** 0：禁止 1：使能

© 2013 - 2024 杰发科技有限公司

35.6.2 分频滤波设置寄存器(PCT_PSR)

表 35-5 PCT_PSR 寄存器

PCT_PSR 分频滤波设置寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								BYP	PS_VAL				CVAL_SEL		PCS	
访问								RW	RW				RW		RW	
复位值								0	0	0	0	0	0	0	0	0

字段	说明
8 BYP	**分频/滤波功能使能控制** 0：使能分频/滤波功能 1：禁用分频/滤波功能 当 BYP=0，CNR 计数分频/滤波后的输出； 当 BYP=1，CNR 直接计数 PCS 所选时钟源（时钟计数模式下）、或者 IPS 所选输入源（脉冲计数模式下）； 只有 PCT 模块未使能时候才可以更改 BYP 配置。
7:4 PS_VAL	**分频/滤波值** 0000b: 对所选时钟源做 2 分频; 滤波功能时候不支持此配置值. 0001b: 对所选时钟源做 4 分频; 脉冲计数模式下对大于等于 2 个时钟周期的输入才递增 CNR 0010b:对所选时钟源做 8 分频; 脉冲计数模式下对大于等于 4 个时钟周期的输入才递增 CNR 0011b: 对所选时钟源做 16 分频; 脉冲计数模式下对大于等于 8 个时钟周期的输入才递增 CNR 0100b: 对所选时钟源做 32 分频; 脉冲计数模式下对大于等于 16 个时钟周期的输入才递增 CNR 0101b: 对所选时钟源做 64 分频; 脉冲计数模式下对大于等于 32 个时钟周期的输入才递增 CNR 0110b: 对所选时钟源做 128 分频; 脉冲计数模式下对大于等于 64 个时钟周期的输入才递增 CNR 0111b: 对所选时钟源做 256 分频; 脉冲计数模式下对大于等于 128 个时钟周期的输入才递增 CNR 1000b: 对所选时钟源做 512 分频; 脉冲计数模式下对大于等于 256 个时钟周期的输入才递增 CNR 1001b: 对所选时钟源做 1024 分频; 脉冲计数模式下对大于等于 512 个时钟周期的输入

© 2013 - 2024 杰发科技有限公司

字段	说明
	才递增 CNR 1010b: 对所选时钟源做 2048 分频; 脉冲计数模式下对大于等于 1024 个时钟周期的输入才递增 CNR 1011b: 对所选时钟源做 4096 分频; 脉冲计数模式下对大于等于 2048 个时钟周期的输入才递增 CNR 1100b: 对所选时钟源做 8192 分频; 脉冲计数模式下对大于等于 4096 个时钟周期的输入才递增 CNR 1101b: 对所选时钟源做 16,384 分频; 脉冲计数模式下对大于等于 8192 个时钟周期的输入才递增 CNR 1110b: 对所选时钟源做 32,768 分频; 脉冲计数模式下对大于等于 16,384 个时钟周期的输入才递增 CNR 1111b: 对所选时钟源做 65,536 分频; 脉冲计数模式下对大于等于 32,768 个时钟周期的输入才递增 CNR **说明：只有 PCT 未使能时候才可以更改 PS_VAL 配置。**
3:2 CVAL_SEL	**PCT 读取 CVAL 值 SEL 设置** 00b: BUSCLK/Function clk >= 4 01b: BUSCLK/Function clk >= 3 10b: BUSCLK/Function clk >= 1 **说明： PCT 配置为脉冲计数器模式时，CVAL_SEL 必须设置为 00。**
1:0 PCS	**PCT 时钟源选择** 00b：HSIDIV2_CLK 01b：LSI1K_CLK 10b：RTC_CLK 11b：PCT_CLK (CKGEN 模块为 PCT 配置的时钟源) **说明：只有 PCT 未使能时候才可以更改 PCS 配置。**

© 2013 - 2024 杰发科技有限公司

35.6.3 超时值寄存器(PCT_CMR)

表 35- 6 PCT_CMR 寄存器

PCT_CMR 超时值寄存器 复位值：

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CMR															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15:0 CMR	**PCT 超时值** 当 PCT 使能的时候，CNR=CMR 并继续递增时候 CF 被置位 1，并产生触发输出； 如果 CMR=0，触发输出将会一直持续直到 PCT 被关闭

35.6.4 当前值寄存器 (PCT_CNR)

表 35- 7 PCT_CNR 寄存器

PCT_CNR 当前值寄存器 复位值：

0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	CNR															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
15:0 CNR	**PCT 当前值** 当被读取时候，CNR 返回实时计数值

© 2013 - 2024 杰发科技有限公司

36 实时计数器（RTC）

36.1 简介

实时计数器（Real Time Clock），主要功能是实时计数，可以用于定时或闹钟功能。在 STOP1、STOP2、VLPS、Standby（Standby 模式仅 AC78408/9 支持）模式下，RTC 具有保持运行并唤醒 MCU 的功能。

36.2 特性

- 32 位向上计数器
- 可编程 20 位预分频器
- 可编程 32 位闹钟计数器
- 寄存器写保护
- 支持外部管脚时钟输入
- 支持 RTC 时钟输出

36.3 结构框图

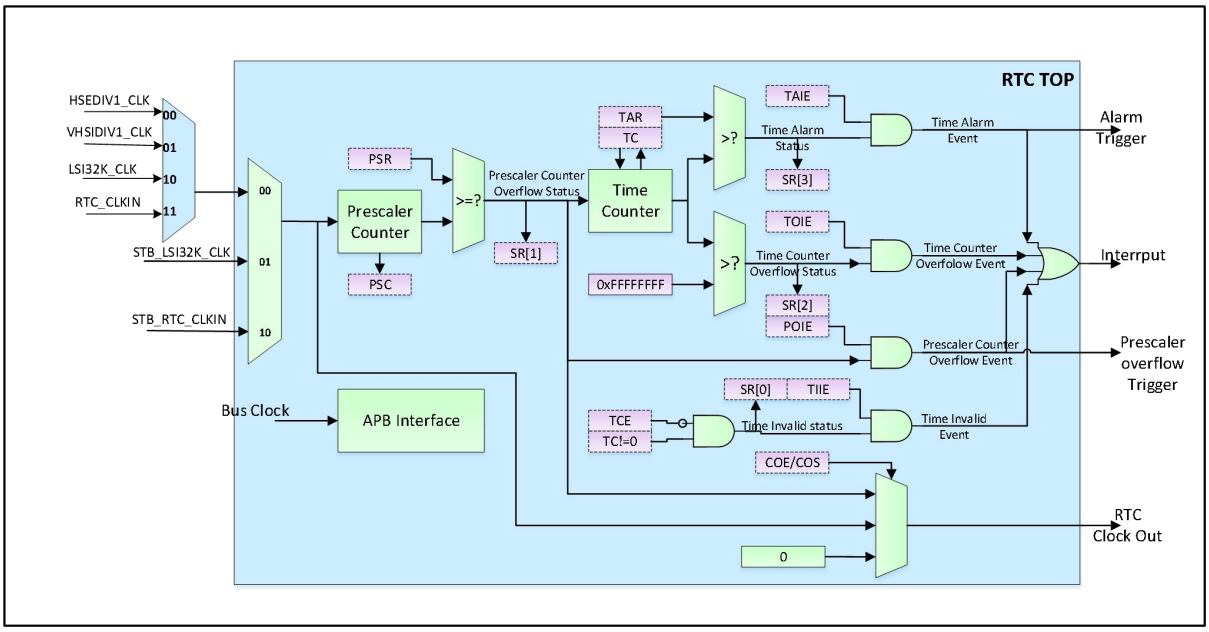


图 36- 1 AC78408/9 RTC 结构框图

© 2013 - 2024 杰发科技有限公司

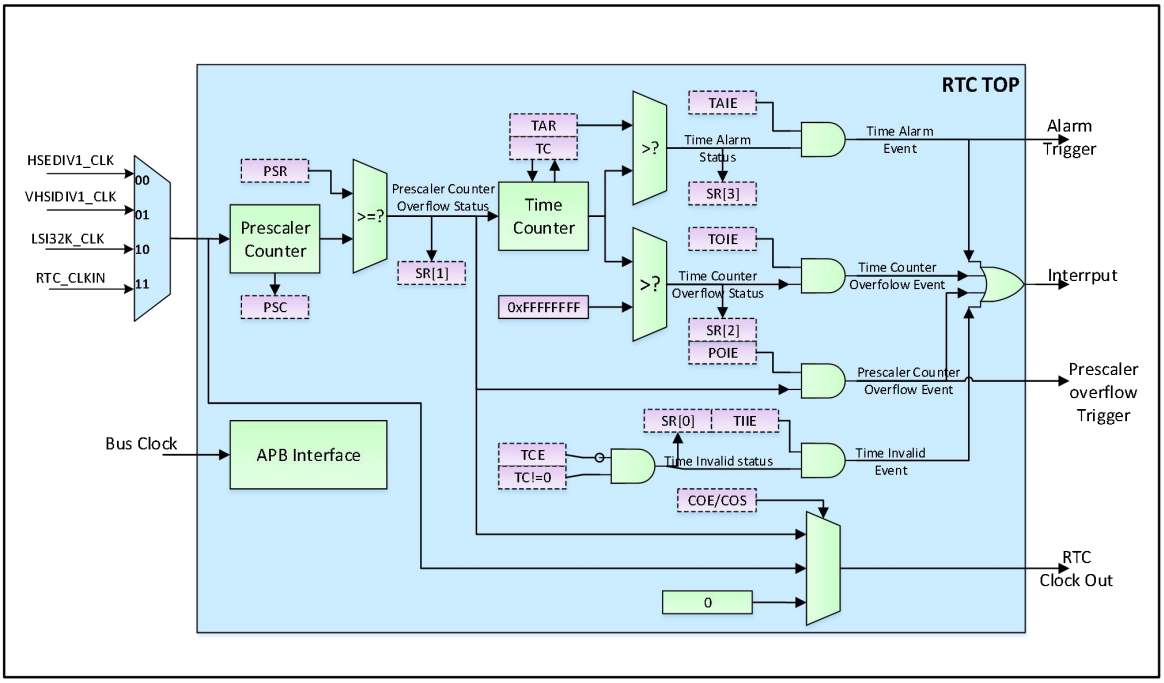


图 36- 2 AC78406/7 RTC 结构框图

36.4 功能描述

36.4.1 时钟源选择

在 AC78408/9 中 RTC 模块有六种时钟源可选择使用：

一：RTC_CTRL[CLKSEL]设置 00，可以选择来自 CKGEN 的四个不同的时钟源，分别为内部低速时钟（LSI）4 分频、外部高速时钟（HSE）、内部超高速时钟（VHSI）及外部 RTC_CLKIN 引脚输入时钟。需注意，来自 CKGEN 的四个时钟源无法唤醒 Standby 模式。

二：RTC_CTRL[CLKSEL]设置 01，可以选择来自 SPM 的 LSI32K 时钟源，该时钟源主要用于需要唤醒 Standby 的场景，也可以作为普通时钟源使用。

三：RTC_CTRL[CLKSEL]设置 10，可以选择来自 SPM 的 RTC_CLKIN 时钟源，该时钟源主要用于需要唤醒 Standby 的场景，也可以作为普通时钟源使用。

在 AC78406/7 中 RTC 模块有四种时钟源可选择使用：内部低速时钟（LSI）4 分频、外部高速时钟（HSE）、内部超高速时钟（VHSI）及外部 RTC_CLKIN 引脚输入时钟。

36.4.2 软件复位

置位 RTC_CTRL[SWR]将复位 RTC 模块所有寄存器，等效于 POR 复位。该位不受软件复位影响，需要软件清除。

© 2013 - 2024 杰发科技有限公司

36.4.3 计时特性

RTC 由一个 32 位实时计数器和一个 20 位预分频器组成。实时计数器是一个 32 位向上增加的计数器，当实时计数器计数到 0xFFFFFFFF 溢出后，将产生实时计数器溢出标志。如果使能了实时计数器溢出中断，则产生一次实时计数器溢出中断请求，溢出后若想再次使能 RTC 计时需要先写 RTC_SR[TCE] = 0，并对实时计数寄存器写入非 0 值清除溢出标志，再置位 RTC_SR[TCE]使能 RTC 开始计时。预分频器是一个 20 位向上计数的计数器，当达到预设的预分频值后实时计数器值加一，同时产生预分频溢出标志，如果使能了预分频中断，则产生一次预分频中断请求，预分频器计数溢出后将从 0 开始新的计数。

实时计数寄存器和预分频器寄存器只有在 RTC_SR[TCE] = 0 时，才可被写入。当实时计数器溢出后，预分频计数寄存器将停止递增，读取实时计数寄存器和预分频计数寄存器的值为 0。软件通过对实时计数寄存器写入非 0 值清除溢出标志，对 RTC_SR[POF]写 1 清除预分频器中断标志。

默认系统上电运行时 RTC_SR[TIF] = 1，若使能 RTC 无效中断，则产生 RTC 无效中断请求，可通过软件对 RTC_TC 寄存器写入非 0 值清除标志。RTC_SR[TIF] = 1 时，写 RTC_SR[TCE] = 1 不会启动 RTC 计数。

36.4.4 闹钟事件

RTC 支持一个 32bit 可编程的闹钟计数器，当实时计数器计数值达到设定的闹钟值并向上增加时产生闹钟溢出标志，如果使能了 RTC 闹钟中断，则产生一次闹钟中断请求。写 RTC_TAR 寄存器可清除此标志。

36.4.5 寄存器写保护

RTC 支持对锁定寄存器、控制寄存器、状态寄存器进行写保护。软件通过写锁定寄存器设置被锁定的寄存器。对被锁定的寄存器的写访问将被忽略，并且不会产生总线错误，直到系统产生 POR 或软件复位，被锁定的寄存器才会被解锁。

需要注意的是当控制寄存器被锁定后，软件复位将被禁用。

控制寄存器中的更新模式字段允许在状态寄存器被锁定时对 TCE 字段的软件写访问。当 RTC_CTRL[UPM] = 0 时，只有 RTC_LR[SRL] = 0 时，才能写入 RTC_SR[TCE]。当 RTC_CTRL[UPM] = 1 时，RTC_SR[TCE] = 0 或 RTC_SR[TOF] = 1 时，RTC_SR[TCE]也可写入。状态寄存器的预分频中断标志位不会被锁定。

36.4.6 RTC 时钟输出

部分管脚可以复用为 RTC_CLKOUT 功能。复用后，RTC 可通过置位 RTC_CTRL[COE] 来使能 RTC_CLKOUT 功能，写 RTC_CTRL[COS]寄存器选择输出 RTC 时钟还是 RTC 预分频器溢出。

© 2013 - 2024 杰发科技有限公司

36.4.7 中断请求

RTC 模块共有 4 个中断源。

表 36-1 RTC 中断汇总

中断源	标志	本地使能	全局使能
闹钟中断	TAF	TAIE	
实时计数器溢出中断	TOF	TOIE	
预分频中断	POF	POIE	
RTC 无效中断	TIF	TIIE	

36.4.8 低功耗模式

在 STOP1/STOP2/VLPS/Standby（Standby 模式仅 AC78408/9 支持）模式下，除无效中断外，任意 RTC 中断都可唤醒 MCU。从 STOP1/STOP2/VLPS/Standby 模式唤醒后，RTC 会继续保持工作。

表 36-2 RTC 唤醒源

模式	唤醒源
STOP1	除无效中断外的所有中断
STOP2	除无效中断外的所有中断
VLPS	除无效中断外的所有中断
Standby（仅 AC78408/9 支持）	除无效中断外的所有中断

36.4.9 调试模式

RTC 模块没有调试模式配置，在进入调试模式时可正常运行。

36.5 应用说明

36.5.1 RTC 基本使用

使用前，需要先在时钟模块中设置 RTC 时钟源，使能配置 RTC 寄存器的时钟，再配置 RTC_CTRL、RTC_TC、RTC_PSR、RTC_SR 等寄存器，初始化 RTC 模块的中断、计数值、预分频值、闹钟值等，最后置位 RTC_SR[TCE]，使能 RTC 开始计数。

RTC 预分频计数器溢出时，标志位 RTC_SR[POF]置位，写 1 清除标志。RTC 计数器溢出时，标志位 RTC_SR[TOF]置位，需先设置 RTC_SR[TCE] = 0，再对 TC 寄存器写非 0 值清除标志。RTC 闹钟计数器溢出时，标志位 RTC_SR[TAF]置位，写 RTC_TAR 寄存器清除标志。在计数器运行时，对 RTC_TC 或 RTC_PSR 的写操作无效。

© 2013 - 2024 杰发科技有限公司

RTC_CTRL[COE]位为 1，RTC_CTRL[COS]位为 0，使能 RTC 预分频溢出翻转指定 GPIO，GPIO 同时需要复用为 RTC_CLKOUT 功能。

36.5.2 RTC 低功耗唤醒

RTC 任意中断可将 MCU 从低功耗状态下唤醒，需在 MCU 进入低功耗状态前使能 RTC 相关中断，同时选择该低功耗模式下还在工作的时钟作为时钟源并启动计时。

注意：
一旦使能 RTC 无效中断，中断立马产生，系统会先处理该中断。

36.5.3 编程指南

RTC 使用配置如下：

1. 配置实时计数器的值，向 RTC_TC 寄存器写入非 0 值；
2. 配置 RTC_TAR 寄存器；
3. 配置 RTC_PSR 寄存器；
4. 配置 RTC_CTRL 寄存器，使能 RTC 中断、RTC_CLKOUT 等；
5. 置位 RTC_SR[TCE]，使能 RTC 计数器。

36.6 寄存器定义

RTC 寄存器只支持 32 位访问。

表 36-3 RTC 寄存器映射

RTC 基地址：0x40009800

地址	名称	宽度	描述
RTC 基地址+0x00	RTC_CTRL	32	控制寄存器
RTC 基地址+0x04	RTC_TAR	32	闹钟寄存器
RTC 基地址+0x08	RTC_TC	32	实时计数寄存器
RTC 基地址+0x0C	RTC_PSR	32	预分频器寄存器
RTC 基地址+0x10	RTC_PSC	32	预分频计数寄存器
RTC 基地址+0x14	RTC_SR	32	状态寄存器
RTC 基地址+0x18	RTC_LR	32	锁定寄存器

© 2013 - 2024 杰发科技有限公司

36.6.1 控制寄存器（RTC_CTRL）

表 36-4 RTC_CTRL 寄存器

RTC_CTRL 控制寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称															CLKSEL	
访问															RW	
复位值															0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									TAIE	TOIE	POIE	TIIE	COE	COS	UPM	SWR
访问									RW	RW	RW	RW	RW	RW	RW	RW
复位值									0	0	0	0	0	0	0	0

字段	说明
17:16 CLKSEL	**时钟源选择(仅 AC78408/9 支持)** 00：CKGEN (需要在 CKGEN 中选择需要的时钟源) 01：STB_LSI32K_CLK(用于唤醒 Standby 的 LSI32K 时钟源) 10：STB_RTC_CLKIN(用于唤醒 Standby 的 RTCIN 时钟源) 注意：该 bit 位为 write once, 只有 POR 复位才会复位该寄存器的该 bit 位。
7 TAIE	**闹钟中断使能** 0： 禁用闹钟中断请求 1： 使能闹钟中断请求 使能闹钟中断。如果 TAIE 置位，则在 TAF 置位时会产生中断请求。
6 TOIE	**实时计数器溢出中断使能** 0： 禁用实时计数器溢出中断请求 1： 使能实时计数器溢出中断请求 使能实时计数器溢出中断。如果 TOIE 置位，则在 TOF 置位时会产生中断请求。
5 POIE	**预分频中断使能** 0： 禁用预分频中断请求 1： 使能预分频中断请求 使能预分频中断。如果 POIE 置位，则在 POF 置位时会产生中断请求。
4 TIIE	**无效中断使能** 0： 禁用无效中断请求 1： 使能预无效中断请求

© 2013 - 2024 杰发科技有限公司

字段	说明
	使能无效中断。如果 TIIE 置位，则在 TIF 置位时会产生中断请求。
3 COE	**RTC 时钟输出使能** 0： 禁用 RTC 时钟输出 1： 使能 RTC 时钟输出
2 COS	**RTC 时钟输出选择** 0： 选择输出 RTC 预分频溢出 1： 选择输出 RTC 时钟
1 UPM	**更新模式** 0： 状态寄存器被锁定时，TCE 写无效 1：状态寄存器被锁定时，在特定条件下 TCE 写有效 特定条件指的是：TIF 或 TOF 位置 1，或 TCE 是禁用状态。
0 SWR	**软件复位使能** 0： 复位无效 1： 使能复位 **注意：** **该位会复位除去本位的所有 RTC 寄存器。该位只可被 POR 或软件清除。**

36.6.2 闹钟寄存器（RTC_TAR）

表 36-5 RTC_TAR 寄存器

RTC_TAR **闹钟寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	TAR[31: 16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TAR[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 TAR	**闹钟计数值** 当计数器被启用后，实时计数器的值等于闹钟计数值并且向上增加时，将置位 TAF。写入 TAR 将清除 TAF 标志。

© 2013 - 2024 杰发科技有限公司

36.6.3 实时计数寄存器（RTC_TC）

表 36-6 RTC_TC 寄存器

RTC_TC 实时计数寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	TC[31: 16]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	TC[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
31:0 TC	**实时计数值** 当计数器被启用，预分频器溢出后 TC 值向上增加，回读返回当前计数值。当 TC 溢出时，将置位 TOF。向 TC 写入非 0 值将清除 TOF 标志。 **注意：TC 只有在计数器被禁用时才能写入。**

36.6.4 预分频器寄存器（RTC_PSR）

表 36-7 RTC_PSR 寄存器

RTC_PSR 预分频器寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称													PSR[19: 16]			
访问													RW			
复位值													0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PSR[15: 0]															
访问	RW															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
19:0 PSR	**预分频值** 当预分频计数器等于预分频值时，预分频计数器溢出。 **注意：** **只有当计数器被禁用时，PSR 才能被写入。**

© 2013 - 2024 杰发科技有限公司

36.6.5 预分频计数寄存器（RTC_PSC）

表 36-8 RTC_PSC 寄存器

RTC_PSC 预分频计数寄存器 复位值：0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称													PSC [19: 16]			
访问													RO			
复位值													0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PSC [15: 0]															
访问	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
19:0 PSC	**预分频计数值** PSC 按照 RTC 功能时钟向上计数，读该寄存器返回分频器计数器当前计数值。当分频器计数器等于 PSR 并递增时，PSC 将溢出。

36.6.6 状态寄存器（RTC_SR）

表 36-9 RTC_SR 寄存器

RTC_SR 状态寄存器 复位值：0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称												TCE	TAF	TOF	POF	TIF
访问												RW	RO	RO	R/W1C	RO
复位值												0	0	0	0	1

字段	说明
4 TCE	**RTC 计数器使能** 0：禁用 RTC 计数器 1：使能 RTC 计数器

© 2013 - 2024 杰发科技有限公司

字段	说明
3 TAF	**闹钟中断标志位** 0：RTC 闹钟计数器没有达到 RTC 闹钟寄存器的值 1：RTC 闹钟计数器达到 RTC 闹钟寄存器的值 该状态位表示 RTC 闹钟计数器达到了 RTC 闹钟寄存器中的值。写闹钟寄存器可清除此位。
2 TOF	**实时计数器溢出中断标志位** 0：RTC 时钟计数器没有溢出 1：RTC 实时计数器溢出 该状态位表示 RTC 实时计数器溢出。对实时计数寄存器写入非 0 值可清除此位。
1 POF	**预分频中断标志** 0： RTC 预分频器计数器没有达到 RTC 预分频器寄存器的值 1： RTC 预分频器计数器达到 RTC 预分频器寄存器的值 该状态位表示 RTC 预分频器计数器达到了 RTC 预分频器寄存器中的值。写入逻辑 0 无效。写入逻辑 1 会清除此位。
0 TIF	**无效标志** 0： RTC 有效 1： RTC 无效 该状态位在 POR 或软件复位后置位。当该位为 1 时，TC 和 PSC 不会递增，计数器不会计数。在使能 RTC 计数前，软件必须通过向 TC 写入非 0 值来清除此位。

36.6.7 锁定寄存器（RTC_LR）

表 36-10 RTC_LR 寄存器

RTC_LR **锁定寄存器** **复位值：0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
复位值																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称														LRL	SRL	CRL
访问														RW	RW	RW
复位值														0	0	0

字段	说明

© 2013 - 2024 杰发科技有限公司

字段	说明
2 LRL	**锁定寄存器锁定位** 0：锁定寄存器没有被锁定 1：锁定寄存器被锁定，写锁定寄存器无效 一旦该位置位，只能被 POR 或软件复位清除。
1 SRL	**状态寄存器锁定位** 0：状态寄存器没有被锁定 1：状态寄存器被锁定，写状态寄存器无效 一旦该位置位，只能被 POR 或软件复位清除。状态寄存器中的 POF 标志位在锁定状态下也可被软件写 1 清除。当 UPM 置位时，TCE 在特定条件下仍可写即使状态寄存器被锁定。
0 CRL	**控制寄存器锁定位** 0：控制寄存器没有被锁定 1：控制寄存器被锁定，写控制寄存器无效 一旦该位置位，只能被 POR 清除。因为控制寄存器被锁定后，软件复位被禁用。

© 2013 - 2024 杰发科技有限公司

37 调试(Debug)

37.1 简介

该器件基于 ARM CoreSight 架构进行调试。外部调试器通过调试接口访问寄存器和存储器，控制程序运行/停止/复位等操作。

37.2 特性

- JTAG 调试接口和 SWD 串行线调试接口
- 6 个硬件断点
- 4 个数据观察点
- ITM 指令跟踪宏单元
- SWO 串行线输出
- 调试安全性

37.3 功能描述

37.3.1 调试安全性

使能 Flash 读保护后，调试端口的功能将受限，debug 模块不能访问整个 MCU 资源，这样能有效保护 Flash 数据。

37.3.2 调试复位

调试模块只会被 POR 复位。

37.3.3 低功耗调试

STOP1/STOP2/VLPS 模式下会断开 debug 连接，不支持调试访问寄存器和内存。

© 2013 - 2024 杰发科技有限公司