HDSC 华大半导体
HUADA SEMICONDUCTOR

HC32F17 系列 / HC32F19 系列

32 位 ARM® Cortex®-M0+ 微控制器

用户手册

前　言

非常感谢大家对华大半导体产品的支持和信赖。

使用本系列产品前，请系统阅读本手册和“数据手册”。

- 本手册的目的和对象读者

  本手册主要介绍本系列的功能、操作事项和使用方法。对象读者为使用本系列实际开发产品的工程师。

  ※ 本手册介绍外设功能的构成和操作说明，但不包括该系列的规格说明。关于芯片规格，详情参见其对应的“数据手册”。

- 样本程序和开发环境

  华大半导体提供外设功能运行用的样本程序和本系列所需的开发环境说明。关于华大微控制器的运行规格和使用方法，请联系本公司。

- 微控制器支持信息：http://www.hdsc.com.cn/mcu.htm

- 华大半导体有限公司（以下简称：“HDSC”）保留随时更改、更正、增强、修改华大半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。HDSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- 用户对 HDSC 产品的选择和使用承担全部责任，用户将 HDSC 产品用于其自己或指定第三方产品上的，HDSC 不提供服务支持且不对此类产品承担任何责任。
- HDSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- HDSC 产品的转售，若其条款与此处规定不同，HDSC 对此类产品的任何保修承诺无效。
- 任何带有“®”或“™”标识的图形或字样是 HDSC 的商标。所有其他在 HDSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- 本通知中的信息取代并替换先前版本中的信息。

©2019 华大半导体有限公司 - 保留所有权利

目　录

前　言 2
目　录 3
产品特性 31
1　功能模块 32
1.1　32 位 CORTEX M0+ 内核 33
1.2　128K / 256K Byte FLASH 33
1.3　16K / 32K Byte RAM 33
1.4　时钟系统 33
1.5　工作模式 34
1.6　通用 IO 端口 34
1.7　中断控制器 34
1.8　复位控制器 35
1.9　DMAC 36
1.10　定时器/计数器 36
1.11　看门狗 WDT 38
1.12　通用异步收发器 UART0~UART3 38
1.13　同步串行接口 SPI 39
1.14　I2C 总线 39
1.15　蜂鸣器 Buzzer 39
1.16　时钟校准电路 40
1.17　唯一识别号 UID 40
1.18　CRC16/32 硬件循环冗余校验码 40
1.19　AES 硬件加密 40
1.20　TRNG 真随机数发生器 40
1.21　12 Bit SARADC 41
1.22　12 Bit DAC 41
1.23　电压比较器 VC 41
1.24　低电压检测器 LVD 42
1.25　运放 OPA 42
1.26　LCD 驱动 42
1.27　嵌入式调试系统 43
1.28　编程模式 43
1.29　高安全性 43
2　引脚配置及功能 44
2.1　封装示意图 44
2.2　引脚功能说明 50
2.3　模块信号说明 62
3　系统结构 64
3.1　概述 64
3.2　系统地址划分 65

3.3 存储器和模块地址分配.....67
4 工作模式.....69
4.1 运行模式.....71
4.2 休眠模式.....73
4.3 深度休眠模式.....75
5 系统控制器（SYSCTRL）.....78
5.1 系统时钟介绍.....78
5.1.1 内部高速 RC 时钟 RCH.....79
5.1.2 内部低速 RC 时钟 RCL.....80
5.1.3 外部低速晶振时钟 XTL.....80
5.1.4 外部高速晶振时钟 XTH.....80
5.1.5 锁相环时钟 PLL.....80
5.1.6 时钟启动过程.....81
5.2 系统时钟切换.....82
5.2.1 标准的时钟切换流程.....82
5.2.2 RCH 不同振荡频率间切换流程.....83
5.2.3 从其它时钟切换到 XTL 示例.....83
5.2.4 从其它时钟切换到 XTH 示例.....84
5.2.5 从其它时钟切换到 RCL 示例.....85
5.2.6 从其它时钟切换到 RCH 示例.....85
5.2.7 PLL 与 RCH 相互切换示例，参考时钟为 RCH.....85
5.2.8 PLL 与 XTH 相互切换示例，参考时钟为 XTH.....86
5.3 时钟校准模块.....88
5.4 中断唤醒控制.....89
5.4.1 从深度休眠模式唤醒后执行中断服务程序的方法.....89
5.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法.....89
5.4.3 使用退出休眠特性.....90
5.5 寄存器.....92
5.5.1 系统控制寄存器 0（SYSCTRL0）.....93
5.5.2 系统控制寄存器 1（SYSCTRL1）.....95
5.5.3 系统控制寄存器 2（SYSCTRL2）.....96
5.5.4 RCH 控制寄存器（RCH_CR）.....97
5.5.5 XTH 控制寄存器（XTH_CR）.....98
5.5.6 RCL 控制寄存器（RCL_CR）.....99
5.5.7 XTL 控制寄存器（XTL_CR）.....100
5.5.8 PLL 控制寄存器（PLL_CR）.....101
5.5.9 外围模块时钟控制寄存器 0（PERI_CLKEN0）.....103
5.5.10 外围模块时钟控制寄存器 1（PERI_CLKEN1）.....105
6 复位控制器（RESET）.....106
6.1 复位控制器介绍.....106
6.1.1 上电下电复位 POR.....106
6.1.2 外部复位管脚复位.....107

6.1.3 WDT 复位 107
6.1.4 PCA 复位 107
6.1.5 LVD 低电压复位 107
6.1.6 Cortex-M0+ SYSRESETREQ 复位 107
6.1.7 Cortex-M0+ LOCKUP 复位 107
6.2 寄存器 108
6.2.1 复位标识寄存器（RESET_FLAG） 108
6.2.2 外围模块复位控制寄存器 0（PERI_RESET0） 110
6.2.3 外围模块复位控制寄存器 1（PERI_RESET1） 112
7 中断控制器（NVIC） 113
7.1 概述 113
7.2 中断优先级 114
7.3 中断向量表 115
7.4 中断输入和挂起行为 116
7.5 中断等待 119
7.6 中断源 120
7.7 中断结构图 122
7.8 寄存器 124
7.8.1 中断使能设置寄存器（NVIC_ISER） 124
7.8.2 中断使能清除寄存器（NVIC_ICER） 125
7.8.3 中断挂起状态设置寄存器（NVIC_ISPR） 125
7.8.4 中断挂起状态清除寄存器（NVIC_ICPR） 126
7.8.5 中断优先级寄存器（NVIC_IPR0） 127
7.8.6 中断优先级寄存器（NVIC_IPR1） 128
7.8.7 中断优先级寄存器（NVIC_IPR2） 129
7.8.8 中断优先级寄存器（NVIC_IPR3） 130
7.8.9 中断优先级寄存器（NVIC_IPR4） 131
7.8.10 中断优先级寄存器（NVIC_IPR5） 132
7.8.11 中断优先级寄存器（NVIC_IPR6） 133
7.8.12 中断优先级寄存器（NVIC_IPR7） 134
7.8.13 中断屏蔽特殊寄存器（PRIMASK） 135
7.9 软件基本操作 136
7.9.1 外部中断使能 136
7.9.2 NVIC 中断使能和清除使能 136
7.9.3 NVIC 中断挂起和清除挂起 136
7.9.4 NVIC 中断优先级 136
7.9.5 NVIC 中断屏蔽 137
8 端口控制器（GPIO） 138
8.1 端口控制器简介 138
8.2 端口控制器主要特性 138
8.3 端口控制器功能描述 139
8.3.1 端口配置功能 139

8.3.2 端口的写入 142
8.3.3 端口的读取 143
8.3.4 端口数字复用功能 144
8.3.5 端口中断功能 147
8.4 端口配置操作流程 148
8.4.1 端口复用配置为模拟端口操作流程 148
8.4.2 端口复用配置为数字通用端口操作流程 148
8.4.3 端口复用配置为数字功能端口操作流程 148
8.4.4 端口复用配置为调试测试端口操作流程 148
8.4.5 端口复用配置为红外输出信号操作流程 148
8.4.6 端口高电平中断操作流程 149
8.4.7 端口低电平中断操作流程 149
8.4.8 端口上升沿中断操作流程 149
8.4.9 端口下降沿中断操作流程 149
8.4.10 端口上拉使能配置操作流程 150
8.4.11 端口下拉使能配置操作流程 150
8.4.12 端口增强驱动配置操作流程 150
8.4.13 端口开漏输出配置操作流程 150
8.4.14 端口位置位操作流程 150
8.4.15 端口位清零操作流程 150
8.4.16 端口位置位清零操作流程 150
8.5 端口控制器寄存器描述 151
8.5.1 端口通用寄存器 156
8.5.1.1 端口 Px 输入输出配置寄存器(PxDIR) (x = A,B,C,D,E,F) 156
8.5.1.2 端口 Px 输入值寄存器(PxIN) (x = A,B,C,D,E,F) 156
8.5.1.3 端口 Px 输出值配置寄存器(PxOUT) (x = A,B,C,D,E,F) 157
8.5.1.4 端口 Px 数模配置寄存器(PxADS) (x = A,B,C,D,E,F) 157
8.5.1.5 端口 Px 位置位寄存器(PxBSET) (x = A,B,C,D,E,F) 158
8.5.1.6 端口 Px 位清零寄存器(PxBCLR) (x = A,B,C,D,E,F) 158
8.5.1.7 端口 Px 位置位清零寄存器(PxBSETCLR) (x = A,B,C,D,E,F) 159
8.5.1.8 端口 Px 上拉使能配置寄存器(PxPU) (x = A,B,C,D,E,F) 160
8.5.1.9 端口 Px 驱动能力配置寄存器(PxDR) (x = A,B,C,D,E,F) 160
8.5.1.10 端口 Px 下拉使能配置寄存器(PxPD) (x = A,B,C,D,E,F) 161
8.5.1.11 端口 Px 开漏输出配置寄存器(PxOD) (x = A,B,C,D,E,F) 161
8.5.1.12 端口 Px 高电平中断使能配置寄存器(PxHIE) (x = A,B,C,D,E,F) 162
8.5.1.13 端口 Px 低电平中断使能配置寄存器(PxLIE) (x = A,B,C,D,E,F) 162
8.5.1.14 端口 Px 上升沿中断使能配置寄存器(PxRIE) (x = A,B,C,D,E,F) 163
8.5.1.15 端口 Px 下降沿中断使能配置寄存器(PxFIE) (x = A,B,C,D,E,F) 163
8.5.1.16 端口 Px 中断状态寄存器(Px_STAT) (x = A,B,C,D,E,F) 164
8.5.1.17 端口 Px 中断清除寄存器(Px_ICLR) (x = A,B,C,D,E,F) 164
8.5.2 端口辅助寄存器 165
8.5.2.1 端口辅助功能配置寄存器 1(GPIO_CTRL1) 165

8.5.2.2 端口辅助功能配置寄存器 2(GPIO_CTRL2) 167
8.5.2.3 端口辅助功能定时器门控选择(GPIO_TIMGS) 169
8.5.2.4 端口辅助功能定时器 ETR 选择(GPIO_TIMES) 170
8.5.2.5 端口辅助功能定时器捕获输入选择(GPIO_TIMCPS) 171
8.5.2.6 端口辅助功能 PCA 捕获选择(GPIO_PCAS) 172
8.5.3 端口 PA 功能选择寄存器 173
8.5.3.1 端口 PA00 功能配置寄存器(PA00_SEL) 173
8.5.3.2 端口 PA01 功能配置寄存器(PA01_SEL) 174
8.5.3.3 端口 PA02 功能配置寄存器(PA02_SEL) 175
8.5.3.4 端口 PA03 功能配置寄存器(PA03_SEL) 176
8.5.3.5 端口 PA04 功能配置寄存器(PA04_SEL) 177
8.5.3.6 端口 PA05 功能配置寄存器(PA05_SEL) 178
8.5.3.7 端口 PA06 功能配置寄存器(PA06_SEL) 179
8.5.3.8 端口 PA07 功能配置寄存器(PA07_SEL) 180
8.5.3.9 端口 PA08 功能配置寄存器(PA08_SEL) 181
8.5.3.10 端口 PA09 功能配置寄存器(PA09_SEL) 182
8.5.3.11 端口 PA10 功能配置寄存器(PA10_SEL) 183
8.5.3.12 端口 PA11 功能配置寄存器(PA11_SEL) 184
8.5.3.13 端口 PA12 功能配置寄存器(PA12_SEL) 185
8.5.3.14 端口 PA13 功能配置寄存器(PA13_SEL) 186
8.5.3.15 端口 PA14 功能配置寄存器(PA14_SEL) 187
8.5.3.16 端口 PA15 功能配置寄存器(PA15_SEL) 188
8.5.4 端口 PB 功能选择寄存器 189
8.5.4.1 端口 PB00 功能配置寄存器(PB00_SEL) 189
8.5.4.2 端口 PB01 功能配置寄存器(PB01_SEL) 190
8.5.4.3 端口 PB02 功能配置寄存器(PB02_SEL) 191
8.5.4.4 端口 PB03 功能配置寄存器(PB03_SEL) 192
8.5.4.5 端口 PB04 功能配置寄存器(PB04_SEL) 193
8.5.4.6 端口 PB05 功能配置寄存器(PB05_SEL) 194
8.5.4.7 端口 PB06 功能配置寄存器(PB06_SEL) 195
8.5.4.8 端口 PB07 功能配置寄存器(PB07_SEL) 196
8.5.4.9 端口 PB08 功能配置寄存器(PB08_SEL) 197
8.5.4.10 端口 PB09 功能配置寄存器(PB09_SEL) 198
8.5.4.11 端口 PB10 功能配置寄存器(PB10_SEL) 199
8.5.4.12 端口 PB11 功能配置寄存器(PB11_SEL) 200
8.5.4.13 端口 PB12 功能配置寄存器(PB12_SEL) 201
8.5.4.14 端口 PB13 功能配置寄存器(PB13_SEL) 202
8.5.4.15 端口 PB14 功能配置寄存器(PB14_SEL) 203
8.5.4.16 端口 PB15 功能配置寄存器(PB15_SEL) 204
8.5.5 端口 PC 功能选择寄存器 205
8.5.5.1 端口 PC00 功能配置寄存器(PC00_SEL) 205
8.5.5.2 端口 PC01 功能配置寄存器(PC01_SEL) 206

8.5.5.3 端口 PC02 功能配置寄存器(PC02_SEL) 207
8.5.5.4 端口 PC03 功能配置寄存器(PC03_SEL) 208
8.5.5.5 端口 PC04 功能配置寄存器(PC04_SEL) 209
8.5.5.6 端口 PC05 功能配置寄存器(PC05_SEL) 210
8.5.5.7 端口 PC06 功能配置寄存器(PC06_SEL) 211
8.5.5.8 端口 PC07 功能配置寄存器(PC07_SEL) 212
8.5.5.9 端口 PC08 功能配置寄存器(PC08_SEL) 213
8.5.5.10 端口 PC09 功能配置寄存器(PC09_SEL) 214
8.5.5.11 端口 PC10 功能配置寄存器(PC10_SEL) 215
8.5.5.12 端口 PC11 功能配置寄存器(PC11_SEL) 216
8.5.5.13 端口 PC12 功能配置寄存器(PC12_SEL) 217
8.5.5.14 端口 PC13 功能配置寄存器(PC13_SEL) 218
8.5.5.15 端口 PC14 功能配置寄存器(PC14_SEL) 219
8.5.5.16 端口 PC15 功能配置寄存器(PC15_SEL) 220
8.5.6 端口 PD 功能选择寄存器 221
8.5.6.1 端口 PD00 功能配置寄存器(PD00_SEL) 221
8.5.6.2 端口 PD01 功能配置寄存器(PD01_SEL) 222
8.5.6.3 端口 PD02 功能配置寄存器(PD02_SEL) 223
8.5.6.4 端口 PD03 功能配置寄存器(PD03_SEL) 224
8.5.6.5 端口 PD04 功能配置寄存器(PD04_SEL) 225
8.5.6.6 端口 PD05 功能配置寄存器(PD05_SEL) 226
8.5.6.7 端口 PD06 功能配置寄存器(PD06_SEL) 227
8.5.6.8 端口 PD07 功能配置寄存器(PD07_SEL) 228
8.5.6.9 端口 PD08 功能配置寄存器(PD08_SEL) 229
8.5.6.10 端口 PD09 功能配置寄存器(PD09_SEL) 230
8.5.6.11 端口 PD10 功能配置寄存器(PD10_SEL) 231
8.5.6.12 端口 PD11 功能配置寄存器(PD11_SEL) 232
8.5.6.13 端口 PD12 功能配置寄存器(PD12_SEL) 233
8.5.6.14 端口 PD13 功能配置寄存器(PD13_SEL) 234
8.5.6.15 端口 PD14 功能配置寄存器(PD14_SEL) 235
8.5.6.16 端口 PD15 功能配置寄存器(PD15_SEL) 236
8.5.7 端口 PE 功能选择寄存器 237
8.5.7.1 端口 PE00 功能配置寄存器(PE00_SEL) 237
8.5.7.2 端口 PE01 功能配置寄存器(PE01_SEL) 238
8.5.7.3 端口 PE02 功能配置寄存器(PE02_SEL) 239
8.5.7.4 端口 PE03 功能配置寄存器(PE03_SEL) 240
8.5.7.5 端口 PE04 功能配置寄存器(PE04_SEL) 241
8.5.7.6 端口 PE05 功能配置寄存器(PE05_SEL) 242
8.5.7.7 端口 PE06 功能配置寄存器(PE06_SEL) 243
8.5.7.8 端口 PE07 功能配置寄存器(PE07_SEL) 244
8.5.7.9 端口 PE08 功能配置寄存器(PE08_SEL) 245
8.5.7.10 端口 PE09 功能配置寄存器(PE09_SEL) 246

8.5.7.11 端口 PE10 功能配置寄存器(PE10_SEL) 247
8.5.7.12 端口 PE11 功能配置寄存器(PE11_SEL) 248
8.5.7.13 端口 PE12 功能配置寄存器(PE12_SEL) 249
8.5.7.14 端口 PE13 功能配置寄存器(PE13_SEL) 250
8.5.7.15 端口 PE14 功能配置寄存器(PE14_SEL) 251
8.5.7.16 端口 PE15 功能配置寄存器(PE15_SEL) 252
8.5.8 端口 PF 功能选择寄存器 253
8.5.8.1 端口 PF00 功能配置寄存器(PF00_SEL) 253
8.5.8.2 端口 PF01 功能配置寄存器(PF01_SEL) 254
8.5.8.3 端口 PF02 功能配置寄存器(PF02_SEL) 255
8.5.8.4 端口 PF03 功能配置寄存器(PF03_SEL) 256
8.5.8.5 端口 PF04 功能配置寄存器(PF04_SEL) 257
8.5.8.6 端口 PF05 功能配置寄存器(PF05_SEL) 258
8.5.8.7 端口 PF06 功能配置寄存器(PF06_SEL) 259
8.5.8.8 端口 PF07 功能配置寄存器(PF07_SEL) 260
8.5.8.9 端口 PF09 功能配置寄存器(PF09_SEL) 261
8.5.8.10 端口 PF10 功能配置寄存器(PF10_SEL) 262
8.5.8.11 端口 PF11 功能配置寄存器(PF11_SEL) 263
9 I2C 总线（I2C） 264
9.1 简介 264
9.2 主要特性 264
9.3 协议描述 264
9.3.1 I2C 总线上数据传输 265
9.3.2 I2C 总线上的应答 266
9.3.3 I2C 总线上的仲裁 267
9.4 功能描述 269
9.4.1 串行时钟发生器 270
9.4.2 输入滤波器 270
9.4.3 地址比较器 270
9.4.4 应答标志位 271
9.4.5 中断产生器 271
9.4.6 工作模式 271
9.4.7 状态码表述 278
9.5 编程示例 280
9.5.1 主机发送示例 280
9.5.2 主机接收示例 280
9.5.3 从机接收示例 281
9.5.4 从机发送示例 282
9.6 寄存器描述 283
9.6.1 I2C 波特率计数器使能寄存器(I2Cx_TMRUN) 283
9.6.2 I2C 波特率计数器配置寄存器(I2Cx_TM) 284
9.6.3 I2C 配置寄存器(I2Cx_CR) 285

9.6.4 I2C 数据寄存器(I2Cx_DATA) 287
9.6.5 I2C 地址寄存器(I2Cx_ADDR) 288
9.6.6 I2C 状态寄存器(I2Cx_STAT) 289
10 串行外设接口（SPI） 290
10.1 SPI 简介 290
10.2 SPI 主要特性 290
10.3 SPI 功能描述 291
10.3.1 SPI 主机模式 291
10.3.2 SPI 从机模式 292
10.3.3 SPI 数据帧格式 293
10.3.4 SPI 状态标志及中断 294
10.3.5 SPI 多机系统配置说明 295
10.3.6 SPI 管脚配置说明 297
10.4 SPI 编程示例 298
10.4.1 SPI 主机发送示例 298
10.4.2 SPI 主机接收示例 298
10.4.3 SPI 从机发送示例 299
10.4.4 SPI 从机接收示例 300
10.6 SPI 寄存器描述 301
10.6.1 SPI 配置寄存器(SPIx_CR) 302
10.6.2 SPI 片选配置寄存器(SPIx_SSN) 304
10.6.3 SPI 状态寄存器(SPIx_STAT) 305
10.6.4 SPI 数据寄存器(SPIx_DATA) 307
10.6.5 SPI 配置寄存器 2(SPIx_CR2) 308
10.6.6 SPI 中断清除寄存器 2(SPIx_ICLR) 309
11 时钟校准模块（CLKTRIM） 310
11.1 CLKTRIM 简介 310
11.2 CLKTRIM 主要特性 310
11.3 CLKTRIM 功能描述 311
11.3.1 CLKTRIM 校准模式 311
11.3.1.1 时钟校准原理 311
11.3.1.2 时钟校准模块硬件结构 311
11.3.1.3 时钟校准软件流程 312
11.3.2 CLKTRIM 监测模式 314
11.3.2.1 时钟监测原理 314
11.3.2.2 时钟监测硬件结构 314
11.3.2.3 时钟监测软件流程 314
11.4 CLKTRIM 寄存器描述 317
11.4.1 配置寄存器(CLKTRIM_CR) 318
11.4.2 参考计数器初值配置寄存器(CLKTRIM_REFCON) 320
11.4.3 参考计数器值寄存器(CLKTRIM_REFCNT) 320
11.4.4 校准计数器值寄存器(CLKTRIM_CALCNT) 321

11.4.5 中断标志位寄存器(CLKTRIM_IFR) 322
11.4.6 中断标志位清除寄存器(CLKTRIM_ICLR) 323
11.4.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON) 324
12 FLASH 控制器 (FLASH) 325
12.1 概述 325
12.2 FLASH 容量划分 325
12.3 功能描述 326
12.3.1 页擦除（Sector Erase） 327
12.3.2 全片擦除（Chip Erase） 328
12.3.3 写操作（Program） 330
12.3.4 读操作（Read） 332
12.4 擦写时间 333
12.5 读等待周期 335
12.6 擦写保护 335
12.6.1 擦写保护位 335
12.6.2 PC 地址擦写保护 335
12.7 寄存器写保护 336
12.8 寄存器 337
12.8.1 TNVS 参数寄存器（FLASH_TNVS） 337
12.8.2 TPGS 参数寄存器（FLASH_TPGS） 338
12.8.3 TPROG 参数寄存器（FLASH_TPROG） 338
12.8.4 TSERASE 寄存器（FLASH_TSERASE） 339
12.8.5 TMERASE 参数寄存器（FLASH_TMERASE） 339
12.8.6 TPRCV 参数寄存器（FLASH_TPRCV） 340
12.8.7 TSRCV 参数寄存器（FLASH_TSRCV） 340
12.8.8 TMRCV 参数寄存器（FLASH_TMRCV） 341
12.8.9 CR 寄存器（FLASH_CR） 342
12.8.10 IFR 寄存器（FLASH_IFR） 343
12.8.11 ICLR 寄存器（FLASH_ICLR） 343
12.8.12 BYPASS 寄存器（FLASH_BYPASS） 344
12.8.13 SLOCK0 寄存器（FLASH_SLOCK0） 344
12.8.14 SLOCK1 寄存器（FLASH_SLOCK1） 345
12.8.15 SLOCK2 寄存器（FLASH_SLOCK2） 346
12.8.16 SLOCK3 寄存器（FLASH_SLOCK3） 347
13 RAM 控制器 (RAM) 348
13.1 概述 348
13.2 功能描述 348
13.2.1 RAM 地址范围 348
13.2.2 读写位宽 348
13.2.3 奇偶校验 348
13.3 寄存器 349
13.3.1 控制寄存器（RAM_CR） 349

13.3.2 奇偶校验出错地址寄存器（RAM_ERRADDR） 350
13.3.3 出错中断标志寄存器（RAM_IFR） 350
13.3.4 出错中断标志清除寄存器（RAM_ICLR） 351
14 DMA 控制器（DMAC） 352
14.1 概述 352
14.2 功能框图 354
14.3 基本功能 357
14.3.1 软件-块（Block）传输 357
14.3.2 软件-突发（Burst）传输 360
14.3.3 硬件-块（Block）传输 362
14.3.4 硬件-突发（Burst）传输 363
14.3.5 通道优先级控制 364
14.4 寄存器 365
14.4.1 DMAC_CONF 366
14.4.2 DMAC_CONFA0、DMAC_CONFA1 367
14.4.3 DMAC_CONFB0、DMAC_CONFB1 370
14.4.4 DMAC_SRCADR0、DMAC_SRCADR1 372
14.4.5 DMAC_DSTADR0、DMAC_DSTADR1 373
15 通用定时器（TIM0/1/2/3） 374
15.1 通用定时器简介 374
15.1.1 基本特性(TIM0/1/2) 374
15.1.2 基本特性(TIM3) 376
15.2 Timer 功能描述 377
15.2.1 定时器时钟 377
15.2.2 定时计数器 377
15.2.3 定时器预除频 377
15.2.4 模式 0 计数定时器功能 378
15.2.4.1 功能框图 379
15.2.4.2 计数波形 380
15.2.4.3 计数功能 381
15.2.4.4 定时功能 381
15.2.4.5 时序图 381
15.2.4.6 Buzzer 功能 382
15.2.4.7 设置示例 383
15.2.5 模式 1 脉宽测量 PWC 384
15.2.5.1 PWC 功能框图 384
15.2.5.2 PWC 波形测量时序图 385
15.2.5.3 PWC 单次触发模式 387
15.2.5.4 设置示例 387
15.2.6 模式 2/3 比较捕获模式 389
15.2.6.1 计数器 389
15.2.6.2 计数器波形 390

15.2.6.3 重复计数 393
15.2.6.4 数据缓存 394
15.2.6.5 比较输出 OCREF 398
15.2.6.6 独立 PWM 输出 401
15.2.6.7 互补 PWM 输出 402
15.2.6.8 有死区的 PWM 输出 403
15.2.6.9 单脉冲输出 404
15.2.6.10 比较中断 405
15.2.6.11 捕获输入 406
15.2.6.12 设置示例 409
15.2.7 模式 2/3 从模式 412
15.2.7.1 门控计数 412
15.2.7.2 触发功能 413
15.2.7.3 复位计数 413
15.2.8 正交编码计数功能 414
15.2.9 Timer 触发 ADC 416
15.2.10 刹车控制 417
15.2.11 Timer 互联 417
15.2.12 GATE 输入互联 418
15.2.13 ETR 输入互联 418
15.2.14 CHx 捕获输入互联 419
15.2.15 DMA 419
15.2.15.1 设置示例 421
15.3 Timer 寄存器描述 422
15.4 模式 0 定时器寄存器描述 423
15.4.1 16 位模式重载寄存器（TIMx_ARR） 423
15.4.2 16 位模式计数寄存器（TIMx_CNT） 423
15.4.3 32 位模式计数寄存器（TIMx_CNT32） 424
15.4.4 控制寄存器（TIMx_M0CR） 425
15.4.5 中断标志寄存器（TIMx_IFR） 427
15.4.6 中断标志清除寄存器（TIMx_ICLR） 427
15.4.7 死区时间寄存器（TIMx_DTR） 428
15.5 脉冲宽度测量 PWC 寄存器描述 429
15.5.1 16 位模式计数寄存器（TIMx_CNT） 429
15.5.2 控制寄存器（TIMx_M1CR） 430
15.5.3 中断标志寄存器（TIMx_IFR） 432
15.5.4 中断标志清除寄存器（TIMx_ICLR） 432
15.5.5 主从模式控制寄存器（TIMx_MSCR） 433
15.5.6 输出控制滤波（TIMx_FLTR） 434
15.5.7 控制寄存器（TIMx_CR0） 435
15.5.8 比较捕获寄存 (TIMx_CCR0A) 435
15.6 模式 2,3 寄存器描述 436

15.6.1 16 位模式重载寄存器（TIMx_ARR）..... 436
15.6.2 16 位模式计数寄存器（TIMx_CNT）..... 436
15.6.3 控制寄存器（TIMx_M23CR）..... 437
15.6.4 中断标志寄存器（TIMx_IFR）..... 440
15.6.5 中断标志清除寄存器（TIMx_ICLR）..... 442
15.6.6 主从模式控制寄存器（TIMx_MSCR）..... 443
15.6.7 输出控制/输入滤波（TIMx_FLTR）..... 445
15.6.8 ADC 触发控制寄存器（TIMx_ADTR）..... 448
15.6.9 通道 0 控制寄存器（TIMx_CRCH0）..... 449
15.6.10 通道 1/2 控制寄存器（TIM3_CRCH1/2）（仅 TIM3 存在）..... 451
15.6.11 死区时间寄存器（TIMx_DTR）..... 453
15.6.12 重复周期设置值（TIMx_RCR）..... 455
15.6.13 通道 0 比较捕获寄存器 (TIMx_CCR0A/B)..... 455
15.6.14 通道 1/2 比较捕获寄存器 (TIM3_CCR1/2 A/B) （仅 TIM3 存在）..... 456
16 可编程计数阵列（PCA）..... 456
16.1 PCA 简介..... 456
16.2 PCA 功能描述..... 458
16.2.1 PCA 定时/计数器..... 459
16.2.1.1 16 位自由计数模式..... 460
16.2.1.2 16 位重载计数模式..... 460
16.2.2 PCA 捕获功能..... 462
16.2.3 PCA 比较功能..... 464
16.2.3.1 比较翻转输出模式..... 464
16.2.3.2 PCA 16 位 PWM 功能..... 466
16.2.3.3 PCA 模块 4 的 WDT 功能..... 467
16.2.3.4 PCA 8 位 PWM 功能..... 468
16.3 PCA 模块与其他模块互连及控制..... 471
16.4 PCA 寄存器描述..... 472
16.4.1 控制寄存器（PCA_CCON）..... 473
16.4.2 模式寄存器（PCA_CMOD）..... 474
16.4.3 计数寄存器（PCA_CNT）..... 475
16.4.4 中断清除寄存器（PCA_ICLR）..... 475
16.4.5 比较捕获模式寄存器（PCA_CCAPM0~4）..... 476
16.4.6 比较捕获数据寄存器高 8 位（PCA_CCAP0~4H）..... 477
16.4.7 比较捕获数据寄存器低 8 位（PCA_CCAP0~4L）..... 477
16.4.8 比较捕获 16 位寄存器（PCA_CCAP0~4）..... 478
16.4.9 比较高速输出标志寄存器（PCA_CCAPO）..... 478
16.4.10 周期寄存器（PCA_CARR）..... 479
16.4.11 增强 PWM 控制（PCA_EPWM）..... 479
17 高级定时器（TIM4/5/6）..... 480
17.1 Advanced Timer 简介..... 480
17.2 Advanced Timer 功能描述..... 482

17.2.1 基本动作 482
17.2.1.1 基本波形模式 482
17.2.1.2 比较输出 483
17.2.1.3 捕获输入 484
17.2.2 时钟源选择 484
17.2.3 计数方向 485
17.2.3.1 锯齿波计数方向 485
17.2.3.2 三角波计数方向 485
17.2.4 数字滤波 485
17.2.5 软件同步 486
17.2.5.1 软件同步启动 486
17.2.5.2 软件同步停止 487
17.2.5.3 软件同步清零 487
17.2.6 硬件同步 487
17.2.6.1 硬件同步启动 487
17.2.6.2 硬件同步停止 487
17.2.6.3 硬件同步清零 488
17.2.6.4 硬件同步捕获输入 488
17.2.6.5 硬件同步计数 489
17.2.7 缓存功能 490
17.2.7.1 缓存传送时间点 491
17.2.7.2 通用周期基准值缓存传送时间点 491
17.2.7.3 通用比较基准值缓存传送时间点 491
17.2.7.4 捕获输入值缓存传送时间点 491
17.2.7.5 清零动作时缓存传送 491
17.2.8 通用 PWM 输出 492
17.2.8.1 PWM 展频输出 492
17.2.8.2 独立 PWM 输出 492
17.2.8.3 互补 PWM 输出 493
17.2.8.4 多相 PWM 输出 495
17.2.9 正交编码计数 497
17.2.9.1 位置计数模式 497
17.2.9.2 公转模式 500
17.2.10 周期间隔响应 503
17.2.11 保护机制 503
17.2.12 中断说明 504
17.2.12.1 计数比较匹配中断 504
17.2.12.2 计数周期匹配中断 504
17.2.12.3 死区时间错误中断 504
17.2.13 DMA 505
17.2.14 刹车保护 505
17.2.14.1 端口刹车与软件刹车 505

17.2.14.2 低功耗模式自动刹车 506
17.2.14.3 输出电平同高同低刹车 506
17.2.14.4 VC 刹车 507
17.2.15 内部互连 508
17.2.15.1 中断触发输出 508
17.2.15.2 AOS 触发 509
17.2.15.3 端口触发 TRIGA-TRIGD 510
17.2.15.4 比较输出 VC 与 Advanced Timer 互连 510
17.3 寄存器描述 511
17.3.1 通用计数基准值寄存器 (TIMx_CNTER) 513
17.3.2 通用周期基准值寄存器（TIMx_PERAR） 513
17.3.3 通用周期缓存寄存器（TIMx_PERBR） 514
17.3.4 通用比较基准值寄存器（TIMx_GCMAR-GCMDR） 514
17.3.5 专用比较基准值寄存器（TIMx_SCMAR-SCMBR） 515
17.3.6 死区时间基准值寄存器（TIMx_DTUAR- DTDAR） 515
17.3.7 通用控制寄存器（TIMx_GCONR） 516
17.3.8 中断控制寄存器（TIMx_ICONR） 518
17.3.9 端口控制寄存器（TIMx_PCONR） 520
17.3.10 缓存控制寄存器（TIMx_BCONR） 523
17.3.11 死区控制寄存器（TIMx_DCONR） 524
17.3.12 滤波控制寄存器（TIMx_FCONR） 525
17.3.13 有效周期寄存器（TIMx_VPERR） 527
17.3.14 状态标志寄存器（TIMx_STFLR） 528
17.3.15 硬件启动事件选择寄存器（TIMx_HSTAR） 530
17.3.16 硬件停止事件选择寄存器（TIMx_HSTPR） 532
17.3.17 硬件清零事件选择寄存器（TIMx_HCELR） 534
17.3.18 硬件捕获 A 事件选择寄存器（TIMx_HCPAR） 536
17.3.19 硬件捕获 B 事件选择寄存器（TIMx_HCPBR） 538
17.3.20 硬件递加事件选择寄存器（TIMx_HCUPR） 540
17.3.21 硬件递减事件选择寄存器（TIMx_HCDOR） 542
17.3.22 软件同步启动寄存器（TIMx_SSTAR） 544
17.3.23 软件同步停止寄存器（TIMx_SSTPR） 545
17.3.24 软件同步清零寄存器（TIMx_SCLRR） 546
17.3.25 中断标志寄存器（TIMx_IFR） 547
17.3.26 中断标志清除寄存器（TIMx_ICLR） 549
17.3.27 展频及中断触发选择（TIMx_CR） 550
17.3.28 AOS 选择控制寄存器（TIMx_AOSSR） 552
17.3.29 AOS 选择控制寄存器标志清除（TIMx_AOSCL） 553
17.3.30 端口刹车控制寄存器（TIMx_PTBKS） 554
17.3.31 端口触发控制寄存器（TIMx_TTRIG） 555
17.3.32 AOS 触发控制寄存器（TIMx_ITRIG） 556
17.3.33 端口刹车极性控制寄存器（TIMx_PTBKP） 557

18 看门狗定时器（WDT）.....558
18.1 WDT 简介.....558
18.2 WDT 功能描述.....559
18.2.1 WDT 溢出后产生中断.....559
18.2.2 WDT 溢出后产生复位.....559
18.3 WDT 寄存器描述.....561
18.3.1 WDT 清除控制寄存器（WDT_RST）.....561
18.3.2 WDT_CON 寄存器.....562
19 通用同步异步收发器（UART）.....563
19.1 简介.....563
19.2 主要特性.....564
19.3 功能描述.....565
19.3.1 工作模式.....565
19.3.1.1 Mode0~Mode3 功能对比.....565
19.3.1.2 Mode0 数据收发说明.....566
19.3.1.3 Mode1 数据收发说明.....567
19.3.1.4 Mode2 数据收发说明.....568
19.3.1.5 Mode3 数据收发说明.....569
19.3.1.6 单线半双工数据收发说明.....570
19.3.2 波特率生成.....570
19.3.2.1 Mode1/Mode3 波特率设置示例.....571
19.3.3 帧错误检测.....574
19.3.4 多机通讯.....574
19.3.4.1 给定地址.....574
19.3.4.2 广播地址.....575
19.3.4.3 举例.....575
19.3.5 DMAC 硬件握手.....575
19.3.6 硬件流控.....576
19.3.7 收发端缓存.....578
19.4 寄存器.....580
19.4.1 数据寄存器（UARTx_SBUF）.....580
19.4.2 控制寄存器（UARTx_SCON）.....581
19.4.3 地址寄存器（UARTx_SADDR）.....583
19.4.4 地址掩码寄存器（UARTx_SADEN）.....583
19.4.5 标志位寄存器（UARTx_ISR）.....584
19.4.6 标志位清除寄存器（UARTx_ICR）.....585
19.4.7 波特率寄存器（UARTx_SCNT）.....585
20 循环冗余校验（CRC）.....586
20.1 概述.....586
20.2 功能框图.....586
20.3 功能描述.....586
20.4 寄存器.....587

20.4.1 控制寄存器（CRC_CR） 587
20.4.2 结果寄存器（CRC_RESULT） 588
20.4.3 数据寄存器（CRC_DATA） 589
20.5 软件基本操作 590
20.5.1 CRC16 编码模式 590
20.5.2 CRC16 检验模式 590
20.5.3 CRC32 编码模式 590
20.5.4 CRC32 检验模式 591
21 真随机数发生器（TRNG） 592
21.1 概述 592
21.2 功能框图 592
21.3 功能描述 592
21.4 寄存器 593
21.4.1 控制寄存器（TRNG_CR） 593
21.4.2 模式寄存器（TRNG_MODE） 594
21.4.3 数据寄存器 0（TRNG_DATA0） 595
21.4.4 数据寄存器 1（TRNG_DATA1） 595
21.5 软件基本操作 596
21.5.1 生成 64bits 真随机数的操作流程（上电第一次） 596
21.5.2 生成 64bits 真随机数的操作流程（非上电第一次生成） 597
22 高级加密标准模块（AES） 598
22.1 功能定义 598
22.1.1 AES 算法简述 598
22.1.2 AES 模块功能描述 600
22.2 模块寄存器说明 601
22.2.1 控制寄存器（AES_CR） 602
22.2.2 数据寄存器（AES_Data） 603
22.2.3 密钥寄存器（AES_Key） 604
22.3 异常机制 606
22.4 本模块操作说明 607
22.4.1 IP 操作的共同点 607
22.4.2 加密操作流程 607
22.4.3 解密操作流程 607
22.4.4 数据示例 608
22.5 运行时间说明 611
23 液晶控制器（LCD） 612
23.1 LCD 简介 612
23.2 LCD 主要特性 612
23.3 LCD 框图 613
23.4 LCD 驱动波形 614
23.4.1 静态驱动波形 614
23.4.2 1/2Duty 1/2Bias 驱动波形 615

23.4.3 1/2Duty 1/3Bias 驱动波形 616
23.4.4 1/3Duty 1/2Bias 驱动波形 617
23.4.5 1/3Buty 1/3Bias 驱动波形 618
23.4.6 1/4Duty 1/2Bias 驱动波形 619
23.4.7 1/4Duty 1/3Bias 驱动波形 620
23.4.8 1/6Duty 1/3Bias 驱动波形 621
23.4.9 1/8Duty 1/3Bias 驱动波形 622
23.5 LCD Bias 产生电路 623
23.5.1 内部电阻模式 623
23.5.2 外部电容模式 624
23.5.3 外部电阻模式 625
23.6 DMA 625
23.7 中断 626
23.8 LCD 显示模式 626
23.8.1 LCD 显示模式 1 (MODE = 1) 627
23.8.2 LCD 显示模式 0 (MODE = 0) 629
23.9 LCD 寄存器 630
23.9.1 配置寄存器 0（LCD_CR0） 631
23.9.2 配置寄存器 1（LCD_CR1） 633
23.9.3 中断清除寄存器（LCD_INTCLR） 634
23.9.4 输出配置寄存器 0（LCD_POEN0） 634
23.9.5 输出配置寄存器 1（LCD_POEN1） 635
23.9.6 LCD_RAM0~7 637
23.9.7 LCD_RAM8~F 637
24 模数转换器（ADC） 638
24.1 模块简介 638
24.2 ADC 框图 639
24.3 转换时序及转换速度 640
24.4 单次转换模式 641
24.5 扫描转换模式 643
24.5.1 顺序扫描转换模式 643
24.5.2 插队扫描转换模式 645
24.5.3 扫描转换触发 DMA 读取 649
24.6 连续转换累加模式 650
24.7 ADC 转换外部触发源 653
24.8 ADC 转换结果比较 654
24.9 ADC 中断 655
24.10 使用温度传感器测量环境温度 656
24.11 ADC 模块寄存器 658
24.11.1 ADC 基本配置寄存器 0（ADC_CR0） 660
24.11.2 ADC 基本配置寄存器 1（ADC_CR1） 662
24.11.3 ADC 顺序扫描转换通道配置寄存器 0（ADC_SQR0） 664

24.11.4 ADC 顺序扫描转换通道配置寄存器 1（ADC_SQR1） 665
24.11.5 ADC 顺序扫描转换通道配置寄存器 2（ADC_SQR2） 666
24.11.6 ADC 插队扫描转换通道配置寄存器（ADC_JQR） 667
24.11.7 ADC 顺序扫描转换通道 x 转换结果（ADC_SqrResult0 - 15） 667
24.11.8 ADC 插队扫描转换通道 x 转换结果（ADC_JqrResult0 - 3） 668
24.11.9 ADC 转换结果（ADC_Result） 668
24.11.10 ADC 转换结果累加值（ADC_ResultAcc） 669
24.11.11 ADC 比较上阈值（ADC_HT） 669
24.11.12 ADC 比较下阈值（ADC_LT） 670
24.11.13 ADC 中断标志寄存器（ADC_IFR） 671
24.11.14 ADC 中断清除寄存器（ADC_ICR） 672
24.11.15 ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger0） 673
24.11.16 ADC 插队扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger1） 675
24.11.17 ADC 单次转换启动控制寄存器（ADC_SglStart） 677
24.11.18 ADC 顺序扫描转换启动控制寄存器（ADC_SqrStart） 677
24.11.19 ADC 插队扫描转换启动控制寄存器（ADC_JqrStart） 678
24.11.20 ADC 一直转换启动停止控制寄存器（ADC_allStart） 679
25 数模转换器（DAC） 680
25.1 DAC 特性 680
25.2 功能说明 681
25.2.1 框图 681
25.2.2 DAC 输出缓冲器使能 682
25.2.3 DAC 通道使能 682
25.2.4 DAC 输出电压 682
25.2.5 DAC 触发选择 683
25.2.6 功能说明 684
25.2.6.1 DAC 数据格式 684
25.2.6.2 DAC 转换 684
25.2.7 噪声生成 686
25.2.8 三角波生成 687
25.2.9 DMA 请求 688
25.3 DAC 寄存器 689
25.3.1 DAC 控制寄存器（DAC_CR） 690
25.3.2 DAC 软件触发寄存器（DAC_SWTR） 692
25.3.3 DAC 12 位右对齐数据保持寄存器（DAC_DHR12R0） 692
25.3.4 DAC 12 位左对齐数据保持寄存器（DAC_DHR12L0） 693
25.3.5 DAC 8 位右对齐数据保持寄存器（DAC_DHR8R0） 693
25.3.6 DAC 数据输出寄存器（DAC_DOR0） 694
25.3.7 DAC 状态寄存器（DAC_SR） 694
25.3.8 DAC 控制寄存器（DAC_ETRS） 695
26 模拟比较器（VC） 696
26.1 模拟电压比较器 VC 简介 696

26.2 电压比较器框架图.....697
26.3 建立/响应时间.....698
26.4 滤波时间.....698
26.5 迟滞功能.....699
26.6 VC 寄存器.....700
26.6.1 VC 配置寄存器（VC_CR）.....700
26.6.2 VC0 配置寄存器（VC0_CR）.....702
26.6.3 VC1 配置寄存器（VC1_CR）.....704
26.6.4 VC0 输出配置寄存器（VC0_OUT_CFG）.....706
26.6.5 VC1 输出配置寄存器（VC1_ OUT_CFG）.....708
26.6.6 VC 中断寄存器（VC_IFR）.....710
26.6.7 VC2 配置寄存器（VC2_CR）.....711
26.6.8 VC2 输出配置寄存器（VC2_ OUT_CFG）.....713
27 低电压检测器（LVD）.....715
27.1 LVD 简介.....715
27.2 LVD 框图.....715
27.3 迟滞功能.....716
27.4 数字滤波.....716
27.5 配置示例.....717
27.5.1 LVD 配置为低电压复位.....717
27.5.2 LVD 配置为电压变化中断.....717
27.6 LVD 寄存器.....718
27.6.1 LVD 配置寄存器（LVD_CR）.....718
27.6.2 LVD 中断寄存器（LVD_IFR）.....720
28 运算放大器（OPA）.....721
28.1 OPA 特性.....721
28.2 OPA 功能描述.....722
28.3 OPA 校零.....723
28.3.1 软件控制.....723
28.3.2 软件触发控制.....723
28.3.3 ADC 触发控制.....723
28.4 OPA 寄存器.....724
28.4.1 OPA 配置寄存器（OPA_CR0）.....725
28.4.2 OPA 配置寄存器（OPA_CR1）.....726
29 模拟其它寄存器.....727
29.1 BGR 配置寄存器（BGR_CR）.....727
30 SWD 调试接口.....728
30.1 SWD 调试附加功能.....728
30.2 ARM® 参考文档.....729
30.3 调试端口引脚.....730
30.3.1 SWD 端口引脚.....730
30.3.2 SW-DP 引脚分配.....730

30.3.3 SWD 引脚上的内部上拉 730
30.4 SWD 端口 731
30.4.1 SWD 协议简介 731
30.4.2 SWD 协议序列 731
30.4.3 SW-DP 状态机（复位、空闲状态、ID 代码） 732
30.4.4 DP 和 AP 读/写访问 732
30.4.5 SW-DP 寄存器 733
30.4.6 SW-AP 寄存器 734
30.5 内核调试 735
30.6 BPU（断点单元） 735
30.6.1 BPU 功能 735
30.7 DWT（数据观察点） 736
30.7.1 DWT 功能 736
30.7.2 DWT 程序计数器采样寄存器 736
30.8 MCU 调试组件 (DBG) 737
30.8.1 对低功耗模式的调试支持 737
30.8.2 对定时器、看门狗的调试支持 737
30.9 调试模式模块工作状态控制（DEBUG_ACTIVE） 738
31 典型应用电路图 740
32 电气特性 741
32.1 测试条件 741
32.1.1 最小和最大数值 741
32.1.2 典型数值 741
32.2 绝对最大额定值 742
32.3 工作条件 744
32.3.1 通用工作条件 744
32.3.2 上电和掉电时的工作条件 744
32.3.3 内嵌复位和 LVD 模块特性 745
32.3.4 内置的参考电压 747
32.3.5 供电电流特性 747
32.3.6 从低功耗模式唤醒的时间 751
32.3.7 外部时钟源特性 752
32.3.7.1 外部输入高速时钟 752
32.3.7.2 外部输入低速时钟 752
32.3.7.3 高速外部时钟 XTH 753
32.3.7.4 低速外部时钟 XTL 755
32.3.8 内部时钟源特性 757
32.3.8.1 内部 RCH 振荡器 757
32.3.8.2 内部 RCL 振荡器 758
32.3.9 PLL 特性 758
32.3.10 存储器特性 759
32.3.11 EFT 特性 759

32.3.12 ESD 特性 760
32.3.13 I/O 端口特性 760
32.3.13.1 输出特性——端口 760
32.3.13.2 输入特性——端口 PA,PB,PC,PD,PE,PF,RESET 762
32.3.13.3 端口外部输入采样要求——Timer Gate/Timer Clock 762
32.3.13.4 端口漏电特性——PA,PB,PC,PD,PE,PF 763
32.3.14 RESETB 引脚特性 763
32.3.15 ADC 特性 763
32.3.16 VC 特性 766
32.3.17 OPA 特性 767
32.3.18 LCD 控制器 768
32.3.19 DAC 特性 768
33 唯一设备 ID 寄存器（80 位） 769
34 封装尺寸 770
35 附录 A SysTick 定时器 776
35.1 SysTick 定时器简介 776
35.2 设置 SysTick 776
35.3 SysTick 寄存器 777
35.3.1 SysTick 控制和状态寄存器（CTRL） 777
35.3.2 SysTick 重载寄存器（LOAD） 777
35.3.3 SysTick 当前值寄存器（VAL） 777
35.3.4 SysTick 校准值寄存器（CALIB） 778
36 附录 B 文档约定 779
36.1 寄存器相关缩写词列表 779
36.2 词汇表 779
版本记录 & 联系方式 780

表目录

表 2-1 模块信号说明 63
表 3-1 地址划分表 68
表 4-1 运行模式下可运行模块图 71
表 4-2 休眠模式下可运行模块图 74
表 4-3 深度休眠模式下可运行模块图 76
表 7-1 Cortex-M0+ 处理器中断一览 113
表 7-2 外部中断与 NVIC 中断输入对应关系 121
表 8-1 端口状态真值表 141
表 8-2 端口复用表 146
表 9-1 I2C 时钟信号波特率 270
表 10-1 SPI 管脚配置说明表 297
表 10-2 SPI 寄存器列表 301
表 10-3 主机模式波特率选择 303
表 11-1 寄存器列表 317
表 15-1 Timer 寄存器列表 422
表 16-1 PCA 比较捕获功能模块设置 470
表 16-2 PCA 寄存器列表 472
表 17-1 Advanced Timer 基本特性 480
表 17-2 Advanced Timer 端口列表 480
表 17-3 AOS 源选择 509
表 17-4 端口触发选择 510
表 17-5 Advanced Timer 寄存器列表 512
表 18-1 WDT 寄存器列表 561
表 19-1 Mode0/1/2/3 数据结构 565
表 19-2 B8 数据含义 566
表 19-3 PCLK=4MHz 波特率计算表 571
表 19-4 PCLK=8MHz 波特率计算表 571
表 19-5 PCLK=16MHz 波特率计算表 572
表 19-6 PCLK=24MHz 波特率计算表 572
表 19-7 PCLK=32MHz 波特率计算表 573
表 19-8 PCLK=48MHz 波特率计算表 573
表 22-1 寄存器列表 601
表 22-2 128 位操作寄存器示例 608
表 22-3 192 位操作寄存器示例 609
表 22-4 192 位操作寄存器示例 610
表 22-5 AES 加解密运行时间 611
表 23-1 LCD 寄存器 630
表 24-1 ADC 寄存器 659
表 25-1 DAC 寄存器 689
表 26-1 VC 寄存器 700

表 27-1 LVD 寄存器 718
表 28-1 OPA 寄存器 724
表 32-1 工作电流特性 750

图目录

图 3-1 系统架构示意图 64
图 3-2 地址区域划分示意图 66
图 4-1 控制模式框图 69
图 5-1 时钟控制模块框图 79
图 5-2 晶振时钟启动示意图 81
图 5-3 时钟切换示意图 87
图 5-4 时钟校准原理图 88
图 6-1 复位来源示意图 106
图 7-1 只使用了高两位的优先级寄存器 114
图 7-2 中断向量表 115
图 7-3 中断激活和挂起状态 116
图 7-4 中断挂起状态被清除然后被重新确认 117
图 7-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行 117
图 7-6 中断处理中产生的中断挂起也可以被确认 118
图 7-7 中断结构图 122
图 8-1 端口电路图 139
图 8-2 AHB/FASTIO 总线端口随系统时钟的变化 142
图 8-3 读取端口引脚数据同步图 143
图 9-1 I2C 传输协议 265
图 10-1 从机接收示意图 292
图 10-2 从机发送示意图 293
图 10-3 主机模式帧格式 293
图 10-4 从机 CPHA 为 0 时数据帧格式 294
图 10-5 从机 CHPA 为 1 时数据帧格式 294
图 10-6 SPI 多主机/多从机系统的示意图 296
图 11-1 时钟源选择示意图 311
图 11-2 时钟校准模块硬件示意图 312
图 11-3 时钟校准波形示意图 313
图 11-4 CLKTRIM 时钟选择 314
图 11-5 时钟监控波形示意图 316
图 15-1 TIM0/1/2 框图 375
图 15-2 TIM3 框图 376
图 15-3 自由计数框图 379
图 15-4 重载计数波形 379
图 15-5 16 位重载计数波形 380
图 15-6 32 位自由计数波形 380
图 15-7 自由计数时序图 381
图 15-8 重载计数时序图（预分频设置为 2） 381
图 15-9 PWC 测量框图 384
图 15-10 高电平脉冲宽度测量 385

图 15-11 下降沿到下降沿周期测量 385
图 15-12 上升沿到上升沿周期测量 385
图 15-13 上升沿到上升沿周期测量单次模式 387
图 15-14 计数器框图 389
图 15-15 无预分频的向上计数 390
图 15-16 带预分频的上计数 391
图 15-17 不带预分频的下计数 391
图 15-18 带预分频的下计数 392
图 15-19 带预分频的上下计数 392
图 15-20 边沿对齐计时器波形(DIR =1) 392
图 15-21 边沿对齐计时器波形(DIR =0) 393
图 15-22 中心对齐计数器波形 393
图 15-23 重复计数器产生更新时序 394
图 15-24 三角波模式下缓存使能 394
图 15-25 三角波模式下缓存无效 395
图 15-26 锯齿波模式下上计数缓存使能 395
图 15-27 锯齿波模式下上计数缓存无效 395
图 15-28 锯齿波模式下计数缓存使能 396
图 15-29 锯齿波模式下计数缓存无效 396
图 15-30 锯齿波模式下计数比较缓存使能 397
图 15-31 OCREF 输出框图 398
图 15-32 锯齿波计数单点比较 OCREF 输出波形（OCMx=111） 399
图 15-33 三角波计数单点比较 OCREF 输出波形（OCMx=111） 399
图 15-34 锯齿波计数双点比较 OCREF 输出（OCMx=111） 400
图 15-35 三角波计数双点比较 OCREF 输出（OCMx=111） 400
图 15-36 独立 PWM 输出框图 401
图 15-37 CCPx=0 时 PWM 输出波形 401
图 15-38 CCPx=1 时 PWM 输出波形 401
图 15-39 互补 PWM 输出框图 402
图 15-40 互补 PWM 输出波形图 402
图 15-41 互补 PWM 输出波形图 403
图 15-42 死区时间 403
图 15-43 三角波模式单脉冲计数 404
图 15-44 锯齿波上计数单脉冲模式 405
图 15-45 锯齿波下计数单脉冲模式 405
图 15-46 中断示意图 406
图 15-47 捕获功能框图 407
图 15-48 捕获时序图 407
图 15-49 CHA 端口选择 407
图 15-50 CHB 端口选择 408
图 15-51 从模式示意图 412
图 15-52 门控功能 413

图 15-53 触发和复位功能 413
图 15-54 编码计数 415
图 15-55 ADC 触发 416
图 16-1 PCA 整体框图 457
图 16-2 PCA 计数器框图 460
图 16-3 PCA 捕获功能框图 463
图 16-4 PCA 比较功能框图 465
图 16-5 PCA 16 位 PWM 功能框图 466
图 16-6 PCA WDT 功能框图 467
图 16-7 PCA PWM 功能框图 469
图 16-8 PCA PWM 输出波形 470
图 17-1 Advanced Timer 框图 481
图 17-2 锯齿波波形（递加计数） 482
图 17-3 三角波波形 482
图 17-4 比较输出动作 483
图 17-5 捕获输入动作 484
图 17-6 捕获输入端口的滤波功能 486
图 17-7 软件同步动作 486
图 17-8 硬件同步动作 489
图 17-9 单缓存方式比较输出时序 490
图 17-10 PWM 展频输出示意图 492
图 17-11 CHA 输出 PWM 波 492
图 17-12 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出 493
图 17-13 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区） 494
图 17-14 6 相 PWM 波 495
图 17-15 三角波 A 模式时带死区时间三相互补 PWM 波输出 496
图 17-16 位置模式时基本计数动作 497
图 17-17 位置模式时相位差计数动作设定(1 倍) 498
图 17-18 位置模式时相位差计数动作设定(2 倍) 498
图 17-19 位置模式时相位差计数动作设定(4 倍) 498
图 17-20 位置模式时方向计数动作 499
图 17-21 公转模式时 Z 相计数动作 500
图 17-22 公转模式时位置计数器输出计数动作 501
图 17-23 公转模式时 Z 相计数和位置计数器输出混合计数动作 501
图 17-24 公转计数模式-混合计数 Z 相屏蔽动作例 1 502
图 17-25 公转计数模式-混合计数 Z 相屏蔽动作例 2 502
图 17-26 周期间隔有效请求信号动作 503
图 17-27 端口刹车与软件刹车示意图 506
图 17-28 输出同高同低刹车示意图 506
图 17-29 VC 刹车控制示意图 507
图 17-30 Timer4/5/6 中断选择 508
图 18-1 WDT 整体框图 558

图 19-1 结构框图 563
图 19-2 Mode0 发送数据 566
图 19-3 Mode0 接收数据 567
图 19-4 Mode1 发送数据 567
图 19-5 Mode1 接收数据 568
图 19-6 Mode2 发送数据 568
图 19-7 Mode2 接收数据 569
图 19-8 Mode3 发送数据 569
图 19-9 Mode3 接收数据 569
图 19-10 UART 硬件流控 576
图 19-11 nRTS 硬件流控信号 577
图 19-12 nCTS 硬件流控信号 577
图 19-13 接收缓存 578
图 19-14 发送缓存 579
图 20-1 CRC 应用示意图 586
图 21-1 TRNG 数据流 592
图 22-1 AES 的加解密示意图 598
图 22-2 AES 的加密流程图 599
图 24-1 ADC 示意框图 639
图 24-2 ADC 转换时序图 640
图 24-3 ADC 顺序扫描转换过程示例 643
图 24-4 ADC 插队扫描转换过程示例 646
图 24-5 ADC 顺序扫描过程中进行插队扫描示例 648
图 24-6 ADC 连续转换累加过程示例 650
图 24-7 ADC 单次转换或顺序扫描转换外部触发源示意图 653
图 24-8 ADC 插队扫描转换外部触发源示意图 653
图 25-1 触发不使能时 DAC 转换时序图 685
图 25-2 LFSR 产生波形的 DAC 转换（使用软件触发） 686
图 25-3 生成 DAC 三角波 687
图 25-4 生成三角波波形的 DAC 转换（使能软件触发） 687
图 26-1 VC 框架图 697
图 26-2 VC 滤波响应时间 698
图 26-3 VC 迟滞功能 699
图 27-1 LVD 框图 715
图 30-1 调试支持框图 728
图 32-1 POR/Brown Out 示意图 745
图 32-2 输出端口 VOH/VOL 实测曲线 761

简介

HC32F17 / HC32F19 系列是一款宽电压工作范围的 MCU。集成 12 位 1Msps 高精度 SARADC，1 个 12 位 DAC 以及集成了比较器、运放、内置高性能 PWM 定时器、LCD 显示、多路 UART、SPI、I2C 等丰富的通讯外设，内建 AES、TRNG 等信息安全模块，具有高整合度、高抗干扰、高可靠性的特点。本产品内核采用 Cortex-M0+ 内核，配合成熟的 Keil & IAR 调试开发软件，支持 C 语言及汇编语言，汇编指令。

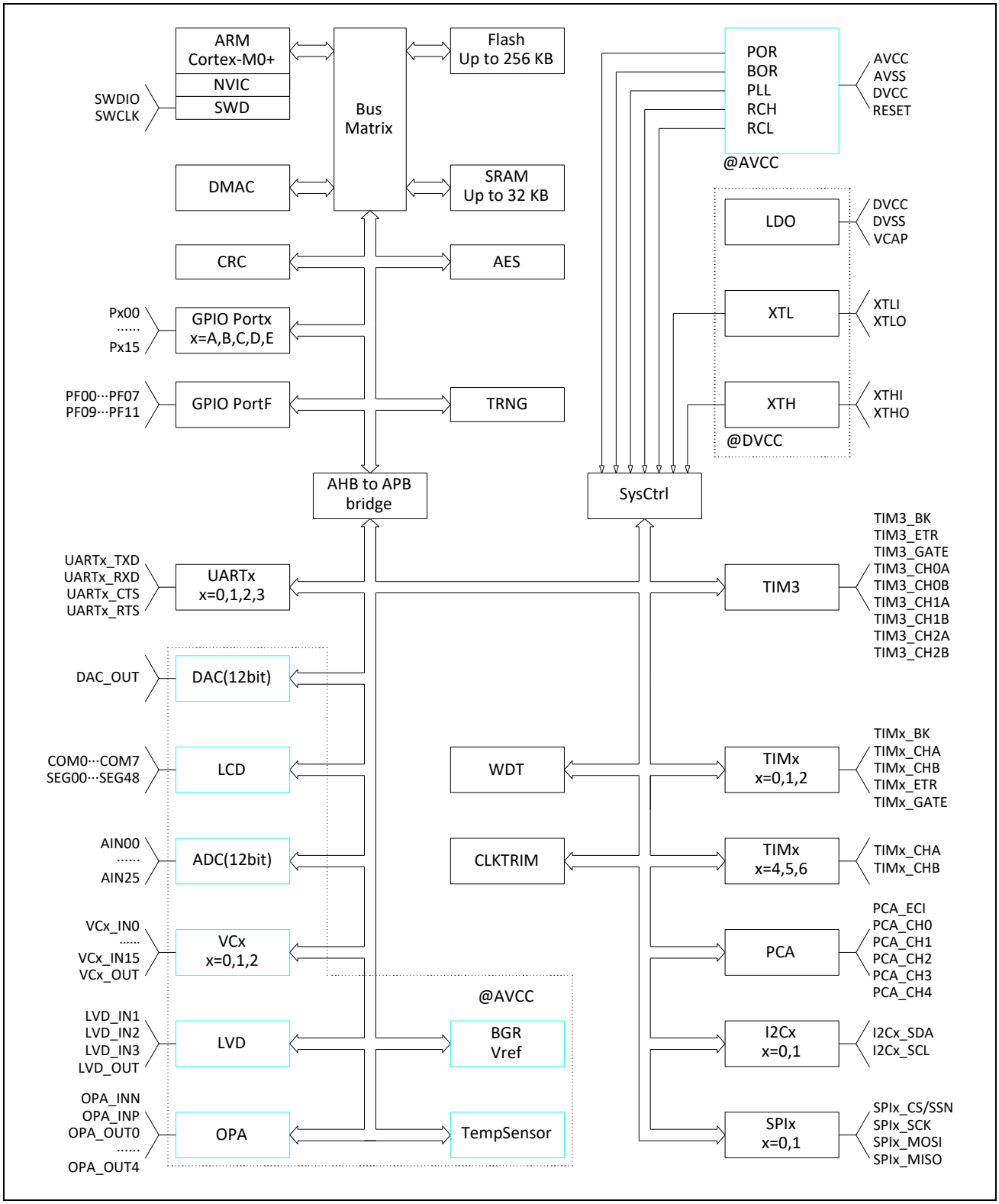
通用 MCU 典型应用

- 物联网应用
- 通信模块，温控器
- 智能交通，报警系统
- 智能家居，家用电器，健身设备

产品特性

- 48MHz Cortex-M0+ 32 位 CPU 平台
- HC32F17 /HC32F19 系列具有灵活的功耗管理系统，低功耗性能
  - 3μA @3V 深度休眠模式：所有时钟关闭，上电复位有效，IO 状态保持，IO 中断有效，所有寄存器、RAM 和 CPU 数据保存状态时的功耗
  - 10μA @32.768KHz 低速工作模式：CPU 运行，外设关闭，从 Flash 运行程序
  - 30μA/MHz@3V@24MHz 休眠模式：CPU 停止，外设关闭，主时钟运行
  - 130μA/MHz@3V@24MHz 工作模式：CPU 运行，外设关闭，从 Flash 运行程序
  - 4μs 唤醒时间，使模式切换更加灵活高效，系统反应更为敏捷
- 128K / 256K 字节 Flash 存储器，具有擦写保护功能，支持 ISP、ICP、IAP。
- 16K / 32K 字节 RAM 存储器，附带奇偶校验，增强系统的稳定性
- 通用 I/O 引脚（88IO/100PIN, 72IO/80PIN 56IO/64PIN, 40IO/48PIN, 26IO/32PIN）
- 时钟、晶振
  - 外部高速晶振 4 ~ 32MHz
  - 外部低速晶振 32.768KHz
  - 内部高速时钟 4/8/16/22.12/24MHz
  - 内部低速时钟 32.8/38.4KHz
  - PLL 时钟 8 ~ 48MHz
  - 硬件支持内外时钟校准和监控
- 定时器/计数器
  - 3 个 1 通道互补输出通用 16 位定时器
  - 1 个 3 通道互补输出通用 16 位定时器
  - 2 个低功耗 16 位定时器，支持级联
  - 3 个高性能 16 位定时器/计数器，支持 PWM 互补，死区保护功能
  - 1 个 20 位可编程看门狗电路，内建专用 10KHz 振荡器提供 WDT 计数
  - 1 个可编程 16 位定时器 PCA，支持 5 通道捕获比较，5 通道 PWM 输出
- 通讯接口
  - 4 路 UART 标准通讯接口
  - 2 路 SPI 标准通讯接口
  - 2 路 I2C 标准通讯接口
- 蜂鸣器频率发生器，支持互补输出
- 硬件 CRC16 / CRC32 模块
- AES128/192/256 硬件协处理器
- TRNG 真随机数发生器
- 2 通道 DMAC
- 4*52 / 6*50 / 8*48 LCD 驱动
- 全球唯一 10 字节 ID 号
- 12 位 1Msps 采样的高速高精度 SARADC，内置跟随器，可测量高输出阻抗的信号
- 1 路 12 位 500Ksps DAC
- 集成 1 个多功能运算放大器，可以作为 DAC 的输出 Buffer
- 集成 3 路电压比较器，具有 6 位 DAC 和可编程比较基准
- 集成低电压侦测器，可配置 16 阶比较电压，可监控端口电压以及电源电压
- SWD 调试解决方案，提供全功能调试器
- 工作条件：-40 ~ 85℃，1.8 ~ 5.5V
- 封装形式：LQFP100/80/64/48, QFN32

1 功能模块


ARM Cortex-M0+
NVIC
SWD
SWDIO
SWCLK
Bus Matrix
Flash Up to 256 KB
DMAC
SRAM Up to 32 KB
CRC
AES
Px00 ..... Px15
GPIO Portx x=A,B,C,D,E
PF00···PF07 PF09···PF11
GPIO PortF
TRNG
AHB to APB bridge
POR
BOR
PLL
RCH
RCL
AVCC
AVSS
DVCC
RESET
@AVCC
LDO
DVCC
DVSS
VCAP
XTL
XTLI
XTLO
XTH
XTHI
XTHO
@DVCC
SysCtrl
UARTx_TXD
UARTx_RXD
UARTx_CTS
UARTx_RTS
UARTx x=0,1,2,3
TIM3
TIM3_BK
TIM3_ETR
TIM3_GATE
TIM3_CH0A
TIM3_CH0B
TIM3_CH1A
TIM3_CH1B
TIM3_CH2A
TIM3_CH2B
DAC_OUT
DAC(12bit)
COM0···COM7
SEG00···SEG48
LCD
WDT
TIMx x=0,1,2
TIMx_BK
TIMx_CHA
TIMx_CHB
TIMx_ETR
TIMx_GATE
AIN00 ..... AIN25
ADC(12bit)
CLKTRIM
TIMx x=4,5,6
TIMx_CHA
TIMx_CHB
VCx_IN0 ..... VCx_IN15
VCx_OUT
VCx x=0,1,2
PCA
PCA_ECI
PCA_CH0
PCA_CH1
PCA_CH2
PCA_CH3
PCA_CH4
@AVCC
LVD_IN1
LVD_IN2
LVD_IN3
LVD_OUT
LVD
BGR Vref
I2Cx x=0,1
I2Cx_SDA
I2Cx_SCL
OPA_INN
OPA_INP
OPA_OUT0 ..... OPA_OUT4
OPA
TempSensor
SPIx x=0,1
SPIx_CS/SSN
SPIx_SCK
SPIx_MOSI
SPIx_MISO

1.1 32 位 CORTEX M0+ 内核

ARM® Cortex®-M0+ 处理器源于 Cortex-M0，包含了一颗 32 位 RISC 处理器，运算能力达到 0.95 Dhrystone MIPS/MHz。同时加入了多项全新设计，改进调试和追踪能力、减少每条指令循环（IPC）数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex-M0+ 处理器全面支持已整合 Keil & IAR 调试器。

Cortex-M0+ 包含了一个硬件调试电路，支持 2-pin 的 SWD 调试界面。

ARM Cortex-M0+ 特性：

指令集	Thumb / Thumb-2
流水线	2级流水线
性能效率	2.46 CoreMark / MHz
性能效率	0.95 DMIPS / MHz in Dhrystone
中断	32个快速中断
中断优先级	可配置4级中断优先级
增强指令	单周期32位乘法器
调试	Serial-wire 调试端口，支持4个硬中断（break point）以及2个观察点(watch point)

1.2 128K / 256K Byte FLASH

内建全集成 Flash 控制器，无需外部高压输入，由全内置电路产生高压来编程。支持 ISP、IAP、ICP 功能。

1.3 16K / 32K Byte RAM

根据客户选择不同的功耗模式，RAM 数据都会被保留。自带硬件奇偶校验位，万一数据被意外破坏，在数据被读取时，硬件电路会立刻产生中断，保证系统的可靠性。

1.4 时钟系统

一个频率为 4~24MHz 可配置的高精度内部时钟 RCH。在配置 24MHz 下，从低功耗模式到工作模式的唤醒时间为 4us，全电压全温度范围内的频率偏差小于 ±2.5%，无需外接昂贵的高频晶体。

一个频率为 4~32MHz 的外部晶振 XTH。

一个频率为 32.768KHz 的外部晶振 XTL。

一个频率为 32.8/38.4KHz 的内部时钟 RCL。

一个频率为 8~48MHz 输出的 PLL。

1.5 工作模式

1) 运行模式(Active Mode)：CPU 运行，周边功能模块运行。
2) 休眠模式(Sleep Mode)：CPU 停止运行，周边功能模块运行。
3) 深度休眠模式(Deep sleep Mode)：CPU 停止运行，高速时钟停止，低功耗功能模块运行。

1.6 通用 IO 端口

最多可提供 88 个 GPIO 端口，其中部分 GPIO 与模拟端口复用。每个端口由独立的控制寄存器位来控制，支持 FAST IO。支持边沿触发中断和电平触发中断，可从各种功耗模式下把 MCU 唤醒到工作模式。支持位置位，位清零，位置位清零操作。支持 Push-Pull CMOS 推挽输出、Open-Drain 开漏输出。内置上拉电阻、下拉电阻，带有施密特触发器输入滤波功能。输出驱动能力可配置，最大支持 18mA 的电流驱动能力。所有通用 IO 可支持外部异步中断。

1.7 中断控制器

Cortex-M0+处理器内置了嵌套向量中断控制器（NVIC），支持最多 32 个中断请求（IRQ）输入；有四个中断优先级，可处理复杂逻辑，能够进行实时控制和中断处理。

32 个中断入口向量地址，分别为：

中断向量号	中断来源
[0]	GPIO_PA
[1]	GPIO_PB
[2]	GPIO_PC/GPIO_PE
[3]	GPIO_PD/GPIO_PF
[4]	DMAC
[5]	TIM3
[6]	UART0/UART2
[7]	UART1/UART3
[8]	保留
[9]	保留

[10]	SPI0
[11]	SPI1
[12]	I2C0
[13]	I2C1
[14]	TIM0
[15]	TIM1
[16]	TIM2
[17]	保留
[18]	TIM4
[19]	TIM5
[20]	TIM6
[21]	PCA
[22]	WDT
[23]	保留
[24]	ADC/DAC
[25]	保留
[26]	VC0
[27]	VC1/VC2
[28]	LVD
[29]	LCD
[30]	RAM FLASH
[31]	CLKTRIM

1.8 复位控制器

本产品具有 7 个复位信号来源，每个复位信号可以让 CPU 重新运行，绝大多数寄存器会被重新复位，程序计数器 PC 会指向起始地址。

	中断来源
[0]	上电掉电复位 POR BOR
[1]	外部 Reset Pin 复位
[2]	WDT 复位
[3]	PCA 复位
[4]	Cortex-M0+ LOCKUP 硬件复位
[5]	Cortex-M0+ SYSRESETREQ 软件复位
[6]	LVD 复位

1.9 DMAC

DMAC（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMAC 能提高系统性能。

- DMAC 配有独立的总线，所以即便是在使用 CPU 总线的同时，DMAC 也可进行传输操作。
- 由 2 条通道组成，能执行 2 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式，并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作时，可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线(AHB)，支持 32 位地址空间(4GB)。

1.10 定时器/计数器

类型	名称	位宽	预除频	计数方向	PWM	捕获	互补输出
通用定时器	TIM0	16/32	1/2/4/8/16 32/64/256	上计数/ 下计数/ 上下计数	2	2	1
	TIM1	16/32	1/2/4/8/16/ 32/64/256	上计数/ 下计数/ 上下计数	2	2	1
	TIM2	16/32	1/2/4/8/16/ 32/64/256	上计数/ 下计数/ 上下计数	2	2	1
	TIM3	16/32	1/2/4/8/16/ 32/64/256	上计数/ 下计数/ 上下计数	6	6	3
可编程计数阵列	PCA	16	2/4/8/16/32	上计数	5	5	无
高级定时器	TIM4	16	1/2/4/8/16/ 64/256/1024	上计数/ 下计数/ 上下计数	2	2	1
	TIM5	16	1/2/4/8/16/ 64/256/1024	上计数/ 下计数/	2	2	1

				上下计数			
	TIM6	16	1/2/4/8/16/ 64/256/1024	上计数/ 下计数/ 上下计数	2	2	1

通用定时器包含四个定时器 TIM0/1/2/3。

通用定时器特性

- PWM 独立输出，互补输出
- 捕获输入
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能

TIM0/1/2 功能完全相同。TIM0/1/2 是同步定时/计数器，可以作为 16 位自动重装载功能的定时/计数器，也可以作为 32 位无重载功能的定时/计数器。TIM0/1/2 每个定时器都具有 2 路捕获比较功能，可以产生 2 路 PWM 独立输出或 1 组 PWM 互补输出。具有死区控制功能。

TIM3 是多通道的通用定时器,具有 TIM0/1/2 的所有功能，可以产生 3 组 PWM 互补输出或 6 路 PWM 独立输出，最多 6 路输入捕获。具有死区控制功能。

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程，以提供输入捕捉，输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

高级定时器 Advanced Timer 包含三个定时器 TIM4/5/6。TIM4/5/6 是功能相同的高性能计数器，可用于计数产生不同形式的时钟波形，1 个定时器可以产生互补的一对 PWM 或者独立的 2 路 PWM 输出，可以捕获外界输入进行脉冲宽度或周期测量。

Advanced Timer 基本的功能及特性如表所示：

波形模式	锯齿波、三角波
基本功能	• 递加、递减计数方向

	• 软件同步
	• 硬件同步
	• 缓存功能
	• 正交编码计数
	• 通用PWM输出
	• 保护机制
	• AOS关联动作
中断类型	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断

1.11 看门狗 WDT

WDT（Watch Dog Timer）是一个可配置的 20 位定时器，在 MCU 异常的情况下提供复位；内建 10KHz 低速时钟输入作为计数器时钟。调试模式下，可选择暂停或继续运行；只有写入特定序列才能重启 WDT。

1.12 通用异步收发器 UART0~UART3

4 路通用异步收发器（Universal Asynchronous Receiver/Transmitter），UART0~UART3。

通用 UART 基本功能：

- 半双工和全双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式
- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- DMAC 硬件传输握手
- 硬件流控
- 支持单线模式

1.13 同步串行接口 SPI

2 路同步串行接口（Serial Peripheral Interface）

SPI 基本特性：

- 通过编程可以配置为主机或者从机
- 四线传输方式，全双工通信
- 主机模式 7 种波特率可配置
- 主机模式最大波特率为 1/2 PCLK
- 从机模式最大波特率为 1/4 PCLK
- 可配置的串行时钟极性和相位
- 支持中断
- 8 位数据传输，先传输高位后低位
- 支持 DMA 软件/硬件访问

1.14 I2C 总线

2 路 I2C，采用串行同步时钟，可实现设备之间以不同的速率传输数据。

I2C 基本特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

1.15 蜂鸣器 Buzzer

4 个通用定时器复用输出为 Buzzer 提供可编程驱动频率。该蜂鸣器端口可提供 18mA 的 sink 电流，互补输出，不需要额外的三极管。

1.16 时钟校准电路

内建时钟校准电路，可以通过外部精准的晶振时钟校准内部 RC 时钟，亦可使用内部 RC 时钟去检验外部晶振时钟是否工作正常。

时钟校准基本特性：

- 校准模式
- 监测模式
- 32 位参考时钟计数器可加载初值
- 32 位待校准时钟计数器可配置溢出值
- 6 种参考时钟源
- 5 种待校准时钟源
- 支持中断方式

1.17 唯一识别号 UID

每颗芯片出厂前具备唯一的 10 字节设备标识号，包括 wafer lot 信息，以及芯片坐标信息等。UID 地址为：0x00100E74 - 0x00100E7D。

1.18 CRC16/32 硬件循环冗余校验码

CRC16 符合 ISO/IEC13239 中给出的多项式 $X^{16} + X^{12} + X^5 + 1$。

CRC32 符合 ISO/IEC13239 中给出的多项式 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$。

1.19 AES 硬件加密

AES（The Advanced Encryption Standard）是美国国家标准技术研究所（NIST）在 2000 年 10 月 2 日正式宣布的新的数据加密标准。AES 的分组长度固定为 128 Bit，而密钥长度支持 128/192/256 Bit。

1.20 TRNG 真随机数发生器

TRNG 是一个真随机数发生器，用来产生真随机数。

1.21 12 Bit SARADC

单调不失码的 12 位逐次逼近型模数转换器，在 24MHz ADC 时钟下工作时，采样率达到 1Msps。参考电压可选择片内精准电压（1.5V 或 2.5V）或从外部输入或电源电压。30 个输入通道，包括 26 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 电源电压、1 路内建 BGR 1.2V 电压、DAC 内部输出。内置电压跟随器，可测量高输出阻抗的信号。

SAR ADC 基本特性：

- 12 位转换精度；
- 1Msps 转换速度；
- 30 个输入通道，包括 26 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、1 路内建 BGR 1.2V 电压、DAC 内部输出；
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5V 参考电压、内置 2.5V 参考电压；
- ADC 的电压输入范围：0~Vref；
- 4 种转换模式：单次转换、顺序扫描连续转换、插队扫描连续转换、连续转换累加；
- 输入通道电压阈值监测；
- 软件可配置 ADC 的转换速率；
- 内置电压跟随器，可测量高输出阻抗的信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

1.22 12 Bit DAC

1 通道 12bit 500Ksps DAC，可以进行数模转换。

1.23 电压比较器 VC

内建 3 路 VC，芯片管脚电压监测/比较电路。16 个可配置的正外部输入通道，11 个可配置的负外部输入通道；5 个内部负输入通道，包括 1 路内部温度传感器电压、1 路内建 BGR 2.5V 参考电压、1 路内建 BGR 1.2V 电压、1 路 64 阶电阻分压。VC 输出可供通用定时器 TIM0/1/2/3 与可编程计数阵列 PCA 捕获、门控、外部计数时钟使用。可根据上升/下降边沿产生异步中断，从低功耗模式下唤醒 MCU。可配置的软件防抖功能。

1.24 低电压检测器 LVD

对芯片电源电压或芯片管脚电压进行检测。16 档电压监测值（1.8～3.3V）。可根据上升/下降边沿产生异步中断或复位。具有硬件迟滞电路和可配置的软件防抖功能。

LVD 基本特性：

- 4 路监测源，AVCC、PC13、PB08、PB07；
- 16 阶阈值电压，1.8~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

1.25 运放 OPA

OPA 可以灵活配置，适用于简易滤波器和电压跟随器应用。可以作为 DAC 输出缓存器使用,也可以配置为运放使用。

1.26 LCD 驱动

LCD 控制器是一款适用于单色无源液晶显示器（LCD）的数字控制器/驱动器，最多具有 8 个公用端子（COM）和 48 个区段端子（SEG），用以驱动 208 (4x52)或 384 (8x48)个 LCD 图像元素。可以选择电容分压或电阻分压，支持内部电阻分压。内部电阻分压可以调节对比度。支持 DMA 硬件数据传输。

LCD 基本特性：

- 高度灵活的帧速率控制。
- 支持静态、1/2、1/3、1/4、1/6 和 1/8 占空比。
- 支持 1/2、1/3 偏置。
- 多达 16 个寄存器的 LCD 数据 RAM。
- 可通过软件配置 LCD 的对比度。
- 3 种驱动波形生成方式
  - 内部电阻分压、外部电阻分压，外部电容分压方式

– 可通过软件配置内部电阻分压方式的功耗，从而匹配 LCD 面板所需的电容电荷

- 支持低功耗模式：LCD 控制器可在 Active、Sleep、DeepSleep 模式下进行显示。
- 可配置帧中断。
- 支持 LCD 闪烁功能且可配置多种闪烁频率
- 未使用的 LCD 区段和公共引脚可配置为数字或模拟功能。

1.27 嵌入式调试系统

嵌入式调试解决方案，提供全功能的实时调试器，配合标准成熟的 Keil/IAR 等调试开发软件。支持 4 个硬断点以及多个软断点。

1.28 编程模式

支持两种编程模式：在线编程、离线编程。

支持两种编程协议：ISP 协议、SWD 协议。

支持统一编程接口：ISP 协议与 SWD 协议共用 SWD 端口。

当复位时 BOOT0（PF11）管脚为高电平，芯片工作于 ISP 编程模式，可通过 ISP 协议对 Flash 进行编程。

当复位时 BOOT0（PF11）管脚为低电平，芯片工作于用户模式，芯片执行 Flash 内的程序代码，可通过 SWD 协议对 Flash 进行编程。

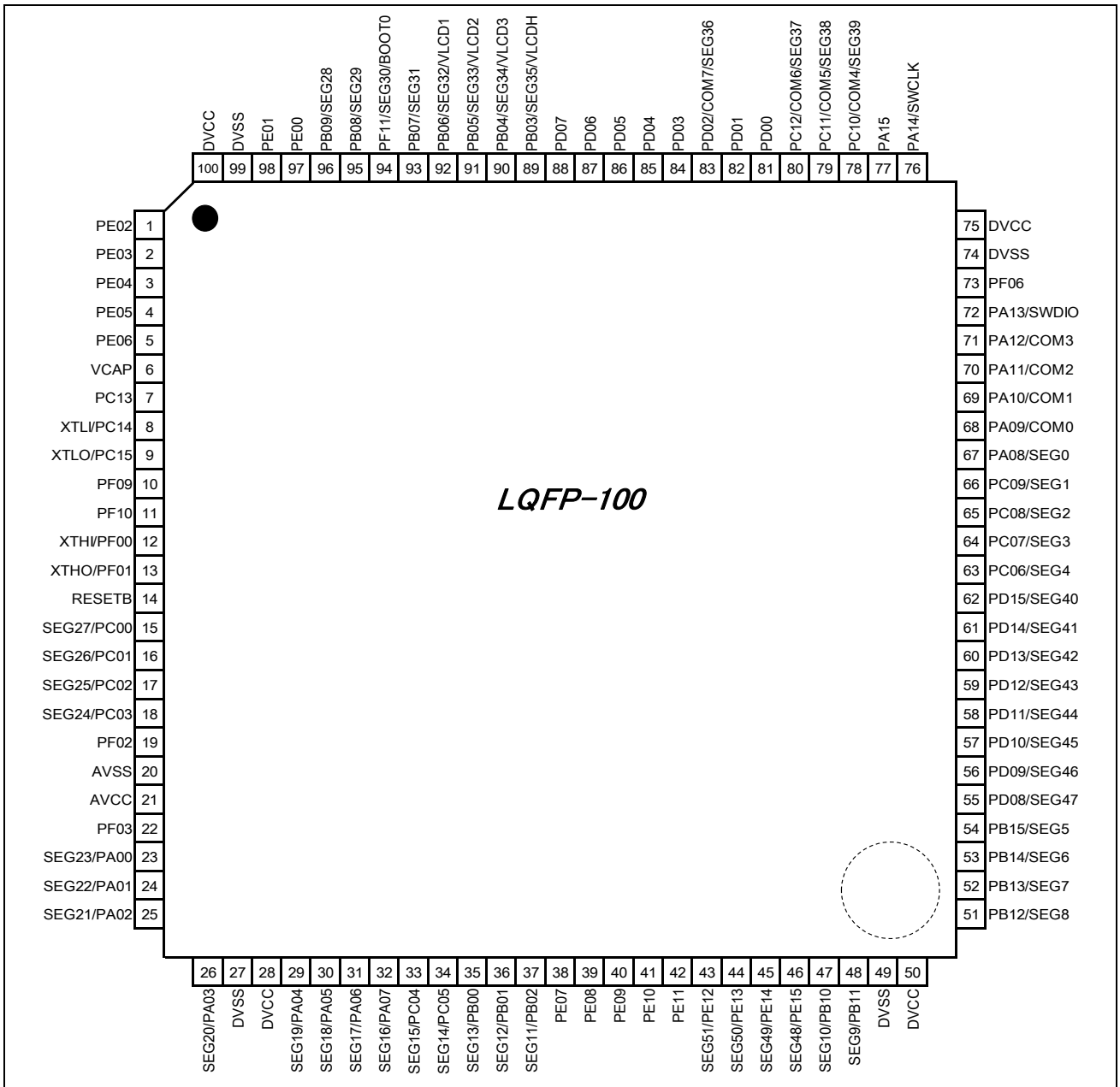
1.29 高安全性

加密型嵌入式调试解决方案，提供全功能的实时调试器。

2 引脚配置及功能

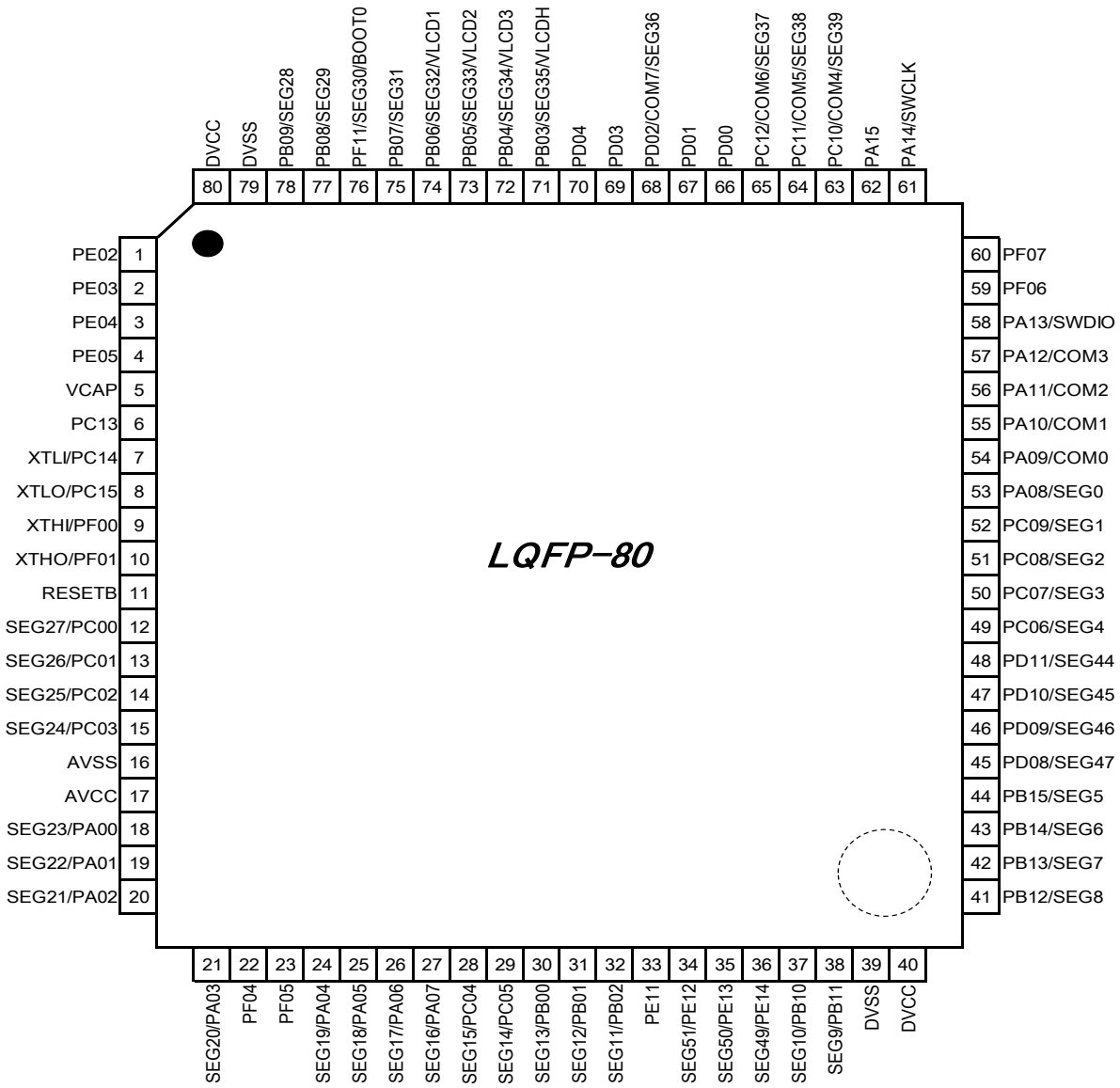
2.1 封装示意图

HC32F176PATA / HC32F196PCTA



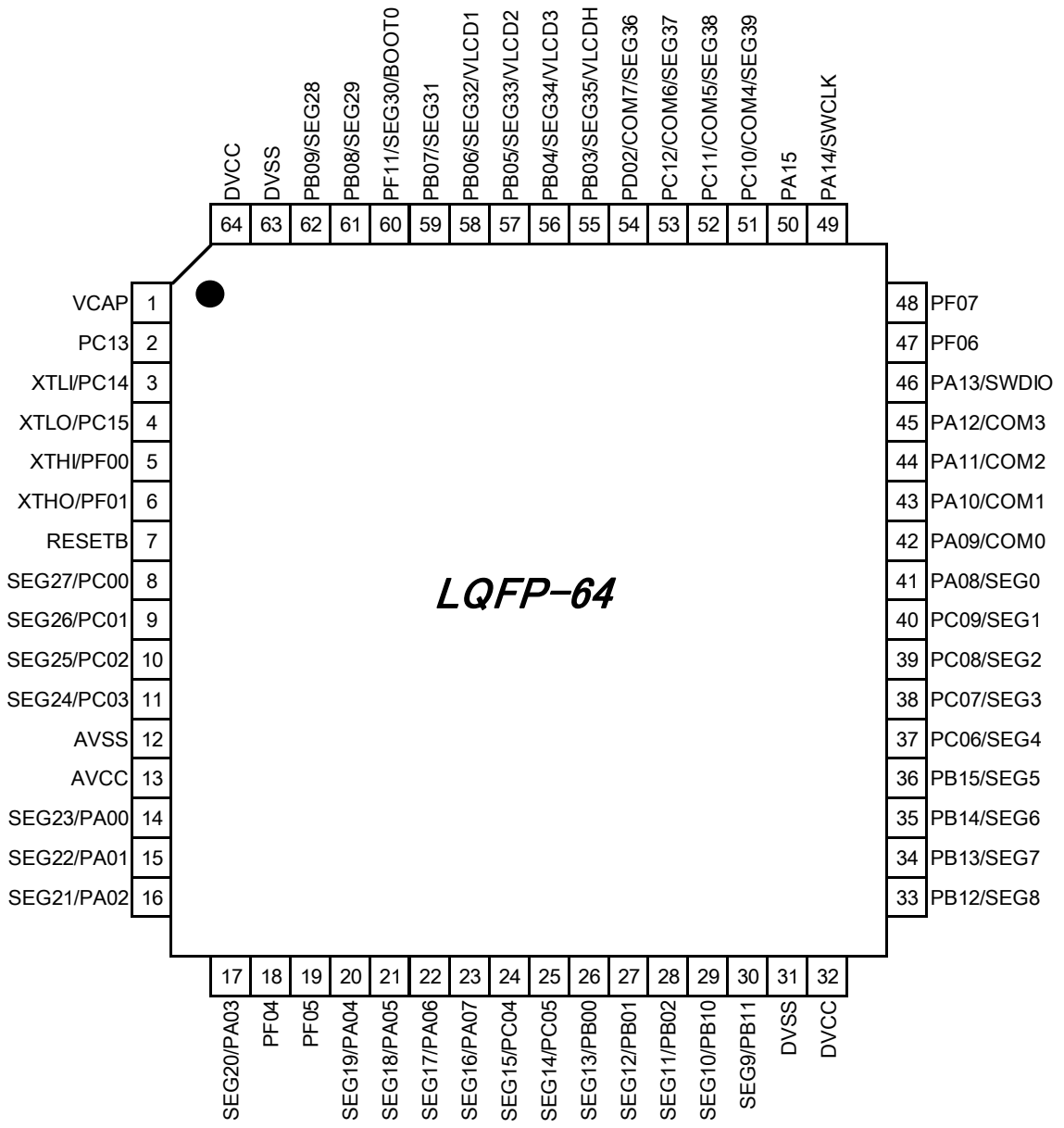
注：

- 在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。
- 该封装未引出的 IO 详见 2.2 引脚功能说明。



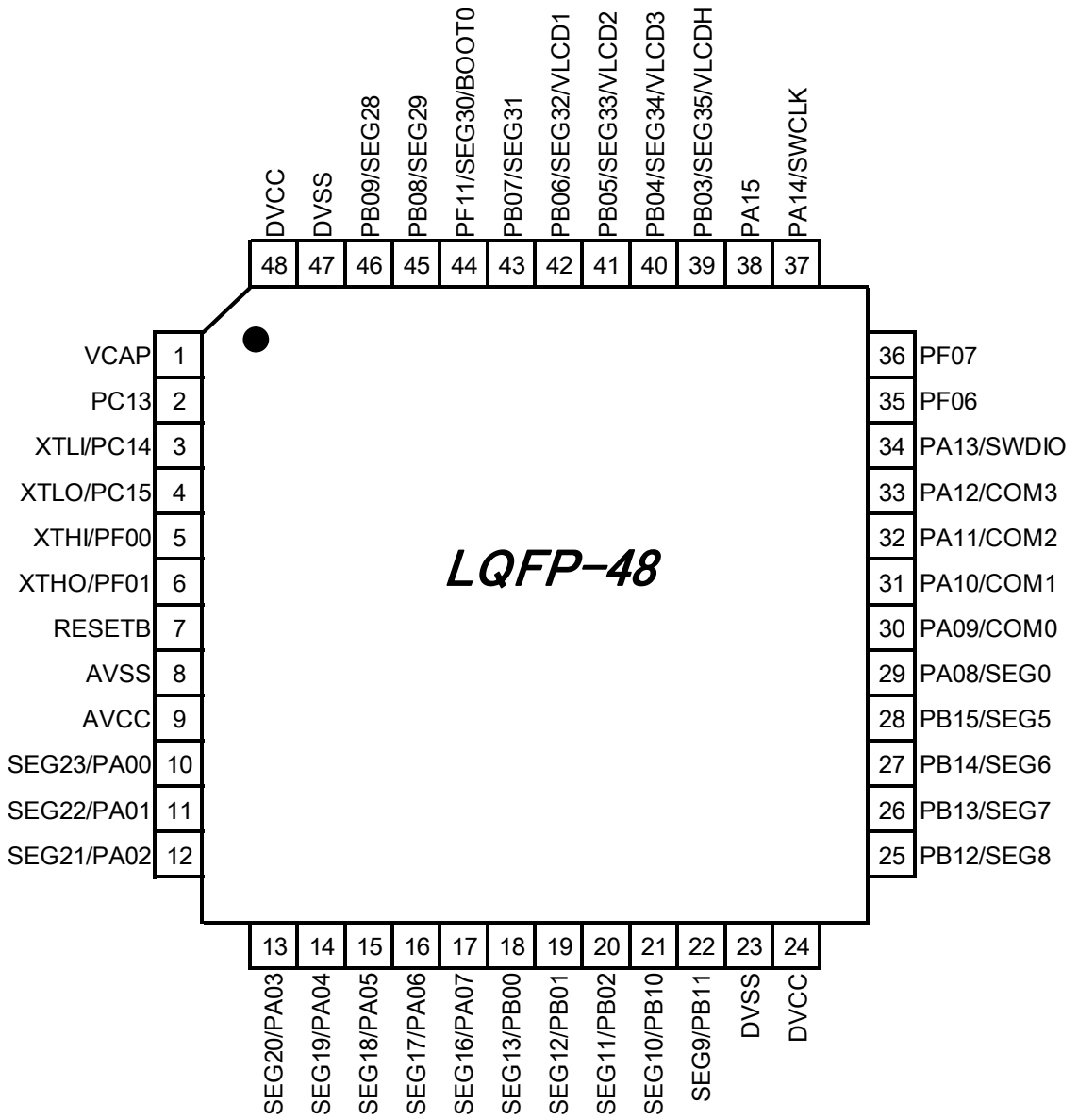
注：

- **在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。**
- **该封装未引出的 IO 详见 2.2 引脚功能说明。**



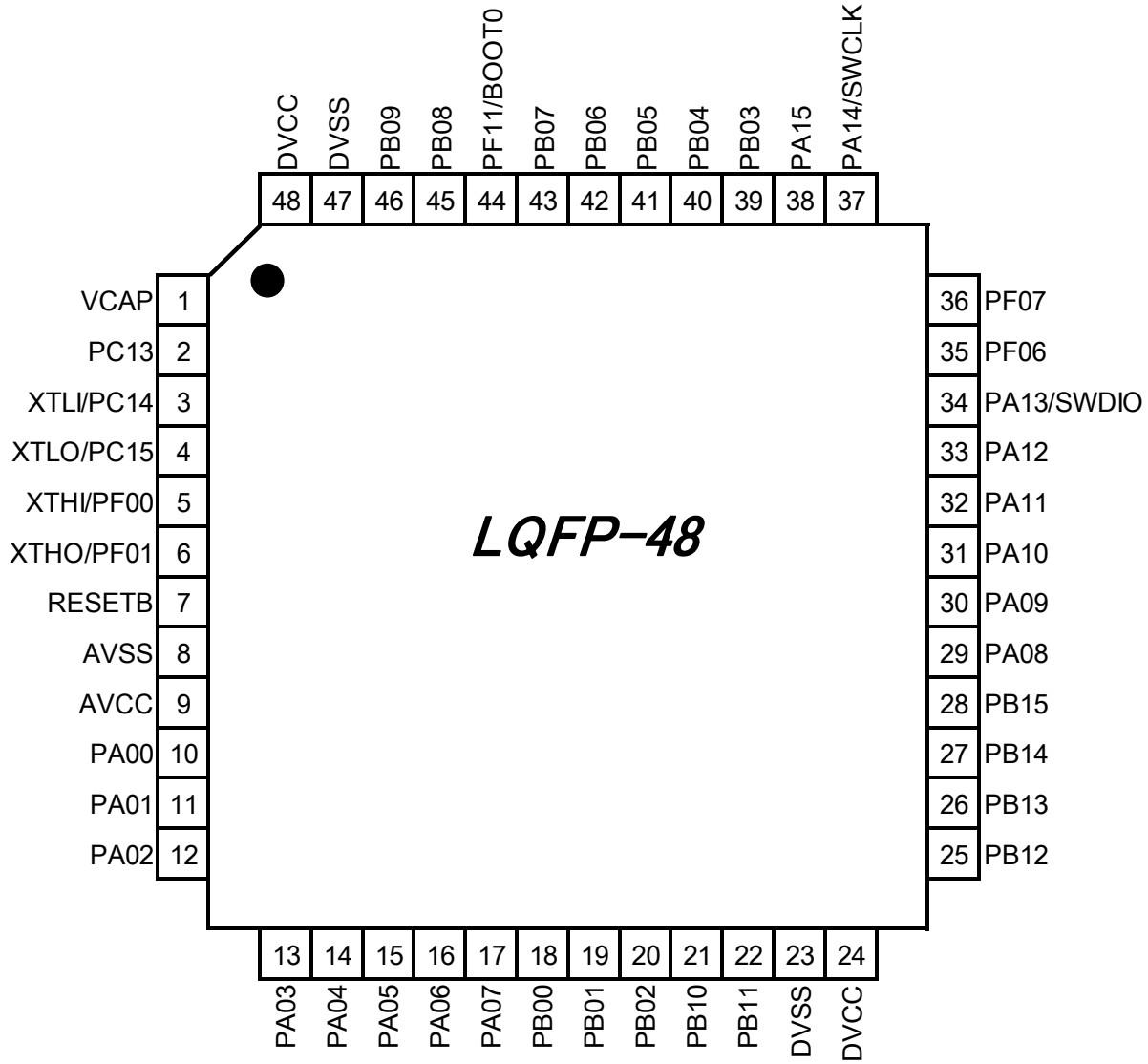
注：

- **在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。**
- **该封装未引出的 IO 详见 2.2 引脚功能说明。**



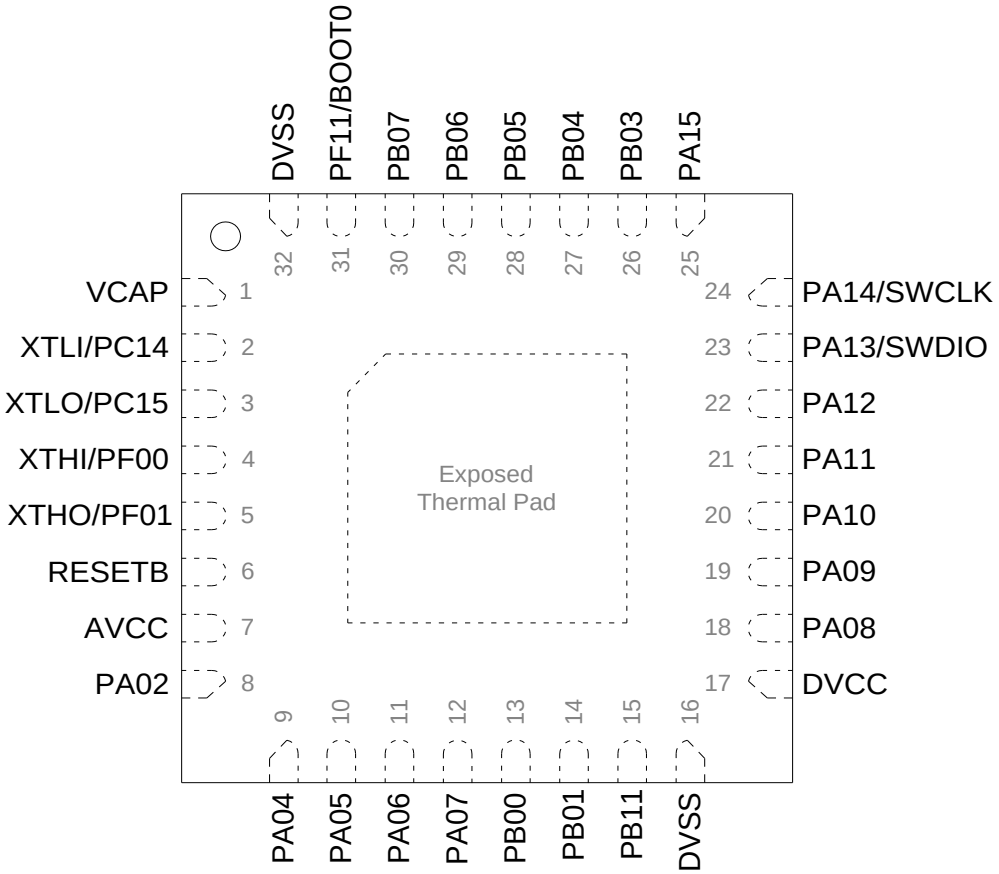
注：

- **在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。**
- **该封装未引出的 IO 详见 2.2 引脚功能说明。**



注：

- 在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。
- 该封装未引出的 IO 详见 2.2 引脚功能说明。



注：

- **Exposed Thermal Pad 需要连接到 DVSS。**
- **在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。**
- **该封装未引出的 IO 详见 2.2 引脚功能说明。**

2.2 引脚功能说明

LQFP100	LQFP80	LQFP64	LQFP48	QFN32	NAME	DIGITAL	ANALOG
1	1				PE02	PCA_ECI	
2	2				PE03	PCA_CH0	
3	3				PE04	PCA_CH1	
4	4				PE05	PCA_CH2	
5					PE06	PCA_CH3	
6	5	1	1	1	VCAP		
7	6	2	2		PC13	TIM3_CH1B	LVD0
8	7	3	3	2	PC14		XTLI
9	8	4	4	3	PC15		XTLO
10					PF09	TIM0_CHA	
11					PF10	TIM0_CHB	
12	9	5	5	4	PF00	I2C0_SDA UART1_TXD	XTHI
13	10	6	6	5	PF01	I2C0_SCL TIM4_CHB UART1_RXD	XTHO
14	11	7	7	6	RESETB		
15	12	8			PC00	UART1_CTS UART2_RTS	AIN10 VC0_INP0 VC1_INN0 SEG27
16	13	9			PC01	TIM5_CHB UART1_RTS UART2_CTS	AIN11 VC0_INP1 VC1_INN1 SEG26
17	14	10			PC02	SPI1_MISO UART2_RXD	AIN12 VC0_INP2 VC1_INN2 SEG25
18	15	11			PC03	SPI1_MOSI UART2_TXD	AIN13 VC0_INP3 VC1_INN3 SEG24
19					PF02		
20	16	12	8		AVSS		
21	17	13	9	7	AVCC		
22					PF03		

23	18	14	10		PA00	UART1_CTS TIM0_ETR VC0_OUT TIM1_CHA TIM3_ETR TIM0_CHA	AIN0 VC0_INP4 VC0_INN0 VC1_INP0 VC1_INN4 SEG23
24	19	15	11		PA01	UART1_RTS TIM0_CHB TIM1_ETR TIM1_CHB HCLK_OUT SPI1_MOSI	AIN1 VC0_INP5 VC0_INN1 VC1_INP1 VC1_INN5 SEG22
25	20	16	12	8	PA02	UART1_TXD TIM0_CHA VC1_OUT TIM1_CHA TIM2_CHA PCLK_OUT SPI1_MISO	AIN2 VC0_INP6 VC0_INN2 VC1_INP2 SEG21
26	21	17	13		PA03	UART1_RXD TIM0_GATE TIM1_CHB TIM2_CHB SPI1_CS TIM3_CH1A TIM5_CHA	AIN3 VC0_INP7 VC0_INN3 VC1_INP3 SEG20
27					DVSS		
28					DVCC		
	22	18			PF04		
	23	19			PF05		
29	24	20	14	9	PA04	SPI0_CS UART1_TXD PCA_CH4 TIM2_ETR TIM5_CHA LVD_OUT TIM3_CH2B	AIN4 VC0_INP8 VC0_INN4 VC1_INP4 DAC_OUT OP3_OUT SEG19

30	25	21	15	10	PA05	SPI0_SCK TIM0_ETR PCA_ECI TIM0_CHA TIM5_CHB XTL_OUT XTH_OUT	AIN5 VC0_INP9 VC0_INN5 VC1_INP5 VC2_INP0 VC2_INN0 OP3_OUT1 SEG18
31	26	22	16	11	PA06	SPI0_MISO PCA_CH0 TIM3_BK TIM1_CHA VC0_OUT TIM3_GATE	AIN6 VC0_INP10 VC0_INN6 OP3_OUT2 SEG17
32	27	23	17	12	PA07	SPI0_MOSI PCA_CH1 HCLK_OUT TIM3_CH0B TIM2_CHA VC1_OUT TIM4_CHB	AIN7 VC0_INP11 VC0_INN7 OP3_OUT3 SEG16
33	28	24			PC04	TIM2_ETR IR_OUT VC2_OUT	AIN14 VC0_INN8 OP3_OUT4 SEG15
34	29	25			PC05	TIM6_CHB PCA_CH4	AIN15 VC0_INN9 OP3_INN SEG14
35	30	26	18	13	PB00	PCA_CH2 TIM3_CH1B TIM5_CHB RCH_OUT RCL_OUT PLL_OUT	AIN8 VC0_INN10 VC1_INN6 OP3_INP SEG13
36	31	27	19	14	PB01	PCA_CH3 PCLK_OUT TIM3_CH2B TIM6_CHB VC2_OUT TCLK_OUT	AIN9/EXVREF VC1_INP6 VC1_INN7 VC2_INP1 VC2_INN1 SEG12

37	32	28	20		PB02	PCA_ECI TIM4_CHA TIM1_BK TIM0_BK TIM2_BK	AIN16 VC1_INP7 VC1_INN8 SEG11
38					PE07	TIM3_ETR	
39					PE08	TIM3_CH0B	
40					PE09	TIM3_CH0A	VC2_INP2
41					PE10	TIM3_CH1B	VC2_INP3
42	33				PE11	TIM3_CH1A	VC2_INP4 VC2_INN2
43	34				PE12	TIM3_CH2B SPI0_CS UART3_CTS	SEG51
44	35				PE13	TIM3_CH2A SPI0_SCK UART3_RTS	AIN25 VC2_INP5 SEG50
45	36				PE14	TIM3_CH0B SPI0_MISO UART3_RXD	AIN24 VC2_INP6 SEG49
46					PE15	TIM3_BK SPI0_MOSI UART3_TXD	AIN23 VC2_INP7 VC2_INN3 SEG48
47	37	29	21		PB10	I2C1_SCL SPI1_SCK TIM1_CHA TIM3_CH1A UART1_RTS	AIN17 VC1_INP8 SEG10
48	38	30	22	15	PB11	I2C1_SDA TIM1_CHB TIM2_GATE TIM6_CHA UART1_CTS	AIN18 VC2_INP8 VC2_INN4 SEG9
49	39	31	23	16	DVSS		
50	40	32	24	17	DVCC		

51	41	33	25		PB12	SPI1_CS TIM3_BK TIM0_BK TIM6_CHA	AIN19 VC1_INP9 SEG8
52	42	34	26		PB13	SPI1_SCK I2C1_SCL TIM3_CH0B TIM1_CHA TIM1_GATE TIM6_CHB	AIN20 VC1_INP10 SEG7
53	43	35	27		PB14	SPI1_MISO I2C1_SDA TIM3_CH1B TIM0_CHA TIM1_BK	AIN21 VC1_INP11 VC2_INP9 VC2_INN5 SEG6
54	44	36	28		PB15	SPI1_MOSI TIM3_CH2B TIM0_CHB TIM0_GATE	AIN22 SEG5
55	45				PD08		SEG47
56	46				PD09		VC2_INP10 SEG46
57	47				PD10		VC2_INP11 VC2_INN6 SEG45
58	48				PD11		VC2_INP12 VC2_INN7 SEG44
59					PD12	UART2_RTS	SEG43
60					PD13	UART2_RX	SEG42
61					PD14	UART2_TX	SEG41
62					PD15	UART2_CTS	SEG40
63	49	37			PC06	PCA_CH0 TIM4_CHA TIM2_CHA UART3_RXD	SEG4

64	50	38			PC07	PCA_CH1 TIM5_CHA TIM2_CHB UART3_TXD	VC2_INP13 VC2_INN8 SEG3
65	51	39			PC08	PCA_CH2 TIM6_CHA TIM2_ETR UART3_CTS	SEG2
66	52	40			PC09	PCA_CH3 TIM4_CHB TIM1_ETR UART3_RTS	SEG1
67	53	41	29	18	PA08	UART0_TXD TIM3_CH0A TIM1_GATE TIM4_CHA TIM3_BK	SEG0
68	54	42	30	19	PA09	UART0_TXD TIM3_CH1A TIM0_BK I2C0_SCL HCLK_OUT TIM5_CHA	COM0
69	55	43	31	20	PA10	UART0_RXD TIM3_CH2A TIM2_BK I2C0_SDA TIM2_GATE PCLK_OUT TIM6_CHA	COM1
70	56	44	32	21	PA11	UART0_CTS TIM3_GATE I2C1_SCL VC0_OUT SPI0_MISO TIM4_CHB	COM2
71	57	45	33	22	PA12	UART0_RTS TIM3_ETR I2C1_SDA VC1_OUT SPI0_MOSI	COM3

72	58	46	34	23	PA13	IR_OUT UART0_RXD LVD_OUT TIM3_ETR VC2_OUT	SWDIO
73	59	47	35		PF06	I2C1_SCL UART0_CTS	
	60	48	36		PF07	I2C1_SDA UART0_RTS	
74					DVSS		
75					DVCC		
76	61	49	37	24	PA14	UART1_TXD UART0_TXD TIM3_CH2A LVD_OUT RCH_OUT RCL_OUT PLL_OUT	SWCLK
77	62	50	38	25	PA15	SPI0_CS UART1_RXD TIM0_ETR TIM0_CHA TIM3_CH1A	
78	63	51			PC10	PCA_CH2	COM4/SEG39
79	64	52			PC11	PCA_CH3	COM5/SEG38
80	65	53			PC12	PCA_CH4	COM6/SEG37
81	66				PD00	SPI1_CS	
82	67				PD01	SPI1_SCK	
83	68	54			PD02	PCA_ECI TIM1_ETR	COM7/SEG36
84	69				PD03	UART1_CTS SPI1_MISO	
85	70				PD04	UART1_RTS SPI1_MOSI	
86					PD05	UART1_TX	

87					PD06	UART1_RX	
88					PD07	UART1_TX	
89	71	55	39	26	PB03	SPI0_SCK TIM0_CHB TIM1_GATE TIM3_CH0A XTL_OUT XTH_OUT	VC1_INN9 SEG35/VLCDH
90	72	56	40	27	PB04	SPI0_MISO PCA_CH0 TIM2_BK UART0_CTS TIM2_GATE TIM3_CH0B	VC0_INP12 VC1_INP12 SEG34/VLCD3
91	73	57	41	28	PB05	SPI0_MOSI TIM1_BK PCA_CH1 UART0_RTS	VC0_INP13 SEG33/VLCD2
92	74	58	42	29	PB06	I2C0_SCL UART0_TXD TIM1_CHB TIM0_CHA TIM3_CH0A	VC0_INP14 VC1_INP14 SEG32/VLCD1
93	75	59	43	30	PB07	I2C0_SDA UART0_RXD TIM2_CHB TIM0_CHB	VC1_INP15 LVD2 SEG31
94	76	60	44	31	BOOT0 /PF11		SEG30
95	77	61	45		PB08	I2C0_SCL TIM1_CHA TIM2_CHA TIM0_GATE TIM3_CH2A UART0_TXD	LVD1 SEG29

96	78	62	46		PB09	I2C0_SDA IR_OUT SPI1_CS TIM2_CHA TIM2_CHB UART0_RXD	SEG28
97					PE00	TIM1_CHA	
98					PE01	TIM2_CHA	
99	79	63	47	32	DVSS		
100	80	64	48		DVCC		

每个引脚的数字功能由 PSEL 位域进行控制，详见下表。

Px_SEL							
0	1	2	3	4	5	6	7
PA00	UART1_CTS		TIM0_ETR	VC0_OUT	TIM1_CHA	TIM3_ETR	TIM0_CHA
PA01	UART1_RTS		TIM0_CHB	TIM1_ETR	TIM1_CHB	HCLK_OUT	SPI1_MOSI
PA02	UART1_TXD	TIM0_CHA	VC1_OUT	TIM1_CHA	TIM2_CHA	PCLK_OUT	SPI1_MISO
PA03	UART1_RXD	TIM0_GATE	TIM1_CHB	TIM2_CHB	SPI1_CS	TIM3_CH1A	TIM5_CHA
PA04	SPI0_CS	UART1_TXD	PCA_CH4	TIM2_ETR	TIM5_CHA	LVD_OUT	TIM3_CH2B
PA05	SPI0_SCK	TIM0_ETR	PCA_ECI	TIM0_CHA	TIM5_CHB	XTL_OUT	XTH_OUT
PA06	SPI0_MISO	PCA_CH0	TIM3_BK	TIM1_CHA	VC0_OUT	TIM3_GATE	
PA07	SPI0_MOSI	PCA_CH1	HCLK_OUT	TIM3_CH0B	TIM2_CHA	VC1_OUT	TIM4_CHB
PA08	UART0_TXD	TIM3_CH0A			TIM1_GATE	TIM4_CHA	TIM3_BK
PA09	UART0_TXD	TIM3_CH1A	TIM0_BK	I2C0_SCL		HCLK_OUT	TIM5_CHA
PA10	UART0_RXD	TIM3_CH2A	TIM2_BK	I2C0_SDA	TIM2_GATE	PCLK_OUT	TIM6_CHA
PA11	UART0_CTS	TIM3_GATE	I2C1_SCL		VC0_OUT	SPI0_MISO	TIM4_CHB
PA12	UART0_RTS	TIM3_ETR	I2C1_SDA		VC1_OUT	SPI0_MOSI	
PA13	IR_OUT	UART0_RXD	LVD_OUT	TIM3_ETR			VC2_OUT
PA14	UART1_TXD	UART0_TXD	TIM3_CH2A	LVD_OUT	RCH_OUT	RCL_OUT	PLL_OUT
PA15	SPI0_CS	UART1_RXD		TIM0_ETR	TIM0_CHA	TIM3_CH1A	
PB00	PCA_CH2	TIM3_CH1B		TIM5_CHB	RCH_OUT	RCL_OUT	PLL_OUT
PB01	PCA_CH3	PCLK_OUT	TIM3_CH2B	TIM6_CHB		VC2_OUT	TCLK_OUT
PB02		PCA_ECI		TIM4_CHA	TIM1_BK	TIM0_BK	TIM2_BK
PB03	SPI0_SCK	TIM0_CHB	TIM1_GATE	TIM3_CH0A		XTL_OUT	XTH_OUT
PB04	SPI0_MISO	PCA_CH0	TIM2_BK	UART0_CTS	TIM2_GATE	TIM3_CH0B	
PB05	SPI0_MOSI		TIM1_BK	PCA_CH1			UART0_RTS
PB06	I2C0_SCL	UART0_TXD	TIM1_CHB	TIM0_CHA		TIM3_CH0A	
PB07	I2C0_SDA	UART0_RXD	TIM2_CHB		TIM0_CHB		
PB08	I2C0_SCL	TIM1_CHA		TIM2_CHA	TIM0_GATE	TIM3_CH2A	UART0_TXD
PB09	I2C0_SDA	IR_OUT	SPI1_CS	TIM2_CHA		TIM2_CHB	UART0_RXD
PB10	I2C1_SCL	SPI1_SCK	TIM1_CHA		TIM3_CH1A		UART1_RTS
PB11	I2C1_SDA	TIM1_CHB		TIM2_GATE	TIM6_CHA		UART1_CTS
PB12	SPI1_CS	TIM3_BK		TIM0_BK			TIM6_CHA
PB13	SPI1_SCK	I2C1_SCL	TIM3_CH0B		TIM1_CHA	TIM1_GATE	TIM6_CHB
PB14	SPI1_MISO	I2C1_SDA	TIM3_CH1B	TIM0_CHA			TIM1_BK
PB15	SPI1_MOSI	TIM3_CH2B	TIM0_CHB	TIM0_GATE			
PC00			UART1_CTS	UART2_RTS			
PC01		TIM5_CHB	UART1_RTS			UART2_CTS	
PC02	SPI1_MISO			UART2_RXD			
PC03	SPI1_MOSI				UART2_TXD		
PC04		TIM2_ETR	IR_OUT	VC2_OUT			
PC05		TIM6_CHB	PCA_CH4				

PC06	PCA_CH0	TIM4_CHA	TIM2_CHA			UART3_RXD	
PC07	PCA_CH1	TIM5_CHA	TIM2_CHB			UART3_TXD	
PC08	PCA_CH2	TIM6_CHA	TIM2_ETR			UART3_CTS	
PC09	PCA_CH3	TIM4_CHB	TIM1_ETR			UART3_RTS	
PC10			PCA_CH2				
PC11			PCA_CH3				
PC12			PCA_CH4				
PC13			TIM3_CH1B				
PC14							
PC15							
PD00		SPI1_CS					
PD01		SPI1_SCK					
PD02	PCA_ECI		TIM1_ETR				
PD03	UART1_CTS	SPI1_MISO					
PD04	UART1_RTS	SPI1_MOSI					
PD05	UART1_TXD						
PD06	UART1_RXD						
PD07	UART1_TXD						
PD08							
PD09							
PD10							
PD11							
PD12		UART2_RTS					
PD13	UART2_RXD						
PD14	UART2_TXD						
PD15		UART2_CTS					
PE00	TIM1_CHA						
PE01	TIM2_CHA						
PE02	PCA_ECI						
PE03	PCA_CH0						
PE04	PCA_CH1						
PE05	PCA_CH2						
PE06	PCA_CH3						
PE07	TIM3_ETR						
PE08	TIM3_CH0B						
PE09	TIM3_CH0A						
PE10	TIM3_CH1B						
PE11	TIM3_CH1A						
PE12	TIM3_CH2B	SPI0_CS	UART3_CTS				
PE13	TIM3_CH2A	SPI0_SCK	UART3_RTS				
PE14	TIM3_CH0B	SPI0_MISO	UART3_RXD				

PE15	TIM3_BK	SPI0_MOSI	UART3_TXD				
PF00	I2C0_SDA		UART1_TXD				
PF01	I2C0_SCL	TIM4_CHB	UART1_RXD				
PF02							
PF03							
PF04							
PF05							
PF06	I2C1_SCL		UART0_CTS				
PF07	I2C1_SDA		UART0_RTS				
PF09	TIM0_CHA						
PF10	TIM0_CHB						
PF11							

2.3 模块信号说明

模块	引脚名称	描述
电源	DVCC	数字电源
	AVCC	模拟电源
	DVSS	数字地
	AVSS	模拟地
	VCAP	LDO内核供电输出（仅限内部电路使用，需外接不小于1uF的去耦电容）
ISP	BOOT0	当复位时 BOOT0（PF11）管脚为高电平，芯片工作于ISP编程模式，可通过ISP协议对Flash进行编程。 当复位时 BOOT0（PF11）管脚为低电平，芯片工作于用户模式，芯片执行Flash内的程序代码，可通过SWD协议对Flash进行编程。
ADC	AIN0~AIN23	ADC输入通道0~23
	ADC_VREF	ADC外部参考电压
VC	VCIN0~VCIN15	VC 输入0~15
	VC0_OUT	VC0比较输出
	VC1_OUT	VC1比较输出
	VC2_OUT	VC2比较输出
LVD	LVDIN0	电压侦测输入0
	LVDIN1	电压侦测输入1
	LVDIN2	电压侦测输入2
	LVD_OUT	电压侦测输出
OPA y=0~4	OPA_INN	OPA负端输入
	OPA_INP	OPA正端输入
	OPA_OUTy	OPA输出
LCD x=0~7 y=0-52 z=1,2,3,H	COMx	LCD公共端输出
	SEGy	LCD区段端输出
	VLCDz	外部电阻模式，外部电容模式使用管脚
UART x=0,1,2,3	UARTx_TXD	UARTx数据发送端
	UARTx_RXD	UARTx数据接收端
	UARTx_CTS	UARTx CTS
	UARTx_RTS	UARTx RTS
SPI x=0,1	SPIx_MISO	SPI模块主机输入从机输出数据信号
	SPIx_MOSI	SPI模块主机输出从机输入数据信号
	SPIx_SCK	SPI模块时钟信号
	SPIx_CS	SPI 片选
I2C	I2Cx_SDA	I2C模块数据信号

x=0,1	I2Cx_SCL	I2C模块时钟信号
通用定时器 TIMx x=0,1,2	TIMx_CHA	Timer的捕获输入比较输出A
	TIMx_CHB	Timer的捕获输入比较输出B
	TIMx_ETR	Timer的外部计数输入信号
	TIMx_GATE	Timer的门控信号
通用定时器 TIM3 y=0,1,2	TIM3_CHyA	Timer的捕获输入比较输出A
	TIM3_CHyB	Timer的捕获输入比较输出B
	TIM3_ETR	Timer的外部计数输入信号
	TIM3_GATE	Timer的门控信号
可编程计数阵列 PCA	PCA_ECI	外部时钟输入信号
	PCA_CH0	捕获输入/比较输出/PWM输出 0
	PCA_CH1	捕获输入/比较输出/PWM输出 1
	PCA_CH2	捕获输入/比较输出/PWM输出 2
	PCA_CH3	捕获输入/比较输出/PWM输出 3
	PCA_CH4	捕获输入/比较输出/PWM输出 4
高级定时器 Advanced Timer	TIM4_CHA	Advanced Timer4 比较输出/捕获输入端A
	TIM4_CHB	Advanced Timer4 比较输出/捕获输入端B
	TIM5_CHA	Advanced Timer5 比较输出/捕获输入端A
	TIM5_CHB	Advanced Timer5 比较输出/捕获输入端B
	TIM6_CHA	Advanced Timer6 比较输出/捕获输入端A
	TIM6_CHB	Advanced Timer6 比较输出/捕获输入端B

表 2-1 模块信号说明

注意：

– IO 端口复位为输入高阻状态，休眠模式和深度休眠模式保持之前的端口状态。

3 系统结构

3.1 概述

本产品系统由以下部分组成：

- 2 个 AHB 总线 Master：
  - Cortex-M0+
  - DMA 控制器
- 4 个 AHB 总线 Slaves：
  - FLASH 存储器
  - SRAM 存储器
  - AHB0，AHB to APB Bridge，包含所有 APB 接口外设
  - AHB1，包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。如下图所示：

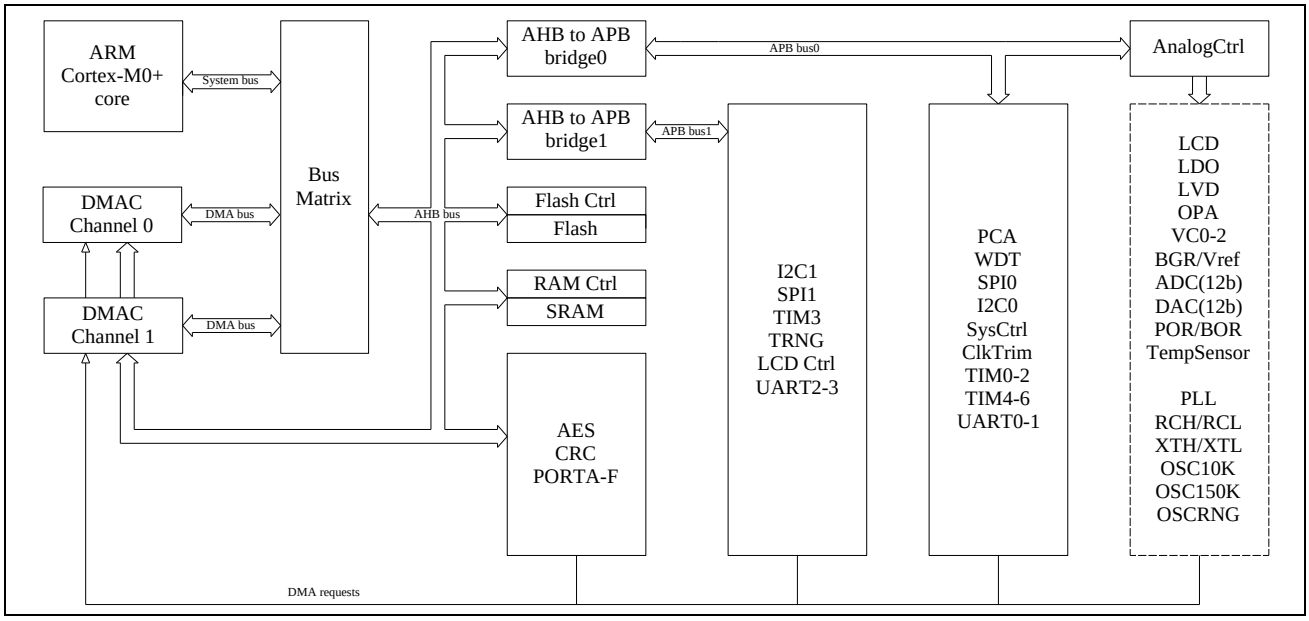
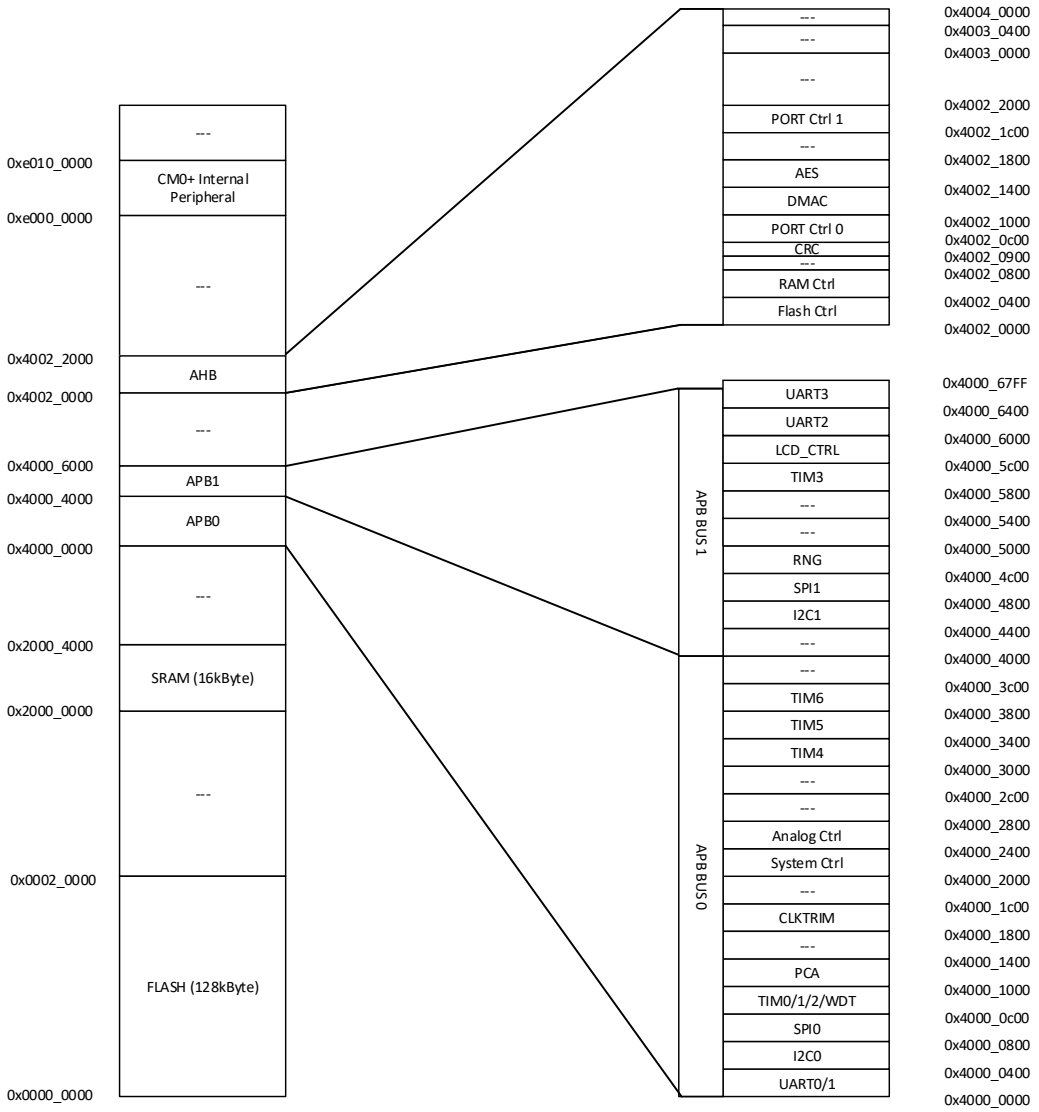


图 3-1 系统架构示意图

3.2 系统地址划分

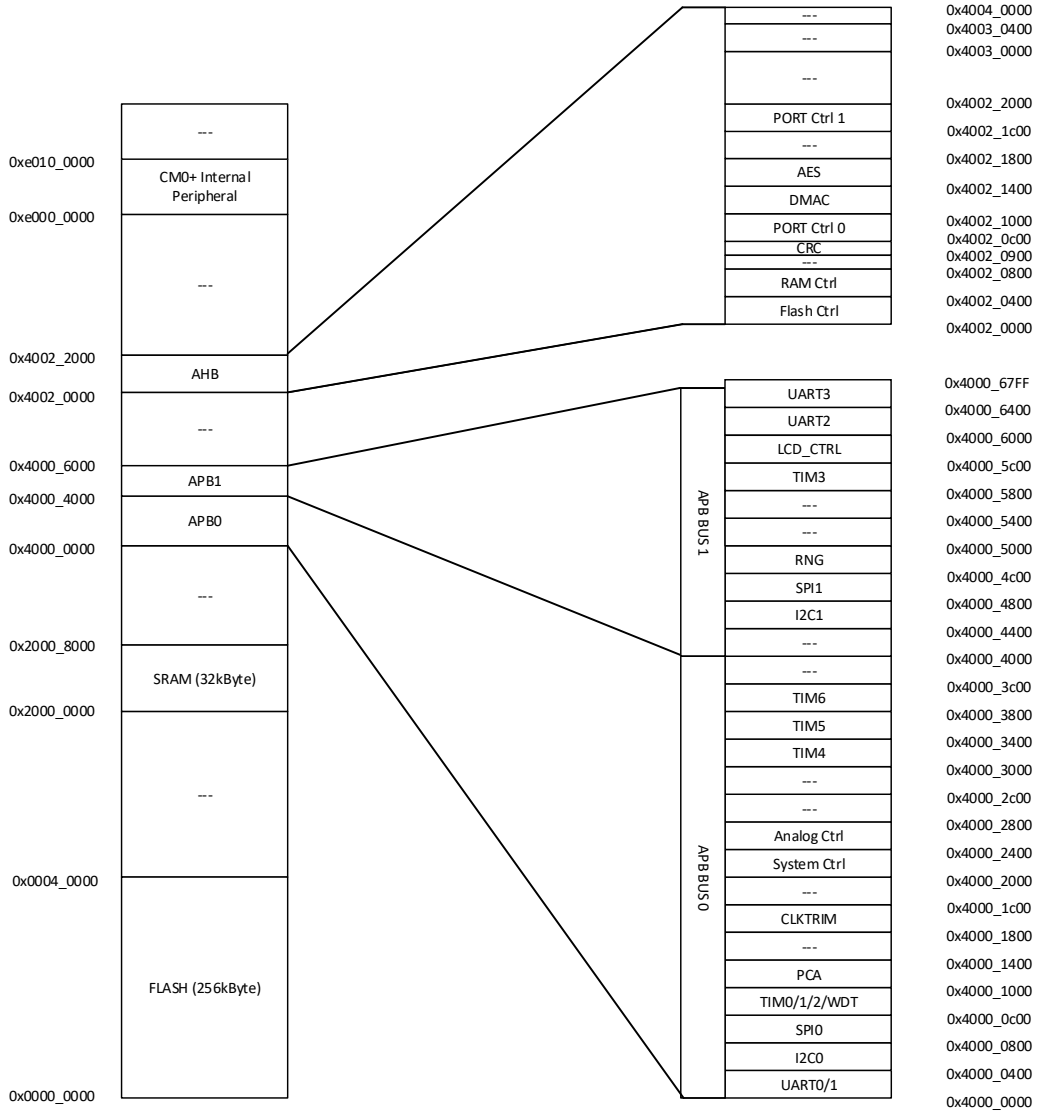
HC32F17 系列系统的地址区域划分，如下图所示：



注意：

1. port ctrl模块分了两段地址

HC32F19 系列系统的地址区域划分，如下图所示：



注意：

1. port ctrl模块分了两段地址

图 3-2　地址区域划分示意图

3.3 存储器和模块地址分配

Boundary Address	Size	Memory Area	Description
0x0000_0000 – 0x0001_FFFF	128KByte	FLASH	
0x0000_0000 – 0x0003_FFFF	256KByte		
0x0001_0000 – 0x1FFF_FFFF	-	Reserved	
0x2000_0000 – 0x2000_3FFF	16KByte	SRAM	
0x2000_0000 – 0x2000_7FFF	32KByte		
0x2000_2000 – 0x3FFF_FFFF	-	Reserved	
0x4000_0000 – 0x4000_00FF	256Byte	UART0	
0x4000_0100 – 0x4000_01FF	256Byte	UART1	
0x4000_0200 – 0x4000_02FF	256Byte	Reserved	
0x4000_0300 – 0x4000_03FF	-	Reserved	
0x4000_0400 – 0x4000_07FF	1KByte	I2C0	
0x4000_0800 – 0x4000_0BFF	1KByte	SPI0	
0x4000_0C00 – 0x4000_0CFF	256Byte	TIM0	
0x4000_0D00 – 0x4000_0DFF	256Byte	TIM1	
0x4000_0E00 – 0x4000_0EFF	256Byte	TIM2	
0x4000_0F00 – 0x4000_0F7F	128Byte	Reserved	
0x4000_0F80 – 0x4000_0FFF	128Byte	WDT	
0x4000_1000 – 0x4000_13FF	1KByte	PCA	
0x4000_1400 – 0x4000_17FF	1KByte	Reserved	
0x4000_1800 – 0x4000_1BFF	1KByte	CLKTRIM	
0x4000_1C00 – 0x4000_1FFF	-	Reserved	
0x4000_2000 – 0x4000_23FF	1KByte	SYSCTRL	
0x4000_2400 – 0x4000_27FF	1KByte	ANALOGCTRL	
0x4000_2800 – 0x4000_2BFF	1KByte	Reserved	
0x4000_2C00 – 0x4000_2FFF	1KByte	Reserved	
0x4000_3000 – 0x4000_33FF	1KByte	TIM4	
0x4000_3400 – 0x4000_37FF	1KByte	TIM5	
0x4000_3800 – 0x4000_3BFF	1KByte	TIM6	
0x4000_3C00– 0x4000_3FFF	-	Reserved	
0x4000_4000– 0x4000_43FF	1KByte	Reserved	
0x4000_4400– 0x4000_47FF	1KByte	I2C1	
0x4000_4800– 0x4000_4BFF	1KByte	SPI1	
0x4000_4C00– 0x4000_4FFF	1KByte	TRNG	
0x4000_5000– 0x4000_53FF	1KByte	Reserved	
0x4000_5400– 0x4000_57FF	1KByte	Reserved	
0x4000_5800– 0x4000_5BFF	1KByte	TIM3	

0x4000_5C00– 0x4000_5FFF	1KByte	LCD	
0x4000_6000 – 0x4000_63FF	1KByte	UART2	
0x4000_6400 – 0x4000_67FF	1KByte	UART3	
0x4000_6800 - 0x4001_FFFF	-	Reserved	
0x4002_0000 - 0x4002_03FF	1KByte	FLASH CTRL	
0x4002_0400 - 0x4002_07FF	1KByte	RAM CTRL	
0x4002_0800 - 0x4002_08FF	256Byte	Reserved	
0x4002_0900 - 0x4002_0BFF	768Byte	CRC	
0x4002_0C00 - 0x4002_0FFF	1KByte	PORT CTRL	
0x4002_1000 - 0x4002_13FF	1KByte	DMAC	
0x4002_1400 - 0x4002_17FF	1KByte	AES	
0x4002_1800 - 0x4002_1BFF	1KByte	Reserved	
0x4002_1C00 - 0x4002_1FFF	1KByte	PORT CTRL1	

表 3-1 地址划分表

4 工作模式

本产品的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压（VCC）为 1.8～5.5V。

本产品有如下几个工作模式：

1) 运行模式：CPU 运行，周边功能模块运行。

2) 休眠模式：CPU 停止运行，周边功能模块运行。

3) 深度休眠模式：CPU 停止运行，高速时钟停止运行。

从运行模式，通过执行软件程序，可进入其他低功耗模式。从其他各种低功耗模式，通过中断触发，可回到运行模式。

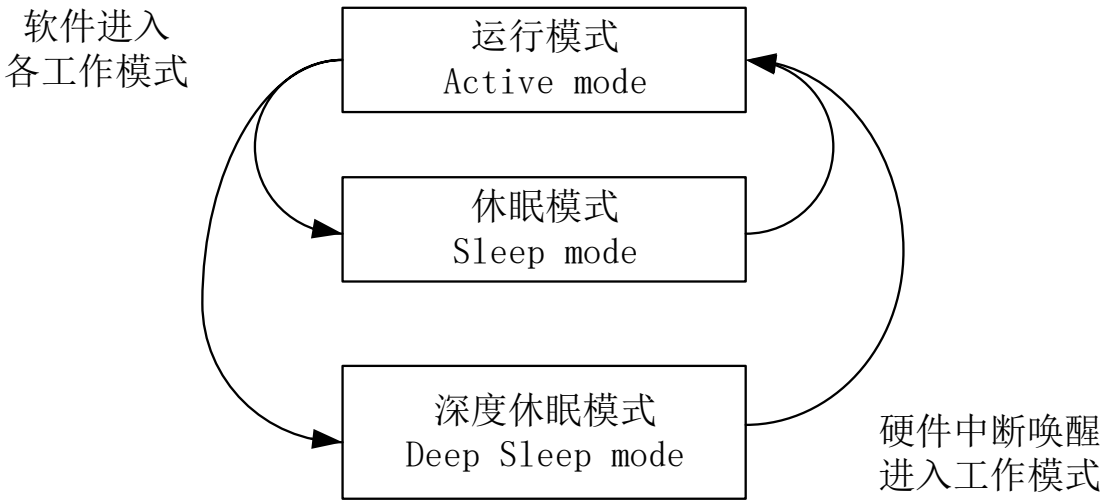


图 4-1 控制模式框图

在各模式下，CPU 可响应所有的中断类型。

中断向量编号	中断来源	运行模式	休眠模式	深度休眠模式
[0]	GPIO_PA	√	√	√
[1]	GPIO_PB	√	√	√
[2]	GPIO_PC/GPIO_PE	√	√	√
[3]	GPIO_PD/GPIO_PF	√	√	√
[4]	DMAC	√	√	
[5]	TIM3	√	√	
[6]	UART0	√	√	
[7]	UART1	√	√	
[8]	–	–	–	–

[9]	–	–	–	–
[10]	SPI0	√	√	
[11]	SPI1	√	√	
[12]	I2C0	√	√	
[13]	I2C1	√	√	
[14]	TIM0	√	√	
[15]	TIM1	√	√	
[16]	TIM2	√	√	
[17]	–	–	–	–
[18]	TIM4	√	√	
[19]	TIM5	√	√	
[20]	TIM6	√	√	
[21]	PCA	√	√	
[22]	WDT	√	√	√
[23]	–	–	–	–
[24]	ADC	√	√	
[25]	–	–	–	–
[26]	VC0	√	√	√
[27]	VC1/VC2	√	√	√
[28]	LVD	√	√	√
[29]	LCD	√	√	√
[30]	FLASH/RAM	√	√	
[31]	CLKTRIM	√	√	√

在各模式下，本产品可响应所有的复位类型。

	复位源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR	√	√	√
[1]	外部 Reset Pin 复位	√	√	√
[2]	LVD 复位	√	√	√
[3]	WDT 复位	√	√	√
[4]	PCA 复位	√	√	
[5]	Cortex-M0+ LOCKUP 硬件复位	√		
[6]	Cortex-M0+ SYSRESETREQ 软件复位	√		

4.1 运行模式

本产品运行模式（Active Mode）：

在系统在电源上电复位后，或从各低功耗唤醒后，微控制器 MCU 处于运行状态。当 CPU 不需继续运行时，可以利用多种低功耗模式来节能，例如等待某个外部事件时。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

运行模式（Active Mode）		
Cortex-M0+	SWD	XTH
FLASH	UART0-3	RCH
RAM	SPI0-1	PLL
DMAC	I2C0-1	ADC
TIM0-3	AES	DAC
TIM4-6	CRC	OPA
PCA	TRNG	
GPIO	XTL	
	RCL	LVD
	LCD	RESET
	CLKTRIM	POR/BOR
	VC0-2	WDT

表 4-1 运行模式下可运行模块图

几种降低运行模式下芯片功耗的方法：

1) 在运行模式下，通过对预分频寄存器（SYSCTRL0.HCLK_PRS, SYSCTRL0.PCLK_PRS）进行编程，可以降低任意一个系统时钟(HCLK，PCLK)的速度。进入休眠模式前，也可以利用预分频器来降低外设的时钟。
2) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗。
3) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗，并让系统进入休眠模式下更多地减少功耗，并在执行 WFI 指令前关闭不使用外设的时钟（PERIx_CLKEN.x）。

4) 使用低功耗模式代替休眠模式，因为本产品的唤醒时间极短（~4us），亦可满足系统的实时响应的需求。

4.2 休眠模式

本产品休眠模式（Sleep Mode）

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但时钟模块、系统时钟、NVIC 中断处理以及周边的功能模块仍都可以工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

- 如何进入休眠模式：

  通过执行 WFI 指令进入休眠状态。根据 Cortex-M0+系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择休眠模式进入机制：

  SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

  SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

- 如何退出休眠模式：

  如果执行 WFI 指令进入休眠模式，任意一个高优先级嵌套向量中断控制器响应的外设中断都能将系统从休眠模式唤醒。

使用注意：

1) SLEEP-ON-EXIT 该位置 1，执行完中断自动进入 sleep，程序不需要写 __wfi()；
2) SLEEP-ON-EXIT 该位清 0，main()执行 __wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 sleeping。等待后续中断触发。
3) SLEEP-ON-EXIT 位不影响 __wfi() 指令的执行。SLEEP-ON-EXIT=0：main() 执行 wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，继续往下执行；
4) 若在中断中进入 sleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。

休眠模式(Sleep Mode)		
Cortex-M0+	SWD	XTH
FLASH	UART0-3	RCH
RAM	SPI0-1	PLL
DMAC	I2C0-1	ADC
TIM0-3	AES	DAC
TIM4-6	CRC	OPA
PCA	TRNG	
GPIO	XTL	
	RCL	LVD
	LCD	RESET
	CLKTRIM	POR/BOR
	VC0-2	WDT

灰色的模块在当前状态下不工作。

表 4-2 休眠模式下可运行模块图

4.3 深度休眠模式

本产品深度休眠模式 （Deep Sleep Mode）

使用 SLEEPDEEP 配合 WFI 指令可以进入深度休眠模式，在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置是否允许，NVIC 中断处理仍可以工作。

- 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度休眠前的状态。
- 系统从低速时钟进入深度休眠模式，由于低速时钟不会自动关闭，保持运行，进入休眠模式。只有 ARM Cortex-M0+不运行，其他模块都运行。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- 系统进入深度休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

如何进入深度休眠模式：

首先设置 Cortex-M0+ 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入休眠状态。根据 Cortex™-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择深度休眠模式进入机制：

SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

如何退出深度休眠模式：

如果执行 WFI 指令进入休眠模式，任意一个被嵌套向量中断控制器响应的外设中断（Deep Sleep 下可运行的 周边模块中断）都能将系统从休眠模式唤醒。

唤醒设置参考 4.4 中断唤醒控制 WIC。

深度休眠模式（Deep Sleep Mode）		
Cortex-M0+	SWD	XTH
FLASH	UART0-3	RCH
RAM	SPI0-1	PLL
DMAC	I2C0-1	ADC
TIM0-3	AES	DAC
TIM4-6	CRC	OPA
PCA	TRNG	
GPIO	XTL	
	RCL	LVD
	LCD	RESET
	CLKTRIM	POR/BOR
	VC0-2	WDT

表 4-3 深度休眠模式下可运行模块图

灰色的模块在当前状态下不工作。

系统控制寄存器(Cortex-M0+ 内核系统控制寄存器)

地址: 0xE000ED10

复位值：0x0000 0000

位	标记	功能描述	读写
31:5	RESERVED	保留	
4	SEVONPEND	设置为1时，每次新的中断挂起都会产生一个事件，如果使用了WFE 休眠，它可用于唤醒处理器	RW
3	RESERVED	保留	
2	SLEEPDEEP	设为1时，执行 WFI 进入深度休眠，本产品进入 Deep sleep 模式 设置为0时，执行 WFI 进入休眠，本产品进入 sleep/Idle 模式	RW
1	SLEEPONEXIT	设为1时，当退出异常处理并返回程序线程时，处理器自动进入休眠模式(WFI) 设置为0时，该特性就会被自动禁止	RW
0	RESERVED	保留	

进入深度休眠后，唤醒后系统时钟有两种选择，默认使用进入深度休眠的时钟，配置寄存器 SYSCTRL0.wakeup_byRCH 为 1 后不管进入深度休眠前是什么时钟，唤醒后都使用内部高速时钟 RCH。如果使用外部晶体振荡这样设置可以加速唤醒系统。

5 系统控制器（SYSCTRL）

5.1 系统时钟介绍

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟。为了确保振荡器精度，内部时钟均具有校准功能。

本产品支持以下五个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 RCH（输出频率为 4~24MHz）
- 内部低速 RC 时钟 RCL（38.4K 与 32.8K 可配置）
- 外部高速晶振时钟 XTH
- 外部低速晶振时钟 XTL
- 锁相环时钟 PLL

注 1：切换系统时钟的时钟源时，请严格按照操作步骤进行切换，详见 5.2。

注 2：XTL 可以不接晶振，直接从 PC14 管脚输入 32.768KHz 的时钟信号。XTH 可以不接晶振，直接从 PF00 管脚输入 4~32MHz 的时钟信号。

本产品还包含以下两个辅助时钟：

- 内部低速 10K 时钟；仅供看门狗和 CLKTRIM 模块使用。
- 内部 150K 时钟：仅供 LVD 和 VC 模块使用。

下图为本产品的时钟架构。

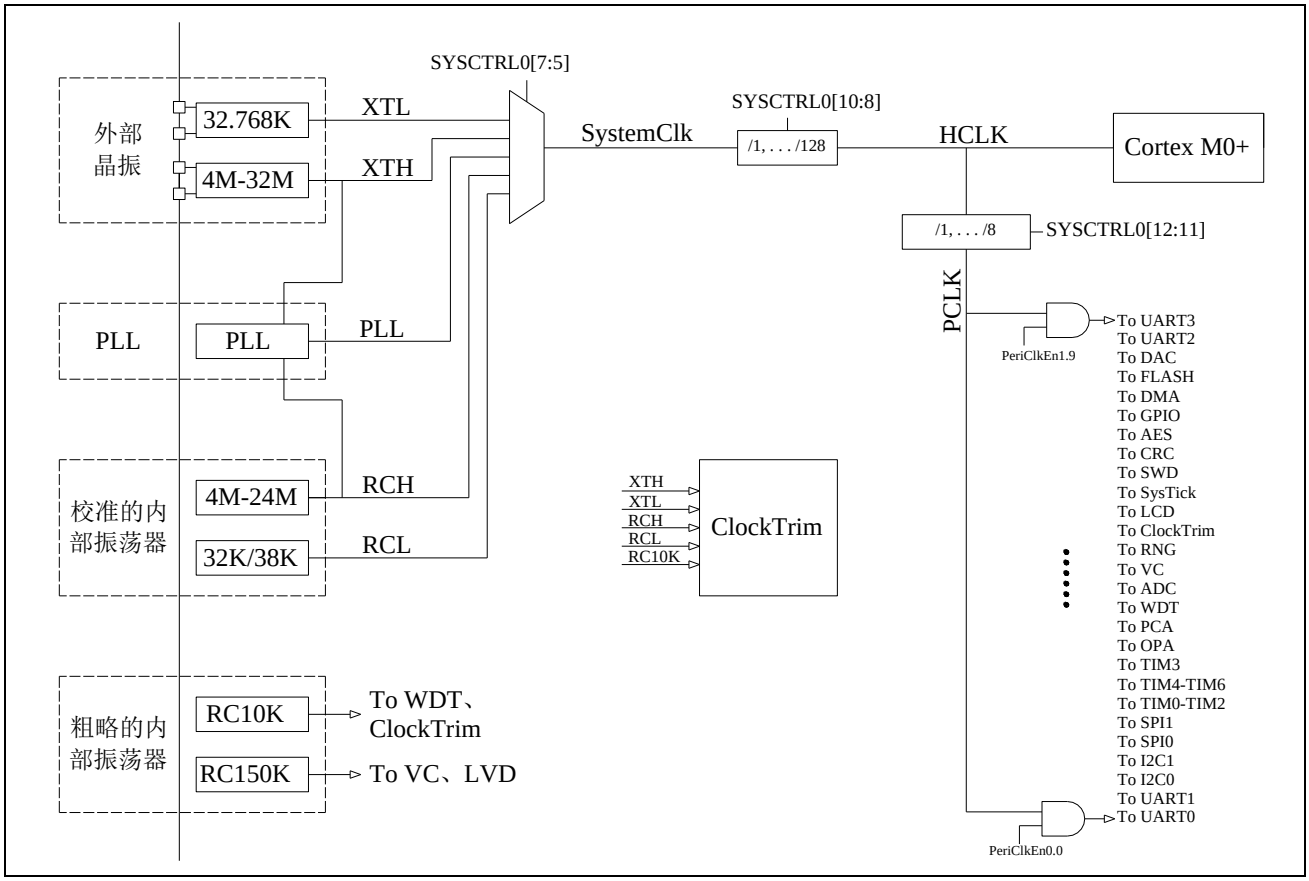


图 5-1 时钟控制模块框图

5.1.1 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为频率为 4MHz 的内部高速时钟；当系统进入 Deep Sleep，此高速时钟会自动关闭。

更改寄存器 RCH_CR[10:0]的数值即可调整 RCH 的输出频率。寄存器数值每增加 1 则 RCH 的输出频率增加约 0.2%，总调整范围为 4~24MHz。

出厂时已预调好的 5 个频率 4MHz、8MHz、16MHz、22.12MHz、24MHz；如需其它频率请手动调整该寄存器的数值。

更改 RCH 输出频率需要按照特定的更改时序，详见系统时钟切换章节。

内部高速时钟从启动到稳定仅需 4us。为了在深度休眠模式下能快速响应中断，建议进入深度休眠模式前将系统时钟切换为 RCH。

5.1.2 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 RCL_CR[9:0]来选择其输出频率，可供选择的频率为38.4KHz、32.768KHz。当系统进入 DeepSleep，此低速时钟不会自动关闭，部分外设模块可以选择 RCL 作为其时钟。

5.1.3 外部低速晶振时钟 XTL

外部低速晶振时钟需外接一个 32.768KHz 的低功耗晶振，具有超高精度。当系统进入 Deep Sleep，此低速时钟不会自动关闭。深度休眠模式下工作的外设模块可以选择 XTL 作为其时钟。

XTL 也可以不接晶振，直接从 PC14 管脚输入 32.768KHz 的时钟信号。从 PC14 输入时钟信号的方法为：配置 PC14 引脚为 GPIO 输入；设置 SYSCTRL1. EXTL_EN 为 1。

注意：

- 晶体及其匹配电容需符合电气参数中**低速外部时钟 XTL** 的相关要求。

5.1.4 外部高速晶振时钟 XTH

外部高速晶振时钟需外接一个 4~32MHz 的高速晶振。当系统进入 Deep Sleep，此高速时钟会自动关闭。

XTH 也可以不接晶振，直接从 PF00 管脚输入 4~32MHz 的时钟信号。从 PF00 输入时钟信号的方法为：配置 PF00 引脚为 GPIO 输入；设置 SYSCTRL1. EXTH_EN 为 1。

注意：

- 晶体及其匹配电容需符合电气参数中**高速外部时钟 XTH** 的相关要求。

5.1.5 锁相环时钟 PLL

内建 PLL 支持 8~48M 的时钟输出。PLL 的参考时钟源为：RCH、XTH 晶振时钟、PF00 管脚输入时钟。

5.1.6 时钟启动过程

上述五种时钟源都需要启动稳定时间，下图以外部 XTH 为例说明时钟的启动稳定过程。

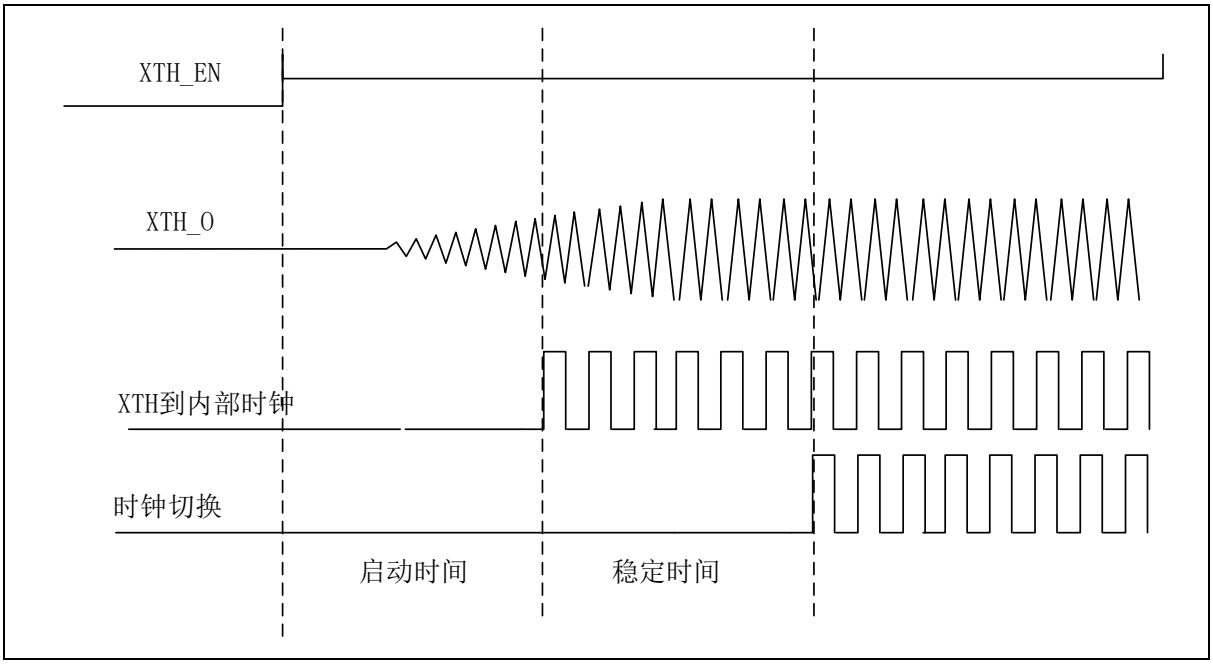


图 5-2　晶振时钟启动示意图

5.2 系统时钟切换

时钟源的切换是由寄存器 SYSCTRL0[7:0] 来控制。系统时钟的时钟源可通过 SYSCTRL0[7：5]在 RCH、RCL、XTH、XTL、PLL 之间进行切换。切换时 RCH、RCL、XTH、XTL 这四个时钟源中的任意两者可以相互切换；PLL 只可与 RCH、XTH 这两个时钟进行切换。时钟切换操作必须按照下文所描述的 7 种时钟切换流程进行，否则可能出现异常。

时钟切换时需要同步配置 FLASH_CR. WAIT。时钟频率不大于 24MHz 则应设置 FLASH_CR.WAIT 为 0；时钟频率大于 24MHz 则应设置 FLASH_CR. WAIT 为 1；时钟频率大于 48MHz 则应设置 FLASH_CR. WAIT 为 2。

注：设置 FLASH_CR. WAIT 的值，需要先向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5 然后再对 FLASH_CR. WAIT 进行赋值，详见 FLASH 控制器章节。

5.2.1 标准的时钟切换流程

操作流程如下：

Step1：如新时钟源需要外部引脚，则将该引脚设置为适当的模式。

注：接外部晶振时需要模拟引脚；接外部时钟输入时需要 GPIO 输入并使能外部时钟输入。

Step2：配置新时钟源的的振荡参数。

Step3：使能新时钟源的振荡器。

Step4：根据当前时钟源和新时钟源两者中较高的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step5：等待新时钟源输出稳定的频率。

Step6：配置 SYSCTRL0. CLKSW，选择系统时钟的来源为新时钟源。

Step7：根据新时钟源的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step8：关闭不再使用的时钟源。

5.2.2 RCH 不同振荡频率间切换流程

RCH 不同振荡频率间切换具有两种方案。

方案 1：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0.HCLK_PRS 为 0x7。

Step3：向上或向下逐级调整 RCH 的输出频率，4M -> 8M -> 16M -> 24M/22.12M 或 24M/22.12M -> 16M -> 8M -> 4M。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.HCLK_PRS 为 0x0。

从 4M 切换到 24M 的示例代码如下所示：

```
M0P_SystemCtrl->SYSCTRL2 = 0X5A5A;
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 7;
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C08 ) );   //4M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C06 ) );   //8M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C04 ) );   //16M
M0P_SystemCtrl->RCH_CR =   *((uint16 *)( 0X00100C00 ) );   //24M
M0P_SystemCtrl->SYSCTRL2 = 0X5A5A;
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 0
```

方案 2：

Step1：将系统时钟切换为 RCL，参见 5.2.5 从其它时钟切换到 RCL 示例。

Step2：将系统时钟切换为 RCH，参见 5.2.6 从其它时钟切换到 RCH 示例。

5.2.3 从其它时钟切换到 XTL 示例

操作流程如下：

Step1：设置 PCADS. 14 及 PCADS. 15 为 1，配置 PC14/PC15 引脚为模拟端口。

Step2：根据晶振特性，配置 XTL_CR[5:0]。

Step3：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step4：设置 SYSCTRL0. XTL_EN 为 1，使能晶振振荡电路。

Step5：查询等待 XTL_CR. Stable 标志变为 1，晶振输出稳定时钟。

Step6：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step7：设置 SYSCTRL0. CLKSW 为 3，将系统时钟切换为 XTL。

Step8：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step9：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

5.2.4 从其它时钟切换到 XTH 示例

Step1：设置 PFADS. 0 及 PFADS. 1 为 1，配置 PF00/PF01 引脚为模拟端口。

Step2：根据晶振特性，配置 XTH_CR[3:0]。

Step3：设置 XTH_CR. Startup 为 3，选择最长的晶振稳定时间。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0. XTH_EN 为 1，使能晶振振荡电路。

Step6：根据当前时钟和 XTH 两者中较高的频率，按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step7：查询等待 XTH_CR. Stable 标志变为 1 后，软件延时 10ms 以上，晶振输出稳定时钟。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0. CLKSW 为 1，将系统时钟切换为 XTH。

Step10：根据 XTH 的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step11：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

5.2.5 从其它时钟切换到 RCL 示例

操作流程如下：

Step1：配置 RCL_CR.TRIM 及 RCL_CR. Startup。

Step2：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：设置 SYSCTRL0. RCL_EN 为 1，使能 RCL 振荡电路。

Step4：查询等待 RCL_CR. Stable 标志变为 1，RCL 输出稳定时钟。

Step5：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 SYSCTRL0. CLKSW 为 2，将系统时钟切换为 RCL。

Step7：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

5.2.6 从其它时钟切换到 RCH 示例

操作流程如下：

Step1：配置 RCH_CR.TRIM。

Step2：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：设置 SYSCTRL0. RCH_EN 为 1，使能 RCH 振荡电路。

Step4：查询等待 RCH_CR. Stable 标志变为 1，RCH 输出稳定时钟。

Step5：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 SYSCTRL0. CLKSW 为 0，将系统时钟切换为 RCH。

Step7：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

5.2.7 PLL 与 RCH 相互切换示例，参考时钟为 RCH

从 RCH 切换到 PLL 操作流程如下：

Step1：设置 PLL_CR.REFSEL 为 3，选择 PLL 时钟源为 RCH。

Step2：根据 RCH 频率，配置 PLL_CR.FRSEL。

Step2：根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step3：设置 PLL_CR. Startup 为 7，选择最长的 PLL 稳定时间。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step6：根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step7：查询等待 PLL_CR. Stable 标志变为 1，PLL 输出稳定时钟。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0. CLKSW 为 4，将系统时钟切换为 PLL。

警告：切换过程中不可以更改 RCH_CR.TRIM。

从 PLL 切换到 RCH 操作流程如下：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0. CLKSW 为 0，将系统时钟切换为 RCH。

Step3：按 Flash 控制器章节流程设置 FLASH_CR. WAIT 为 0。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.PLL_EN 为 0，关闭 PLL。

警告：切换过程中不可以更改 RCH_CR.TRIM。

5.2.8 PLL 与 XTH 相互切换示例，参考时钟为 XTH

从 RCH 切换到 PLL 操作流程如下：

Step1：设置 PLL_CR.REFSEL 为 0，选择 PLL 时钟源为 XTH。

Step2：根据 XTH 频率，配置 PLL_CR.FRSEL。

Step2：根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step3：设置 PLL_CR. Startup 为 7，选择最长的 PLL 稳定时间。

Step4：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5：设置 SYSCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step6：根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step7：查询等待 PLL_CR. Stable 标志变为 1，PLL 输出稳定时钟。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0. CLKSW 为 4，将系统时钟切换为 PLL。

从 PLL 切换到 XTH 操作流程如下：

Step1：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：设置 SYSCTRL0. CLKSW 为 1，将系统时钟切换为 XTH。

Step7：根据 XTH 的频率，按 Flash 控制器章节流程配置 FLASH_CR. WAIT。

Step8：向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 SYSCTRL0.PLL_EN 为 0，关闭 PLL。

下图为时钟切换时序图：

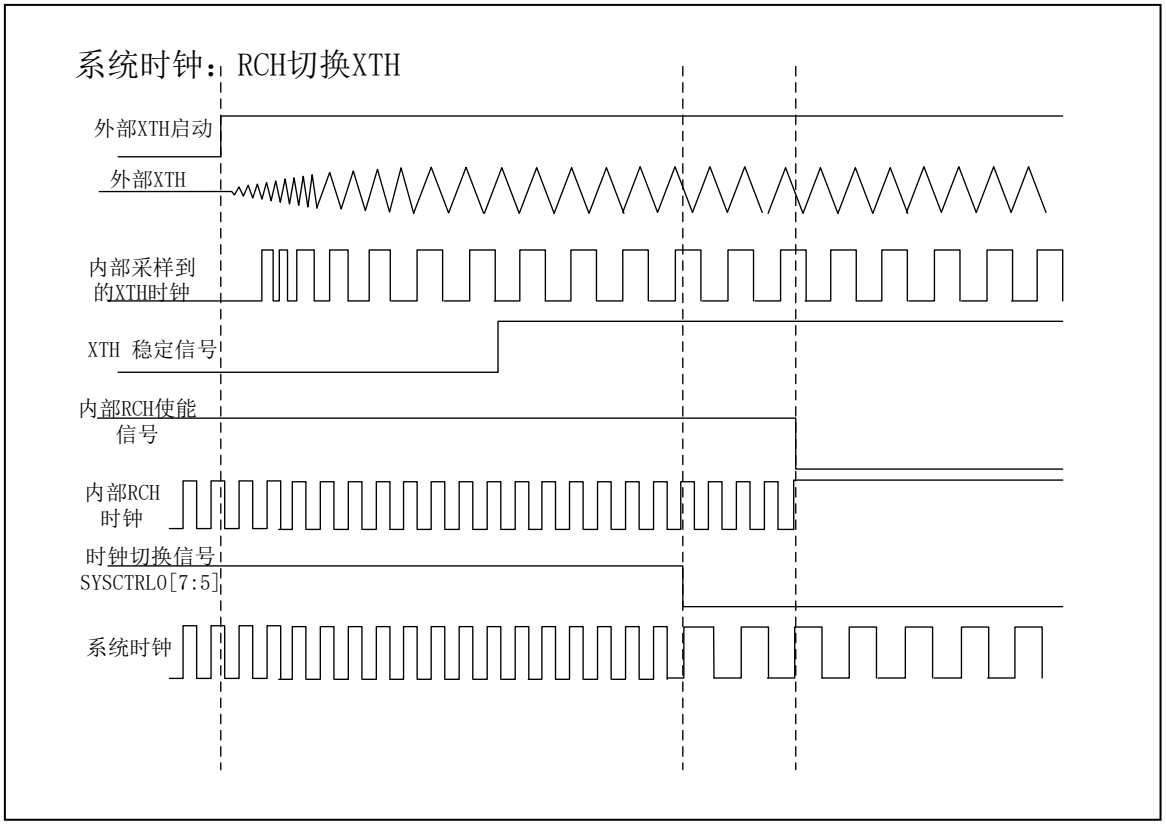


图 5-3 时钟切换示意图

5.3 时钟校准模块

本产品内嵌时钟校准电路，如下图所示，系统时钟的 5 个源均可以相互校准，当选择好参考时钟以及被校准时钟后，设置寄存器 REFCNT 值，置位 cali.start 启动时钟校准电路，此时两个 32 位计数器（递增、递减）同时工作，当递减计数器等于 0 时，cali.finish 被置位，表明校准结束，此时软件可以读取 CALCNT 值，这样就很容易得到参考时钟与被校准时钟之间的频率关系。

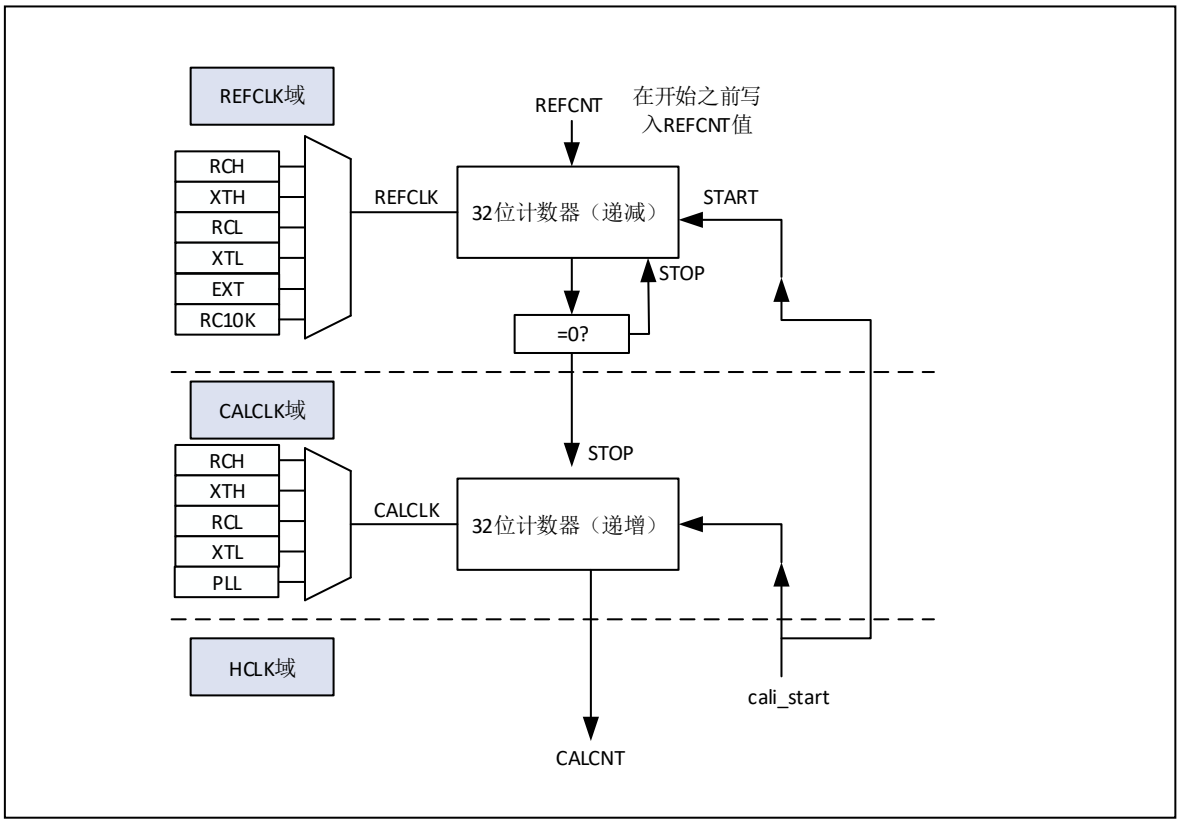


图 5-4 时钟校准原理图

5.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级>当前等级	Y	Y
0	IRQ 优先级≦当前等级	N	N
1	IRQ 优先级>当前等级	Y	N
1	IRQ 优先级≦当前等级	N	N

5.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 执行 WFI 指令以进入深度休眠模式
5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务程序

例程：

```
SCB->SCR |= 0x00000004u;
while(1)
 {
     __asm("WFI");
 }
```

5.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 PRIMASK 为 1
4. 设置 SCB->SCR.SLEEPDEEP 为 1

5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
7. 清除中断标志，清除中断挂起状态
8. 执行用户定义的操作

例程：

```
__asm("CPSID I");     //Set PRIMASK
SCB->SCR |= 0x00000004u;
while(1)
{
        __asm("WFI");
        M0P_TIMx->ICLR = 0x00;                  //Clear Timx Int Flag
        NVIC_ClearPendingIRQ( TIMx_IRQn ); //Clear Timx Pending Flag
        …     //执行用户定义的操作
}
```

5.4.3 使用退出休眠特性

退出休眠（sleep-on-exit）非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠模式。利用退出休眠特性，处理器可以尽可能多的处于休眠模式。

Cortex-M0 利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，为了下次进入异常时，不用再进行压栈操作，处理器不会执行出栈的过程。

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 设置 SCB->SCR.SLEEPONEXIT 为 1
5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序

7. 退出中断服务时自动进入休眠模式

例程：

```
SCB->SCR |= 0x00000004u;
SCB->SCR |= 0x00000002u;
while(1)
 {
        __asm("WFI");
  }
```

5.5 寄存器

基地址 0x40002000

寄存器	偏移地址	描述
SYSCTRL0	0x000	系统控制寄存器0
SYSCTRL1	0x004	系统控制寄存器1
SYSCTRL2	0x008	系统控制寄存器2
RCH_CR	0x00C	RCH 控制寄存器
XTH_CR	0x010	XTH 控制寄存器
RCL_CR	0x014	RCL 控制寄存器
XTL_CR	0x018	XTL 控制寄存器
PERI_CLKEN0	0x020	外围模块时钟控制寄存器0
PERI_CLKEN1	0x024	外围模块时钟控制寄存器1
PLL_CR	0x03C	PLL 控制寄存器

表 5-1 系统控制寄存器表格

5.5.1 系统控制寄存器 0（SYSCTRL0）

偏移地址：0x000

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wakeup_byRCH	Reserved		PCLK_PRS		HCLK_PRS			CLKSW			PLL_EN	XTL_EN	RCL_EN	XTH_EN	RCH_EN
RW			RW		RW			RW			RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	wakeup_byRCH	1：从 Deep Sleep 唤醒后，system clock来源为RCH，原时钟继续使能。 0：从 Deep Sleep 唤醒后，不改变system clock来源。
14:13	Reserved	保留
12:11	PCLK_PRS	PCLK时钟来源选择 00： HCLK 01： HCLK/2 10： HCLK/4 11： HCLK/8
10:8	HCLK_PRS	HCLK时钟来源选择 000： SystemClk 001： SystemClk/2 010： SystemClk/4 011： SystemClk/8 100： SystemClk/16 101： SystemClk/32 110： SystemClk/64 111： SystemClk/128
7:5	CLKSW	SystemClk时钟来源选择 000：内部高速时钟 RCH 001：外部高速晶振 XTH 010：内部低速时钟 RCL 011：外部低速晶振 XTL 100：内部 PLL
4	PLL_EN	PLL使能控制 0： 关闭 1： 使能

3	XTL_EN	外部低速晶振 XTL 使能控制 0： 关闭 1： 使能 注：需要将PC14、PC15设置成模拟端口。
2	RCL_EN	内部低速时钟 RCL 使能控制 0： 关闭 1： 使能
1	XTH_EN	外部高速晶振 XTH 使能控制 0： 关闭 1： 使能 注：当系统进入DeepSleep，此高速时钟会自动关闭
0	RCH_EN	内部高速时钟 RCH 使能信号。 0： 关闭 1： 使能 注：当系统进入DeepSleep，此高速时钟会自动关闭。

注意：

– 每次改写 SYSCTRL0，SYSCTRL1 的值，均需要先对 SYSCTRL2 依次写入 0x5A5、0xA5A5。这样的步骤可有效防止对 SYSCTRL0，SYSCTRL1 寄存器的误操作。

5.5.2 系统控制寄存器 1（SYSCTRL1）

偏移地址：0x004

复位值: 0x00000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							SWD_USE_IO	Res.	LOCKUP_EN	Res.		XTL_ALWAYS_ON	EXTL_EN	EXTH_EN	Res.
							RW		RW			RW	RW	RW	

位	标记	功能描述
31:9	Reserved	保留位
8	SWD_ USE_IO	SWD 端口功能配置 0：SWD 端口 1：GPIO 端口
7	Res.	保留位
6	LOCKUP_EN	Cortex-M0+ LockUp 功能配置 0：关闭 1：使能 注：使能后，CPU读到无效指令时会复位MCU，可增强系统可靠性。
5:4	Res.	保留位
3	XTL_ ALWAYS_ON	XTL 高级使能控制 1：SYSCTRL0.XTL_EN 只可置位。 0：SYSCTRL0.XTL_EN 可置位可清零。
2	EXTL_EN	外部 XTL 时钟输入控制 1：XTL 输出时钟从 PC14 输入。 0：XTL 输出时钟由晶振产生。 注：使用PC14 输入时钟时，需设置SYSCTRL0.XTL_EN为1。
1	EXTH_EN	外部 XTH 输入控制 1：XTH 输出时钟从 PF00输入。 0：XTH 输出时钟由晶振产生。 注：使用PF00 输入时钟时，需设置SYSCTRL0.XTH_EN为1。
0	Reserved	保留位

5.5.3 系统控制寄存器 2（SYSCTRL2）

偏移地址：0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCTRL2															
WO															

位	标记	功能描述
31:16	Reserved	保留位
15:0	SYSCTRL2	寄存器 SYSCTL0，SYSCTRL1 保护系列控制寄存器，对 SYSCTRL2 先写 0x5A5A，再写 0xA5A5，启动对于寄存器 SYSCTL0，SYSCTRL1 的写操作，只要对寄存器 SYSCTL0，SYSCTRL1 写操作了，这个保护位自动回复保护状态，需要重新写入系列打开保护。

5.5.4 RCH 控制寄存器（RCH_CR）

偏移地址：0x00C

复位值: 0x00000126

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Stable	TRIM										
				RO	RW										

位	标记	功能描述
31:12	Reserved	保留位
11	stable	RCH 时钟稳定标志位。 1：代表 RCH 已经稳定，可以被内部电路使用。 0：代表 RCH 未稳定，不可以被内部电路使用。
10:0	TRIM	时钟频率调整，更改该寄存器的数值即可调整RCH的输出频率。寄存器数值每增加1则RCH的输出频率增加约0.2%，总调整范围为4~24MHz。 Flash中已保存了5组频率的校准值，将Flash内的校准值读出并写入RCH_CR.TRIM即可获得精准的频率。 24M校准值地址：0x00100C00 - 0x00100C01 22.12M校准值地址：0x00100C02 - 0x00100C03 16M校准值地址：0x00100C04 - 0x00100C05 8M校准值地址：0x00100C06 - 0x00100C07 4M校准值地址：0x00100C08 - 0x00100C09

5.5.5 XTH 控制寄存器（XTH_CR）

偏移地址：0x010

复位值: 0x00000022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									Stable	Startup		xth_fsel		Driver	
									RO	RW		RW		RW	

位	标记	功能描述
31:7	Reserved	保留位
6	stable	外部高速时钟 XTH 稳定标志位。 1：代表 XTH 已经稳定，可以被内部电路使用。 0：代表 XTH 未稳定，不可以被内部电路使用。 注意：为了增加系统可靠性，在查询到该标志后，需要软件延时 10ms 以上，才可将系统时钟切换为 XTH。
5:4	Startup	外部高速时钟 XTH 稳定时间选择 00: 256个周期; 01: 1024个周期; 10: 4096个周期; 11: 16384个周期; 注：强烈建议将 XTH 的稳定时间设置为 11。如果 XTH 稳定时间不足，在进行时钟切换时或从深度休眠唤醒时，系统不能稳定工作。
3:2	xth_fsel	外部晶振工作频率选择 11: 24M~32M 10: 16M~24M 01: 8M~16M 00: 4M~8M
1:0	Driver	外部晶振驱动能力选择 11: 最强驱动能力 10: 默认驱动能力 (推荐值) 01: 弱驱动能力 00: 最弱驱动能力 注:需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。

5.5.6 RCL 控制寄存器（RCL_CR）

偏移地址：0x014

复位值: 0x0000033Fh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Stable	Startup		TRIM									
			RO	RW		RW									

位	标记	功能描述
31:13	Reserved	保留位
12	stable	内部低速时钟 RCL 稳定标志位。 1：代表 RCL 已经稳定，可以被内部电路使用。 0：代表 RCL 未稳定，不可以被内部电路使用。
11:10	Startup	内部低速时钟 RCL 稳定时间选择 11: 256个周期; 10: 64个周期; 01: 16个周期; 00: 4个周期;
9:0	TRIM	内部低速时钟频率调整，Flash中保存了2组频率的校准值。 将Flash内的校准值读出并写入RCL_CR.TRIM即可获得精准的频率。 38.4K校准值地址： 0x00100C20 - 0x00100C21 32.768K校准值地址：0x00100C22 - 0x00100C23

5.5.7 XTL 控制寄存器（XTL_CR）

偏移地址：0x018

复位值: 0x00000021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									Stable	Startup		Amp_sel		Driver	
									RO	RW		RW		RW	

位	标记	功能描述
31:7	Reserved	
6	stable	外部低速晶振 XTL 稳定标志位。 1：代表 XTL 已经稳定，可以被内部电路使用。 0：代表 XTL 未稳定，不可以被内部电路使用。
5:4	Startup	外部低速晶振 XTL 稳定时间选择 00: 256 个周期; 01: 1024 个周期; 10: 4096 个周期; 11: 16384 个周期;
3:2	amp_sel	XTL晶振振荡幅度的调整。 11: 最大振幅 10: 较大振幅（推荐值） 01: 正常振幅 00: 最小振幅
1:0	Driver	XTL晶振驱动能力选择 11: 最强驱动能力 10: 较强驱动能力 01: 一般驱动能力 （推荐值） 00: 最弱驱动能力 注:需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。

5.5.8 PLL 控制寄存器（PLL_CR）

偏移地址：0x03C

复位值: 0x000010B0F

31	30 2	28	27	26	25	24	23	22	21	20	19	18	17 16
Reserved												Stable	Startup
												RO	RW

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stableopt	FRSEL		LFSEL		IBSEL		DVIN			FOSC			REFSEL	
RW	RW		RW		RW		RW			RW			RW	

位	标记	功能描述
31:19	Reserved	保留
18	Stable	PLL 稳定标志 0: PLL未稳定 1:PLL已经稳定
17:15	Startup	PLL 稳定时间选择 000: 128个PLL周期 001: 256个PLL周期 010: 512个PLL周期 011: 1024个PLL周期 100: 2048个PLL周期 101: 4096个PLL周期 110: 8192个PLL周期 111: 16384个PLL周期
14:13	FRSEL	PLL 输入频率选择，根据PLL输入时钟的频率进行如下配置 00：4M~6M 01：6M~12M 10：12M~20M 11：20M~24M
12:11	LFSEL	PLL 滤波器控制位， 请保持默认值
10:9	IBSEL	PLL 偏置电流选择， 请保持默认值
8:5	DIVN	PLL 输出时钟的倍频系数, PLL输出频率与输入频率的倍数 输出频率 = DIVN*输入频率， DIVN 允许范围0X2~0XC
4:2	FOSC	PLL 输出频率范围选择，根据PLL输出时钟频率进行如下配置 000：8M~12M

		001：12M~18M 010：18M~24M 011：24M~36M 1xx：36M~48M
1:0	REFSEL	输入时钟选择 x0：XTH晶振生成的时钟 01：XTH从管脚PF00输入的时钟。（详见5.1.4） 11：RCH时钟

5.5.9 外围模块时钟控制寄存器 0（PERI_CLKEN0）

复位值: 0x8080_0000

偏移地址：0x020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLASH	Res.	DMA	GPIO	AES	CRC	SWD	TICK	Res.	LCD	Trim	Res.		TRNG	VC	ADC
RW		RW	RW	RW	RW	RW	RW		RW	RW			RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDT	PCA	OPA	Res.	TIM3	ADVTIM	Res.	BASETIM	SPI1	SPI0	I2C1	I2C0	Res.		UART1	UART0
RW	RW	RW		RW	RW		RW	RW	RW	RW	RW			RW	RW

位	标记	功能描述
31	FLASH	FLASH 控制器模块时钟使能。关闭后 FLASH 配置寄存器不可写，FLASH 中的程序仍然可以运行。 1：使能；0：关闭
30	Res.	保留位
29	DMA	DMAC 模块时钟使能。 1：使能；0：关闭
28	GPIO	GPIO 模块时钟使能。 1：使能；0：关闭
27	AES	AES 模块时钟使能。 1：使能；0：关闭
26	CRC	CRC 模块时钟使能。 1：使能；0：关闭
25	SWD	SWD 模块时钟使能。 1：使能；0：关闭
24	TICK	SYSTICK 控制寄存器时钟使能。 1：使能；0：关闭
23	Res.	保留位
22	LCD	LCD 模块时钟使能。 1：使能；0：关闭
21	TRIM	CLKTRIM 模块时钟使能。 1：使能；0：关闭
20:19	Res.	保留位
18	TRNG	TRNG 模块时钟使能。 1：使能；0：关闭
17	VC	VC、LVD，模块时钟使能。

		1：使能；0：关闭
16	ADC	ADC、BGR 模块时钟使能。 1：使能；0：关闭
15	WDT	WDT 模块时钟使能。 1：使能；0：关闭
14	PCA	PCA 模块时钟使能。 1：使能；0：关闭
13	OPA	OPA 模块时钟使能。 1：使能；0：关闭
12	Res.	保留位
11	TIM3	TIM3 模块时钟使能。 1：使能；0：关闭
10	ADVTIM	TIM456 模块时钟使能。 1：使能；0：关闭
9	Res.	保留位
8	BASETIM	TIM012 模块时钟使能。 1：使能；0：关闭
7	SPI1	SPI1 模块时钟使能。 1：使能；0：关闭
6	SPI0	SPI0 模块时钟使能。 1：使能；0：关闭
5	I2C1	I2C1 模块时钟使能。 1：使能；0：关闭
4	I2C0	I2C0 模块时钟使能。 1：使能；0：关闭
3:2	Res.	保留位
1	UART1	UART1 模块时钟使能。 1：使能；0：关闭
0	UART0	UART0 模块时钟使能。 1：使能；0：关闭

5.5.10 外围模块时钟控制寄存器 1（PERI_CLKEN1）

复位值: 0x0000_0000

偏移地址：0x024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						UART3	UART2	Reserved				DAC	Reserved		
						R/W	R/W					WO			

位	标记	功能描述
31:10	Reseved	保留位
9	UART3	UART3 模块时钟使能。 1：使能；0：关闭
8	UART2	UART2 模块时钟使能。 1：使能；0：关闭
7:4	Reseved	保留位
3	DAC	DAC 模块时钟使能。 1：使能；0：关闭 **此控制为不可读，读出值始终为0，使用读修改写其他控制位时DAC使能会被关闭**
2:0	Reseved	保留位

6 复位控制器（RESET）

6.1 复位控制器介绍

本产品具有 7 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序会从复位向量处开始执行。

- 数字区域上电掉电复位 POR
- 外部 Reset PAD，低电平为复位信号
- WDT 复位
- PCA 复位
- LVD 低电压复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示。复位标志均由硬件置位，需要用户软件清零。芯片复位时，如果查询到 Reset_flag. POR15V 或 Reset_flag. POR5V 为 1 则为**上电复位**。**上电复位**时用户程序应当将寄存器 Reset_flag 清零，则下一次复位时可通过 Reset_flag 的相关比特判断复位来源。

下图描述各区域的复位来源。

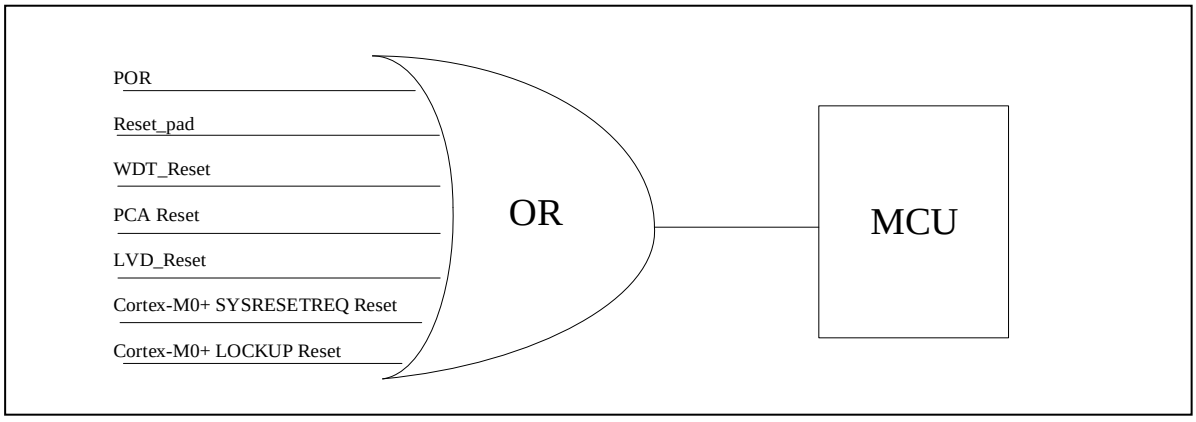


图 6-1 复位来源示意图

6.1.1 上电下电复位 POR

本产品有两个供电区域：VCC 区域、VCAP 区域。所有的模拟模块及 IO 工作于 VCC

区域；其它模块工作于 VCAP 区域。

VCC 区域上电时，当 VCC 电压低于 POR 阈值电压时（典型值为 1.65V），会产生 POR5V 信号；VCC 区域下电时，当 VCC 电压低于 BOR 阈值电压时（典型值为 1.5V），会产生 POR5V 信号。

VCAP 区域上电时，当 VCAP 电压低于 POR 阈值电压时，会产生 POR15V 信号；VCAP 区域下电时，当 VCAP 电压低于 BOR 阈值电压时，会产生 POR15V 信号。

POR5V 信号和 POR15V 信号均会将芯片的寄存器复位到初始化状态。

6.1.2 外部复位管脚复位

当外部复位引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20uS（典型值）的毛刺信号，因此，加到复位引脚上的低电平信号必须大于 20uS，才能确保芯片可靠复位。

6.1.3 WDT 复位

看门狗复位，请参看 WDT 一章说明。

6.1.4 PCA 复位

PCA 复位，请参看 PCA 一章说明。

6.1.5 LVD 低电压复位

LVD 复位，请参考 LVD 一章说明。

6.1.6 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位

6.1.7 Cortex-M0+ LOCKUP 复位

当 Cortex-M0+ 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个 CORE 区域。

6.2 寄存器

6.2.1 复位标识寄存器（RESET_FLAG）

复位值: 00000000_00000000_00000000_xxxxxx11b

地址：0x4000201C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RSTB	sysreq	lockup	PCA	WDT	LVD	Por15	Por5v
								RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0

位	标记	功能描述
31:8	Reserved	保留位
7	RSTB	RESETB 端口复位标志，需要软件初始化及清除，上电状态不定 1：发生端口复位 0：无端口复位发生 写0清除，写1无效
6	Sysreq	Cotrex-M0+ CPU 软件复位标志，需要软件初始化及清除，上电状态不定 1：发生 Cotrex-M0+ CPU 软件复位 0：无 Cotrex-M0+ CPU 软件复位发生 写0清除，写1无效
5	Lockup	Cotrex-M0+ CPU Lockup 复位标志，需要软件初始化及清除，上电状态不定 1：发生 Cotrex-M0+ CPU Lockup 复位 0：无 Cotrex-M0+ CPU Lockup 复位发生 写0清除，写1无效
4	PCA	PCA 复位标志，需要软件初始化及清除，上电状态不定 1：发生 PCA 复位 0：无 PCA 复位发生 写0清除，写1无效
3	WDT	WDT 复位标志，需要软件初始化及清除，上电状态不定 1：发生 WDT 复位 0：无 WDT 复位发生 写0清除，写1无效
2	LVD	LVD 复位标志，需要软件初始化及清除，上电状态不定 1：发生 LVD 复位 0：无 LVD 复位发生 写0清除，写1无效

1	POR15V	VCAP域复位标志 1：VCAP域发生复位 0：VCAP域无复位发生 写 0 清除，写 1 无效
0	POR5V	VCC电源域复位标志 1：VCC电源域发生复位 0：VCC电源域无复位发生 写0清除，写1无效

6.2.2 外围模块复位控制寄存器 0（PERI_RESET0）

复位值: 0x7F7F6FFF

地址：0x40002028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DMA	GPIO	AES	CRC	SWD	TICK	Res.	LCD	Trim	Res.		TRNG	VC	ADC
		RW	RW	RW	RW	RW	RW		RW	RW			RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	PCA	OPA	Res.	TIM3	ADV TIM	Res.	BASE TIM	SPI1	SPI0	I2C1	I2C0	Res.		UART 1	UART 0
	RW	RW		RW	RW		RW	RW	RW	RW	RW			RW	RW

位	标记	功能描述
31:30	Res.	保留位
29	DMA	DMAC 模块复位使能。 1：正常工作；0：模块处于复位状态
28	GPIO	GPIO 模块复位使能。 1：正常工作；0：模块处于复位状态
27	AES	AES 模块复位使能。 1：正常工作；0：模块处于复位状态
26	CRC	CRC 模块复位使能。 1：正常工作；0：模块处于复位状态
25	SWD	SWD 模块复位使能。 1：正常工作；0：模块处于复位状态
24	TICK	SYSTICK 模块复位使能。 1：正常工作；0：模块处于复位状态
23	Res.	保留位
22	LCD	LCD 模块复位使能。 1：正常工作；0：模块处于复位状态
21	TRIM	CLKTRIM 模块复位使能。 1：正常工作；0：模块处于复位状态
20:19	Res.	保留位
18	TRNG	TRNG 模块复位使能。 1：正常工作；0：模块处于复位状态
17	VC	VC，LVD，模块复位使能。 1：正常工作；0：模块处于复位状态
16	ADC	ADC 模块复位使能。 1：正常工作；0：模块处于复位状态

15	Res.	保留位
14	PCA	PCA 模块复位使能。 1：正常工作；0：模块处于复位状态
13	OPA	OPA 模块复位使能。 1：正常工作；0：模块处于复位状态
12	Res.	保留位
11	TIM3	TIM3 模块复位使能。 1：正常工作；0：模块处于复位状态
10	ADVTIM	TIM456 模块复位使能。 1：正常工作；0：模块处于复位状态
9	Res.	保留位
8	BASETIM	TIM012 模块复位使能。 1：正常工作；0：模块处于复位状态
7	SPI1	SPI1 模块复位使能。 1：正常工作；0：模块处于复位状态
6	SPI0	SPI0 模块复位使能。 1：正常工作；0：模块处于复位状态
5	I2C1	I2C1 模块复位使能。 1：正常工作；0：模块处于复位状态
4	I2C0	I2C0 模块复位使能。 1：正常工作；0：模块处于复位状态
3:2	Res.	保留位
1	UART1	UART1 模块复位使能。 1：正常工作；0：模块处于复位状态
0	UART0	UART0 模块复位使能。 1：正常工作；0：模块处于复位状态

6.2.3 外围模块复位控制寄存器 1（PERI_RESET1）

复位值: 0x0000_0318

地址：0x4000202C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						UART3	UART2	Reserved				DAC	Reserved		
						R/W	R/W					R0/W			

位	标记	功能描述
31:10	Reseved	保留位
9	UART3	UART3 模块复位使能。 1：正常工作；0：模块处于复位状态
8	UART2	UART2 模块复位使能。 1：正常工作；0：模块处于复位状态
7:4	Reseved	保留位
3	DAC	DAC 模块复位使能。 1：正常工作；0：模块处于复位状态 **此控制为不可读，读出值始终为0，使用读修改写其他控制位时DAC模块处于复位状态**
2:0	Reseved	保留位

7 中断控制器（NVIC）

7.1 概述

Cortex-M0+ 处理器内置了嵌套向量中断控制器（NVIC），支持最多 32 个中断请求（IRQ）输入，以及 1 个不可屏蔽中断（NMI）输入（在本产品系统中并未使用）。另外，处理器还支持多个内部异常。

每个异常源都有一个单独的异常编号，每种异常类型都有对应的优先级，有些异常的优先级是固定的，而有些则是可编程的。具体如下表所示：

异常编号	异常类型	优先级	描述
1	复位	-3（最高）	复位
2	NMI	-2	不可屏蔽中断（在本系统中没有使用）
3	硬件错误	-1	错误处理异常
4-10	保留	NA	…
11	SVC	可编程	通过SVC指令调用管理程序
12-13	保留	NA	…
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
…	…	…	…
47	中断#31	可编程	外部中断#31

表 7-1 Cortex-M0+ 处理器中断一览

本章节只对处理器的 32 个外部中断请求（中断#0 到中断#31）做详细介绍，处理器内部异常的具体情况可参考其他相关文档。同时，本章节只讨论处理器内核中 NVIC 的中断处理机制，外设模块自身的中断产生机制不在这里展开讨论。

7.2 中断优先级

每一个外部中断都对应一个优先级寄存器，每个优先级都是 2 位宽，并且使用中断优先级寄存器的最高两位，每个寄存器占 1 个字节（8 位）。在这个设定下，可以使用的优先级为 0x00（最高）、0x40、0x80 和 0xc0（最低）。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用		未使用，读出为 0					

图 7-1 只使用了高两位的优先级寄存器

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

如果处理器正在运行的另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断#0 和中断#1 使能且具有相同的优先级，在它们同时被触发时，中断#0 会首先执行。

7.3 中断向量表

当 Cortex-M0+处理器要处理中断服务请求时，它需要首先确定异常处理的起始地址，所需的信息叫做向量表，如图 7-2 所示。向量表存储在存储器空间的开始位置，包含了系统中可用异常（中断）的异常（中断）向量，以及主栈指针（MSP）的初始值。

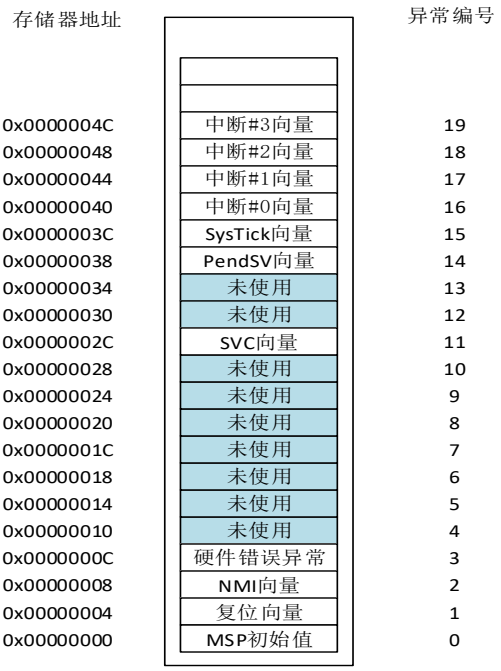


图 7-2 中断向量表

其中，中断向量的存储顺序同中断编号一致，由于每个向量都是 1 个字（4 字节），中断向量的地址为中断编号乘 4，每个中断向量都是中断处理的起始地址。

7.4 中断输入和挂起行为

Cortex-M0+ 处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。该过程如图 7-3 所示：

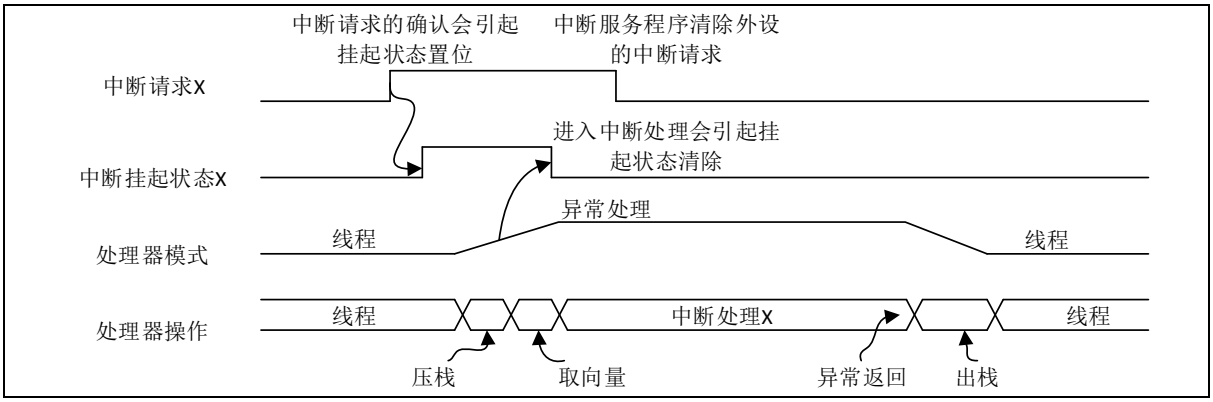


图 7-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 NVIC_CLRPEND 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。该过程如图 7-4 所示：

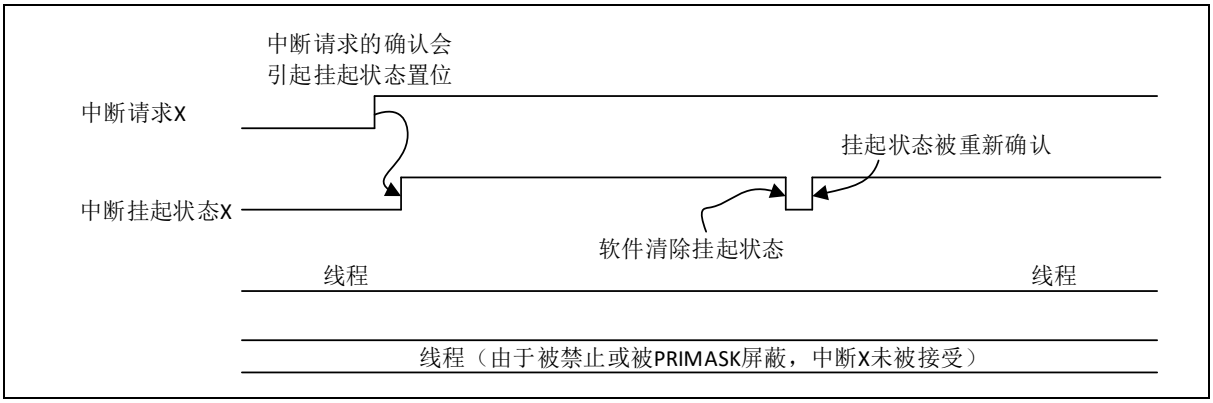


图 7-4 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。该过程如图 7-5 所示：

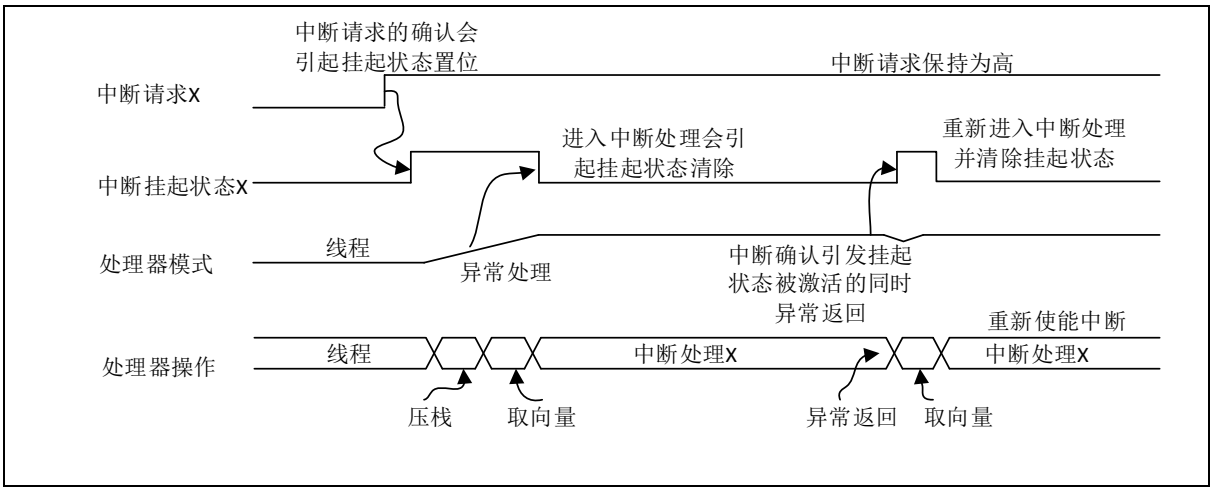


图 7-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在终端服务程序执行的过程中产生外设中断请求，该请求会被当作新的中断请求，并且在本次中断退出后，还会引起中断服务程序的再次执行。该过程如图 7-6 所示：

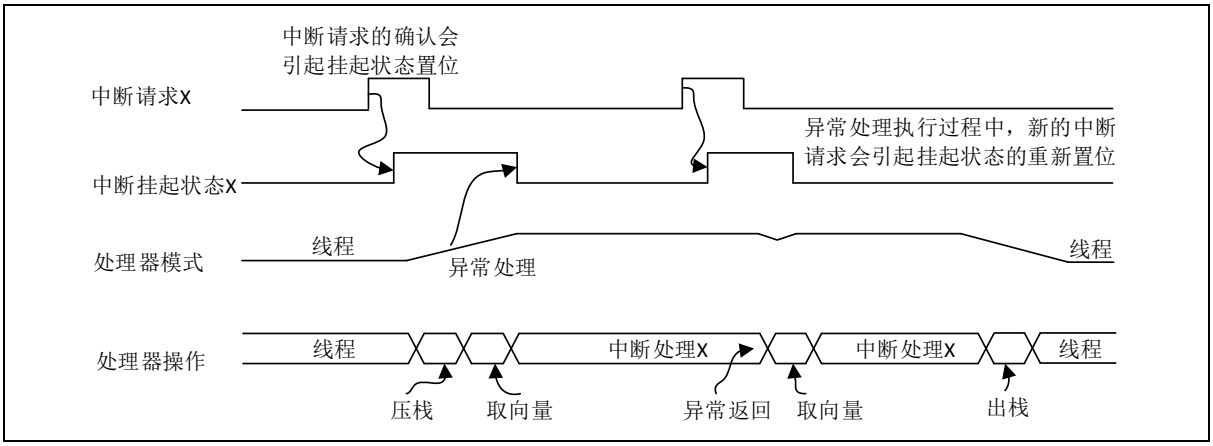


图 7-6 中断处理中产生的中断挂起也可以被确认

7.5 中断等待

通常情况下，NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理器时钟周期开始，一直到中断处理开始执行结束。计算中断等待需要具备以下前提：

- 该中断使能并且没有被 SCS_PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态，在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输，如果存储器系统需要等待，那么发生总线传输时产生的等待状态则可能使得中断延迟。

下面的几种情况可能会导致不同的中断等待：

- 中断的末尾连锁，如果中断返回时产生了另外一个中断请求，处理器就会跳过出栈和压栈过程，这样就减少了中断等待时间。
- 延迟到达，如果中断发生时，另外一个低优先级的中断正在进行压栈处理，由于延迟到达机制的存在，高优先级的中断会首先执行，这样也会导致高优先级中断的等待时间减小。

7.6 中断源

因为 Cortex-M0+ 处理器的 NVIC 支持最多 32 个外部中断，而在本系统中，外部中断源大于 32 个，所以部分外部中断复用在同一个 NVIC 中断输入上，其中 NMI（不可屏蔽中断）并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示：

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#0	PORTA	v	v	v
中断#1	PORTB	v	v	v
中断#2	PORTC/PORTE	v	v	v
中断#3	PORTD/PORTF	v	v	v
中断#4	DMAC	v	v	-
中断#5	TIM3	v	v	-
中断#6	UART0/2	v	v	-
中断#7	UART1/3	v	v	-
中断#8	-	-	-	-
中断#9	-	-	-	-
中断#10	SPI0	v	v	-
中断#11	SPI1	v	v	-
中断#12	I2C0	v	v	-
中断#13	I2C1	v	v	-
中断#14	TIM0	v	v	-
中断#15	TIM1	v	v	-
中断#16	TIM2	v	v	-
中断#17	-	-	-	-
中断#18	TIM4	v	v	-
中断#19	TIM5	v	v	-
中断#20	TIM6	v	v	-
中断#21	PCA	v	v	-
中断#22	WDT	v	v	v
中断#23	-	-	-	-
中断#24	ADC/DAC	v	v	-
中断#25	-	-	-	-
中断#26	VC0	v	v	v
中断#27	VC1/VC2	v	v	v
中断#28	LVD	v	v	v
中断#29	LCD	v	v	v
中断#30	FLASH/RAM	v	v	-

中断#31	CLKTRIM	v	v	v

表 7-2 外部中断与 NVIC 中断输入对应关系

注意：

– 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。

7.7 中断结构图

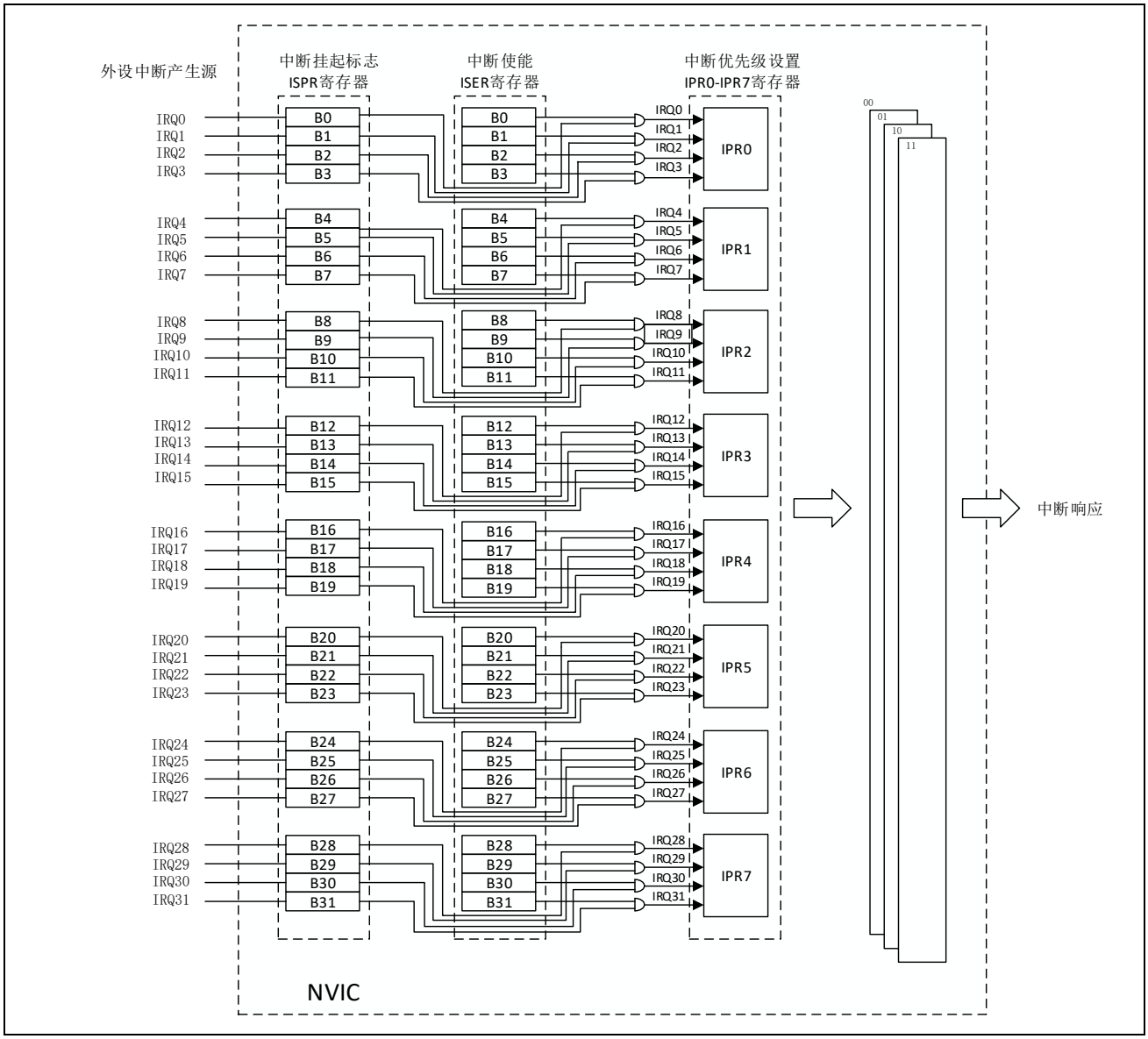


图 7-7 中断结构图

本系统的中断结构框图如图 7-7 所示。需要注意几点：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ 有多于 1 个外设中断输入复用，必须分别读取这些外设的中断标志位来判定是哪个外设的中断。
- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 NVIC_ISER 置位与否，中断挂起寄存器 SCS_SEPEND 都会被置位，表示相应的外设中断源有中断产生。
- 只有中断使能寄存器 NVIC_ISER 置位，相应的中断 IRQ 才会给处理器相应，执行

相应的中断程序。

- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 NVIC_ISPR 由硬件自动清除。
- 中断优先级寄存器 NVIC_IPR0- NVIC_IPR7 设置了 32 个中断源的优先级，00 优先级最高，11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

7.8 寄存器

基地址：0xE000 E000

寄存器	偏移地址	描述
NVIC_ISER	0x100	中断请求使能寄存器
NVIC_ICER	0x180	中断请求清除使能寄存器
NVIC_ISPR	0x200	中断设置挂起寄存器
NVIC_ICPR	0x280	中断清除挂起寄存器
NVIC_IPR0	0x400	中断#0-中断#3优先级寄存器
NVIC_IPR1	0x404	中断#4-中断#7优先级寄存器
NVIC_IPR2	0x408	中断#8-中断#11优先级寄存器
NVIC_IPR3	0x40C	中断#12-中断#15优先级寄存器
NVIC_IPR4	0x410	中断#16-中断#19优先级寄存器
NVIC_IPR5	0x414	中断#20-中断#23优先级寄存器
NVIC_IPR6	0x418	中断#24-中断#27优先级寄存器
NVIC_IPR7	0x41C	中断#28-中断#31优先级寄存器
PRIMASK	-	中断屏蔽特殊寄存器

7.8.1 中断使能设置寄存器（NVIC_ISER）

偏移地址：0x100

复位值：0x0000 0000

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

位	标记	功能描述
31:0	SETENA [31:0]	设置使能中断#0到中断#31；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 …… [31]:IRQ31

7.8.2 中断使能清除寄存器（NVIC_ICER）

偏移地址：0x180

复位值：0x0000 0000

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

位	标记	描述
31:0	CLRENA	清除使能中断#0到中断#31；写“1”清零，写“0”无效

7.8.3 中断挂起状态设置寄存器（NVIC_ISPR）

偏移地址：0x200

复位值：0x0000 0000

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

位	标记	功能描述
31:0	SETPEND	设置中断#0到中断#31的挂起状态；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 …… [31]:IRQ31

7.8.4 中断挂起状态清除寄存器（NVIC_ICPR）

偏移地址：0x280

复位值：0x0000 0000

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

位	标记	描述
31:0	CLRPEND	清除中断#0到中断#31的挂起状态；写“1”清零，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 …… [31]:IRQ31

7.8.5 中断优先级寄存器（NVIC_IPR0）

偏移地址：0x400

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]]															
RW															

位	标记	功能描述
31:0	IPR0[31:0]	中断#0到中断#3的优先级； [31:30]：中断#3 的优先级 [23:22]：中断#2 的优先级 [15:14]：中断#1 的优先级 [7:6]： 中断#0 的优先级 其中，00 优先级最高，11 优先级最低

7.8.6 中断优先级寄存器（NVIC_IPR1）

偏移地址：0x404

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]]															
RW															

位	标记	功能描述
31:0	IPR1[31:0]	中断#4到中断#7的优先级； [31:30]：中断#7 的优先级 [23:22]：中断#6 的优先级 [15:14]：中断#5 的优先级 [7:6]： 中断#4 的优先级 其中，00 优先级最高，11 优先级最低

7.8.7 中断优先级寄存器（NVIC_IPR2）

偏移地址：0x408

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]]															
RW															

位	标记	功能描述
31:0	IPR2[31:0]	中断#8到中断#11的优先级； [31:30]：中断#11 的优先级 [23:22]：中断#10 的优先级 [15:14]：中断#9 的优先级 [7:6]： 中断#8 的优先级 其中，00 优先级最高，11 优先级最低

7.8.8 中断优先级寄存器（NVIC_IPR3）

偏移地址：0x40C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]]															
RW															

位	标记	功能描述
31:0	IPR3[31:0]	中断#12到中断#15的优先级； [31:30]：中断#15 的优先级 [23:22]：中断#14 的优先级 [15:14]：中断#13 的优先级 [7:6]： 中断#12 的优先级 其中，00 优先级最高，11 优先级最低

7.8.9 中断优先级寄存器（NVIC_IPR4）

偏移地址：0x410

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR4[15:0]]															
RW															

位	标记	功能描述
31:0	IPR4[31:0]	中断#16到中断#19的优先级； [31:30]：中断#19 的优先级 [23:22]：中断#18 的优先级 [15:14]：中断#17 的优先级 [7:6]： 中断#16 的优先级 其中，00 优先级最高，11 优先级最低

7.8.10 中断优先级寄存器（NVIC_IPR5）

偏移地址：0x414

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]]															
RW															

位	标记	功能描述
31:0	IPR5[31:0]	中断#20到中断#23的优先级； [31:30]：中断#23 的优先级 [23:22]：中断#22 的优先级 [15:14]：中断#21 的优先级 [7:6]： 中断#20 的优先级 其中，00 优先级最高，11 优先级最低

7.8.11 中断优先级寄存器（NVIC_IPR6）

偏移地址：0x418

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]]															
RW															

位	标记	功能描述
31:0	IPR6[31:0]	中断#24到中断#27的优先级； [31:30]：中断#27 的优先级 [23:22]：中断#26 的优先级 [15:14]：中断#25 的优先级 [7:6]： 中断#24 的优先级 其中，00 优先级最高，11 优先级最低

7.8.12 中断优先级寄存器（NVIC_IPR7）

偏移地址：0x41C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]]															
RW															

位	标记	功能描述
31:0	IPR7[31:0]	中断#28到中断#31的优先级； [31:30]：中断#31 的优先级 [23:22]：中断#30 的优先级 [15:14]：中断#29 的优先级 [7:6]： 中断#28 的优先级 其中，00 优先级最高，11 优先级最低

7.8.13 中断屏蔽特殊寄存器（PRIMASK）

偏移地址：---

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

BIT	符号	描述
31:1	Reserved	
0	PRIMASK	置位后，除了 NMI 和硬件错误异常外的其他中断都会被屏蔽掉 清零后，所有异常和中断都不会被屏蔽掉 该特殊寄存器需要通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。在处理对时间敏感的应用时，需要操作 PRIMASK 寄存器。

7.9 软件基本操作

7.9.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考外设模块各自的章节描述。

7.9.2 NVIC 中断使能和清除使能

Cortex-M0+ 处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零使能位。这样就有了 32 位的中断使能寄存器 NVIC_ISER 和 32 位的清零使能寄存器 NVIC_ICER。如果想使能某一个中断，则对 NVIC_ISER 寄存器的相应位置 1。如果想清零某一个中断，则对 NVIC_ICER 寄存器的相应位置 1。

注意，这里提到的中断使能仅仅是针对处理器 NVIC 而言的，每个外设的中断生成与否，是由外设的中断控制寄存器决定，与 NVIC_ISER 和 NVIC_ICER 无关。

7.9.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持合法。

当处理器开始进入中断处理，硬件会自动引起挂起状态的清除。

可以通过操作中断设置挂起 NVIC_ISPR 和中断清除挂起 NVIC_ICPR 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

7.9.4 NVIC 中断优先级

设置 NVIC_IPR0- NVIC_IPR7 寄存器决定 SCS_IRQ0- SCS_IRQ32 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+ 处理器支持。

7.9.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 SCS_PRIMASK 实现。特殊寄存器 SCS_PRIMASK 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI（本系统不支持）和硬件错误异常处于使能。实际上，当 SCS_PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0（可和值的最高优先级）。

可以通过多种方法编程 SCS_PRIMASK 寄存器，使用汇编语言，可以利用 MS R 指令在设置和清除 SCS_PRIMASK 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 PRIMASK。

```
void __enable_irq(void);   //清除 PRIMASK
void __disable_irq(void);   //设置 PRIMASK
```

8 端口控制器（GPIO）

8.1 端口控制器简介

本系列有 88 个数字通用输入输出端口 PA[15:0], PB[15:0], PC[15:0], PD[7:0]。模拟模块 ADC/VC/LVD/LCD 的输入输出信号、各功能模块（如 SPI，UART，I2C，Timer 等）的输入输出信号可以和数字通用输入输出端口复用。

每个端口都可以配置成内部上拉（pull up）/下拉（pull down）的输入,高阻输入（floating input），推挽输出（CMOS output），开漏输出（open drain output），两档驱动能力输出。

数字端口被配置成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口输入值寄存器的结果为“0”。

每个数字端口被配置为输入时，都可以提供外部中断，中断类型可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发 4 种，查询 Px_STAT[n]的中断标志位即可知道相应的中断触发端口。另外，每个数字端口的中断都可以把芯片从休眠模式/深度休眠模式唤醒到工作模式。

芯片复位后端口为高阻输入（floating input），目的是防止芯片被异常复位时，对外部器件产生异常动作。但为了避免高阻输入而产生的漏电，用户要在芯片启动之后对端口进行相应的配置（配置成内部上拉/下拉输入或者输出）。

8.2 端口控制器主要特性

端口控制器支持以下特性：

- 端口输入值/输出值寄存器支持 FAST IO/AHB 总线读写
- 其他寄存器支持 AHB 总线接口读写
- 模拟功能引脚/数字通用引脚/数字功能引脚复用
- 支持上拉/下拉/两档驱动/开漏输出功能选择
- 支持工作模式/休眠模式/深度休眠模式下中断
- 支持高电平/低电平/上升沿/下降沿触发中断
- 支持位置位，位清零，位置位清零功能

注：FAST IO 介绍请参考 ARM Cortex-M0+_IntegrationAndImplementationManual.pdf。

8.3 端口控制器功能描述

8.3.1 端口配置功能

每一个端口都可以根据系统需求通过配置寄存器（**PxADS**）把端口配置成模拟端口或数字端口。当 PxADS 为‘1’时，端口配置成模拟端口，当 PxADS 为‘0’时，端口配置成数字端口。

端口电路结构如下图所示：

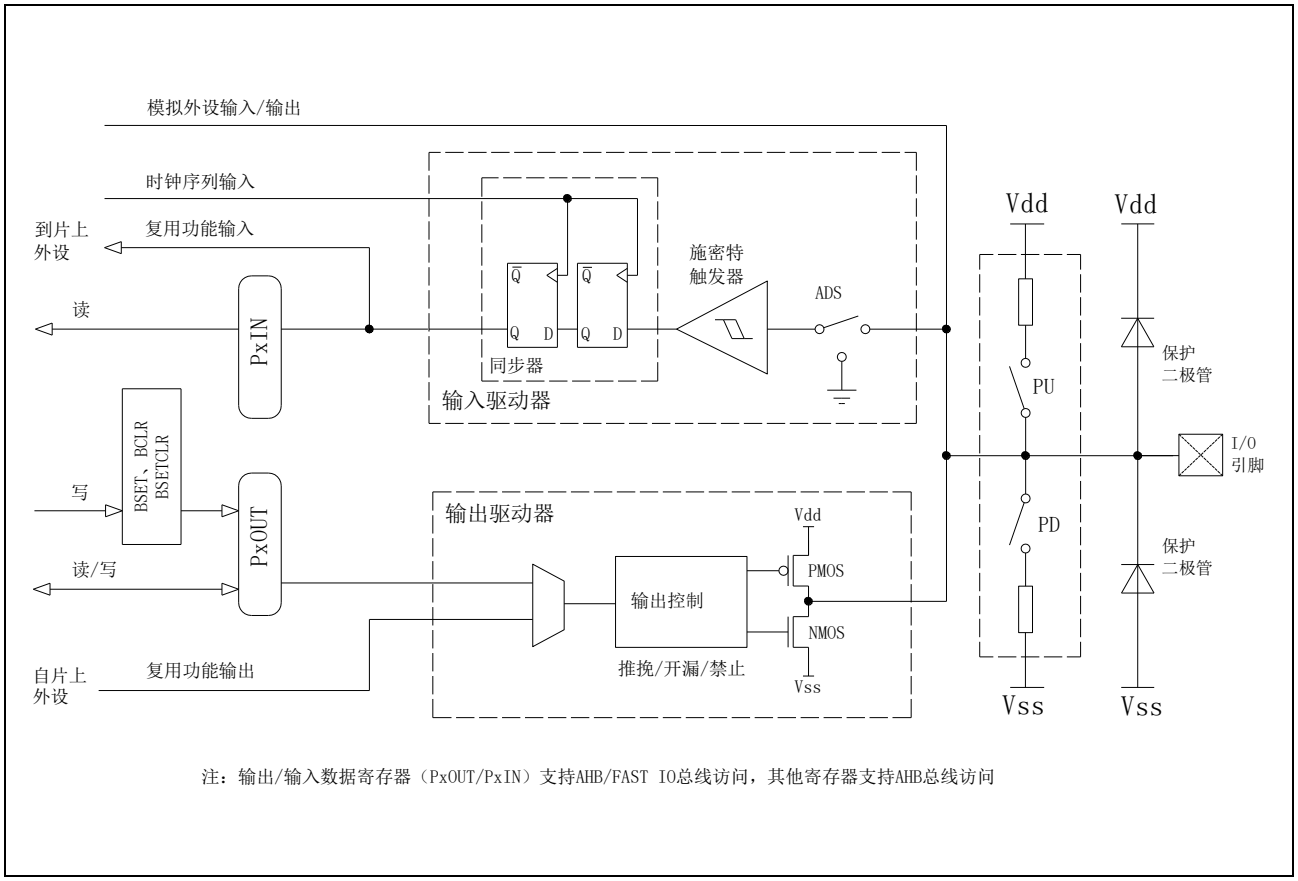


图 8-1 端口电路图

端口配置为数字端口时，可以接受数字通用输入输出信号（设置 Px_SEL 寄存器为‘0’），也可以通过配置寄存器 Px_SEL 接受各功能模块（如 SPI，UART，I2C，Timer 等）的输入输出信号，详见端口数字复用功能模块。

还可以配置相应的寄存器实现以下特性：

1） 内部上拉（PxPU）/下拉（PxPD）

上拉寄存器（PxPU）、下拉寄存器（PxPD）分别对应端口上拉使能和端口下拉使能，当对应位为‘1’时，设置对应位引脚上拉/下拉使能，为‘0’时，禁止对应位引脚上拉/下拉。

2） 两档驱动输出（PxDR）

能够通过 PxDR 寄存器改变驱动能力，PxDR 为‘1’时为低驱动能力，PxDR 为‘0’时为高驱动能力。

3） 开漏输出（PxOD）

通过 PxOD 寄存器设置管脚输出状态。当 PxOD 为‘1’时，端口开漏输出使能，为‘0’时，端口推挽输出使能。开漏引脚不连接外部的上拉电阻时，只能输出低电平，如果需要同时具备输出高电平的功能时，则需要上拉电阻。

4）方向选择（PxDIR）

用于设置端口引脚的方向。PxDIR 为‘0’的时候端口为输出，PxDIR 为‘1’的时候端口为输入。

5）输入电平状态 （PxIN）

通过读取 PxIN 寄存器能够获得同步后的管脚电平，PxIN 为‘1’时为高电平，PxIN 为‘0’时为低电平。可通过 AHB/FAST IO 总线访问。

6）输出高低电平选择（PxOUT）

当端口引脚配置为输出时，如果 PxOUT 为‘1’，端口引脚输出为高电平，如果 PxOUT 为‘0’，则输出低电平。可通过 AHB/FAST IO 总线访问。

7）位置位（PxBSET）、位清零（PxBCLR）、位置位清零（PxBSETCLR）

位置位与位清零适用于在不改变其他位的值的情况下，对用户想要改变的位进行设置，得到相应的值。位置位是将对应的值设置为‘1’，位清零是将对应的值设置为‘0’。

注意：

- 以上特性在配置为模拟端口时无效。

端口的状态和寄存器配置的关系如下表：

IO 状态	IO 方向	PxADS	PxDIR	PxOUT	PxIN	PxBSET	PxBCLR	PxBSETCLR	PxPU	PxPD	PxOD	PxDR	Px_SEL
模拟	输入/输出	1	W	W	0	W	W	W	W	W	W	W	W
浮空	输入	0	1	W	X	W	W	W	0	0	W	W	0
下拉	输入	0	1	W	0	W	W	W	0	1	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	0	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	1	W	W	0
1	输入	0	1	W	1	W	W	W	W	W	W	W	0
0	输入	0	1	W	0	W	W	W	W	W	W	W	0
1	输出	0	0	1	1	0	0	0	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	0	W	0
1	输出	0	0	W	1	1	0	0	W	W	0	W	0
0	输出	0	0	W	0	0	1	0	W	W	0	W	0
（SET）1	输出	0	0	W	（SET）1	0	0	1	W	W	0	W	0
（CLR）0	输出	0	0	W	（CLR）0	0	0	1	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	1	W	0
Z	输出	0	0	1	X	0	0	0	0	0	1	W	0
0	输出	0	0	1	0	0	0	0	0	1	1	W	0
1	输出	0	0	1	1	0	0	0	1	0	1	W	0
1	输出	0	0	1	1	0	0	0	1	1	1	W	0
注：	0 - Logic low		1 - Logic high		W - Whatever 0 or 1		X - unknow state		Z - high impedance				

表 8-1 端口状态真值表

8.3.2 端口的写入

端口输入值/输出值寄存器（PxIN/PxOUT）支持 AHB 总线和 FAST IO 总线的读写（通过寄存器 GPIO_CTRL2.ahb_sel 位控制），对于其他的寄存器则只支持 AHB 总线的读写。对于这两种不同的总线，系统时钟（HCLK）对这两种总线的处理周期并不相同。下面两幅图为对于两种总线端口翻转的最快时序：

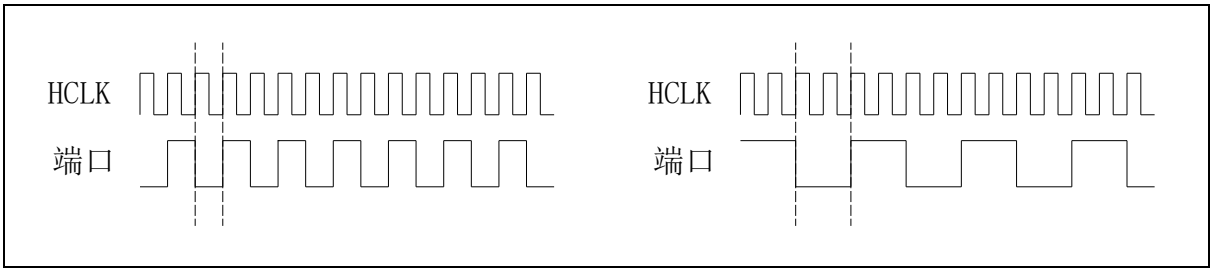


图 8-2 AHB/FASTIO 总线端口随系统时钟的变化
（左为 FAST IO，右为 AHB）

对于 AHB 总线，每两个 HCLK 周期，IO 翻转一次，而对于 FAST IO 总线，每个 HCLK 周期，IO 翻转一次。

8.3.3 端口的读取

每一个端口可以通过读取 PxIN 寄存器来获得端口引脚电平。如图 8-1 所示，PxIN 寄存器的各个位与其前面的锁存器组成了一个同步器从而避免了系统时钟状态发生改变时在短时间内引脚电平变化而造成的信号不稳定，但是同时也引入了延迟。读取端口引脚数据的同步图如下：

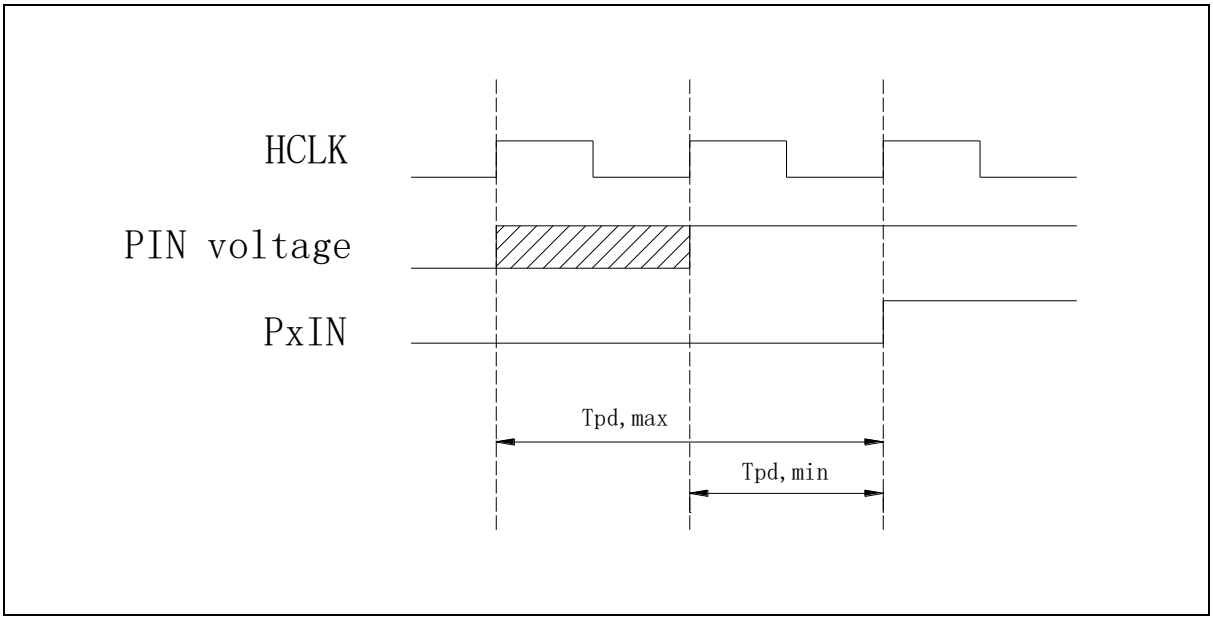


图 8-3 读取端口引脚数据同步图

在系统时钟上升沿之后的时钟时期，管脚电平信号会锁存在内部寄存器，如阴影部分所示，在下一次系统时钟上升沿之后，稳定的管脚电平信号能够被读取。之后再系统时钟上升沿时，数据锁存到 PxIN 寄存器中。信号装换延迟 Tpd 为 1-2 个系统时钟。

注意：

- 未连接引脚的处理：

  在使用过程中如果有未接引脚，为避免在其他数字输入使能模式时引脚没有确定的电平造成悬空消耗电流，推荐赋予引脚一个确定的电平，但未用引脚与 Vcc 或 GND 连接时，可能会在引脚偶然作为输出时出现冲击电流。

8.3.4 端口数字复用功能

端口数字复用是端口控制器的主要功能之一。通过配置寄存器，可以灵活地把端口配置成数字通用端口/数字功能端口。

如下表所示：Px_SEL 寄存器用于数字通用端口/数字功能端口切换，每一个端口都可以独立配置成系统所需的功能端口。

Px_SEL							
0	1	2	3	4	5	6	7
PA00	UART1_CTS		TIM0_ETR	VC0_OUT	TIM1_CHA	TIM3_ETR	TIM0_CHA
PA01	UART1_RTS		TIM0_CHB	TIM1_ETR	TIM1_CHB	HCLK_OUT	SPI1_MOSI
PA02	UART1_TXD	TIM0_CHA	VC1_OUT	TIM1_CHA	TIM2_CHA	PCLK_OUT	SPI1_MISO
PA03	UART1_RXD	TIM0_GATE	TIM1_CHB	TIM2_CHB	SPI1_CS	TIM3_CH1A	TIM5_CHA
PA04	SPI0_CS	UART1_TXD	PCA_CH4	TIM2_ETR	TIM5_CHA	LVD_OUT	TIM3_CH2B
PA05	SPI0_SCK	TIM0_ETR	PCA_ECI	TIM0_CHA	TIM5_CHB	XTL_OUT	XTH_OUT
PA06	SPI0_MISO	PCA_CH0	TIM3_BK	TIM1_CHA	VC0_OUT	TIM3_GATE	
PA07	SPI0_MOSI	PCA_CH1	HCLK_OUT	TIM3_CH0B	TIM2_CHA	VC1_OUT	TIM4_CHB
PA08	UART0_TXD	TIM3_CH0A			TIM1_GATE	TIM4_CHA	TIM3_BK
PA09	UART0_TXD	TIM3_CH1A	TIM0_BK	I2C0_SCL		HCLK_OUT	TIM5_CHA
PA10	UART0_RXD	TIM3_CH2A	TIM2_BK	I2C0_SDA	TIM2_GATE	PCLK_OUT	TIM6_CHA
PA11	UART0_CTS	TIM3_GATE	I2C1_SCL		VC0_OUT	SPI0_MISO	TIM4_CHB
PA12	UART0_RTS	TIM3_ETR	I2C1_SDA		VC1_OUT	SPI0_MOSI	
PA13	IR_OUT	UART0_RXD	LVD_OUT	TIM3_ETR			VC2_OUT
PA14	UART1_TXD	UART0_TXD	TIM3_CH2A	LVD_OUT	RCH_OUT	RCL_OUT	PLL_OUT
PA15	SPI0_CS	UART1_RXD		TIM0_ETR	TIM0_CHA	TIM3_CH1A	
PB00	PCA_CH2	TIM3_CH1B		TIM5_CHB	RCH_OUT	RCL_OUT	PLL_OUT
PB01	PCA_CH3	PCLK_OUT	TIM3_CH2B	TIM6_CHB		VC2_OUT	TCLK_OUT
PB02		PCA_ECI		TIM4_CHA	TIM1_BK	TIM0_BK	TIM2_BK
PB03	SPI0_SCK	TIM0_CHB	TIM1_GATE	TIM3_CH0A		XTL_OUT	XTH_OUT
PB04	SPI0_MISO	PCA_CH0	TIM2_BK	UART0_CTS	TIM2_GATE	TIM3_CH0B	
PB05	SPI0_MOSI		TIM1_BK	PCA_CH1			UART0_RTS
PB06	I2C0_SCL	UART0_TXD	TIM1_CHB	TIM0_CHA		TIM3_CH0A	
PB07	I2C0_SDA	UART0_RXD	TIM2_CHB		TIM0_CHB		
PB08	I2C0_SCL	TIM1_CHA		TIM2_CHA	TIM0_GATE	TIM3_CH2A	UART0_TXD
PB09	I2C0_SDA	IR_OUT	SPI1_CS	TIM2_CHA		TIM2_CHB	UART0_RXD
PB10	I2C1_SCL	SPI1_SCK	TIM1_CHA		TIM3_CH1A		UART1_RTS
PB11	I2C1_SDA	TIM1_CHB		TIM2_GATE	TIM6_CHA		UART1_CTS
PB12	SPI1_CS	TIM3_BK		TIM0_BK			TIM6_CHA
PB13	SPI1_SCK	I2C1_SCL	TIM3_CH0B		TIM1_CHA	TIM1_GATE	TIM6_CHB
PB14	SPI1_MISO	I2C1_SDA	TIM3_CH1B	TIM0_CHA			TIM1_BK

PB15	SPI1_MOSI	TIM3_CH2B	TIM0_CHB	TIM0_GATE			
PC00			UART1_CTS	UART2_RTS			
PC01		TIM5_CHB	UART1_RTS			UART2_CTS	
PC02	SPI1_MISO			UART2_RXD			
PC03	SPI1_MOSI				UART2_TXD		
PC04		TIM2_ETR	IR_OUT	VC2_OUT			
PC05		TIM6_CHB	PCA_CH4				
PC06	PCA_CH0	TIM4_CHA	TIM2_CHA			UART3_RXD	
PC07	PCA_CH1	TIM5_CHA	TIM2_CHB			UART3_TXD	
PC08	PCA_CH2	TIM6_CHA	TIM2_ETR			UART3_CTS	
PC09	PCA_CH3	TIM4_CHB	TIM1_ETR			UART3_RTS	
PC10			PCA_CH2				
PC11			PCA_CH3				
PC12			PCA_CH4				
PC13			TIM3_CH1B				
PC14							
PC15							
PD00		SPI1_CS					
PD01		SPI1_SCK					
PD02	PCA_ECI		TIM1_ETR				
PD03	UART1_CTS	SPI1_MISO					
PD04	UART1_RTS	SPI1_MOSI					
PD05	UART1_TXD						
PD06	UART1_RXD						
PD07	UART1_TXD						
PD08							
PD09							
PD10							
PD11							
PD12		UART2_RTS					
PD13	UART2_RXD						
PD14	UART2_TXD						
PD15		UART2_CTS					
PE00	TIM1_CHA						
PE01	TIM2_CHA						
PE02	PCA_ECI						
PE03	PCA_CH0						
PE04	PCA_CH1						
PE05	PCA_CH2						
PE06	PCA_CH3						
PE07	TIM3_ETR						

PE08	TIM3_CH0B						
PE09	TIM3_CH0A						
PE10	TIM3_CH1B						
PE11	TIM3_CH1A						
PE12	TIM3_CH2B	SPI0_CS	UART3_CTS				
PE13	TIM3_CH2A	SPI0_SCK	UART3_RTS				
PE14	TIM3_CH0B	SPI0_MISO	UART3_RXD				
PE15	TIM3_BK	SPI0_MOSI	UART3_TXD				
PF00	I2C0_SDA		UART1_TXD				
PF01	I2C0_SCL	TIM4_CHB	UART1_RXD				
PF02							
PF03							
PF04							
PF05							
PF06	I2C1_SCL		UART0_CTS				
PF07	I2C1_SDA		UART0_RTS				
PF09	TIM0_CHA						
PF10	TIM0_CHB						
PF11							

表 8-2 端口复用表

8.3.5 端口中断功能

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，分别对应的中断使能寄存器为高电平中断使能寄存器（PxHIE））/低电平中断使能寄存器（PxLIE）/上升沿中断使能寄存器（PxRIE）/下降沿中断使能寄存器（PxHIE）。当中断触发时，通过查询中断状态寄存器（Px_STAT）就可以判断是哪一个端口触发了中断，通过清零中断清除寄存器（Px_ICLR）就可以清除对应的中断状态标志位。

8.4 端口配置操作流程

8.4.1 端口复用配置为模拟端口操作流程

a) 设置寄存器 PxADS[n]为 1

8.4.2 端口复用配置为数字通用端口操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1：端口方向为输入，CPU 可以读取端口的状态 PxIN[n]。

d) 设置寄存器 PxDIR[n]为 0：端口方向为输出

e) 设置寄存器 PxOUT[n]为 1：端口输出高电平

f) 设置寄存器 PxOUT[n]为 0：端口输出低电平

8.4.3 端口复用配置为数字功能端口操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 1~7（根据系统需求，参考端口复用表）

c) 设置寄存器 PxDIR[n]（根据系统需求）

d) 设置寄存器 PxPU[n]/PxPD[n]/PxOD[n]（根据系统需求）

8.4.4 端口复用配置为调试测试端口操作流程

参考 SWD 调试接口。

8.4.5 端口复用配置为红外输出信号操作流程

端口 PA13,PB09,PC04 可以把频率为 38K 的内部时钟信号调制成红外信号输出。

a) 设置寄存器 PAADS[13]/PBADS[9]/PCADS[4]为 0

b) 设置寄存器 PA13_SEL = 1/PB09_SEL = 2/PC04_SEL = 3

c) 设置寄存器 PADIR[13] = 0/PBDIR[9] = 0/PCDIR[4] = 0：端口方向为输出

d) 设置寄存器 GPIO_CTRL1 的 bit14 选择红外信号输出极性

e) 设置寄存器 PAOUT[13]/PBOUT[9]/PCOUT[4]门控红外信号的输出

8.4.6 端口高电平中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_HIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

8.4.7 端口低电平中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ LIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

8.4.8 端口上升沿中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ RIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

8.4.9 端口下降沿中断操作流程

a) 设置寄存器 PxADS[n]为 0

b) 设置寄存器 Px_SEL 为 0

c) 设置寄存器 PxDIR[n]为 1

d) 设置寄存器 Px_ FIE[n]为 1

e) 中断触发后读取中断状态寄存器 Px_STAT[n]

f) 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]

8.4.10 端口上拉使能配置操作流程

a) 设置寄存器 PxPU[n]为 1

8.4.11 端口下拉使能配置操作流程

a) 设置寄存器 PxPU[n]为 0

b) 设置寄存器 PxPD[n]为 1

注：当 PxPU[n],PxPD[n]同时置 1 时，PxPU[n]优先级高，PxPD[n]无效。

8.4.12 端口增强驱动配置操作流程

a) 设置寄存器 PxDR[n]为 0

8.4.13 端口开漏输出配置操作流程

a) 设置寄存器 PxOD[n]为 1

8.4.14 端口位置位操作流程

a) 设置寄存器 PxBSET[n]为 1

8.4.15 端口位清零操作流程

a) 设置寄存器 PxBCLR[n]为 1

8.4.16 端口位置位清零操作流程

a) 设置寄存器 PxBSETCLR[n]为 1

8.5 端口控制器寄存器描述

寄存器列表 1　　基地址 1：0x40020C00

偏移量	寄存器名称	访问	寄存器描述
0x00	PA00_SEL	RW	端口 PA00 功能配置寄存器
0x04	PA01_SEL	RW	端口 PA01 功能配置寄存器
0x08	PA02_SEL	RW	端口 PA02 功能配置寄存器
0x0c	PA03_SEL	RW	端口 PA03 功能配置寄存器
0x10	PA04_SEL	RW	端口 PA04 功能配置寄存器
0x14	PA05_SEL	RW	端口 PA05 功能配置寄存器
0x18	PA06_SEL	RW	端口 PA06 功能配置寄存器
0x1c	PA07_SEL	RW	端口 PA07 功能配置寄存器
0x20	PA08_SEL	RW	端口 PA08 功能配置寄存器
0x24	PA09_SEL	RW	端口 PA09 功能配置寄存器
0x28	PA10_SEL	RW	端口 PA10 功能配置寄存器
0x2c	PA11_SEL	RW	端口 PA11 功能配置寄存器
0x30	PA12_SEL	RW	端口 PA12 功能配置寄存器
0x34	PA13_SEL	RW	端口 PA13 功能配置寄存器
0x38	PA14_SEL	RW	端口 PA14 功能配置寄存器
0x3c	PA15_SEL	RW	端口 PA15 功能配置寄存器
0x40	PB00_SEL	RW	端口 PB00 功能配置寄存器
0x44	PB01_SEL	RW	端口 PB01 功能配置寄存器
0x48	PB02_SEL	RW	端口 PB02 功能配置寄存器
0x4c	PB03_SEL	RW	端口 PB03 功能配置寄存器
0x50	PB04_SEL	RW	端口 PB04 功能配置寄存器
0x54	PB05_SEL	RW	端口 PB05 功能配置寄存器
0x58	PB06_SEL	RW	端口 PB06 功能配置寄存器
0x5c	PB07_SEL	RW	端口 PB07 功能配置寄存器
0x60	PB08_SEL	RW	端口 PB08 功能配置寄存器
0x64	PB09_SEL	RW	端口 PB09 功能配置寄存器
0x68	PB10_SEL	RW	端口 PB10 功能配置寄存器
0x6c	PB11_SEL	RW	端口 PB11 功能配置寄存器
0x70	PB12_SEL	RW	端口 PB12 功能配置寄存器
0x74	PB13_SEL	RW	端口 PB13 功能配置寄存器
0x78	PB14_SEL	RW	端口 PB14 功能配置寄存器
0x7c	PB15_SEL	RW	端口 PB15 功能配置寄存器
0x80	PC00_SEL	RW	端口 PC00 功能配置寄存器
0x84	PC01_SEL	RW	端口 PC01 功能配置寄存器
0x88	PC02_SEL	RW	端口 PC02 功能配置寄存器
0x8c	PC03_SEL	RW	端口 PC03 功能配置寄存器

0x90	PC04_SEL	RW	端口 PC04 功能配置寄存器
0x94	PC05_SEL	RW	端口 PC05 功能配置寄存器
0x98	PC06_SEL	RW	端口 PC06 功能配置寄存器
0x9c	PC07_SEL	RW	端口 PC07 功能配置寄存器
0xa0	PC08_SEL	RW	端口 PC08 功能配置寄存器
0xa4	PC09_SEL	RW	端口 PC09 功能配置寄存器
0xa8	PC10_SEL	RW	端口 PC10 功能配置寄存器
0xac	PC11_SEL	RW	端口 PC11 功能配置寄存器
0xb0	PC12_SEL	RW	端口 PC12 功能配置寄存器
0xb4	PC13_SEL	RW	端口 PC13 功能配置寄存器
0xb8	PC14_SEL	RW	端口 PC14 功能配置寄存器
0xbc	PC15_SEL	RW	端口 PC15 功能配置寄存器
0xc0	PD00_SEL	RW	端口 PD00 功能配置寄存器
0xc4	PD01_SEL	RW	端口 PD01 功能配置寄存器
0xc8	PD02_SEL	RW	端口 PD02 功能配置寄存器
0xcc	PD03_SEL	RW	端口 PD03 功能配置寄存器
0xd0	PD04_SEL	RW	端口 PD04 功能配置寄存器
0xd4	PD05_SEL	RW	端口 PD05 功能配置寄存器
0xd8	PD06_SEL	RW	端口 PD06 功能配置寄存器
0xdc	PD07_SEL	RW	端口 PD07 功能配置寄存器
0xe0	PD08_SEL	RW	端口 PD08 功能配置寄存器
0xe4	PD09_SEL	RW	端口 PD09 功能配置寄存器
0xe8	PD10_SEL	RW	端口 PD10 功能配置寄存器
0xec	PD11_SEL	RW	端口 PD11 功能配置寄存器
0xf0	PD12_SEL	RW	端口 PD12 功能配置寄存器
0xf4	PD13_SEL	RW	端口 PD13 功能配置寄存器
0xf8	PD14_SEL	RW	端口 PD14 功能配置寄存器
0xfc	PD15_SEL	RW	端口 PD15 功能配置寄存器
0x100	PADIR	RW	端口 PA 输入输出配置寄存器
0x104	PAIN	RO	端口 PA 输入值寄存器
0x108	PAOUT	RW	端口 PA 输出值配置寄存器
0x10c	PAADS	RW	端口 PA 数模配置寄存器
0x110	PABSET	RW	端口 PA 位置位寄存器
0x114	PABCLR	RW	端口 PA 位清零寄存器
0x118	PABSETCLR	RW	端口 PA 位置位清零寄存器
0x11c	PADR	RW	端口 PA 驱动能力配置寄存器
0x120	PAPU	RW	端口 PA 上拉使能配置寄存器
0x124	PAPD	RW	端口 PA 下拉使能配置寄存器
0x12c	PAOD	RW	端口 PA 开漏输出配置寄存器
0x130	PAHIE	RW	端口 PA 高电平中断使能配置寄存器
0x134	PALIE	RW	端口 PA 低电平中断使能配置寄存器

0x138	PARIE	RW	端口 PA 上升沿中断使能配置寄存器
0x13c	PAFIE	RW	端口 PA 下降沿中断使能配置寄存器
0x200	PA_STAT	RO	端口 PA 中断状态寄存器
0x210	PA_ICLR	RW	端口 PA 中断清除寄存器
0x140	PBDIR	RW	端口 PB 输入输出配置寄存器
0x144	PBIN	RO	端口 PB 输入值寄存器
0x148	PBOUT	RW	端口 PB 输出值配置寄存器
0x14c	PBADS	RW	端口 PB 数模配置寄存器
0x150	PBBSET	RW	端口 PB 位置位寄存器
0x154	PBBCLR	RW	端口 PB 位清零寄存器
0x158	PBBSETCLR	RW	端口 PB 位置位清零寄存器
0x15c	PBDR	RW	端口 PB 驱动能力配置寄存器
0x160	PBPU	RW	端口 PB 上拉使能配置寄存器
0x164	PBPD	RW	端口 PB 下拉使能配置寄存器
0x16c	PBOD	RW	端口 PB 开漏输出配置寄存器
0x170	PBHIE	RW	端口 PB 高电平中断使能配置寄存器
0x174	PBLIE	RW	端口 PB 低电平中断使能配置寄存器
0x178	PBRIE	RW	端口 PB 上升沿中断使能配置寄存器
0x17c	PBFIE	RW	端口 PB 下降沿中断使能配置寄存器
0x240	PB_STAT	RO	端口 PB 中断状态寄存器
0x250	PB_ICLR	RW	端口 PB 中断清除寄存器
0x180	PCDIR	RW	端口 PC 输入输出配置寄存器
0x184	PCIN	RO	端口 PC 输入值寄存器
0x188	PCOUT	RW	端口 PC 输出值配置寄存器
0x18c	PCADS	RW	端口 PC 数模配置寄存器
0x190	PCBSET	RW	端口 PC 位置位寄存器
0x194	PCBCLR	RW	端口 PC 位清零寄存器
0x198	PCBSETCLR	RW	端口 PC 位置位清零寄存器
0x19c	PCDR	RW	端口 PC 驱动能力配置寄存器
0x1a0	PCPU	RW	端口 PC 上拉使能配置寄存器
0x1a4	PCPD	RW	端口 PC 下拉使能配置寄存器
0x1ac	PCOD	RW	端口 PC 开漏输出配置寄存器
0x1b0	PCHIE	RW	端口 PC 高电平中断使能配置寄存器
0x1b4	PCLIE	RW	端口 PC 低电平中断使能配置寄存器
0x1b8	PCRIE	RW	端口 PC 上升沿中断使能配置寄存器
0x1bc	PCFIE	RW	端口 PC 下降沿中断使能配置寄存器
0x280	PC_STAT	RO	端口 PC 中断状态寄存器
0x290	PC_ICLR	RW	端口 PC 中断清除寄存器
0x1c0	PDDIR	RW	端口 PD 输入输出配置寄存器
0x1c4	PDIN	RO	端口 PD 输入值寄存器
0x1c8	PDOUT	RW	端口 PD 输出值配置寄存器

0x1cc	PDADS	RW	端口 PD 数模配置寄存器
0x1d0	PDBSET	RW	端口 PD 位置位寄存器
0x1d4	PDBCLR	RW	端口 PD 位清零寄存器
0x1d8	PDBSETCLR	RW	端口 PD 位置位清零寄存器
0x1dc	PDDR	RW	端口 PD 驱动能力配置寄存器
0x1e0	PDPU	RW	端口 PD 上拉使能配置寄存器
0x1e4	PDPD	RW	端口 PD 下拉使能配置寄存器
0x1ec	PDOD	RW	端口 PD 开漏输出配置寄存器
0x1f0	PDHIE	RW	端口 PD 高电平中断使能配置寄存器
0x1f4	PDLIE	RW	端口 PD 低电平中断使能配置寄存器
0x1f8	PDRIE	RW	端口 PD 上升沿中断使能配置寄存器
0x1fc	PDFIE	RW	端口 PD 下降沿中断使能配置寄存器
0x2c0	PD_STAT	RO	端口 PD 中断状态寄存器
0x2d0	PD_ICLR	RW	端口 PD 中断清除寄存器
0x304	GPIO_CTRL1	RW	端口辅助功能配置寄存器 1
0x308	GPIO_CTRL2	RW	端口辅助功能配置寄存器 2
0x30c	GPIO_TIMGS	RW	端口辅助功能定时器门控选择
0x310	GPIO_TIMES	RW	端口辅助功能定时器ETR选择
0x314	GPIO_TIMCPS	RW	端口辅助功能定时器捕获输入选择
0x318	GPIO_PCAS	RW	端口辅助功能PCA捕获选择
0x1000	PE00_SEL	RW	端口 PE00 功能配置寄存器
0x1004	PE01_SEL	RW	端口 PE01 功能配置寄存器
0x1008	PE02_SEL	RW	端口 PE02 功能配置寄存器
0x100c	PE03_SEL	RW	端口 PE03 功能配置寄存器
0x1010	PE04_SEL	RW	端口 PE04 功能配置寄存器
0x1014	PE05_SEL	RW	端口 PE05 功能配置寄存器
0x1018	PE06_SEL	RW	端口 PE06 功能配置寄存器
0x101c	PE07_SEL	RW	端口 PE07 功能配置寄存器
0x1020	PE08_SEL	RW	端口 PE08 功能配置寄存器
0x1024	PE09_SEL	RW	端口 PE09 功能配置寄存器
0x1028	PE10_SEL	RW	端口 PE10 功能配置寄存器
0x102c	PE11_SEL	RW	端口 PE11 功能配置寄存器
0x1030	PE12_SEL	RW	端口 PE12 功能配置寄存器
0x1034	PE13_SEL	RW	端口 PE13 功能配置寄存器
0x1038	PE14_SEL	RW	端口 PE14 功能配置寄存器
0x103c	PE15_SEL	RW	端口 PE15 功能配置寄存器
0x1040	PF00_SEL	RW	端口 PF00 功能配置寄存器
0x1044	PF01_SEL	RW	端口 PF01 功能配置寄存器
0x1048	PF02_SEL	RW	端口 PF02 功能配置寄存器
0x104c	PF03_SEL	RW	端口 PF03 功能配置寄存器
0x1050	PF04_SEL	RW	端口 PF04 功能配置寄存器

0x1054	PF05_SEL	RW	端口 PF05 功能配置寄存器
0x1058	PF06_SEL	RW	端口 PF06 功能配置寄存器
0x105c	PF07_SEL	RW	端口 PF07 功能配置寄存器
0x1064	PF09_SEL	RW	端口 PF09 功能配置寄存器
0x1068	PF10_SEL	RW	端口 PF10 功能配置寄存器
0x106c	PF11_SEL	RW	端口 PF11 功能配置寄存器
0x1100	PEDIR	RW	端口 PE 输入输出配置寄存器
0x1104	PEIN	RO	端口 PE 输入值寄存器
0x1108	PEOUT	RW	端口 PE 输出值配置寄存器
0x110c	PEADS	RW	端口 PE 数模配置寄存器
0x1110	PEBSET	RW	端口 PE 位置位寄存器
0x1114	PEBCLR	RW	端口 PE 位清零寄存器
0x1118	PEBSETCLR	RW	端口 PE 位置位清零寄存器
0x111c	PEDR	RW	端口 PE 驱动能力配置寄存器
0x1120	PEPU	RW	端口 PE 上拉使能配置寄存器
0x1124	PEPD	RW	端口 PE 下拉使能配置寄存器
0x112c	PEOD	RW	端口 PE 开漏输出配置寄存器
0x1130	PEHIE	RW	端口 PE 高电平中断使能配置寄存器
0x1134	PELIE	RW	端口 PE 低电平中断使能配置寄存器
0x1138	PERIE	RW	端口 PE 上升沿中断使能配置寄存器
0x113c	PEFIE	RW	端口 PE 下降沿中断使能配置寄存器
0x1200	PE_STAT	RO	端口 PE 中断状态寄存器
0x1210	PE_ICLR	RW	端口 PE 中断清除寄存器
0x1140	PFDIR	RW	端口 PF 输入输出配置寄存器
0x1144	PFIN	RO	端口 PF 输入值寄存器
0x1148	PFOUT	RW	端口 PF 输出值配置寄存器
0x114c	PFADS	RW	端口 PF 数模配置寄存器
0x1150	PFBSET	RW	端口 PF 位置位寄存器
0x1154	PFBCLR	RW	端口 PF 位清零寄存器
0x1158	PFBSETCLR	RW	端口 PF 位置位清零寄存器
0x115c	PFDR	RW	端口 PF 驱动能力配置寄存器
0x1160	PFPU	RW	端口 PF 上拉使能配置寄存器
0x1164	PFPD	RW	端口 PF 下拉使能配置寄存器
0x116c	PFOD	RW	端口 PF 开漏输出配置寄存器
0x1170	PFHIE	RW	端口 PF 高电平中断使能配置寄存器
0x1174	PFLIE	RW	端口 PF 低电平中断使能配置寄存器
0x1178	PFRIE	RW	端口 PF 上升沿中断使能配置寄存器
0x117c	PFFIE	RW	端口 PF 下降沿中断使能配置寄存器
0x1240	PF_STAT	RO	端口 PF 中断状态寄存器
0x1250	PF_ICLR	RW	端口 PF 中断清除寄存器

8.5.1 端口通用寄存器

8.5.1.1 端口 Px 输入输出配置寄存器(PxDIR) (x = A,B,C,D,E,F)

地址偏移量：0x100(PA),0x140(PB),0x180(PC),0x1C0(PD),0x1100(PE),0x1140(PF)

复位值：0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDIR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDIR	端口Px输入输出配置寄存器 (对应Px15-Px00) 1：配置成输入　　0：配置成输出 注：每个bit对应一个端口，例如：PxDIR[15]对应端口Px15

8.5.1.2 端口 Px 输入值寄存器(PxIN) (x = A,B,C,D,E,F)

地址偏移量：0x104(PA),0x144(PB),0x184(PC),0x1C4(PD),0x1104(PE),0x1144(PF)

复位值：NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxIN[15:0]															
ro															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxIN	端口Px输入值寄存器 (对应Px15-Px00) 1：输入为高电平 0：输入为低电平

8.5.1.3 端口 Px 输出值配置寄存器(PxOUT) (x = A,B,C,D,E,F)

地址偏移量：0x108(PA),0x148(PB),0x188(PC),0x1C8(PD),0x1108(PE),0x1148(PF)

复位值：NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOUT[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOUT	端口Px输出值配置寄存器 (对应Px15-Px00) 1：输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0：输出低电平。

8.5.1.4 端口 Px 数模配置寄存器(PxADS) (x = A,B,C,D,E,F)

地址偏移量：0x10C(PA),0x14C(PB),0x18C(PC),0x1CC(PD),0x110C(PE),0x114C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxADS[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxADS	端口Px数模配置寄存器 (对应Px15-Px00) 1：配置为模拟端口 0：配置为数字端口

8.5.1.5 端口 Px 位置位寄存器(PxBSET) (x = A,B,C,D,E,F)

地址偏移量：0x110(PA),0x150(PB),0x190(PC),0x1D0(PD),0x1110(PE),0x1150(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBSET[15:0]															
W1															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxBSET	端口Px位置位寄存器 (对应Px15-Px00) 1：置位 0：保持

8.5.1.6 端口 Px 位清零寄存器(PxBCLR) (x = A,B,C,D,E,F)

地址偏移量：0x114(PA),0x154(PB),0x194(PC),0x1D4(PD),0x1114(PE),0x1154(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBCLR[15:0]															
W1															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxBCLR	端口Px位清零寄存器 (对应Px15-Px00) 1：清零 0：保持

8.5.1.7 端口 Px 位置位清零寄存器(PxBSETCLR) (x = A,B,C,D,E,F)

地址偏移量：0x118(PA),0x158(PB),0x198(PC),0x1D8(PD),0x1118(PE),0x1158(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PxBSET[15:0]															
W1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBCLR[15:0]															
W1															

位	标记	功能描述
31:16	PxBSET	端口Px位置位寄存器 1：置位 0：保持
15:0	PxBCLR	端口Px位清零寄存器 1：清零 0：保持

注意：

– PxBSET 和 PxBCLR 相同位同时置 1 时，PxBCLR 具有高优先级。即该端口被清零。

8.5.1.8 端口 Px 上拉使能配置寄存器(PxPU) (x = A,B,C,D,E,F)

地址偏移量：0x120(PA),0x160(PB),0x1A0(PC),0x1E0(PD),0x1120(PE),0x1160(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPU[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPU	端口Px上拉使能配置寄存器 1：使能 0：禁止

8.5.1.9 端口 Px 驱动能力配置寄存器(PxDR) (x = A,B,C,D,E,F)

地址偏移量：0x11C(PA),0x15C(PB),0x19C(PC),0x1DC(PD),0x111C(PE),0x115C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDR	端口Px驱动能力配置寄存器 1：低驱动能力 0：高驱动能力

8.5.1.10 端口 Px 下拉使能配置寄存器(PxPD) (x = A,B,C,D,E,F)

地址偏移量：0x124(PA),0x164(PB),0x1A4(PC),0x1E4(PD),0x1124(PE),0x1164(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPD[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPD	端口Px下拉使能配置寄存器 1：使能 0：禁止

8.5.1.11 端口 Px 开漏输出配置寄存器(PxOD) (x = A,B,C,D,E,F)

地址偏移量：0x12C(PA),0x16C(PB),0x1AC(PC),0x1EC(PD),0x112C(PE),0x116C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOD[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOD	端口Px开漏输出配置寄存器 1：将端口输出方式设置为开漏输出 0：将端口输出方式设置为推挽输出

8.5.1.12 端口 Px 高电平中断使能配置寄存器(PxHIE) (x = A,B,C,D,E,F)

地址偏移量：0x130(PA),0x170(PB),0x1B0(PC),0x1F0(PD),0x1130(PE),0x1170(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxHIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxHIE	端口Px高电平中断使能配置寄存器 1：使能 0：禁止

8.5.1.13 端口 Px 低电平中断使能配置寄存器(PxLIE) (x = A,B,C,D,E,F)

地址偏移量：0x134(PA),0x174(PB),0x1B4(PC),0x1F4(PD),0x1134(PE),0x1174(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxLIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxLIE	端口Px低电平中断使能配置寄存器 1：使能 0：禁止

8.5.1.14 端口 Px 上升沿中断使能配置寄存器(PxRIE) (x = A,B,C,D,E,F)

地址偏移量：0x138(PA),0x178(PB),0x1B8(PC),0x1F8(PD),0x1138(PE),0x1178(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxRIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxRIE	端口Px上升沿中断使能配置寄存器 1：使能 0：禁止

8.5.1.15 端口 Px 下降沿中断使能配置寄存器(PxFIE) (x = A,B,C,D,E,F)

地址偏移量：0x13C(PA),0x17C(PB),0x1BC(PC),0x1FC(PD),0x113C(PE),0x117C(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxFIE[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxFIE	端口Px下降沿中断使能配置寄存器 1：使能 0：禁止

8.5.1.16 端口 Px 中断状态寄存器(Px_STAT) (x = A,B,C,D,E,F)

地址偏移量：0x200(PA),0x240(PB),0x280(PC),0x2C0(PD),0x1200(PE),0x1240(PF)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px_STAT[15:0]															
ro															

位	标记	功能描述
31:16	Reserved	保留
15:0	Px_STAT	端口Px中断状态寄存器 1：中断触发 0：无中断触发

8.5.1.17 端口 Px 中断清除寄存器(Px_ICLR) (x = A,B,C,D,E,F)

地址偏移量：0x210(PA),0x250(PB),0x290(PC),0x2D0(PD),0x1210(PE),0x1250(PF)

复位值：0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px_ICLR[15:0]															
rw															

位	标记	功能描述
31:16	Reserved	保留
15:0	Px_ICLR	端口Px中断清除寄存器 1：保留中断标志位 0：清除中断标志位

8.5.2 端口辅助寄存器

8.5.2.1 端口辅助功能配置寄存器 1(GPIO_CTRL1)

地址偏移量：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ir_pol	hclk_en	pclk_en	hclk_sel		pclk_sel		ssn0_sel				ext_clk_sel			
	rw	rw	rw	rw		rw		rw				rw			

位	标记	功能描述
31:15	Reserved	保留
14	ir_pol	IR输出极性选择. 0 – 正向输出 1 – 反向输出
13	hclk_en	hclk输出门控. 0 – 门控 1 – 输出
12	pclk_en	pclk输出门控. 0 – 门控 1 – 输出
11:10	hclk_sel	hclk输出分频选择. 0 0 hclk 0 1 hclk/2 1 0 hclk/4 1 1 hclk/8
9:8	pclk_sel	pclk输出分频选择. 0 0 pclk 0 1 pclk/2 1 0 pclk/4 1 1 pclk/8
7:4	ssn0_sel	SPI0 SSN信号来源选择. 0000 高电平 0001 PA03 0010 PA04

			0011	PA06	
			0100	PA08	
			0101	PA09	
			0110	PA12	
			0111	PA15	
			1000	PB01	
			1001	PB02	
			1010	PB05	
			1011	PB06	
			1100	PB09	
			1101	PB10	
			1110	PB12	
			1111	PB14	
3:0	ext_clk_sel	外部时钟信号来源选择.			
			0000	高电平	
			0001	PA03	
			0010	PA04	
			0011	PA06	
			0100	PA08	
			0101	PA09	
			0110	PA12	
			0111	PA15	
			1000	PB01	
			1001	PB02	
			1010	PB05	
			1011	PB06	
			1100	PB09	
			1101	PB10	
			1110	PB12	
			1111	PB14	

8.5.2.2 端口辅助功能配置寄存器 2(GPIO_CTRL2)

地址偏移量：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ahb_sel	Res							tclk_div		tclk_sel		ssn1_sel			
rw								rw		rw		rw			

位	标记	功能描述
31:4	Reserved	保留
15	ahb_sel	AHB总线访问模式 0 – 使用FASTIO总线访问PxIN,PxOUT寄存器 1 -- 使用AHB总线访问PxIN,PxOUT寄存器
14:8	Reserved	保留
7:6	tclk_div	tclk输出分频选择. 0 0 tclk 0 1 tclk/2 1 0 tclk/4 1 1 tclk/8
5:4	tclk_sel	tclk输出信号源选择. 0 0 - 0 1 RCH 1 0 XTH 1 1 PLL
3:0	ssn1_sel	SPI1 SSN信号来源选择. 0000 高电平 0001 PA03 0010 PA04 0011 PA06 0100 PA08 0101 PA09 0110 PA12 0111 PA15 1000 PB01 1001 PB02

			1010	PB05	
			1011	PB06	
			1100	PB09	
			1101	PB10	
			1110	PB12	
			1111	PB14	

8.5.2.3 端口辅助功能定时器门控选择(GPIO_TIMGS)

地址偏移量：0x30C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIM3_G			TIM2_G			TIM1_G			TIM0_G		
				R/W			R/W			R/W			R/W		

位	标记	功能描述
31:12	Reserved	保留
11:9	TIM3_G	Timer3 定时器GATE输入选择，选择见下表
8:6	TIM2_G	Timer2 定时器GATE输入选择，选择见下表
5:3	TIM1_G	Timer1 定时器GATE输入选择，选择见下表
2:0	TIM0_G	Timer0 定时器GATE输入选择，选择见下表

	TIM0_g	TIM1_g	TIM2_g	TIM3_g	-
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	-	UART0_RXD	UART0_RXD	-
010	UART1_RXD	-	UART1_RXD	UART1_RXD	-
011	VC0_OUT	VC0_OUT	VC0_OUT	UART2	VC0_OUT
100	VC1_OUT	VC1_OUT	VC1_OUT	UART3	VC1_OUT
101	PA03	PA08	PA10	VC0_OUT	PB03
110	PB08	PB03	PB04	PA06	PB05
111	PB15	PB13	PB11	PA11	PC00

8.5.2.4 端口辅助功能定时器 ETR 选择(GPIO_TIMES)

地址偏移量：0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIM3_E			TIM2_E			TIM1_E			TIM0_E		
				R/W			R/W			R/W			R/W		

位	标记	功能描述
31:12	Reserved	保留
11:9	TIM3_E	Timer3 定时器ETR输入选择，选择见下表
8:6	TIM2_E	Timer2 定时器ETR输入选择，选择见下表
5:3	TIM1_E	Timer1 定时器ETR输入选择，选择见下表
2:0	TIM0_E	Timer0 定时器ETR输入选择，选择见下表

	TIM0_e	TIM1_e	TIM2_e	TIM3_e	-
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	-	UART0_RXD	-	UART0_RXD	-
010	-	UART1_RXD	-	UART1_RXD	LVD_OUT
011	VC0_OUT	VC1_OUT	VC0_OUT	VC1_OUT	VC0_OUT
100	LVD_OUT	LVD_OUT	-	-	VC1_OUT
101	PA00	PA01	PA04	PA00	PB04
110	PA05	PC09	PC04	PA12	PB06
111	PA15	PD02	PC08	PA13	PC03

8.5.2.5 端口辅助功能定时器捕获输入选择(GPIO_TIMCPS)

地址偏移量：0x314

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM3_CB		TIM3_CA		TIM2_CA		TIM1_CA		TIM0_CA						
	R/W		R/W		R/W		R/W		R/W						

位	标记	功能描述
31:15	Reserved	保留
14:12	TIM3_CB	Timer3 定时器CH0B输入选择，选择见下表
11:9	TIM3_CA	Timer3 定时器CH0A输入选择，选择见下表
8:6	TIM2_CA	Timer2 定时器CHA输入选择，选择见下表
5:3	TIM1_CA	Timer1 定时器CHA输入选择，选择见下表
2:0	TIM0_CA	Timer0 定时器CHA输入选择，选择见下表

	TIM0_CHA	TIM1_CHA	TIM2_CHA	TIM3_CH0A	TIM3_CH0B
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	UART1_RXD	-	-	UART0_RXD
010	PA00	PA00	VC0_OUT	-	UART1_RXD
011	PA02	PA02	PA02	-	-
100	PA05	PA06	PA07	VC0_OUT	VC1_OUT
101	PA15	PB08	PB08	PA08	PA07
110	PB06	PB10	PB09	PB03	PB04
111	PB14	PB13	PC06	PB06	PB13

8.5.2.6 端口辅助功能 PCA 捕获选择(GPIO_PCAS)

地址偏移量：0x318

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										PCA_CH0			PCA_ECI		
										R/W			R/W		

位	标记	功能描述
31:6	Reserved	保留
5:3	PCA_ECI	PCA ECI时钟输入选择，选择见下表
2:0	PCA_CH0	PCA CH0 捕获口输入选择，选择见下表

	pca_eci	pca_ch0
000	PX_SEL	PX_SEL
001	-	-
010	LVD_OUT	-
011	VC0_OUT	LVD_OUT
100	VC1_OUT	VC1_OUT
101	PA05	PA06
110	PB02	PB04
111	PD02	PC06

8.5.3 端口 PA 功能选择寄存器

8.5.3.1 端口 PA00 功能配置寄存器(PA00_SEL)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA00_sel	端口PA00功能选择. 000 ---- GPIO PA00 001 ---- UART1_CTS UART1模块CTS信号 010 ---- Reserved Reserved 011 ---- TIM0_ETR Timer0模块外部时钟输入信号 100 ---- VC0_OUT VC0模块输出/反向输出信号 101 ---- TIM1_CHA Timer1模块通道A信号 110 ---- TIM3_ETR Timer3模块外部时钟输入信号 111 ---- TIM0_CHA Timer0模块通道A信号

8.5.3.2 端口 PA01 功能配置寄存器(PA01_SEL)

地址偏移量：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA01_sel	端口PA01功能选择. 000 ---- GPIO PA01 001 ---- UART1_RTS UART1模块RTS信号 010 ---- Reserved Reserved 011 ---- TIM0_CHB Timer0模块通道B信号 100 ---- TIM1_ETR Timer1模块外部时钟输入信号 101 ---- TIM1_CHB Timer1模块通道B信号 110 ---- HCLK_OUT AHB总线时钟输出信号 111 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号

8.5.3.3 端口 PA02 功能配置寄存器(PA02_SEL)

地址偏移量：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved-															PA02_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA02_sel	端口PA02功能选择. 000 ---- GPIO PA02 001 ---- UART1_TXD UART1模块TXD信号 010 ---- TIM0_CHA Timer0模块通道A信号 011 ---- VC1_OUT VC1模块输出/反向输出信号 100 ---- TIM1_CHA Timer1模块通道A信号 101 ---- TIM2_CHA Timer2模块通道A信号 110 ---- PCLK_OUT APB总线时钟输出信号 111 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号

8.5.3.4 端口 PA03 功能配置寄存器(PA03_SEL)

地址偏移量：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA03_sel	端口PA03功能选择. 000 ---- GPIO PA03 001 ---- UART1_RXD UART1模块RXD信号 010 ---- TIM0_GATE Timer0模块门控信号 011 ---- TIM1_CHB Timer1模块通道B信号 100 ---- TIM2_CHB Timer2模块通道B信号 101 ---- SPI1_CS SPI1模块主机模式片选信号 110 ---- TIM3_CH1A Timer3模块通道1A信号 111 ---- TIM5_CHA Timer6模块通道1A信号

8.5.3.5 端口 PA04 功能配置寄存器(PA04_SEL)

地址偏移量：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PA04_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PA04_sel	端口PA04功能选择. 000 ---- GPIO PA04 001 ---- SPI0_CS SPI0模块主机模式片选信号 010 ---- UART1_TXD UART1模块TXD信号 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- TIM2_ETR Timer2模块外部时钟输入信号 101 ---- TIM5_CHA Timer6模块通道1A信号 110 ---- LVD_OUT LVD模块输出信号 111 ---- TIM3_CH2B Timer3模块通道2B信号

8.5.3.6 端口 PA05 功能配置寄存器(PA05_SEL)

地址偏移量：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA05_sel	端口PA05功能选择. 000 ---- GPIO PA05 001 ---- SPI0_SCK SPI0模块时钟信号 010 ---- TIM0_ETR Timer0模块外部时钟输入信号 011 ---- PCA_ECI PCA模块外部时钟输入信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- TIM5_CHB Timer6模块通道1B信号 110 ---- XTL_OUT 外部32K晶振输出信号 111 ---- XTH_OUT 外部32M晶振输出信号

8.5.3.7 端口 PA06 功能配置寄存器(PA06_SEL)

地址偏移量：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA06_sel	端口PA06功能选择. 000 ---- GPIO PA06 001 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 010 ---- PCA_CH0 PCA模块通道0捕获/比较信号 011 ---- TIM3_BK Timer3模块刹车信号 100 ---- TIM1_CHA Timer1模块通道A信号 101 ---- VC0_OUT VC0模块输出/反向输出信号 110 ---- TIM3_GATE Timer3模块门控信号 111 ---- Reserved Reserved

8.5.3.8 端口 PA07 功能配置寄存器(PA07_SEL)

地址偏移量：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA07_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA07_sel	端口PA07功能选择. 000 ---- GPIO PA07 001 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 010 ---- PCA_CH1 PCA模块通道1捕获/比较信号 011 ---- HCLK_OUT AHB总线时钟输出信号 100 ---- TIM3_CH0B Timer3模块通道0B信号 101 ---- TIM2_CHA Timer2模块通道A信号 110 ---- VC1_OUT VC1模块输出/反向输出信号 111 ---- TIM4_CHB Timer6模块通道0B信号

8.5.3.9 端口 PA08 功能配置寄存器(PA08_SEL)

地址偏移量：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA08_sel	PA08_sel: 端口PA08功能选择. 000 ---- GPIO PA08 001 ---- UART0_TXD UART0模块TXD信号 010 ---- TIM3_CH0A Timer3模块通道0A信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- TIM1_GATE Timer1模块门控信号 110 ---- TIM4_CHA Timer6模块通道0A信号 111 ---- TIM3_BK Timer3模块刹车信号

8.5.3.10 端口 PA09 功能配置寄存器(PA09_SEL)

地址偏移量：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA09_sel	端口PA09功能选择. 000 ---- GPIO PA09 001 ---- UART0_TXD UART0模块TXD信号 010 ---- TIM3_CH1A Timer3模块通道1A信号 011 ---- TIM0_BK Timer0模块刹车信号 100 ---- I2C0_SCL I2C0模块时钟信号 101 ---- Reserved Reserved 110 ---- HCLK_OUT AHB总线时钟输出信号 111 ---- TIM5_CHA Timer6模块通道1A信号

8.5.3.11 端口 PA10 功能配置寄存器(PA10_SEL)

地址偏移量：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA10_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA10_sel	PA10_sel: 端口PA10功能选择. 000 ---- GPIO PA10 001 ---- UART0_RXD UART0模块RXD信号 010 ---- TIM3_CH2A Timer3模块通道2A信号 011 ---- TIM2_BK Timer2模块刹车信号 100 ---- I2C0_SDA I2C0模块数据信号 101 ---- TIM2_GATE Timer2模块门控信号 110 ---- PCLK_OUT APB总线时钟输出信号 111 ---- TIM6_CHA Timer6模块通道2A信号

8.5.3.12 端口 PA11 功能配置寄存器(PA11_SEL)

地址偏移量：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA11_sel	PA11_sel: 端口PA11功能选择. 000 ---- GPIO PA11 001 ---- UART0_CTS UART0模块CTS信号 010 ---- TIM3_GATE Timer3模块门控信号 011 ---- I2C1_SCL I2C1模块时钟信号 100 ---- Reserved Reserved 101 ---- VC0_OUT VC0模块输出/反向输出信号 110 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 111 ---- TIM4_CHB Timer6模块通道0B信号

8.5.3.13 端口 PA12 功能配置寄存器(PA12_SEL)

地址偏移量：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA12_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA12_sel	PA12_sel: 端口PA12功能选择. 000 ---- GPIO PA12 001 ---- UART0_RTS UART0模块RTS信号 010 ---- TIM3_ETR Timer3模块外部时钟输入信号 011 ---- I2C1_SDA I2C1模块数据信号 100 ---- Reserved Reserved 101 ---- VC1_OUT VC1模块输出/反向输出信号 110 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 111 ---- Reserved Reserved

8.5.3.14 端口 PA13 功能配置寄存器(PA13_SEL)

地址偏移量：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA13_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA13_sel	PA13_sel: 端口PA13功能选择. 000 ---- GPIO PA13 001 ---- IR_OUT 红外输出信号 010 ---- UART0_RXD UART0模块RXD信号 011 ---- LVD_OUT LVD模块输出信号 100 ---- TIM3_ETR Timer3模块外部时钟输入信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- VC2_OUT VC2模块输出/反向输出信号

8.5.3.15 端口 PA14 功能配置寄存器(PA14_SEL)

地址偏移量：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PA14_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PA14_sel	PA14_sel: 端口PA14功能选择. 000 ---- GPIO PA14 001 ---- UART1_TXD UART1模块TXD信号 010 ---- UART0_TXD UART0模块TXD信号 011 ---- TIM3_CH2A Timer3模块通道2A信号 100 ---- LVD_OUT LVD模块输出信号 101 ---- RCH_OUT 内部24M RC时钟输出信号 110 ---- RCL_OUT 内部38K RC时钟输出信号 111 ---- PLL_OUT 内部PLL时钟输出信号

8.5.3.16 端口 PA15 功能配置寄存器(PA15_SEL)

地址偏移量：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PA15_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PA15_sel	PA15_sel: 端口PA15功能选择. 000 ---- GPIO PA15 001 ---- SPI0_CS SPI0模块主机模式片选信号 010 ---- UART1_RXD UART1模块RXD信号 011 ---- Reserved Reserved 100 ---- TIM0_ETR Timer0模块外部时钟输入信号 101 ---- TIM0_CHA Timer0模块通道A信号 110 ---- TIM3_CH1A Timer3模块通道1A信号 111 ---- Reserved Reserved

8.5.4 端口 PB 功能选择寄存器

8.5.4.1 端口 PB00 功能配置寄存器(PB00_SEL)

地址偏移量：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB00_sel	端口PB00功能选择. 000 ---- GPIO PB00 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- TIM3_CH1B Timer3模块通道1B信号 011 ---- Reserved Reserved 100 ---- TIM5_CHB Timer6模块通道1B信号 101 ---- RCH_OUT 内部24M RC时钟输出信号 110 ---- RCL_OUT 内部38K RC时钟输出信号 111 ---- PLL_OUT 内部PLL时钟输出信号

8.5.4.2 端口 PB01 功能配置寄存器(PB01_SEL)

地址偏移量：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB01_sel	PB01_sel: 端口PB01功能选择. 000 ---- GPIO PB01 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- PCLK_OUT APB总线时钟输出信号 011 ---- TIM3_CH2B Timer3模块通道2B信号 100 ---- TIM6_CHB Timer6模块通道2B信号 101 ---- Reserved Reserved 110 ---- VC2_OUT VC2模块输出/反向输出信号 111 ---- TCLK_OUT 内部时钟输出信号

8.5.4.3 端口 PB02 功能配置寄存器(PB02_SEL)

地址偏移量：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB02_sel	PB02_sel: 端口PB02功能选择. 000 ---- GPIO PB02 001 ---- Reserved Reserved 010 ---- PCA_ECI PCA模块外部时钟输入信号 011 ---- Reserved Reserved 100 ---- TIM4_CHA Timer6模块通道0A信号 101 ---- TIM1_BK Timer1模块刹车信号 110 ---- TIM0_BK Timer0模块刹车信号 111 ---- TIM2_BK Timer2模块刹车信号

8.5.4.4 端口 PB03 功能配置寄存器(PB03_SEL)

地址偏移量：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB03_sel	PB03_sel: 端口PB03功能选择. 000 ---- GPIO PB03 001 ---- SPI0_SCK SPI0模块时钟信号 010 ---- TIM0_CHB Timer0模块通道B信号 011 ---- TIM1_GATE Timer1模块门控信号 100 ---- TIM3_CH0A Timer3模块通道0A信号 101 ---- Reserved Reserved 110 ---- XTL_OUT 外部32K晶振输出信号 111 ---- XTH_OUT 外部32M晶振输出信号

8.5.4.5 端口 PB04 功能配置寄存器(PB04_SEL)

地址偏移量：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB04_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB04_sel	PB04_sel: 端口PB04功能选择. 000 ---- GPIO PB04 001 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 010 ---- PCA_CH0 PCA模块通道0捕获/比较信号 011 ---- TIM2_BK Timer2模块刹车信号 100 ---- UART0_CTS UART0模块CTS信号 101 ---- TIM2_GATE Timer2模块门控信号 110 ---- TIM3_CH0B Timer3模块通道0B信号 111 ---- Reserved Reserved

8.5.4.6 端口 PB05 功能配置寄存器(PB05_SEL)

地址偏移量：0x54

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB05_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB05_sel	PB05_sel: 端口PB05功能选择. 000 ---- GPIO PB05 001 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 010 ---- Reserved Reserved 011 ---- TIM1_BK Timer1模块刹车信号 100 ---- PCA_CH1 PCA模块通道1捕获/比较信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- UART0_RTS UART0模块RTS信号

8.5.4.7 端口 PB06 功能配置寄存器(PB06_SEL)

地址偏移量：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB06_sel	PB06_sel: 端口PB06功能选择. 000 ---- GPIO PB06 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- UART0_TXD UART0模块TXD信号 011 ---- TIM1_CHB Timer1模块通道B信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- Reserved Reserved 110 ---- TIM3_CH0A Timer3模块通道0A信号 111 ---- Reserved Reserved

8.5.4.8 端口 PB07 功能配置寄存器(PB07_SEL)

地址偏移量：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB07_sel	PB07_sel: 端口PB07功能选择. 000 ---- GPIO PB07 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- UART0_RXD UART0模块RXD信号 011 ---- TIM2_CHB Timer2模块通道B信号 100 ---- Reserved Reserved 101 ---- TIM0_CHB Timer0模块通道B信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.4.9 端口 PB08 功能配置寄存器(PB08_SEL)

地址偏移量：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB08_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB08_sel	PB08_sel: 端口PB08功能选择. 000 ---- GPIO PB08 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- TIM1_CHA Timer1模块通道A信号 011 ---- Reserved Reserved 100 ---- TIM2_CHA Timer2模块通道A信号 101 ---- TIM0_GATE Timer0模块门控信号 110 ---- TIM3_CH2A Timer3模块通道2A信号 111 ---- UART0_TXD UART0模块TXD信号

8.5.4.10 端口 PB09 功能配置寄存器(PB09_SEL)

地址偏移量：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB09_sel	PB09_sel: 端口PB09功能选择. 000 ---- GPIO PB09 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- IR_OUT 红外输出信号 011 ---- SPI1_CS SPI1模块主机模式片选信号 100 ---- TIM2_CHA Timer2模块通道A信号 101 ---- Reserved Reserved 110 ---- TIM2_CHB Timer2模块通道B信号 111 ---- UART0_RXD UART0模块RXD信号

8.5.4.11 端口 PB10 功能配置寄存器(PB10_SEL)

地址偏移量：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB10_sel	PB10_sel: 端口PB10功能选择. 000 ---- GPIO PB10 001 ---- I2C1_SCL I2C1模块时钟信号 010 ---- SPI1_SCK SPI1模块时钟信号 011 ---- TIM1_CHA Timer1模块通道A信号 100 ---- Reserved Reserved 101 ---- TIM3_CH1A Timer3模块通道1A信号 110 ---- Reserved Reserved 111 ---- UART1_RTS UART1模块RTS信号

8.5.4.12 端口 PB11 功能配置寄存器(PB11_SEL)

地址偏移量：0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB11_sel	PB11_sel: 端口PB11功能选择. 000 ---- GPIO PB11 001 ---- I2C1_SDA I2C1模块数据信号 010 ---- TIM1_CHB Timer1模块通道B信号 011 ---- Reserved Reserved 100 ---- TIM2_GATE Timer2模块门控信号 101 ---- TIM6_CHA Timer6模块通道2A信号 110 ---- Reserved Reserved 111 ---- UART1_CTS UART1模块CTS信号

8.5.4.13 端口 PB12 功能配置寄存器(PB12_SEL)

地址偏移量：0x70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB12_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB12_sel	PB12_sel: 端口PB12功能选择. 000 ---- GPIO PB12 001 ---- SPI1_CS SPI1模块主机模式片选信号 010 ---- TIM3_BK Timer3模块刹车信号 011 ---- Reserved Reserved 100 ---- TIM0_BK Timer0模块刹车信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- TIM6_CHA Timer6模块通道2A信号

8.5.4.14 端口 PB13 功能配置寄存器(PB13_SEL)

地址偏移量：0x74

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB13_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB13_sel	PB13_sel: 端口PB13功能选择. 000 ---- GPIO PB13 001 ---- SPI1_SCK SPI1模块时钟信号 010 ---- I2C1_SCL I2C1模块时钟信号 011 ---- TIM3_CH0B Timer3模块通道0B信号 100 ---- Reserved Reserved 101 ---- TIM1_CHA Timer1模块通道A信号 110 ---- TIM1_GATE Timer1模块门控信号 111 ---- TIM6_CHB Timer6模块通道2B信号

8.5.4.15 端口 PB14 功能配置寄存器(PB14_SEL)

地址偏移量：0x78

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PB14_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PB14_sel	PB14_sel: 端口PB14功能选择. 000 ---- GPIO PB14 001 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 010 ---- I2C1_SDA I2C1模块数据信号 011 ---- TIM3_CH1B Timer3模块通道1B信号 100 ---- TIM0_CHA Timer0模块通道A信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- TIM1_BK Timer1模块刹车信号

8.5.4.16 端口 PB15 功能配置寄存器(PB15_SEL)

地址偏移量：0x7C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PB15_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PB15_sel	PB15_sel: 端口PB15功能选择. 000 ---- GPIO PB15 001 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 010 ---- TIM3_CH2B Timer3模块通道2B信号 011 ---- TIM0_CHB Timer0模块通道B信号 100 ---- TIM0_GATE Timer0模块门控信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5 端口 PC 功能选择寄存器

8.5.5.1 端口 PC00 功能配置寄存器(PC00_SEL)

地址偏移量：0x80

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC00_sel	PC00_sel: 端口PC00功能选择. 000 ---- GPIO PC00 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- UART1_CTS UART1模块CTS信号 100 ---- UART2_RTS UART2模块RTS信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.2 端口 PC01 功能配置寄存器(PC01_SEL)

地址偏移量：0x84

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC01_sel	PC01_sel: 端口PC01功能选择. 000 ---- GPIO PC01 001 ---- Reserved Reserved 010 ---- TIM5_CHB Timer6模块通道1B信号 011 ---- UART1_RTS UART1模块RTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- UART2_CTS UART2模块CTS信号 111 ---- Reserved Reserved

8.5.5.3 端口 PC02 功能配置寄存器(PC02_SEL)

地址偏移量：0x88

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC02_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC02_sel	PC02_sel: 端口PC02功能选择. 000 ---- GPIO PC02 001 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- UART2_RXD UART2模块RXD信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.4 端口 PC03 功能配置寄存器(PC03_SEL)

地址偏移量：0x8C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC03_sel	PC03_sel: 端口PC03功能选择. 000 ---- GPIO PC03 001 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- UART2_TXD UART2模块TXD信号 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.5 端口 PC04 功能配置寄存器(PC04_SEL)

地址偏移量：0x90

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC04_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC04_sel	PC04_sel: 端口PC04功能选择. 000 ---- GPIO PC04 001 ---- Reserved Reserved 010 ---- TIM2_ETR Timer2模块外部时钟输入信号 011 ---- IR_OUT 红外输出信号 100 ---- VC2_OUT VC2模块输出/反向输出信号 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.6 端口 PC05 功能配置寄存器(PC05_SEL)

地址偏移量：0x94

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC05_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC05_sel	PC05_sel: 端口PC05功能选择. 000 ---- GPIO PC05 001 ---- Reserved Reserved 010 ---- TIM6_CHB Timer6模块通道2B信号 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.7 端口 PC06 功能配置寄存器(PC06_SEL)

地址偏移量：0x98

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC06_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC06_sel	PC06_sel: 端口PC06功能选择. 000 ---- GPIO PC06 001 ---- PCA_CH0 PCA模块通道0捕获/比较信号 010 ---- TIM4_CHA Timer6模块通道0A信号 011 ---- TIM2_CHA Timer2模块通道A信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- UART3_RXD UART3模块RXD信号 111 ---- Reserved Reserved

8.5.5.8 端口 PC07 功能配置寄存器(PC07_SEL)

地址偏移量：0x9C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC07_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC07_sel	PC07_sel: 端口PC07功能选择. 000 ---- GPIO PC07 001 ---- PCA_CH1 PCA模块通道1捕获/比较信号 010 ---- TIM5_CHA Timer6模块通道1A信号 011 ---- TIM2_CHB Timer2模块通道B信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- UART3_TXD UART3模块TXD信号 111 ---- Reserved Reserved

8.5.5.9 端口 PC08 功能配置寄存器(PC08_SEL)

地址偏移量：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PC08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PC08_sel	PC08_sel: 端口PC08功能选择. 000 ---- GPIO PC08 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- TIM6_CHA Timer6模块通道2A信号 011 ---- TIM2_ETR Timer2模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- UART3_CTS UART3模块CTS信号 111 ---- Reserved Reserved

8.5.5.10 端口 PC09 功能配置寄存器(PC09_SEL)

地址偏移量：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC09_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC09_sel	PC09_sel: 端口PC09功能选择. 000 ---- GPIO PC09 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- TIM4_CHB Timer6模块通道0B信号 011 ---- TIM1_ETR Timer1模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- UART3_RTS UART3模块RTS信号 111 ---- Reserved Reserved

8.5.5.11 端口 PC10 功能配置寄存器(PC10_SEL)

地址偏移量：0xA8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC10_sel	PC10_sel: 端口PC10功能选择. 000 ---- GPIO PC10 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- PCA_CH2 PCA模块通道2捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.12 端口 PC11 功能配置寄存器(PC11_SEL)

地址偏移量：0xAC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC11_sel	PC11_sel: 端口PC11功能选择. 000 ---- GPIO PC11 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- PCA_CH3 PCA模块通道3捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.13 端口 PC12 功能配置寄存器(PC12_SEL)

地址偏移量：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC12_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC12_sel	PC12_sel: 端口PC12功能选择. 000 ---- GPIO PC12 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- PCA_CH4 PCA模块通道4捕获/比较信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.14 端口 PC13 功能配置寄存器(PC13_SEL)

地址偏移量：0xB4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC13_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC13_sel	PC13_sel: 端口PC13功能选择. 000 ---- GPIO PC13 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- TIM3_CH1B Timer3模块通道1B信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.15 端口 PC14 功能配置寄存器(PC14_SEL)

地址偏移量：0xB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC14_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC14_sel	PC14_sel: 端口PC14功能选择. 000 ---- GPIO PC14 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.5.16 端口 PC15 功能配置寄存器(PC15_SEL)

地址偏移量：0xBC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PC15_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PC15_sel	PC15_sel: 端口PC15功能选择. 000 ---- GPIO PC15 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6 端口 PD 功能选择寄存器

8.5.6.1 端口 PD00 功能配置寄存器(PD00_SEL)

地址偏移量：0xC0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD00_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD00_sel	端口PD00功能选择. 000 ---- GPIO PD00 001 ---- Reserved Reserved 010 ---- SPI1_CS SPI1模块主机模式片选信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.2 端口 PD01 功能配置寄存器(PD01_SEL)

地址偏移量：0xC4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD01_sel	端口PD01功能选择. 000 ---- GPIO PD01 001 ---- Reserved Reserved 010 ---- SPI1_SCK SPI1模块时钟信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.3 端口 PD02 功能配置寄存器(PD02_SEL)

地址偏移量：0xC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD02_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD02_sel	PD02_sel: 端口PD02功能选择. 000 ---- GPIO PD02 001 ---- PCA_ECI PCA模块外部时钟输入信号 010 ---- Reserved Reserved 011 ---- TIM1_ETR Timer1模块外部时钟输入信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.4 端口 PD03 功能配置寄存器(PD03_SEL)

地址偏移量：0xCC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD03_sel	端口PD03功能选择. 000 ---- GPIO PD03 001 ---- UART1_CTS UART1模块CTS信号 010 ---- SPI1_MISO SPI1模块主机输入从机输出数据信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.5 端口 PD04 功能配置寄存器(PD04_SEL)

地址偏移量：0xD0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD04_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD04_sel	端口PD04功能选择. 000 ---- GPIO PD04 001 ---- UART1_RTS UART1模块RTS信号 010 ---- SPI1_MOSI SPI1模块主机输出从机输入数据信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.6 端口 PD05 功能配置寄存器(PD05_SEL)

地址偏移量：0xD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PD05_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PD05_sel	端口PD05功能选择. 000 ---- GPIO PD05 001 ---- UART1_TXD UART1模块TXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.7 端口 PD06 功能配置寄存器(PD06_SEL)

地址偏移量：0xD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD06_sel	PD06_sel: 端口PD06功能选择. 000 ---- GPIO PD06 001 ---- UART1_RXD UART1模块RXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.8 端口 PD07 功能配置寄存器(PD07_SEL)

地址偏移量：0xDC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD07_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD07_sel	PD07_sel: 端口PD07功能选择. 000 ---- GPIO PD07 001 ---- UART1_TXD UART1模块TXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.9 端口 PD08 功能配置寄存器(PD08_SEL)

地址偏移量：0xE0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD08_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD08_sel	PD08_sel: 端口PD08功能选择. 000 ---- GPIO PD08 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.10 端口 PD09 功能配置寄存器(PD09_SEL)

地址偏移量：0xE4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD09_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD09_sel	PD09_sel: 端口PD09功能选择. 000 ---- GPIO PD09 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.11 端口 PD10 功能配置寄存器(PD10_SEL)

地址偏移量：0xE8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD10_sel	PD10_sel: 端口PD10功能选择. 000 ---- GPIO PD10 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.12 端口 PD11 功能配置寄存器(PD11_SEL)

地址偏移量：0xEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD11_sel	PD11_sel: 端口PD11功能选择. 000 ---- GPIO PD11 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.13 端口 PD12 功能配置寄存器(PD12_SEL)

地址偏移量：0xF0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD12_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD12_sel	PD12_sel: 端口PD12功能选择. 000 ---- GPIO PD12 001 ---- Reserved Reserved 010 ---- UART2_RTS UART2模块RTS信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.14 端口 PD13 功能配置寄存器(PD13_SEL)

地址偏移量：0xF4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD13_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD13_sel	PD13_sel: 端口PD13功能选择. 000 ---- GPIO PD13 001 ---- UART2_RXD UART2模块RXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.15 端口 PD14 功能配置寄存器(PD14_SEL)

地址偏移量：0xF8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD14_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD14_sel	PD14_sel: 端口PD14功能选择. 000 ---- GPIO PD14 001 ---- UART2_TXD UART2模块TXD信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.6.16 端口 PD15 功能配置寄存器(PD15_SEL)

地址偏移量：0xFC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PD15_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PD15_sel	PD15_sel: 端口PD15功能选择. 000 ---- GPIO PD15 001 ---- Reserved Reserved 010 ---- UART2_CTS UART2模块CTS信号 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7 端口 PE 功能选择寄存器

8.5.7.1 端口 PE00 功能配置寄存器(PE00_SEL)

地址偏移量：0x1000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE00_sel	端口PE00功能选择. 000 ---- GPIO PE00 001 ---- TIM1_CHA Timer1模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.2 端口 PE01 功能配置寄存器(PE01_SEL)

地址偏移量：0x1004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE01_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE01_sel	端口PE01功能选择. 000 ---- GPIO PE01 001 ---- TIM2_CHA Timer2模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.3 端口 PE02 功能配置寄存器(PE02_SEL)

地址偏移量：0x1008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved-													PE02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE02_sel	端口PE02功能选择. 000 ---- GPIO PE02 001 ---- PCA_ECI PCA模块外部时钟输入信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.4 端口 PE03 功能配置寄存器(PE03_SEL)

地址偏移量：0x100C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PE03_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PE03_sel	端口PE03功能选择. 000 ---- GPIO PE03 001 ---- PCA_CH0 PCA模块通道0捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.5 端口 PE04 功能配置寄存器(PE04_SEL)

地址偏移量：0x1010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PE04_sel	
														rw	

位	标记	功能描述
31:3	Reserved	保留
2:0	PE04_sel	端口PE04功能选择. 000 ---- GPIO PE04 001 ---- PCA_CH1 PCA模块通道1捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.6 端口 PE05 功能配置寄存器(PE05_SEL)

地址偏移量：0x1014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE05_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE05_sel	端口PE05功能选择. 000 ---- GPIO PE05 001 ---- PCA_CH2 PCA模块通道2捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.7 端口 PE06 功能配置寄存器(PE06_SEL)

地址偏移量：0x1018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE06_sel	端口PE06功能选择. 000 ---- GPIO PE06 001 ---- PCA_CH3 PCA模块通道3捕获/比较信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.8 端口 PE07 功能配置寄存器(PE07_SEL)

地址偏移量：0x101C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE07_sel	端口PE07功能选择. 000 ---- GPIO PE07 001 ---- TIM3_ETR Timer3模块外部时钟输入信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.9 端口 PE08 功能配置寄存器(PE08_SEL)

地址偏移量：0x1020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE08_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE08_sel	PE08_sel: 端口PE08功能选择. 000 ---- GPIO PE08 001 ---- TIM3_CH0B Timer3模块通道0B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.10 端口 PE09 功能配置寄存器(PE09_SEL)

地址偏移量：0x1024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PE09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PE09_sel	端口PE09功能选择. 000 ---- GPIO PE09 001 ---- TIM3_CH0A Timer3模块通道0A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.11 端口 PE10 功能配置寄存器(PE10_SEL)

地址偏移量：0x1028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE10_sel	PE10_sel: 端口PE10功能选择. 000 ---- GPIO PE10 001 ---- TIM3_CH1B Timer3模块通道1B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.12 端口 PE11 功能配置寄存器(PE11_SEL)

地址偏移量：0x102C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE11_sel	PE11_sel: 端口PE11功能选择. 000 ---- GPIO PE11 001 ---- TIM3_CH1A Timer3模块通道1A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.13 端口 PE12 功能配置寄存器(PE12_SEL)

地址偏移量：0x1030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE12_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE12_sel	PE12_sel: 端口PE12功能选择. 000 ---- GPIO PE12 001 ---- TIM3_CH2B Timer3模块通道2B信号 010 ---- SPI0_CS SPI0模块主机模式片选信号 011 ---- UART3_CTS UART3模块CTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.14 端口 PE13 功能配置寄存器(PE13_SEL)

地址偏移量：0x1034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE13_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE13_sel	PE13_sel: 端口PE13功能选择. 000 ---- GPIO PE13 001 ---- TIM3_CH2A Timer3模块通道2A信号 010 ---- SPI0_SCK SPI0模块时钟信号 011 ---- UART3_RTS UART3模块RTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.15 端口 PE14 功能配置寄存器(PE14_SEL)

地址偏移量：0x1038

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE14_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE14_sel	PE14_sel: 端口PE14功能选择. 000 ---- GPIO PE14 001 ---- TIM3_CH0B Timer3模块通道0B信号 010 ---- SPI0_MISO SPI0模块主机输入从机输出数据信号 011 ---- UART3_RXD UART3模块RXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.7.16 端口 PE15 功能配置寄存器(PE15_SEL)

地址偏移量：0x103C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PE15_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PE15_sel	PE15_sel: 端口PE15功能选择. 000 ---- GPIO PE15 001 ---- TIM3_BK Timer3模块刹车信号 010 ---- SPI0_MOSI SPI0模块主机输出从机输入数据信号 011 ---- UART3_TXD UART3模块TXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8 端口 PF 功能选择寄存器

8.5.8.1 端口 PF00 功能配置寄存器(PF00_SEL)

地址偏移量：0x1040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF00_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF00_sel	端口PF00功能选择. 000 ---- GPIO PF00 001 ---- I2C0_SDA I2C0模块数据信号 010 ---- Reserved Reserved 011 ---- UART1_TXD UART1模块TXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.2 端口 PF01 功能配置寄存器(PF01_SEL)

地址偏移量：0x1044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF01_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF01_sel	PF01_sel: 端口PF01功能选择. 000 ---- GPIO PF01 001 ---- I2C0_SCL I2C0模块时钟信号 010 ---- TIM4_CHB Timer6模块通道0B信号 011 ---- UART1_RXD UART1模块RXD信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.3 端口 PF02 功能配置寄存器(PF02_SEL)

地址偏移量：0x1048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF02_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF02_sel	PF02_sel: 端口PF02功能选择. 000 ---- GPIO PF02 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.4 端口 PF03 功能配置寄存器(PF03_SEL)

地址偏移量：0x104C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF03_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF03_sel	PF03_sel: 端口PF03功能选择. 000 ---- GPIO PF03 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.5 端口 PF04 功能配置寄存器(PF04_SEL)

地址偏移量：0x1050

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF04_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF04_sel	PF04_sel: 端口PF04功能选择. 000 ---- GPIO PF04 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.6 端口 PF05 功能配置寄存器(PF05_SEL)

地址偏移量：0x1054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF05_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF05_sel	PF05_sel: 端口PF05功能选择. 000 ---- GPIO PF05 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.7 端口 PF06 功能配置寄存器(PF06_SEL)

地址偏移量：0x1058

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF06_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF06_sel	PF06_sel: 端口PF06功能选择. 000 ---- GPIO PF06 001 ---- I2C1_SCL I2C1模块时钟信号 010 ---- Reserved Reserved 011 ---- UART0_CTS UART0模块CTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.8 端口 PF07 功能配置寄存器(PF07_SEL)

地址偏移量：0x105C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF07_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF07_sel	PF07_sel: 端口PF07功能选择. 000 ---- GPIO PF07 001 ---- I2C1_SDA I2C1模块数据信号 010 ---- Reserved Reserved 011 ---- UART0_RTS UART0模块RTS信号 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.9 端口 PF09 功能配置寄存器(PF09_SEL)

地址偏移量：0x1064

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PF09_sel		
													rw		

位	标记	功能描述
31:3	Reserved	保留
2:0	PF09_sel	PF09_sel: 端口PF09功能选择. 000 ---- GPIO PF09 001 ---- TIM0_CHA Timer0模块通道A信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.10 端口 PF10 功能配置寄存器(PF10_SEL)

地址偏移量：0x1068

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF10_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF10_sel	PF10_sel: 端口PF10功能选择. 000 ---- GPIO PF10 001 ---- TIM0_CHB Timer0模块通道B信号 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

8.5.8.11 端口 PF11 功能配置寄存器(PF11_SEL)

地址偏移量：0x106C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PF11_sel
															rw

位	标记	功能描述
31:3	Reserved	保留
2:0	PF11_sel	PF11_sel: 端口PF11功能选择. 000 ---- GPIO PF11 001 ---- Reserved Reserved 010 ---- Reserved Reserved 011 ---- Reserved Reserved 100 ---- Reserved Reserved 101 ---- Reserved Reserved 110 ---- Reserved Reserved 111 ---- Reserved Reserved

9 I2C 总线（I2C）

9.1 简介

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器（I2Cx_STAT)能反映 I2C 总线控制器和 I2C 总线的状态。

9.2 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

9.3 协议描述

I2C 总线使用连接设备的"SCL"(串行时钟总线)和"SDA"(串行数据总线)来传送信息。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节（最高位 MSB 开始传输），后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

9.3.1 I2C 总线上数据传输

通常标准 I2C 传输协议包含四个部分：起始(S)或重复起始信号(Sr)，从机地址及读写位，传输数据，停止信号(P)。

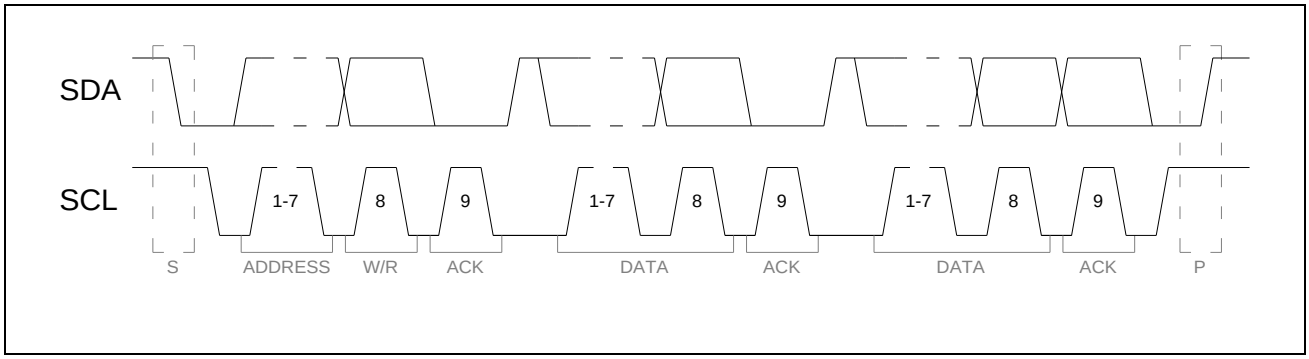


图 9-1 I2C 传输协议

- 起始信号、重复起始信号、停止信号

  当总线处于空闲状态下（SCL 和 SDA 线同时为高），SDA 线上出现由高到低的信号，表明总线上产生了起始信号。

  当两个起始信号之间没有停止信号时，即产生了重复起始信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信（例如：从写入设备到从设备读出）而不释放总线。

  当 SCL 线为高时，SDA 线上出现由低到高的信号，被定义为停止信号。主机向总线发出停止信号结束数据传送。

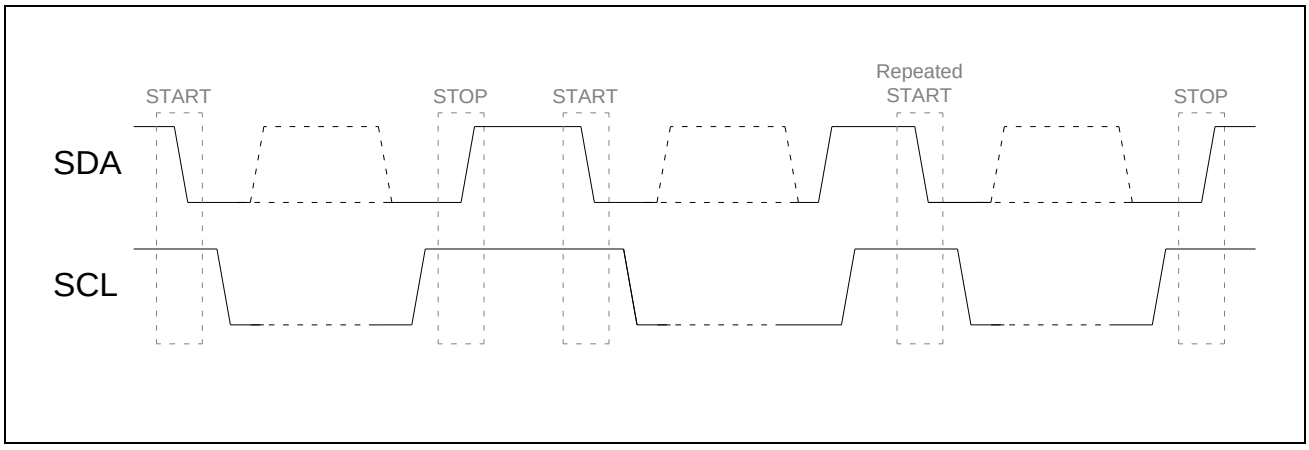


图 9-2 START 和 STOP 条件

- 从机地址及读写位

  当起始信号产生后，主机立即传输数据的第一字节：7 位从机地址 + 读写位，读写位控制从机的数据传输方向（0：写；1：读）。被主机寻址的从机会通过在第 9 个 SCL

时钟周期将 SDA 置为低电平作为应答。

- 传输数据

  数据传输过程中，一个 SCL 时钟脉冲传输一个数据位，且 SDA 线只有在 SCL 为低时才可以改变。

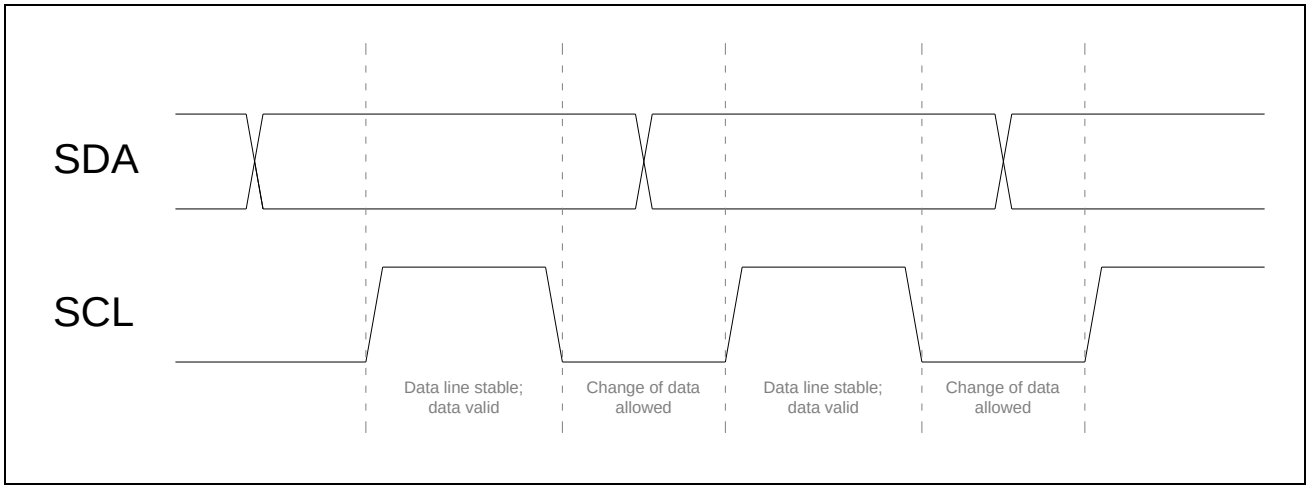


图 9-3 I2C 总线上位传输

9.3.2 I2C 总线上的应答

每传输一个字节，后面跟随一个应答位。通过将 SDA 线拉低，来允许接收端回应发送端。ACK 为一个低电平信号，当时钟信号为高时，SDA 保持低电平则表明接收端已成功接收到发送端的数据。

当主机作为发送器件时，如果从机上产生无响应信号(NACK)，主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。当主机作为接收器件时，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

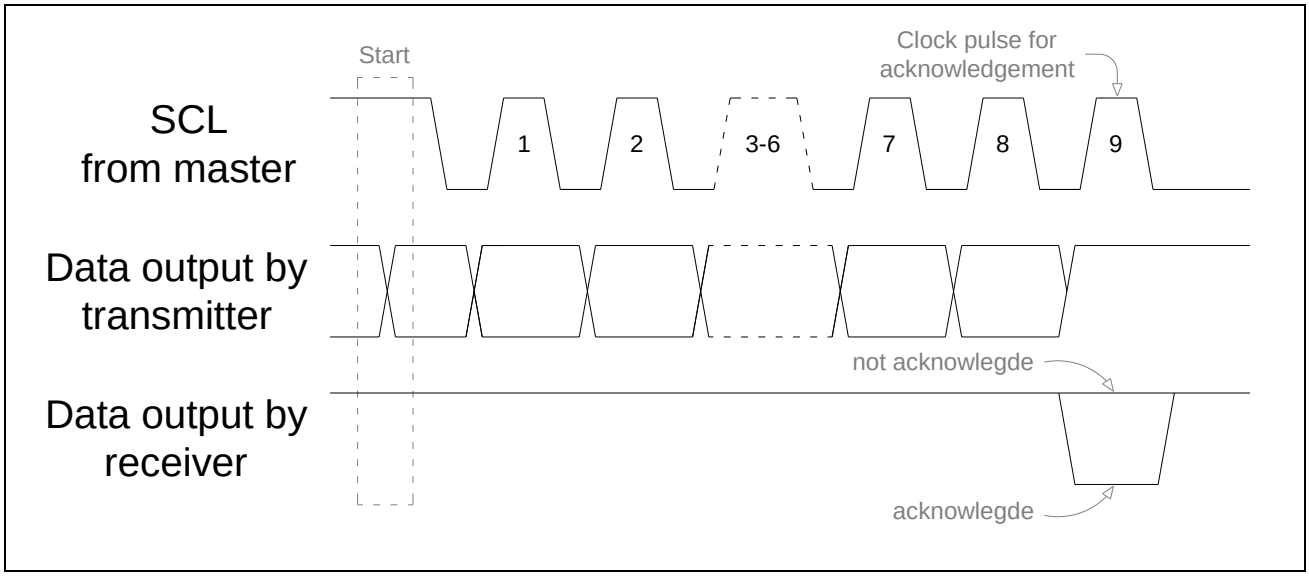


图 9-4 I2C 总线上应答信号

9.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：SCL 线上的同步和 SDA 线上的仲裁

- SCL 线上的同步（时钟同步）

  由于 I2C 总线具有线“与”的逻辑功能，SCL 线上只要有一个节点发送低电平，总线上就表现低电平。当所有的节点都发送高电平时，总线才能表现为高电平。所以，时钟低电平的时间由时钟电平期最长的器件决定，而时钟的高电平时间由时钟高电平期最短的器件决定。由于 I2C 这种特性，当多个主机同时发送时钟信号时，在总线上表示的是统一的时钟信号。如果从机希望主机降低传送速度可以通过将 SCL 主动拉低延长其低电平时间来通知主机，当主机在准备下一次传送时发现 SCL 的电平被拉低时进行等待，直到从机完成操作并释放 SCL 线的控制权。

- SDA 线上的仲裁

  SDA 线上的仲裁也是由于 I2C 总线具有线“与”的逻辑功能。主机在发送数据后，通过比较总线上的数据来决定是否退出竞争。丢失仲裁的主机立即切换到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。仲裁失败的主机继续输出时钟脉冲（在 SCL 上），直到发送完当前的串行字节。通过这种原理可以保证 I2C 总线在多个主机企图控制总线时保证数据的不丢失。

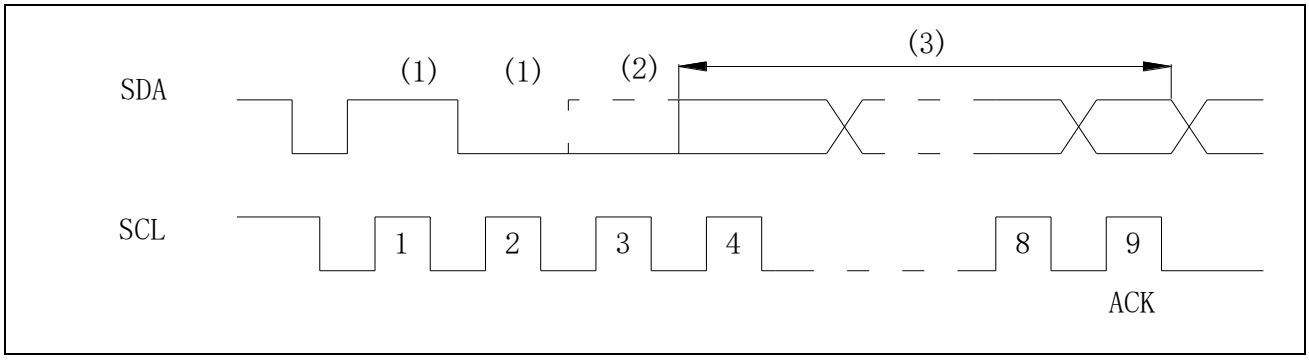


图 9-5 I2C 总线上的仲裁

a） 另一器件发送串行数据；

b） 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1（虚线）。仲裁丢失，I2C 进入从接收模式；

c） 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为下个字节的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。

9.4 功能描述

I2C 总线使用双线在连接到总线"SCL"（串行时钟线）和"SDA"（串行数据线）的设备间传送信息。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。由于只有无方向端口，I2C 组件需要使用到引脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。I2C 总线状态能在状态寄存器中进行查询。

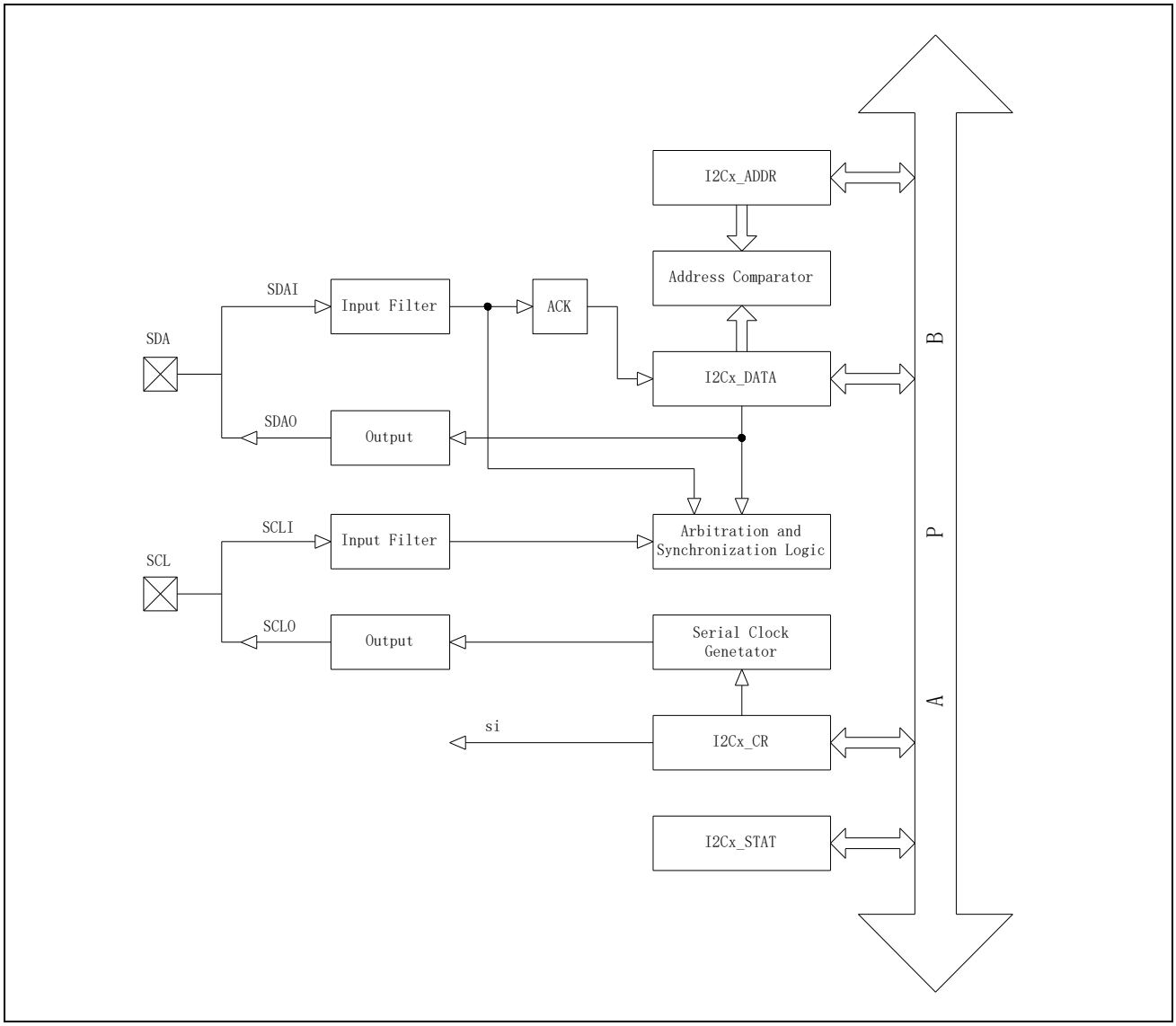


图 9-6 I2C 功能模块图

9.4.1 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 PCLK 信号的频率关系为 Fscl = Fpclk / 8 / (I2Cx_TM.tm+1)，其中 I2Cx_TM.tm 应大于 0。

下表列出了 PCLK 频率与 I2Cx_TM.tm 组合时，SCL 信号的输出频率值。

PCLK（KHz）	I2Cx_TM.tm						
	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

表 9-1 I2C 时钟信号波特率

9.4.2 输入滤波器

输入信号与 PCLK 同步，低于 PCLK 周期的尖锋脉冲信号会被滤除。每个滤波器由 3 个触发器组成。第一个触发器用来直接锁存输入信号，并将数据载入由另外两个构成的移位寄存器中。当第二和第三个触发器的状态是"11"或"00"时，内部的滤除信号会各自被置 1 或置 0。

9.4.3 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用"I2Cx_ADDR"寄存器对自己的从机地址进行编程。并且会根据"I2Cx_ADDR"寄存器的"i2cadr"位与首次接收到的 8 位字节或广播地址（0x00）相比较。如果任何一者相同，"I2Cx_CR"寄存器的"si"位会被置 1 并产生一个中断请求。

9.4.4 应答标志位

"I2Cx_CR"寄存器的"aa"标志位为应答标志位。当"aa"位为 1 时，I2C 模块收到数据后回应答位，当"aa"位为 0 时，I2C 模块收到数据后回非应答位。

9.4.5 中断产生器

"I2Cx_CR"寄存器的"si"标志位为中断标志位。每当状态寄存器（I2Cx_STAT）的值产生变化时（除变为 0xF8 外），"si"标志位都会被置 1。当产生中断时，通过查询状态寄存器（I2Cx_STAT）可获知 I2C 总线的状态，以确定中断的实际来源。为了进行下一步的操作，"si"标志位必须通过软件清零。

9.4.6 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbps 而在高速模式下可达 400Kbps，在超高速模式下可达 1Mbps，并且可以在四种模式下工作：主机发送模式、主机接收模式、从机接收模式、从机发送模式。 还有一种特殊模式广播呼叫模式，其操作方式与从机接收模式类似。

- 主机发送模式

  主机发送多个字节到从机，主机产生时钟，所以需要在 I2Cx_TM 中填入设定值。主机发送模式中需要将 I2Cx_CR.sta 置 1。当总线空闲时，主机发起一个起始位 START。若成功 I2Cx_CR.si 被置 1。接下来把从机地址和写位（SLA+W）写入 I2Cx_DATA 中，清零"si"位后，总线上发出 SLA+W。

  主机发出 SLA+W 收到从机应答位 ACK 后，"si"被置 1。接下来根据用户定义格式发送数据。所有数据发送完后，将 I2Cx_CR.sto 置 1，清零"si"位后发出 STOP 信号，也可以发送重复起始信号进行新一轮数据传输。

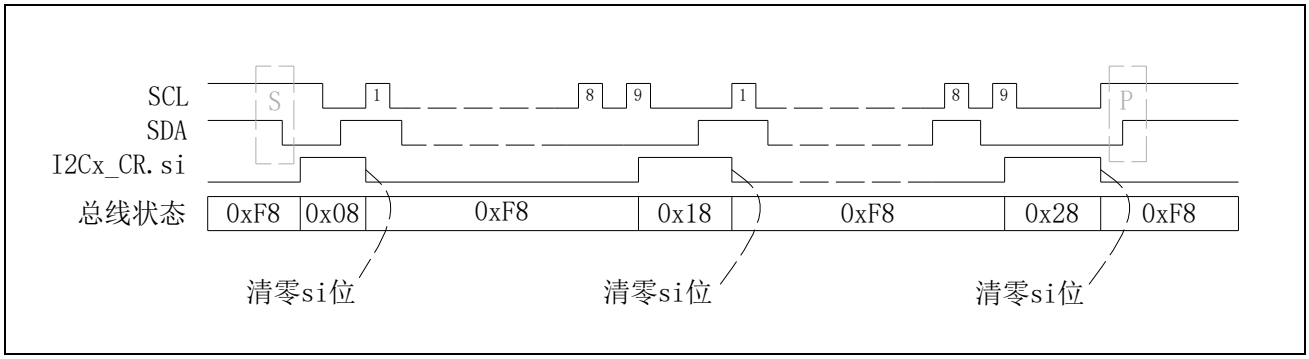


图 9-7 主发送模式数据同步图

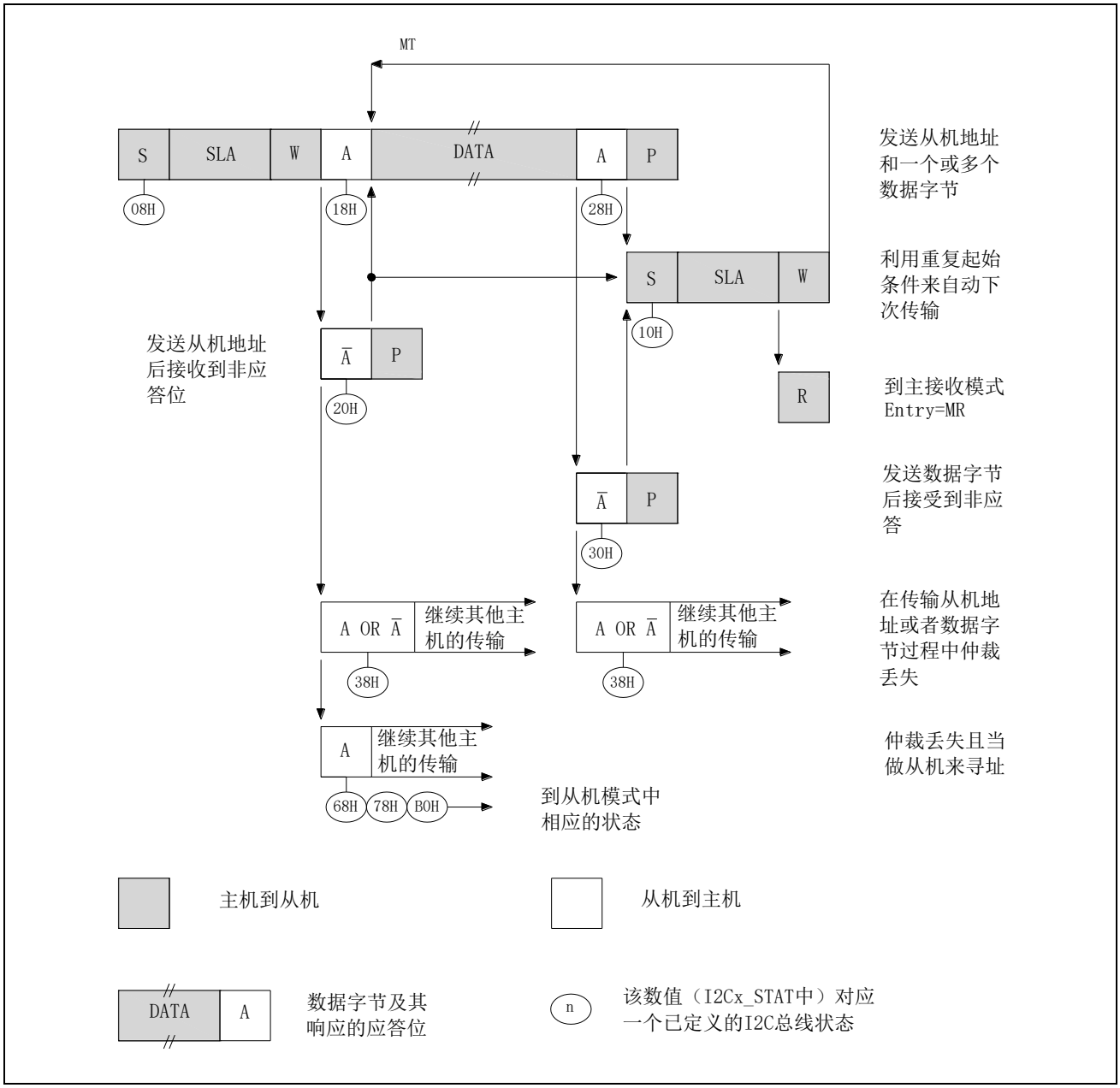


图 9-8 I2C 主机发送状态图

- 主机接收模式

主机接收模式，由从机传输数据。初始化设置与主机发送模式相同，主机发送起始位以后， I2Cx_DATA 应写入从机地址和“读位” (SLA+R)。收到从机应答位 ACK 后 I2Cx_CR.si 被置 1。"si"清 0 后开始接收从机数据，若 I2Cx_CR.aa 为 1，主机收到数据后回应答位；若为 0 主机收到数据后不回应答 NACK。然后主机可以发停止信号或重复起始信号开始下一轮的数据传输。

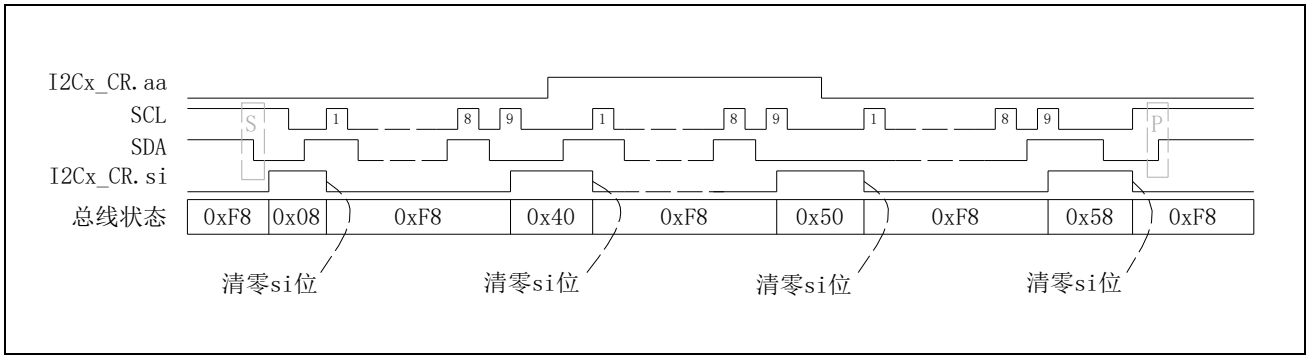


图 9-9 主接收模式数据同步图

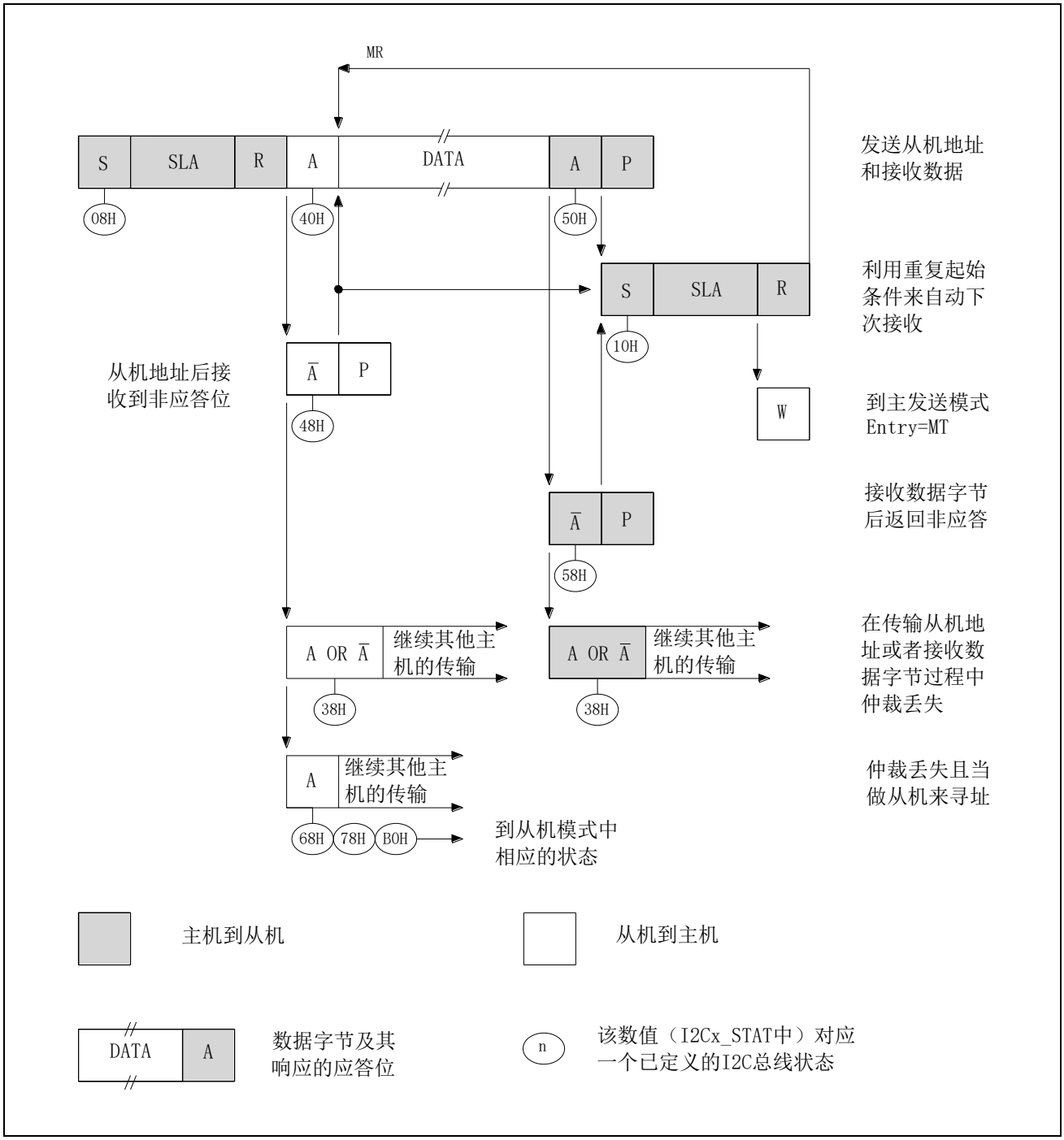


图 9-10 I2C 主机接收状态图

- 从机接收模式

  在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2Cx_ADDR 应写入从机地址，I2Cx_CR.aa 置 1 用以响应主机的寻址。上述初始化后，从机进入空闲模式，等待“写”信号（SLA+W）。若主机仲裁失败，也会直接进入从机接收模式。

  当从机被“写”信号 SLA+W 寻址到后，需要清零"si"位，以便从主机接收数据。如果在传输过程中 I2Cx_CR.aa=0，从机将在下一字节返回无应答位 NACK，从机也将转为未寻址从机，与主机联系终止，不再接收数据，且 I2C_DATA 保持之前接收到的数据。从机地址识别可通过置位"aa"来恢复，这意味着"aa"位可临时将 I2C 模块从 I2C 总线上分离出来。

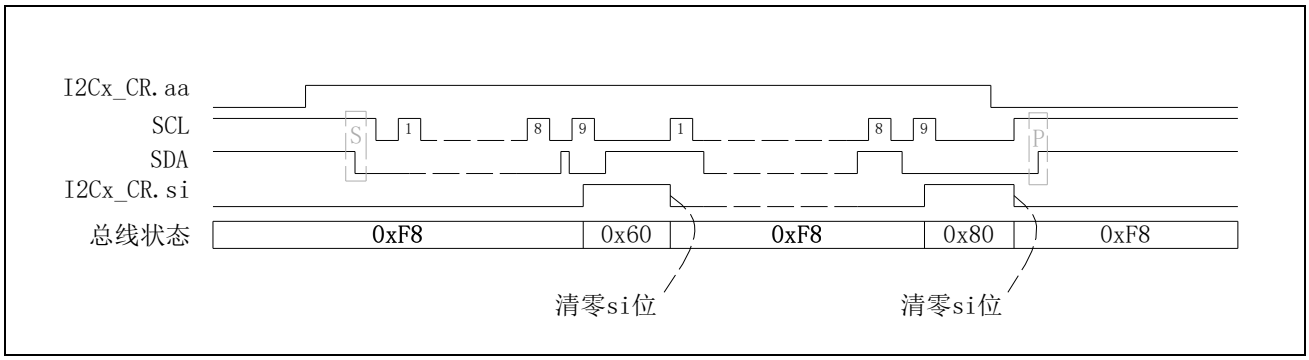


图 9-11　从接收模式数据同步图

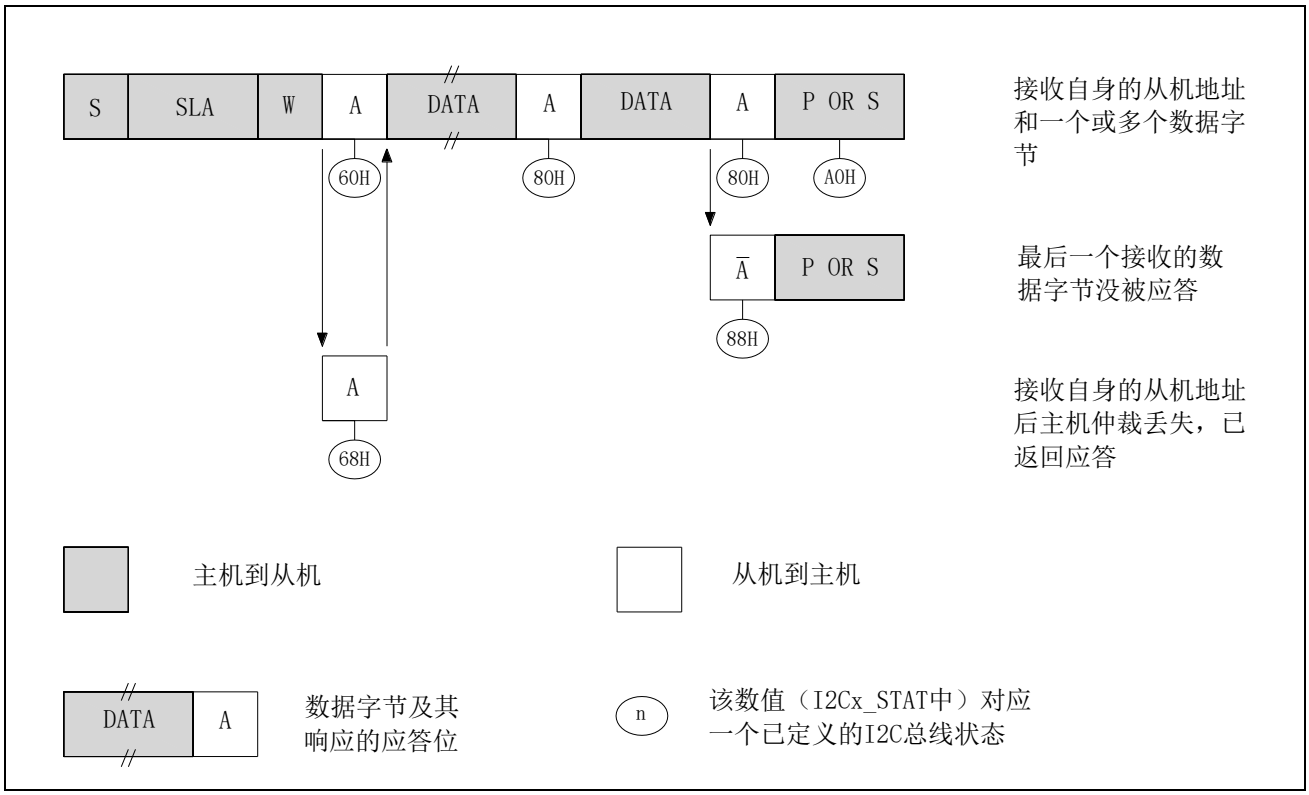


图 9-12　从机接收状态图

- 从机发送模式

  从机发送模式，数据由从机发送给主机。当初始化 I2Cx_ADDR 及 I2Cx_CR.aa 值后，器件等待直到自身地址被“读”信号(SLA+R)寻址。若主机仲裁失败，也可进入从机发送模式。

  当从机被“读”信号 SLA+R 寻址，需要将"si"清零用以向主机发送数据。通常主机接收每字节数据后会返回应答位。

  如果传输过程中 I2Cx_CR.aa 清零 ，从机将发送最后一个字节数据，并在接下去的传输中发送全 1 数据，并将自身变为未寻址从机。

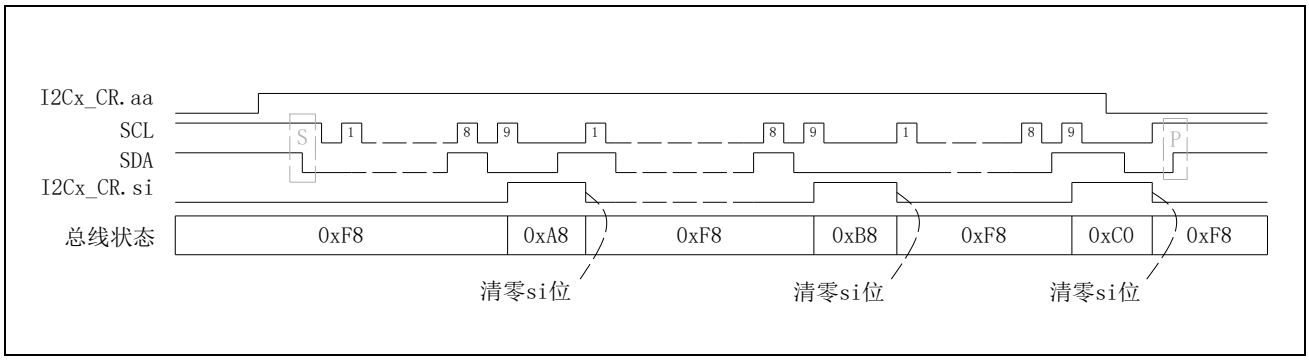


图 9-13　从发送模式数据同步图

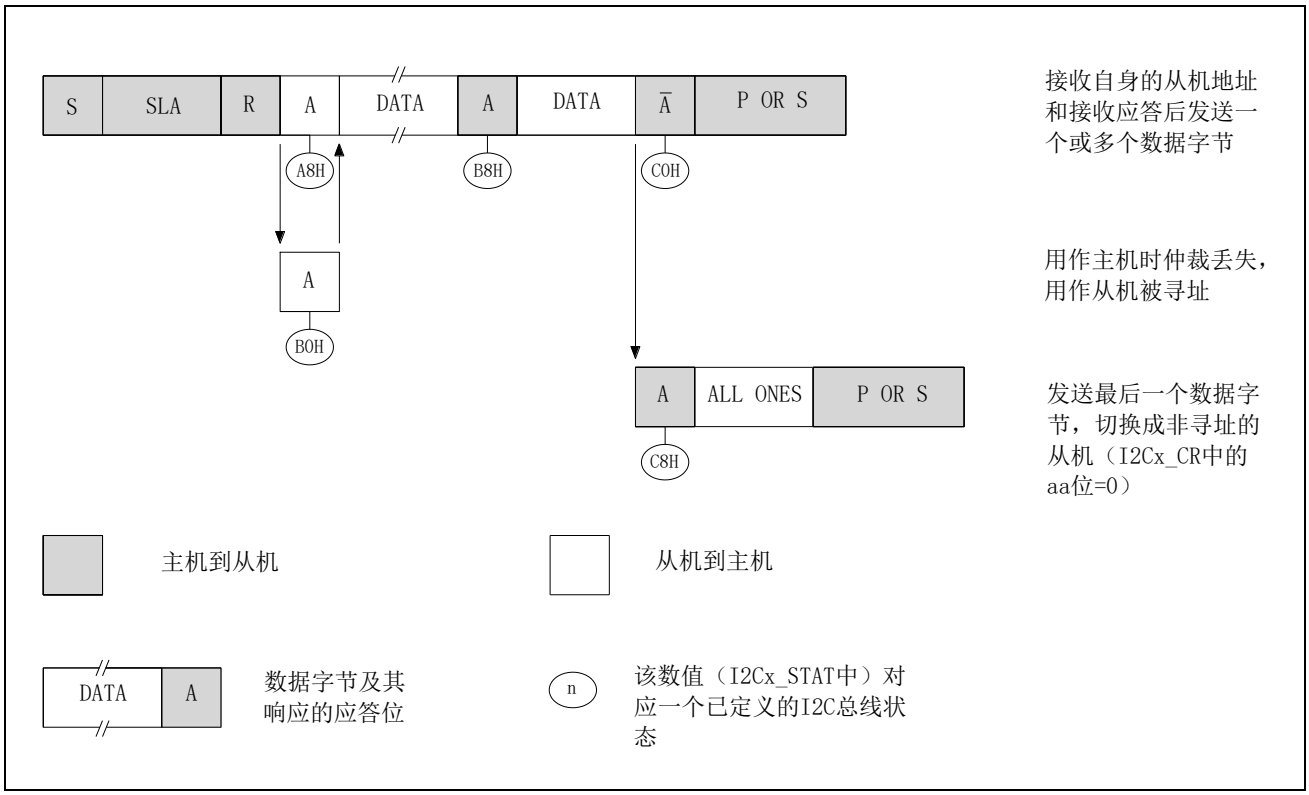


图 9-14　I2C 从机发送状态图

- 广播呼叫模式

  广播呼叫模式是一种特殊的从机接收模式， 寻址方式为 0x00，从机地址和读写都为 0。当 I2Cx_ADDR.GC 及 I2Cx_CR.aa 都为置 1，使能接收广播呼叫模式。在该模式下 I2Cx_STAT 值与普通从机接收模式 I2Cx_STAT 值不同。仲裁失败也可能进入广播呼叫模式。

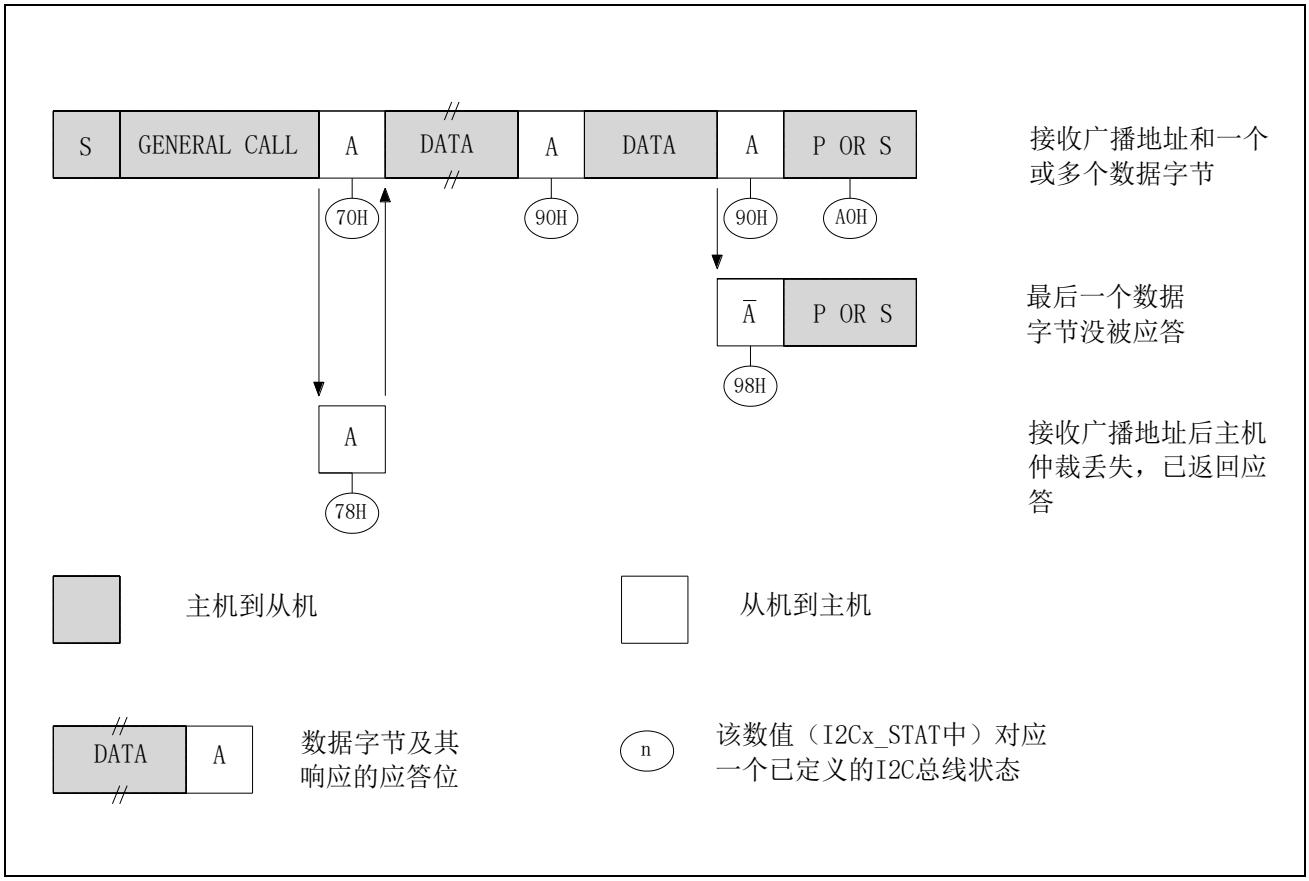


图 9-15 I2C 广播呼叫状态图

9.4.7 状态码表述

I2C 状态寄存器中有两种特殊的状态：F8H 和 00H。

F8H：这个状态码表示没有任何可用的相关信息，因为串行中断标志"si"还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

00H：该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时"si"置位。

状态代码	描述
主发送模式	
08H	已发送起始条件
10H	已发送重复起始条件
18H	已发送 SLA+W，已接收 ACK
20H	已发送 SLA+W，已接收非 ACK
28H	已发送 I2Cx_DATA 中的数据，已接收 ACK
30H	已发送 I2Cx_DATA 中的数据，已接收非 ACK
38H	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	
08H	已发送起始条件
10H	已发送重复起始条件
38H	在非 ACK 中丢失仲裁
40H	已发送 SLA+R，已接收 ACK
48H	已发送 SLA+R，已接收非 ACK
50H	已接收数据字节，ACK 已返回
58H	已接收数据字节，非 ACK 已返回
从接收模式	
60H	已接收自身的 SLA+W，已返回 ACK

68H	主控时在 SLA+读写丢失仲裁，已接收自身的 SLA+W，已返回 ACK
80H	前一次寻址使用自身从地址，已接收数据字节，已返回 ACK
88H	前一次寻址使用自身从地址，已接收数据字节，已返回非 ACK
A0H	静态寻址时，接收到停止条件或重复起始条件
从发送模式	
A8H	已接收自身的 SLA+R，已返回 ACK
B0H	当主机时丢失仲裁，已接收自身 SLA+R，已返回 ACK
B8H	已发送数据，已接收 ACK
C0H	已发送数据字节，已接收非 ACK
C8H	装入的数据字节已被发送，已接收 ACK
广播呼叫模式	
70H	已接收广播地址（0x00），已返回 ACK
78H	主控时在 SLA+读写丢失仲裁，已接收广播地址，已返回 ACK
90H	前一次寻址使用广播地址，已接收数据字节，已返回 ACK
98H	前一次寻址使用广播地址，已接收数据字节，已返回非 ACK
A0H	静态寻址时，接收到停止条件或重复起始条件
其余杂项状态	
F8H	无可用的相关状态信息，si=0
00H	传输过程出现总线错误，或外部干扰使 I2C 进入未定义的状态

表 9-2 I2C 状态码表述

9.5 编程示例

9.5.1 主机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。

Step2：配置 I2Cx_TM，使 SCL 的时钟速率符合应用需求。

Step3：设置 I2Cx_TMRUN 为 1，使能 SCL 时钟发生器。

Step4：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step5：设置 I2Cx_CR.sta 为 1，总线尝试发送 Start 信号。

Step6：等待 I2Cx_CR.si 变为 1，Start 信号已发送到总线上。

Step7：查询 I2Cx_STAT，如果该寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理。

Step8：向 I2Cx_DATA 中写入 SLA+W，设置 I2Cx_CR.si 为 0，发送 SLA+W。

Step9：等待 I2Cx_CR.si 变为 1，SLA+W 已发送到总线上。

Step10：查询 I2Cx_STAT，如果该寄存器值为 0x18，继续执行下一步骤。否则进行出错处理。

Step11：向 I2Cx_DATA 写入待发送的数据，设置 I2Cx_CR.si 为 0，发送数据。

Step12：等待 I2Cx_CR.si 变为 1，数据已发送到总线上。

Step13：查询 I2Cx_STAT，如果该寄存器值为 0x28，继续执行下一步骤。否则进行出错处理。

Step14：如待发送的数据未完成，则跳转到 Step11 继续执行。

Step15：设置 I2Cx_CR.sto 为 1，设置 I2Cx_CR.si 为 0，总线尝试发送 Stop 信号。

Step16：等待 I2Cx_CR.si 变为 1，Stop 信号已发送到总线上。

9.5.2 主机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。

Step2：配置 I2Cx_TM，使 SCL 的时钟速率符合应用需求。

Step3：设置 I2Cx_TMRUN 为 1，使能 SCL 时钟发生器。

Step4：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step5：设置 I2Cx_CR.sta 为 1，总线尝试发送 Start 信号。

Step6：等待 I2Cx_CR.si 变为 1，Start 信号已发送到总线上。

Step7：查询 I2Cx_STAT，如果寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理。

Step8：向 I2Cx_DATA 写入 SLA+R，设置 I2Cx_CR.si 为 0，发送 SLA+R。

Step9：等待 I2Cx_CR.si 变为 1，SLA+R 已发送到总线上。

Step10：查询 I2Cx_STAT，如果寄存器值为 0x40，继续执行下一步骤，否则进行出错处理。

Step11：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step12：设置 I2Cx_CR.si 为 0，从机发送数据，主机根据 I2Cx_CR.aa 发送 ACK 或 NACK。

Step13：等待 I2Cx_CR.si 变为 1，从 I2Cx_DATA 读取已接收到的数据。

Step14：查询到 I2Cx_STAT，如果该寄存器值为 0x50 或 0x58，继续执行下一步骤，否则进行出错处理。

Step15：如果待接收的数据只差最后一个字节，设置 I2Cx_CR.aa 为 0，使能非应答标志。

Step16：如待接收的数据未完成，则跳转到 Step12 继续执行。

Step17：设置 I2Cx_CR.sto 为 1，设置 I2Cx_CR.si 为 0，总线尝试发送 Stop 信号。

Step18：等待 I2Cx_CR.si 变为 1，Stop 信号已发送到总线上。

9.5.3 从机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。

Step2：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step3：配置 I2Cx_ADDR 为从机地址。

Step4：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step5：等待 I2Cx_CR.si 变为 1，被 SLA+W 寻址。

Step6：查询 I2Cx_STAT，如果该寄存器值为 0x60，继续执行下一步骤，否则进行出错处理。

Step7：设置 I2Cx_CR.si 为 0，主机发送数据，从机根据 I2Cx_CR.aa 返回 ACK 或 NACK。

Step8：等待 I2Cx_CR.si 变为 1，从 I2Cx_DATA 中读取已接收到的数据。

Step9：查询 I2Cx_STAT，如果该寄存器值为 0x80，继续执行下一步骤，否则进行出错处理

Step10：如待接收的数据未完成，则跳转到 Step7 继续执行。

Step11：设置 I2Cx_CR.aa 为 0，设置 I2Cx_CR.si 为 0。

9.5.4 从机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。

Step2：设置 I2Cx_CR.ens 为 1，使能 I2C 模块。

Step3：配置 I2Cx_ADDR 为从机地址。

Step4：设置 I2Cx_CR.aa 为 1，使能应答标志。

Step5：等待 I2Cx_CR.si 变为 1，被 SLA+R 寻址。

Step6：查询 I2Cx_STAT，如果该寄存器的值为 0xA8，继续执行下一步骤，否则进行出错处理。

Step7：向 I2Cx_DATA 写入待发送的数据，设置 I2Cx_CR.si 为 0，发送数据。

Step8：等待 I2Cx_CR.si 变为 1，数据已发送到总线上。

Step9：查询 I2Cx_STAT，如果该寄存器的值为 0xB8 或 0xC0 时，继续执行下一步骤，否则进行出错处理。

Step10：如待发送的数据未完成，则跳转到 Step7 继续执行。

Step11：设置 I2Cx_CR.aa 为 0，设置 I2Cx_CR.si 为 0。

9.6 寄存器描述

寄存器列表

I2C0 基地址：0x40000400

I2C1 基地址：0x40004400

偏移量	寄存器名称	访问	寄存器描述
0x00	I2Cx_TMRUN	RW	I2C 波特率计数器使能寄存器.
0x04	I2Cx_TM	RW	I2C 波特率计数器配置寄存器.
0x08	I2Cx_CR	RW	I2C 配置寄存器.
0x0c	I2Cx_DATA	RW	I2C 数据寄存器.
0x10	I2Cx_ADDR	RW	I2C 地址寄存器.
0x14	I2Cx_STAT	RO	I2C 状态寄存器.

表 9-3 寄存器列表

9.6.1 I2C 波特率计数器使能寄存器(I2Cx_TMRUN)

地址偏移量：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31：1	Reserved	
0	tme	波特率计数器使能. 0 – 禁止 1 – 使能

9.6.2 I2C 波特率计数器配置寄存器(I2Cx_TM)

地址偏移量：0x04

复位值：0x0000 0000

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

位	标记	功能描述
31:8	Reserved	
7:0	tm	tm:波特率计数器配置值. Fscl = Fpclk / 8 / (tm+1) ， 其中 tm >0

9.6.3 I2C 配置寄存器(I2Cx_CR)

地址偏移量：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ens	sta	sto	si	aa	Res	h1m
									RW	RW	RW	RW	RW		RW

位	标记	功能描述
31:7	Reserved	
6	ens	I2C 模块使能控制 0 – 禁止 1 – 使能
5	sta	I2C 总线控制 0 – 无功能 1 – 向总线发送 START
4	sto	I2C 总线控制 0 – 无功能 1 – 向总线发送 STOP
3	si	I2C 中断标志 读出 1，已发生 I2C 中断 写入 0，I2C 进行一下步操作
2	aa	应答控制位 0 – 发送 NAK 1 – 发送 ACK
1	Reserved	

0	h1m	I2C 滤波参数配置 0 – 高级滤波，更高的抗干扰性能 1 – 简单滤波，更快的通信速率 注：当 I2Cx_TM 寄存器小于 10 时，只可配置为简单滤波，否则通信速率达不到目标值

9.6.4 I2C 数据寄存器(I2Cx_DATA)

地址偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cdat							
								RW							

位	标记	功能描述
31:8	Reserved	
7:0	i2cdat	I2C 数据寄存器 在 I2C 发送模式下，写入待发送的数据 在 I2C 接收模式下，读出收到的数据

9.6.5 I2C 地址寄存器(I2Cx_ADDR)

地址偏移量：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cadr							GC
								RW							RW

位	标记	功能描述
31:8	Reserved	
7:1	i2cadr	I2C 从机模式地址.
0	GC	广播地址应答使能 0 – 禁止 1 – 使能

9.6.6 I2C 状态寄存器(I2Cx_STAT)

地址偏移量：0x14

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2csta							
								RO							

位	标记	功能描述
31:8	Reserved	
7:0	i2csta	I2C 状态寄存器 状态值的具体定义详见【状态码表述】章节

10 串行外设接口（SPI）

10.1 SPI 简介

SPI 接口是工作于全双工模式下的同步串行数据通信接口，使用 4 个引脚进行通信：MISO、MOSI、SCK、CS/SSN。当 SPI 作为主机时，输出 CS 和 SCK 信号以控制通信过程。当 SPI 作为从机时，在 SSN 和 SCK 信号的控制下进行通信。

10.2 SPI 主要特性

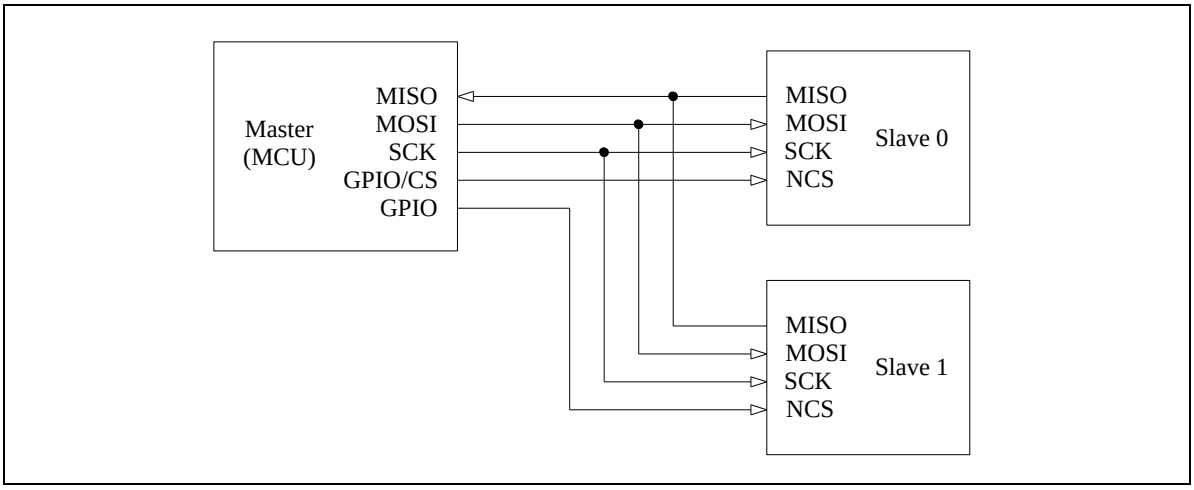
- 支持 SPI 主机模式、SPI 从机模式
- 支持标准四线全双工通信
- 支持配置串行时钟极性和相位
- 主机模式支持 7 种通信速率
- 主机模式最高通信速率为 1/2 PCLK
- 从机模式最高通信速率为 1/4 PCLK
- 帧长度固定为 8 比特，优先传送 MSB
- 支持 DMA 软件/硬件访问

10.3 SPI 功能描述

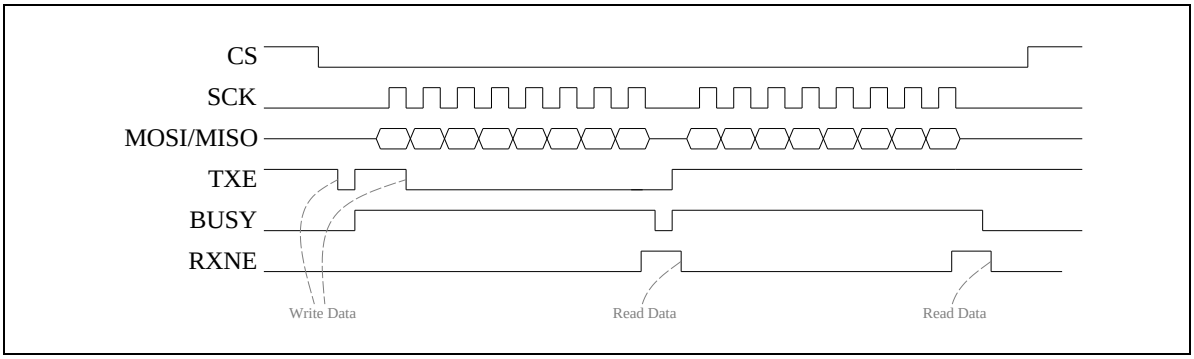
10.3.1 SPI 主机模式

每个数据帧的长度固定为 8 比特，发送数据的第一位固定为 MSB。设置 SPIx_CR.mstr 为 1，则 SPI 接口工作于主机模式。将数据写入 SPIx_Data 寄存器，即可启动 SPI 传输，SCK 引脚会自动产生串行时钟；在串行时钟边沿，移位寄存器中的数据被发送到 MOSI 引脚，MISO 引脚的数据被接收到移位寄存器中。SCK 引脚输出时钟的频率由 SPIx_CR[spr2:spr0]进行控制，其输出频率的范围为 PCLK /2~PCLK/128；CS 引脚的输出电平由 SPIx_SSN.ssn 进行控制，GPIO 引脚的输出电平由 GPIO 相关寄存器控制。

主机模式的典型应用框图如下所示。

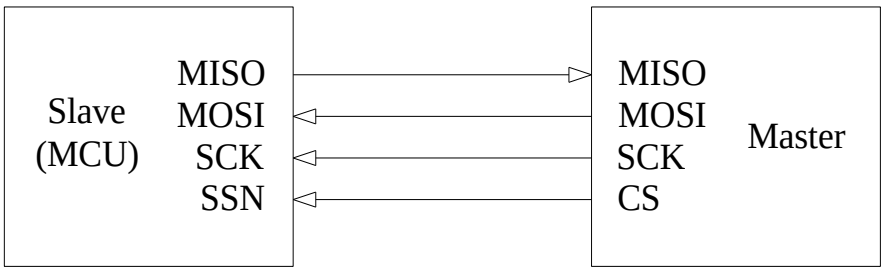


主机模式的通信示意如下图所示，其中 CPOL=0，CPHA=0。



10.3.2 SPI 从机模式

每个数据帧的长度固定为 8 比特，接收数据的第一位固定为 MSB。设置 SPIx_CR.mstr 为 0，则 SPI 接口工作于从机模式。该模式下，SCK 引脚作为输入脚且串行时钟来自于外部主机；SSN 引脚作为输入脚且片选信号来自外部主机或固定为低电平。SSN 引脚详见 GPIO 章节辅助寄存器。从机模式的典型应用框图如下所示。



当 SPI 从机从 SPI 主机接收到一字节数据时，RXNE 位会被置高；用户程序应尽快读取收到的数据。从机模式接收数据的通信时序如下所示，其中 CPOL=0，CPHA=0。

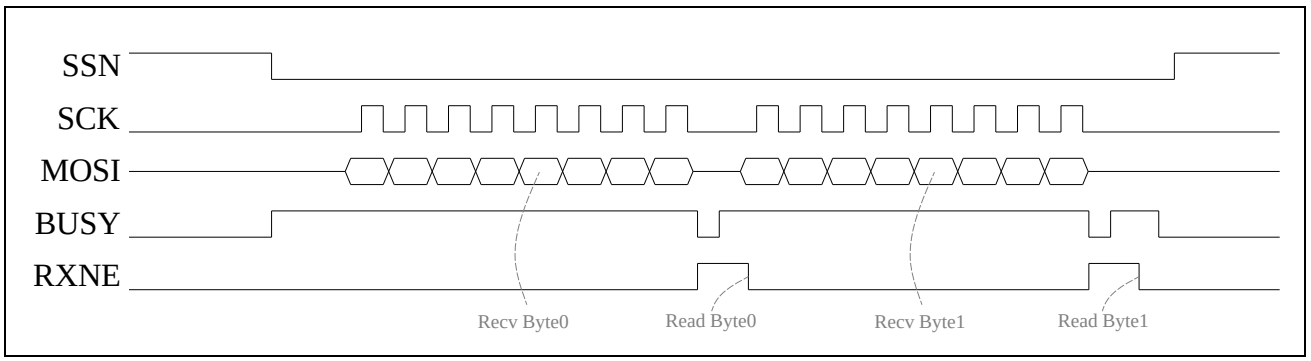


图 10-1 从机接收示意图

当 SPI 从机需要发送数据到主机时，在主机拉低 NSS 之前应依次进行如下五步操作：设置 PERI_RESET0.SPIx 位为 0、设置 PERI_RESET0.SPIx 位为 1、配置 SPI 通信参数、向 SPIx_DATA 寄存器中写入待发送的第一字节数据；在 NSS 拉低后，每当查询到 TXE 标志为 1 时，应尽快向 SPIx_DATA 寄存器中写入待发送的后续数据。从机模式发送数据的通信时序如下所示，其中 CPOL=0，CPHA=0。

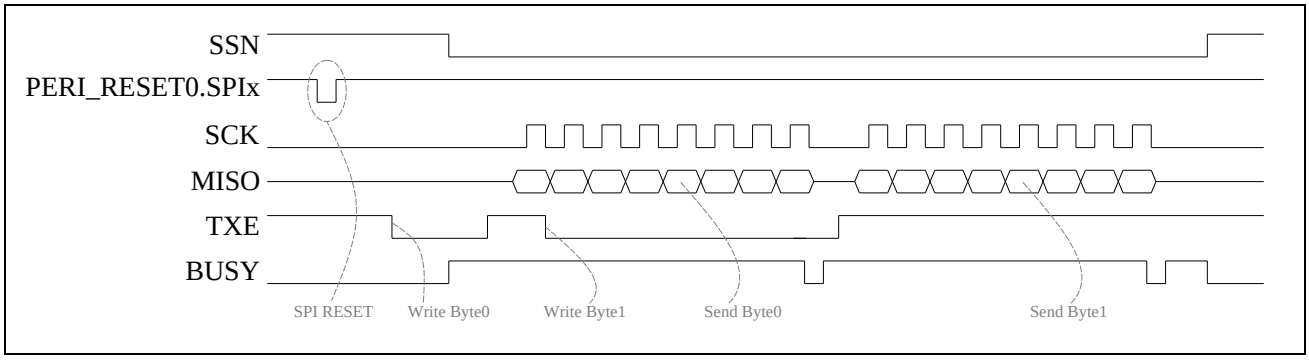


图 10-2 从机发送示意图

10.3.3 SPI 数据帧格式

SPI 接口帧格式取决于时钟极性位 CPOL 和时钟相位位 CPHA 的配置。

当 CPOL 为 0 时，SCK 线空闲状态为低电平。当 CPOL 为 1 时，SCK 线空闲状态为高电平。当 CPHA 为 0 时，数据会在第一个 SCK 时钟转换信号跳变时被采样。当 CPHA 为 1 时，数据会在第二个 SCK 时钟信号跳变时被采样。

SPI 接口主机帧格式如下图所示。

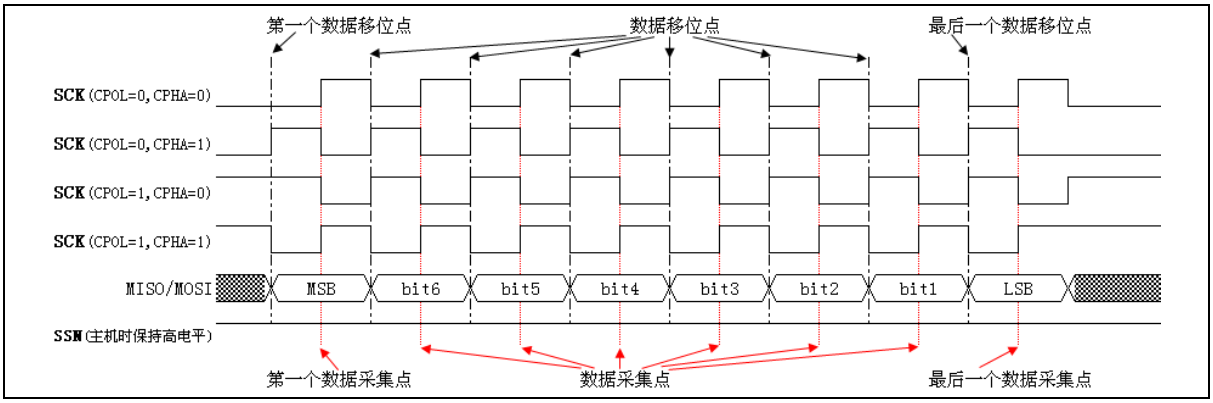


图 10-3 主机模式帧格式

SPI 接口从机帧格式如下图所示。

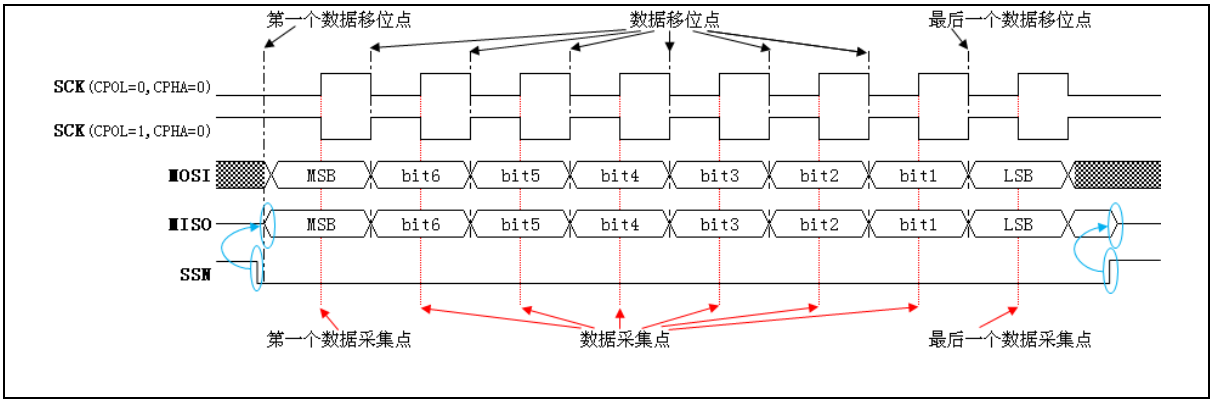


图 10-4 从机 CPHA 为 0 时数据帧格式

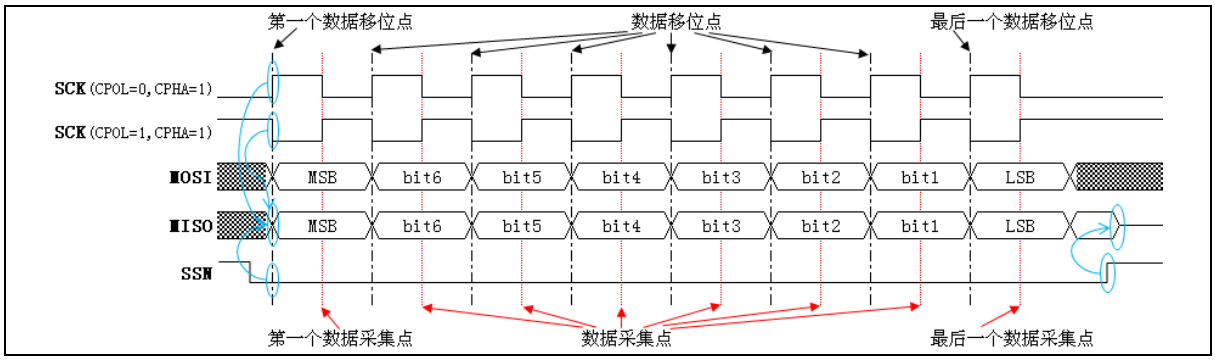


图 10-5 从机 CHPA 为 1 时数据帧格式

10.3.4 SPI 状态标志及中断

SPI 在工作中会产生如下四种状态标志，其产生条件及清除方法如下所示。

- 当发送缓冲器（SPIx_Data）里的数据被转移到发送移位寄存器后，SPIx_STAT.txe 会被硬件置位，说明发送缓冲器为空，可以写入下一笔数据。向 SPIx_DATA 写入数据即可清除该标志位。
- 当接收移位寄存器里的数据被转移到接收缓冲器（SPIx_Data）时， SPIx_STAT.rxne 会被硬件置位，说明接收缓冲器非空，需要用户尽快读取该数据。从 SPIx_DATA 读出数据即可清除该标志位。
- 当 SPI 工作于主机模式且外部 SSN 输入为低电平，SPIx_STAT.mdf 会被硬件置位，说明有其它 SPI 主机在占用总线。当 SSN 输入为高电平时，SPIx_STAT.mdf 会被硬件自动清除。

- 当 SPI 工作于从机模式，如果正在进行数据传输时 SSN 管脚被拉高，则 SPIx_STAT.sserr 会被置位。设置 SPIx_CR.spen 为 0 可以清除该标志。

如果允许 SPI 中断 (SPIx_CR2.int_en=1)，以下三种情况均可产生中断：

- SPI 发送缓冲器为空，即 SPIx_STAT.txe 为 1
- SPI 接收缓冲器非空，即 SPIx_STAT.rxne 为 1
- SPI 的主机模式错误，即 SPIx_STAT.mdf 为 1

在中断服务程序中需要对 SPIx_ICLR 写入 0x00 以清除内部中断标志。

10.3.5 SPI 多机系统配置说明

- 当 SPI 模块作为主机且工作于单主机系统时，可通过 SPI_CS 管脚或 GPIO 管脚控制从机。当选择 SPI_CS 管脚作为从机的片选信号时，设置 SPIx_SSN.ssn 为 0 即可选中从机，设置 SPIx_SSN.ssn 为 1 即可释放从机。当选择 GPIO 管脚作为从机的片选信号时，设置 GPIOx_OUT 寄存器相应的比特为 0 即可选中从机，设置 GPIOx_OUT 寄存器相应的比特为 1 即可释放从机。
- 当 SPI 模块从为从机时，根据需要配置 SPI_SSN 的来源（详见 GPIO 端口辅助控制器）。当 SSN 为低时，即可选中本从机以进行通信；当 SSN 为高时，则本机处于未选中态。
- 当 SPI 模式工作于多主机多从机时，所有的从机片选信号都通过 GPIO 管脚连接，主机还必须通过 GPIO 管脚和其他主机的 SSN 信号相连，来监测总线是否被占用。下图所示的 Master0 需要进行通信的操作方法为：等待 Master0.SSN 变为高；从 GPIO0 输出低以通知 Master1 释放 SPI 总线；从 GPIO2 输出低以选中 Slave1；与 Slave1 进行通信；从 GPIO2 输出高以释放 Slave1；从 GPIO0 输出高以释放 Master1。

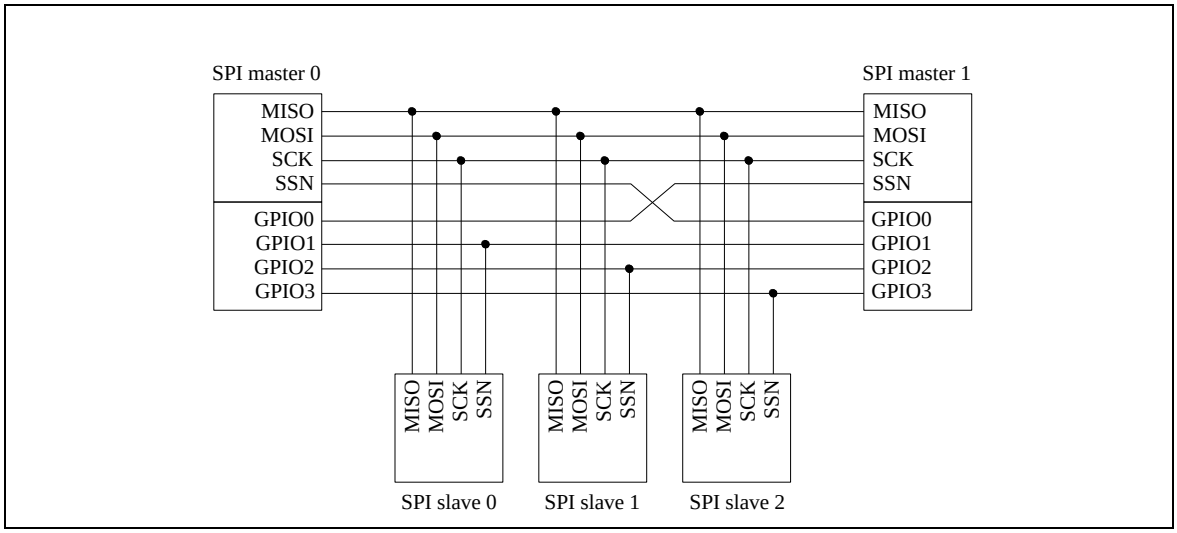


图 10-6 SPI 多主机/多从机系统的示意图

10.3.6 SPI 管脚配置说明

SPI 可以在一些特殊的管脚配置下保持部分或全部的功能。

具体情况如下表（“√”代表该管脚被配置并使用，空白代表该管脚未被配置）：

	SPI_CS(主)/ SPI_SSN(从)	SCK	MOSI	MISO	功能说明
主机模式	√	√	√	√	一般配置 全部主机功能正常
		√	√	√	全部主机功能正常
	√	√	√		主机发送功能正常
	√	√		√	主机接收功能正常
		√	√		主机发送功能正常
		√		√	主机接收功能正常
从机模式	√	√	√	√	一般配置 全部从机功能正常
	√	√	√		从机接收功能正常
	√	√		√	从机发送功能正常
	√ 固定低电平	√	√	√	全部从机功能正常
	√ 固定低电平	√	√		从机接收功能正常
	√ 固定低电平	√		√	从机发送功能正常

表 10-1 SPI 管脚配置说明表

注意：

- 表中未列举的情况暂不支持。
- 主机模式下，即使不使用 SPIx_CS 片选输出，发送数据前需要设置 SPIx.SSN 为 1，发送数据后需要设置 SPIx.SSN 为 0。
- 从机模式且片选输入固定为低电平时，为维持正常功能，必须满足 SPIx_CR.cpha=1。

10.4 SPI 编程示例

10.4.1 SPI 主机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 CS/SCK/MISO/MOSI 映射到需要的管脚；并配置 CS/SCK/MOSI 管脚为输出模式，配置 MISO 管脚为输入模式。

Step2：设置 SPIx_CR.mstr 为 1，使 SPI 工作于主机模式。

Step3：配置 SPIx_CR[spr2:spr0]，使 SCK 输出的时钟速率符合应用需求。

Step4：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step5：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step6：设置 SPIx_SSN.ssn 为 0，使 CS 管脚输出低电平以选中从机。

Step7：当查询到 SPIx_STAT.txe 为 1 时，将待发送的数据尽快写入 SPIx_DATA。

Step8：如待发送的数据未完成，则跳转到 Step7 继续执行。

Step9：查询等待 SPIx_STAT.txe 变为 1，最后一个字节数据已开始发送。

Step10：查询等待 SPIx_STAT.busy 变为 0，SPI 总线空闲。

Step11：设置 SPIx_SSN.ssn 为 1，使 CS 管脚输出高电平以释放从机。

注意：

– 可使用 GPIO 代替 CS 实现片选输出，多用于多机通信系统。

– 在传输过程必需设置 SPIx_SSN.ssn 为 0，传输完成后必需设置 SPIx_SSN.ssn 为 1。

10.4.2 SPI 主机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 CS/SCK/MISO/MOSI 映射到需要的管脚；并配置 CS/SCK/MOSI 管脚为输出模式，配置 MISO 管脚为输入模式。

Step2：设置 SPIx_CR.mstr 为 1，使 SPI 工作于主机模式。

Step3：配置 SPIx_CR[spr2:spr0]，使 SCK 输出的时钟速率符合应用需求。

Step4：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step5：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step6：设置 SPIx_SSN.ssn 为 0，使 CS 管脚输出低电平以选中从机。

Step7：向 SPIx_DATA 写入任意数据以触发主机发送 SCK。

Step8：查询等待 SPIx_STAT.rxne 变为 1，已收到从机发送的数据。

Step9：从 SPIx_DATA 读出收到的数据。

Step10：如果待接收的数据未完成，则跳转到 Step7 继续执行。

Step11：设置 SPIx_SSN.ssn 为 1，使 CS 管脚输出高电平以释放从机。

注意：

– 可使用 GPIO 代替 CS 实现片选输出，多用于多机通信系统。

– 在传输过程必需设置 SPIx_SSN.ssn 为 0，传输完成后必需设置 SPIx_SSN.ssn 为 1。

10.4.3 SPI 从机发送示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SSN/SCK/MISO/MOSI 映射到需要的管脚；并配置 SSN/SCK/MOSI 管脚为输入模式，配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2：设置 PERI_RESET0.SPIx 为 0，使 SPI 模块处于复位状态。

Step3：设置 PERI_RESET0.SPIx 为 1，使 SPI 模块处于工作状态。

Step4：设置 SPIx_CR.mstr 为 0，使 SPI 工作于从机模式。

Step5：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step6：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step7：将待发送的第一个字节数据写入 SPIx_DATA。

Step8：查询等待 SSN 管脚拉低，主机选中 SPI 从机。

Step9：当查询到 SPIx_STAT.txe 为 1 时，将待发送的数据尽快写入 SPIx_DATA。

Step10：当查询到 SSN 管脚为低且待发送的数据尚未完成，跳转到 Step9。

Step11：查询等待 SSN 管脚拉高，主机释放 SPI 从机。

注意：

– 每当 SPI 从机需要发送数据时，均需从 Step2 开始执行初始化操作及发送操作。

10.4.4 SPI 从机接收示例

Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SSN/SCK/MISO/MOSI 映射到需要的管脚；并配置 SSN/SCK/MOSI 管脚为输入模式，配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2：设置 SPIx_CR.mstr 为 0，使 SPI 工作于从机模式。

Step3：配置 SPIx_CR.cpol 及 SPIx_CR.cpha，使数据帧格式符合应用需求。

Step4：设置 SPIx_CR.spen 为 1，使能 SPI 接口。

Step5：查询等待 SSN 管脚拉低，主机选中 SPI 从机。

Step6：查询等待 SPIx_STAT.rxne 变为 1，已收到主机发送的数据。

Step7：从 SPIx_DATA 读出收到的数据。

Step8：如果待接收的数据未完成，则跳转到 Step6 继续执行。

Step9：查询等待 SSN 管脚拉高，主机释放 SPI 从机。

10.6 SPI 寄存器描述

寄存器列表

SPI0 基地址：0x40000800

SPI1 基地址：0x40004800

偏移量	寄存器名称	访问	寄存器描述
0x00	SPIx_CR	RW	SPIx 配置寄存器
0x04	SPIx_SSN	RW	SPIx 片选配置寄存器
0x08	SPIx_STAT	RO	SPIx 状态寄存器
0x0c	SPIx_DATA	RW	SPIx 数据寄存器
0x10	SPIx_CR2	RW	SPIx 配置寄存器 2
0x14	SPIx_ICLR	WO	SPIx 中断清除寄存器

表 10-2 SPI 寄存器列表

10.6.1 SPI 配置寄存器(SPIx_CR)

地址偏移量：0x00

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spr2	spen	Res	mstr	cpol	cpha	spr1	spr0
								RW	RW		RW	RW	RW	RW	RW

位	标记	功能描述
31:8	Reserved	
7	spr2	波特率选择位 2 参考 spr0.
6	spen	SPI 模块使能控制 0 – 禁止 1 – 使能
5	Reserved	
4	mstr	SPI 工作模式配置 0 – 从机模式 1 – 主机模式
3	cpol	SCK 线空闲状态配置 0 – 低电平 1 – 高电平
2	cpha	时钟相位配置 0 – 第一边沿 1 – 第二边沿
1	spr1	波特率选择位 1 参考 spr0

<table>
<tr>
<td>0</td>
<td>spr0</td>
<td>波特率选择位 0
<table>
<tr><th>spr2</th><th>spr1</th><th>spr0</th><th>SCK Rate</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>PCLK /2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>PCLK /4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>PCLK /8</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>PCLK /16</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>PCLK /32</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>PCLK /64</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>PCLK /128</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
表 10-3 主机模式波特率选择</td>
</tr>
</table>

10.6.2 SPI 片选配置寄存器(SPIx_SSN)

地址偏移量：0x04

复位值：0x000000FF

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

位	标记	功能描述
31:1	Reserved	
0	ssn	主机模式时 SPI_CS 输出电平配置 0：SPI_CS 端口输出低电平 1：SPI_CS 端口输出高电平

10.6.3 SPI 状态寄存器(SPIx_STAT)

地址偏移量：0x08

复位值：0x00000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spif	Res.	sserr	mdf	busy	txe	rxne	Res.
								RO		RO	RO	RO	RO	RO	

位	标记	功能描述
31:8	Reserved	
7	spif	接收完成标志 1：已从 SPI 总线上接收到一个字节 0：正在接收中
6	Reserved	
5	sserr	从机模式 SSN 错误标志
4	mdf	主机模式时，冲突标志 1：SSN 管脚电平为低 0：SSN 管脚电平为高
3	busy	SPI 总线传输状态标志 1：SPI 总线正在传输数据 0：SPI 总线空闲
2	txe	发送缓冲器状态标志 1：发送缓冲器空 0：发送缓冲器非空
1	rxne	接收缓冲器状态标志 1：接收缓冲器非空 0：接收缓冲器空

0	Reserved	

10.6.4 SPI 数据寄存器(SPIx_DATA)

地址偏移量：0x0c

复位值：0x00000000

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

位	标记	功能描述
31:8	Reserved	
7:0	spdat	数据寄存器 在发送模式，向这个寄存器写入待发送的字节； 在接收模式，从这个寄存器读取接收到的字节；

10.6.5 SPI 配置寄存器 2(SPIx_CR2)

地址偏移量：0x10

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									rxneie	txeie	hdma_tx	hdma_rx	int_en	Reserved	
									RW	RW	RW	RW	RW		

位	标记	功能描述
31:7	Reserved	
6	rxneie	接收缓冲器非空中断使能 0 – 禁止 1 – 使能
5	txeie	发送缓冲器空中断使能 0 – 禁止 1 – 使能
4	hdma_tx	DMA 硬件访问发送使能. 0 – 禁止 1 – 使能
3	hdma_rx	DMA 硬件访问接收使能. 0 – 禁止 1 – 使能
2	int_en	SPI 中断使能. 0 – 禁止 1 – 使能
1:0	Reserved	请保持这两位的值均为 1。

10.6.6 SPI 中断清除寄存器 2(SPIx_ICLR)

地址偏移量：0x14

复位值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															int_clr
															R1W0

位	标记	功能描述
31:1	Reserved	
0	int_clr	SPI 中断清除 0 – 清除 1 – 保持

11 时钟校准模块（CLKTRIM）

11.1 CLKTRIM 简介

CLKTRIM(Clock Trimming)模块是一个专门用来校准/监测时钟的电路。在校准模式下选择精准的时钟源来校准不精准的时钟源，反复校准，调节不精准时钟源的参数，直到被校准时钟源的频率达到精度要求。校准模式下计数值会有一定的误差，但是在允许的精度误差范围内。在监测模式下选择稳定的时钟源来监测系统工作时钟，在设定的监测周期下，监测系统工作时钟是否有失效的情况发生并产生中断。在校准模式和监控模式下，所需的时钟源都必须初始化和使能，具体的配置过程请参考第 4 章系统控制器。

11.2 CLKTRIM 主要特性

CLKTRIM 支持以下特性：

- 校准模式
- 监测模式
- 32 位参考时钟计数器可加载初值
- 32 位待校准时钟计数器可配置溢出值
- 6 种参考时钟源
- 5 种待校准时钟源
- 支持中断方式

11.3 CLKTRIM 功能描述

11.3.1 CLKTRIM 校准模式

11.3.1.1 时钟校准原理

在校准一个频率不精准，但是有参数设置可以调节频率的时钟（待校准时钟 CAL_CLK）时，需要一个精准的时钟作为参考时钟(REF_CLK)。设定校准时间 Ttrim，待校准时钟频率 Fcal, 参考时钟频率 Fref,使用待校准时钟和参考时钟同时计数，在校准时间结束时停止计数。读取待校准时钟计数器的值 M，参考时钟计数器的值 N，

得到等式 Ttrim = M/Fcal = N/Fref

推导出　Fcal = Fref * M / N

由公式判断出，参考时钟的频率误差率越小，待校准时钟的误差率就越小。

计算出待校准时钟频率后，如果误差率超出系统要求的范围，可以调节待校准时钟的参数后，再次重复上面的步骤校准，直到待校准时钟频率的误差率满足系统的要求。

11.3.1.2 时钟校准模块硬件结构

时间校准模块的待校准时钟有 5 个时钟源，参考时钟有 6 个时钟源，如图 11-1 所示：

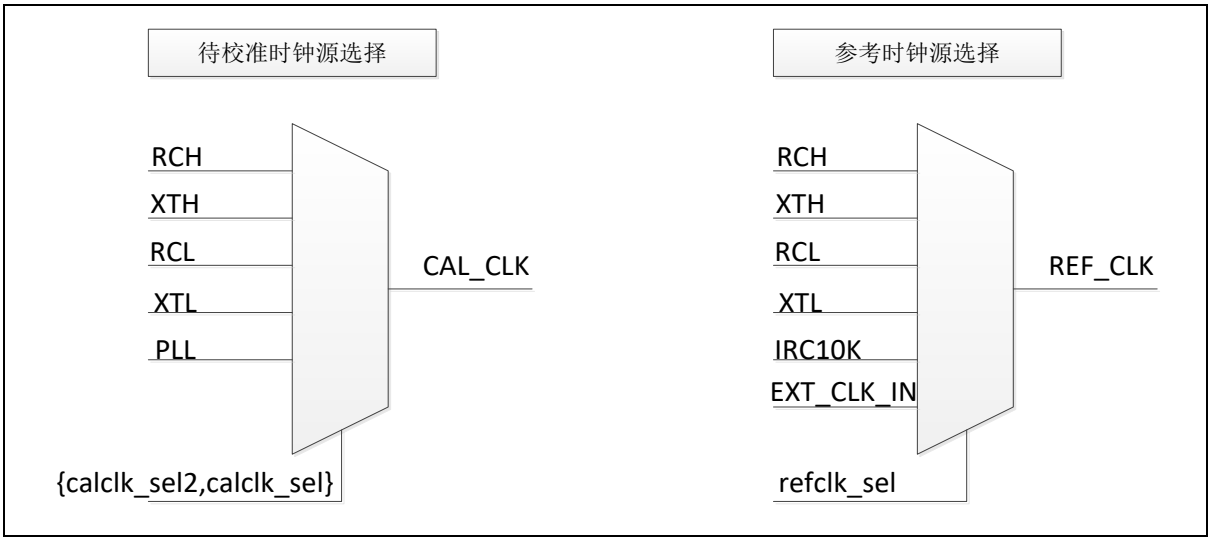


图 11-1　时钟源选择示意图

待校准时钟的选择由寄存器 CLKTRIM_CR.calclk_sel2,CLKTRIM_CR.calclk_sel 来配置。

参考时钟的选择由寄存器 CLKTRIM_CR.refclk_sel 来配置。

如图 11-2 所示，时钟校准模块有 2 个 32 位计数器，

1 个是以参考时钟为时钟，可配置初值的减计数器，

初值由寄存器 CLKTRIM_REFCON .rcntval 来配置，

计数器值可以从寄存器 CLKTRIM_REFCNT 读出，

当计数器减计数到 0 时，停止计数并产生中断。

1 个是以待校准时钟为时钟，可配置溢出值的加计数器，

溢出值由寄存器 CLKTRIM_CALCON. ccntval 来配置，

计数器值可以从寄存器 CLKTRIM_CALCNT 读出，

当计数器加计数到溢出值时，停止计数并产生中断。

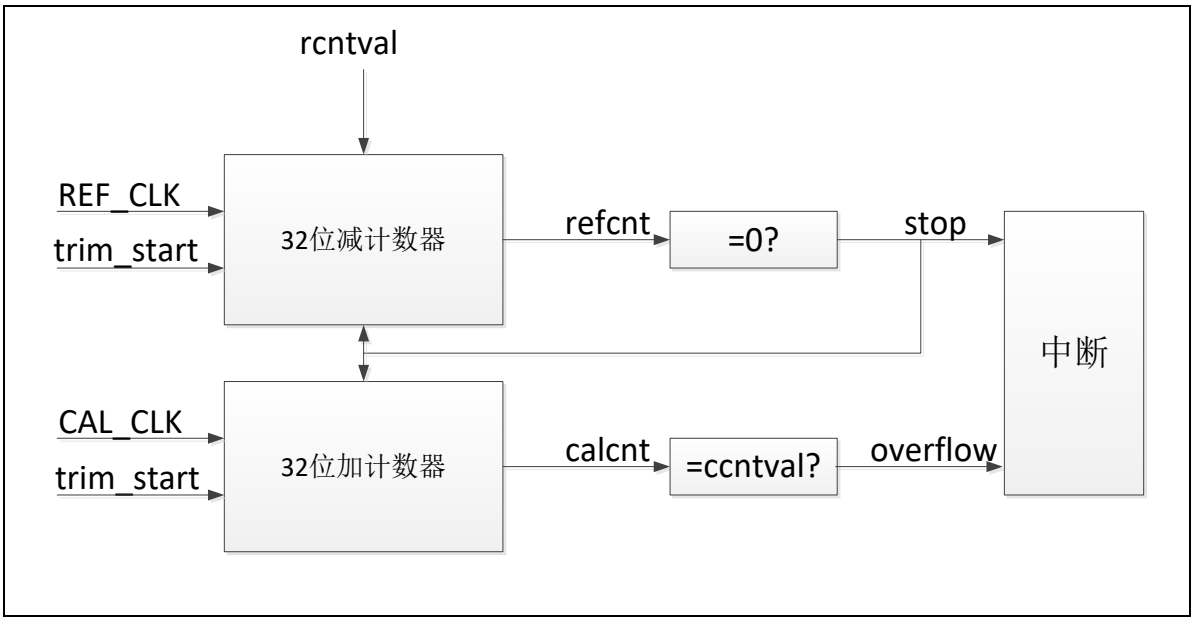


图 11-2 时钟校准模块硬件示意图

11.3.1.3 时钟校准软件流程

1. 设置 CLKTRIM_CR .refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR .calclk_sel 寄存器选择被校准时钟。
3. 设置 CLKTRIM_REFCON .rcntval 寄存器为校准时间。
4. 设置 CLKTRIM_CR .IE 寄存器使能中断。
5. 设置 CLKTRIM_CR .trim_start 寄存器开始校准。
6. 参考时钟计数器和待校准时钟计数器开始计数。
7. 当参考时钟计数器从初始值减计数到 0 时，CLKTRIM_IFR.stop 置 1，触发中断。
8. 中断服务子程序判断 CLKTRIM_IFR.stop 为 1，读取寄存器 CLKTRIM_REFCNT

和 CLKTRIM_CALCNT 的值，

9. 清零 CLKTRIM_CR .trim_start 寄存器结束校准。
10. 根据时钟校准原理章节提到的公式计算出待校准时钟的频率，

    其中 M = 寄存器 CLKTRIM_CALCNT 的值，

    N = 寄存器 CLKTRIM_REFCON .rcntval 的值

    当待校准时钟的频率不满足误差率要求时，调整待校准时钟参数，

    重新执行步骤 5~10，直到待校准时钟满足误差率要求。

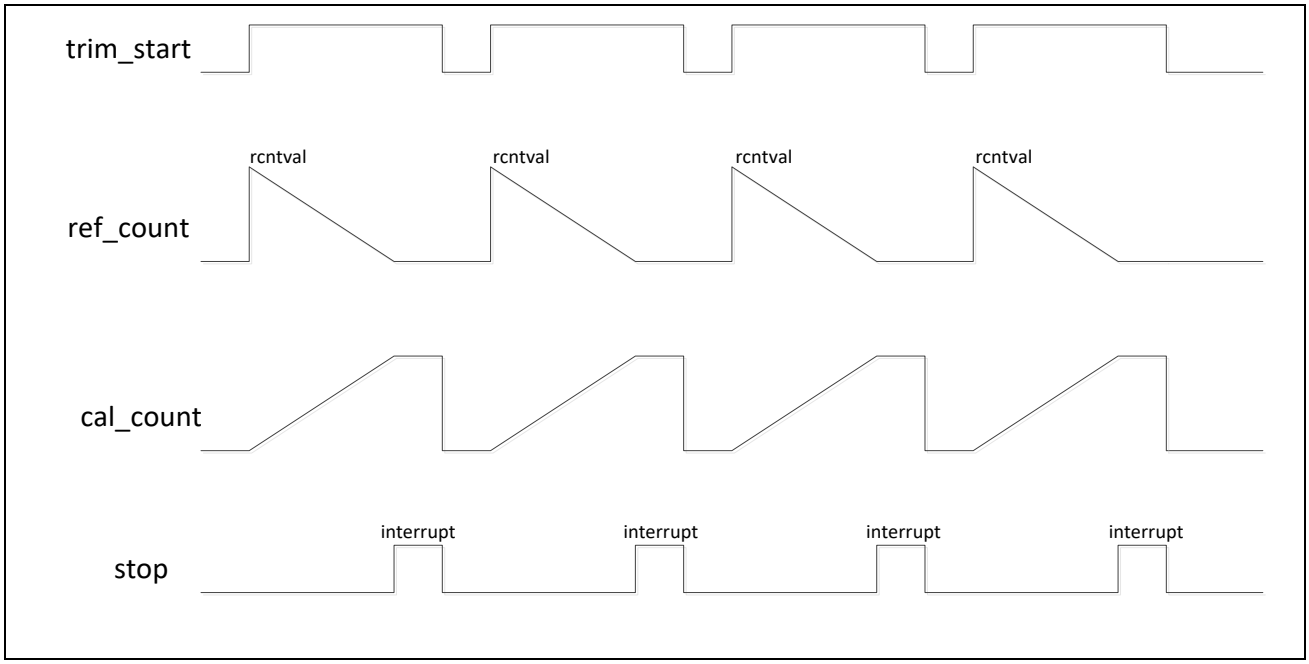


图 11-3 时钟校准波形示意图

注意：

– 校准模式在校准过程中有可能因为校准时间设置过长，发生待校准时钟计数器在 CLKTRIM_IFR.stop 置 1 之前溢出的情况，CLKTRIM_IFR. calcnt_of 置 1，触发中断。中断服务子程序发现 CLKTRIM_IFR. calcnt_of 置 1 时，清零 CLKTRIM_CR .trim_start 寄存器结束校准。

这种情况下校准是无法正确进行的，必须调整校准时间，重新校准。

具体步骤是：

1. 设置 CLKTRIM_REFCON .rcntval 寄存器调整校准时间。
2. 设置 CLKTRIM_CR .trim_start 寄存器重新开始校准。

11.3.2 CLKTRIM 监测模式

11.3.2.1 时钟监测原理

为了监测系统工作时钟（被监测时钟 CAL_CLK）是否正常工作时，需要一个稳定的时钟作为参考时钟(REF_CLK)。设定监测时间 Ttrim，被监测时钟频率 Fcal,参考时钟频率 Fref, 被监测时钟计数器溢出值 CALVAL, 使用被监测时钟和参考时钟同时计数，在监测时间结束时停止计数，判断被监测时钟计数器是否处于溢出状态。如果被监测时钟计数器已溢出，表示在监测时间内被监测时钟工作正常，继续进行下一次监测。如果被监测时钟计数器没有溢出，表示在监测时间内被监测时钟工作异常，可能被监测时钟已经停止或者频率大幅变化，产生被监测时钟工作异常中断，由硬件切换系统工作时钟。

11.3.2.2 时钟监测硬件结构

时间校准模块的被监测时钟有 2 个时钟源，参考时钟有 6 个时钟源，如图 11-4 所示：

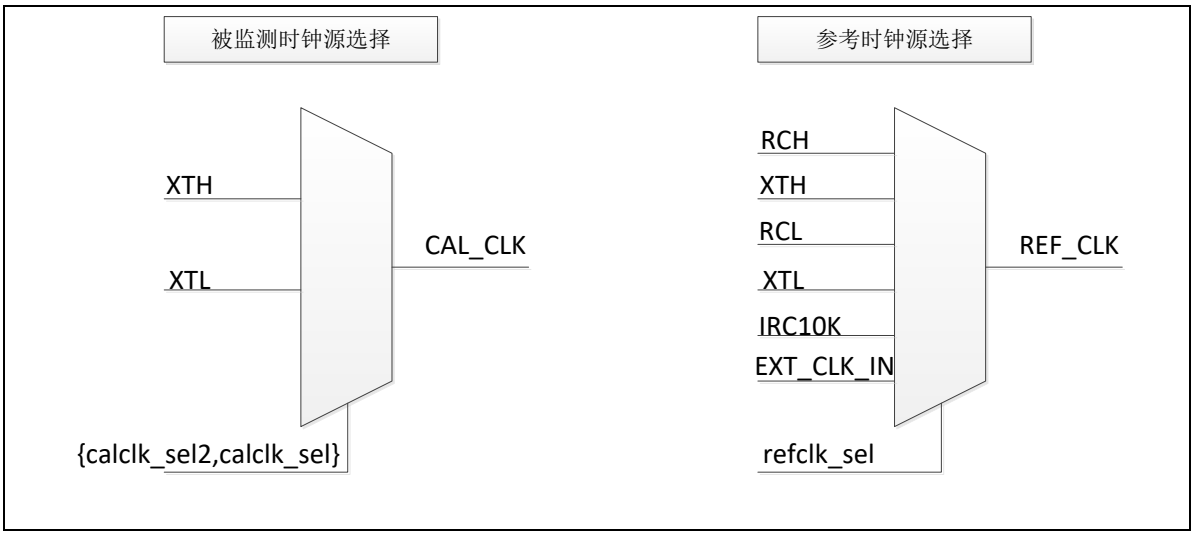


图 11-4 CLKTRIM 时钟选择

被监测时钟的选择由寄存器 CLKTRIM_CR.calclk_sel2,CLKTRIM_CR.calclk_sel 来配置，参考时钟的选择由寄存器 CLKTRIM_CR.refclk_sel 来配置。

11.3.2.3 时钟监测软件流程

1. 设置 CLKTRIM_CR .refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR .calclk_sel 寄存器选择被监控时钟。

3. 设置 CLKTRIM_REFCON .rcntval 寄存器为监控间隔时间。
4. 设置 CLKTRIM_CALCON. ccntval 寄存器为被监控时钟计数器溢出时间。
5. 设置 CLKTRIM_CR .mon_en 寄存器使能监控功能。
6. 设置 CLKTRIM_CR .IE 寄存器使能中断。
7. 设置 CLKTRIM_CR .trim_start 寄存器开始监控。
8. 参考时钟计数器和被监控时钟计数器开始计数。
9. 当参考时钟计数器计数到达监控间隔时间时，CLKTRIM_IFR.stop 置 1，同时判断被监控时钟计数器是否溢出，即 CLKTRIM_IFR. calcnt_of 是否置 1。如果溢出，即 CLKTRIM_IFR. calcnt_of 为 1，表示被监控时钟工作正常。如果没有溢出，即 CLKTRIM_IFR. calcnt_of 为 0，表示被监控时钟失效，CLKTRIM_IFR .xtl_fault/xth_fault 置 1，触发中断。
10. 处理中断服务子程序，清除中断标志位 CLKTRIM_IFR .xtl_fault/xth_fault，清零 CLKTRIM_CR .trim_start 寄存器结束监测。

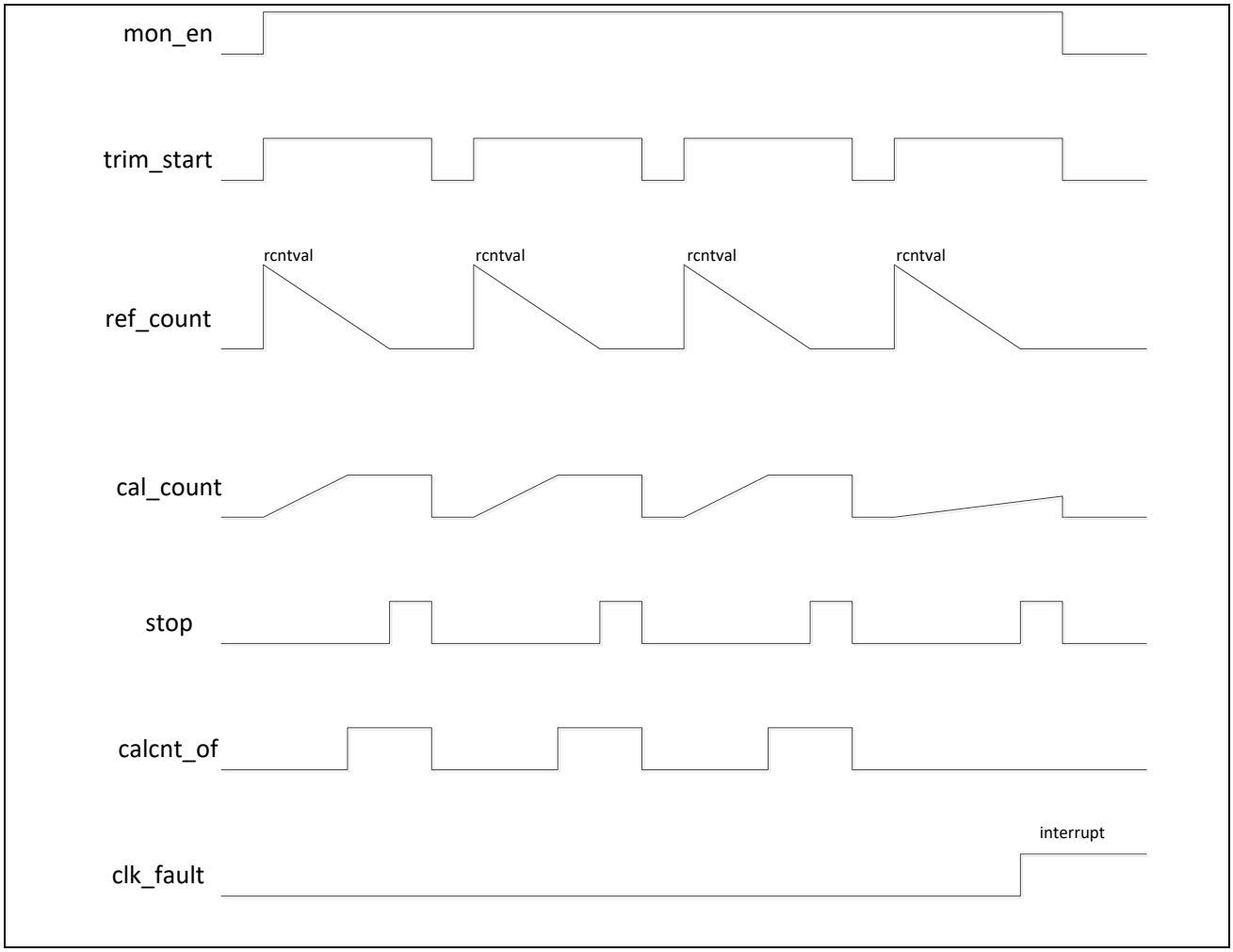


图 11-5 时钟监控波形示意图

11.4 CLKTRIM 寄存器描述

寄存器列表　　基地址：0x40001800

偏移量	寄存器名称	访问	寄存器描述
0x00	CLKTRIM_CR	RW	配置寄存器
0x04	CLKTRIM_REFCON	RW	参考计数器初值配置寄存器
0x08	CLKTRIM_REFCNT	RO	参考计数器值寄存器
0x0c	CLKTRIM_CALCNT	RO	校准计数器值寄存器
0x10	CLKTRIM_IFR	RO	中断标志位寄存器
0x14	CLKTRIM_ICLR	RW	中断标志位清除寄存器
0x18	CLKTRIM_CALCON	RW	校准计数器溢出值配置寄存器

表 11-1　寄存器列表

11.4.1 配置寄存器(CLKTRIM_CR)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							calclk_sel2	IE	mon_en	calclk_sel		refclk_sel			trim_start
							RW	RW	RW	RW		RW			RW

位	标记	功能描述
31:9	Reserved	
8	calclk_sel2	待校准/监测时钟选择高位寄存器
7	IE	中断使能寄存器　0 – 禁止　　1 – 使能
6	mon_en	监测模式使能寄存器　0 – 禁止　　1 – 使能
5:4	calclk_sel	待校准/监测时钟选择低位寄存器 calclk_sel2，calclk_sel　　 000　　RCH 001　　XTH 010　　RCL 011　　XTL 100　　PLL 101 - 111　　保留
3:1	refclk_sel	参考时钟选择寄存器 000 ---- RCH 001 ---- XTH 010 ---- RCL 011 ---- XTL

		100 ---- IRC10K 101 ---- EXT_CLK_IN
0	trim_start	校准/监测开始寄存器　　0 – 停止　　1 – 开始

11.4.2 参考计数器初值配置寄存器(CLKTRIM_REFCON)

地址偏移量：0x04

复位值：0x00000000

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

位	标记	功能描述
31:0	rcntval	参考计数器初始值

11.4.3 参考计数器值寄存器(CLKTRIM_REFCNT)

地址偏移量：0x08

复位值：0x00000000

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

位	标记	功能描述
31:0	refcnt	参考计数器值

11.4.4 校准计数器值寄存器(CLKTRIM_CALCNT)

地址偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
calcnt[31:16]															
RO															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
calcnt[15:0]															
RO															

位	标记	功能描述
31:0	calcnt	校准计数器值

11.4.5 中断标志位寄存器(CLKTRIM_IFR)

地址偏移量：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											pll_fault	xth_fault	xtl_fault	calcnt_of	stop
											RO	RO	RO	RO	RO

位	标记	功能描述
31:5	Reserved	
4	pll_fault	PLL 失效标志. CLKTRIM_ICLR.pll_fault_clr 写零清除此标志位
3	xth_fault	XTH 失效标志. CLKTRIM_ICLR.xth_fault_clr 写零清除此标志位
2	xtl_fault	XTL 失效标志. CLKTRIM_ICLR.xtl_fault_clr 写零清除此标志位
1	calcnt_of	校准计数器溢出标志. CLKTRIM_CR.start 写零清除此标志位
0	stop	参考计数器停止标志. CLKTRIM_CR.start 写零清除此标志位

11.4.6 中断标志位清除寄存器(CLKTRIM_ICLR)

地址偏移量：0x14

复位值：0x1111 1111

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											pll_fault_clr	xth_fault_clr	xtl_fault_clr	Reserved	
											R1W0	R1W0	R1W0		

位	标记	功能描述
31:5	Reserved	
4	pll_fault_clr	清除 PLL 失效标志，写零清除。
3	xth_fault_clr	清除 XTH 失效标志，写零清除。
2	xtl_fault_clr	清除 XTL 失效标志，写零清除。
1:0	Reserved	

11.4.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON)

地址偏移量：0x18

复位值：0xffff ffff

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

位	标记	功能描述
31:0	ccntval	校准计数器溢出值

12 FLASH 控制器 (FLASH)

12.1 概述

本系统包含一块 64k/128k/192k/256k 字节（Byte）容量的 FLASH 存储器，共划分为 128/256/384/512 个页（Sector），每个页（Sector）的容量为 512 字节（Byte）。FLASH 控制器支持对 FLASH 存储器的擦除、编程以及读操作。本控制器还支持对 FLASH 存储器擦写保护，以及控制寄存器的写保护。

12.2 FLASH 容量划分

地址	序号		地址	序号
0x0E00 – 0x0FFF	Sector7	……	0x3FE00 – 0x3FFFF	Sector511
0x0C00 – 0x0DFF	Sector6	……	0x3FC00 – 0x3FDFF	Sector510
0x0A00 – 0x0BFF	Sector5	……	0x3FA00 – 0x3FBFF	Sector509
0x0800 – 0x09FF	Sector4	……	0x3F800 – 0x3F9FF	Sector508
0x0600 – 0x07FF	Sector3	……	0x3F600 – 0x3F7FF	Sector507
0x0400 – 0x05FF	Sector2	……	0x3F400 – 0x3F5FF	Sector506
0x0200 – 0x03FF	Sector1	……	0x3F200 – 0x3F3FF	Sector505
0x0000 – 0x01FF	Sector0	……	0x3F000 – 0x3F1FF	Sector504

表 12-1 FLASH 容量划分

注意：

- 64KByte FLASH 由 Sector0-127 组成；
- 128KByte FLASH 由 Sector0-255 组成；
- 192KByte FLASH 由 Sector0-383 组成；
- 256KByte FLASH 由 Sector0-511 组成。

12.3 功能描述

本控制器支持对 FLASH 的 Byte（8 位）、half-word（16 位）和 word（32 位）三种位宽的数据读写操作。注意，Byte 操作的目标地址必须按字节对齐，half-word 操作的目标地址必须按半字对齐（地址最低位为 1’b0），word 操作的地址必须按字对齐（地址最低两位为 2’b00）。如果目标地址没有按照位宽对齐，则该操作无效，并且 CPU 会进入 Hard Fault 出错中断。

本控制器采用高安全性的硬件设计，具有 FLASH 操作源防御功能：仅当 FLASH 操作函数的地址位于 0~32K，才能正确执行 FLASH 擦写操作。

FLASH 地址的 0~32K 具有更高的安全性，重要函数须放置此区域内；例如重要程序入口、中断入口函数、高安全算法模块、UID、AES、真随机数的算法配合，组成高安全认证系统。

12.3.1 页擦除（Sector Erase）

页擦除每次可以擦除用户指定的一个页（Sector）。擦除操作完成后，页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR. BUSY 变为 0）；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

页擦除操作步骤如下：

Step1：配置 FLASH 擦写参数，详见擦写时序章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 2，设置 Flash 操作模式为 Sector 擦除。

Step4：检查 FLASH_CR. OP 是否为 2，如不为 2 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除该 Sector 的擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。

例：（(unsigned char *) 0x00000200) = 0x00。

Step9：等待 FLASH_CR. BUSY 变为 0，Sector 擦除操作完成。

Step10：如需擦除其它 Sector，重复 Step5 – Step9。

注意：

- **对 FLASH 进行页擦除的代码所在的地址必须小于 32768。**

12.3.2 全片擦除（Chip Erase）

全片擦除可以一次性擦除全部的页（Sector）。擦除操作完成后，所有页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则该操作会被禁止。因为该操作会擦除当前 PC 所在的程序段。如发生这种情况，出错标志位会被置起；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

全片擦除操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 12.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 3，设置 Flash 操作模式为 Chip 擦除。

Step4：检查 FLASH_CR. OP 是否为 3，如不为 3 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK0 为 0xFFFF FFFF，去除 Sector0~127 的擦写保护。

Step7：检查 FLASH_SLOCK0 是否为 0xFFFF FFFF，如不为 0xFFFF FFFF 则跳转到 则跳转到 Step5。

Step8：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9：设置 FLASH_SLOCK1 为 0xFFFF FFFF，去除 Sector128~255 的擦写保护。

Step10：检查 FLASH_SLOCK1 是否为 0xFFFF FFFF，如不为 0xFFFF FFFF 则跳转到 Step8。

Step11：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12：设置 FLASH_SLOCK2 为 0xFFFF FFFF，去除 Sector256~383 的擦写保护。

Step13：检查 FLASH_SLOCK2 是否为 0xFFFF FFFF，如不为 0xFFFF FFFF 则跳转到 Step11。

Step14：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step15：设置 FLASH_SLOCK3 为 0xFFFF FFFF，去除 Sector384~511 的擦写保护。

Step16：检查 FLASH_SLOCK3 是否为 0xFFFF FFFF，如不为 0xFFFF FFFF 则跳转到 Step14。

Step17：对待擦除的 Chip 内的任意地址进行写操作，触发 Chip 擦除。

例：（(unsigned char *）0x00000000）= 0x00。

Step18：等待 FLASH_CR. BUSY 变为 0，Chip 擦除操作完成。

注意：

- **对 FLASH 进行片擦除的代码所在的地址必须小于 32768。**

12.3.3 写操作（Program）

写操作只能将 FLASH 内的比特数据由 1 写成 0，故写入数据前应确保待写入的地址内的数据为 0xFF。支持写入 3 种数据长度：Byte（8bits）、Half-word（16bits）、Word（32bits），写入的数据以小端模式存放在 FLASH 中，即低地址存放数据的低字节。如果写操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR. BUSY 变为 0）；如果该写操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR. BUSY 变为 0）。

Byte 写操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 12.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Byte 写操作，触发写入操作。

例：（（unsigned char *）0x00001231）= 0x5A。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Byte 到其它地址，重复 Step8 – Step9。

Half-word 写操作步骤如下：

Step1：配置 FLASH 擦写时间，详见 12.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Half-word 写操作，触发写入操作。

*例：**（*(unsigned short int *) 0x00001232*）= *0xABCD*。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Half-word 到其它地址，重复 Step8 - Step9。

Word 写操作步骤如下：

Step1：配置 FLASH 擦写参数，详见 12.4 章节。

Step2：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：配置 FLASH_CR. OP 为 1，设置 Flash 操作模式为写入。

Step4：检查 FLASH_CR. OP 是否为 1，如不为 1 则跳转到 Step2。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7：检查 FLASH_SLOCK 相应的比特是否为 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8：对待写入的目标地址进行 Word 写操作，触发写入操作。

*例：**（*(unsigned int *) 0x00001234*）= *0x55667788*。

Step9：等待 FLASH_CR. BUSY 变为 0，写入操作完成。

Step10：如需写 Word 到其它地址，重复 Step8 - Step9。

注意：

- **对 FLASH 进行写操作的代码所在的地址必须小于 32768。**

12.3.4 读操作（Read）

支持读出 3 种数据长度：Byte（8bits）、Half-word（16bits）、Word（32bits）,读出的数据为小端模式，即低地址存放数据的低字节。读操作无需操作步骤，任何时刻都可以读出 FLASH 内的数据。

例：

Byte 读操作： *temp = *（(unsigned char *）0x00001231）*

Half-word 读操作 *temp = *（(unsigned short int *）0x00001232）*

Word 读操作 *temp = *（(unsigned int *）0x00001234）*

12.4 擦写时间

FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。上电时默认装载了 HCLK 为 4MHz 时的擦写参数；如果对 Flash 进行擦写时的 HCLK 频率不为 4MHz，则用户程序应装载 HCLK 频率所对应的擦写参数。对 FLASH 进行操作时，支持的最高 HCLK 为 48MHz。

擦写时序参数相关的寄存器为：FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE 、 FLASH_TMERASE 、 FLASH_TPRCV 、 FLASH_TSRCV 、 FLASH_TMRCV。如果 HCLK 由默认 4MHz 升高为 8MHz，则上述 FLASH_Tx 寄存器的值应该设为默认值的 2 倍，即保持当前 Tsysclk*FLASH_Tx 的结果与默认值相等即可。

以下表格为不同频率下相应的 FLASH 擦写时间参数：

Parameter	4M	8M	16M	24M	32M	48M
TNVS	0x20	0x40	0x80	0xC0	0x100	0x180
TPGS	0x17	0x2E	0x5C	0x8A	0xB8	0x114
TPROG	0x1B	0x36	0x6C	0xA2	0xD8	0x144
TSERASE	0x4650	0x8CA0	0x11940	0x1A5E0	0x23280	0x34BC0
TMERASE	0x222E0	0x445C0	0x88B80	0xCD140	0x111700	0x19A280
TPRCV	0x18	0x30	0x60	0x90	0xC0	0x120
TSRCV	0xF0	0x1E0	0x3C0	0x5A0	0x780	0xB40
TMRCV	0x3E8	0x7D0	0xFA0	0x1770	0x1F40	0x2EE0

表 12-2 不同频率下 FLASH 擦写时间参数

配置系统频率为 8MHz 时擦写参数的操作步骤如下：

Step1：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2：向 FLASH_TNVS 寄存器写入 0x40，如读出的该寄器值不为 0x40，则跳转到上一步。

Step3：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step4：向 FLASH_TPGS 寄存器写入 0x2E，如读出的该寄器值不为 0x2E，则跳转到上一步。

Step5：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6：向 FLASH_TPROG 寄存器写入 0x36，如读出的该寄器值不为 0x36，则跳转到上一步。

Step7：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step8：向 FLASH_TSERASE 寄存器写入 0x8CA0，如读出的该寄器值不为 0x8CA0，则跳转到上一步。

Step9：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10：向 FLASH_TMERASE 寄存器写入 0x445C0，如读出的该寄器值不为 0x445C0，则跳转到上一步。

Step11：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12：向 FLASH_TPRCV 寄存器写入 0x30，如读出的该寄器值不为 0x30，则跳转到上一步。

Step13：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step14：向 FLASH_TSRCV 寄存器写入 0x1E0，如读出的该寄器值不为 0x1E0，则跳转到上一步。

Step15：向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step16：向 FLASH_TMRCV 寄存器写入 0x7D0，如读出的该寄器值不为 0x7D0，则跳转到上一步。

12.5 读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz。当 HCLK 频率超过 24MHz 小于 48MHz 时，必须为 FLASH 的读取时间插入等待周期，即设置 FLASH_CR.WAIT 为 1。当插入等待周期时，FLASH 每两个周期才会完成一次读取操作。

12.6 擦写保护

12.6.1 擦写保护位

整个 64k/128k/192k/256k 字节 FLASH 存储器被划分为 128/256/384/512 个页，每 4 个页共用一个擦写保护位。当页被保护时，对该页进行的擦写操作均无效并产生报警标志位和中断信号。当 FLASH 存储器中的任意页被保护时，对该 FLASH 的全片擦写无效，并产生报警标志位和中断信号。

12.6.2 PC 地址擦写保护

CPU 在 FLASH 中运行程序时，如果当前 PC 指针正好落在待擦写的页地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。

12.7 寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

需要通过写序列方式才能更改的寄存器如下所示：

FLASH_TNVS 、 FLASH_TPGS 、 FLASH_TPROG 、 FLASH_TSERASE 、 FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV、FLASH_CR、FLASH_SLOCK0、FLASH_SLOCK1、FLASH_SLOCK2、FLASH_SLOCK3。

不需要通过写序列方式即可更改的寄存器如下所示：

FLASH_ICLR、FLASH_BYPASS。

通过写序列方式修改寄存器值的具体操作步骤如下所示：

Step1：向 FLASH_BYPASS 寄存器写入 0x5A5A。

Step2：向 FLASH_BYPASS 寄存器写入 0xA5A5。

Step3：对待修改的寄存器写入目标值。

Step4：验证待修改的寄存器当前值是否与目标相同，如不则跳转到 Step1。

Step5：执行其他操作。

注意：

- **写 0x5a5a、0xa5a5、写目标寄存器，这三步写操作之间不可插入任何写操作（写 ROM、RAM、REG），否则无法改写目标寄存器的数值。如改写失败，需要重新进行这三步操作。**

12.8 寄存器

基地址：0x4002 0000

寄存器	偏移地址	描述
FLASH_TNVS	0x00	Tnvs 时间参数
FLASH_TPGS	0x04	Tpgs 时间参数
FLASH_TPROG	0x08	Tprog 时间参数
FLASH_TSERASE	0x0C	Tserase 时间参数
FLASH_TMERASE	0x10	Tmerase 时间参数
FLASH_TPRCV	0x14	Tprcv 时间参数
FLASH_TSRCV	0x18	Tsrcv 时间参数
FLASH_TMRCV	0x1C	Tmrcv时间参数
FLASH_CR	0x20	控制寄存器
FLASH_IFR	0x24	中断标志寄存器
FLASH_ICLR	0x28	中断标志清除寄存器
FLASH_BYPASS	0x2C	0x5a5a-0xa5a5 Bypass序列寄存器
FLASH_SLOCK0	0x30	Sector0-127擦写保护寄存器
FLASH_SLOCK1	0x34	Sector128-255擦写保护寄存器
FLASH_SLOCK2	0x40	Sector256-383擦写保护寄存器
FLASH_SLOCK3	0x44	Sector384-512擦写保护寄存器

12.8.1 TNVS 参数寄存器（FLASH_TNVS）

偏移地址：0x00

复位值：0x0000 0020

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

位	标记	功能描述
31:9	Reserved	
8:0	TNVS	计算公式：TNVS = 8*HCLK，HCLK的单位为MHz。 4MHz示例：TNVS = 8*4 = 32。

12.8.2 TPGS 参数寄存器（FLASH_TPGS）

偏移地址：0x04

复位值：0x0000 0017

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:9	Reserved	
8:0	TPGS	计算公式：TPGS = 5.75*HCLK，HCLK的单位为MHz。 4MHz示例：TPGS = 5.75*4 = 23。

12.8.3 TPROG 参数寄存器（FLASH_TPROG）

偏移地址：0x08

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:9	Reserved	
8:0	TPROG	计算公式：TPROG = 6.75*HCLK，HCLK的单位为MHz。 4MHz示例：TPROG = 6.75*4 = 27。

12.8.4 TSERASE 寄存器（FLASH_TSERASE）

偏移地址：0x0C

复位值：0x0000 4650

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														TSERASE	
														RW	

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

位	标记	功能描述
31:18	Reserved	
17:0	TSERASE	计算公式：TSERASE = 4500*HCLK，HCLK的单位为MHz。 4MHz示例：TSERASE = 4500*4 = 27。

12.8.5 TMERASE 参数寄存器（FLASH_TMERASE）

偏移地址：0x10

复位值：0x000222E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TMERASE				
											RW				

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

位	标记	功能描述
31:21	Reserved	
20:0	TMERASE	计算公式：TMERASE = 35000*HCLK，HCLK的单位为MHz。 4MHz示例：TMERASE = 35000*4 = 140000。

12.8.6 TPRCV 参数寄存器（FLASH_TPRCV）

偏移地址：0x14

复位值：0x0000 0018

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

位	标记	功能描述
31:12	Reserved	
11:0	TPRCV	计算公式：TPRCV = 6*HCLK，HCLK的单位为MHz。 4MHz示例：TPRCV = 6*4 = 24。

12.8.7 TSRCV 参数寄存器（FLASH_TSRCV）

偏移地址：0x18

复位值：0x0000 00F0

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

位	标记	功能描述
31:12	Reserved	
11:0	TSRCV	计算公式：TSRCV = 60*HCLK，HCLK的单位为MHz。 4MHz示例：TSRCV = 60*4 = 240。

12.8.8 TMRCV 参数寄存器（FLASH_TMRCV）

偏移地址：0x1C

复位值：0x0000 03E8

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

位	标记	功能描述
31:14	Reserved	
13:0	TMRCV	计算公式：TMRCV = 250*HCLK，HCLK的单位为MHz。 4MHz示例：TMRCV = 250*4 = 1000。

12.8.9 CR 寄存器（FLASH_CR）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DPSTB_EN	Reserved		IE		BUSY	WAIT		OP	
						RW			RW		RO	RW		RW	

位	标记	功能描述
31:10	Reserved	
9	DPSTB_EN	FLASH dpstb使能Mask位； 0：当系统进入deepsleep模式，FLASH不进入低功耗模式； 1：当系统进入 deepsleep 模式，FLASH 进入低功耗模式；
8:7	Reserved	
6:5	IE	IE[6]：FLASH 擦写被保护地址中断使能；0：不使能；1：使能 IE[5]：FLASH 擦写 PC 值中断使能；0：不使能；1：使能
4	BUSY	空闲/忙标志位；0：空闲状态；1：忙状态；
3:2	WAIT	读FLASH 周期； 0~24MHz: 00/11，1个周期； 24~48MHz：01：2个周期； 48~72MHz：10:3个周期；（本系列产品最高时钟为48MHz）
1:0	OP	FLASH 操作； 00：读（read）；01：写（program）；10：页擦除（sector erase）；11：全片擦除（chip erase）

12.8.10 IFR 寄存器（FLASH_IFR）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IF1	IF0
														RO	RO

位	标记	描述
31:2	Reserved	
1	IF1	擦写保护报警中断标志位
0	IF0	擦写 PC 地址报警中断标志位

12.8.11 ICLR 寄存器（FLASH_ICLR）

偏移地址：0x28

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ICLR1	ICLR0
														R1W0	R1W0

位	标记	功能描述
31:4	Reserved	
3:2	Reserved	写无效，读为0x3
1	ICLR1	清除保护报警中断标志位；写0清除；写1无效；
0	ICLR0	清除 PC 地址报警中断标志位；写0清除；写1无效；

12.8.12 BYPASS 寄存器（FLASH_BYPASS）

偏移地址：0x2C

复位值：0x0000 0000

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

位	标记	描述
31:16	Reserved	
15:0	BYSEQ	在修改本模块寄存器之前，必须对 BYSEQ[15:0] 寄存器写入 0x5a5a-0xa5a5 序列。每次写入该 Bypass 序列后，只可以修改一次寄存器。如需再次修改寄存器，必须再次输入 0x5a5a-0xa5a5 序列。

12.8.13 SLOCK0 寄存器（FLASH_SLOCK0）

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK0[15:0]															
RW															

位	标记	描述
31:0	SLOCK0	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit[0] 对应：Sector0-1-2-3 Bit [1] 对应：Sector4-5-6-7 Bit [2] 对应：Sector8-9-10-11 Bit [3] 对应：Sector12-13-14-15 …… Bit [31] 对应：Sector124-125-126-127

12.8.14 SLOCK1 寄存器（FLASH_SLOCK1）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK1[15:0]															
RW															

位	标记	描述
31:0	SLOCK1	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit [0] 对应：Sector128-129-130-131 Bit [1] 对应：Sector132-133-134-135 Bit [2] 对应：Sector136-137-138-139 Bit [3] 对应：Sector140-141-142-143 …… Bit [31] 对应：Sector251-252-253-254

12.8.15 SLOCK2 寄存器（FLASH_SLOCK2）

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK2[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK2[15:0]															
RW															

位	标记	描述
31:0	SLOCK2	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit[0] 对应：Sector256-257-258-259 Bit [1] 对应：Sector260-261-262-263 Bit [2] 对应：Sector264-265-266-267 Bit [3] 对应：Sector268-269-270-271 …… Bit [31] 对应：Sector380-381-382-383

12.8.16 SLOCK3 寄存器（FLASH_SLOCK3）

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK3[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK3[15:0]															
RW															

位	标记	描述
31:0	SLOCK3	Sector 擦写保护位；0：不允许擦写；1：允许擦写 Bit[0] 对应：Sector384-385-386-387 Bit [1] 对应：Sector388-389-390-391 Bit [2] 对应：Sector392-393-394-395 Bit [3] 对应：Sector396-397-398-399 …… Bit [31] 对应：Sector252-253-254-255

13 RAM 控制器 (RAM)

13.1 概述

本系统中包含一块容量为 8k/16k/24k/32k 字节（Byte）的 SRAM，支持字节（8 位）、半字（16 位 bits）、字（32 位）三种读写操作。可在系统时钟频率下进行读写操作，无须等待周期。此外，本控制器还支持奇偶校验，可对每个字节（Byte）的 SRAM 数据进行奇偶校验，并产生奇偶校验出错中断。

13.2 功能描述

13.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示：

地址范围	大小	Memory类型
0x2000_0000 – 0x2000_7FFF	32KByte	SRAM
0x2000_0000 – 0x2000_3FFF	16KByte	SRAM

表 13-1 RAM 地址映射

13.2.2 读写位宽

本控制器支持 字节（8 位）、半字（16 位）、字（32 位） 三种位宽的读写操作。字节操作的地址必须按字节对齐，半字操作的目标地址必须按半字对齐（地址最低位为 1’b0），字操作的地址必须按字对齐（地址最低两位为 2’b00）。如果读写操作的目标地址没有按照位宽规定对齐，该操作无效，并且系统会产生 Hard Fault 出错中断。

13.2.3 奇偶校验

本控制器支持 SRAM 数据的奇偶校验。当对 SRAM 写数据时，对每个字节的数据做奇偶校验，并把 1bit 校验值和 8bits 数据一起存入 SRAM 中。当对 SRAM 读数据时，控制器会读取 8bits 数据和 1bit 校验值，并做奇偶校验，如果校验出错，则置位奇偶校验出错标志位，在中断使能情况下，会产生出错中断。

注意：在奇偶校验使能时，在读取 SRAM 数据前，必须初始化 SRAM，否则可能会误触发奇偶校验报警标志位或者中断。

13.3 寄存器

基地址：0x4002 0400

寄存器	偏移地址	描述
RAM_CR	0x00	控制寄存器
RAM_ERRADDR	0x04	出错地址寄存器
RAM_IFR	0x08	出错中断标志寄存器
RAM_ICLR	0x0C	出错中断标志清除寄存器

表 13-2 寄存器基地址

13.3.1 控制寄存器（RAM_CR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:2	Reserved	保留
1	IE	出错报警中断使能信号；1：使能报警中断，0：关闭报警中断；
0	Reserved	保留

13.3.2 奇偶校验出错地址寄存器（RAM_ERRADDR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ERRADDR														
	R														

位	标记	功能描述
31:15	Reserved	
14:0	ERRADDR	奇偶校验出错Byte地址；中断标志位清除后，该地址同时被清除；

13.3.3 出错中断标志寄存器（RAM_IFR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															ERR
															RO

位	标记	功能描述
31:1	Reserved	
0	ERR	奇偶校验出错标志位

13.3.4 出错中断标志清除寄存器（RAM_ICLR）

偏移地址：0x0C

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															ERRCLR
															R1W0

位	标记	功能描述
31:1	Reserved	
0	ERRCLR	出错中断标志清除位；写1：无效，写0：清零

14 DMA 控制器（DMAC）

14.1 概述

DMAC（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMAC 能提高系统性能。

DMAC 概述：

- DMAC 配有独立的总线，所以即便是在使用 CPU 总线的同时，DMAC 也可进行传输操作。
- 由 2 条通道组成，能执行 2 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式，并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作时，可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线(AHB)，支持 32 位地址空间(4GB)。

各通道功能概述：

- 可递增或固定传输源和传输目标地址。
- 具有传输源和传输目标地址重装功能（即传输完成后回到原设置值的功能）。
- 传输数据的大小可以从以下三种规格中选择：
  1) 传输数据宽度：（从字节/半字/字中选择）
  2) 设置块数：（从 1 至 16 中选择）
  3) 设置传输长度：（从 1 至 65536 中选择）
- 可指定是否发出传输成功完成和传输出错中断。
- 可选择以下四种传输模式：
  1) 软件-Block 传输
  2) 软件-Burst 传输
  3) 硬件-Block 传输
  4) 硬件-Burst 传输

传输模式概述：

软件传输是通过 CPU 直接指令启动 DMAC 的方法。

硬件传输是外设发出传输请求时将外设的请求信号作为 DMAC 的传输请求信号以直接启动 DMAC 的方法。外设请求可来自于 UART0/1，SPI0/1，TIM0/1/2/3/4/5/6，ADC，LCD。当这些外设有数据传输请求时，数据可以通过当前通道预先的传输配置来实现传输，而不用 CPU 介入。

14.2 功能框图

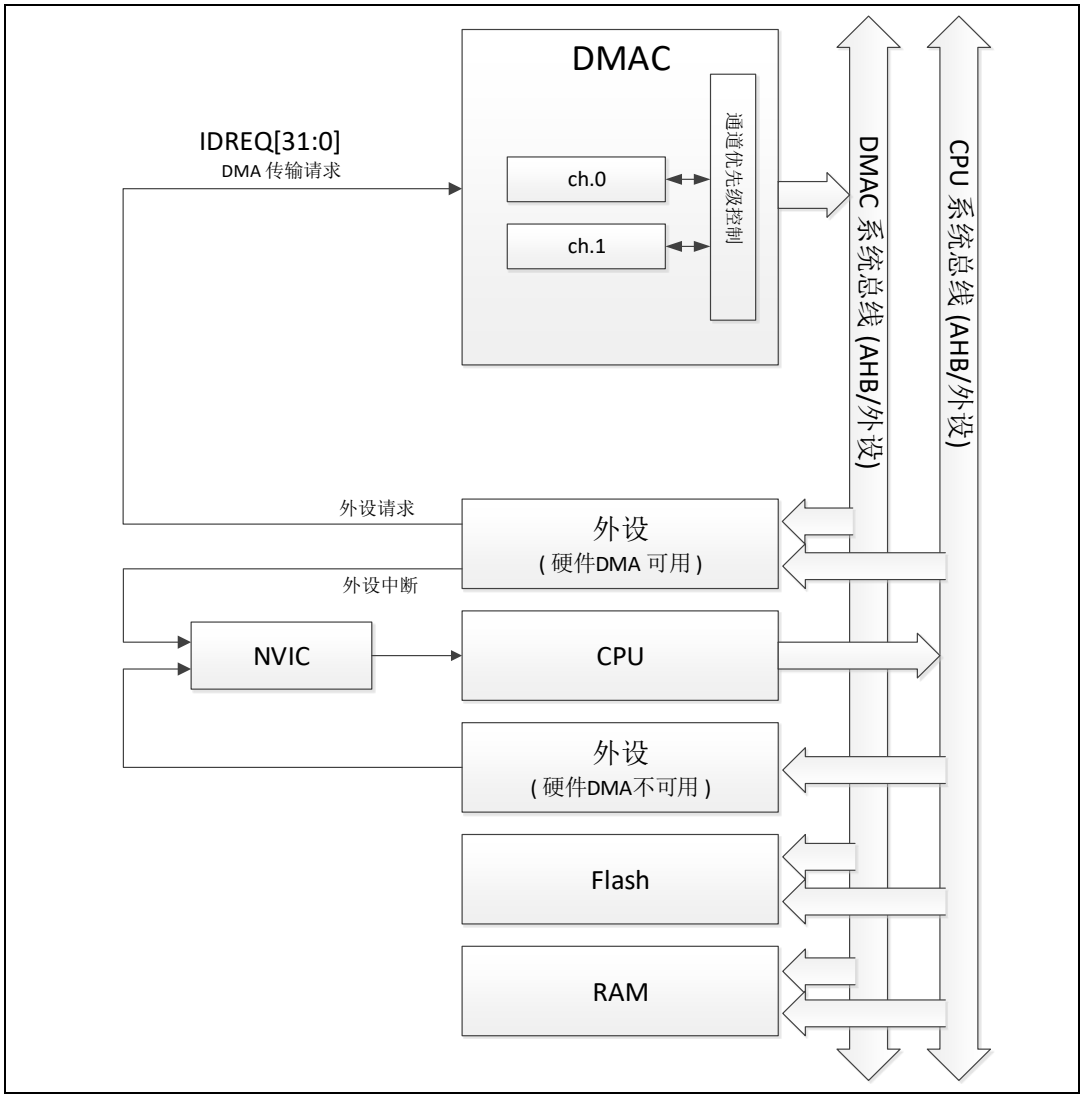


图 14-1 功能框图

框图说明

- DMAC

  DMAC 最多配置 2 条通道。各通道执行独立传输。通道之间发生冲突时，优先级控制器控制这些通道的传输操作。

- 系统连接

  DMAC 通过系统总线连接至 CPU、FLASH、RAM 以及各种外设。DMAC 自身有独立于 CPU 总线的总线，可在访问 CPU 总线时执行传输操作。通过指定各通道传输目标地址和传输源地址以进行存储器与外设之间传输数据，系统中的任何地址区

域均可访问。由于部分区域不能从 DMAC 访问，需检查内存映射。

- 硬件传输请求连接

  被中断控制器（指图 14-1 中的 DRQSEL）所选择支持硬件传输的外设中断信号，可以作为发送至 CPU 的中断信号或者同时作为发送至 DMAC 的 DMA 传输请求信号。

  根据硬件请求执行 DMA 传输时，通过设置 DRQSEL，连接各外设的中断信号，作为提前发给 DMAC 的传输请求信号。不支持硬件传输的外设中断信号不能作为 DMA 传输请求信号。

  输入 DMAC 的 DMA 传输请求信号有 40 个。硬件传输时，DMAC 各通道从上述 40 条传输请求信号中选择 1 条信号执行操作。使用 CONFA: TRI_SEL 寄存器进行选择。详见以下表格：

IDREQ 编号	对应外设传输请求
0	SPI0 RX
1	SPI0 TX
2	SPI1 RX
3	SPI1 TX
4	ADC JRQ0 RX
5	ADC SRQ0 RX
6	LCD TX
7	/
8	UART0 RX
9	UART0 TX
10	UART1 RX
11	UART1 TX
12	/
13	/
14	/
15	/

16	DAC
17	/
18	TIM0 CHA
19	TIM0 CHB
20	TIM1 CHA
21	TIM1 CHB
22	TIM2 CHA
23	TIM2 CHB
24	TIM3 CHA
25	TIM3 CHB
26	TIM4 CHA
27	TIM4 CHB
28	TIM5 CHA
29	TIM5 CHB
30	TIM6 CHA
31	TIM6 CHB
32	UART4 RX
33	UART4 TX
34	UART5 RX
35	UART5 TX

表 14-1 传输请求和对应外设请求信号

- DMAC 中断

  尽管图 14-1 未作说明，用于传输完成通知的中断信号连接至 NVIC。各通道有 8 个中断源，合并成 1 个输出。

14.3 基本功能

本节说明各传输模式下 DMAC 的操作。为了加速各模块间的数据搬运，PCLK 要与 HCLK 同频率下，DMA 才能发挥性能优势。

14.3.1 软件-块（Block）传输

本节说明软件-块传输。

图 14-2 所示为软件-块传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求块传输 (ST=1, TRI_SEL[5:0]=000000, MODE=00)
- 传输源启动地址： SA(SRC_ADR=SA)
- 传输源地址控制： 可递增和重装 (FS=0, RS=1)
- 传输目标启动地址： DA(DST_ADR=DA)
- 传输目标地址控制： 不可递增和重装 (FD=0, RD=0)
- 传输数据大小： 半字（16 位），块数= 2，传输长度= 3 (WIDTH=01, BC=1, TC=2)
- BC/TC 重装： 可重装 (RC=1)

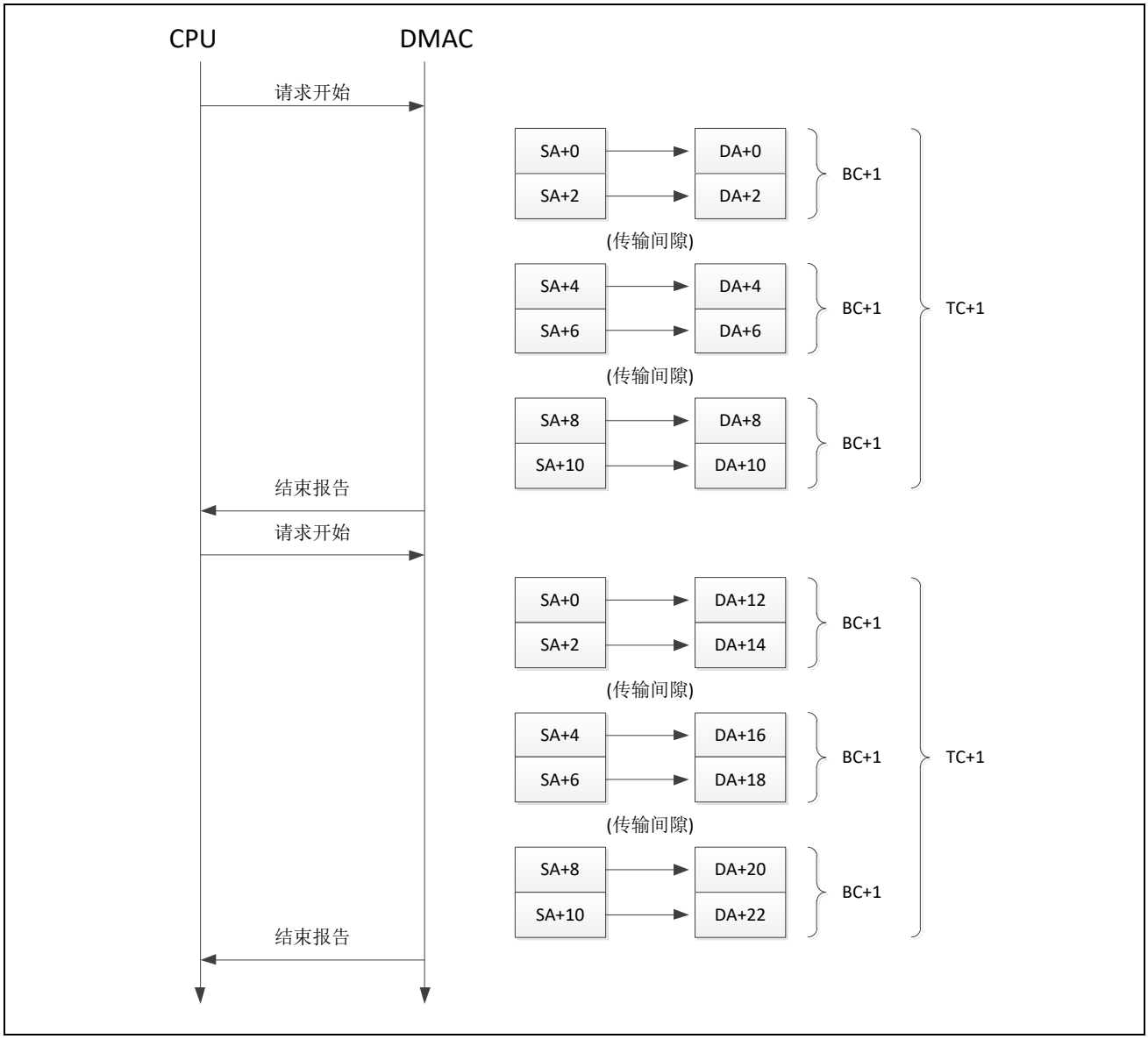


图 14-2 软件-块传输操作示例

通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以半字（16 位）执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在源地址至目的地址区域执行块数 (=BC+1) 传输。
- 块传输时，每完成一次块传输则产生一个传输间隔。
- DMAC 根据传输数量 (=TC+1)和传输块数 (=BC+1)执行数据传输。根据 CPU 传输请求所传输的数据量大小等于“数据位宽 (WIDTH) × 块数 (BC+1) × 传输数量 (TC+1)”。

- 传输完成后，DMAC 通知 CPU 传输完成。
- 如传输完成后再次指示启动传输，因传输源地址已设置为重装 (RS=1)，将从之前的传输源地址重新启动传输。如果未将传输目标地址设置为重装 (RD=0)，将从先前传输结束地址的下一地址开始传输。同样，如果指定了 BC/TC 重装，下次传输将重装与上次传输相同的块数和传输数量值。
- 传输间隔期间不执行传输，插入传输间隔的目的是防止某一条 DMAC 通道占用系统总线的访问权。如多条通道均有传输请求，DMAC 切换在传输间隔计时期间执行传输操作的通道。通过调节 BC 及 TC 的设置可控制传输间隔生成的频率。

此外，传输间隔计时期间总线访问权也转至 CPU。此产品的系统总线为多层配置，带有专用于 DMA 的特殊系统总线。因此，如 CPU 与访问目标之间无冲突，CPU 操作的同时可执行传输。即使 CPU 与访问目标之间发生冲突，只要 DMAC 传输处在不同的地址段（RAM 和外设，或 FLASH 和 RAM 等），CPU 操作几乎不受影响。但如传输处在相同的地址段（RAM 和 RAM 等），CPU 性能可能会受影响，具体取决于所用块数，对此必须引起注意。

14.3.2 软件-突发（Burst）传输

本节说明软件-突发传输。

图 14-3 所示为软件-突发传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求突发传输 (ST=1, TRI_SEL[5:0]=000000,MODE=01)
- 传输源启动地址： SA(SRC_ADR=SA)
- 传输源地址： 固定和可重装 (FS=1, RS=1)
- 传输目标启动地址： DA(DST_ADR=DA)
- 传输目标地址： 递增和不可重装 (FD=0, RD=0)
- 传输数据大小： 字（32 位），块数=3，传输数量=2(WIDTH=10,BC=2,TC=1)
- 传输数量重装： 重装的传输数量 (RC=1)

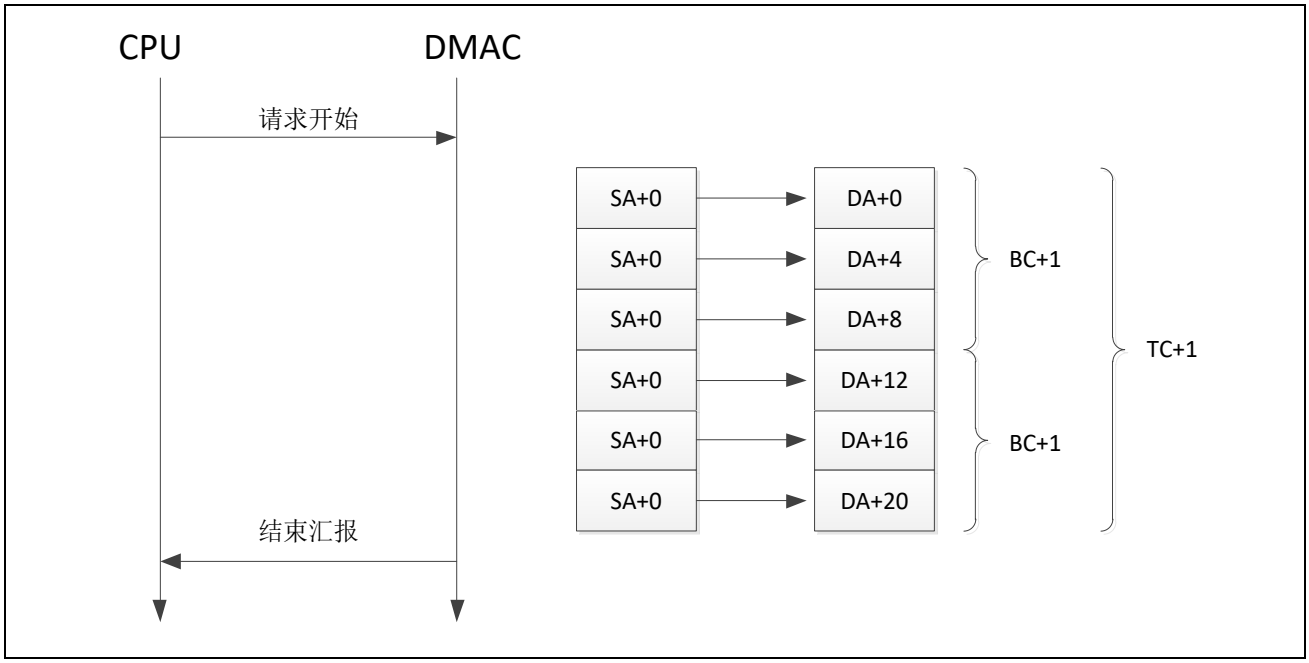


图 14-3 软件-突发传输操作示例

通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以字（32 位）执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在地址 SA 至地址 DA 区域执行块数 (=BC+1) 传输。由于指定了固定的传输源地址，传输源地址与传输源启动地址 (SA+0) 相同。

- 突发传输时，连续执行传输，不产生传输间隔。
- DMAC 根据传输数量 (=TC+1) 执行块数 (=BC+1) 数据传输。根据 CPU 传输请求所传输的数据大小等于"数据位宽 (WIDTH) × 块数 (BC+1) × 传输数量 (TC+1)。"
- 传输完成后，DMAC 通知 CPU 传输完成。

与块传输不同的是突发传输不产生传输间隔。由于受控通道占用了系统总线访问权，该通道可作为优先通道。

14.3.3 硬件-块（Block）传输

本节说明硬件-块传输。

图 14-4 所示为硬件-块传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 硬件-块传输（ST=0，TRI_SEL= 传输请求源的外设，MODE=00）
- 传输数据大小： 块数= 4，传输数量= 3 (BC=3,TC=2)

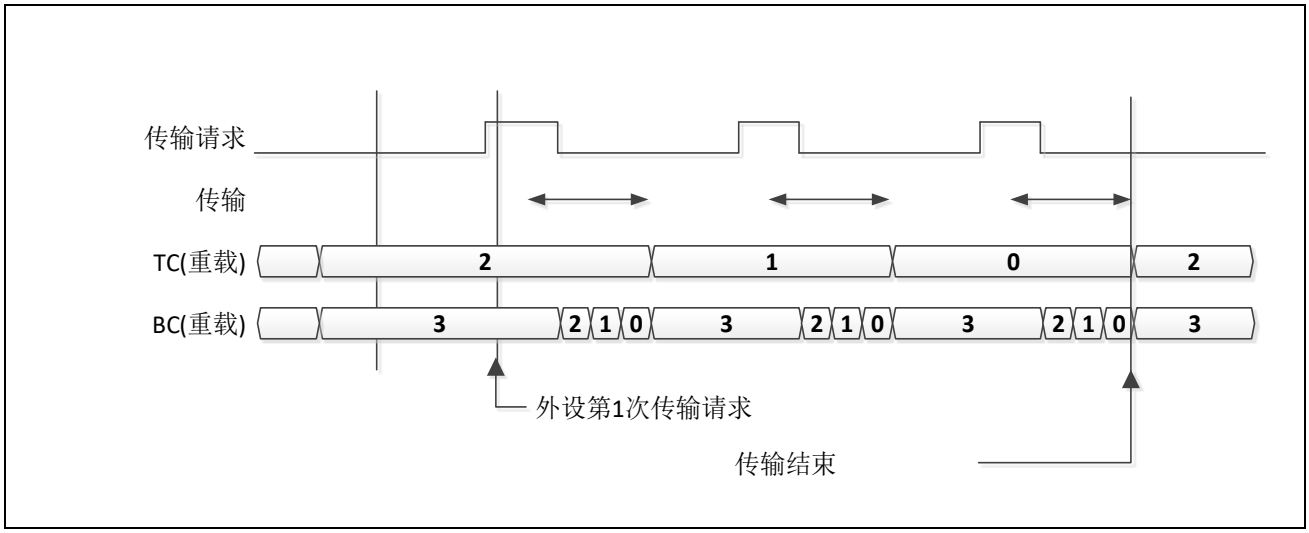


图 14-4 硬件-块传输操作示例

硬件-块传输操作如下：

通过 CPU 指定传输内容，指令启动操作。 DMAC 等待外设的传输请求。收到传输请求后，DMAC 执行块数 (=BC+1) 传输，然后等待下一次传输请求。等待时产生传输间隔。总共收到（TC+1）次传输请求，DMAC 执行 (TC+1)次传输操作。执行的传输总数为 (BC+1)×(TC+1)。外设发出的传输请求必须与 DMAC 传输数量 (TC+1) 相匹配。完成所有传输后，DMAC 通知 CPU 传输完成。

14.3.4 硬件-突发（Burst）传输

本节说明硬件-突发传输。

图 14-5 所示为硬件-突发传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 输请求源的外设，MODE=01）
- 传输数据大小： 块数= 4，传输数量= 5 (BC=3,TC=4)

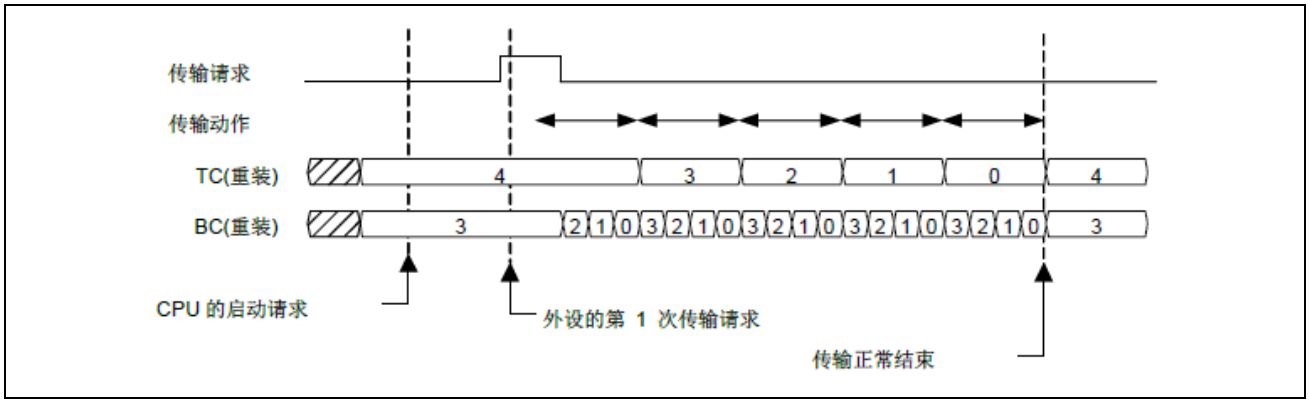


图 14-5 硬件-突发传输操作示例

硬件-突发传输操作如下所述：

通过 CPU 指定传输内容，指令启动操作。 DMAC 等待外设的传输请求。收到第一次传输请求后，DMAC 执行通过 (BC+1)×(TC+1) 计算出的所有次数的传输。硬件-突发传输过程不产生传输间隙。完成所有传输后，DMAC 通知 CPU 传输完成。

14.3.5 通道优先级控制

如 2 个通道均有传输请求，各通道传输间隔计时期间 DMAC 切换其中要传输的通道。此时根据优先控制确定执行传输的下一条通道。可通过 PRIO 从固定优先或循环优先中选择优先控制。

固定优先模式操作 (PRIO=0)

固定优先模式下，所有接到传输请求的通道中最小编号通道优先执行传输操作。（优先顺序：CH0>CH1）

首先执行最高优先级通道的传输（CH0）。最高优先级通道在传输间隔计时期间停止传输操作时，执行第二优先通道的传输操作（CH1）。因此，第一优先通道和第二优先通道交替执行传输操作。

循环优先模式操作 (PRIO=1)

循环优先模式下，所有通道均等地执行其传输操作。

14.4 寄存器

DMAC 基地址：0x4002 1000

寄存器	偏移地址	控制通道	描述
DMAC_CONF	0x00	ALL	所有通道配置寄存器
DMAC_CONFA0	0x10	CH0	通道0配置A寄存器
DMAC_CONFB0	0x14	CH0	通道0配置B寄存器
DMAC_SRCADR0	0x18	CH0	通道0传输源地址寄存器
DMAC_DSTADR0	0x1C	CH0	通道0传输目的地址寄存器
DMAC_CONFA1	0x20	CH1	通道1配置A寄存器
DMAC_CONFB1	0x24	CH1	通道1配置B寄存器
DMAC_SRCADR1	0x28	CH1	通道1传输源地址寄存器
DMAC_DSTADR1	0x2C	CH1	通道1传输目的地址寄存器

14.4.1 DMAC_CONF

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	ST	Res	PRIO	HALT[3:0]				Reserved							
RW	RW	R	RW	RW											

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

位	标记	功能描述
31	EN	该位控制所有通道的使能和禁止传输操作。 当该位置"1"时，使能所有通道的操作，每个通道按照各自设置工作。 当该位清"0"时，所有通道操作都被禁止，直到该位重新置"1"。正在传输的通道强制停止传输。
30	ST	该位指示所有通道的传输状态。 如果下列任何情况在传输中发生，DMAC 将该位置”1”。 – 当"0"写入 CONF:EN 位，并且所有通道传输结束。 – 当除"0000"外的值写入 CONF:HALT 位，并且所有通道传输暂停。 当 CONF:EN=1 和 CONF:HALT=0000 有效，并且所有的通道使能操作，DMAC 将本位清零。 尽管本位的属性是 RW, CPU 写"1"不影响 DMAC 的操作。然而如果 CONF 寄存器需要更新而又不想改变当前位，应该首先读出这个位，然后重新写入同样的值。
29	Reserved	
28	PRIO	PRIO控制各个通道的传输优先级。 当该位置 "0"，各通道优先级固定。CH0 优先级 ＞CH1 优先级。 当该位置 "1"，各通道优先级采用轮询方式决定。
27:24	HALT[3:0]	HALT位控制所有通道的传输暂停或取消。 当这些位设置为"0000"以外的值时，所有正在传输的通道进入暂停状态。当设置为"0000"，所有传输继续。 即便来自外部或者外设的传输请求有效，暂停状态的通道忽略传输请求。如果是块传输和突发传输，即便暂停清除，相关的通道也不启动传输。为了完成传输过程中暂停的传输，需要在暂停取消后发动另外的传输请求。 这些位可以用于暂停传输而不复位该通道的配置寄存器。
23:0	Reserved	

14.4.2 DMAC_CONFA0、DMAC_CONFA1

偏移地址：0x10（DMAC_CONFA0）

0x20（DMAC_CONFA1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENS	PAS	ST	TRI_SEL[6:0]							Reserved		BC[3:0]			
RW	RW	RW	RW									RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC[15:0]															
RW															

位	标记	功能描述
31	ENS	ENS 位控制单独通道的使能和禁止功能。 当该位置"1"，相关通道使能，等待触发，开始传输功能(CONF:EN 必须置 "1")。 如果 MSK 位 (CONFB[0]) 位清 "0"，DMAC 在完成传输后清除本位，即"0"。 当该位为"0"，相关通道禁止，不进行数据传输。同时，如果正好在传输过程中，那么停止传输。 该位可用于强制停止正在传输操作的相关通道，同时复位配置寄存器。
30	PAS	PAS 位控制各自通道的传输暂停或取消。 当该位置 "1" ，相关正在传输的通道进入暂停状态。清除该位到 0，从新开始传输。 该位清除为"0" 当通道的传输结束后。 即便外部或外设传输请求有效，暂停状态通道忽略传输请求。如果是块传输和突发传输，相关通道不会开始传输，即使是清除了暂停位。为了完成暂停的传输，在清除暂停后发送另外的传输请求。 该位可用于暂停传输而不复位相关通道的配置寄存器。
29	ST	ST 位用于产生单独通道的软件传输请求。 当该位置"1"，软件传输触发产生，相关通道开始传输。传输完成后，DMAC 清除该位至"0"。 当该位在传输中置"0"，传输停止。
28:22	TRI_SEL[6:0]	TRI_SEL[6:0]位选择传输请求的触发源。 设置 IS[6:0]为： "0000000" -- 传输触发设定为软件请求 "100000" -- SPI0 RX "1000001" -- SPI0 TX "1000010" -- SPI1 RX "1000011" -- SPI1 TX "1000100" -- ADC JRQ0 RX "1000101" -- ADC SRQ0 RX "1000110" -- LCD TX "1000111" -- Reserved

		"1001000" -- UART0 RX "1001001" -- UART0 TX "1001010" -- UART1 RX "1001011" -- UART1 TX "1001100" -- Reserved "1001101" -- Reserved "1001110" -- Reserved "1001111" -- Reserved "1010000" -- DAC "1010001" -- Reserved "1010010" -- TIM0A "1010011" -- TIM0B "1010100" -- TIM1A "1010101" -- TIM1B "1010110" -- TIM2A "1010111" -- TIM2B "1011000" -- TIM3A "1011001" -- TIM3B "1011010" -- TIM4A "1011011" -- TIM4B "1011100" -- TIM5A "1011101" -- TIM5B "1011110" -- TIM6A "1011111" -- TIM6B "1100000" -- UART4 RX "1100001" -- UART4 TX "1100010" -- UART5 RX "1100011" -- UART5 TX
21:20	Reserved	
19:16	BC[3:0]	BC[3:0]位指定块/突发传输的个数。 设置为请求传输时，设置"0000"。 设置"BC[3:0]=块数量 - 1"。最大允许传输的块数为 16。 这几位在传输的工程中可以去读。通常情况，当一个源地址访问，或者一个目的地址的访问成功完成，BC[3:0] 减去 1。 – 在 CONFB:RC=1 的情况下： 传输开始时设置的值，在传输完成后重载。 – 在 CONFB:RC=0 的情况下： 成功完成传输后，该位清零，同时在传输暂停引起的传输非成功完成时，该值保持不变。
15:0	TC[15:0]	TC[15:0] 位用于指定块/突发/请求传输的传输次数。 设定值"TC = 传输数量 - 1". 最大传输数为 65536。 传输时，这些值是可以读的。正常情况，当一次块传输完成，TC 减去 1。 – 在 CONFB:RC=1 的情况下

		传输开始时设置的值，在传输完成后重载。 – 在 CONFB:RC=0 的情况下 成功完成传输后，该位清零，同时在传输暂停引起的传输非成功完成时，该值保持不变。

14.4.3 DMAC_CONFB0、DMAC_CONFB1

偏移地址：0x14 （DMAC_CONFB0）

0x24 （DMAC_CONFB1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		MODE[1:0]		WIDTH[1:0]		FS	FD	RC	RS	RD	ERR_IE	FIS_IE	STAT[2:0]		
		RW		RW		RW	RW	RW	RW	RW	RW	RW	RW		

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

位	标记	功能描述
31:30	Reserved	
29:28	MODE[1:0]	该位控制传输模式： 00：Block 传输（初始值） 01：Burst 传输 10：Reserved 11：Reserved
27:26	WIDTH[1:0]	该位指示传输数据的宽度： 00：字节（8-Bit）（初始值） 01：半字（16-Bit） 10：字（32-Bit） 11：Reserved
25	FS	该位用于指定源地址自增或者固定： 0：按 CONFB:WIDTH[1:0]的数据位宽自增传输源地址（初始值） 1：固定传输源地址
24	FD	该位用于指定目的地址自增或者固定： 0：按 CONFB:WIDTH[1:0]的数据位宽自增传输目的地址（初始值） 1：固定传输目的地址
23	RC	该位控制CONFA:BC[3:0]和CONFA:TC[15:0]的重载功能。 当该位置”1”时，CONFA:BC[3:0]和 CONFA:TC[15:0]在传输完成后重新加载传输开始时的设定值。 0：禁止重新加载 BC/TC 值（初始值） 1：使能重新加载 BC/TC 值
22	RS	该位控制传输源地址的重载功能 0：禁止重新加载传输源地址（初始值） 1：使能重新加载传输源地址
21	RD	该位控制传输目的地址的重载功能

		0：禁止重新加载传输目的地址（初始值） 1：使能重新加载传输目的地址
20	ERR_IE	该位使能或禁止在传输未成功完成中断的产生。 0：禁止传输出错时产生中断（初始值） 1：使能传输出错时产生中断 当该位设置为”1”时，在如下情况下，传输出错中断发出。 – 地址溢出 CONFB:STAT[2:0]=”001” – 由外设发出的停止传输引起的停止，或者设置 DMAC_CONF:EN、DDMAC_CONFA:ENS 引起的禁止传输 CONFB:STAT[2:0]=010 – 传输源地址访问错误 CONFB:STAT[2:0]=011 — 传输目的地址访问错误 CONFB:STAT[2:0]=100
19	FIS_IE	该位使能或禁止传输成功完成中断的产生。 0：禁止传输成功完成时产生中断（初始值） 1：使能传输成功完成时产生中断 当该位置”1”时，如果 CONF:STAT 在传输完成时为成功完成传输，中断发生。
18:16	STAT[2:0]	该位使用3-bit代码指示传输的当前状态。 如果一个成功传输完成中断或者传输非成功完成中断发生，中断信号可以通过写“000”到该位来清除当前状态。 000 – 初始值 001 – 传输错误引起中止（地址溢出） 010 – 传输停止请求引起中止 (外设停止请求引起的停止或者 EB/DE 位引起的禁止传输) 011 – 传输错误引起中止 (传输源地址访问错误) 100 – 传输错误引起中止 (传输目的地地址访问错误) 101 – 成功传输完成 110 – 保留 111 -- 传输暂停
15:1	Reserved	
0	MSK	该位用于在传输完成时，屏蔽来自 DMAC 的 EB 位(CONFA[31])清除功能。 – 如果 MSK=0， DMAC 在传输完成时清除 CONFA:ENS 位。 – 如果 MSK=1， DMAC 在传输完成时不清除 CONFA:ENS 位。这个功能允许连续传输而不需要 CPU 干预。 这个功能用于硬件传输。使用这个功能时，可同时使能重载功能CONFB:RC，CONFB:RS和CONFB:RD位。

14.4.4 DMAC_SRCADR0、DMAC_SRCADR1

偏移地址：0x18（DMAC_SRCADR0）

0x28（DMAC_SRCADR1）

复位值：0x0000 0000

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

位	标记	功能描述
31:0	SRCADR[31:0]	SRCADR[31:0]位指定传输源地址。 不能设置相对于传输数据位宽 (CONFB:WIDTH[1:0])非对齐的地址。传输时可读取这些位的值。 – 如果 CONFB:FS="1"，传输源地址设定为固定值，并且不会改变。 – 如果 CONFB:FS="0"，且 CONFB:RS="0"，传输时源地址根据 CONFB:WIDTH[1:0]的数据位宽自增。如果传输成功完成，这个值为传输完成地址的下一地址。如果传输未成功完成，这个值为暂停时所设定的值。 – 如果 CONFB:FS="0"且 CONFB:RS="1"，传输时源地址根据 CONFB:WIDTH[1:0]的数据位宽自增。传输完成后，传输开始时设定的源地址重新加载到本寄存器位中。

14.4.5 DMAC_DSTADR0、DMAC_DSTADR1

偏移地址：0x1C（DMAC_DSTADR0）

0x2C（DMAC_DSTADR1）

复位值：0x0000 0000

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

位	符号	描述
31:0	DSTADR[31:0]	DSTADR[31:0] 位指定传输目的地址。 不能设置相对于传输数据位宽 (CONFB:WIDTH[1:0])非对齐的地址。传输时可读取这些位的值。 – 如果 CONFB:FD="1"，传输目的地址设置为固定地址，并且不改变。 – 如果 CONFB:FD="0"，且 CONFB:RD="0", 传输时目的地址根据 CONFB:WIDTH[1:0]的数据位宽自增。如果传输成功完成，这个值为传输完成地址的下一地址。如果传输未成功完成，这个值为暂停时所设定的值。 – 如果 CONFB:FD="0" 且 CONFB:RD="1", 传输时目的地址根据 CONFB:WIDTH[1:0]的数据位宽自增。传输完成后，传输开始时设定的目的地址重新加载到本寄存器位中。

15 通用定时器（TIM0/1/2/3）

15.1 通用定时器简介

TIM0/1/2 是分别由 1 个计数单元和 2 个比较单元组成的定时器。每个定时器支持 2 个独立 PWM 输出或 1 对互补 PWM 输出和 1 个独立 PWM 输出。支持 2 个捕获输入。TIM0/1/2 可以组成 3 对互补 PWM 输出。

TIM3 是由 1 个计数单元和 6 个比较单元组成的定时器，支持 6 个独立 PWM 输出或 3 对互补 PWM 输出。支持 6 个捕获输入。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

15.1.1 基本特性(TIM0/1/2)

- 2 路独立 PWM 输出 CHA,CHB,
- 1 路互补 PWM 输出（CHA,CHB）+1 路独立 PWM 输出（gate）
- 1 路互补 PWM 输出（CHA,CHB）+1 路捕获功能(gate)
- 最多 2 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发

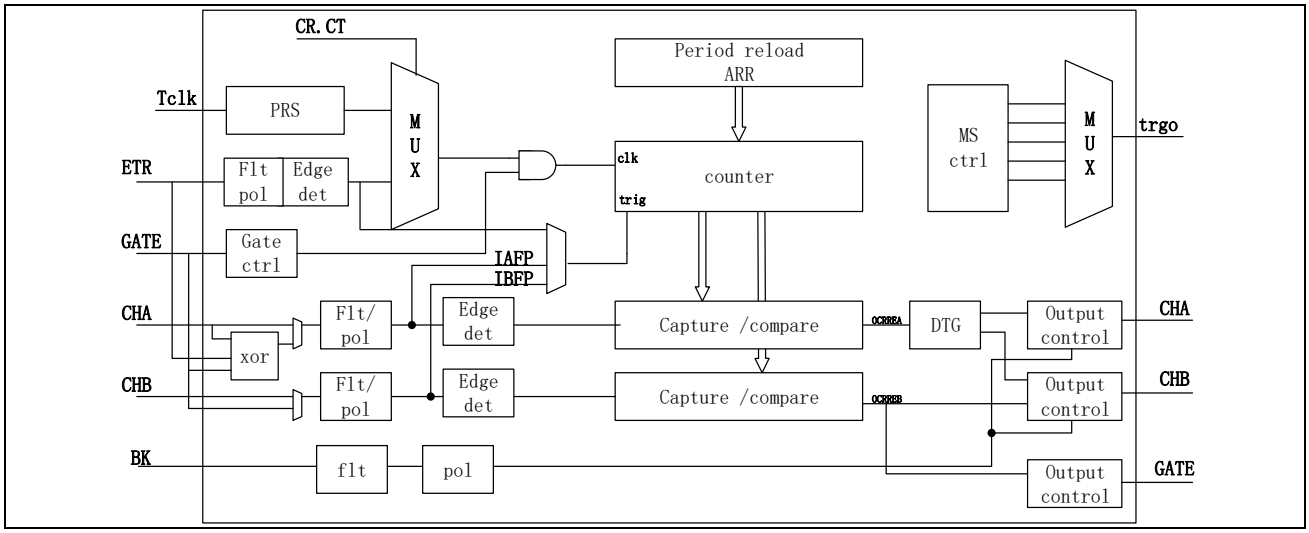


图 15-1 TIM0/1/2 框图

15.1.2 基本特性(TIM3)

- 6 路独立 PWM 输出 CH0A,CH0B, CH1A,CH1B, CH2A,CH2B
- 3 路互补 PWM 输出（CHxA,CHxB）+1 路独立 PWM 输出（gate）
- 3 路互补 PWM 输出（CHxA,CHxB）+1 路捕获功能(gate)
- 最多 6 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发

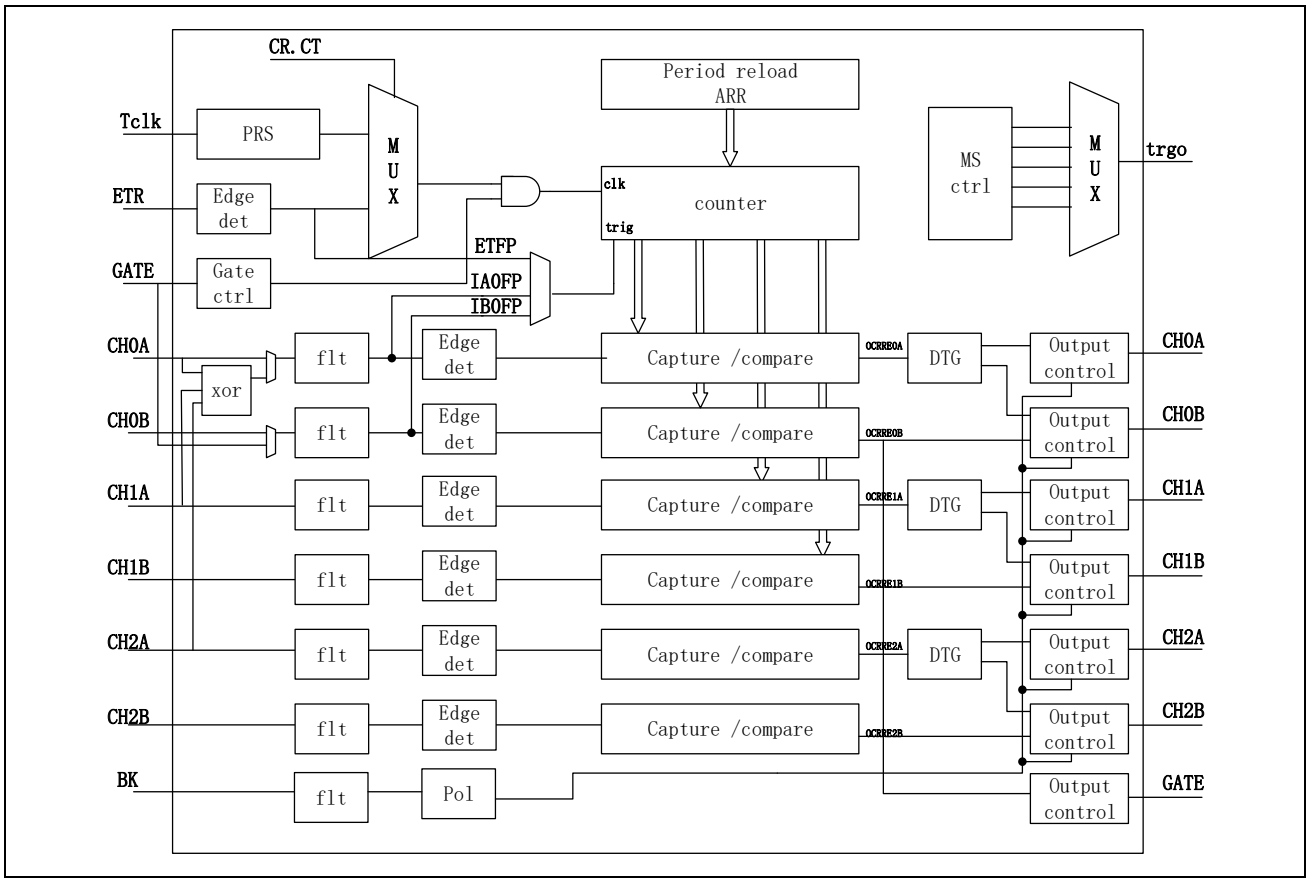


图 15-2 TIM3 框图

15.2 Timer 功能描述

15.2.1 定时器时钟

定时器使用 PCLK 作为定时器时钟，下文定时器时钟使用 TCLK 标示。

15.2.2 定时计数器

可编程通用定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

15.2.3 定时器预除频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

15.2.4 模式 0 计数定时器功能

这种模式下，计数器向上计数。计数器支持两种计数模式，重载模式与自由计数模式，可以选择外部时钟 ETR 计数或系统时钟计数。门控 Gate 可以控制计数屏蔽。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB，这种模式下 CHA,CHB 反相。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出，然后从 ARR 再开始计数，计数周期为 0Xffff-ARR+1；自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

计数器可用通过外部门控功能控制计数器是否计数。控制关系如下

M0CR.GATE	M0CR.GATEP	端口 GATE 输入	M0CR.CTEN	计数器
x	x	x	0	不计数
0	x	x	1	计数
1	0	高电平	1	计数
1	0	低电平	1	不计数
1	1	高电平	1	不计数
1	1	低电平	1	计数

15.2.4.1 功能框图

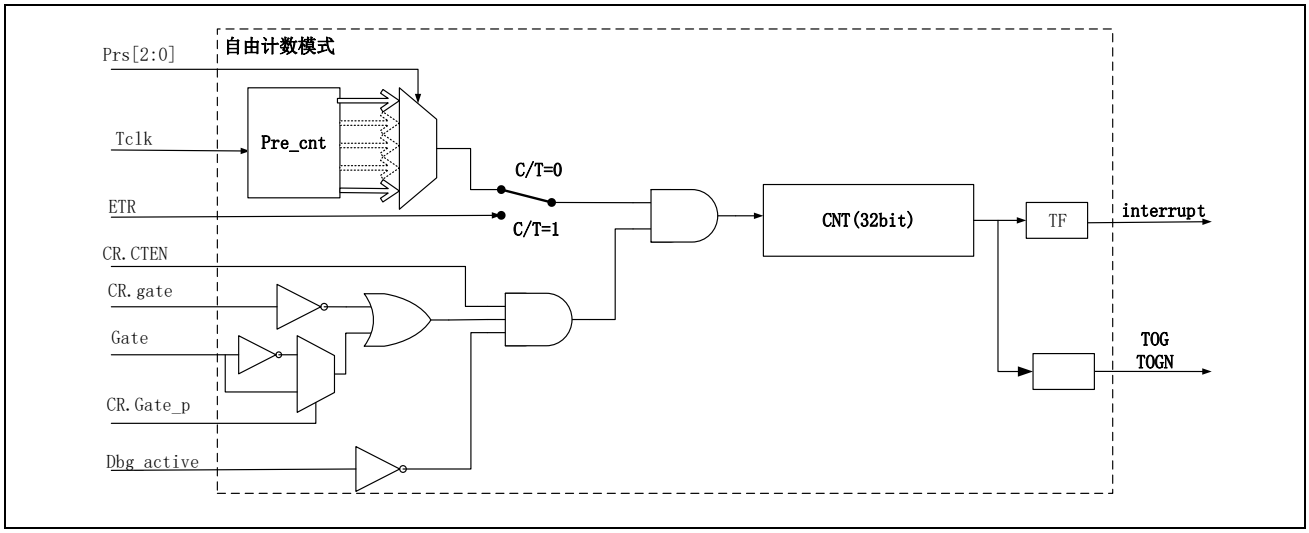


图 15-3 自由计数框图

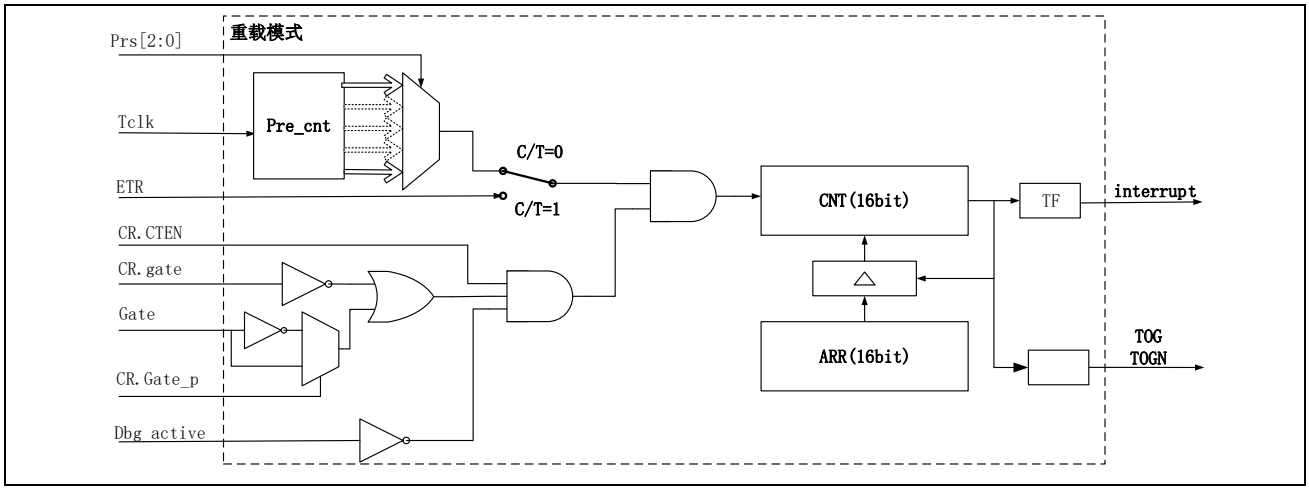


图 15-4 重载计数波形

15.2.4.2 计数波形

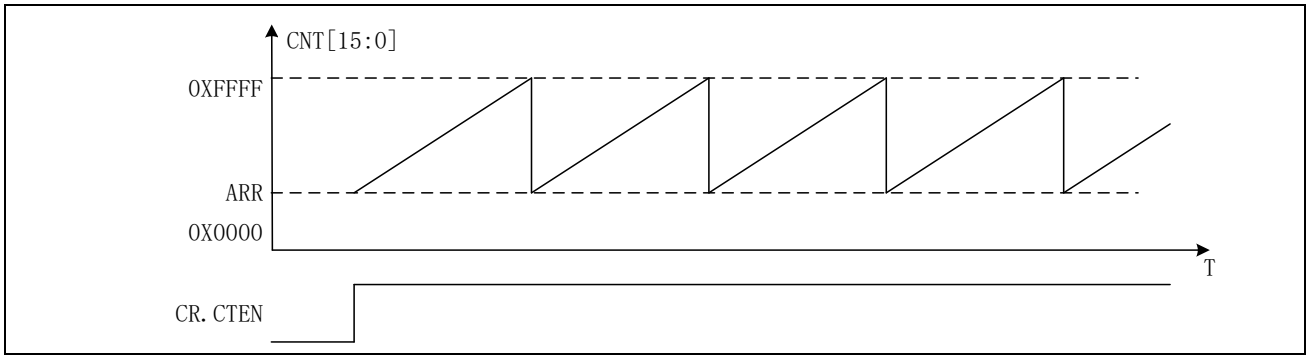


图 15-5 16位重载计数波形

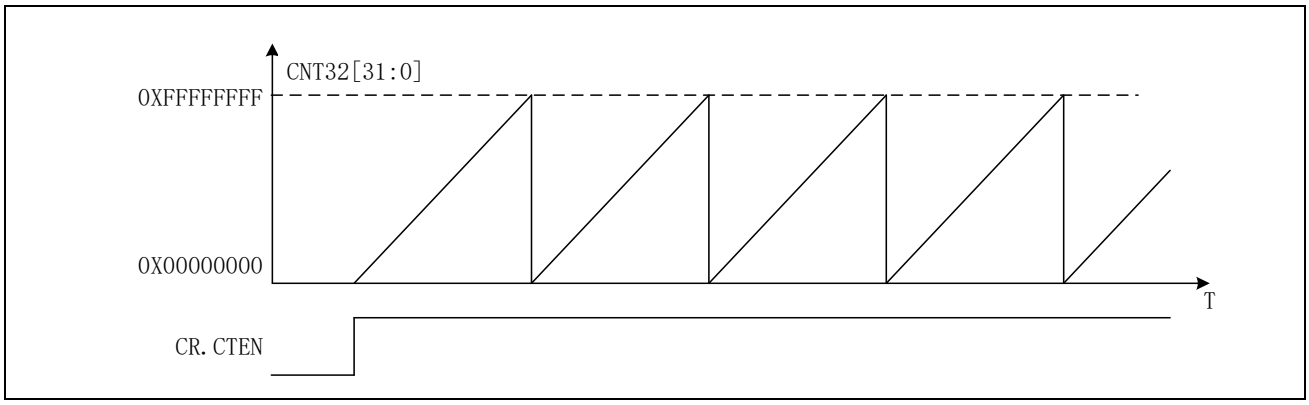


图 15-6 32位自由计数波形

15.2.4.3 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 tclk 采样，因此外部输入时钟频率不能超过系统的 tclk 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

15.2.4.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

15.2.4.5 时序图

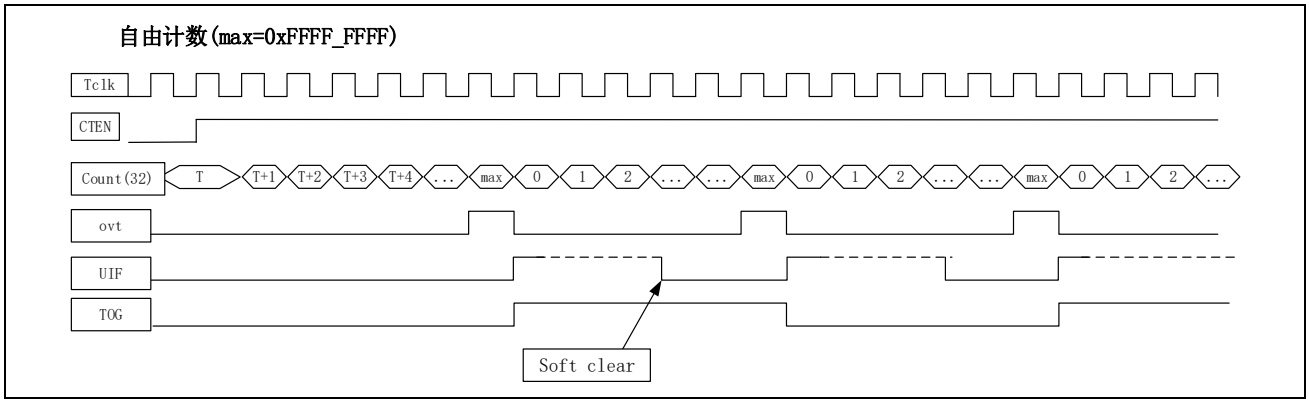


图 15-7 自由计数时序图

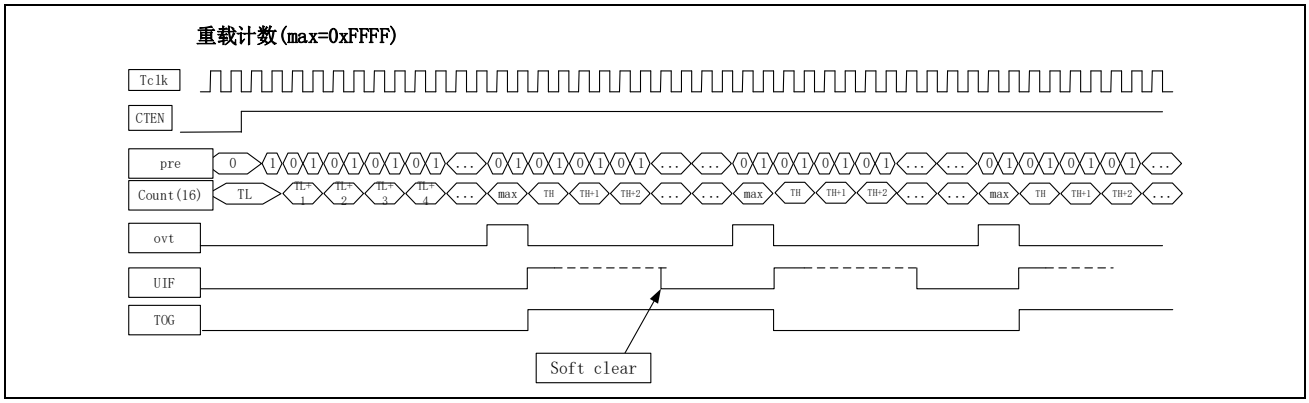


图 15-8 重载计数时序图（预分频设置为2）

15.2.4.6 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 CR.TOG_EN 为 0 可以同时设置端口 CHA,CHB 输出为 0。

在计数时钟为 4M 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNTL 初始值	CNTH 重载值
1000Hz	0.5ms	2000	63536	0XF830	0XF830
2000Hz	0.25ms	1000	64536	0XFC18	0XFC18
4000Hz	0.125ms	500	65036	0XFE0C	0XFE0C

15.2.4.7 设置示例

重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

门控外部时钟自由计数设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置计数器初值 CNT
3. 使能门控功能 M0CR.GATE
4. 选择门控有效电平 M0CR.GATEP
5. 清除中断标志
6. 使能中断 M0CR.UIE
7. 使能外部时钟模式 M0CR.CT
8. 开启定时器 M0CR.CTEN

BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

15.2.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。

第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CMAR，并且产生捕获中断 CAF，如果计数器发生溢出，会产生溢出标志。设置溢出中断使能会产生溢出中断。

M1CR.edg1st	0	0	1	1
M1CR.edg2nd	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

15.2.5.1 PWC 功能框图

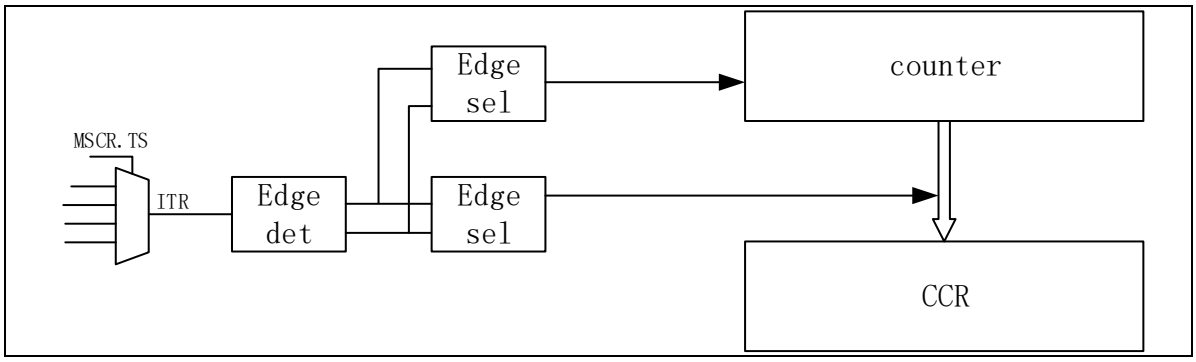


图 15-9 PWC测量框图

MSCR.TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：无效 110：端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效） 111：端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）

15.2.5.2 PWC 波形测量时序图

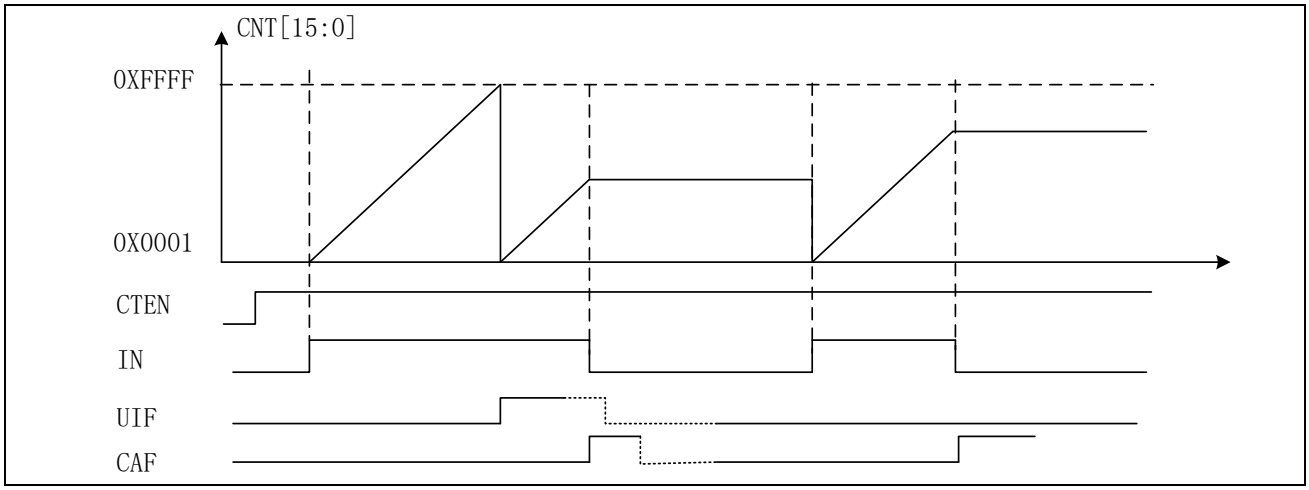


图 15-10 高电平脉冲宽度测量

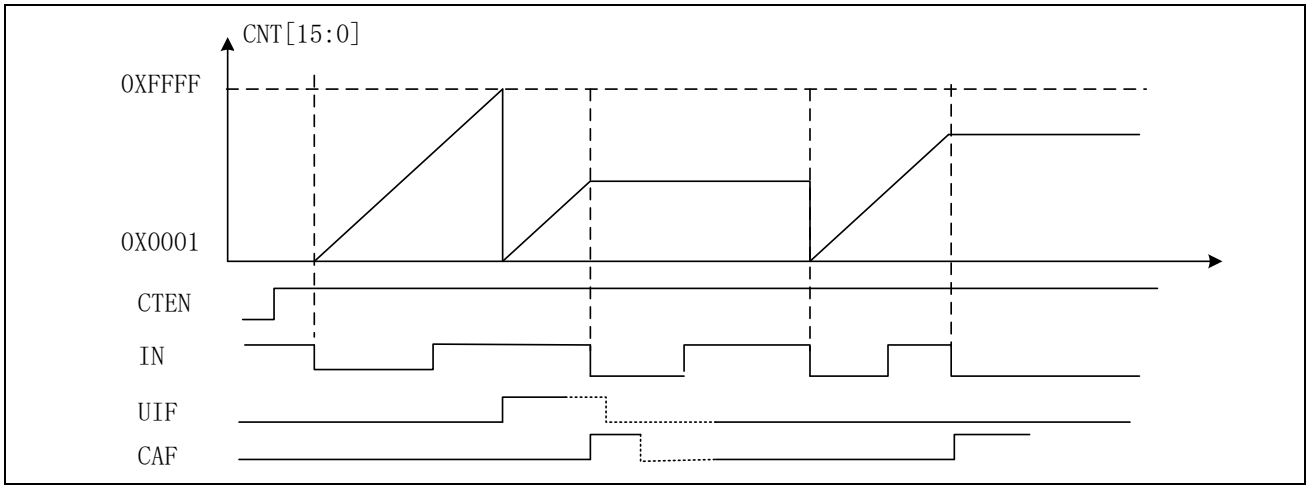


图 15-11 下降沿到下降沿周期测量

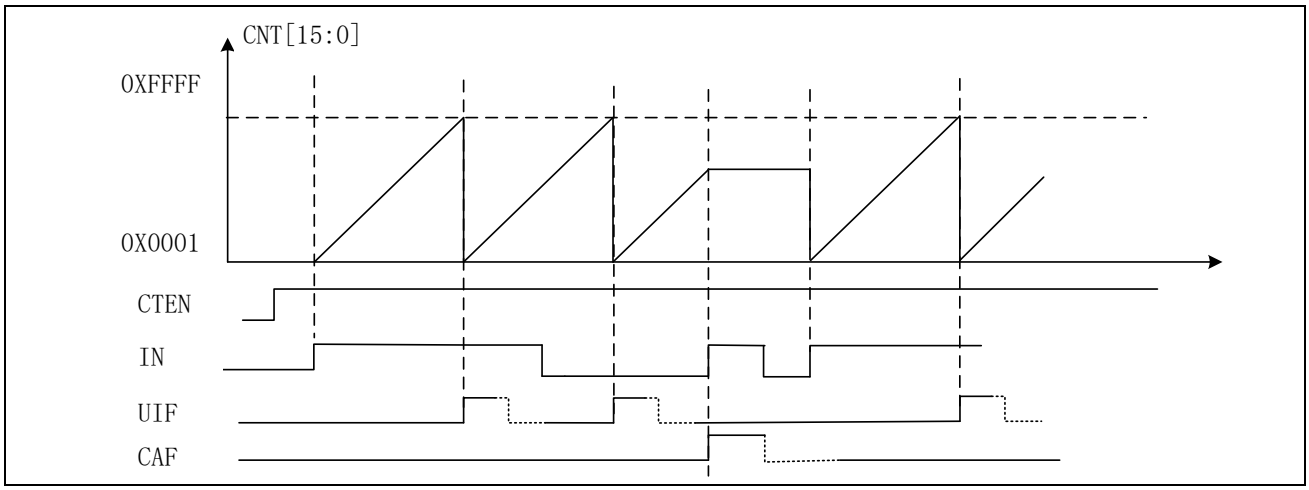


图 15-12 上升沿到上升沿周期测量

通过寄存 MSCR.TS 选择测量信号源。

000 ETFP： ETR 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

001 ITR0： Timer 内部互联信号 0，其他 timer 的 TRGO 输出

010 ITR1： Timer 内部互联信号 1，其他 timer 的 TRGO 输出

011 ITR2： Timer 内部互联信号 2，其他 timer 的 TRGO 输出

100 ITR3： Timer 内部互联信号 3，其他 timer 的 TRGO 输出

101 IA0ED： 无效

110 IAFP： CH0A 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

111 IBFP： CH0B 外部输输入滤波后的相位选择信号，可选择外部滤波与输入反向

	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

注：关于 TRGO 输出参考寄存器描述。

15.2.5.3 PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量，测量完成后 CTEN 将被清除。

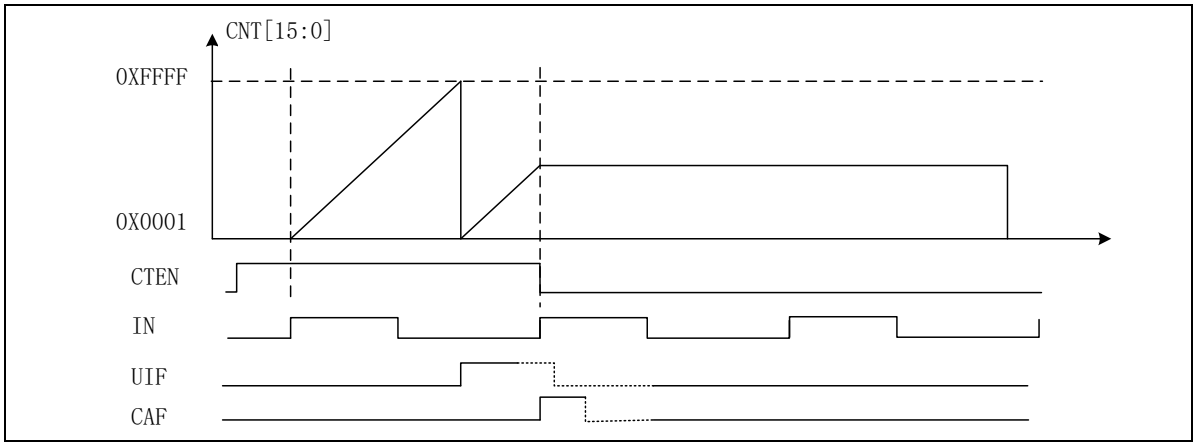


图 15-13 上升沿到上升沿周期测量单次模式

15.2.5.4 设置示例

脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE
6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN
8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
9. 等待下次测量

脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量低电平
5. 清除中断标志
6. 使能溢出中断 M1CR.UIE

7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

15.2.6 模式 2/3 比较捕获模式

15.2.6.1 计数器

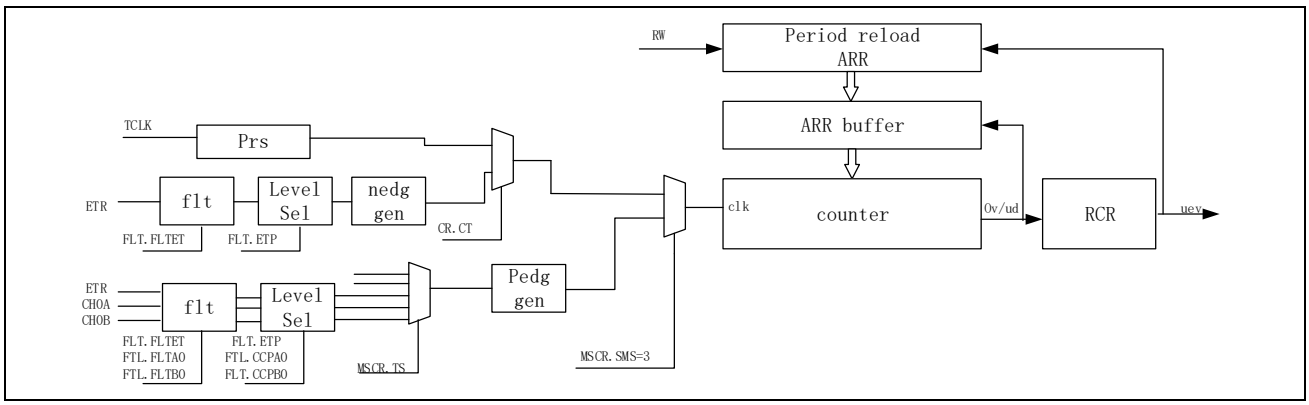


图 15-14 计数器框图

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

- 计数器基本单元包括：
- 计数器寄存器 CNT
- 预分频寄存器 CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLT,CR0,MSCR,CR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并行缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才有自动装载寄存器更新到缓存寄存器。

时钟选择及门控功能，触发功能，复位功能参考模式 2/3 从模式章节。

15.2.6.2 计数器波形

模式 2 为锯齿波计数波形，通过设置 CR.DIR 可以更改计数方向。

设置 CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值（TIMx_ARR），然后重新 从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次(TIMx_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_CR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_IFR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下图示显示 ARR=0X2C 时不同计数方向的计数器波形

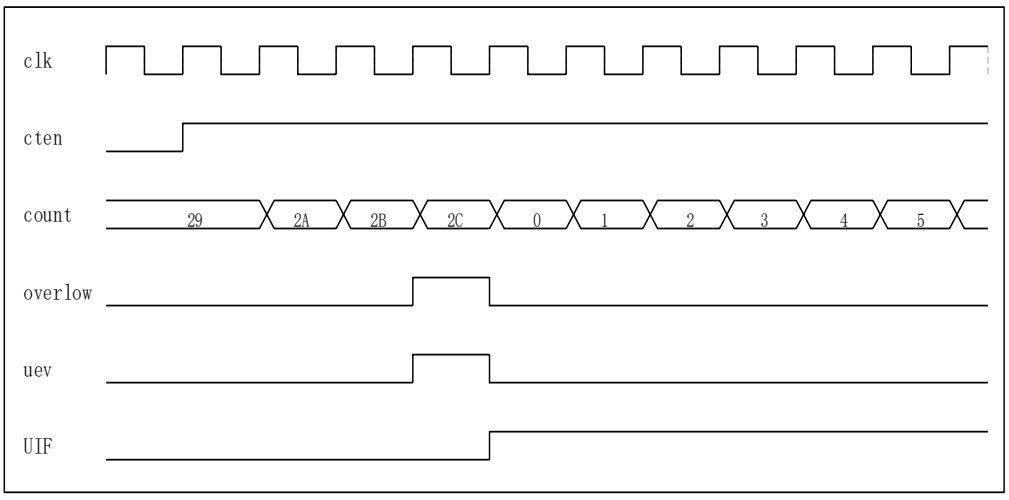


图 15-15 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1，

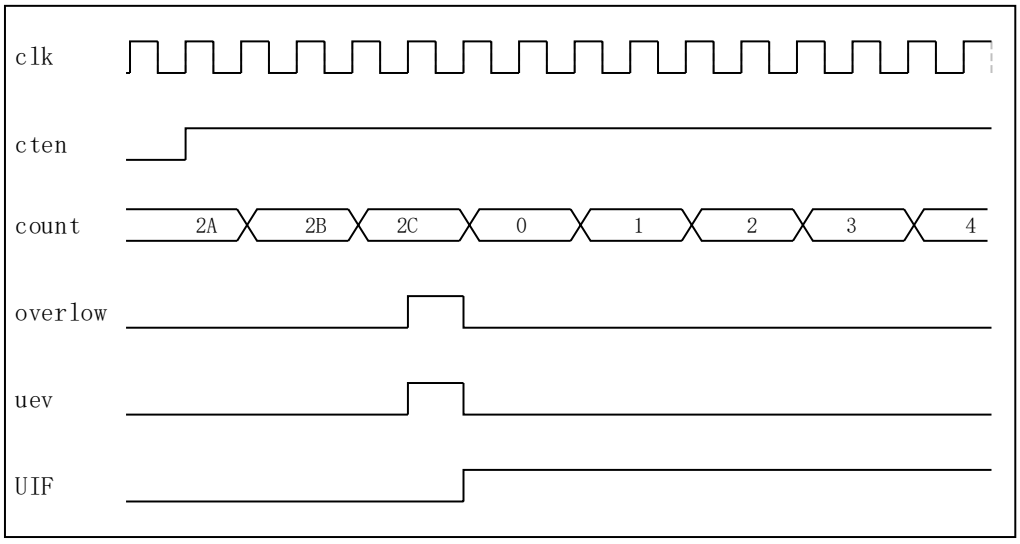


图 15-16　带预分频的上计数

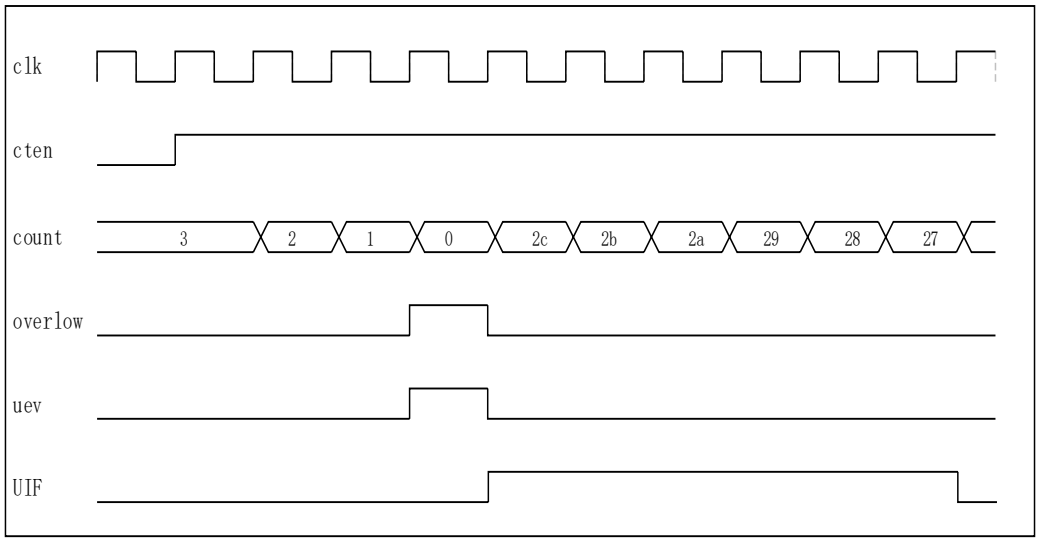


图 15-17　不带预分频的下计数

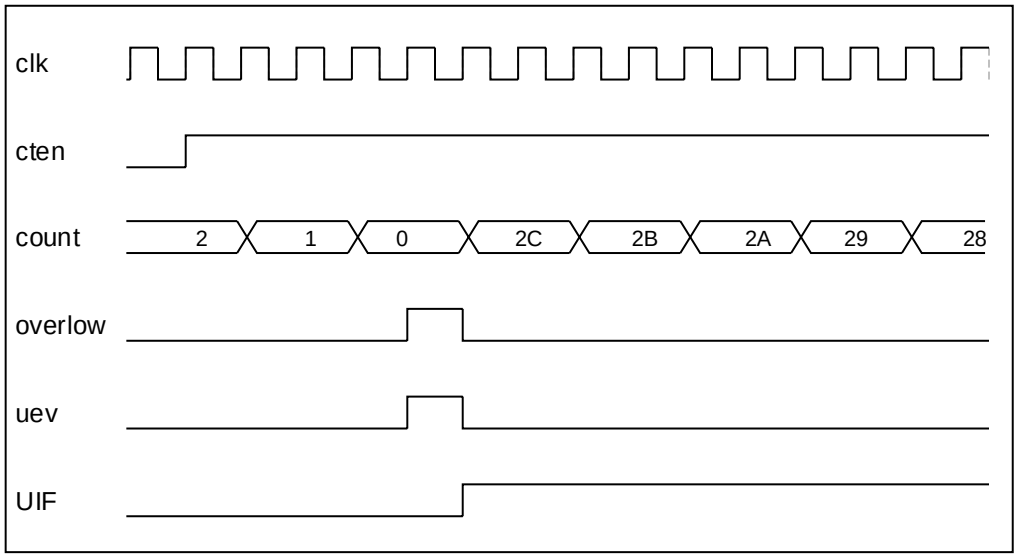


图 15-18 带预分频的下计数

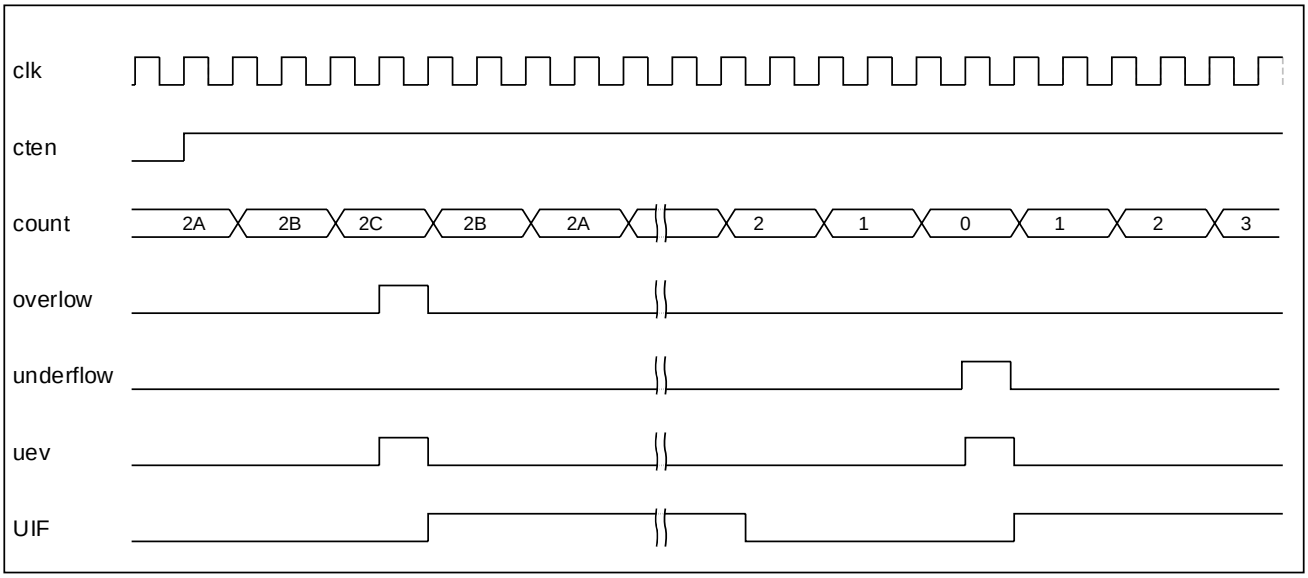


图 15-19 带预分频的上下计数

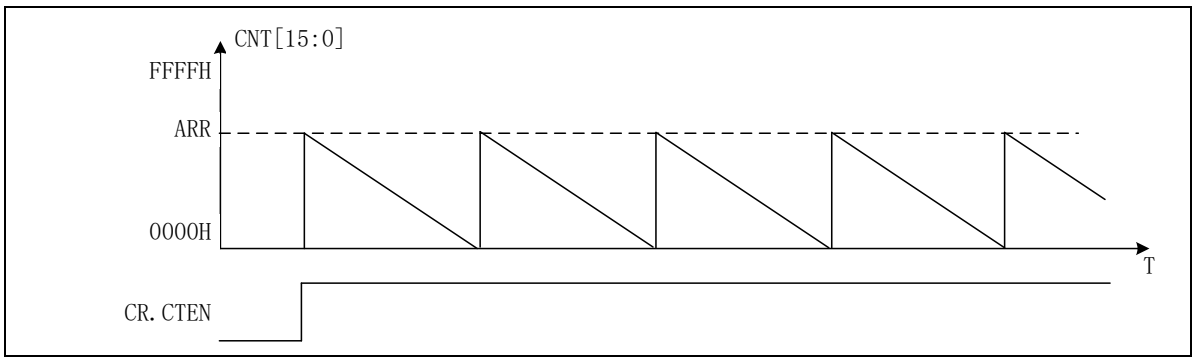


图 15-20 边沿对齐计时器波形(DIR =1)

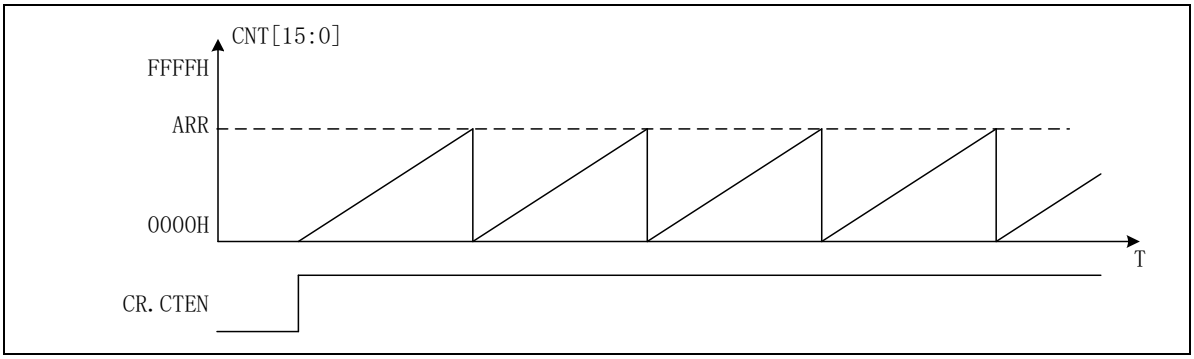


图 15-21 边沿对齐计时器波形(DIR =0)

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 CR.DIR 方向位是只读的。写值无效。从其他模式切换到中心对齐模式DIR 自动清 0。软件事件更新和从模式外部触发复位模式DIR 自动清零。

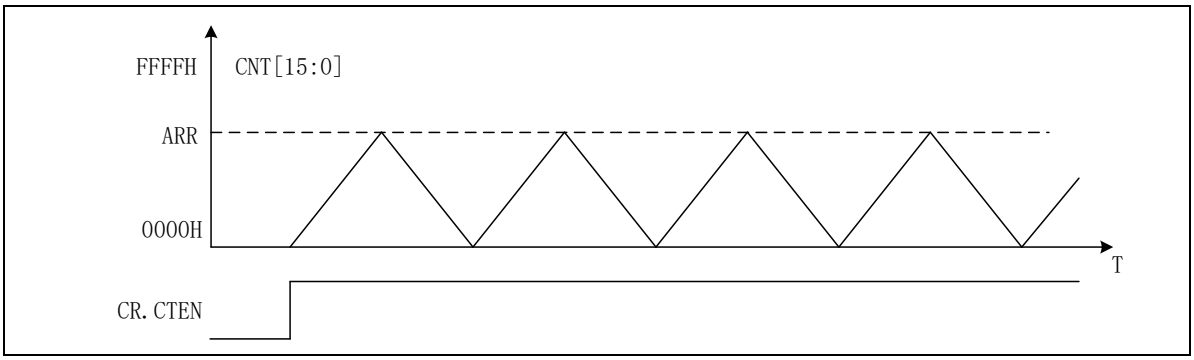


图 15-22 中心对齐计数器波形

15.2.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存的值更新到比较缓存寄存中。

重复计数器在下面条件成立时递减

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

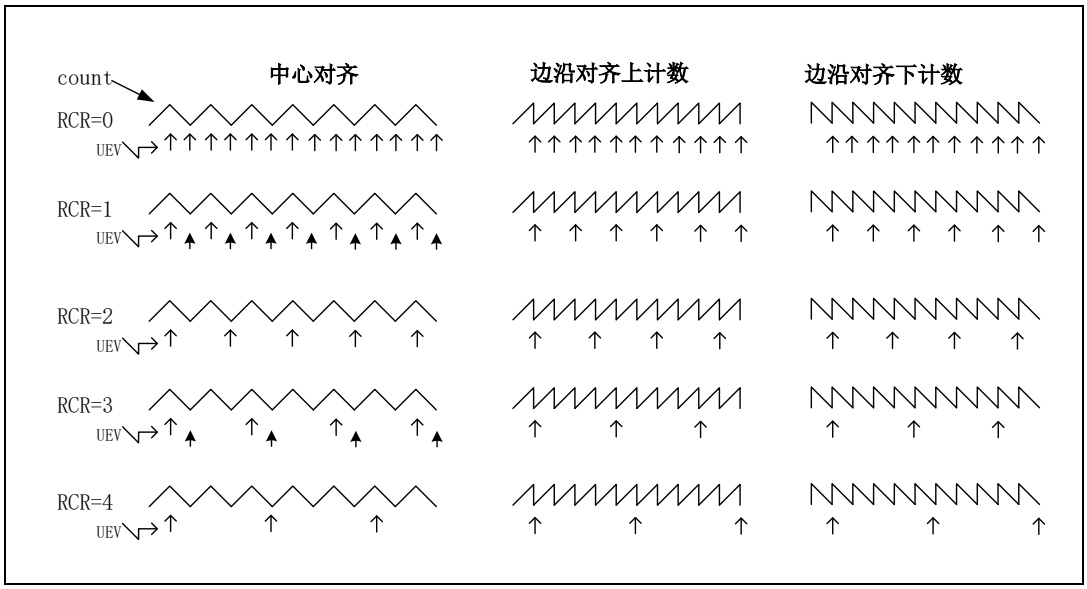


图 15-23 重复计数器产生更新时序

除了上下溢出通过重复计数器可以产生事件更新 UEV 外，还可以通过写寄存器 CR.UG 产生软件及从模式复位事件更新 UEV；这时需要配置 CR.URS。

15.2.6.4 数据缓存

自动重载数据 ARR 与比较寄存器都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCR 才会生效。

自动重载值在不同计数模式下的更新时序图如下

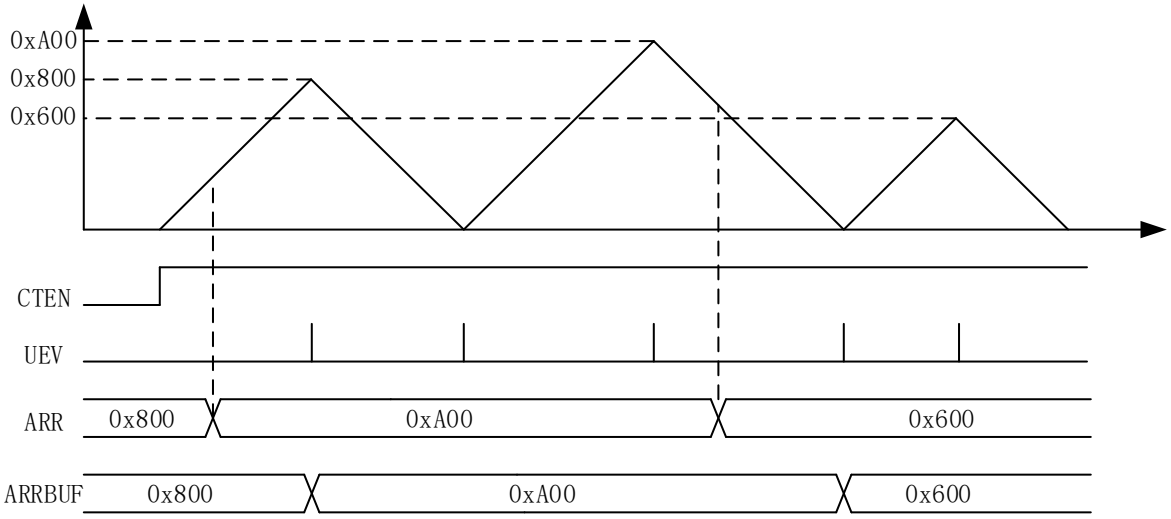


图 15-24 三角波模式下缓存使能

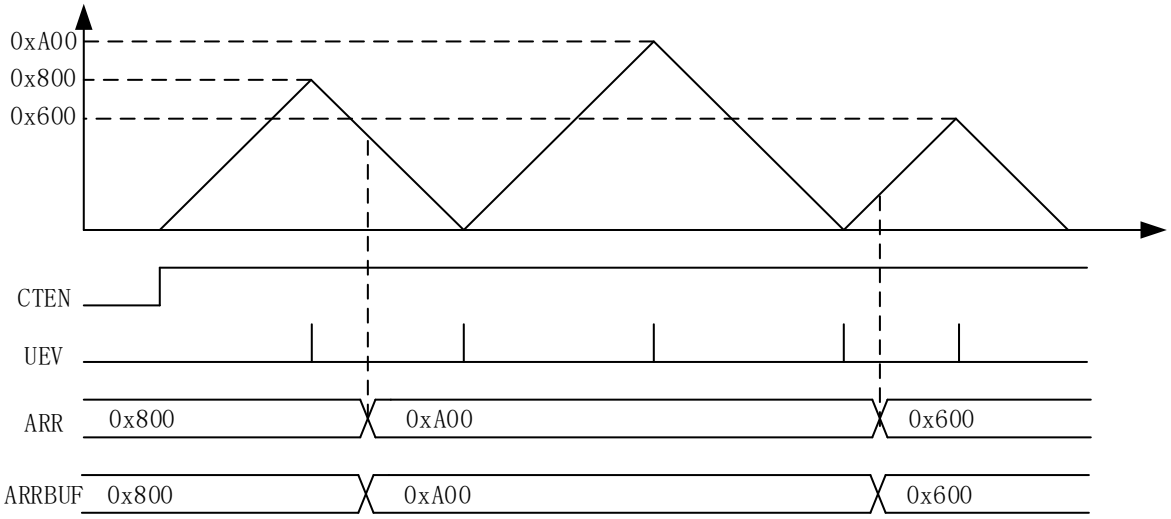


图 15-25 三角波模式下缓存无效

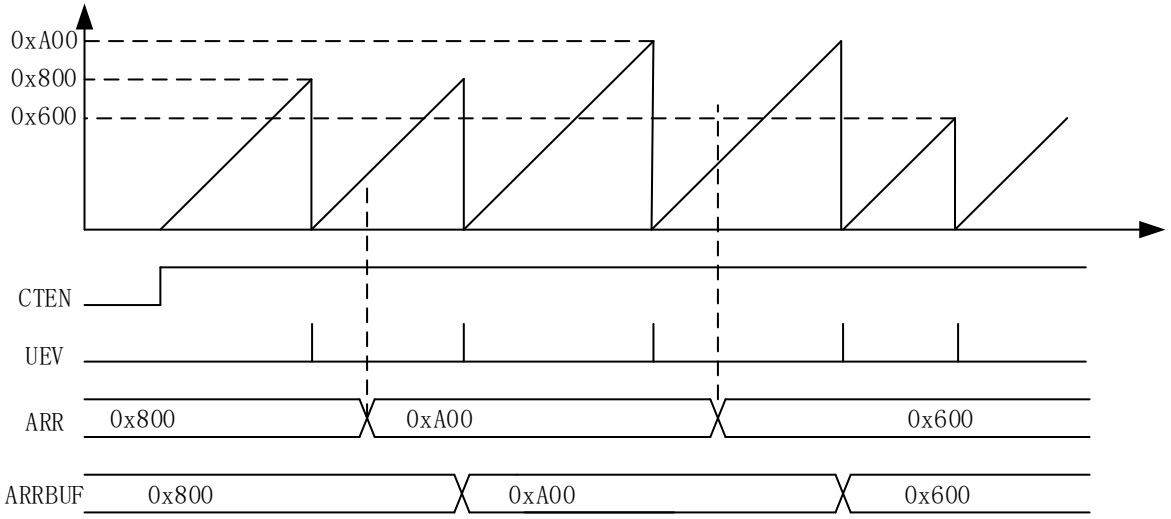


图 15-26 锯齿波模式下上计数缓存使能

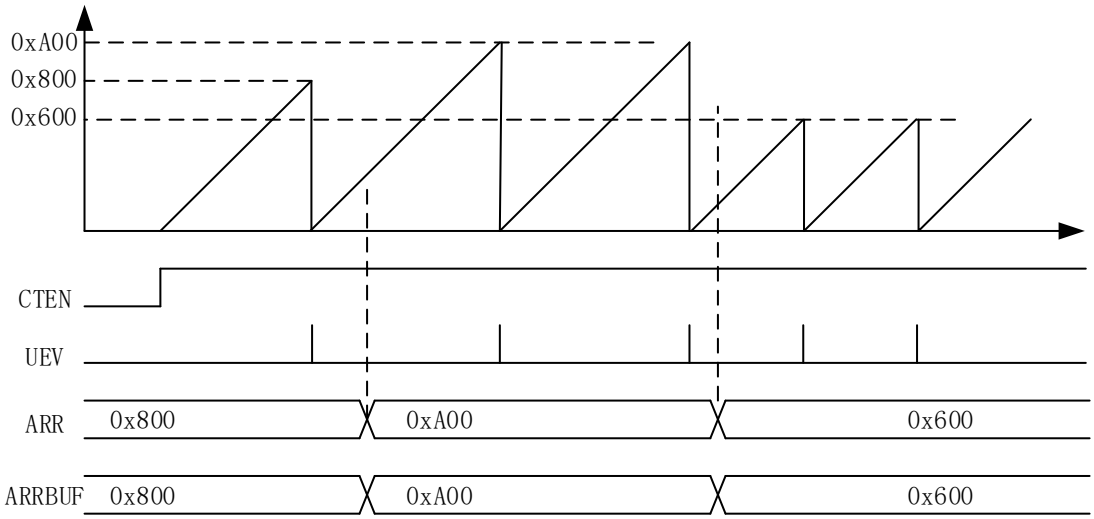


图 15-27 锯齿波模式下上计数缓存无效

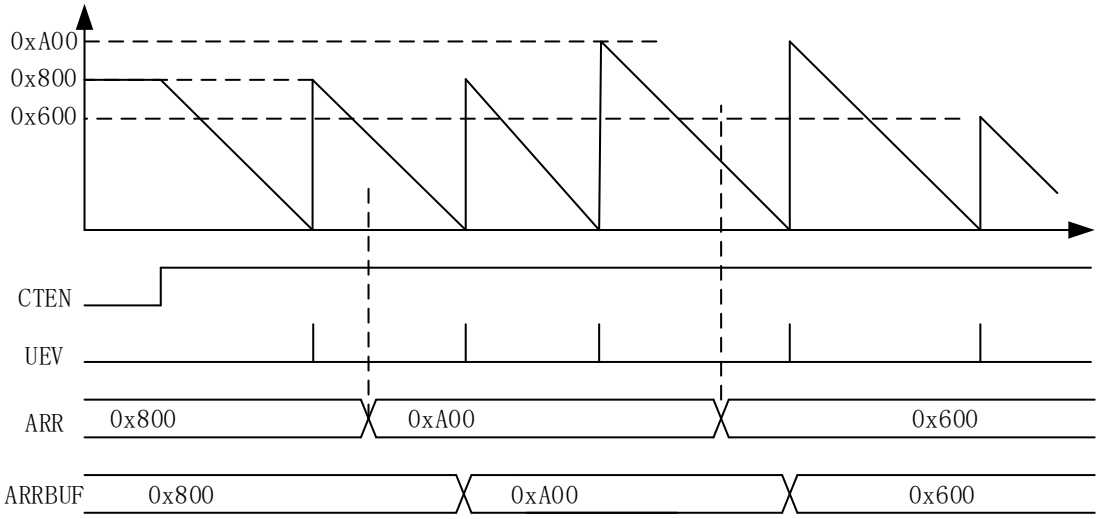


图 15-28　锯齿波模式下计数缓存使能

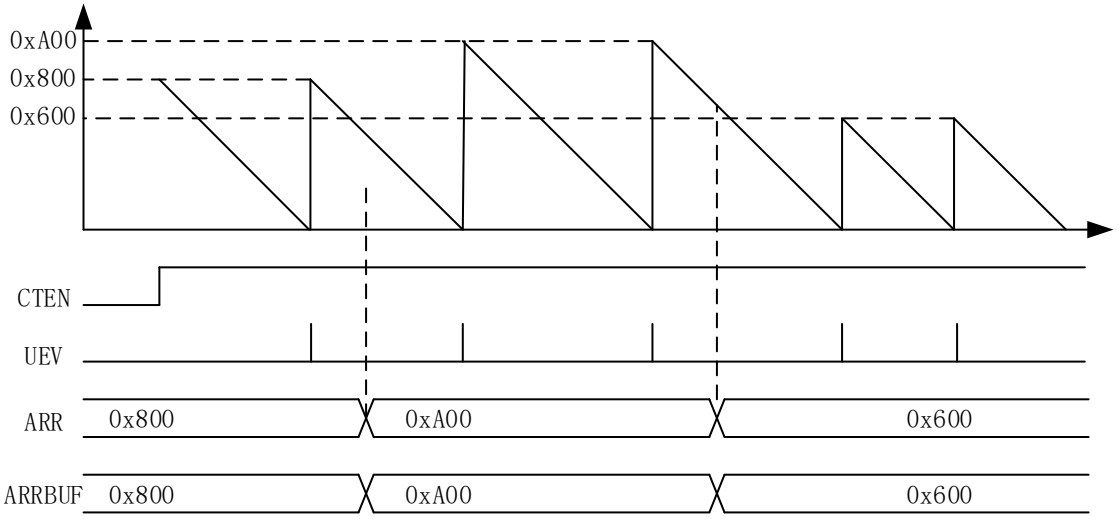


图 15-29　锯齿波模式下计数缓存无效

在三角波模式与锯齿波上计数模式时，如果缓存不使能，更改的 ARR 时，当前计数器的值要小于要更改的 ARR 周期值，否则当前周期会计数到 0XFFFF。

比较缓存与周期缓存更新状态一致，这里不一一列出时序图。

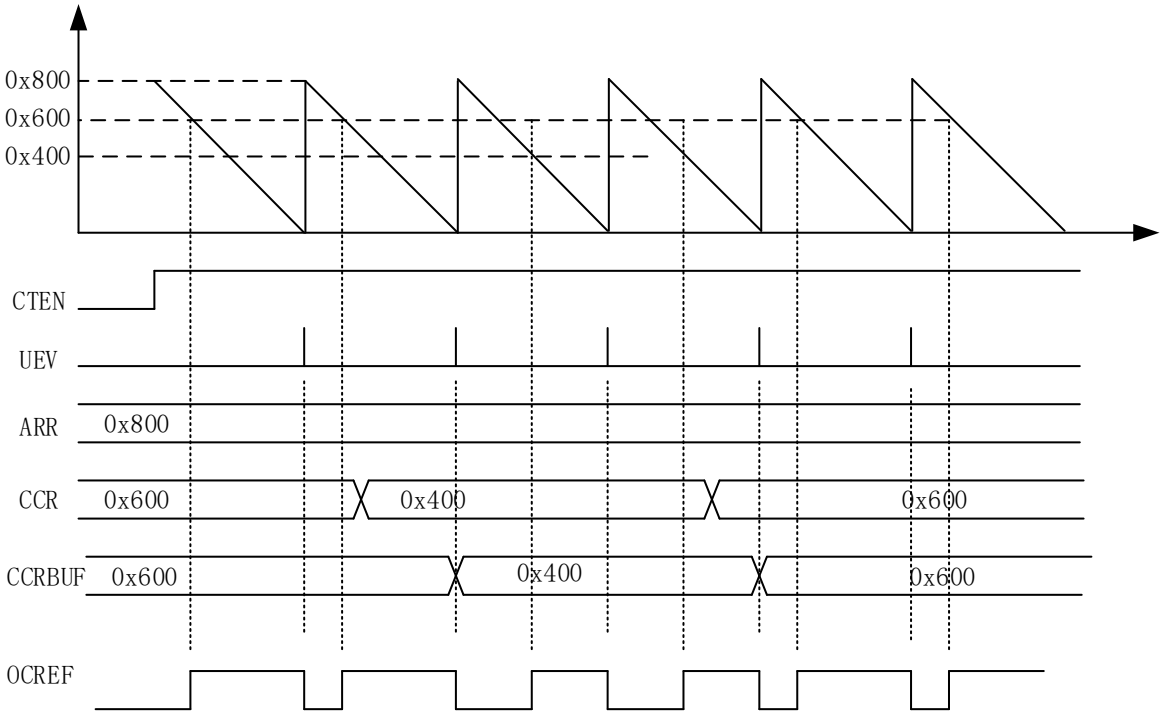


图 15-30 锯齿波模式下计数比较缓存使能

15.2.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA,CCRB 一起控制 OCREFA 的比较输出。

OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

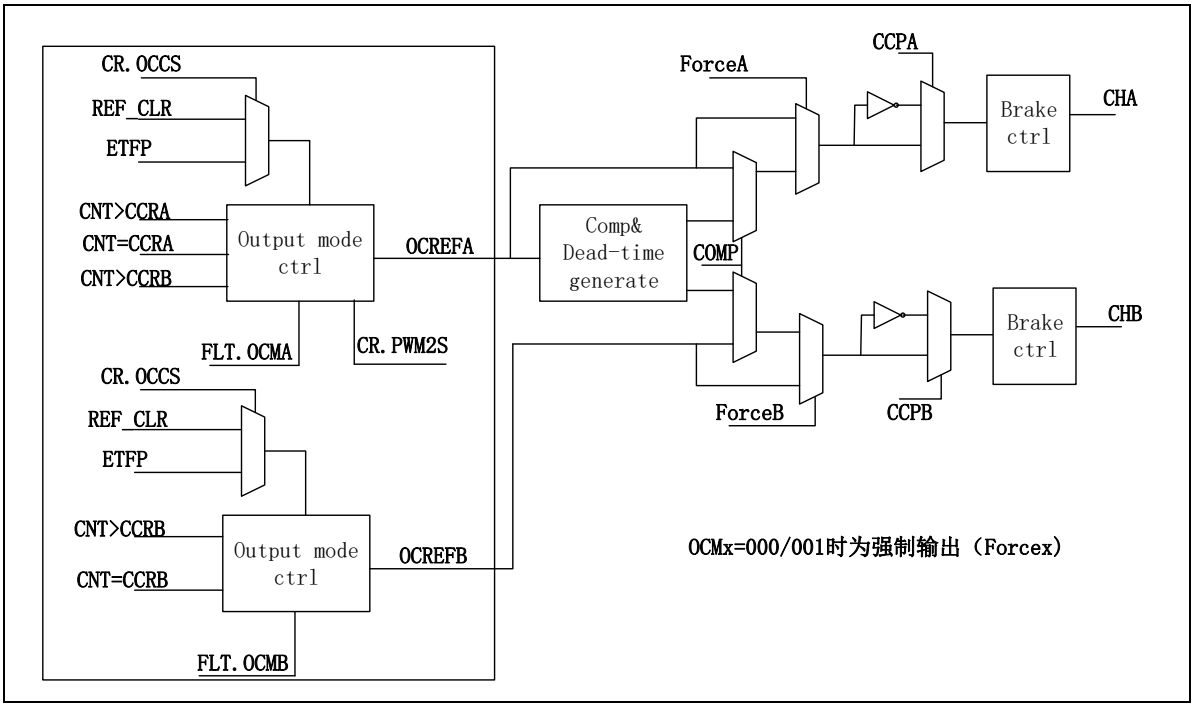


图 15-31 OCREF 输出框图

OCREF 输出使用 OCMx 选择

000：强制为 0

001：强制为 1

010：比较匹配时强制为 0

011：比较匹配时强制为 1

100：比较匹配时翻转

101：比较匹配时输出一个计数周期的高电平

110：PWM 模式 1

单点比较：

上计数时 CNT<CCRxy 输出高，下计数时 CNT>CCRxy 输出为低电平

双点比较：

1）锯齿波上计数 CCRxA<CNT≤CCRxB 输出为低电平

2）锯齿波下计数 CCRxA<CNT≤CCRxB 输出为高电平

3）三角波上计数 CNT<CCRxA 输出高，下计数 CNT>CCRxB 为低电平

111：PWM 模式 2

单点比较：

上计数时 CNT<CCRxy 输出低，下计数时 CNT>CCRxy 输出为高　电平

双点比较：

1）锯齿波上计数 CCRxA≤CNT<CCRxB 输出为高电平

2）锯齿波下计数 CCRxA≤CNT<CCRxB 输出为低电平

3）三角波上计数 CNT<CCRxA 输出低，下计数 CNT>CCRxB 为高电平

注：强制输出有高优先级，当强制输出有效时，互补输出控制无效。

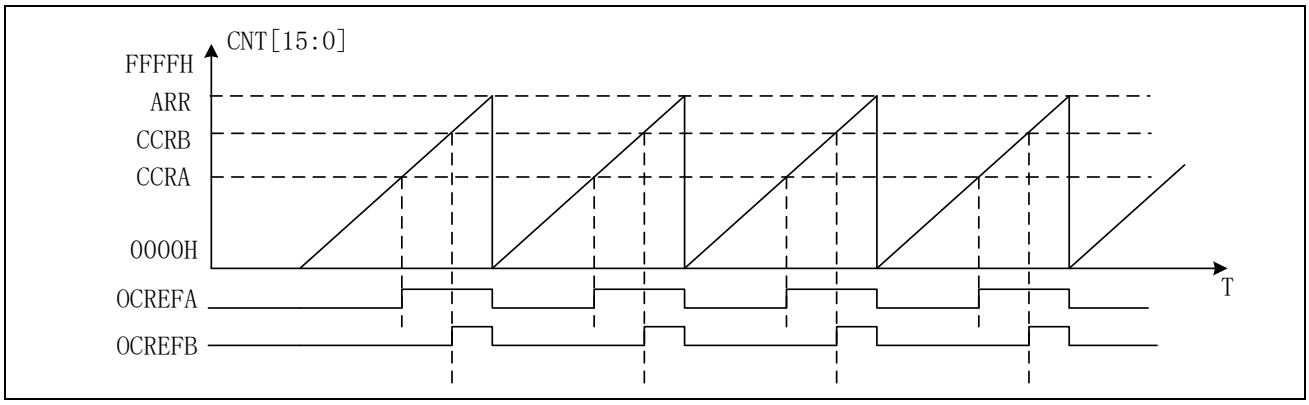


图 15-32　锯齿波计数单点比较OCREF 输出波形（OCMx=111）

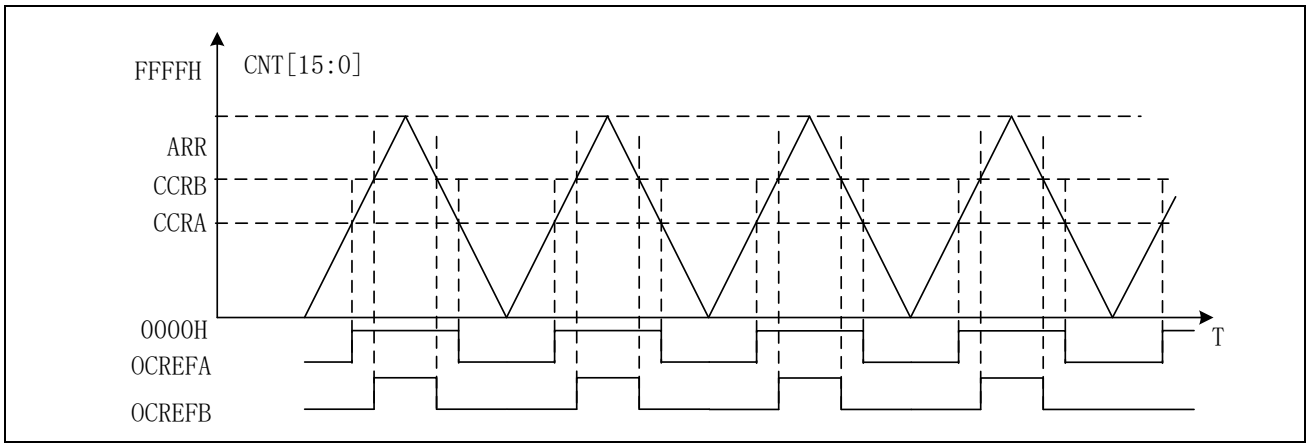


图 15-33　三角波计数单点比较 OCREF 输出波形（OCMx=111）

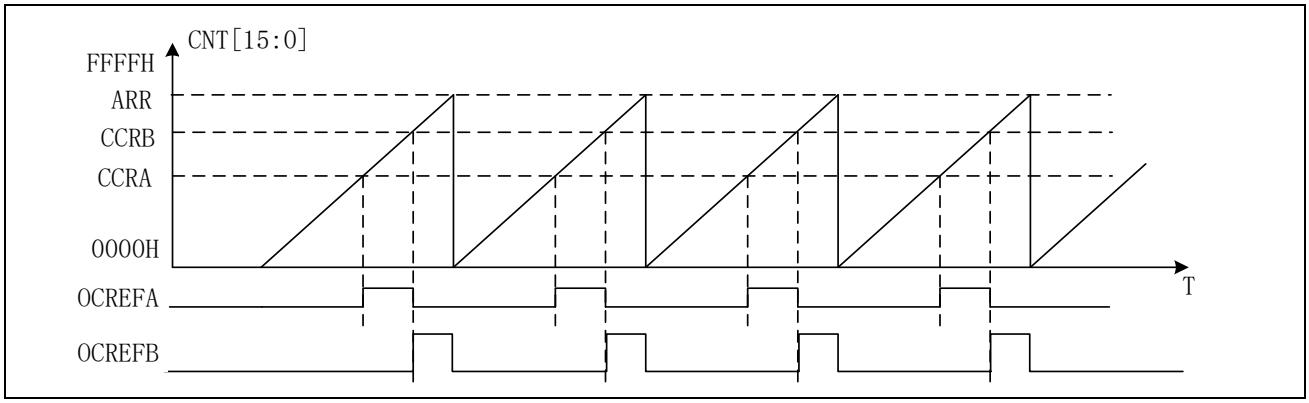


图 15-34 锯齿波计数双点比较 OCREF 输出（OCMx=111）

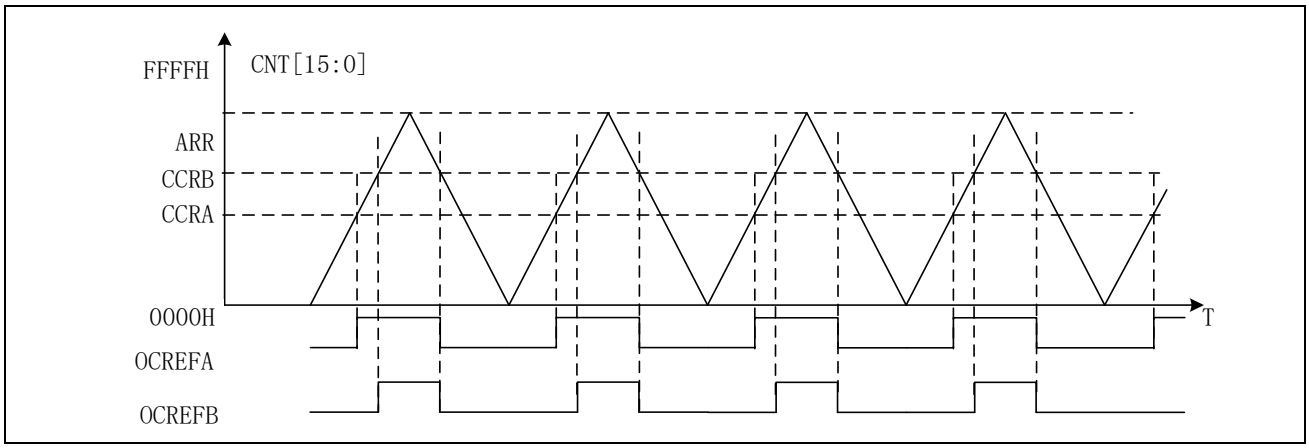


图 15-35 三角波计数双点比较 OCREF 输出（OCMx=111）

15.2.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFB 控制 CHB 的输出。通过 CRCHx.CCPA, CRCHx.CCPB 可以控制 CHA，CHB 输出的反向。

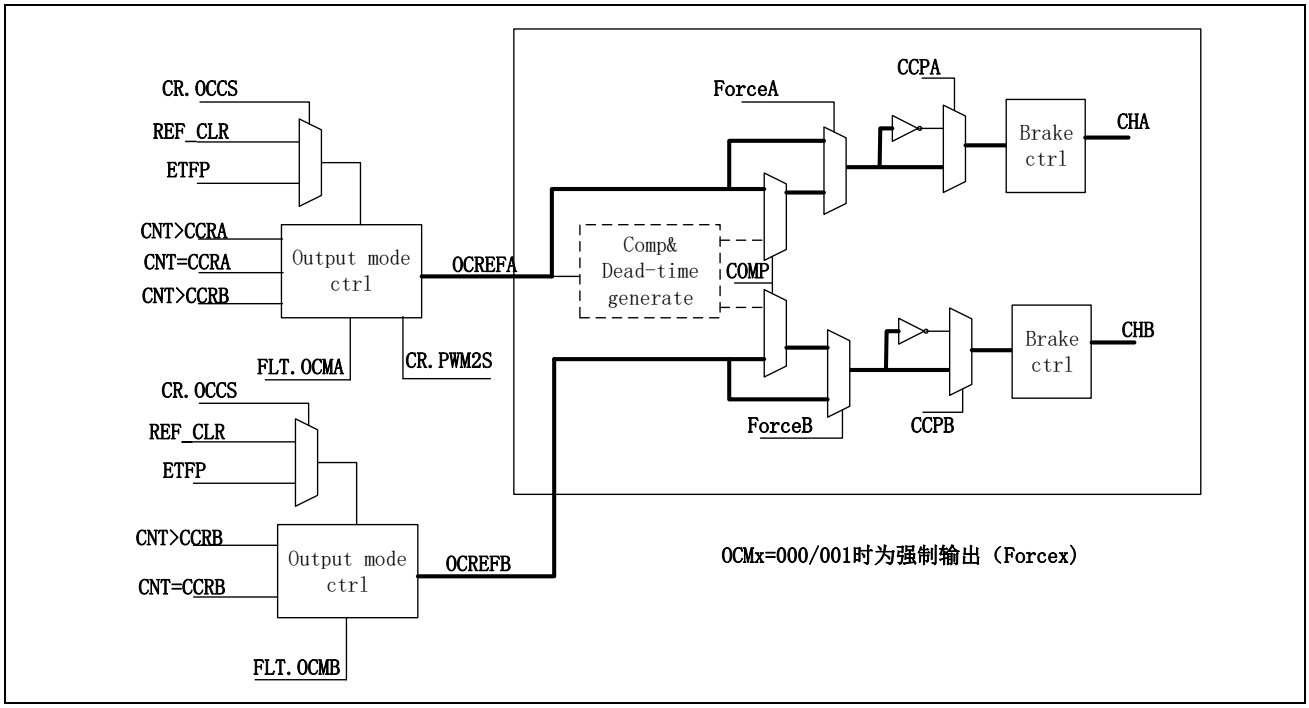


图 15-36 独立 PWM 输出框图

PWM 输出与 OCREF 关系

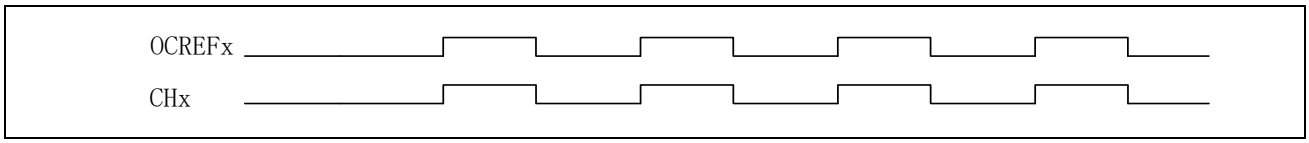


图 15-37 CCPx=0时PWM输出波形

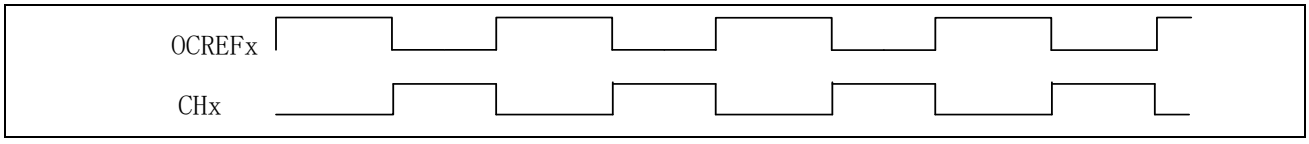


图 15-38 CCPx=1 时 PWM 输出波形

15.2.6.7 互补 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFA 同时控制 CHB 的输出。比较寄存器 CCRxB 可以作为专用比较控制 ADC 触发。

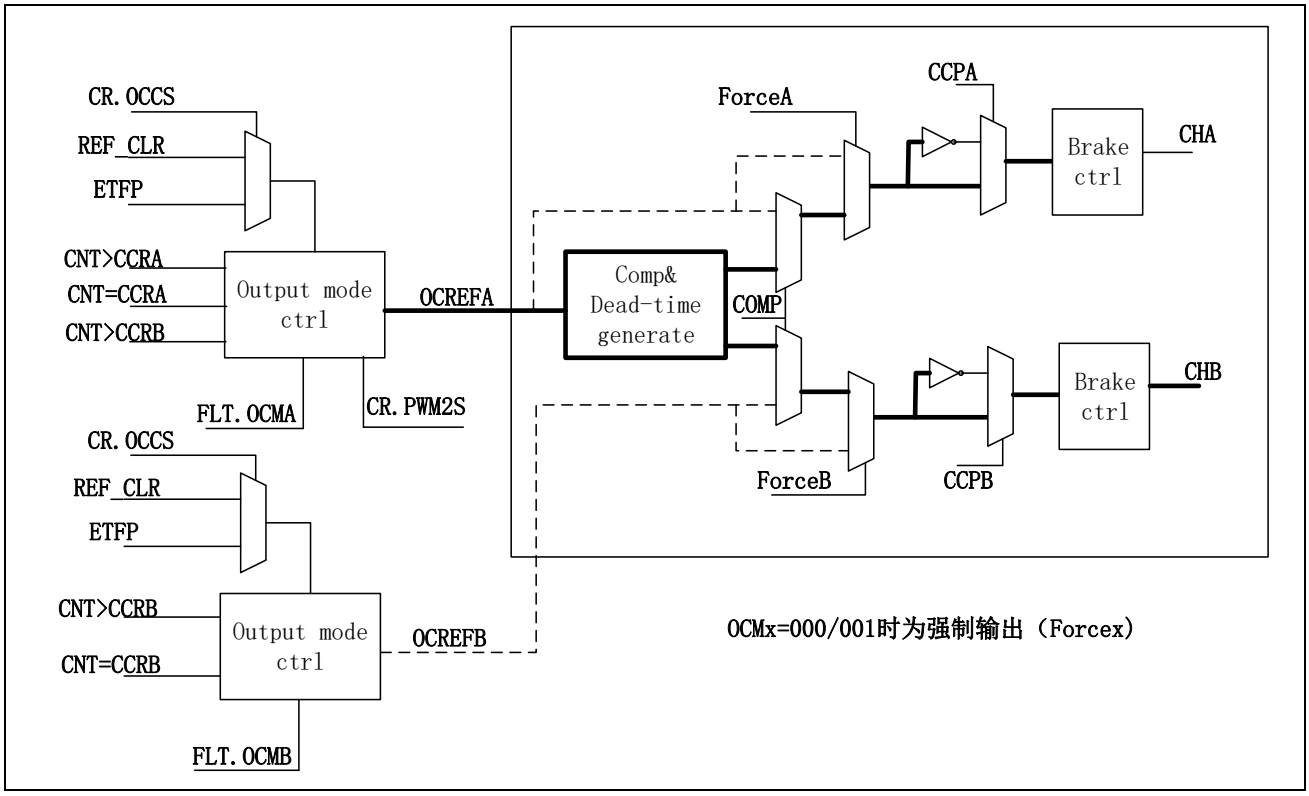


图 15-39 互补 PWM 输出框图

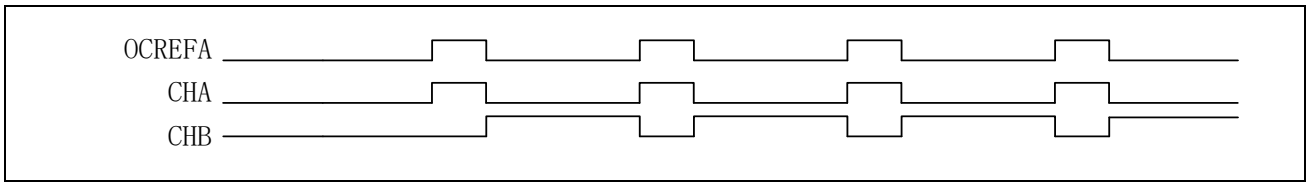


图 15-40 互补PWM输出波形图

15.2.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

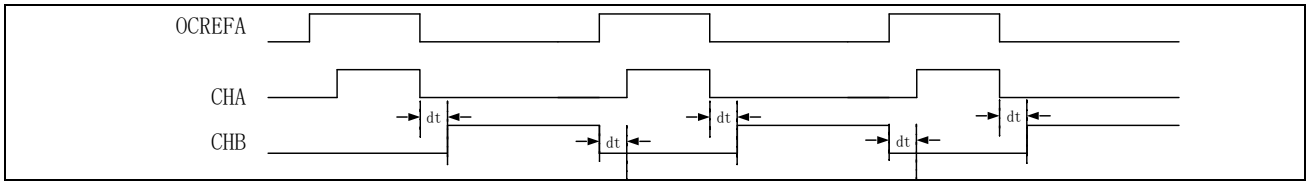
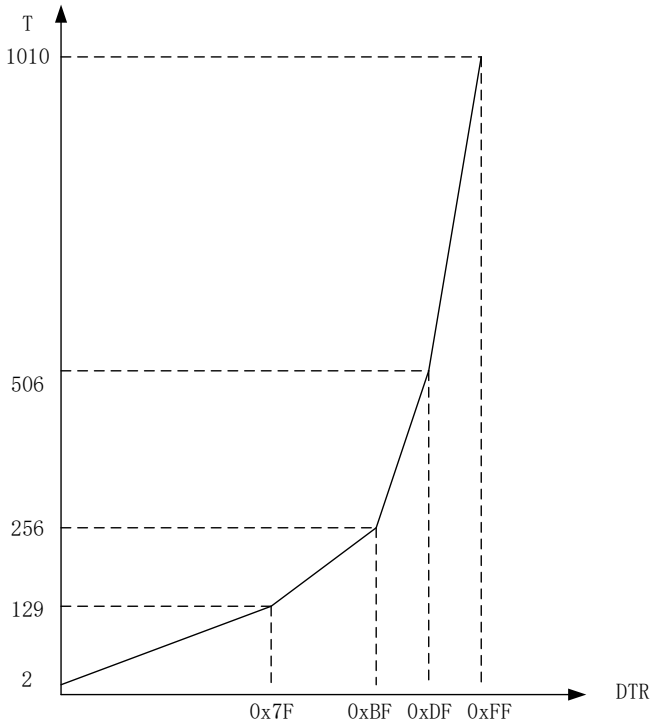


图 15-41 互补PWM输出波形图

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下

DTR[7] =0	T=DTR[6:0]+2	2-129	step=1
DTR[7:6] =10	T={DTR[5:0]+64}*2 +2	130-256	step=2
DTR[7:5] =110	T={DTR[4:0]+32}*8 +2	258-506	step=8
DTR[7:5] =111	T={DTR[4:0]+32}*16 +2	514-1010	step=16



100M clock 死区时间由2.56us调整为10.1us

图 15-42 死区时间

15.2.6.9 单脉冲输出

单脉冲模式 (ONE SHOT) 是上述 PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将

TIMx_M23CR 寄存器中的 ONESHOT 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0,上计数模式计数值不要设置到大于等于 ARR。

只有当比较值与计数器初始值不同时，才能正确产生输出脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时：CNT<CCRxy≦ ARR（特别注意，0<CCRxy），
- 递减计数时：CNT>CCRxy。

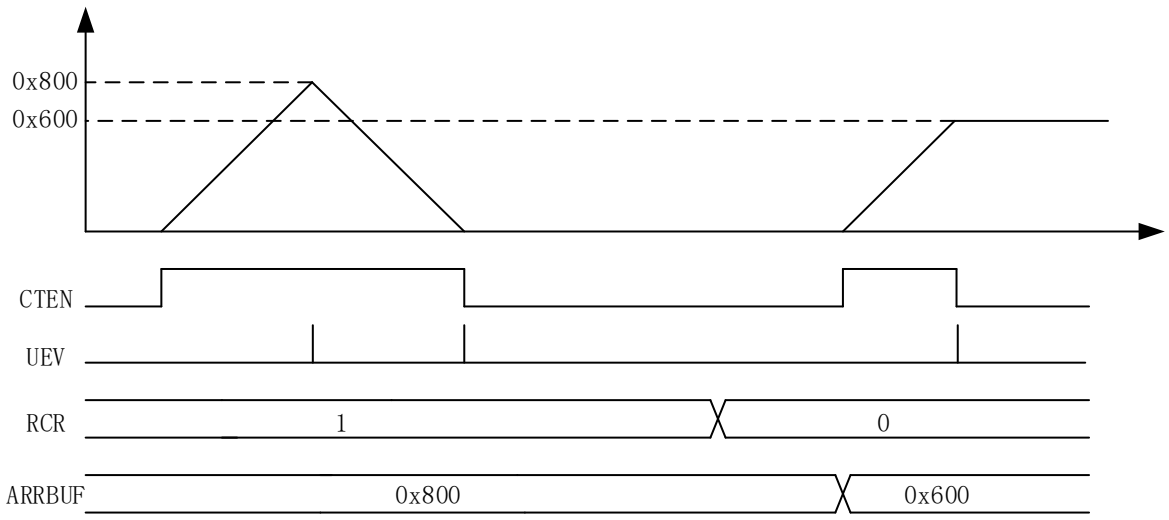


图 15-43 三角波模式单脉冲计数

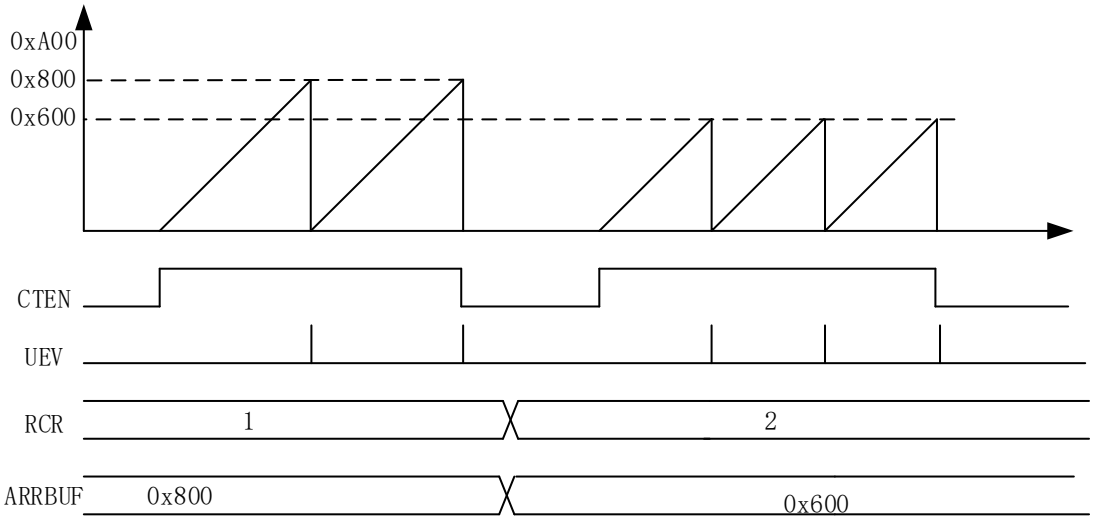


图 15-44 锯齿波上计数单脉冲模式

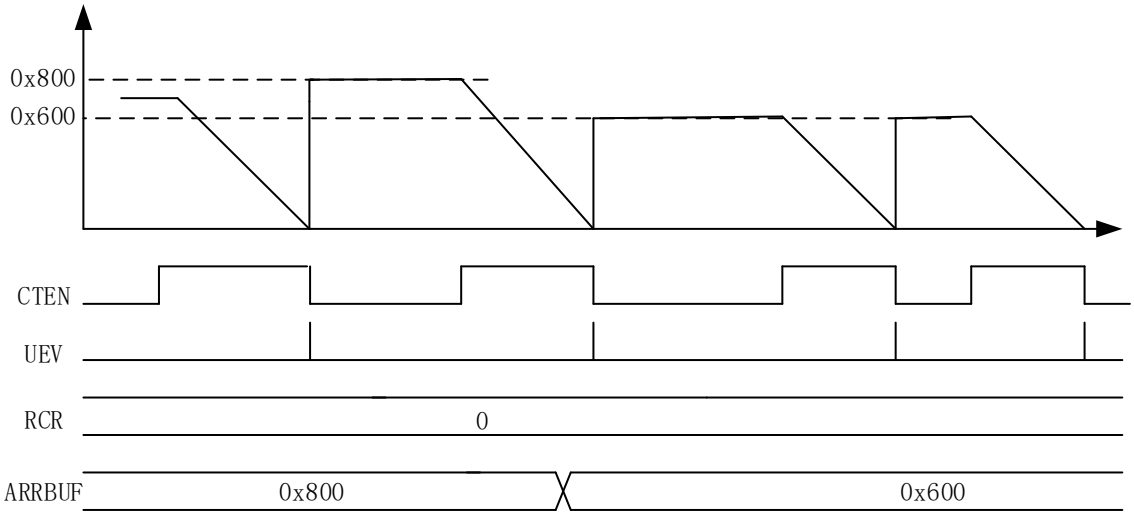


图 15-45 锯齿波下计数单脉冲模式

15.2.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0,1,2;y=A,B)将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 CR.CIS 控制。

CR.CIS=2’b00 时 比较匹配无输出

CR.CIS=2’b01 时 上计数时比较匹配

CR.CIS=2’b10 时 下计数时比较匹配

CR.CIS=2’b11 时 上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制。

CRCHx.CISB=2’b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2’b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2’b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2’b11 时 通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。详见 Timer 触发 ADC 章节。

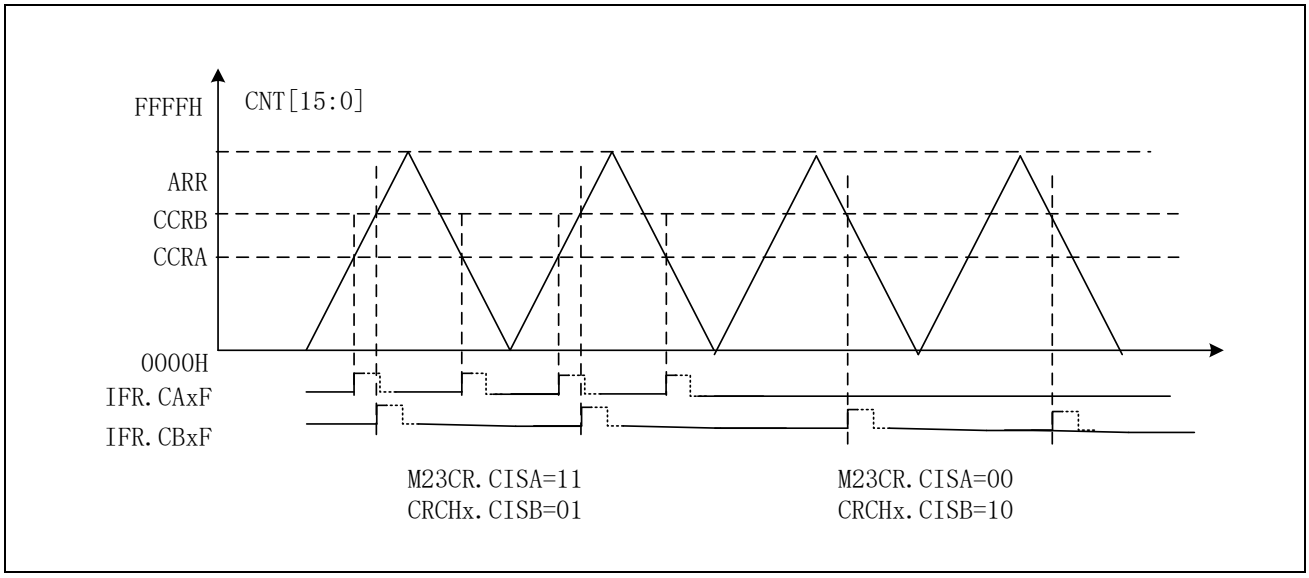


图 15-46 中断示意图

15.2.6.11 捕获输入

CR.MODE=2/3

在三角波计数或者锯齿波计数模式下都可以设置捕获功能，可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。

每个通道的捕获边沿可以单独设置，通过寄存器 CRCHx.CFy/CRy(x=0/1/2;y=A/B)选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

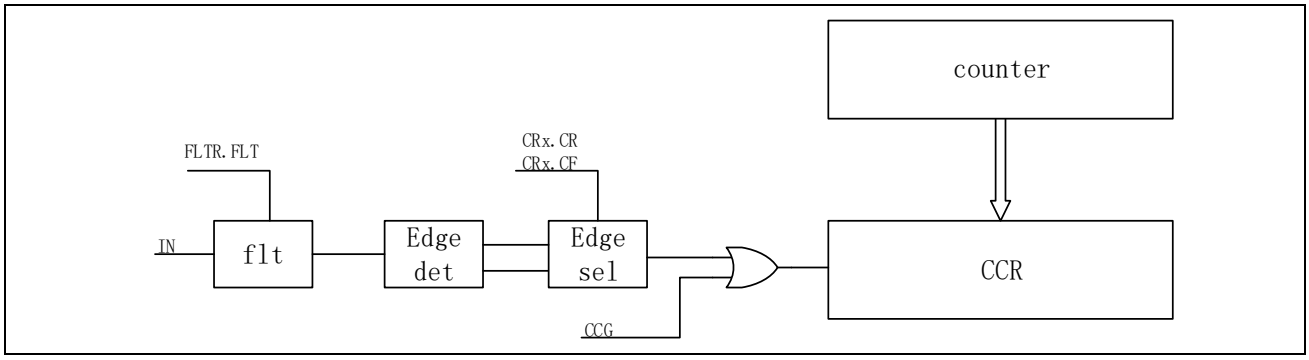


图 15-47 捕获功能框图

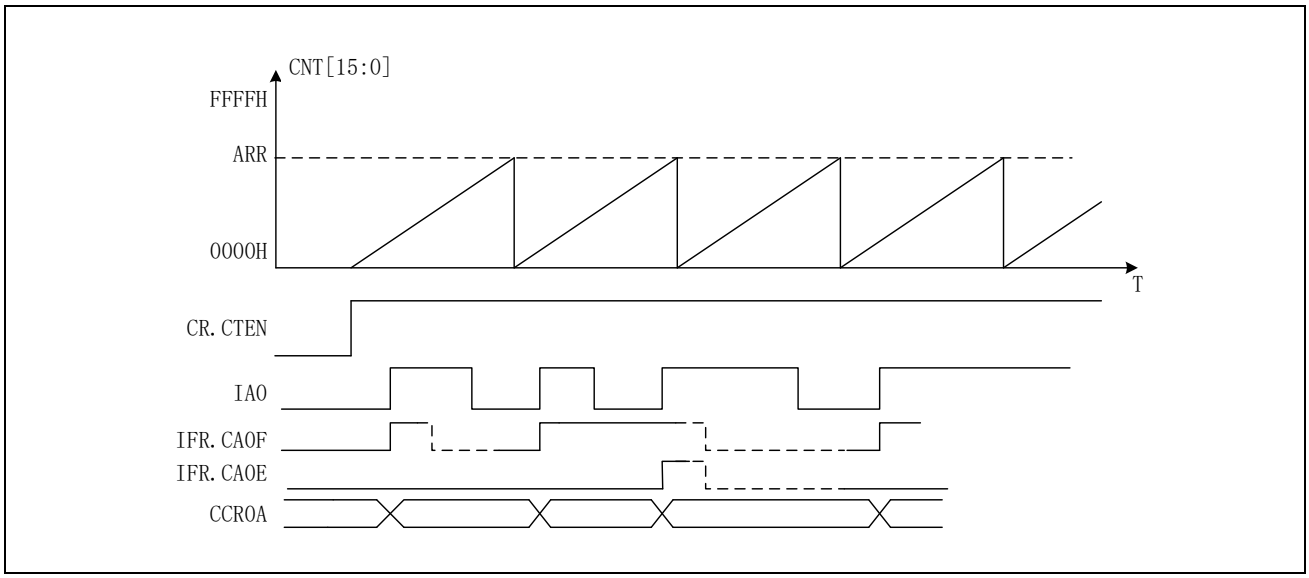


图 15-48 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A，CH1A，CH2A 的异或输入。

IA0S	0	1
Timer0/1/2	CHA0	CHA0 ETR GATE异或输入
Timer3	CHA0	CHA0 CHA CHA2异或输入

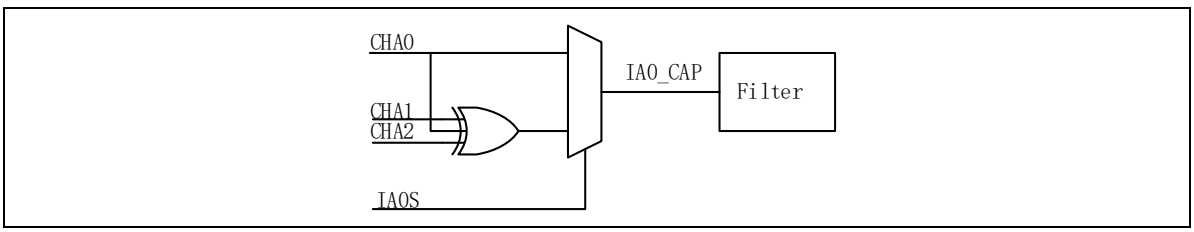


图 15-49 CHA 端口选择

CH0B 的捕获输入通过 MSCR. IB0S 选择 CH0B 输入还是内容 MSCR.TS 选择的信号。

<table>
<tr><td>IB0S</td><td>0</td><td>1</td></tr>
<tr><td>Timer0/1/2/3</td><td>CHB0</td><td>内部触发MSCR.TS选择信号
<table>
<tr><td>MCSCR.TS</td><td>通道B捕获输入</td></tr>
<tr><td>000</td><td>ETR 滤波相位选择输出信号，滤波相位选择可配</td></tr>
<tr><td>001</td><td>内部互联ITR0</td></tr>
<tr><td>010</td><td>内部互联ITR1</td></tr>
<tr><td>011</td><td>内部互联ITR2</td></tr>
<tr><td>100</td><td>内部互联ITR3</td></tr>
<tr><td>101</td><td>CH0A 边沿</td></tr>
<tr><td>110</td><td>CH0A 滤波输出信号，滤波功能可配</td></tr>
<tr><td>111</td><td>CH0B 滤波输出信号，滤波功能可配</td></tr>
</table></td></tr>
</table>

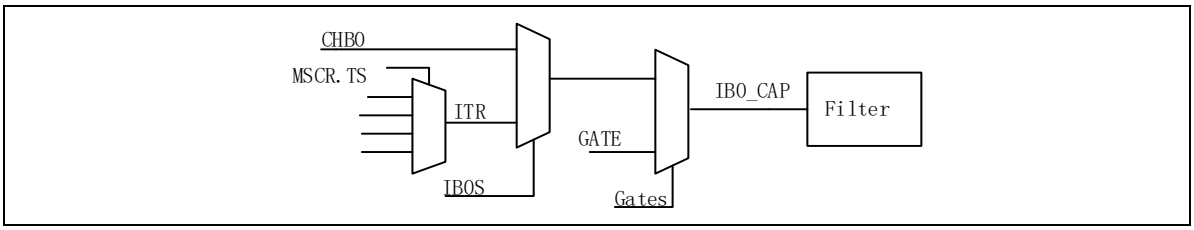


图 15-50 CHB 端口选择

Gates 信号为选择 PWM 互补输出功能生成的信号，当 PWM 互补输出时，比较捕获寄存器 CCR0B 没有使用，自动切换到使用 GATE 作为捕获/比较输出。

15.2.6.12 设置示例

边沿对齐独立 PWM 输出设置

1. 设置模式 CR.MODE=2
2. 设置锯齿波计数方向 CR.DIR
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA,CCRxB
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 CR.CTEN

中心对齐互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA(CCRxB 不需要设置，PWM 输出与这个寄存器无关)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 CR.CTEN

三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置两个比较使能 CR.PWM2S
4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
6. 设置 PWM 比较值 CCRxA,CCRxB,上计数比较点为 CCRxA,下计数比较点为 CCRxB
7. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断
10. 设置输出极性 FLT.CCPAx FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DT
13. 使能输出 DTR.MOE
14. 使能定时器 CR.CTEN

捕获功能设置

设置 CH0B 为上升沿捕获功能

1. 选择计数器计数方式，设置 mode=2
2. 选择 CH0B 为捕获模式 CRCHx.CSB=1
3. 根据需要选择输入滤波 FLT.FLTB0
4. 选择捕获的边沿 CRCHx.CRB=1
5. 设置 ARR 改变周期值
6. 启动定时器 CR.CTEN
7. 清除相关中断标志
8. 清除捕获标志，使能相应中断
9. 查询到捕获标志后，读取 CCR0B 获得捕获值

注：定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

互补 PWM 输出时使用 GATE 作为 PWM 输出功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCR0B 设置 GATE PWM 比较值
3. 设置 FLT.OCMB0=6/7 设置 PWM 输出模式
4. 使能 PWM 输出 DTR.MOE

互补 PWM 输出时使用 GATE 下降沿作为捕获输入功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 GATE 作为捕获输入 CR.CSG
3. 选择 GATE 捕获输入边沿设置 CR.CFG(GATE 作为捕获输入 B 通道滤波设置无效)
4. 启动定时器 CR.CTEN
5. 清除相关中断标志
6. 清除捕获标志，使能相应中断
7. 查询到捕获标志后，读取 CCR0B 获得捕获值

15.2.7 模式 2/3 从模式

定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

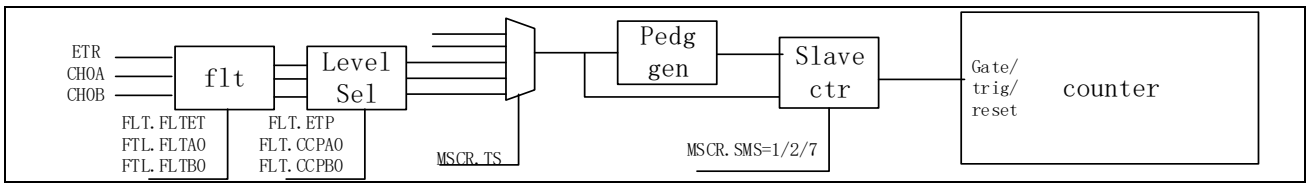


图 15-51 从模式示意图

从模式功能选择通过 MSCR.SMS 选择；

MSCR.SMS	从模式功能选择 001： 复位功能; 010： 触发模式; 111： 门控功能
MSCR.TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP，滤波功能可配置 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP，滤波功能可配置 111：端口CH0B的滤波相位选择后的信号IBFP，滤波功能可配置

15.2.7.1 门控计数

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 CH0A 为低电平时向上计数：

- 配置通道 CH0A 以检测 CH0A 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 FLT.FLTA0=000)。置 FLTR 寄存器中 CCPA0=1 以确定极性(只检测低电平)。
- 置 SMCR 寄存器中 SMS=111，配置定时器为门控模式；置 SMCR 寄存器中 TS=110，选择 CH0A 作为输入源。
- 置 CR 寄存器中 CTEN=1，启动计数器。在门控模式下，如果 CTEN=0，则计数器不能启动，不论触发输入电平如何。只要 CH0A 为低，计数器开始依据内部时钟计数，一旦 CH0A 变高则停止计数。

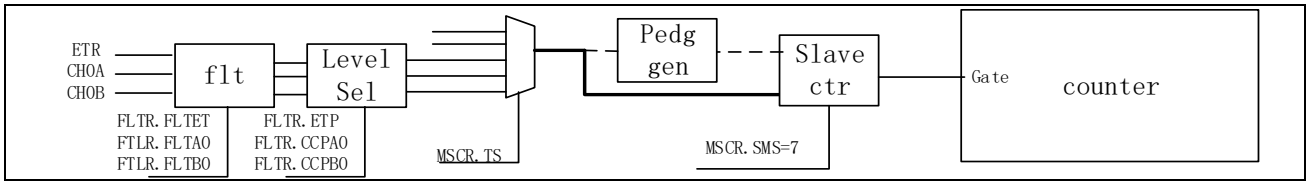


图 15-52 门控功能

15.2.7.2 触发功能

使用外部触发（CH0A、CH0B、ETR）可以使定时器同步启动，使用定时器内部互联信号结合 MSCR.MSM 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 CR.TG 启动软件触发功能。

输入端上选中的事件使能计数器。在下面的例子中，计数器在 CH0B 输入的上升沿开始向上计数：

- 配置通道 CH0B 检测 CH0B 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 FLT.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性(不反向)。
- 置 SMCR 寄存器中 SMS=010，配置定时器为触发模式；置 SMCR 寄存器中 TS=111，选择 CH0B 作为输入源。当 CH0B 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。CH0B 上升沿和计数器启动计数之间的延时，取决于 CH0B 输入端的重同步电路。

  注意：如果使用下降沿触发，先选择触发极性，然后再选择模式，否则会产生误触发。

15.2.7.3 复位计数

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 CR 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(ARR，CCRx)都被更新了。

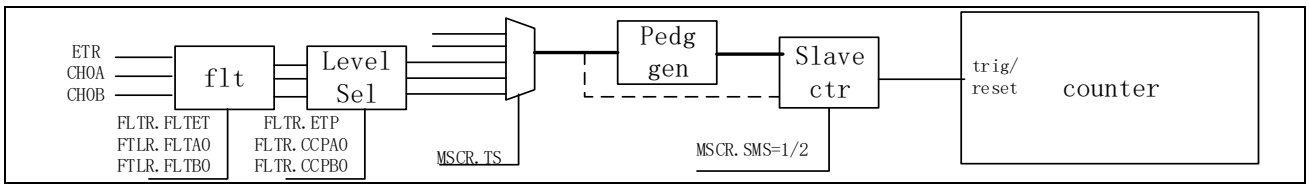


图 15-53 触发和复位功能

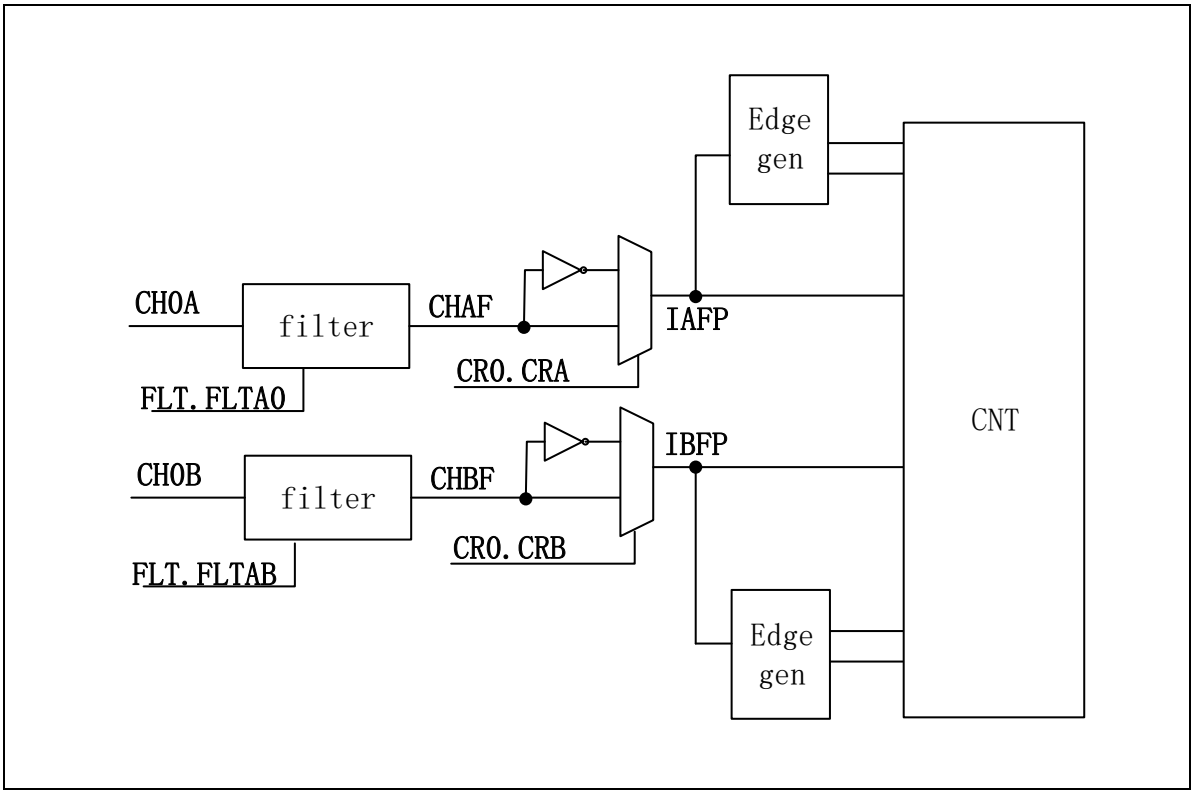
15.2.8 正交编码计数功能

MSCR.SMS=4/5/6，对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP, IBFP 的相位关系进行编码计数。IAFP,IBFP 为端口输入 CH0A，CH0B 的滤波相位选择的信号。模式 1 使用 CH0A 的边沿计数。模式 2 使用 CH0B 的边沿计数。模式 3 使用 CH0A,CH0B 的边沿都计数。

为了保证计数相位的正确性，需要保证 A/B 输入的相位大于一个脉冲宽度的相位差，A/B 输入脉冲宽度需要大于两个脉冲宽度。

			IAFP		IBFP	
	IBFP	IAFP	Rising	falling	Rising	falling
MOD1	High		Down	Up	-	-
	Low		up	Down	-	-
MOD2		High	-	-	up	Down
		Low	-	-	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号，IAFP/IBFP 为端口滤波相位选择后的信号。


Edge
gen
CH0A
filter
CHAF
IAFP
CR0. CRA
FLT. FLTA0
CNT
IBFP
CH0B
filter
CHBF
CR0. CRB
FLT. FLTAB
Edge
gen

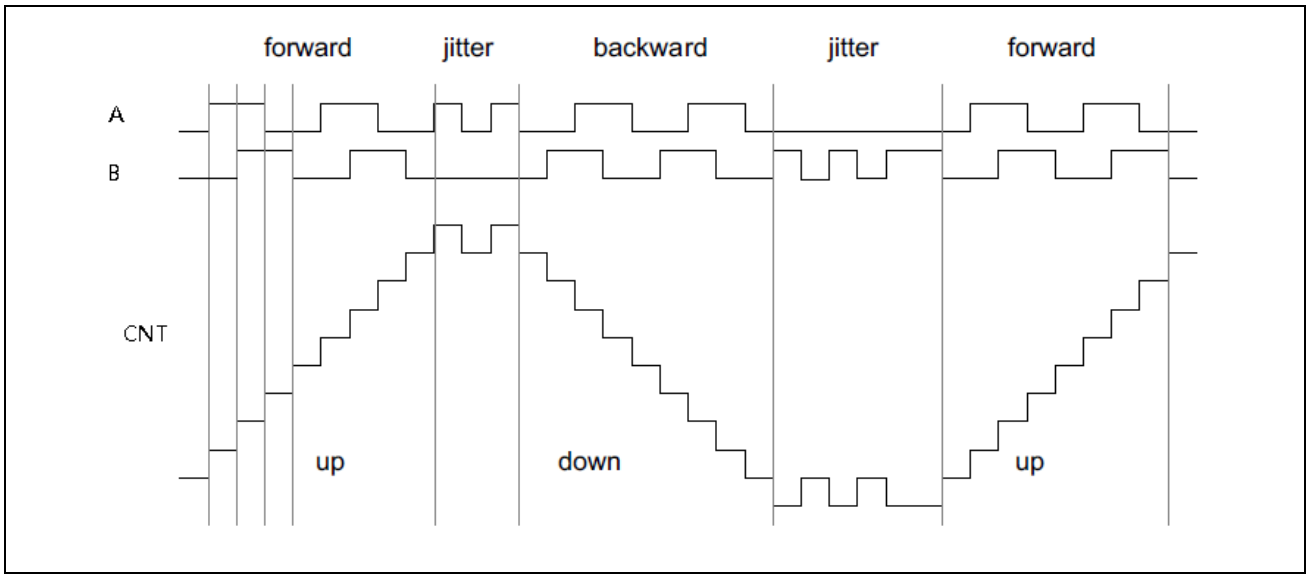

forward
jitter
backward
jitter
forward
A
B
CNT
up
down
up

图 15-54 编码计数

15.2.9 Timer 触发 ADC

CCMR0A,CCMR1A,CCMR2A 比较匹配可以配置触发 ADC，中心对齐 PWM 时只能通过控制寄存器 CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCMR0B,CCMR1B,CCMR2B 比较匹配可以配置触发 ADC，中心对齐 PWM 时，可以通过寄存器 CRx.CISB 分别控制三个匹配触发点（上升，下降，上升下降）。

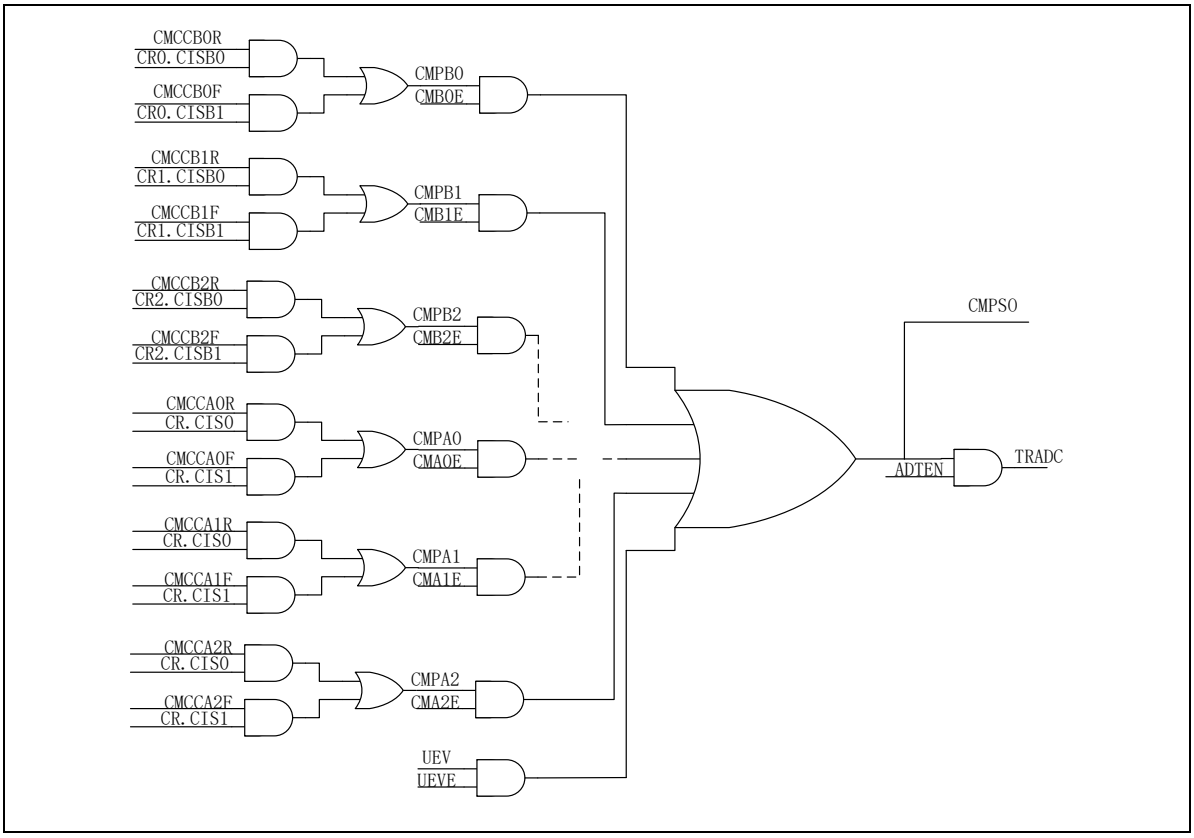


图 15-55 ADC 触发

互补 PWM 输出时 B 比较作为 ADC 触发功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCRxB 设置 ADC 触发比较点
3. 设置 CRCHx.CISB 选择上计数下计数比较匹配（锯齿波不需要选择）
4. 选择 ADC 触发比较的源 ADTR
5. 使能 ADC 触发 ADTR.ADTE

15.2.10 刹车控制

VC 比较输出可以控制刹车功能，外部 BK 端口输入可控制刹车，系统 fail 可以控制刹车功能。通过 CR.BG 可以实现软件刹车功能，控制输出端口到设定的状态。

两通道的输出的 Tim0/1/2 可以选择使用 Tim0 的刹车端口控制 TIM1/2 的刹车功能，也可以使用各自的刹车输入控制各自的刹车功能。当 DTR.BKSEL 设置为 1 时，TIM1/2 共用 TIM0 的刹车输入。

15.2.11 Timer 互联

TRGO 输出信号可以连接到其他定时器的 ITR 信号。连接关系如下：

	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 TIM0 为主模式 MSCR.MSM=1
3. 设置 TIM0 的 CTEN 触发另外两个定时器 MSCR.MMS=1
4. 保持 TIM1/2 的 MSCR.MSM=0
5. 设置 TIM1/2 为触发模式 MSCR.SMS=2
6. 选择 TIM1/2 的触发源为 TIM0 的 TRGO, MSCR.TS=1
7. 最后使能定时器 TIM0，启动定时器 CR.CTEN=1

定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM1 的 UG 触发 TIM0/2）示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 TIM1 为触发主模式 MSCR.MSM=1
3. 设置 TIM1 的 UG 触发另外两个定时器 MSCR.MMS=0

4. 保持 TIM0/2 的 MSCR.MSM=0
5. 设置 TIM0/2 为触发模式 MSCR.SMS=2
6. 选择 TIM0/2 的触发源为 TIM0 的 TRGO, MSCR.TS=2
7. 最后使能定时器 TIM1，启动定时器 CR.UG=1

15.2.12 GATE 输入互联

GATE 输入可以从端口通过 PX_SEL 选择直接输入，也可通过端口功能寄存器 GPIO_TIMGS 选择可以连通到其他模块或端口。

当 TIMx_G=0x0 时，门控 GATE 输入为 PX_SEL 选择的端口输入，当 TIMx_G=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_g	TIM1_g	TIM2_g	TIM3_g
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	-	UART0_RXD	UART0_RXD
010	UART1_RXD	-	UART1_RXD	UART1_RXD
011	VC0_OUT	VC0_OUT	VC0_OUT	-
100	VC1_OUT	VC1_OUT	VC1_OUT	-
101	PA03	PA08	PA10	VC0_OUT
110	PB08	PB03	PB04	PA06
111	PB15	PB13	PB11	PA11

15.2.13 ETR 输入互联

ETR 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO_TIMES 选择可以连通到其他模块或端口。

当 TIMx_E=0x0 时，外部时钟 EXT 输入为 PX_SEL 选择的端口输入，当 TIMx_E=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_e	TIM1_e	TIM2_e	TIM3_e
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	-	UART0_RXD	-	UART0_RXD
010	-	UART1_RXD	-	UART1_RXD
011	VC0_OUT	VC1_OUT	VC0_OUT	VC1_OUT
100	LVD_OUT	LVD_OUT	-	-

101	PA00	PA01	PA04	PA00
110	PA05	PC09	PC04	PA12
111	PA15	PD02	PC08	PA13

15.2.14 CHx 捕获输入互联

Timer0/1/2 的 CHA，Timer3 的 CH0A/CH0B 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO_TIMCPS 选择可以连通到其他模块或端口。

当 TIMx_CHy=0x0 时，捕获输入是 PX_SEL 选择的端口输入，当 TIMx_CHy=0x1~0x7 时，连接其他模块的输入或输出。

	TIM0_CHA	TIM1_CHA	TIM2_CHA	TIM3_CH0A	TIM3_CH0B
000	PX_SEL	PX_SEL	PX_SEL	PX_SEL	PX_SEL
001	UART0_RXD	UART1_RXD	-	-	UART0_RXD
010	PA00	PA00	VC0_OUT	-	UART1_RXD
011	PA02	PA02	PA02	-	-
100	PA05	PA06	PA07	VC0_OUT	VC1_OUT
101	PA15	PB08	PB08	PA08	PA07
110	PB06	PB10	PB09	PB03	PB04
111	PB14	PB13	PC06	PB06	PB13

15.2.15 DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求，A 请求触发源可选择比较捕获 A，外部触发，B 请求触发源可选择比较捕获 B，事件更新。

IDREQ	Interrupt Signal of Peripheral
18	TIM0A
19	TIM0B
20	TIM1A
21	TIM1B
22	TIM2A

23	TIM2B
24	TIM3A
25	TIM3B

TIMxA 的触发源可选择比较捕获 A,ETRIG

TIMxB 的触发源可选择比较捕获 B,UEV

15.2.15.1 设置示例

捕获数据 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考数据捕获流程设置数据捕获
9. 使能 CRCHx.CDy 使能捕获数据触发 DAM 传输。

Note：源地址设置为捕获通道寄存器，源地址固定，目标地址加。

脉冲调整 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考脉冲调整输出设置脉冲调制输出
9. 根据需要选择硬件触发 DMA 的条件（UEV 触发或者比较匹配触发）

Note：目标地址设置为捕获通道寄存器，源地址变化，目标地址固定。

15.3 Timer 寄存器描述

Timer	基地址	描述
Timer0	0x40000C00	Timer0基地址
Timer1	0x40000D00	Timer1基地址
Timer2	0x40000E00	Timer2基地址
Timer3	0X40005800	Timer3基地址

寄存器	偏移地址	描述
TIMx_ARR	0X000	Timer重载寄存器/周期
TIMx_CNT	0X004	Timer 16位模式计数寄存器
TIMx_CNT32	0X008	Timer 32位模式计数寄存器 注：TIMx_CNT32是TIMx_ARR与TIMx_CNT的重映射寄存器。为了方便32位定时器写操作。
TIMx_M0CR	0X00C	Timer模式0控制寄存器（按不同模式进行描述）
TIMx_M1CR	0X00C	Timer模式1控制寄存器（按不同模式进行描述）
TIMx_M2CR	0X00C	Timer模式23控制寄存器（按不同模式进行描述）
TIMx_IFR	0X010	Timer中断标志
TIMx_ICLR	0X014	Timer中断清除寄存器
TIMx_MSCR	0X018	主从模式控制
TIMx_FLTR	0X01c	滤波控制
TIMx_ADTR	0X020	ADC触发控制
TIMx_CRCH0	0x024	比较单元0控制寄存器
TIMx_CRCH1	0X028	比较单元1控制寄存器
TIMx_CRCH2	0x02C	比较单元2控制寄存器
TIMx_DTR	0X030	死区寄存器
TIMx_RCR	0X034	重复计数寄存器
TIMx_ARRDM	0X038	Timer重载寄存器/周期映射地址
TIMx_CCR0A	0X03C	比较0A寄存器
TIMx_CCR0B	0x040	比较0B寄存器
TIMx_CCR1A	0X044	比较1A寄存器
TIMx_CCR1B	0X048	比较1B寄存器
TIMx_CCR2A	0X04C	比较2A寄存器
TIMx_CCR2B	0X050	比较2B寄存器

表 15-1 Timer 寄存器列表

15.4 模式 0 定时器寄存器描述

15.4.1 16 位模式重载寄存器（TIMx_ARR）

偏移地址：0x000

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值寄存器

15.4.2 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器 计数值寄存器

15.4.3 32 位模式计数寄存器（TIMx_CNT32）

偏移地址：0x008

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32[15:0]															
RW															

位	符号	描述
31:0	CNT32	32位定时器 计数值寄存器 注：仅在模式0 32位定时器自由计数模式下有效，其他模式禁止写该寄存器

15.4.4 控制寄存器（TIMx_M0CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8
Resvered		Mode		Resvered	UIE	GATEP	GATE
		RW			RW	RW	RW

7	6	5	4	3	2	1	0
Resvered	PRS			TOGEN	CT	MD	CTEN
	RW			RW	RW	RW	RW

位	符号	描述
31:14	Reserved	保留位
13:12	MODE	工作模式 **00 定时器模式0;** 01 PWC模式 10 锯齿波模式; 11三角波模式
11	Reserved	保留位
10	UIE	中断使能控制，写1后使能中断
9	GATEP	端口 GATE 极性控制 0：端口GATE高电平有效 1：端口GATE低电平有效
8	GATE	定时器门控使能 0：无门控，CTEN=1 时定时器工作； 1：门控使能，端口 GATE 有效并且CTEN=1时定时器才工作；
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000：1； 001：2； 010：4； 011：8； 100：16； 101：32； 110：64； 111：256；
3	TOGEN	模式0下翻转输出使能 1：翻转输出使能 0：翻转输出关闭CHA,CHB输出为低电平
2	CT	计数时钟选择 0： 内部计数时钟TCLK 1： 外部计数时钟ETR;

1	MD	模式选择32定时/16定时模式选择 0： 32位自由计数 1： 16位重载计数
0	CTEN	定时器使能 0： 定时器停止； 1： 定时器使能

15.4.5 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															RO

位	符号	描述
31:1	Res.	保留位
0	UIF	溢出中断

15.4.6 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															R1W0

位	符号	描述
31:1	REV	保留位
0	UIF	溢出中断清除，写0清除

15.4.7 死区时间寄存器（TIMx_DTR）

偏移地址：0x030

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			MOE	Resvered											
			RW												

位	符号	描述
31:13	Reserved	保留位
12	MOE	翻转输出使能 0：翻转输出为输入状态 1：翻转端口为输出状态
11:0	Reserved	保留位

15.5 脉冲宽度测量 PWC 寄存器描述

15.5.1 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	计数值寄存器

15.5.2 控制寄存器（TIMx_M1CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8
Resvered	Oneshot	Mode		Resvered	UIE	Edg2nd	Edg1st
	RW	RW			RW	RW	RW

7	6	5	4	3	2	1	0
Resvered	PRS			Resvered	CT	Resvered	CTEN
	RW				RW		RW

位	符号	描述
31:15	Reserved	保留位
14	Oneshot	单次触发模式选择 1：完成一次脉冲测量自动结束，再次测量需要重新使能CTEN 0：循环测量
13:12	MODE	工作模式 00 定时器模式0; **01 PWC模式** 10 锯齿波模式; 11三角波模式
11	Reserved	保留位
10	UIE	中断使能控制，写1后使能中断 计数到0XFFFF才会溢出并产生溢出标志
9	Edg2nd	脉冲宽度测量结束边沿选择
8	Edg1st	脉冲宽度测量开始边沿选择
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000：1； 001：2； 010：4； 011：8； 100：16； 101：32； 110：64； 111：256；
3	Reserved	保留位
2	CT	计数时钟选择 0： 内部计数时钟TCLK 1： 外部计数时钟ETR;

edg2nd Edg1st	00	01	10	11
测量	上沿-上沿周期	低电平宽度	高电平宽度	下沿-下沿周期

1	Reserved	保留位
0	CTEN	脉冲宽度测量使能 0：禁止； 1：使能

15.5.3 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	Res.	UIF
													RO		RO

位	符号	描述
31:3	Res	保留位
2	CA0F	脉冲宽度测量中断标志
1	Res.	保留
0	UIF	溢出中断标志

15.5.4 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	Res.	UIF
													R1W0		R1W0

位	符号	描述
31:3	Res.	保留位
2	CA0F	脉冲宽度测量中断标志清除，写0清除
1	Res.	保留
0	UIF	溢出中断清除，写0清除

15.5.5 主从模式控制寄存器（TIMx_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resverved			IB0S	IA0S	Resvered			TS			Resvered				
			RW	RW				RW							

位	符号	描述
31:13	Res.	保留位
12	IB0S	CH0B 输入选择 0： CH0B; 1：内部触发TS选择信号； 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0： CH0A; 1： CH0A CH1A CH2A XOR(TIM3) 0： CH0A; 1： CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	Res.	保留位
7:5	TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP 111：端口CH0B的滤波相位选择后的信号IBFP;
4:0	Res.	保留位

15.5.6 输出控制滤波（TIMx_FLTR）

偏移地址：0x01C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET			Resvered											
RW	RW														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvered									FLTB0			Resvered	FLTA0		
									RW				RW		

位	符号	描述
31	ETP	ETR 输入相位选择 0：同相位; 1：反向输入；
30:28	FLTET	ETR滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效
27:7	Resvered	保留位
6:4	FLTB0	CHB 输入滤波控制； 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效
3	Resvered	保留位
2:0	FLTA0	CHA 输入滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效

15.5.7 控制寄存器（TIMx_CR0）

偏移地址：0x024;

复位值：0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:9	Resvered	保留位
8	CIEA	CIEA 脉冲宽度测量完成中断使能 0：禁止 1：使能
7:0	Resvered	保留位

15.5.8 比较捕获寄存 (TIMx_CCR0A)

偏移地址：0x03C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0A															
R															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0A	脉冲宽度测量结果

15.6 模式 2,3 寄存器描述

15.6.1 16 位模式重载寄存器（TIMx_ARR）

偏移地址：0x000(0X038 dummy address)

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	重载寄存器/周期寄存器，具有缓存功能 在计数器未使能或者缓存没有使能时，缓存寄存器可以立刻更新

15.6.2 16 位模式计数寄存器（TIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	重载定时器 计数值寄存器

注：使用 PWM 比较输出时，初始化 CNT 值需要小于 ARR 的值。

15.6.3 控制寄存器（TIMx_M23CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24
Resvered				DIR	BG	UG	TG
				RW/RO	W1	W1	W1

23	22	21	20	19	18	17	16
OCCE	CIS		BIE	TIE	TDE	URS	OCCS
RW	RW		RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
CSG	Oneshot	Mode		UDE	UIE	CFG	CRG
RW	RW	RW		RW	RW	RW	RW

7	6	5	4	3	2	1	0
BUFPEN	PRS			Pwm2s	CT	Comp	CTEN
RW	RW			RW	RW	RW	RW

位	符号	描述
31:28	Reserved	保留位
27	DIR	计数方向，只有在锯齿波模式下可以写。其他模式下只读，写无效 0：上计数 1：下计数 注：从其他模式切换到中心对齐模式DIR自动清0。软件事件更新和从模式外部触发复位模式DIR自动清零。
26	BG	软件刹车，自动清零 写1产生软件刹车； 写0无效
25	UG	软件更新，自动清零 写1产生软件更新； 写0无效 初始化计数器并更新缓存寄存器到相应寄存器（缓存使能），预除频计数器也会被清零。
24	TG	软件触发，自动清零,需要在触发模式SMS=2且mode=2/3下都可以触发。 写1产生软件触发； 写0无效
23	OCCE	OCREF 清除使能

		1：OCREF_CLR信号可以清除OCREF输出 0：OCREF输出不受OCREF_CLR影响
22:21	CIS	中心对齐A比较中断模式(B比较中断单独控制在CRx寄存器中CISB) 00：无中断， 01：上沿中断， 10：下沿中断， 11：上下沿都中断
20	BIE	刹车中断使能 1：中断使能 0：中断禁止
19	TIE	触发中断使能 1：中断使能 0：中断禁止
18	TDE	触发DMA使能 1：中断使能 0：中断禁止
17	URS	更新源 0：上溢出/下溢出/软件更新UG/从模式复位; 1：上溢出/下溢出
16	OCCS	OCREF 清除源选择 0：电压比较器VC输出，VC选择在VCx_OUTCFG寄存器设置 1：ETR端口滤波相位选择后的信号 当OCCE有效时，OC_clr可以清除OCREF的比较输出信号为零，（当OCMx>1时有效），下一个uev事件后继续比较输出
15	CSG	GATE 在PWM互补模式下捕获/比较选择;（只有在PWM互补输出时有效） 使用CCR0B作为GATE的比较或捕获通道 1：捕获； 0：比较
14	Oneshot	单次触发模式选择 1：发生事件更新后定时器停止。 0：循环计数
13:12	MODE	工作模式 00 定时器模式0; 01 PWC模式 **10 锯齿波模式; 11三角波模式**
11	UDE	更新DMA使能 1：使能更新触发DMA 0：禁止更新触发DMA
10	UIE	UIE 更新中断使能 1：使能更新中断 0：禁止更新中断
9	CFG	GATE作为捕获输入时，下沿捕获有效控制（只有在PWM互补输出时有效）

		1：下沿捕获有效 0：下沿捕获无效
8	CRG	GATE作为捕获输入时，上沿捕获有效控制（只有在PWM互补输出时有效） 1：上沿捕获有效 0：上沿捕获无效
7	BUFPEN	重载缓存使能 1：周期缓存使能，写入后下个事件更新后才影响到周期值。 0：周期缓存无效，写入后立刻影响周期值
6:4	PRS	内部时钟分频选择 000：1；　001：2；　010：4；　011：8； 100：16；　101：32；　110：64；　111：256；
3	PWM2S	OCREFA双点比较选择(缺省值为1) 0：双点比较使能，使用CCRA,CCRB比较控制OCREFA输出 1：单点比较使能，只使用CCRA比较控制OCREFA输出 注：OCREFB不受影响，仍然使用CCRB控制OCREFB输出
2	CT	计数时钟选择 0：　内部计数时钟TCLK 1：　外部计数时钟ETR;
1	Comp	PWM互补输出模式选择 0：独立PWM输出 1：互补PWM输出
0	CTEN	定时器使能 0：禁止； 1：使能 可以外部触发使能，在oneshot模式下结束时该位自动清零

15.6.4 中断标志寄存器（TIMx_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1A	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO		RO

位	符号	描述
31:16	Reserved	保留位
15	TIF	触发中断标志
14	BIF	刹车中断标志
13	CB2E	通道CH2B捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志 0：无数据丢失；1：数据丢失
10	CA2E	通道CH2A捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志 0：无数据丢失；1：数据丢失（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志 0：无数据丢失；1：数据丢失
7	CB2F	通道CH2B发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
6	CB1F	通道CH1B发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
5	CB0F	通道CH0B发生捕获/比较匹配标志 0：没有发生 1：发生
4	CA2F	通道CH2A发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
3	CA1F	通道CH1A发生捕获/比较匹配标志 0：没有发生 1：发生（仅TIM3存在）
2	CA0F	通道CH0A发生捕获/比较匹配标志 0：没有发生 1：发生
1	Res.	保留位

0	UIF	事件更新中断标志 0：没有发生　1：发生

15.6.5 中断标志清除寄存器（TIMx_ICLR）

偏移地址：0x014

复位值： 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1A	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF
R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0		R1W0

位	符号	描述
31:16	REV	保留位
15	TIF	触发中断标志清除，写0清除
14	BIF	刹车中断标志清除，写0清除
13	CB2E	通道CH2B捕获数据丢失标志清除，写0清除（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志清除，写0清除（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志清除，写0清除
10	CA2E	通道CH2A捕获数据丢失标志清除，写0清除（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志清除，写0清除（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志清除，写0清除
7	CB2F	通道CH2B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
6	CB1F	通道CH1B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
5	CB0F	通道CH0B捕获/比较匹配标志清除，写0清除
4	CA2F	通道CH2A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
3	CA1F	通道CH1A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
2	CA0F	通道CH0A捕获/比较匹配标志清除，写0清除
1	Res.	保留
0	UIF	事件更新中断清除，写0清除

15.6.6 主从模式控制寄存器（TIMx_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resverved			IB0S	IA0S	SMS			TS			***MSM***	***CCDS***	MMS		
			RW	RW	RW			RW			RW	RW	RW		

位	符号	描述
31:13	Res	保留位
12	IB0S	CH0B 输入选择 0：CH0B; 1：内部触发TS选择信号； 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0： CH0A; 1： CH0A CH1A CH2A XOR(TIM3) 0： CH0A; 1： CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	SMS	从模式功能选择 000： 使用内部时钟; 001： 复位功能; 010： 触发模式; 011： 外部时钟模式 100： 正交编码计数模式1 ; 101： 正交编码计数模式2; 110： 正交编码计数模式3; 111： 门控功能
7:5	TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：端口CH0A的边沿信号; 110：端口CH0A的滤波相位选择后的信号IAFP 111：端口CH0B的滤波相位选择后的信号IBFP;

<table>
<tr><td>4</td><td>MSM</td><td>主从选择
0：无延时
1：延时使能，使主送计数器同时启动.
注：使用触发模式时，从模式设置为0，主模式设置为1，可以使主从计数同时启动</td></tr>
<tr><td>3</td><td>CCDS</td><td>比较模式下 DMA 比较触发选择；
0 ：比较匹配触发DMA;
1： 比较匹配不触发DMA，事件更新代替比较匹配触发DMA</td></tr>
<tr><td>2:0</td><td>MMS</td><td>主模式输出选择，用于内部互联，连接到其他定时器的ITRx
<table>
<tr><td>定时器0/1/2</td><td>定时器3</td></tr>
<tr><td>000：软件更新UG，写CR.UG
001：定时器使能CTEN
010：定时器事件更新UEV;
011：比较匹配选择输出CMPSO;
100：定时器比较参数输出OCREF0A
101：定时器比较参数输出OCREF0B
110：定时器比较参数输出OCREF0B
111：定时器比较参数输出OCREF0B</td><td>000：软件更新UG，写CR.UG
001：定时器使能CTEN
010：定时器事件更新UEV;
011：比较匹配选择输出CMPSO;
100：定时器比较参数输出OCREF0A
101：定时器比较参数输出OCREF1A
110：定时器比较参数输出OCREF2A
111：定时器比较参数输出OCREF0B</td></tr>
</table></td></tr>
</table>

15.6.7 输出控制/输入滤波（TIMx_FLTR）

偏移地址：0x01C

复位值：0x00000000

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16
ETP	FLTET	BKP	FLTBK	CCPB2	OCMB2	CCPA2	OCMA2
-	-	-	-	-	FLTB2	-	FLTA2
RW	RW	RW	RW	RW	RW	RW	RW

15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
CCPB1	OCMB1	CCPA1	OCMA1	CCPB0	OCMB0	CCPA0	OCMA0
-	FLTB1	-	FLTA1	CCPB0	FLTB0	CCPA0	FLTA0
RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31	ETP	ETR 输入相位选择 0：同相位; 1：反向输入；
30:28	FLTET	ETR滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效
27	BKP	刹车BK输入相位选择 0：同相位; 1：反向输入；
26:24	FLTBK	刹车输入滤波控制 滤波设置 0xx：滤波无效 100：pclk 3个连续有效; 101：pclk/4 3个连续有效 110：pclk/16 3个连续有效; 111：pclk/64 3个连续有效 注：为了保证OCMB0的PWM输出设置，GATE 在PWM互补模式下作为捕获输入，滤波设置无效。
23	CCPB2	比较功能：CH2B通道比较输出相位控制 0：正常输出; 1：反向输出

22:20	OCMB2 FLTB2	比较功能：CH2B通道比较控制;参考OCMB0 捕获功能：CH2B输入通道滤波设置，参考FLTBK
19	CCPA2	比较功能：CH2A通道比较输出相位控制 0：正常输出; 1：反向输出
18:16	OCMA2 FLTA2	比较功能：CH2A通道比较控制;参考OCMB0 捕获功能：CH2A输入通道滤波设置，参考FLTBK
15	CCPB1	比较功能：CH1B通道比较输出相位控制 0：正常输出; 1：反向输出
14：12	OCMB1 FLTB1	比较功能：CH1B通道比较控制;参考OCMB0 捕获功能：CH1B输入通道滤波设置，参考FLTBK
11	CCPA1	比较功能：CH1A通道比较输出相位控制 0：正常输出; 1：反向输出
10:8	OCMA1 FLTA1	比较功能：CH1A通道比较控制;参考OCMB0 捕获功能：CH1A输入通道滤波设置，参考FLTBK
7	CCPB0	比较功能：输出比较模式 CCPBx比较输出CHBx端口极性控制 0：正常输出; 1：反向输出 编码计数与从模式门控功能：输入相位控制 CCPB0从模式门控，复位，外部触发，外部时钟使用CH0B端口输入极性控制 0：正常输入; 1：反向输入
6:4	OCMB0 FLTB0	比较功能：CH0B通道比较控制 000：强制为0 001：强制为1 010：比较匹配时强制为0 011：比较匹配时强制为1 100：比较匹配时翻转 101：比较匹配时输出一个计数周期的高电平 110：PWM 模式1 单点比较： 上计数时CNT<CCRxy输出高，下计数时CNT>CCRxy输出为低电平 双点比较： 1）锯齿波上计数 CCRxA<CNT≤CCRxB输出为低电平 2）锯齿波下计数 CCRxA<CNT≤CCRxB输出为高电平 3）三角波上计数CNT<CCRxA输出高，下计数CNT>CCRxB为低电平 111 PWM 模式2 单点比较： 上计数时CNT<CCRxy输出低，下计数时CNT>CCRxy输出为高电平

		双点比较： 1）锯齿波上计数 CCRxA≤CNT<CCRxB输出为高电平 2）锯齿波下计数 CCRxA≤CNT<CCRxB输出为低电平 3）三角波上计数CNT<CCRxA输出低，下计数CNT>CCRxB为高电平 捕获功能：CH0B输入通道滤波设置，参考FLTBK
3	CCPA0	比较功能：CCPAx比较输出CHAx端口极性控制 0：正常输出; 1：反向输出 编码计数与从模式门控功能：输入相位控制 CCPA0从模式门控，复位，外部触发，外部时钟使用CH0A端口输入极性控制 0：正常输入; 1：反向输入
2:0	OCMA0 FLTA0	比较功能：A通道比较控制；参考OCMB0 捕获功能：CH0A输入通道滤波设置，参考FLTBK

15.6.8 ADC 触发控制寄存器（TIMx_ADTR）

偏移地址：0x020

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADTE	CMB2E	CMB1E	CMB0E	CMA2E	CMA1E	CMA0E	UEVE
								RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:8	Resvered	保留位
7	ADTE	使能ADC触发全局控制 1：使能 0：禁止
6	CMB2E	通道2B比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
5	CMB1E	通道1B比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
4	CMB0E	通道0B比较匹配触发ADC使能 1：使能 0：禁止
3	CMA2E	通道2A比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
2	CMA1E	通道1A比较匹配触发ADC使能，只有TIM3存在 1：使能 0：禁止
1	CMA0E	通道0A比较匹配触发ADC使能 1：使能 0：禁止
0	UEVE	事件更新触发ADC使能 1：使能 0：禁止

15.6.9 通道 0 控制寄存器（TIMx_CRCH0）

偏移地址：0x024

复位值： 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa	
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配；01上升匹配；10 下降匹配；11双匹配
11	CDEB	B捕获比较触发DMA使能 0：禁止 1：使能
10	CDEA	A捕获比较触发DMA使能 0：禁止 1：使能
9	CIEB	B捕获比较触发中断使能 0：禁止 1：使能
8	CIEA	A捕获比较触发中断使能 0：禁止 1：使能
7	BUFEB	比较功能：B比较缓存使能控制 0：禁止 1：使能
6	BUFEA	比较功能：A比较缓存使能控制 0：禁止 1：使能

5	CSB	B通道捕获/比较功能选择 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择 0：比较模式 1：捕获模式
3	CFB	B通道下降沿捕获使能 0：禁止 1：使能
2	CRB	B通道上升沿捕获使能 0：禁止 1：使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平;
1	CFA	A通道下降沿捕获使能 0：禁止 1：使能
0	CRA	A通道上升沿捕获使能 0：禁止 1：使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平;

15.6.10 通道 1/2 控制寄存器（TIM3_CRCH1/2）（仅 TIM3 存在）

偏移地址：

TIM3_CRCH1：0x028；

TIM3_CRCH2：0x02C

复位值： 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA
CCGB	CCGA	CISB		CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa	
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发，硬件自动清零。 在比较模式下只产生中断； 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配；01上升匹配；10 下降匹配；11双匹配
11	CDEB	B捕获比较触发DMA使能 0：禁止 1：使能
10	CDEA	A捕获比较触发DMA使能 0：禁止 1：使能
9	CIEB	B捕获比较触发中断使能 0：禁止 1：使能
8	CIEA	A捕获比较触发中断使能 0：禁止 1：使能
7	BUFEB	比较功能：B比较缓存使能控制 0：禁止 1：使能

6	BUFEA	比较功能：A比较缓存使能控制 0：禁止 1：使能
5	CSB	B通道捕获/比较功能选择 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择 0：比较模式 1：捕获模式
3	CFB	B通道下降沿捕获使能 0：禁止 1：使能
2	CRB	B通道上升沿捕获使能 0：禁止 1：使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00：高阻输出 01：保持之前输出l 10：强制输出低电平 11：强制输出高电平;
1	CFA	A通道下降沿捕获使能 0：禁止 1：使能
0	CRA	A通道上升沿捕获使能 0：禁止 1：使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00：高阻输出 01：保持之前输出l 10：强制输出低电平 11：强制输出高电平;

15.6.11 死区时间寄存器（TIMx_DTR）

偏移地址：0x030

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VCE	Safeen	MOE	AOE	BKE	DTEN	Bksel	DTR							
	RW	RW	RW	RW	RW	RW	RW	RW							

位	符号	描述
31:15	Reserved	保留位
14	VCE	VC Brake enable 0：禁止 1：使能
13	Safeen	Safety 刹车使能（osc fail,brown down,lockup） 0：禁止 1：使能
12	MOE	PWM输出使能 0：禁止 1：使能
11	AOE	PWM输出自动使能 0：禁止 1：使能
10	BKE	刹车使能 0：禁止 1：使能
9	DTEN	死区控制使能 0：禁止 1：使能
8	bksel	刹车选择 0：使用自己的刹车控制； 1：TIM1/2使用 TIM0的的刹车控制 注：TIM0/TIM3选择无效。
7:0	DTR	死区时间寄存器 DTR[7] =0 T=DTR[6:0]+2 2-129 step=1 DTR[7:6] =10 T={DTR[5:0]+64}*2 +2 130-256 step=2

		DTR[7:5] =110	T={DTR[4:0]+32}*8 +2 258-506 step=8
		DTR[7:5] =111	T={DTR[4:0]+32}*16 +2 514-1010 step=16

15.6.12 重复周期设置值（TIMx_RCR）

偏移地址：0x034

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:8	Reserved	保留位
7:0	RCR	重复周期计数值 设置RCR+1个周期个上溢出下溢出后产生事件更新，计数器上溢出或下溢出时内部RCR_CNT减1，当计数到零后RCR_CNT重载RCR的值，并且产生事件更新UEV信号

15.6.13 通道 0 比较捕获寄存器 (TIMx_CCR0A/B)

偏移地址：

TIMx_CCR0A ： 0x03C；

TIMx_CCR0B ： 0x040

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0y															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0y	比较捕获寄存器，比较具有缓存功能(y=A,B)

15.6.14 通道 1/2 比较捕获寄存器 (TIM3_CCR1/2 A/B) （仅 TIM3 存在）

偏移地址：

TIM3_CCR1A：0x044

TIM3_CCR1B：0x048

TIM3_CCR2A：0x04C

TIM3_CCR2B：0x050

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CCRxy	比较捕获寄存器，比较具有缓存功能(x=1,2；y=A,B)

16 可编程计数阵列（PCA）

16.1 PCA 简介

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程，以提供输入捕捉，输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

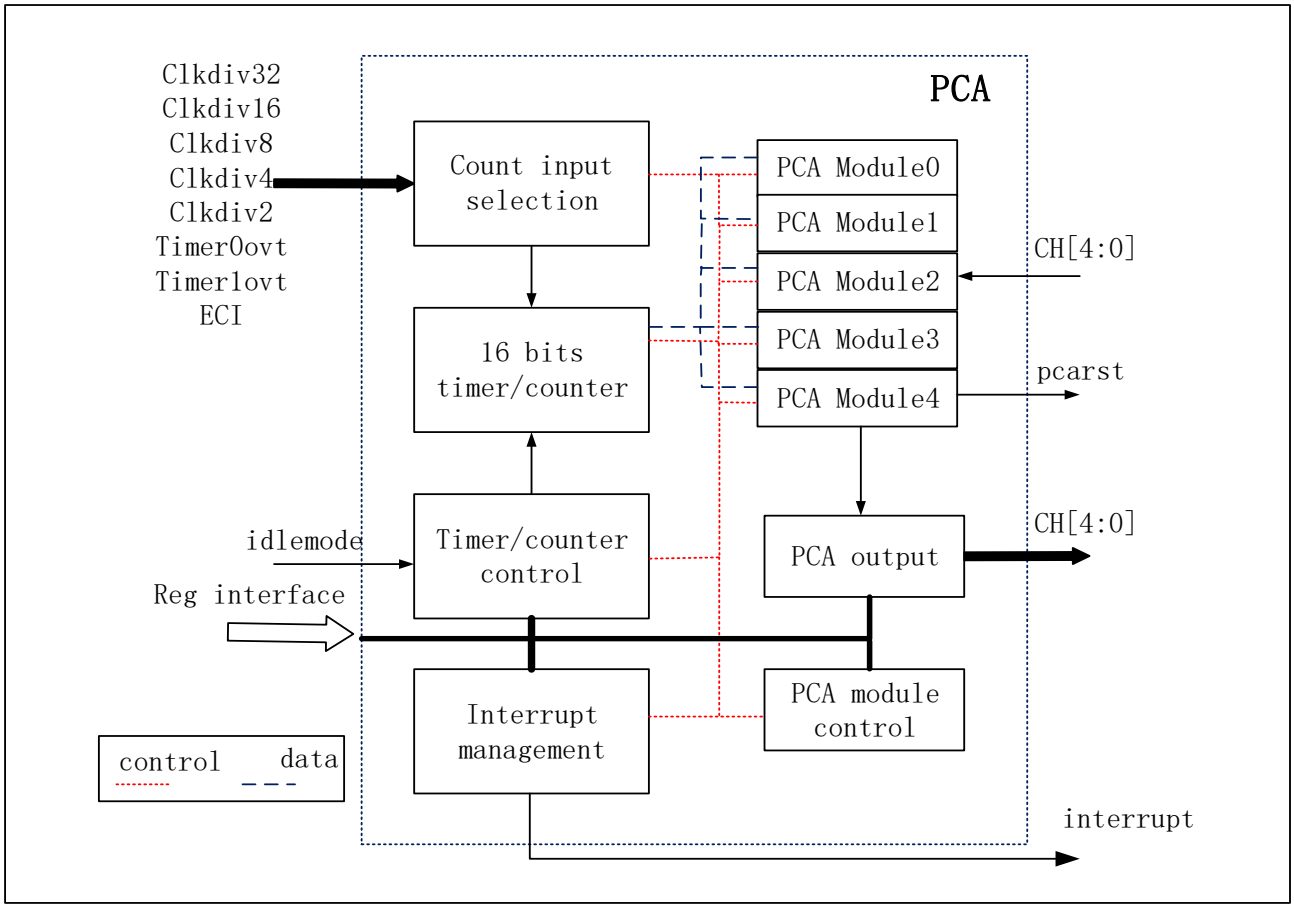


图 16-1 PCA 整体框图

16.2 PCA 功能描述

五个模块都可被配置为独立工作，有三种工作方式：边沿触发捕捉、输出比较、8/16 位脉宽调制。每个模块在系统控制器中都有属于自己的功能寄存器，这些寄存器用于配置模块的工作方式和与模块交换数据。

每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx)，以及 1 个 16 位比较器，和各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 8 位 PWM 模式下，寄存器(CCAPxL)用来控制输出波形的占空比，CCAPxH 为 8 位比较缓存。16 位 PWM 模式下 CARR 用于控制 PWM 输出的周期，CCAPx 寄存器控制占空比。

每个模块都可以独立编程的操作在任何以下模式：

- 16 位捕获模式的上升沿，下降沿或任意沿触发。
- 比较模式：16 位高速输出，16 位看门狗定时器（模块 4）或 16/8 位脉冲宽度调制。
- 关闭状态。

比较/捕获模块模式寄存器（CCAPMx）确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CCON.CR 位即可控制 PCA 定时/计数器的运行。在一个比较/捕获模块捕获，软件定时器，高速输出，设置模块的比较/捕获标志（CCON.CCFx），并产生 PCA 中断请求，如果相应的使能位在 CCAPMx 寄存器设置。CPU 可以在任何时候读写 CCAPx 寄存器。

16.2.1 PCA 定时/计数器

CNT 的这组特殊功能寄存器可用作为一个 16 位定时器/计数器。这是一个 16 位向上计数的计数器。如果 CMOD.CFIE 位被置“1”时，当 CNT 溢出时硬件自动设置 PCA 溢出标志（CCON.CF）并产生 PCA 中断请求。CMOD.CPS[2：0]三位选择八个信号输入到定时器/计数器。

- 系统时钟 PCLK 的 32 分频。
- 系统时钟 PCLK 的 16 分频。
- 系统时钟 PCLK 的 8 分频。
- 系统时钟 PCLK 的 4 分频。
- 系统时钟 PCLK 的 2 分频。
- 定时器 0 的溢出。每次定时器 0 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- 定时器 1 的溢出。每次定时器 1 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- ECI。CPU 每过 4 个 PCLK 时钟周期就对 PCA ECI 进行采样，当每次采样结果从高变低时，CL 自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8,以满足采样需求。

设置运行控制器（CCON.CR）启动 PCA 定时/计数器。当 CMOD.CIDL 置“1”后，PCA 定时器/计数器可以继续运行在空闲模式下。CPU 可以随时读取 CNT 的数值，但当计数启动后（CCON.CR=1）时，为了防止计数错误，CNT 是禁止写入的。

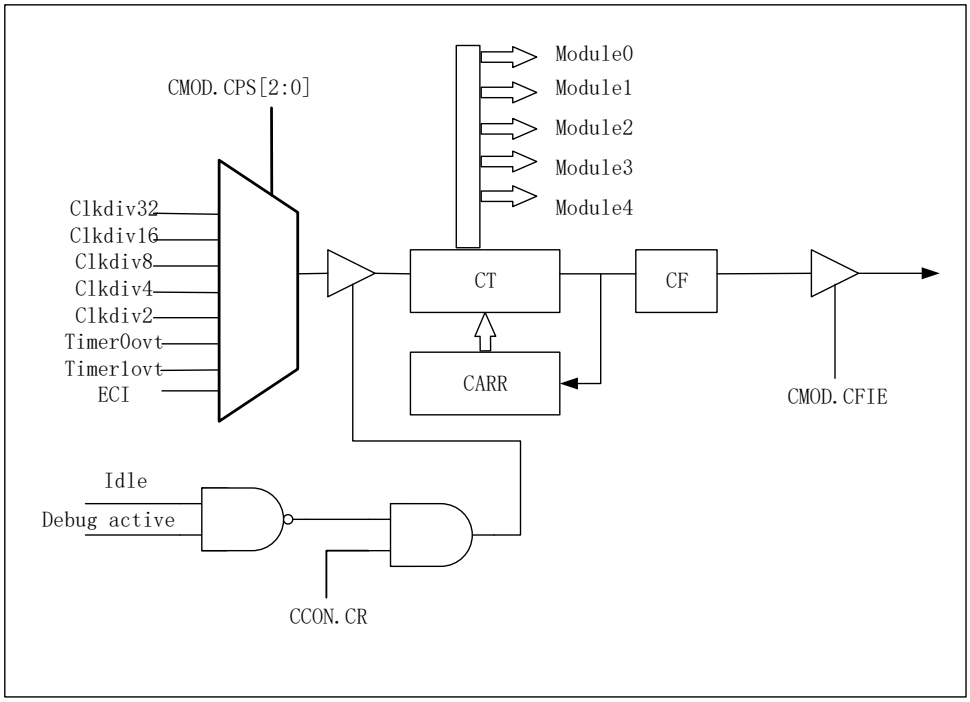


图 16-2 PCA 计数器框图

16.2.1.1 16 位自由计数模式

计数器计数到最大值 0XFFFF 溢出后计数器的值变为 0，重新开始向上计数，如果需要更改计数周期值可以停止 PCA 更改计数器的初值然后继续计数。可用于 PCA 捕获模式，8 位 PWM 模式。

设置流程

1. 保持 PCA_EPWM 的 EPMW 为 0
2. 设置 PCA_CMOD.CPS 选择计数时钟
3. 根据需要设置 PCA_CMOD.CFIE 设置计数溢出中断使能
4. 时钟 PCA_CCON.CR 启动 PCA 计数器
5. 溢出更改计数器初值需要停止 PCA 计数器

16.2.1.2 16 位重载计数模式

计数器计数到与寄存器 CARR 的值相同时溢出后计数器的值变为 0，继续开始向上计数，可用于 PCA 捕获模式，16 位 PWM 模式。

设置流程

1. 设置 PCA_EPWM 的 EPMW 为 1

2. 设置 PCA_CARR 设置计数周期值
3. 设置 PCA_CMOD.CPS 选择计数时钟
4. 根据需要设置 PCA_CMOD.CFIE 设置计数溢出中断使能
5. 时钟 PCA_CCON.CR 启动 PCA 计数器

16.2.2 PCA 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期，脉冲宽度，占空比和相位差的功能。

引脚上出现的电平跳变导致 PCA 捕捉 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比较寄存器（CCAPx）。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何变化（正沿或负沿）。当捕捉发生时，CCON 中的捕捉/比较标志（CCFn）被置为逻辑‘1’并产生一个中断请求（如果 CCF 中断被允许）。当 CPU 转向中断服务程序时，CCFn 位不能被硬件自动清除，用户软件写 INTCL 寄存器清除此标志位。如果 CCPMx.CAPP 以及 CCAPMx.CAPN 位都被设置为逻辑‘1’，可以通过直接读对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。

分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期，以保证输入信号能够被硬件识别。

CPU 可以在任何时候读取或写入 CCAPx 的寄存器。

捕获设置：

- 当需要在外部上升沿进行捕获，CCPMx.CAPP =“1”以及 CCAPMx.CAPN = “0”
- 当需要在外部下降沿进行捕获，CCPMx.CAPP =“0”以及 CCAPMx.CAPN = “1”
- 当需要在外部上升、下降沿进行捕获，CCPMx.CAPP =“1”以及 CCAPMx.CAPN = “1”
- 根据需要配置捕获中断及中断处理程序。
- 按照定时/计数器配置 PCA 计数器启动。

注意：

– 随后由同一模块的捕获值会覆盖现有捕获的值。为了保持捕获的值，在中断服务程序中将它保存在 RAM 里面，这个操作必须在下一次事件出现之前完成，否则就会丢失前面一次捕获采样值。

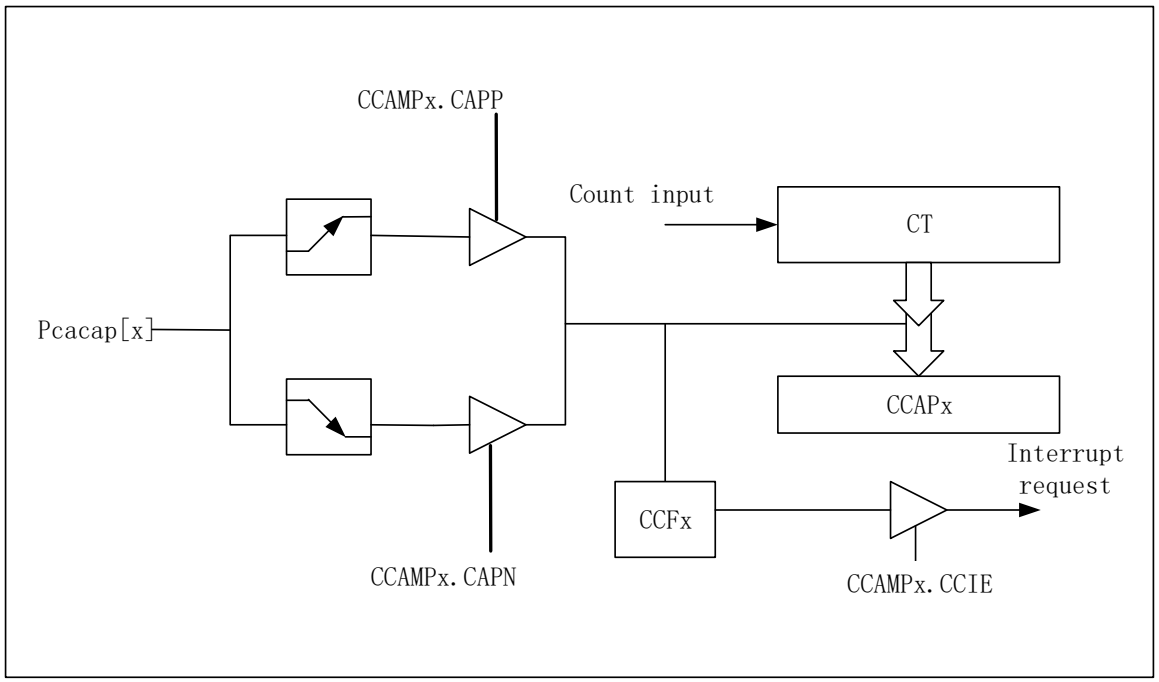


图 16-3 PCA 捕获功能框图

16.2.3 PCA 比较功能

比较功能提供如下功能，高速输出模式，WDT 模式，16 为 PWM 模式和 8 位 PWM 模式。在前三个功能中，比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 CCAPx 寄存器中的 16 位值。在 8 位 PWM 模式下，PCA 模块不断地将 PCA 定时器/计数器低字节寄存器（CNT）与一个在 CCAPxL 模块寄存器 8 位的值进行比较。每 4 个时钟周期比较一次，即与最快的 PCA 定时器/计数器的时钟速率相匹配。

设置 CCAPMx.ECOM 位选择该模块的比较功能。

若要正确使用在比较模式下的模块，请遵守以下的一般程序：

- 选择 PCA 模块的操作模式。
- 选择 PCA 定时器/计数器的输入信号。
- 比较值加载到模块的比较/捕获寄存器对。
- 设置 PCA 定时器/计数器运行控制位。
- 匹配后产生中断，清除模块的比较/捕获标志。

16.2.3.1 比较翻转输出模式

在比较翻转输出模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器（CCAPx）发生匹配时，模块 PCA 的 CH[x]引脚上的逻辑电平将发生变化。这可以提供比切换 IO 输出有更高精度，因为这个输出不会被中断响应而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗，精度都有所欠缺。

要设定一个比较/捕获模块的比较翻转输出模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器（CCAPx）之间的匹配切换 PCA 的 CH[x]信号，并设置模块的比较/捕获标志（CCON.CCFx）。通过软件设置或清除 PCA 的 CH[x]信号，用户可以选择匹配切换信号从低到高或高到低。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位（CCAPMx.CCIE）当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。如果用户在中断程序中不去改变比较/捕获寄存器，PCA 并重新计数比较值，如相匹配则发生下一次翻转。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器（CCAPx）。

注意：

– 为了防止无效的匹配，而更新这些寄存器，用户软件应该写 CCAPxL 的首先然后 CCAPxH。写到 CCAPxL 清除禁用比较功能 ECOM 位，而写到 CCAPxH 设置的 ECOM 位，重新启用比较功能。

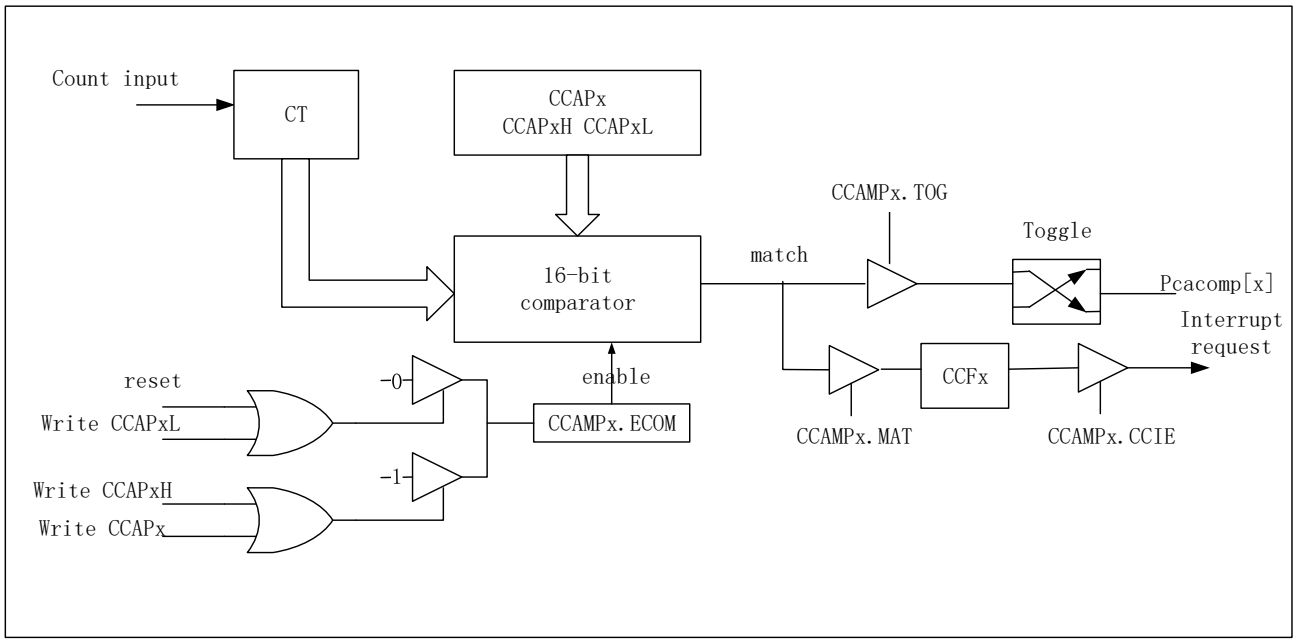


图 16-4 PCA 比较功能框图

16.2.3.2 PCA 16 位 PWM 功能

在比较输出 PWM 模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器（CCAPx）发生匹配时，模块 PCA 的 CH[x]引脚上的逻辑电平将发生变化。计数器溢出时，CH[x]引脚上的逻辑电平将清零。这可以提供一组 16 位的 PWM 输出。

要设定一个比较/捕获模块的 PWM 模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位及 EPWM 寄存器。PCA 定时器/计数器和比较/捕获的寄存器（CCAPx）之间的匹配切换 PCA 的 CH[x]信号，并设置模块的比较/捕获标志（CCON.CCFx）。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位（CCAPMx.CCIE）当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。

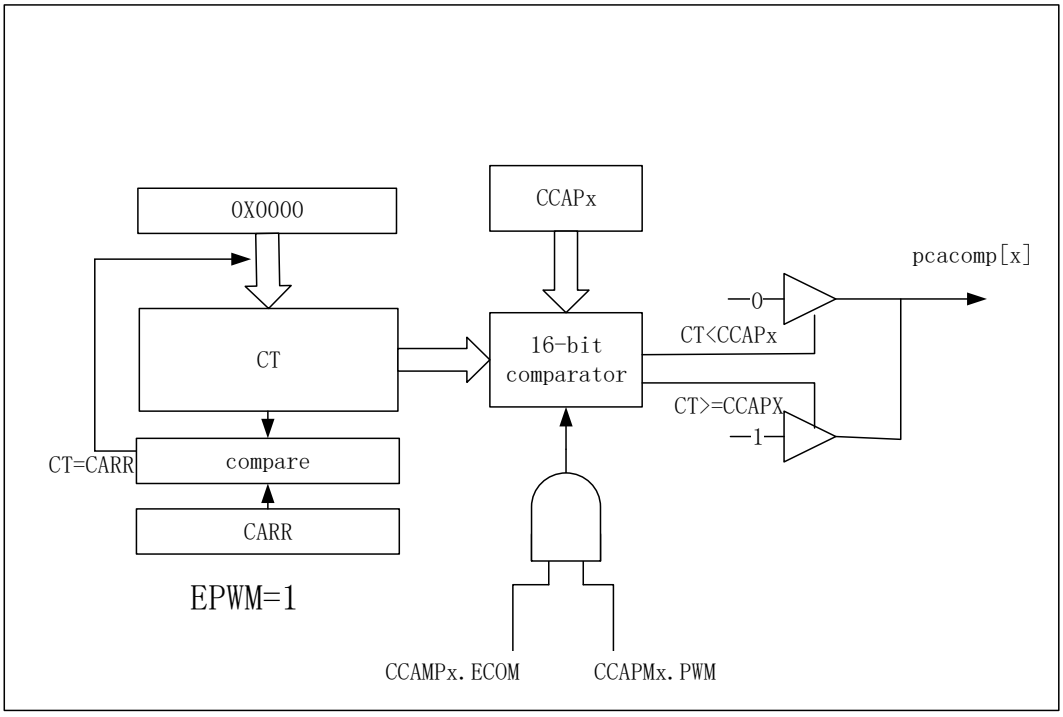


图 16-5 PCA 16 位 PWM 功能框图

16.2.3.3 PCA 模块 4 的 WDT 功能

本系列除了一个 WDT 硬件模块，PCA 的模块 4，还提供一个可编程频率的 16 位 WDT。当 PCA 定时器/计数器的计数值与模块 4 中存储的值比较/捕获寄存器（CCAP4）相匹配时，这种模式产生复位信号。PCA 的 WDT 复位信号做为一个独立的复位信号。与外部复位（RST），硬件看门狗复位（WDTRST）和 LVD 低电压复位，POR 上电下电复位相结合。用户可以自由结合或单独使用它们。模块 4 是具有 WDT 模式唯一 PCA 的模块。当不设置为 WDT 时，它可以在其它模式中独立使用。

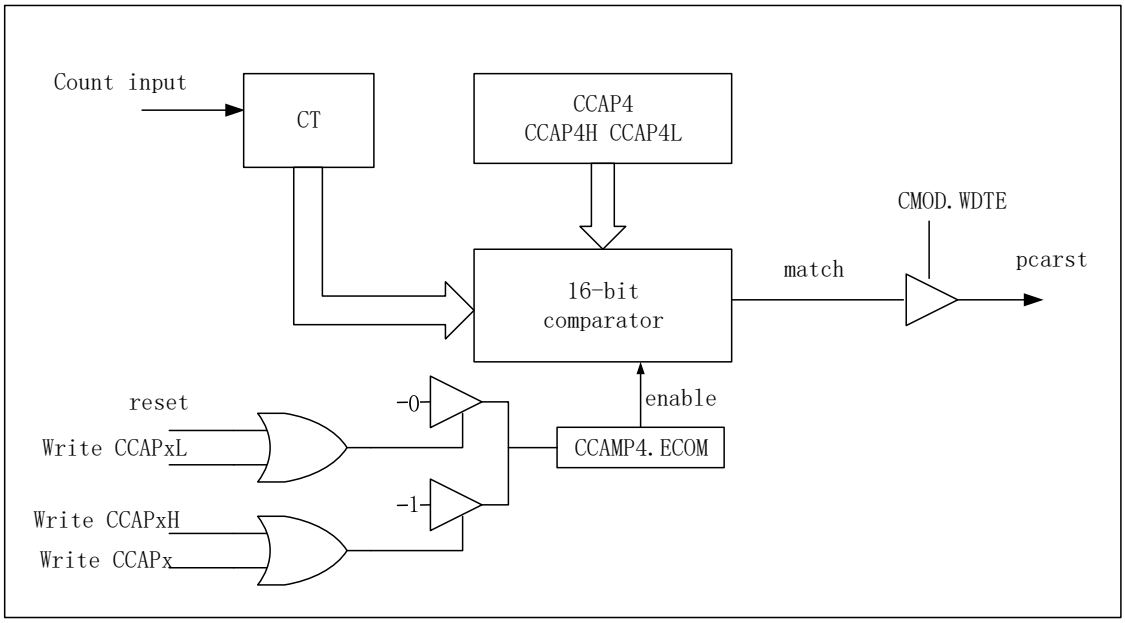


图 16-6 PCA WDT 功能框图

当把 PCA 模块 4 当 WDT 使用时，必须设置 CCAPM4.ECOM4，CCAPM4.MAT4 以及 CMOD.WDTE。另外 PCA 定时器/计数器可以设置 CMOD.CPS 来选择不同的输入计数频率。

在比较/捕获寄存器(CCAP4)输入一个 16 位的比较值。在 PCA 定时器/计数器(CNT)，输入一个 16 位的初始值或使用复位值（0000h）。这些值乘以的 PCA 输入脉冲率之间的差额确定的 WDT 匹配运行时间。设置定时器/计数器运行控制位（CCON.CR）启动 PCA WDT。每次匹配时，PCA 的 WDT 产生复位信号。要防止一个 PCA WDT 复位，用户有三种选择。

- 定期的比较值 CCAP4 的改变，所以匹配永远不会发生。

- 定期更改 PCA 定时器/计数器值（CNT）所以匹配永远不会发生。
- 通过在匹配前清除的CMOD.WDTE位来禁用模块复位输出信号，后来重新启用它。

前两个选项是更可靠的，因为 WDT 在第三个选项没有被禁用。

第二个选项是不推荐，如果其他 PCA 模块都在使用，因为五个模块共享一个共同的时间基。因此，在大多数应用中的第一个选项是最好的。

PCA WDT 配置流程

1) 配置 WDT 比较/捕获寄存器 PCA_CCAP4
2) 配置 PCA 计数寄存器 PCA_CNT
3) 配置 PCA_CCAMP4 选择比较匹配功能
4) 配置 PCA_CMOD 选择输入时钟，使能 WDT 功能
5) 启动 PCA
6) 选择清除 PCA WDT 清除方式在 PCA WDT 复位前清除 PCA WDT

16.2.3.4 PCA 8 位 PWM 功能

脉宽调制是一种使用程序来控制波形占空比，周期，相位的技术。5 个 PCA 模块都可以被独立地用于在对应 PCA 的 CH[x]引脚产生脉宽调制（PWM）输出，脉冲宽度为 8 位分辨率。PWM 输出的频率取决于 PCA 计数器/定时器的时基。使用模块的捕捉/比较寄存器 CCAPxL 来改变 PWM 输出信号的占空比。当 PCA 计数器/定时器的低字节（CL）与 CCAPxL 中的值相等时，PCA 的 CH[x]引脚上的输出被置“1”；当 CL 中的计数值溢出时，PCA 的 CH[x]输出被复位“0”。当计数器/定时器的低字节 CL 溢出时（从 0xFF 到 0x00），保存在 CCAPxH 中的值被自动装入到 CCAPxL，不需软件干预。

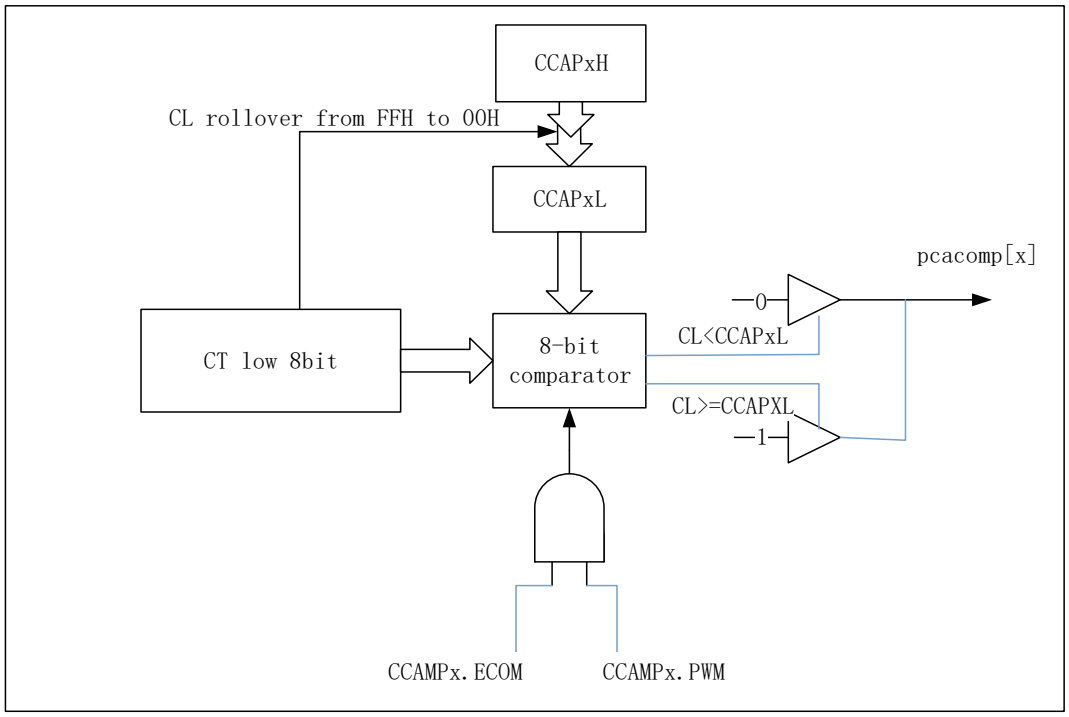


图 16-7 PCA PWM 功能框图

在这种模式下，PCA 定时器/计数器（CL）的低字节中的值是不断在低字节比较/捕获寄存器（CCAPxL）的值相比。当 CL＜CCAPxL，输出波形为低。当两者匹配时（CL= CCAPxL），输出波形去到高，直到 CL 溢出从 FFH 到 00H，结束期间仍然很高。在溢出时，在 CCAPxH 的值自动装载到 CCAPxL 内，一个新的周期的开始。

在 CCAPxL 的值决定当前波形的占空比。在 CCAPxH 的值确定下一个波形的占空比。改变 CCAPxL 中的值即可更改的脉冲宽度调制。正如图所示，8 位值在 CCAPxL 可以从 0（100％占空比），到 255（0.4％占空比）。要改变 CCAPxL 值而不会产生毛刺，需要在高字节寄存器（CCAPxH）写入一个新值。当 CL 超过 FFH 滚动到 00h，这个值是由硬件自动加载到 CCAPxL。

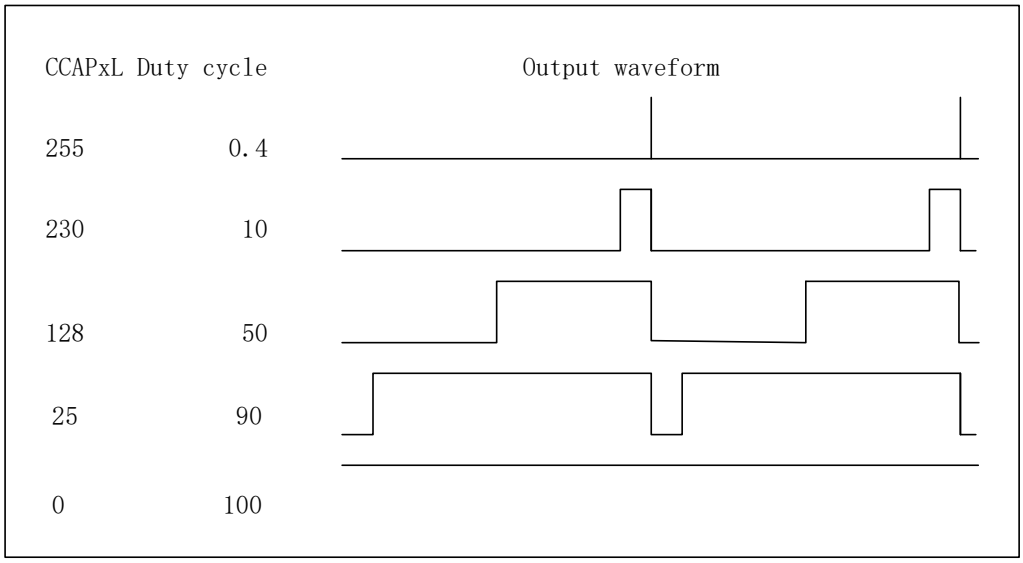


图 16-8 PCA PWM 输出波形

要设定一个比较/捕获模块在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。另外 PCA 定时器/计数器由编程 CMOD.CSP[2:0]可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位（CCON.CR）启动 PCA 定时器/计数器。

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	EPWM	工作方式
X	1	0	0	0	0	X	0	用正沿触发捕获
X	0	1	0	0	0	X	0	用负沿触发捕获
X	1	1	0	0	0	X	0	用跳变沿触发捕获
1	0	0	1	0	0	X	0	软件定时器
1	0	0	1	1	0	X	0	高速输出
1	0	0	0	0	1	X	0	8位脉冲宽度调制器
1	0	0	1	1	0	X	1	16位脉冲宽度调制器

表 16-1 PCA 比较捕获功能模块设置

16.3 PCA 模块与其他模块互连及控制

PCA 通过端口功能选择可以连通到其他模块或端口。

		0	1	2	3	4	5	6	7
GPIO_PCAS[2:0]	PCA_ECI	PX_SEL	Res.	LVD	VC0	VC1	PA05	PB02	PD02
GPIO_PCAS[5:3]	PCA_CH0	PX_SEL	Res.	Res.	LVD	VC1	PA06	PB04	PC06

当 GPIO_PCAS[2:0]=0x0 时，PCA_ECI 输入为 PX_SEL 选择的端口输入，当 GPIO_PCAS[2:0]=0x1~0X7 时，连接其他模块的输入或输出。

当 GPIO_PCAS[5:3]=0x0 时，PCA_CH0 捕获输入为 PX_SEL 选择的端口输入，当 GPIO_PCAS[5:3]=0x1~0X7 时，连接其他模块的输入或输出。

16.4 PCA 寄存器描述

基地址 0X40001000

寄存器	偏移地址	描述
PCA_CCON	0X000	PCA 控制寄存器
PCA_CMOD	0X004	PCA 模式寄存器
PCA_CNT	0X008	PCA 计数寄存器
PCA_INT_CL	0X00C	PCA 中断清除寄存器
PCA_CCAPM0	0x010	PCA 比较/捕获模块0模式寄存器
PCA_CCAPM1	0x014	PCA 比较/捕获模块1模式寄存器
PCA_CCAPM2	0x018	PCA 比较/捕获模块2模式寄存器
PCA_CCAPM3	0x01C	PCA 比较/捕获模块3模式寄存器
PCA_CCAPM4	0x020	PCA 比较/捕获模块4模式寄存器
PCA_CCAP0H	0X024	PCA 比较/捕获模块0高8位寄存器
PCA_CCAP0L	0X028	PCA 比较/捕获模块0低8位寄存器
PCA_CCAP1H	0X02C	PCA 比较/捕获模块1高8位寄存器
PCA_CCAP1L	0X030	PCA 比较/捕获模块1低8位寄存器
PCA_CCAP2H	0X034	PCA 比较/捕获模块2高8位寄存器
PCA_CCAP2L	0X038	PCA 比较/捕获模块2低8位寄存器
PCA_CCAP3H	0X03C	PCA 比较/捕获模块3高8位寄存器
PCA_CCAP3L	0X040	PCA 比较/捕获模块3低8位寄存器
PCA_CCAP4H	0X044	PCA 比较/捕获模块4高8位寄存器
PCA_CCAP4L	0X048	PCA 比较/捕获模块4低8位寄存器
PCA_CCAPO	0X04C	PCA PWM与高速输出标志寄存器
PCA_CCAP0	0X050	PCA 比较/捕获模块0 的16位寄存器
PCA_CCAP1	0X054	PCA 比较/捕获模块1 的16位寄存器
PCA_CCAP2	0X058	PCA 比较/捕获模块2 的16位寄存器
PCA_CCAP3	0X05C	PCA 比较/捕获模块3 的16位寄存器
PCA_CCAP4	0X060	PCA 比较/捕获模块4 的16位寄存器
PCA_CARR	0X064	PCA 周期装载寄存器
PCA_EPWM	0X068	PCA PWM增强寄存器

表 16-2 PCA 寄存器列表

16.4.1 控制寄存器（PCA_CCON）

偏移地址：0x000

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved	CF	CR	Reserved	CCF4	CCF3	CCF2	CCF1	CCF0
	RO	RW		RO	RO	RO	RO	RO

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志（写无效） 当PCA计数溢出时，CF由硬件置位，如果CMOD寄存器的CFIE位为1，则CF标志可以产生中断 1：发生计数器溢出；0：无溢出；
6	CR	PCA计数器运行控制位 1：启动PCA计数器计数 0：关闭PCA计数器计数
5	Reserved	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM4.CCIE置位时，这个标志位会产生一个PCA中断
3	CCF3	PCA计数器模块3 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM3.CCIE置位时，这个标志位会产生一个PCA中断
2	CCF2	PCA计数器模块2 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM2.CCIE置位时，这个标志位会产生一个PCA中断
1	CCF1	PCA计数器模块1 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM1.CCIE置位时，这个标志位会产生一个PCA中断
0	CCF0	PCA计数器模块0 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM0.CCIE置位时，这个标志位会产生一个PCA中断

16.4.2 模式寄存器（PCA_CMOD）

偏移地址：0x004

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved	CIDL	WDTE	Reserved		CPS			CFIE
	RW	RW			RW			RW

位	符号	描述
31:8	Reserved	保留位
7	CIDL	空闲模式IDLE下，PCA是否停止工作 1：休眠模式（sleep）下，PCA停止工作 0：休眠模式（sleep）下，PCA继续工作
6	WDTE	PCA WDT功能使能控制位 1：启动PCA模块4 WDT功能 0：关闭PCA模块4 WDT功能
5:4	Reserved	保留位
3:1	CPS[2:0]	时钟分频选择及时钟源选择 000：PCLK/32 001：PCLK/16 010：PCLK/8 011：PCLK/4 100：PCLK/2 101：timer0 overflow 110：timer1 overflow 111：ECI 外部时钟，时钟PCLK四分频采样
0	CFIE	PCA计数器中断使能控制信号 1：使能中断 0：关闭中断

16.4.3 计数寄存器（PCA_CNT）

偏移地址：0x008

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数器的值 只有在PCA停止状态CNT才可以写入。否则写入无效

16.4.4 中断清除寄存器（PCA_ICLR）

偏移地址：0x00C

复位值: 0x0000 009Fh

31-8	7	6	5	4	3	2	1	0
Reserved	CF	Reserved		CCF4	CCF3	CCF2	CCF1	CCF0
	R1W0			R1W0	R1W0	R1W0	R1W0	R1W0

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志清除（软件写0清零，写1无效），读出值为1
6:5	RSV	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
3	CCF3	PCA计数器模块3 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
2	CCF2	PCA计数器模块2 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
1	CCF1	PCA计数器模块1 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
0	CCF0	PCA计数器模块0 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1

16.4.5 比较捕获模式寄存器（PCA_CCAPM0~4）

偏移地址

CCAPM0：0x010;　　CCAPM1：0x014;　　CCAPM2：0x018;

CCAPM3：0x01C;　　CCAPM4：0x020;

复位值: 0x0000 0000h

31-8	7	6	5	4	3	2	1	0
Reserved		ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE
		RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:7	Reserved	保留位
6	ECOM	允许比较器功能控制位 1：允许比较器功能；0：禁止比较强功能； 当PCA用于软件计数器，高速输出，PWM模式，WDT模式，要置位ECOM 写CCAMPHx或CCAMPx寄存器会自动置位ECOM；写CCAMPLx寄存器会自动清除ECOM位
5	CAPP	正沿捕获控制位 1：允许上升沿捕获；0：禁止上升沿捕获
4	CAPN	负沿捕获控制位 1：允许下降沿捕获；0：禁止下降沿捕获
3	MAT	允许匹配控制位 1：PCA计数值与模块的比较/捕获寄存器的值一旦匹配，将置位CCON寄存的中断标志CCFx(x=0-4) 0：禁止匹配功能
2	TOG	翻转控制位 1：工作在PCA高速输出模式，PCA计数器的值与模块的比较/捕获寄存器的值一旦匹配，CCPx引脚翻转 0：禁止翻转功能
1	PWM	脉宽调制控制位 1：允许CCPx引脚作为PWM输出 0：禁止PWM脉宽调制功能 **只有CCAPMx[6:0]=100_0010时，PWM功能才有效**
0	CCIE	PCA使能中断 1：使能比较/捕获中断 0：PCA比较/捕获功能中断禁止

16.4.6 比较捕获数据寄存器高 8 位（PCA_CCAP0~4H）

偏移地址

CCAP0H：0x024;　　CCAP1H：0x02C;　　CCAP2H：0x034;

CCAP3H：0x03C;　　CCAP4H：0x044;

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx[15：8]							
	RW							

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[15:8]	比较/捕获模式高8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的高8位；写CCAPxH寄存器会自动置位寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比装载寄存器，在计数器低8位溢出时，装载寄存器会自动更新到PWM比较寄存器

16.4.7 比较捕获数据寄存器低 8 位（PCA_CCAP0~4L）

偏移地址

CCAP0L：0x028;　　CCAP1L：0x030;　　CCAP2L：0x038;

CCAP3L：0x040;　　CCAP4L：0x048;

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx[7：0]							
	RW							

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[7:0]	比较/捕获模式低8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的低8位；写CCAPxL寄存器会自动清除寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比比较寄存器，在PWM模式，计数器的低8位的值小于CCAPx[7:0]的值PWM输出低电平，否则PWM输出高电平。

16.4.8 比较捕获 16 位寄存器（PCA_CCAP0~4）

偏移地址

CCAP0：0x050;　　CCAP1：0x054;　　CCAP2：0x058;

CCAP3：0x05C;　　CCAP4：0x060;

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCAPx[15：0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCAPx	比较/捕获模式16位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值；写CCAPx寄存器会置位寄存器CCAPMx的ECOM位。 写CCAPX寄存器相当于写CCAPxL及CCAPxH这两个8位寄存器。在比较/捕获模式下可以直接读写这个寄存器，在PWM模式下，使用CCAPxL及CCAPxH寄存器

16.4.9 比较高速输出标志寄存器（PCA_CCAPO）

偏移地址：0x04C

复位值: 0x0000 0000h

31:8	7	6	5	4	3	2	1	0
Reserved				CCAPO4	CCAPO3	CCAPO2	CCAPO1	CCAPO0
				RW	RW	RW	RW	RW

位	符号	描述
31:5	Reserved	保留位
4	CCAPO4	比较模块4的输出值
3	CCAPO3	比较模块3的输出值
2	CCAPO2	比较模块2的输出值
1	CCAPO1	比较模块1的输出值
0	CCAPO0	比较模块0的输出值

16.4.10 周期寄存器（PCA_CARR）

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARR[15：0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CARR	计数周期重载寄存器

16.4.11 增强 PWM 控制（PCA_EPWM）

复位值: 0x0000 0000h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	符号	描述
31:1	Reserved	保留位
0	EPWM	16 bit PWM 使能

17 高级定时器（TIM4/5/6）

17.1 Advanced Timer 简介

Advanced Timer 是一个包含三个定时器 Timer4/5/6。Timer4/5/6 功能相同的高性能计数器，可用于计数产生不同形式的时钟波形，1 个定时器可以产生互补的一对 PWM 或者独立的 2 路 PWM 输出，可以捕获外界输入进行脉冲宽度或周期测量。

Advanced Timer 基本的功能及特性如表所示。

波形模式	锯齿波、三角波
基本功能	• 递加、递减计数方向
	• 软件同步
	• 硬件同步
	• 缓存功能
	• 正交编码计数
	• 通用 PWM 输出
	• 保护机制
	• AOS 关联动作
中断类型	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断

表 17-1 Advanced Timer 基本特性

端口名	方向	功能
TIMx_CHA	输入/输出	正交编码计数时钟输入端口或捕获输入端口或比较输出端口（x=4~6）
TIMx_CHB		
TIMTRIA	输入	硬件计数时钟输入端口或捕获输入端口 硬件启动、停止、清零条件输入端口
TIMTRIB		
TIMTRIC		
TIMTRID		

表 17-2 Advanced Timer 端口列表

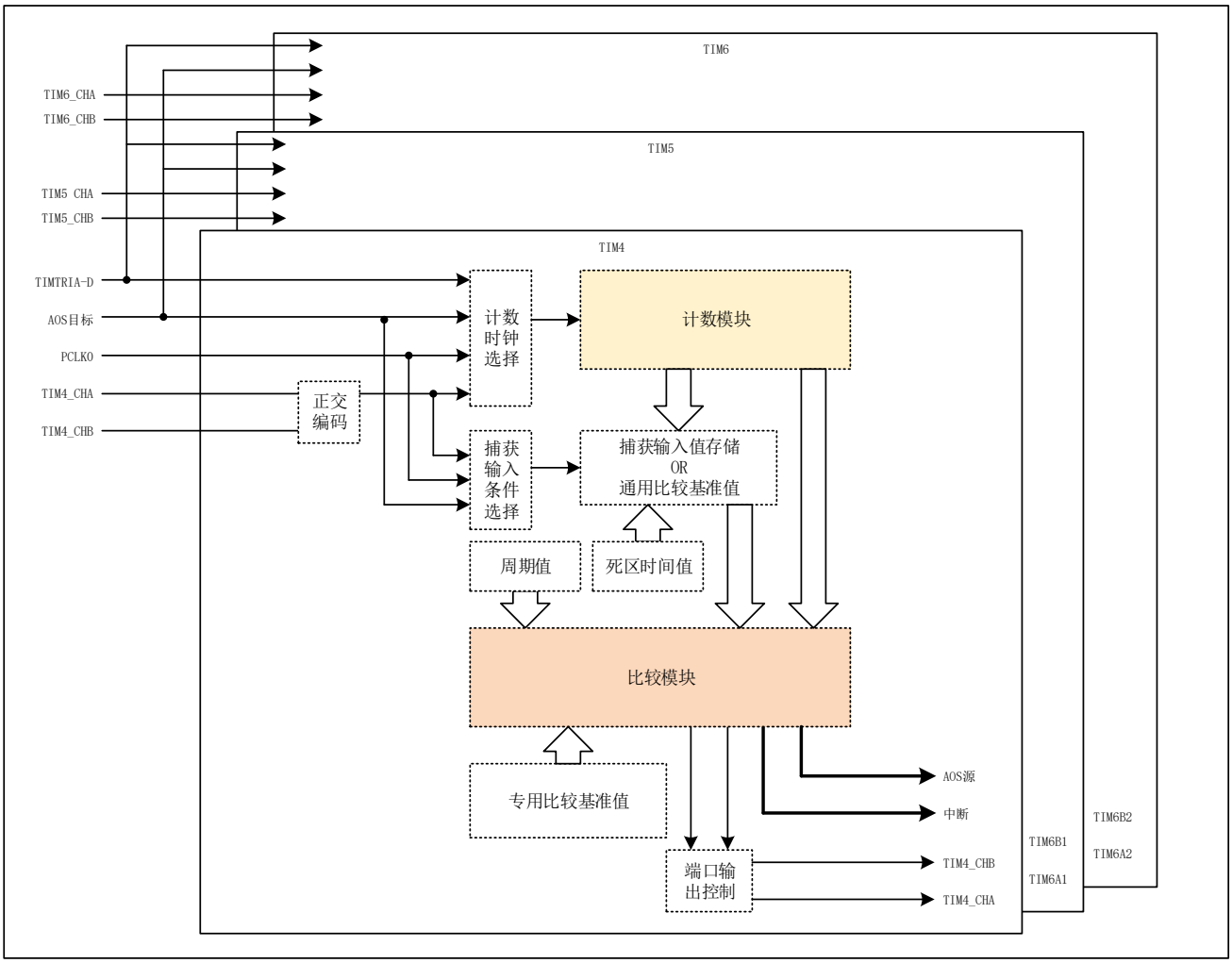


图 17-1 Advanced Timer 框图

17.2 Advanced Timer 功能描述

17.2.1 基本动作

17.2.1.1 基本波形模式

Timer4/5/6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。波形模式又由于不同的内部计数动作有所细分，三角波模式分为三角波 A 模式、三角波 B 模式。锯齿波和三角波的基本波形如图 17-2 图 17-3 所示。三角波 A 模式与三角波 B 模式区别在于缓存传送有差别，三角波 A 模式一个周期只发生一次缓存传送（谷点），而三角波 B 模式一个周期发生两次缓存传送（峰点和谷点）。

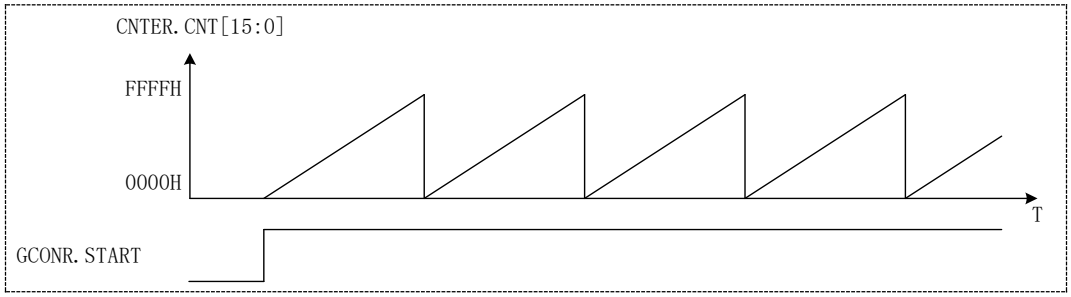


图 17-2 锯齿波波形（递加计数）

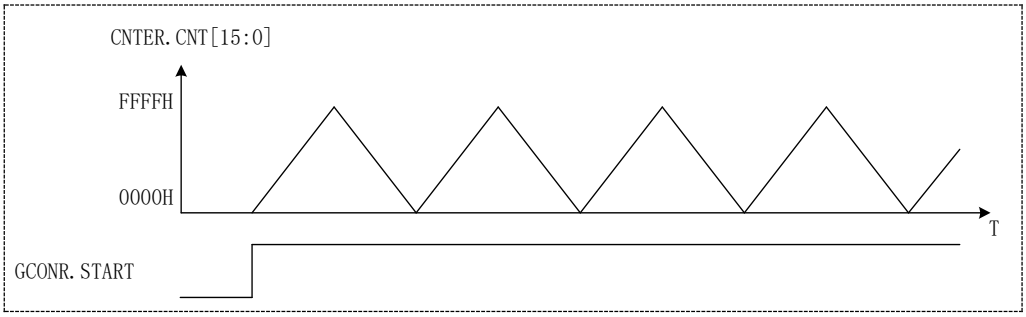


图 17-3 三角波波形

17.2.1.2 比较输出

Timer4/5/6 一个定时器有 2 个比较输出端口（CHxA、CHxB），可在计数值与计数基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器分别对应了 CHxA、CHxB 的计数比较基准值。当计数器的计数值和 GCMAR 相等时，CHxA 端口输出指定的电平；当计数器的计数值和 GCMBR 相等时，CHxB 端口输出指定电平。

CHxA、CHxB 端口的计数起始电平、停止电平、计数比较匹配时的电平等，可由端口控制寄存器（PCONR）的 PCONR.STACA、PCONR.STPCA、PCONR.STASTPSA、PCONR.CMPCA[1:0]、PCONR.PERCA[1:0] 和 PCONR.STACB、PCONR.STPCB、PCONR.STASTPSB、PCONR.CMPCB[1:0]、PCONR.PERCB[1:0] 位设定。图 17-4 为比较输出的动作例。

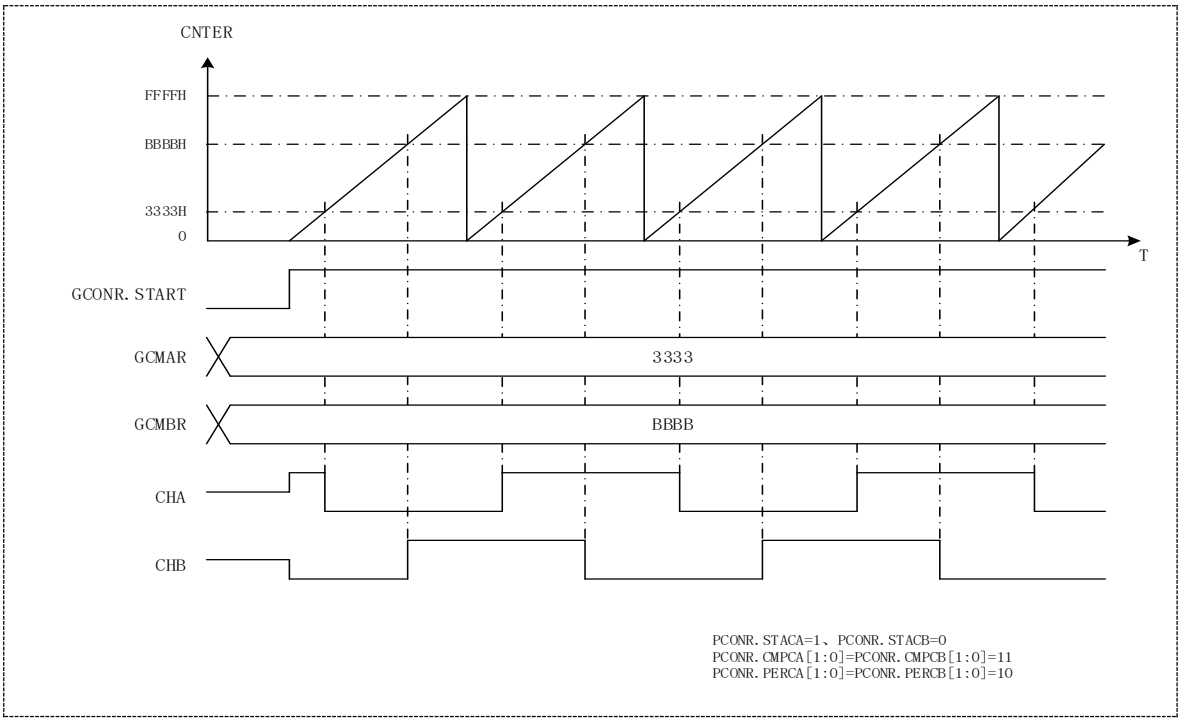


图 17-4 比较输出动作

17.2.1.3 捕获输入

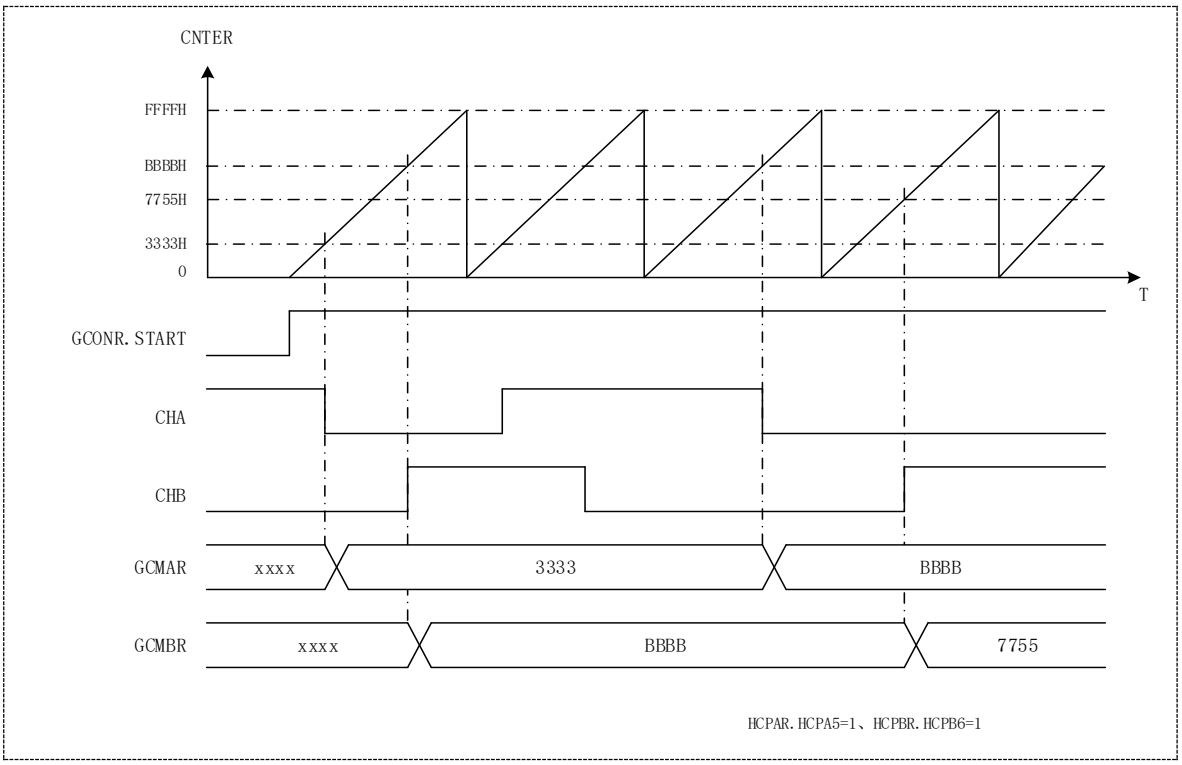


图 17-5 捕获输入动作

Timer4/5/6 都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的计数值。设定端口控制寄存器（PCONR）的 PCONR.CAPCA、PCONR.CAPCB 位为 1，捕获输入功能就有效了。当设定了对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器（GCMAR、GCMBR）中。

每组捕获输入的条件可以是 AOS 事件触发、TIMTRIA-TIMTRID 输入、CHxA 或 CHxB 的输入等，具体的条件选择可通过硬件捕获事件选择寄存器（HCPAR、HCPBR）来设定。图 17-5 为捕获输入的动作例。

17.2.2 时钟源选择

Timer4/5/6 的计数时钟可以有以下几种选择：

a. PCLK 及 PCLK 的 2、4、8、16、64、256、1024 分频 (GCONR.CKDIV[2:0] 设定)

b. AOS 事件触发输入（HCUPR.HCUP [19:16] 或 HCDOR.HCDO [19:16] 设定）

c. CHxA 和 CHxB 的正交编码输入（HCUPR.HCUP[7:0] 或 HCDOR.HCDO [7:0] 设定）

d. TIMTRIA-TIMTRID 的端口输入（HCUPR.HCUP [15:8] 或 HCDOR.HCDO [15:8] 设定）

从上述描述可以看到，b、c、d 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d 时钟时，a 时钟自动无效。

17.2.3 计数方向

Timer4/5/6 的计数器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

17.2.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在计数器计数中或停止时设定。

在递加计数中时，设定 GCONR.DIR=0（递减计数），则计数器计数到上溢后变为递减计数模式；在递减计数中时，设定 GCONR.DIR=1（递加计数），则计数器计数到下溢后变为递加计数模式。

在计数停止时，设定 GCONR.DIR 位。则计数开始后直至上溢或下溢时，GCONR.DIR 的设定才会反映到计数中。

17.2.3.2 三角波计数方向

三角波模式时，计数方向只能在计数器停止时设定。在计数中设定计数方向无效。

在计数停止时，设定 GCONR.DIR 位。则计数开始后直至上溢或下溢时，GCONR.DIR 的设定才会反映到计数中。

17.2.4 数字滤波

Timer4/5/6 的 CHxA、CHxB、TIMTRIA~D 端口输入都有数字滤波功能。可通过设定滤波控制寄存器（FCONR）的相关使能位开启对应端口的滤波功能。滤波用的基准时钟也通过滤波控制寄存器（FCONR）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 17-6 所示。

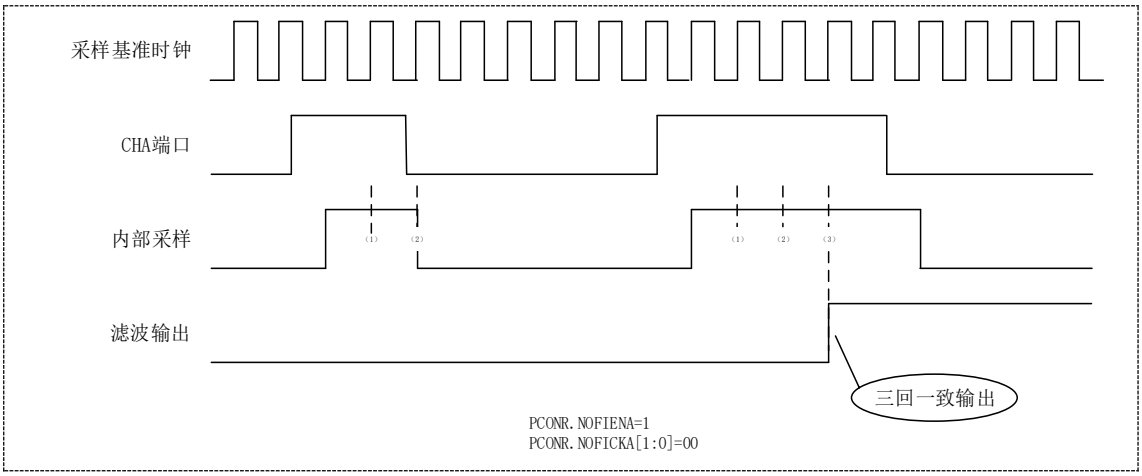


图 17-6 捕获输入端口的滤波功能

TIMTRIA~D 端口是一组 Timer4/5/6 间共用的端口，该组端口的数字滤波功能在只在 Timer4 实现，其他定时器 Timer5/6 对该组端口的数字滤波功能设定无效。

17.2.5 软件同步

17.2.5.1 软件同步启动

Timer4/5/6 可通过设定软件同步启动寄存器（SSTAR）的相关位，实现目标 Timer4/5/6 的同步启动。

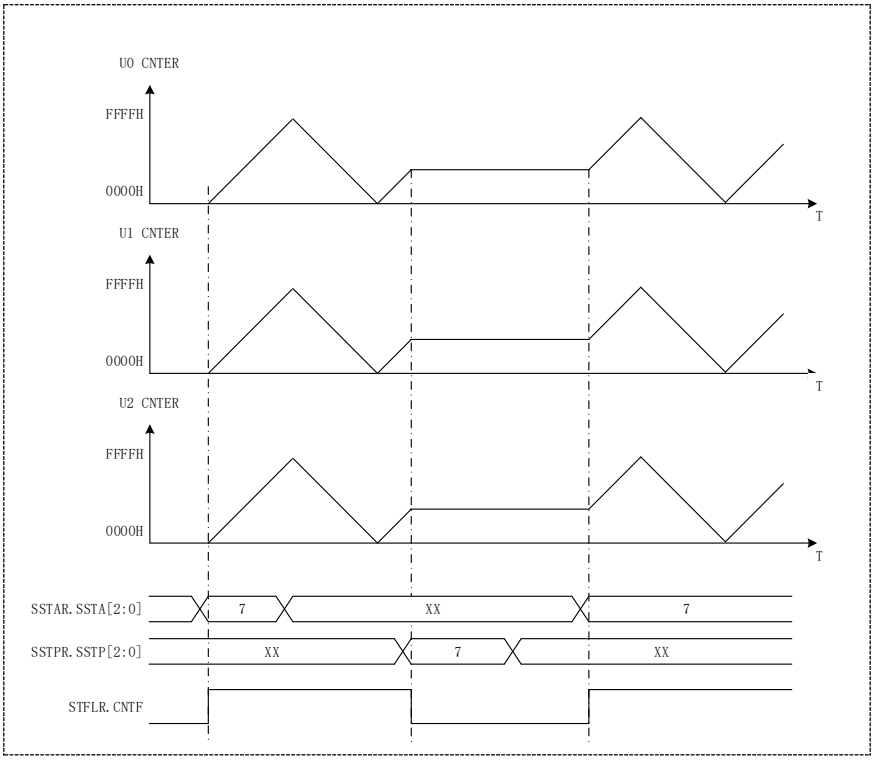


图 17-7 软件同步动作

17.2.5.2 软件同步停止

Timer4/5/6 可通过设定软件同步停止寄存器（SSTPR）的相关位，实现目标 Timer4/5/6 的同步停止。

17.2.5.3 软件同步清零

Timer4/5/6 可通过设定软件同步清零寄存器（SCLRR）的相关位，实现目标 Timer4/5/6 的同步清零。

如图 17-7 所示、若设定 Timer4 的 SSTAR.SSTA0=SSTAR.SSTA1=SSTAR.SSTA2,即可实现 Timer4/5/6 的软件同步启动。

软件同步动作相关寄存器（SSTAR、SSTPR、SCLRR）是一组独立于 Timer4/5/6 外、各个 TIM 间共用的寄存器，这组寄存器的各个位只在写 1 时有效，写 0 无效。在读取 SSTAR 寄存器时，会读出各个定时器的计数器状态，在读取 SSTPR 或 SCLRR 时，会读出 0。

17.2.6 硬件同步

每个定时器除独立拥有 2 个通用输入端口（CHxA、CHxB）外，还共同拥有 4 个外部通用输入端口（TIMTRIA、TIMTRIB、TIMTRIC、TIMTRID）及 4 个 AOS 目标，可实现定时器间的硬件同步动作。

17.2.6.1 硬件同步启动

各 Timer4/5/6 均可选择用硬件方式启动计数器，选择相同硬件启动条件的定时器即可在启动条件有效时实现同步启动。具体的硬件启动条件由硬件启动事件选择寄存器（HSTAR）的设定来决定。

17.2.6.2 硬件同步停止

各 Timer4/5/6 均可选择用硬件方式停止计数器，选择相同硬件停止条件的定时器即可在停止条件有效时实现同步停止。具体的硬件停止条件由硬件停止事件选择寄存器（HSTPR）的设定来决定。

17.2.6.3 硬件同步清零

各 Timer4/5/6 均可选择用硬件方式清零计数器，选择相同硬件清零条件的定时器即可在清零条件有效时实现同步清零。具体的硬件清零条件由硬件清零事件选择寄存器（HCLRR）的设定来决定。

17.2.6.4 硬件同步捕获输入

各 Timer4/5/6 均可选择用硬件方式实现捕获输入功能，选择相同捕获输入功能条件的定时器即可在捕获输入功能条件有效时实现同步捕获输入。具体的硬件捕获输入功能条件由硬件捕获事件选择寄存器（HCPAR、HCPBR）的设定来决定。

17.2.6.5 硬件同步计数

Timer4/5/6 均可选择用硬件输入作为 CLOCK 进行计数，选择相同硬件计数条件的定时器即可在硬件计数 CLOCK 有效时实现同步计数。具体的硬件计数条件由硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）的设定来决定。选择硬件同步计数功能时，只是选择了外部输入时钟源，不影响计数器的启动、停止、清零动作。计数器的启动、停止、清零等还需要单独设定。

图 17-8 所示、Timer4/5/6 的硬件同步动作例。

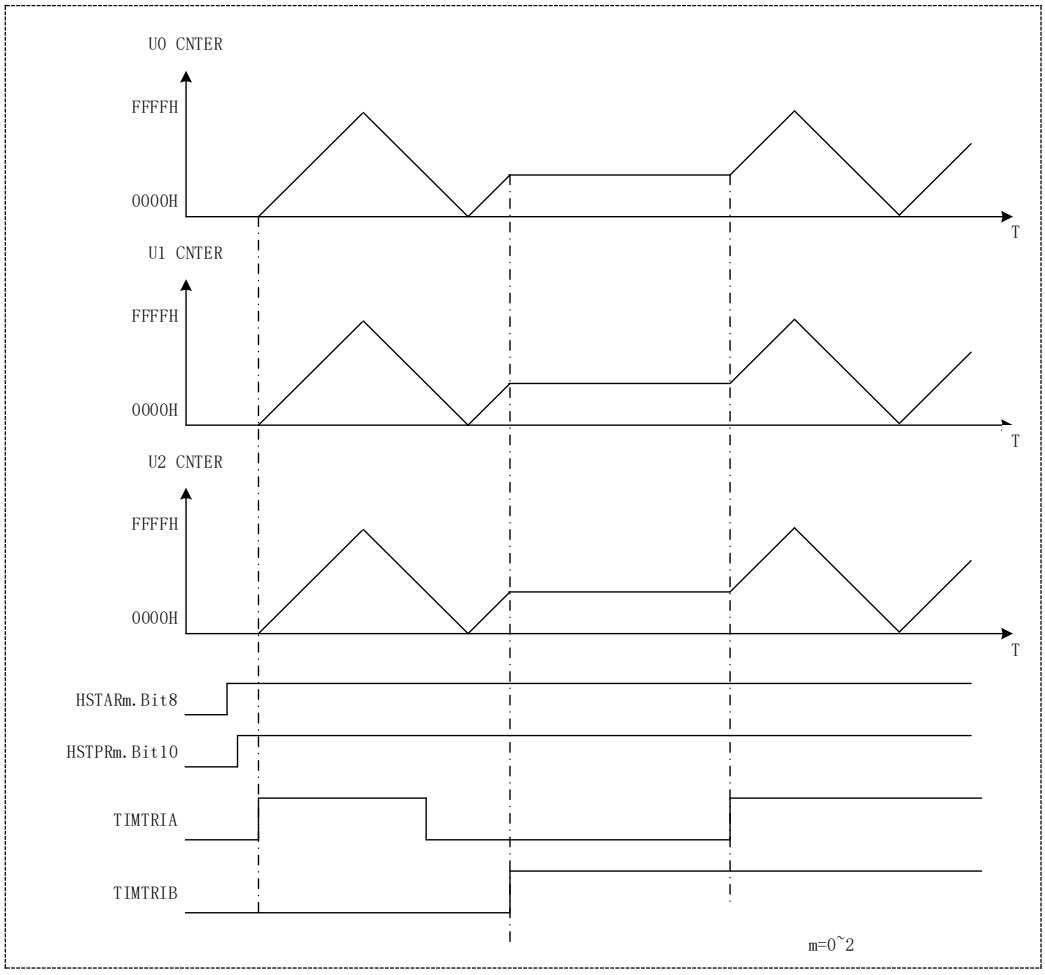


图 17-8 硬件同步动作

17.2.7 缓存功能

缓存动作是指通过设定缓存控制寄存器（BCONR），在缓存传送时间点，选择发生以下事件：

a. 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中

b. 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）中（比较输出时）

c. 通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）

图 17-9 所示，是比较输出动作时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

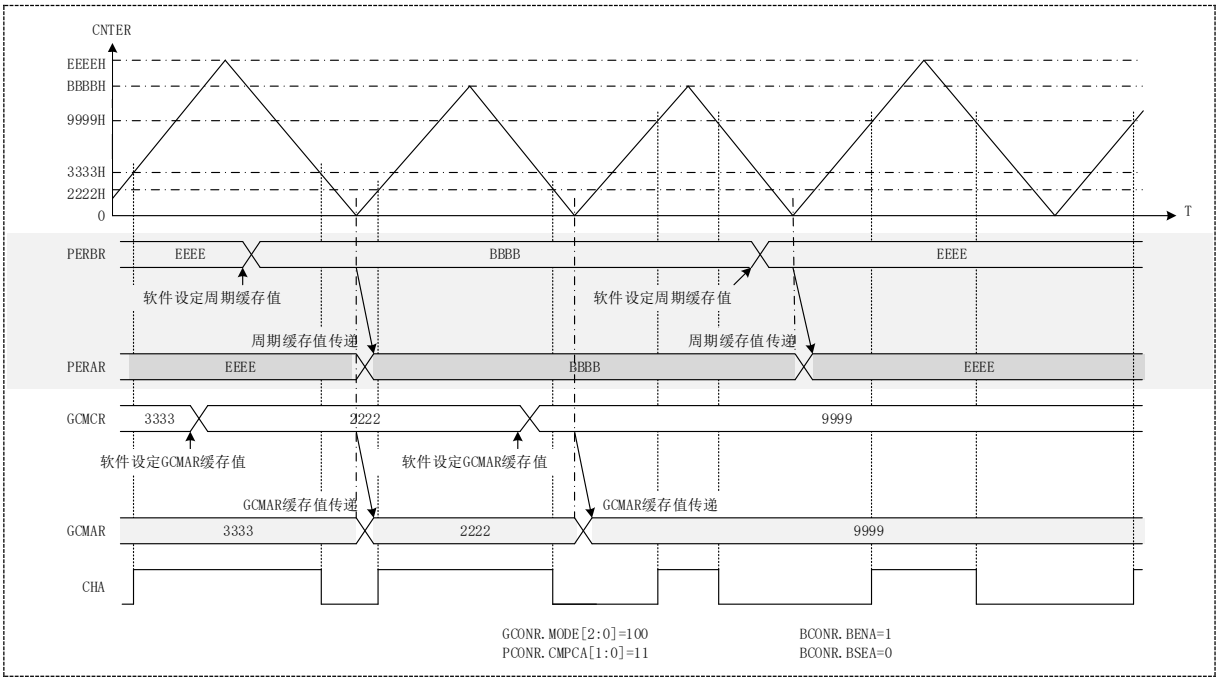


图 17-9 单缓存方式比较输出时序

17.2.7.1 缓存传送时间点

17.2.7.2 通用周期基准值缓存传送时间点

周期基准值缓存传送时间点为锯齿波时递加计数上溢点或递减计数下溢点、三角波时计数谷点。

17.2.7.3 通用比较基准值缓存传送时间点

锯齿波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在上溢点或下溢点。

三角波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点。

三角波 B 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点和计数峰点。

17.2.7.4 捕获输入值缓存传送时间点

捕获输入动作缓存传送时间点为捕获输入动作时。

17.2.7.5 清零动作时缓存传送

在锯齿波计数模式或硬件计数模式时，正常的比较输出动作期间若有清零动作产生，通用周期基准值、通用比较基准值、等会根据相应的缓存动作设定状况发生一次缓存传送。

17.2.8 通用 PWM 输出

17.2.8.1 PWM 展频输出

为了降低 PWM 输出对外部的干扰，在 PWM 输出级有展频配置。每个 PWM 输出周期会微调 PWM 输出的相位。

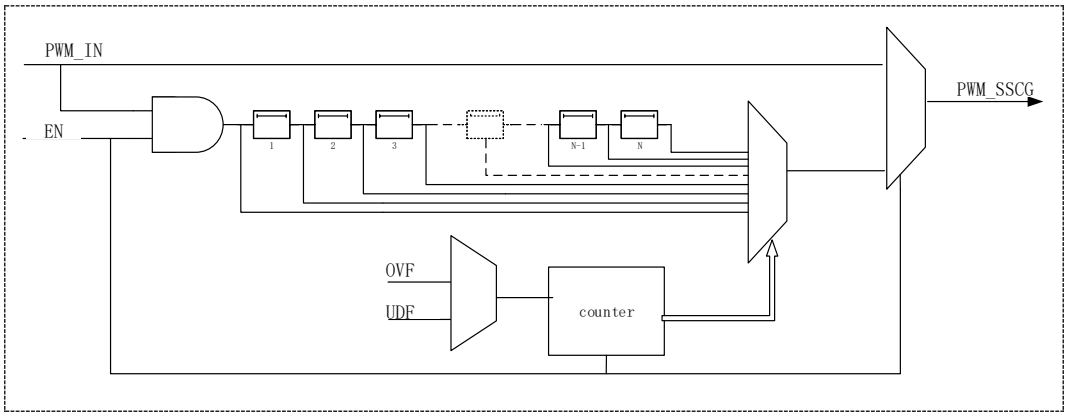


图 17-10 PWM 展频输出示意图

17.2.8.2 独立 PWM 输出

每个定时器的 2 个端口 CHxA、CHxB 能独立的输出 PWM 波。如图 17-11，定时器 Timer6 的 CHA 端口输出 PWM 波。

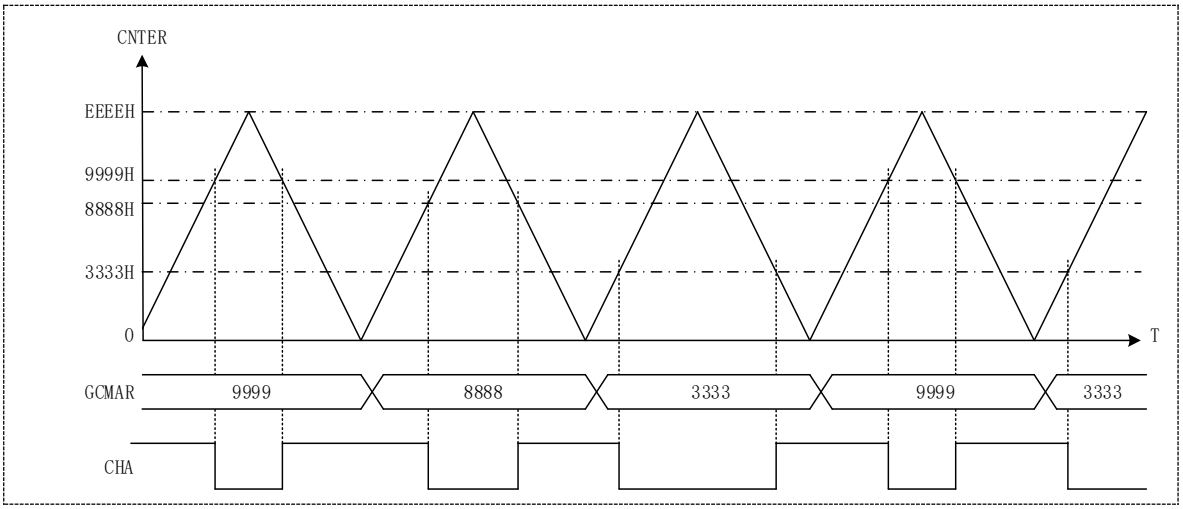


图 17-11 CHA 输出 PWM 波

17.2.8.3 互补 PWM 输出

CHxA 端口和 CHxB 端口，在不同的模式下可组合输出互补 PWM 波形。

软件设定 GCMBR 互补 PWM 输出

软件设定 GCMBR 互补 PWM 输出是指在锯齿波模式和三角波 A 模式、三角波 B 模式下，用于 CHxB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由寄存器直接设定，与通用比较基准值寄存器（GCMAR）的值没有直接关系。

图 17-12 为软件设定 GCMBR 互补 PWM 波的输出例。

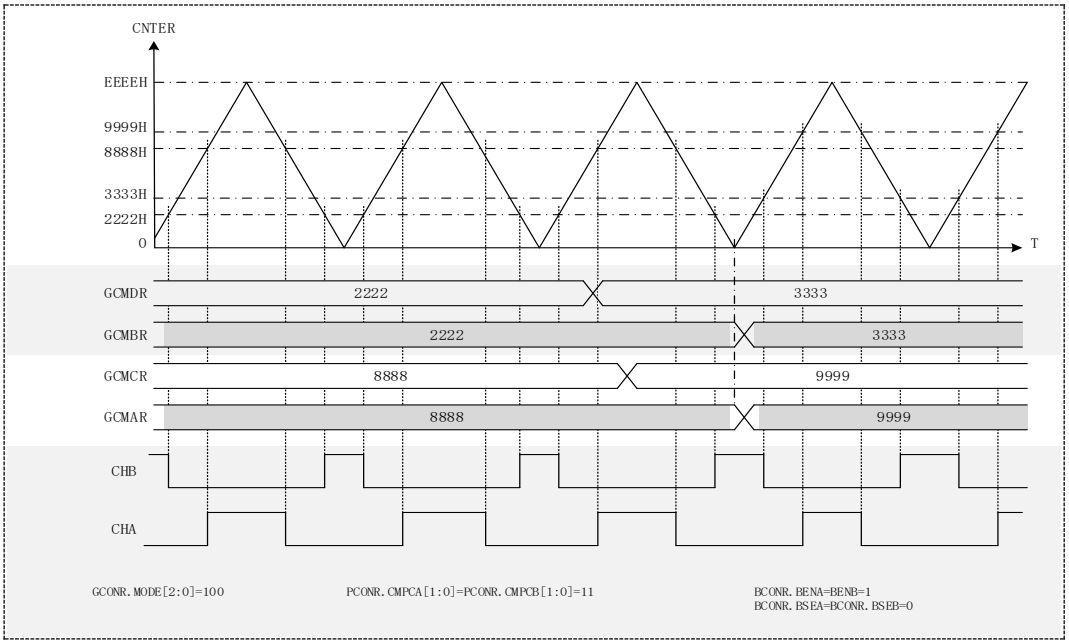


图 17-12 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出

硬件设定 GCMBR 互补 PWM 输出

硬件设定 GCMBR 互补 PWM 输出是指在三角波 A 模式、三角波 B 模式下，用于 CHxB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由通用比较基准值寄存器（GCMAR）和死区时间基准值寄存器（DTUAR、DTDAR）的值运算决定。

图 17-13 为硬件设定 GCMBR 互补 PWM 波输出例。

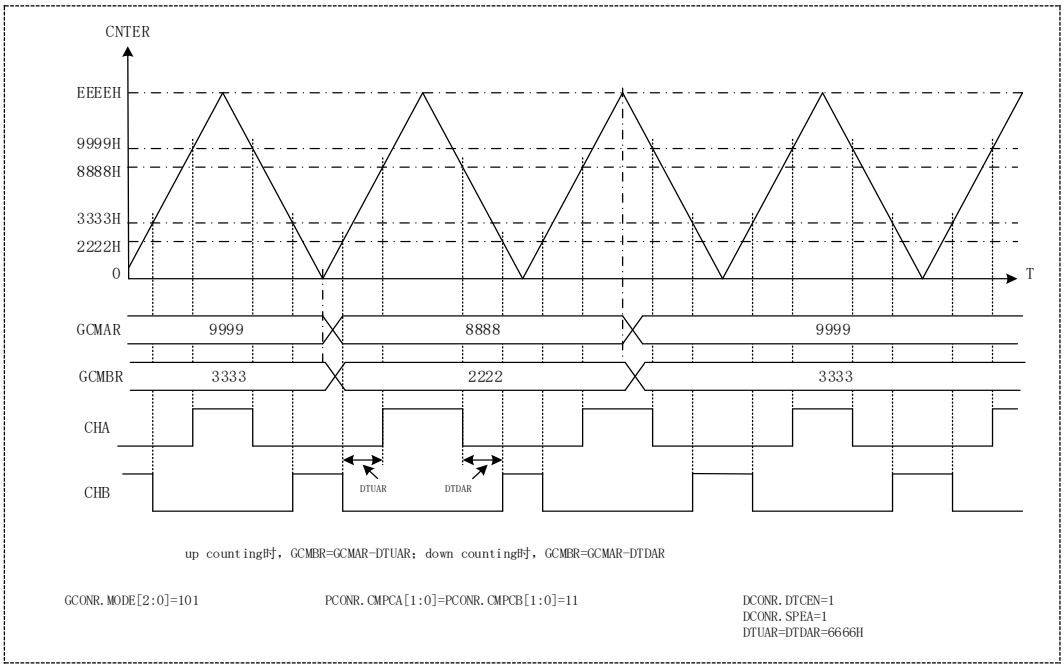


图 17-13 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）

17.2.8.4 多相 PWM 输出

每个定时器的 CHxA、CHxB 端口都能输出 2 相独立的 PWM 波或一组互补 PWM 波，多个定时器间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。如图 17-14，Timer4，Timer5，Timer6 组合输出 6 相 PWM 波；如图 17-15，Timer4，Timer5，Timer6 组合输出 3 相互补 PWM 波。

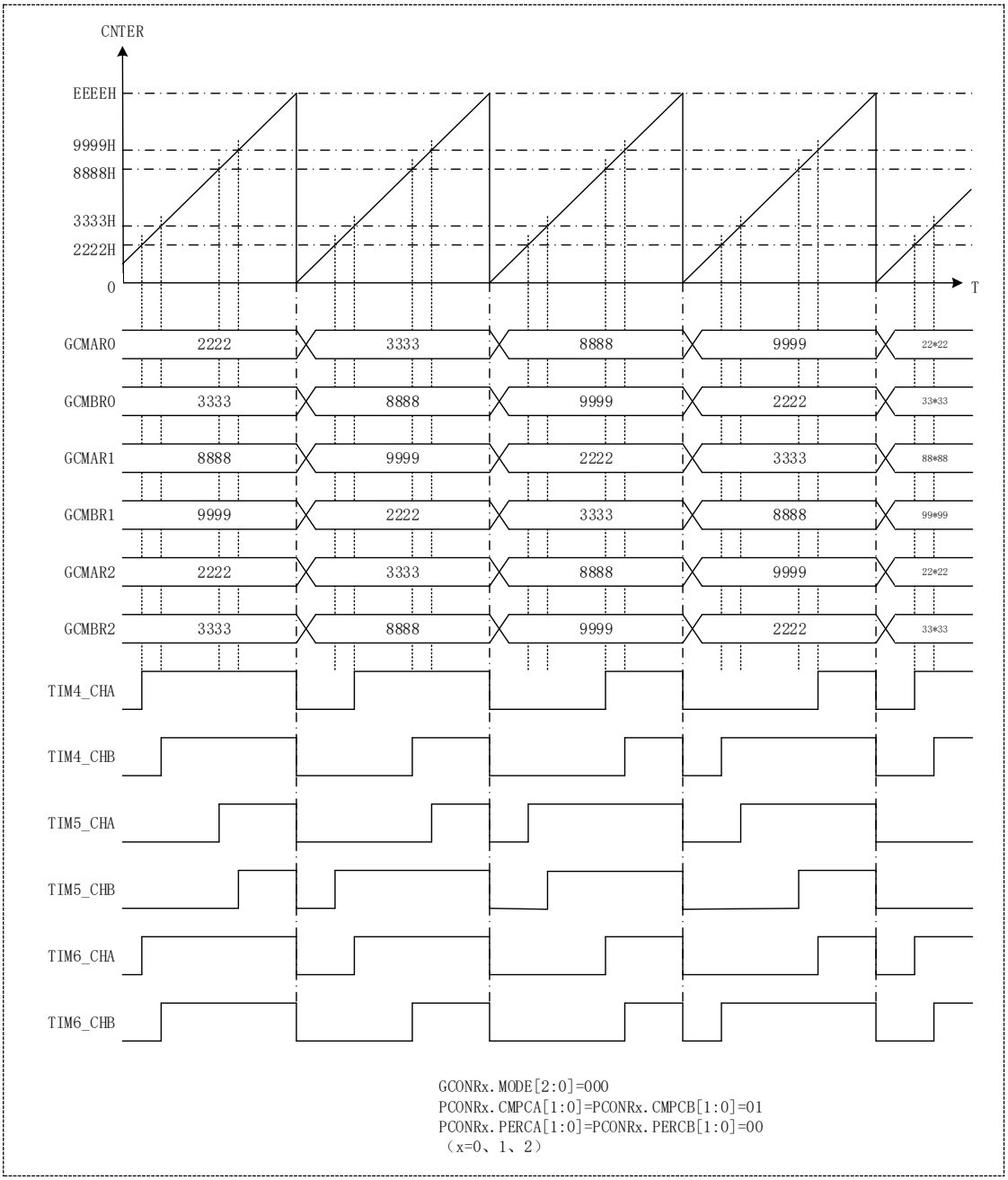


图 17-14　6 相 PWM 波

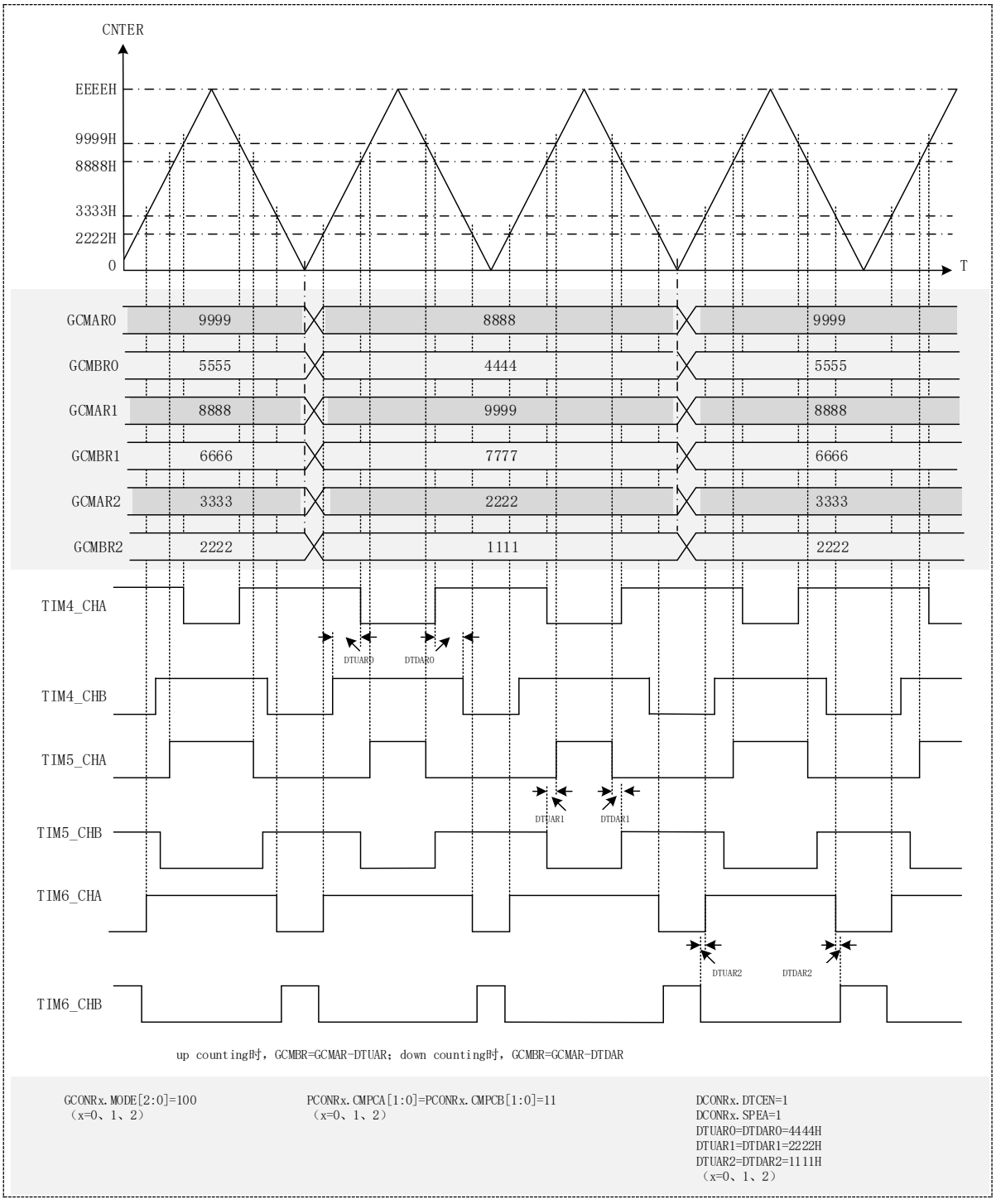


图 17-15　三角波 A 模式时带死区时间三相互补 PWM 波输出

17.2.9 正交编码计数

将 CHxA 输入看作 AIN 输入、CHxB 输入看作 BIN 输入、TIMTRIA-D 中的任意一个输入看作 ZIN 输入，Advanced Timer 就可以实现三路输入的正交编码计数。

一个定时器的 AIN、BIN 单独动作可以实现位置计数模式；两个定时器的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，一个定时器用于位置计数，一个定时器用于公转计数。

公转计数模式时，每两个定时器组合（定时器 4、5 组合，定时器 4 作为位置计数单元，定时器 5 作为公转计数单元）分别实现位置计数和公转计数。

AIN 和 BIN 的计数条件通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 CHxA 和 CHxB 的正交关系实现；ZIN 的输入动作通过设定位置单元的硬件清零事件选择寄存器（HCLRR）实现位置计数单元的位置计数器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转计数单元的公转计数器计数。

17.2.9.1 位置计数模式

正交编码位置模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

基本计数

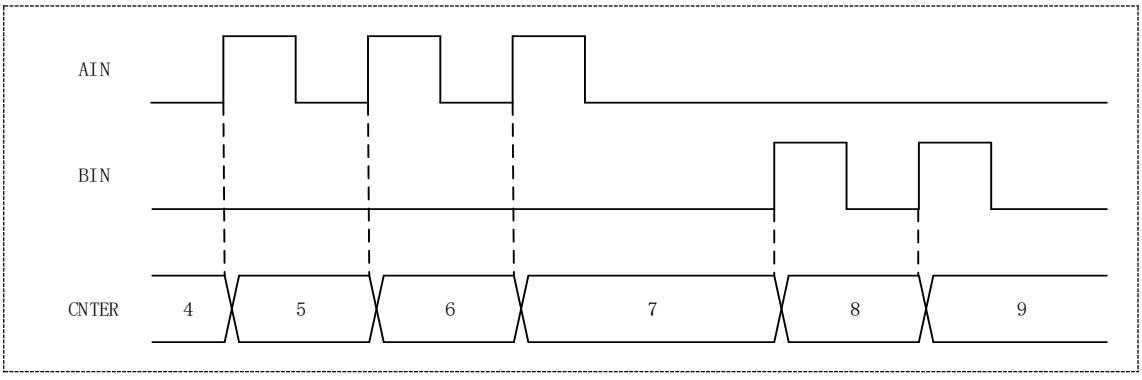


图 17-16 位置模式时基本计数动作

通过设定 HCUPR 与 HCDOR 寄存器，可以灵活的实现各种方式的相位差计数。

相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图图 17-17~图 17-19 所示。

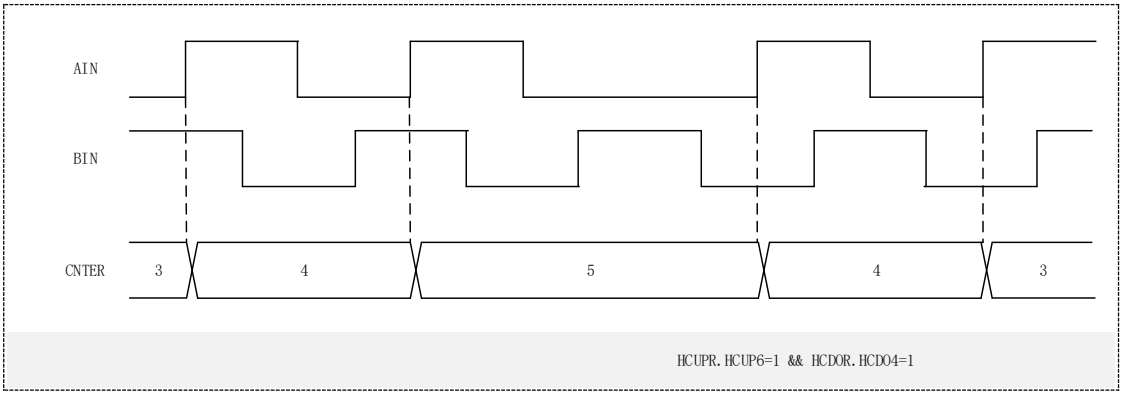


图 17-17 位置模式时相位差计数动作设定(1 倍)

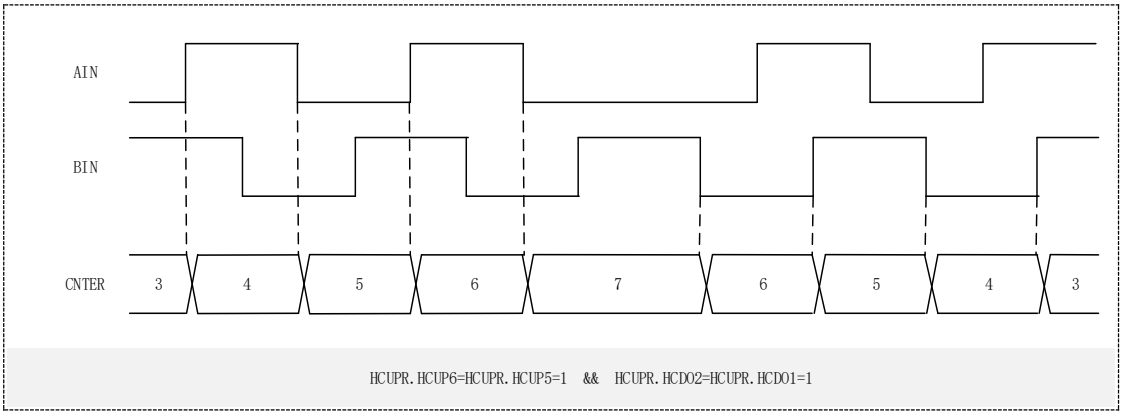


图 17-18 位置模式时相位差计数动作设定(2 倍)

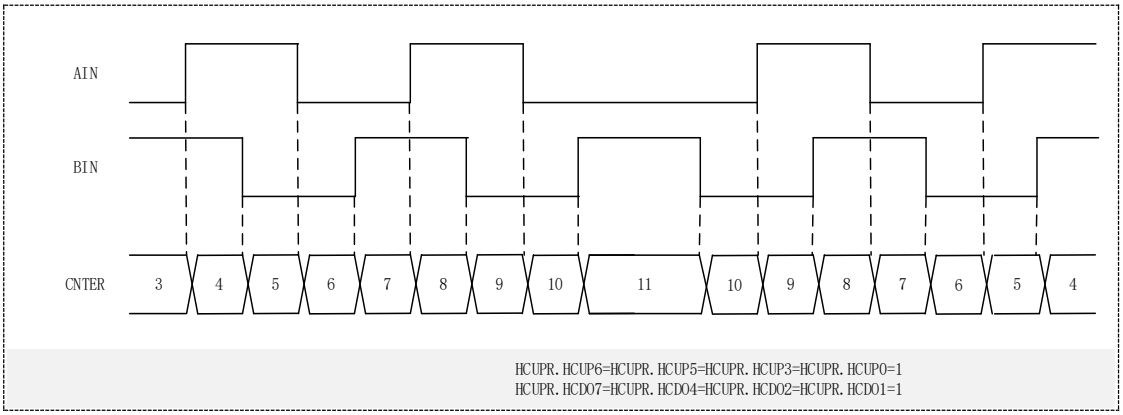


图 17-19 位置模式时相位差计数动作设定(4 倍)

方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如图 17-20 所示。

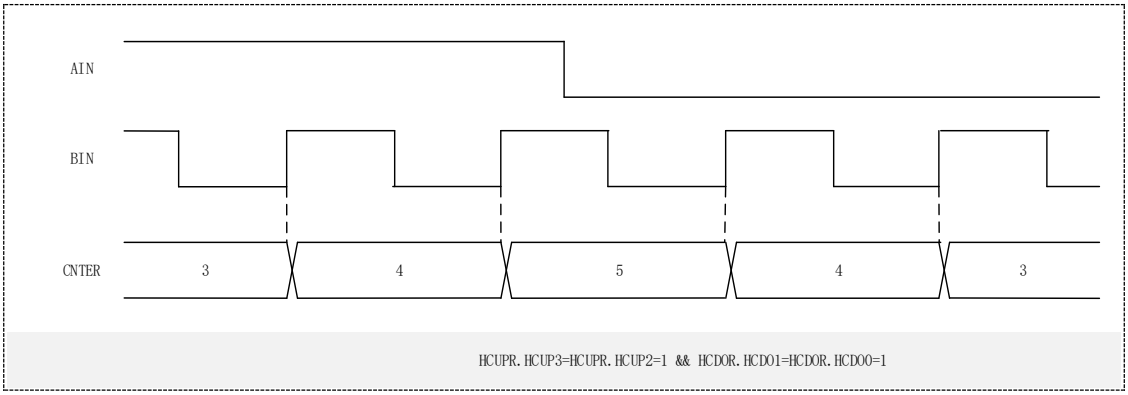


图 17-20 位置模式时方向计数动作

17.2.9.2 公转模式

正交编码公转模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现对公转圈数等的判断。公转模式时根据公转计数器的计数方式，可实现 Z 相计数功能、位置计数器输出计数功能和 Z 相计数与位置计数器输出混合计数功能。即使用两个 Advanced Timer 实现此功能。

Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。

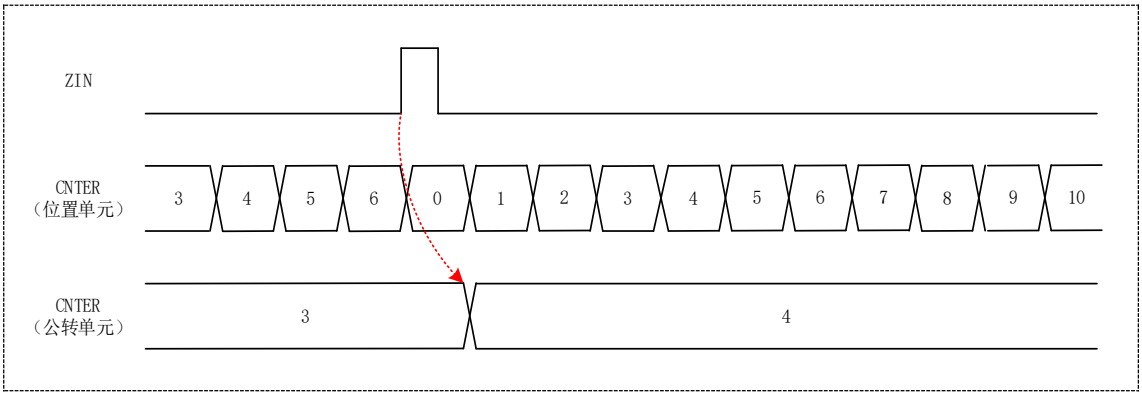


图 17-21 公转模式时 Z 相计数动作

位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的计数器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过 AOS 模块的联动选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加(递减)事件选择寄存器(HCUPR 或 HCDOR)的递加（递减）事件选择 Bit16:Bit19 中的 1 位，同时 AOS 模块设定对应的递加（递减）事件的事件源为位置计数单元的计数溢出事件，具体请参考 AOS 章节。如图 17-22 所示。

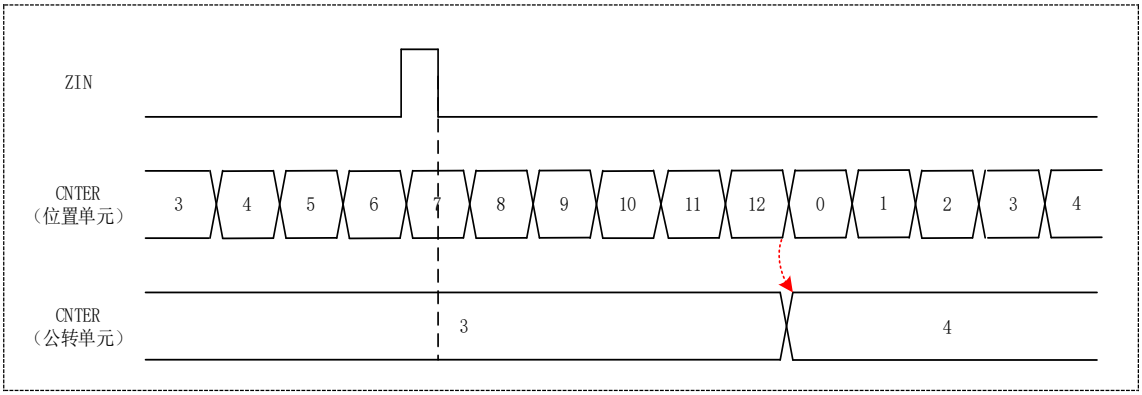


图 17-22　公转模式时位置计数器输出计数动作

混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如图 17-23 所示。

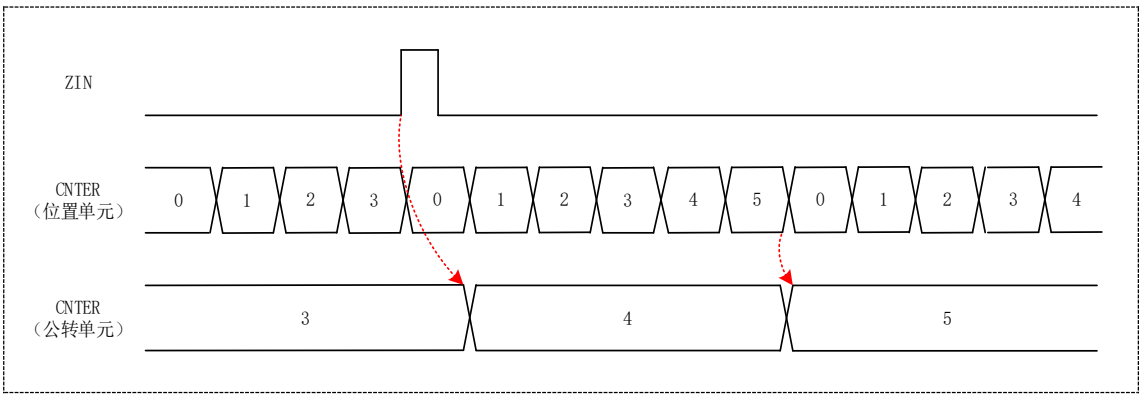


图 17-23　公转模式时 Z 相计数和位置计数器输出混合计数动作

Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置计数器的上溢点或下溢点后的几个周期内（GCONR.ZMSK[0:1]设定），将 ZIN 的有效输入屏蔽，不进行公转计数单元的计数和位置计数单元的清零。

位置计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKPOS 为 1 时，位置计数单元的 Z 相屏蔽功能使能，Z 相屏蔽的周期数由 GCONR.ZMSK 设定；公转计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。

图 17-24 是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不

清零；之后再来的 ZIN 相输入正常动作。

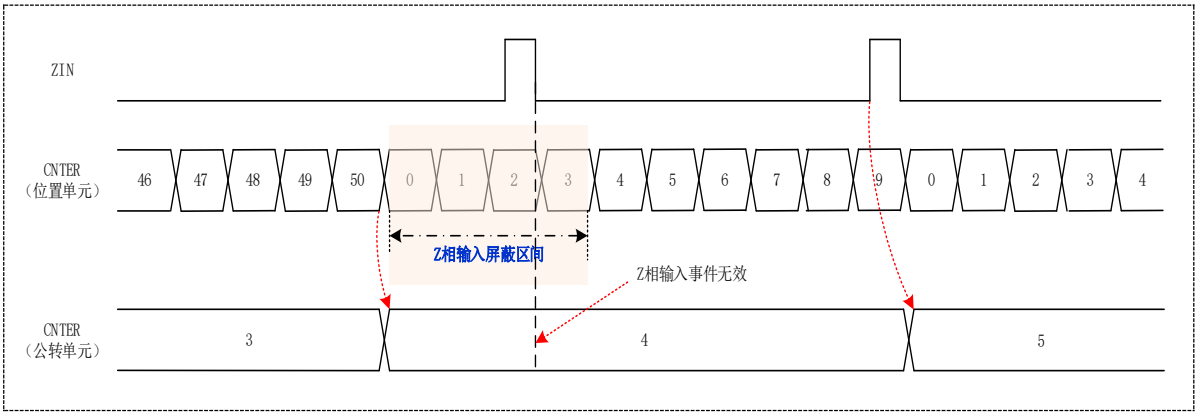


图 17-24 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 17-25 是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维持了 3 个周期），开始向下计数。在位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

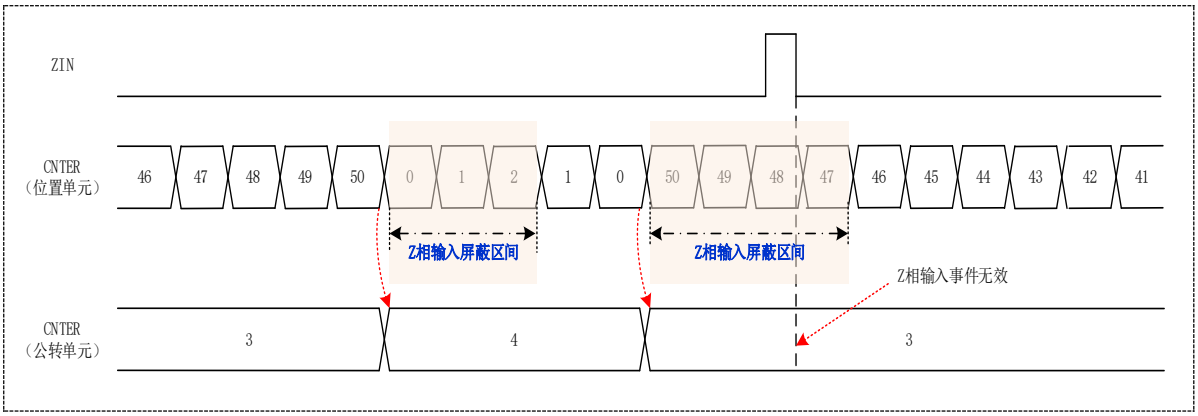


图 17-25 公转计数模式-混合计数 Z 相屏蔽动作例 2

17.2.10 周期间隔响应

Timer4/5/6 的通用比较基准值寄存器（GCMAR~GCMDR），在计数比较匹配时可分别产生专用有效请求信号，送到 AOS 模块中用于和其它模块关联动作。

该请求信号可以每间隔几个周期后产生一次有效的请求信号。通过设定有效周期寄存器（VPERR）的 VPERR.PCNTS 位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和比较基准值寄存器 GCMAR 或 GCMBR 的值相等，也不会输出有效的请求信号。图 17-26 所示是周期间隔有效请求信号的动作例。

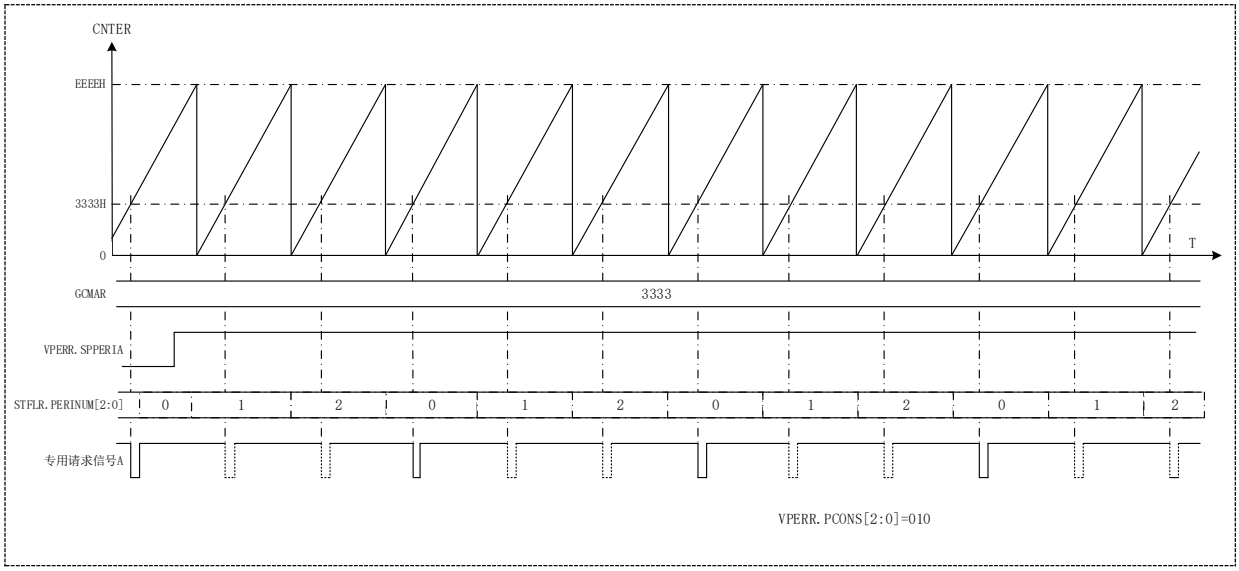


图 17-26 周期间隔有效请求信号动作

17.2.11 保护机制

Advanced Timer 可以对端口的输出状态进行保护控制。

Advanced Timer 有 4 个共用的端口输出无效事件接口，这 4 个接口连接刹车控制模块输出的 4 组刹车事件。每个接口上选通的异常状况事件可从刹车控制设定，当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。

端口在正常输出期间，若监测到从刹车控制过来的刹车事件，则端口的输出状态可变为预先设定好的状态。通用 PWM 输出端口在刹车控制异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（PCONR.DISVALA、PCONR.DISVALB 的设定决定）。

例如，若 PCONR.DISSELA[1:0]=01&PCONR.DISVALA=01 设定时，则在 CHxA

端口正常输出期间，若输出无效条件 1 上产生刹车事件，则 CHxA 端口上输出变为高阻态。

17.2.12 中断说明

Timer4/5/6 各含有 3 类共计 9 个中断。分别是 4 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个计数周期匹配中断、1 个死区时间错误中断。

17.2.12.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR-GCMDR）共计 4 个，可分别与计数值比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMDF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENA~ICONR.INTEND 中相应位为 1 使能中断，则对应的中断请求也会被触发。

在硬件捕获事件选择寄存器（HCPAR、HCPBR）选择的捕获输入有效条件产生时，捕获输入动作发生。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENA 或 ICONR.INTENB 位为 1 使能中断，则对应的中断请求被触发。

17.2.12.2 计数周期匹配中断

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，状态标志寄存器（STFLR）的 STFLR.OVFF 或 STFLR.UDFF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENOVF 位与 ICONR.INTENUDF 位使能中断，则在对应的时间点可触发计数周期匹配中断。

17.2.12.3 死区时间错误中断

将死区时间基准值寄存器（DTUAR、DTDAR）的值加载到通用比较基准值寄存器（GCMBR）中时，若超过周期限制，则会产生死区时间错误，状态标志寄存器（STFLR）的 STFLR.DTEF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENDE 位使能中断，则会在该时刻触发死区时间错误中断。

17.2.13 DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求,A 请求触发源可选择通用比较捕获 A,C,专用比较 A，计数上溢出，B 请求触发源可选择通用比较捕获 B、D,专用比较 B，计数下溢出。

IDREQ	Interrupt Signal of Peripheral
26	TIM4A
27	TIM4B
28	TIM5A
29	TIM5B
30	TIM6A
31	TIM6B

TIMxA 的 触发源可选择比较捕获 A，C，计数上溢出，专用比较 A

TIMxB 的 触发源可选择比较捕获 B，D，计数下溢出，专用比较 B

17.2.14 刹车保护

当可以设定无效条件 0~3，配置 PCONR.DISVALA, PCONR.DISVALB。无效条件有效时硬件自动将端口状态改变为预设状态（高电平，低电平，高阻态，保持正常输出）。

17.2.14.1 端口刹车与软件刹车

端口经过极性选择控制，有效使能后，经过数字滤波，同步，产生端口刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 3。端口刹车标志需要软件清除。

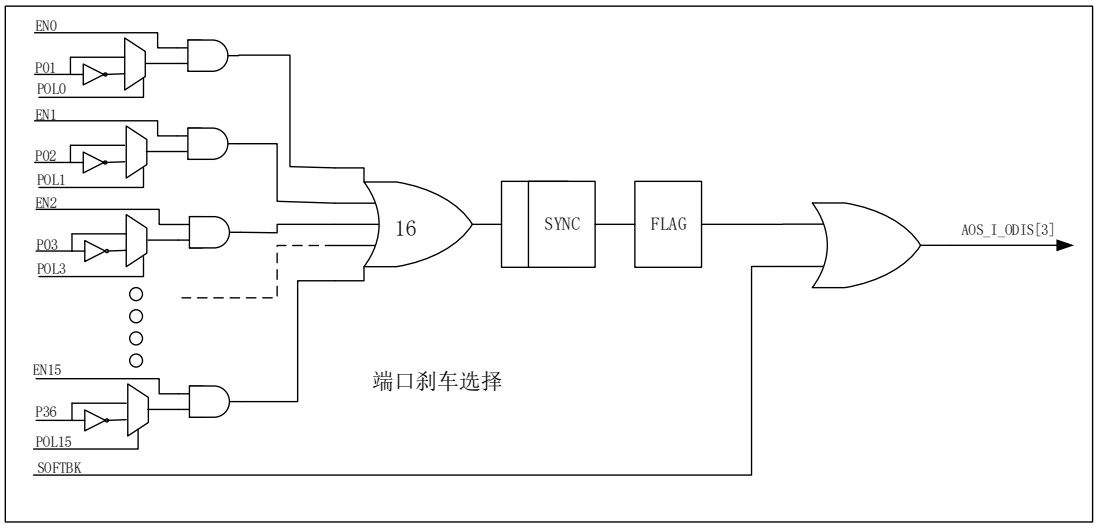


图 17-27 端口刹车与软件刹车示意图

17.2.14.2 低功耗模式自动刹车

系统进入低功耗模式，时钟停止后 PWM 将不能正常工作。低功耗模式作为 Advanced Timer 的无效条件 2 控制 PWM 刹车。

17.2.14.3 输出电平同高同低刹车

输出电平经过电平监测，有效使能后，经过同步，产生同高同低刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 1。同高同低刹车标志需要软件清除。

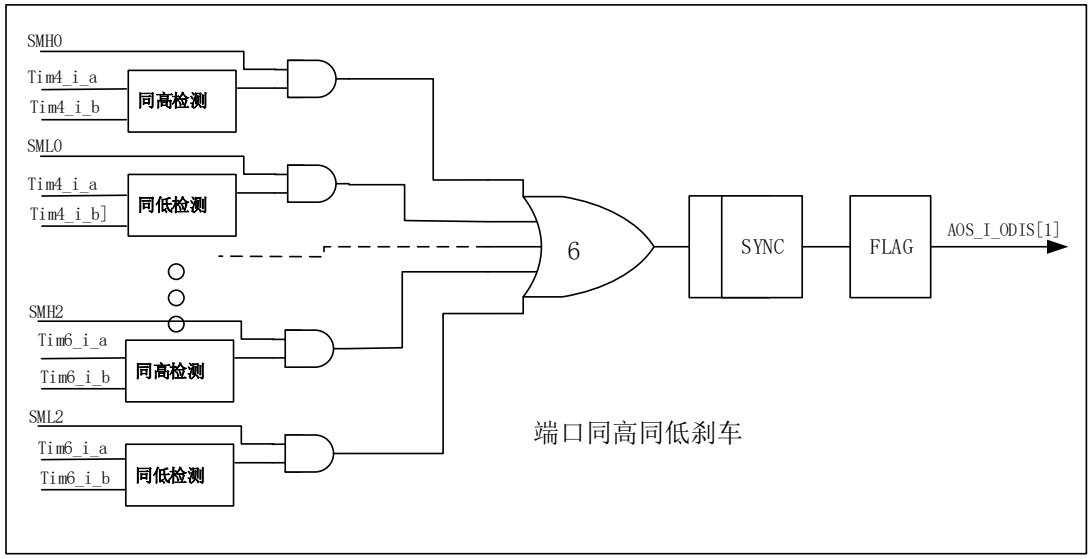


图 17-28 输出同高同低刹车示意图

17.2.14.4 VC 刹车

VC0，VC1，VC2 中断标志经过使能后作为 Advanced Timer 的无效条件 0。

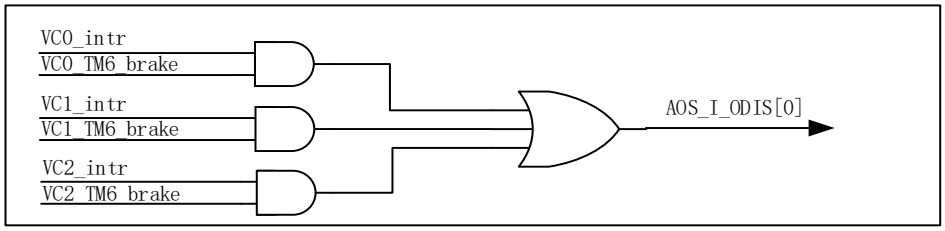


图 17-29 VC 刹车控制示意图

17.2.15 内部互连

17.2.15.1 中断触发输出

由于 Timer4/5/6 的一个中断包含多个中断源。控制触发 ADC 与控制 AOS 的中断信号有单独控制可以选择不同的源，可以选择上溢出、下溢出，4 个比较匹配共 6 个 TIMx 的中断源的任意中断源作为触发条件。

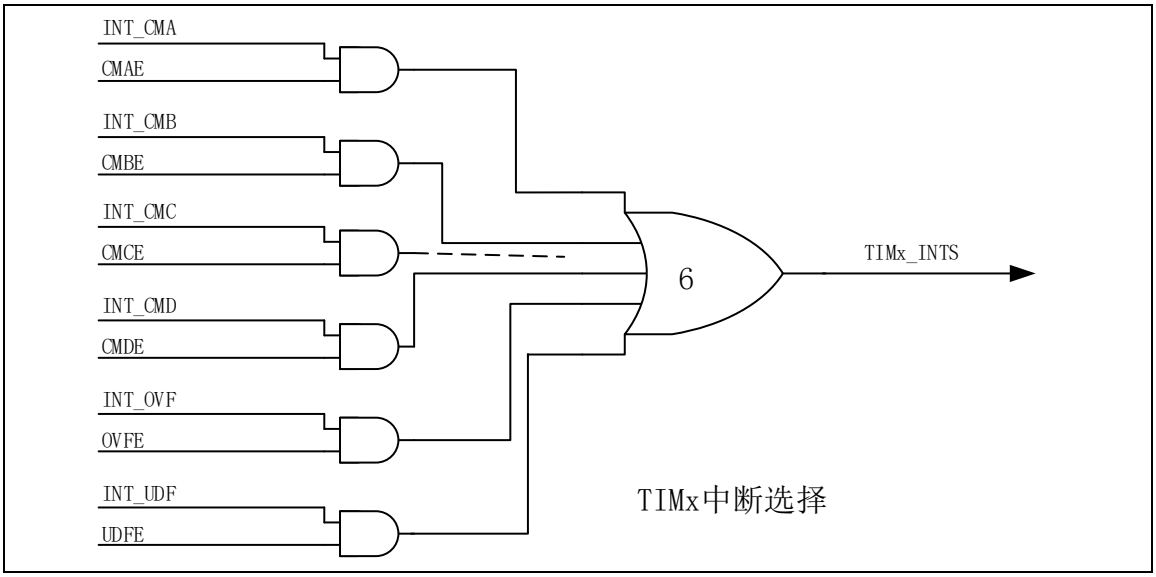


图 17-30 Timer4/5/6 中断选择

17.2.15.2 AOS 触发

AOS 是系统的内部信号，通过选择控制后可以触发 Advanced Timer 的计数器的开始，停止，清零，加 1，减 1 等功能。Advanced Timer 有 4 路 AOS 触发，每路触发可以选择不同模块的中断源。选择后的信号产生单脉冲触发输入到 Advanced Timer，控制 Advanced Timer 的计数器的开始，停止，清零。

Timer4/5/6 内部使用寄存器选择不同的 AOS_I_TRIG 作为自己的触发信号。如可使用 HSTAR 寄存器可以使用中断触发相应定时器的硬件启动。

	AOS_i_trig0	AOS_i_trig1	AOS_i_trig2	AOS_i_trig3
选择控制信号	ITRIG.IAOS0S	ITRIG.IAOS1S	ITRIG.IAOS2S	ITRIG.IAOS3S
0000	TIM0_INT	TIM0_INT	TIM0_INT	TIM0_INT
0001	TIM1_INT	TIM1_INT	TIM1_INT	TIM1_INT
0010	TIM2_INT	TIM2_INT	TIM2_INT	TIM2_INT
0011	-	-	-	-
0100	TIM4_INTS	TIM4_INTS	TIM4_INTS	TIM4_INTS
0101	TIM5_INTS	TIM5_INTS	TIM5_INTS	TIM5_INTS
0110	TIM6_INTS	TIM6_INTS	TIM6_INTS	TIM6_INTS
0111	UART0_INT	UART0_INT	UART0_INT	UART0_INT
1000	UART1_INT	UART1_INT	UART1_INT	UART1_INT
1001	-	-	-	-
1010	VC0_INT	VC0_INT	VC0_INT	VC0_INT
1011	VC1_INT	VC1_INT	VC1_INT	VC1_INT
1100	-	-	-	-
1101	PCA_INT	PCA_INT	PCA_INT	PCA_INT
1110	SPI_INT	SPI_INT	SPI_INT	SPI_INT
1111	ADC_INT	ADC_INT	ADC_INT	ADC_INT

表 17-3 AOS 源选择

17.2.15.3 端口触发 TRIGA-TRIGD

端口触发可以控制 Advanced Timer 的硬件启动、停止、清零、捕获、计数器加减计数等功能，有数字滤波功能可选，端口可以配置为芯片的任意一个端口上。

选择控制信号分别独立控制	TRIGA	TRIGB	TRIGC	TRIGD
0000	PA3	PA3	PA3	PA3
0001	PB3	PB3	PB3	PB3
0010	PC3	PC3	PC3	PC3
0011	PD3	PD3	PD3	PD3
0100	PA7	PA7	PA7	PA7
0101	PB7	PB7	PB7	PB7
0110	PC7	PC7	PC7	PC7
0111	PD7	PD7	PD7	PD7
1000	PA11	PA11	PA11	PA11
1001	PB11	PB11	PB11	PB11
1010	PC11	PC11	PC11	PC11
1011	PD1	PD1	PD1	PD1
1100	PA15	PA15	PA15	PA15
1101	PB15	PB15	PB15	PB15
1110	PC5	PC5	PC5	PC5
1111	PD5	PD5	PD5	PD5

表 17-4 端口触发选择

17.2.15.4 比较输出 VC 与 Advanced Timer 互连

VC 内部可以互连到 Advanced Timer 的捕获输入端，可以对 VC 输出的边沿进行捕获；VC0 与 CHA 连接，VC1 与 CHB 连接；控制在 VC 控制寄存器。

17.3 寄存器描述

CH0 基地址 0x40003000

CH1 基地址 0x40003400

CH2 基地址 0x40003800

寄存器	偏移地址	描述
TIMx_CNTER	0x000	通用计数基准值寄存器
TIMx_PERAR	0x004	通用周期基准值寄存器
TIMx_PERBR	0x008	通用周期基准值缓存寄存器
TIMx_GCMAR	0x010	通用比较A基准值寄存器
TIMx_GCMBR	0x014	通用比较B基准值寄存器
TIMx_GCMCR	0x018	通用比较C基准值寄存器
TIMx_GCMDR	0x01C	通用比较D基准值寄存器
TIMx_SCMAR	0x028	专用比较A基准值寄存器
TIMx_SCMBR	0x02C	专用比较B基准值寄存器
TIMx_DTUAR	0x040	死区时间基准值寄存器
TIMx_DTDAR	0x044	死区时间基准值寄存器
TIMx_GCONR	0x050	通用控制寄存器
TIMx_ICONR	0x054	中断控制寄存器
TIMx_PCONR	0x058	端口控制寄存器
TIMx_BCONR	0x05C	缓存控制寄存器
TIMx_DCONR	0x060	死区控制寄存器
TIMx_FCONR	0x068	滤波控制寄存器
TIMx_VPERR	0x06C	有效周期寄存器
TIMx_STFLR	0x070	状态标志寄存器
TIMx_HSTAR	0x074	硬件启动事件选择寄存器
TIMx_HSTPR	0x078	硬件停止事件选择寄存器
TIMx_HCLRR	0x07C	硬件清零事件选择寄存器
TIMx_HCPAR	0x080	硬件捕获事件选择寄存器
TIMx_HCPBR	0x084	硬件捕获事件选择寄存器
TIMx_HCUPR	0x088	硬件递减事件选择寄存器
TIMx_HCDOR	0x08C	硬件递减事件选择寄存器
TIMx_IFR	0x100	中断标志寄存器
TIMx_ICLR	0x104	中断清除寄存器
TIMx_CR	0x108	展频及中断触发选择寄存器
TIMx_AOSSR	0x110	AOS 选择寄存器，三个通道共用
TIMx_AOSCL	0x114	AOS 刹车标志清除寄存器，三个通道共用
TIMx_PTBKS	0x118	端口刹车控制寄存器，三个通道共用

TIMx_TTRIG	0x11C	端口触发控制寄存器，三个通道共用
TIMx_ITRIG	0x120	AOS 触发控制寄存器，三个通道共用
TIMx_PTBKP	0x124	端口刹车极性控制寄存器，三个通道共用
TIMx_SSTAR	0x3F4	软件同步启动寄存器
TIMx_SSTPR	0x3F8	软件同步停止寄存器
TIMx_SCLRR	0x3FC	软件同步清零寄存器

表 17-5 Advanced Timer 寄存器列表

17.3.1 通用计数基准值寄存器 (TIMx_CNTER)

地址偏移量：0x000

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	CNT[15:0]	当前计数器的计数值

17.3.2 通用周期基准值寄存器（TIMx_PERAR）

地址偏移量：0x004

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERA[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	PERA[15:0]	计数周期值，设定每轮计数的计数周期值

17.3.3 通用周期缓存寄存器（TIMx_PERBR）

地址偏移量：0x008

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERB[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	PERB[15:0]	缓存计数周期值，计数周期的缓存值

17.3.4 通用比较基准值寄存器（TIMx_GCMAR-GCMDR）

地址偏移量：0x0010, 0x0014, 0x0018, 0x001C

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCMA-D [15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	GCMA-D [15:0]	计数比较基准值，比较基准值设定，与计数值相等时匹配信号有效

17.3.5 专用比较基准值寄存器（TIMx_SCMAR-SCMBR）

地址偏移量：0x0028, 0x002C

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCMA-B[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	SCMA-B [15:0]	计数比较基准值，比较基准值设定，与计数值相等时匹配信号有效

17.3.6 死区时间基准值寄存器（TIMx_DTUAR- DTDAR）

地址偏移量：0x040, 0x044

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTUA/DTDA [15:0]															
RW															

位	符号	功能描述
31:16	Reserved	-
15:0	DTUA/DA [15:0]	死区时间值，死区时间设定值

17.3.7 通用控制寄存器（TIMx_GCONR）

地址偏移量：0x050

复位值: 0x00000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ZMSK[1:0]		ZMSK POS	ZMSK REV
												RW		RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DIR	Res.	CKDIV[2:0]			MODE[2:0]			START
							RW		RW			RW			RW

位	标记	功能描述
31:20	Reserved	-
19:18	ZMSK[1:0]	Z相输入屏蔽周期数 正交编码Z相输入被屏蔽的计数周期值 00：Z相输入屏蔽功能无效 01：位置计数上溢后或下溢后的4个计数周期内的Z相输入被屏蔽 10：位置计数上溢后或下溢后的8个计数周期内的Z相输入被屏蔽 11：位置计数上溢后或下溢后的16个计数周期内的Z相输入被屏蔽
17	ZMSKPOS	Z相输入位置计数器选择 0：Z相输入时该定时器作为位置计数器，在屏蔽周期期间内位置计数器清零功能正常动作 1：Z相输入时该定时器作为位置计数器，在屏蔽周期期间内位置计数器清零功能被屏蔽
16	ZMSKREV	Z相输入公转计数器选择 0：Z相输入时该定时器作为公转计数器，在屏蔽周期期间内公转计数器计数功能正常动作 1：Z相输入时该定时器作为公转计数器，在屏蔽周期期间内公转计数器计数功能被屏蔽
15:9	Reserved	-
8	DIR	计数方向 0：递减计数；1：递加计数
7	Reserved	-
6:4	CKDIV[2:0]	计数时钟选择 000：PCLK0　001：PCLK0/2　010：PCLK0/4　011：PCLK0/8 100：PCLK0/16　101：PCLK0/64　110：PCLK0/256　111：PCLK0/1024
3:1	MODE[2:0]	计数模式

		000：锯齿波A模式　100：三角波A模式　101：三角波B模式 请不要设定其它值
0	START	计数器启动　0：计数器关闭；1：计数器启动 *注：该位在软件停止条件或硬件停止条件有效时，会自动变为0*

17.3.8 中断控制寄存器（TIMx_ICONR）

地址偏移量：0x054

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												INTENSBD	INTENSBU	INTENSAD	INTENSAU
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSAMH	INTENSAML	Reserved					INTENDE	INTENUDF	INTENOVF	Reserved		INTEND	INTENC	INTENB	INTENA
RW	RW						RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	INTENSBD	专用向下计数触发ADC使能B
18	INTENSBU	专用向上计数触发ADC使能B
17	INTENSAD	专用向下计数触发ADC使能A
16	INTENSAU	专用向上计数触发ADC使能A
15	INTENSAMH	同高中断使能
14	INTENSAML	同低中断使能
13:9	Reserved	-
8	INTENDE	死区时间错误中断使能 0：死区时间错误时，该中断无效 1：死区时间错误时，该中断使能
7	INTENUDF	下溢中断使能 0：锯齿波时下溢发生或三角波时计数到谷点，该中断无效 1：锯齿波时下溢发生或三角波时计数到谷点，该中断使能
6	INTENOVF	上溢中断使能 0：锯齿波时上溢发生或三角波时计数到峰点，该中断无效 1：锯齿波时上溢发生或三角波时计数到峰点，该中断使能
5:4	Reserved	-
3	INTEND	计数匹配中断使能D 0：GCMDR 寄存器与计数值相等时，该中断无效 1：GCMDR 寄存器与计数值相等时，该中断使能
2	INTENC	计数匹配中断使能C 0：GCMCR 寄存器与计数值相等时，该中断无效

		1：GCMCR 寄存器与计数值相等时，该中断使能
1	INTENB	计数匹配中断使能B 0：GCMBR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：GCMBR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能
0	INTENA	计数匹配中断使能A 0：GCMAR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断无效 1：GCMAR 寄存器与计数值相等时，或者发生捕获输入事件时，该中断使能

17.3.9 端口控制寄存器（TIMx_PCONR）

地址偏移量：0x058

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DISVALB		DISSELB		OUTENB	PERCB		CMPCB		STASTPSB	STPCB	STACB	CAPCB
			RW		RW		RW	RW		RW		RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DISVALA		DISSELA		OUTENA	PERCA		CMPCA		STASTPSA	STPCA	STACA	CAPCA
			RW		RW		RW	RW		RW		RW	RW	RW	RW

位	标记	功能
31:29	Reserved	-
28:27	DISVALB	CHxB 输出状态控制 00：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口正常输出 01：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出高阻态 10：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出低电平 11：强制输出无效条件0~3中被选择的条件成立时，CHxB 端口输出高电平
26:25	DISSELB	强制输出无效条件选择B 00：选择强制输出无效条件0；01：选择强制输出无效条件1 10：选择强制输出无效条件2；11：选择强制输出无效条件3
24	OUTENB	输出使能B 0：Advanced Timer 功能时的 CHxB 端口输出无效 1：Advanced Timer 功能时的 CHxB 端口输出有效
23:22	PERCB	周期值匹配时端口状态设定B 00：计数器计数值与周期值相等时，CHxB 端口输出保持为低电平 01：计数器计数值与周期值相等时，CHxB 端口输出设定为高电平 10：计数器计数值与周期值相等时，CHxB 端口输出设定为先前状态 11：计数器计数值与周期值相等时，CHxB 端口输出设定为反转电平
21:20	CMPCB	比较值匹配时端口状态设定B 00：计数器计数值与 GCMBR 相等时，CHxB 端口输出保持为低电平 01：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为高电平 10：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为先前状态 11：计数器计数值与 GCMBR 相等时，CHxB 端口输出设定为反转电平
19	STASTPSB	计数开始停止端口状态选择B 0：计数开始或停止时，CHxB 端口输出由 STACB、STPCB 决定 1：计数开始或停止时，CHxB 端口输出设定为先前状态 *注：此处的计数开始是指初始计数开始或停止再开始；计数停止是指初始时停*

		止或计数开始后再停止
18	STPCB	计数停止端口状态设定B 0：计数停止时，CHxB 端口输出设定为低电平 1：计数停止时，CHxB 端口输出设定为高电平
17	STACB	计数开始端口状态设定B 0：计数开始时，CHxB 端口输出设定为低电平 1：计数开始时，CHxB 端口输出设定为高电平
16	CAPCB	功能模式选择B 0：比较输出功能；1：捕获输入功能
15:13	Reserved	-
12:11	DISVALA	CHxA 输出状态控制 00：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口正常输出 01：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出高阻态 10：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出低电平 11：强制输出无效条件0~3中被选择的条件成立时，CHxA 端口输出高电平
10:9	DISSELA	强制输出无效条件选择A 00：选择强制输出无效条件0；01：选择强制输出无效条件1 10：选择强制输出无效条件2；11：选择强制输出无效条件3
8	OUTENA	输出使能A 0：Advanced Timer 功能时的 CHxA 端口输出无效 1：Advanced Timer 功能时的 CHxA 端口输出有效
7:6	PERCA	周期值匹配时端口状态设定A 00：计数器计数值与周期值相等时，CHxA 端口输出保持为低电平 01：计数器计数值与周期值相等时，CHxA 端口输出设定为高电平 10：计数器计数值与周期值相等时，CHxA 端口输出设定为先前状态 11：计数器计数值与周期值相等时，CHxA 端口输出设定为反转电平
5:4	CMPCA	比较值匹配时端口状态设定A 00：计数器计数值与 GCMAR 相等时，CHxA 端口输出保持为低电平 01：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为高电平 10：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为先前状态 11：计数器计数值与 GCMAR 相等时，CHxA 端口输出设定为反转电平
3	STASTPSA	计数开始停止端口状态选择A 0：计数开始或停止时，CHxA 端口输出由 STACA、STPCA 决定 1：计数开始或停止时，CHxA 端口输出设定为先前状态 *注：此处的计数开始是指初始计数开始或停止再开始；计数停止是指初始时停止或计数开始后再停止*
2	STPCA	计数停止端口状态设定A 0：计数停止时，CHxA 端口输出设定为低电平； 1：计数停止时，CHxA 端口输出设定为高电平
1	STACA	计数开始端口状态设定A 0：计数开始时，CHxA 端口输出设定为低电平 1：计数开始时，CHxA 端口输出设定为高电平

0	CAPCA	功能模式选择A 0：比较输出功能；1：捕获输入功能

17.3.10 缓存控制寄存器（TIMx_BCONR）

地址偏移量：0x05C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BENP	Reserved					BENB	Res.	BENA
							RW						RW		RW

位	标记	功能
31:9	Reserved	-
8	BENP	周期值缓存传送 0：缓存传送无效 1：缓存传送使能（PERBR->PERAR）
7:3	Reserved	-
2	BENB	通用比较值缓存传送B 0：缓存传送无效 1：缓存传送使能 比较输出功能时：（GCMDR->GCMBR）；捕获输入功能时：（GCMBR->GCMDR）
1	Reserved	-
0	BENA	通用比较值缓存传送A 0：缓存传送无效 1：缓存传送使能 比较输出功能时：（GCMCR->GCMAR）；捕获输入功能时：（GCMAR->GCMCR）

17.3.11 死区控制寄存器（TIMx_DCONR）

地址偏移量：0x060

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							SEPA	Reserved							DTCEN
							RW								RW

位	标记	功能
31:9	Reserved	-
8	SEPA	分离设定 0：DTUAR 和 DTDAR 分别设定 1：DTDAR 的值和 DTUAR 的值自动相等
7:1	Reserved	-
0	DTCEN	死区功能 0：死区功能无效 1：死区功能有效

17.3.12 滤波控制寄存器（TIMx_FCONR）

地址偏移量：0x068

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	NOFICKTD		NOFIENTD	Res.	NOFICKTC		NOFIENTC	Res.	NOFICKTB		NOFIENTB	Res.	NOFICKTA		NOFIENTA
	RW		RW		RW		RW		RW		RW		RW		RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									NOFICKGB		NOFIENGB	Res.	NOFICKGA		NOFIENGA
									RW		RW		RW		RW

位	标记	功能
31	Reserved	-
30:29	NOFICKTD	TRID 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
28	NOFIENTD	TRID 端口捕获输入滤波使能，0无效；1使能
27	Reserved	-
26:25	NOFICKTC	TRIC 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
24	NOFIENTC	TRIC 端口捕获输入滤波使能，0无效；1使能
23	Reserved	-
22:21	NOFICKTB	TRIB 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
20	NOFIENTB	TRIB 端口捕获输入滤波使能，0无效；1使能
19	Reserved	-
18:17	NOFICKTA	TRIA 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
16	NOFIENTA	TRIA 端口捕获输入滤波使能，0无效；1使能
15:7	Reserved	-
6:5	NOFICKGB	CHxIB 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
4	NOFIENGB	CHxIB 端口捕获输入滤波使能，0无效；1使能
3	Reserved	-
2:1	NOFICKGA	CHxIA 端口滤波采样基准时钟选择 00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64
0	NOFIENGA	CHxIA 端口捕获输入滤波使能，0无效；1使能

注意：

– TRIGA-D 滤波设置只有在 TIM4 中设置有效，在 Timer5/6 设置无效。

17.3.13 有效周期寄存器（TIMx_VPERR）

地址偏移量：0x06C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PCNTS			PCNTE	
											RW			RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												GEPERID	GEPERIC	GEPERIB	GEPERIA
												RW	RW	RW	RW

位	标记	功能
31:21	Reserved	-
20:18	PCNTS	有效周期选择 000：有效周期选择功能无效 001：每隔1个周期有效一次 010：每隔2个周期有效一次 011：每隔3个周期有效一次 100：每隔4个周期有效一次 101：每隔5个周期有效一次 110：每隔6个周期有效一次 111：每隔7个周期有效一次
17:16	PCNTE	有效周期计数条件选择 00：有效周期选择功能无效 01：锯齿波计数上、下溢点或三角波波谷做为计数条件 10：锯齿波计数上、下溢点或三角波波峰做为计数条件 11：锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件
15:4	Reserved	-
3	GEPERID	通用信号有效周期选择D 0：有效周期选择功能无效；1：有效周期选择功能使能
2	GEPERIC	通用信号有效周期选择C 0：有效周期选择功能无效；1：有效周期选择功能使能
1	GEPERIB	通用信号有效周期选择B 0：有效周期选择功能无效；1：有效周期选择功能使能
0	GEPERIA	通用信号有效周期选择A 0：有效周期选择功能无效；1：有效周期选择功能使能

17.3.14 状态标志寄存器（TIMx_STFLR）

地址偏移量：0x070

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIRF	Reserved							VPERNUM			Reserved				
R								R							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	Reserved		CMDF	CMCF	CMBF	CMAF
			RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31	DIRF	计数方向 0：递减计数 1：递加计数
30:24	Reserved	-
23:21	VPERNUM	周期次数 有效周期选择功能使能时，计数后的周期次数
20:13	Reserved	-
12	CMSBDF	向下计数专用比较基准值匹配B
11	CMSBUF	向上计数专用比较基准值匹配B
10	CMSADF	向下计数专用比较基准值匹配A
9	CMSAUF	向上计数专用比较基准值匹配A
8	DTEF	死区时间错误 0：未发生死区时间错误；1：发生死区时间错误
7	UDFF	下溢匹配 0：未发生锯齿波下溢或三角波计数到谷点 1：发生锯齿波下溢或三角波计数到谷点
6	OVFF	上溢匹配 0：未发生锯齿波上溢或三角波计数到峰点 1：发生锯齿波上溢或三角波计数到峰点
5:4	Reserved	-
3	CMDF	计数匹配D 0：GCMDR 寄存器的值与计数值不相等；1：GCMDR 寄存器的值与计数值相等
2	CMCF	计数匹配C 0：GCMCR寄存器的值与计数值不相等；1：GCMCR 寄存器的值与计数值相等

1	CMBF	计数匹配B 0：GCMBR 寄存器的值与计数值不相等，且未发生 CHxB 捕获完成动作 1：GCMBR 寄存器的值与计数值相等，或发生 CHxB 捕获完成动作
0	CMAF	计数匹配A 0：GCMAR 寄存器的值与计数值不相等，且未发生 CHxA 捕获完成动作 1：GCMAR 寄存器的值与计数值相等，或发生 CHxA 捕获完成动作

17.3.15 硬件启动事件选择寄存器（TIMx_HSTAR）

地址偏移量：0x074

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STARTS	Reserved														
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTA15	HSTA14	HSTA13	HSTA12	HSTA11	HSTA10	HSTA9	HSTA8	HSTA7	HSTA6	HSTA5	HSTA4	HSTA3	HSTA2	HSTA1	HSTA0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31	STARTS	硬件启动使能 0：硬件启动无效 1：硬件启动有效 *注：硬件启动有效时，SSTAR 的设定无效*
30:16	Reserved	-
15	HSTA15	硬件启动条件15：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
14	HSTA14	硬件启动条件14：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
13	HSTA13	硬件启动条件13：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
12	HSTA12	硬件启动条件12：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
11	HSTA11	硬件启动条件11：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
10	HSTA10	硬件启动条件10：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
9	HSTA9	硬件启动条件9：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效

8	HSTA8	硬件启动条件8：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
7	HSTA7	硬件启动条件7：CHxB 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
6	HSTA6	硬件启动条件6：CHxB 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
5	HSTA5	硬件启动条件5：CHxA 端口上采样到下降沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
4	HSTA4	硬件启动条件4：CHxA 端口上采样到上升沿 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
3	HSTA3	硬件启动条件3：从 AOS 来的事件触发3有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
2	HSTA2	硬件启动条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
1	HSTA1	硬件启动条件1：从 AOS 来的事件触发1有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效
0	HSTA0	硬件启动条件0：从 AOS 来的事件触发0有效 0：条件匹配时，硬件启动无效 1：条件匹配时，硬件启动有效

17.3.16 硬件停止事件选择寄存器（TIMx_HSTPR）

地址偏移量：0x078

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STOPS	Reserved														
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTP15	HSTA14	HSTP13	HSTP12	HSTP11	HSTP10	HSTP9	HSTP8	HSTP7	HSTP6	HSTP5	HSTP4	HSTP3	HSTP2	HSTP1	HSTP0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31	STOPS	硬件停止使能 0：硬件停止无效 1：硬件停止有效 *注：硬件停止有效时，软件停止的设定无效*
30:16	Reserved	-
15	HSTP15	硬件停止条件15：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
14	HSTP14	硬件停止条件14：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
13	HSTP13	硬件停止条件13：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
12	HSTP12	硬件停止条件12：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
11	HSTP11	硬件停止条件11：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
10	HSTP10	硬件停止条件10：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
9	HSTP9	硬件停止条件9：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效

8	HSTP8	硬件停止条件8：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
7	HSTP7	硬件停止条件7：CHxB 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
6	HSTP6	硬件停止条件6：CHxB 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
5	HSTP5	硬件停止条件5：CHxA 端口上采样到下降沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
4	HSTP4	硬件停止条件4：CHxA 端口上采样到上升沿 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
3	HSTP3	硬件停止条件3：从 AOS 来的事件触发3有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
2	HSTP2	硬件停止条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
1	HSTP1	硬件停止条件1：从 AOS 来的事件触发1有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效
0	HSTP0	硬件停止条件0：从 AOS 来的事件触发0有效 0：条件匹配时，硬件停止无效 1：条件匹配时，硬件停止有效

17.3.17 硬件清零事件选择寄存器（TIMx_HCELR）

地址偏移量：0x07C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLEARS	Reserved														
RW															

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

位	标记	功能
31	CLEARS	硬件清零使能 0：硬件清零无效 1：硬件清零有效 *注：硬件清零有效时，软件清零的设定无效*
30:16	Reserved	-
15	HCEL15	硬件清零条件15：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
14	HCEL14	硬件清零条件14：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
13	HCEL13	硬件清零条件13：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
12	HCEL12	硬件清零条件12：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
11	HCEL11	硬件清零条件11：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
10	HCEL10	硬件清零条件10：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
9	HCEL9	硬件清零条件9：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效

8	HCEL8	硬件清零条件8：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
7	HCEL7	硬件清零条件7：CHxB 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
6	HCEL6	硬件清零条件6：CHxB 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
5	HCEL5	硬件清零条件5：CHxA 端口上采样到下降沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
4	HCEL4	硬件清零条件4：CHxA 端口上采样到上升沿 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
3	HCEL3	硬件清零条件3：从 AOS 来的事件触发3有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
2	HCEL2	硬件清零条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
1	HCEL1	硬件清零条件1：从 AOS 来的事件触发1有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效
0	HCEL0	硬件清零条件0：从 AOS 来的事件触发0有效 0：条件匹配时，硬件清零无效 1：条件匹配时，硬件清零有效

17.3.18 硬件捕获 A 事件选择寄存器（TIMx_HCPAR）

地址偏移量：0x080

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCPA15	HCPA14	HCPA13	HCPA12	HCPA11	HCPA10	HCPA9	HCPA8	HCPA7	HCPA6	HCPA5	HCPA4	HCPA3	HCPA2	HCPA1	HCPA0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	HCPA15	硬件捕获A条件15：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
14	HCPA14	硬件捕获A条件14：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
13	HCPA13	硬件捕获A条件13：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
12	HCPA12	硬件捕获A条件12：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
11	HCPA11	硬件捕获A条件11：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
10	HCPA10	硬件捕获A条件10：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
9	HCPA9	硬件捕获A条件9：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
8	HCPA8	硬件捕获A条件8：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
7	HCPA7	硬件捕获A条件7：CHxB 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效

		1：条件匹配时，硬件捕获A有效
6	HCPA6	硬件捕获A条件6：CHxB 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
5	HCPA5	硬件捕获A条件5：CHxA 端口上采样到下降沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
4	HCPA4	硬件捕获A条件4：CHxA 端口上采样到上升沿 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
3	HCPA3	硬件捕获A条件3：从 AOS 来的事件触发3有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
2	HCPA2	硬件捕获A条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
1	HCPA1	硬件捕获A条件1：从 AOS 来的事件触发1有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效
0	HCPA0	硬件捕获A条件0：从 AOS 来的事件触发0有效 0：条件匹配时，硬件捕获A无效 1：条件匹配时，硬件捕获A有效

17.3.19 硬件捕获 B 事件选择寄存器（TIMx_HCPBR）

地址偏移量：0x084

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCPB15	HCPB14	HCPB13	HCPB12	HCPB11	HCPB10	HCPB9	HCPB8	HCPB7	HCPB6	HCPB5	HCPB4	HCPB3	HCPB2	HCPB1	HCPB0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	HCPB15	硬件捕获B条件15：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
14	HCPB14	硬件捕获B条件14：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
13	HCPB13	硬件捕获B条件13：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
12	HCPB12	硬件捕获B条件12：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
11	HCPB11	硬件捕获B条件11：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
10	HCPB10	硬件捕获B条件10：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
9	HCPB9	硬件捕获B条件9：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
8	HCPB8	硬件捕获B条件8：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
7	HCPB7	硬件捕获B条件7：CHxB 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效

		1：条件匹配时，硬件捕获B有效
6	HCPB6	硬件捕获B条件6：CHxB 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
5	HCPB5	硬件捕获B条件5：CHxA 端口上采样到下降沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
4	HCPB4	硬件捕获B条件4：CHxA 端口上采样到上升沿 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
3	HCPB3	硬件捕获B条件3：从 AOS 来的事件触发3有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
2	HCPB2	硬件捕获B条件2：从 AOS 来的事件触发2有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
1	HCPB1	硬件捕获B条件1：从 AOS 来的事件触发1有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效
0	HCPB0	硬件捕获B条件0：从 AOS 来的事件触发0有效 0：条件匹配时，硬件捕获B无效 1：条件匹配时，硬件捕获B有效

17.3.20 硬件递加事件选择寄存器（TIMx_HCUPR）

地址偏移量：0x088

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCUP19	HCUP18	HCUP17	HCUP16
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCUP15	HCUP14	HCUP13	HCUP12	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	HCUP19	硬件递加条件：从 AOS 来的事件触发3有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
18	HCUP18	硬件递加条件：从 AOS 来的事件触发2有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
17	HCUP17	硬件递加条件：从 AOS 来的事件触发1有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
16	HCUP16	硬件递加条件：从 AOS 来的事件触发0有效 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
15	HCUP15	硬件递加条件：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
14	HCUP14	硬件递加条件：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
13	HCUP13	硬件递加条件：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
12	HCUP12	硬件递加条件：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效

11	HCUP11	硬件递加条件：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
10	HCUP10	硬件递加条件：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
9	HCUP9	硬件递加条件：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
8	HCUP8	硬件递加条件：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
7	HCUP7	硬件递加条件：CHxB 端口为高电平时，CHxA 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
6	HCUP6	硬件递加条件：CHxB 端口为高电平时，CHxA 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
5	HCUP5	硬件递加条件：CHxB 端口为低电平时，CHxA 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
4	HCUP4	硬件递加条件：CHxB 端口为低电平时，CHxA 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
3	HCUP3	硬件递加条件：CHxA 端口为高电平时，CHxB 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
2	HCUP2	硬件递加条件：CHxA 端口为高电平时，CHxB 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
1	HCUP1	硬件递加条件：CHxA 端口为低电平时，CHxB 端口上采样到下降沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效
0	HCUP0	硬件递加条件：CHxA 端口为低电平时，CHxB 端口上采样到上升沿 0：条件匹配时，硬件递加无效 1：条件匹配时，硬件递加有效

17.3.21 硬件递减事件选择寄存器（TIMx_HCDOR）

地址偏移量：0x08C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCDO19	HCDO18	HCDO17	HCDO16
												RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCDO15	HCDO14	HCDO13	HCDO12	HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:20	Reserved	-
19	HCDO19	硬件递减条件：从 AOS 来的事件触发3有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
18	HCDO18	硬件递减条件：从 AOS 来的事件触发2有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
17	HCDO17	硬件递减条件：从 AOS 来的事件触发1有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
16	HCDO16	硬件递减条件：从 AOS 来的事件触发0有效 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
15	HCDO15	硬件递减条件：TIMTRID 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
14	HCDO14	硬件递减条件：TIMTRID 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
13	HCDO13	硬件递减条件：TIMTRIC 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
12	HCDO12	硬件递减条件：TIMTRIC 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效

11	HCDO11	硬件递减条件：TIMTRIB 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
10	HCDO10	硬件递减条件：TIMTRIB 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
9	HCDO9	硬件递减条件：TIMTRIA 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
8	HCDO8	硬件递减条件：TIMTRIA 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
7	HCDO7	硬件递减条件：CHxB 端口为高电平时，CHxA 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
6	HCDO6	硬件递减条件：CHxB 端口为高电平时，CHxA 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
5	HCDO5	硬件递减条件：CHxB 端口为低电平时，CHxA 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
4	HCDO4	硬件递减条件：CHxB 端口为低电平时，CHxA 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
3	HCDO3	硬件递减条件：CHxA 端口为高电平时，CHxB 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
2	HCDO2	硬件递减条件：CHxA 端口为高电平时，CHxB 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
1	HCDO1	硬件递减条件：CHxA 端口为低电平时，CHxB 端口上采样到下降沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效
0	HCDO0	硬件递减条件：CHxA 端口为低电平时，CHxB 端口上采样到上升沿 0：条件匹配时，硬件递减无效 1：条件匹配时，硬件递减有效

17.3.22 软件同步启动寄存器（TIMx_SSTAR）

地址偏移量：0x3F4

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SSTA2	SSTA1	SSTA0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SSTA2	Timer6 软件启动 0：软件启动无效 1：软件启动使能
1	SSTA1	Timer5 软件启动 0：软件启动无效 1：软件启动使能
0	SSTA0	Timer4 软件启动 0：软件启动无效 1：软件启动使能

17.3.23 软件同步停止寄存器（TIMx_SSTPR）

地址偏移量：0x3F8

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SSTP2	SSTP1	SSTP0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SSTP2	Timer6 软件停止 0：软件停止无效 1：软件停止使能
1	SSTP1	Timer5 软件停止 0：软件停止无效 1：软件停止使能
0	SSTP0	Timer4 软件停止 0：软件停止无效 1：软件停止使能

17.3.24 软件同步清零寄存器（TIMx_SCLRR）

地址偏移量：0x3FC

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SCLR2	SCLR1	SCLR0
													RW	RW	RW

位	标记	功能
31:3	Reserved	
2	SCLR2	Timer6 软件清零 0：软件清零无效 1：软件清零使能
1	SCLR1	Timer5 软件清零 0：软件清零无效 1：软件清零使能
0	SCLR0	Timer4 软件清零 0：软件清零无效 1：软件清零使能

17.3.25 中断标志寄存器（TIMx_IFR）

地址偏移量：0x100

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMHF	SAMLF	Reserved					DTEF	UDFF	OVFF	Reserved		CMDF	CMCF	CMBF	CMAF
RO	RO						RO	RO	RO			RO	RO	RO	RO

位	标记	功能
31:16	Reserved	-
15	SAMHF	CHxA/B 端口高状态中断标志 0：CHxA 和 CHxB 端口上没有同时出现高电平 1：CHxA 和 CHxB 端口上同时出现高电平
14	SAMLF	CHxA/B 端口低状态中断标志 0：CHxA 和 CHxB 端口上没有同时出现低电平 1：CHxA 和 CHxB 端口上同时出现低电平
13:9	Reserved	-
8	DTEF	死区时间错误中断标志 0：未发生死区时间错误；1：发生死区时间错误
7	UDFF	下溢匹配中断标志 0：未发生锯齿波下溢或三角波计数到谷点 1：发生锯齿波下溢或三角波计数到谷点
6	OVFF	上溢匹配中断标志 0：未发生锯齿波上溢或三角波计数到峰点 1：发生锯齿波上溢或三角波计数到峰点
5:4	Reserved	-
3	CMDF	计数匹配D中断标志 0：GCMDR 寄存器的值与计数值不相等；1：GCMDR 寄存器的值与计数值相等
2	CMCF	计数匹配C中断标志 0：GCMCR 寄存器的值与计数值不相等；1：GCMCR 寄存器的值与计数值相等
1	CMBF	计数匹配B中断标志 0：GCMBR 寄存器的值与计数值不相等，且未发生 CHxB 捕获完成动作 1：GCMBR 寄存器的值与计数值相等，或发生 CHxB 捕获完成动作
0	CMAF	计数匹配A中断标志

		0：GCMAR 寄存器的值与计数值不相等，且未发生 CHxA 捕获完成动作 1：GCMAR 寄存器的值与计数值相等，或发生 CHxA 捕获完成动作

17.3.26 中断标志清除寄存器（TIMx_ICLR）

地址偏移量：0x104

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMHC	SAMLC	Reserved					DTEC	UDFC	OVFC	Reserved		CMDC	CMCC	CMBC	CMAC
R1W0	R1W0						R1W0	R1W0	R1W0			R1W0	R1W0	R1W0	R1W0

位	标记	功能
31:16	Reserved	-
15	SAMHC	CHxA/B 端口高状态中断标志清除，写1无效，写0清除对应中断
14	SAMLC	CHxA/B 端口低状态中断标志清除，写1无效，写0清除对应中断
13:9	Reserved	-
8	DTEC	死区时间错误中断标志清除，写1无效，写0清除对应中断
7	UDFC	下溢匹配中断标志清除，写1无效，写0清除对应中断
6	OVFC	上溢匹配中断标志清除，写1无效，写0清除对应中断
5:4	Reserved	-
3	CMDC	计数匹配D中断标志清除，写1无效，写0清除对应中断
2	CMCC	计数匹配C中断标志清除，写1无效，写0清除对应中断
1	CMBC	计数匹配B中断标志清除，写1无效，写0清除对应中断
0	CMAC	计数匹配A中断标志清除，写1无效，写0清除对应中断

17.3.27 展频及中断触发选择（TIMx_CR）

地址偏移量：0x108

复位值: 0x0000 0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									Dma_s_cmb	Dma_s_cma	Dma_g_udf	Dma_g_ovf	Reserved		Dma_g_cmd
									RW	RW	RW	RW			RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dma_g_cmc	Dma_g_cmb	Dma_g_cma	CMSBE	CMSAE	DITENS	DITENB	DITNA	UDFE	OVFE	Reserved		CMDE	CMCE	CMBE	CMAE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW

位	标记	功能
31:23	Reserved	-
22	Dma_s_cmb	专用比较DMA使能
21	Dma_s_cma	专用比较DMA使能
20	Dma_g_udf	下溢出DMA使能
19	Dma_g_ovf	上溢出DMA使能
18:17	Reserved	
16	Dma_g_cmd	通用比较DMA使能
15	Dma_g_cmc	通用比较DMA使能
14	Dma_g_cmb	通用比较DMA使能
13	Dma_g_cma	通用比较DMA使能
12	CMSBE	专用比较基准值匹配B使能触发ADC
11	CMSAE	专用比较基准值匹配A使能触发ADC
10	DITENS	PWM 展频计数选择 0: 选择下溢出，1：选择上溢出
9	DITENB	PWM 通道B展频使能 0：使能无效，1：使能有效，每个周期改变 PWM 的输出延时
9	DITENA	PWM 通道A展频使能 0：使能无效，1：使能有效，每个周期改变 PWM 的输出延时
7	UDFE	下溢匹配使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
6	OVFE	上溢匹配使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
5:4	Reserved	-

3	CMDE	计数匹配D使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
2	CMCE	计数匹配C使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
1	CMBE	计数匹配B使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg
0	CMAE	计数匹配A使能触发 ADC 0：使能无效，1：使能有效，这个匹配可以控制 ADC/AOS_i_tirg

17.3.28 AOS 选择控制寄存器（TIMx_AOSSR）

地址偏移量：0x110

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMH2	SMH1	SMH0	SML2	SML1	SML0	SOFTBK	Reserved		BFILTEN	BFILTS		FSAME	FBRAKE
		RW	RW	RW	RW	RW	RW	RW			RW	RW		R	R

位	标记	功能
31:14	Reserved	-
13	SMH2	通道2同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
12	SMH1	通道1同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
11	SMH0	通道0同高选择 0：选择无效，1：选择有效，出现同高时 AOS_i_odis[1]
10	SML2	通道2同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
9	SML1	通道1同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
8	SML0	通道0同低选择 0：选择无效，1：选择有效，出现同低时 AOS_i_odis[1]
7	SOFTBK	软件刹车：写1实现软件刹车
13	Reserved	-
4	BFILTEN	端口刹车滤波使能
3:2	BFILTS	端口刹车滤波时钟选择
1	FSAME	同高同低刹车标志，只读
0	FBRAKE	端口刹车标志，只读

17.3.29 AOS 选择控制寄存器标志清除（TIMx_AOSCL）

地址偏移量：0x114

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSAME	FBRAKE
														RO	RO

位	标记	功能
31:2	Reserved	-
1	FSAME	同高同低刹车标志清除，写0清除，写1无效，读恒为1
0	FBRAKE	端口刹车标志清除，写0清除，写1无效，读恒为1

17.3.30 端口刹车控制寄存器（TIMx_PTBKS）

地址偏移量：0x118

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	EN15	PD5 刹车端口使能：1 选择，0无效
14	EN14	PC5 刹车端口使能：1 选择，0无效
13	EN13	PB15 刹车端口使能：1 选择，0无效
12	EN12	PA15 刹车端口使能：1 选择，0无效
11	EN11	PD1 刹车端口使能：1 选择，0无效
10	EN10	PC11 刹车端口使能：1 选择，0无效
9	EN9	PB11 刹车端口使能：1 选择，0无效
8	EN8	PA11 刹车端口使能：1 选择，0无效
7	EN7	PD7 刹车端口使能：1 选择，0无效
6	EN6	PC7 刹车端口使能：1 选择，0无效
5	EN5	PB7 刹车端口使能：1 选择，0无效
4	EN4	PA7 刹车端口使能：1 选择，0无效
3	EN3	PD3 刹车端口使能：1 选择，0无效
2	EN2	PC3 刹车端口使能：1 选择，0无效
1	EN1	PB3 刹车端口使能：1 选择，0无效
0	EN0	PA3 刹车端口使能：1 选择，0无效

17.3.31 端口触发控制寄存器（TIMx_TTRIG）

地址偏移量：0x11C

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGDS				TRIGCS				TRIGBS				TRIGAS			
RW				RW				RW				RW			

位	标记	功能
31:16	Reserved	-
15:12	TRIGDS	TIMx 触发D端口选择
11:8	TRIGCS	TIMx 触发C端口选择
7:4	TRIGBS	TIMx 触发B端口选择
3:0	TRIGAS	TIMx 触发A端口选择

控制信号与端口选择如下

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
PA3	PB3	PC3	PD3	PA7	PB7	PC7	PD7	PA11	PB11	PC11	PD1	PA15	PB15	PC15	PD5

17.3.32 AOS 触发控制寄存器（TIMx_ITRIG）

地址偏移量：0x120

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAOS3S				IAOS2S				IAOS1S				IAOS0S			
RW				RW				RW				RW			

位	标记	功能
31:16	Reserved	-
15:12	IAOS3S	TIMx AOS3 触发源选择
11:8	IAOS2S	TIMx AOS2 触发源选择
7:4	IAOS1S	TIMx AOS1 触发源选择
3:0	IAOS0S	TIMx AOS0 触发源选择

控制信号（IAOSxS）与中断源选择如下 (x=0,1,2,3)

0000	0001	0010	0011	0100	0101	0110	0111
TIM0_INT	TIM1_INT	TIM2_INT	-	TIM4_INTS	TIM5_INTS	TIM6_INTS	UART0_INT
1000	1001	1010	1011	1100	1101	1110	1111
UART1_INT	-	VC0_INT	VC1_INT	-	PCA_INT	SPI_INT	ADC_INT

17.3.33 端口刹车极性控制寄存器（TIMx_PTBKP）

地址偏移量：0x124

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器，任意一个定时器其更改后，在另外两个定时器的值会同时更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL15	POL14	POL13	POL12	POL11	POL10	POL9	POL8	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能
31:16	Reserved	-
15	POL15	PD5 刹车端口极性选择：1 低电平有效，0 高电平有效
14	POL14	PC5 刹车端口极性选择：1 低电平有效，0 高电平有效
13	POL13	PB15 刹车端口极性选择：1 低电平有效，0 高电平有效
12	POL12	PA15 刹车端口极性选择：1 低电平有效，0 高电平有效
11	POL11	PD1 刹车端口极性选择：1 低电平有效，0 高电平有效
10	POL10	PC11 刹车端口极性选择：1 低电平有效，0 高电平有效
9	POL9	PB11 刹车端口极性选择：1 低电平有效，0 高电平有效
8	POL8	PA11 刹车端口极性选择：1 低电平有效，0 高电平有效
7	POL7	PD7 刹车端口极性选择：1 低电平有效，0 高电平有效
6	POL6	PC7 刹车端口极性选择：1 低电平有效，0 高电平有效
5	POL5	PB7 刹车端口极性选择：1 低电平有效，0 高电平有效
4	POL4	PA7 刹车端口极性选择：1 低电平有效，0 高电平有效
3	POL3	PD3 刹车端口极性选择：1 低电平有效，0 高电平有效
2	POL2	PC3 刹车端口极性选择：1 低电平有效，0 高电平有效
1	POL1	PB3 刹车端口极性选择：1 低电平有效，0 高电平有效
0	POL0	PA3 刹车端口极性选择：1 低电平有效，0 高电平有效

18 看门狗定时器（WDT）

18.1 WDT 简介

WDT 可用来检测和解决由软件错误引起的故障。当 WDT 计数器达到设定的溢出时间后，会触发中断或产生系统复位。WDT 由专用的 10KHz 片内振荡器驱动。

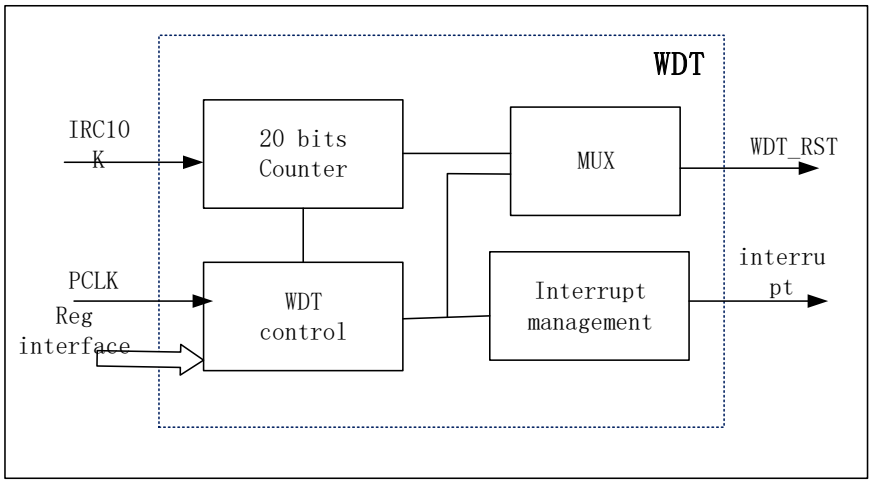


图 18-1 WDT 整体框图

18.2 WDT 功能描述

- 20Bit 自由运行的递增计数器，溢出时间可配置为 1.6ms – 50s。
- 溢出后的动作可配置为中断或复位。
- WDT 时钟由独立的 RC 振荡器提供，可在 Sleep 和 DeepSleep 模式下工作。
- WDTCON 寄存器只有在 WDT 未被启动时才能修改，以防止启动后无意之间修改 WDT 的配置。

18.2.1 WDT 溢出后产生中断

在本模式下，WDT 将按所设定的时间周期性地产生中断。在中断服务程序中需要清除 WDT 中断标志。

配置方法如下所示：

Step1：配置 WDT_CON. WOV，选择 WDT 计时溢出时间。

Step2：设置 WDT_CON. WINT_EN 为 1，选择 WDT 溢出后产生中断。

Step3：使能 NVIC 中断向量表中的 WDT 中断。

Step4：向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step5:在中断服务程序中向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清除中断标志。

18.2.2 WDT 溢出后产生复位

在本模式下，WDT 计数器溢出后会产生 Reset 信号，该信号会复位 MCU。用户程序需要在 WDT 溢出前清零 WDT 计数器，从而避免产生 WDT 复位。

配置方法如下所示：

Step1：配置 WDT_CON. WOV，选择 WDT 计数器溢出时间。

Step2：设置 WDT_CON. WINT_EN 为 0，选择 WDT 溢出后产生复位。

Step3：向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step4：在 WDT 溢出前向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清零 WDT 计数器。

注意：

– 由于 WDT 振荡器是低精度的 RC 振荡器，强烈建议在 WDT 计数器计数值到达溢

出值的一半之前对 WDT 进行清零。

18.3 WDT 寄存器描述

基地址 0X40000F00

寄存器	偏移地址	描述
WDT_RST	0X080	WDT清除控制寄存器
WDT_CON	0X084	WDT控制寄存器

表 18-1 WDT 寄存器列表

18.3.1 WDT 清除控制寄存器（WDT_RST）

偏移地址：0x080

复位值: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved	WDTRST							
	WO							

位	符号	描述
31:8	Reserved	保留位，读为0
7:0	WDTRST	看门狗启动/清零控制 当看门狗未启动时，向该寄存器依次写入0x1E、0xE1，启动WDT定时器。 当看门狗已启动时，向该寄存器依次写入0x1E、0xE1，清零WDT定时器及中断标志。

18.3.2 WDT_CON 寄存器

偏移地址：0x084

复位值: 0x0000 000F

注：该寄存器只有在 WDT 未运行时才可以写入。

31:16	15:8	7	6	5	4	3	2	1	0
Reserved	**WCNTL**	**WDTINT**	Res.	WINT_EN	WDTR	WOV			
	RO	RO		RW	RO	RW			

位	符号	描述
31:16	Reserved	保留位，读为0
15:8	WCNTL	WDT计数器低8位
7	WDTINT	WDT中断标志 1：已发生WDT中断，向WDT_RST寄存器依次写入0x1E、0xE1以清除该中断标志。 0：未发生WDT中断 。
5	WINT_EN	WDT溢出后的动作配置 1：WDT溢出后产生中断。 0：WDT溢出后产生复位。
4	WDTR	WDT运行标志 1：WDT正在运行 0：WDT停止
3:0	WOV[3:0]	WDT计时溢出时间配置 0000：1.6ms　　1000：500ms 0001：3.2ms　　1001：820ms 0010：6.4ms　　1010：1.64s 0011：13ms　　1011：3.28s 0100：26ms　　1100：6.55s 0101：51ms　　1101：13.1s 0110：102ms　　1110：26.2s 0111：205ms　　1111：52.4s

19 通用同步异步收发器（UART）

19.1 简介

通用同步异步收发器（UART）能够灵活地与外部设备进行全双工数据交换，它支持同步单向通信、单线半双工通信以及多处理器通信。常用于短距离、低速的串行通信中。UART 通过可编程波特率发生器提供了多种波特率。UART 支持多种工作模式。

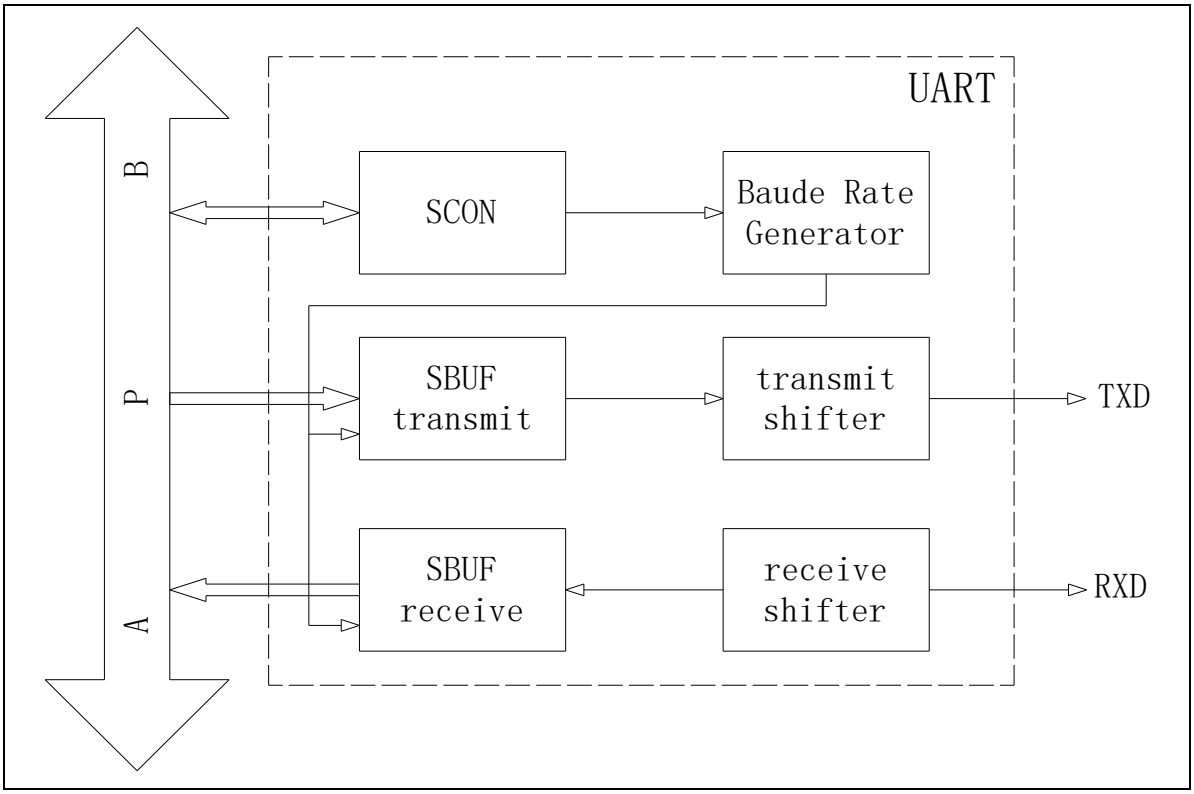


图 19-1 结构框图

19.2 主要特性

通用 UART 模块支持以下基本功能：

- 全双工传输、半双工传输、单线半双工传输
- 可编程串行通信功能
  - 两种字符长度：8 比特、 9 比特
  - 三种校验方式：无检验、奇校验、偶校验
  - 三种停止长度：1 比特、 2 比特、 1.5 比特
- 16 比特波特率发生器
- 多机通讯
- 硬件地址识别
- 硬件流控
- DMAC 硬件传输握手

19.3 功能描述

19.3.1 工作模式

UART 支持多种工作模式：同步半双工模式、异步全双工模式、单线半双工模式。通过 UARTx_SCON.SM 和 UARTx_SCON. HDSEL 搭配，即可配置出所需要的各种工作模式。

19.3.1.1 Mode0~Mode3 功能对比

配置 UARTx_SCON.SM 可选择不同的传输模式：Mode0~Mode3。这四种工作模式的主要功能对比如下表所示：

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8bit	Data(8bit)	$BaudRate = \frac{f_{PCLK}}{12}$
Mode1	异步模式 全双工	10~11bit	Start (1bit) + Data(8bit) + Stop(1~2bit)	$BaudRate = \frac{f_{PCLK}}{OVER * SCNT}$
Mode2	异步模式 全双工	11~12bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1~2bit)	$BaudRate = \frac{f_{PCLK}}{OVER}$
Mode3	异步模式 全双工	11~12bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1~2bit)	$BaudRate = \frac{f_{PCLK}}{OVER * SCNT}$

表 19-1 Mode0/1/2/3 数据结构

注：

- Mode0 只能作为主机发送 UART 同步移位时钟，不可以作为从机接收外部输入的 UART 同步移位时钟。
- f_{PCLK}代表当前 PCLK 的频率。
- OVER 的定义详见 UARTx_SCON。
- SCNT 的定义详见 UARTx_SCNT。
- B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

应用场景	UARTx_SCON. ADRDET	UARTx_SCON. B8CONT[1:0]	B8 数据含义
奇偶校验	--	01/10	接收时，B8 是所收到的 8-Bit 数据的奇偶校验位； 发送时，B8 是待发送的 8-Bit 数据的奇偶校验位；
多机通讯	1	--	B8=1，代表当前是地址帧； B8=0，代表当前是数据帧；
其他	0	00/11	接收/发送数的第 8 比特

表 19-2 B8 数据含义

注意：

- 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制；

19.3.1.2 Mode0 数据收发说明

发送数据时，清除 UARTx_SCON.REN 位，并将数据写入 UARTx_SBUF 寄存器中。此时，发送数据从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

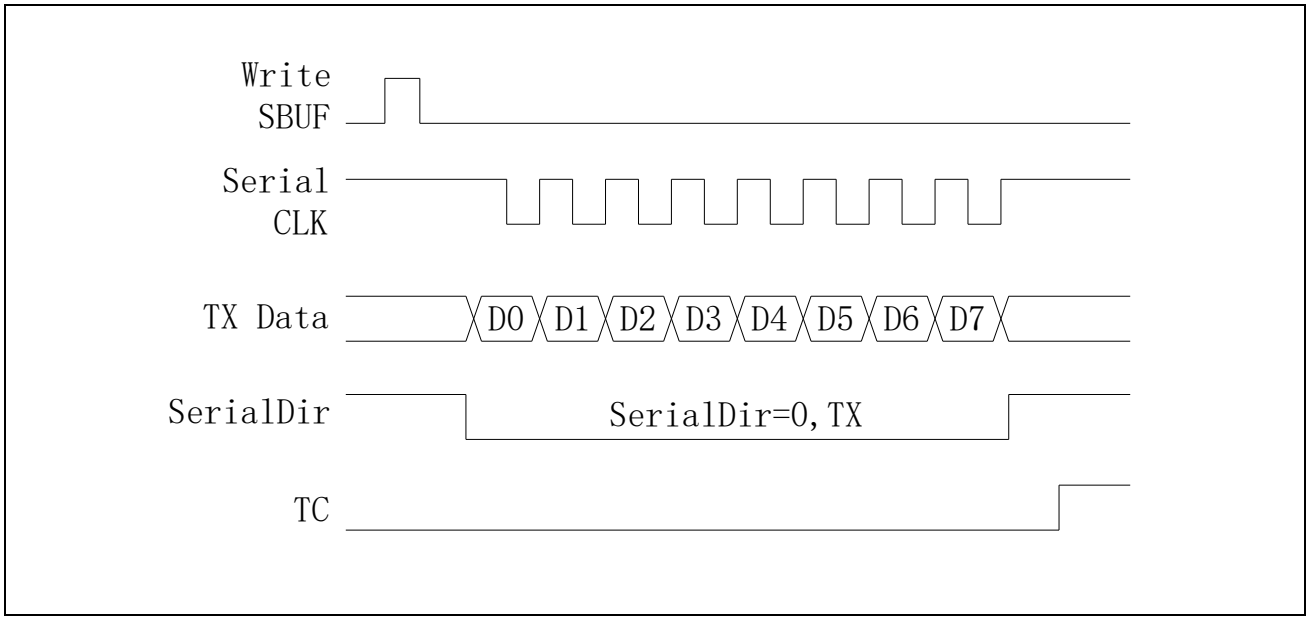


图 19-2 Mode0 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清零。当接收结束，数据可从 UARTx_SBUF 寄存器读取。此时，接收数据从 RXD 输入（低位在先，高位在后），同步移位时钟从 TXD 输出。

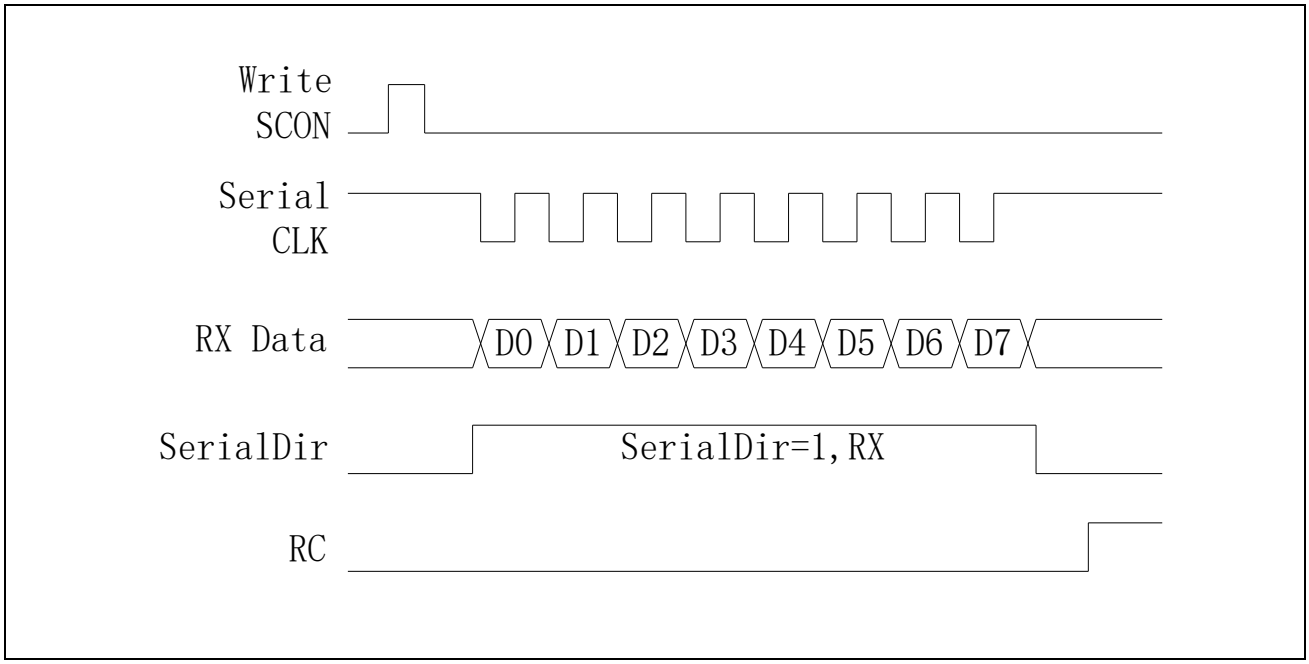


图 19-3 Mode0 接收数据

19.3.1.3 Mode1 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

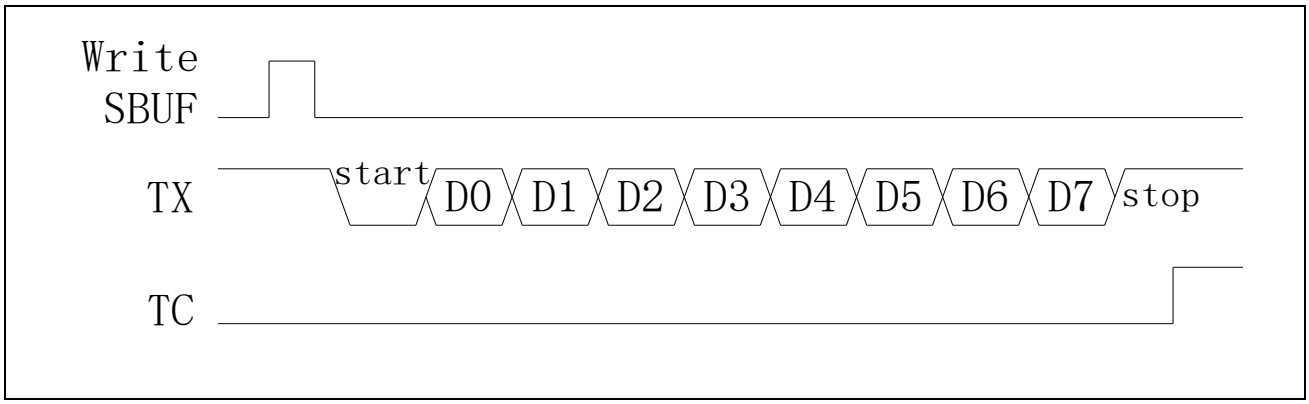


图 19-4 Mode1 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

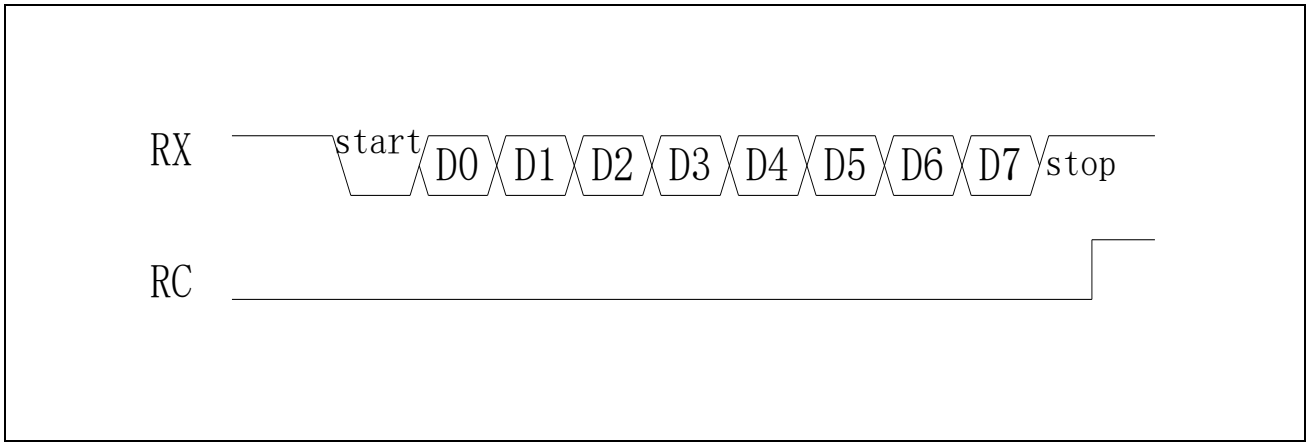


图 19-5 Mode1 接收数据

19.3.1.4 Mode2 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

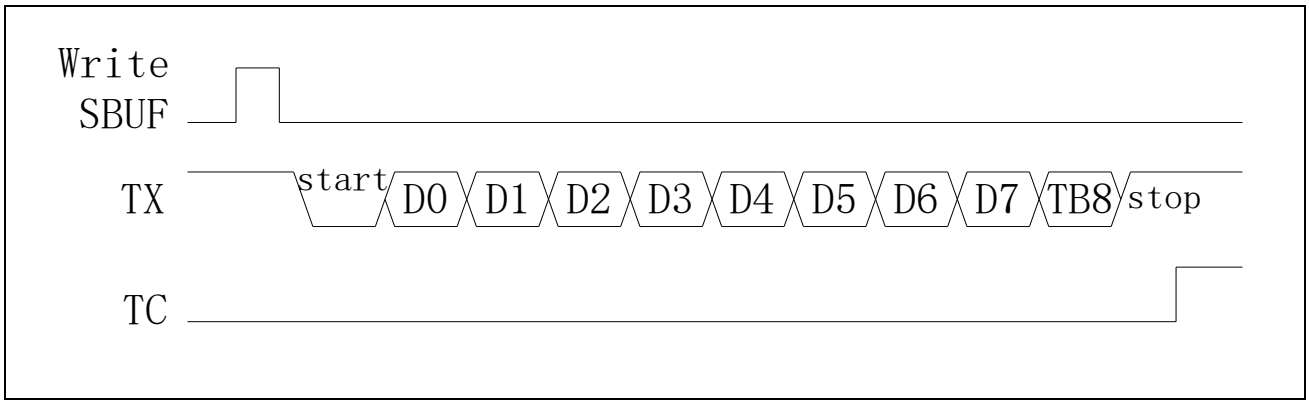


图 19-6 Mode2 发送数据

接收数据时，需将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上的数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读取。

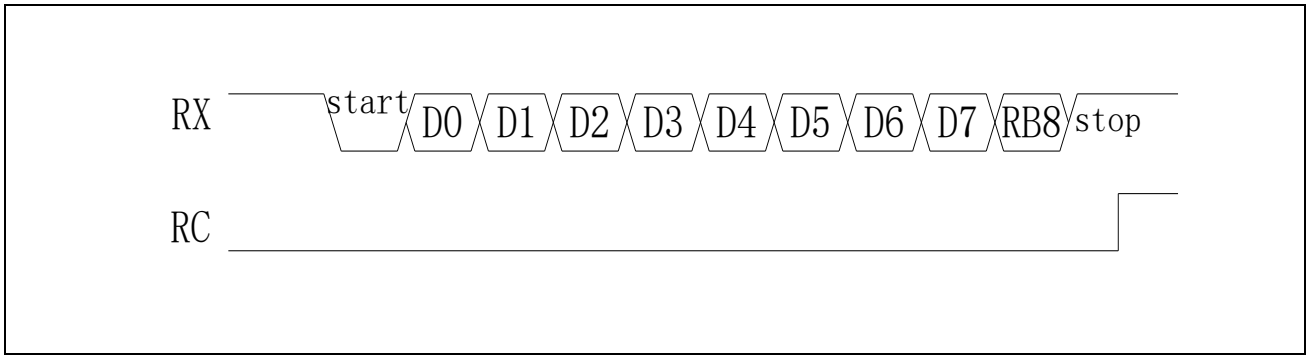


图 19-7 Mode2 接收数据

19.3.1.5 Mode3 数据收发说明

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

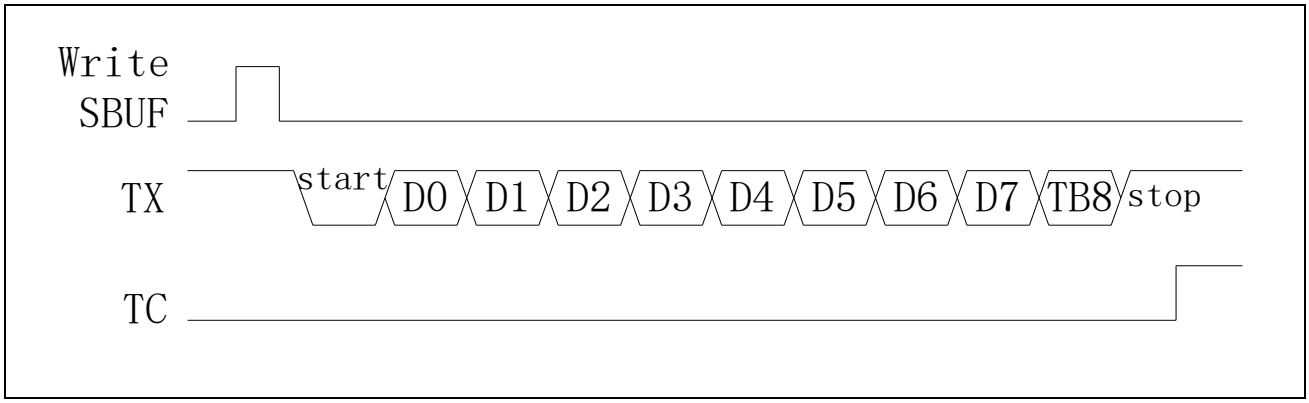


图 19-8 Mode3 发送数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

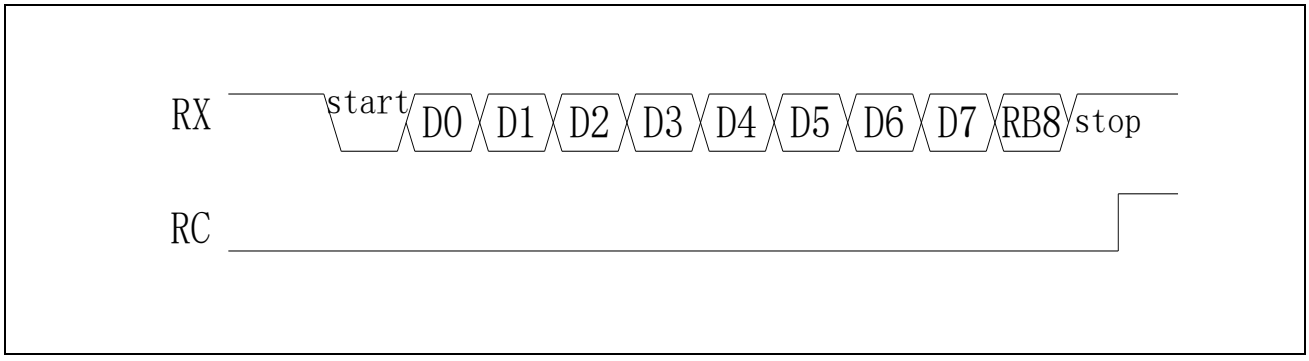


图 19-9 Mode3 接收数据

19.3.1.6 单线半双工数据收发说明

当 UARTx_SCON.HDSEL=1’b1，可进入单线半双工传输模式：

- TX 与 RX 信号线在模块内部相连。
- RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。
- 当发送缓存为空时，TX 信号始终为输入（接收状态），一旦向发送缓存填入一个数据，TX 信号变为输出（发送状态），当发送完成后，发送缓存变为空，TX 信号又回到输入。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

注意：

1. 即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态），接收过程将被打断，并得到一个错误的接收数据。
2. 单线半双工模式与 Mode0 模式的异同：

   同：两者都是用一根信号分时复用地完成数据收发，

   异：Mode0 为同步传输，有同步时钟，而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1-2-3 的波特率公式，仅仅是收发功能复用到了 TX 信号上。

19.3.2 波特率生成

Mode0~Mode3 生成波特率的公式不尽相同，详见下方所示。

Mode0 波特率生成工式：$BaudRate = \frac{f_{PCLK}}{12}$

Mode1 波特率生成工式：$BaudRate = \frac{f_{PCLK}}{OVER*SCNT}$

Mode2 波特率生成工式：$BaudRate = \frac{f_{PCLK}}{OVER}$

Mode3 波特率生成工式：$BaudRate = \frac{f_{PCLK}}{OVER*SCNT}$

注：

- f_{PCLK}代表当前 PCLK 的 频率。
- OVER 的定义详见 UARTx_SCON。
- SCNT 的定义详见 UARTx_SCNT

19.3.2.1 Mode1/Mode3 波特率设置示例

波特率	PCLK = 4 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%
128000	4	125000.00	-2.34%	2	125000.00	-2.34%
250000	2	250000.00	0.00%	1	250000.00	0.00%

表 19-3 PCLK=4MHz 波特率计算表

波特率	PCLK = 8 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%
57600	17	58823.53	2.12%	9	55555.56	-3.55%
76800	13	76923.08	0.16%	7	71428.57	-6.99%
115200	9	111111.11	-3.55%	4	125000.00	8.51%
128000	8	125000.00	-2.34%	4	125000.00	-2.34%
256000	4	250000.00	-2.34%	2	250000.00	-2.34%
500000	2	500000.00	0.00%	1	500000.00	0.00%

表 19-4 PCLK=8MHz 波特率计算表

波特率	PCLK = 16 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%
4800	417	4796.16	-0.08%	208	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%
76800	26	76923.08	0.16%	13	76923.08	0.16%
115200	17	117647.06	2.12%	9	111111.11	-3.55%
128000	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	8	250000.00	-2.34%	4	250000.00	-2.34%
500000	4	500000.00	0.00%	2	500000.00	0.00%
1000000	2	1000000.00	0.00%	1	1000000.00	0.00%

表 19-5 PCLK=16MHz 波特率计算表

波特率	PCLK = 24 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1250	2400.00	0.00%	625	2400.00	0.00%
4800	625	4800.00	0.00%	313	4792.33	-0.16%
9600	313	9584.66	-0.16%	156	9615.38	0.16%
19200	156	19230.77	0.16%	78	19230.77	0.16%
38400	78	38461.54	0.16%	39	38461.54	0.16%
57600	52	57692.31	0.16%	26	57692.31	0.16%
76800	39	76923.08	0.16%	20	75000.00	-2.34%
115200	26	115384.62	0.16%	13	115384.62	0.16%
128000	23	130434.78	1.90%	12	125000.00	-2.34%
256000	12	250000.00	-2.34%	6	250000.00	-2.34%
1000000	3	1000000.00	0.00%	2	750000.00	-25.00%
1500000	2	1500000.00	0.00%	1	1500000.00	0.00%

表 19-6 PCLK=24MHz 波特率计算表

波特率	PCLK = 32 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	833	4801.92	0.04%	417	4796.16	-0.08%
9600	417	9592.33	-0.08%	208	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%
76800	52	76923.08	0.16%	26	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%
128000	31	129032.26	0.81%	16	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%

表 19-7 PCLK=32MHz 波特率计算表

波特率	PCLK = 48 MHz					
	OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%
9600	625	9600.00	0.00%	313	9584.66	-0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%
115200	52	115384.62	0.16%	26	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%
256000	23	260869.57	1.90%	12	250000.00	-2.34%
1000000	6	1000000.00	0.00%	3	1000000.00	0.00%
2000000	3	2000000.00	0.00%	2	1500000.00	-25.00%
3000000	2	3000000.00	0.00%	1	3000000.00	0.00%

表 19-8 PCLK=48MHz 波特率计算表

19.3.3 帧错误检测

当工作在 Mode1/2/3 时，UART 具有帧错误检测功能，如果接收数据时硬件未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。检测到帧错误时，UARTx_ISR.FE 置 1。UARTx_ISR.FE 要及时软件清零。

19.3.4 多机通讯

当工作在 Mode2/3 时，将 UARTx_SCON.ADRDET 位置“1”，可开启多机通讯功能。主机可以通过 UARTx_SBUF[8]来区分当前的发送帧是地址帧（UARTx_SBUF[8]=1）还是数据帧（UARTx_SBUF[8]=0）。

- 当为数据帧时，该帧数据不会存入到从机的 UARTx_SBUF 寄存器中，从机也不会产生接收中断。
- 当为地址帧时，由于多机通讯中自动地址识别功能已开启，使得从机可以检测接收到的地址与其自身地址是否相符合。
  - 如果地址相符合，从机会对 UARTx_ISR.RC 置“1”，UARTx_SBUF[8]置“1”，同时将该地址帧存入到 UARTx_SBUF 寄存器中。从机软件看到 UARTx_SBUF[8]=1 并且 UARTx_ISR.RC=1 后，将 UARTx_SCON.ADEDET 位清“0”，接受数据帧。
  - 如果地址不符合，表明主机并不是寻址该从机，从机硬件保持 UARTx_SBUF[8]和 UARTx_ISR.RC 为“0”，软件保持 UARTx_SCON.ADRDET 位为“1”，继续处于地址监听状态。

注：如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 stop 位代替。当从机接收到匹配的地址帧和有效的 stop 位时，UARTx_ISR.RC 会被置“1”。

19.3.4.1 给定地址

UART 设备的 UARTx_SADDR 寄存器用来表示自己的设备给定地址。

UARTx_SADEN 寄存器是地址掩码，当 UARTx_SADEN 的某一位为“0”可定义地址中的无关位，不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时

寻址一个或者多个从机设备。

注意：如果需要给出唯一匹配地址，UARTx_SADEN 寄存器必须设为 8'hFF。给定地址公式如下所示：

GivenAddr = SADDR & SADEN

19.3.4.2 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

BroadCastAddr = SADDR|SADEN

19.3.4.3 举例

假设某从机的 UARTx_SADDR 和 UARTx_SADEN 配置如下：

SADDR：8'b01101001

SADEN：8'b11111011

那么其给定地址和广播地址如下：

Given： 8'b01101x01

Broadcast: 8'b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

8'b01101001 和 8'b01101101 (given address)

8'b11111011 和 8'b11111111 (broadcast address)。

19.3.5 DMAC 硬件握手

UART 模块支持 DMAC 的硬件握手逻辑。

- 将 UARTx_SCON.DMACTXEN 设置为 1，可以打开 UART TX 的 DMAC 硬件握手逻辑。当发送缓存为空时，UART 会向 DMAC 发出数据搬运请求 TX REQ。DMAC 收到该信号，则从 DMAC 源地址搬运一个帧的发送数据到 UARTx_SBUF 中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。
- 将 UARTx_SCON.DMACRXEN 设置为 1，可以打开 UART RX 的 DMAC 硬件握手逻辑。当一帧接收完成，UART 会向 DMAC 发出数据搬运请求 RX REQ。DMAC

收到该信号，则从 UARTx_SBUF 中把接收数据搬运至 DMAC 的目标地址中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

19.3.6 硬件流控

通过增加 nCTS 和 nRTS 信号可以实现 UART 硬件流控的功能，即 UART 硬件模块根据 nCTS 和 nRTS 的高低电平自动控制数据的收发，而无需通过软件来判断。两个 UART 模块之间的硬件流控示意图如下所示：

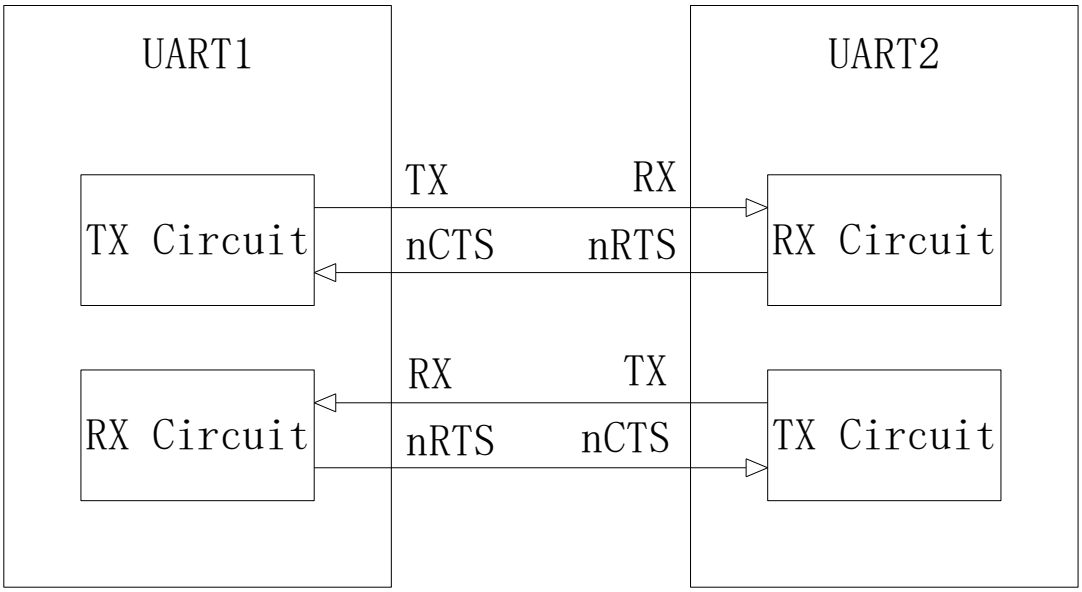


图 19-10 UART 硬件流控

- **nRTS 流控**

nRTS 流控使能时（UARTx_SCON.RTSEN 设置为 1）：

– 当 UART 接收缓存空时，会将 nRTS 变为有效（连接到低电平）。

– 当接收缓存满时，会将 nRTS 变为无效（连接到高电平），表明发送过程会在当前帧结束后停止。

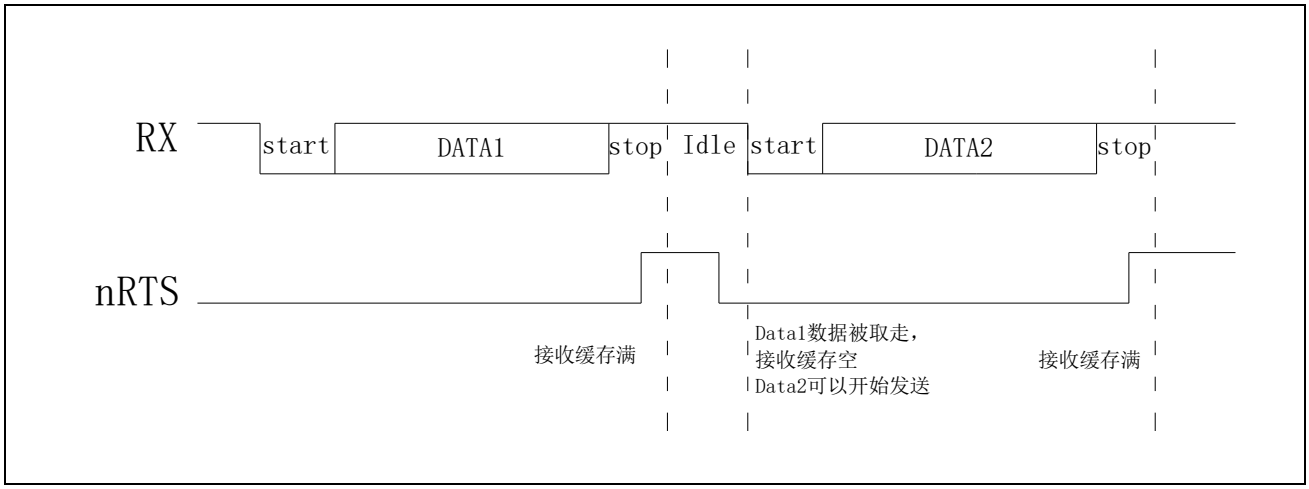


图 19-11 nRTS 硬件流控信号

- **CTS 流控**

nCTS 流控使能时（UARTx_SCON.CTSEN 设置为 1），在 UART 发送下一帧数据之前，先判断 nCTS 的高低电平：

– 如果 nCTS 为有效（连接到低电平），则 UART 发送下一帧数据。

– 如果 nCTS 为无效（连接到高电平），则 UART 当前帧发送完成后，暂停发送下一帧数据。

当 nCTS 流控使能时，一旦 nCTS 信号发生翻转，UARTx_SFLAG.CTSIE 便会硬件置 1。如果 UARTx_SCON.CTSIE 置 1，则会产生中断，同时，nCTS 信号的高低电平会记录在 UARTx_SFLAG.CTS 标志位中。

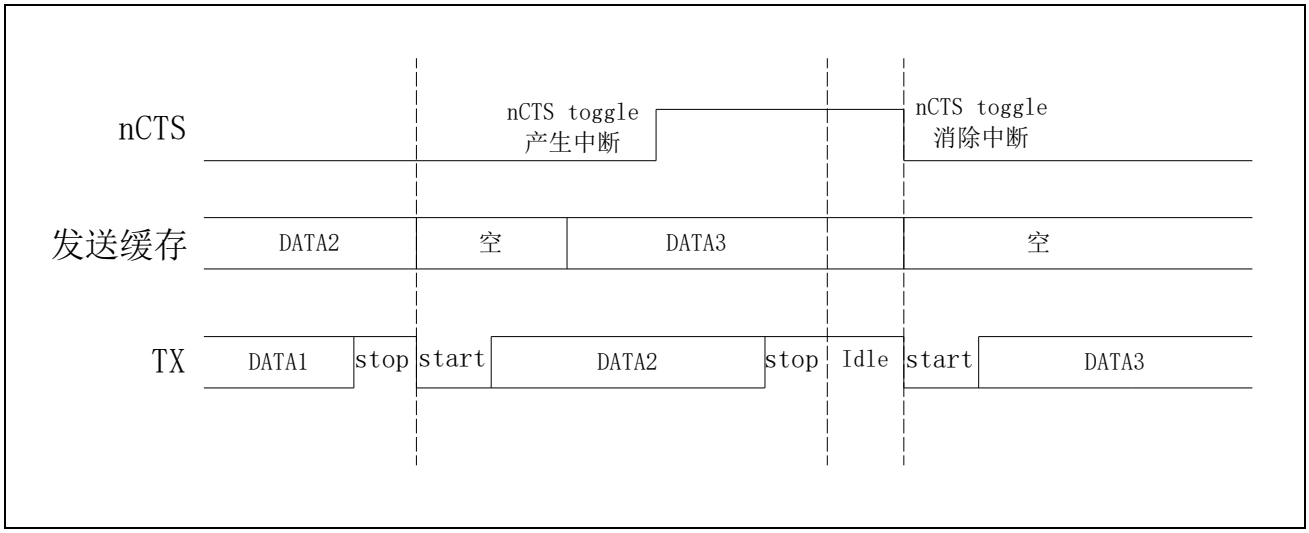


图 19-12 nCTS 硬件流控信号

19.3.7 收发端缓存

- **接收缓存**

UART 模块接收端有一个帧（8/9-Bit）的接收缓存，即保存接收的数据帧直至下一帧数据的 Stop 位被接收才更新该数据帧。

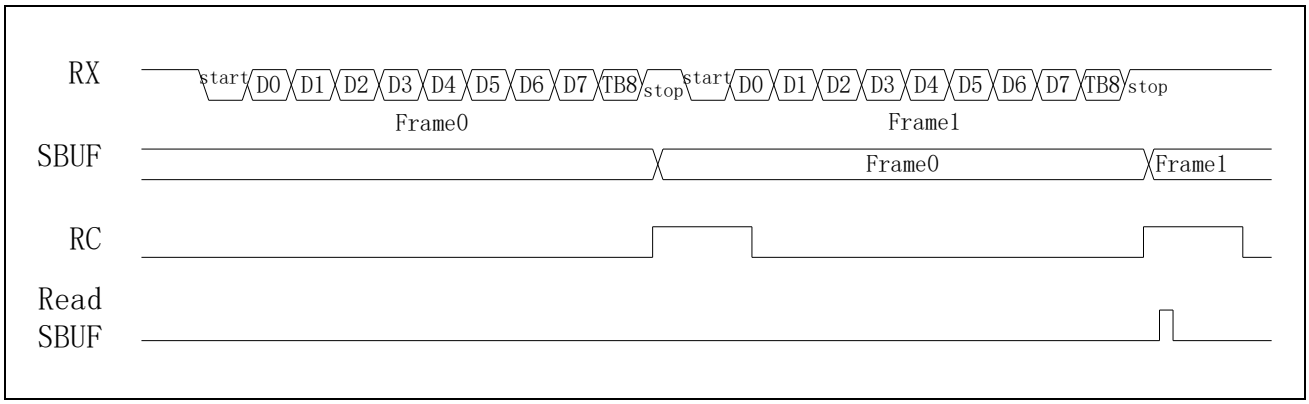


图 19-13 接收缓存

- **发送缓存**

UART 模块发送端有一个帧(8/9-Bit)的发送缓存，当 UART 在发送当前帧时，下一个发送数据软件写入 UARTx_SBUF。

当 UARTx_ISR.TXE=0 时，表明当前发送缓存满，UARTx_SBUF 不能写入下一个发送数据。否则该数据会硬件丢弃。

当 UARTx_ISR.TXE=1 时，表明当前发送缓存空，UARTx_SBUF 可以写入下一个发送数据，在完成当前数据传输后，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

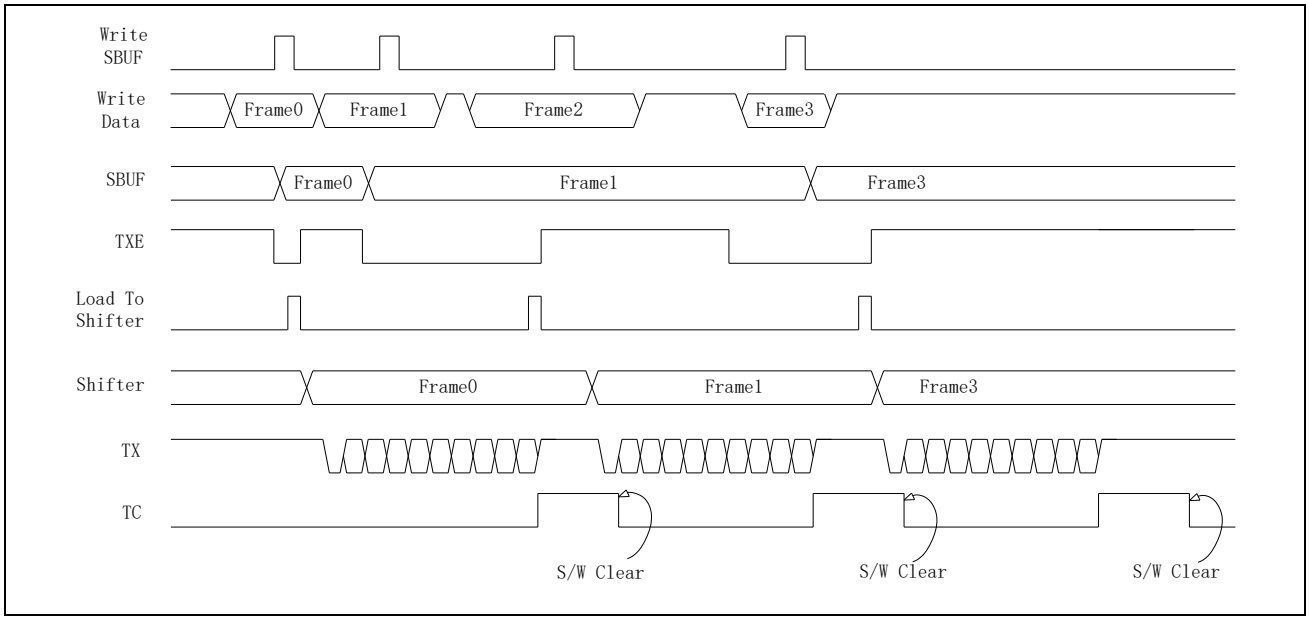

Write SBUF
Write Data
Frame0
Frame1
Frame2
Frame3
SBUF
Frame0
Frame1
Frame3
TXE
Load To Shifter
Shifter
Frame0
Frame1
Frame3
TX
TC
S/W Clear
S/W Clear
S/W Clear

图 19-14　发送缓存

19.4 寄存器

UART0 基地址：0x4000 0000

UART1 基地址：0x4000 0100

UART2 基地址：0x4000 6000

UART3 基地址：0x4000 6400

寄存器	偏移地址	描述
UARTx_SBUF	0x00	数据寄存器
UARTx_SCON	0x04	控制寄存器
UARTx_SADDR	0x08	地址寄存器
UARTx_SADEN	0x0C	地址掩码寄存器
UARTx_ISR	0x10	中断标志位寄存器
UARTx_ICR	0x14	中断标志位清除寄存器
UARTx_SCNT	0x18	波特率寄存器

19.4.1 数据寄存器（UARTx_SBUF）

偏移地址：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31:9	Reserved	
8	DATA[8]	在Mode0/1下，读取该位为0，写入该位无效； 在Mode2/3下，该位表示Bit8数据位，分以下两种情况： （1）当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位PE置1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送； （2）当硬件奇偶校验位关闭时，接收时该位为接收数据Bit8；发送时该位为发送数据Bit8； 注意：当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受B8CONT控制；
7:0	DATA[7:0]	发送数据时，将发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读取。

19.4.2 控制寄存器（UARTx_SCON）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN
									RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOPBIT		PEIE	Reserved			OVER	TXEIE	SM		ADRDET	REN	B8CONT		TCIE	RCIE
RW		RW				RW	RW	RW		RW	RW	RW		RW	RW

位	标记	功能描述
31:23	Reserved	
22	HDSEL	单线半双工模式使能位；0: 关闭单线半双工模式；1：打开单线半双工模式；
21	FEIE	帧错误中断使能位；0：关闭中断；1：打开中断；
20	CTSIE	CTS信号翻转中断使能位；0：关闭中断；1：打开中断；
19	CTSEN	硬件flow control信号使能位；0：关闭flow control信号；1：打开flow control信号；
18	RTSEN	
17	DMATXEN	TX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
16	DMARXEN	RX DMAC的硬件握手信号使能位；0：关闭硬件握手信号；1：打开硬件握手信号；
15:14	STOPBIT	stop bit长度选择；00:1-bit；01:1.5-bit；10:2-bit；11：reserved； 注意：Mode0 时虽然没有 Stop Bit，但仍需把 STOPBIT[1:0]保持为 2’b00；
13	PEIE	奇偶校验错误中断使能位；0：关闭奇偶校验错误中断；1：打开奇偶校验错误中断； 数据接收标志产生于数据停止位接收完成，硬件奇偶校验与数据接收中断由于停止位设置不同，奇偶校验错误中断会提前于数据接收中断。使用此中断使能时请注意。 建议如下： 方法1：关闭PEIE中断使能，接收数据中断后软件判断ISR.PE奇偶校验是否正确。 方法2：关闭PEIE中断使能，接收数据中断通过接收SBUF.BIT8软件判断奇偶校验是否正确。
12:10	Reserved	
9	OVER	Mode0： 无效； Mode1/3： 0：16采样分频； 1：8采样分频； Mode2: 0：32采样分频； 1：16采样分频；
8	TXEIE	TX空中断使能位；0：TX Buffer空中断关闭；1：TX Buffer空中断打开；
7:6	SM	工作模式；00：mode0；01：mode1；10：mode2；11：mode3；
5	ADRDET	多机通讯地址自动识别使能位；0：关闭；1：打开；
4	REN	Mode0： 0：发送；1：接收； Mode1/2/3：0：发送；1：接收/发送；

3:2	B8CONT	Bit8数据控制位； 00：由软件读写SBUF[8]来决定；01：硬件偶校验； 10：硬件奇校验；　　11：保留；
1	TCIE	发送中断使能位；0：发送中断关闭；1：发送中断打开；
0	RCIE	接收中断使能位；0：接收中断关闭；1：接收中断打开；

19.4.3 地址寄存器（UARTx_SADDR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:8	Reserved	
7:0	SADDR	从机设备地址寄存器

19.4.4 地址掩码寄存器（UARTx_SADEN）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:8	RESERVED	
7:0	SADEN	从机设备地址掩码寄存器

19.4.5 标志位寄存器（UARTx_ISR）

偏移地址：0x10

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CTS	CTSIF	PE	TXE	FE	TC	RC
									RO	RO	RO	RO	RO	RO	RO

位	符号	描述
31:7	Reserved	
6	CTS	CTS信号标志位；硬件置1；硬件清零；0：CTS信号为低电平；1：CTS信号为高电平；
5	CTSIF	CTS中断标志位；硬件置1；软件清零；0：CTS信号没有发生反转；1：CTS信号发生反转；
4	PE	奇偶校验错误标志位；硬件置1；软件清零；0：无奇偶校验错误；1：奇偶校验错误；
3	TXE	Tx Buffer空标志位；硬件置1；硬件清零；0：Tx Buffer非空；1：Tx Buffer空
2	FE	帧错误标志位；0：硬件置1；软件清零；
1	TC	发送完毕中断标志位；硬件置1；软件清零；0：发送未完成；1：发送完毕；
0	RC	接收完毕中断标志位；硬件置1；软件清零；0：接收未完成；1：接收完成；

19.4.6 标志位清除寄存器（UARTx_ICR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CTSIFCF	PECF	Res.	FECF	TCCF	RCCF
										R1W0	R1W0		R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	
5	CTSIFCF	CTSIF标志清楚位；写0清除；写1无效；
4	PECF	PE标志清楚位；写0清除；写1无效；
3	Reserved	
2	FECF	FE标志清除位；写0清零；写1无效；
1	TCCF	TC标志清除位；写0清零；写1无效；
0	RCCF	RC标志清除位；写0清零；写1无效；

19.4.7 波特率寄存器（UARTx_SCNT）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:16	Reserved	
15:0	SCNT	波特率计数器

20 循环冗余校验（CRC）

20.1 概述

在许多应用中，都需要用 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛运用。本模块实现 CRC16 和 CRC32 两种算法对数据进行运算和校验。

20.2 功能框图

以下示意了 CRC 算法在数据传输中的一个最典型的应用：

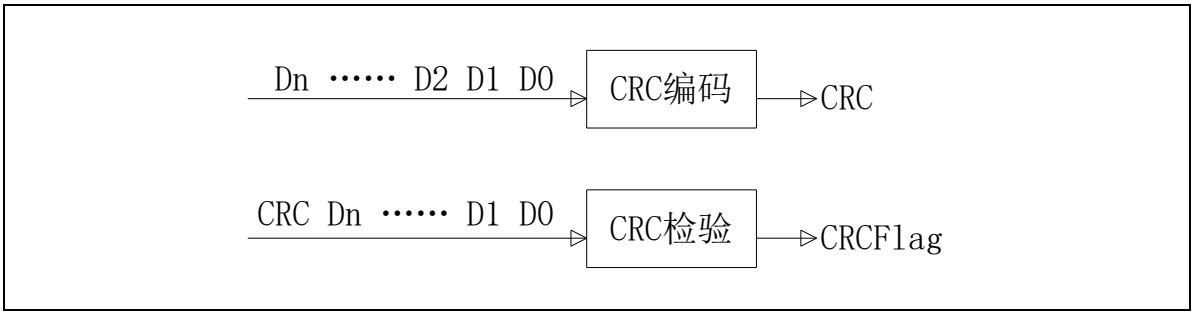


图 20-1 CRC 应用示意图

20.3 功能描述

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。其中，CRC32 的生成多项式为 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$，32 位初值为“0xFFFFFFFF”。CRC16 的生成多项式为 $x^{16}+x^{12}+x^{5}+1$，16 位初值为“0xFFFF”。

本模块功能包括：

- CRC 编码和 CRC 校验；
- 3 种位宽访问方式 8 位、16 位、32 位：

  8 位位宽下输入数据示例为 0x00, 0x11, 0x22 , 0x33, 0x44, 0x55, 0x66, 0x77；

  16 位位宽下输入数据示例为 0x1100, 0x3322, 0x5544, 0x7766；

  32 位位宽下输入数据示例为 0x33221100, 0x77665544；

20.4 寄存器

基地址：0x4002 0900

寄存器	偏移地址	描述
CRC_CR	0x00	CRC 控制寄存器，选择CRC16/32，读取校验结果标志位。
CRC_RESULT	0x04	CRC 结果寄存器，计算完成后对该寄存器读取即获得结果。
CRC_DATA	0x80-0xFF	CRC 数据寄存器，用于输入需要运算的数据。

20.4.1 控制寄存器（CRC_CR）

偏移地址：0x00

复位值：0x0000 0001

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

位	符号	功能描述
31:2	Reserved	
1	FLAG	CRC校验结果标志位；0：当前校验错误；1：当前校验正确；
0	CR	CRC功能选择；0：CRC16；1：CRC32

20.4.2 结果寄存器（CRC_RESULT）

偏移地址：0x04

复位值：0x0000 0000

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

位	符号	描述
31:0	RESULT	CRC结果寄存器；选择CRC16时，取RESULT[15:0]；选择CRC32时，取RESULT[31:0]；

20.4.3 数据寄存器（CRC_DATA）

偏移地址：0x80-0xFF

复位值：0x0000 0000

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

位	符号	功能描述
31:0	DATA	本寄存器用于输入需要运算的数据；本寄存器的地址是一个范围（0x80-0xFF），对该地址范围内的任何地址进行操作，都会认为对本寄存器进行操作。这样定义的目的就是方便软件可以用STM指令对本寄存器进行连续的32位数据写入操作，以加快运算速度。同时本寄存器也支持8/16位的输入方式。 需要特别注意的是，无论是计算还是检验，对本寄存器进行的数据写入操作，必须符合“位宽一致、先低后高”的原则，即“每次写入数据，都必须写入到与本次有效数据位宽相等的寄存器中，并且较低位数据的写入先于较高位数据”。举例如下： 如果待写入数据为0xFEDCBA9876543210，那么 按字节（单次8位）写入时： *((unsigned **char** *) 0x40020980) = 0x10 ; *((unsigned **char** *) 0x40020981) = 0x32 ; …… *((unsigned **char** *) 0x40020986) = 0xDC ; *((unsigned **char** *) 0x40020987) = 0xFE ; 按半字（单次16位）写入时： *((unsigned **short** *) 0x40020980) = 0x3210 ; *((unsigned **short** *) 0x40020982) = 0x7654 ; *((unsigned **short** *) 0x40020984) = 0xBA98 ; *((unsigned **short** *) 0x40020986) = 0xFEDC ; 按全字（单次32位）写入时： *((unsigned **int** *) 0x40020980) = 0x76543210 ; *((unsigned **int** *) 0x40020984) = 0xFEDCBA98 ;

20.5 软件基本操作

20.5.1 CRC16 编码模式

编码模式可以对原始数据编码以计算其 CRC 值，操作流程如下所示：

Step 1： 向 CRC_CR.CR 写入 1’b0，选择 CRC16。

Step 2： 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。

Step 3： 将待编码的原始数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）

Step 4： 读取 CRC_RESULT[15:0]，即为 CRC 值。

Step 5： 如果需要计算其他数据，重复 step1~step4。

20.5.2 CRC16 检验模式

检验模式可以检验已编码的数据是否被篡改，操作流程如下所示：

Step 1： 向 CRC_CR.CR 写入 1’b0，选择 CRC16。

Step 2： 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。

Step 3： 将已编码的数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）

Step 4： 读取 CRC_CR.FLAG，以判定编码的数据是否被篡改。当 CRC_FLAG=0，表示校验失败；当 CRC_FLAG=1，表示校验成功。

Step 5： 如果需要检验其他数据，重复 step1~step4。

20.5.3 CRC32 编码模式

编码模式可以对原始数据编码以计算其 CRC 值，操作流程如下所示：

Step 1： 向 CRC_CR.CR 写入 1，选择 CRC32。

Step 2： 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算。

Step 3： 将待编码的原始数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）

Step 4： 读取 CRC_RESULT[31:0]，即为 CRC 值。

Step 5： 如果需要计算其他数据，重复 step1~step4。

20.5.4 CRC32 检验模式

检验模式可以检验已编码的数据是否被篡改，操作流程如下所示：

Step 1： 向 CRC_CR.CR 写入 1，选择 CRC32。

Step 2： 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算。

Step 3： 将已编码的数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）

Step 4： 读取 CRC_CR.FLAG，以判定编码的数据是否被篡改。当 CRC_CR.FLAG=0，表示校验失败；当 CRC_CR.FLAG=1，表示校验成功。

Step 5： 如果需要检验其他数据，重复 step1~step4。

21 真随机数发生器（TRNG）

21.1 概述

真随机数模块产生 64 位真随机数。

21.2 功能框图

以下示意了 TRNG 模块的数据流：

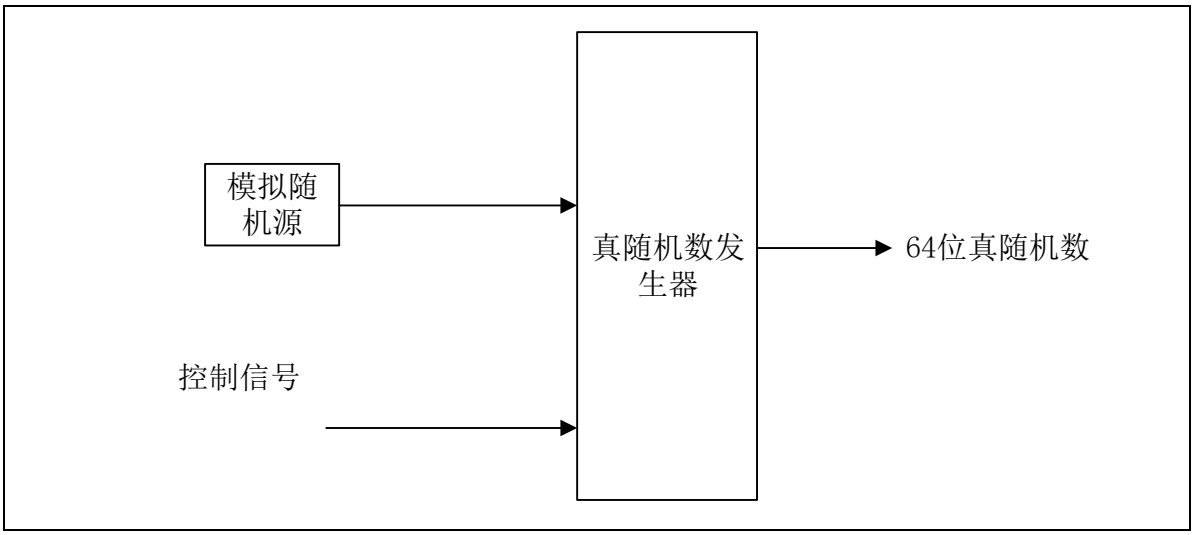


图 21-1 TRNG 数据流

21.3 功能描述

本模块采用内部的模拟随机源，每次启动都可以产生 64bits 真随机数。此外，还可以对真随机数生成的方式进行软件配置，详细内容可查看寄存器描述章节。生成的 64 位真随机数分别存放在 DATA0 和 DATA1 寄存器中。

21.4 寄存器

基地址：0x4000 4C00

寄存器	偏移地址	描述
TRNG_CR	0x00	控制寄存器
TRNG_MODE	0x04	模式寄存器
TRNG_DATA0	0x0C	数据寄存器0
TRNG_DATA1	0x10	数据寄存器1

21.4.1 控制寄存器（TRNG_CR）

偏移地址：0x00

复位值：0x0000 0000

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

位	标记	功能描述
31:2	Reserved	
1	TRNG_RUN	软件写入“1”，开始产生新的64bits随机数；运行完毕后，硬件清零； 0：随机数产生完成； 1：写1启动随机数产生，读1表示随机数正在产生；
0	TRNGcir_EN	随机源电路使能位： 0：关闭随机源； 1：打开随机源；

21.4.2 模式寄存器（TRNG_MODE）

偏移地址：0x04

复位值：0x0000 0010

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

位	标记	描述
31:5	Reserved	
4:2	TRNG_CNT	64bits TRNG的反馈移位次数 3’b000:移位0次（即输出随机源的采样值） 3’b001:移位8次 3’b010:移位16次 3’b011:移位32次 3’b100:移位64次 3’b101:移位128次 3’b110:移位256次 3’b111:Reserved
1	TRNG_FDBK	在移位操作时，64bits TRNG的反馈信号是否与随机源进行异或操作 0：不进行异或操作； 1：进行异或操作；
0	TRNG_LOAD	在产生新的随机数时，64bits TRNG是否从随机源获得新的初始值 0：不装载新的初始值（产生伪随机数）； 1：装载新的初始值（产生真随机数）；

21.4.3 数据寄存器 0（TRNG_DATA0）

偏移地址：0x0C

复位值：---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA0[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[15:0]															
RO															

位	标记	功能描述
31:0	DATA0	软件对本寄存器读取将得到低32位的随机数

21.4.4 数据寄存器 1（TRNG_DATA1）

偏移地址：0x10

复位值：---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
RO															

位	标记	功能描述
31:0	DATA1	软件对本寄存器读取将得到高32位的随机数

21.5 软件基本操作

21.5.1 生成 64bits 真随机数的操作流程（上电第一次）

上电第一次生成 64bits 真随机数，需要经过以下操作：

Step1：打开随机源电路：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“1”，启动随机源电路，开始输出串行真随机数。

Step2：选择重新装载初始值：将真随机数模式寄存器的 Bit0（TRNG_MODE. TRNG_LOAD）设置为“1”，使新产生的真随机数的初始值从随机源获得。

Step3：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE. TRNG_FDBK）设置为“1”，将反馈信号与随机源异或后输入 PRNG 中。

Step4：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE. TRNG_CNT）为“110”，选择移位 256 次。

Step5：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR. TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step6：选择不重新装载初始值：将真随机数模式寄存器的 Bit0（TRNGModeReg. TRNG_Load）设置为“0”。

Step7：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE. TRNG_FDBK）设置为“0”，将反馈信号直接输入 PRNG 中。

Step8：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为“100”，选择移位 64 次。

Step9：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step10：读取真随机数：软件在查询到真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN）变为“0”后，通过读取真随机数数据寄存器 0(TRNG_DATA0)和真随机数数据寄存器 1(TRNG_DATA1)，得到 64Bits 真随机数。

Step11：完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“0”，关闭随机源电路。

21.5.2 生成 64bits 真随机数的操作流程（非上电第一次生成）

非上电第一次生成 64bits 真随机数，需要经过以下操作：

Step1：打开随机源电路：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“1”，启动随机源电路，开始输出串行真随机数。

Step2：选择不重新装载初始值：将真随机数模式寄存器的 Bit0（TRNG_MODE.TRNG_LOAD）设置为“0”。

Step3：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE.TRNG_FDBK）设置为“1”，将反馈信号与随机源异或后输入 PRNG 中。

Step4：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为”110”，选择移位 256 次。

Step5：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step6: 选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（TRNG_MODE.TRNG_FDBK）设置为“0”，将反馈信号直接输入 PRNG 中。

Step7：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4－Bit2（TRNG_MODE.TRNG_CNT）为”100”，选择移位 64 次。

Step8：读取真随机数：软件在查询到真随机数控制寄存器的 Bit1（TRNG_CR.TRNG_RUN）变为“0”后，通过读取真随机数数据寄存器 0（TRNG_Data0）和真随机数数据寄存器 1（TRNG_Data1），得到 64Bits 真随机数。

如果需要继续生成新的真随机数，那么回到 Step2，直到满足要求。

Step9：完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：对真随机数控制寄存器的 Bit0（TRNG_CR.TRNGcir_En）写入“0”，关闭随机源电路。

22 高级加密标准模块（AES）

22.1 功能定义

22.1.1 AES 算法简述

AES（The Advanced Encryption Standard）是美国国家标准技术研究所（NIST）在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分组和一个密钥，输出是一个密文分组；对解密而言，输入是一个密文分组和一个密钥，而输出是一个明文分组。此过程如图 22-1 所示：

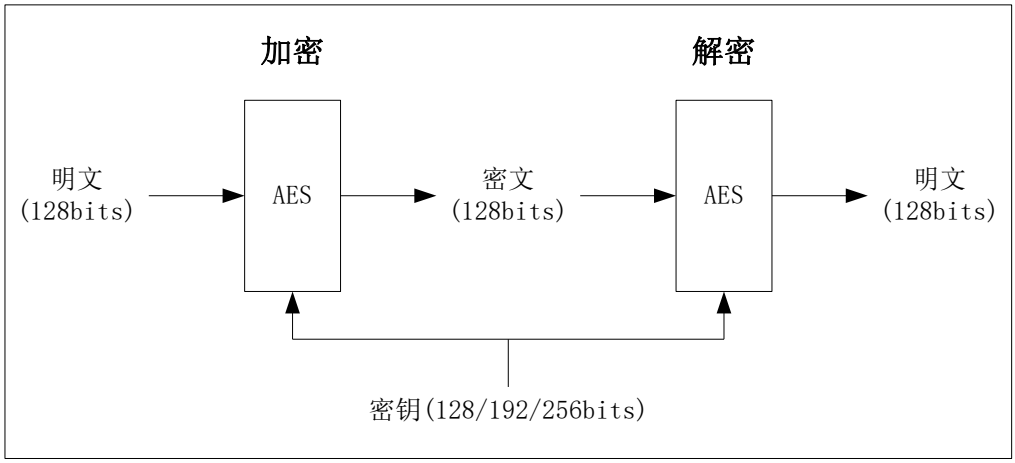


图 22-1 AES 的加解密示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态（state），AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在 GF(2^8)中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如图 22-2 所示：

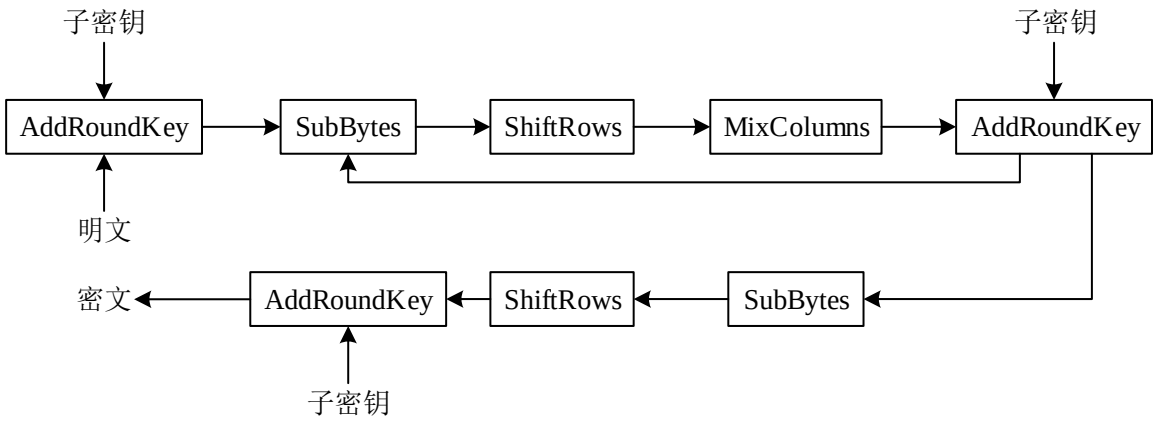


图 22-2 AES 的加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》

22.1.2 AES 模块功能描述

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述；
- 支持 128、192 和 256 位密钥。

22.2 模块寄存器说明

AES 基地址 0x40021400

表 22-1 寄存器列表

寄存器	偏移地址	描述
AES_CR	0x00或0x40	控制寄存器
AES_Data	0x10~0x1C	数据寄存器
AES_Key	0x20~0x3C	密钥寄存器

22.2.1 控制寄存器（AES_CR）

偏移地址：0x00 或 0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											Keysize		Reserved	Mode	Start
											RW			RW	RW

位	符号	描述
31:5	Reserved	保留位，读为0
4:3	Keysize	2’b00：密钥长度为128位 2’b01：密钥长度为192位 2’b10：密钥长度为256位 2’b11：密钥长度为128位
2	Reserved	保留位，读为0
1	Mode	0：加密运算 1：解密运算
0	Start	0：本模块运算结束或未被启动 1：启动本模块进行运算

说明：

1. AES_CR.Start 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
2. 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.Start＝0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

22.2.2 数据寄存器（AES_Data）

偏移地址：0x10~0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data[15:0]															
RW															

位	符号	描述
31:0	Data	存放AES算法的128比特明文/密文

说明：

1. 数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128 位明文	128 位密文	128 位密文	128 位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：0xFFEEDDCCBBAA99887766554433221100

偏移地址	寄存器名称	填入数据
0x10	AES_Data0	0x33221100
0x14	AES_Data1	0x77665544
0x18	AES_Data2	0xBBAA9988
0x1C	AES_Data3	0xFFEEDDCC

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.Start＝0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.Start＝0 时）才能进行，否则对本寄存器的读取将得到全 0。

22.2.3 密钥寄存器（AES_Key）

偏移地址：0x20~0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Key[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Key[15:0]															
RW															

位	符号	描述
31:0	Key	存放AES算法的128、192、256比特密钥

1. 密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。对应的操作顺序如下：

128 位密钥举例：0x0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C

192 位密钥举例：0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	0x17161514

256 位密钥举例：0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504

0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	0x17161514
0x38	AES_Key6	0x1B1A1918
0x3C	AES_Key7	0x1F1E1D1C

2. 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.Start＝0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.Start＝0 时）才能进行，否则对本寄存器的读取将得到全 0。

22.3 异常机制

- 只支持 32 位访问，其它位宽的访问会导致系统异常，进入硬件异常中断。
- 访问 AES 模块的偏移地址大于等于 0x80 的地址，会导致系统异常，进入硬件异常中断。

22.4 本模块操作说明

本模块共有两个功能：加密、解密。对两个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

22.4.1 IP 操作的共同点

1. 在 AES 加解密过程中，数据寄存器会改变，如果下次运算的被操作数据就是本次运算的结果，那么就无需重新写入数据了。
2. 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址 0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
3. 判断模块运算结束的方法：不断读取 AES_CR.Start，如果其值变为 0，则表示运算结束。

22.4.2 加密操作流程

Step 1：将待加密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2：将加密密钥写入密钥寄存器（AES_KEY）中。

Step 3：根据密钥长度设置 AES_CR.Keysize

Step 4：将 AES_CR.Mode 设置为 0，启动加密模式。

Step 5：向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6：等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7：读取数据寄存器（AES_DATA），获得 128 位密文。

22.4.3 解密操作流程

Step 1：将待解密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2：将解密密钥写入密钥寄存器（AES_KEY）中。

Step 3：根据密钥长度设置 AES_CR.Keysize

Step 4：将 AES_CR.Mode 设置为 1，启动解密模式。

Step 5：向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6：等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7：读取数据寄存器（AES_DATA），获得 128 位明文。

22.4.4 数据示例

128 位明文：0xFFEEDDCCBBAA99887766554433221100

128 位密钥：0x0F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-2 128 位操作寄存器示例

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

128 位明文：0xFFEEDDCCBBAA99887766554433221100

192 位密钥：0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-3 192 位操作寄存器示例

加密前

寄存器	值（密钥）	寄存器	值（明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		

加密后

寄存器	值（密钥）	寄存器	值（密文）
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110		
Key5	0x17161514		

128 位明文：0xFFEEDDCCBBAA99887766554433221100

256 位密钥：0x1F1E1D1C1B1A19181716151413121110 0F0E0D0C0B0A0908070605040 3020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 22-4　192 位操作寄存器示例

加密前

寄存器	值 （密钥）	寄存器	值 （明文）
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

加密后

寄存器	值 （密钥）	寄存器	值 （密文）
Key0	0x03020100	Data0	0xCAB7A28E
Key1	0x07060504	Data1	0xBF456751
Key2	0x0B0A0908	Data2	0x9049FCAE
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

22.5 运行时间说明

本模块从启动一次运算（AES_CR.Start 写入 1）到该次运算结束（AES_CR.Start 恢复到 0）所需时间如表 22-5 所示：

表 22-5 AES 加解密运行时间

	128 位密钥	192 位密钥	256 位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

23 液晶控制器（LCD）

23.1 LCD 简介

LCD 控制器是一款适用于单色无源液晶显示器(LCD)的数字控制器/驱动器，最多具有 8 个公用端子（COM）和 48 个区段端子（SEG），用以驱动 208 (4x52)或 384 (8x48)个 LCD 图像元素。端子的确切数量取决于数据手册中所述的器件引脚。

LCD 由若干区段（像素或完整符号）组成，这些区段均可点亮或熄灭。每个区段都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，相应的区段可见。区段电压必须为交流，以避免液晶中出现电泳效应（这将影响显示效果）。之后，必须在区段两端生成波形以避免出现直流。

词汇表

液晶（LCD）：无源显示面板，带有直接引向区段的端子。

公用（COM）：连接到多个区段的电气连接端子。

偏置（BIAS）：驱动 LCD 时使用的电压等级，定义为 1/(驱动 LCD 显示的电压等级数–1)。

区段（SEG）：最小可视单元（LCD 显示器上的最小组成元素，线条或点）。

占空比（DUTY）：定义为 1/(LCD 显示器上的公用端子数)的数字。

帧：写入区段的波形的一个周期。

帧速率：每秒帧数，即每秒激励 LCD 区段的次数。

23.2 LCD 主要特性

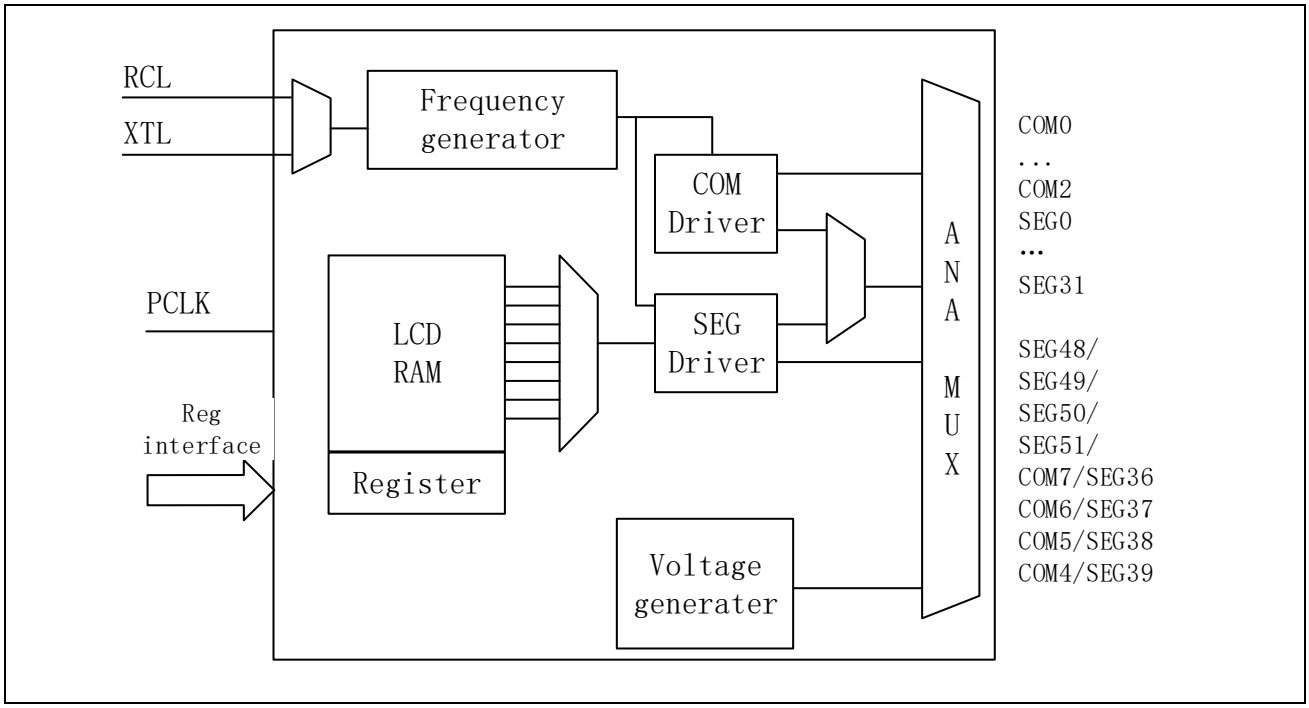
- 高度灵活的帧速率控制。
- 支持静态、1/2、1/3、1/4、1/6 和 1/8 占空比。
- 支持 1/2、1/3 偏置。
- 多达 16 个寄存器的 LCD 数据 RAM。
- 可通过软件配置 LCD 的对比度。
- 3 种驱动波形生成方式

– 内部电阻分压、外部电阻分压，外部电容分压方式

– 可通过软件配置内部电阻分压方式的功耗，从而匹配 LCD 面板所需的电容电荷

- 支持低功耗模式：LCD 控制器可在 Active、Sleep、DeepSleep 模式下进行显示。
- 可配置帧中断。
- 支持 LCD 闪烁功能且可配置多种闪烁频率
- 未使用的 LCD 区段和公共引脚可配置为数字或模拟功能。

23.3 LCD 框图



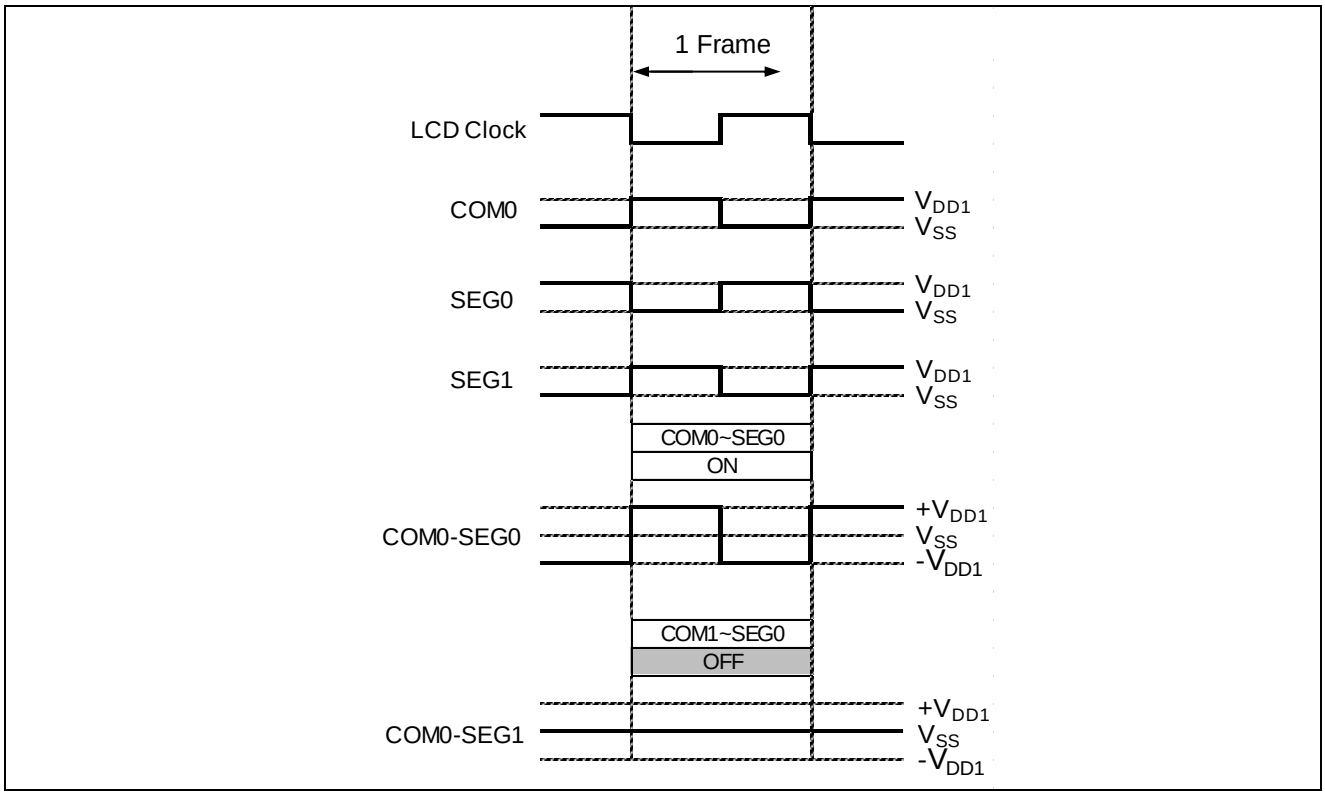
23.4 LCD 驱动波形

LCD 支持 5 种占空比（Duty）的驱动波形：静态、1/2、1/3、1/4、1/6 和 1/8，由 LCD_CR0.Duty 进行设置。LCD 支持 2 种偏置（Bias）的驱动波形：1/2、1/3，由 LCD_CR0.Bias 进行设置。建议的组合方式如下表所示：

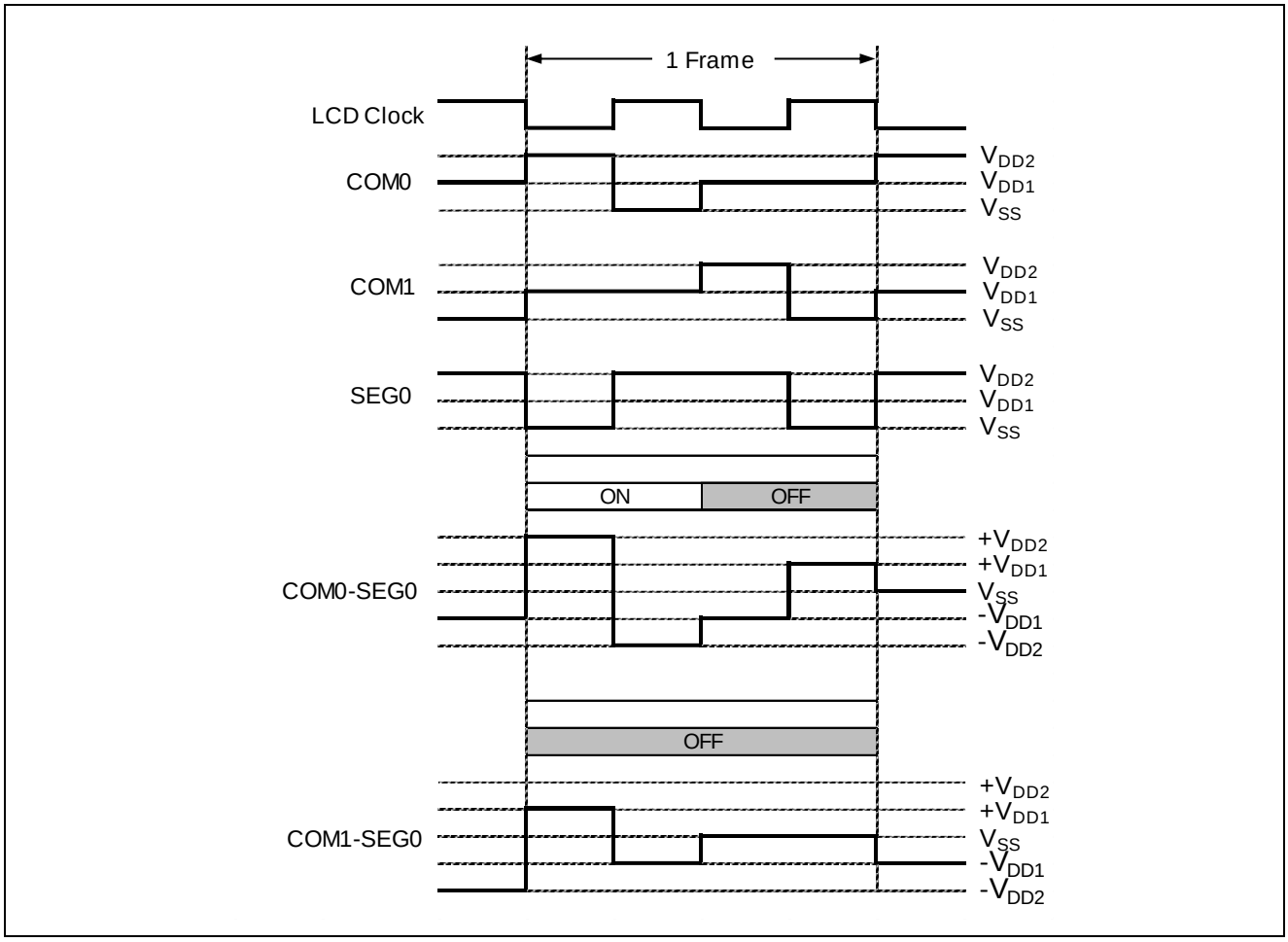
	1/2 Duty	1/3 Duty	1/4 Duty	1/6 Duty	1/8 Duty
1/2 Bias	√	√	**不推荐**	**不推荐**	**不推荐**
1/3 Bias	**不推荐**	**不推荐**	√	√	√

各模式下的驱动波形如下方所示：

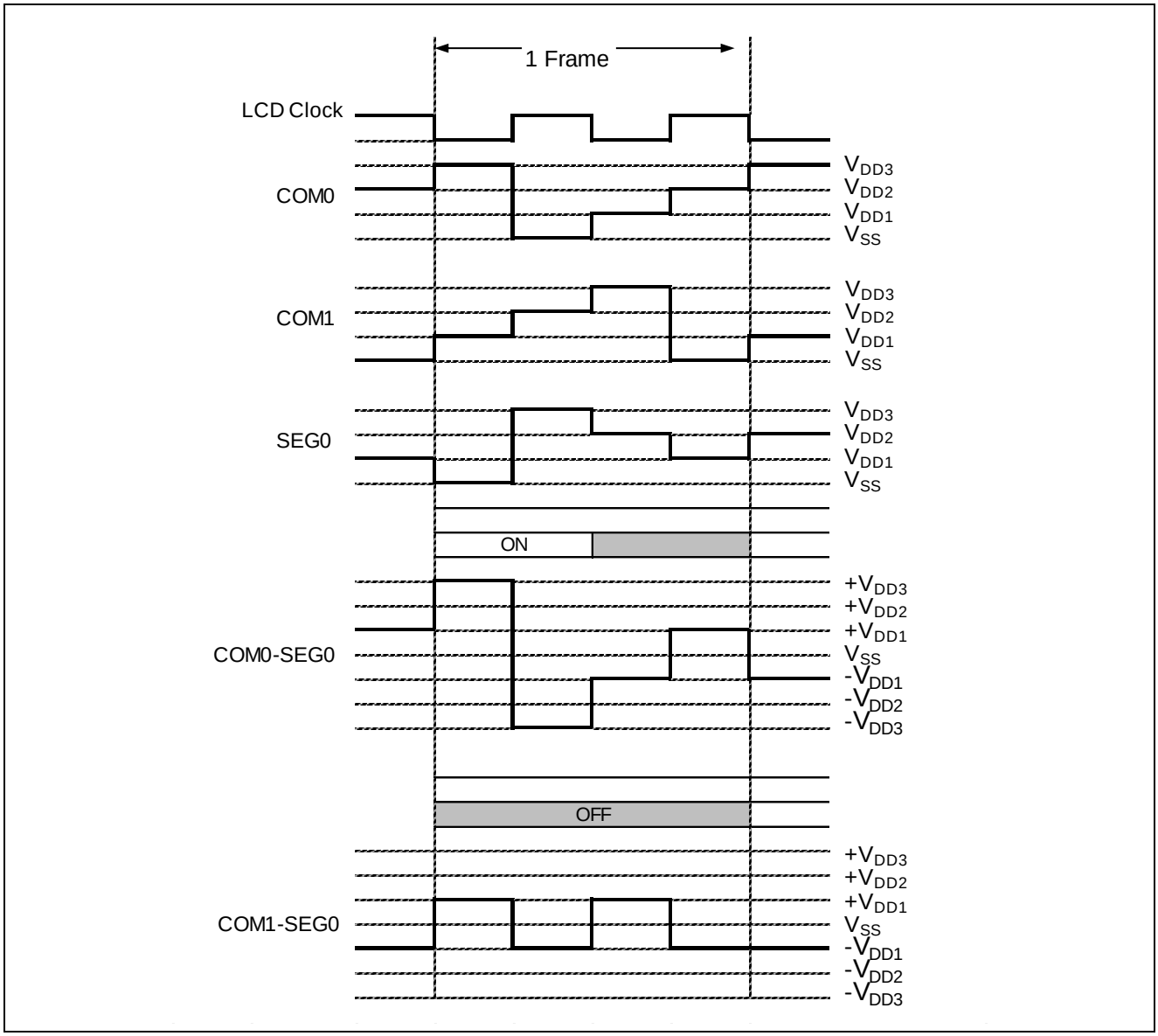
23.4.1 静态驱动波形



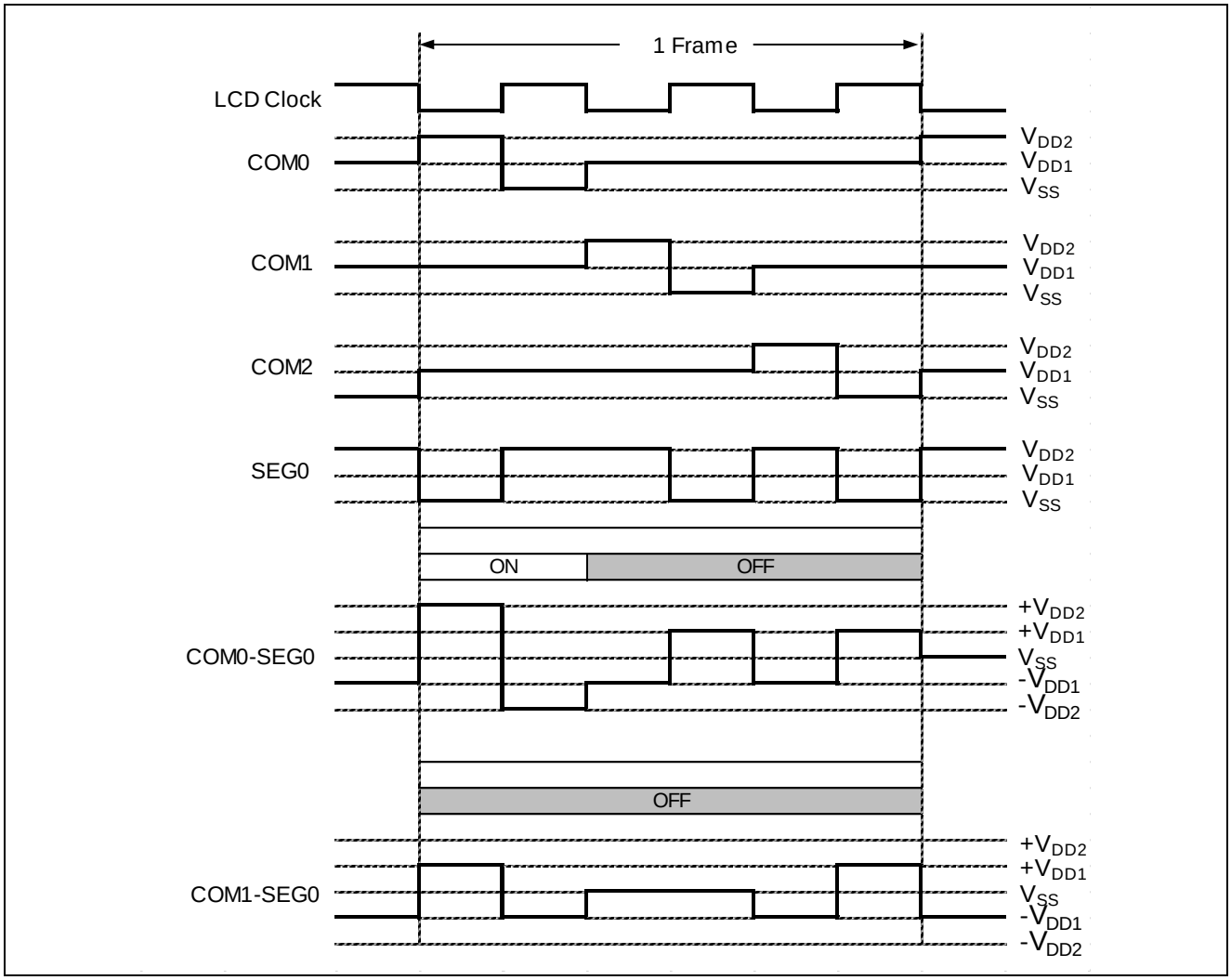
23.4.2 1/2Duty 1/2Bias 驱动波形



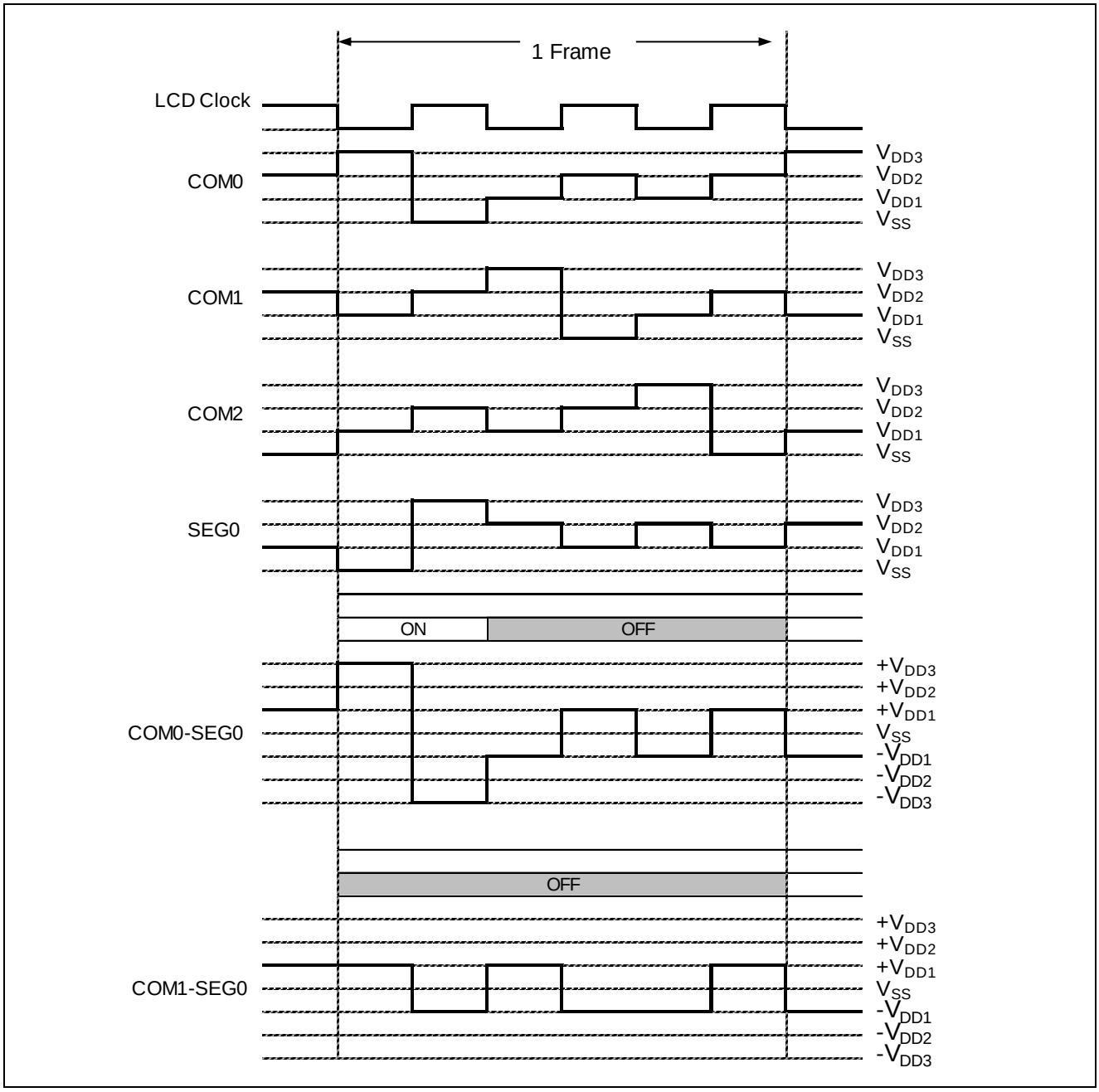
23.4.3 1/2Duty 1/3Bias 驱动波形



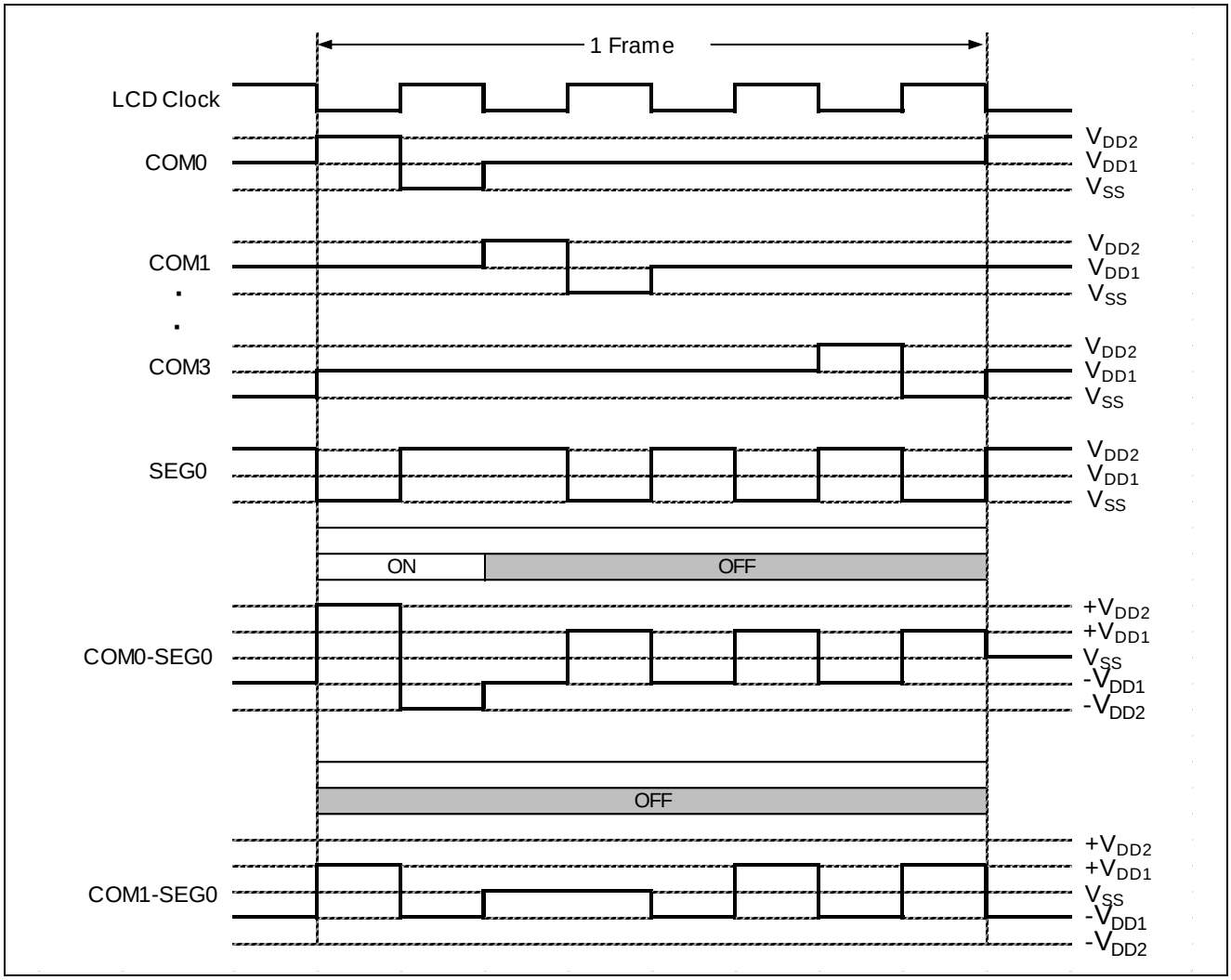
23.4.4 1/3Duty 1/2Bias 驱动波形



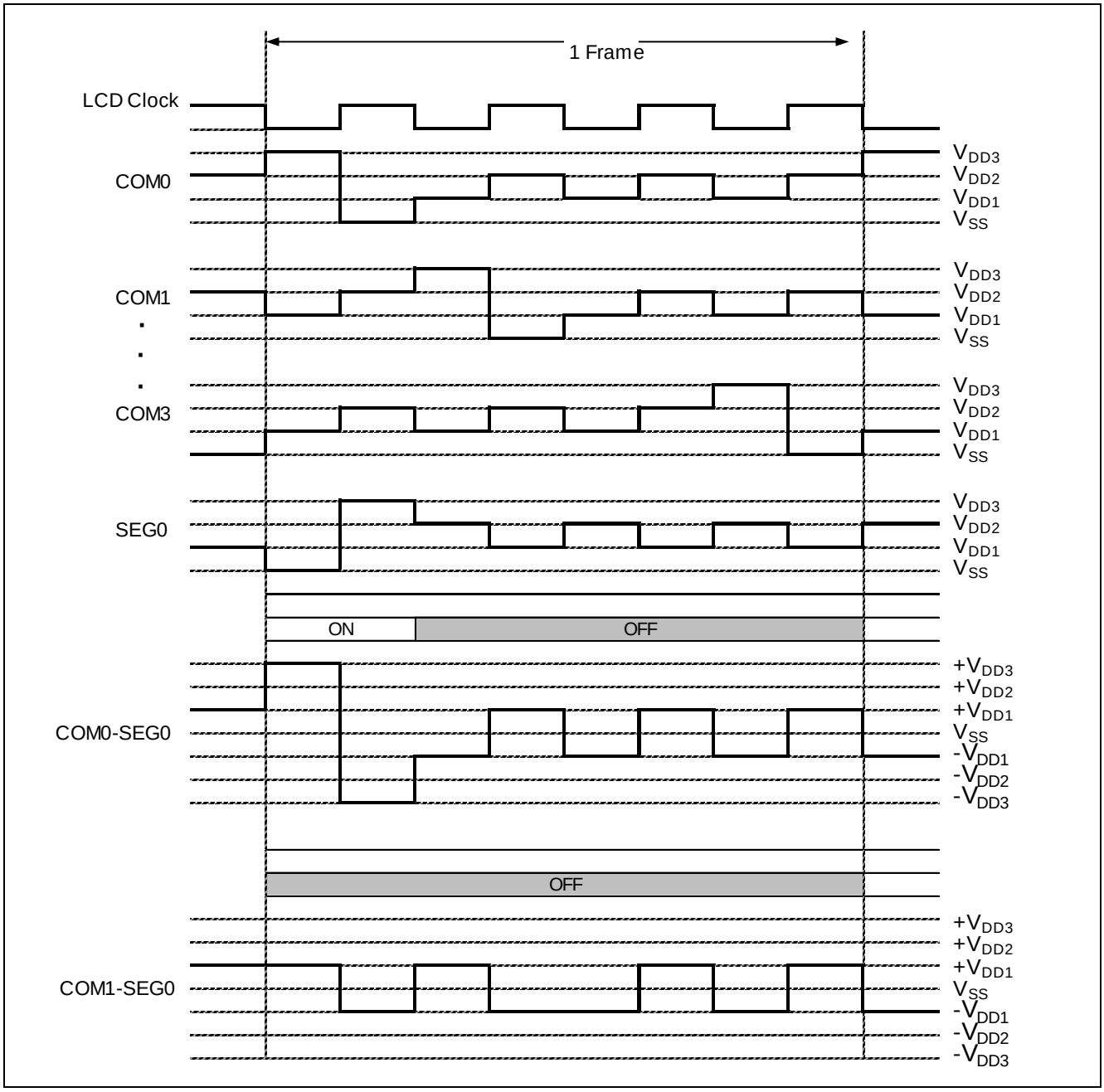
23.4.5 1/3Buty 1/3Bias 驱动波形



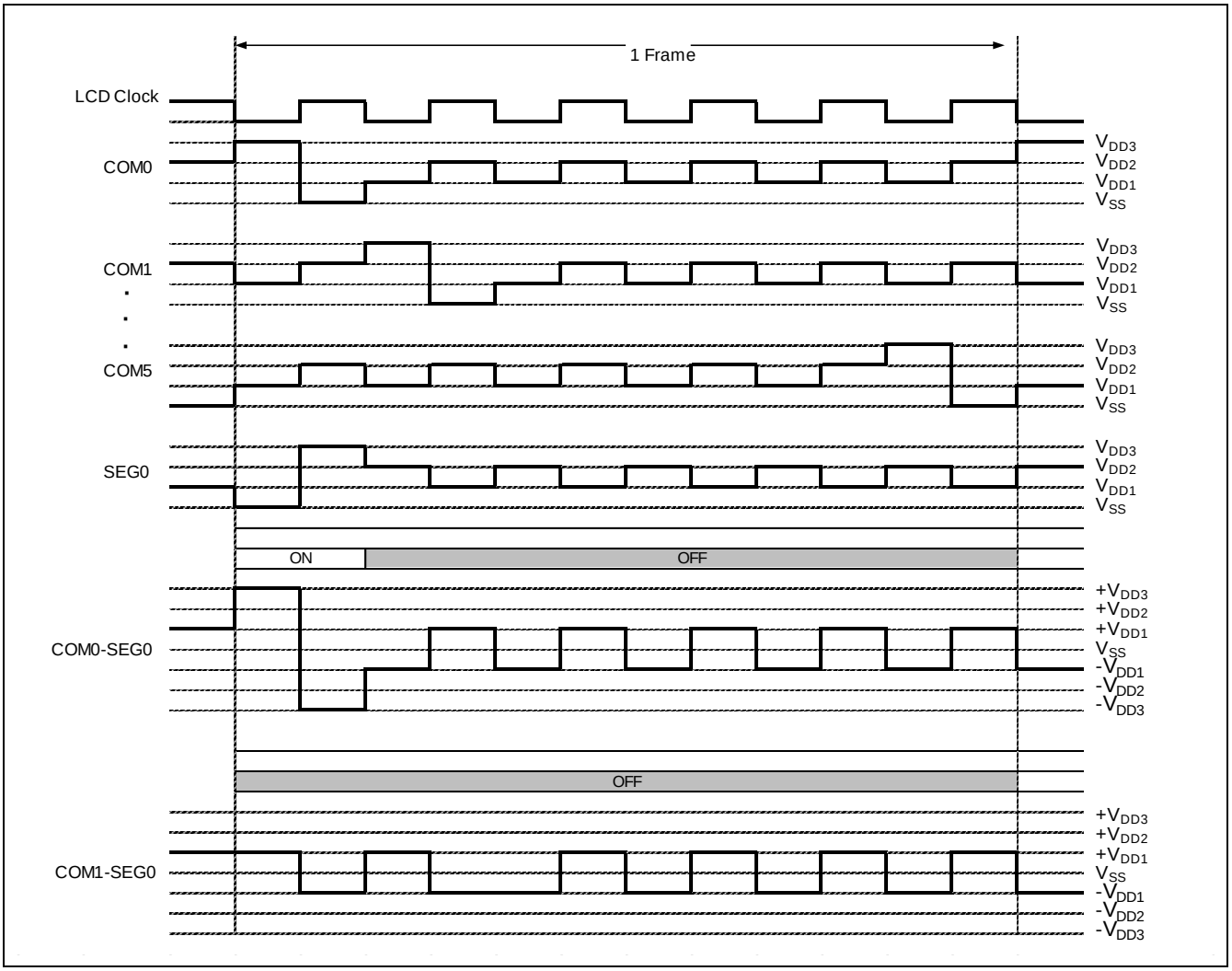
23.4.6 1/4Duty 1/2Bias 驱动波形



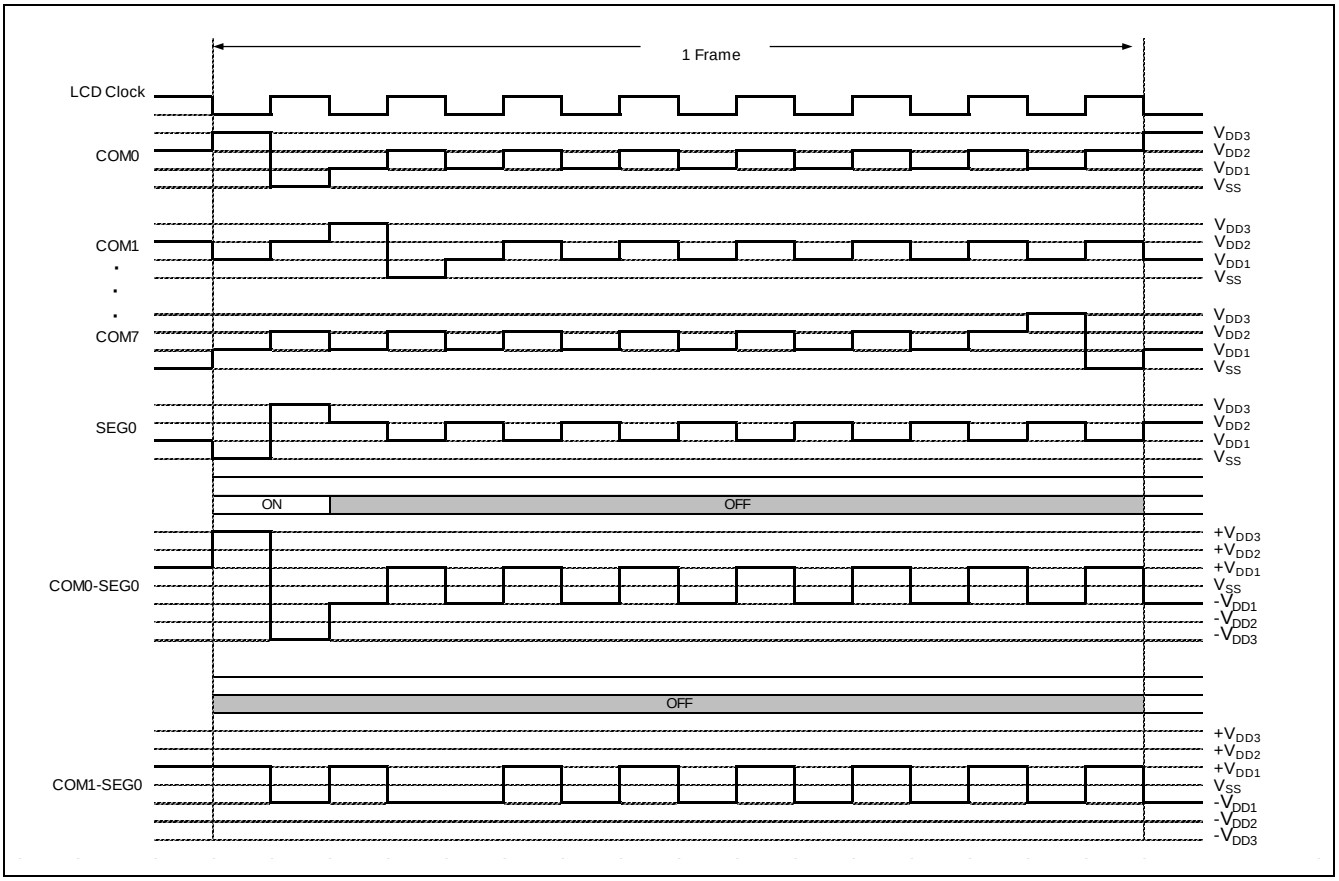
23.4.7 1/4Duty 1/3Bias 驱动波形



23.4.8 1/6Duty 1/3Bias 驱动波形



23.4.9 1/8Duty 1/3Bias 驱动波形



23.5 LCD Bias 产生电路

LCD 的 Bias 电压具有 3 种来源：内部电阻分压、外部电阻分压、外部电容分压。

1）当选择内部电阻分压时，芯片会自动切换到内部的电路，这种模式驱动能力较弱。当选择 1/2bias 时，大功耗/中功耗/小功耗电阻分别为 240K/360K/720K；当选择 1/3bias 时，大功耗/中功耗/小功耗电阻分别为 360K/540K/1080K。

2）当选择外部电阻分压时，根据系统功耗和 LCD 驱动能力来调整阻值，推荐使用 500K 电阻。当选择外部电容分压时，驱动能力较强，推荐使用 100nF 电容。但这两种电路需要用户在芯片的外围引脚搭建相关电路，并占用 4 个 IO 口。

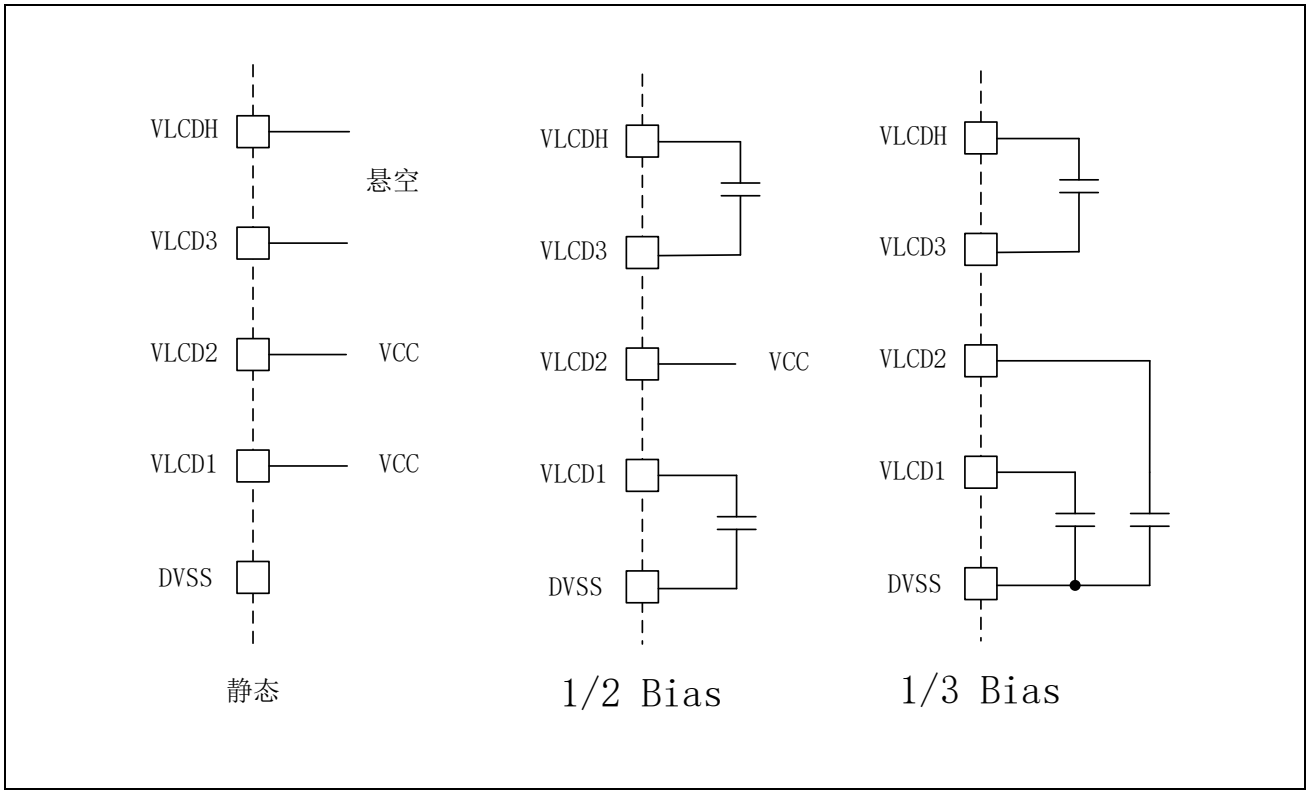
23.5.1 内部电阻模式

内部电阻模式 VLCDH,VLCD1~VLCD3 可以作为 LCD SEG 输出或者 IO 端口使用。

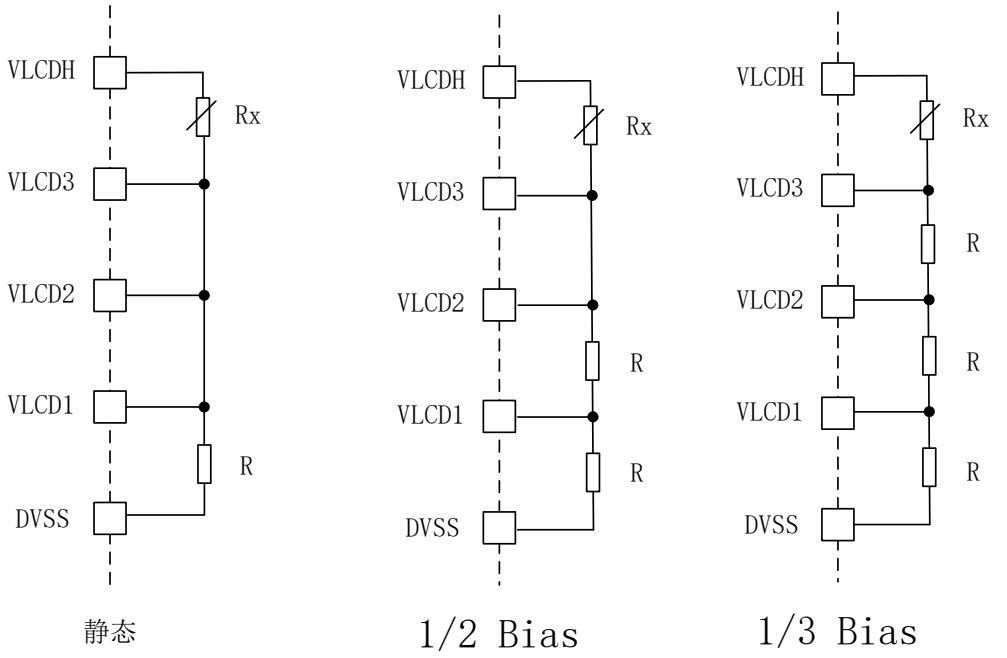
内部电阻模式，LCD 的驱动电压由 CR0.Contrast 控制，如下表所示：

CR0.Contrast	VLCD(1/3 bias)	VLCD(1/2 bias)
0	1.00 * VCC	1.00 * VCC
1	0.94 * VCC	0.92 * VCC
2	0.9 * VCC	0.85 * VCC
3	0.85 * VCC	0.8 * VCC
4	0.81 * VCC	0.75 * VCC
5	0.78 * VCC	0.70 * VCC
6	0.75 * VCC	0.66 * VCC
7	0.72 * VCC	0.63 * VCC
8	0.70 * VCC	0.61 * VCC
9	0.67 * VCC	0.58 * VCC
10	0.65 * VCC	0.55 * VCC
11	0.63 * VCC	0.53 * VCC
12	0.61 * VCC	0.51 * VCC
13	0.59 * VCC	0.48 * VCC
14	0.57 * VCC	0.47 * VCC
15	0.55 * VCC	0.45 * VCC

23.5.2 外部电容模式


VLCDH
悬空
VLCD3
VLCD2
VCC
VLCD1
VCC
DVSS
静态
VLCDH
VLCD3
VLCD2
VCC
VLCD1
DVSS
1/2 Bias
VLCDH
VLCD3
VLCD2
VLCD1
DVSS
1/3 Bias

23.5.3 外部电阻模式



注意：

- Rx 为可调电阻，用于调节 LCD 显示对比度
- 根据使用 LCD 屏幕选择合适的电阻 R

23.6 DMA

LCD 支持软件和硬件触发 DMA 数据传输，可以将需要显示的内容从 RAM 或 ROM 中自动搬到 LCD 显示 RAM 中。硬件触发使用的是帧中断信号。LCD 使用的 DMA 通道号为[6]。

DMA 数据传送配置流程：

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择 LCD DMA
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址

6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 使能 LCD DMA 触发

23.7 中断

当 LCD 设置有效时，LCD 中断可以配置为帧数产生中断。

23.8 LCD 显示模式

LCD 支持两种显示模式。一种以 COM 为显示单元，同一个 SEG 的所有 COM 段在同一字节中（模式 0）。另外一种为同一个 COM 的不同 SEG 在同一个字节中（模式 1）。根据 LCD 面板选择合适的显示方式可以简化程序操作。

23.8.1 LCD 显示模式 1 (MODE = 1)

1/8 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
COM6																																	LCDRAM6
COM7																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35					SEG40	SEG41	SEG42	SEG43	SEG44	SEG45	SEG46	SEG47																	
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
COM3																																	LCDRAMB
COM4																																	LCDRAMC
COM5																																	LCDRAMD
COM6																																	LCDRAME
COM7																																	LCDRAMF

1/6 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37			SEG40	SEG41	SEG42	SEG43	SEG44	SEG45	SEG46	SEG47	SEG48	SEG49	SEG50	SEG51													
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
COM3																																	LCDRAMB
COM4																																	LCDRAMC
COM5																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

1/4 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
																																	LCDRAM4
																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39	SEG40	SEG41	SEG42	SEG43	SEG44	SEG45	SEG46	SEG47	SEG48	SEG49	SEG50	SEG51													
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
COM3																																	LCDRAMB
																																	LCDRAMC
																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

1/3 Duty 1/2 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
																																	LCDRAM3
																																	LCDRAM4
																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39	SEG40	SEG41	SEG42	SEG43	SEG44	SEG45	SEG46	SEG47	SEG48	SEG49	SEG50	SEG51													
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
																																	LCDRAMB
																																	LCDRAMC
																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

23.8.2 LCD 显示模式 0 (MODE = 0)

1/8 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG29								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7
SEG32								SEG40								SEG48																LCDRAM8
SEG33								SEG41								SEG49																LCDRAM9
SEG34								SEG42								SEG50																LCDRAMA
SEG35								SEG43								SEG51																LCDRAMB
								SEG44																								LCDRAMC
								SEG45																								LCDRAMD
								SEG46																								LCDRAME
								SEG47																								LCDRAMF

1/6 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG29								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7
SEG32								SEG40								SEG48																LCDRAM8
SEG33								SEG41								SEG49																LCDRAM9
SEG34								SEG42								SEG50																LCDRAMA
SEG35								SEG43								SEG51																LCDRAMB
SEG36								SEG44																								LCDRAMC
SEG37								SEG45																								LCDRAMD
								SEG46																								LCDRAME
								SEG47																								LCDRAMF

1/4 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG29								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7
SEG32								SEG40								SEG48																LCDRAM8
SEG33								SEG41								SEG49																LCDRAM9
SEG34								SEG42								SEG50																LCDRAMA
SEG35								SEG43								SEG51																LCDRAMB
SEG36								SEG44																								LCDRAMC
SEG37								SEG45																								LCDRAMD
SEG38								SEG46																								LCDRAME
SEG39								SEG47																								LCDRAMF

1/3 Duty 1/2 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2						
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG29								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7
SEG32								SEG40								SEG48																LCDRAM8
SEG33								SEG41								SEG49																LCDRAM9
SEG34								SEG42								SEG50																LCDRAMA
SEG35								SEG43								SEG51																LCDRAMB
SEG36								SEG44																								LCDRAMC
SEG37								SEG45																								LCDRAMD
SEG38								SEG46																								LCDRAME
SEG39								SEG47																								LCDRAMF

23.9 LCD 寄存器

基地址 0x40005C00

寄存器	偏移地址	描述
LCD_CR0	0x000	LCD 配置寄存器0
LCD_CR1	0x004	LCD 配置寄存器1
LCD_INTCLR	0x008	LCD 中断清除寄存器
LCD_POEN0	0x00C	LCD 输出配置寄存器
LCD_POEN1	0x010	LCD 输出配置寄存器
LCD_RAM0	0x040	LCD RAM0
LCD_RAM1	0x044	LCD RAM1
LCD_RAM2	0x048	LCD RAM2
LCD_RAM3	0x04C	LCD RAM3
LCD_RAM4	0x050	LCD RAM4
LCD_RAM5	0x054	LCD RAM5
LCD_RAM6	0x058	LCD RAM6
LCD_RAM7	0x05C	LCD RAM7
LCD_RAM8	0x060	LCD RAM8
LCD_RAM9	0x064	LCD RAM9
LCD_RAMA	0x068	LCD RAM10
LCD_RAMB	0x06C	LCD RAM11
LCD_RAMC	0x070	LCD RAM12
LCD_RAMD	0x074	LCD RAM13
LCD_RAME	0x078	LCD RAM14
LCD_RAMF	0x07C	LCD RAM15

表 23-1 LCD 寄存器

23.9.1 配置寄存器 0（LCD_CR0）

偏移地址 0x000

复位值 0x000000DA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Contrast				BSEL			Duty			Bias	CpClk		LcdClk		EN
RW				RW			RW			RW	RW		RW		RW

位	标记	功能描述
31:16	Reserved	保留
15:12	Contrast	LCD对比度调整 注：仅当Bias电压来源选择内部电阻分压时有效。 Constrast值越大，LCD波形的幅度越小。 0X0时，LCD波形幅度最大，对比度最大； …… 0XF时，LCD波形幅度最小，对比度最小；
11:9	BSEL	Bias电压来源选择 111：Reserved 110：内部电阻分压，大功耗模式 101：Reserved 100：内部电阻分压，小功耗模式 011：Reserved 010：内部电阻分压，中功耗模式 001：电容分压模式，需要外部电路配合 000：外部电阻模式，需要外部电路配合
8:6	Duty	LCD duty配置 000: 静态 001: 1/2 duty 010: 1/3 duty 011: 1/4 duty 100: Reserved 101: 1/6 duty 110: Reserved 111: 1/8 duty

duty	PAD	FUNCTION
1/4 duty	COM4SEG39PAD	SEG39
	COM5SEG38PAD	SEG38

			COM6SEG37PAD	SEG37
			COM7SEG36PAD	SEG36
		1/6 duty	COM4SEG39PAD	COM4
			COM5SEG38PAD	COM5
			COM6SEG37PAD	SEG37
			COM7SEG36PAD	SEG36
		1/8 duty	COM4SEG39PAD	COM4
			COM5SEG38PAD	COM5
			COM6SEG37PAD	COM6
			COM7SEG36PAD	COM7
5	Bias	LCD Bias配置 0: 1/3 bias 1: 1/2 bias		
4:3	CpClk	电压泵时钟频率选择 00: 2k Hz 01: 4k Hz 10: 8k Hz 11: 16k Hz		
2:1	LcdClk	LCD扫描频率选择 00: 64Hz 01: 128Hz 10: 256Hz 11: 512Hz 注：LCD帧频率 = LCD扫描频率×Duty		
0	EN	LCD使能控制 1：使能 0：禁止		

23.9.2 配置寄存器 1（LCD_CR1）

偏移地址 0x004

复位值 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				INTF	DmaEn	IE	Mode	ClkSrc	BlinkEn	BlinkCnt					
				RO	RW	RW	RW	RW	RW	RW					

位	标记	功能描述
31:11	Reserved	保留
11	INTF	中断标志 1：中断 0：无中断
10	DmaEn	DMA硬件触发使能 1：使能LCD中断触发DMA, 0：禁止LCD中断触发DMA
9	IE	中断使能 1：使能 0：禁止
8	Mode	LCD RAM显示模式选择 0 模式0 1 模式1
7	ClkSrc	LCD时钟源选择 1：XTL 0：RCL
8	BlinkEn	LCD闪屏配置 1：使能 0：禁止
5:0	BlinkCnt	闪屏频率与LCD中断间隔设置 注：LCD闪烁频率为 = LCD帧频率 / (BlinkCnt+1) LCD中断间隔= (BlinkCnt+1)*(1/LCD 帧频率)

23.9.3 中断清除寄存器（LCD_INTCLR）

偏移地址 0x008

复位值 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					INTF	Reserved									
					R0W0										

位	标记	功能描述
31:11	Reserved	保留
10	INTF	中断标志清除，写0清除，写1无效
9:0	Reserved	保留

23.9.4 输出配置寄存器 0（LCD_POEN0）

偏移地址 0x00C

复位值 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:0	Sx	Segx输出控制位 0: SEG输出使能 1: SEG输出关闭，可以使用其他功能，如 IO,模拟输入输出

23.9.5 输出配置寄存器 1（LCD_POEN1）

偏移地址 0x010

复位值 0x01FF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved							C3	C2	C1	C0	S51	S50	S49	S48	S47
							RW	RW	RW	RW	RW	RW	RW	RW	RW

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S46	S45	S44	MUX	S43	S42	S41	S40	S39 C4	S38 C5	S37 C6	S36 C7	S35	S34	S33	S32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
3:0	Sx	Segx输出控制位 0: SEG输出使能 1: SEG输出关闭，可以使用其他功能，如 IO,模拟输入输出
7:4	SxCy	Segx/COMy输出控制位 0: SEG/COM输出使能 1: SEG/COM输出关闭，可以使用其他功能，如 IO,模拟输入输出 SEG COM 管脚功能选择由CR0.DUTY决定
11:8	Sx	Segx输出控制位 0: SEG输出使能 1: SEG输出关闭，可以使用其他功能，如 IO,模拟输入输出
12	MUX	SEG32~SEG35端口功能选择，详细参考选择表格
20:13	Sx	SEG40-SEG51端口功能选择
24:21	Cx	COMx输出控制位 0: COM输出使能 1: COM输出关闭，可以使用其他功能，如 IO,模拟输入输出

VLCDXSEGXPAD 状态描述		寄存器如何配置				
		MUX	S<35：32>	BSEL		
	VLCDHSEG35=IO	1	1111	X	X	X

VLCDXSEGXPAD 选择 GPIO，LCD disable(LCD_ON=0)时		VLCD3SEG34=IO VLCD2SEG33=IO VLCD1SEG32=IO					
VLCDXSEGXPAD 选择 IO，LCD enable(LCD_ON=1)时，这时只能选择内部电阻工作模式	小电阻（大电流）模式	VLCDHSEG35=IO VLCD3SEG34=IO VLCD2SEG33=IO VLCD1SEG32=IO	1	1111	1	1	0
	中电阻（中电流）模式	VLCDHSEG35=IO VLCD3SEG34=IO VLCD2SEG33=IO VLCD1SEG32=IO	1	1111	0	1	0
	大电阻（小电流）模式	VLCDHSEG35=IO VLCD3SEG34=IO VLCD2SEG33=IO VLCD1SEG32=IO	1	1111	1	0	0
选择外部电阻工作模式，内部电阻断路		VLCDHSEG35=VOUT VLCD3SEG34=VLCD3 VLCD2SEG33=VLCD2 VLCD1SEG32=VLCD1	0	0000	0	0	0
选择内部电阻工作模式	小电阻（大电流）模式	VLCDHSEG35=SEG35/IO VLCD3SEG34=SEG34/IO VLCD2SEG33=SEG33/IO VLCD1SEG32=SEG32/IO	0	XXXX	1	1	0
	中电阻（中电流）模式	VLCDHSEG35=SEG35/IO VLCD3SEG34=SEG34/IO VLCD2SEG33=SEG33/IO VLCD1SEG32=SEG32/IO	0	XXXX	0	1	0
	大电阻（小电流）模式	VLCDHSEG35=SEG35/IO VLCD3SEG34=SEG34/IO VLCD2SEG33=SEG33/IO VLCD1SEG32=SEG32/IO	0	XXXX	1	0	0
选择外部电容工作模式，内部电阻断路		VLCDHSEG35=DH1 VLCD3SEG34=DH2 VLCD2SEG33=VLCD2 VLCD1SEG32=VLCD1	0	0000	0	0	1

23.9.6 LCD_RAM0~7

偏移地址 0x040~0x5c

复位值：无

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:0	Dx	LCD点输出，显示参考LCD显示模式 0 对应的SEG COM交叉点不亮；1 对应的SEG COM交叉点亮；

23.9.7 LCD_RAM8~F

偏移地址 0x060~0x7C

复位值：无

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								D23	D22	D21	D20	D19	D18	D17	D16
								RW	RW	RW	RW	RW	RW	RW	RW

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:24	Reserved	保留
23:0	Dx	LCD点输出，显示参考LCD显示模式 0 对应的SEG COM交叉点不亮；1 对应的SEG COM交叉点亮；

24 模数转换器（ADC）

24.1 模块简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 12 位转换精度；
- 1M SPS 转换速度；
- 30 路输入通道，包括 26 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、1 路内建 BGR 1.2V 电压、1 路 DAC 内部输出；
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；
- ADC 的电压输入范围：0~Vref；
- 4 种转换模式：单次转换、顺序扫描连续转换、插队扫描连续转换、连续转换累加；
- 输入通道电压阈值监测；
- 软件可配置 ADC 的转换速率；
- 内置信号跟随器，可转换高阻信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

24.2 ADC 框图

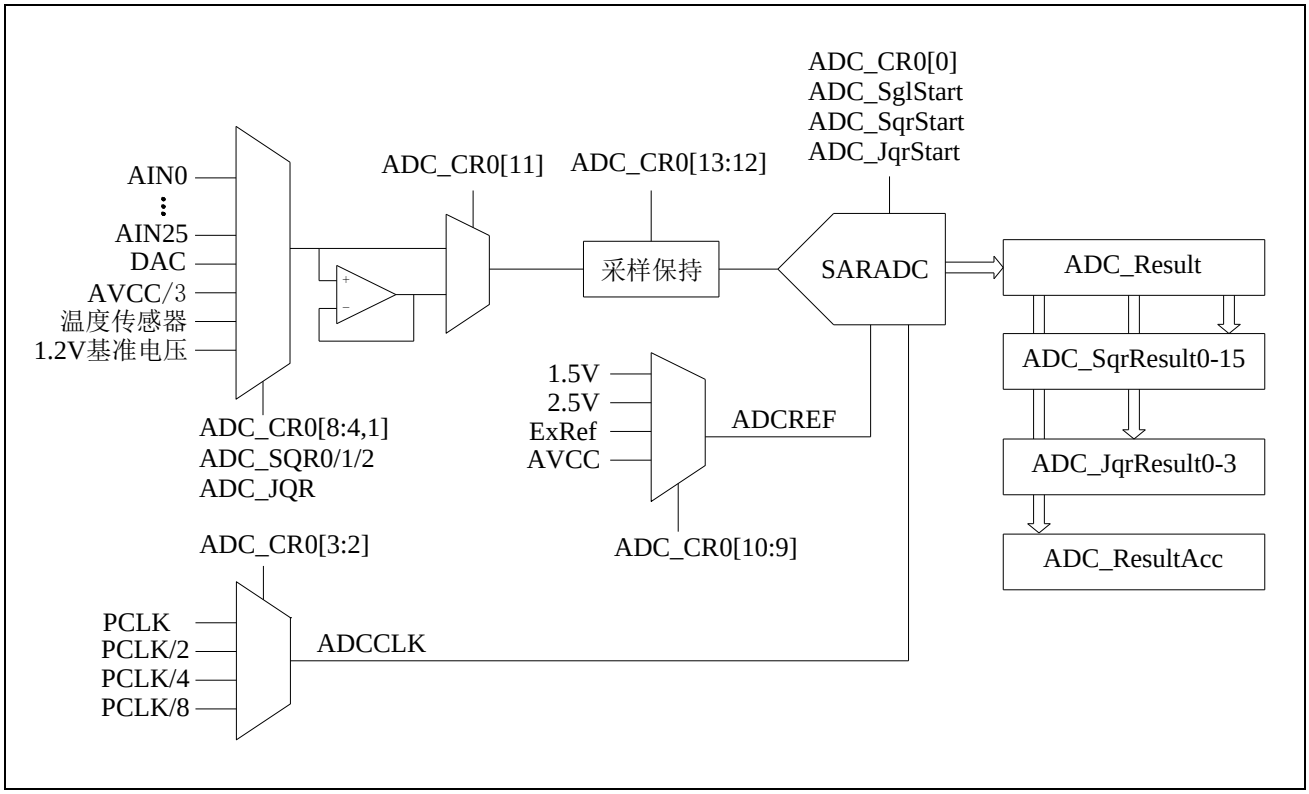


图 24-1 ADC 示意框图

24.3 转换时序及转换速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由转换过程及逐次比较过程组成。其中转换过程需要 4~12 个 ADCCLK，由 ADC_CR0.SAM 配置；逐次比较过程需要 16 个 ADCCLK。所以，一次 ADC 转换共需要 20~28 个 ADCCLK。

ADC 转换速度的单位为 SPS，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADCCLK 的频率 / 一次 ADC 转换所需要的 ADCCLK 的个数。

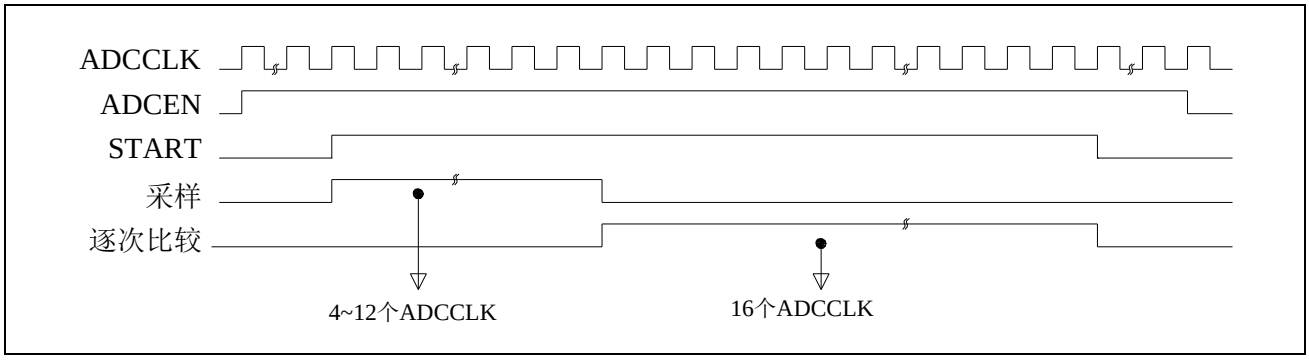


图 24-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关，最高转换速度如下表所示：

ADC 参考电压	AVCC 电压	最高转换速度	最大 ADCCLK 频率
内部 1.5V	1.8~5.5V	200K SPS	4MHz
内部 2.5V	2.8~5.5V	200K SPS	4MHz
AVCC / ExRef	1.8~2.4V	200K SPS	4MHz
AVCC / ExRef	2.4~2.7V	500K SPS	16MHz
AVCC / ExRef	2.7~5.5V	1M SPS	24MHz

24.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 30 路 ADC 通道进行转换。该模式既可通过设置 ADC_SglStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_IFR.SGLIF 位会自动置 1，转换结果保存在 ADC_Result 寄存器中。

通过 ADC_SglStart.Start 位启动 ADC 单次转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_CR0.SGLMux，选择待转换的通道。

Step11：设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step12：设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step13：等待 ADC_IFR. SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step14：如需对其它通道进行转换，重复执行 Step10~Step13。

Step15：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 单次转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_CR0.SGLMux，选择待转换的通道。

Step13：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step14：配置 ADC_ExtTrigger0，选择外部触发条件。

Step15：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step16：如需对其它通道进行转换，重复执行 Step12~Step15。

Step17：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

24.5 扫描转换模式

扫描转换模式分为顺序扫描转换和插队扫描转换两种模式。两种模式各自单独工作时，均可连续对多个通道进行多次转换；当两种模式同时工作时，则优先对插队扫描配置的通道进行转换。

24.5.1 顺序扫描转换模式

顺序扫描转换模式最多可进行 16 次连续转换，转换的总次数由 ADC_SQR2.CNT 进行配置；可配置所有 30 个通道进行转换，待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_SqrResultx~ADC_SqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 8 次转换的顺序扫描转换过程。其中顺序扫描转换通道 7，4，1 配置为 AIN1，转换通道 6，3，0 配置为 AIN0，转换通道 5，2 配置为 AIN5。ADC_SqrStart.Start 置 1 后，ADC 模块会依次对顺序扫描转换通道 7~0 进行转换。

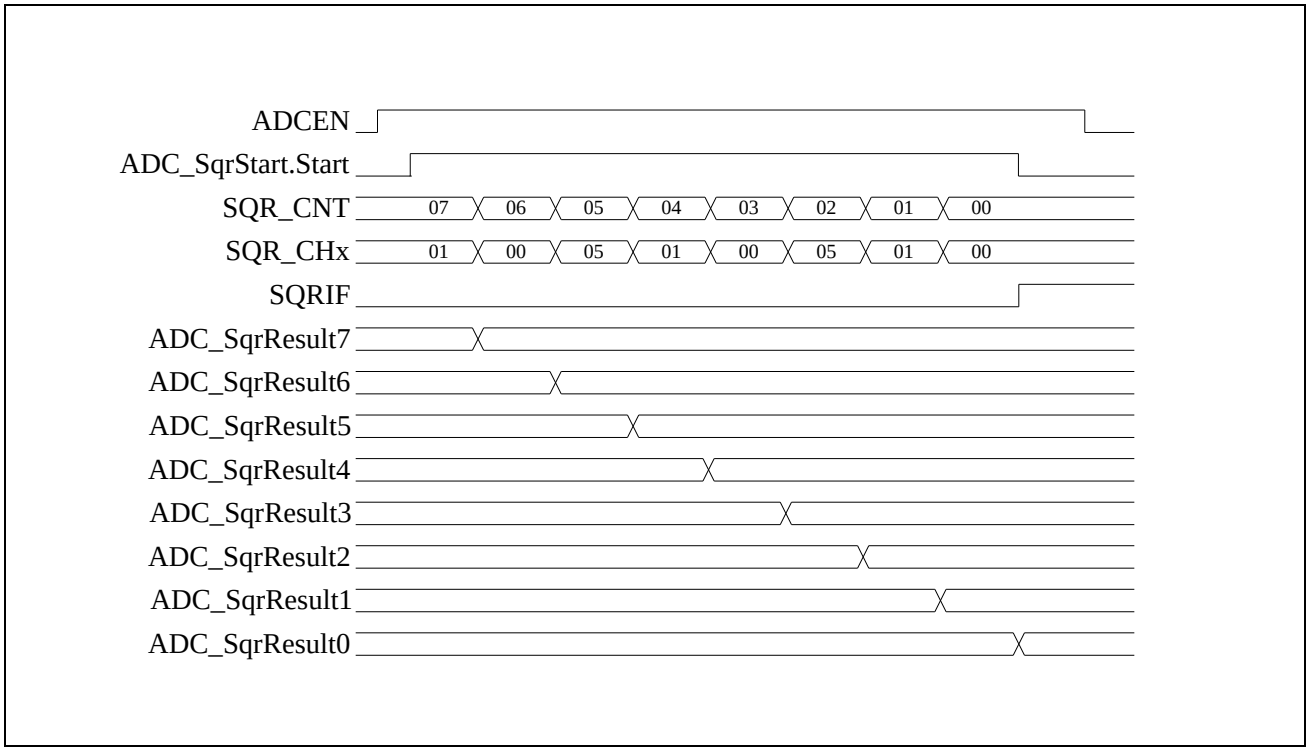


图 24-3 ADC 顺序扫描转换过程示例

通过 ADC_SqrStart.Start 位启动 ADC 顺序扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step11：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step9 中配置的转换通道数保持一致。

Step12：设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。

Step13：设置 ADC_SqrStart.Start 为 1，启动 ADC 顺序扫描转换。

Step14：等待 ADC_IFR.SQRIF 变为 1，读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step15：如需对其它通道进行转换，重复执行 Step10~Step14。

Step16：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 顺序扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step13：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step11 中配置的转换通道数保持一致。

Step14：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step15：配置 ADC_ExtTrigger0，选择外部触发条件。

Step16：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step17：如需对其它通道进行转换，重复执行 Step12~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

24.5.2 插队扫描转换模式

插队扫描转换模式最多可进行 4 次连续转换，转换的总次数由 ADC_JQR.CNT 进行配置；可配置所有 30 个通道进行转换，待转换通道由 ADC_JQR.CHxMux 进行配置。该模式既可通过设置 ADC_JqrStart.Start 位启动也可通过设置 ADC_ExtTrigger1 的外部触发启动。启动转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.JQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_JqrResultx~ ADC_JqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 4 次转换的插队扫描转换的过程。其中插队扫描转换通道 3，

2，1，0 分别设置为 AIN5，AIN0，AIN1，AIN5。ADC_JqrStart.Start 置 1 后，ADC 模块会依次对插队扫描转换通道 3~0 进行转换。

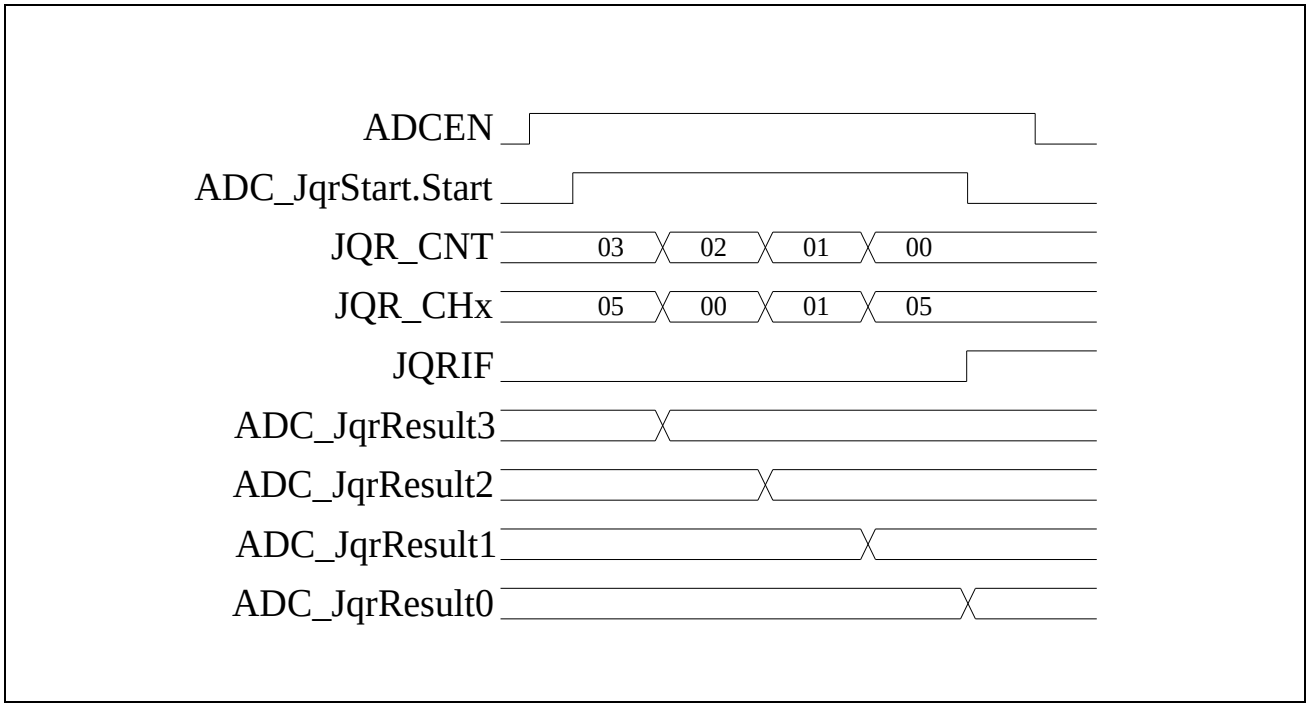


图 24-4 ADC 插队扫描转换过程示例

通过 ADC_JqrStart.Start 位启动 ADC 插队扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10：配置 ADC_JQR.CHxMux，选择插队扫描转换通道。

Step11：配置 ADC_JQR.CNT，选择插队扫描转换的总转换次数。

注：总转换次数需与 Step9 中配置的转换通道数保持一致。

Step12：设置 ADC_ICR.JQRIC 为 0，清除 ADC_IFR.JQRIF 标志。

Step13：设置 ADC_JqrStart.Start 为 1，启动 ADC 插队扫描转换。

Step14：等待 ADC_IFR.JQRIF 变为 1，读取 ADC_JqrResultx~ADC_JqrResult0 寄存器以获取相应通道的转换结果。

Step15：如需对其它通道进行转换，重复执行 Step10~Step14。

Step16：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 插队扫描转换操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step10：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step11：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step12：配置 ADC_JQR.CHxMux，选择插队扫描转换通道。

Step13：配置 ADC_JQR.CNT，选择插队扫描转换的总转换次数。

注：总转换次数需与 Step11 中配置的转换通道数保持一致。

Step14：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step15：配置 ADC_ExtTrigger1，选择外部触发条件。

Step16：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC

中断服务程序中读取 ADC_JqrResultx~ ADC_JqrResult0 寄存器以获取相应通道的转换结果。

Step17：如需对其它通道进行转换，重复执行 Step12~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

插队扫描转换的执行优先级高于顺序扫描转换，当顺序扫描转换正在进行时，如果插队扫描转换启动，则顺序扫描在完成当前通道的转换后暂停，等到插队扫描转换完成后，再继续执行剩下的通道转换。下图演示了对 AIN0，AIN1，AIN5，AIN8 进行顺序扫描转换时，启动对 AIN2，AIN6 进行插队扫描转换的过程。其中顺序扫描转换通道 3，2，1，0 分别设置为 AIN1，AIN0，AIN5，AIN8，插队扫描转换通道 1，0 分别设置为 AIN6，AIN2。在顺序扫描进行 AIN0 转换的过程中，启动了插队扫描，顺序扫描会将当前 AIN0 的转换完成然后暂停，等到插队扫描完成 AIN6 和 AIN2 的转换后，顺序扫描再进行 AIN5 和 AIN8 的转换。

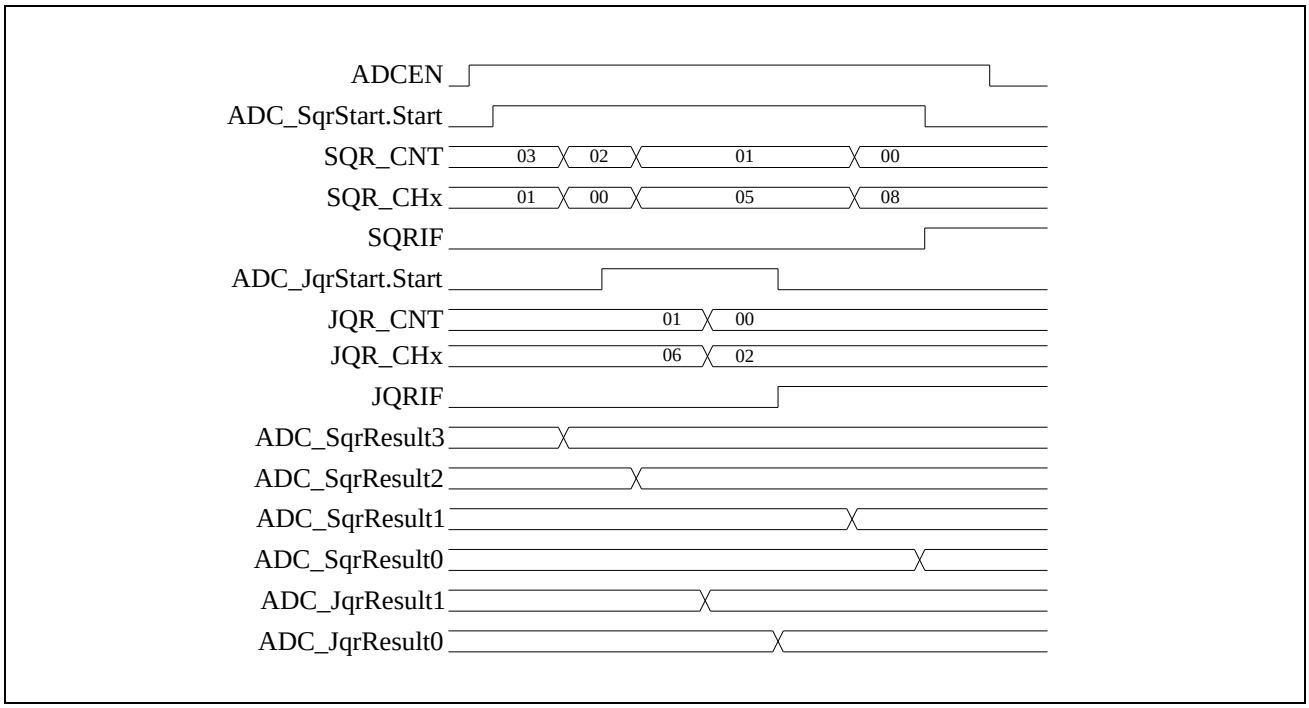


图 24-5 ADC 顺序扫描过程中进行插队扫描示例

24.5.3 扫描转换触发 DMA 读取

顺序扫描和插队扫描转换完成后，可自动触发 DMA 读取转换结果。顺序扫描模式下，通过配置 ADC_CR1. DmaSqr 为 1 使能该功能；插队扫描模式下，通过配置 ADC_CR1. DmaJqr 为 1 使能该功能。

24.6 连续转换累加模式

在连续转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可配置所有 30 个通道进行转换。ADC 转换的总次数由 ADC_SQR2.CNT 进行配置；待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动连续转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果的累加值保存在 ADC_ResultAcc 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。顺序扫描转换通道 9，6，3，0 配置为 AIN0，转换通道 8，5，2 配置为 AIN1，转换通道 7，4，1 配置为 AIN5。ADC_SqrStart.Start 置 1 后，ADC 模块会按照顺序扫描转换通道配置依次进行转换，直到 SQR_CNT 的计数值变为 0。每次转换完成时，ADC_ResultAcc 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

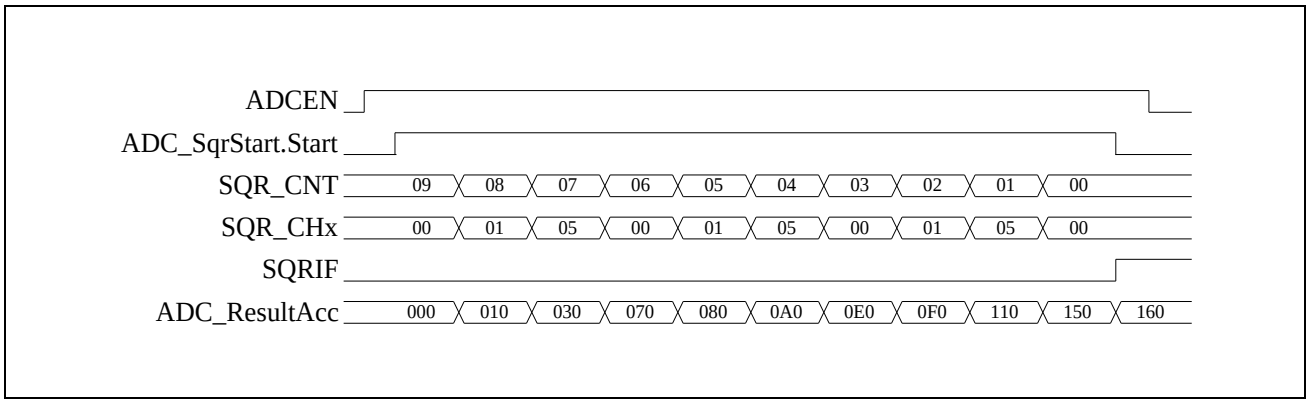


图 24-6 ADC 连续转换累加过程示例

通过 ADC_SqrStart.Start 位启动 ADC 连续转换累加操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR1.RAccEn 为 1，使能 ADC 转换自动累加功能。

Step8：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step9：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step10：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step11：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step12：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step10 中配置的转换通道数保持一致。

Step13：设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。

Step14：设置 ADC_CR1.RAccClr 为 0，清零 ADC_ResultAcc 寄存器。

Step15：设置 ADC_SqrStart.Start 为 1，启动 ADC 顺序扫描转换。

Step16：等待 ADC_IFR.SQRIF 变为 1，读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step17：如需对其它通道进行转换，重复执行 Step11~Step16。

Step18：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 连续转换累加操作流程：

Step1：配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3：设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step6：设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7：设置 ADC_CR0.IE 为 1，使能 ADC 中断。

Step8：使能 NVIC 中断向量表中的 ADC 中断。

Step9：设置 ADC_CR1.RAccEn 为 1，使能 ADC 转换自动累加功能。

Step10：配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step11：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step12：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step13：配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step14：配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step12 中配置的转换通道数保持一致。

Step15：设置 ADC_IFR 为 0x0，清除 ADC 中断标志。

Step16：设置 ADC_CR1.RAccClr 为 0，清零 ADC_ResultAcc 寄存器。

Step17：配置 ADC_ExtTrigger0，选择外部触发条件。

Step18：当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step19：如需对其它通道进行转换，重复执行 Step13~Step18。

Step20：设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

24.7 ADC 转换外部触发源

ADC 转换既可通过软件配置启动，也可通过外部触发启动。

配置 ADC_ExtTrigger0 寄存器可设置 ADC 单次转换或顺序扫描转换的外部触发源。

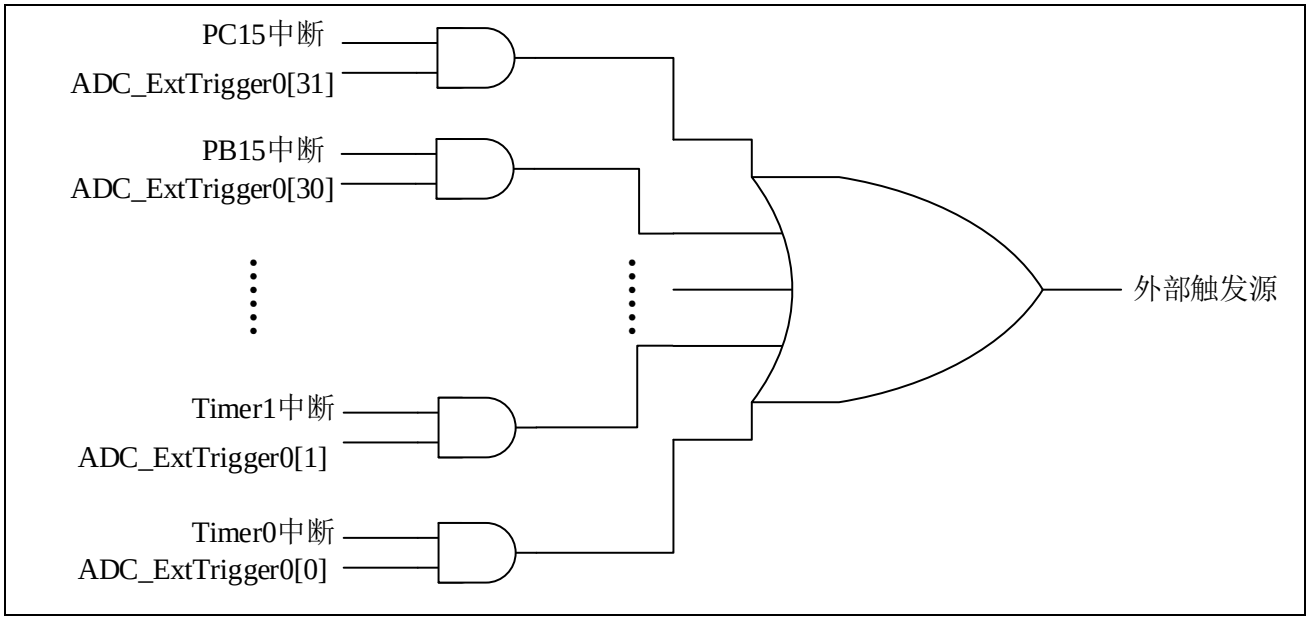


图 24-7 ADC 单次转换或顺序扫描转换外部触发源示意图

配置 ADC_ExtTrigger1 寄存器可设置 ADC 插队扫描转换的外部触发源。

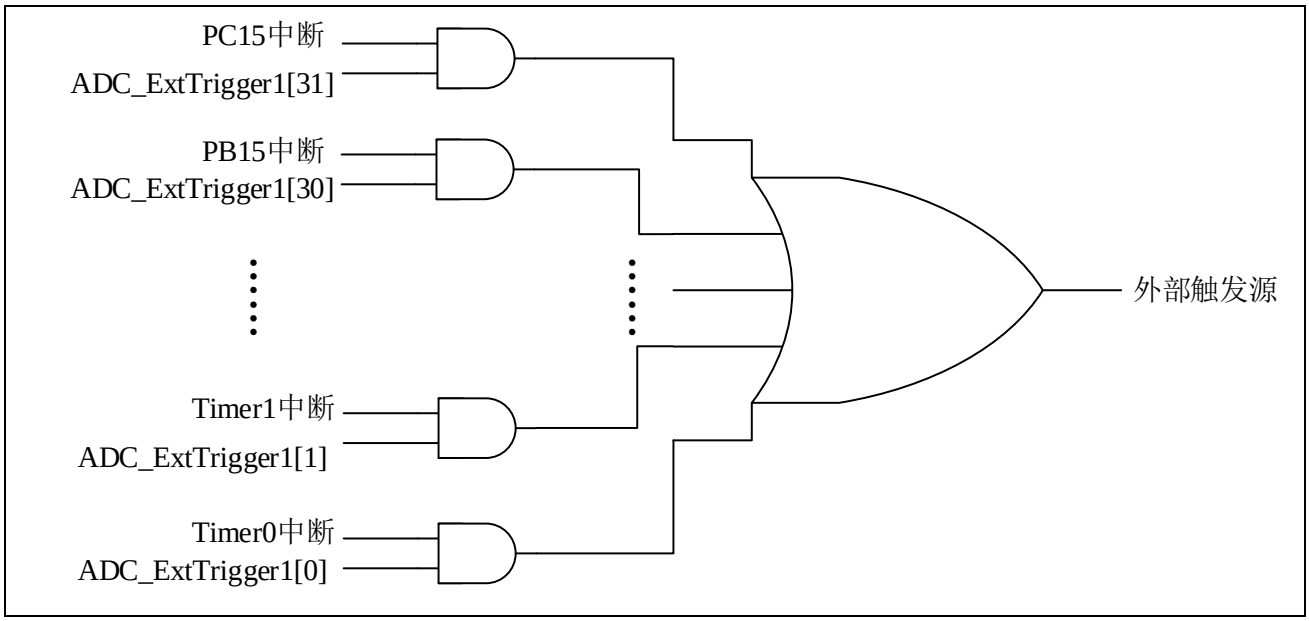


图 24-8 ADC 插队扫描转换外部触发源示意图

24.8 ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 HtCmp、LtCmp、RegCmp 置 1。该功能可实现对模拟量的自动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序界入。监测通道选择通过 ADC_CR1.ThCh 进行配置。

上阈值比较：当 ADC 转换结果位于[ADC_HT，4095]区间内则 ADC_IFR.HTIF 置 1；向 ADC_ICR.HTIC 写入 0 则清零 ADC_IFR.HTIF。

下阈值比较：当 ADC 转换结果位于[0,ADC_LT)区间内则 ADC_IFR.LTIF 置 1；向 ADC_ICR.LTIC 写入 0 则清零 ADC_IFR.LTIF。

区间值比较：当 ADC 转换结果位于[ADC_LT，ADC_HT)区间内则 ADC_IFR.REGIF 置 1；向 ADC_ICR.REGIC 写入 0 则清零 ADC_IFR.REGIF。

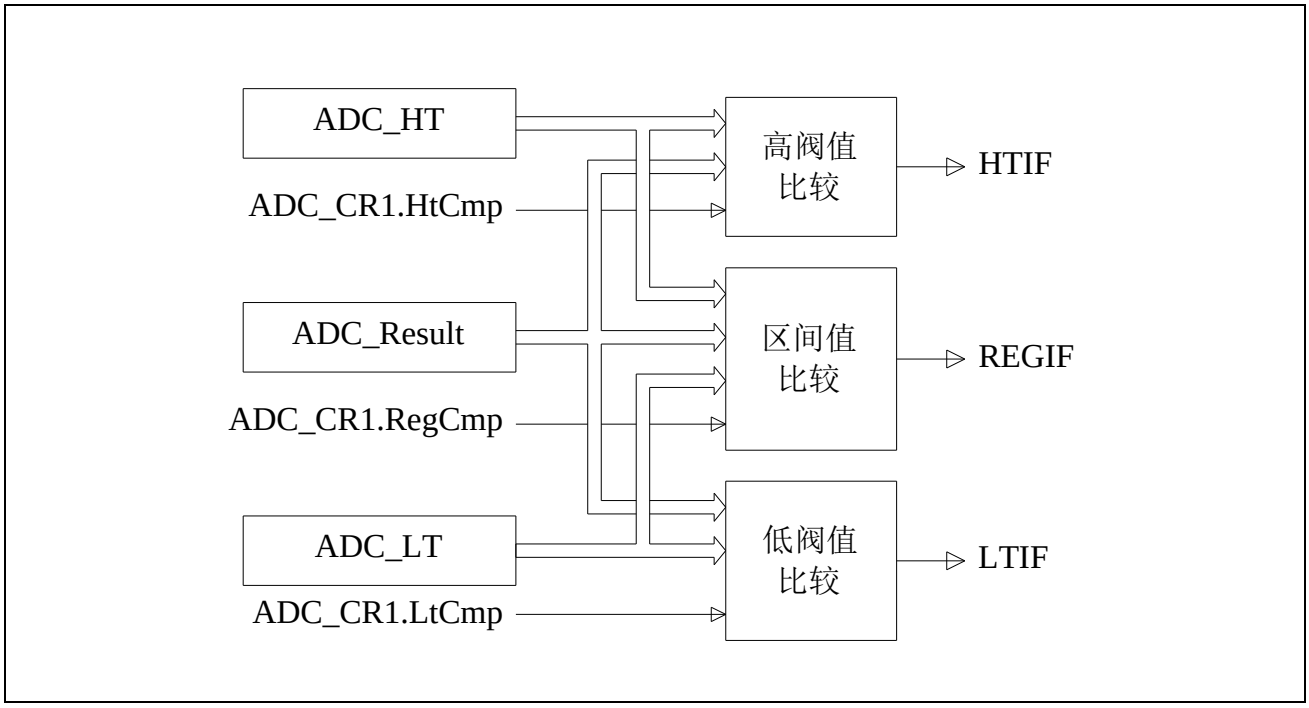


图 28-9

24.9 ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能
ADC 插队扫描转换完成	ADC_IFR.JQRIF	ADC_CR0.IE
ADC 顺序扫描转换完成	ADC_IFR.SQRIF	
ADC 转换结果位于区间值区域（大于等于下阈值小于上阈值）	ADC_IFR.REGIF	
ADC 转换结果位于上阈值区域(大于等于上阈值)	ADC_IFR.HTIF	
ADC 转换结果比较下阈值区域（小于下阈值）	ADC_IFR.LTIF	
ADC 单次转换完成	ADC_IFR.SGLIF	

24.10 使用温度传感器测量环境温度

温度传感器的输出电压会随环境温度的改变而变化，故根据温度传感器的输出电压即可计算出相应的环境温度。当 ADC 模块的测量通道选择温度传感器的输出电压时，即可测量环境温度。

计算公式如下：

环境温度 = 25 + 0.0839 × Vref × (AdcValue – Trim)

其中：Vref 为当前 ADC 模块的参考电压，取值为 1.5 或 2.5。

AdcValue 为 ADC 模块测量温度传感器输出电压的结果，取值为 0~4095。

Trim 为 16Bits 的校准值，计算时需要从 Flash 存储器中读出，其存放地址详见下表。

ADC 参考电压	校准值存放地址	校准值精度
内部 1.5V	0x00100C34	±3℃
内部 2.5V	0x00100C36	±3℃

计算示例如下：

条件 1：Vref=2.5、AdcValue=0x7E5、Trim=0x76C：

温度 1：25 + 0.0839 × 2.5 × (0x7E5 – 0x76C) = 50℃。

条件 2：Vref=1.5、AdcValue=0x72D、Trim=0x76C：

温度 3：25 + 0.0839 × 1.5 × (0x72D – 0x76C) = 17.2℃。

通过 ADC 测量环境温度操作流程：

Step1：设置 BGR_CR 为 3，使能 BGR 模块和温度传感器模块。

Step2：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step3：延时 20us，等待 ADC 及 BGR 模块启动完成。

Step4：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step5: 设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

Step6：配置 ADC_CR0.Ref，选择 ADC 的参考电压为内部 1.5V 或内部 2.5V。

Step7：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step8：设置 ADC_CR0.SGLMux 为 0x1C，选择待转换的通道为温度传感器的输出。

Step9：设置 ADC_CR0.Buf 为 1，使能输入信号放大器。

Step10：设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step11：设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step12：等待 ADC_IFR.SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step13：设置 ADC_CR0.En 及 BGR_CR 为 0，关闭 ADC 模块、BGR 模块、温度传感器模块。

Step14：读取温度传感器校准值，根据公式计算当前的环境温度。

24.11 ADC 模块寄存器

基地址 0x40002400

寄存器	偏移地址	描述
ADC_CR0	0x004	ADC 基本配置寄存器0
ADC_CR1	0x008	ADC 基本配置寄存器1
ADC_SQR0	0x040	ADC 顺序扫描转换通道配置寄存器0
ADC_SQR1	0x044	ADC 顺序扫描转换通道配置寄存器1
ADC_SQR2	0x048	ADC 顺序扫描转换通道配置寄存器2
ADC_JQR	0x04C	ADC 插队扫描转换通道配置寄存器
ADC_SqrResult0	0x050	ADC 顺序扫描转换通道0转换结果
ADC_SqrResult1	0x054	ADC 顺序扫描转换通道1转换结果
ADC_SqrResult2	0x058	ADC 顺序扫描转换通道2转换结果
ADC_SqrResult3	0x05C	ADC 顺序扫描转换通道3转换结果
ADC_SqrResult4	0x060	ADC 顺序扫描转换通道4转换结果
ADC_SqrResult5	0x064	ADC 顺序扫描转换通道5转换结果
ADC_SqrResult6	0x068	ADC 顺序扫描转换通道6转换结果
ADC_SqrResult7	0x06C	ADC 顺序扫描转换通道7转换结果
ADC_SqrResult8	0x070	ADC 顺序扫描转换通道8转换结果
ADC_SqrResult9	0x074	ADC 顺序扫描转换通道9转换结果
ADC_SqrResult10	0x078	ADC 顺序扫描转换通道10转换结果
ADC_SqrResult11	0x07C	ADC 顺序扫描转换通道11转换结果
ADC_SqrResult12	0x080	ADC 顺序扫描转换通道12转换结果
ADC_SqrResult13	0x084	ADC 顺序扫描转换通道13转换结果
ADC_SqrResult14	0x088	ADC 顺序扫描转换通道14转换结果
ADC_SqrResult15	0x08C	ADC 顺序扫描转换通道15转换结果
ADC_JqrResult0	0x090	ADC 插队扫描转换通道0转换结果
ADC_JqrResult1	0x094	ADC 插队扫描转换通道1转换结果
ADC_JqrResult2	0x098	ADC 插队扫描转换通道2转换结果
ADC_JqrResult3	0x09C	ADC 插队扫描转换通道3转换结果
ADC_Result	0x0A0	ADC 转换结果
ADC_ResultAcc	0x0A4	ADC 转换结果累加值
ADC_HT	0x0A8	ADC 比较上阈值
ADC_LT	0x0AC	ADC 比较下阈值
ADC_IFR	0x0B0	ADC 中断标志寄存器
ADC_ICR	0x0B4	ADC 中断清除寄存器
ADC_ExtTrigger0	0x0B8	ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器
ADC_ExtTrigger1	0x0BC	ADC 插队扫描转换外部中断触发源配置寄存器
ADC_SglStart	0x0C0	ADC单次转换启动控制寄存器

ADC_SqrStart	0x0C4	ADC顺序扫描转换启动控制寄存器
ADC_JqrStart	0x0C8	ADC插队扫描转换启动控制寄存器
ADC_allstart	0x0CC	ADC一直转换启动停止控制寄存器

表 24-1 ADC 寄存器

24.11.1 ADC 基本配置寄存器 0（ADC_CR0）

偏移地址 0x004

复位值 0x000067F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	InRefEn	SAM		Buf	Ref		SGLMux					CkDiv		RES.	En
RW	RW	RW		RW	RW		RW					RW			RW

位	标记	功能描述
31:16	Reserved	保留
15	IE	ADC中断控制 1：使能中断 0：禁止中断
14	InRefEn	ADC内部参考电压使能 1：使能内部参考电压 0：禁止内部参考电压
13:12	SAM	ADC采样周期选择 00：4个转换周期 01：6个转换周期 10：8个转换周期 11：12个转换周期
11	Buf	内置跟随器使能控制 0：关闭跟随器，外部输入信号与ADC直接相连。 1：打开跟随器，外部输入信号通过跟随器后与ADC相连，用于测量高阻信号。 注1：使能该功能时，最大速率为200Ksps 注2：以下条件必须使能该功能 待转换的信号输出阻抗较高 待转换的信号为：1/3VCC、内置温度传感器输出电压、内部基准1.2V输出电压、DAC输出
10:9	Ref	ADC参考电压选择 00：内部1.5V 01：内部2.5V 10：外部参考电压ExRef（PB01）

		11：AVCC电压
8:4	SGLMux	单次转换模式转换通道选择 00000：AIN0（PA00） 00001：AIN1（PA01） 00010：AIN2（PA02） 00011：AIN3（PA03） 00100：AIN4（PA04） 00101：AIN5（PA05） 00110：AIN6（PA06） 00111：AIN7（PA07） 01000：AIN8（PB00） 01001：AIN9（PB01） 01010：AIN10（PC00） 01011：AIN11（PC01） 01100：AIN12（PC02） 01101：AIN13（PC03） 01110：AIN14（PC04） 01111：AIN15（PC05） 10000：AIN16（PB02） 10001：AIN17（PB10） 10010：AIN18（PB11） 10011：AIN19（PB12） 10100：AIN20（PB13） 10101：AIN21（PB14） 10110：AIN22（PB15） 10111：AIN23（PE15） 11000：AIN24（PE14） 11001：AIN25（PE13）
		11010：DAC输出　　注：ADC_CR0.Buf必须为1 11011：1/3VCC　　注：ADC_CR0.Buf必须为1 11100：内置温度传感器输出电压　　注：ADC_CR0.Buf必须为1 11101：内部基准1.2V输出电压　　注：ADC_CR0.Buf必须为1
3:2	CkDiv	ADC时钟选择 00：PCLK时钟 01：PCLK时钟2分频 10：PCLK时钟4分频 11：PCLK时钟8分频
1	Reserved	保留
0	En	ADC使能控制 1：使能ADC 0：禁止ADC

24.11.2 ADC 基本配置寄存器 1（ADC_CR1）

偏移地址 0x008

复位值 0x00008000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAccClr	RegCmp	HtCmp	LtCmp	RAccEn	Mode	DmaJqr	DmaSqr	ThCh					Align	Reserved	
WO	RW	RW	RW	RW	RW	RW	RW	RW					RW		

位	标记	功能描述
31:16	Reserved	保留
15	RAccClr	ADC转换结果累加寄存器清零 1：无作用； 0：ADC转换结果累加寄存器（ADC_ResultAcc）清零。
14	RegCmp	ADC区间比较控制 1：使能区间比较 0：禁止区间比较
13	HtCmp	ADC高阈值比较控制 1：使能高阈值比较 0：禁止高阈值比较
12	LtCmp	ADC低阈值比较控制 1：使能低阈值比较 0：禁止低阈值比较
11	RAccEn	ADC转换结果自动累加控制 1：使能ADC转换结果自动累加功能 0：禁止ADC转换结果自动累加功能
10	Mode	ADC转换模式选择 1：扫描转换模式 0：单次转换模式
9	DmaJqr	插队扫描触发DMA读取控制 1：使能插队扫描转换触发DMA读取 0：禁止插队扫描转换触发DMA读取
8	DmaSqr	顺序扫描触发DMA读取控制 1：使能顺序扫描转换触发DMA读取

		0：禁止顺序扫描转换触发DMA读取
7:3	ThCh	阈值比较通道选择 00000：选择通道0进行阈值比较 00001：选择通道1进行阈值比较 00010：选择通道2进行阈值比较 …… 11101：选择通道 29 进行阈值比较
2	Align	转换结果对齐控制 1：转换结果16Bits左对齐存储 0：转换结果16Bits右对齐存储
1:0	Reserved	保留

24.11.3 ADC 顺序扫描转换通道配置寄存器 0（ADC_SQR0）

偏移地址 0x040

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH5Mux					CH4Mux					CH3Mux			
		RW					RW					RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2Mux					CH1Mux					CH0Mux				
	RW					RW					RW				

位	标记	功能描述
31:30	Reserved	保留
29:25	CH5Mux	顺序扫描转换通道5选择，设置参见ADC_CR0.SGLMux
24:20	CH4Mux	顺序扫描转换通道4选择，设置参见ADC_CR0.SGLMux
19:15	CH3Mux	顺序扫描转换通道3选择，设置参见ADC_CR0.SGLMux
14:10	CH2Mux	顺序扫描转换通道2选择，设置参见ADC_CR0.SGLMux
9:5	CH1Mux	顺序扫描转换通道1选择，设置参见ADC_CR0.SGLMux
4:0	CH0Mux	顺序扫描转换通道0选择，设置参见ADC_CR0.SGLMux

24.11.4 ADC 顺序扫描转换通道配置寄存器 1（ADC_SQR1）

偏移地址 0x044

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH11Mux					CH10Mux					CH9Mux			
		RW					RW					RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH8Mux					CH7Mux					CH6Mux				
	RW					RW					RW				

位	标记	功能描述
31:30	Reserved	保留
29:25	CH11Mux	顺序扫描转换通道11选择，设置参见ADC_CR0.SGLMux
24:20	CH10Mux	顺序扫描转换通道10选择，设置参见ADC_CR0.SGLMux
19:15	CH9Mux	顺序扫描转换通道9选择，设置参见ADC_CR0.SGLMux
14:10	CH8Mux	顺序扫描转换通道8选择，设置参见ADC_CR0.SGLMux
9:5	CH7Mux	顺序扫描转换通道7选择，设置参见ADC_CR0.SGLMux
4:0	CH6Mux	顺序扫描转换通道6选择，设置参见ADC_CR0.SGLMux

24.11.5 ADC 顺序扫描转换通道配置寄存器 2（ADC_SQR2）

偏移地址 0x048

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CNT				CH15Mux			
								RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH14Mux					CH13Mux					CH12Mux				
	RW					RW					RW				

位	标记	功能描述
31:24	Reserved	保留
23:20	CNT	顺序扫描转换次数
19:15	CH15Mux	顺序扫描转换通道15选择，设置参见ADC_CR0.SGLMux
14:10	CH14Mux	顺序扫描转换通道14选择，设置参见ADC_CR0.SGLMux
9:5	CH13Mux	顺序扫描转换通道13选择，设置参见ADC_CR0.SGLMux
4:0	CH12Mux	顺序扫描转换通道12选择，设置参见ADC_CR0.SGLMux

24.11.6 ADC 插队扫描转换通道配置寄存器（ADC_JQR）

偏移地址 0x04C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CNT		CH3Mux			
										RW		RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH2Mux					CH1Mux					CH0Mux				
	RW					RW					RW				

位	标记	功能描述
31:24	Reserved	保留
21:20	CNT	插队扫描转换次数
19:15	CH3Mux	插队扫描转换通道3选择，设置参见ADC_CR0.SGLMux
14:10	CH2Mux	插队扫描转换通道2选择，设置参见ADC_CR0.SGLMux
9:5	CH1Mux	插队扫描转换通道1选择，设置参见ADC_CR0.SGLMux
4:0	CH0Mux	插队扫描转换通道0选择，设置参见ADC_CR0.SGLMux

24.11.7 ADC 顺序扫描转换通道 x 转换结果（ADC_SqrResult0 - 15）

偏移地址 0x050 ~ 0x8C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 顺序扫描转换通道x转换结果

24.11.8 ADC 插队扫描转换通道 x 转换结果（ADC_JqrResult0 - 3）

偏移地址 0x090 ~ 0x9C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 插队扫描转换通道x转换结果

24.11.9 ADC 转换结果（ADC_Result）

偏移地址 0x0A0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC转换结果

24.11.10 ADC 转换结果累加值（ADC_ResultAcc）

偏移地址 0x0A4

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ResultAcc[19:16]			
												RO			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ResultAcc[15:0]															
RO															

位	标记	功能描述
31:20	Reserved	保留
19:0	ResultAcc	ADC转换累加值

24.11.11 ADC 比较上阈值（ADC_HT）

偏移地址 0x0A8

复位值 0x00000FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:12	Reserved	保留
11:0	HT	ADC转换结果比较上阈值

24.11.12 ADC 比较下阈值（ADC_LT）

偏移地址 0x0AC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				LT											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	LT	ADC转换结果比较下阈值

24.11.13 ADC 中断标志寄存器（ADC_IFR）

偏移地址 0x0B0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										JQRIF	SQRIF	REGIF	HHT_INTF	LLT_INTF	S_INTF
										RO	RO	RO	RO	RO	RO

位	标记	功能描述
31:6	Reserved	保留
5	JQRIF	ADC插队扫描转换完成标志 1：ADC插队扫描转换完成 0：ADC插队扫描转换未完成
4	SQRIF	ADC顺序扫描转换完成标志 1：ADC顺序扫描转换完成 0：ADC顺序扫描转换未完成
3	REGIF	ADC转换结果比较区间标志 1：ADC转换结果位于[ADC_LT ， ADC_HT)区间内 0：ADC转换结果位于[ADC_LT ， ADC_HT)区间外
2	HTIF	ADC转换结果比较上阈值标志 1：ADC转换结果位于[ADC_HT，4095]区间内 0：ADC转换结果位于[ADC_HT，4095]区间外
1	LTIF	ADC转换结果比较下阈值标志 1：ADC转换结果位于[0，ADC_LT)区间内 0：ADC转换结果位于[0，ADC_LT)区间外
0	SGLIF	ADC单次转换完成标志 1：ADC单次转换完成 0：ADC单次转换未完成

24.11.14 ADC 中断清除寄存器（ADC_ICR）

偏移地址 0x0B4

复位值 0x0000003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										JQRIC	SQRIC	REGIC	HHT_INTC	LLT_INTC	S_INTC
										R1W0	R1W0	R1W0	R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	保留
5	JQRIC	写0清除ADC插队扫描转换完成标志 写1无作用
4	SQRIC	写0清除ADC顺序扫描转换完成标志 写1无作用
3	REGIC	写0清除ADC转换结果比较区间标志 写1无作用
2	HTIC	写0清除ADC转换结果比较上阈值 写1无作用
1	LTIC	写0清除ADC转换结果比较下阈值标志 写1无作用
0	SGLIC	写0清除ADC单次转换完成标志 写1无作用

24.11.15 ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger0）

偏移地址 0x0B8

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0	PCA	Res.	VC1	VC0	Reserved		UART1	UART0	TIM6	TIM5	TIM4	TIM3	TIM2	TIM1	TIM0
RW	RW		RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换
29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI1	SPI1中断触发ADC转换
15	SPI0	SPI0中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	Reserved	保留
12	VC1	VC1中断触发ADC转换
11	VC0	VC0中断触发ADC转换
10:9	Reserved	保留

8	UART1	UART1中断触发ADC转换
7	UART0	UART0中断触发ADC转换
6	TIM6	Timer6中断触发ADC转换
5	TIM5	Timer5中断触发ADC转换
4	TIM4	Timer4中断触发ADC转换
3	TIM3	Timer3中断触发ADC转换
2	TIM2	Timer2中断触发ADC转换
1	TIM1	Timer1中断触发ADC转换
0	TIM0	Timer0中断触发ADC转换

Notes:

1) TIM4/5/6 中断触发 ADC 自动转换，除了需要使能 TIM4/5/6 的相应中断外，还需要配置 Advanced Timer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。

2) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。

24.11.16 ADC 插队扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger1）

偏移地址 0x0BC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0	PCA	Res.	VC1	VC0	Reserved		UART1	UART0	TIM6	TIM5	TIM4	TIM3	TIM2	TIM1	TIM0
RW	RW		RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换
29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI1	SPI1中断触发ADC转换
15	SPI0	SPI0中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	Reserved	保留
12	VC1	VC1中断触发ADC转换
11	VC0	VC0中断触发ADC转换
10:9	Reserved	保留
8	UART1	UART1中断触发ADC转换
7	UART0	UART0中断触发ADC转换

6	TIM6	Timer6中断触发ADC转换
5	TIM5	Timer5中断触发ADC转换
4	TIM4	Timer4中断触发ADC转换
3	TIM3	Timer3中断触发ADC转换
2	TIM2	Timer2中断触发ADC转换
1	TIM1	Timer1中断触发ADC转换
0	TIM0	Timer0中断触发ADC转换

Notes:

1) TIM4/5/6 中断触发 ADC 自动转换，除了需要使能 TIM4/5/6 的相应中断外，还需要配置 Advanced Timer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。

2) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。

24.11.17 ADC 单次转换启动控制寄存器（ADC_SglStart）

偏移地址 0x0C0

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC单次转换启动控制 1：启动ADC单次转换 0：停止ADC单次转换

24.11.18 ADC 顺序扫描转换启动控制寄存器（ADC_SqrStart）

偏移地址 0x0C4

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC顺序扫描转换启动控制 1：启动ADC顺序扫描转换 0：停止ADC顺序扫描转换

24.11.19 ADC 插队扫描转换启动控制寄存器（ADC_JqrStart）

偏移地址 0x0C8

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1：启动ADC插队扫描转换 0：停止ADC插队扫描转换

24.11.20 ADC 一直转换启动停止控制寄存器（ADC_allStart）

偏移地址 0x0CC

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1：启动ADC一直转换 0：停止ADC一直转换,需要等待到当次转换结束 读0无转换；1正在转换

25 数模转换器（DAC）

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，在 12 位模式下，数据可以采用左对齐或右对齐方式。DAC 可以与 DMA 控制器配合使用，可使用与 ADC 共享的外部输入参考电压 ExRef 以获得更精确的转换结果。DAC 的输出可以选择使用内部缓冲器以实现更高的电流驱动。

25.1 DAC 特性

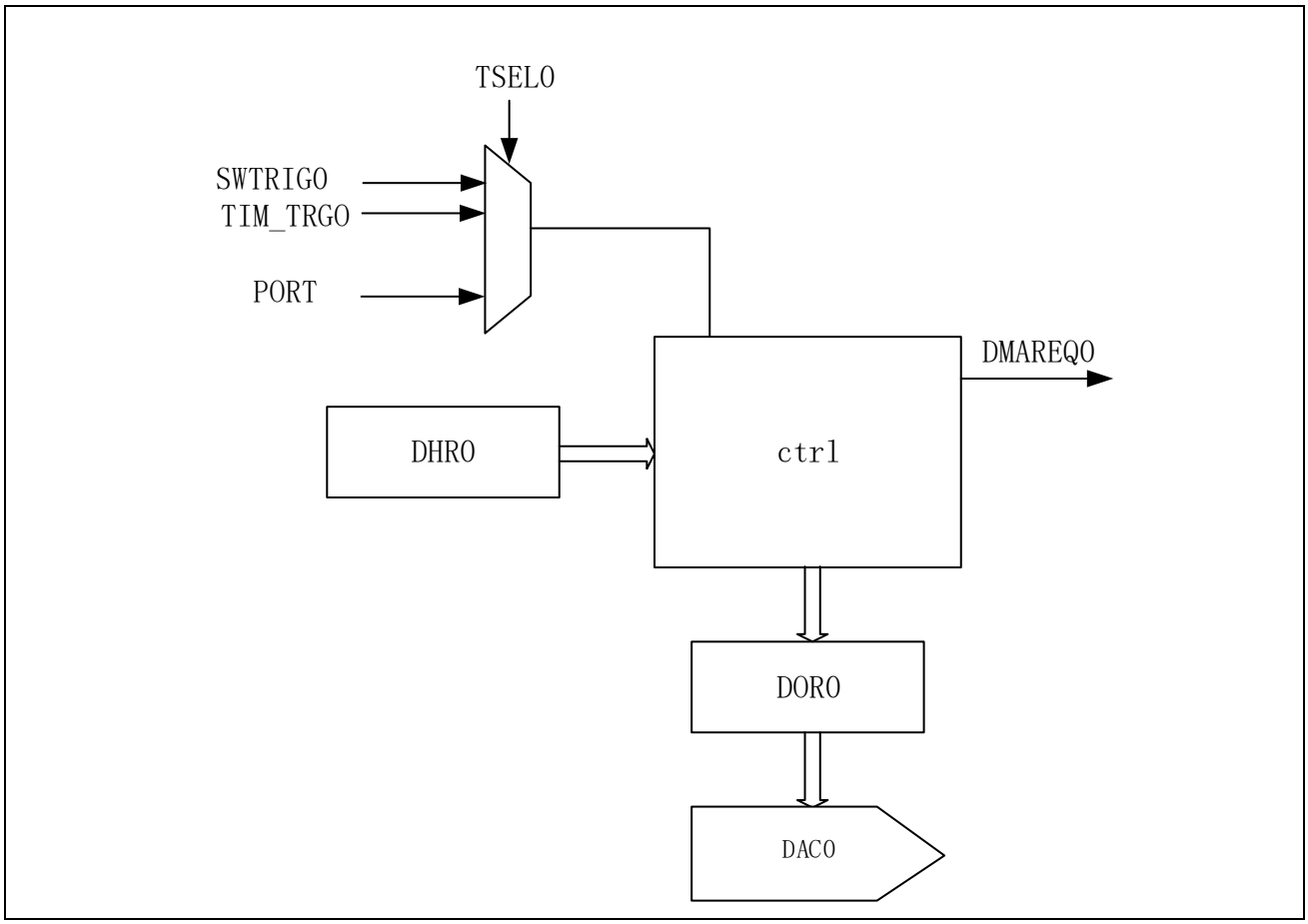
器件集成了 1 个 DAC 转换器，具有一个输出通道：DAC_OUT。

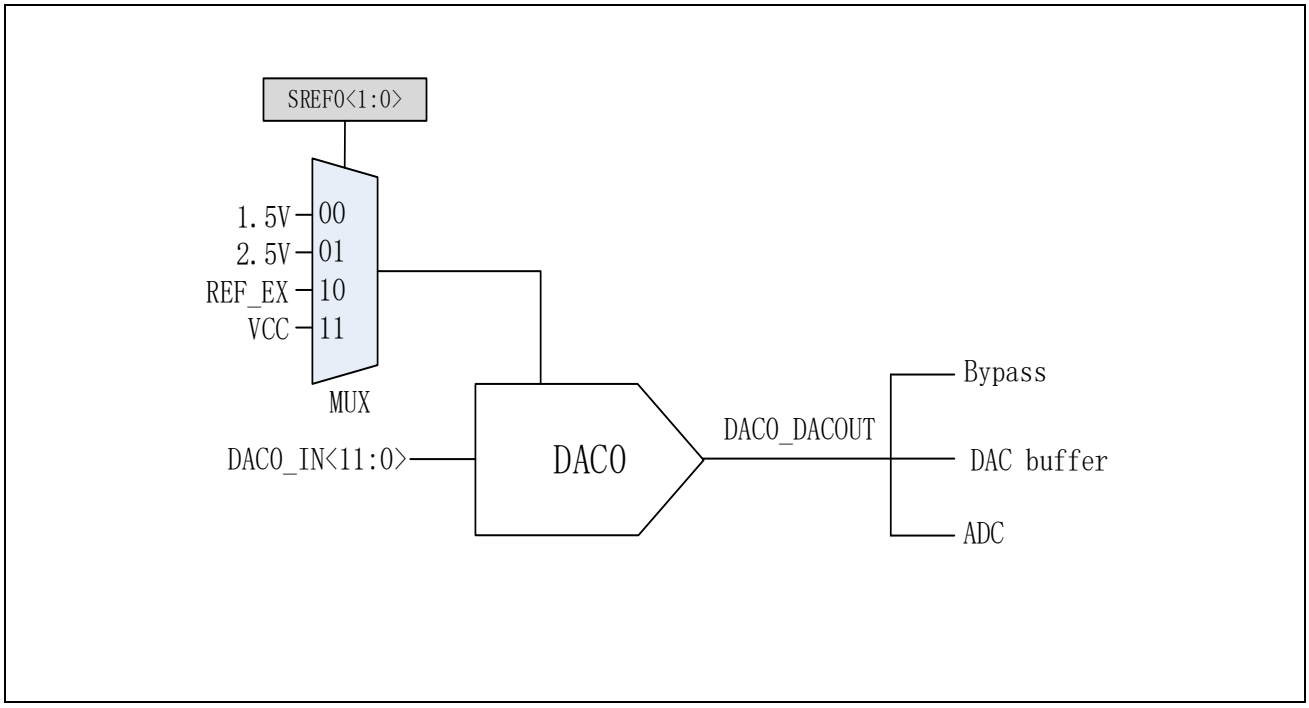
DAC 主要特性如下：

- 一个数据保持寄存器
- 12 位模式下数据采用左对齐或右对齐
- 生成噪声波
- 生成三角波
- DMA 功能（包括下溢检测）
- 通过外部触发信号进行转换
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5V 参考电压、内置 2.5V 参考电压

25.2 功能说明

25.2.1 框图





25.2.2 DAC 输出缓冲器使能

DAC 可以使用输出缓冲器，用来降低输出阻抗并在不增加外部运算放大器的情况下直接驱动外部负载。关于输出缓冲器的设置请参考<运算放大器 OPA>章节。

通过 DAC_CR 寄存器中的 BOFF0 位，可使能或禁止 DAC 通道输出缓冲器。

25.2.3 DAC 通道使能

将 DAC_CR 寄存器中的 EN0 位置 1，即可使能 DAC 通道。经过一段启动时间 t_{WAKEUP} 后，DAC 通道被真正使能。

EN0 位只会使能模拟 DAC 通道。即使 EN0 位复位，DAC 通道数字接口仍处于使能状态。

25.2.4 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 VREF+ 之间的输出电压。DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DAC_{out} = \frac{V_{ref} * DOR}{4096}$$

25.2.5 DAC 触发选择

如果 TEN0 控制位置 1，可通过外部事件（定时计数器、外部中断线）或软件触发 DAC 转换。TSEL0[2:0]控制位将决定哪个事件将触发转换，如下表所示。

触发源	说明	TSEL0
TIM0_TRADC	片上定时器信号	000
TIM1_TRADC		001
TIM2_TRADC		010
TIM3_TRADC		011
TIM4_TRADC		100
TIM5_TRADC		101
SWTRIG	软件控制	110
EXTI	外部引脚	111

每当 DAC 检测到所选定时器 TRGO 输出，或所选外部触发端口上的上升沿时，DAC_DHR0 寄存器中存储的数据即会转移到 DAC_DOR0 寄存器中。经过三个 PCLK 周期后，DAC_DOR0 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DAC_DHR0 寄存器中的内容加载到 DAC_DOR0 寄存器中后，SWTRIG 即由硬件自动复位。数据从 DAC_DHR0 寄存器转移到 DAC_DOR0 寄存器只需一个 PCLK 时钟周期。

注意：

- **EN0 位置 1，即 DAC 使能时，无法更改 TSEL0[2:0]位的内容。**

25.2.6 功能说明

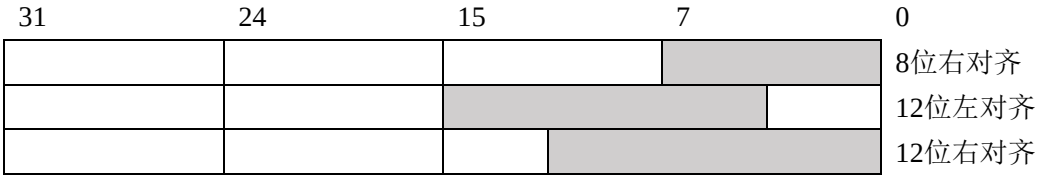
25.2.6.1 DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

存在三种可能：

- 8 位右对齐：软件必须将数据加载到 DAC_DHR8R0[7:0]位（实际是存储到 DHR0[11:4]位）
- 12 位左对齐：软件必须将数据加载到 DAC_DHR12L0[15:4]位（实际是存储到 DHR0[11:0]位）
- 12 位右对齐：软件必须将数据加载到 DAC_DHR12R0[11:0]位（实际是存储到 DHR0[11:0]位）

根据加载的 DAC_DHR0 寄存器，用户写入的数据将被相应的移位并存储到 DHR0（内部的数据保持寄存器，非存储器映射寄存器）中。之后，DHR0 寄存器将被自动加载，或者通过软件触发或外部事件触发加载到 DOR0 寄存器。



25.2.6.2 DAC 转换

DAC_DOR0 寄存器无法直接写入，任何数据都必须通过加载到 DAC_DHR0 寄存器中（写入 DAC_DHR8R0、DAC_DHR12L0、DAC_DHR12R0）才能传输到 DAC 通道。如果未选择触发（DAC_CR 寄存器中的 TEN0 位复位），那么经过一个 PCLK 时钟周期后，DAC_DHR0 寄存器中存储的数据将自动转移到 DAC_DOR0 寄存器。但是，如果选择硬件触发（置位 DAC_CR 寄存器中的 TEN0 位），数据将在触发条件到来的三个 PCLK 时钟周期后完成转移。

当 DAC_DOR0 加载了 DAC_DHR0 内容时，模拟输出电压将在一段时间 $t_{SETTLING}$ 后可用，具体时间长短取决于电源电压和模拟输出负载。

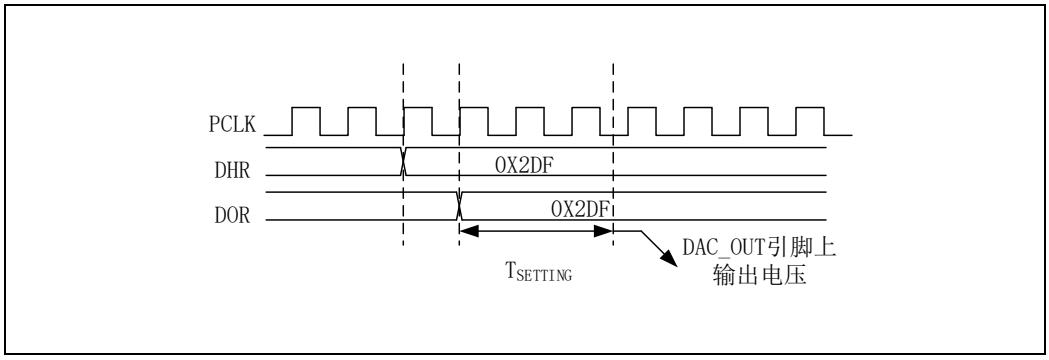


图 25-1 触发不使能时 DAC 转换时序图

生成单个 LFSR（线性反馈移位寄存器）的独立触发

要将 DAC 配置为此转换模式（请参见第 29.2.7 节：生成噪声），需要遵循以下顺序：

1. 将 DAC 通道触发使能位 TEN0 置 1。
2. 通过设置 TSEL0[2:0]位配置触发源。
3. 将 WAVE0[1:0]位配置为“01”，并在 MAMP0[3:0]位中配置相应的 LFSR 掩码值。
4. 将 DAC 通道转换数据加载到所需 DAC_DHR0 寄存器（DHR12R0、DHR12L0 或 DHR8R0）。

DAC 通道触发信号到达时，LFSR0 计数器内容根据掩码屏蔽后与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 PCLK 时钟周期之后），LFSR0 计数器随即更新。

生成单个三角波的独立触发

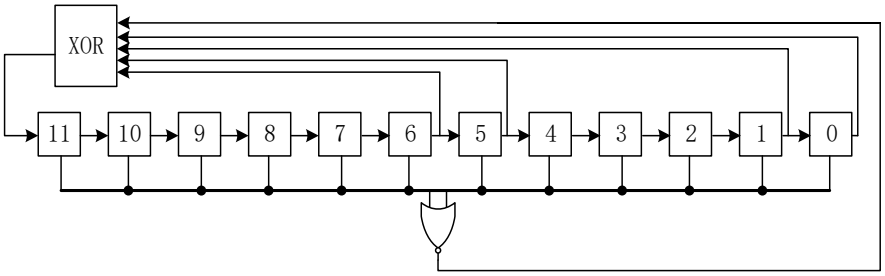
要将 DAC 配置为此转换模式（请参见第 29.2.8 节：生成三角波），需要遵循以下顺序：

1. 将 DAC 通道触发使能位 TEN0 置 1。
2. 通过设置 TSEL0[2:0]位配置触发源。
3. 将 WAVE0[1:0]位配置为“1x”，并在 MAMP0[3:0]位中配置相应的最大振幅值。
4. 将 DAC 通道转换数据加载到所需 DAC_DHR0 寄存器（DHR12R0、DHR12L0 或 DHR8R0）。

DAC 通道触发信号到达时，三角波计数器内容按照所设的振幅与 DHR0 寄存器内容相加，所得总和转移到 DAC_DOR0 中（三个 PCLK 时钟周期之后），三角波计数器随即更新。

25.2.7 噪声生成

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 WAVE0[1:0]置为“01”即可选择生成噪声。LFSR 中的预加载值为 0xAAA。在每次发生触发事件后，经过三个 PCLK 时钟周期，该寄存器会依照特定的计算算法完成更新。



LFSR 值可以通过 DAC_CR 寄存器中的 MAMP0[3:0]位来部分或完全屏蔽，所得到的值将与 DAC_DHR0 的内容相加，然后存储到 DAC_DOR0 寄存器中。如果发生溢出，溢出位将被舍弃。

如果 LFSR 为 0x000，将向其注入“1”（防锁定机制）。

可以通过复位 WAVE0[1:0]位来将 LFSR 波形产生功能关闭。

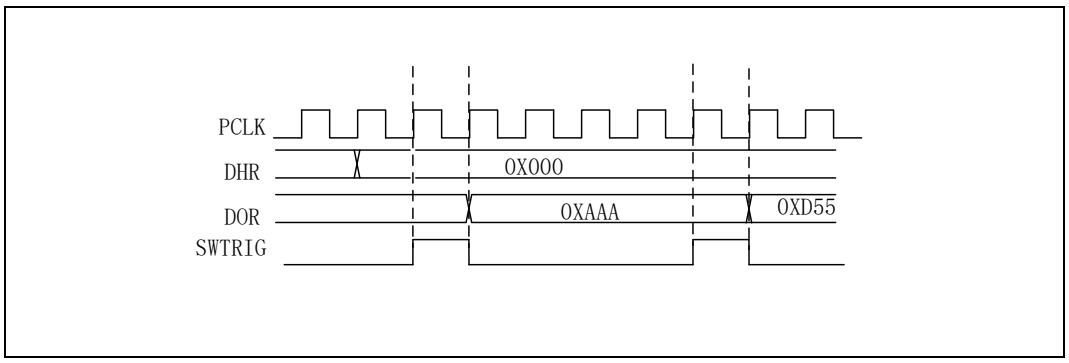


图 25-2 LFSR 产生波形的 DAC 转换（使用软件触发）

25.2.8 三角波生成

可以在直流或慢变信号上叠加一个小幅三角波。将 WAVE0[1:0]置为“10”即可选择 DAC 生成三角波。振幅通过 DAC_CR 寄存器中的 MAMP0[3:0]位进行配置。每次发生触发事件后，经过三个 PCLK 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DAC_DHR0 寄存器内容相加，所得总和将存储到 DAC_DOR0 寄存器中。只要小于 MAMP0[3:0]位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVE0[1:0]位来将三角波产生功能关闭。

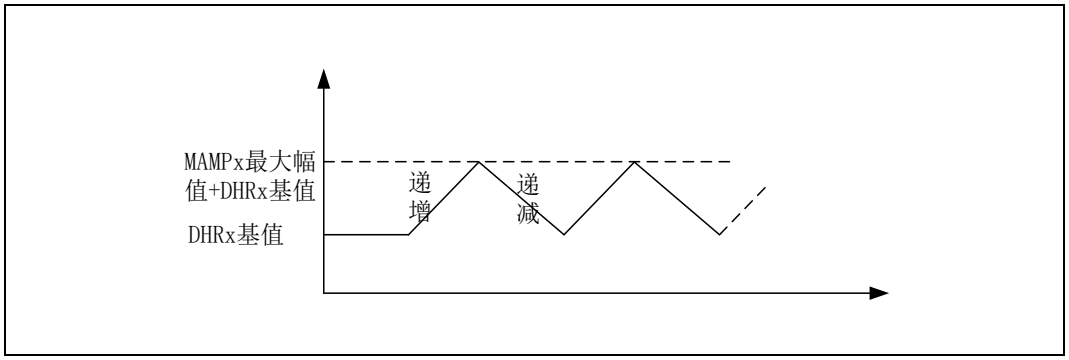


图 25-3 生成 DAC 三角波

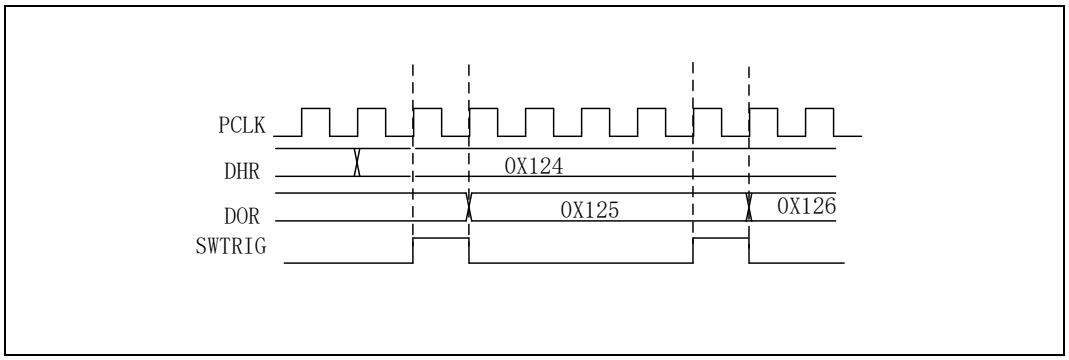


图 25-4 生成三角波波形的 DAC 转换（使能软件触发）

25.2.9 DMA 请求

DAC 通道具有 DMA 功能。DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAEN0 位置 1 时，如果发生触发，则将产生 DAC DMA 请求。DAC_DHR0 寄存器的值随后转移到 DAC_DOR0 寄存器。

DMA 下溢

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的应答，将不会发出新的请求，并且 DAC_SR 寄存器中的 DMA 通道下溢标志 DMAUDR0 将置 1，以报告这一错误状况。DMA 数据传输随即禁止，并且不再处理其它 DMA 请求。DAC 通道仍将继续转换旧有数据。

软件应通过写入“0”来将 DMAUDR0 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于 DAC 通道，如果使能 DAC_CR 寄存器中的 DMAUDRIE0 位，还将产生中断。

25.3 DAC 寄存器

基地址 0x40002500

PERI_CLKEN1.DAC 与 PERI_RESET1.DAC 控制为不可读，读出的值恒为 0，使用读修改写更改这两个寄存器的其他控制位时会修改掉 DAC 控制位的值成 0，需注意！

寄存器	偏移地址	描述
DAC_CR	0x000	控制寄存器
DAC_SWTR	0x004	软件触发寄存器
DAC_DHR12R0	0x008	DAC 12位右对齐数据寄存器
DAC_DHR12L0	0x00C	DAC 12位左对齐数据寄存器
DAC_DHR8R0	0X010	DAC 8位右对齐数据寄存器
DAC_DOR0	0x02C	DAC数据输出寄存器
DAC_SR	0X034	状态寄存器
DAC_ETRS	0X038	DAC测试寄存器/外部触发选择寄存器

表 25-1 DAC 寄存器

25.3.1 DAC 控制寄存器（DAC_CR）

偏移地址 0x00

复位值 0x0000 C000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SREF0		DMAUDRIE0	DMAEN0	MAMP0				WAVE0		TSEL0			TEN0	BOFF0	EN0
RW		RW	RW	RW				RW		RW			RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15:14	SREF0	DAC参考电压选择 00：内部1.5V 01：内部2.5V 10：外部参考电压ExRef（PB01） 11：AVCC电压
13	DMAUDRIE0	DAC DMA下溢中断使能 0:禁止DAC通道DMA下溢中断 1:使能DAC通道DMA下溢中断
12	DMAEN0	DAC通道DMA使能 0:禁止DAC通道DMA模式 1:使能DAC通道DMA模式
11:8	MAMP0	DAC通道掩码/振幅选择器 0000：不屏蔽LFSR的位0/三角波振幅等于1 0001：不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010：不屏蔽LFSR的位[2:0]/三角波振幅等于7 0011：不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100：不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101：不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110：不屏蔽LFSR的位[6:0]/三角波振幅等于127 0111：不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000：不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001：不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010：不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011：不屏蔽LFSR的位[11:0]/三角波振幅等于4095

7:6	WAVE0	DAC通道噪声/三角波生成使能 00:禁止生成波 01:使能生成噪声波 1x 使能生成三角波
5:3	TSEL0	DAC通道触发器选择 000 TIM0 TRADC触发 001 TIM1 TRADC触发 010 TIM2 TRADC触发 011 TIM3 TRADC触发 100 TIM4 TRADC触发 101 TIM5 TRADC触发 110 软件触发 111 外部端口触发
2	TEN0	DAC通道触发使能 0：禁止DAC通道触发，写入DAC_DHR0寄存器的数据在一个PCLK时钟周期之后转移到DAC_DOR0寄存器 1：使能DAC通道触发，DAC_DHR0寄存器的数据在三个PCLK时钟周期之后转移到DAC_DOR0寄存器
1	BOFF0	DAC通道输出缓冲器禁止 0：使能DAC通道输出缓冲器 1：禁止DAC通道输出缓冲器
0	EN0	DAC通道使能 0：禁止DAC通道 1：使能DAC通道

25.3.2 DAC 软件触发寄存器（DAC_SWTR）

偏移地址 0x04

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															SWTRIG0
															W1

位	标记	功能描述
31:1	Reserved	保留
0	SWTRIG0	DAC软件触发 写1软件触发，自动清零，写0无动作

25.3.3 DAC 12 位右对齐数据保持寄存器（DAC_DHR12R0）

偏移地址 0x08

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR0											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	DHR0	DAC通道 12位右对齐数据 这些位由软件写入，用于为DAC通道指定12位数据

25.3.4 DAC 12 位左对齐数据保持寄存器（DAC_DHR12L0）

偏移地址 0x0C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR0												Reserved			
RW															

位	标记	功能描述
31:16	Reserved	保留
15:4	DHR0	DAC通道 12 位左对齐数据 这些位由软件写入，用于为DAC通道指定12位数据
3:0	Reserved	保留

25.3.5 DAC 8 位右对齐数据保持寄存器（DAC_DHR8R0）

偏移地址 0x10

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DHR0							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	DHR0	DAC通道 8 位右对齐数据 这些位由软件写入，用于为DAC通道指定8位数据

25.3.6 DAC 数据输出寄存器（DAC_DOR0）

偏移地址 0x2C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DOR0											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	DOR0	DAC通道的数据输出

25.3.7 DAC 状态寄存器（DAC_SR）

偏移地址 0x34

复位值 0x00004000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMAUDR0	Reserved												
		RW													

位	标记	功能描述
31:14	Reserved	保留
13	DMAUDR0	DAC通道DMA下溢标志 0：DAC通道未发生DMA下溢错误状况 1：DAC通道发生DMA下溢错误状况（当前所选触发源以高于DMA服务能力的频率驱动DAC通道转换）
12:0	Reserved	保留

25.3.8 DAC 控制寄存器（DAC_ETRS）

偏移地址 0x038

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PTIRGSEL			RSV			
									RW			RW			

位	标记	功能描述
31:7	Reserved	保留
6:4	PTRIGSEL	端口触发选择 000 PA9 001 PB9 010 PC9 011 PD9 100 PE9 101 PF9 11X 保留
3:0	Reserved	保留

26 模拟比较器（VC）

26.1 模拟电压比较器 VC 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“＋”输入端电压高于“－”输入端电压时，电压比较器输出为高电平；当“＋”输入端电压低于“－”输入端电压时，电压比较器输出为低电平。本产品内部集成的模拟电压比较器 VC 具有以下特性：

- 支持电压比较功能；
- 支持内部 64 阶 VCC 分压（使用分压来源电压需要大于 1.8V）
- 支持 16 个外部输入端口和片内 BGR 输出的参考电压作为电压比较器的输入；
- 支持三种软件可配置的中断触发方式：高电平触发/上升沿触发/下降沿触发；
- 电压比较器的输出可以作为 通用定时器 和 低功耗定时器控制输入；
- 电压比较器的输出可以作为 高级定时器和通用定时器的刹车输入或者捕获输入；
- 支持在深度休眠模式下工作，电压比较器的中断输出可以将芯片从深度休眠模式下唤醒；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。

  注意：使用 VC 需要使能 BGR，参考 BGR 寄存器描述。

26.2 电压比较器框架图

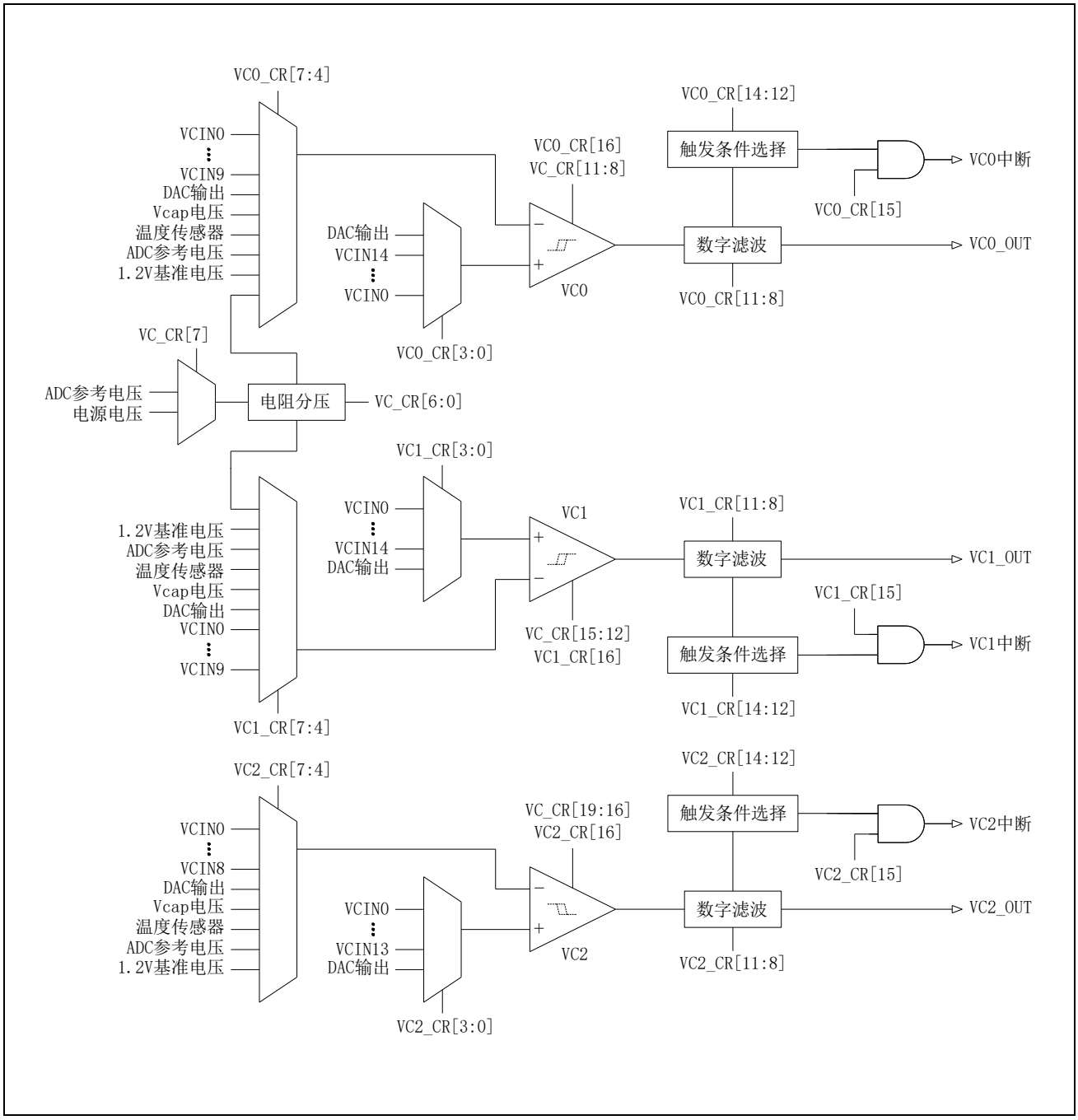


图 26-1 VC 框架图

26.3 建立/响应时间

当使用电压比较器时，从 VC 使能或者 VC 的两端输入电压变化到输出正确的结果的时间由 VC 控制寄存器（VC_CR）中的 BIAS_SEL 控制位决定，电流越大 VC 响应越快，典型值从 200nS 到 20uS 四档可调。

如果需要内部 BGR 提供的参考电压（1.5V 或 2.5V）作为比较基准电压。内部 BGR 的启动时间大约为 20us，电压比较器需要等待内部 BGR 稳定后才能正常输出。

26.4 滤波时间

在电压比较器固有的建立/响应时间之外，用户可以设置更长的滤波时间来过滤掉系统噪声，比如马达停止时的大电流噪声。

从寄存器 VC_IFR.VCx_Filter 可以读出数字滤波后的信号电平；当 GPIO 功能配置为 VCx_OUT 时，数字滤波后的信号可以从 GPIO 输出以方便测量。

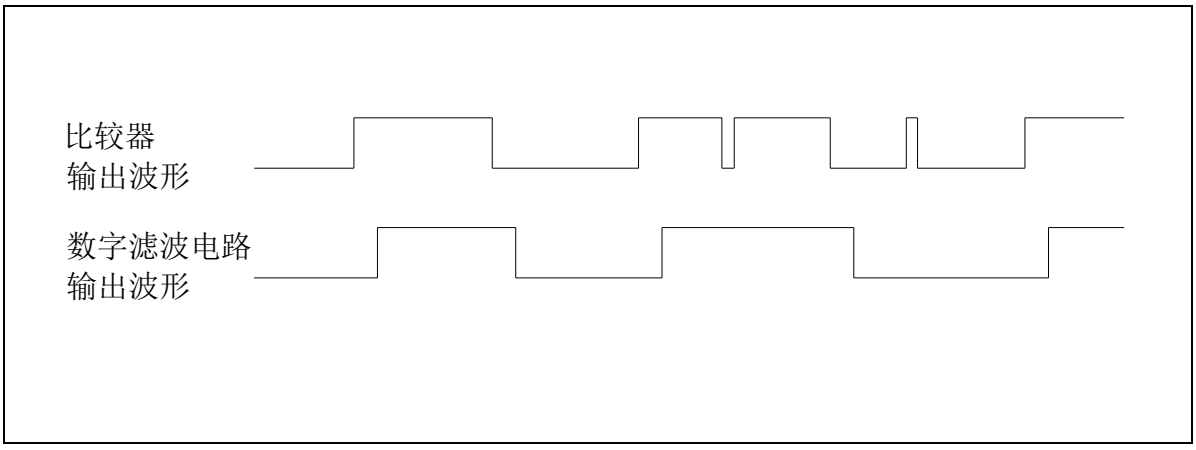


图 26-2 VC 滤波响应时间

26.5 迟滞功能

电压比较器可以选择迟滞功能，迟滞功能使能后的图示如下：

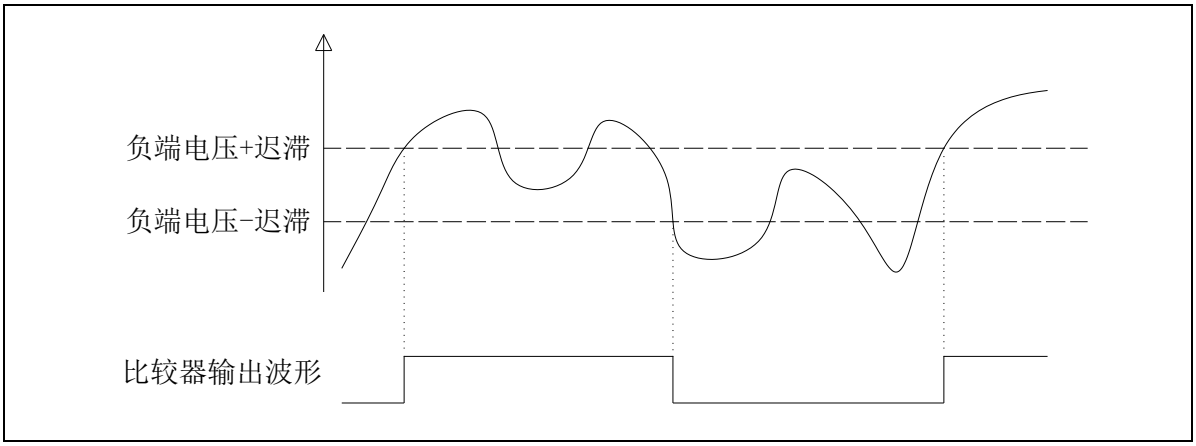


图 26-3 VC 迟滞功能

26.6 VC 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
VC_CR	0x010	VC0/1 配置寄存器0
VC0_CR	0x014	VC0 配置寄存器
VC1_CR	0x018	VC1 配置寄存器
VC0_ OUT_CFG	0x01C	VC0 输出配置寄存器
VC1_ OUT_CFG	0x020	VC1 输出配置寄存器
VC_INT	0x024	VC 中断寄存器
VC2_CR	0x150	VC2 配置寄存器
VC2_ OUT_CFG	0x154	VC2输出配置寄存器

表 26-1 VC 寄存器

26.6.1 VC 配置寄存器（VC_CR）

偏移地址 0x010

复位值 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												VC2_HYS_SEL		VC2_BIAS_SEL	
												R/W		R/W	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1_HYS_SEL		VC1_BIAS_SEL		VC0_HYS_SEL		VC0_BIAS_SEL		VC_REF2P5_SEL	VC_DIV_EN	VC_DIV					
R/W		R/W		R/W		R/W		R/W	R/W	R/W					

位	标记	功能描述
31:16	Reserved	保留
19:18	VC2_HYS_SEL	VC2 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
17:16	VC2_BIAS_SEL	VC2 功耗选择（功耗越大，响应速度越快） 00:300nA 01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us)

		11:20uA(需要开启BGR，BGR启动时间大约20us)
15:14	VC1_HYS_SEL	VC1 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
13:12	VC1_BIAS_SEL	VC1 功耗选择（功耗越大，响应速度越快） 00:300nA 01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us) 11:20uA(需要开启BGR，BGR启动时间大约20us)
11:10	VC0_HYS_SEL	VC0 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
9:8	VC0_BIAS_SEL	VC0 功耗选择（功耗越大，响应速度越快） 00:300nA 01:1.2uA 10:10uA(需要开启BGR，BGR启动时间大约20us) 11:20uA(需要开启BGR，BGR启动时间大约20us)
7	VC_REF2P5_SEL	VC_DIV 参考电压Vref选择 0:VCC 1:ADC_CR0.SREF所选择的参考电压
6	VC_DIV_EN	6位 DAC 使能 1：使能
5:0	VC_DIV	6位 DAC 配置 000000: 1/64 Vref 000001:2/64 Vref 000010:3/64 Vref 000011:4/64 Vref … 111110:63/64 Vref 111111: Vref

26.6.2 VC0 配置寄存器（VC0_CR）

偏移地址 0x014

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time			FLTEN	n_sel				p_sel			
RW	RW	RW	RW	RW			RW	RW				RW			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波 0：VC 无滤波

7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PA0 0001: select channel 1 input PA1 0010: select channel 2 input PA2 0011: select channel 3 input PA3 0100: select channel 4 input PA4 0101: select channel 5 input PA5 0110: select channel 6 input PA6 0111: select channel 7 input PA7 1000: select channel 8 input PC4 1001: select channel 9 input PC5 1010:DAC 1011:电阻分压输出电压 1100:内置温度传感器输出电压 1101:内部基准1.2V输出电压 1110:ADC的REF (使用需要使能ADC) 1111:VCAP引脚的电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PC0 0001: select channel 1 input PC1 0010: select channel 2 input PC2 0011: select channel 3 input PC3 0100: select channel 4 input PA0 0101: select channel 5 input PA1 0110: select channel 6 input PA2 0111: select channel 7 input PA3 1000: select channel 8 input PA4 1001: select channel 9 input PA5 1010: select channel 10 input PA6 1011: select channel 11 input PA7 1100: select channel 12 input PB4 1101: select channel 13 input PB5 1110: select channel 14 input PB6 1111: DAC

26.6.3 VC1 配置寄存器（VC1_CR）

偏移地址 0x018

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time			FLTEN	n_sel				p_sel			
RW	RW	RW	RW	RW			RW	RW				RW			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波 0：VC 无滤波

7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PC0 0001: select channel 1 input PC1 0010: select channel 2 input PC2 0011: select channel 3 input PC3 0100: select channel 4 input PA0 0101: select channel 5 input PA1 0110: select channel 6 input PB0 0111: select channel 7 input PB1 1000: select channel 8 input PB2 1001: select channel 9 input PB3 1010: DAC1 1011:电阻分压输出电压 1100:内置温度传感器输出电压 1101:内部基准1.2V输出电压 1110:ADC的REF (使用需要使能ADC) 1111:VCAP引脚的电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PA0 0001: select channel 1 input PA1 0010: select channel 2 input PA2 0011: select channel 3 input PA3 0100: select channel 4 input PA4 0101: select channel 5 input PA5 0110: select channel 6 input PB1 0111: select channel 7 input PB2 1000: select channel 8 input PB10 1001: select channel 9 input PB12 1010: select channel 10 input PB13 1011: select channel 11 input PB14 1100: select channel 12 input PB4 1101: DAC 1110: select channel 14 input PB6 1111: select channel 15 input PB7

26.6.4 VC0 输出配置寄存器（VC0_OUT_CFG）

偏移地址 0x01C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_TIM6	TIM5	INV_TIM5	TIM4	INV_TIM4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC0 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC0 filter 结果输出到 TIM6 捕获输入CHA使能 1：使能； 0：禁止。
13	INV_TIM6	VC0 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC0 filter 结果输出到 TIM5 捕获输入CHA使能 1：使能； 0：禁止。
11	INV_TIM5	VC0 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC0 filter 结果输出到 TIM4 捕获输入CHA使能 1：使能； 0：禁止。
9	INV_TIM4	VC0 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC0 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC0 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC0 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC0 filter 结果输出到 TIM1 REFCLR 使能控制 1：使能； 0：禁止。
1	TIM0RCLR	VC0 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。

0	INV_Timer	VC0 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

26.6.5 VC1 输出配置寄存器（VC1_ OUT_CFG）

偏移地址 0x020

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_TIM6	TIM5	INV_TIM5	TIM4	INV_TIM4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC1 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC1 filter 结果输出到 TIM6 捕获输入CHB使能 1：使能； 0：禁止。
13	INV_TIM6	VC1 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC1 filter 结果输出到 TIM5 捕获输入CHB使能 1：使能； 0：禁止。
11	INV_TIM5	VC1 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC1 filter 结果输出到 TIM4 捕获输入CHB使能 1：使能； 0：禁止。
9	INV_TIM4	VC1 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC1 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC1 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC1 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC1 filter 结果输出到 TIM1 REFCLR 使能控制 1：使能； 0：禁止。
1	TIM0RCLR	VC1 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。

0	INV_Timer	VC1 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

26.6.6 VC 中断寄存器（VC_IFR）

偏移地址 0x024

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										VC2_Filter	VC2_INTF	VC1_Filter	VC0_Filter	VC1_INTF	VC0_INTF
										RO	RW0	RO	RO	RW0	RW0

位	标记	功能描述
31:4	Reserved	保留
5	VC2_Filter	VC2 Filter 后的状态
4	VC2_INTF	VC2 中断标志，1发生 VC2 中断；0未发生中断；写0清除中断标志，写1无效
3	VC1_Filter	VC1 Filter 后的状态
2	VC0_Filter	VC0 Filter 后的状态
1	VC1_INTF	VC1 中断标志，1发生 VC1 中断；0未发生中断；写0清除中断标志，写1无效
0	VC0_INTF	VC0 中断标志，1发生 VC0 中断；0未发生中断；写0清除中断标志，写1无效

26.6.7 VC2 配置寄存器（VC2_CR）

偏移地址 0x150

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															EN
															R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	fallin	debounce_time			FLTEN	n_sel				p_sel			
R/W	R/W	R/W	R/W	R/W			R/W	R/W				R/W			

位	标记	功能描述
31:17	Reserved	保留
16	EN	电压比较器使能 1：使能电压比较器 0：关闭电压比较器
15	IE	VC 中断使能 1：使能；0：禁止
14	level	VC 输出信号触发中断选择 1：使能高电平触发 INT flag 0：禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1：使能上升沿触发 INT flag 0：禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1：使能下降沿触发 INT flag 0：禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us 011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注意：滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1：启动 VC 滤波 0：VC 无滤波

7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input PA5 0001: select channel 1 input PB1 0010: select channel 2 input PE11 0011: select channel 3 input PE15 0100: select channel 4 input PB11 0101: select channel 5 input PB14 0110: select channel 6 input PD10 0111: select channel 7 input PD11 1000: select channel 8 input PC7 1001: DAC 1010: DAC 1011:NC 1100:内置温度传感器输出电压 1101:内部基准1.2V输出电压 1110:ADC的REF (使用需要使能ADC) 1111:VCAP引脚的电压
3:0	P_SEL	电压比较器“+”端输入选择 0000: select channel 0 input PA5 0001: select channel 1 input PB1 0010: select channel 2 input PE9 0011: select channel 3 input PE10 0100: select channel 4 input PE11 0101: select channel 5 input PE13 0110: select channel 6 input PE14 0111: select channel 7 input PE15 1000: select channel 8 input PB11 1001: select channel 9 input PB14 1010: select channel 10 input PD9 1011: select channel 11 input PD10 1100: select channel 12 input PD11 1101: select channel 13 input PC7 1110: DAC 1111: DAC

26.6.8 VC2 输出配置寄存器（VC2_ OUT_CFG）

偏移地址 0x154

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_Timer6	TIM5	INV_Timer5	TIM4	INV_Timer4	Reserved			TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	INV_Timer
R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC2 作为 Advanced Timer 刹车控制 1：使能； 0：禁止。
14	TIM6	VC2 filter 结果输出到 TIM6 捕获输入CHB使能 1：使能； 0：禁止。
13	INV_TIM6	VC2 filter 结果输出到 TIM6 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
12	TIM5	VC2 filter 结果输出到 TIM5 捕获输入CHB使能 1：使能； 0：禁止。
11	INV_TIM5	VC2 filter 结果输出到 TIM5 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
10	TIM4	VC2 filter 结果输出到 TIM4 捕获输入CHB使能 1：使能； 0：禁止。
9	INV_TIM4	VC2 filter 结果输出到 TIM4 反向使能 1：使能反向； 0：禁止反向，输入与 VC 输出同向。
8:6	Res.	保留
5	TIMBK	VC2 filter 结果输出到Timer0/1/2/3 刹车控制 1：使能； 0：禁止。
4	TIM3RCLR	VC2 filter 结果输出到 TIM3 REFCLR 使能控制 1：使能； 0：禁止。
3	TIM2RCLR	VC2 filter 结果输出到 TIM2 REFCLR 使能控制 1：使能； 0：禁止。
2	TIM1RCLR	VC2 filter 结果输出到 TIM1 REFCLR 使能控制 1：使能； 0：禁止。
1	TIM0RCLR	VC2 filter 结果输出到 TIM0 REFCLR 使能控制 1：使能； 0：禁止。

0	INV_Timer	VC2 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1：使能反向； 0：禁止反向，输入与 VC 输出同向。

27 低电压检测器（LVD）

27.1 LVD 简介

LVD 可用于监测 VCC 及芯片管脚的电压。当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号，用户可根据该信号执行一些紧急任务。

LVD 具有以下特性：

- 4 路监测源，AVCC、PC13、PB08、PB07；
- 16 阶阈值电压，1.8~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

27.2 LVD 框图

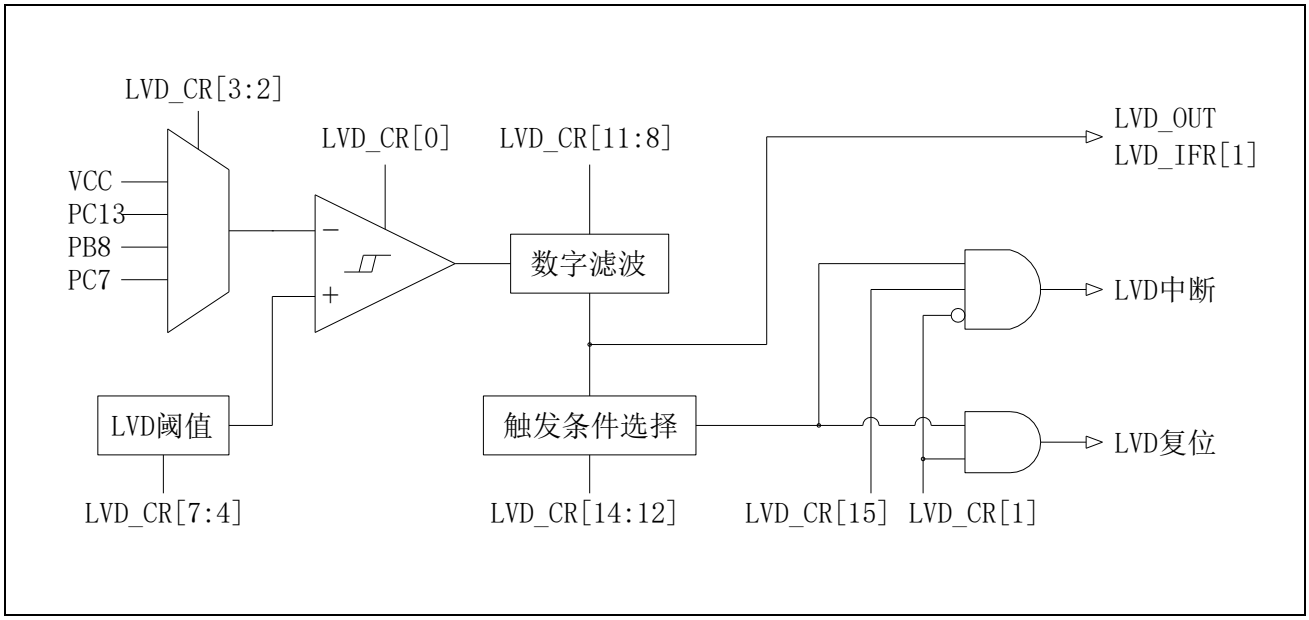


图 27-1 LVD 框图

27.3 迟滞功能

LVD 内置的电压比较器具有迟滞功能，可以增强芯片的抗干扰能力。LVD 的输出信号会等到输入信号高于或低于阈值电压 20mV 后才发生翻转，其输入输出信号示意如下：

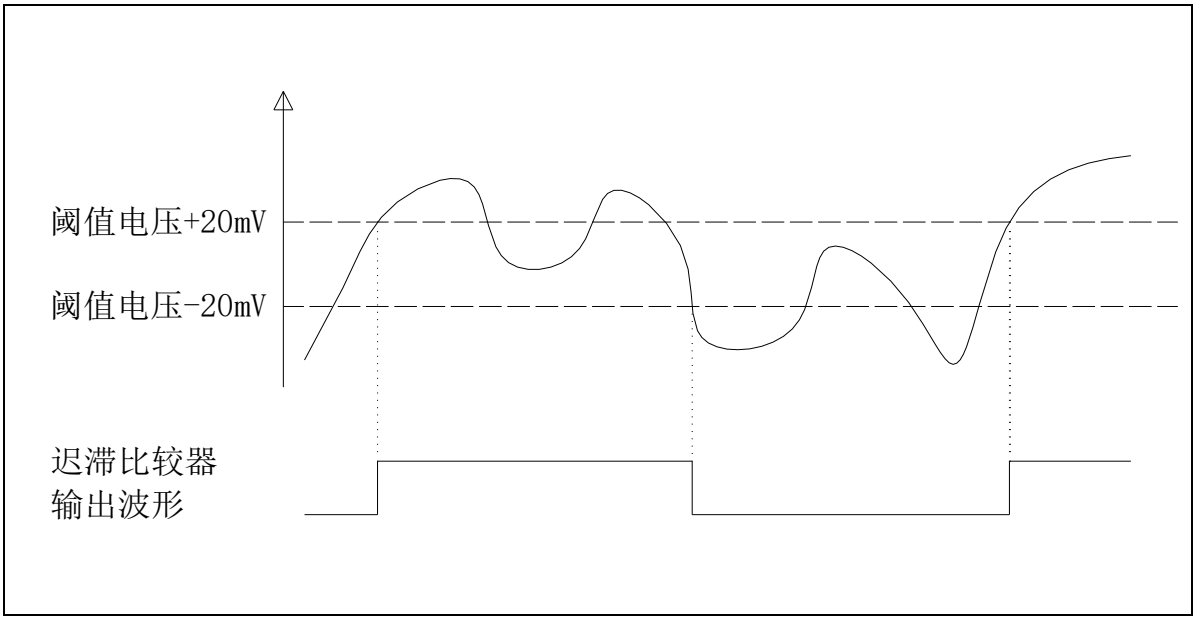


图 27-2 LVD 迟滞响应

27.4 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。

从寄存器LVD_IFR[1]可以读出数字滤波后的信号电平；当GPIO功能配置为LVD_OUT时，数字滤波后的信号可以从 GPIO 输出以方便测量。

使能数字滤波模块，滤波示意如下所示：

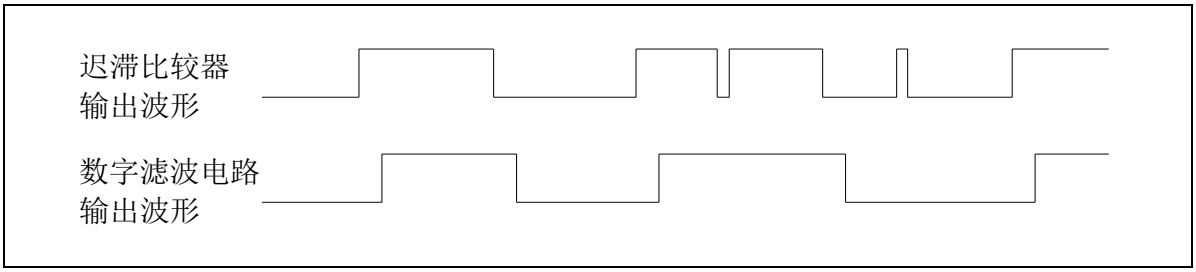


图 27-3 LVD 滤波输出

27.5 配置示例

27.5.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。

配置方法如下所示：

Step1：配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2：配置 LVD_CR. VTDS，选择 LVD 的阈值电压。

Step3：配置 LVD_CR. Debounce_time，选择 LVD 滤波时间。

Step4：配置 LVD_CR. FLTEN，使能 LVD 滤波。

Step5：设置 LVD_CR. HTEN 为 1，选择高电平触发 LVD 动作。

Step6：设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。

Step7：设置 LVD_CR.LVDEN 为 1，使能 LVD。

27.5.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。

配置方法如下所示：

Step1：配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2：配置 LVD_CR. VTDS，选择 LVD 的阈值电压。

Step3：配置 LVD_CR. Debounce_time，选择 LVD 滤波时间。

Step4：配置 LVD_CR. FLTEN，使能 LVD 滤波。

Step5：设置 LVD_CR. RTEN 和 LVD_CR. FTEN 为 1，选择电平变化触发 LVD 动作。

Step6：设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。

Step7：设置 LVD_CR.IE 为 1，使能 LVD 中断。

Step8：使能 NVIC 中断向量表中的 LVD 中断。

Step9：设置 LVD_CR.LVDEN 为 1，使能 LVD。

Step10：在 LVD 的中断服务程序中执行用户需要进行的操作；退出中断服务程序前向 LVD_IFR 写入 0x00 以清除中断标志。

27.6 LVD 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
LVD_CR	0x028	LVD 配置寄存器
LVD_IFR	0x02C	LVD 中断标志寄存器

表 27-1 LVD 寄存器

27.6.1 LVD 配置寄存器（LVD_CR）

偏移地址 0x028

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	HTEN	RTEN	FTEN	Debounce_time			FLTEN	VTDS				Source_sel		ACT	LVDEN
RW	RW	RW	RW	RW			RW	RW				RW		RW	RW

位	标记	功能描述
31:16	Reserved	保留
15	IE	LVD中断使能 1：使能； 0：禁止。
14	HTEN	高电平触发使能 (被监测电压低于阈值电压) 1：使能； 0：禁止。
13	RTEN	上升沿触发使能 (被监测电压从高于阈值电压变为低于阈值电压) 1：使能； 0：禁止。
12	FTEN	下降沿触发使能 (被监测电压从低于阈值电压变为高于阈值电压) 1：使能； 0：禁止。
11:9	Debounce_time	数字滤波时间配置 111：滤波时间大约为28.8ms 110：滤波时间大约为7.2ms 101：滤波时间大约为1.8ms 100：滤波时间大约为450us

		011：滤波时间大约为112us 010：滤波时间大约为28us 001：滤波时间大约为14us 000：滤波时间大约为7us 注：滤波时间仅在FLTEN为1时有效。
8	FLTEN	数字滤波使能配置 1: 使能数字滤波 0：禁止数字滤波
7:4	VTDS	LVD 阈值电压选择 1111：3.3v 1110：3.2v 1101：3.1v 1100：3.0v 1011：2.9v 1010：2.8v 1001：2.7v 1000：2.6v 0111：2.5v 0110：2.4v 0101：2.3v 0100：2.2v 0011：2.1v 0010：2.0v 0001：1.9v 0000：1.8v
3:2	Source_sel	LVD 监测来源选择 11：PB07端口输入电压 10：PB08端口输入电压 01：PC13端口输入电压 00：AVCC 电源电压
1	ACT	LVD触发动作选择 1：系统复位 0：NVIC中断
0	LVDEN	LVD使能控制 1：使能LVD 0：禁止LVD

27.6.2 LVD 中断寄存器（LVD_IFR）

偏移地址 0x02C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														LVD_Filter	INTF
														RO	RW0

位	标记	功能描述
31:2	Reserved	保留
1	LVD_Filter	LVD Filter 后的状态
0	INTF	LVD中断标志： 1：发生LVD中断； 0：未发生中断； 写0清除中断标志，写1无效。

28 运算放大器（OPA）

OPA 模块是一个通用运算放大器，适合于简易放大器和电压跟随器应用，并可配置为内部 DAC 的跟随输出。OPA 的输出可以配置到不同的管脚，方便使用 ADC 的相应通道进行采样。OPA 的输入范围是 0V 到 AVCC，输出范围是 0.1V 到 AVCC-0.2V。

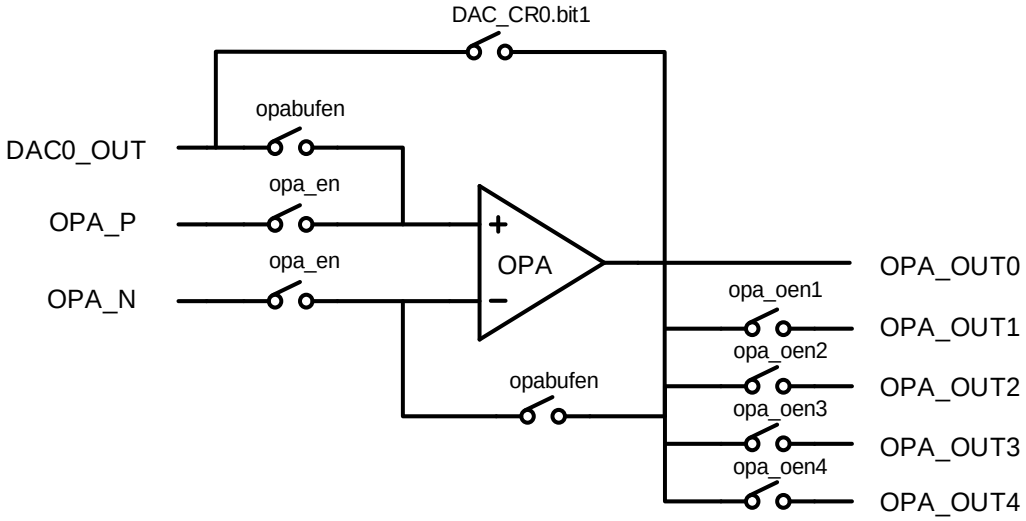
28.1 OPA 特性

- 1 个独立配置运放
- 输入范围 0～AVCC，输出范围 0.1V～AVCC-0.2V
- 四个可选输出管脚 OPA_OUT1～OPA_OUT4
- 可配置为以下模式
  - 通用运放模式（general purpose OPA）
  - DAC 电压跟随器

28.2 OPA 功能描述

OPA 可以作为通用运算放大器，使用外部元件组成相应的信号调理电路来实现所需的功能，也可以作为内部 DAC 的缓冲器输出以提高驱动能力。

OPA 的示意图如下：



当 OPA 作为内部 DAC 的缓冲器使用时，需要按照如下设置：

- 将寄存器 DAC_CR 的 BOFF0 位（bit1）置 0
- 将寄存器 OPA_CR0 的 EN 位（bit0）置 0
- 将寄存器 OPA_CR0 的 OPABUFEN 位（bit2）置 1

则 DAC 的输出信号经过 OPA 跟随后输出到管脚。

28.3 OPA 校零

OPA 支持不同模式的自动校零功能，可以软件自己控制，也可以硬件控制。高精度 OPA 应用，在每次使用 OPA 前，需要进行校零。校零建议时间 10us 以上，校零后请立即采样。

28.3.1 软件控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 设置 OPA_CR. CLK_SW_SET 为 1
- 软件写 OPA_CR. AZ_PULSE 为 1
- 设定校零时间，等待校零时间结束，清除 OPA_CR. AZ_PULSE 为 0

28.3.2 软件触发控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 选择需要校零的时间参数 OPA_CR.CLK_SEL
- 设置软件触发 OPA_CR. TRIGGER

28.3.3 ADC 触发控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 选择需要校零的时间参数 OPA_CR.CLK_SEL
- 使能 ADC 触发 ADCTR_EN

28.4 OPA 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
OPA_CR0	0x030	OPA 控制寄存器0
OPA_CR1	0x03C	OPA 控制寄存器1

表 28-1 OPA 寄存器

28.4.1 OPA 配置寄存器（OPA_CR0）

偏移地址 0x030

复位值 0x00000180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						BIASSEL			OPA OEN3	OPA OEN3	OPA OEN2	OPAO EN1	OPA BUF EN	AZEN	EN
						RW			RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:10	Reserved	保留
9:7	BIASSEL	OPA偏置电流选择 缺省值为011，一般不需要更改
6	OPA_OEN4	OPA输出4使能
5	OPA_OEN3	OPA输出3使能
4	OPA_OEN2	OPA输出2使能
3	OPA_OEN1	OPA输出1使能
2	OPABUFEN	DAC使用OPA单元作为缓冲输出，使能时，OPA使能需要关闭
1	AZEN	自动校零使能
0	OPA_EN	OPA使能

28.4.2 OPA 配置寄存器（OPA_CR1）

偏移地址 0x03C

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CLK_SEL				CLK_SW_SET	AZ_PULSE	TRIGGER	ADCTR_EN
								RW				RW	RW	W1	RW

位	标记	功能描述
31:8	Reserved	保留
7:4	CLK_SEL	自动校准脉冲宽度设置（建议自动校准时间为10us） 0000:1个PCLK周期 0001:2个PCLK周期 0010:4个PCLK周期 0011:8个PCLK周期 0100:16个PCLK周期 0101:32个PCLK周期 0110:64个PCLK周期 0111:128个PCLK周期 1000:256个PCLK周期 1001:512个PCLK周期 1010:1024个PCLK周期 1011:2048个PCLK周期 11xx:4096个PCLK周期
3	CLK_SW_SET	自动校零选择 1：软件校准使能 0：软件校准禁止，软件触发校准使能
2	AZ_PULSE	软件校准 1：校准 0：停止
1	TRIGGER	软件触发校准设置，自动清零 1：软件触发校准 0：无效
0	ADCTR_EN	ADC启动触发OPA自动校准使能 1：使能 0：禁止

29 模拟其它寄存器

基地址 0x40002400

寄存器	偏移地址	描述
BGR_CR	0x000	BGR 控制寄存器

29.1 BGR 配置寄存器（BGR_CR）

偏移地址 0x000

复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TS_EN	BGR_EN
														RW	RW

位	标记	功能描述
31:2	Reserved	保留
1	TS_EN	内置温度传感器使能控制 1：使能内部温度传感器，必须使能 BGR 0：禁止内部温度传感器 注意：温度传感器使能 20us 后，才能输出稳定信号。
0	BGR_EN	BGR 使能控制 1：使能 BGR 0：禁止 BGR 注意： 1) PERI0_CLKEN.ADC 为 1 时，才可以操作此寄存器。 2) BGR 使能 20us 后，才能输出稳定的高精度参考电压。BGR 稳定后才可以被其它模块使用，故用户操作中应加入等待 BGR 稳定的步骤。 3)当使用 ADC,VC,OPA 时，必须使能 BGR

30 SWD 调试接口

本系列使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当本系列微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

注意：

– SWD 能把 DeepSleep 唤醒，请注意不要以 SWD 接口断开作为 DeepSleep 的状态。

30.1 SWD 调试附加功能

本产品使用了 ARM Cortex-M0+ CPU，该内核包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 MCU 相连并进行调试时，将使用调试功能。

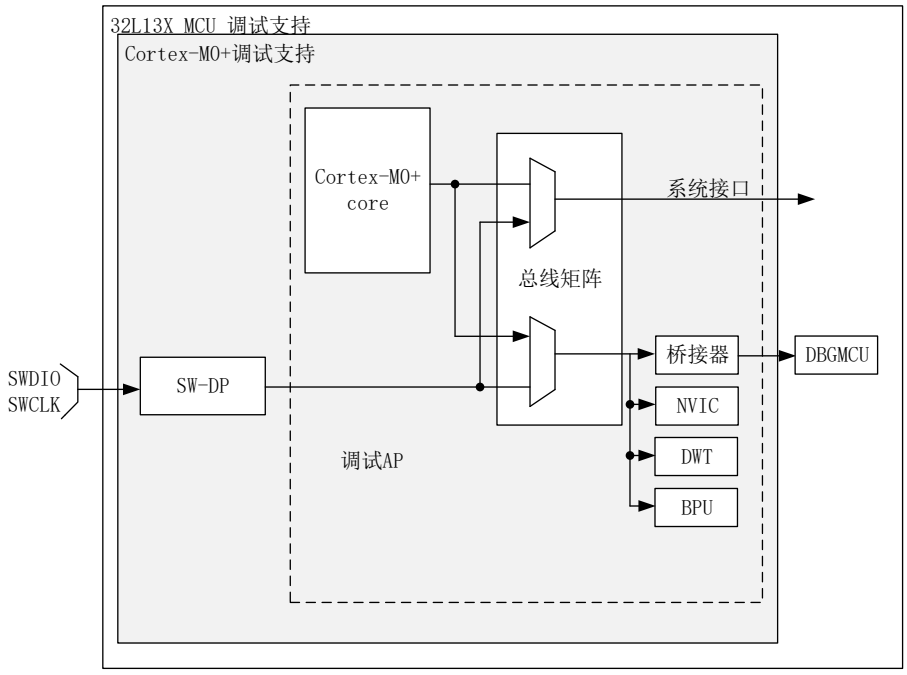


图 30-1 调试支持框图

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

注：

– 有关 ARM® Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册。

30.2 ARM® 参考文档

- Cortex®-M0+技术参考手册 (TRM)

  可从 www.infocenter.arm.com 获取。
- ARM® 调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

30.3 调试端口引脚

30.3.1 SWD 端口引脚

本系列的 SWD 接口，需要用到 2 个引脚，如下表所示。

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入/输出	PA13

30.3.2 SW-DP 引脚分配

如果烧录程序时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 PA13/PA14 引脚均被初始化为可被调试器使用的专用引脚。用户可设置 SYSCTRL1 .SWD_USE_IO 寄存器来禁止 SWD 引脚的调试功能，SWD 引脚将被释放以用作普通 GPIO。SWD 引脚的配置与功能小结如下表所示：

【加密芯片】选项	SWD_USE_IO 配置	PA13/PA14 功能
加密	0	NA
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

30.3.3 SWD 引脚上的内部上拉

用户软件释放 SWI/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

30.4 SWD 端口

30.4.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM® 建议采用 100 k）。这些上拉电阻可在内部配置。无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

30.4.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

位	名称	说明
0	启动	必须为1
1	APnDP	0：DP 访问；1：AP 访问
2	RnW	0：写请求；1：读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动。

位	名称	说明
0	ACK	001：FAULT 010：WAIT 100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

位	名称	说明
0:31	WDATA 或RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

30.4.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM®代码，设置为 **0x0BB11477**（相当于 Cortex®-M0+）。注意：

– 在目标读取此 ID 代码前，SW-DP 状态机是不工作的。

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 *Cortex®-M0+ TRM* 和 *CoreSight 设计套件 r1p0TRM*。

30.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的

READOK 标志，以便了解 AP 读访问是否成功。

- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

30.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	**RW**	**SELECT** 寄存器的 **CTRLSEL** 位	寄存器	注释
00	读取		IDCODE	制造商代码设置为 Cortex®-M0+ 的默认 ARM® 代码。 **0x0BB11477**（标识 SW-DP）
00	写		ABORT	
01	读 / 写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
01	读 / 写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的 持续时间）
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数 据，无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器 窗口
11	读 / 写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有 用（在执行下个 AP 事务时提供读取 AP 请求 的结果）。此读取缓冲区捕获 AP 中的数据，显示为

前一次读取的结果，无需启动新操作。

30.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	**A[3:2]** 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 – 位 31:24：APSEL：选择当前 AP (select the current AP) – 位 23:8：保留 – 位 7:4：APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果

30.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	*32* 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	*17* 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	*32* 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	*32* 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

30.6 BPU（断点单元）

Cortex®-M0+ BPU 实现提供四个断点寄存器。

30.6.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M ARM®和 ARM® CoreSight 组件技术参考手册。

30.7 DWT（数据观察点）

Cortex®-M0+ DWT 实现提供了两个观察点寄存器组。

30.7.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM® 中所述。

30.7.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

30.8 MCU 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

30.8.1 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

30.8.2 对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

30.9 调试模式模块工作状态控制（DEBUG_ACTIVE）

复位值 0x00000FFF(仅在 SWD 调试模式下，此寄存器设置才起作用)

地址：0x4000_2038

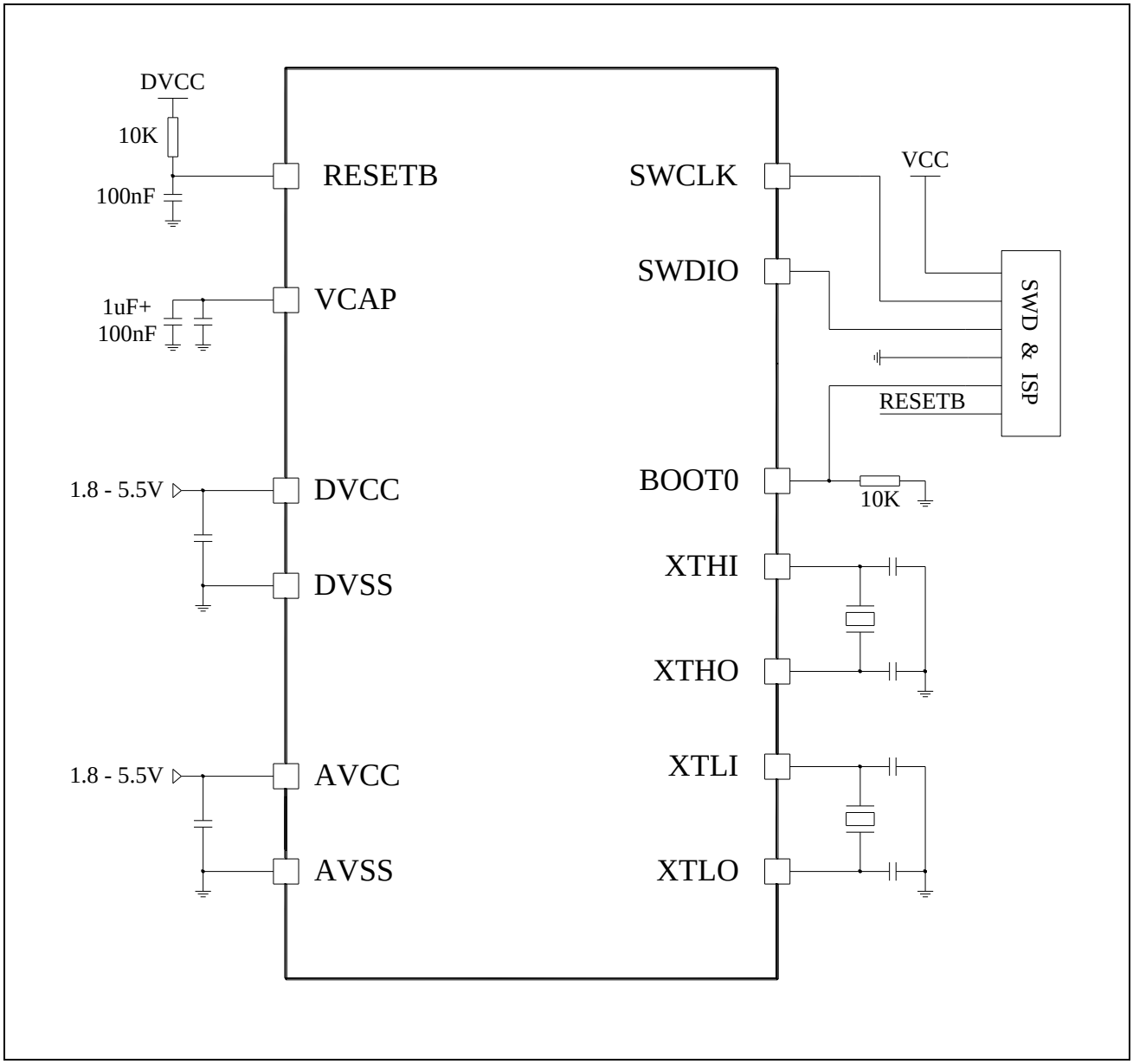
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TIM3	Reserved		WDT	PCA	TIM6	TIM5	TIM4	Res.	TIM2	TIM1	TIM0
				RW			RW	RW	RW	RW	RW		RW	RW	RW

位	标记	功能描述
31:12	Reserved	保留
11	TIM3	调试时， Timer3 计数功能配置 1：在 SWD 调试界面下，暂停 Timer3 计数功能 0：在 SWD 调试界面下，Timer3 正常计数功能
10:9	Reserved	保留
8	WDT	调试时， WDT 计数功能配置 1：在 SWD 调试界面下，暂停 WDT 计数功能 0：在 SWD 调试界面下，WDT 正常计数功能
7	PCA	调试时， PCA 计数功能配置 1：在 SWD 调试界面下，暂停 PCA 计数功能 0：在 SWD 调试界面下，PCA 正常计数功能
6	TIM6	调试时， Timer6 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能
5	TIM5	调试时， Timer5 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能
4	TIM4	调试时， Timer4 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Time r 正常计数功能
3	Reserved	保留
2	TIM2	调试时， Timer2 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能
1	TIM1	调试时， Timer1 计数功能配置 1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能
0	TIM0	调试时， Timer0 计数功能配置

		1：在 SWD 调试界面下，暂停 Timer 计数功能 0：在 SWD 调试界面下，Timer 正常计数功能

31 典型应用电路图



注意：

– AVCC 与 DVCC 电压必须相同。

– 每组电源都需要一个去耦电容，去耦电容尽量靠近相应电源管脚。

32 电气特性

32.1 测试条件

除非特别说明，所有电压的都以 VSS 为基准。

32.1.1 最小和最大数值

除非特别说明，在生产线上通过对 100%的产品在环境温度 TA=25°C 和 TA=TAmax 下执行的测试(TAmax 与选定的温度范围匹配)，所有最小和最大值将在最坏的环境温度、供电电压和时钟频率条件下得到保证。

在每个表格下方的注解中说明为通过综合评估、设计模拟和/或工艺特性得到的数据，不会在生产线上进行测试；在综合评估的基础上，最小和最大数值是通过样本测试后，取其平均值再加减三倍的标准分布(平均±3Σ)得到。

32.1.2 典型数值

除非特别说明，典型数据是基于 TA=25°C 和 VCC=3.3V(1.8V≤VCC≤5.5V 电压范围)。这些数据仅用于设计指导而未经测试。

典型的 ADC 精度数值是通过对一个标准的批次采样，在所有温度范围下测试得到，95%产品的误差小于等于给出的数值(平均±2Σ)。

32.2 绝对最大额定值

加在器件上的载荷如果超过“绝对最大额定值”列表中给出的值，可能会导致器件永久性地损坏。这里只是给出能承受的最大载荷，并不意味在此条件下器件的功能性操作无误。器件长期工作在最大值条件下会影响器件的可靠性。

符号	描述	最小值	最大值	单位
VCC - VSS	外部主供电电压(包含AVCC和DVCC)(1)	-0.3	5.5	V
V_{IN}	在其它引脚上的输入电压(2)	VSS-0.3	VCC + 0.3	V
| ΔVCCx |	不同供电引脚之间的电压差		50	mV
| VSSx - VSS |	不同接地引脚之间的电压差		50	mV
V_{ESD}(HBM)	ESD静电放电电压(人体模型)	参考绝对最大值电气参数		V

表 32-1 电压特性

1. 所有的电源(DVCC,AVCC)和地(DVSS, AVSS)引脚必须始终连接到外部允许范围内的供电系统上。
2. $I_{INJ(PIN)}$绝对不可以超过它的极限，即保证 V_{IN} 不超过其最大值。如果不能保证 V_{IN} 不超过其最大值，也要保证在外部限制 $I_{INJ(PIN)}$不超过其最大值。当 V_{IN}>VCC 时，有一个正向注入电流；当 V_{IN}<VSS 时，有一个反向注入电流。

符号	描述	最大值(1)	单位
I_{VCC}	经过DVCC/AVCC电源线的总电流(供应电流)(1)	300	mA
I_{VSS}	经过VSS地线的总电流(流出电流)(1)	300	mA
I_{IO}	任意I/O和控制引脚上的输出灌电流	25	mA
	任意I/O和控制引脚上的输出电流	-25	mA
$I_{INJ(PIN)}$(2) (3)	RESETB引脚的注入电流	+/-5	mA
	XTH的XTHI引脚和XTL的XTLI引脚的注入电流	+/-5	mA
	其他引脚的注入电流(4)	+/-5	mA
$\sum I_{INJ(PIN)}$(2)	所有I/O和控制引脚上的总注入电流(4)	+/-25	mA

表 32-2 电流特性

1. 所有的电源(DVCC,AVCC)和地(DVSS,AVSS)引脚必须始终连接到外部允许范围内的供电系统上。
2. $I_{INJ(PIN)}$绝对不可以超过它的极限，即保证 V_{IN} 不超过其最大值。如果不能保证 VIN 不超过其最大值，也要保证在外部限制 $I_{INJ(PIN)}$不超过其最大值。当 V_{IN}>VCC 时，有一个正向注入电流；当 VIN<VSS 时，有一个反向注入电流。
3. 反向注入电流会干扰器件的模拟性能。
4. 当几个 I/O 口同时有注入电流时，$\sum I_{INJ(PIN)}$的最大值为正向注入电流与反向注入电流的即时绝对值之和。该结果基于在器件 4 个 I/O 端口上$\sum I_{INJ(PIN)}$最大值的特性。

符号	描述	数值	单位
TSTG	储存温度范围	-60 ~ + 150	℃
TJ	最大结温度	105	℃

表 32-3 温度特性

32.3 工作条件

32.3.1 通用工作条件

符号	参数	条件	最小值	最大值	单位
fHCLK	内部AHB时钟频率		0	48	MHz
fPCLK0	内部APB0时钟频率		0	48	MHz
fPCLK1	内部APB1时钟频率		0	48	MHz
DVCC	数字部分工作电压		1.8	5.5	V
AVCC(1)	模拟部分工作电压	必须与DVCC(2)相同	1.8	5.5	V
PD	功率耗散 TA=85℃	LQFP100		476	mW
	功率耗散 TA=85℃	LQFP80		465	mW
	功率耗散 TA=85℃	LQFP64		455	mW
	功率耗散 TA=85℃	LQFP48		364	mW
	功率耗散 TA=85℃	QFN32		357	mW
TA	环境温度	最大功率消耗	-40	85	℃
		低功率消耗(3)	-40	105	℃
TJ	结温度范围		-40	105	℃

表 32-4 通用工作条件

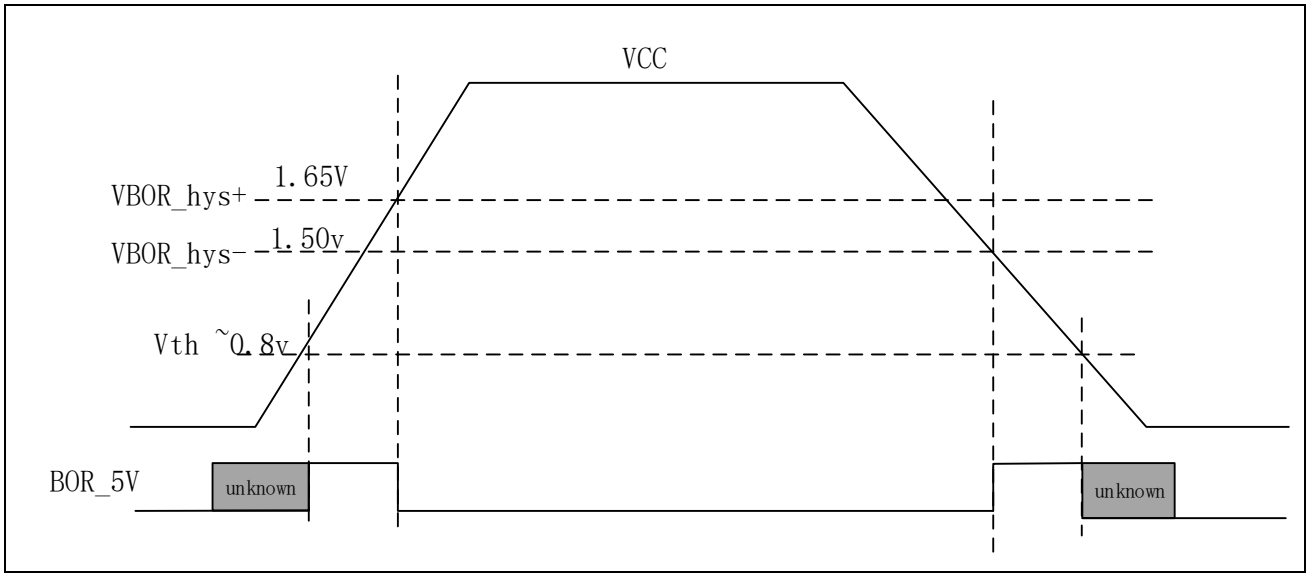
1. 当使用 ADC 时，参见 ADC 电气参数。
2. 建议使用相同的电源为 DVCC 和 AVCC 供电，在上电和正常操作期间，DVCC 和 AVCC 之间最多允许有 300mV 的差别。
3. 在较低的功率耗散的状态下，只要 T_J 不超过 T_{Jmax}，T_A 可以扩展到这个范围。

32.3.2 上电和掉电时的工作条件

符号	参数	条件	最小值	最大值	单位
tVcc	VCC上升速率		0	5	V/μs
tVcc	VCC下降速率		0	5	V/μs

表 32-5 上电和掉电的工作条件

32.3.3 内嵌复位和 LVD 模块特性



1. 设计保证，不在生产中测试。

图 32-1 POR/Brown Out 示意图

符号	参数	条件	最小值	典型值	最大值	单位
Vpor	POR 释放电压（上电过程） BOR 检测电压（掉电过程）		1.45	1.50	1.65	V

表 32-6 POR/Brown Out

符号	参数	条件	最小值	典型值	最大值	单位
Vex	外部输入电压范围		0		VCC	V
Vlevel	检测阈值	LVD_CR.VTDS=0000 LVD_CR.VTDS =0001 LVD_CR.VTDS =0010 LVD_CR.VTDS =0011 LVD_CR.VTDS =0100 LVD_CR.VTDS=0101 LVD_CR.VTDS=0110 LVD_CR.VTDS=0111 LVD_CR.VTDS=1000 LVD_CR.VTDS=1001 LVD_CR.VTDS=1010 LVD_CR.VTDS=1011 LVD_CR.VTDS=1100 LVD_CR.VTDS=1101 LVD_CR.VTDS=1110 LVD_CR.VTDS=1111		1.8±3.5% 1.9±3.5% 2.0±3.5% 2.1±3.5% 2.2±3.5% 2.3±3.5% 2.4±3.5% 2.5±3.5% 2.6±3.5% 2.7±3.5% 2.8±3.5% 2.9±3.5% 3.0±3.5% 3.1±3.5% 3.2±3.5% 3.3±3.5%		V
Icomp	功耗			0.12		uA
Tresponse	响应时间			80		μs
Tsetup	建立时间			400		μs
Vhyste	迟滞电压			40		mV
Tfilter	滤波时间	LVD_debounce = 000 LVD_debounce = 001 LVD_debounce = 010 LVD_debounce = 011 LVD_debounce = 100 LVD_debounce = 101 LVD_debounce = 110 LVD_debounce = 111		7 14 28 112 450 1800 7200 28800		μs

表 32-7 LVD 模块特性

32.3.4 内置的参考电压

符号	参数	条件	最小值	典型值	最大值	单位
V_{REF25}	Internal 2.5V Reference Voltage	常温25°C 3.3V	2.475	2.5	2.525	V
V_{REF25}	Internal 2.5V Reference Voltage	-40~85°C; 2.8~5.5V	2.463	2.5	2.525	V[1]
V_{REF15}	Internal 1.5V Reference Voltage	常温25°C 3.3V	1.485	1.5	1.515	V
V_{REF15}	Internal 1.5V Reference Voltage	-40~85°C; 1.8~5.5V	1.477	1.5	1.519	V[1]
T_{Coeff}	Internal 2.5V 1.5V temperature coefficient	-40 ~ 85°C			120	ppm/°C

1. 数据基于考核结果，不在生产中测试

32.3.5 供电电流特性

电流消耗是多种参数和因素的综合指标，这些参数和因素包括工作电压、环境温度、I/O 引脚的负载、产品的软件配置、工作频率、I/O 脚的翻转速率、程序在存储器中的位置以及执行的代码等。

微控制器处于下列条件：

- 所有的 I/O 引脚都处于输入模式，并连接到一个静态电平上——VCC 或 VSS(无负载)。
- 所有的外设都处于关闭状态，除非特别说明。
- 闪存存储器的访问时间调整到 fHCLK 的频率(0~24MHz 时为 0 个等待周期，24~48MHz 时为 1 个等待周期)。
- 当开启外设时：fPCLK0 = fHCLK，fPCLK1 = fHCLK。

Symbol	Parameter	Conditions			Typ(1)	Max(2)	Unit
I_{DD} (Run in RAM)	All peripherals clock ON, Run while(1) in RAM	VCAP=1.5V V_{CC}=3.3V T_A=2xC	RCH clock source	4M	750		uA
				8M	1460		
				16M	2850		
				22.12M	3940		
				24M	4270		
			PLL RCH4M to xxM clock source	32M	5750		
				48M	8540		
				4M	350		uA

	All peripherals clock OFF, Run while(1) in RAM	VCAP=1.5V V_{CC}=3.3V T_A=2xC	RCH clock source	8M	660		
				16M	1250		
				22.12M	1710		
				24M	1850		
			PLL RCH4M to xxM clock source	32M	2560		
				48M	3770		
I_{DD} (Run CoreMark)	All peripherals clock OFF, Run CoreMark in Flash	VCAP=1.5V V_{CC}=3.3V T_A=2xC	RCH clock source	4M	790		uA
				8M	1470		
				16M	2780		
				22.12M	3720		
				24M	4000		
			PLL RCH4M to xxM	48M FlashWait=1	6080		
I_{DD} (Run mode)	All peripherals clock ON, Run while(1) in Flash	VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	RCH clock source	4M	1000	1440	uA
				8M	1890	2710	
				16M	3710	5160	
				22.12M	5010	7010	
				24M	5400	7570	
		VCAP=1.5V V_{CC} =1.8-5.5V T_A=N40C-85C	PLL RCH4M to xxM clock source	16M	3930	5000	uA
				24M	5480	7090	
				32M FlashWait=1	6590	7650	
				40M FlashWait=1	8100	9470	
				48M FlashWait=1	9610	11200	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH8M to xxM clock source	16M	3990	5040	uA
				24M	5530	7140	
				32M FlashWait=1	6640	7690	
				40M FlashWait=1	8160	9480	
				48M FlashWait=1	9670	11250	
	All peripherals clock OFF, Run while(1) in Flash	VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	RCH clock source	4M	610	1000	uA
				8M	1090	1840	
				16M	2080	3360	
				22.12M	2770	4480	
				24M	2970	4810	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH4M to xxM clock source	16M	2290	3150	uA
				24M	3060	4370	
				32M FlashWait=1	3410	4030	
				40M FlashWait=1	4110	4950	

				48M FlashWait=1	4860	5870	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH8M to xxM clock source	16M	2340	3220	uA
				24M	3120	4420	
				32M FlashWait=1	3460	4080	
				40M FlashWait=1	4160	4990	
				48M FlashWait=1	4910	5910	
I_{DD} (Sleep mode)	All peripherals clock ON	VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	RCH clock source	4M	550	630	uA
				8M	1060	1190	
				16M	2050	2290	
				22.12M	2830	3160	
				24M	3070	3420	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH4M to xxM clock source	16M	2290	2560	uA
				24M	3200	3600	
				32M FlashWait=1	4190	4720	
				40M FlashWait=1	5200	5860	
				48M FlashWait=1	6190	6990	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH8M to xxM clock source	16M	2350	2620	uA
				24M	3250	3660	
				32M FlashWait=1	4240	4770	
				40M FlashWait=1	5250	5890	
				48M FlashWait=1	6250	7020	
	All peripherals clock OFF	VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	RCH clock source	4M	150	190	uA
				8M	260	320	
				16M	450	530	
				22.12M	610	710	
				24M	650	750	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH4M to xxM clock source	16M	690	780	uA
				24M	790	900	
				32M FlashWait=1	990	1110	
				40M FlashWait=1	1200	1350	
				48M FlashWait=1	1410	1580	
		VCAP=1.5V V_{CC}=1.8-5.5V T_A=N40C-85C	PLL RCH8M to xxM clock source	16M	740	850	uA
				24M	840	960	
				32M FlashWait=1	1040	1170	
				40M FlashWait=1	1250	1400	
				48M FlashWait=1	1460	1640	
I_{DD} (LP Run)	All peripherals clock ON, Run while(1) in Flash	VCAP=1.5V V_{CC}=1.8-5.5V	XTL32K clock source Driver=0x0	T_A=N40-25C	13	22	uA
				T_A=50C	14	21	
				T_A=85C	21	29	

	All peripherals clock OFF, Run while(1) in Flash	VCAP=1.5V V_{CC}=1.8-5.5V	XTL32K clock source Driver=0x0	T_A=N40-25C	10	18	uA
				T_A=50C	11	17	
				T_A=85C	18	26	
I_{DD} (LP Sleep)	All peripherals clock ON	VCAP=1.5V V_{CC}=1.8-5.5V	XTL32K clock source Driver=0x0	T_A=N40-25C	8	9	uA
				T_A=50C	9	10	
				T_A=85C	16	19	
	All peripherals clock OFF	VCAP=1.5V V_{CC}=1.8-5.5V	XTL32K clock source Driver=0x0	T_A=N40-25C	5	5	uA
				T_A=50C	6	7	
				T_A=85C	13	16	
I_{DD} (Deep Sleep)	XTL32K +DeepSleep	VCAP=1.5V V_{CC}=1.8-5.5V	XTL32K Driver=0x0	T_A=N40-25C	2980	3170	nA
				T_A=50C	3720	4110	
				T_A=85C	8380	9790	
	IRC32K +DeepSleep	VCAP=1.5V V_{CC}=1.8-5.5V		T_A=N40-25C	2920	3100	nA
				T_A=50C	3660	4010	
				T_A=85C	8320	9650	
	WDT +DeepSleep	VCAP=1.5V V_{CC}=1.8-5.5V		T_A=N40-25C	2710	2840	nA
				T_A=50C	3420	3740	
				T_A=85C	8080	9500	
	DeepSleep	VCAP=1.5V V_{CC}=1.8-5.5V		T_A=N40-25C	2600	2730	nA
				T_A=50C	3320	3630	
				T_A=85C	7980	9360	

1. 若没有其他指定条件，该 Typ 的值是在 25 °C & V_{CC} = 3.3V 测得。

2. 若没有其他指定条件，该 Max 的值是 V_{CC} = 1.8-5.5 & Temperature = N40 - 85 °C 范围内的最大值。

3. 数据基于考核结果，不在生产中测试

表 32-1 工作电流特性

32.3.6 从低功耗模式唤醒的时间

唤醒时间是在 RCH 振荡器的唤醒阶段测量得到。唤醒时使用的时钟源依当前的操作模式而定：

- 休眠模式：时钟源是 RCH 振荡器
- 深度休眠模式：时钟源是进入深度休眠时所使用的时钟是 RCH 振荡器

Symbol	Papameter	Conditions	Min	Typ	Max	Unit
T_{wu}	休眠模式唤醒时间			1.8		μs
	深度休眠唤醒时间	F_{MCLK} = 4MHz		9.0		μs
		F_{MCLK} = 8MHz		6.0		μs
		F_{MCLK} = 16MHz		5.0		μs
		F_{MCLK} = 24MHz		4.0		μs

1．唤醒时间的测量是从唤醒事件开始至用户程序读取第一条指令。

32.3.7 外部时钟源特性

32.3.7.1 外部输入高速时钟

符号	参数	条件	最小值	典型值	最大值	单位
fXTH_ext	用户外部时钟频率(1)		0	8	32	MHz
VXTHH	输入引脚高电平电压		0.7VCC		VCC	V
VXTHL	输入引脚低电平电压		VSS		0.3VCC	V
Tr(XTH)	上升的时间(1)				20	ns
Tf(XTH)	下降的时间(1)				20	ns
Tw(XTH)	输入高或低的时间(1)		16			ns
Cin(XTH)	输入容抗(1)			5		pF
Duty	占空比		40		60	%
IL	输入漏电流				±1	μA

由设计保证，不在生产中测试。

32.3.7.2 外部输入低速时钟

符号	参数	条件	最小值	典型值	最大值	单位
fXTH_ext	用户外部时钟频率(1)		0	32.768	1000	KHz
VXTHH	输入引脚高电平电压		0.7VCC		VCC	V
VXTHL	输入引脚低电平电压		VSS		0.3VCC	V
Tr(XTH)	上升的时间(1)				50	ns
Tf(XTH)	下降的时间(1)				50	ns
Tw(XTH)	输入高或低的时间(1)		450			ns
Cin(XTH)	输入容抗(1)			5		pF
Duty	占空比		30		70	%
IL	输入漏电流				±1	μA

由设计保证，不在生产中测试。

32.3.7.3 高速外部时钟 XTH

高速外部时钟(XTH)可以使用一个 4~32MHz 的晶体/陶瓷谐振器构成的振荡器产生。本节中所给出的信息是基于使用下表中列出的典型外部元器件，通过综合特性评估得到的结果。在应用中，谐振器和负载电容必须尽可能地靠近振荡器的引脚，以减小输出失真和启动时的稳定时间。有关晶体谐振器的详细参数(频率、封装、精度等)，请咨询相应的生产厂商。

外部 XTH 晶振(1) (2)

符号	参数	条件	最小值	典型值	最大值	单位
F_{CLK}	振荡频率		4		32	MHz
ESR_{CLK}	支持的晶振ESR范围	32M		30	60	Ohm
		4M		400	1500	Ohm
C_{LX}(3)	负载电容	按晶体制造商要求进行配置。				
Duty	占空比		40	50	60	%
Idd(4)	电流	32M Xtal, C_L=12pF, ESR=30ohm		600		uA
T_{start}(5)	启动时间	32MHz @ XTH_CR.Driver=1111		300		μs
		4MHz @ XTH_CR.Driver=0011		2		ms

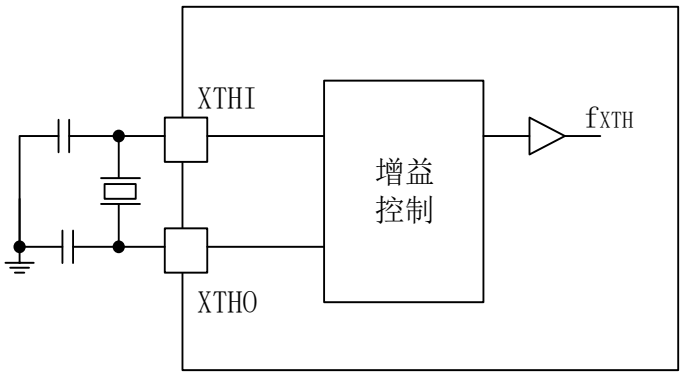
1. 谐振器的特性参数由晶体/陶瓷谐振器制造商给出。
2. 由综合评估得出，不在生产中测试。
3. C_{LX} 指 XTAL 的两个管脚的负载电容，用户***必须***按晶体制造商的要求选择该电容的容值。

   如果晶体制造商给出了***负载电容的容值***，则匹配电容的容值应为晶体制造商所给出的负载电容容值的两倍。

   如果晶体制造商给出了***匹配电容的容值***，则直接使用晶体制造商所给出的匹配电容的容值即可。

   例：晶体制造商给出晶体的***负载电容***为 8pF 时，匹配电容的容值应为 16pF。考虑 PCB 与 MCU 引脚之间的分布电容，建议选择容值为 15pF 或 12pF 的匹配电容。

   晶体制造商给出晶体的***匹配电容***为 12pF 时，匹配电容的容值应为 12pF。考虑 PCB 与 MCU 引脚之间的分布电容，建议选择容值为 10pF 或 8pF 的匹配电容。
4. 电流跟随频率变化而变化，测试条件：XTH_CR.Driver=1110
5. Tstart 是启动时间，是从软件使能 XTH 开始测量，直至得到稳定的 32MHz/4MHz 振荡这段时间。这个数值是在 XTH_CR.Startup=10 设置下，使用一个标准的晶体谐振器上测量得到，它可能因晶体制造商和型号的不同而变化较大。



注意：

– 晶体的匹配电容**必须**按照晶体制造商的技术手册的要求进行配置。

如果晶体制造商给出了***负载电容的容值***，则匹配电容的容值应为晶体制造商所给出的负载电容容值的两倍。

如果晶体制造商给出了***匹配电容的容值***，则直接使用晶体制造商所给出的匹配电容的容值即可。

32.3.7.4 低速外部时钟 XTL

低速外部时钟(XTL)可以使用一个 32.768KHz 的晶体/陶瓷谐振器构成的振荡器产生。本节中所给出的信息是基于典型外部元器件，通过综合特性评估得到的结果。在应用中，谐振器和负载电容必须尽可能地靠近振荡器的引脚，以减小输出失真和启动时的稳定时间。有关晶体谐振器的详细参数(频率、封装、精度等)，请咨询相应的生产厂商。

外部 XTL 晶振(1)

符号	参数	条件	最小值	典型值	最大值	单位
F_{CLK}	振荡频率			32.768		KHz
ESR_{CLK}	支持的晶振ESR范围			65	85	KOhm
C_{Lx}(2)	负载电容	按晶体制造商要求进行配置。				
DC_{ACLK}	占空比		30	50	70	%
Idd(3)	电流	ESR= 65 KOhm C_L=12 pF		350	1000	nA
T_{start}(4)	启动时间	ESR=65 KOhm, C_L=12 pF, 40% - 60% duty cycle has been reached		500		ms

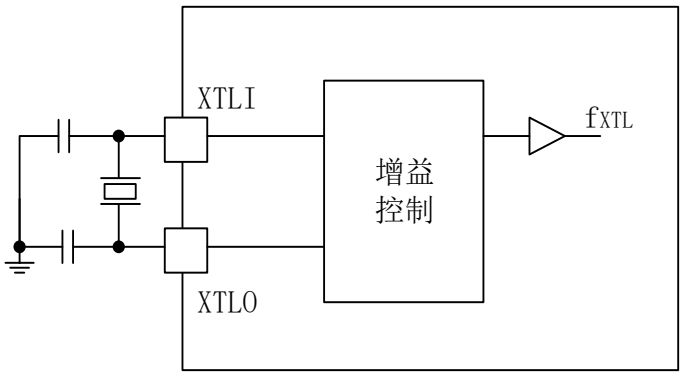
1. 由综合评估得出，不在生产中测试。
2. C_{LX} 指 XTAL 的两个管脚的负载电容，用户***必须***按晶体制造商的要求选择该电容的容值。

   如果晶体制造商给出了***负载电容的容值***，则匹配电容的容值应为晶体制造商所给出的负载电容容值的两倍。

   如果晶体制造商给出了***匹配电容的容值***，则直接使用晶体制造商所给出的匹配电容的容值即可。

   例：晶体制造商给出晶体的***负载电容***为 8pF 时，匹配电容的容值应为 16pF。考虑 PCB 与 MCU 引脚之间的分布电容，建议选择容值为 15pF 或 12pF 的匹配电容。

   晶体制造商给出晶体的***匹配电容***为 12pF 时，匹配电容的容值应为 12pF。考虑 PCB 与 MCU 引脚之间的分布电容，建议选择容值为 10pF 或 8pF 的匹配电容。
3. 典型值为XTL_CR.Driver=1001时的功耗。选择具有较小ESR值的高质量振荡器，可以通过减小XTL_CR.Driver设置值以优化电流消耗。
4. Tstart 是启动时间，是从软件使能 XTL 开始测量，直至得到稳定的 32768 振荡这段时间。这个数值是在 **XTL_CR.Driver=1001 和 XTL_CR.Startup=10** 设置下，使用一个标准的晶体谐振器上测量得到，它可能因晶体制造商和型号的不同而变化较大。



注意：

– 晶体的匹配电容***必须***按照晶体制造商的技术手册的要求进行配置。

  如果晶体制造商给出了***负载电容的容值***，则匹配电容的容值应为晶体制造商所给出的负载电容容值的两倍。

  如果晶体制造商给出了***匹配电容的容值***，则直接使用晶体制造商所给出的匹配电容的容值即可。

32.3.8 内部时钟源特性

32.3.8.1 内部 RCH 振荡器

Symbol	Papameter	Conditions	Min	Typ	Max	Unit
Dev	RCH振荡器精度	User trimming step for given VCC and TA conditions		0.25		%
		VCC = 1.8 ~ 5.5V T_{AMB} = -40 ~ 85°C	-2.5		+2.5	%
		VCC = 1.8 ~ 5.5V T_{AMB} = -20 ~ 50°C	-2.0		+2.0	%
F_{CLK}	振荡频率		4.0	4.0 8.0 16.0 22.12 24.0	24.0	MHz
I_{CLK}	功耗	F_{MCLK} = 4MHz		80		μA
		F_{MCLK} = 8MHz		100		μA
		F_{MCLK} = 16MHz		120		μA
		F_{MCLK} = 24MHz		140		μA
DC_{CLK}	占空比(1)		45	50	55	%

1. 由综合评估得出，不在生产中测试。

32.3.8.2 内部 RCL 振荡器

Symbol	Papameter	Conditions	Min	Typ	Max	Unit
Dev	RCL振荡器精度	User trimming step for given VCC and TA conditions		0.5		%
		VCC = 1.8 ~ 5.5V T_{AMB} = -40 ~ 85°C	-2.5		+2.5	%
		VCC = 1.8 ~ 5.5V T_{AMB} = -20 ~ 50°C	-1.5		+1.5	%
F_{CLK}	振荡频率			38.4 32.768		KHz
T_{CLK}	启动时间			150		μs
DC_{CLK}	占空比(1)		25	50	75	%
I_{CLK}	功耗			0.35		μA

1. 由综合评估得出，不在生产中测试。

32.3.9 PLL 特性

符号	参数	条件	最小值	典型值	最大值	单位
Fin(1)	输入时钟		4	4	24	MHz
	输入时钟占空比		40		60	%
Fout	输出频率		8	-	48	MHz
Duty(1)	输出占空比		48%	-	52%	
Tlock(1)	锁定时间	输入频率4MHz	-	100	200	μs

1. 由综合评估得出，不在生产中测试。

32.3.10 存储器特性

符号	参数	条件	最小值	典型值	最大值	单位
EC_{FLASH}	擦写次数	Regulator voltage=1.5V, T_{AMB} = 25℃	20K			cycles
RET_{FLASH}	数据保存期限	T_{AMB} = 85℃	20			Years
		常温	100			Years
T_{b_prog}	编程时间（字节）		22		30	μs
T_{w_prog}	编程时间（字）		40		52	μs
T_{p_erase}	页擦除时间		4		5	ms
T_{m_erase}	整片擦除时间		30		40	ms

32.3.11 EFT 特性

芯片复位可以使系统恢复正常操作。

符号	级别/类型
EFT to IO (IEC61000-4-4)	2KV Class:4
EFT to Power (IEC61000-4-4)	4KV Class:4

软件建议

软件的流程中必须包含程序跑飞的控制，如：

- 被破坏的程序计数器
- 意外的复位
- 关键数据被破坏(控制寄存器等)

在进行 ESD 测试时，可以把超出应用要求的电压直接施加在芯片上，当检测到意外动作的地方，软件部分需要加强以防止发生不可恢复的错。

32.3.12 ESD 特性

使用特定的测量方法，对芯片进行强度测试以决定它的电气敏感性方面的性能。

符号	参数	条件	最小值	典型值	最大值	单位
$VESD_{HBM}$	ESD @ Human Body Mode			4		KV
$VESD_{CDM}$	ESD @ Charge Device Mode			1		KV
$VESD_{MM}$	ESD @ machine Mode			200		V
Ilatchup	Latch up current			200		mA

32.3.13 I/O 端口特性

32.3.13.1 输出特性——端口

符号	参数	条件	最小值	最大值	单位
V_{OH}	High level output voltage Source Current	Sourcing 4 mA, VCC = 3.3 V (see Note 1)	VCC-0.25		V
		Sourcing 8 mA, VCC = 3.3 V (see Note 2)	VCC-0.6		V
V_{OL}	Low level output voltage Sink Current	Sinking 5 mA, VCC = 3.3 V (see Note 1)		VSS+0.25	V
		Sinking 14 mA, VCC = 3.3 V (see Note 2)		VSS+0.6	V
V_{OHD}	High level output voltage Double source Current	Sourcing 8 mA, VCC = 3.3 V (see Note 1)	VCC-0.25		V
		Sourcing 18 mA, VCC = 3.3V (see Note 2)	VCC-0.6		V
V_{OLD}	Low level output voltage Double Sink Current	Sinking 8 mA, VCC = 3.3 V (see Note 1)		VSS+0.25	V
		Sinking 18 mA, VCC = 3.3 V (see Note 2)		VSS+0.6	V

表 32-4 端口输出特性

NOTES: 1. The maximum total current, I_{OH}(max) and I_{OL}(max), for all outputs combined, should not exceed 40 mA to satisfy the maximum specified voltage drop.

2. The maximum total current, I_{OH}(max) and I_{OL}(max), for all outputs combined, should not exceed 100 mA to satisfy the maximum specified voltage drop.

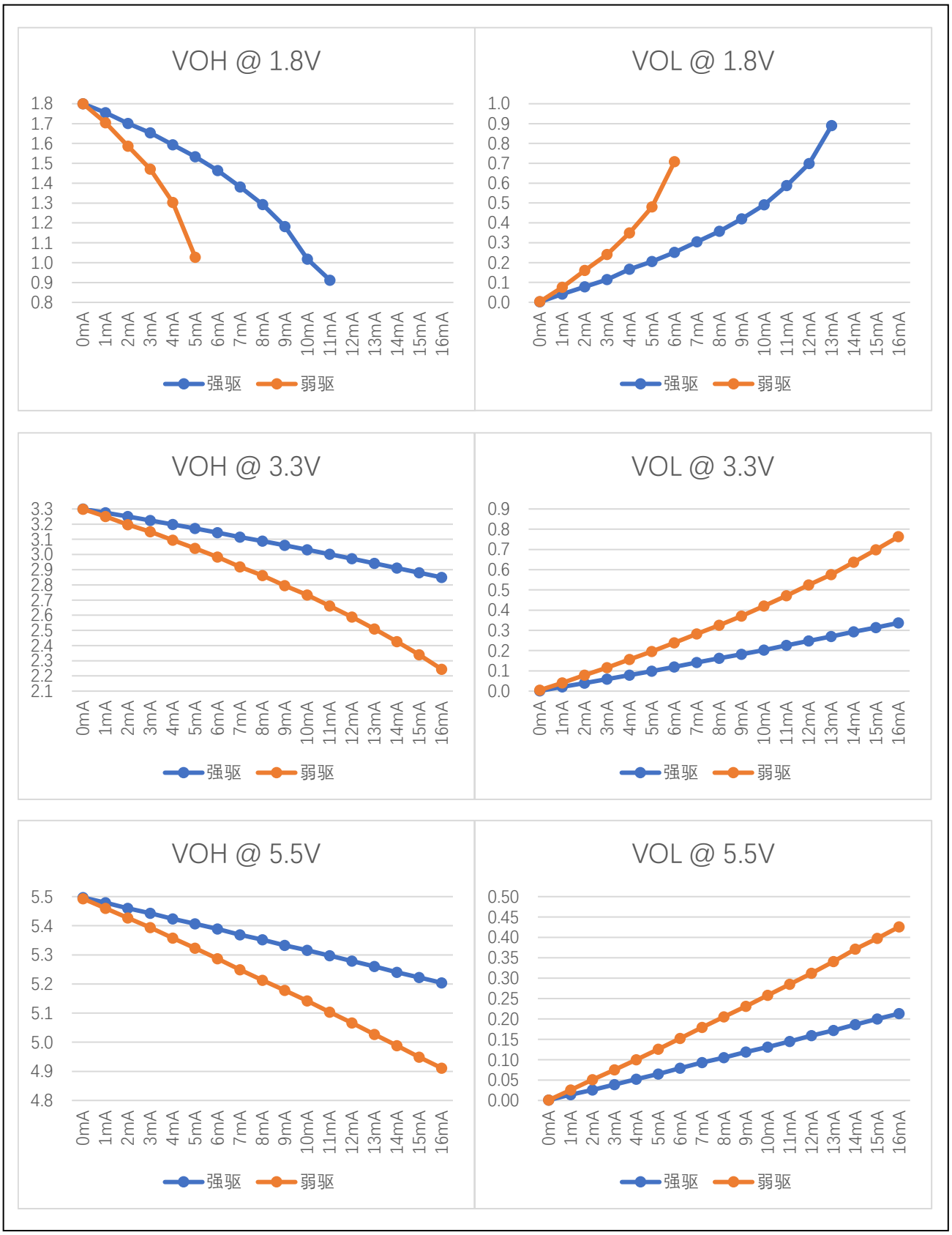


图 32-2 输出端口 VOH/VOL 实测曲线

32.3.13.2 输入特性——端口 PA,PB,PC,PD,PE,PF,RESET

符号	参数	条件	最小值	典型值	最大值	单位
V_{IH}	Positive-going input threshold voltage	VCC=1.8V	1.2			V
		VCC=3.3V	2.0			V
		VCC=5.5V	3.3			V
V_{IL}	Negative-going input threshold voltage	VCC=1.8V			0.5	V
		VCC=3.3V			1.0	V
		VCC=5.5V			1.6	V
$V_{hys(1)}$	Input voltage hysteresis (V_{IH} - V_{IL})	VCC=1.8V		0.3		V
		VCC=3.3V		0.4		V
		VCC=5.5V		0.6		V
$R_{pullhigh}$	Pullup resistor	Pullup enabled VCC=3.3V		80		Kohm
$R_{pulllow}$	Pulldown resistor	Pulldown enabled VCC=3.3V		40		Kohm
C_{input}	Input capacitance			5		pF

1. 由综合评估得出，不在生产中测试。

32.3.13.3 端口外部输入采样要求——Timer Gate/Timer Clock

符号	参数	条件	最小值	典型值	最大值	单位
t(int)	External interrupt timing	External trigger signal for the interrupt flag (see Note 1)	1.8V	30		ns
			3.3V	30		ns
			5.5V	30		ns
t(cap)	Timer capture timing	Timer4/5/6 capture pulse width Fsystem = 4MHz	1.8V	0.5		μs
			3.3V	0.5		μs
			5.5V	0.5		μs
t(clk)	Timer clock frequency applied to pin	Timer0/1/2/4/5/6 external clock input Fsystem = 4MHz	1.8V		PCLK/2	MHz
			3.3V		PCLK/2	MHz
			5.5V		PCLK/2	MHz
t(pca)(2)	PCA clock frequency applied to pin	PCA external clock input Fsystem = 4MHz	1.8V		PCLK/8	MHz
			3.3V		PCLK/8	MHz
			5.5V		PCLK/8	MHz

NOTES: 1. The external signal sets the interrupt flag every time the minimum $t_{(int)}$ parameters are met. It may be set even with trigger signals shorter than $t_{(int)}$.

2. 由综合评估得出，不在生产中测试。

32.3.13.4 端口漏电特性——PA,PB,PC,PD,PE,PF

符号	参数	条件	最小值	典型值	最大值	单位
$I_{lkg(Px.y)}$	Leakage current	$V_{(Px.y)}$ (see Note 1,2)		±50		nA

NOTES: 1. The leakage current is measured with VSS or VCC applied to the corresponding pin(s), unless otherwise noted.

2. The port pin must be selected as input.

32.3.14 RESETB 引脚特性

RESETB 引脚输入驱动使用 CMOS 工艺，它连接了一个不能断开的上拉电阻。

符号	参数	条件	最小值	典型值	最大值	单位
$V_{IL(RESETB)}$ (1)	输入低电平电压		-0.3		0.8	
$V_{IH(RESETB)}$	输入高电平电压		0.8*VCC		VCC+0.5	
$V_{hys(RESETB)}$	施密特触发器电压迟滞			200		mV
R_{PU}	弱上拉等效电阻	$V_{IN} = V_{SS}$		80		KΩ
$V_{F(RESETB)}$ (1)	输入滤波脉冲				100	ns
$V_{NF(RESETB)}$ (1)	输入非滤波脉冲		300			ns

1．由设计保证，不在生产中测试。

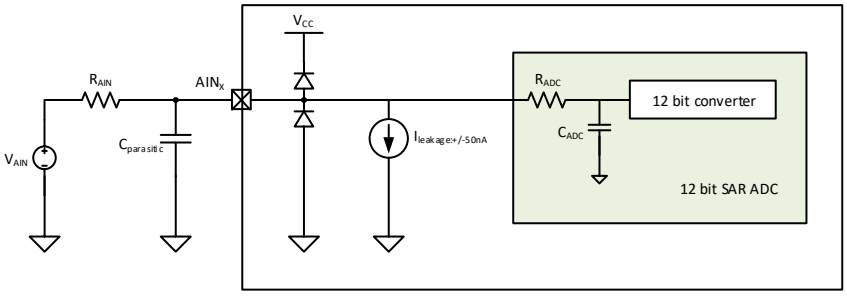
32.3.15 ADC 特性

符号	参数	条件	最小值	典型值	最大值	单位
V_{ADCIN}	Input voltage range	Single ended	0		$V_{ADCREFIN}$	V
$V_{ADCREFIN}$	Input range of external reference voltage	Single ended	0		AVCC	V
I_{ADC1}	Active current including reference generator and buffer	200Ksps		2		mA
I_{ADC2}	Active current without reference generator and buffer	1Msps		0.5		mA
C_{ADCIN}	ADC input capacitance			16	19.2	pF
R_{ADC}(1)	ADC sampling switch impedance			1.5		KOhm
R_{AIN}(1)	ADC external input resistor(2)				100	KOhm
F_{ADCCLK}	ADC clock Frequency				24M	Hz
$T_{ADCSTART}$	Startup time of reference generator and ADC core			20		μs

$T_{ADCCONV}$	Conversion time		20	24	28	cycles
ENOB	Effective Bits	1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.8V REF=EXREF		10.3		Bit
		1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.8V REF=VCC		10.3		Bit
		200Ksps@VCC>=1.8V REF=internal 1.5V		9.4		Bit
		200Ksps@VCC>=2.8V REF=internal 2.5V		9.4		Bit
SNR	Signal to Noise Ratio	1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.8V REF=EXREF		68.2		dB
		1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.8V REF=VCC		68.2		dB
		200Ksps@VCC>=1.8V REF=internal 1.5V		60		dB
		200Ksps@VCC>=2.8V REF=internal 2.5V		60		dB
DNL(1)	Differential non-linearity	200Ksps； VREF=EXREF/AVCC	-1		1	LSB
INL(1)	Integral non-linearity	200Ksps； VREF=EXREF/AVCC	-3		3	LSB
E_o	Offset error			0		LSB
E_g	Gain error			0		LSB

1．由设计保证，不在生产中测试。

2．ADC 的典型应用如下图所示：



对于 0.5LSB 采样误差精度要求的条件下，外部输入阻抗的计算公式如下：

$$R_{AIN} = \frac{M}{F_{ADC} * C_{ADC} * (N + 1) * \ln(2)} - R_{ADC}$$

其中F_{ADC}为 ADC 时钟频率，寄存器 ADC_CR0<3:2>可设定其与 PCLK 的关系，如下表。

下表为 ADC 时钟频率F_{ADC}和 PCLK 分频比关系：

ADC_CR0<3:2>	N
00	1
01	2
10	4
11	8

M 为采样周期个数，由寄存器 ADC_CR0<13:12>设定。

下表为采样时间t_{sa}和 ADC 时钟频率F_{ADC}的关系：

ADC_CR0<13:12>	M
00	4
01	6
10	8
11	12

下表为 ADC 时钟频率F_{ADC}和外部电阻R_{AIN}的关系（M=12,采样误差 0.5LSB 的条件下）：

R_{AIN}（KOhm）	F_{ADC}(KHz)
10	5600
30	2100
50	1300
80	820
100	660
120	550
150	450

对于上述典型应用，应注意：

- 尽量减小 ADC 输入端口AIN_X的寄生电容$C_{PARACITIC}$；
- 除了考虑R_{AIN}值外，如果信号源V_{AIN}的内阻较大时，也需要加入考虑。

32.3.16 VC 特性

符号	参数	条件	最小值	典型值	最大值	单位
Vin	Input voltage range		0		5.5	V
Vincom	Input common mode range		0		VCC-0.2	V
Voffset	Input offset	常温25°C 3.3V	-10		+10	mV
Icomp	Comparator’s current	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		0.3 1.2 10 20		uA
Tresponse	Comparator’s response time when one input cross another	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		20 5 1 0.2		μs
Tsetup	Comparator’s setup time when ENABLE. Input signals unchanged.	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		20 5 1 0.2		μs
Twarmup	From main bandgap enable to 1.2V BGR reference、Temp sensor voltage、ADC internal 1.5V、2.5V reference stable			20		μs
Tfilter	Digital filter time	VC_debounce = 000 VC_debounce = 001 VC_debounce = 010 VC_debounce = 011 VC_debounce = 100 VC_debounce = 101 VC_debounce = 110 VC_debounce = 111		7 14 28 112 450 1800 7200 28800		μs

32.3.17 OPA 特性

OPA：(AVCC=2.2 ~ 5.5 V, AVSS=0 V, Ta=- 40 ~ +85℃)

符号	参数	条件	最小值	典型值	最大值	单位
Vi	输入电压		0	-	AVCC	V
Vo	输出电压(1)		0.1	-	AVCC-0.2	V
Io	输出电流(1)				1	mA
RL	负载电阻(1)		5K			Ohm
Tstart	初始化时间(2)				20	μs
Vio	输入失调电压	Vic=AVCC/2, Vo=AVCC/2, RL=5KΩ, Rs=50 pF		±6		mV
PM	相位裕度(1)	Vic=AVCC/2, Vo=AVCC/2 RL=5KΩ, CL=50pF		80	-	deg
UGBW	单位增益带宽(1)	Vic=AVCC/2, Vo=AVCC/2 RL=5KΩ, CL=50pF		9.3		MHz
SR	压摆率(1)	RL=5KΩ, CL=50pF		8		V/μs

1．由设计保证，不在生产中测试。

2．需要同时设置 BGR_CR<0>=1

32.3.18 LCD 控制器

符号	参数	工作条件	最小	典型	最大	单位
I_{LCD}	电流	VCC=3.3V		3.3		uA
RH	低驱动电阻			1M		Ω
RL	高驱动电阻			360K		Ω
VLCDH	LCD 可调最高电压				VCC	V
VLCD3	LCD 最高电压				VLCDH	V
VLCD2	LCD 2/3电压				2/3 VLCDH	V
VLCD1	LCD 1/3电压				1/3 VLCDH	V
VLCD0	LCD 最低电压		0			V
$\triangle V_{XX}$	LCD 电压偏差	TA=-40~85℃		±5%		

32.3.19 DAC 特性

符号	参数	工作条件	最小	典型	最大	单位
V_{DACOUT}	Output voltage range	AVDD voltage reference, single ended	0		Vcc	V
V_{DACCM}	Output common mode voltage range		0		Vcc	V
I_{DAC}	Active current	500KSamples/s		15u		uA
SR_{DAC}	Sample rate				500	Ksps
$t_{DACCONV}$	Conversion time		2			μs
$t_{DACSETTLE}$	Setting time			5		μs
SNR_{DAC}	Signal to Noise Ratio			59		dB
$SNDR_{DAC}$	Signal to Noise and Distortion Ratio			57		dB
$SFDR_{DAC}$	Spurious Free Dynamic Range			56		dB
$V_{DACOFFSET}$	Offset voltage	w/o buffer		2		mV
DNL_{DAC}	Differential non-linearity			±1		LSB
INL_{DAC}	Integral non-linearity			±5		LSB

33 唯一设备 ID 寄存器（80 位）

唯一设备标识符最适合：

- 用作序列号
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

80 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。

80 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

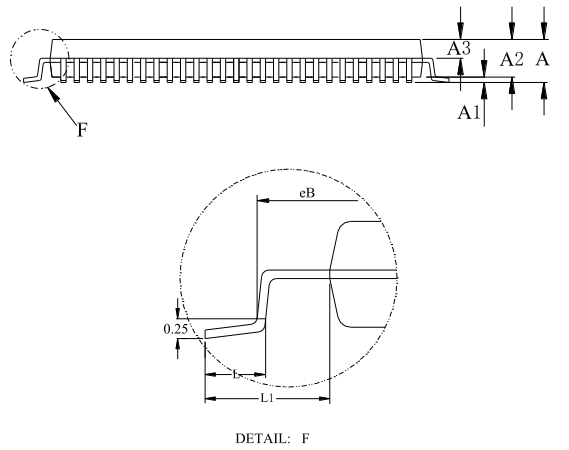
基址：0x0010 0E74

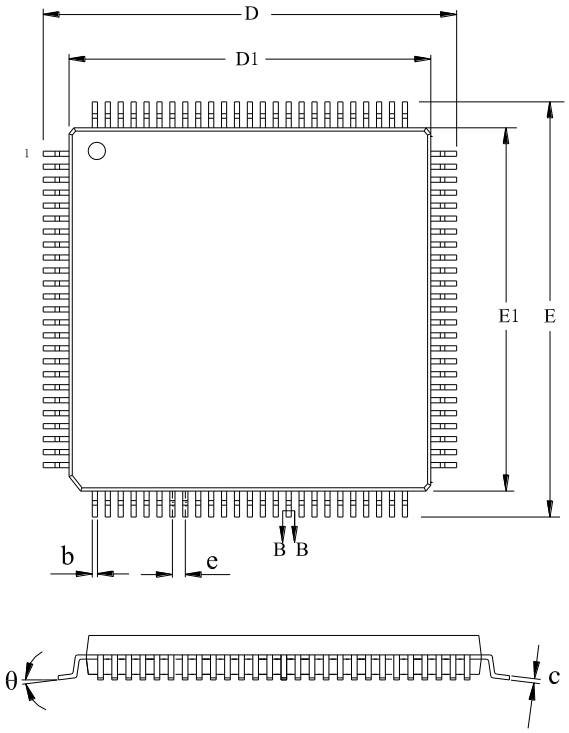
UID 规则

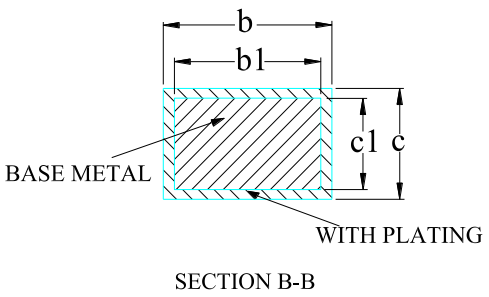
偏移地址	描述	UID Bits（80 bits）							
		7	6	5	4	3	2	1	0
0	Lot Number	UID[7:0]							
1		UID[15:8]							
2		UID[23:16]							
3		UID[31:24]							
4		UID[39:32]							
5		UID[47:40]							
6	X Coordinate on the wafer	UID[55:48]							
7	Y Coordinate on the wafer	UID[63:56]							
8	Wafer Number	UID[71:64]							
9	Rev ID	UID[79:72]							

34 封装尺寸

LQFP100 封装







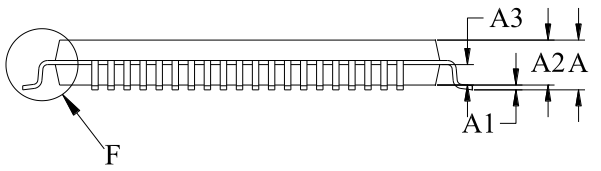
SECTION B-B

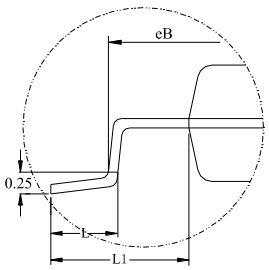
Symbol	Millimeter		
	Min	Nom	Max
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	--	0.26
b1	0.17	0.20	0.23
c	0.13	--	0.17
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
eB	15.05	--	15.35
e	0.50BSC		
L	0.45	--	0.75
L1	1.00REF		
θ	0	--	7°

NOTE:

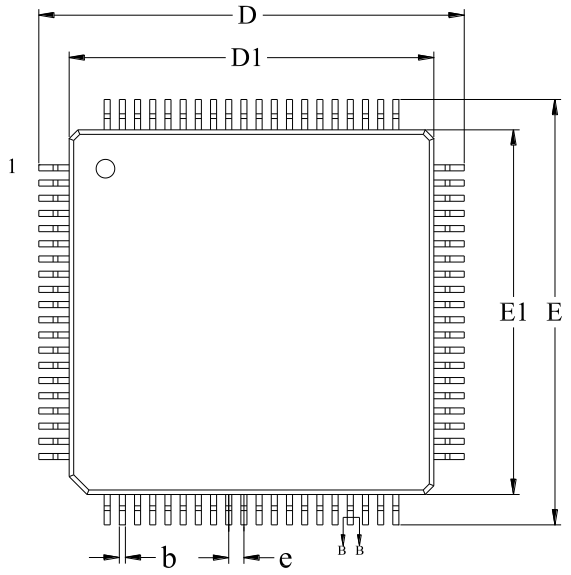
- **Dimensions “D1” and “E1” do not include mold flash.**

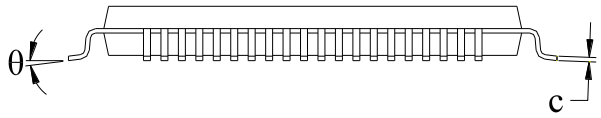
LQFP80 封装

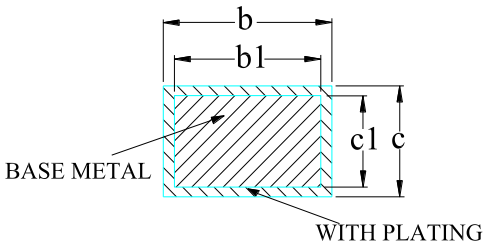




DETAIL: F







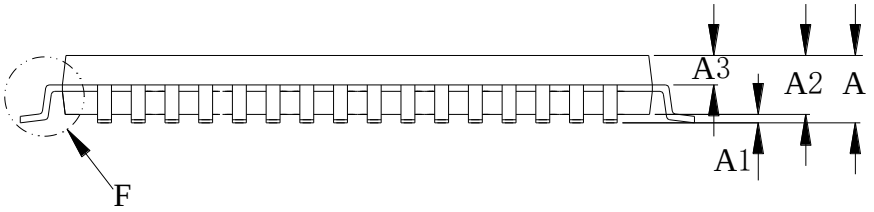
SECTION B-B

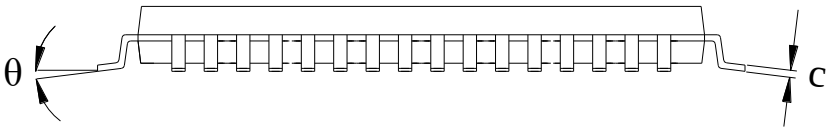
Symbol	Millimeter		
	Min	Nom	Max
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	--	0.26
b1	0.17	0.20	0.23
c	0.13	--	0.17
c1	0.12	0.13	0.14
D	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
eB	13.05	--	13.25
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
θ	0	--	7°

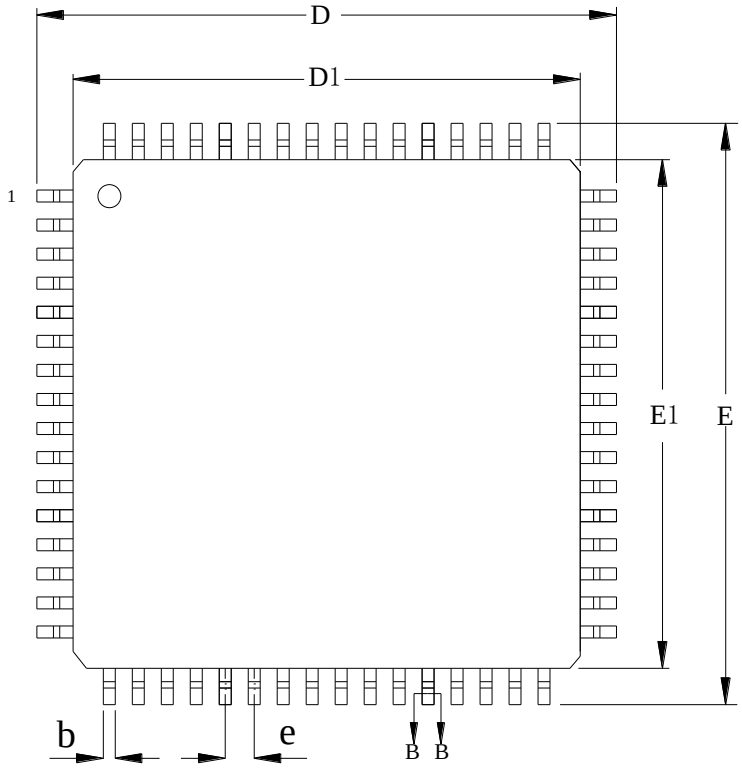
NOTE:

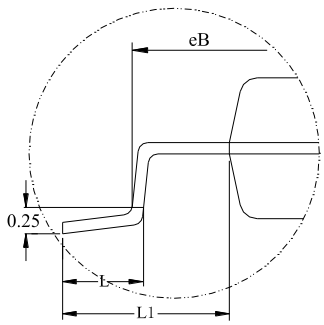
- **Dimensions “D1” and “E1” do not include mold flash.**

LQFP64 封装

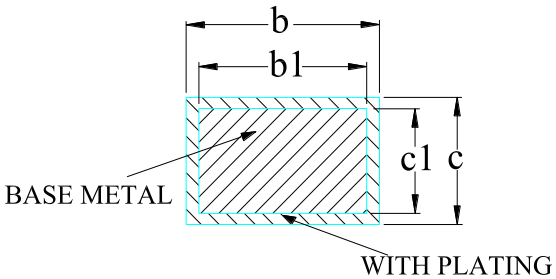








DETAIL: F



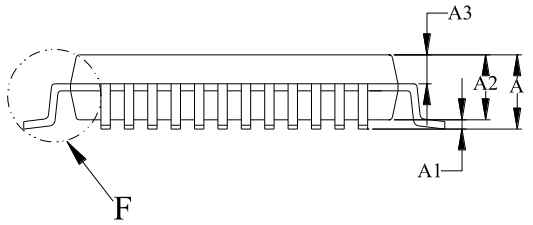
SECTION B-B

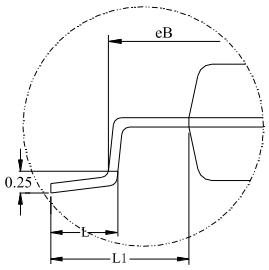
Symbol	LQFP64 (10x10)			LQFP64 (7x7)		
	Min	Nom	Max	Min	Nom	Max
A	--	--	1.60	--	--	1.60
A1	0.05	--	0.15	0.05	--	0.15
A2	1.35	1.40	1.45	1.35	1.40	1.45
A3	0.59	0.64	0.69	0.59	0.64	0.69
b	0.18	--	0.26	0.16	--	0.24
b1	0.17	0.20	0.23	0.15	0.18	0.21
c	0.13	--	0.17	0.13	--	0.17
c1	0.12	0.13	0.14	0.12	0.13	0.14
D	11.80	12.00	12.20	8.80	9.00	9.20
D1	9.90	10.00	10.10	6.90	7.00	7.10
E	11.80	12.00	12.20	8.80	9.00	9.20
E1	9.90	10.00	10.10	6.90	7.00	7.10
eB	11.05	--	11.25	8.10	--	8.25
e	0.50BSC			0.40BSC		
L	0.45	--	0.75	0.45	--	0.75
L1	1.00REF			1.00REF		
θ	0°	--	7°	0°	--	7°

NOTE:

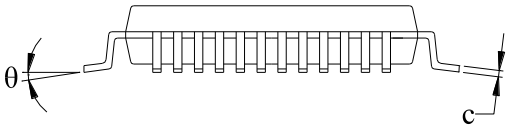
- **Dimensions “D1” and “E1” do not include mold flash.**

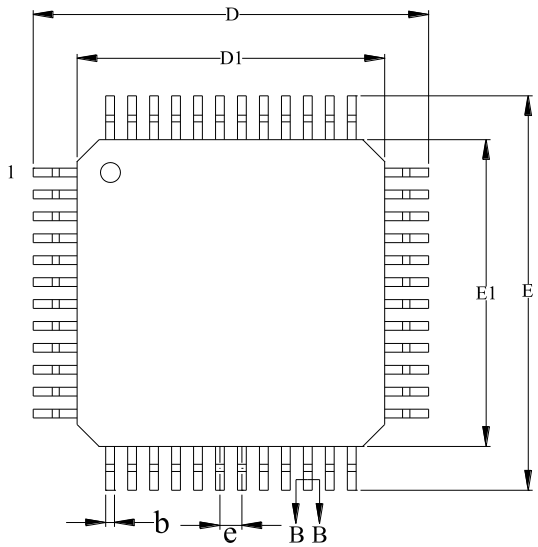
LQFP48 封装

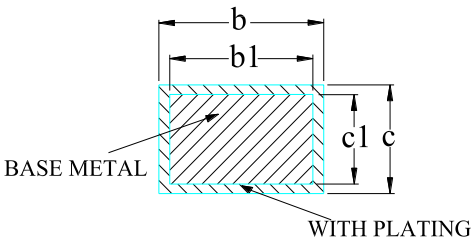




DETAIL: F







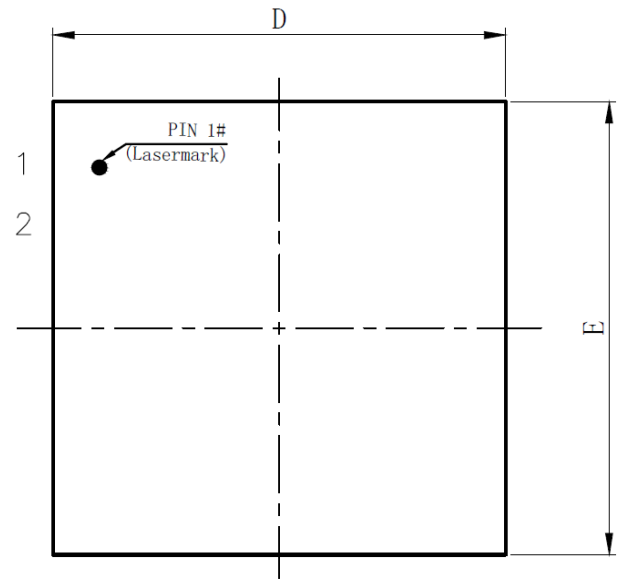
SECTION B-B

Symbol	Millimeter		
	Min	Nom	Max
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	--	0.26
b1	0.17	0.20	0.23
c	0.13	--	0.17
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	--	8.25
e	0.50BSC		
L	0.40	--	0.65
L1	1.00REF		
θ	0	--	7°

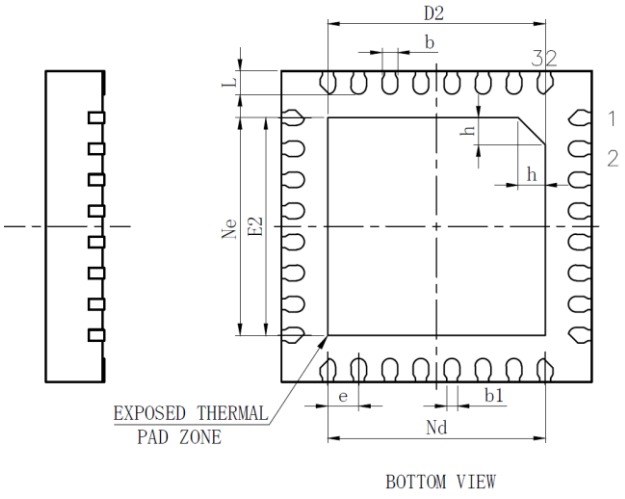
NOTE:

- **Dimensions "D1" and "E1" do not include mold flash.**

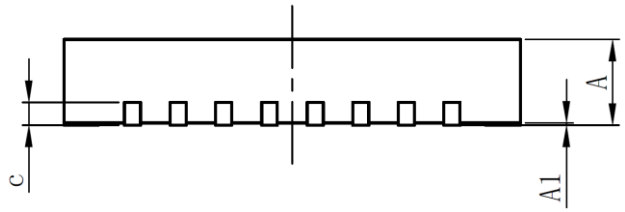
QFN32 封装



TOP VIEW



BOTTOM VIEW



SIDE VIEW

Symbol	Millimeter		
	Min	Nom	Max
A	0.70	0.75	0.80
	0.80	0.85	0.90
	0.85	0.90	0.95
A1	0	0.02	0.05
b	0.15	0.20	0.25
b1	0.14REF		
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.70	2.80	2.90
e	0.40BSC		
Ne	2.80BSC		
Nd	2.80BSC		
E	3.90	4.00	4.10
E2	2.70	2.80	2.90
L	0.25	0.30	0.35
h	0.30	0.35	0.40
L/F 载体尺寸	122 x 122		

35 附录 A SysTick 定时器

35.1 SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行OS维护的工作。Cortex-M0处理器中有一个称为SysTick的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询计数标志。

35.2 设置 SysTick

由于 SysTick 定时器的重新装载值和当前值在复位时都是未定义的，为了防止产生异想不到的结果，SysTick 的设置代码需要遵循一定的流程。

启动→禁止 SysTick→设置重载值寄存器→清除当前值寄存器→使能 SysTick→完成

SysTick 有一个校准值寄存器，它提供的信息有助于所需重载值的计算。如果微处理器中有这个定时的校准，那么这个寄存器的 TENMS 域中就会包含 10 毫秒所对应的计数值。

35.3 SysTick 寄存器

地址	名称	CMSIS 符号	全名
0XE000E010	SYS_CSR	SysTick->CTRL	SysTick控制和状态寄存器
0XE000E014	SYS_RVR	SysTick->LOAD	SysTick重装载寄存器
0XE000E018	SYS_CVR	SysTick->VAL	SysTick当前值寄存器
0XE000E01C	SYS_CALIR	SysTick->CALIB	SysTick校准值寄存器

35.3.1 SysTick 控制和状态寄存器（CTRL）

位	符号	功能描述	类型	复位值
31:17	Reserved	-	-	-
16	COUNTFLAG	当定时器到0时，该位置1，读取寄存器会被清零	RO	0
15:3	Reserved	-	-	-
2	CLKSOURCE	SysTick时钟源选择 1:使用内核时钟 0:使用参考时钟	RW	0
1	TICKINT	SysTick中断使能 1：中断使能 0：中断禁止	RW	0
0	ENABLE	置1时SysTick 定时器使能，要不然计数会被禁止。	RW	0

35.3.2 SysTick 重载寄存器（LOAD）

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

35.3.3 SysTick 当前值寄存器（VAL）

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	CURRENT	读出值为SysTick定时器的当前数值，写入任何值都会清除寄存器，COUNTFLAG也会清零（不会引起SysTick异常）	RW	未定义

35.3.4 SysTick 校准值寄存器（CALIB）

位	符号	功能描述	类型	复位值
31	NOREF	如果读出值为1，表示没有使用外部参考时钟，SysTick使用内核时钟；如果是0，则使用外部参考时钟（外部低速晶振XTL）	RO	-
30	SKEW	如果设为1，则TENMS域不准确	RO	
29:24	Reserved	-	-	
23:0	TENMS	10毫秒校准值	RO	-

36 附录 B 文档约定

36.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

RW	读写，软件可以读写这些位。
RO	只读，软件只能读取这些位。
WO	只写，软件只能写入该位。读取该位时将返回无效数据。
W1	只写 1，硬件自动清 0，写 0 无效
R0W1	软件读取该位为 0，写入 1 将该位清零。写入 0 对该位的值无影响。
RW0	软件可以读写该位，写 1 无效，写 0 清除
R1W0	软件读取该位为 1,，写入 0 将该位清零。写入 1 对该位的值无影响。
RC	软件可以读取该位。读取该位时，将自动清零。 写入“0”对该位的值无影响。
Res, Reserverd	保留位，必须保持复位值。

36.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

Word：32 位数据。

Half Word：16 位数据。

Byte：8 位数据。

IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。

ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。

AHB：高级高性能总线。

APB：低速外设总线。

DMA：直接存储器访问。

TIM：定时器

版本记录 & 联系方式

版 本	修 订 日 期	修 订 内 容 摘 要
Rev1.0	2019/09/11	初稿发布。



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email：mcu@hdsc.com.cn

网址：http://www.hdsc.com.cn/mcu.htm

通信地址：上海市张江高科园区碧波路 572 弄 39 号

邮编：201203

