Arm® Cortex®-M

32位微控制器

NuMicro® 家族
M031/M032 系列
技术参考手册

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro® microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

目录

1 概述 22
2 特性 23
2.1 M031/M032特性 23
3 料号信息 30
3.1 封装类型 30
3.2 M031/M032系列选型指南 31
3.2.1 M031基本系列（M031Fx / M031Ex / M031Tx） 31
3.2.2 M031基本系列（M031Lx） 33
3.2.3 M031基本系列（M031Sx） 34
3.2.4 M031基本系列（M031Kx） 35
3.2.5 M032 USB系列（M032Fx / M032Ex / M032Tx） 36
3.2.6 M032 USB系列（M032Lx） 37
3.2.7 M032 USB系列（M032Sx） 38
3.2.8 M032 USB系列（M032Kx） 39
3.2.9 命名规则 40
3.3 M031/M032系列特性比较表 41
4 引脚配置 43
4.1 引脚配置 43
4.1.1 M031 系列引脚图 43
4.1.2 M031系列多功能引脚框图 49
4.1.3 M032系列引脚框图 111
4.1.4 M032系列多功能引脚框图 117
4.2 引脚映射 167
4.3 引脚功能描述 172
5 框图 178
6 功能描述 179
6.1 Arm® Cortex®-M0 内核 179
6.2 时钟控制器 181
6.2.1 概述 181
6.2.2 时钟发生器 183
6.2.3 系统时钟和SysTick时钟 185
6.2.4 外设时钟 187
6.2.5 掉电模式时钟 187
6.2.6 时钟输出 188
6.2.7 USB时钟源 188

6.2.8 寄存器映射 190
6.2.9 寄存器描述 191
6.3 系统管理 222
6.3.1 概述 222
6.3.2 系统复位 222
6.3.3 系统电源分配 228
6.3.4 电源模式和唤醒源 229
6.3.5 系统内存映射 232
6.3.6 SRAM内存结构 234
6.3.7 带奇偶校验功能的SRAM内存结构 234
6.3.8 芯片总线矩阵 236
6.3.9 IRC自动校准 236
6.3.10 寄存器锁控制器 237
6.3.11 UART0_TXD/USCI0_DAT1调制PWM 241
6.3.12 寄存器映射 243
6.3.13 寄存器描述 245
6.3.14 系统定时器（SysTick） 291
6.3.15 可嵌套向量中断控制器（NVIC） 296
6.3.16 系统控制寄存器 312
6.4 Flash 存储控制器(FMC) 322
6.4.1 概述 322
6.4.2 特性 322
寄存器描述 323
6.4.3 框图 323
6.4.4 功能描述 326
6.4.5 寄存器映射 354
6.4.6 寄存器描述 355
6.5 通用I/O (GPIO) 373
6.5.1 概述 373
6.5.2 特性 373
6.5.3 框图 374
6.5.4 基本配置 374
6.5.5 功能描述 375
6.5.6 寄存器映射 379
6.5.7 寄存器描述 382
6.6 PDMA控制器(PDMA) 395
6.6.1 概述 395
6.6.2 特性 395
6.6.3 框图 395
6.6.4 基本配置 395

6.6.5 功能描述.....396
6.6.6 寄存器映射.....402
6.6.7 寄存器描述.....403
6.7 定时器控制器(TMR) 432
6.7.1 概述.....432
6.7.2 特性.....432
6.7.3 框图.....433
6.7.4 基本配置.....434
6.7.5 功能描述.....435
6.7.6 寄存器映射.....441
6.7.7 寄存器描述.....443
6.8 看门狗定时器(WDT) 453
6.8.1 概述.....453
6.8.2 特性.....453
6.8.3 框图.....453
6.8.4 基本配置.....453
6.8.5 功能描述.....454
6.8.6 寄存器映射.....457
6.8.7 寄存器描述.....458
6.9 窗口看门狗定时器 (WWDT) 462
6.9.1 概述.....462
6.9.2 特性.....462
6.9.3 框图.....462
6.9.4 基本配置.....462
6.9.5 功能描述.....463
6.9.6 寄存器映射.....467
6.9.7 寄存器描述.....468
6.10 实时时钟(RTC) 473
6.10.1概述.....473
6.10.2特性.....473
6.10.3框图.....474
6.10.4基本配置.....474
6.10.5功能描述.....474
6.10.6寄存器映射.....479
6.10.7寄存器描述.....480
6.11 基本PWM 发生器和捕获定时器(BPWM) 496
6.11.1概述.....496
6.11.2特性.....496
6.11.3框图.....497

6.11.4 基本配置 500
6.11.5 功能描述 502
6.11.6 寄存器映射 517
6.11.7 寄存器描述 520
6.12 PWM发生器和捕获定时器 (PWM) 554
6.12.1 概述 554
6.12.2 特性 554
6.12.3 框图 555
6.12.4 基本配置 559
6.12.5 功能描述 560
6.12.6 寄存器映射 580
6.12.7 寄存器描述 585
6.13 UART接口控制器 (UART) 637
6.13.1 概述 637
6.13.2 特征 637
6.13.3 框图 638
6.13.4 基本配置 642
6.13.5 功能描述 645
6.13.6 寄存器映射 661
6.13.7 寄存器描述 663
6.14 SPI接口(SPI) 694
6.14.1 概述 694
6.14.2 特征 694
6.14.3 框图 695
6.14.4 基本配置 695
6.14.5 功能描述 696
6.14.6 时序图 712
6.14.7 编程例程 714
6.14.8 寄存器映射 717
6.14.9 寄存器描述 718
6.15 QSPI接口 (QSPI) 739
6.15.1 概述 739
6.15.2 特征 739
6.15.3 框图 740
6.15.4 基本配置 741
6.15.5 功能描述 741
6.15.6 时序图 757
6.15.7 编程示例 759
6.15.8 寄存器映射 761
6.15.9 寄存器描述 762

6.16 I²C 串行接口控制器(I²C) 776
6.16.1概述 776
6.16.2特性 776
6.16.3框图 777
6.16.4基本配置 777
6.16.5功能描述 778
6.16.6寄存器映射 798
6.16.7寄存器描述 800
6.17 USCI –通用串行控制接口控制器(USCI) 823
6.17.1概述 823
6.17.2特性 823
6.17.3框图 823
6.17.4功能描述 823
6.18 USCI – UART模式 834
6.18.1概述 834
6.18.2特性 834
6.18.3框图 834
6.18.4基本配置 835
6.18.5功能描述 835
6.18.6寄存器映射 844
6.18.7寄存器描述 845
6.19 USCI - SPI模式 867
6.19.1概述 867
6.19.2特性 867
6.19.3框图 868
6.19.4基本配置 868
6.19.5功能描述 869
6.19.6寄存器映射 881
6.19.7寄存器描述 882
6.20 USCI - I²C模式 905
6.20.1概述 905
6.20.2特性 905
6.20.3框图 906
6.20.4基本配置 906
6.20.5功能描述 907
6.20.6寄存器映射 925
6.20.7寄存器描述 926
6.21 外部总线接口(EBI) 945
6.21.1概述 945

6.21.2特性.....945
6.21.3框图.....946
6.21.4基本配置.....946
6.21.5功能描述.....948
6.21.6寄存器表.....958
6.21.7寄存器描述.....959
6.22 USB 2.0 全速设备控制器(USBD) 963
6.22.1概述.....963
6.22.2特性.....963
6.22.3框图.....964
6.22.4基本配置.....964
6.22.5功能描述.....964
6.22.6寄存器表.....970
6.22.7寄存器描述.....972
6.23 CRC控制器 (CRC)..... 992
6.23.1概述.....992
6.23.2特性.....992
6.23.3框图.....992
6.23.4基本配置.....993
6.23.5功能描述.....993
6.23.6寄存器映射.....995
6.23.7寄存器描述.....996
6.24 硬件除法器(HDIV)..... 1001
6.24.1概述.....1001
6.24.2特性.....1001
6.24.3基本配置.....1001
6.24.4功能描述.....1001
6.24.5寄存器表.....1002
6.24.6寄存器描述.....1003
6.25 模数转换器(ADC)..... 1008
6.25.1概述.....1008
6.25.2特性.....1008
6.25.3框图.....1009
6.25.4基本配置.....1010
6.25.5功能描述.....1010
6.25.6寄存器映射.....1020
6.25.7寄存器描述.....1022
6.26 模拟比较器控制器(ACMP)..... 1039
6.26.1概述.....1039

6.26.2特性.....1039
6.26.3框图.....1040
6.26.4基本配置.....1040
6.26.5功能描述.....1041
6.26.6寄存器表.....1046
6.26.7寄存器描述.....1047
6.27 外设互联.....1057
6.27.1概述.....1057
6.27.2外设互连矩阵表.....1057
6.27.3功能描述.....1057
7 应用电路.....1060
7.1 供电电源电路.....1060
7.2 外设应用电路.....1061
8 电气特性.....1062
9 缩写.....1063
9.1 缩写.....1063
10 修订历史.....1065

图集

图4.1-1 M031系列TSSOP 20引脚框图 43
图4.1-2 M031系列TSSOP 28引脚框图 44
图4.1-3 M031系列QFN 33引脚框图 45
图4.1-4 M031系列LQFP 48引脚框图 46
图4.1-5 M031系列LQFP 64引脚框图 47
图4.1-6 M031系列LQFP 128引脚框图 48
图4.1-7 M031FB0AE多功能引脚框图 49
图4.1-8 M031FC1AE多功能引脚框图 50
图4.1-9 M031EB0AE多功能引脚框图 51
图4.1-10 M031EC1AE多功能引脚框图 52
图4.1-11 M031TB0AE多功能引脚框图 54
图4.1-12 M031TC1AE多功能引脚框图 56
图4.1-13 M031TD2AE多功能引脚框图 58
图4.1-13 M031TE3AE多功能引脚框图 60
图4.1-14 M031LC2AE多功能引脚框图 62
图4.1-15 M031LD2AE多功能引脚框图 65
图4.1-16 M031LE3AE多功能引脚框图 68
图4.1-17 M031LG6AE多功能引脚框图 71
图4.1-18 M031LG8AE多功能引脚框图 74
图4.1-19 M031SC2AE多功能引脚框图 77
图4.1-20 M031SD2AE多功能引脚框图 80
图4.1-21 M031SE3AE多功能引脚框图 83
图4.1-22 M031SG6AE多功能引脚框图 86
图4.1-23 M031SG8AE多功能引脚框图 89
图4.1-24 M031SIAAE多功能引脚框图 92
图4.1-25 M031KG6AE多功能引脚框图 96
图4.1-26 M031KG8AE多功能引脚框图 101
图4.1-27 M031KIAAE多功能引脚框图 106
图4.1-28 M032系列TSSOP 20引脚框图 111
图4.1-29 M032系列TSSOP 28引脚框图 112
图4.1-30 M032系列QFN 33引脚框图 113
图4.1-31 M032系列LQFP 48引脚框图 114
图4.1-32 M032系列LQFP 64引脚框图 115

图4.1-33 M032系列LQFP 128引脚框图 116
图4.1-34 M032FC1AE多功能引脚框图..... 117
图4.1-35 M032EC1AE多功能引脚框图 118
图4.1-36 M032TC1AE多功能引脚框图..... 120
图4.1-37 M032TD2AE多功能引脚框图..... 122
图4.1-38 M032LC2AE多功能引脚框图..... 124
图4.1-39 M032LD2AE多功能引脚框图..... 127
图4.1-40 M032LE3AE多功能引脚框图..... 130
图4.1-41 M032LG6AE多功能引脚框图..... 133
图4.1-42 M032LG8AE多功能引脚框图..... 136
图4.1-43 M032SE3AE多功能引脚框图..... 139
图4.1-44 M032SG6AE多功能引脚框图 142
图4.1-45 M032SG8AE多功能引脚框图 145
图4.1-46 M032SIAAE多功能引脚框图..... 148
图4.1-47 M032KG6AE多功能引脚框图 152
图4.1-48 M032KG8AE多功能引脚框图 157
图4.1-49 M032KIAAE多功能引脚框图..... 162
图 5-1 NuMicro® M031/M032框图 178
图 6-1 功能框图..... 179
图 6.2-1时钟发生器全局示意图 （1/2） 182
图 6.2-2 时钟发生器全局示意图（2/2） 183
图 6.2-3 时钟发生器框图 184
图 6.2-4系统时钟框图..... 185
图 6.2-5 HXT停止保护过程..... 186
图 6.2-6 LXT 停止保护过程 187
图 6.2-7 系统时钟控制框图..... 187
图 6.2-8 时钟输出框图..... 188
图 6.2-9 USBD时钟源..... 189
图 6.3-1系统复位源..... 223
图 6.3-2 nRESET复位时序 225
图 6.3-3 上电复位（POR）波形图 225
图 6.3-4 低压复位（LVR）波形图 226
图 6.3-5 掉电检测复位（BOD）波形..... 227
图 6.3-6 NuMicro® M031电源分布框图..... 228

图 6.3-7 电源模式状态机 230
图 6.3-8 SRAM 内存组织 234
图 6.3-9 NuMicro® M031总线矩阵图 236
图 6.4-1 16/32/64/128/256KB型号Flash存储器控制框图 324
图 6.4-2 512KB 型号Flash存储器控制框图 326
图 6.4-3 数据Flash共享APROM空间 327
图 6.4-4 双BANK区APROM示例(512K字节) 327
图 6.4-5 地址操作模式 328
图 6.4-6 16/32/64/128K字节Flash芯片 SPROM安全模式 334
图 6.4-7 256/512K字节Flash芯片 SPROM安全模式 334
图 6.4-8 16/32/64/128K字节芯片存储器映射 335
图 6.4-9 256/512K字节芯片存储器映射 336
图 6.4-10 16/32/64/128K字节Flash芯片系统存储器映射带IAP模式 337
图 6.4-11 256/512K字节Flash芯片系统存储器映射带IAP模式 338
图 6.4-12 LDROM带IAP 模式 339
图 6.4-13 APROM带IAP 模式 339
图 6.4-14 16/32/64/128K字节Flash的芯片不带IAP模式系统内存映射 340
图 6.4-15 256/512K字节Flash的芯片不带IAP模式系统内存映射 341
图 6.4-16启动源选择 342
图 6.4-17 ISP示范流程 345
图 6.4-18 VECMAP加速中断处理 346
图 6.4-19 ISP 32位编程流程 347
图 6.4-20 ISP 64位编程流程 348
图 6.4-21 多字编程固件运行在SRAM 349
图 6.4-22 多字编程流程 350
图 6.4-23 CRC-32 Checksum 计算 351
图 6.4-24 CRC-32 Checksum计算流程 352
图 6.4-25 全是1校验流程 353
图 6.5-1 GPIO控制器框图 374
图 6.5-2推挽输出 375
图 6.5-3 开漏输出 375
图 6.5-4准双向I/O模式 376
图 6.5-5 GPIO上升沿触发中断 377
图 6.5-6 GPIO下降沿触发中断 377

图 6.6-1 PDMA 控制器框图 395
图 6.6-2 描述表数据结构 396
图 6.6-3 基本模式有限状态机 397
表 6.6-4描述表链表结构 398
表 6.6-5 Scatter-Gather 模式有限状态机 398
图 6.6-6基本模式下单次传输跟批量传输示意图 400
图 6.6-7 PDMA通道0时间溢出计数操作示意图 401
图 6.7-1定时器框图 433
图 6.7-2定时器时钟源 434
图 6.7-3连续计数模式 436
图 6.7-4外部捕获模式 437
图 6.7-5复位计数模式 438
图 6.7-6 内部定时器触发 439
图 6.7-7 定时器间触发捕获时序 440
图 6.8-1 看门狗定时器框图 453
图 6.8-2 看门狗定时器时钟控制 454
图 6.8-3 看门狗定时器定时溢出间隔和复位周期时序图 455
图 6.9-1 WWDT 框图 462
图 6.9-2 WWDT 时钟控制 463
图 6.9-3 WWDT 复位和重载动作 464
图 6.9-4 当 CNTDAT > CMPDAT WWDT 重载计数器 464
图 6.9-5 当WWDT_CNT < WINCMP WWDT重载计数器 465
图 6.9-6 WWDT中断和复位信号 465
图 6.10-1 RTC框图 474
图 6.13-1 UART 时钟控制框图 640
图 6.13-2 UART 模块框图 641
图 6.13-3 自动波特率测量 648
图 6.13-4发送延时操作 648
图 6.13-5 UART nCTS唤醒示例 1 649
图 6.13-6 UART nCTS 唤醒示例2 649
图 6.13-7 UART 数据唤醒 650
图 6.13-8 UART接收数据 FIFO 达到阈值唤醒 650
图 6.13-9 UART RS-485 AAD模式地址匹配唤醒 651
图 6.13-10 UART数据接收 FIFO 阈值超时唤醒 651

图 6.13-11自动流控框图 654
图 6.13-12 UART nCTS自动流控使能 655
图 6.13-13 UART nRTS自动流控功能使能 655
图 6.13-14 UART nRTS软件控制的自动流控 656
图 6.13-15 IrDA 控制模块框图 656
图 6.13-16 IrDA TX/RX 时序图 657
图 6.13-17 RS-485自动方向控制模式下 nRTS 引脚驱动电平 658
图 6.13-18 RS-485 nRTS 软件控制时的驱动电平 659
图 6.13-19 RS-485帧结构 660
图 6.14-1 SPI 框图 695
图 6.14-2 SPI 外设时钟 696
图 6.14-3 SPI 全双工主机模式应用框图 697
图 6.14-4 SPI全双工从机模式应用框图 697
图 6.14-5在一个数据字32-位长度 698
图 6.14-6自动从机选择(SSACTPOL = 0, SUSPITV > 0x2) 699
图 6.14-7自动从机选择(SSACTPOL = 0, SUSPITV < 0x3) 699
图 6.14-8字节重排序功能 700
图 6.14-9字节休眠时序图 700
图 6.14-10 SPI半双工主机模式应用框图 701
图 6.14-11 SPI半双工从机模式应用框图 701
图 6.14-12 FIFO阀值比较器 702
图 6.14-13 8~16位数据长度的发送 FIFO 缓存示例 703
图 6.14-14 17~32 位数据长度的发送 FIFO 缓存示例 704
图 6.14-15 接收 FIFO 缓存示例对于16位数据长度 705
图 6.14-16 接收 FIFO 缓存示例对于32位数据长度 705
图 6.14-17 TX下溢事件和从机溢出运行事件 706
图 6.14-18从机模式位个数错误 706
图 6.14-19 I^2S数据格式时序图 708
图 6.14-20 MSB对齐数据格式时序图 708
图 6.14-21 PCM模式 A 时序图 708
图 6.14-22 PCM模式 B 时序图 709
图 6.14-23各种I^2S 模式的FIFO 内容 710
图 6.14-24主机模式下的SPI时序 713
图 6.14-25主机模式下的SPI时序(交替SPIx_CLK时钟相位) 713

图 6.14-26从机模式下的SPI时序 714
图 6.14-27从机模式下的SPI时序(交替SPIx_CLK时钟相位) 714
图 6.15-1 QSPI 框图 740
图 6.15-2 QSPI 外设时钟 742
图 6.15-3 QSPI QSPI全双工主机模式应用框图 742
图 6.15-4 QSPI QSPI全双工从机模式应用框图 743
图 6.15-5 一次32-位的传输事务 743
图 6.15-6自动从机选择(SSACTPOL = 0, SUSPITV > 0x2) 744
图 6.15-7自动从机选择(SSACTPOL = 0, SUSPITV < 0x3) 745
图 6.15-8字节重排功能 745
图 6.15-9字节挂起波形图 746
图 6.15-10 QSPI 半双工主机模式应用框图 746
涂 6.15-11 QSPI 半双工从机模式应用框图 746
图 6.15-12 2位传输模式系统架构 748
图 6.15-13 2位传输模式的时序图(主机模式) 748
图 6.15-14双输出模式的位时序 749
图 6.15-15双输入模式的位时序 749
图 6.15-16四I/O输出模式位顺序 750
图 6.15-17四I/O输入模式位顺序 750
图 6.15-18 FIFO 阈值比较器 751
图 6.15-19发送FIFO缓存示例 752
图 6.15-20接收FIFO缓存示例 753
图 6.15-21 TX 下溢事件和从机溢出事件 753
图 6.15-22 2位传输模式FIFO缓存示例 754
图 6.15-23 TX下溢事件(QSPI0从机三线模式使能) 754
图 6.15-24从机模式位计数错误 755
图 6.15-25从机超时事件 755
图 6.15-26 QSPI 主机模式时序 757
图 6.15-27 QSPI主机模式时序(QSPIx_CLK交替相位) 758
图 6.15-28 QSPI从机模式时序 758
图 6.15-29 QSPI从机模式时序(QSPIx_CLK交替相位) 759
图6.16-1 I^2C控制器框图 777
图6.16-2 I^2C总线时序 778
图6.16-3 I^2C协议 778

图6.16-4起始（START）和停止（STOP）信号条件 779
图6.16-5在I²C 总线上位传输 780
图6.16-6在 I²C总线上应答 780
图6.16-7主机向从机发送7位数据 780
图6.16-8主机从从机读取7位数据 781
图6.16-9根据当前I²C状态控制I²C总线 781
图6.16-10主机传输模式控制流程 782
图6.16-11主机接收模式控制流程 783
图6.16-12从机模式控制流程 784
图6.16-13 GC模式 785
图6.16-14仲裁丢失 786
图6.16-15总线管理主要元素包协议图 788
图6.16-167位可寻址的设备与主机通信 789
图6.16-177位可寻址的设备对警报响应地址的响应 789
图6.16-18总线管理报警功能 790
图6.16-19总线管理超时时序 790
图6.16-20总线时钟低电平超时时序 791
图6.16-21设置时间错误调整 792
图6.16-22保持时间错误调整 792
图6.16-23 I²C数据移位方向 793
图6.16-24 I²C 超时计数模块框图 795
图6.16-25 I²C唤醒相关信号波形 796
图6.16-26 EEPROM随机读 797
图6.16-27随机读的协议 798
图6.17-1USCI框图 823
图6.17-2输入调节 USCIx_DAT[1:0] 和 USCIx_CTL[1:0] 824
图6.17-3输入调节USCIx_CLK 825
图6.17-4数据缓冲框图 826
图6.17-5数据存取结构 827
图6.17-6传送数据路径 827
图6.17-7接收数据路径 828
图6.17-8相关协议时钟发生器 829
图6.17-9基本时钟分频计数器 830
图6.17-10时序测量计数器框图 830

图6.17-11采样时间计数器 831
图6.17-12事件和中断结构 832
图6.18-1USCI-UART模式框图 834
图6.18-2 UART全双工通信的UART信号连接 836
图6.18-3 UART标准帧格式 836
图6.18-4 UART位时序 (数据采样时间) 838
图6.18-5 UART自动波特率控制 840
图6.18-6输入数据唤醒 841
图6.18-7nCTS唤醒情况1 841
图6.18-8nCTS唤醒情况2 841
图6.19-1SPI 主机模式应用框图 867
图6.19-2SPI 从机模式应用框图 867
图6.19-3USCI SPI 模式框图 868
图6.19-44-线全双工SPI 通信信号 (主机模式) 869
图6.19-54线全双工SPI 通信信号 (从机模式) 870
图6.19-6不同时钟模式下的SPI通信(SCLKMODE=0x0) 871
图6.19-7不同时钟模式下的SPI通信(SCLKMODE=0x1) 871
图6.19-8不同时钟模式下的SPI通信(SCLKMODE=0x2) 872
图6.19-9不同时钟模式下的SPI通信(SCLKMODE=0x3) 872
图6.19-10 MSB优先格式下16位数据长度的一次传输 873
图6.19-11两组传输字间的暂停 873
图6.19-12自动从机选择(SUSPITV ≧ 0x3) 874
图6.19-13自动从机选择(SUSPITV < 0x3) 874
图6.19-14一个输出数据通道半双工通信 (SPI 主机模式) 875
图6.19-15一个输入数据通道半双工通信(SPI 主机模式) 876
图6.19-16主机模式下的SPI 时序 877
图6.19-17主机模式下的SPI 时序(串行总线时钟反相) 878
图6.19-18从机模式下的SPI 时序 878
图6.19-19从机模式下的SPI 时序(串行总线时钟反相) 879
图6.20-1 I^2C总线时序 905
图6.20-2USCI I^2C模式框图 906
图6.20-3 I^2C 协议 907
图6.20-4 START 和 STOP 状态 907
图6.20-5I^2C 总线上的位传输 908

图6.20-6总线上的应答信号 909
图6.20-7仲裁丢失 910
图6.20-8依据I²C 状态控制I²C 总线 912
图6.20-9 7位地址情况下主机向从机传输数据 912
图6.20-10 7位地址情况下主机向从机读取数据 913
图6.20-1110位地址情况下主机向从机传输数据 913
图6.20-1210位地址情况下主机向从机读取数据 913
图6.20-13 7位地址主机发送模式控制流程 914
图6.20-147位地址主机接收模式控制流程 915
图6.20-15 10位地址主机发送模式控制流程 916
图6.20-16 10位地址主机接收模式控制流程 917
图6.20-17 7位地址从机模式控制流程 918
图6.20-18 10位地址从机模式控制流程 919
图6.20-19 7位地址的GC模式 920
图6.20-20建立时间错误调整 922
图6.20-21保持时间错误调整 922
图6.20-22 I²C超时计数模块框图 923
图6.20-23 EEPROM随机读取 924
图6.20-24随机读取协议 924
图 6.21-1 EBI框图 946
图 6.21-2 16位 EBI 数据宽度与16位设备的连接 949
图 6.21-3 8位 EBI 数据宽度与8位设备的连接 949
图 6.21-4 16位EBI数据宽度与16位设备的分开模式连接 950
图 6.21-5 8位EBI数据宽度与8位设备的分开模式连接 950
图 6.21-6 16位数据宽度的时序控制波形 952
图 6.21-7 8位数据宽度的时序控制波形 953
图 6.21-8 16位数据模式字节写的时序控制波形 954
图 6.21-9插入空闲周期的时序控制波形 955
图 6.21-10 16位数据宽度分开模式下的时序控制波形 956
图 6.21-11 连续数据访问模式的时序控制波形 957
图 6.22-1 USB框图 964
图 6.22-2 NEVWK 中断操作流程 966
图 6.22-3 端点SRAM架构 967
图 6.22-4 Setup 事务后接着是Data IN 事务 967

图 6.22-5 数据输出传输 968
图 6.22-6 LPM状态转换图 969
图 6.23-1 CRC产生器框图 993
图 6.23-2校验和位反转功能框图 994
图 6.23-3写数据位顺序反转功能框图 994
图 6.24-1 硬件除法器操作流程 1001
图 6.25-1 AD控制器框图 1009
图 6.25-2 ADC外设时钟控制 1011
图 6.25-3 单次模式转换时序图 1012
图 6.25-4 Burst模式转换时序图 1013
图 6.25-5使能通道单周期扫描模式时序图 1014
图 6.25-6使能通道连续扫描模式时序图 1015
图 6.25-7 16次平均校准模式 1016
图 6.25-8 A/D转换结果监控逻辑框图 1017
图 6.25-9 A/D控制器中断 1018
图 6.25-10 ADC单端输入转换结果映射框图 1023
图 6.25-11 ADC差分输入转换结果映射框图 1024
图 6.26-1 模拟比较器框图 1040
图 6.26-2 ACMP0比较器迟滞功能 1041
图 6.26-3 窗口锁定模式 1042
图 6.26-4 滤波器功能的例子 1042
图 6.26-5 比较器控制器中断 1043
图 6.26-6 比较器参考电压框图 1043
图 6.26-7 窗口比较模式例子 1044
图 6.26-8 窗口比较模式例子 1045

表集

表1-1 NuMicro® M031/M032 系列关键特性支持表 22
表4.1-1 M031EB0AE多功能引脚表 50
表4.1-2 M031FC1AE多功能引脚表 50
表4.1-3 M031EB0AE多功能引脚表 52
表4.1-4 M031EC1AE多功能引脚表 53
表4.1-5 M031TB0AE多功能引脚表 55
表4.1-6 M031TC1AE多功能引脚表 57
表4.1-7 M031TD2AE多功能引脚表 59
表4.1-7 M031TE3AE多功能引脚表 61
表4.1-8 M031LC2AE多功能引脚表 64
表4.1-9 M031LD2AE多功能引脚表 67
表4.1-10 M031LE3AE多功能引脚表 70
表4.1-11 M031LG6AE多功能引脚表 73
表4.1-12 M031LG8AE多功能引脚表 76
表4.1-13 M031SC2AE多功能引脚表 79
表4.1-14 M031SD2AE多功能引脚表 82
表4.1-15 M031SE3AE多功能引脚表 85
表4.1-16 M031SG6AE多功能引脚表 88
表4.1-17 M031SG8AE多功能引脚表 92
表4.1-18 M031SIAAE多功能引脚表 95
表4.1-19 M031KG6AE多功能引脚表 100
表4.1-20 M031KG8AE多功能引脚表 105
表4.1-21 M031KIAAE多功能引脚表 110
表4.1-22 M032FC1AE多功能引脚表 117
表4.1-23 M032EC1AE多功能引脚表 119
表4.1-24 M032TC1AE多功能引脚表 121
表4.1-25 M032TD2AE多功能引脚表 123
表4.1-26 M032LC2AE多功能引脚表 126
表4.1-27 M032LD2AE多功能引脚表 129
表4.1-28 M032LE3AE多功能引脚表 132
表4.1-29 M032LG6AE多功能引脚表 135
表4.1-30 M032LG8AE多功能引脚表 138
表4.1-31 M032SE2AE多功能引脚表 141

表4.1-32 M032SG6AE多功能引脚表 144
表4.1-33 M032SG8AE多功能引脚表 147
表4.1-34 M032SIAAE多功能引脚表 150
表4.1-35 M032KG6AE多功能引脚表 156
表4.1-36 M032KG8AE多功能引脚表 161
表4.1-37 M032KIAAE多功能引脚表 166
表4.2-1引脚映射表 171
表4.3-1引脚功能描述表 177
表 6.2-1 PLL输出频率公式符号定义 213
表 6.3-1寄存器复位值 225
表 6.3-2 电源模式表 229
表 6.3-3 电源模式差异表 229
表 6.3-4 电源模式差异表 229
表 6.3-5 不同电源模式下的时钟 231
表 6.3-6 再次进入掉电状态的条件 232
表 6.3-7 片上控制器地址空间分配 233
表 6.3-8异常模式 296
表 6.3-9 中断号码表 298
表 6.3-10 优先级分组 317
表 6.4-1 不同芯片FMC特性比较表 323
表 6.4-2 双Bank区地址范围 326
表 6.4-3向量映射支持 342
表 6.4-4 ISP 命令列表 344
表 6.4-5 Flash编程的FMC控制寄存器 347
表 6.5-1去抖动功能设定表 378
表 6.6-1 通道优先级表 397
表 6.7-1定时器0~定时器3 MFP 表 435
表 6.8-1 看门狗定时器定时溢出间隔周期选择 455
表 6.9-1 WWDT 预分频值选择 463
表 6.9-2 CMPDAT设置限制 466
表 6.13-1 NuMicro® M031/M032系列UART特征 638
表 6.13-2 UART 中断 642
表 6.13-3 UART 接口控制器引脚 644
表 6.13-4 UART 控制器波特率计算公式 645

表 6.13-5 UART控制器波特率参数设置示范表 645
表 6.13-6 UART控制器波特率寄存器设置示范表 646
表 6.13-7波特率补偿例表1 647
表6.13-8波特率补偿示表2 647
表 6.13-9 UART 控制器中断源和标志位列表 653
表 6.13-10 UART线控的数据位和停止位长度设置 653
表 6.13-11 UART 线控校验位设置 654
表 6.14-1 SPI/I^{2}S 接口引脚描述(SPI0) 696
表 6.14-2主机模式和单声道模式虚拟数据数 711
表 6.14-3主机模式和立体声模式虚拟数据数 711
表 6.14-4 I^{2}S从机模式和单声道模式虚拟数据数 711
表 6.14-5 PCM从机模式和单声道模式虚拟数据数 712
表 6.14-6 I^{2}S从机模式和立体声模式虚拟数据数 712
表 6.14-7 PCM从机模式和立体声模式虚拟数据数 712
表6.16-1不同芯片I^{2}C特性比较表 777
表6.17-1不同协议的输入信号 824
表6.17-2不同协议输出信号 825
表6.17-3数据传输事件和中断处理 832
表6.17-4具体协议事件和中断处理 832
表6.18-1协议输入信号 836
表6.18-2UART协议的输出信号 836
表6.18-3波特率关系 839
表6.19-1串行总线时钟配置 870
表6.20-1 I^{2}C 传输速率与PCLK之间的关系 921
表 6.21-1 EBI特性比较表 945
表 6.21-2 EBI地址映射 948
表 6.21-3 时序控制参数 951
表 6.22-1 USB链路电源管理 (Lx)状态 968
表 6.25-1 ADC特性比较表 1009
表 6.25-2 ADC 差分模式通道选择 1019
表 6.26-1 不同芯片校准功能特性比较表 1039
表 6.26-2 窗口比较逻辑真值表 1044
表 6.27-1 外设互联矩阵表 1057
表 9.1-1 缩写列表 1064

1 概述

NuMicro® M031/M032系列32位微处理器基于Arm® Cortex®-M0内核并带32位硬件乘法/除法器。具有1.8 ~ 3.6 V工作电压，5V I/O耐压，在-40°C ~105°C范围内可达到48/72MHz。

M031/M032系列为需要1.8V低压接口连接的应用提供了一个解决方案，增强型快速2 MSPS转换率12位ADC、比较器和高达24通道96/144 MHz的PWM控制。它支持对电压、电流和传感器数据进行快速而精确的数据转换，然后对外部设备进行快速响应控制。此外，M031/M032系列还提供了大量的外围设备，包括通用串行控制接口（USCI），可以灵活地设置为UART/SPI/I^2C，多达10组UART、4组SPI、4组I^2C和单线UART接口，用于主设备和从设备之间的数据通信。

M031/M032系列提供从16K字节到512K字节的闪存，SRAM从2K字节到96K字节。支持的封装形式从TSSOP 20脚、TSSOP 28脚、QFN 33脚、LQFP 48脚到LQFP 64脚和LQFP 128脚，管脚兼容不同的料号，使系统设计和零件更换变得容易。

M032料号全部基于M031并为USB相关应用增加无晶振USB 2.0全速设备特性。

对于开发者，Nuvoton提供NuMaker-PFM评估版及Nuvoton Nu-Link除错工具，第三方开发软件支持例如Keil® MDK, IAR EWARM以及带GNU GCC编译器的Eclippse IDE。

产品线	UART	I^2C	SPI/ I^2S	QSPI	USCI	Timer	PWM	RTC	PDMA	EBI	ADC	ACMP	Divider	USBD	IEC60730
M031/M032	8	2	1	1	2	4	24	1	9	1	16	2	1	√	√

表 1-1 NuMicro® M031/M032 系列关键特性支持表

NuMicro® M031/M032系列适合于很广泛的应用场合，例如：

- 激光测距仪
- 空气检测仪/净化器
- 移动液晶面板控制器
- 物联网传感器装置
- 人机界面控制器
- 微型打印机
- 游戏键盘和鼠标
- WPC无线充电

2 特性

2.1 M031/M032特性

核心及系统	
Arm® Cortex®-M0	• Arm® Cortex®-M0 处理器，可运行于 72 MHz – 72 MHz在2.0V-3.6V – 48 MHz在1.8V-3.6V • 内置嵌套式向量中断控制器（NVIC） • 24 位系统定时器 • 支持可屏蔽中断 • 通过 WFI 和 WFE 指令，支持低功耗休眠功能
欠压检测（BOD）	• 两级BOD选择，带欠压中断及复位（2.5V/2.0V）
低压复位（LVR）	• 低压复位1.7V阈值电压
安全	• 96位唯一标识符（UID） • 128位客户唯一标识符（UCID）
32位硬件除法器（HDIV）	• 有符号（二进制补码）整数计算 • 32位被除数、16位除数的计算能力 • 输出32位商和32位余数（16位余数随着符号扩展到32位）
存储器	
Flash	• 双bank 512KB内置应用代码空间（APROM），可OTA升级 • 单bank 256KB内置应用代码空间（APROM） • 最大8 KB内嵌内存为用户自定义引导代码（LDROM） • 2048字节只可执行加密保护空间（SPROM） • 内嵌内存支持512字节或者2048字节页擦除 • 快速内存编程验证带CRC-32校验计算 • 片上Flash支持在线编程（ICP）、在系统编程（ISP）和在应用编程（IAP）功能 • 支持通过SWD/ICE接口的2线ICP更新
SRAM	• 内嵌多达96 KB SRAM – 32 KB SRAM位于bank 0支持硬件奇偶校验和保持模式 – 32/32 KB SRAM位于bank 1和bank 2 • 支持字节、半字、字访问 • 支持PDMA模式

循环冗余计算（CRC）	• 支持 CRC-CCITT，CRC-8，CRC-16 和 CRC-32 多项式 • 可编程初始值和种子值 • 输入数据和 CRC 校验和支持可编程顺序逆设置和补码设置 • 8 位，16 位和 32 位数据宽度 • 8 位写模式 1-AHB 时钟操作周期 • 16 位写模式 2-AHB 时钟操作周期 • 32 位写模式 4-AHB 时钟操作周期 • 使用 DMA 在执行 CRC 操作时写入数据
外设DMA （PDMA）	• 多达 9 个独立且可配置的通道，用于在存储器和外围设备之间自动传输数据 • 基本和分散加载传输模式 • 每个通道都支持使用分散加载传输模式的循环缓冲区管理 • 固定优先级和循环优先级模式 • 单一和批量传输类型 • 字节、半字和字传输单元，计数高达 65536 • 增量或固定源和目标地址
时钟	
外部时钟源	• 4~32 MHz外部高速晶振（HXT）提供精确的时间操作 • 32.7688 kHz外部低速晶振（LXT）可为RTC和低功耗系统运行提供时钟 • 支持外部晶体振荡器和时钟故障检测异常（NMI）
内部时钟源	• 48 MHz 内部高速 RC 振荡器（HIRC）可提供给 USB 时钟，无需外部晶振 • 38.4 kHz 内部低速 RC 振荡器（LIRC）用于看门狗时钟和唤醒操作 • 144 MHz 片内 PLL，时钟源来自 HIRC 或 HXT，允许 CPU 运行到最大 CPU 频率不需要高速晶振
实时时钟（RTC）	• RTC时钟源包括低速外部晶振（LXT） • 可从空闲或者掉电模式唤醒CPU • 支持5秒内±5ppm软件时钟精度补偿 • 支持报警寄存器（秒、分、时、日、月、年） • 支持RTC时钟节拍和报警匹配中断 • 自动闰年识别 • 支持输出1Hz 时钟校准
计时器	
32位计时器	• 最多4组32位定时器，带有24位递增计数器和一个独立时钟源的8位预分频计数器

	• 单次，周期，反转输出和连续计数运行模式 • 支持来自外部引脚输入的事件计数功能 • 支持脉宽测量和复位24位递增计数器的外部捕获引脚 • 如果定时器中断信号产生，支持芯片唤醒功能
PWM （PWM）	• 最多两组PWM模块，每组模块提供三个16位计数器和6个通道输出 • 多达12个独立输入捕获通道，带16位分辨率计数器 • 支持12位预分频的死区插入 • 向上、向下或上下PWM计数类型 • 支持3个互补成对PWM输出通道的互补模式 • 计数器同步启动功能 • 带自动恢复的刹车功能 • 每个PWM通道支持屏蔽功能和三态输出 • 能够触发ADC启动转换
基本PWM （BPWM）	• 两组带12位时钟预分频的16位计数器，12个可达144MHz的PWM输出通道 • 多达 6 个独立输入捕获通道，带 16 位分辨率计数器 • 向上、向下或上下 PWM 计数类型 • 计数器同步启动功能 • 每个 PWM 通道支持屏蔽功能和三态输出 • 能够触发ADC启动转换
看门狗	• 20位向上计数器为WDT溢出时间间隔 • 支持多个时钟源从LIRC （默认选择）, HCLK/2048 和LXT带9个可选超时时间段 • 可用作掉电模式或者空闲模式下的唤醒 • 看门狗溢出可触发中断或复位芯片 • 支持看门狗定时器复位延时周期，包括1026、130、18或3个WDT_CLK的复位延时时间 • 芯片上电或者复位时可配置强制WDT使能
窗口看门狗	• 时钟源来自HCLK/2048或LIRC；窗口设定由11位预分频的6位计数器 • 在空闲/掉电模式下挂起
模拟接口	
ADC	• 模拟输入电压范围: 0 ~ AV_{DD} • 一个12位，2MSPS SAR ADC，最多16个单端输入通道或8对差分输入；10位精度保证 • 带隙电压VBG输入的内部通道 • 支持外部V_{REF} 脚 • 支持校准能力

	• 四种操作模式：单次模式，Burst模式，单周期扫描模式和连续扫描模式 • 模拟到数字转换可被软件触发（ADST），外部引脚（STADC）以及定时器0~3溢出脉冲触发 • 每个通道的转换结果存储在相应数据寄存器内，并带有有效和覆盖标志 • 支持通过比较模式功能监视转换结果 • 可配置ADC外部采样时间 • PDMA操作 • 支持浮动检测功能
模拟比较器 （ACMP）	• 两组模拟比较器 • 在正输入端支持四个多路I/O管脚 • 在负输入端支持I/O引脚，带隙电压和16级AV_{DD}或V_{REF}分压 • 支持掉电模式下通过中断唤醒 • 支持PWM刹车事件触发和PWM循环控制 • 支持窗口比较模式和窗口锁存模式 • 支持滞后功能 • 支持校准功能
通信接口	
低功耗UART	• 低功耗UART波特率最高7.2 MHz • 支持自动波特率检测和波特率补偿功能 • 支持低功耗UART （LPUART）: 时钟源为LXT（32.768 KHz），波特率为9600bps时UART可正常工作在掉电模式即使系统时钟停止 • 16位 FIFOs 带可编程阀值触发 • 自动流控制（nCTS和nRTS） • 支持IrDA （SIR）功能 • 支持RS-485 9位模式和传输方向控制 • 支持 nCTS, 输入数据, 数据 FIFO 达到阀值以及RS-485 地址匹配（AAD 模式） 在空闲模式下唤醒功能 • 支持硬件或软件使能编程nRTS引脚控制RS-485传输方向 • 支持唤醒功能 • 8位接收器FIFO超时检测功能 • 支持中断错误、帧错误、奇偶校验错误和接收/发送FIFO溢出检测功能 • PDMA操作 • 支持单线功能模式

I²C	• 两组I²C设备，支持主机/从机模式 • 支持标准模式（100 kbps），快速模式（400 kbps），高速模式（1 Mbps） • 支持7位模式 • 可编程时钟，可实现多种速率控制 • 支持多地址识别（四个从机地址带掩码选择） • 支持多地址掉电模式唤醒功能 • PDMA操作 • I²C端口0支持SMBus和PMBus
Quad SPI	• SPI Quad控制器，支持主机/从机模式 • 在1.8V~3.6V电源系统下，主机模式最高24 MHz，从机模式最高16MHz • 支持双线和四线I/O传输模式 • 支持单数据通道半双工传输 • 支持只接收模式 • 发送位长度可配置8到32位 • 提供独立发送和接收8级FIFO缓冲区 • 支持MSB 优先或LSB优先传输序列 • 支持字节重新排序功能 • 支持字节或字挂起模式 • 支持3线，无从机选择信号，双向接口 • PDMA操作 • 支持2位传输模式
SPI/I²S	• SPI/I²S 控制器，支持主机/从机模式 **SPI** • 在 1.8V~3.6V 电源系统下，主机模式最高 24 MHz，从机模式最高 16 MHz • 发送位长度可配置 8 到 32 位 • 提供单独发送和接收4级32位（或8级16位）FIFO缓冲区 • 支持 MSB 优先或 LSB 优先传输序列 • 支持字节重新排序功能 • 支持字节或字挂起模式 • 支持单数据通道半双工传输 • 支持只接收模式 • PDMA操作 **I²S** • 支持8位、16位、24位和32位的单声道和立体声音频数据。

	• 提供独立发送和接收4级FIFO缓冲区 • 支持 PCM A模式, PCM B模式, I^2S和MSB对齐的数据格式 • PDMA操作 • 当缓冲区和可编程边界交叉时，生成中断请求
通用串行控制接口（**USCI**）	• 两组USCI，可配置为UART, SPI或I^2C功能 • 支持单字节TX和RX缓存模式 **UART** • 支持一个发送缓存和两个接收缓存用于数据传输 • 支持硬件自动流量控制和可编程流控制触发电平 • 9位数据传输 • 通过波特率发生器的内置捕获事件检测波特率 • 支持唤醒功能 • PDMA操作 **SPI** • 支持主机和从机运行模式 • 支持一个发送缓存和两个接收缓存用于数据传输 • 可配置传输字长度从4位到16位 • 支持 MSB 优先或 LSB 优先传输序列 • 支持字传输挂起功能 • 支持 3 线，无从机选择信号，双向接口 • 支持唤醒功能：输入从机选择信号改变 • PDMA操作 • 支持单数据通道半双工传输 **I^2C** • 支持兼容主机和从机设备 • 支持一个发送缓存和两个接收缓存用于数据传输 • 支持标准模式（100 kbps）, 快速模式（400 kbps）, 高速模式（1 Mbps） • 支持7位模式 （10位模式不支持） • 支持10位总线超时 • 支持总线监控模式 • 支持掉电状况下由数据或是地址匹配唤醒 • 支持多地址识别 • 支持设备地址标识 • 可编程建立/保持时间
外部总线接口 （**EBI**）	• 支持多达两个存储块，可单独调整时序参数。 • 每个块支持专用的外部芯片选择管脚，具有极性控制和高达1 MB的寻址空间 • 8位/16位数据宽度

	• 支持16位数据长度时字节写 • 不同访问条件下的可配置空闲周期：写命令完成空闲（W2X）和连续读空闲（R2R） • 支持地址/数据复用模式 • 支持地址总线和数据总线分离模式 • 支持LCD接口i80模式 • PDMA操作
GPIO	• 支持四种I/O模式：准双向，推挽输出，开漏输出和高阻态输入 • 可配置中断源设置为边沿/电平触发 • 准双向I/O模式可使能I/O引脚内部上拉电阻 • 支持5V输入耐压功能，模拟引脚（PA.10，PA.11，PB.0~PB.15，PF.2~PF.5）除外 • 使能引脚中断功能将使能唤醒功能 • 输入施密特触发功能
高级连接	
USB 2.0 全速片上收发器	• 符合USB 2.0版规范 • 支持总线空闲3ms以上进入挂起功能 • 8个可配置端点，可配置为同步，批量，中断和控制类型 • 512个可配置为端点缓存的RAM • 远程唤醒能力 • 支持无晶振功能 • 帧开始（SOF）锁定时钟脉冲的产生 • USB 2.0链路电源管理

3 料号信息

3.1 封装类型

Part No.	TSSOP20	TSSOP28	QFN33	LQFP48	LQFP64	LQFP128
M031xB	M031FB0AE	M031EB0AE	M031TB0AE			
M031xC	M031FC1AE	M031EC1AE	M031TC1AE	M031LC2AE	M031SC2AE	
M031xD			M031TD2AE	M031LD2AE	M031SD2AE	
M031xE			M031TE3AE	M031LE3AE	M031SE3AE	
M031xG				M031LG6AE M031LG8AE	M031SG6AE M031SG8AE	M031KG6AE M031KG8AE
M031xI					M031SIAAE	M031KIAAE
M032xC	M032FC1AE	M032EC1AE	M032TC1AE	M032LC2AE		
M032xD			M032TD2AE	M032LD2AE		
M032xE				M032LE3AE	M032SE3AE	
M032xG				M032LG6AE M032LG8AE	M032SG6AE M032SG8AE	M032KG6AE M032KG8AE
M032xI					M032SIAAE	M032KIAAE

3.2 M031/M032系列选型指南

3.2.1 M031基本系列（M031Fx / M031Ex / M031Tx）

型号		M031							
		FB0AE	FC1AE	EB0AE	EC1AE	TB0AE	TC1AE	TD2AE	TE3AE
Flash （KB）		16	32	16	32	16	32	64	128
SRAM （KB）		2	4	2	4	2	4	8	16
LDROM （KB）		2	2	2	2	2	2	2	4
SPROM （字节）		512							
系统频率（MHz）		48							
PLL （MHz）		-	-	-	-	-	-	96	96
I/O		15	15	23	23	27	27	27	27
32位定时器		2	4	2	4	2	4	4	4
连接性	USCI	-	-	-	-	-	-	1	1
	UART	3							
	SPI/I²S	1							
	QSPI	-							
	I²C/SMBus	2/0							
	USB FS	-							
PWM		6	6	6	6	6	6	12	12
BPWM		-							
PDMA		-	2	-	2	-	2	5	5
EBI		-							
HDIV		√							
CRC		√							
IEC-60730		-							
HXT		√							
LXT		-	-	-	-	-	√	√	√
RTC		-							
模拟比较器		-	-	-	-	-	-	2	2
12-bit SAR ADC		7	7	9	9	10	10	10	10
封装		TSSOP20	TSSOP20	TSSOP28	TSSOP28	QFN33	QFN33	QFN33	QFN33

3.2.2 M031基本系列（M031Lx）

型号		M031				
		LC2AE	LD2AE	LE3AE	LG6AE	LG8AE
Flash （KB）		32	64	128	256	256
SRAM （KB）		8	8	16	32	64
LDROM （KB）		2	2	4	4	4
SPROM （字节）		512	512	512	2048	2048
系统频率 （MHz）		48	48	48	72	72
PLL （MHz）		96	96	96	144	144
I/O		42				
32位定时器		4				
连接性	USCI	1	1	1	2	2
	UART	3	3	3	6	6
	SPI/I^2S	1				
	QSPI	-	-	-	1	1
	I²C/SMBus	2/0	2/0	2/0	2/1	2/1
	USB FS	-	-	-	-	-
PWM		12				
BPWM		-	-	-	12	12
PDMA		5	5	5	7	7
EBI		-	-	√	√	√
CRC		√				
HDIV		√				
IEC-60730		-	-	-	√	√
HXT		√				
LXT		√				
RTC		-	-	-	√	√
模拟比较器		2				
12-bit SAR ADC		12				
封装		LQFP48				

3.2.3 M031基本系列（M031Sx）

型号		M031					
		SC2AE	SD2AE	SE3AE	SG6AE	SG8AE	SIAAE
Flash （KB）		32	64	128	256	256	512
SRAM （KB）		8	8	16	32	64	96
LDROM （KB）		2	2	4	4	4	8
SPROM （字节）		512	512	512	2048	2048	2048
系统频率 （MHz）		48	48	48	72	72	72
PLL （MHz）		96	96	96	144	144	144
I/O		55					
32位定时器		4					
连接性	USCI	1	1	1	2	2	2
	UART	3	3	3	6	6	8
	SPI/I^2S	1					
	QSPI	-	-	-	1	1	1
	I^2C/SMBus	2/0	2/0	2/0	2/1	2/1	2/1
	USB FS	-					
PWM		12					
BPWM		-	-	-	12	12	12
PDMA		5	5	5	7	7	9
EBI		-	-	√	√	√	√
CRC		√					
HDIV		√					
IEC-60730		-	-	-	√	√	√
HXT		√					
LXT		√					
RTC		-	-	-	√	√	√
模拟比较器		2					
12-bit SAR ADC		16					
封装		LQFP64					

3.2.4 M031基本系列（M031Kx）

型号		M031		
		KG6AE	KG8AE	KIAAE
Flash （KB）		256	256	512
SRAM （KB）		32	64	96
LDROM （KB）		4	4	8
SPROM （字节）		2048		
系统频率 （MHz）		72		
PLL （MHz）		144		
I/O		111		
32位定时器		4		
连接性	USCI	2		
	UART	6	6	8
	SPI/I^2S	1		
	QSPI	1		
	I^2C/SMBus	2/1		
	USB FS	-		
PWM		12		
BPWM		12		
PDMA		7	7	9
EBI		√		
CRC		√		
HDIV		√		
IEC-60730		√		
HXT		√		
LXT		√		
RTC		√		
模拟比较器		2		
12-bit SAR ADC		16		
封装		LQFP128		

3.2.5 M032 USB系列（M032Fx / M032Ex / M032Tx）

型号		M032			
		FC1AE	EC1AE	TC1AE	TD2AE
Flash （KB）		32	32	32	64
SRAM （KB）		4	4	4	8
LDROM （KB）		2			
SPROM （字节）		512			
系统频率 （MHz）		48			
PLL （MHz）		-			
I/O		11	19	23	23
32位定时器		2	2	2	4
连接性	USCI	1	1	1	2
	UART	1			
	SPI/I²S	1			
	QSPI	-	-	-	1
	I²C/SMBus	-			
	USB FS	√			
PWM		-			
BPWM		6	6	6	12
PDMA		-	-	-	5
EBI		-			
CRC		-			
HDIV		-	-	-	√
IEC-60730		-			
HXT		-			
LXT		-			
RTC		-			
模拟比较器		-			
12-bit SAR ADC		3	9	10	10
封装		TSSOP20	TSSOP28	QFN33	QFN33

3.2.6 M032 USB系列（M032Lx）

型号		M032				
		LC2AE	LD2AE	LE3AE	LG6AE	LG8AE
Flash （KB）		32	64	128	256	256
SRAM （KB）		8	8	16	32	64
LDROM （KB）		2	2	4	4	4
SPROM （字节）		512	512	512	2048	2048
系统频率 （MHz）		48	48	48	72	72
PLL （MHz）		-	-	96	144	144
I/O		38				
32位定时器		4				
连接性	USCI	2	2	1	2	2
	UART	1	1	3	6	6
	SPI/I^2S	1				
	QSPI	1	1	-	1	1
	I^2C/SMBus	-	-	2/0	2/1	2/1
	USB FS	√				
PWM		-	-	12	12	12
BPWM		12	12	-	12	12
PDMA		5	5	5	7	7
EBI		-	-	√	√	√
CRC		-	-	√	√	√
HDIV		√				
IEC-60730		-	-	-	√	√
HXT		-	-	√	√	√
LXT		-	-	√	√	√
RTC		-	-	-	√	√
模拟比较器		-	-	2	2	2
12-bit SAR ADC		12				
封装		LQFP48				

3.2.7 M032 USB系列（M032Sx）

型号		M032			
		SE3AE	SG6AE	SG8AE	SIAAE
Flash （KB）		128	256	256	512
SRAM （KB）		16	32	64	96
LDROM （KB）		4	4	4	8
SPROM （字节）		512	2048	2048	2048
系统频率 （MHz）		48	72	72	72
PLL （MHz）		96	144	144	144
I/O		51			
32位定时器		4			
连接性	USCI	1	2	2	2
	UART	3	6	6	8
	SPI/I²S	1			
	QSPI	-	1	1	1
	I²C/SMBus	2/0	2/1	2/1	2/1
	USB FS	√			
PWM		12			
BPWM		-	12	12	12
PDMA		5	7	7	9
EBI		√			
CRC		√			
HDIV		√			
IEC-60730		-	√	√	√
HXT		√			
LXT		√			
RTC		-	√	√	√
模拟比较器		2			
12-bit SAR ADC		16			
封装		LQFP64			

3.2.8 M032 USB系列（M032Kx）

型号		M032		
		KG6AE	KG8AE	KIAAE
Flash （KB）		256	256	512
SRAM （KB）		32	64	96
LDROM （KB）		4	4	8
SPROM （字节）		2048		
系统频率 （MHz）		72		
PLL （MHz）		144		
I/O		107		
32位定时器		4		
连接性	USCI	2		
	UART	6	6	8
	SPI/I^2S	1		
	QSPI	1		
	I^2C/SMBus	2/1		
	USB FS	√		
PWM		12		
BPWM		12		
PDMA		7	7	9
EBI		√		
CRC		√		
HDIV		√		
IEC-60730		√		
HXT		√		
LXT		√		
RTC		√		
模拟比较器		2		
12-bit SAR ADC		16		
封装		LQFP128		

3.2.9 命名规则

M0	32	K	I	A	A	E
内核	产品线	封装	Flash	SRAM	保留	温度
Cortex®-M0	31: Base 32: USB	F: TSSOP20 （4.4x6.5 mm） E: TSSOP28 （4.4x9.7 mm） T: QFN33 （4x4 mm） L: LQFP48 （7x7 mm） S: LQFP64 （7x7 mm） K: LQFP128 （14x14 mm）	B: 16 KB C: 32 KB D: 64 KB E: 128 KB G: 256 KB I: 512 KB	0: 2 KB 1: 4 KB 2: 8/12 KB 3: 16 KB 6: 32 KB 8: 64 KB A: 96 KB		E:-40°C ~ 105°C

3.3 M031/M032系列特性比较表

章节	小节	-	M031xB/C/D/E	M031xG/I
		M032xC/D	M032xE	M032xG/I
系统管理	6.3.6 SRAM内存结构	●	●	-
	6.3.7 带奇偶校验功能的SRAM内存结构	-	-	●
FMC	6.4.4.3 物理和虚拟地址概念	-	-	●
	6.4.4.4 APROM 重启地址操作模式选项	-	-	-/●
	6.4.4.14 *3.3.1.1 高速缓存控制器*	-	-	●
	6.4.4.15 片上Flash存储器编程 64位编程和多字编程	-	-	●
	6.4.4.17 Flash 全为1校验	-	-	●
	ISP控制寄存器(FMC_ISPCTL) INTEN （FMC_ISPCTL[24]）	-	●	
ADC	6.25.5.11 PWM触发	-	●	●
	6.25.5.12 BPWM触发	●	-	●
	6.25.5.17 浮空检测功能	●	-	●
I^2C	6.16.5.2 操作模式 - 总线管理 (SMBus/PMBus 兼容) - 设备识别 – 从机地址 - 总线协议 - 地址解析协议(ARP) - 接收命令和数据应答控制 - 主机通知协议 - 总线管理警报 - 包错误检查 - 超时 - 总线管理超时 - 总线时钟低电平超时 - 总线空闲侦测	-	-	●
ACMP	6.26.5.7 校准功能	-	-/-/-/●	●
EBI	6.21.5.3 EBI数据宽度连接-地址总线与数据总线分开模式	-	-	●
	6.21.5.4 EBI操作控制 - 连续数据访问模式	-	-	●
USBD	6.22.7 寄存器描述	●	-	-

章节	小节	-	M031xB/C/D/E	M031xG/I
		M032xC/D	M032xE	M032xG/I
	USB配置寄存器 (USB_CFGx) DSQSYNC OUT Token传输			

4 引脚配置

用户可以在第3章或使用NuTool - PinConfig找到引脚配置信息。NuTool-PinConfigure包含所有型号的Nuvoton NuMicro®系列芯片，帮助用户正确、方便地配置多功能GPIO。

4.1 引脚配置

4.1.1 M031 系列引脚图

4.1.1.1 M031系列TSSOP 20引脚框图

对应料号:M031FB0AE, M031FC1AE

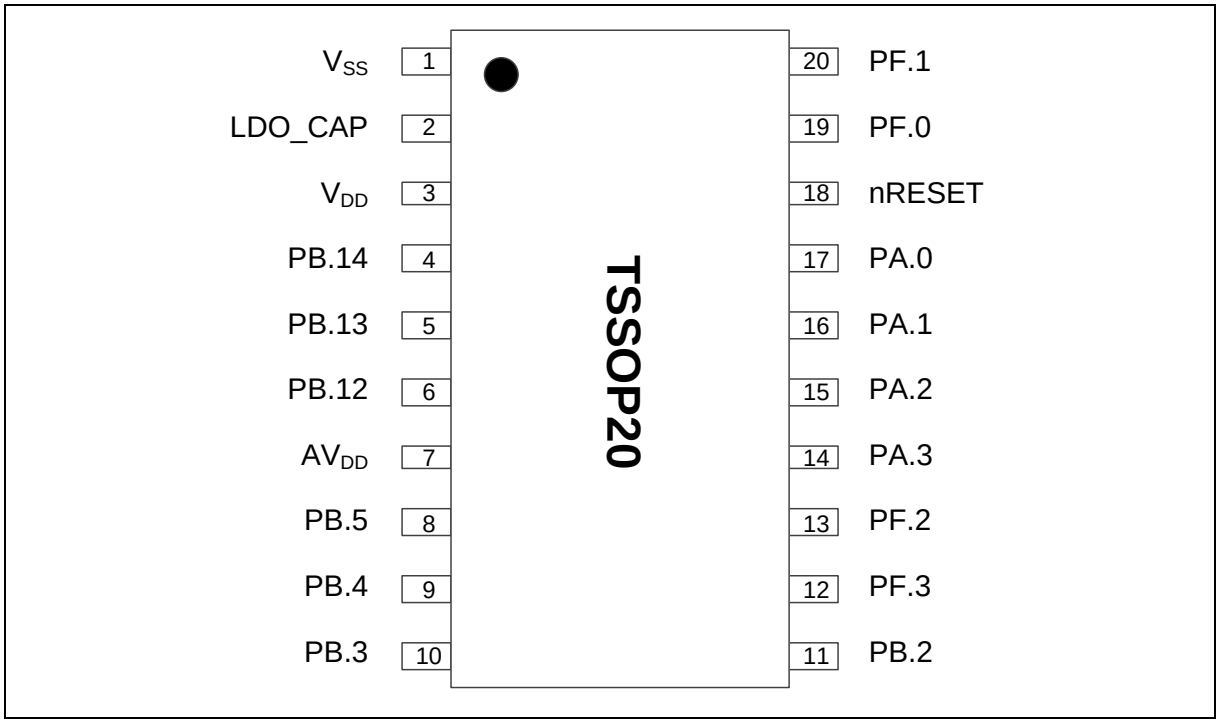


图 4.1-1 M031 系列 TSSOP 20 引脚框图

4.1.1.2 M031系列TSSOP 28引脚框图

对应料号:M031EB0AE, M031EC1AE

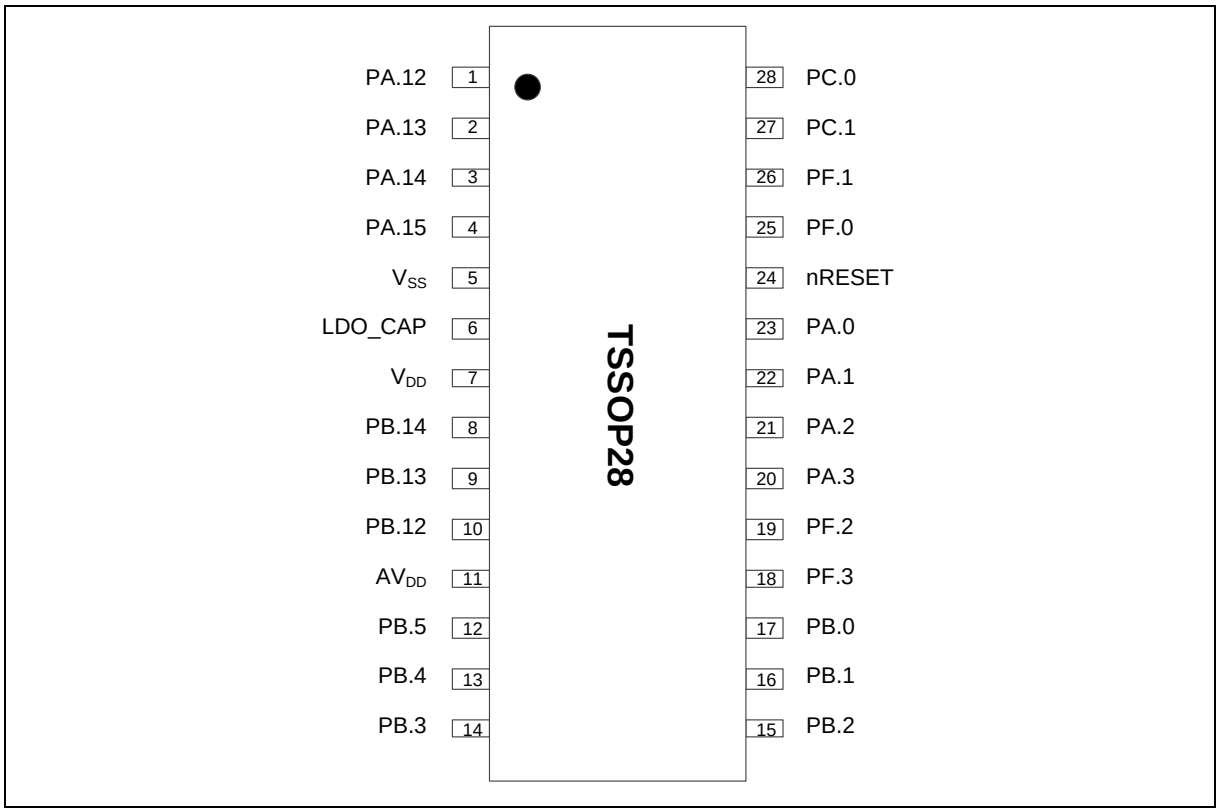


图 4.1-2 M031 系列 TSSOP 28 引脚框图

4.1.1.3 M031系列QFN 33引脚框图

对应料号:M031TB0AE, M031TC1AE, M031TD2AE, M031TE3AE

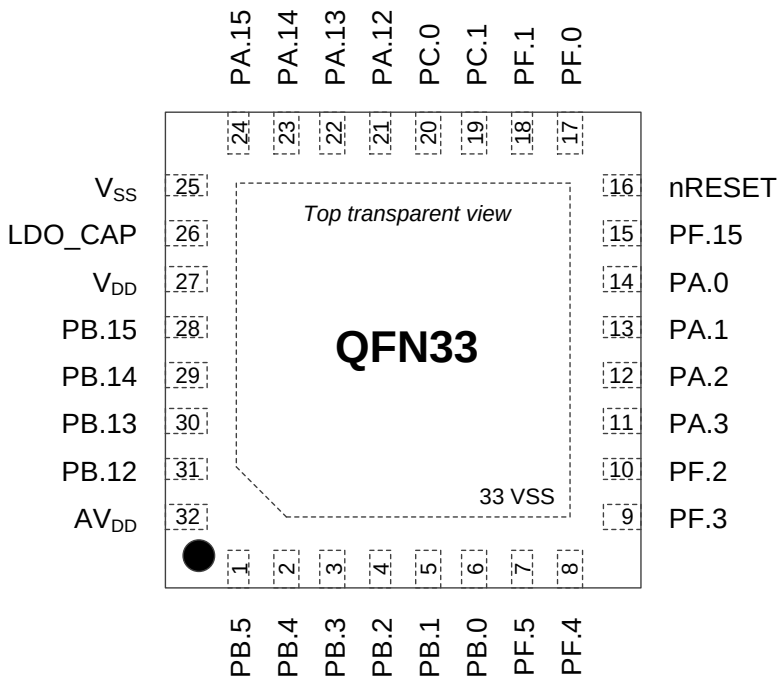


图 4.1-3 M031 系列 QFN 33 引脚框图

4.1.1.4 M031系列LQFP 48引脚框图

对应料号:M031LC2AE, M031LD2AE, M031LE3AE, M031LG6AE, M031LG8AE

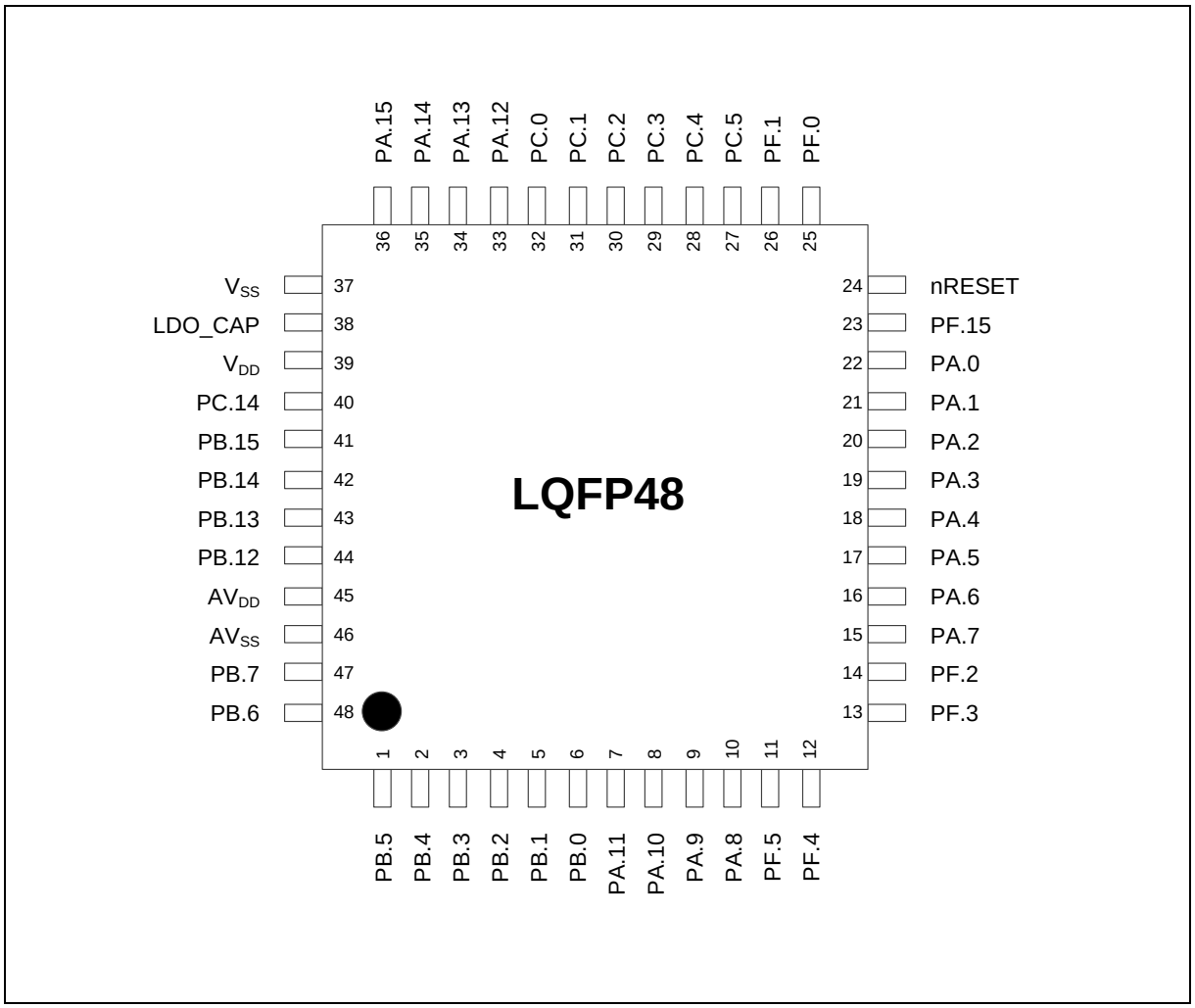


图 4.1-4 M031 系列 LQFP 48 引脚框图

4.1.1.5 M031系列LQFP 64引脚框图

对应料号:M031SC2AE, M031SD2AE, M031SE3AE, M031SG6AE, M031SG8AE, M031SIAAE

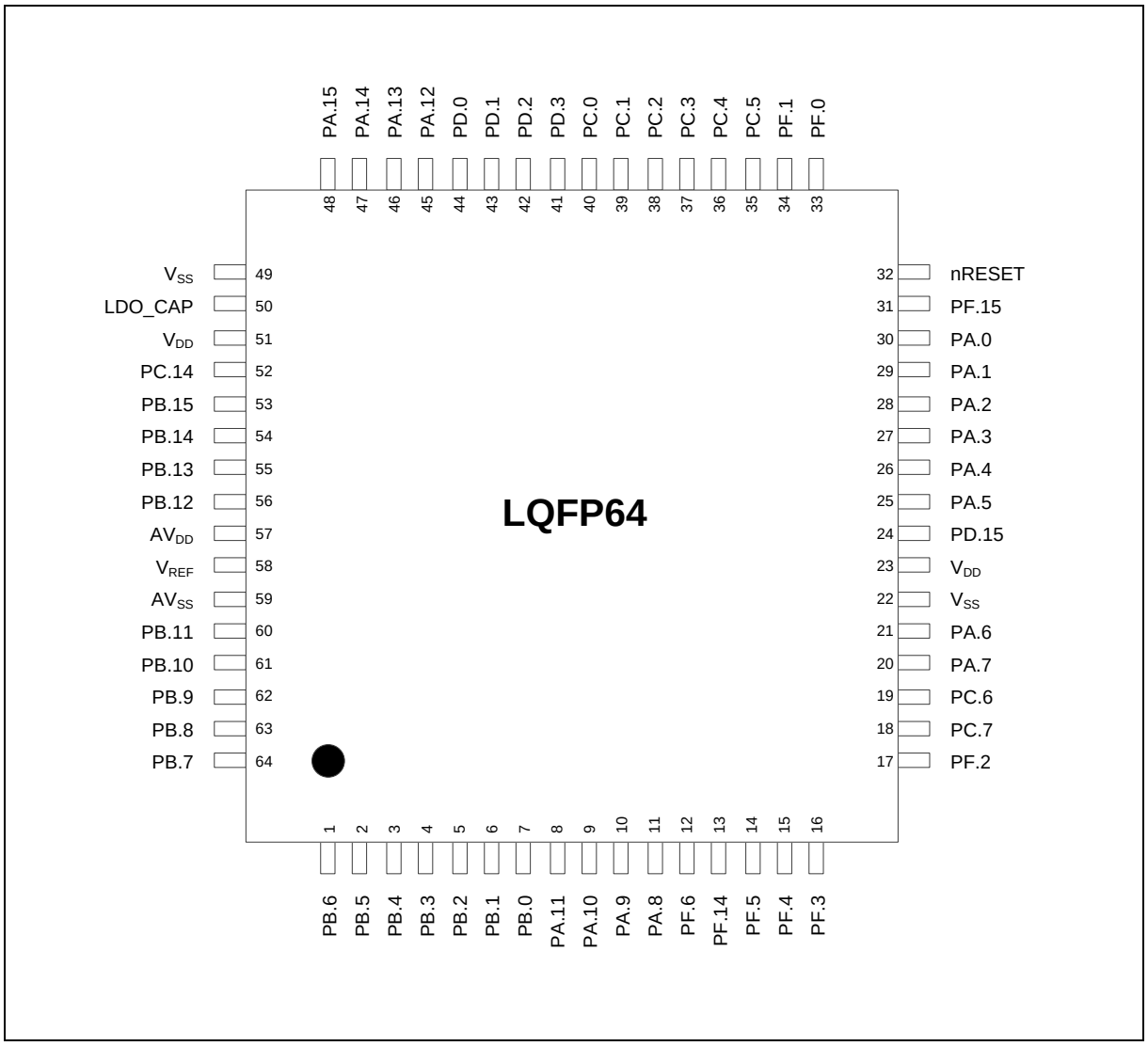


图 4.1-5 M031 系列 LQFP 64 引脚框图

4.1.1.6 M031系列LQFP 128引脚框图

对应料号:M031KG6AE, M031KG8AE, M031KIAAE

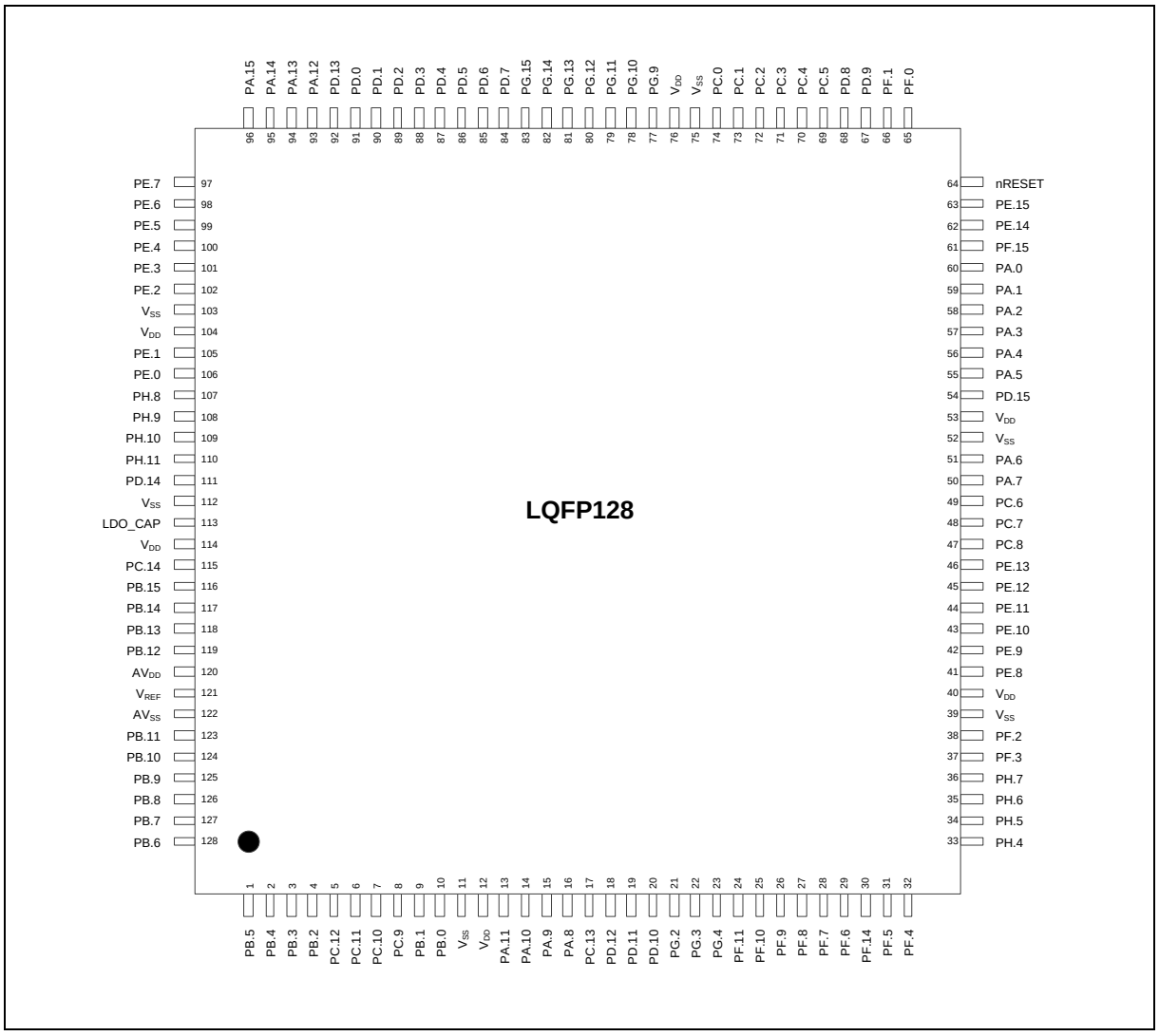


图 4.1-6 M031 系列 LQFP 128 引脚框图

4.1.2 M031系列多功能引脚框图

4.1.2.1 M031系列TSSOP 20多功能引脚框图

对应料号:M031FB0AE, M031FC1AE

M031FB0AE

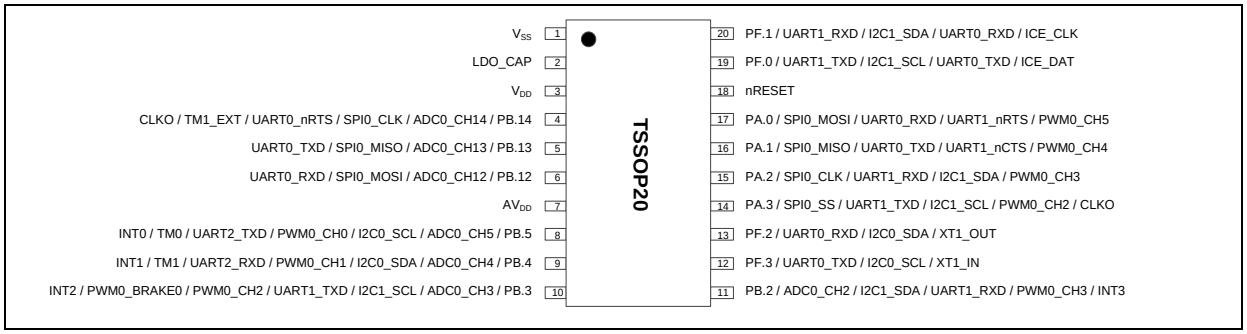


图 4.1-7 M031FB0AE 多功能引脚框图

引脚	M031FB0AE引脚功能
1	VSS
2	LDO_CAP
3	V_{DD}
4	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
5	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD
6	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD
7	AV_{DD}
8	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
9	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
10	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / INT2
11	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / INT3
12	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
13	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
14	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO
15	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
16	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
17	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
18	nRESET
19	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
20	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK

表 4.1-1 M031EB0AE 多功能引脚表

M031FC1AE

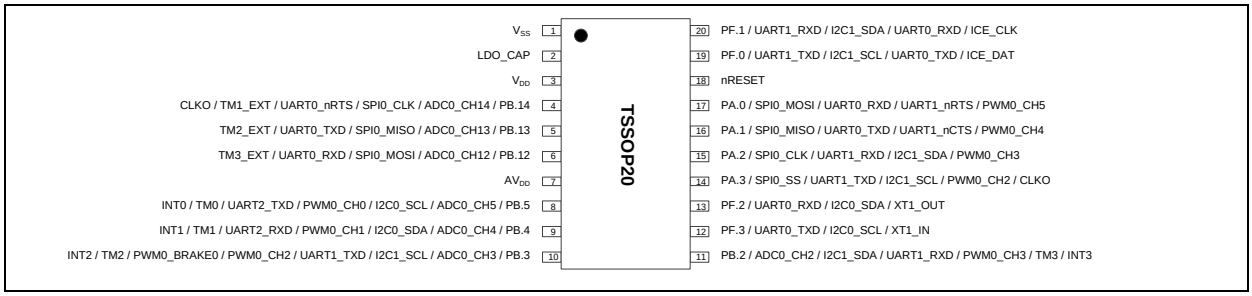


图 4.1-8 M031FC1AE 多功能引脚框图

引脚	**M031FC1AE引脚功能**
1	VSS
2	LDO_CAP
3	V_{DD}
4	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
5	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / TM2_EXT
6	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / TM3_EXT
7	AV_{DD}
8	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
9	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
10	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
11	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
12	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
13	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
14	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO
15	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
16	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
17	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
18	nRESET
19	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
20	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK

表 4.1-2 M031FC1AE 多功能引脚表

4.1.2.2 M031系列TSSOP 28多功能引脚框图

对应料号:M031EB0AE, M031EC1AE

M031EB0AE

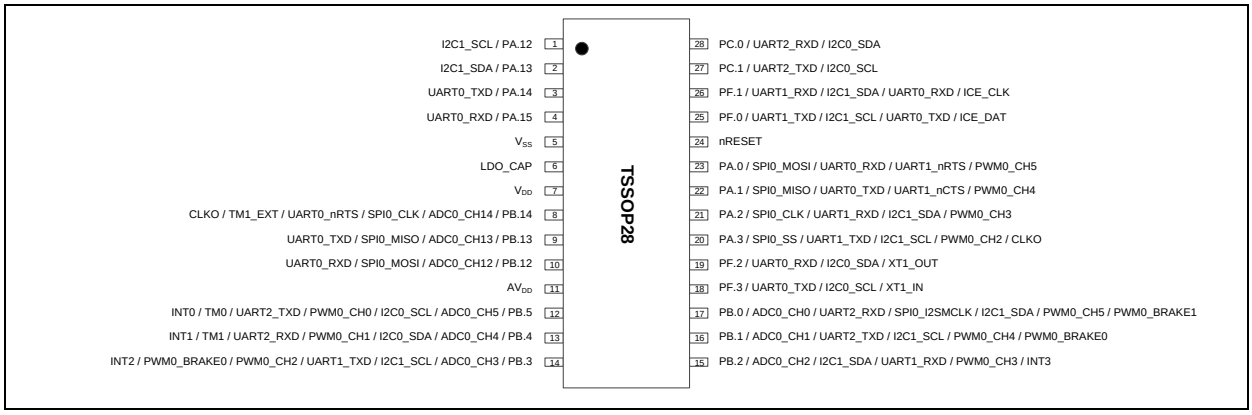


图 4.1-9 M031EB0AE 多功能引脚框图

引脚	M031EB0AE引脚功能
1	PA.12 / I2C1_SCL
2	PA.13 / I2C1_SDA
3	PA.14 / UART0_TXD
4	PA.15 / UART0_RXD
5	VSS
6	LDO_CAP
7	V_{DD}
8	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
9	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD
10	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD
11	AV_{DD}
12	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
13	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
14	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / INT2
15	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / INT3
16	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE0
17	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE1
18	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
19	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
20	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO

引脚	M031EB0AE引脚功能
21	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
22	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
23	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.1 / UART2_TXD / I2C0_SCL
28	PC.0 / UART2_RXD / I2C0_SDA

表 4.1-3 M031EB0AE 多功能引脚表

M031EC1AE

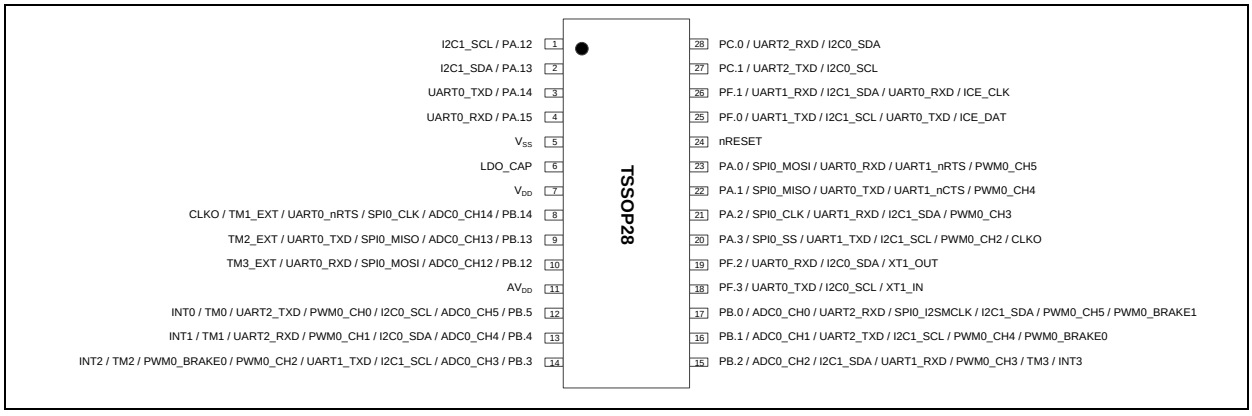


图 4.1-10 M031EC1AE 多功能引脚框图

引脚	M031EC1AE引脚功能
1	PA.12 / I2C1_SCL
2	PA.13 / I2C1_SDA
3	PA.14 / UART0_TXD
4	PA.15 / UART0_RXD
5	VSS
6	LDO_CAP
7	V_{DD}
8	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
9	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / TM2_EXT
10	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / TM3_EXT
11	AV_{DD}
12	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0

引脚	M031EC1AE引脚功能
13	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
14	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
15	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
16	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE0
17	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE1
18	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
19	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
20	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO
21	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
22	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
23	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.1 / UART2_TXD / I2C0_SCL
28	PC.0 / UART2_RXD / I2C0_SDA

表 4.1-4 M031EC1AE 多功能引脚表

4.1.2.3 M031系列QFN 33多功能引脚框图

对应料号:M031TB0AE, M031TC1AE, M031TD2AE, M031TE3AE

M031TB0AE

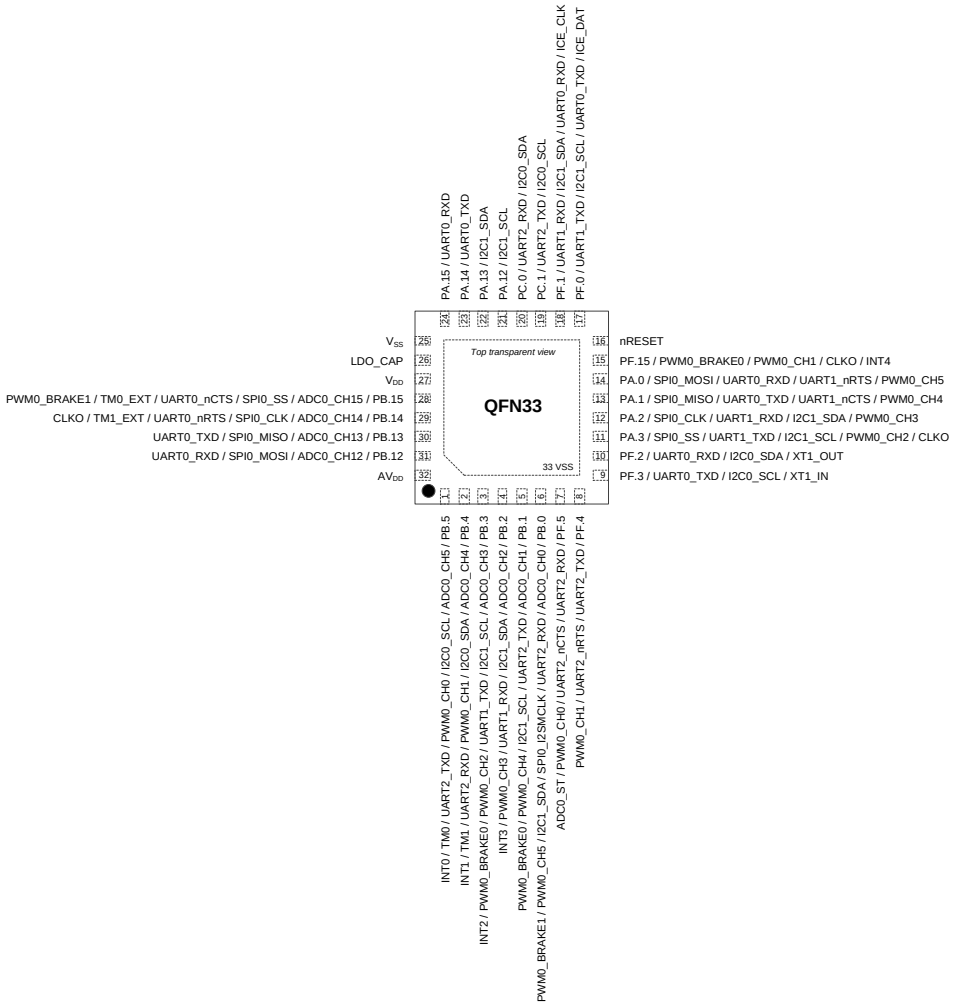


图 4.1-11 M031TB0AE 多功能引脚框图

引脚	M031TB0AE引脚功能
1	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M031TB0AE引脚功能
3	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / INT2
4	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE1
7	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / ADC0_ST
8	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1
9	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO
12	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
14	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
15	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / CLKO / INT4
16	nRESET
17	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	PC.1 / UART2_TXD / I2C0_SCL
20	PC.0 / UART2_RXD / I2C0_SDA
21	PA.12 / I2C1_SCL
22	PA.13 / I2C1_SDA
23	PA.14 / UART0_TXD
24	PA.15 / UART0_RXD
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / UART0_nCTS / TM0_EXT / PWM0_BRAKE1
29	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD
31	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD
32	AV_{DD}

表 4.1-5 M031TB0AE 多功能引脚表

M031TC1AE

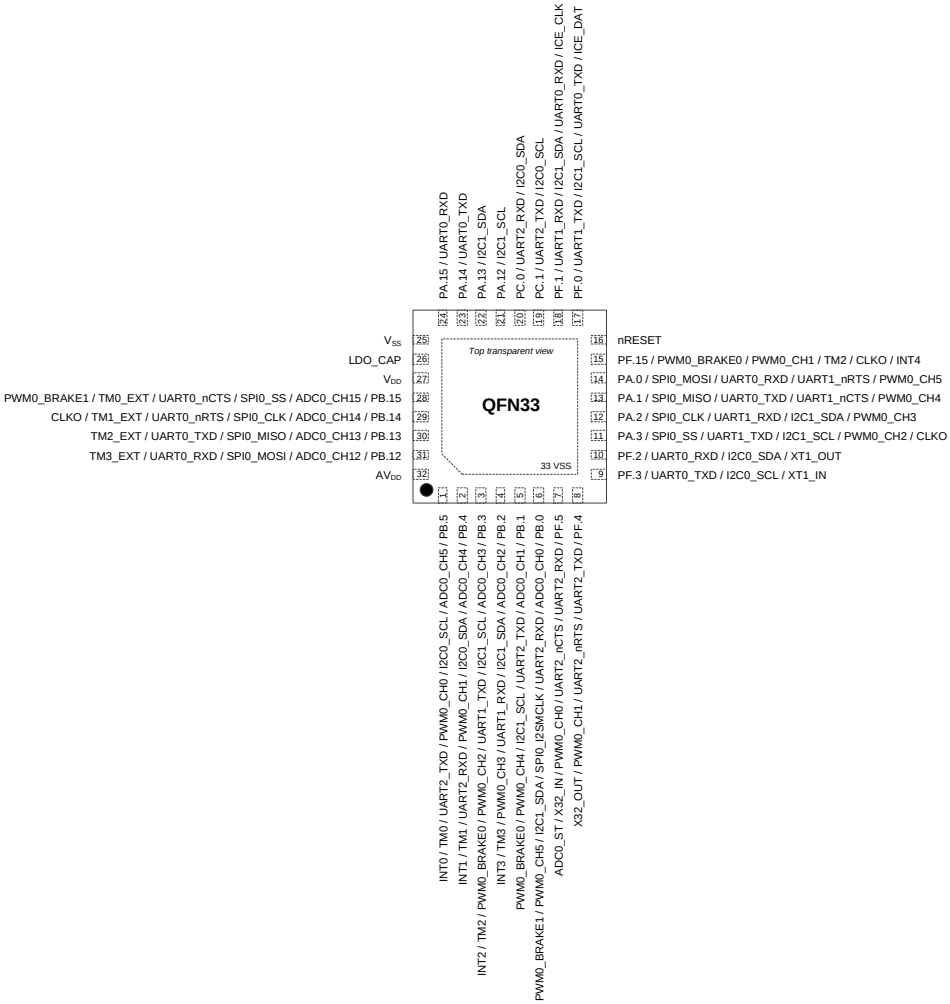


图 4.1-12 M031TC1AE 多功能引脚框图

引脚	M031TC1AE引脚功能
1	PB.5 / ADC0_CH5 / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE0

引脚	M031TC1AE引脚功能
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE1
7	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
8	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
9	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO
12	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
14	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
15	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
16	nRESET
17	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	PC.1 / UART2_TXD / I2C0_SCL
20	PC.0 / UART2_RXD / I2C0_SDA
21	PA.12 / I2C1_SCL
22	PA.13 / I2C1_SDA
23	PA.14 / UART0_TXD
24	PA.15 / UART0_RXD
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / UART0_nCTS / TM0_EXT / PWM0_BRAKE1
29	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / TM2_EXT
31	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / TM3_EXT
32	AV_{DD}

表 4.1-6 M031TC1AE 多功能引脚表

M031TD2AE

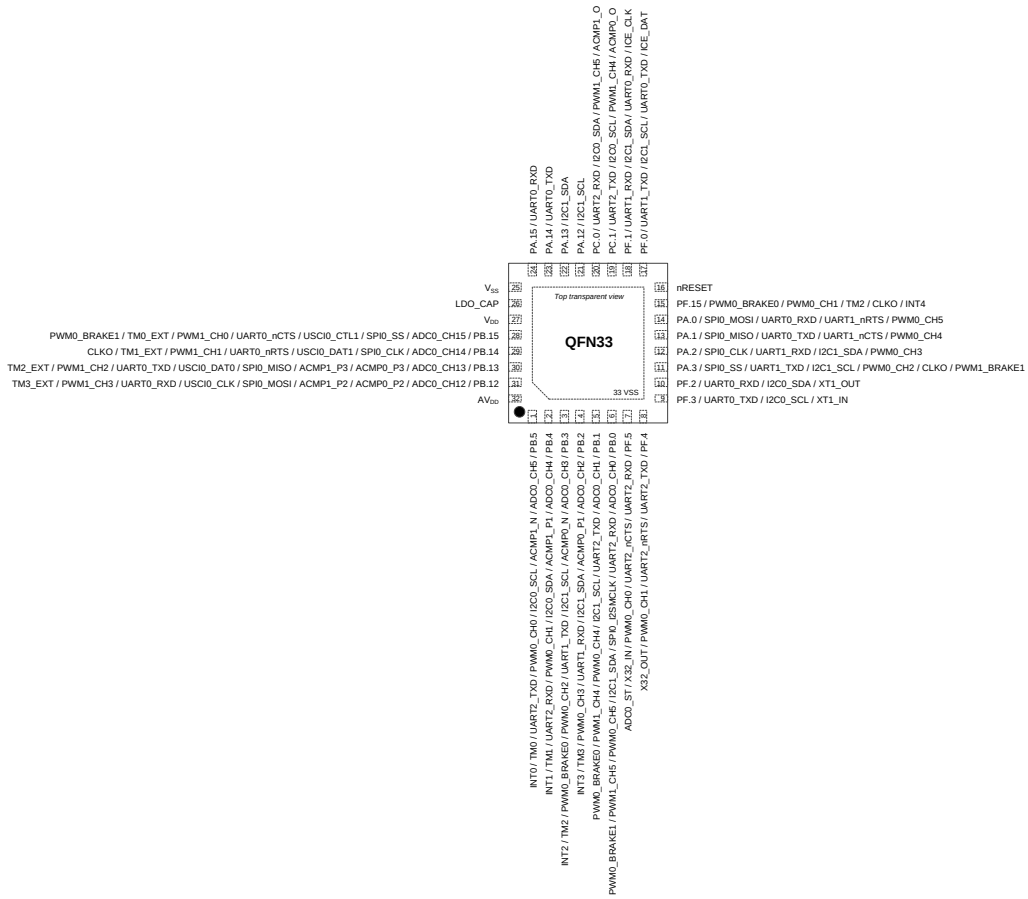


图 4.1-13 M031TD2AE 多功能引脚框图

引脚	M031TD2AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0

引脚	M031TD2AE引脚功能
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
8	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
9	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
12	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
14	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
15	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
16	nRESET
17	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
20	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
21	PA.12 / I2C1_SCL
22	PA.13 / I2C1_SDA
23	PA.14 / UART0_TXD
24	PA.15 / UART0_RXD
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
29	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
31	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
32	AV_{DD}

表 4.1-7 M031TD2AE 多功能引脚表

M031TE3AE

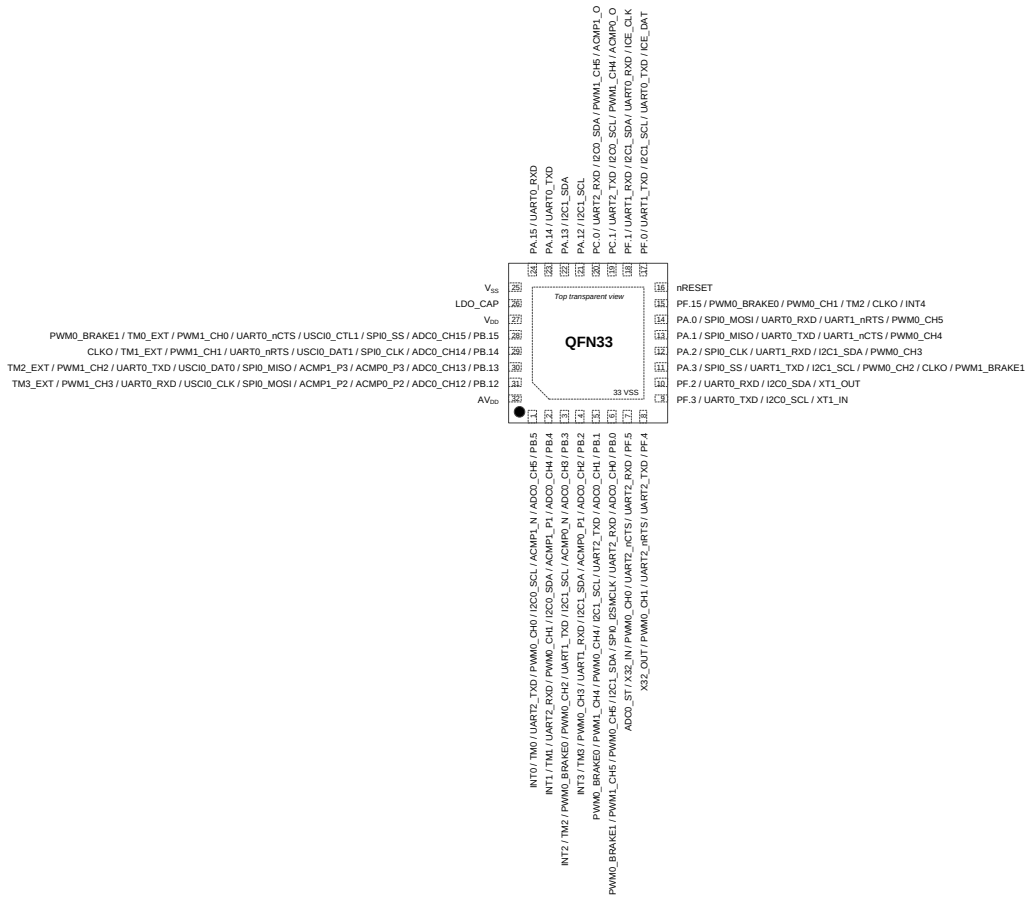


图 4.1-14 M031TE3AE 多功能引脚框图

引脚	M031TE3AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0

引脚	M031TE3AE引脚功能
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
8	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
9	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
12	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
14	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
15	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
16	nRESET
17	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
20	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
21	PA.12 / I2C1_SCL
22	PA.13 / I2C1_SDA
23	PA.14 / UART0_TXD
24	PA.15 / UART0_RXD
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
29	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
31	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
32	AV_{DD}

表4.1-8 M031TE3AE多功能引脚表

4.1.2.4 M031系列LQFP 48多功能引脚框图

对应料号:M031LC2AE, M031LD2AE, M031LE3AE, M031LG6AE, M031LG8AE

M031LC2AE

图 4.1-15 M031LC2AE 多功能引脚框图

引脚	M031LC2AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M031LC2AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / USCI0_CLK / TM0_EXT
8	PA.10 / ACMP1_P0 / USCI0_DAT0 / TM1_EXT
9	PA.9 / USCI0_DAT1 / UART1_TXD / TM2_EXT
10	PA.8 / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
13	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	PA.7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
18	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
19	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
22	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	PC.4 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	PC.3 / UART2_nRTS / PWM1_CH2
30	PC.2 / UART2_nCTS / PWM1_CH3
31	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
32	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	PA.12 / I2C1_SCL
34	PA.13 / I2C1_SDA
35	PA.14 / UART0_TXD
36	PA.15 / UART0_RXD
37	VSS

引脚	M031LC2AE引脚功能
38	LDO_CAP
39	V_{DD}
40	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
41	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / UART1_TXD / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / UART1_RXD / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-9 M031LC2AE 多功能引脚表

M031LD2AE

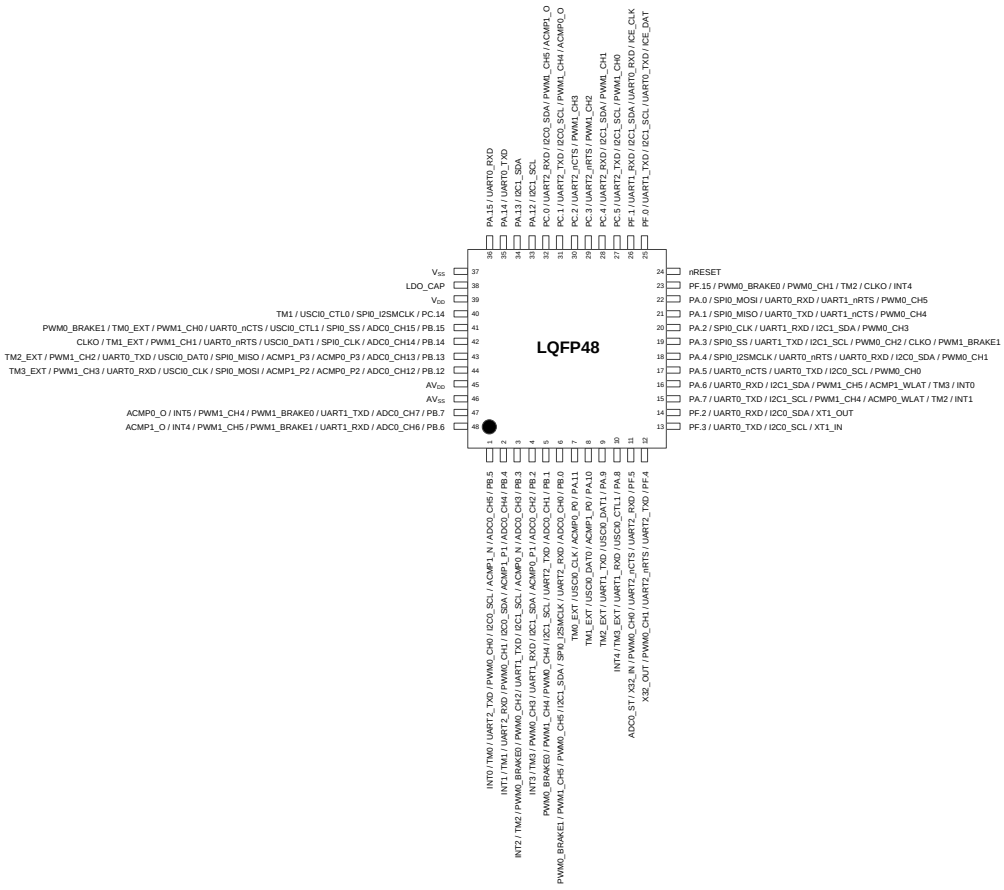


图 4.1-16 M031LD2AE 多功能引脚框图

引脚	**M031LD2AE引脚功能**
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0

引脚	M031LD2AE引脚功能
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / USCI0_CLK / TM0_EXT
8	PA.10 / ACMP1_P0 / USCI0_DAT0 / TM1_EXT
9	PA.9 / USCI0_DAT1 / UART1_TXD / TM2_EXT
10	PA.8 / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
13	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	PA.7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
18	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
19	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
22	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	PC.4 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	PC.3 / UART2_nRTS / PWM1_CH2
30	PC.2 / UART2_nCTS / PWM1_CH3
31	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
32	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	PA.12 / I2C1_SCL
34	PA.13 / I2C1_SDA
35	PA.14 / UART0_TXD
36	PA.15 / UART0_RXD
37	VSS
38	LDO_CAP
39	V_{DD}

引脚	M031LD2AE引脚功能
40	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
41	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / UART1_TXD / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / UART1_RXD / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-10 M031LD2AE 多功能引脚表

M031LE3AE

图 4.1-17 M031LE3AE 多功能引脚框图

引脚	M031LE3AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1

引脚	M031LE3AE引脚功能
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / XT1_OUT
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
18	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
19	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
22	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.5 / EBI_AD5 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	PC.4 / EBI_AD4 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	PC.3 / EBI_AD3 / UART2_nRTS / PWM1_CH2
30	PC.2 / EBI_AD2 / UART2_nCTS / PWM1_CH3
31	PC.1 / EBI_AD1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
32	PC.0 / EBI_AD0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	PA.12 / I2C1_SCL
34	PA.13 / I2C1_SDA
35	PA.14 / UART0_TXD
36	PA.15 / UART0_RXD
37	VSS
38	LDO_CAP
39	V_{DD}
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / TM1

引脚	M031LE3AE引脚功能
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / UART1_TXD / EBI_nCS0 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / UART1_RXD / EBI_nCS1 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-11 M031LE3AE 多功能引脚表

M031LG6AE

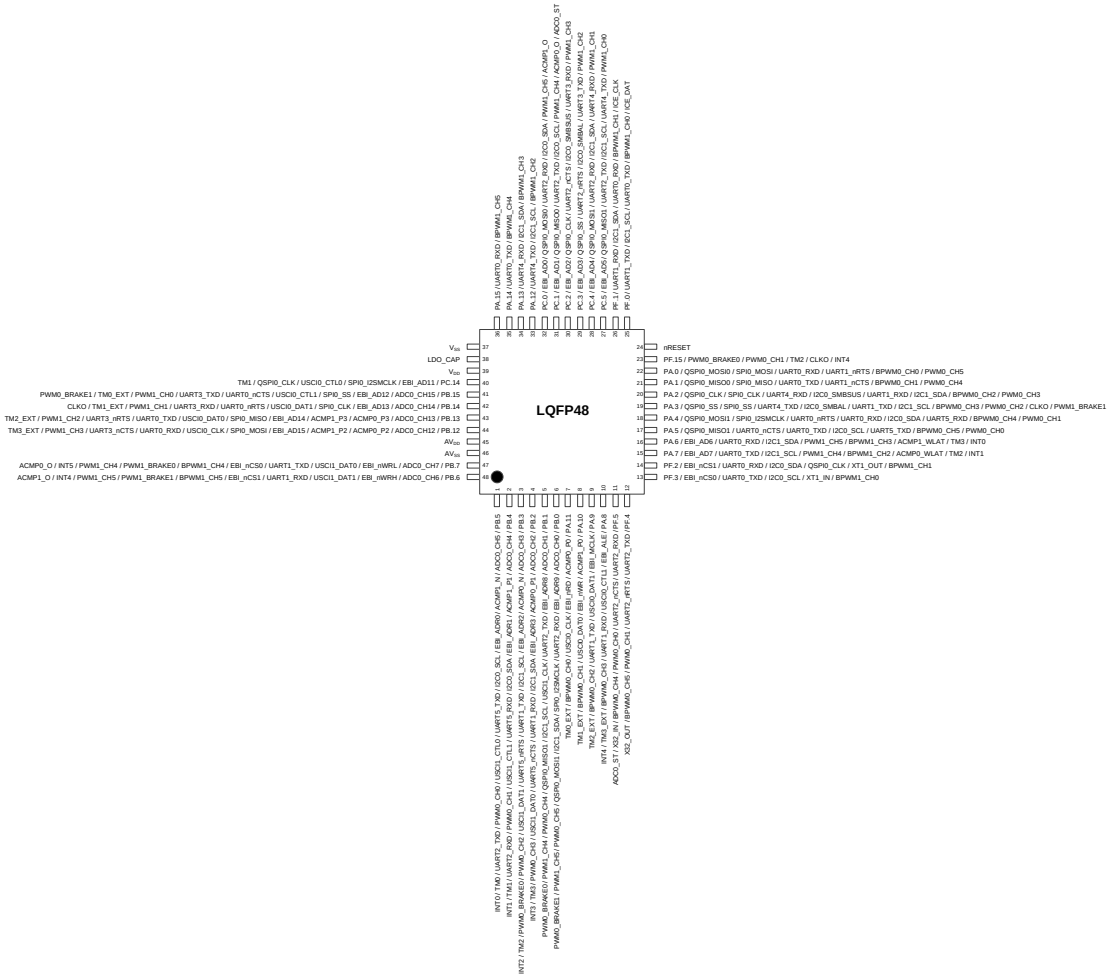


图 4.1-18 M031LG6AE 多功能引脚框图

引脚	M031LG6AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M031LG6AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
19	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
27	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
28	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
29	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
30	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
31	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
32	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
34	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3

引脚	M031LG6AE引脚功能
35	PA.14 / UART0_TXD / BPWM1_CH4
36	PA.15 / UART0_RXD / BPWM1_CH5
37	VSS
38	LDO_CAP
39	V_{DD}
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-12 M031LG6AE 多功能引脚表

M031LG8AE

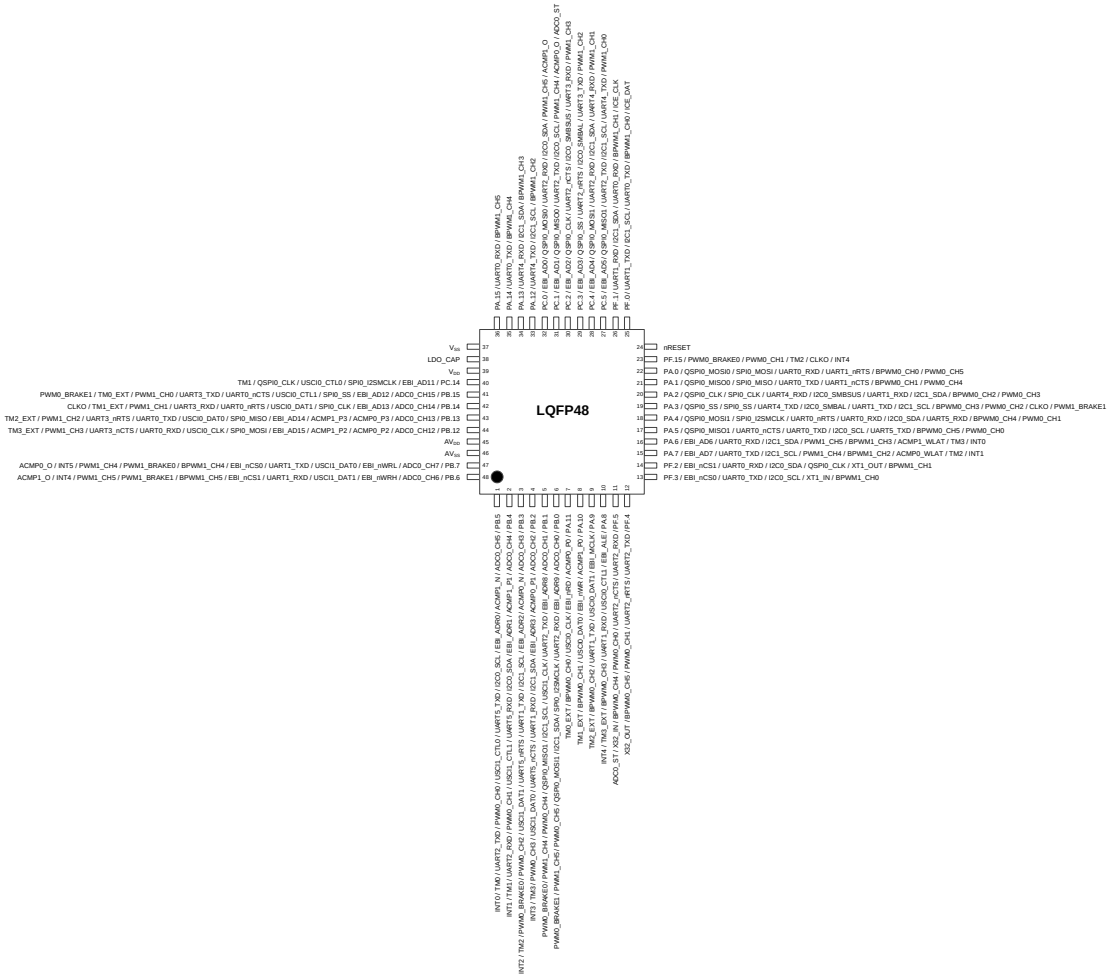


图 4.1-19 M031LG8AE 多功能引脚框图

引脚	M031LG8AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M031LG8AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
19	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
27	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
28	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
29	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
30	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
31	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
32	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
34	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3

引脚	M031LG8AE引脚功能
35	PA.14 / UART0_TXD / BPWM1_CH4
36	PA.15 / UART0_RXD / BPWM1_CH5
37	VSS
38	LDO_CAP
39	V_{DD}
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-13 M031LG8AE 多功能引脚表

4.1.2.5 M031系列LQFP 64多功能引脚框图

对应料号:M031SC2AE, M031SD2AE, M031SE3AE, M031SG6AE, M031SG8AE, M031SIAAE

M031SC2AE

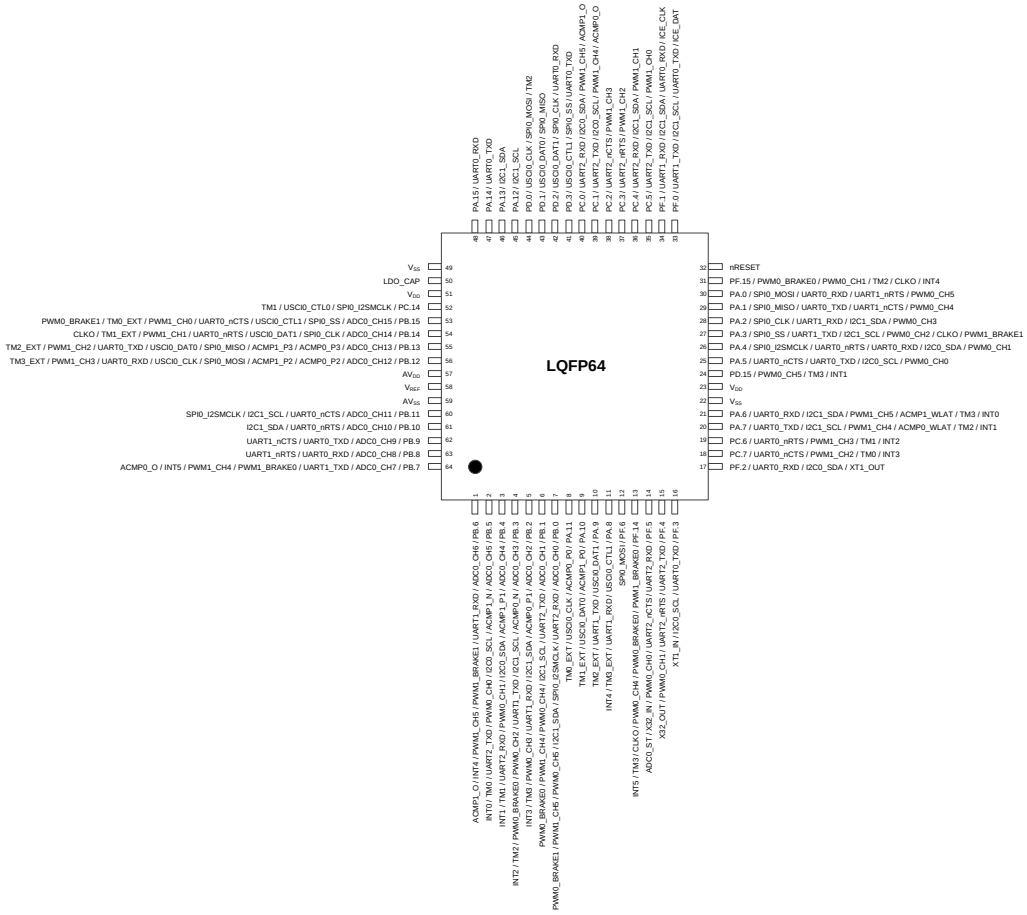


图 4.1-20 M031SC2AE 多功能引脚框图

引脚	M031SC2AE引脚功能
1	PB.6 / ADC0_CH6 / UART1_RXD / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M031SC2AE引脚功能
4	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / USCI0_CLK / TM0_EXT
9	PA.10 / ACMP1_P0 / USCI0_DAT0 / TM1_EXT
10	PA.9 / USCI0_DAT1 / UART1_TXD / TM2_EXT
11	PA.8 / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
12	PF.6 / SPI0_MOSI
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
16	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
17	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
18	PC.7 / UART0_nCTS / PWM1_CH2 / TM0 / INT3
19	PC.6 / UART0_nRTS / PWM1_CH3 / TM1 / INT2
20	PA.7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
26	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
27	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
30	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	PC.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
36	PC.4 / UART2_RXD / I2C1_SDA / PWM1_CH1
37	PC.3 / UART2_nRTS / PWM1_CH2

引脚	M031SC2AE引脚功能
38	PC.2 / UART2_nCTS / PWM1_CH3
39	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
40	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / USCI0_CTL1 / SPI0_SS / UART0_TXD
42	PD.2 / USCI0_DAT1 / SPI0_CLK / UART0_RXD
43	PD.1 / USCI0_DAT0 / SPI0_MISO
44	PD.0 / USCI0_CLK / SPI0_MOSI / TM2
45	PA.12 / I2C1_SCL
46	PA.13 / I2C1_SDA
47	PA.14 / UART0_TXD
48	PA.15 / UART0_RXD
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
53	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / UART0_nCTS / I2C1_SCL / SPI0_I2SMCLK
61	PB.10 / ADC0_CH10 / UART0_nRTS / I2C1_SDA
62	PB.9 / ADC0_CH9 / UART0_TXD / UART1_nCTS
63	PB.8 / ADC0_CH8 / UART0_RXD / UART1_nRTS
64	PB.7 / ADC0_CH7 / UART1_TXD / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-14 M031SC2AE 多功能引脚表

M031SD2AE

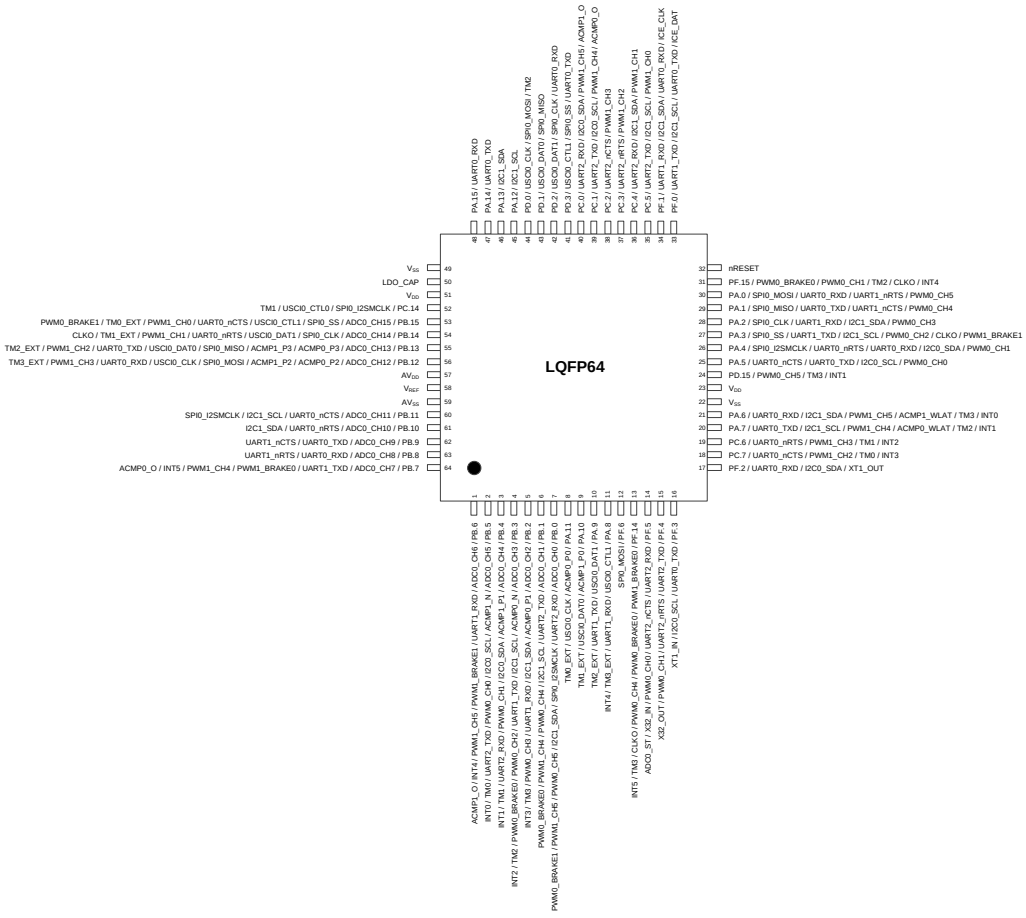


图 4.1-21 M031SD2AE 多功能引脚框图

引脚	M031SD2AE引脚功能
1	PB.6 / ADC0_CH6 / UART1_RXD / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
4	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3

引脚	M031SD2AE引脚功能
6	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / USCI0_CLK / TM0_EXT
9	PA.10 / ACMP1_P0 / USCI0_DAT0 / TM1_EXT
10	PA.9 / USCI0_DAT1 / UART1_TXD / TM2_EXT
11	PA.8 / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
12	PF.6 / SPI0_MOSI
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
16	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
17	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
18	PC.7 / UART0_nCTS / PWM1_CH2 / TM0 / INT3
19	PC.6 / UART0_nRTS / PWM1_CH3 / TM1 / INT2
20	PA.7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
26	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
27	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
30	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	PC.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
36	PC.4 / UART2_RXD / I2C1_SDA / PWM1_CH1
37	PC.3 / UART2_nRTS / PWM1_CH2
38	PC.2 / UART2_nCTS / PWM1_CH3
39	PC.1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O

引脚	M031SD2AE引脚功能
40	PC.0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / USCI0_CTL1 / SPI0_SS / UART0_TXD
42	PD.2 / USCI0_DAT1 / SPI0_CLK / UART0_RXD
43	PD.1 / USCI0_DAT0 / SPI0_MISO
44	PD.0 / USCI0_CLK / SPI0_MOSI / TM2
45	PA.12 / I2C1_SCL
46	PA.13 / I2C1_SDA
47	PA.14 / UART0_TXD
48	PA.15 / UART0_RXD
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
53	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / UART0_nCTS / I2C1_SCL / SPI0_I2SMCLK
61	PB.10 / ADC0_CH10 / UART0_nRTS / I2C1_SDA
62	PB.9 / ADC0_CH9 / UART0_TXD / UART1_nCTS
63	PB.8 / ADC0_CH8 / UART0_RXD / UART1_nRTS
64	PB.7 / ADC0_CH7 / UART1_TXD / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-15 M031SD2AE 多功能引脚表

M031SE3AE

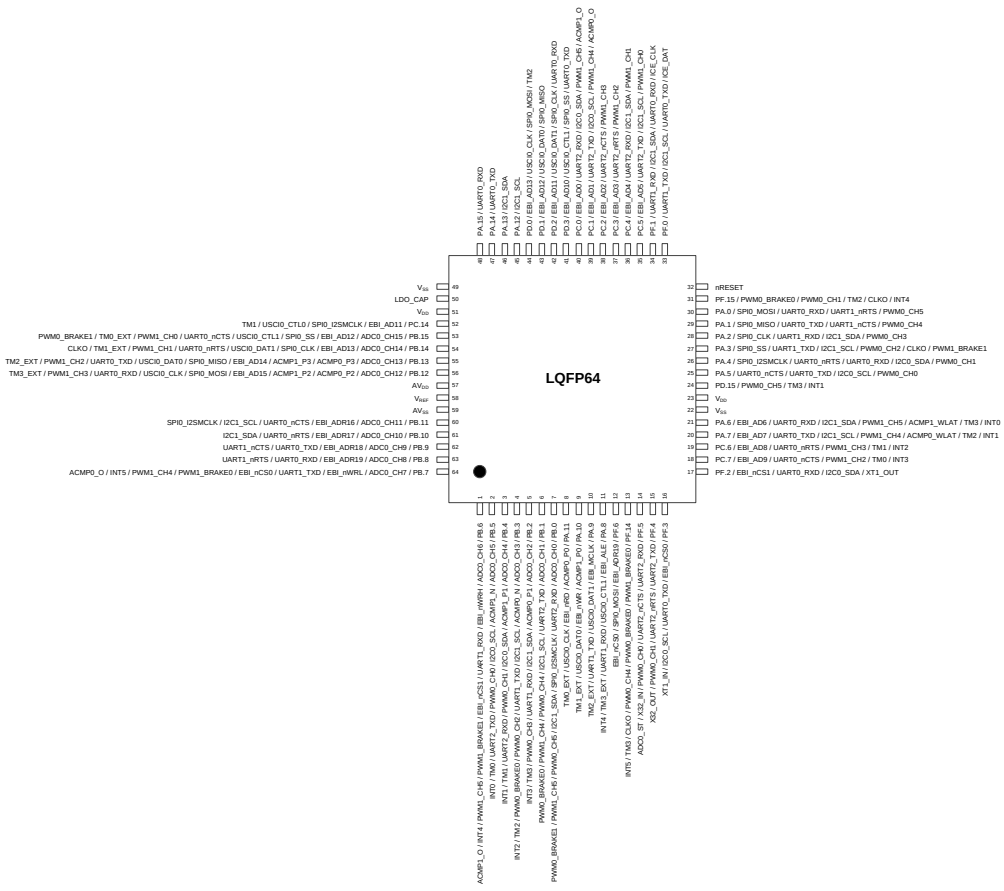


图 4.1-22 M031SE3AE 多功能引脚框图

引脚	**M031SE3AE引脚功能**
1	PB.6 / ADC0_CH6 / EBI_nWRH / UART1_RXD / EBI_nCS1 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
4	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3

引脚	M031SE3AE引脚功能
6	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / XT1_OUT
18	PC.7 / EBI_AD9 / UART0_nCTS / PWM1_CH2 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART0_nRTS / PWM1_CH3 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
26	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
27	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
30	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	PC.5 / EBI_AD5 / UART2_TXD / I2C1_SCL / PWM1_CH0
36	PC.4 / EBI_AD4 / UART2_RXD / I2C1_SDA / PWM1_CH1
37	PC.3 / EBI_AD3 / UART2_nRTS / PWM1_CH2
38	PC.2 / EBI_AD2 / UART2_nCTS / PWM1_CH3
39	PC.1 / EBI_AD1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O

引脚	M031SE3AE引脚功能
40	PC.0 / EBI_AD0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / TM2
45	PA.12 / I2C1_SCL
46	PA.13 / I2C1_SDA
47	PA.14 / UART0_TXD
48	PA.15 / UART0_RXD
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / I2C1_SCL / SPI0_I2SMCLK
61	PB.10 / ADC0_CH10 / EBI_ADR17 / UART0_nRTS / I2C1_SDA
62	PB.9 / ADC0_CH9 / EBI_ADR18 / UART0_TXD / UART1_nCTS
63	PB.8 / ADC0_CH8 / EBI_ADR19 / UART0_RXD / UART1_nRTS
64	PB.7 / ADC0_CH7 / EBI_nWRL / UART1_TXD / EBI_nCS0 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-16 M031SE3AE 多功能引脚表

M031SG6AE

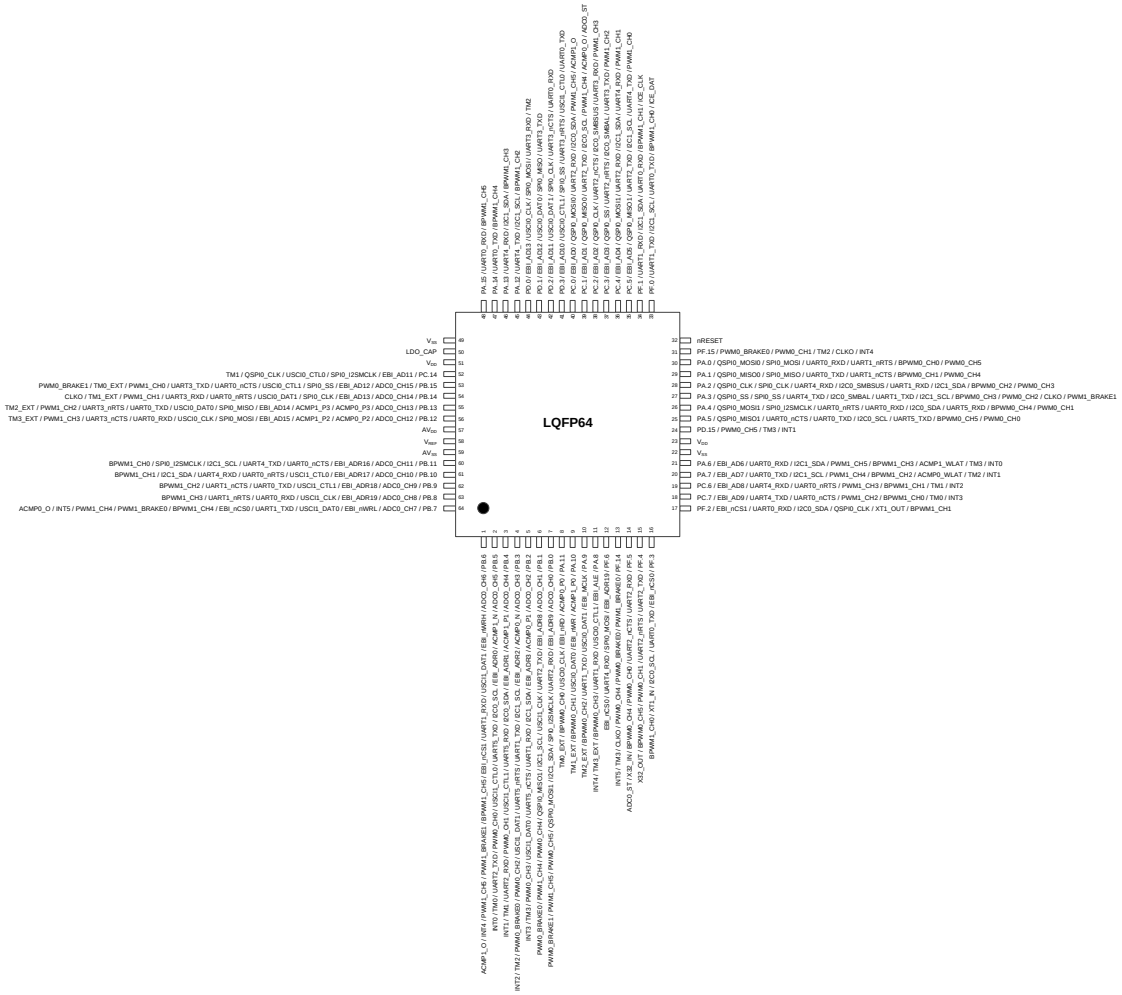


图 4.1-23 M031SG6AE 多功能引脚框图

引脚	M031SG6AE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M031SG6AE引脚功能
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK

引脚	M031SG6AE引脚功能
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
46	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
47	PA.14 / UART0_TXD / BPWM1_CH4
48	PA.15 / UART0_RXD / BPWM1_CH5
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-17 M031SG6AE 多功能引脚表

M031SG8AE

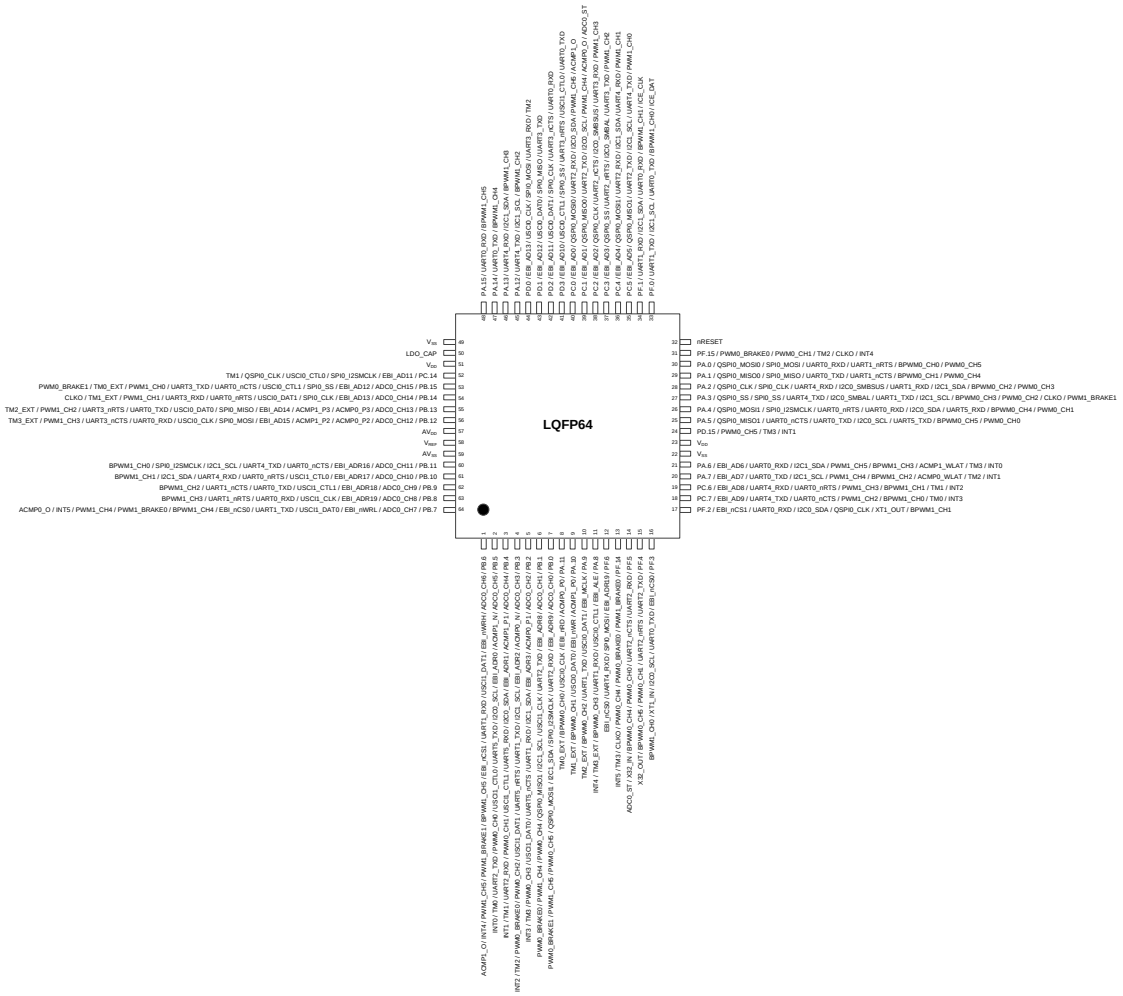


图 4.1-24 M031SG8AE 多功能引脚框图

引脚	M031SG8AE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0

引脚	M031SG8AE引脚功能
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET

引脚	M031SG8AE引脚功能
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
46	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
47	PA.14 / UART0_TXD / BPWM1_CH4
48	PA.15 / UART0_RXD / BPWM1_CH5
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-18 M031SG8AE 多功能引脚表

M031SIAAE

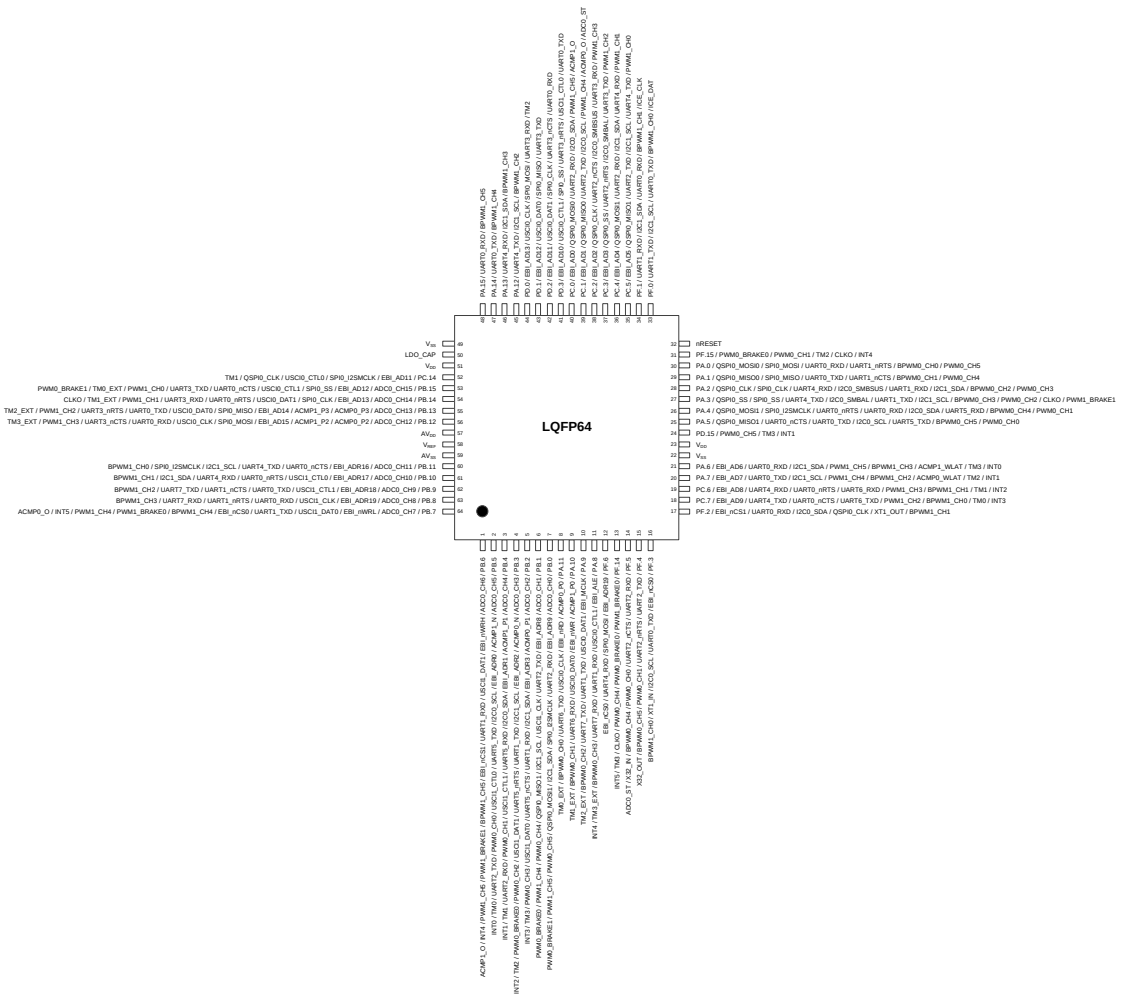


图 4.1-25 M031SIAAE 多功能引脚框图

引脚	M031SIAAE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

引脚	M031SIAAE引脚功能
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / UART6_TXD / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / UART6_RXD / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / UART7_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / UART7_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / UART6_TXD / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / UART6_RXD / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4

引脚	M031SIAAE引脚功能
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
46	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
47	PA.14 / UART0_TXD / BPWM1_CH4
48	PA.15 / UART0_RXD / BPWM1_CH5
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / UART7_TXD / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / UART7_RXD / BPWM1_CH3

引脚	M031SIAAE引脚功能
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-19 M031SIAAE 多功能引脚表

4.1.2.6 M031系列LQFP 128多功能引脚框图

对应料号:M031KG6AE, M031KG8AE, M031KIAAE

M031KG6AE

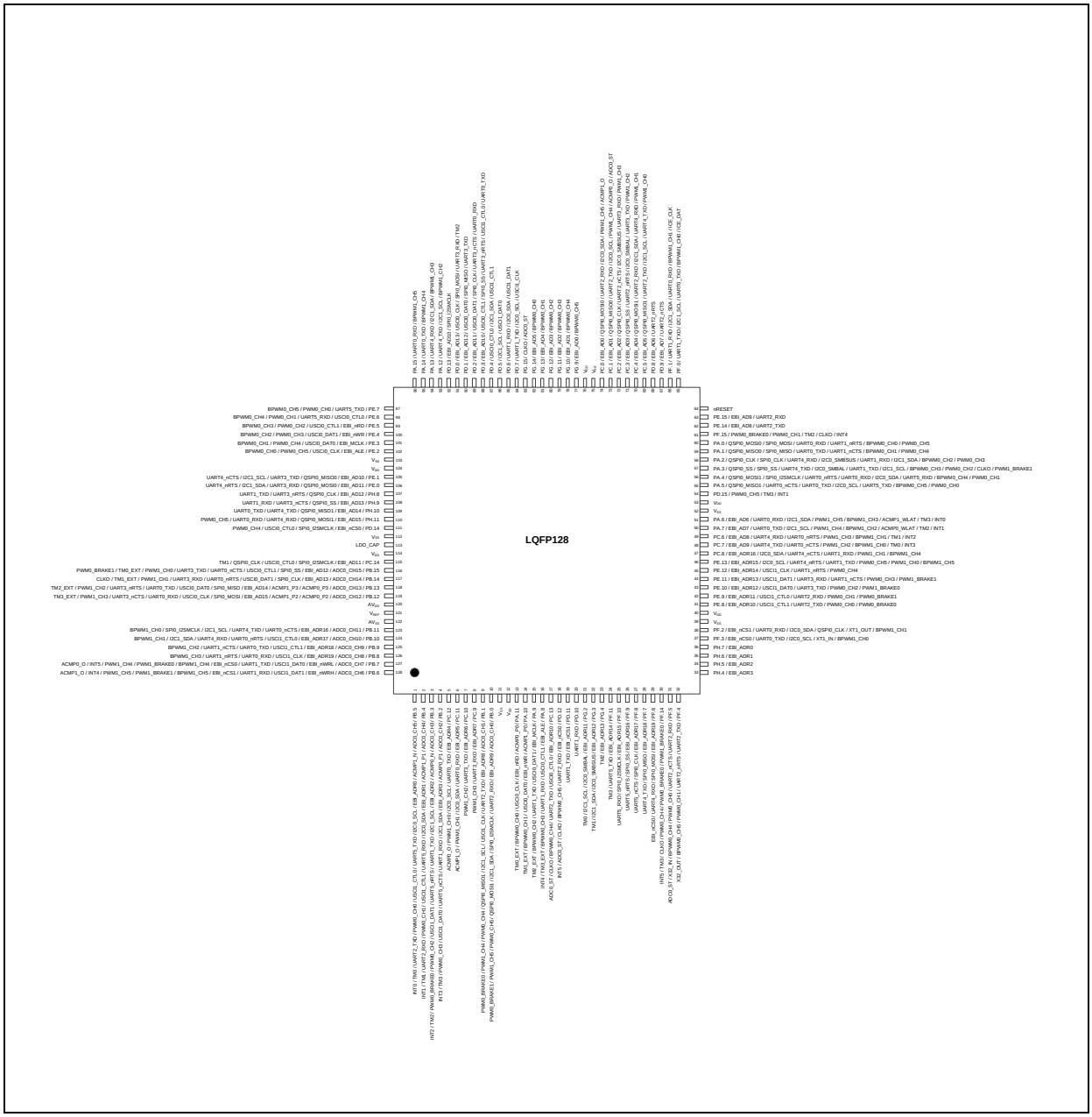


图 4.1-26 M031KG6AE 多功能引脚框图

引脚	M031KG6AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD

引脚	M031KG6AE引脚功能
	/ TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3
34	PH.5 / EBI_ADR2

引脚	M031KG6AE引脚功能
35	PH.6 / EBI_ADR1
36	PH.7 / EBI_ADR0
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD
63	PE.15 / EBI_AD9 / UART2_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS

引脚	M031KG6AE引脚功能
68	PD.8 / EBI_AD6 / UART2_nRTS
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / BPWM0_CH3
80	PG.12 / EBI_AD3 / BPWM0_CH2
81	PG.13 / EBI_AD4 / BPWM0_CH1
82	PG.14 / EBI_AD5 / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
94	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
95	PA.14 / UART0_TXD / BPWM1_CH4
96	PA.15 / UART0_RXD / BPWM1_CH5
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / PWM0_CH4 / BPWM0_CH1

引脚	M031KG6AE引脚功能
102	PE.2 / EBI_ALE / USCI0_CLK / PWM0_CH5 / BPWM0_CH0
103	VSS
104	V_{DD}
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-20 M031KG6AE 多功能引脚表

M031KG8AE

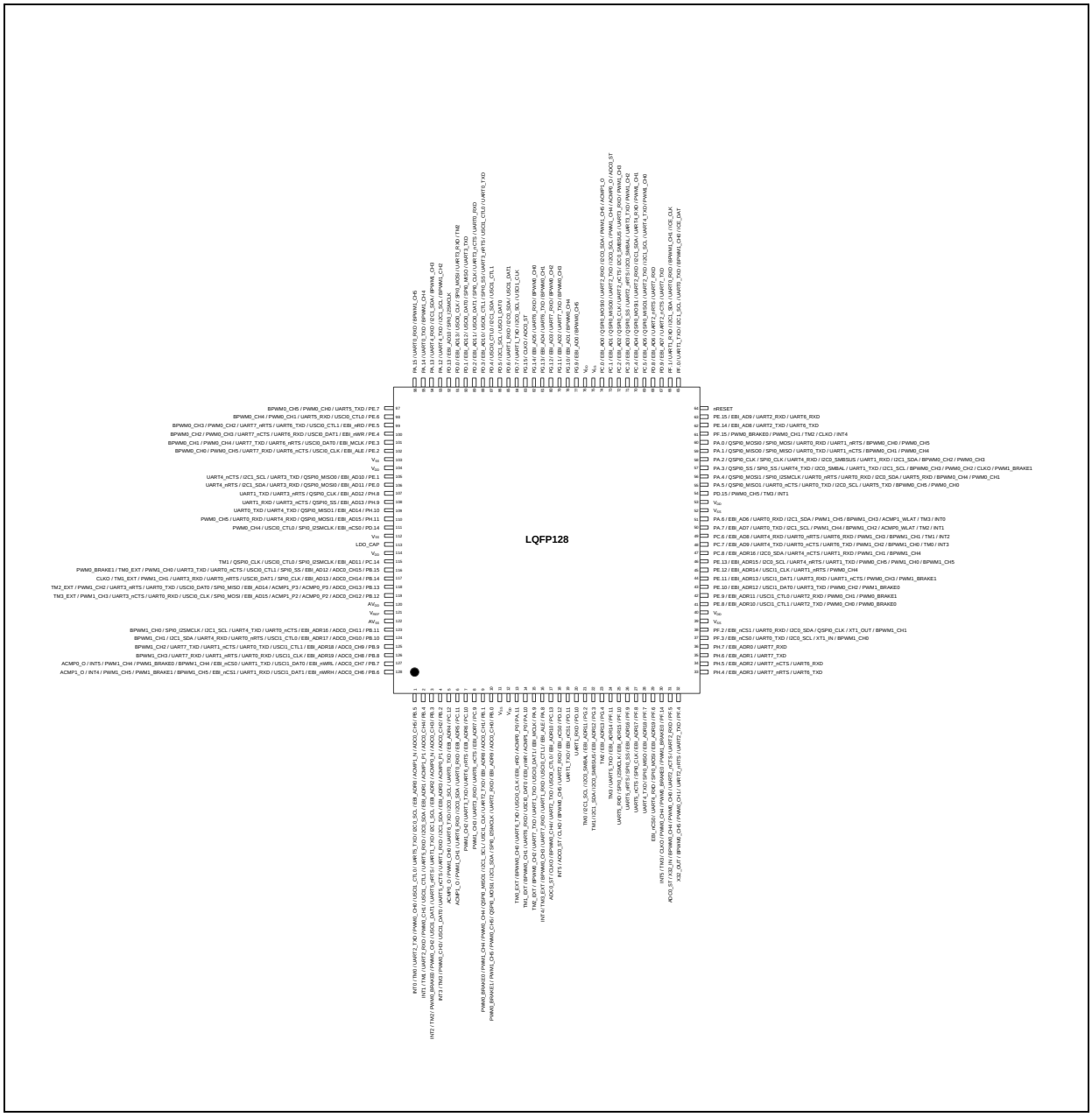


图 4.1-27 M031KG8AE 多功能引脚框图

引脚	M031KG8AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M031KG8AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3
34	PH.5 / EBI_ADR2
35	PH.6 / EBI_ADR1
36	PH.7 / EBI_ADR0

引脚	M031KG8AE引脚功能
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD
63	PE.15 / EBI_AD9 / UART2_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS
68	PD.8 / EBI_AD6 / UART2_nRTS
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0

引脚	M031KG8AE引脚功能
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / BPWM0_CH3
80	PG.12 / EBI_AD3 / BPWM0_CH2
81	PG.13 / EBI_AD4 / BPWM0_CH1
82	PG.14 / EBI_AD5 / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
94	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
95	PA.14 / UART0_TXD / BPWM1_CH4
96	PA.15 / UART0_RXD / BPWM1_CH5
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / PWM0_CH4 / BPWM0_CH1
102	PE.2 / EBI_ALE / USCI0_CLK / PWM0_CH5 / BPWM0_CH0
103	VSS

引脚	M031KG8AE引脚功能
104	V_{DD}
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-21 M031KG8AE 多功能引脚表

M031KIAAE

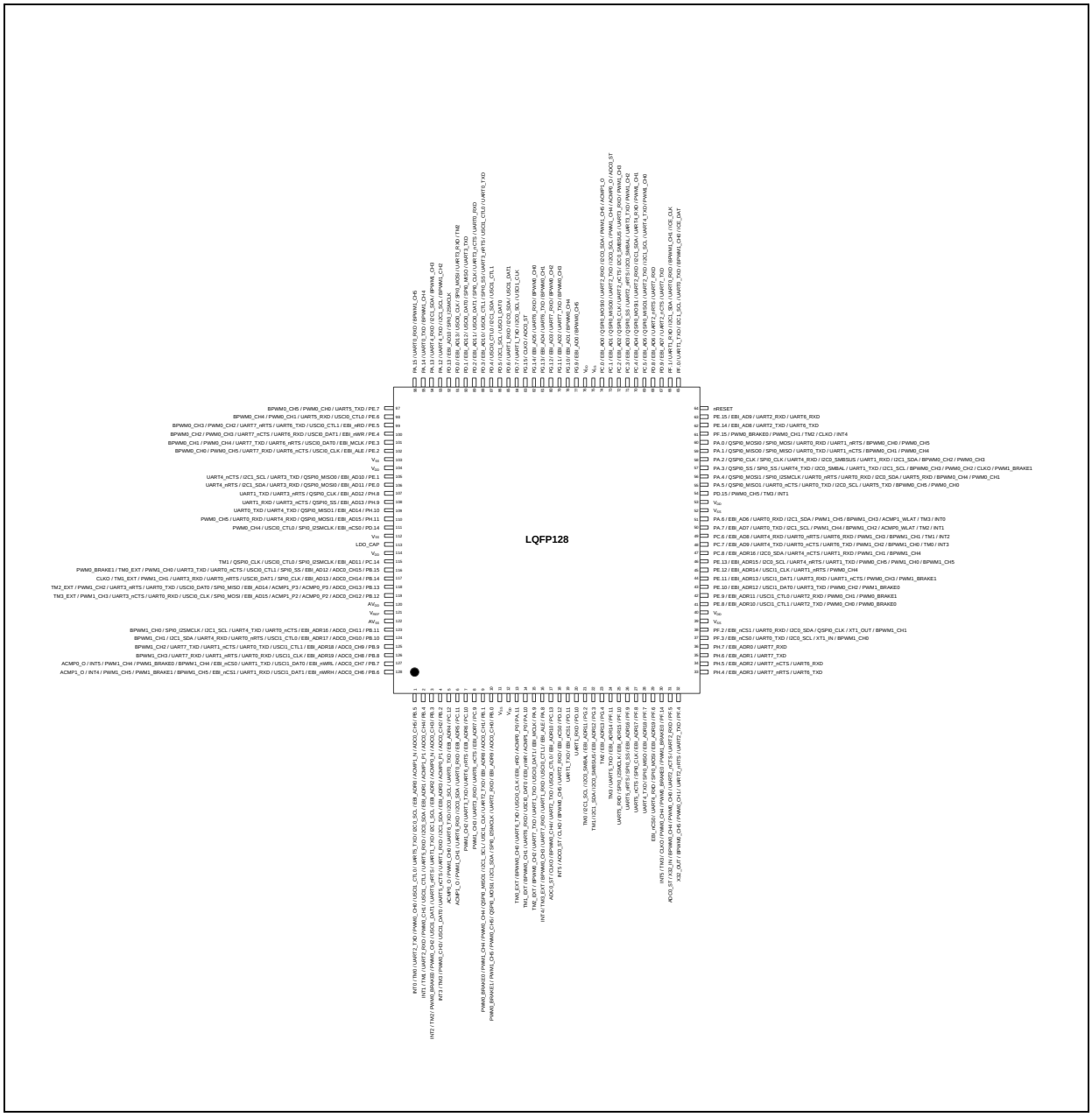


图 4.1-28 M031KIAAE 多功能引脚框图

引脚	M031KIAAE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M031KIAAE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / UART6_TXD / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / UART6_RXD / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART6_nRTS / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART6_nCTS / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / UART6_TXD / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / UART6_RXD / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / UART7_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / UART7_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3 / UART7_nRTS / UART6_TXD
34	PH.5 / EBI_ADR2 / UART7_nCTS / UART6_RXD
35	PH.6 / EBI_ADR1 / UART7_TXD
36	PH.7 / EBI_ADR0 / UART7_RXD

引脚	M031KIAAE引脚功能
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / UART6_TXD / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / UART6_RXD / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD / UART6_TXD
63	PE.15 / EBI_AD9 / UART2_RXD / UART6_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS / UART7_TXD
68	PD.8 / EBI_AD6 / UART2_nRTS / UART7_RXD
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0

引脚	M031KIAAE引脚功能
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / UART7_TXD / BPWM0_CH3
80	PG.12 / EBI_AD3 / UART7_RXD / BPWM0_CH2
81	PG.13 / EBI_AD4 / UART6_TXD / BPWM0_CH1
82	PG.14 / EBI_AD5 / UART6_RXD / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	PA.12 / UART4_TXD / I2C1_SCL / BPWM1_CH2
94	PA.13 / UART4_RXD / I2C1_SDA / BPWM1_CH3
95	PA.14 / UART0_TXD / BPWM1_CH4
96	PA.15 / UART0_RXD / BPWM1_CH5
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / UART6_TXD / UART7_nRTS / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / UART6_RXD / UART7_nCTS / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / UART6_nRTS / UART7_TXD / PWM0_CH4 / BPWM0_CH1
102	PE.2 / EBI_ALE / USCI0_CLK / UART6_nCTS / UART7_RXD / PWM0_CH5 / BPWM0_CH0
103	VSS

引脚	M031KIAAE引脚功能
104	V_{DD}
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / UART7_TXD / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / UART7_RXD / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-22 M031KIAAE 多功能引脚表

4.1.3 M032系列引脚框图

4.1.3.1 M032系列TSSOP 20引脚框图

对应料号:M032FC1AE

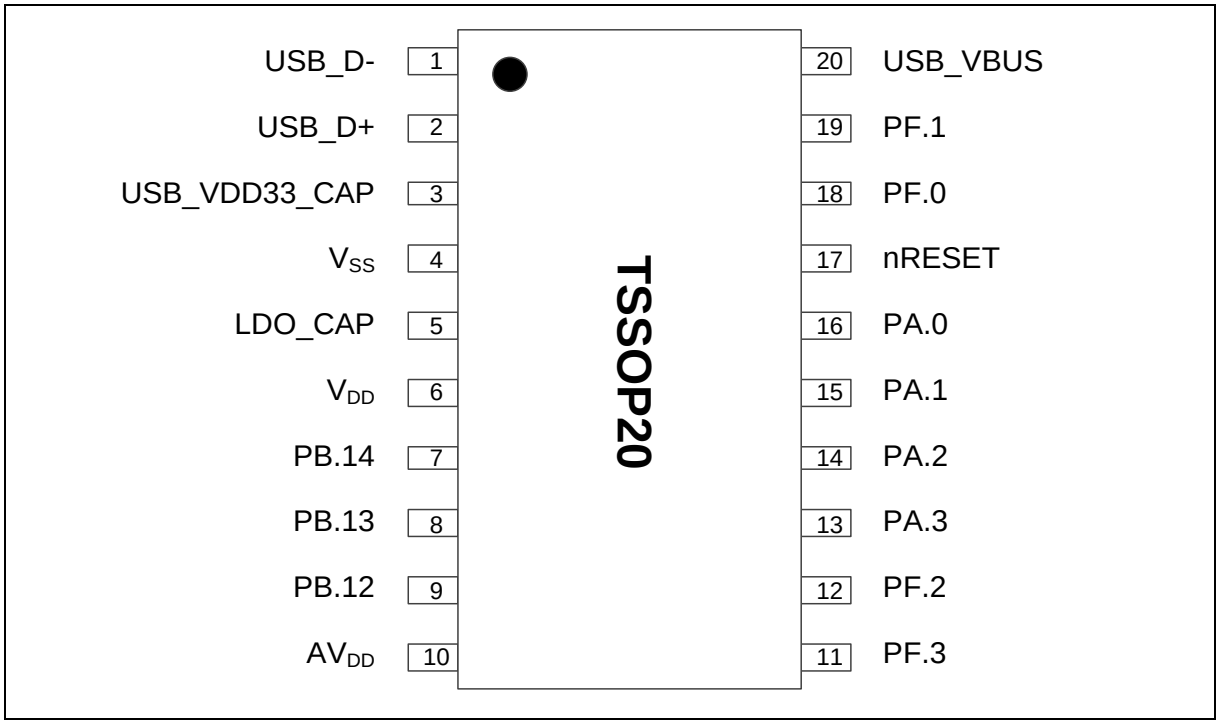


图 4.1-29 M032 系列 TSSOP 20 引脚框图

4.1.3.2 M032系列TSSOP 28引脚框图

对应料号:M032EC1AE

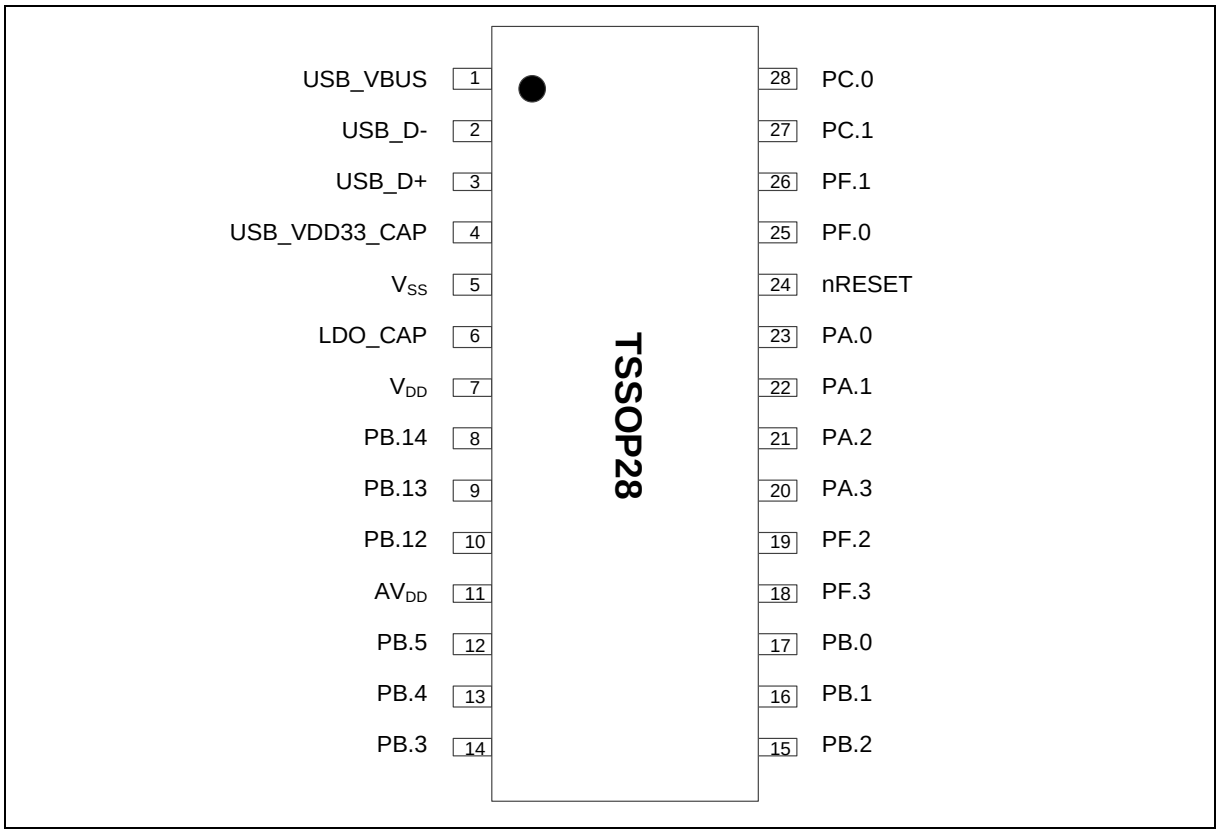


图 4.1-30 M032 系列 TSSOP 28 引脚框图

4.1.3.3 M032系列QFN 33引脚框图

对应料号:M032TC1AE, M032TD2AE

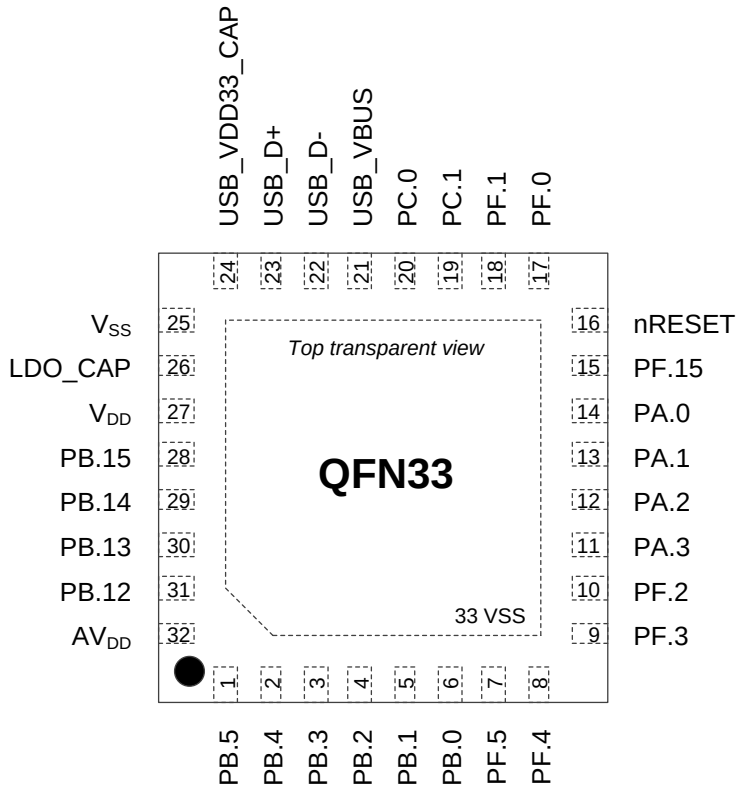


图 4.1-31 M032 系列 QFN 33 引脚框图

4.1.3.4 M032系列LQFP 48引脚框图

对应料号:M032LC2AE, M032LD2AE ,M032LE3AE, M032LG6AE, M032LG8AE

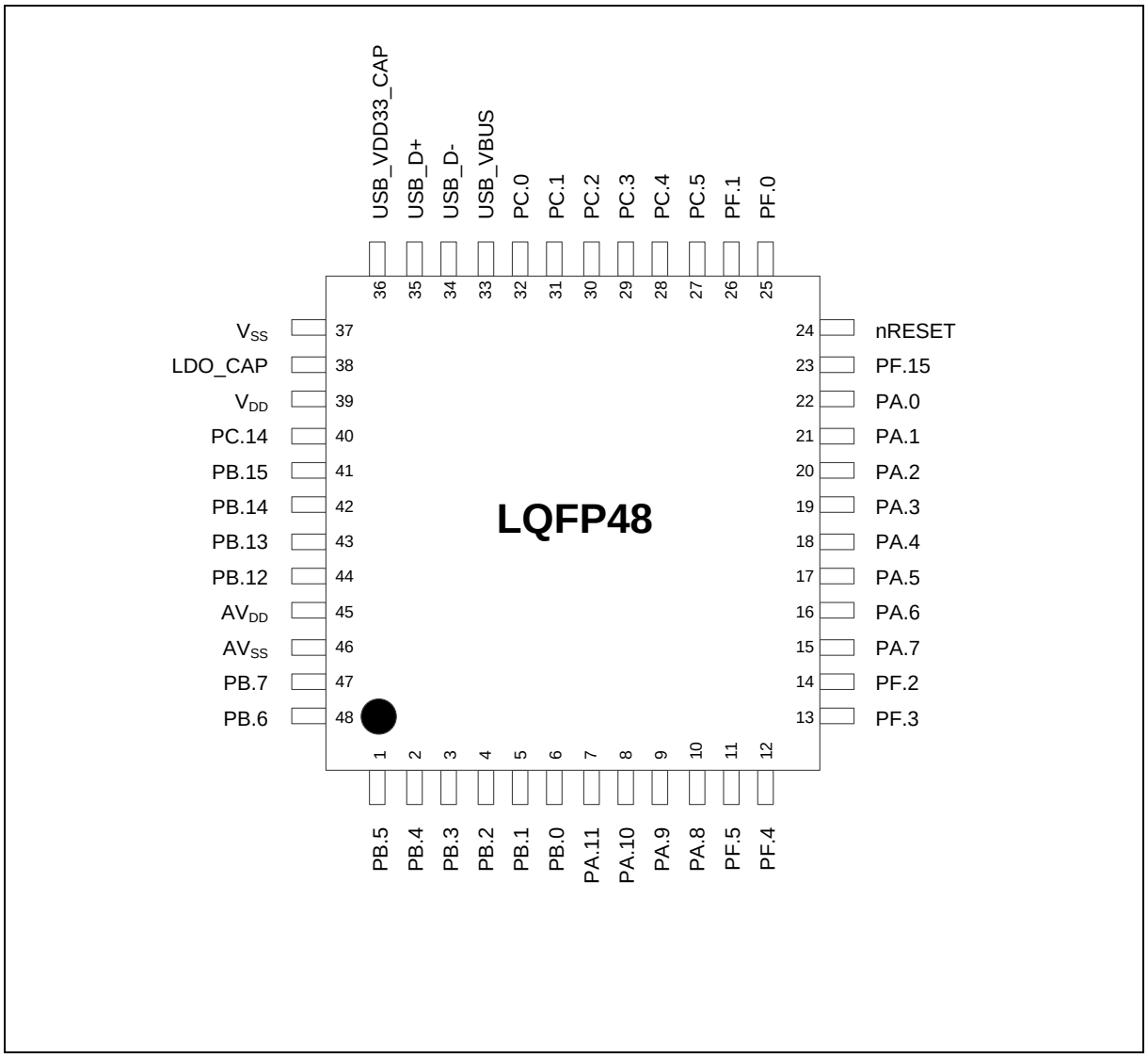


图 4.1-32 M032 系列 LQFP 48 引脚框图

4.1.3.5 M032系列LQFP 64引脚框图

对应料号:M032SE3AE, M032SG6AE, M032SG8AE, M032SIAAE

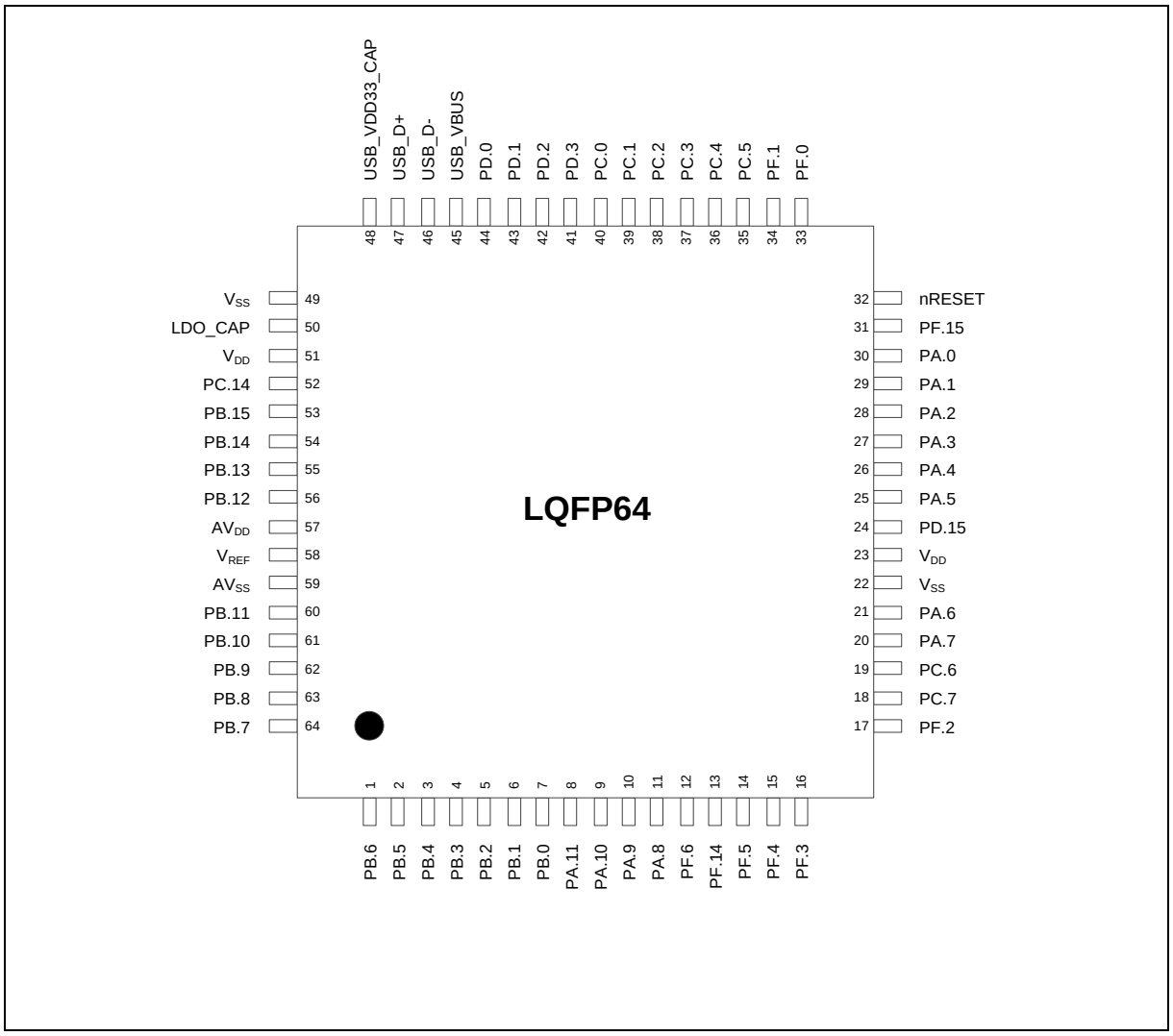


图 4.1-33 M032 系列 LQFP 64 引脚框图

4.1.3.6 M032系列LQFP 128引脚框图

对应料号:M032KG6AE, M032KG8AE, M032KIAAE

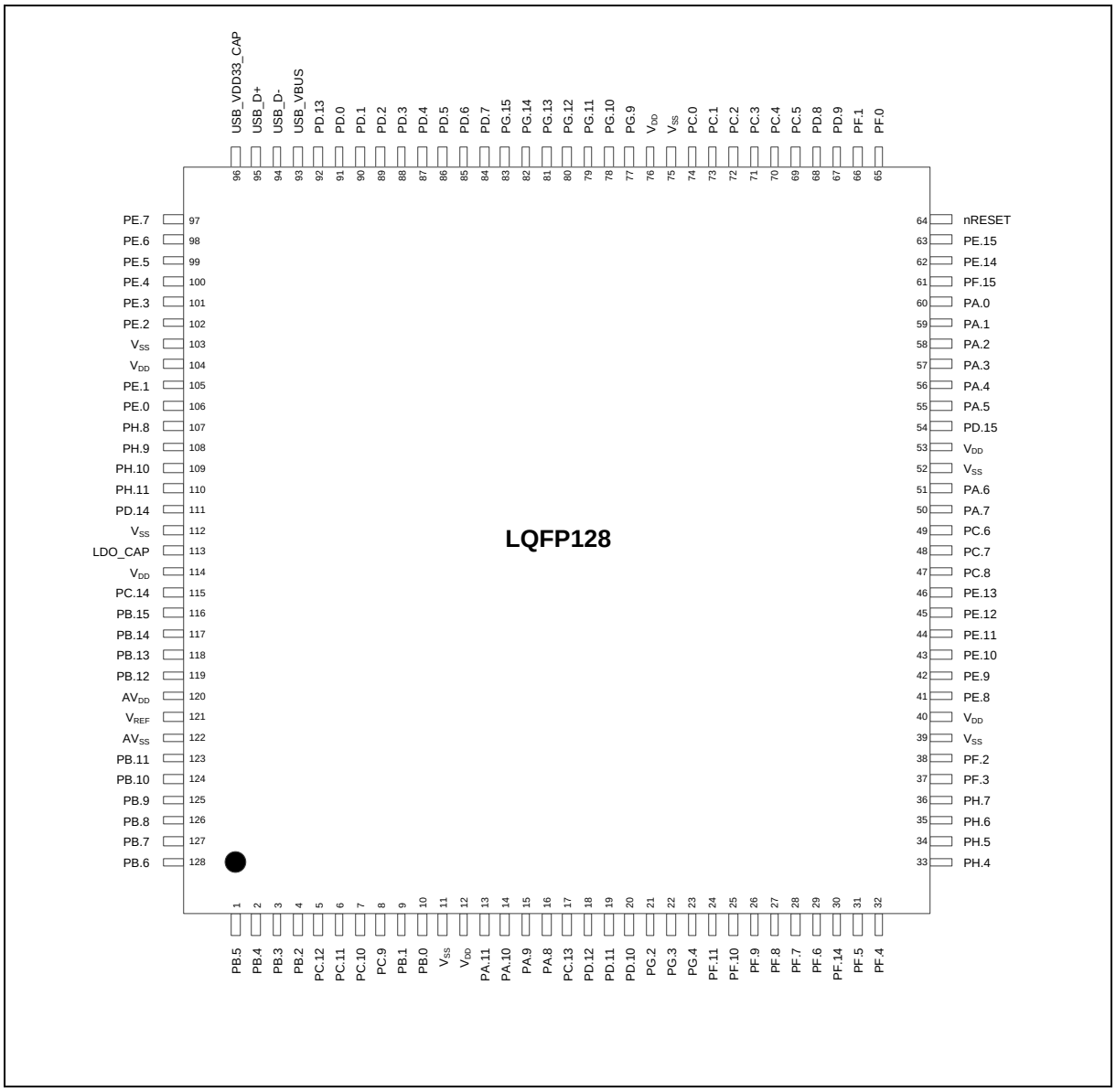


图 4.1-34 M032 系列 LQFP 128 引脚框图

4.1.4 M032系列多功能引脚框图

4.1.4.1 M032系列TSSOP 20多功能引脚框图

对应料号:M032FC1AE

M032FC1AE

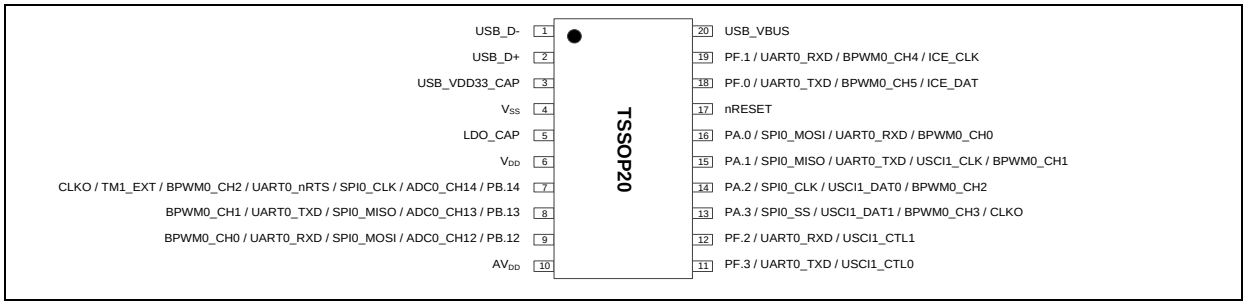


图 4.1-35 M032FC1AE 多功能引脚框图

引脚	M032FC1AE引脚功能
1	USB_D-
2	USB_D+
3	USB_V_{DD}33_CAP
4	VSS
5	LDO_CAP
6	V_{DD}
7	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / BPWM0_CH2 / TM1_EXT / CLKO
8	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / BPWM0_CH1
9	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
10	AV_{DD}
11	PF.3 / UART0_TXD / USCI1_CTL0
12	PF.2 / UART0_RXD / USCI1_CTL1
13	PA.3 / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO
14	PA.2 / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
15	PA.1 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
16	PA.0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
17	nRESET
18	PF.0 / UART0_TXD / BPWM0_CH5 / ICE_DAT
19	PF.1 / UART0_RXD / BPWM0_CH4 / ICE_CLK
20	USB_VBUS

表 4.1-23 M032FC1AE 多功能引脚表

4.1.4.2 M032系列TSSOP 28多功能引脚框图

对应料号:M032EC1AE

M032EC1AE

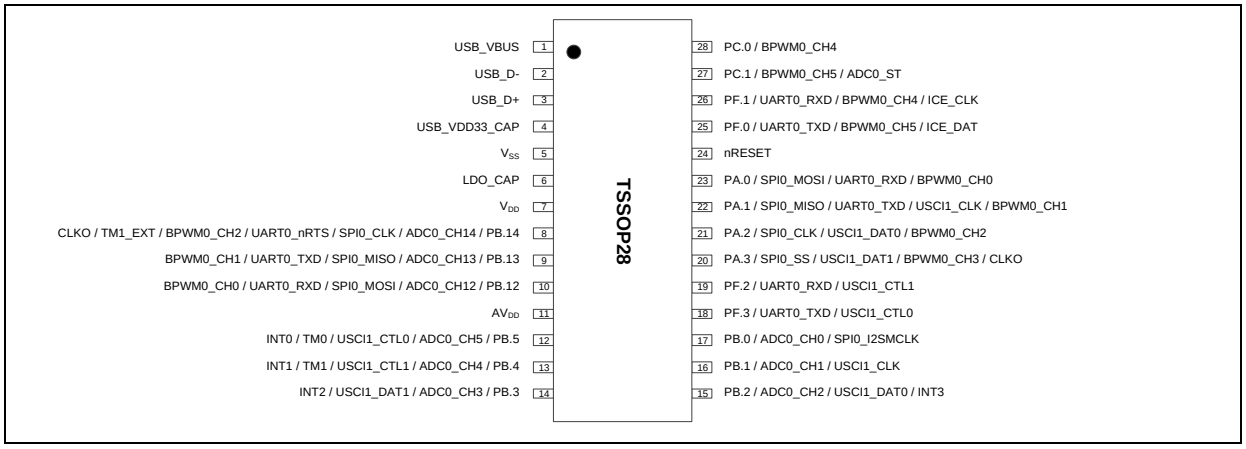


图 4.1-36 M032EC1AE 多功能引脚框图

引脚	M032EC1AE引脚功能
1	USB_VBUS
2	USB_D-
3	USB_D+
4	USB_V_{DD}33_CAP
5	VSS
6	LDO_CAP
7	V_{DD}
8	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / BPWM0_CH2 / TM1_EXT / CLKO
9	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / BPWM0_CH1
10	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
11	AV_{DD}
12	PB.5 / ADC0_CH5 / USCI1_CTL0 / TM0 / INT0
13	PB.4 / ADC0_CH4 / USCI1_CTL1 / TM1 / INT1
14	PB.3 / ADC0_CH3 / USCI1_DAT1 / INT2
15	PB.2 / ADC0_CH2 / USCI1_DAT0 / INT3
16	PB.1 / ADC0_CH1 / USCI1_CLK
17	PB.0 / ADC0_CH0 / SPI0_I2SMCLK
18	PF.3 / UART0_TXD / USCI1_CTL0
19	PF.2 / UART0_RXD / USCI1_CTL1
20	PA.3 / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO

引脚	M032EC1AE引脚功能
21	PA.2 / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
22	PA.1 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
23	PA.0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
24	nRESET
25	PF.0 / UART0_TXD / BPWM0_CH5 / ICE_DAT
26	PF.1 / UART0_RXD / BPWM0_CH4 / ICE_CLK
27	PC.1 / BPWM0_CH5 / ADC0_ST
28	PC.0 / BPWM0_CH4

表 4.1-24 M032EC1AE 多功能引脚表

4.1.4.3 M032系列QFN 33多功能引脚框图

对应料号:M032TC1AE, M032TD2AE

M032TC1AE

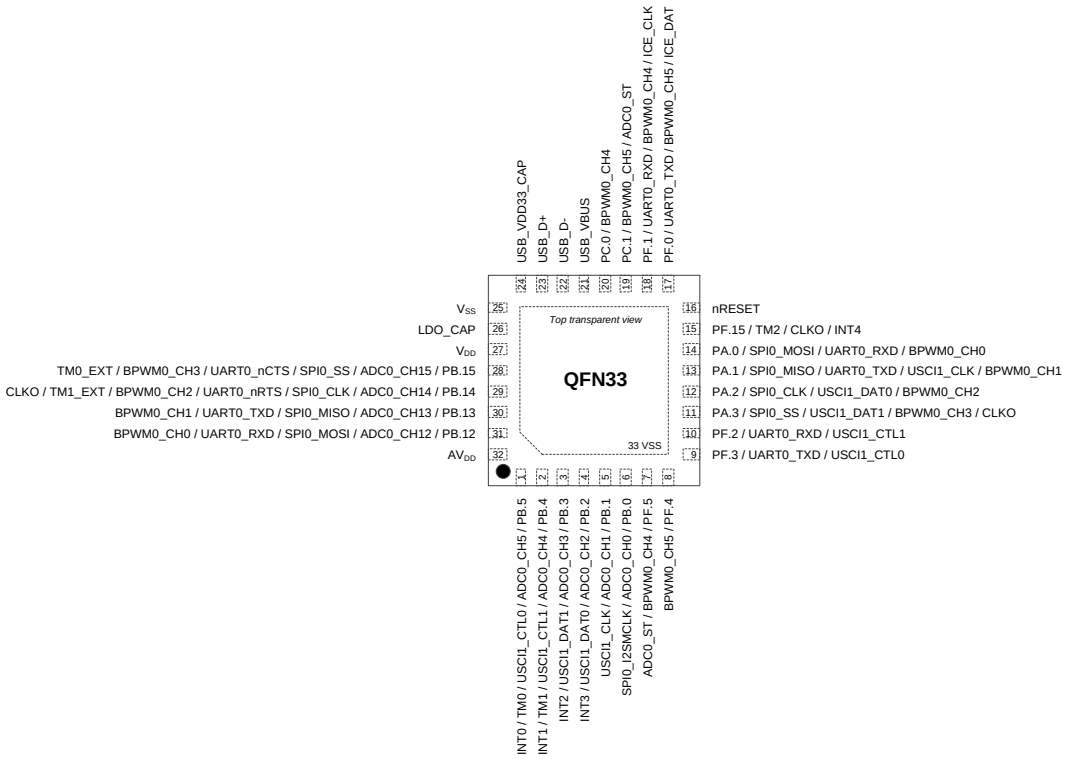


图 4.1-37 M032TC1AE 多功能引脚框图

引脚	M032TC1AE引脚功能
1	PB.5 / ADC0_CH5 / USCI1_CTL0 / TM0 / INT0
2	PB.4 / ADC0_CH4 / USCI1_CTL1 / TM1 / INT1
3	PB.3 / ADC0_CH3 / USCI1_DAT1 / INT2

引脚	M032TC1AE引脚功能
4	PB.2 / ADC0_CH2 / USCI1_DAT0 / INT3
5	PB.1 / ADC0_CH1 / USCI1_CLK
6	PB.0 / ADC0_CH0 / SPI0_I2SMCLK
7	PF.5 / BPWM0_CH4 / ADC0_ST
8	PF.4 / BPWM0_CH5
9	PF.3 / UART0_TXD / USCI1_CTL0
10	PF.2 / UART0_RXD / USCI1_CTL1
11	PA.3 / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO
12	PA.2 / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
13	PA.1 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
14	PA.0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
15	PF.15 / CLKO / INT4
16	nRESET
17	PF.0 / UART0_TXD / BPWM0_CH5 / ICE_DAT
18	PF.1 / UART0_RXD / BPWM0_CH4 / ICE_CLK
19	PC.1 / BPWM0_CH5 / ADC0_ST
20	PC.0 / BPWM0_CH4
21	USB_VBUS
22	USB_D-
23	USB_D+
24	USB_V_{DD}33_CAP
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / UART0_nCTS / BPWM0_CH3 / TM0_EXT
29	PB.14 / ADC0_CH14 / SPI0_CLK / UART0_nRTS / BPWM0_CH2 / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / SPI0_MISO / UART0_TXD / BPWM0_CH1
31	PB.12 / ADC0_CH12 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
32	AV_{DD}

表 4.1-25 M032TC1AE 多功能引脚表

M032TD2AE

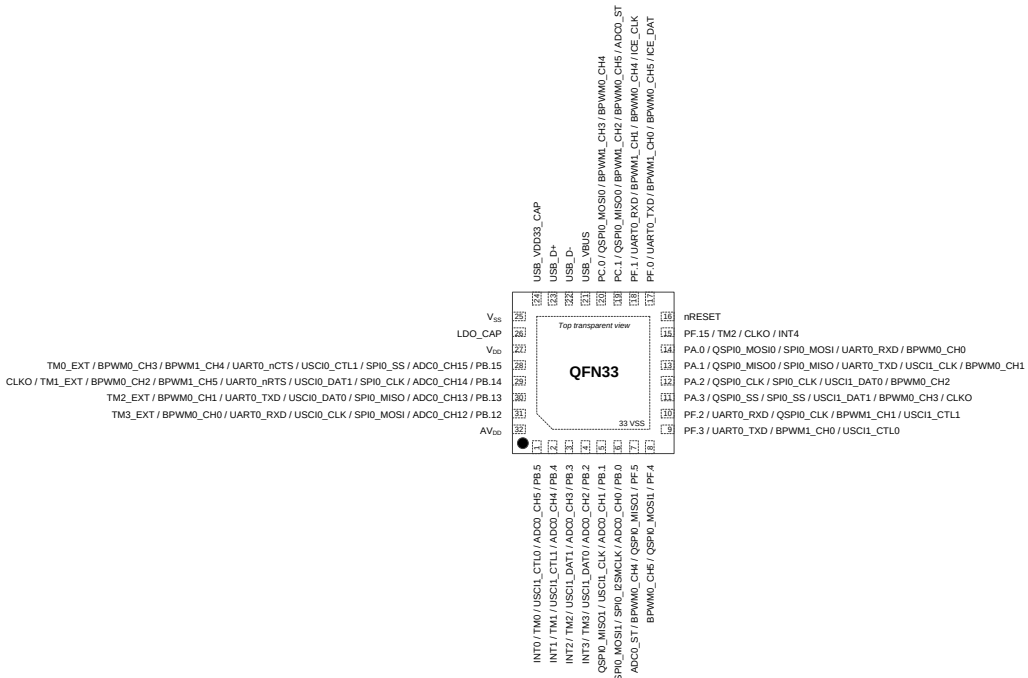


图 4.1-38 M032TD2AE 多功能引脚框图

引脚	M032TD2AE引脚功能
1	PB.5 / ADC0_CH5 / USCI1_CTL0 / TM0 / INT0
2	PB.4 / ADC0_CH4 / USCI1_CTL1 / TM1 / INT1
3	PB.3 / ADC0_CH3 / USCI1_DAT1 / TM2 / INT2
4	PB.2 / ADC0_CH2 / USCI1_DAT0 / TM3 / INT3
5	PB.1 / ADC0_CH1 / USCI1_CLK / QSPI0_MISO1

引脚	M032TD2AE引脚功能
6	PB.0 / ADC0_CH0 / SPI0_I2SMCLK / QSPI0_MOSI1
7	PF.5 / QSPI0_MISO1 / BPWM0_CH4 / ADC0_ST
8	PF.4 / QSPI0_MOSI1 / BPWM0_CH5
9	PF.3 / UART0_TXD / BPWM1_CH0 / USCI1_CTL0
10	PF.2 / UART0_RXD / QSPI0_CLK / BPWM1_CH1 / USCI1_CTL1
11	PA.3 / QSPI0_SS / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO
12	PA.2 / QSPI0_CLK / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
13	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
14	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
15	PF.15 / TM2 / CLKO / INT4
16	nRESET
17	PF.0 / UART0_TXD / BPWM1_CH0 / BPWM0_CH5 / ICE_DAT
18	PF.1 / UART0_RXD / BPWM1_CH1 / BPWM0_CH4 / ICE_CLK
19	PC.1 / QSPI0_MISO0 / BPWM1_CH2 / BPWM0_CH5 / ADC0_ST
20	PC.0 / QSPI0_MOSI0 / BPWM1_CH3 / BPWM0_CH4
21	USB_VBUS
22	USB_D-
23	USB_D+
24	USB_V_{DD}33_CAP
25	VSS
26	LDO_CAP
27	V_{DD}
28	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / BPWM1_CH4 / BPWM0_CH3 / TM0_EXT
29	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / BPWM1_CH5 / BPWM0_CH2 / TM1_EXT / CLKO
30	PB.13 / ADC0_CH13 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / BPWM0_CH1 / TM2_EXT
31	PB.12 / ADC0_CH12 / SPI0_MOSI / USCI0_CLK / UART0_RXD / BPWM0_CH0 / TM3_EXT
32	AV_{DD}

表 4.1-26 M032TD2AE 多功能引脚表

4.1.4.4 M032系列LQFP 48多功能引脚框图

对应料号:M032LC2AE, M032LD2AE, M032LE3AE, M032LG6AE, M032LG8AE

M032LC2AE

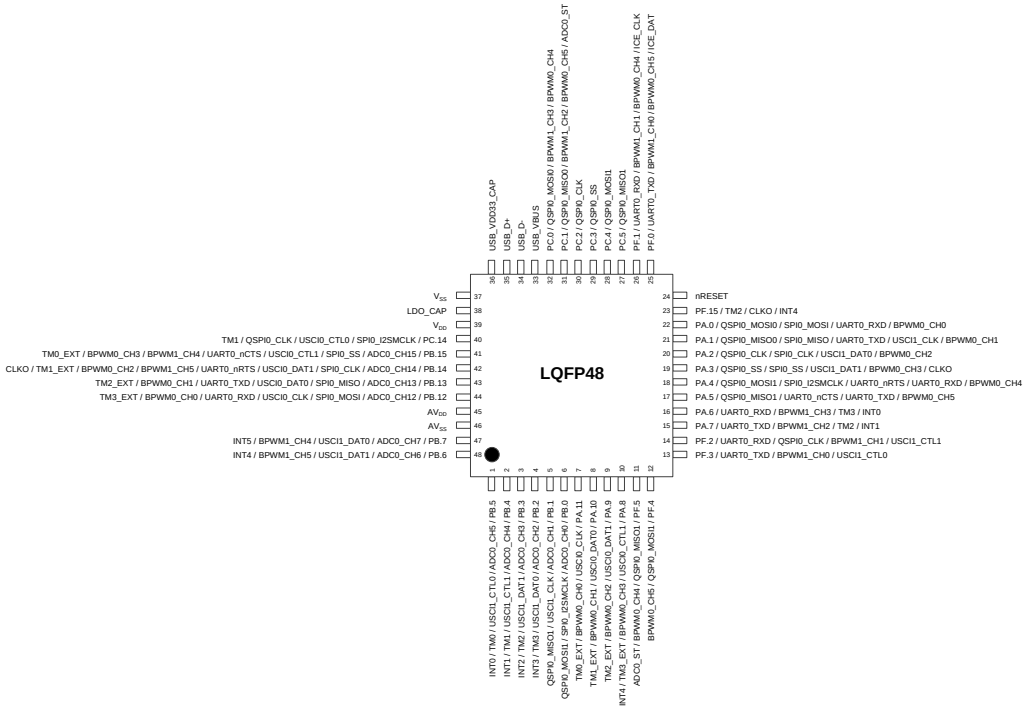


图 4.1-39 M032LC2AE 多功能引脚框图

引脚	M032LC2AE引脚功能
1	PB.5 / ADC0_CH5 / USCI1_CTL0 / TM0 / INT0
2	PB.4 / ADC0_CH4 / USCI1_CTL1 / TM1 / INT1
3	PB.3 / ADC0_CH3 / USCI1_DAT1 / TM2 / INT2

引脚	M032LC2AE引脚功能
4	PB.2 / ADC0_CH2 / USCI1_DAT0 / TM3 / INT3
5	PB.1 / ADC0_CH1 / USCI1_CLK / QSPI0_MISO1
6	PB.0 / ADC0_CH0 / SPI0_I2SMCLK / QSPI0_MOSI1
7	PA.11 / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / USCI0_DAT1 / BPWM0_CH2 / TM2_EXT
10	PA.8 / USCI0_CTL1 / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / QSPI0_MISO1 / BPWM0_CH4 / ADC0_ST
12	PF.4 / QSPI0_MOSI1 / BPWM0_CH5
13	PF.3 / UART0_TXD / BPWM1_CH0 / USCI1_CTL0
14	PF.2 / UART0_RXD / QSPI0_CLK / BPWM1_CH1 / USCI1_CTL1
15	PA.7 / UART0_TXD / BPWM1_CH2 / TM2 / INT1
16	PA.6 / UART0_RXD / BPWM1_CH3 / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / BPWM0_CH5
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / BPWM0_CH4
19	PA.3 / QSPI0_SS / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO
20	PA.2 / QSPI0_CLK / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
23	PF.15 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART0_TXD / BPWM1_CH0 / BPWM0_CH5 / ICE_DAT
26	PF.1 / UART0_RXD / BPWM1_CH1 / BPWM0_CH4 / ICE_CLK
27	PC.5 / QSPI0_MISO1
28	PC.4 / QSPI0_MOSI1
29	PC.3 / QSPI0_SS
30	PC.2 / QSPI0_CLK
31	PC.1 / QSPI0_MISO0 / BPWM1_CH2 / BPWM0_CH5 / ADC0_ST
32	PC.0 / QSPI0_MOSI0 / BPWM1_CH3 / BPWM0_CH4
33	USB_VBUS
34	USB_D-
35	USB_D+
36	USB_V_{DD}33_CAP
37	VSS

引脚	M032LC2AE引脚功能
38	LDO_CAP
39	V_{DD}
40	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / BPWM1_CH4 / BPWM0_CH3 / TM0_EXT
42	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / BPWM1_CH5 / BPWM0_CH2 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / BPWM0_CH1 / TM2_EXT
44	PB.12 / ADC0_CH12 / SPI0_MOSI / USCI0_CLK / UART0_RXD / BPWM0_CH0 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / USCI1_DAT0 / BPWM1_CH4 / INT5
48	PB.6 / ADC0_CH6 / USCI1_DAT1 / BPWM1_CH5 / INT4

表 4.1-27 M032LC2AE 多功能引脚表

M032LD2AE

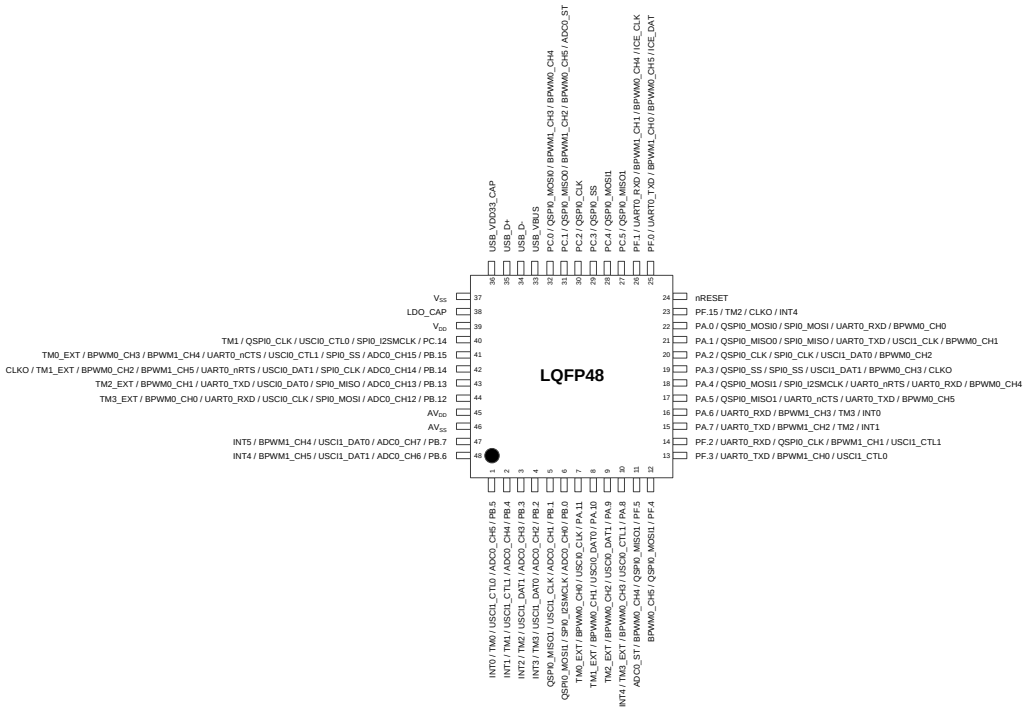


图 4.1-40 M032LD2AE 多功能引脚框图

引脚	M032LD2AE引脚功能
1	PB.5 / ADC0_CH5 / USCI1_CTL0 / TM0 / INT0
2	PB.4 / ADC0_CH4 / USCI1_CTL1 / TM1 / INT1
3	PB.3 / ADC0_CH3 / USCI1_DAT1 / TM2 / INT2
4	PB.2 / ADC0_CH2 / USCI1_DAT0 / TM3 / INT3
5	PB.1 / ADC0_CH1 / USCI1_CLK / QSPI0_MISO1

引脚	M032LD2AE引脚功能
6	PB.0 / ADC0_CH0 / SPI0_I2SMCLK / QSPI0_MOSI1
7	PA.11 / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / USCI0_DAT1 / BPWM0_CH2 / TM2_EXT
10	PA.8 / USCI0_CTL1 / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / QSPI0_MISO1 / BPWM0_CH4 / ADC0_ST
12	PF.4 / QSPI0_MOSI1 / BPWM0_CH5
13	PF.3 / UART0_TXD / BPWM1_CH0 / USCI1_CTL0
14	PF.2 / UART0_RXD / QSPI0_CLK / BPWM1_CH1 / USCI1_CTL1
15	PA.7 / UART0_TXD / BPWM1_CH2 / TM2 / INT1
16	PA.6 / UART0_RXD / BPWM1_CH3 / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / BPWM0_CH5
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / BPWM0_CH4
19	PA.3 / QSPI0_SS / SPI0_SS / USCI1_DAT1 / BPWM0_CH3 / CLKO
20	PA.2 / QSPI0_CLK / SPI0_CLK / USCI1_DAT0 / BPWM0_CH2
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / USCI1_CLK / BPWM0_CH1
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / BPWM0_CH0
23	PF.15 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART0_TXD / BPWM1_CH0 / BPWM0_CH5 / ICE_DAT
26	PF.1 / UART0_RXD / BPWM1_CH1 / BPWM0_CH4 / ICE_CLK
27	PC.5 / QSPI0_MISO1
28	PC.4 / QSPI0_MOSI1
29	PC.3 / QSPI0_SS
30	PC.2 / QSPI0_CLK
31	PC.1 / QSPI0_MISO0 / BPWM1_CH2 / BPWM0_CH5 / ADC0_ST
32	PC.0 / QSPI0_MOSI0 / BPWM1_CH3 / BPWM0_CH4
33	USB_VBUS
34	USB_D-
35	USB_D+
36	USB_V_{DD}33_CAP
37	VSS
38	LDO_CAP
39	V_{DD}

引脚	M032LD2AE引脚功能
40	PC.14 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / BPWM1_CH4 / BPWM0_CH3 / TM0_EXT
42	PB.14 / ADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / BPWM1_CH5 / BPWM0_CH2 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / BPWM0_CH1 / TM2_EXT
44	PB.12 / ADC0_CH12 / SPI0_MOSI / USCI0_CLK / UART0_RXD / BPWM0_CH0 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / USCI1_DAT0 / BPWM1_CH4 / INT5
48	PB.6 / ADC0_CH6 / USCI1_DAT1 / BPWM1_CH5 / INT4

表 4.1-28 M032LD2AE 多功能引脚表

M032LE3AE

图 4.1-41 M032LE3AE 多功能引脚框图

引脚	M032LE3AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0

引脚	M032LE3AE引脚功能
6	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / XT1_OUT
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
18	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
19	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
22	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.5 / EBI_AD5 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	PC.4 / EBI_AD4 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	PC.3 / EBI_AD3 / UART2_nRTS / PWM1_CH2
30	PC.2 / EBI_AD2 / UART2_nCTS / PWM1_CH3
31	PC.1 / EBI_AD1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
32	PC.0 / EBI_AD0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	USB_VBUS
34	USB_D-
35	USB_D+
36	USB_V_{DD}33_CAP
37	VSS
38	LDO_CAP
39	V_{DD}

引脚	M032LE3AE引脚功能
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / UART1_TXD / EBI_nCS0 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / UART1_RXD / EBI_nCS1 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-29 M032LE3AE 多功能引脚表

M032LG6AE

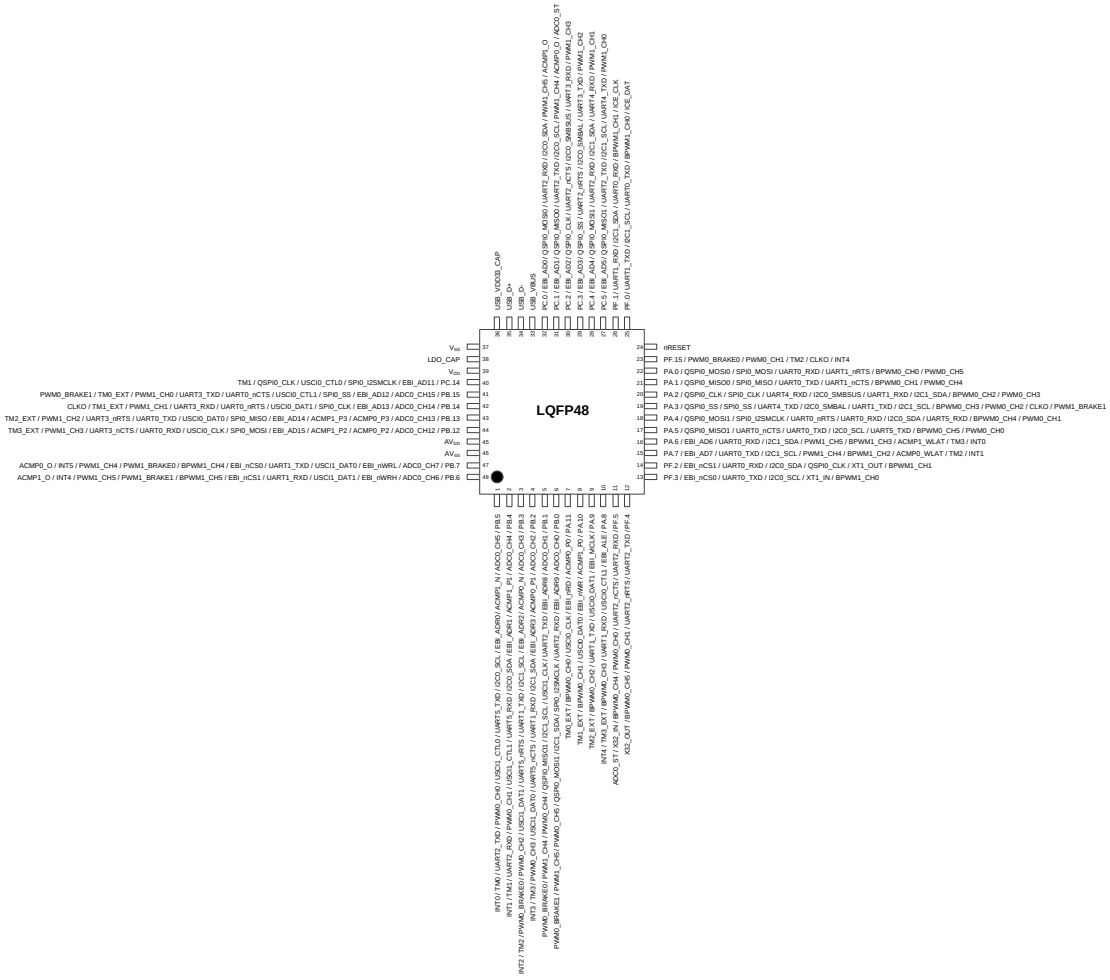


图 4.1-42 M032LG6AE 多功能引脚框图

引脚	M032LG6AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M032LG6AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
19	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
27	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
28	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
29	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
30	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
31	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
32	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	USB_VBUS
34	USB_D-

引脚	M032LG6AE引脚功能
35	USB_D+
36	USB_V_{DD}33_CAP
37	VSS
38	LDO_CAP
39	V_{DD}
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-30 M032LG6AE 多功能引脚表

M032LG8AE

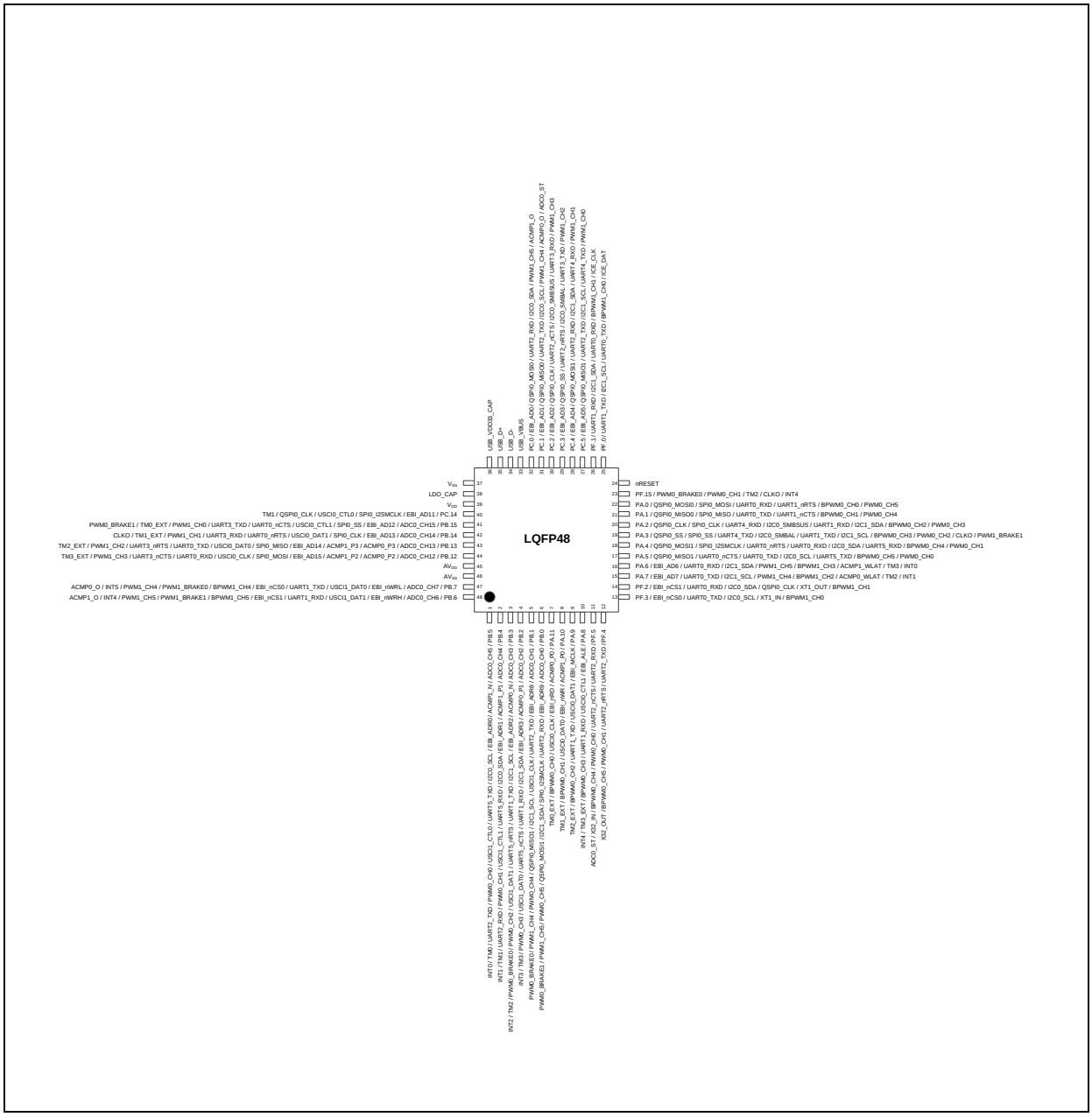


图 4.1-43 M032LG8AE 多功能引脚框图

引脚	M032LG8AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M032LG8AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
6	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
7	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
10	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
13	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
14	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
15	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
16	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
17	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
18	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
19	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
20	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
21	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
22	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
23	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
27	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
28	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
29	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
30	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
31	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
32	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
33	USB_VBUS
34	USB_D-

引脚	M032LG8AE引脚功能
35	USB_D+
36	USB_V_{DD}33_CAP
37	VSS
38	LDO_CAP
39	V_{DD}
40	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
41	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
42	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
43	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
44	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
45	AV_{DD}
46	AVSS
47	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
48	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-31 M032LG8AE 多功能引脚表

4.1.4.5 M032系列LQFP 64多功能引脚框图

对应料号:M032SE3AE, M032SG6AE, M032SG8AE, M032SIAAE

M032SE3AE

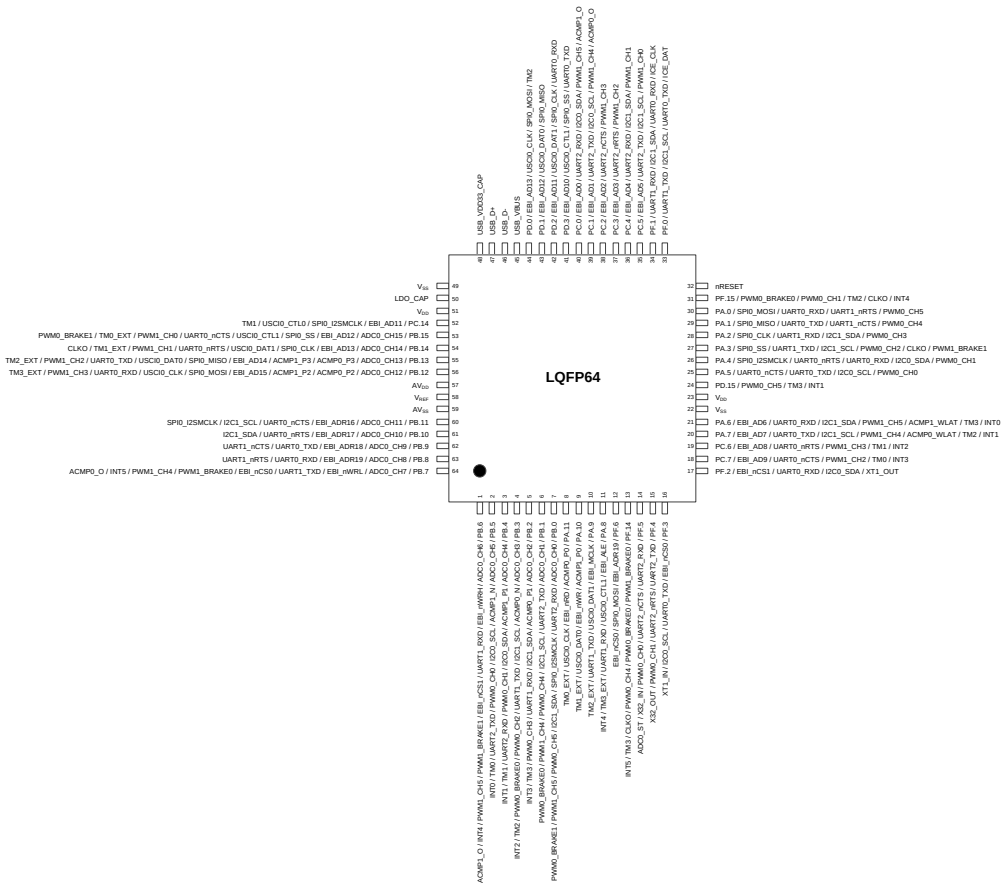


图 4.1-44 M032SE3AE 多功能引脚框图

引脚	M032SE3AE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / UART1_RXD / EBI_nCS1 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / I2C0_SCL / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M032SE3AE引脚功能
4	PB.3 / ADC0_CH3 / ACMP0_N / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / I2C1_SDA / UART1_RXD / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / XT1_OUT
18	PC.7 / EBI_AD9 / UART0_nCTS / PWM1_CH2 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART0_nRTS / PWM1_CH3 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / UART0_nCTS / UART0_TXD / I2C0_SCL / PWM0_CH0
26	PA.4 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / PWM0_CH1
27	PA.3 / SPI0_SS / UART1_TXD / I2C1_SCL / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / SPI0_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / PWM0_CH4
30	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	PC.5 / EBI_AD5 / UART2_TXD / I2C1_SCL / PWM1_CH0
36	PC.4 / EBI_AD4 / UART2_RXD / I2C1_SDA / PWM1_CH1
37	PC.3 / EBI_AD3 / UART2_nRTS / PWM1_CH2

引脚	M032SE3AE引脚功能
38	PC.2 / EBI_AD2 / UART2_nCTS / PWM1_CH3
39	PC.1 / EBI_AD1 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O
40	PC.0 / EBI_AD0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / TM2
45	USB_VBUS
46	USB_D-
47	USB_D+
48	USB_V_{DD}33_CAP
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / I2C1_SCL / SPI0_I2SMCLK
61	PB.10 / ADC0_CH10 / EBI_ADR17 / UART0_nRTS / I2C1_SDA
62	PB.9 / ADC0_CH9 / EBI_ADR18 / UART0_TXD / UART1_nCTS
63	PB.8 / ADC0_CH8 / EBI_ADR19 / UART0_RXD / UART1_nRTS
64	PB.7 / ADC0_CH7 / EBI_nWRL / UART1_TXD / EBI_nCS0 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-32 M032SE2AE 多功能引脚表

M032SG6AE

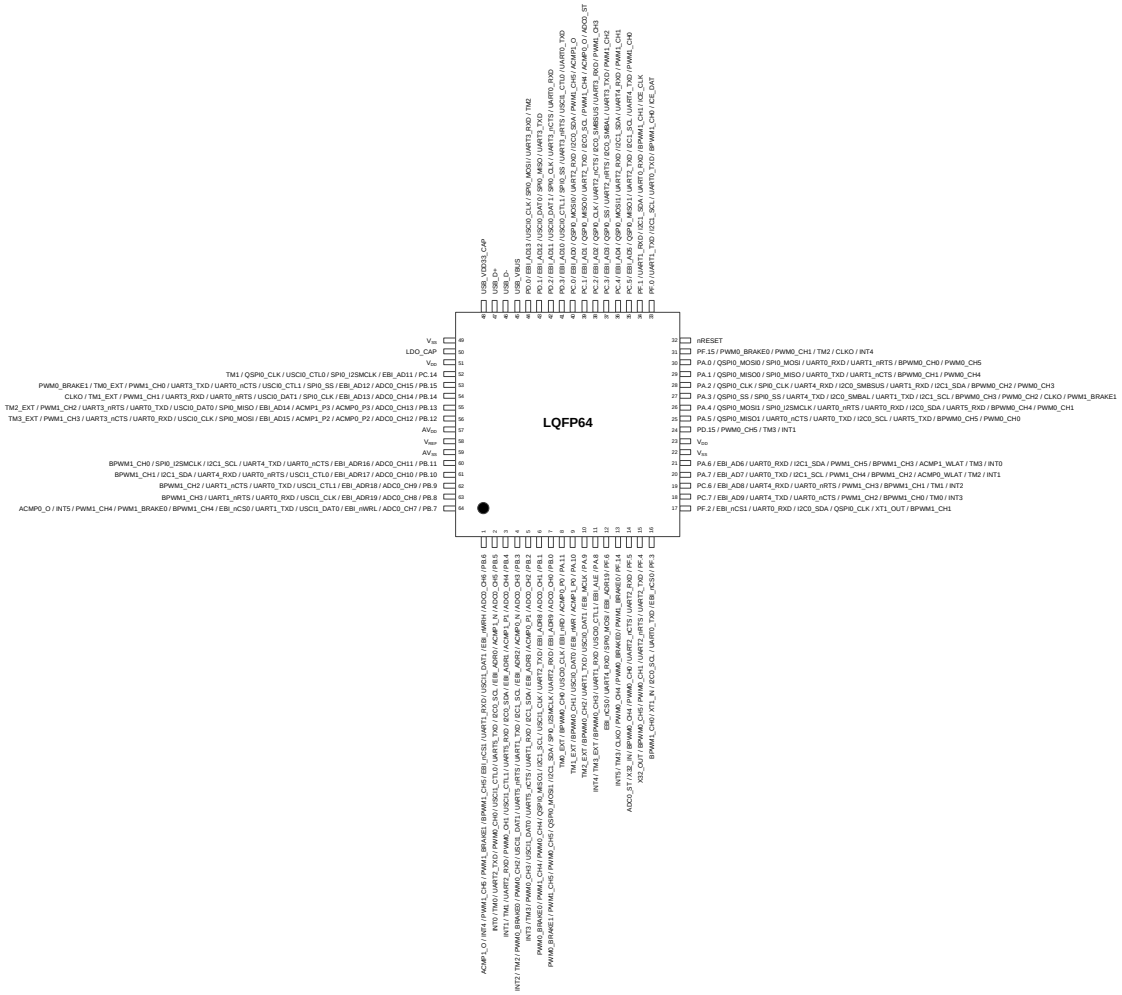


图 4.1-45 M032SG6AE 多功能引脚框图

引脚	M032SG6AE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M032SG6AE引脚功能
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK

引脚	M032SG6AE引脚功能
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	USB_VBUS
46	USB_D-
47	USB_D+
48	USB_V_{DD}33_CAP
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-33 M032SG6AE 多功能引脚表

M032SG8AE

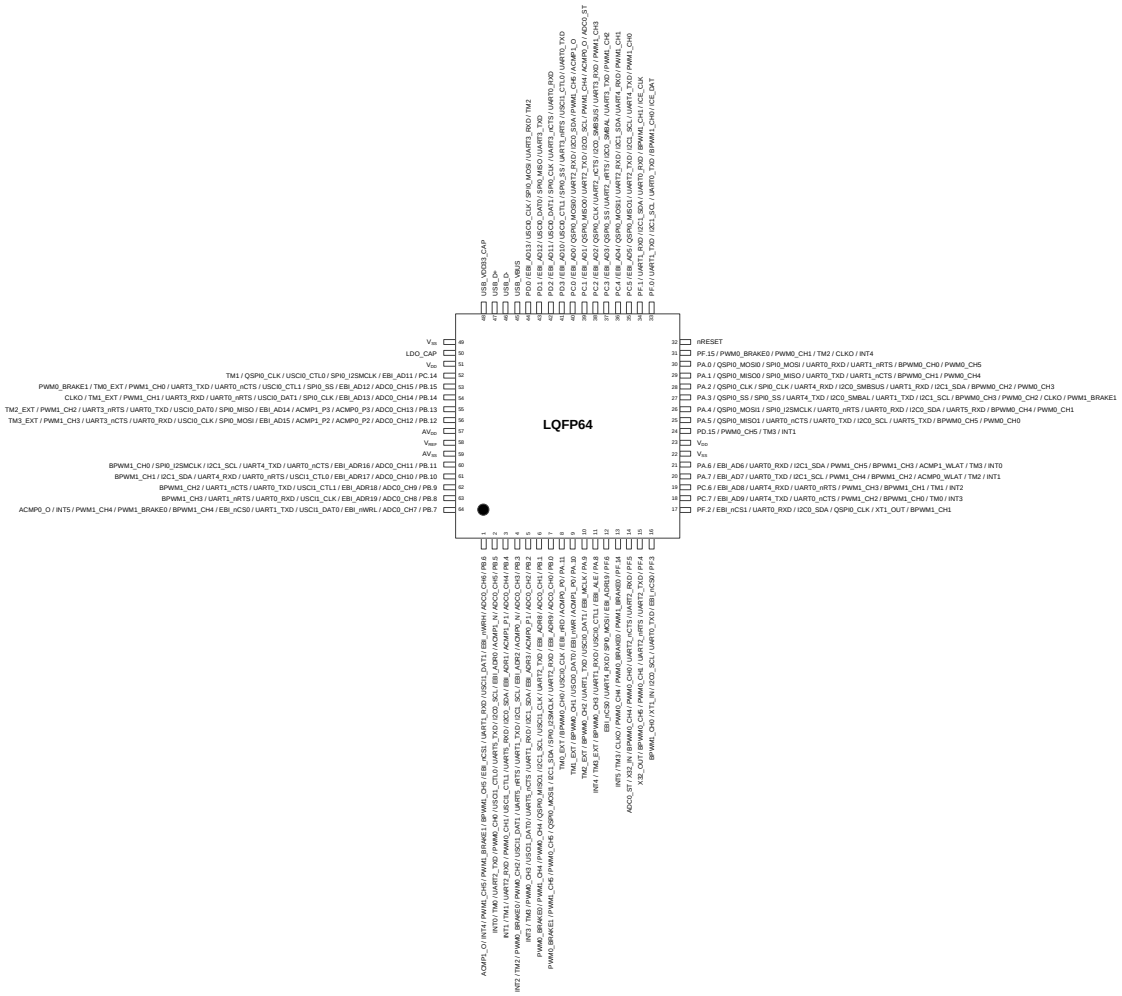


图 4.1-46 M032SG8AE 多功能引脚框图

引脚	M032SG8AE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD

引脚	M032SG8AE引脚功能
	/ TM1 / INT1
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT

引脚	M032SG8AE引脚功能
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	USB_VBUS
46	USB_D-
47	USB_D+
48	USB_V_{DD}33_CAP
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-34 M032SG8AE 多功能引脚表

M032SIAAE

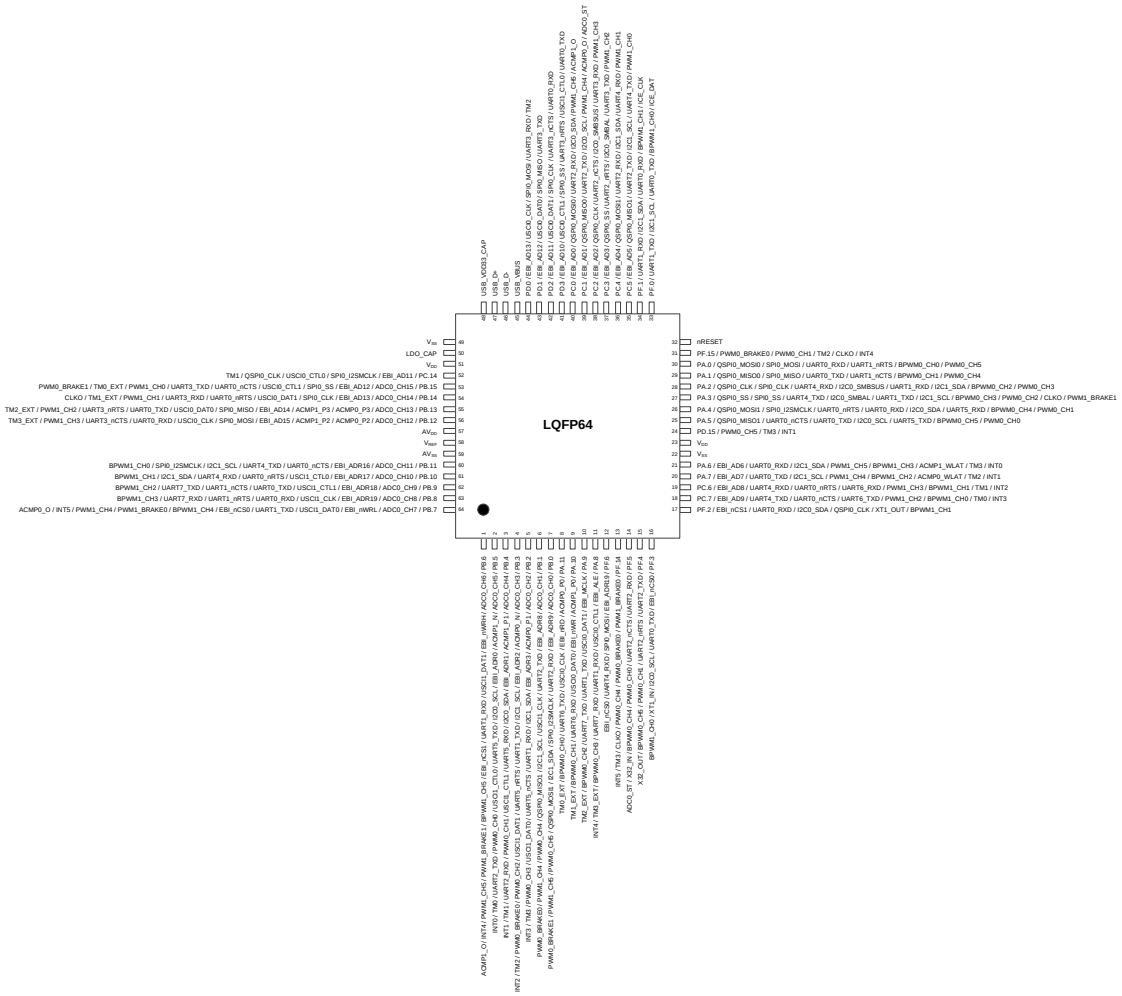


图 4.1-47 M032SIAAE 多功能引脚框图

引脚	M032SIAAE引脚功能
1	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O
2	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
3	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1

引脚	M032SIAAE引脚功能
4	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
5	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
6	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
7	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
8	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / UART6_TXD / BPWM0_CH0 / TM0_EXT
9	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / UART6_RXD / BPWM0_CH1 / TM1_EXT
10	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / UART7_TXD / BPWM0_CH2 / TM2_EXT
11	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / UART7_RXD / BPWM0_CH3 / TM3_EXT / INT4
12	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
13	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
14	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
15	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
16	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
17	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
18	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / UART6_TXD / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
19	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / UART6_RXD / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
20	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
21	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
22	VSS
23	V_{DD}
24	PD.15 / PWM0_CH5 / TM3 / INT1
25	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
26	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
27	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
28	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
29	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
30	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
31	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
32	nRESET
33	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
34	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK

引脚	M032SIAAE引脚功能
35	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
36	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
37	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
38	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
39	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
40	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
41	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
42	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
43	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
44	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
45	USB_VBUS
46	USB_D-
47	USB_D+
48	USB_V_{DD}33_CAP
49	VSS
50	LDO_CAP
51	V_{DD}
52	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
53	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
54	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
55	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
56	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
57	AV_{DD}
58	V_{REF}
59	AVSS
60	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
61	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
62	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / UART7_TXD / BPWM1_CH2
63	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / UART7_RXD / BPWM1_CH3
64	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O

表 4.1-35 M032SIAAE 多功能引脚表

4.1.4.6 M032系列LQFP 128多功能引脚框图

对应料号:M032KG6AE, M032KG8AE, M032KIAAE

M032KG6AE

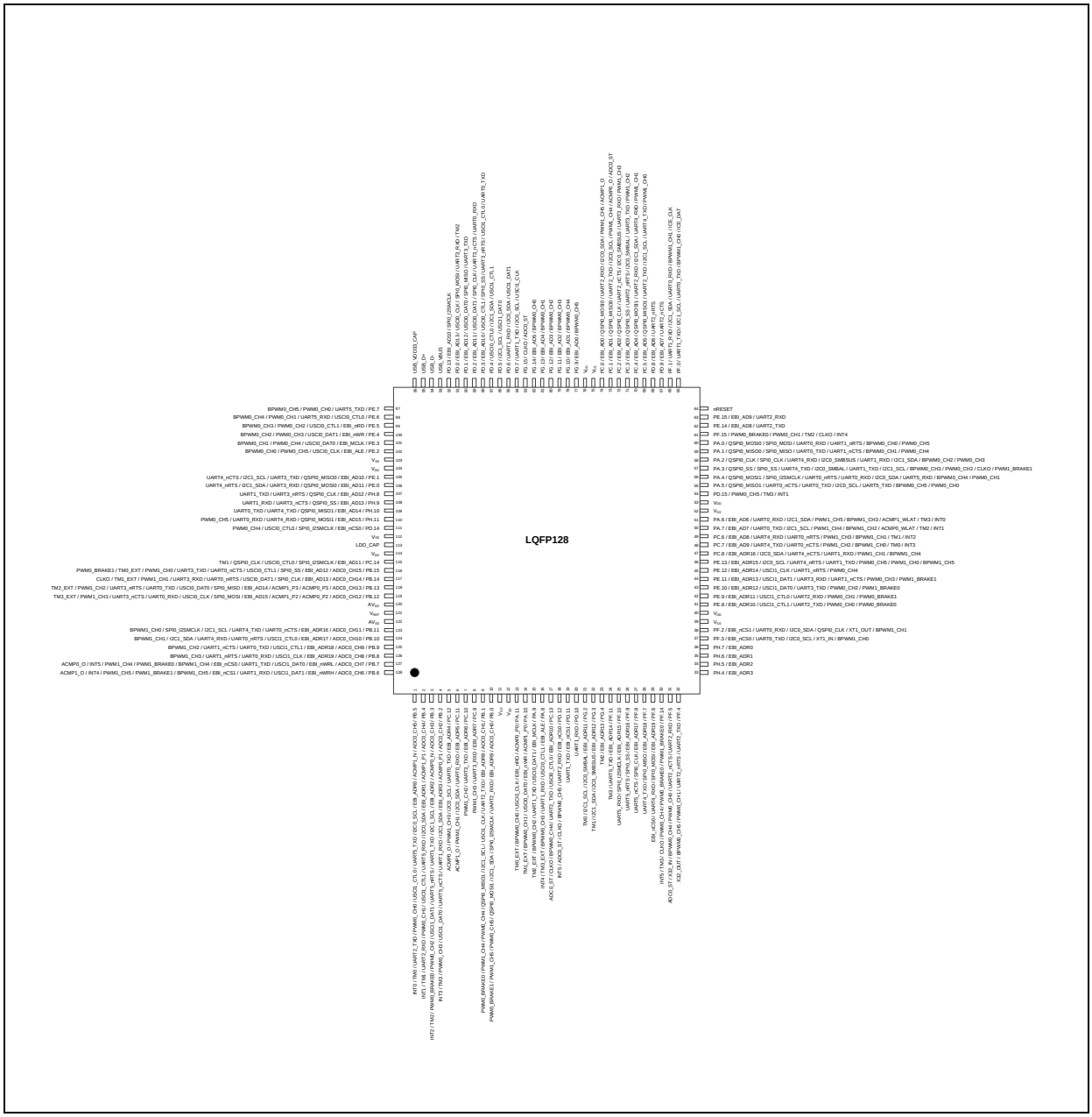


图 4.1-48 M032KG6AE 多功能引脚框图

引脚	M032KG6AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD

引脚	M032KG6AE引脚功能
	/ TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3
34	PH.5 / EBI_ADR2

引脚	M032KG6AE引脚功能
35	PH.6 / EBI_ADR1
36	PH.7 / EBI_ADR0
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD
63	PE.15 / EBI_AD9 / UART2_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS

引脚	M032KG6AE引脚功能
68	PD.8 / EBI_AD6 / UART2_nRTS
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / BPWM0_CH3
80	PG.12 / EBI_AD3 / BPWM0_CH2
81	PG.13 / EBI_AD4 / BPWM0_CH1
82	PG.14 / EBI_AD5 / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	USB_VBUS
94	USB_D-
95	USB_D+
96	USB_V_{DD}33_CAP
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / PWM0_CH4 / BPWM0_CH1

引脚	M032KG6AE引脚功能
102	PE.2 / EBI_ALE / USCI0_CLK / PWM0_CH5 / BPWM0_CH0
103	VSS
104	V_{DD}
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-36 M032KG6AE 多功能引脚表

M032KG8AE

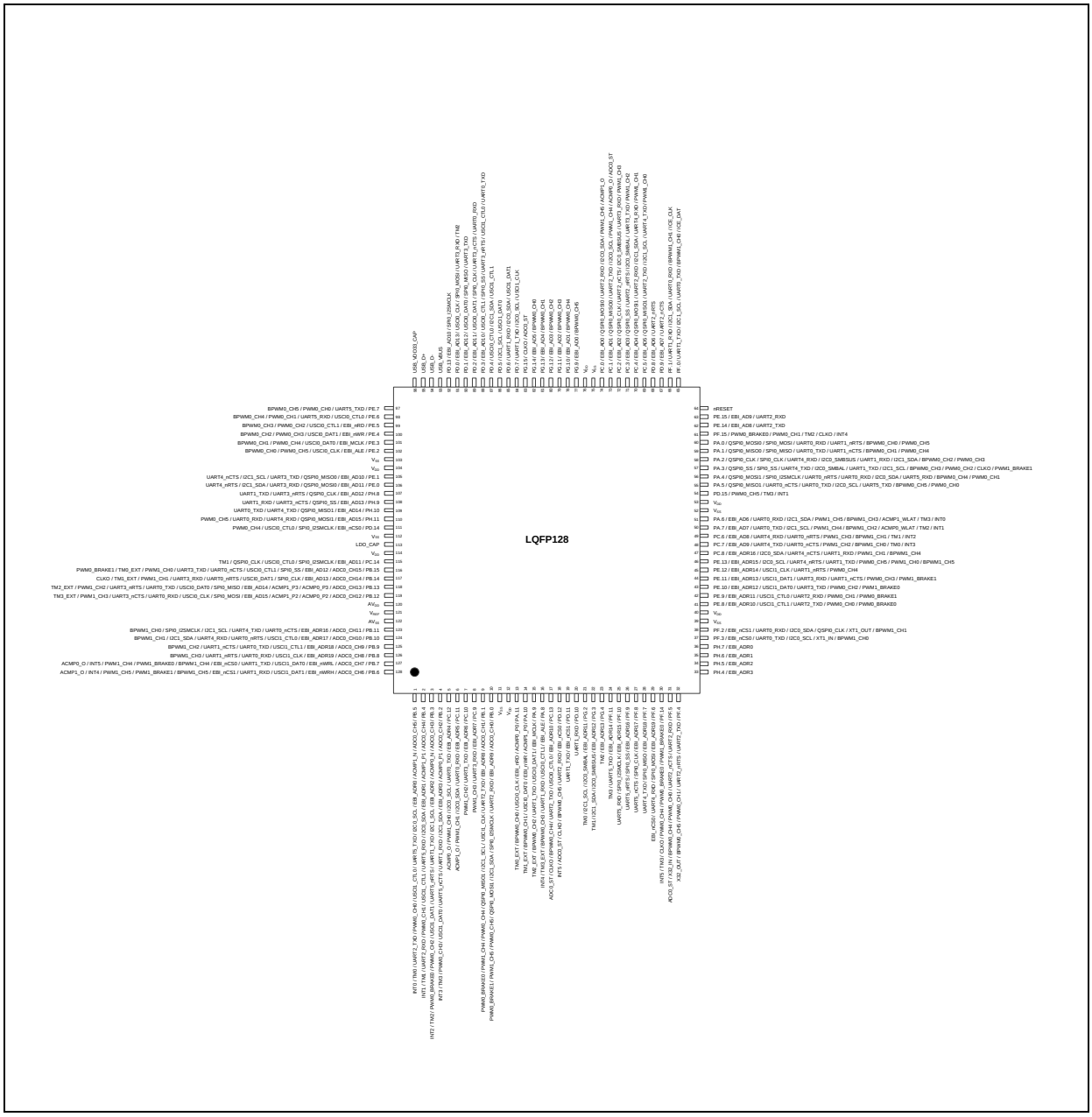


图 4.1-49 M032KG8AE 多功能引脚框图

引脚	M032KG8AE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2

引脚	M032KG8AE引脚功能
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 / PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3
34	PH.5 / EBI_ADR2
35	PH.6 / EBI_ADR1
36	PH.7 / EBI_ADR0

引脚	M032KG8AE引脚功能
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD
63	PE.15 / EBI_AD9 / UART2_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS
68	PD.8 / EBI_AD6 / UART2_nRTS
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0

引脚	M032KG8AE引脚功能
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / BPWM0_CH3
80	PG.12 / EBI_AD3 / BPWM0_CH2
81	PG.13 / EBI_AD4 / BPWM0_CH1
82	PG.14 / EBI_AD5 / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	USB_VBUS
94	USB_D-
95	USB_D+
96	USB_V_{DD}33_CAP
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / PWM0_CH4 / BPWM0_CH1
102	PE.2 / EBI_ALE / USCI0_CLK / PWM0_CH5 / BPWM0_CH0
103	VSS

引脚	M032KG8AE引脚功能
104	V_{DD}
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-37 M032KG8AE 多功能引脚表

M032KIAAE

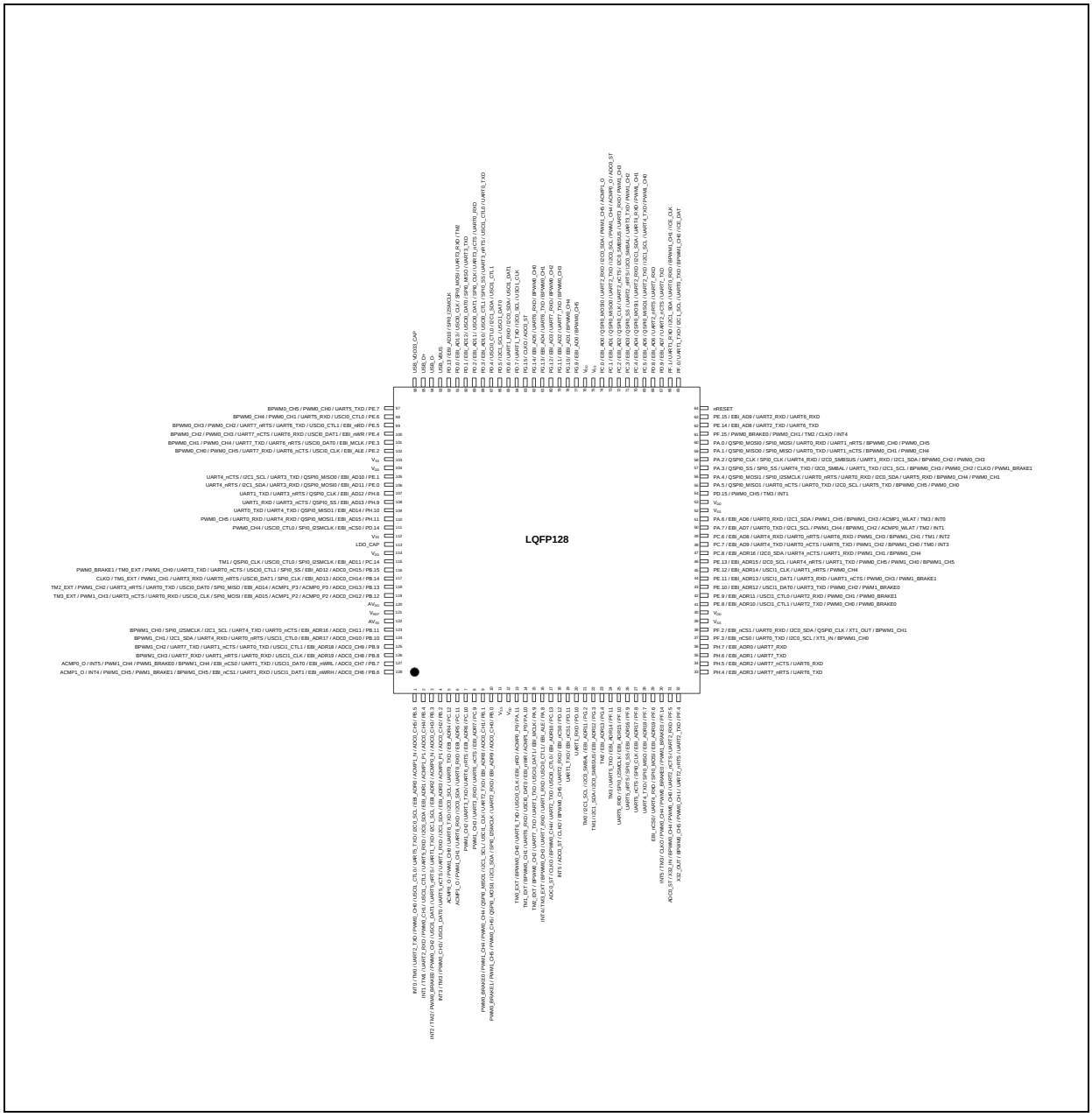


图 4.1-50 M032KIAAE 多功能引脚框图

引脚	M032KIAAE引脚功能
1	PB.5 / ADC0_CH5 / ACMP1_N / EBI_ADR0 / I2C0_SCL / UART5_TXD / USCI1_CTL0 / PWM0_CH0 / UART2_TXD / TM0 / INT0
2	PB.4 / ADC0_CH4 / ACMP1_P1 / EBI_ADR1 / I2C0_SDA / UART5_RXD / USCI1_CTL1 / PWM0_CH1 / UART2_RXD / TM1 / INT1
3	PB.3 / ADC0_CH3 / ACMP0_N / EBI_ADR2 / I2C1_SCL / UART1_TXD / UART5_nRTS / USCI1_DAT1 / PWM0_CH2 / PWM0_BRAKE0 / TM2 / INT2
4	PB.2 / ADC0_CH2 / ACMP0_P1 / EBI_ADR3 / I2C1_SDA / UART1_RXD / UART5_nCTS / USCI1_DAT0 /

引脚	M032KIAAE引脚功能
	PWM0_CH3 / TM3 / INT3
5	PC.12 / EBI_ADR4 / UART0_TXD / I2C0_SCL / UART6_TXD / PWM1_CH0 / ACMP0_O
6	PC.11 / EBI_ADR5 / UART0_RXD / I2C0_SDA / UART6_RXD / PWM1_CH1 / ACMP1_O
7	PC.10 / EBI_ADR6 / UART6_nRTS / UART3_TXD / PWM1_CH2
8	PC.9 / EBI_ADR7 / UART6_nCTS / UART3_RXD / PWM1_CH3
9	PB.1 / ADC0_CH1 / EBI_ADR8 / UART2_TXD / USCI1_CLK / I2C1_SCL / QSPI0_MISO1 / PWM0_CH4 / PWM1_CH4 / PWM0_BRAKE0
10	PB.0 / ADC0_CH0 / EBI_ADR9 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA / QSPI0_MOSI1 / PWM0_CH5 / PWM1_CH5 / PWM0_BRAKE1
11	VSS
12	V_{DD}
13	PA.11 / ACMP0_P0 / EBI_nRD / USCI0_CLK / UART6_TXD / BPWM0_CH0 / TM0_EXT
14	PA.10 / ACMP1_P0 / EBI_nWR / USCI0_DAT0 / UART6_RXD / BPWM0_CH1 / TM1_EXT
15	PA.9 / EBI_MCLK / USCI0_DAT1 / UART1_TXD / UART7_TXD / BPWM0_CH2 / TM2_EXT
16	PA.8 / EBI_ALE / USCI0_CTL1 / UART1_RXD / UART7_RXD / BPWM0_CH3 / TM3_EXT / INT4
17	PC.13 / EBI_ADR10 / USCI0_CTL0 / UART2_TXD / BPWM0_CH4 / CLKO / ADC0_ST
18	PD.12 / EBI_nCS0 / UART2_RXD / BPWM0_CH5 / CLKO / ADC0_ST / INT5
19	PD.11 / EBI_nCS1 / UART1_TXD
20	PD.10 / UART1_RXD
21	PG.2 / EBI_ADR11 / I2C0_SMBAL / I2C1_SCL / TM0
22	PG.3 / EBI_ADR12 / I2C0_SMBSUS / I2C1_SDA / TM1
23	PG.4 / EBI_ADR13 / TM2
24	PF.11 / EBI_ADR14 / UART5_TXD / TM3
25	PF.10 / EBI_ADR15 / SPI0_I2SMCLK / UART5_RXD
26	PF.9 / EBI_ADR16 / SPI0_SS / UART5_nRTS
27	PF.8 / EBI_ADR17 / SPI0_CLK / UART5_nCTS
28	PF.7 / EBI_ADR18 / SPI0_MISO / UART4_TXD
29	PF.6 / EBI_ADR19 / SPI0_MOSI / UART4_RXD / EBI_nCS0
30	PF.14 / PWM1_BRAKE0 / PWM0_BRAKE0 / PWM0_CH4 / CLKO / TM3 / INT5
31	PF.5 / UART2_RXD / UART2_nCTS / PWM0_CH0 / BPWM0_CH4 / X32_IN / ADC0_ST
32	PF.4 / UART2_TXD / UART2_nRTS / PWM0_CH1 / BPWM0_CH5 / X32_OUT
33	PH.4 / EBI_ADR3 / UART7_nRTS / UART6_TXD
34	PH.5 / EBI_ADR2 / UART7_nCTS / UART6_RXD
35	PH.6 / EBI_ADR1 / UART7_TXD
36	PH.7 / EBI_ADR0 / UART7_RXD
37	PF.3 / EBI_nCS0 / UART0_TXD / I2C0_SCL / XT1_IN / BPWM1_CH0

引脚	M032KIAAE引脚功能
38	PF.2 / EBI_nCS1 / UART0_RXD / I2C0_SDA / QSPI0_CLK / XT1_OUT / BPWM1_CH1
39	VSS
40	V_{DD}
41	PE.8 / EBI_ADR10 / USCI1_CTL1 / UART2_TXD / PWM0_CH0 / PWM0_BRAKE0
42	PE.9 / EBI_ADR11 / USCI1_CTL0 / UART2_RXD / PWM0_CH1 / PWM0_BRAKE1
43	PE.10 / EBI_ADR12 / USCI1_DAT0 / UART3_TXD / PWM0_CH2 / PWM1_BRAKE0
44	PE.11 / EBI_ADR13 / USCI1_DAT1 / UART3_RXD / UART1_nCTS / PWM0_CH3 / PWM1_BRAKE1
45	PE.12 / EBI_ADR14 / USCI1_CLK / UART1_nRTS / PWM0_CH4
46	PE.13 / EBI_ADR15 / I2C0_SCL / UART4_nRTS / UART1_TXD / PWM0_CH5 / PWM1_CH0 / BPWM1_CH5
47	PC.8 / EBI_ADR16 / I2C0_SDA / UART4_nCTS / UART1_RXD / PWM1_CH1 / BPWM1_CH4
48	PC.7 / EBI_AD9 / UART4_TXD / UART0_nCTS / UART6_TXD / PWM1_CH2 / BPWM1_CH0 / TM0 / INT3
49	PC.6 / EBI_AD8 / UART4_RXD / UART0_nRTS / UART6_RXD / PWM1_CH3 / BPWM1_CH1 / TM1 / INT2
50	PA.7 / EBI_AD7 / UART0_TXD / I2C1_SCL / PWM1_CH4 / BPWM1_CH2 / ACMP0_WLAT / TM2 / INT1
51	PA.6 / EBI_AD6 / UART0_RXD / I2C1_SDA / PWM1_CH5 / BPWM1_CH3 / ACMP1_WLAT / TM3 / INT0
52	VSS
53	V_{DD}
54	PD.15 / PWM0_CH5 / TM3 / INT1
55	PA.5 / QSPI0_MISO1 / UART0_nCTS / UART0_TXD / I2C0_SCL / UART5_TXD / BPWM0_CH5 / PWM0_CH0
56	PA.4 / QSPI0_MOSI1 / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / UART5_RXD / BPWM0_CH4 / PWM0_CH1
57	PA.3 / QSPI0_SS / SPI0_SS / UART4_TXD / I2C0_SMBAL / UART1_TXD / I2C1_SCL / BPWM0_CH3 / PWM0_CH2 / CLKO / PWM1_BRAKE1
58	PA.2 / QSPI0_CLK / SPI0_CLK / UART4_RXD / I2C0_SMBSUS / UART1_RXD / I2C1_SDA / BPWM0_CH2 / PWM0_CH3
59	PA.1 / QSPI0_MISO0 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1 / PWM0_CH4
60	PA.0 / QSPI0_MOSI0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0 / PWM0_CH5
61	PF.15 / PWM0_BRAKE0 / PWM0_CH1 / TM2 / CLKO / INT4
62	PE.14 / EBI_AD8 / UART2_TXD / UART6_TXD
63	PE.15 / EBI_AD9 / UART2_RXD / UART6_RXD
64	nRESET
65	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / BPWM1_CH0 / ICE_DAT
66	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / BPWM1_CH1 / ICE_CLK
67	PD.9 / EBI_AD7 / UART2_nCTS / UART7_TXD
68	PD.8 / EBI_AD6 / UART2_nRTS / UART7_RXD
69	PC.5 / EBI_AD5 / QSPI0_MISO1 / UART2_TXD / I2C1_SCL / UART4_TXD / PWM1_CH0
70	PC.4 / EBI_AD4 / QSPI0_MOSI1 / UART2_RXD / I2C1_SDA / UART4_RXD / PWM1_CH1

引脚	M032KIAAE引脚功能
71	PC.3 / EBI_AD3 / QSPI0_SS / UART2_nRTS / I2C0_SMBAL / UART3_TXD / PWM1_CH2
72	PC.2 / EBI_AD2 / QSPI0_CLK / UART2_nCTS / I2C0_SMBSUS / UART3_RXD / PWM1_CH3
73	PC.1 / EBI_AD1 / QSPI0_MISO0 / UART2_TXD / I2C0_SCL / PWM1_CH4 / ACMP0_O / ADC0_ST
74	PC.0 / EBI_AD0 / QSPI0_MOSI0 / UART2_RXD / I2C0_SDA / PWM1_CH5 / ACMP1_O
75	VSS
76	V_{DD}
77	PG.9 / EBI_AD0 / BPWM0_CH5
78	PG.10 / EBI_AD1 / BPWM0_CH4
79	PG.11 / EBI_AD2 / UART7_TXD / BPWM0_CH3
80	PG.12 / EBI_AD3 / UART7_RXD / BPWM0_CH2
81	PG.13 / EBI_AD4 / UART6_TXD / BPWM0_CH1
82	PG.14 / EBI_AD5 / UART6_RXD / BPWM0_CH0
83	PG.15 / CLKO / ADC0_ST
84	PD.7 / UART1_TXD / I2C0_SCL / USCI1_CLK
85	PD.6 / UART1_RXD / I2C0_SDA / USCI1_DAT1
86	PD.5 / I2C1_SCL / USCI1_DAT0
87	PD.4 / USCI0_CTL0 / I2C1_SDA / USCI1_CTL1
88	PD.3 / EBI_AD10 / USCI0_CTL1 / SPI0_SS / UART3_nRTS / USCI1_CTL0 / UART0_TXD
89	PD.2 / EBI_AD11 / USCI0_DAT1 / SPI0_CLK / UART3_nCTS / UART0_RXD
90	PD.1 / EBI_AD12 / USCI0_DAT0 / SPI0_MISO / UART3_TXD
91	PD.0 / EBI_AD13 / USCI0_CLK / SPI0_MOSI / UART3_RXD / TM2
92	PD.13 / EBI_AD10 / SPI0_I2SMCLK
93	USB_VBUS
94	USB_D-
95	USB_D+
96	USB_V_{DD}33_CAP
97	PE.7 / UART5_TXD / PWM0_CH0 / BPWM0_CH5
98	PE.6 / USCI0_CTL0 / UART5_RXD / PWM0_CH1 / BPWM0_CH4
99	PE.5 / EBI_nRD / USCI0_CTL1 / UART6_TXD / UART7_nRTS / PWM0_CH2 / BPWM0_CH3
100	PE.4 / EBI_nWR / USCI0_DAT1 / UART6_RXD / UART7_nCTS / PWM0_CH3 / BPWM0_CH2
101	PE.3 / EBI_MCLK / USCI0_DAT0 / UART6_nRTS / UART7_TXD / PWM0_CH4 / BPWM0_CH1
102	PE.2 / EBI_ALE / USCI0_CLK / UART6_nCTS / UART7_RXD / PWM0_CH5 / BPWM0_CH0
103	VSS
104	V_{DD}

引脚	M032KIAAE引脚功能
105	PE.1 / EBI_AD10 / QSPI0_MISO0 / UART3_TXD / I2C1_SCL / UART4_nCTS
106	PE.0 / EBI_AD11 / QSPI0_MOSI0 / UART3_RXD / I2C1_SDA / UART4_nRTS
107	PH.8 / EBI_AD12 / QSPI0_CLK / UART3_nRTS / UART1_TXD
108	PH.9 / EBI_AD13 / QSPI0_SS / UART3_nCTS / UART1_RXD
109	PH.10 / EBI_AD14 / QSPI0_MISO1 / UART4_TXD / UART0_TXD
110	PH.11 / EBI_AD15 / QSPI0_MOSI1 / UART4_RXD / UART0_RXD / PWM0_CH5
111	PD.14 / EBI_nCS0 / SPI0_I2SMCLK / USCI0_CTL0 / PWM0_CH4
112	VSS
113	LDO_CAP
114	V_{DD}
115	PC.14 / EBI_AD11 / SPI0_I2SMCLK / USCI0_CTL0 / QSPI0_CLK / TM1
116	PB.15 / ADC0_CH15 / EBI_AD12 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / PWM1_CH0 / TM0_EXT / PWM0_BRAKE1
117	PB.14 / ADC0_CH14 / EBI_AD13 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / PWM1_CH1 / TM1_EXT / CLKO
118	PB.13 / ADC0_CH13 / ACMP0_P3 / ACMP1_P3 / EBI_AD14 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / PWM1_CH2 / TM2_EXT
119	PB.12 / ADC0_CH12 / ACMP0_P2 / ACMP1_P2 / EBI_AD15 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / PWM1_CH3 / TM3_EXT
120	AV_{DD}
121	V_{REF}
122	AVSS
123	PB.11 / ADC0_CH11 / EBI_ADR16 / UART0_nCTS / UART4_TXD / I2C1_SCL / SPI0_I2SMCLK / BPWM1_CH0
124	PB.10 / ADC0_CH10 / EBI_ADR17 / USCI1_CTL0 / UART0_nRTS / UART4_RXD / I2C1_SDA / BPWM1_CH1
125	PB.9 / ADC0_CH9 / EBI_ADR18 / USCI1_CTL1 / UART0_TXD / UART1_nCTS / UART7_TXD / BPWM1_CH2
126	PB.8 / ADC0_CH8 / EBI_ADR19 / USCI1_CLK / UART0_RXD / UART1_nRTS / UART7_RXD / BPWM1_CH3
127	PB.7 / ADC0_CH7 / EBI_nWRL / USCI1_DAT0 / UART1_TXD / EBI_nCS0 / BPWM1_CH4 / PWM1_BRAKE0 / PWM1_CH4 / INT5 / ACMP0_O
128	PB.6 / ADC0_CH6 / EBI_nWRH / USCI1_DAT1 / UART1_RXD / EBI_nCS1 / BPWM1_CH5 / PWM1_BRAKE1 / PWM1_CH5 / INT4 / ACMP1_O

表 4.1-38 M032KIAAE 多功能引脚表

4.2 引脚映射

同一个封装的不同型号可能具有不同的功能。请参阅第3.2节中的选择指南、4.1节中的引脚配置或 NuTool - PinConfig.。

对应料号: M031xB, M031xC, M031xD, M031xE, M031xG, M031xI, M032xC, M032xD, M032xE, M032xG, M032x系列。

	M031 系列						M032 系列					
引脚名	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin
PB.5	8	12	1	1	2	1		12	1	1	2	1
PB.4	9	13	2	2	3	2		13	2	2	3	2
PB.3	10	14	3	3	4	3		14	3	3	4	3
PB.2	11	15	4	4	5	4		15	4	4	5	4
PC.12						5						5
PC.11						6						6
PC.10						7						7
PC.9						8						8
PB.1		16	5	5	6	9		16	5	5	6	9
PB.0		17	6	6	7	10		17	6	6	7	10
V_{SS}						11						11
V_{DD}						12						12
PA.11				7	8	13				7	8	13
PA.10				8	9	14				8	9	14
PA.9				9	10	15				9	10	15
PA.8				10	11	16				10	11	16
PC.13						17						17
PD.12						18						18
PD.11						19						19
PD.10						20						20
PG.2						21						21
PG.3						22						22
PG.4						23						23
PF.11						24						24
PF.10						25						25
PF.9						26						26
PF.8						27						27
PF.7						28						28

	M031 系列						M032 系列					
引脚名	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin
PF.6					12	29					12	29
PF.14					13	30					13	30
PF.5			7	11	14	31			7	11	14	31
PF.4			8	12	15	32			8	12	15	32
PH.4						33						33
PH.5						34						34
PH.6						35						35
PH.7						36						36
PF.3	12	18	9	13	16	37	11	18	9	13	16	37
PF.2	13	19	10	14	17	38	12	19	10	14	17	38
V_{SS}						39						39
V_{DD}						40						40
PE.8						41						41
PE.9						42						42
PE.10						43						43
PE.11						44						44
PE.12						45						45
PE.13						46						46
PC.8						47						47
PC.7					18	48					18	48
PC.6					19	49					19	49
PA.7				15	20	50				15	20	50
PA.6				16	21	51				16	21	51
V_{SS}					22	52					22	52
V_{DD}					23	53					23	53
PD.15					24	54					24	54
PA.5				17	25	55				17	25	55
PA.4				18	26	56				18	26	56
PA.3	14	20	11	19	27	57	13	20	11	19	27	57
PA.2	15	21	12	20	28	58	14	21	12	20	28	58
PA.1	16	22	13	21	29	59	15	22	13	21	29	59
PA.0	17	23	14	22	30	60	16	23	14	22	30	60
PF.15			15	23	31	61			15	23	31	61
PE.14						62						62

	M031 系列						M032 系列					
引脚名	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin
PE.15						63						63
nRESET	18	24	16	24	32	64	17	24	16	24	32	64
PF.0 ICE_DAT	19	25	17	25	33	65	18	25	17	25	33	65
PF.1 ICE_CLK	20	26	18	26	34	66	19	26	18	26	34	66
PD.9						67						67
PD.8						68						68
PC.5				27	35	69				27	35	69
PC.4				28	36	70				28	36	70
PC.3				29	37	71				29	37	71
PC.2				30	38	72				30	38	72
PC.1		27	19	31	39	73		27	19	31	39	73
PC.0		28	20	32	40	74		28	20	32	40	74
V_{SS}						75						75
V_{DD}						76						76
PG.9						77						77
PG.10						78						78
PG.11						79						79
PG.12						80						80
PG.13						81						81
PG.14						82						82
PG.15						83						83
PD.7						84						84
PD.6						85						85
PD.5						86						86
PD.4						87						87
PD.3					41	88					41	88
PD.2					42	89					42	89
PD.1					43	90					43	90
PD.0					44	91					44	91
PD.13						92						92
NC												
NC												

	M031 系列						M032 系列					
引脚名	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin
NC												
NC												
PA.12		1	21	33	45	93						
PA.13		2	22	34	46	94						
PA.14		3	23	35	47	95						
PA.15		4	24	36	48	96						
USB_VBUS							20	1	21	33	45	93
USB_D-							1	2	22	34	46	94
USB_D+							2	3	23	35	47	95
USB_V_{DD}33_CAP							3	4	24	36	48	96
PE.7						97						97
PE.6						98						98
PE.5						99						99
PE.4						100						100
PE.3						101						101
PE.2						102						102
V_{SS}						103						103
V_{DD}						104						104
PE.1						105						105
PE.0						106						106
PH.8						107						107
PH.9						108						108
PH.10						109						109
PH.11						110						110
PD.14						111						111
V_{SS}	1	5	25	37	49	112	4	5	25	37	49	112
LDO_CAP	2	6	26	38	50	113	5	6	26	38	50	113
V_{DD}	3	7	27	39	51	114	6	7	27	39	51	114
PC.14				40	52	115				40	52	115
PB.15			28	41	53	116			28	41	53	116
PB.14	4	8	29	42	54	117	7	8	29	42	54	117
PB.13	5	9	30	43	55	118	8	9	30	43	55	118
PB.12	6	10	31	44	56	119	9	10	31	44	56	119
AV_{DD}	7	11	32	45	57	120	10	11	32	45	57	120

	M031 系列						M032 系列					
引脚名	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin	20 Pin	28 Pin	32 Pin	48 Pin	64 Pin	128 Pin
V_{REF}					58	121					58	121
AV_{SS}				46	59	122				46	59	122
PB.11					60	123					60	123
PB.10					61	124					61	124
PB.9					62	125					62	125
PB.8					63	126					63	126
PB.7				47	64	127				47	64	127
PB.6				48	1	128				48	1	128

表 4.2-1 引脚映射表

4.3 引脚功能描述

组	引脚名	类型	描述
ACMP0	ACMP0_N	A	模拟比较器 0 的负输入引脚。
	ACMP0_O	O	模拟比较器 0 输出引脚。
	ACMP0_P0	A	模拟比较器 0 的正输入 0 引脚。
	ACMP0_P1	A	模拟比较器 0 的正输入 1 引脚。
	ACMP0_P2	A	模拟比较器 0 的正输入 2 引脚。
	ACMP0_P3	A	模拟比较器 0 的正输入 3 引脚。
	ACMP0_WLAT	I	模拟比较器 0 窗口锁存器输入引脚
ACMP1	ACMP1_N	A	模拟比较器 1 的负输入引脚。
	ACMP1_O	O	模拟比较器 1 的输出引脚。
	ACMP1_P0	A	模拟比较器 1 的正输入 0 引脚。
	ACMP1_P1	A	模拟比较器 1 的正输入 1 引脚。
	ACMP1_P2	A	模拟比较器 1 的正输入 2 引脚。
	ACMP1_P3	A	模拟比较器 1 的正输入 3 引脚。
	ACMP1_WLAT	I	模拟比较器 1 窗口锁存器输入引脚
ADC0	ADC0_CH0	A	ADC0 通道 0 模拟输入。
	ADC0_CH1	A	ADC0 通道 1 模拟输入。
	ADC0_CH2	A	ADC0 通道 2 模拟输入。
	ADC0_CH3	A	ADC0 通道 3 模拟输入。
	ADC0_CH4	A	ADC0 通道 4 模拟输入。
	ADC0_CH5	A	ADC0 通道 5 模拟输入。
	ADC0_CH6	A	ADC0 通道 6 模拟输入。
	ADC0_CH7	A	ADC0 通道 7 模拟输入。
	ADC0_CH8	A	ADC0 通道 8 模拟输入。
	ADC0_CH9	A	ADC0 通道 9 模拟输入。
	ADC0_CH10	A	ADC0 通道 10 模拟输入。
	ADC0_CH11	A	ADC0 通道 11 模拟输入。
	ADC0_CH12	A	ADC0 通道 12 模拟输入。
	ADC0_CH13	A	ADC0 通道 13 模拟输入。
	ADC0_CH14	A	ADC0 通道 14 模拟输入。
	ADC0_CH15	A	ADC0 通道 15 模拟输入。
	ADC0_ST	I	ADC0 外部触发输入引脚。
BPWM0	BPWM0_CH0	I/O	BPWM0 通道 0 输出/捕获输入。
	BPWM0_CH1	I/O	BPWM0 通道 1 输出/捕获输入。
	BPWM0_CH2	I/O	BPWM0 通道 2 输出/捕获输入。

组	引脚名	类型	描述
	BPWM0_CH3	I/O	BPWM0 通道 3 输出/捕获输入。
	BPWM0_CH4	I/O	BPWM0 通道 4 输出/捕获输入。
	BPWM0_CH5	I/O	BPWM0 通道 5 输出/捕获输入。
BPWM1	BPWM1_CH0	I/O	BPWM1 通道 0 输出/捕获输入。
	BPWM1_CH1	I/O	BPWM1 通道 1 输出/捕获输入。
	BPWM1_CH2	I/O	BPWM1 通道 2 输出/捕获输入。
	BPWM1_CH3	I/O	BPWM1 通道 3 输出/捕获输入。
	BPWM1_CH4	I/O	BPWM1 通道 4 输出/捕获输入。
	BPWM1_CH5	I/O	BPWM1 通道 5 输出/捕获输入。
CLKO	CLKO	O	时钟输出
EBI	EBI_AD0	I/O	EBI 地址/数据总线位 0。
	EBI_AD1	I/O	EBI 地址/数据总线位 1。
	EBI_AD2	I/O	EBI 地址/数据总线位 2。
	EBI_AD3	I/O	EBI 地址/数据总线位 3。
	EBI_AD4	I/O	EBI 地址/数据总线位 4。
	EBI_AD5	I/O	EBI 地址/数据总线位 5。
	EBI_AD6	I/O	EBI 地址/数据总线位 6。
	EBI_AD7	I/O	EBI 地址/数据总线位 7。
	EBI_AD8	I/O	EBI 地址/数据总线位 8。
	EBI_AD9	I/O	EBI 地址/数据总线位 9。
	EBI_AD10	I/O	EBI 地址/数据总线位 10。
	EBI_AD11	I/O	EBI 地址/数据总线位 11。
	EBI_AD12	I/O	EBI 地址/数据总线位 12。
	EBI_AD13	I/O	EBI 地址/数据总线位 13。
	EBI_AD14	I/O	EBI 地址/数据总线位 14。
	EBI_AD15	I/O	EBI 地址/数据总线位 15。
	EBI_ADR0	O	EBI 地址总线位 0。
	EBI_ADR1	O	EBI 地址总线位 1。
	EBI_ADR2	O	EBI 地址总线位 2。
	EBI_ADR3	O	EBI 地址总线位 3。
	EBI_ADR4	O	EBI 地址总线位 4。
	EBI_ADR5	O	EBI 地址总线位 5。
	EBI_ADR6	O	EBI 地址总线位 6。
	EBI_ADR7	O	EBI 地址总线位 7。
	EBI_ADR8	O	EBI 地址总线位 8。
	EBI_ADR9	O	EBI 地址总线位 9。

组	引脚名	类型	描述
	EBI_ADR10	O	EBI 地址总线位 10。
	EBI_ADR11	O	EBI 地址总线位 11。
	EBI_ADR12	O	EBI 地址总线位 12。
	EBI_ADR13	O	EBI 地址总线位 13。
	EBI_ADR14	O	EBI 地址总线位 14。
	EBI_ADR15	O	EBI 地址总线位 15。
	EBI_ADR16	O	EBI 地址总线位 16。
	EBI_ADR17	O	EBI 地址总线位 17。
	EBI_ADR18	O	EBI 地址总线位 18。
	EBI_ADR19	O	EBI 地址总线位 19。
	EBI_ALE	O	EBI 地址锁存使能输出引脚。
	EBI_MCLK	O	EBI 外部时钟输出引脚。
	EBI_nCS0	O	EBI 芯片选择 0 输出引脚。
	EBI_nCS1	O	EBI 芯片选择 1 个输出引脚。
	EBI_nRD	O	EBI 读取使能输出引脚。
	EBI_nWR	O	EBI 写使能输出引脚。
	EBI_nWRH	O	EBI 高字节写使能输出引脚
	EBI_nWRL	O	EBI 低字节写使能输出引脚。
GPIO	PA.x~PH.x	I/O	通用数字 I / O 引脚。
I2C0	I2C0_SCL	I/O	I2C0 时钟引脚。
	I2C0_SDA	I/O	I2C0 数据输入/输出引脚。
	I2C0_SMBAL	O	I2C0 SMBus SMBALTER 引脚
	I2C0_SMBSUS	O	I2C0 SMBus SMBSUS 引脚（PMBus CONTROL 引脚）
I2C1	I2C1_SCL	I/O	I2C1 时钟引脚。
	I2C1_SDA	I/O	I2C1 数据输入/输出引脚。
ICE	ICE_CLK	I	串行有线调试器时钟引脚 注意：建议在 ICE_CLK 引脚上使用 100kΩ 上拉电阻
	ICE_DAT	I/O	串行有线调试器数据引脚 注意：建议在 ICE_DAT 引脚上使用 100kΩ 上拉电阻
	nRESET	I	外部复位输入：低电平有效，内部上拉。将此引脚设置为低电平复位至初始状态。 注意：建议在 nRESET 引脚上使用 10kΩ 上拉电阻和 10 uF 电容器。
INT0	INT0	I	外部中断 0 输入引脚。
INT1	INT1	I	外部中断 1 输入引脚。
INT3	INT3	I	外部中断 3 输入引脚。
INT4	INT4	I	外部中断 4 输入引脚。
INT5	INT5	I	外部中断 5 输入引脚。
PWM0	PWM0_BRAKE0	I	PWM0 制动 0 输入引脚。
	PWM0_BRAKE1	I	PWM0 制动 1 输入引脚。

组	引脚名	类型	描述
	PWM0_CH0	I/O	PWM0 通道 0 输出/捕获输入。
	PWM0_CH1	I/O	PWM0 通道 1 输出/捕获输入。
	PWM0_CH2	I/O	PWM0 通道 2 输出/捕获输入。
	PWM0_CH3	I/O	PWM0 通道 3 输出/捕获输入。
	PWM0_CH4	I/O	PWM0 通道 4 输出/捕获输入。
	PWM0_CH5	I/O	PWM0 通道 5 输出/捕获输入。
PWM1	PWM1_BRAKE0	I	PWM1 制动 0 输入引脚。
	PWM1_BRAKE1	I	PWM1 制动 1 输入引脚。
	PWM1_CH0	I/O	PWM1 通道 0 输出/捕获输入。
	PWM1_CH1	I/O	PWM1 通道 1 输出/捕获输入。
	PWM1_CH2	I/O	PWM1 通道 2 输出/捕获输入。
	PWM1_CH3	I/O	PWM1 通道 3 输出/捕获输入。
	PWM1_CH4	I/O	PWM1 通道 4 输出/捕获输入。
	PWM1_CH5	I/O	PWM1 通道 5 输出/捕获输入。
Power	V_{DD}	P	用于 I / O 端口的电源和用于内部 PLL 和数字电路的 LDO 源。
	V_{SS}	P	数字电路的接地引脚。
	AV_{DD}	P	内部模拟电路的电源。
	AV_{SS}	P	模拟电路的接地引脚。
	V_{REF}	A	ADC 参考电压输入。 注意：该引脚需要连接一个 1uF 电容器。
	LDO_CAP	A	LDO 输出引脚。 注意：该引脚需要连接一个 1uF 电容器。
QSPI0	QSPI0_CLK	I/O	四路 SPI0 串行时钟引脚。
	QSPI0_MISO0	I/O	四路 SPI0 MISO0（主机输入，从机输出）引脚。
	QSPI0_MISO1	I/O	四路 SPI0 MISO1（主机输入，从机输出）引脚。
	QSPI0_MOSI0	I/O	四路 SPI0 MOSI0（主机输出，从机输入）引脚。
	QSPI0_MOSI1	I/O	四路 SPI0 MOSI1（主机输出，从机输入）引脚。
	QSPI0_SS	I/O	四路 SPI0 从选择引脚。
SPI0	SPI0_CLK	I/O	SPI0 串行时钟引脚。
	SPI0_I2SMCLK	I/O	SPI0 I2S 主时钟输出引脚
	SPI0_MISO	I/O	SPI0 MISO（主机输入，从机输出）引脚。
	SPI0_MOSI	I/O	SPI0 MOSI（主机输出，从机输入）引脚。
	SPI0_SS	I/O	SPI0 从选择引脚。
TM0	TM0	I/O	Timer0 事件计数器输入/切换输出引脚。
	TM0_EXT	I/O	Timer0 外部捕获输入/切换输出引脚。
TM1	TM1	I/O	Timer1 事件计数器的输入/切换输出引脚。
	TM1_EXT	I/O	Timer1 外部捕获输入/切换输出引脚。

组	引脚名	类型	描述
TM2	TM2	I/O	Timer2 事件计数器输入/切换输出引脚。
	TM2_EXT	I/O	Timer2 外部捕获输入/切换输出引脚。
TM3	TM3	I/O	Timer3 事件计数器输入/切换输出引脚。
	TM3_EXT	I/O	Timer3 外部捕获输入/切换输出引脚。
UART0	UART0_RXD	I	UART0 数据接收器输入引脚。
	UART0_TXD	O	UART0 数据发送器输出引脚。
	UART0_nCTS	I	UART0 清除发送输入引脚。
	UART0_nRTS	O	UART0 请求发送输出引脚。
UART1	UART1_RXD	I	UART1 数据接收器输入引脚。
	UART1_TXD	O	UART1 数据发送器输出引脚。
	UART1_nCTS	I	UART1 清除发送输入引脚。
	UART1_nRTS	O	UART1 请求发送输出引脚。
UART2	UART2_RXD	I	UART2 数据接收器输入引脚。
	UART2_TXD	O	UART2 数据发送器输出引脚。
	UART2_nCTS	I	UART2 清除发送输入引脚。
	UART2_nRTS	O	UART2 请求发送输出引脚。
UART3	UART3_RXD	I	UART3 数据接收器输入引脚。
	UART3_TXD	O	UART3 数据发送器输出引脚。
	UART3_nCTS	I	UART3 清除发送输入引脚。
	UART3_nRTS	O	UART3 请求发送输出引脚。
UART4	UART4_RXD	I	UART4 数据接收器输入引脚。
	UART4_TXD	O	UART4 数据发送器输出引脚。
	UART4_nCTS	I	UART4 清除发送输入引脚。
	UART4_nRTS	O	UART4 请求发送输出引脚。
UART5	UART5_RXD	I	UART5 数据接收器输入引脚。
	UART5_TXD	O	UART5 数据发送器输出引脚。
	UART5_nCTS	I	UART5 清除发送输入引脚。
	UART5_nRTS	O	UART5 请求发送输出引脚。
UART6	UART6_RXD	I	UART6 数据接收器输入引脚。
	UART6_TXD	O	UART6 数据发送器输出引脚。
	UART6_nCTS	I	UART6 清除发送输入引脚。
	UART6_nRTS	O	UART6 请求发送输出引脚。
UART7	UART7_RXD	I	UART7 数据接收器输入引脚。
	UART7_TXD	O	UART7 数据发送器输出引脚。
	UART7_nCTS	I	UART7 清除发送输入引脚。
	UART7_nRTS	O	UART7 请求发送输出引脚。

组	引脚名	类型	描述
USB	USB_VBUS	P	USB 主机或 HUB 的电源。
	USB_D-	A	USB 差分信号 D-。
	USB_D+	A	USB 差分信号 D +。
	USB_V_{DD}33_CAP	A	内部功率调节器输出 3.3V 去耦引脚。
USCI0	USCI0_CLK	I/O	USCI0 时钟引脚。
	USCI0_CTL0	I/O	USCI0 控制 0 引脚。
	USCI0_CTL1	I/O	USCI0 控制 1 引脚。
	USCI0_DAT0	I/O	USCI0 数据 0 引脚。
	USCI0_DAT1	I/O	USCI0 数据 1 引脚。
USCI1	USCI1_CLK	I/O	USCI1 时钟引脚。
	USCI1_CTL0	I/O	USCI1 控制 0 引脚。
	USCI1_CTL1	I/O	USCI1 控制 1 引脚。
	USCI1_DAT0	I/O	USCI1 数据 0 引脚。
	USCI1_DAT1	I/O	USCI1 数据 1 引脚。
X32	X32_IN	I	外部 32.768 kHz 晶体输入引脚。
	X32_OUT	O	外部 32.768 kHz 晶体输出引脚。
XT1	XT1_IN	I	外部高速晶体输入引脚。
	XT1_OUT	O	外部高速晶体输出引脚。

表 4.3-1 引脚功能描述表

5 框图

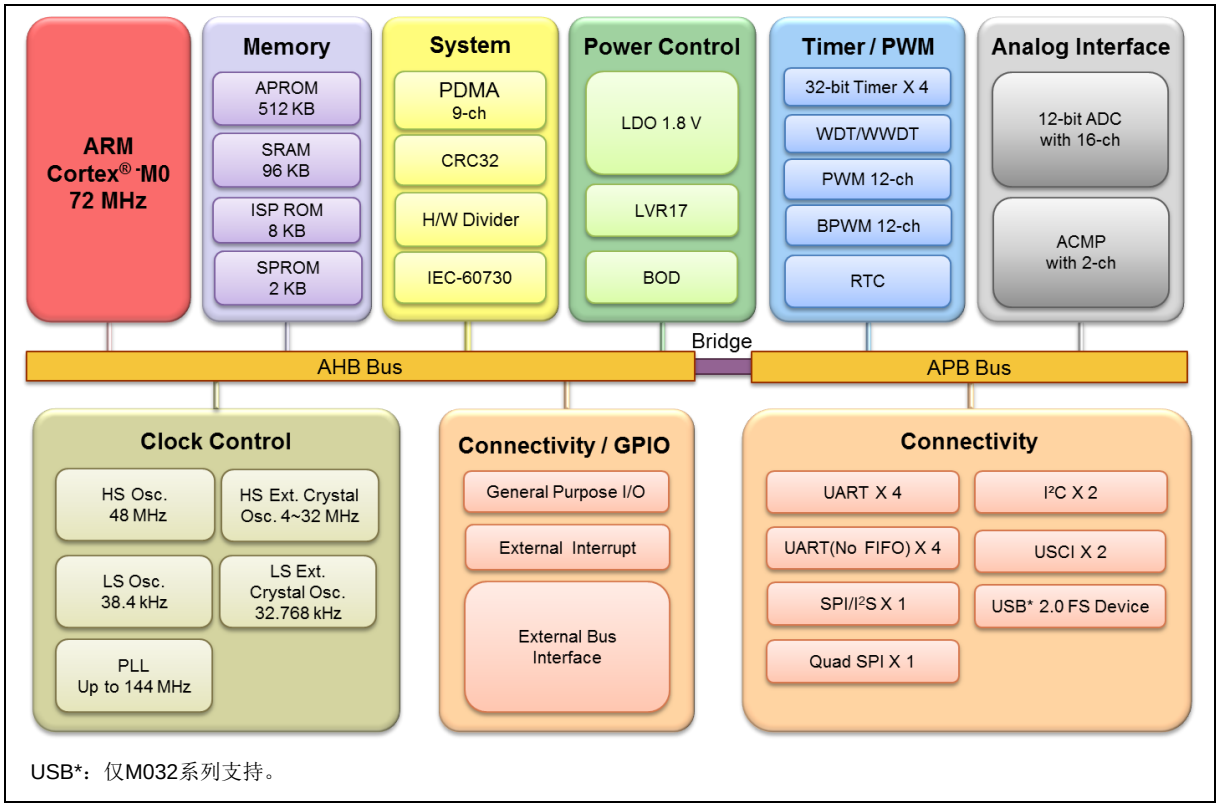


图 5-1 NuMicro® M031/M032 框图

6 功能描述

6.1 Arm® Cortex®-M0 内核

Cortex®-M0处理器是一个可配置的，多级流水线的32位RISC处理器，有一个AMBA AHB-Lite接口和一个嵌套式中断向量控制器（NVIC）。它还具有可选的硬件调试功能。 处理器可执行与其他Cortex®-M处理器兼容的Thumb指令。支持两种处理模式：Thread模式和Handler模式。异常时进入Handler模式。异常只能在Handler模式下返回。复位后进入Thread模式，异常返回后也可以进入Thread模式。图 6-1展示了处理器的功能框图。

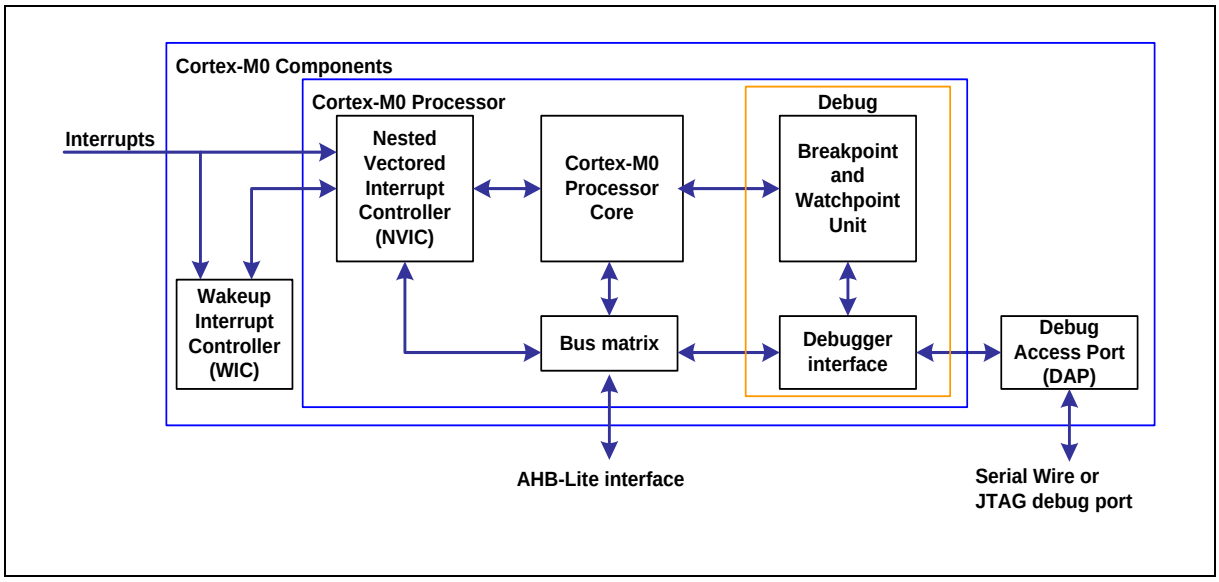


图 6-1 功能框图

设备提供：

- 低门数处理器:
  - Arm®6-M Thumb® 指令集
  - Thumb-2 技术
  - Arm®6-M 兼容24位SysTick定时器
  - 一个 32位硬件乘法器
  - 系统接口支持小端数据访问
  - 准确而及时的中断处理能力
  - 加载/存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
  - C应用程序二进制接口的异常兼容模式。这个ARMv6-M的模式允许用户使用纯C函数实现中断处理
  - 使用等待中断指令（WFI），等待事件指令（WFE）进入低功耗休眠模式，或者从中断退出休眠模式
- NVIC：
  - 32个外部中断输入，每个有4个优先级可设定

- 专用的不可屏蔽中断（NMI）输入
- 同时支持电平触发和脉冲触发中断
- 支持中断唤醒控制器（WIC），且提供超低功耗休眠模式

- 调试模式：
  - 四个硬件断点
  - 两个观察点
  - 用于非侵入式代码分析的程序计数采集寄存器（PCSR）
  - 单步和向量捕获能力
- 总线接口：
  - 提供简单的集成到所有系统外设和存储器的单一32位AMBA-3 AHB-Lite系统接口
  - 支持DAP（Debug Access Port）的单个32位从机接口

6.2 时钟控制器

6.2.1 概述

时钟控制器为整个芯片提供时钟源，包括系统时钟和所有外围设备时钟。该控制器还通过单独时钟的开或关，时钟源选择和分频器来进行功耗控制。只有在CPU使能低功耗PDEN（CLK_PWRCTL[7]）位和Cortex®-M0内核执行WFI指令后，芯片才进入低功耗模式。直到唤醒中断发生，芯片才会退出低功耗模式。在低功耗模式下，时钟控制器会关闭4～32MHz外部高速晶振（HXT），48Mhz内部高速RC振荡器和可编程PLL输出时钟频率（PLLFOUT）来降低整个系统功耗。图 6.2-1和图 6.2-2展示了时钟生成器和时钟源控制器。

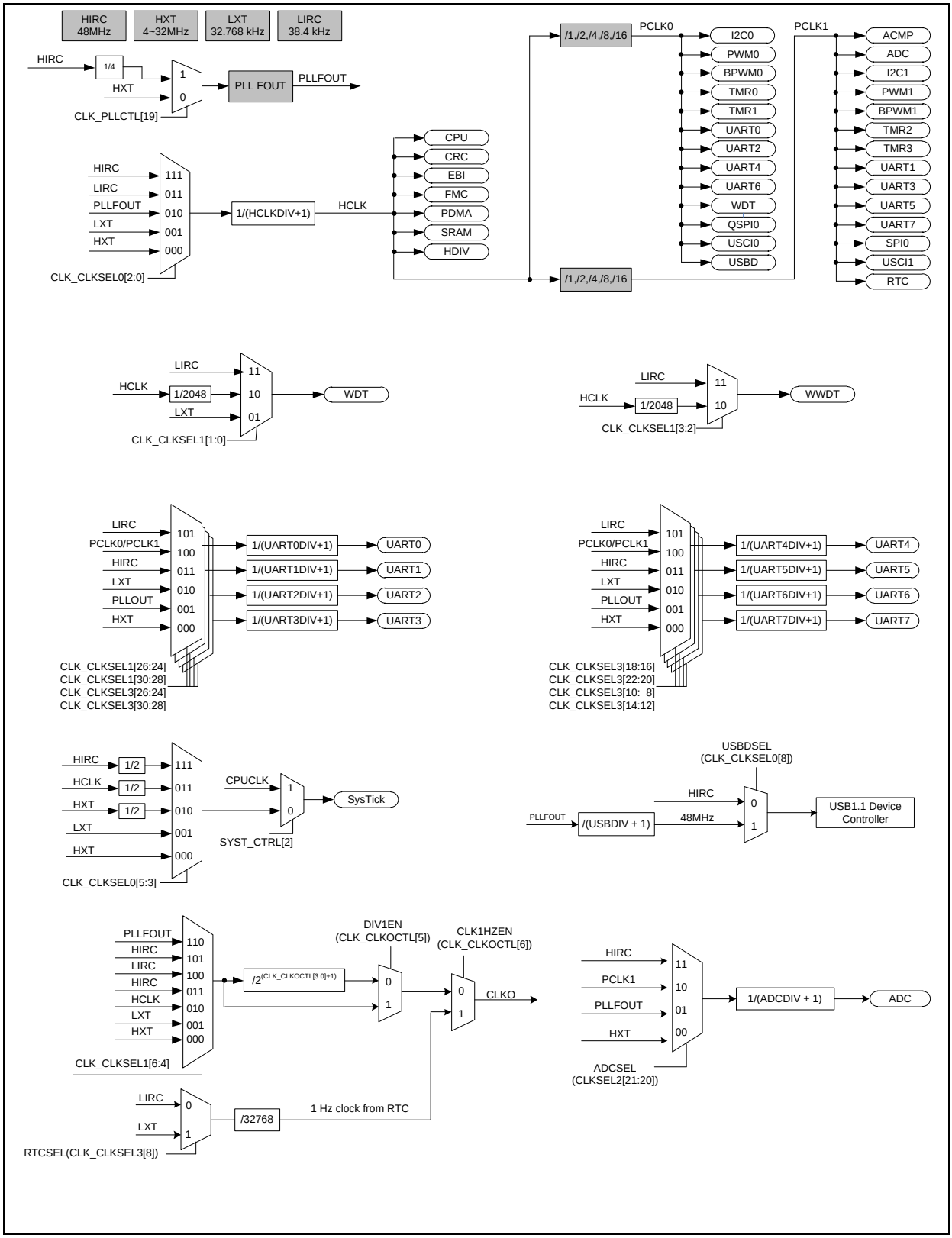


图 6.2-1 时钟发生器全局示意图 （1/2）

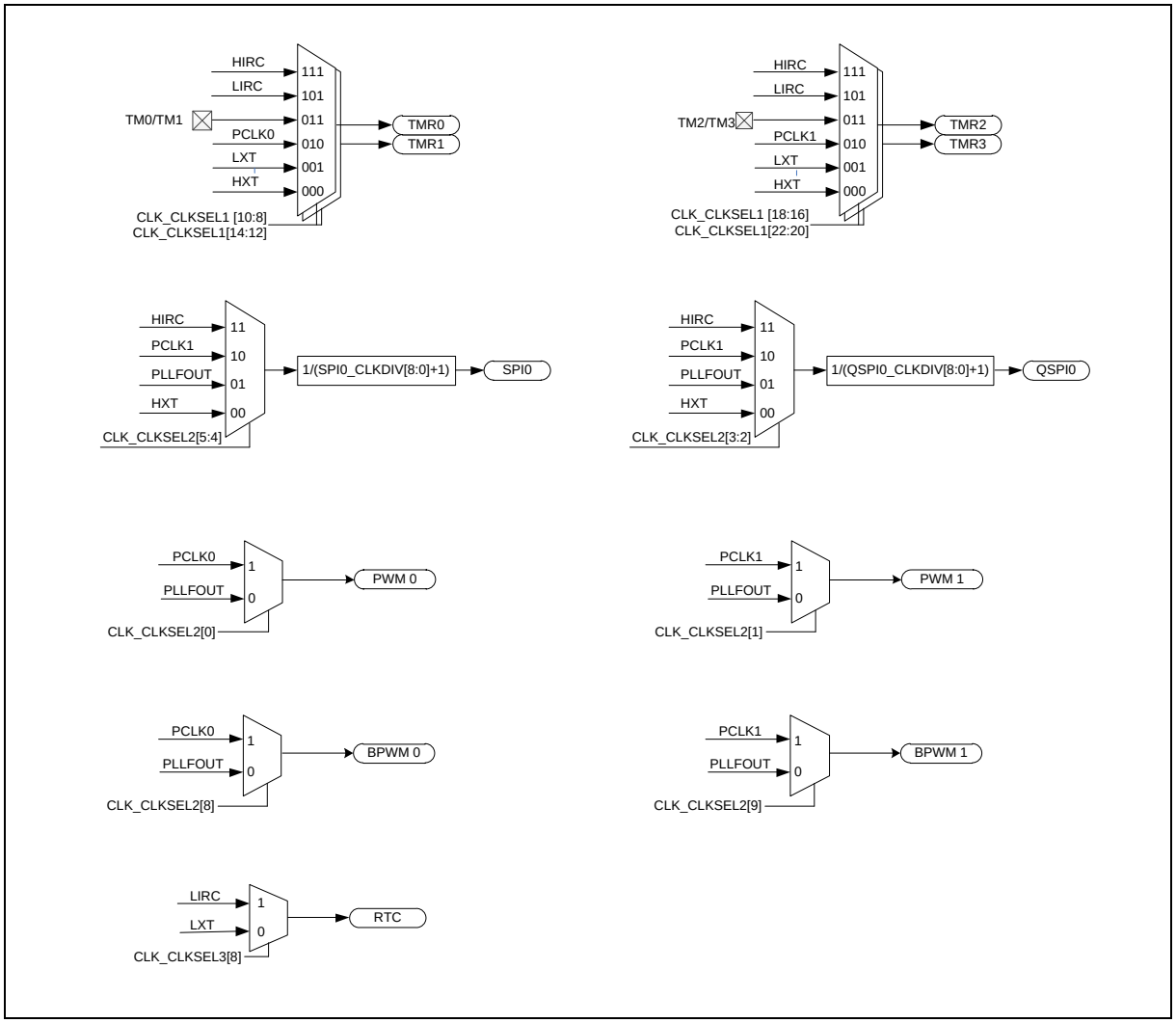


图 6.2-2 时钟发生器全局示意图（2/2）

6.2.2 时钟发生器

时钟发生器由如下5个时钟源组成:

- 32.768 kHz 外部低速晶体振荡器（LXT）
- 4~32 MHz 外部高速晶体振荡器（HXT）
- 可编程PLL输出时钟频率（PLLFOUT），PLL 由外部 4~32 MHz 高速晶振（HXT）或内部48M高速振荡器（HIRC/4）提供时钟源
- 48 MHz 内部高速RC振荡器（HIRC）
- 38.4 kHz 内部低速RC振荡器（LIRC）

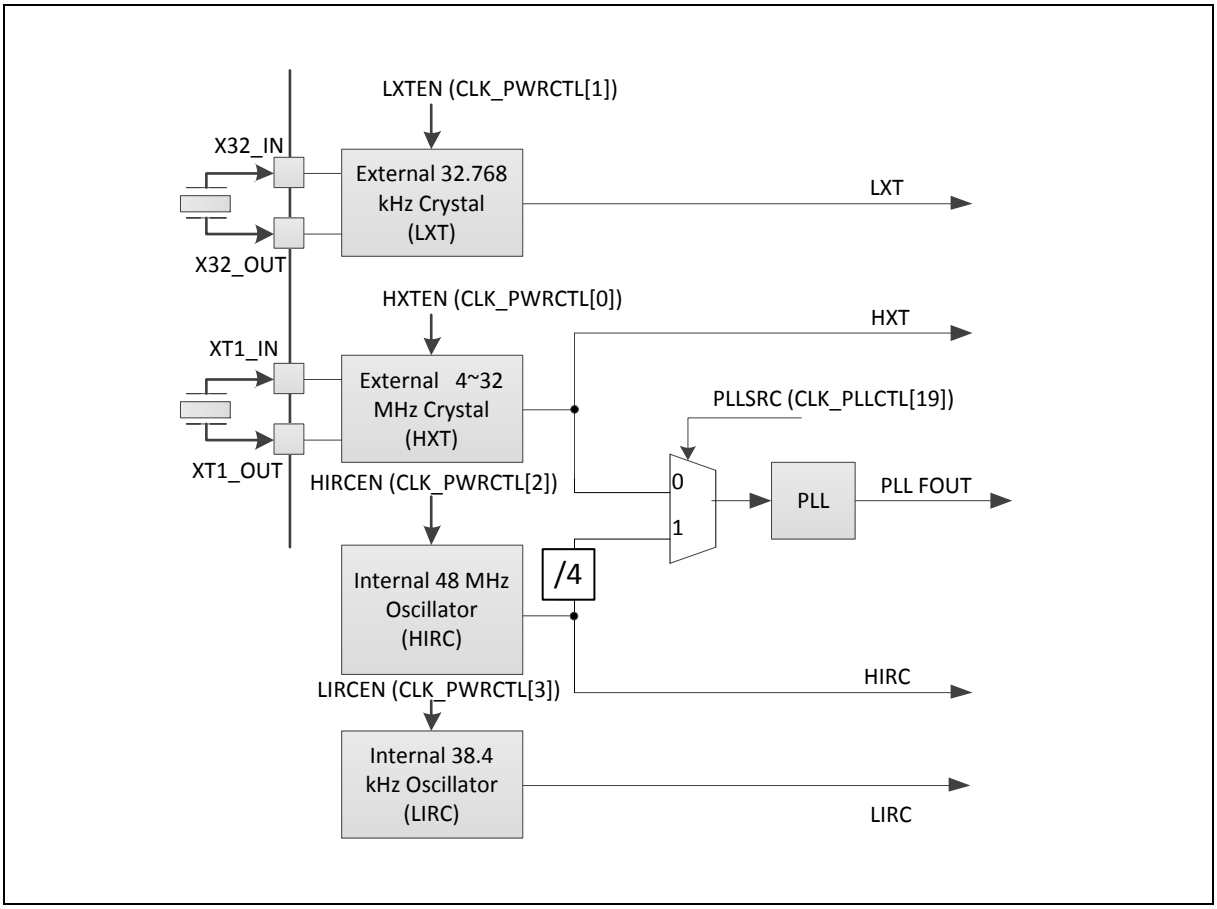


图 6.2-3 时钟发生器框图

6.2.3 系统时钟和SysTick时钟

系统时钟有5个时钟源，由时钟发生器产生。时钟源切换取决于寄存器HCLKSEL（CLK_CLKSEL0[2:0]）。其框图如下图 6.2-4所示。

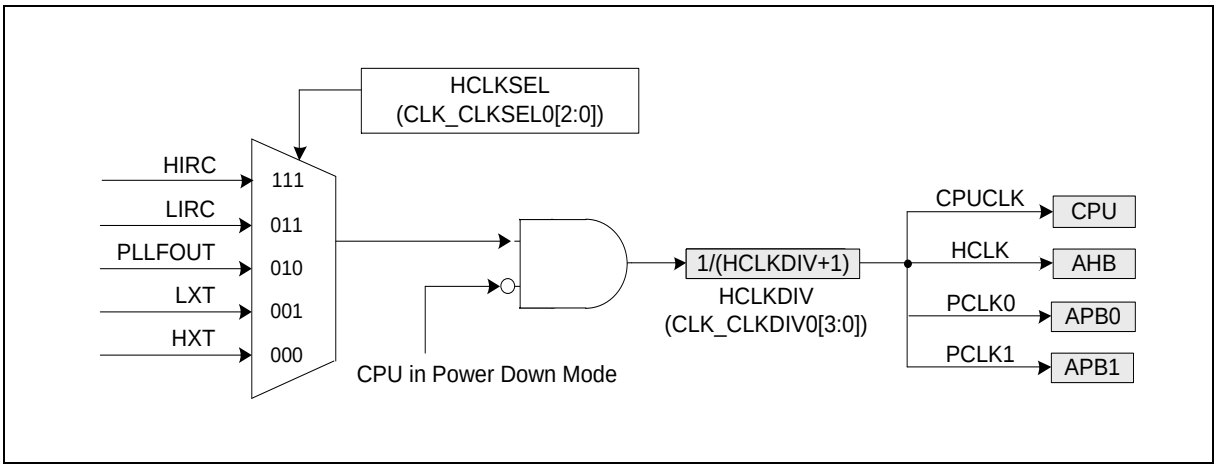


图 6.2-4 系统时钟框图

有两个时钟失败检测器来观察HXT和LXT时钟源的状况，而且均有独立的使能和中断控制。当HXT检测器使能时，HIRC时钟自动使能。当LXT检测器使能时，LIRC时钟自动使能。

当HXT时钟检测器使能，如果检测到HXT时钟停止且满足下面条件：系统的时钟源来自HXT或者系统时钟源来自PLL（PLL的输入时钟为HXT），系统时钟将自动切换到HIRC。如果检测到HXT时钟停止条件，HXTFIF （CLK_CLKDSTS[0]）将被设置为1，此时如果HXTFIE （CLK_CLKDCTL[5]）有置位，芯片将进入中断。用户可以试着去恢复HXT，通过禁止HXT和重新使能HXT来确认时钟稳定标志位是否已设置为1。如果HXT时钟稳定标志位有设置为1，这就意味着HXT在重新使能后已恢复振荡，此时用户可以重新把系统时钟切换到HXT。

HXT时钟停止检测和系统时钟切换到HIRC的过程如下图 6.2-5所示。

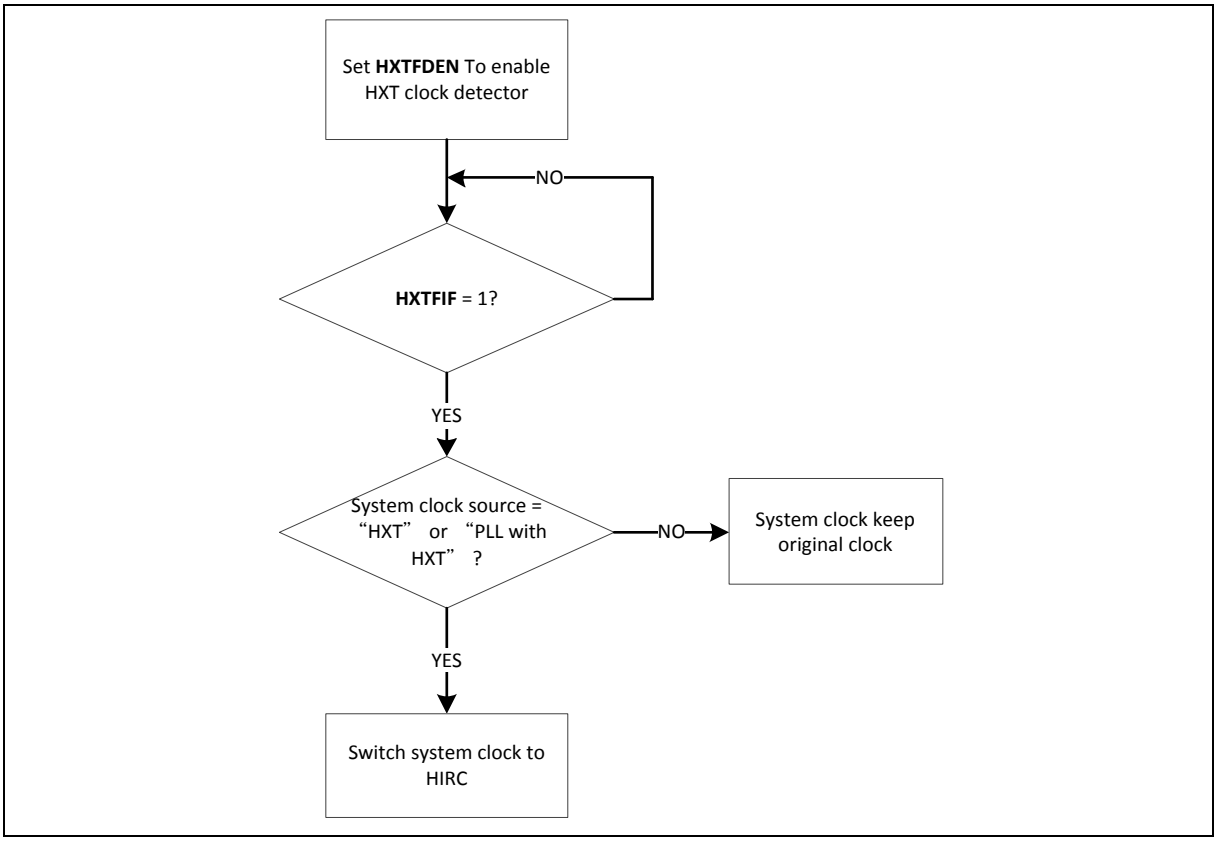


图 6.2-5 HXT 停止保护过程

当LXT时钟检测器使能，如果检测到LXT时钟停止且系统的时钟源来自LXT，系统时钟将自动切换到HIRC。如果检测到LXT时钟停止条件，HXTFIF （CLK_CLKDSTS[1]）将被设置为1，此时如果LXTFIEN （CLK_CLKDCTL[13]）有置位，芯片将进入中断。用户可以试着去恢复LXT，通过禁止LXT和重新使能LXT来确认时钟稳定标志位是否已设置为1。如果LXT时钟稳定标志位有设置为1，这就意味着LXT在重新使能后已恢复振荡，此时用户可以重新把系统时钟切换到LXT。

LXT时钟停止检测和系统时钟切换到LIRC的过程如下图 6.2-6所示。

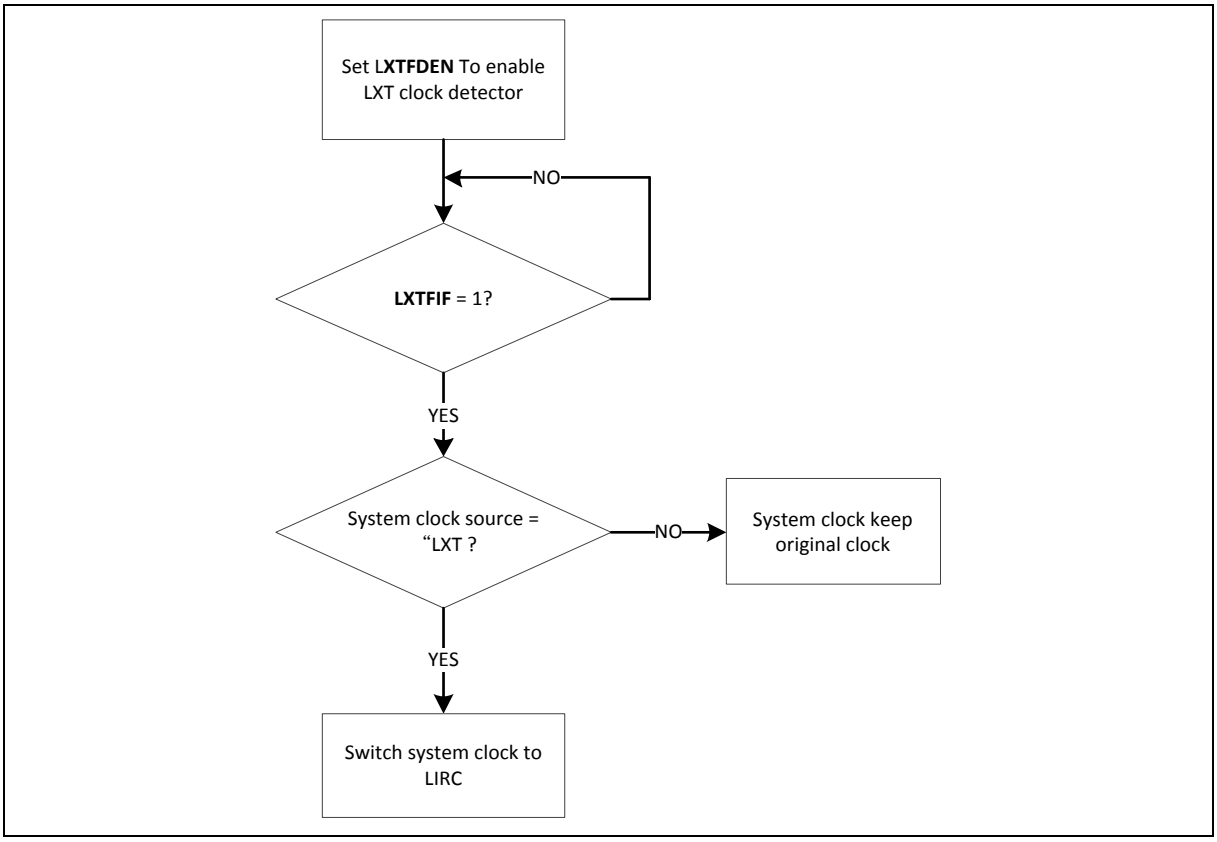


图 6.2-6 LXT 停止保护过程

Cortex®-M0内核的SysTick时钟源可以选择CPU时钟或外部时钟（SYST_CTRL[2]）。如果使用外部时钟，SysTick时钟（STCLK）有5个可选时钟源。时钟源切换取决于寄存器STCLKSEL（CLK_CLKSEL0[5:3]）。其框图如下图6.2-7所示。

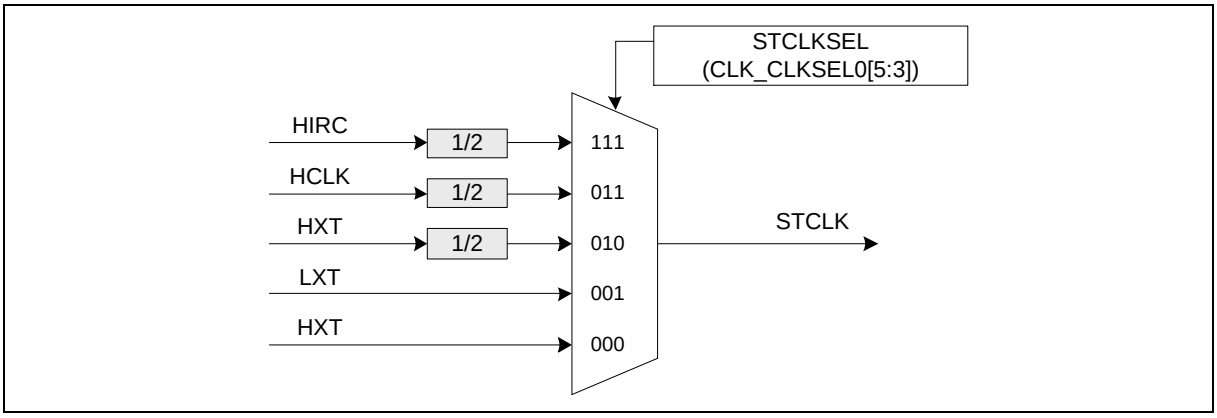


图 6.2-7 系统时钟控制框图

6.2.4 外设时钟

外设时钟可以有不同的时钟源做切换设置，取决于不同的外设。请参考6.2.9节中CLK_CLKSELx寄存器描述。

6.2.5 掉电模式时钟

当芯片进入掉电模式，系统时钟和一些时钟源以及一些外设时钟将被关闭。也有一些时钟源与外设时钟仍在工作。

如下时钟仍在工作：

- 时钟发生器
  - 38.4 kHz内部低速RC振荡器（LIRC）时钟
  - 32.768 kHz外部低速晶振（LXT）时钟
- 外设时钟（当模块的时钟源来自外部低速晶振LXT时钟或内部低速RC振荡器LIRC）

6.2.6 时钟输出

该设备带有一个2的若干次幂的频率分频器，该分频器由16个链式的二分频移位寄存器构成。其中哪一级的值被输出到CLKO功能引脚上，由一个16选1的多路转换器选择。因此共有16种时钟分频选择，分频范围从$F_{in}/2_1$ 到$F_{in}/2_{16}$，此处Fin是到时钟分频器的时钟输入频率。

输出公式：**$F_{out} = F_{in}/2^{(N+1)}$**，其中$F_{in}$为输入时钟频率，$F_{out}$为时钟分频器输出频率，N为FREQSEL（CLK_CLKOCTL[3:0]）中的4位值。

当往CLKOEN （CLK_CLKOCTL[4]）写1，分级计数器开始计数。往CLKOEN（CLK_CLKOCTL[4]）写0，分级计数器持续计数，直到分频时钟达到低电平并会保持在低电平状态。

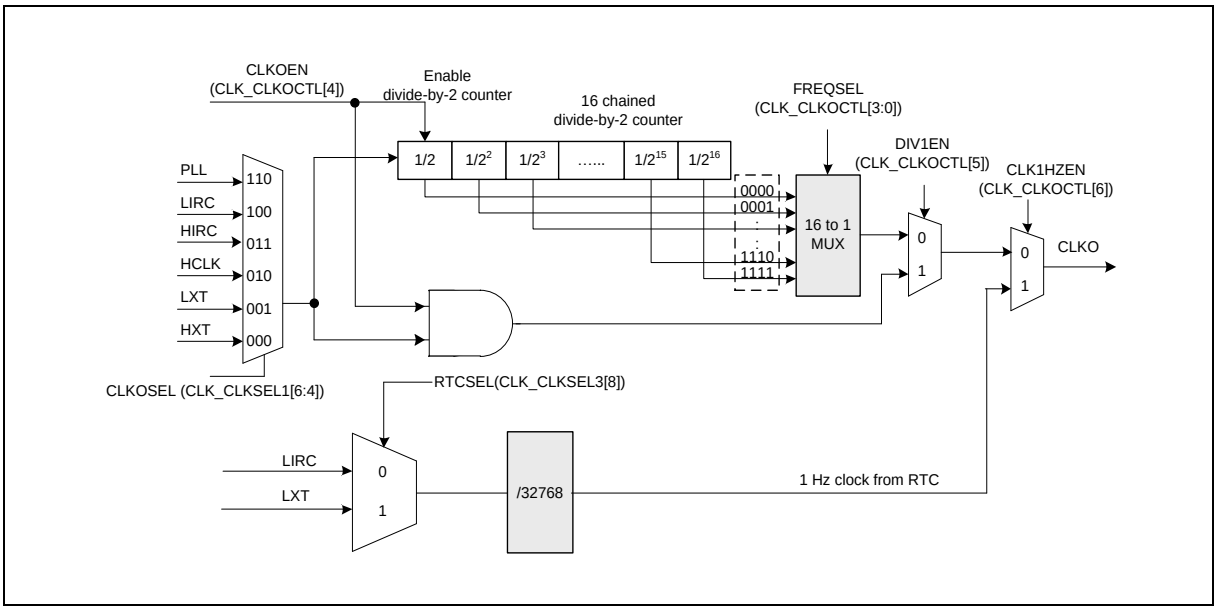


图 6.2-8 时钟输出框图

6.2.7 USB时钟源

USBD的时钟源来自48 Mhz HIRC产生或者可编程PLL的输出。产生的时钟如下图 6.2-9所示。

USBDIV是时钟分频器输出频率，输出公式是

（PLLFOUT 频率） / （USBDIV + 1）

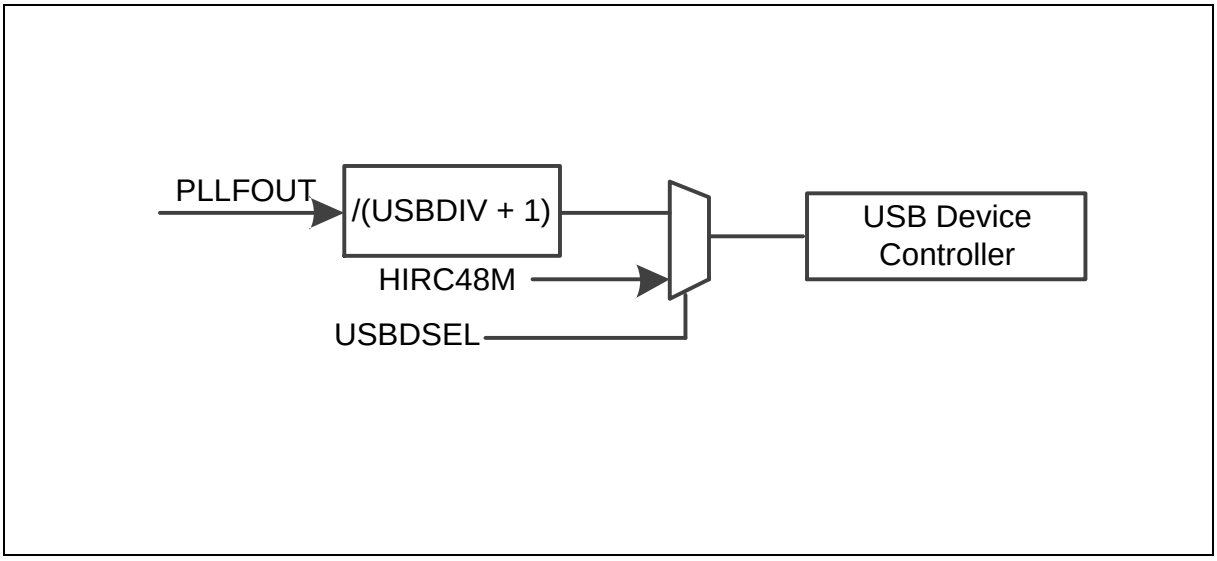


图 6.2-9 USBD 时钟源

6.2.8 寄存器映射

R: 只读，**W:** 只写，**R/W:** 可读写

寄存器	偏移	R/W	描述	复位值
CLK基地址： **CLK_BA = 0x4000_0200**				
CLK_PWRCTL	CLK_BA+0x00	R/W	系统掉电控制寄存器	0x0231_001X
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB设备时钟使能控制寄存器	0x0000_0004
CLK_APBCLK0	CLK_BA+0x08	R/W	APB设备时钟使能控制寄存器0	0x0000_0001
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB设备时钟使能控制寄存器1	0x0000_0000
CLK_CLKSEL0	CLK_BA+0x10	R/W	时钟源选择控制寄存器0	0x0000_003F
CLK_CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0x4477_773B
CLK_CLKSEL2	CLK_BA+0x18	R/W	时钟源选择控制寄存器2	0x0020_032B
CLK_CLKSEL3	CLK_BA+0x1C	R/W	时钟源选择控制寄存器3	0x4444_4400
CLK_CLKDIV0	CLK_BA+0x20	R/W	时钟分频数目寄存器0	0x0000_0000
CLK_CLKDIV4	CLK_BA+0x30	R/W	时钟分频数目寄存器4	0x0000_0000
CLK_PCLKDIV	CLK_BA+0x34	R/W	APB时钟分频器寄存器	0x0000_0000
CLK_PLLCTL	CLK_BA+0x40	R/W	PLL控制寄存器	0x0005_C25E
CLK_STATUS	CLK_BA+0x50	R	时钟状态监测寄存器	0x0000_00XX
CLK_CLKOCTL	CLK_BA+0x60	R/W	时钟输出控制寄存器	0x0000_0000
CLK_CLKDCTL	CLK_BA+0x70	R/W	时钟失败检测器控制寄存器	0x0000_0000
CLK_CLKDSTS	CLK_BA+0x74	R/W	时钟失败检测器状态寄存器	0x0000_0000
CLK_CDUPB	CLK_BA+0x78	R/W	时钟频率检测器上边界寄存器	0x0000_0000
CLK_CDLOWB	CLK_BA+0x7c	R/W	时钟频率检测器下边界寄存器	0x0000_0000
CLK_LDOCTL	CLK_BA+0x80	R/W	LDO控制寄存器	0x0000_0000
CLK_HXTFSEL	CLK_BA+0xB4	R/W	HXT滤波器选择控制寄存器	0x0000_0000

6.2.9 寄存器描述

系统掉电控制寄存器 （CLK_PWRCTL）

寄存器	偏移	R/W	描述	复位值
CLK_PWRCTL	CLK_BA+0x00	R/W	系统掉电控制寄存器	0x0231_001X

31	30	29	28	27	26	25	24
Reserved					LXTGAIN		LXTSELXT
23	22	21	20	19	18	17	16
Reserved	HXTGAIN			Reserved			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDEN	PDWKIF	PDWKIEN	PDWKDLY	LIRCEN	HIRCEN	LXTEN	HXTEN

位	描述	
[31:27]	**Reserved**	保留
[26:25]	**LXTGAIN**	**LXT增益控制位（写保护）** 00 = LXT晶振ESR = 35K, CL=12.5pF 10 = LXT晶振ESR = 70K, CL=12.5pF **注：**此位写保护。参考SYS_REGLCTL寄存器。
[24]	**LXTSELXT**	**LXT模式选择** 0 = LXT工作在晶振模式。PF.4和PF.5被配置为外部低速晶振引脚（LXT） 1 = LXT工作在外部时钟模式。PF.5被配置为外部时钟输入脚 **注1：**当LXTSELXT = 1，PF.5 MFP 必须被配置为GPIO模式。X32_IN直流电源特性与GPIO相同。 **注 2：**此位写保护。参考SYS_REGLCTL寄存器。
[22:20]	**HXTGAIN**	**HXT增益控制位（写保护）** 该位写保护，操作前需要进行解锁操作。 增益控制用来放大晶振的增益，来确保晶振工作正常。增益控制使能时比增益关闭时，晶振需消耗更多的能量。 000 = HXT 频率低于4 MHz 001 = HXT 频率在4 MHz 到8 MHz之间 010 = HXT 频率在 8 MHz 到12 MHz之间 011 = HXT 频率在 12 MHz 到16 MHz之间 100 = HXT 频率在 16 MHz 到24 MHz之间 111 = HXT 频率在 24 MHz 到32 MHz之间 其他：保留

		注：此位写保护。参考SYS_REGLCTL寄存器。
[19:8]	**Reserved**	保留
[7]	**PDEN**	**系统掉电使能位 （写保护）** 当该位置1后，掉电模式使能，芯片持续运行，直到CPU睡眠模式激活，芯片将进入掉电模式。 当芯片从掉电模式唤醒，该位将自动清0。为了进入下一次睡眠模式，用户必须重新将该位置1。 在掉电模式，HXT和HIRC将被禁用，但LXT和LIRC不由掉电模式控制。如果用户在进入掉电模式前禁用LIRC，则该位应在LIRC禁用50us后设置。 在掉电模式，PLL和系统时钟将禁用，并且忽略时钟源选择。如果外设的时钟源来自LXT或LIRC，外设时钟不被掉电模式控制。 0 = 执行WFI指令后,芯片运行在正常模式或空闲模式 1 = 芯片立即进入掉电模式，或等待CPU睡眠指令WFI **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[6]	**PDWKIF**	**掉电模式唤醒中断状态** 由“掉电唤醒事件”置位，表明MCU从掉电模式下恢复。 如果出现任何唤醒源，则置位，参考电源模式和唤醒源一章。 **注1:** 该位写1清0。 **注2:** 只有在PDWKIEN （CLK_PWRCTL[5]）设置为1后,该位才工作。
[5]	**PDWKIEN**	**掉电模式唤醒中断使能位（写保护）** 0 = 掉电模式唤醒中断禁用 1 = 掉电模式唤醒中断使能 **注1:** 当PDWKIF和PDWKIEN都为高时产生唤醒中断。 **注2:** 此位写保护。参考SYS_REGLCTL 寄存器。
[4]	**PDWKDLY**	**使能唤醒延时计数器（写保护）** 当芯片从掉电模式唤醒后，时钟控制将延时若干时钟周期来等待系统时钟稳定。 如果芯片工作在外部高速晶振（HXT），延时时钟周期为4960个时钟周期，如果芯片工作在内部高速RC振荡器（HIRC），延时时钟周期为512个时钟周期。 0 = 时钟周期延时禁用 1 = 时钟周期延时使能 **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[3]	**LIRCEN**	**LIRC 使能位 （写保护）** 0 = 内部低速RC振荡器（LIRC）禁止 1 = 内部低速RC振荡器（LIRC）使能 **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[2]	**HIRCEN**	**HIRC 使能位 （写保护）** 0 = 内部高速RC振荡器（HIRC）禁止 1 = 内部高速RC振荡器（HIRC）使能 注: 此位写保护。参考SYS_REGLCTL寄存器。
[1]	**LXTEN**	**LXT 使能位 （写保护）** 0 = 外部低速晶振（LXT）禁止 1 = 外部低速晶振（LXT）使能 **注1：**通过上电复位复位。 **注2：**此位写保护。参考SYS_REGLCTL 寄存器。

[0]	**HXTEN**	**HXT 使能位 （写保护）** 0 = 外部高速晶振（HXT）禁止 1 = 外部高速晶振（HXT）使能 **注1**：通过上电复位复位。 **注2**：此位写保护. 参考SYS_REGLCTL 寄存器。

AHB设备时钟使能控制寄存器（CLK_AHBCLK）

此寄存器中的位用于使能/禁止系统时钟、AHB总线设备的时钟。

寄存器	偏移	R/W	描述	复位值
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB设备时钟使能控制寄存器	0x0000_0004

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

位	描述	
[31:21]	**Reserved**	保留
[20]	**Reserved**	保留
[19:8]	**Reserved**	保留
[7]	**CRCCKEN**	**CRC 发生器控制器时钟使能位** 0 = CRC 外设时钟禁止 1 = CRC 外设时钟使能
[6:5]	**Reserved**	保留
[4]	**HDIV_EN**	**硬件除法器控制器时钟使能位** 0 = 硬件除法器外设时钟禁止 1 = 硬件除法器外设时钟使能
[3]	**EBICKEN**	**EBI 控制器时钟使能位** 0 = EBI 外设时钟禁止 1 = EBI 外设时钟使能
[2]	**ISPCKEN**	**Flash ISP 控制器时钟使能位** 0 = Flash ISP 外设时钟禁止 1 = Flash ISP 外设时钟使能
[1]	**PDMACKEN**	**PDMA 控制器时钟使能位** 0 = PDMA 外设时钟禁止 1 = PDMA 外设时钟使能
[0]	**Reserved**	保留

APB设备时钟使能控制寄存器0 （CLK_APBCLK0）

此寄存器中的位用于使能/禁止外部接口控制器的时钟。

寄存器	偏移	R/W	描述	复位值
CLK_APBCLK0	CLK_BA+0x08	R/W	APB设备时钟使能控制寄存器0	0x0000_0001

31	30	29	28	27	26	25	24
Reserved			ADCCKEN	USBDCKEN	Reserved		
23	22	21	20	19	18	17	16
UART7CKEN	UART6CKEN	UART5CKEN	UART4CKEN	UART3CKEN	UART2CKEN	UART1CKEN	UART0CKEN
15	14	13	12	11	10	9	8
Reserved		SPI0CKEN	QSPI0CKEN	Reserved		I2C1CKEN	I2C0CKEN
7	6	5	4	3	2	1	0
ACMP01CKEN	CLKOCKEN	TMR3CKEN	TMR2CKEN	TMR1CKEN	TMR0CKEN	RTCCKEN	WDTCKEN

位	描述	
[31:29]	**Reserved**	保留
[28]	**ADCCKEN**	**模拟数字转换 （ADC） 时钟使能位** 0 = ADC 时钟禁止 1 = ADC 时钟使能
[27]	**USBDCKEN**	**USB 设备时钟使能位** 0 = USB 设备时钟禁止 1 = USB 设备时钟使能
[26:24]	**Reserved**	保留
[23]	**UART7CKEN**	**UART7时钟使能位** 0 = UART7时钟禁止 1 = UART7时钟使能
[22]	**UART6CKEN**	**UART6时钟使能位** 0 = UART6时钟禁止 1 = UART6时钟使能
[21]	**UART5CKEN**	**UART5时钟使能位** 0 = UART5时钟禁止 1 = UART5时钟使能
[20]	**UART4CKEN**	**UART4时钟使能位** 0 = UART4时钟禁止 1 = UART4时钟使能
[19]	**UART3CKEN**	**UART3时钟使能位**

		0 = UART3时钟禁止 1 = UART3时钟使能
[18]	**UART2CKEN**	**UART2时钟使能位** 0 = UART2时钟禁止 1 = UART2时钟使能
[17]	**UART1CKEN**	**UART1时钟使能位** 0 = UART1时钟禁止 1 = UART1时钟使能
[16]	**UART0CKEN**	**UART0时钟使能位** 0 = UART0时钟禁止 1 = UART0时钟使能
[15:14]	**Reserved**	保留
[13]	**SPI0CKEN**	**SPI0时钟使能位** 0 = SPI0时钟禁止 1 = SPI0时钟使能
[12]	**QSPI0CKEN**	**QSPI0时钟使能位** 0 = QSPI0时钟禁止 1 = QSPI0时钟使能
[11:10]	**Reserved**	保留
[9]	**I2C1CKEN**	**I2C1时钟使能位** 0 = I2C1时钟禁止 1 = I2C1时钟使能
[8]	**I2C0CKEN**	**I2C0时钟使能位** 0 = I2C0时钟禁止 1 = I2C0时钟使能
[7]	**ACMP01CKEN**	**模拟比较器0/1时钟使能位** 0 = 模拟比较器0/1时钟禁止 1 = 模拟比较器0/1时钟使能
[6]	**CLKOCKEN**	**CLKO时钟使能位** 0 = CLKO时钟禁止 1 = CLKO时钟使能
[5]	**TMR3CKEN**	**Timer3时钟使能位** 0 = Timer3时钟禁止 1 = Timer3时钟使能
[4]	**TMR2CKEN**	**Timer2时钟使能位** 0 = Timer2时钟禁止 1 = Timer2时钟使能
[3]	**TMR1CKEN**	**Timer1时钟使能位** 0 = Timer1时钟禁止 1 = Timer1时钟使能

[2]	**TMR0CKEN**	**Timer0时钟使能位** 0 = Timer0时钟禁止 1 = Timer0时钟使能
[1]	**RTCCKEN**	**RTC时钟使能位** 0 = RTC时钟禁止 1 = RTC时钟使能
[0]	**WDTCKEN**	**看门狗定时器时钟使能位（写保护）** 0 = 看门狗时钟禁止 1 = 看门狗时钟使能 **注1：**此位写保护。参考SYS_REGLCTL寄存器。 **注2：**此位将被开机复位，看门狗复位或芯片软件复位置位。

APB设备时钟使能控制寄存器1 （CLK_APBCLK1）

此寄存器中的位用于使能/禁止外部接口控制器的时钟。

寄存器	偏移	R/W	描述	复位值
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB设备时钟使能控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**BPWM1CKEN**	**BPWM0CKEN**	**PWM1CKEN**	**PWM0CKEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**USCI1CKEN**	**USCI0CKEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

位	描述	
[31:20]	**Reserved**	保留
[19]	**BPWM1CKEN**	**BPWM1时钟使能位** 0 = BPWM1时钟禁止 1 = BPWM1时钟使能
[18]	**BPWM0CKEN**	**BPWM0时钟使能位** 0 = BPWM0时钟禁止 1 = BPWM0时钟使能
[17]	**PWM1CKEN**	**PWM1时钟使能位** 0 = PWM1时钟禁止 1 = PWM1时钟使能
[16]	**PWM0CKEN**	**PWM0时钟使能位** 0 = PWM0时钟禁止 1 = PWM0时钟使能
[15:10]	**Reserved**	保留
[9]	**USCI1CKEN**	**USCI1时钟使能位** 0 = USCI1时钟禁止 1 = USCI1时钟使能
[8]	**USCI0CKEN**	**USCI0时钟使能位** 0 = USCI0时钟禁止 1 = USCI0时钟使能
[7:0]	**Reserved**	保留

时钟源选择控制寄存器0 （CLK_CLKSEL0）

寄存器	偏移	R/W	描述	复位值
CLK_CLKSEL0	CLK_BA+0x10	R/W	时钟源选择控制寄存器0	0x0000_003F

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

位	描述	
[31:9]	**Reserved**	保留
[8]	**USBDSEL**	**USB 设备时钟源选择位 （写保护）** 这些位是被保护位。意味着编程此位需要写“59h”、“16h”、“88h”到地址0x4000_0100以禁用寄存器保护。请参考地址GCR_BA+0x100处的寄存器REGWRPROT。 0 = 时钟源来自HIRC 1 = 时钟源来自PLL分频器 **注：**如果不支持PLL功能，时钟源选项“1”将被改为HIRC。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[7:6]	**Reserved**	保留
[5:3]	**STCLKSEL**	**Cortex®-M0 SysTick时钟源选择（写保护）** 如果SYST_CTRL[2]=0，SysTick使用时钟源如下所列。 000 = 时钟源来自HXT 001 = 时钟源来自LXT 010 = 时钟源来自HXT/2 011 = 时钟源来自HCLK/2 111 = 时钟源来自HIRC/2 其他 = 保留 **注：**如果 SysTick时钟源不是来自HCLK（例如 SYST_ CTRL [2] = 0），SysTick时钟源必须小于或等于HCLK/2。 **注：**此位写保护。参考SYS_REGLCTL寄存器。 **注：**如果不支持LXT或HXT，时钟源选项“000”，“001”或“010”将被停止。 请参照3.2章节NuMicro® M031/M032系列选型指南的详细信息。
[2:0]	**HCLKSEL**	**HCLK时钟源选择（写保护）** 切换始终前，相关时钟源（预选和新选）必须打开。 000 = 时钟源来自HXT

		001 = 时钟源来自LXT 010 = 时钟源来自PLL 011 = 时钟源来自LIRC 111= 时钟源来自HIRC 其他 = 保留 **注：**此位写保护。参考SYS_REGLCTL寄存器。 **注：**随上电复位复位 **注：**如果不支持PLL，时钟源选择“010” 将被改为HIRC。 **注：**如果不支持LXT或HXT，时钟源选择“000”或“001” 将保持前一个时钟选择。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

时钟源选择控制寄存器1 （CLK_CLKSEL1）

在时钟切换之前，必须打开相关的时钟源（预选和新选）。

寄存器	偏移	R/W	描述	复位值
CLK_CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0x4477_773B

31	30	29	28	27	26	25	24
Reserved	**UART1SEL**			**Reserved**	**UART0SEL**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**TMR3SEL**			**Reserved**	**TMR2SEL**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TMR1SEL**			**Reserved**	**TMR0SEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CLKOSEL**			**WWDTSEL**		**WDTSEL**	

位	描述	
[31]	**Reserved**	保留
[30:28]	**UART1SEL**	**UART1时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK1 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK1。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[27]	**Reserved**	保留
[26:24]	**UART0SEL**	**UART0时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK0 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK0。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

[23]	**Reserved**	保留
[22:20]	**TMR3SEL**	**TIMER3时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自外部低速晶振（LXT） 010 = 时钟源来自PCLK1 011 = 时钟源来自外部时钟TM3引脚 101 = 时钟源来自内部低速RC振荡器（LIRC） 111 = 时钟源来自内部高速RC振荡器（HIRC） 其他 = 保留 **注：**如果不支持LXT或HXT，时钟源选择“000” 或“001” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[19]	**Reserved**	保留
[18:16]	**TMR2SEL**	**TIMER2时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自外部低速晶振（LXT） 010 = 时钟源来自PCLK1 011 = 时钟源来自外部时钟TM2引脚 101 = 时钟源来自内部低速RC振荡器（LIRC） 111 = 时钟源来自内部高速RC振荡器（HIRC） 其他 = 保留 **注：**如果不支持LXT或HXT，时钟源选择“000” 或“001” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[15]	**Reserved**	保留
[14:12]	**TMR1SEL**	**TIMER1时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自外部低速晶振（LXT） 010 = 时钟源来自PCLK0 011 = 时钟源来自外部时钟TM1引脚 101 = 时钟源来自内部低速RC振荡器（LIRC） 111 = 时钟源来自内部高速RC振荡器（HIRC） 其他 = 保留 **注：**如果不支持LXT或HXT，时钟源选择“000” 或“001” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[11]	**Reserved**	保留
[10:8]	**TMR0SEL**	**TIMER0时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自外部低速晶振（LXT） 010 = 时钟源来自PCLK0 011 = 时钟源来自外部时钟TM0引脚 101 = 时钟源来自内部低速RC振荡器（LIRC） 111 = 时钟源来自内部高速RC振荡器（HIRC） 其他 = 保留

		注：如果不支持LXT或HXT，时钟源选择“000” 或“001” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[7]	**Reserved**	保留
[6:4]	**CLKOSEL**	**除频器时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自外部低速晶振（LXT） 010 = 时钟源来自HCLK. 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自内部低速RC振荡器（LIRC） 101 = 时钟源来自内部高速RC振荡器（HIRC） 110 = 时钟源来自PLL **注：**如果不支持PLL，时钟源选择“110” 将被改为HIRC。 **注：**如果不支持LXT或HXT，时钟源选择“000” 或“001” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[3:2]	**WWDTSEL**	**窗口看门狗时钟源选择（写保护）** 10 = 时钟源来自HCLK/2048. 11 = 时钟源来自 内部低速RC振荡器（LIRC）. 其他 = 保留
[1:0]	**WDTSEL**	**看门狗时钟源选择（写保护）** 00 = 保留 01 = 时钟源来自外部低速晶振（LXT） 10 = 时钟源来自HCLK/2048 11 = 时钟源来自内部低速RC振荡器（LIRC） **注：** 此位写保护。参考SYS_REGLCTL寄存器。 2. 如果CONFIG0[31], CONFIG0[4], CONFIG0[3] 全为1，此位将被强制置1。 **注：**如果不支持LXT，时钟源选择“01” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

时钟源选择控制寄存器2 （CLK_CLKSEL2）

在时钟切换之前，必须打开相关的时钟源（预选和新选）。

寄存器	偏移	R/W	描述	复位值
CLK_CLKSEL2	CLK_BA+0x18	R/W	时钟源选择控制寄存器2	0x0020_032B

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**ADCSEL**		**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**BPWM1SEL**	**BPWM0SEL**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**SPI0SEL**		**QSPI0SEL**		**PWM1SEL**	**PWM0SEL**

位	描述	
[31:22]	**Reserved**	保留
[21:20]	**ADCSEL**	**ADC时钟源选择** 00 = 时钟源来自外部高速晶振（HXT） 01 = 时钟源来自PLL 10 = 时钟源来自PCLK1 11 = 时钟源来自内部高速RC振荡器（HIRC）时钟 **注：**如果不支持PLL，时钟源选择“01” 将被改为PCLK1。 **注：**如果不支持HXT，时钟源选择“00” 将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[19:10]	**Reserved**	保留
[9]	**BPWM1SEL**	**BPWM1时钟源选择** BPWM1外设时钟源被定义为BPWM1SEL。 0 = 时钟源来自PLL 1 = 时钟源来自PCLK1 **注：**如果不支持PLL，时钟源选择“0” 将被改为PCLK1。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[8]	**BPWM0SEL**	**BPWM0时钟源选择** BPWM0外设时钟源被定义为BPWM0SEL。 0 = 时钟源来自PLL 1 = 时钟源来自PCLK0 **注：**如果不支持PLL，时钟源选择“0” 将被改为PCLK0。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[7:6]	**Reserved**	保留

[5:4]	**SPI0SEL**	**SPI0时钟源选择** 00 = 时钟源来自外部高速晶振（HXT） 01 = 时钟源来自PLL 10 = 时钟源来自PCLK1 11 = 时钟源来自内部高速RC振荡器（HIRC） **注：**如果不支持PLL，时钟源选择“01” 将被改为PCLK1。 **注：**如果不支持HXT，时钟源选择“00” 将停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[3:2]	**QSPI0SEL**	**QSPI0时钟源选择** 00 = 时钟源来自外部高速晶振（HXT） 01 = 时钟源来自PLL 10 = 时钟源来自PCLK0 11 = 时钟源来自内部高速RC振荡器（HIRC） **注：**如果不支持PLL，时钟源选择“01” 将被改为PCLK0。 **注：**如果不支持HXT，时钟源选择“00” 将停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[1]	**PWM1SEL**	**PWM1时钟源选择** PWM1外设时钟源被定义为PWM1SEL。 0 = 时钟源来自PLL 1 = 时钟源来自PCLK1 **注：**如果不支持PLL，时钟源选择“0” 将被改为PCLK1。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[0]	**PWM0SEL**	**PWM0时钟源选择** PWM0外设时钟源被定义为PWM0SEL。 0 = 时钟源来自PLL 1 = 时钟源来自PCLK0 **注：**如果不支持PLL，时钟源选择“0” 将被改为PCLK0。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

时钟源选择控制寄存器3 （CLK_CLKSEL3）

在时钟切换之前，必须打开相关的时钟源（预选和新选）。

寄存器	偏移	R/W	描述	复位值
CLK_CLKSEL3	CLK_BA+0x1C	R/W	时钟源选择控制寄存器3	0x4444_4400

31	30	29	28	27	26	25	24
Reserved	**UART3SEL**			**Reserved**	**UART2SEL**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**UART5SEL**			**Reserved**	**UART4SEL**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**UART7SEL**			**Reserved**	**UART6SEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

位	描述	
[31]	**Reserved**	保留
[30:28]	**UART3SEL**	**UART3时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK1 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK1。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[27]	**Reserved**	保留
[26:24]	**UART2SEL**	**UART2时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK0 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK0。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

[23]	**Reserved**	保留
[22:20]	**UART5SEL**	**UART5时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK1 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK1。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[19]	**Reserved**	保留
[18:16]	**UART4SEL**	**UART4时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK0 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK0。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[15]	**Reserved**	保留
[14:12]	**UART7SEL**	**UART7时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC） 100 = 时钟源来自PCLK1 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK1。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[11]	**Reserved**	保留
[10:8]	**UART6SEL**	**UART6时钟源选择** 000 = 时钟源来自外部高速晶振（HXT） 001 = 时钟源来自PLL 010 = 时钟源来自外部低速晶振（LXT） 011 = 时钟源来自内部高速RC振荡器（HIRC）

		100 = 时钟源来自PCLK0 101 = 时钟源来自内部低速RC振荡器（LIRC） 其他 = 保留 **注：**如果不支持PLL，时钟源选项“001”将被改为PCLK0。 **注：**如果不支持LXT或HXT，时钟源选项“000”或“001”将被停止。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[7:0]	**Reserved**	保留

时钟分频数目寄存器0 （CLK_CLKDIV0）

寄存器	偏移	R/W	描述	复位值
CLK_CLKDIV0	CLK_BA+0x20	R/W	时钟分频数目寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADCDIV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART1DIV				**UART0DIV**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBDIV				**HCLKDIV**			

位	描述	
[31:24]	**Reserved**	保留
[23:16]	**ADCDIV**	**ADC时钟源时钟除频数** ADC时钟频率 = （ADC时钟源时钟频率） / （ADCDIV + 1）
[15:12]	**UART1DIV**	**UART1时钟源时钟除频数** UART1时钟频率 = （UART1时钟源时钟频率） / （UART1DIV + 1）
[11:8]	**UART0DIV**	**UART0时钟源时钟除频数** UART0时钟频率 = （UART0时钟源时钟频率） / （UART0DIV + 1）
[7:4]	**USBDIV**	**USB时钟源时钟除频数** USB时钟频率 = （PLL频率） / （USBDIV + 1）
[3:0]	**HCLKDIV**	**HCLK时钟源时钟除频数** HCLK时钟频率 = （HCLK时钟源时钟频率） / （HCLKDIV + 1）

时钟分频数目寄存器4 （CLK_CLKDIV4）

寄存器	偏移	R/W	描述	复位值
CLK_CLKDIV4	CLK_BA+0x30	R/W	时钟分频数目寄存器4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
UART7DIV				**UART6DIV**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART5DIV				**UART4DIV**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART3DIV				**UART2DIV**			

位	描述	
[31:4]	**Reserved**	保留
[23:20]	**UART7DIV**	**UART7时钟源时钟除频数** UART7时钟频率 = （UART7时钟源时钟频率） / （UART7DIV + 1）
[19:16]	**UART6DIV**	**UART6时钟源时钟除频数** UART6时钟频率 = （UART6时钟源时钟频率） / （UART6DIV + 1）
[15:12]	**UART5DIV**	**UART5时钟源时钟除频数** UART5时钟频率 = （UART5时钟源时钟频率） / （UART5DIV + 1）
[11:8]	**UART4DIV**	**UART4时钟源时钟除频数** UART4时钟频率 = （UART4时钟源时钟频率） / （UART4DIV + 1）
[7:4]	**UART3DIV**	**UART3时钟源时钟除频数** UART3时钟频率 = （UART3时钟源时钟频率） / （UART3DIV + 1）
[3:0]	**UART2DIV**	**UART2时钟源时钟除频数** UART2时钟频率 = （UART2时钟源时钟频率） / （UART2DIV + 1）

APB时钟分频器寄存器（CLK_PCLKDIV）

寄存器	偏移	R/W	描述	复位值
CLK_PCLKDIV	CLK_BA+0x34	R/W	APB时钟分频器寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**APB1DIV**			**Reserved**	**APB0DIV**		

位	描述	
[31:7]	**Reserved**	保留
[6:4]	**APB1DIV**	**APB1时钟分频器** APB1时钟可来自HCLK分频 000: PCLK1 = HCLK 001: PCLK1 = 1/2 HCLK 010: PCLK1 = 1/4 HCLK 011: PCLK1 = 1/8 HCLK 100: PCLK1 = 1/16 HCLK 其他：保留
[3]	**Reserved**	保留
[2:0]	**APB0DIV**	**APB0时钟分频器** APB0时钟可来自HCLK分频 000: PCLK0 = HCLK 001: PCLK0 = 1/2 HCLK 010: PCLK0 = 1/4 HCLK 011: PCLK0 = 1/8 HCLK 100: PCLK0 = 1/16 HCLK 其他：保留

PLL控制寄存器（CLK_PLLCTL）

PLL参考时钟输入来自于4~32 MHz外部高速晶振（HXT）时钟输入或来自于48 MHz 内部高速振荡器（HIRC/4）。此寄存器被用来控制PLL输出频率和PLL运行模式。

编程这些位需要写“59h”、“16h”、“88h”到地址0x4000_0100以禁用寄存器保护。请参考地址GCR_BA+0x100处的寄存器REGWRPROT。

CLK_PLLCTL只能当 PD（CLK_PLLCTL[16]） 置1时修改。

寄存器	偏移	R/W	描述	复位值
CLK_PLLCTL	CLK_BA+0x40	R/W	PLL控制寄存器	0x0005_C25E

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
STBSEL	**Reserved**			**PLLSRC**	**OE**	**BP**	**PD**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
OUTDIV		**INDIV**					**FBDIV**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FBDIV							

位	描述	
[31:24]	**Reserved**	保留
[23]	**STBSEL**	**PLL稳定计数选择 （写保护）** 0 = PLL 稳定时间为6144个PLL时钟源时钟（适合于时钟源频率小于或等于12 MHz ） 1 = PLL 稳定时间为16128个PLL时钟源时钟（适合于时钟源频率大于12 MHz） **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[22:20]	**Reserved**	保留
[19]	**PLLSRC**	**PLL 时钟源选择（写保护）** 0 = PLL时钟源来自外部高速晶振（HXT） 1 = PLL时钟源来自 48 MHz 内部高速震荡时钟（HIRC/4） **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[18]	**OE**	**PLL OE （FOUT 使能）引脚控制（写保护）** 0 = PLL FOUT使能 1 = PLL FOUT被固定为低 **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[17]	**BP**	**PLL 旁路控制（写保护）** 0 = PLL在正常模式（默认） 1 = PLL时钟输出和PLL输入时钟FIN相同 **注:** 此位写保护。参考SYS_REGLCTL寄存器。

[16]	**PD**	**掉电模式（写保护）** 如果在CLK_PWRCTL寄存器设置PDEN位为1，PLL也将进入掉电模式。 0 = PLL在正常模式 1 = PLL在掉电模式（默认） **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[15:14]	**OUTDIV**	**PLL 输出分频控制（写保护）** 请参考下表公式。 **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[13:9]	**INDIV**	**PLL 输入分频控制（写保护）** 请参考下表公式。 **注:** 此位写保护。参考SYS_REGLCTL寄存器。
[8:0]	**FBDIV**	**PLL反馈除频控制器（写保护）** 请参考下表公式。 **注:** 此位写保护。参考SYS_REGLCTL寄存器。

输出时钟频率设置

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

约束条件:

1. $3.2MHz < FIN < 150MHz$
2. $800kHz < \frac{FIN}{2 * NR} < 8MHz$
3. $200MHz < FCO = FIN * \frac{NF}{NR} < 500MHz,$
   $FCO > 250MHz \ \ is \ \ preferred$

符号	描述
FOUT	输出时钟频率
FIN	输入（参考）时钟频率
NR	输入除频（INDIV + 2）
NF	反馈除频（FBDIV + 2）
NO	OUTDIV = “00” : NO = 1 OUTDIV = “01” : NO = 2 OUTDIV = “10” : NO = 2 OUTDIV = “11” : NO = 4

表 6.2-1 PLL 输出频率公式符号定义

时钟状态监测寄存器（CLK_STATUS）

该寄存器中的位用于检测芯片时钟源是否稳定，时钟切换是否失败。

寄存器	偏移	R/W	描述	复位值
CLK_STATUS	CLK_BA+0x50	R	时钟状态监测寄存器	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLKSFAIL	**Reserved**		**HIRCSTB**	**LIRCSTB**	**PLLSTB**	**LXTSTB**	**HXTSTB**

位	描述	
[31:8]	**Reserved**	保留
[7]	**CLKSFAIL**	**时钟切换失败标志（只读）** 当软件切换系统时钟源时，该位将更新。如果切换的目标时钟稳定，该位将置0；如果切换的目标时钟不稳定，该位将置1。 0 = 时钟切换成功 1 = 时钟切换失败 **注：**该位写清零。
[6:5]	**Reserved**	保留
[4]	**HIRCSTB**	**HIRC时钟源稳定标志（只读）** 0 = 内部高速RC振荡器（HIRC） 时钟源不稳定或禁止 1 = 内部高速RC振荡器（HIRC）时钟源稳定且使能
[3]	**LIRCSTB**	**LIRC时钟源稳定标志（只读）** 0 = 内部低速RC振荡器（LIRC） 时钟源不稳定或禁止 1 = 内部低速RC振荡器（LIRC）时钟源稳定且使能
[2]	**PLLSTB**	**内部PLL时钟源稳定标志（只读）** 0 = 内部PLL 时钟源不稳定或禁止 1 = 内部PLL时钟源稳定且使能 **注：**如果不支持PLL，此位无效。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[1]	**LXTSTB**	**LXT时钟源稳定标志（只读）** 0 = 外部低速晶振（LXT） 时钟源不稳定或禁止 1 = 外部低速晶振（LXT） 时钟源稳定且使能 **注：**如果不支持LXT，此位无效。

		请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。
[0]	**HXTSTB**	**HXT时钟源稳定标志（只读）** 0 = 外部高速晶振（HXT） 时钟源不稳定或禁止 1 = 外部高速晶振（HXT）时钟源稳定且使能 **注：**如果不支持HXT，此位无效。 请参照3.2章节NuMicro® M031/M032 系列选型指南的详细信息。

时钟输出控制寄存器（CLK_CLKOCTL）

寄存器	偏移	R/W	描述	复位值
CLK_CLKOCTL	CLK_BA+0x60	R/W	时钟输出控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CLK1HZEN**	**DIV1EN**	**CLKOEN**	**FREQSEL**			

位	描述	
[31:6]	**Reserved**	保留
[6]	**CLK1HZEN**	**1Hz时钟输出使能位** 0 = 32.768 kHz频率补偿的1Hz时钟输出禁止 1 = 32.768 kHz频率补偿的1Hz时钟输出使能
[5]	**DIV1EN**	**时钟输出分频 1使能位** 0 = 时钟将输出由FREQSEL分频后的时钟源频率 1 = 时钟将输出时钟源频率
[4]	**CLKOEN**	**时钟输出使能位** 0 = 时钟输出功能禁止 1 = 时钟输出功能使能
[3:0]	**FREQSEL**	**时钟输出频率选择** 时钟频率公式为 $F_{out} = F_{in}/2^{(N+1)}$ F_{in} 是输入时钟频率 F_{out} 是分频输出时钟的频率 N 是FREQSEL[3:0]的4位值

时钟失败检测器控制寄存器（CLK_CLKDCTL）

寄存器	偏移	R/W	描述	复位值
CLK_CLKDCTL	CLK_BA+0x70	R/W	时钟失败检测器控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**HXTFQIEN**	**HXTFQDEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**LXTFIEN**	**LXTFDEN**	**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**HXTFIEN**	**HXTFDEN**	**Reserved**			

位	描述	
[31:18]	**Reserved**	保留
[17]	**HXTFQIEN**	**HXT时钟频率范围检测中断使能位** 0 = 外部高速晶振（HXT）时钟频率范围检测失败中断禁止 1 = 外部高速晶振（HXT）时钟频率范围检测失败中断使能
[16]	**HXTFQDEN**	**HXT时钟频率范围检测使能位** 0 = 外部高速晶振（HXT）时钟频率范围检测禁止 1 = 外部高速晶振（HXT）时钟频率范围检测使能
[15:14]	**Reserved**	保留
[13]	**LXTFIEN**	**LXT 时钟错误中断使能位** 0 = 外部低速晶振（LXT）时钟错误中断禁止 1 = 外部低速晶振（LXT）时钟错误中断使能
[12]	**LXTFDEN**	**LXT时钟错误检测使能位** 0 = 外部低速晶振（LXT） 时钟错误中断禁止 1 = 外部低速晶振（LXT）时钟错误中断使能
[11:6]	**Reserved**	保留
[5]	**HXTFIEN**	**HXT时钟错误中断使能位** 0 = 外部高速晶振（HXT） 时钟错误中断禁止 1 = 外部高速晶振（HXT）时钟错误中断使能
[4]	**HXTFDEN**	**HXT时钟错误检测使能位** 0 = 外部高速晶振（HXT） 时钟错误中断禁止 1 = 外部高速晶振（HXT）时钟错误中断使能
[3:0]	**Reserved**	保留

时钟失败检测器状态寄存器（CLK_CLKDSTS）

寄存器	偏移	R/W	描述	复位值
CLK_CLKDSTS	CLK_BA+0x74	R/W	时钟失败检测器状态寄存器	0x0000_0000

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

位	描述	
[31:9]	**Reserved**	保留
[8]	**HXTFQIF**	**HXT时钟频率范围检测中断位（写保护）** 0 = 外部高速晶振（HXT）时钟频率正常 1 = 外部高速晶振（HXT）时钟频率不正常 **注：**该位写清零。
[7:2]	**Reserved**	保留
[1]	**LXTFIF**	**LXT时钟失败中断位（写保护）** 0 = 外部低速晶振（LXT） 时钟正常 1 = 外部低速晶振（LXT） 时钟停止 **注：**该位写清零。
[0]	**HXTFIF**	**HXT时钟失败中断位（写保护）** 0 = 外部高速晶振（HXT） 时钟正常 1 = 外部高速晶振（HXT） 时钟停止 **注：**该位写清零。

时钟频率检测器上边界寄存器（CLK_CDUPB）

寄存器	偏移	R/W	描述	复位值
CLK_CDUPB	CLK_BA+0x78	R/W	时钟频率检测器上边界寄存器	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留
[9:0]	**UPERBD**	**HXT 时钟频率范围检测器上边界** 该位定义频率监测窗口的最大值。 当HXT频率监测值高于该寄存器的值时，HXT频率检测失败中断标志将设置为1。

时钟频率检测器下边界寄存器（CLK_CDLOWB）

寄存器	偏移	R/W	描述	复位值
CLK_CDLOWB	CLK_BA+0x7c	R/W	时钟频率检测器下边界寄存器	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留
[9:0]	**LOWERBD**	**HXT时钟频率范围检测器下边界** 该位定义频率监测窗口的最小值。 当HXT频率监测值低于该寄存器的值时，HXT频率检测失败中断标志将设置为 1。

当超出范围的频率HIRC_period*1024> HXT_period*CLK_DUPB 或 HIRC_period*1024 < HXT_period*CLK_CDLOWB时，将产生中断。

HXT滤波器选择控制寄存器（CLK_HXTFSEL）

寄存器	偏移	R/W	描述	复位值
CLK_HXTFSEL	CLK_BA+0xB4	R/W	HXT滤波器选择控制寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**HXTFSEL**	**HXT滤波器选择** 0 = HXT 频率大于12 MHz 1 = HXT 频率小于等于12 MHz **注:** 此位不能在HXT运行时改变。

6.3 系统管理

6.3.1 概述

系统管理包括以下部分:

- 系统复位
- 系统电源分布
- SRAM内存组织
- 系统定时器（SysTick）
- 嵌套向量中断控制器（NVIC）
- 系统控制寄存器

6.3.2 系统复位

系统复位可被以下列出的事件触发。通过读SYS_RSTSTS寄存器的复位事件标志可确定复位源。硬件复位源来自外设信号。软件复位通过设置控制寄存器触发。

- 硬件复位源
  - 上电复位（POR）
  - nRESET引脚低电平
  - 看门狗超时复位和窗口看门狗复位（WDT/WWDT 复位）
  - 低电复位（LVR）
  - 掉电检测复位（BOD 复位）
  - CPU锁死复位
- 软件复位源
  - CHIP复位通过往CHIPRST（SYS_IPRST0[0]）写1来复位整个芯片
  - MCU复位通过往SYSRESETREQ（AIRCR[2]）写1来复位芯片但保持芯片从APROM或LDROM启动的设置
  - CPU复位通过往CPURST（SYS_IPRST0[1]）写1来复位Cortex®-M0内核
  - nRESET去抖动时间32us

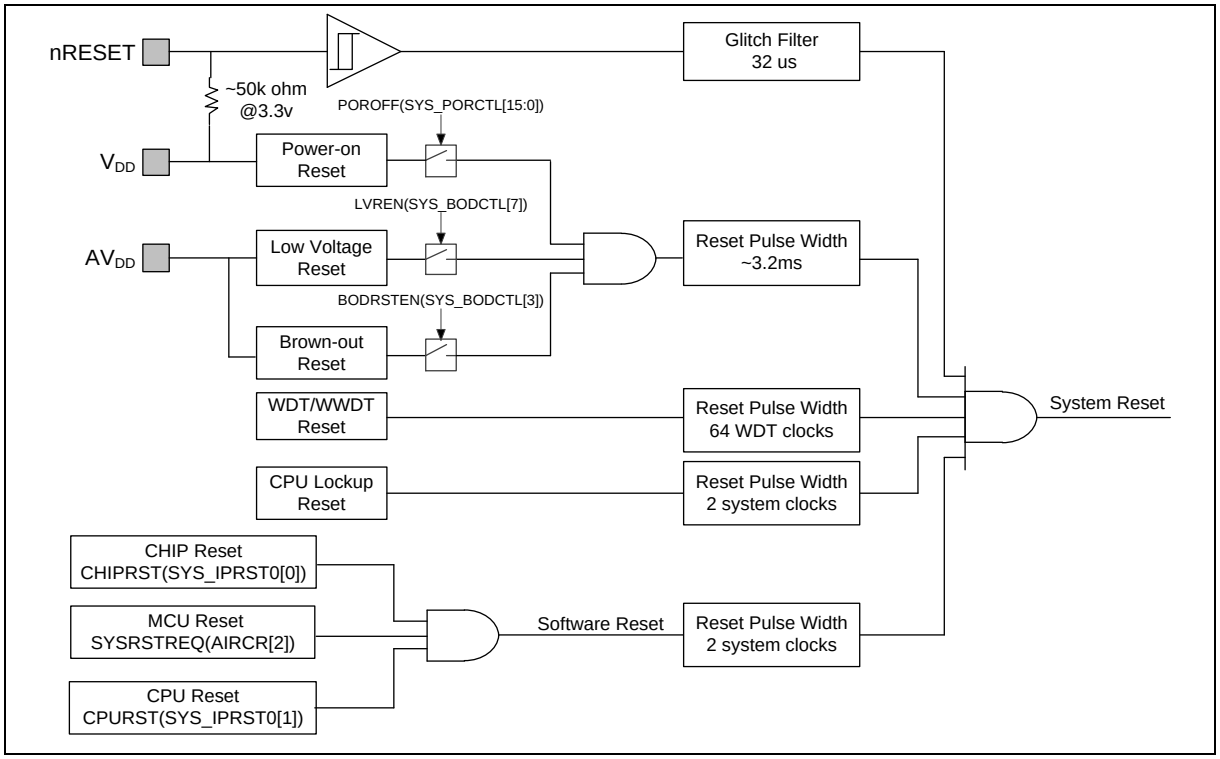


图 6.3-1 系统复位源

NuMicro®家族芯片有9个复位源。通常，CPU复位用于只复位Cortex®-M0；其他复位源将复位Cortex®-M0和所有外设。然而，每个复位源之间有微小的差别，如表 6.3-1

复位源寄存器	POR	nRESET	WDT	LVR	BOD	Lockup	CHIP	MCU	CPU
SYS_RSTSTS	0x001	Bit 1 = 1	Bit 2 = 1	Bit 3 = 1	Bit 4 = 1	Bit 8 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
CHIPRST（SYS_IPRST0[0]）	0x001	-	-	-	-	-	-	-	-
BODEN（SYS_BODCTL[0]） BODVL（SYS_BODCTL[16]） BODRSTEN（SYS_BODCTL[3]）	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	-	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	-
HXTEN（CLK_PWRCTL[0]）	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	从 CONFIG0 重载	
LXTEN（CLK_PWRCTL[1]）	0x0	-	-	-	-	-	-	-	-
LXTSELXT（CLK_PWRCTL[24]）	0x0	-	-	-	-	-	-	-	-
LXTGAIN	0x1	-	-	-	-	-	-	-	-

（CLK_PWRCTL[25:26]）									
WDTCKEN（CLK_APBCLK0[0]）	0x1	-	0x1	-	-	-	0x1	-	-
HCLKSEL（CLK_CLKSEL0[2:0]）	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	-
WDTSEL（CLK_CLKSEL1[1:0]）	0x3	0x3	-	-	-	-	-	-	-
HXTSTB（CLK_STATUS[0]）	0x0	-	-	-	-	-	-	-	-
LXTSTB（CLK_STATUS[1]）	0x0	-	-	-	-	-	-	-	-
PLLSTB（CLK_STATUS[2]）	0x0	-	-	-	-	-	-	-	-
HIRCSTB（CLK_STATUS[4]）	0x0	-	-	-	-	-	-	-	-
CLKSFAIL（CLK_STATUS[7]）	0x0	0x0	-	-	-	-	-	-	-
RSTEN（WDT_CTL[1]） WDTEN（WDT_CTL[7]）	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	-	从CONFIG0重载	-	-
WDT_CTL except bit 1 and bit 7.	0x0700	0x0700	0x0700	0x0700	0x0700	-	0x0700	-	-
WDT_ALTCTL	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_RLDCNT	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CTL	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	-	0x3F0800	-	-
WWDT_STATUS	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CNT	0x3F	0x3F	0x3F	0x3F	0x3F	-	0x3F	-	-
BS（FMC_ISPCTL[1]）	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	-	从CONFIG0重载	-	-
FMC_DFBA	从CONFIG1重载	从CONFIG1重载	从CONFIG1重载	从CONFIG1重载	从CONFIG1重载	-	从CONFIG1重载	-	-
CBS（FMC_ISPSTS[2:1]）	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	从CONFIG0重载	-	从CONFIG0重载	-	-
VECMAP（FMC_ISPSTS[23:9]）	基于CONFIG0	基于CONFIG0	基于CONFIG0	基于CONFIG0	基于CONFIG0	-	基于CONFIG0	-	-

	重载	重载	重载	重载	重载		重载		
其他外设寄存器	复位值								-
FMC 寄存器	复位值								
注意: '-' 表示寄存器的值保持原始设定									

表 6.3-1 寄存器复位值

6.3.2.1 nRESET复位

nRESET复位指的是通过拉低nRESET引脚产生一个复位信号，nRESET引脚是一个异步复位输入引脚，可以用来随时复位系统。当nRESET电压低于0.2VDD并且持续32 us（干扰脉冲滤波）芯片将会被复位。nRESET复位将控制芯片处于复位状态，直到nRESET电压上升到0.7VDD以上，并且持续32us（干扰脉冲滤波）。如果上次复位源是nRESET复位，PINRF（SYS_RSTSTS[1]）将会被置1。如下图 6.3-2 nRESET复位时序。

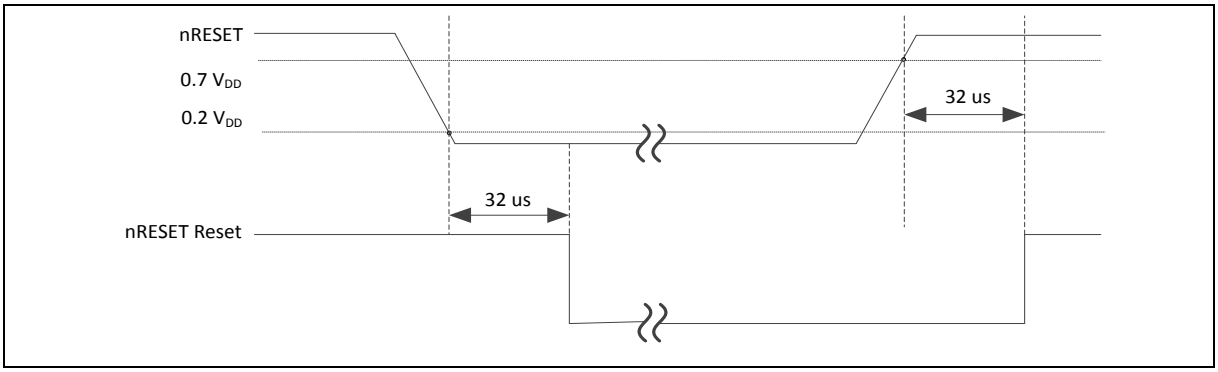


图 6.3-2 nRESET 复位时序

6.3.2.2 上电复位（POR）

上电复位（POR）用来产生稳定的系统复位信号，并在上电时强制系统复位，以避免MCU不可预期的行为。当给MCU供电时，POR模块将检测到电压上升，并向系统产生复位信号，直到电压适合MCU运行。在POR复位时，PORF（SYS_RSTSTS[0]）将设置为1，以指示存在POR复位事件。PORF（SYS_RSTSTS[0]）位可以通过向其写1来清除。图 6.3-3展示了上电复位波形。

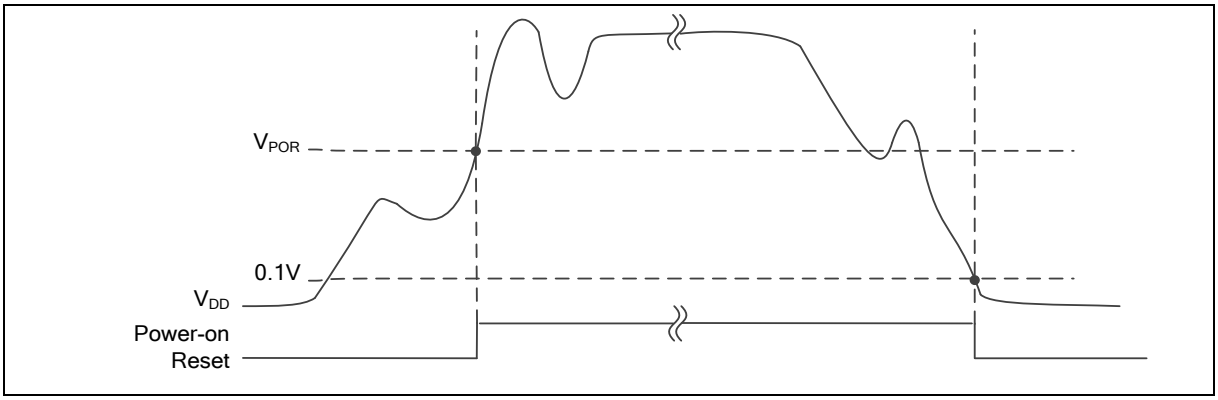


图 6.3-3 上电复位（POR）波形图

6.3.2.3 低电复位（LVR）

通过设置低电复位使能位LVREN （SYS_BODCTL[7]）为1，低电复位功能将被使能，延时200us后，LVR检测电路稳定并且LVR功能被激活。然后LVR功能将在系统运行期间检测AVDD。当AVDD 电压低于VLVR且保持这种状态的时间长于LVRDGSEL （SYS_BODCTL[14:12]）设置的干扰脉冲滤波时间，芯片将被复位。LVR复位将控制芯片处于复位状态，直到AVDD 电压上升到VLVR 以上，且保持这种状态的时间长于LVRDGSEL （SYS_BODCTL[14:12]）设置的干扰脉冲滤波时间。如果上一次复位源是LVR复位。默认设置是使能低电复位功能，但未开启干扰脉冲滤波功能。图 6.3-4展示了低电复位波形。

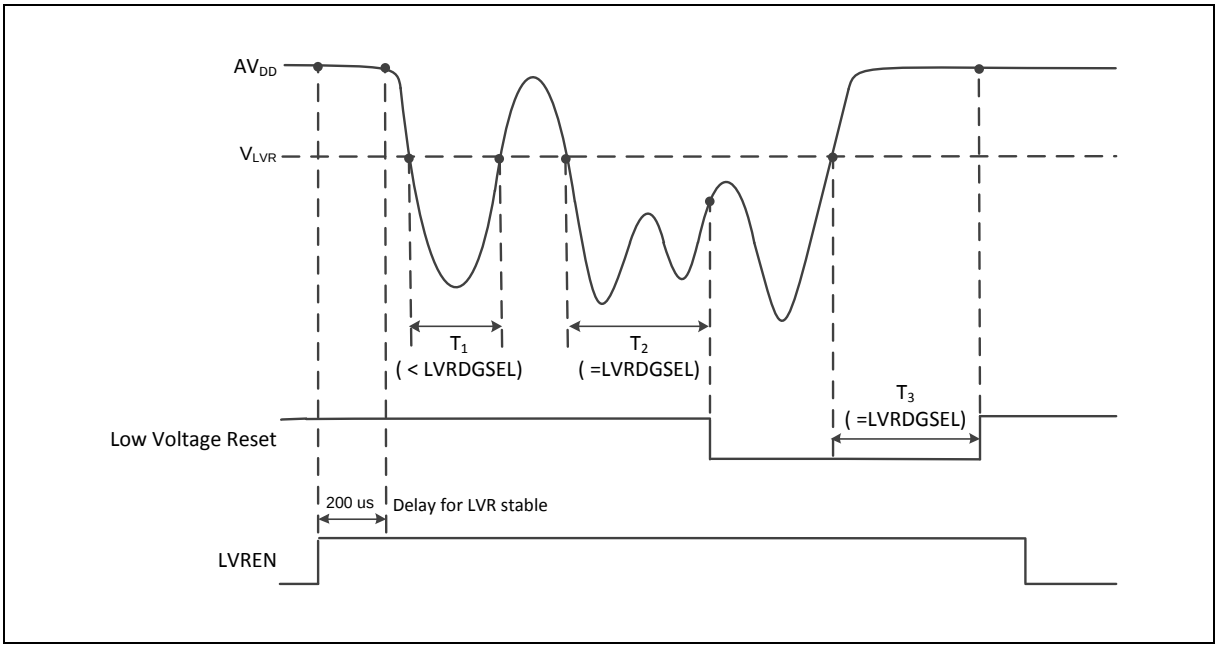


图 6.3-4 低压复位（LVR）波形图

6.3.2.4 掉电检测复位（BOD复位）

配置BODEN （SYS_BODCTL[0]）可使能掉电检测复位功能，掉电检测复位功能会在系统运行时检测AV_{DD}。当AV_{DD}电压低于已使能了BODEN和设置了掉电检测门限电压BODVL（SYS_BODCTL[16]）的V_{BOD}，且该状态保持时间长于设置的干扰脉冲滤波时间BODDGSEL（SYS_BODCTL[10:8]），芯片将会被复位。BOD复位将保持芯片处于复位状态，直到AV_{DD}电压上升到V_{BOD}以上，且保持该状态时间长于BODDGSEL所设置的干扰脉冲滤波时间。默认的BODEN、BODVL 和 BODRSTEN（SYS_BODCTL[3]） 分别通过内存控制器用户配置寄存器的CBODEN （CONFIG0 [19]）、CBOV （CONFIG0 [23:21]） 和 CBORST（CONFIG0[20]）设置。用户通过设置CONFIG0寄存器决定BOD的初始设置。图 6.3-5展示了掉电检测波形。

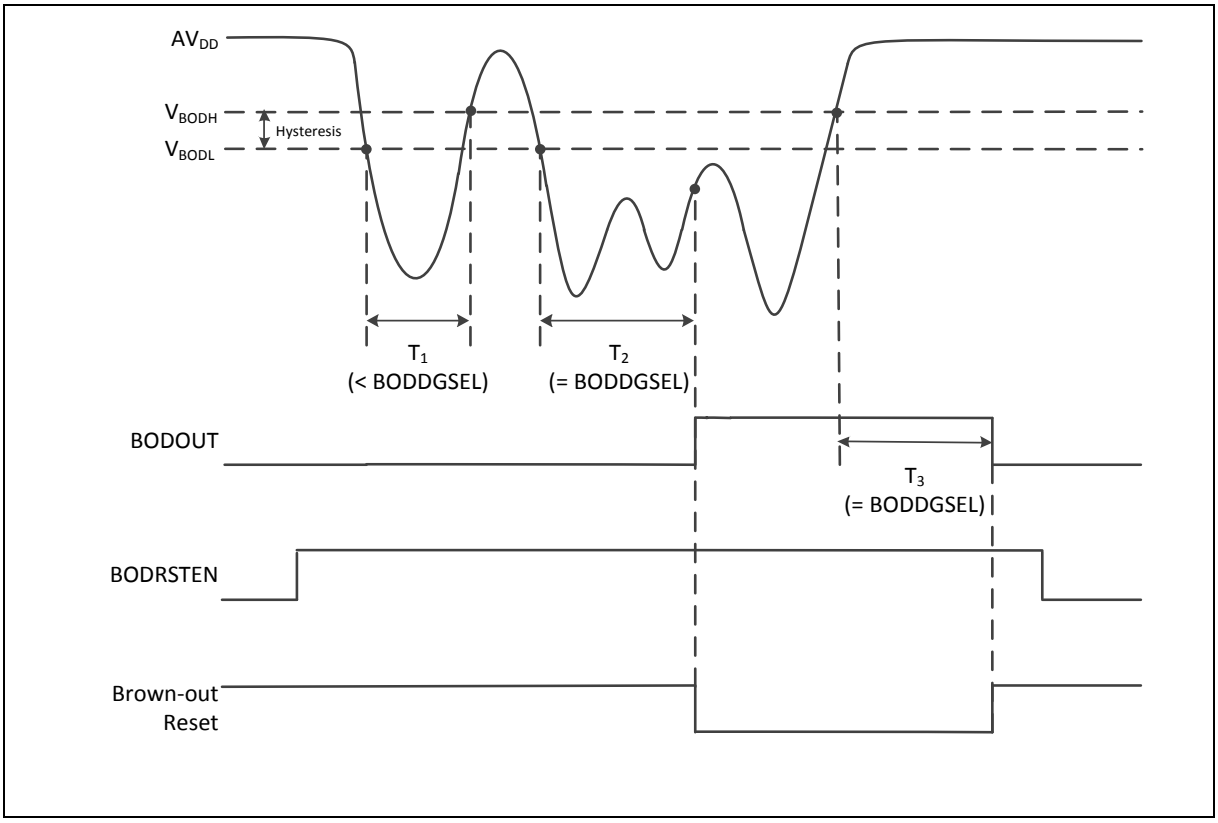


图 6.3-5 掉电检测复位（BOD）波形

6.3.2.5 看门狗定时器复位（WDT）

在大多数工业应用中，系统可靠性是非常重要的。从故障状态自动恢复MCU是提高系统可靠性的一种方法。看门狗定时器（WDT）被广泛用于检查系统是否工作正常。如果MCU崩溃或失控，可能会导致看门狗超时。用户可以决定在看门狗超时期间启用系统复位以恢复系统，并在复位后对系统崩溃/失控采取行动。

软件可以检查复位是否由看门狗超时引起，以指示前一复位是否是看门狗复位，通过检查WDTRF（SYS_RSTSTS[2]） 来处理看门狗超时复位后MCU的故障。

6.3.2.6 CPU锁死复位

当CPU产生硬件错误和芯片内核软件给出严重错误指示后，芯片进入锁定状态这是由于处理器内置的系统状态保护硬件激活后，由于不可恢复异常，CPU被锁定的结果。当芯片进入调试模式时, CPU 锁定复位将被忽略。

6.3.2.7 CPU复位，CHIP复位和MCU复位

CPU复位意味着只有Cortex®-M0核心被复位，并且所有其他外围设备在CPU复位之后保持相同的状态。用户可以将CPURST（SYS_IPRST0[1]）置为1以产生CPU复位信号。

CHIP复位与上电复位相同。CPU和所有外围设备被复位，BS（FMC_ISPCTL[1]）位从CONFIG0设置自动重载。用户可以将CHIPRST（SysIIPRST0[0]）置为1以产生CHIP复位信号。

MCU复位与CHIP复位类似。不同之处在于，BS（FMC_ISPCTL[1]）不会从CONFIG0设置重新加载，并且保持其原始的软件设置，以便从APROM或LDROM启动。用户可以将SYSRESETREQ（AIRCR[2]）置为1以产生MCU复位。

6.3.3 系统电源分配

芯片电源分配可分为三个部分：

- 模拟电源由AV_{DD}和AV_{SS}提供，为模拟组件提供工作电源。
- 数字电源由V_{DD}和V_{SS}用于向内部稳压器供电，内部稳压器为数字操作和I/O引脚提供固定的1.8V电源。
- USB收发器电源由VBUS提供，为USB收发器的运行提供工作电源。

内部电压调节器LDO和V_{DD}的输出需要一个外部电容，该电容应该靠近相应的引脚。图 6.3-6 展示了NuMicro® M031电源分布框图。

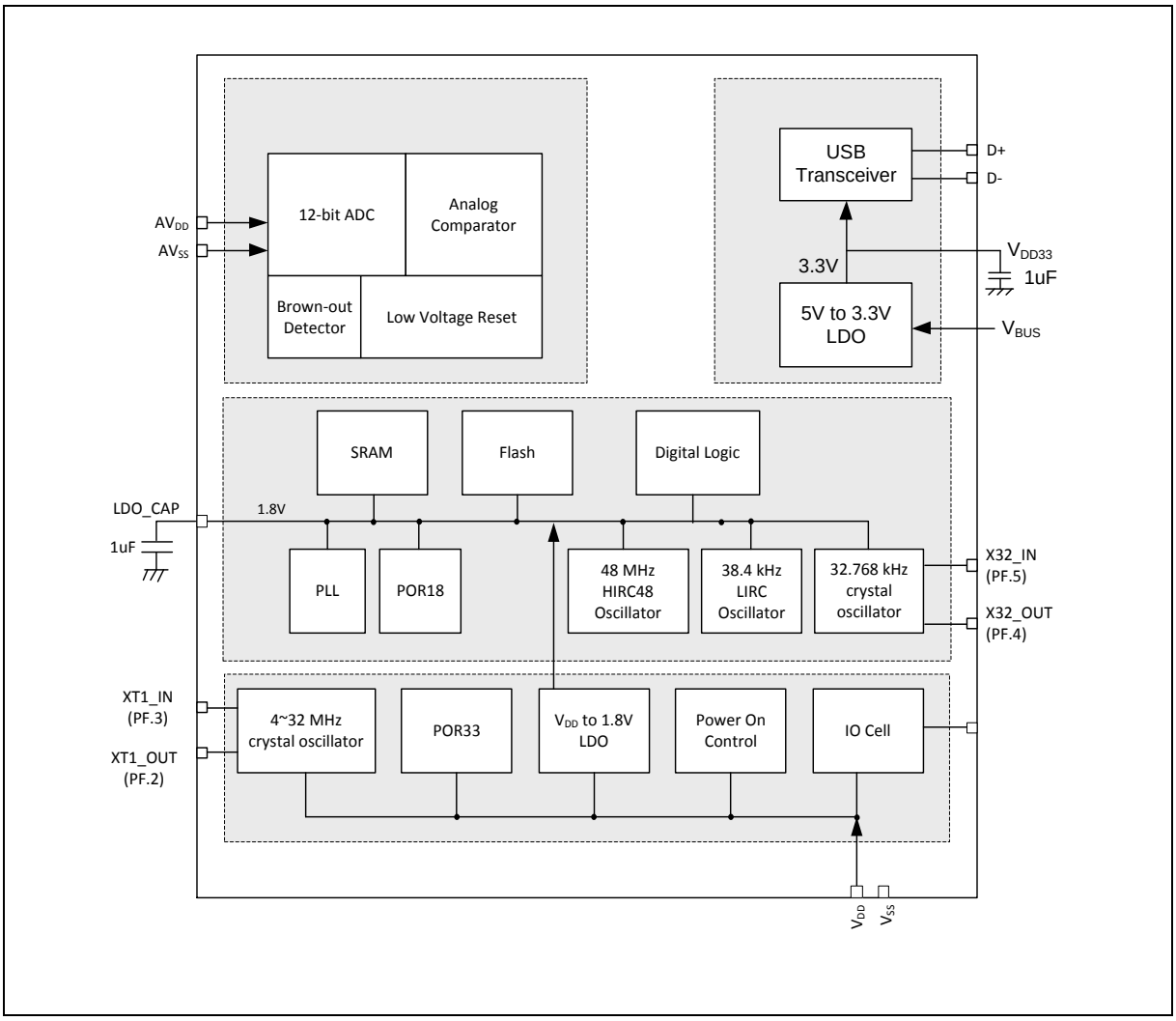


图 6.3-6 NuMicro® M031 电源分布框图

6.3.4 电源模式和唤醒源

M031/M032系列有电源管理单元，支持以下几种运行模式来降低功耗。 表 6.3-2列举了M031/M032系列所有的电源模式。

模式	CPU运行最大频率（MHz）	LDO_CAP（V）	时钟禁止
正常模式	72 MHz在2.0V-3.6V 48 MHz在1.8V-3.6V	1.8	所有时钟通过控制寄存器禁止
空闲模式	CPU进入睡眠模式	1.8	仅CPU时钟禁止
掉电模式	CPU进入掉电模式	1.8	大部分时钟被禁止，除了 LIRC/LXT。如果他们的时钟源选择为LIRC/LXT，仅有WDT/Timer/UART/ RTC外设时钟仍然使能

表 6.3-2 电源模式表

每个电源模式有不同的电源模式进入设置和离开条件。表 6.3-3显示了每个电源模式的进入设置。当芯片电源开启时，芯片以正常模式运行。用户可以通过设置SLEEPDEEP （SCR[2]），PDEN （CLK_PWRCT:[7]）和执行WFI指令来进入每个模式。

寄存器/指令模式	SLEEPDEEP（SCR[2]）	PDEN（CLK_PWRCTL[7]）	CPU运行WFI指令
正常模式	0	0	NO
空闲模式（CPU进入睡眠模式）	0	0	YES
掉电模式（CPU进入深度睡眠模式）	1	1	YES

表 6.3-3 电源模式差异表

在空闲模式和掉电模式中有几个唤醒源。表 6.3-4列出了每个电源模式的可用时钟。

电源模式	正常模式	空闲模式	掉电模式
定义	CPU 在工作状态	CPU 在睡眠状态	CPU在睡眠状态且所有时钟停止除了LXT和LIRC。SRAM 数据保持
进入条件	系统复位后芯片工作在正常模式	CPU 执行WFI指令	CPU使能睡眠模式和掉电模式后只执行WFI指令
唤醒源	N/A	所有中断	WDT, I²C, Timer, UART, BOD, GPIO, EINT, USCI, USBD, ACMP和RTC
可用时钟	All	除CPU时钟外所有时钟	LXT和LIRC
唤醒后	N/A	CPU 返回正常模式	CPU返回正常模式

表 6.3-4 电源模式差异表

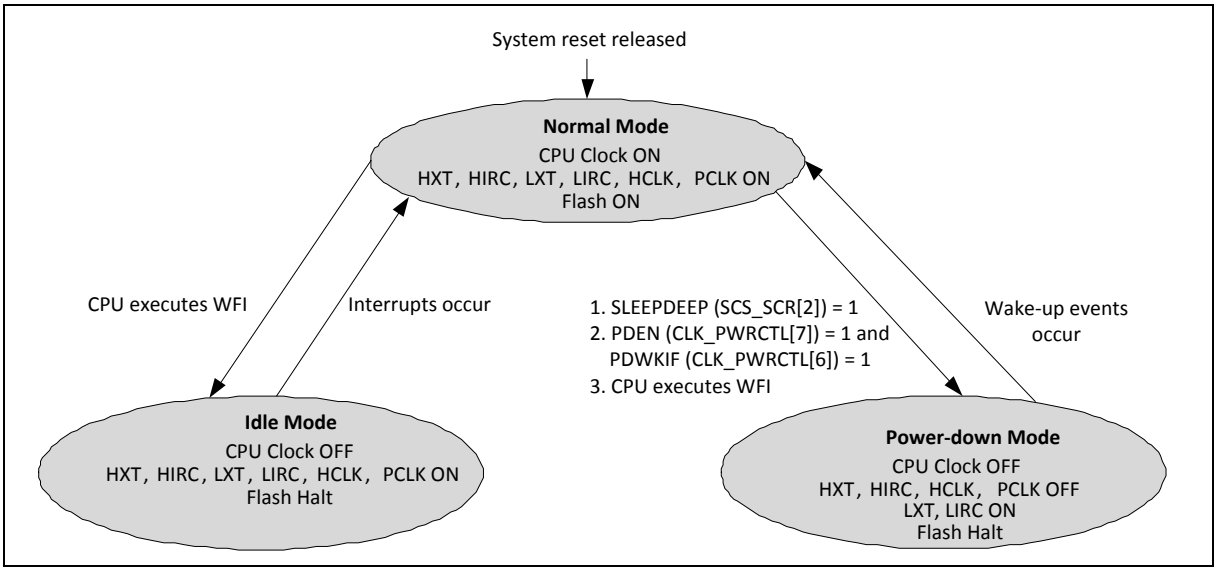


图 6.3-7 电源模式状态机

1. LXT（32768 Hz XTL）开或关取决于软件设定。
2. LIRC（38.4 kHz OSC）开或关取决于软件设定。
3. 如果TIMER时钟源选择LIRC/LXT且LIRC/LXT已打开。
4. 如果WDT时钟源选择LIRC且LIRC已打开。
5. 如果UART时钟源选择LXT且LXT已打开。
6. 如果RTC时钟源选择LIRC/LXT且LIRC/LXT已打开。

	正常模式	空闲模式	掉电模式
HXT（4~32 MHz XTL）	ON	ON	Halt
HIRC48（48 MHz OSC）	ON	ON	Halt
LXT（32768 Hz XTL）	ON	ON	ON/OFF[1]
LIRC（38.4 kHz OSC）	ON	ON	ON/OFF[2]
PLL	ON/OFF	ON/OFF	Halt
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/PCLK	ON	ON	Halt
SRAM保持	ON	ON	ON
FLASH	ON	ON	Halt
GPIO	ON	ON	Halt
PDMA	ON	ON	Halt
TIMER	ON	ON	ON/OFF[3]
PWM	ON	ON	Halt
BPWM	ON	ON	Halt

WDT	ON	ON	ON/OFF[4]
WWDT	ON	ON	Halt
UART	ON	ON	ON/OFF[5]
USCI	ON	ON	Halt
I^2C	ON	ON	Halt
SPI	ON	ON	Halt
QSPI	ON	ON	Halt
USBD	ON	ON	Halt
ADC	ON	ON	Halt
ACMP	ON	ON	Halt
RTC	ON	ON	ON/OFF[6]

表 6.3-5 不同电源模式下的时钟

掉电模式下的唤醒源：

WDT, I²C, Timer, UART, USCI, BOD, GPIO, USBD, ACMP和RTC。

当芯片进入掉电模式后,下述唤醒源可以唤醒芯片到正常模式。表 6.3-5列出了对于各种外设怎样才能使系统再次进入掉电模式的条件。

*用户必须在设置PDEN（CLK_PWRCTL[7]）和执行WFI指令进入掉电模式前等待这些条件完成。

唤醒源	唤醒条件	系统能再次进入掉电模式的条件*
BOD	欠压检测中断	在软件写1清除（SYS_BODCTL[4]）后
INT	外部中断	在软件写1清除Px_INTSRC[n] 位后
GPIO	GPIO 中断	在软件写1清除Px_INTSRC[n]位后
TIMER	Timer 中断	在软件写1清除 TWKF （TIMERx_INTSTS[1]） 和 TIF （TIMERx_INTSTS[0]）后
WDT	WDT 中断	在软件写1清除 WKF （WDT_CTL[5]）（写保护）后
RTC	闹铃中断	软件写1清除ALMIF（RTC_INTSTS[0]） 位后
	定时器嘀嗒中断	软件写1清除TICKIF（RTC_INTSTS[1]） 位后
UART0/1/4/5	nCTS 唤醒	软件写1清除CTSWKF（UARTx_WKSTS[0]） 位后
	传入数据唤醒	软件写1清除DATWKF（UARTx_WKSTS[1]） 位后
	接收FIFO阀值唤醒	软件写1清除RFRTWKF（UARTx_WKSTS[2]） 位后
	RS-485 AAD模式唤醒	软件写1清除RS485WKF（UARTx_WKSTS[3]） 位后
	接收FIFO阀值超时唤醒	软件写1清除TOUTWKF（UARTx_WKSTS[4]） 位后
UART2/3/6/7	nCTS 唤醒	软件写1清除CTSWKF（UARTx_WKSTS[0]） 位后
	传入数据唤醒	软件写1清除DATWKF（UARTx_WKSTS[1]） 位后

USCI UART	CTS触发	软件写1清除WKF（UUART_WKSTS[0]） 位后
	数据触发	软件写1清除WKF（UUART_WKSTS[0]） 位后
USCI I^{2}C	数据触发	软件写1清除WKF（UI2C_WKSTS[0]） 位后
	地址匹配	软件写1清除WKAKDONE（UI2C_PROTSTS[16]位， 然后写1清除WKF（UI2C_WKSTS[0]） 位后
USCI SPI	SS触发	软件写1清除WKF（USPI_WKSTS[0]） 位后
I^{2}C	地址匹配	软件写1清除WKIF（I2C_WKSTS[0]） 位后
USBD	远程唤醒	软件写1清除BUSIF（USBD_INTSTS[0]） 位后
ACMP	比较器掉电唤醒中断	软件写1清除WKIF0（ACMP_STATUS[8]）位和WKIF1（ACMP_STATUS[9]）位后

表 6.3-6 再次进入掉电状态的条件

6.3.5 系统内存映射

NuMicro® M031/M032系列提供4G字节地址空间。分配给控制器的地址空间如表 6.3-7。每个片上外设的详细的寄存器定义、地址空间和编程将在接下来的章节中描述。M031/M032系列只支持小端数据格式。

地址空间	标识	控制器
Flash 和 SRAM 存储空间		
0x0000_0000 – 0x0007_FFFF	FLASH_BA	FLASH 存储空间（512Kb）
0x2000_0000 – 0x2001_7FFF	SRAM0_BA	SRAM存储空间（96Kb）
0x6000_0000 – 0x6FFF_FFFF	EXTMEM_BA	外部存储空间（256MB）
外设控制器空间（0x4000_0000 – 0x400F_FFFF）		
0x4000_0000 – 0x4000_01FF	SYS_BA	系统控制寄存器
0x4000_0200 – 0x4000_02FF	CLK_BA	时钟控制寄存器
0x4000_0300 – 0x4000_03FF	NMI_BA	非屏蔽中断寄存器
0x4000_4000 – 0x4000_4FFF	GPIO_BA	GPIO控制寄存器
0x4000_8000 – 0x4000_8FFF	PDMA_BA	外设DMA控制寄存器
0x4000_C000 – 0x4000_CFFF	FMC_BA	内存控制寄存器
0x4001_0000 – 0x4001_0FFF	EBI_BA	外部总线接口控制寄存器
0x4001_4000 – 0x4001_7FFF	HDIV_BA	硬件除法器寄存器
0x4003_1000 – 0x4003_1FFF	CRC_BA	CRC生成器寄存器
APB控制器空间（0x4000_0000 ~ 0x400F_FFFF）		
0x4004_0000 – 0x4004_0FFF	WDT_BA	看门狗定时器寄存器
0x4004_1000 – 0x4004_1FFF	RTC_BA	RTC控制寄存器

0x4004_3000 – 0x4004_3FFF	ADC_BA	模拟数字转换（ADC）控制寄存器
0x4004_5000 – 0x4004_5FFF	ACMP01_BA	模拟比较器0/1控制寄存器
0x4005_0000 – 0x4005_0FFF	TMR01_BA	Timer0/Timer1控制寄存器
0x4005_1000 – 0x4005_1FFF	TMR23_BA	Timer2/Timer3控制寄存器控制寄存器
0x4005_8000 – 0x4005_8FFF	PWM0_BA	PWM0控制寄存器
0x4005_9000 – 0x4005_9FFF	PWM1_BA	PWM1控制寄存器
0x4005_A000 – 0x4005_AFFF	BPWM0_BA	BPWM0控制寄存器
0x4005_B000 – 0x4005_BFFF	BPWM1_BA	BPWM1控制寄存器
0x4006_0000 – 0x4006_0FFF	QSPI0_BA	QSPI0控制寄存器
0x4006_1000 – 0x4006_1FFF	SPI0_BA	SPI0控制寄存器
0x4007_0000 – 0x4007_0FFF	UART0_BA	UART0控制寄存器
0x4007_1000 – 0x4007_1FFF	UART1_BA	UART1控制寄存器
0x4007_2000 – 0x4007_2FFF	UART2_BA	UART2控制寄存器
0x4007_3000 – 0x4007_3FFF	UART3_BA	UART3控制寄存器
0x4007_4000 – 0x4007_4FFF	UART4_BA	UART4控制寄存器
0x4007_5000 – 0x4007_5FFF	UART5_BA	UART5控制寄存器
0x4007_6000 – 0x4007_6FFF	UART6_BA	UART6控制寄存器
0x4007_7000 – 0x4007_7FFF	UART7_BA	UART7控制寄存器
0x4008_0000 – 0x4008_0FFF	I2C0_BA	I2C0控制寄存器
0x4008_1000 – 0x4008_1FFF	I2C1_BA	I2C1控制寄存器
0x400C_0000 – 0x400C_0FFF	USBD_BA	USB设备控制寄存器
0x400D_0000 – 0x400D_0FFF	USCI0_BA	USCI0控制寄存器
0x400D_1000 – 0x400D_1FFF	USCI1_BA	USCI1控制寄存器
系统控制空间（0xE000_E000 ~ 0xE000_EFFF）		
0xE000_E010 – 0xE000_E0FF	SCS_BA	系统定时器控制寄存器
0xE000_E100 – 0xE000_ECFF	SCS_BA	外部中断控制寄存器
0xE000_ED00 – 0xE000_ED8F	SCS_BA	系统控制寄存器

表 6.3-7 片上控制器地址空间分配

6.3.6 SRAM内存结构

M031支持内嵌SRAM，总共16K字节

- 支持总共16K字节SRAM
- 支持字节/半字/字写
- 支持地址溢出报错

图 6.3-8 展示了SRAM 内存组织。地址空间从0x2000_4000到0x3FFF_FFFF是非法内存空间，如果CPU访问这些非法的内存空间,芯片将会进入硬件错误。

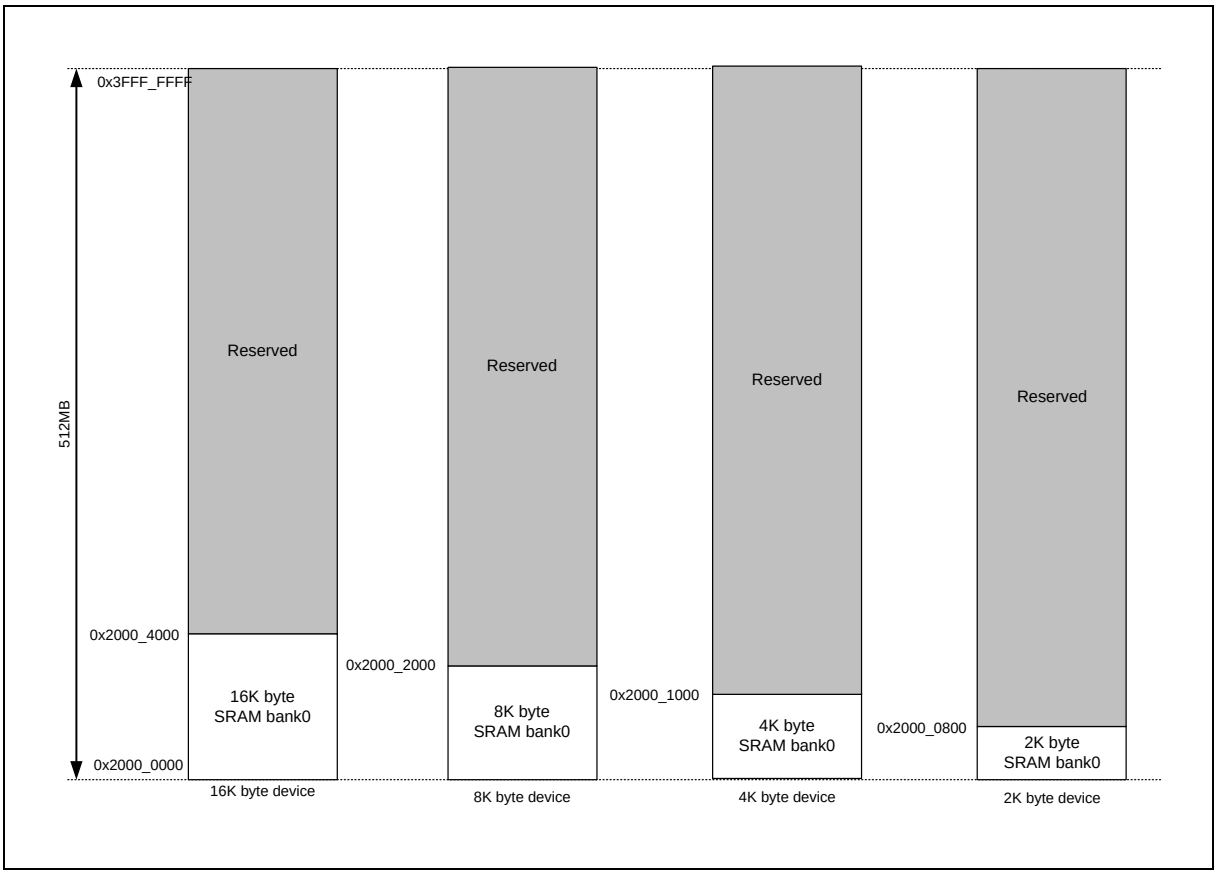


图 6.3-8 SRAM 内存组织

6.3.7 带奇偶校验功能的SRAM内存结构

M031支持内嵌SRAM，总共96K字节

- 支持总共96K字节SRAM
- 支持SRAM bank0 段0奇偶校验功能（32Kb）
- 支持字节/半字/字写
- 支持地址溢出报错

图 6.3-10 SRAM 内存组织。0x2001_8000到0x3FFF_FFFF之间的地址是非法内存空间，如果CPU访问这些非法内存地址，芯片将进入硬件错误。SRAM bank0有三个部分。第0段带奇偶校验功能寻址到32Kb，第1段寻址到32Kb，第2段寻址到32Kb。SRAM段0具有字节奇偶校验错误检查功能。当CPU访问SRAM第0段时，奇偶校验错误检查机制是动态运行的。发生奇偶校验错误时，PERRIF（

SYS_SRAM_STATUS[0]）将被置1，SYS_SRAM_ERRADDR寄存器将保留奇偶校验错误地址。如果PERRIEN（SYS_SRAM_INTCTL[0]）设置为1，当发生SRAM奇偶校验错误时，芯片将进入中断。当发生SRAM奇偶校验错误时，芯片将停止检测SRAM奇偶校验错误，直到用户写入1以清除PERRIF（SYS_SRAM_STATUS[0]）位。

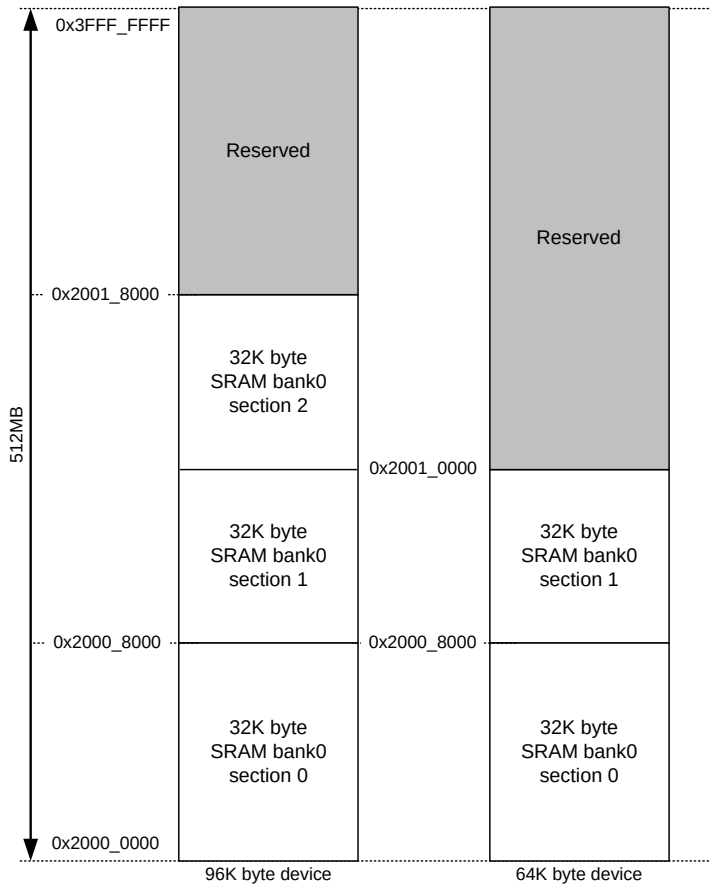


图 6.3-10 SRAM 内存组织

6.3.8 芯片总线矩阵

M031/M032系列支持总线矩阵来管理主机间的访问仲裁。访问仲裁采用轮询算法作为总线优先级。

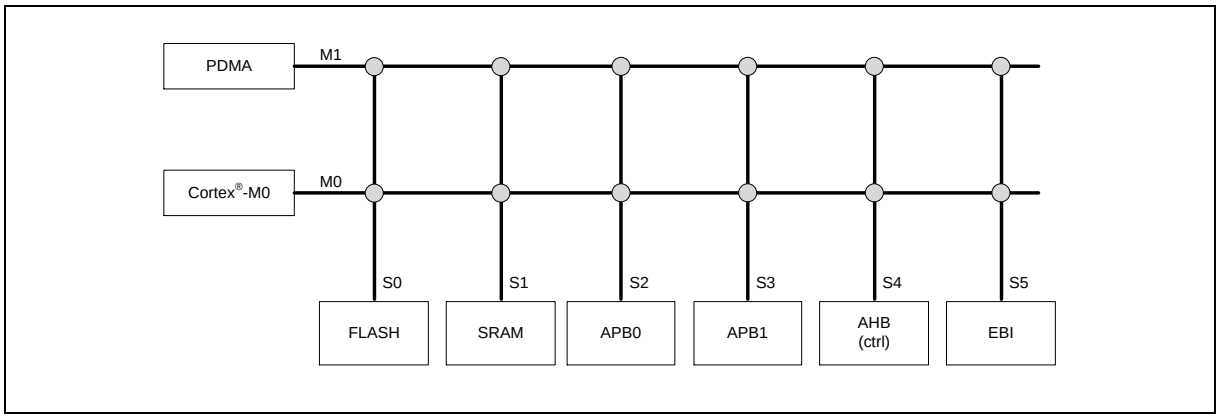


图 6.3-9 NuMicro® M031 总线矩阵图

6.3.9 IRC自动校准

该芯片支持自动校准功能: HIRC校准（48 MHz RC振荡器）,根据精确的外部32.768kHz晶体振荡器或内部USB同步模式，自动得到精确的输出频率，全温度范围内，精度达到0.25%。

在HIRC校准时，系统需要一个精确的48 MHz时钟。这种情况下, 如果既不使用PLL作为系统时钟源，也没有 32.768 kHz 晶振，用户必须设置 REFCKSEL （SYS_HIRCTRIMCTL [10] 参考时钟选择）为 “1”, 设置 FREQSEL （SYS_HIRCTRIMCTL [1:0]校准时钟选择） 为 “01”，自动校准功能将被使能。 当中断状态位 FREQLOCK （SYS_HIRCTRIMSTS[0] HIRC 频率锁定状态）为“1” 时，HIRC输出频率是准确的，偏差范围在0.25%以内。

6.3.10 寄存器锁控制器

部分系统控制寄存器需要保护，以避免误操作干扰芯片运行。这些系统控制寄存器在通电复位后被保护，直到用户禁用寄存器保护。用户如果想要编程这些保护寄存器，必须通过特定的编程来禁用寄存器保护。禁用寄存器保护的方法是依次往地址为0x4000_0100的寄存器SYS_REGLCTL写入“59h”, “16h” “88h”。 任何不同的数据、不同的写入顺序或者在这三个数据写入的过程中，写其它地址，将打断禁用寄存器保护方法流程。所有受保护的控制寄存器都注明“（写保护）”，并添加一个注释在寄存器描述中“**注意**：此位是写保护的。参考SYS_REGLCTL寄存器”。

寄存器	位	描述
CLK_PWRCTL	[26:25] LXTGAIN	LXT增益控制位（写保护）
CLK_PWRCTL	[22:20] HXTGAIN	HXT增益控制位（写保护）
CLK_PWRCTL	[7] PDEN	系统掉电使能（写保护）
CLK_PWRCTL	[5] PDWKIEN	掉电模式唤醒中断使能位（写保护）
CLK_PWRCTL	[4] PDWKDLY	唤醒延时计数器使能位（写保护）
CLK_PWRCTL	[3] LIRCEN	LIRC使能位（写保护）
CLK_PWRCTL	[2] HIRCEN	HIRC使能位（写保护）
CLK_PWRCTL	[1] LXTEN	LXT使能位（写保护）
CLK_PWRCTL	[0] HXTEN	HXT使能位（写保护）
CLK_APBCLK0	[0] WDTCKEN	看门狗定时器时钟使能位（写保护）
CLK_CLKSEL0	[8] USBDSEL	USB 设备时钟源选择（写保护）
CLK_CLKSEL0	[5:3] STCLKSEL	Cortex®-M0 SysTick时钟源选择（写保护）
CLK_CLKSEL0	[2:0] HCLKSEL	HCLK时钟源选择（写保护）
CLK_CLKSEL1	[3:2] WWDTSEL	窗口看门狗定时器时钟源选择（写保护）
CLK_CLKSEL1	[1:0] WDTSEL	看门狗定时器时钟源选择（写保护）
CLK_PLLCTL	[23] STBSEL	PLL稳定计数选择（写保护）
CLK_PLLCTL	[19] PLLSRC	PLL时钟源选择（写保护）
CLK_PLLCTL	[18] OE	PLL OE（FOUT 使能）引脚控制（写保护）
CLK_PLLCTL	[17] BP	PLL旁路控制（写保护）
CLK_PLLCTL	[16] PD	掉电模式（写保护）
CLK_PLLCTL	[15:14] OUTDIV	PLL输出除频控制器（写保护）
CLK_PLLCTL	[13:9] INDIV	PLL输入除频控制器（写保护）
CLK_PLLCTL	[8:0] FBDIV	PLL反馈分频器控制（写保护）
CLK_CLKDSTS	[8] HXTFQIF	HXT时钟频率范围检测器中断标志（写保护）
CLK_CLKDSTS	[1] LXTFIF	LXT时钟失败中断位（写保护）

CLK_CLKDSTS	[0] HXTFIF	HXT时钟失败中断位（写保护）
SYS_IPRST0	[7] CRCRST	CRC 计算控制器复位（写保护）
SYS_IPRST0	[4] HDIV_RST	HDIV控制器复位（写保护）
SYS_IPRST0	[3] EBIRST	EBI控制器复位（写保护）
SYS_IPRST0	[2] PDMARST	PDMA控制器复位（写保护）
SYS_IPRST0	[1] CPURST	处理器内核复位一次（写保护）
SYS_IPRST0	[0] CHIPRST	芯片复位一次（写保护）
SYS_BODCTL	[20] LVRVL	低压复位检测门限电压选择（写保护）
SYS_BODCTL	[16] BODVL	欠压复位检测门限电压选择（写保护）
SYS_BODCTL	[14:12] LVRDGSEL	低压复位输出干扰滤波时间选择（写保护）
SYS_BODCTL	[10:8] BODDGSEL	欠压检测器输出干扰滤波时间选择（写保护）
SYS_BODCTL	[7] LVREN	低电复位使能位（写保护）
SYS_BODCTL	[5] BODLPM	欠压复位检测低电模式（写保护）
SYS_BODCTL	[3] BODRSTEN	欠压复位使能位（写保护）
SYS_BODCTL	[0] BODEN	欠压复位检测使能位（写保护）
SYS_PORCTL	[15:0] POROFF	上电复位使能位（写保护）
SYS_SRAM_BISTCTL	[18] SRS2	SRAM Bank0 段2 BIST选择（写保护）
SYS_SRAM_BISTCTL	[17] SRS1	SRAM Bank0 段1 BIST 选择（写保护）
SYS_SRAM_BISTCTL	[16] SRS0	SRAM Bank0 段0 BIST 选择（写保护）
SYS_SRAM_BISTCTL	[7] PDMABIST	PDMA BIST使能位（写保护）
SYS_SRAM_BISTCTL	[4] USBBIST	USB BIST使能位（写保护）
SYS_SRAM_BISTCTL	[2] FMCBIST	FMC缓存BIST使能位（写保护）
SYS_SRAM_BISTCTL	[0] SRBIST	SRAM BIST使能位（写保护）
SYS_PORDISAN	[15:0] POROFFAN	上电复位使能位（写保护）
NMIEN	[15] UART1_INT	UART1非屏蔽源使能（写保护）
NMIEN	[14] UART0_INT	UART0非屏蔽源使能（写保护）
NMIEN	[13] EINT5	PB.7, PD.12 或PF.14 脚外部中断非屏蔽源使能（写保护）
NMIEN	[12] EINT4	PA.8, PB.6或PF.15 脚外部中断非屏蔽源使能（写保护）
NMIEN	[11] EINT3	PB.2或PC.7 脚外部中断非屏蔽源使能（写保护）
NMIEN	[10] EINT2	PB.3或PC.6 脚外部中断非屏蔽源使能（写保护）
NMIEN	[9] EINT1	PA.7, PB.4或PD.15引脚非屏蔽源使能（写保护）
NMIEN	[8] EINT0	PA.6 或PB.5 引脚外部中断非屏蔽源使能（写保护）
NMIEN	[6] RTC_INT	RTC非屏蔽源使能（写保护）

NMIEN	[4] CLKFAIL	时钟失败检测和IRC自动校准中断非屏蔽源使能（写保护）
NMIEN	[3] SRAM_PERR	SRAM奇偶校验错误非屏蔽源使能（写保护）
NMIEN	[2] PWRWU_INT	掉电模式唤醒非屏蔽源使能（写保护）
NMIEN	[1] IRC_INT	IRC TRIM非屏蔽源使能（写保护）
NMIEN	[0] BODOUT	BOD非屏蔽源使能（写保护）
FMC_ISPCTL	[24] INTEN	ISP中断使能位（写保护）
FMC_ISPCTL	[6] ISPFF	ISP失败标志（写保护）
FMC_ISPCTL	[5] LDUEN	LDROM更新使能位（写保护）
FMC_ISPCTL	[4] CFGUEN	CONFIG更新使能位（写保护）
FMC_ISPCTL	[3] APUEN	APROM更新使能位（写保护）
FMC_ISPCTL	[2] SPUEN	SPROM更新使能位（写保护）
FMC_ISPCTL	[1] BS	启动选择（写保护）
FMC_ISPCTL	[0] ISPEN	ISP使能位（写保护）
FMC_ISPTRG	[0] ISPGO	ISP开始触发（写保护）
FMC_FTCTL	[9] CACHEINV	闪存缓存失效（写保护）
FMC_FTCTL	[7] BBOFF	闪存分支缓冲区禁用控制（写保护）
FMC_FTCTL	[6:4] FOM	频率优化模式（写保护）
FMC_ISPSTS	[6] ISPFF	ISP失败标志（写保护）
TIMER0_CTL	[31] ICEDEBUG	ICE调试模式确认禁用位（写保护）
TIMER1_CTL	[31] ICEDEBUG	ICE调试模式确认禁用位（写保护）
TIMER2_CTL	[31] ICEDEBUG	ICE调试模式确认禁用位（写保护）
TIMER3_CTL	[31] ICEDEBUG	ICE调试模式确认禁用位（写保护）
WDT_CTL	[31] ICEDEBUG	ICE调试模式确认禁用位（写保护）
WDT_CTL	[11:8] TOUTSEL	WDT超时间隔选择（写保护）
WDT_CTL	[7] WDTEN	WDT使能位（写保护）
WDT_CTL	[6] INTEN	WDT超时中断使能位（写保护）
WDT_CTL	[5] WKF	WDT超时唤醒位（写保护）
WDT_CTL	[4] WKEN	WDT超时唤醒功能控制（写保护）
WDT_CTL	[1] RSTEN	WDT超时复位使能位（写保护）
WDT_ALTCTL	[1:0] RSTDSEL	WDT复位延时选择（写保护）
BPWM_CTL0	[31] DBGTRIOFF	ICE调试模式确认禁用（写保护）
BPWM_CTL0	[30] DBGHALT	ICE调试模式计数器停止（写保护）
PWM_CTL0	[31] DBGTRIOFF	ICE调试模式确认禁用位（写保护）

PWM_CTL0	[30] DBGHALT	ICE调试模式计数器停止（写保护）
PWM_DTCTL0_1	[24] DTCKSEL	死区时钟选择（写保护）
PWM_DTCTL0_1	[16] DTEN	使能PWM配对的死区插入（PWM_CH0, PWM_CH1）（PWM_CH2, PWM_CH3）（PWM_CH4, PWM_CH5）（写保护）
PWM_DTCTL0_1	[11:0] DTCNT	死区时间计数器（写保护）
PWM_DTCTL2_3	[24] DTCKSEL	死区时钟选择（写保护）
PWM_DTCTL2_3	[16] DTEN	使能PWM配对的死区插入（PWM_CH0, PWM_CH1）（PWM_CH2, PWM_CH3）（PWM_CH4, PWM_CH5）（写保护）
PWM_DTCTL2_3	[11:0] DTCNT	死区时间计数器（写保护）
PWM_DTCTL4_5	[24] DTCKSEL	死区时钟选择（写保护）
PWM_DTCTL4_5	[16] DTEN	使能PWM配对的死区插入（PWM_CH0, PWM_CH1）（PWM_CH2, PWM_CH3）（PWM_CH4, PWM_CH5）（写保护）
PWM_DTCTL4_5	[11:0] DTCNT	死区计数器（写保护）
PWM_BRKCTL0_1	[19:18] BRKAODD	PWM奇数通道刹车行为选择（写保护）
PWM_BRKCTL0_1	[17:16] BRKAEVEN	PWM偶数通道刹车行为选择（写保护）
PWM_BRKCTL0_1	[15] SYSLBEN	使能系统故障作为电平检测刹车源（写保护）
PWM_BRKCTL0_1	[13] BRKP1LEN	使能引脚BKP1作为电平检测刹车源（写保护）
PWM_BRKCTL0_1	[12] BRKP0LEN	使能引脚BKP0作为电平检测刹车源（写保护）
PWM_BRKCTL0_1	[9] CPO1LBEN	使能ACMP1_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL0_1	[8] CPO0LBEN	使能ACMP0_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL0_1	[7] SYSEBEN	使能系统故障作为边沿检测刹车源（写保护）
PWM_BRKCTL0_1	[5] BRKP1EEN	使能PWMx_BRAKE1引脚作为边沿检测刹车源（写保护）
PWM_BRKCTL0_1	[4] BRKP0EEN	使能PWMx_BRAKE0引脚作为边沿检测刹车源（写保护）
PWM_BRKCTL0_1	[1] CPO1EBEN	使能ACMP1_O数字输出作为边沿检测刹车源（写保护）
PWM_BRKCTL0_1	[0] CPO0EBEN	使能ACMP0_O数字输出作为边沿检测刹车源（写保护）
PWM_BRKCTL2_3	[19:18] BRKAODD	PWM奇数通道刹车行为选择（写保护）
PWM_BRKCTL2_3	[17:16] BRKAEVEN	PWM偶数通道刹车行为选择（写保护）
PWM_BRKCTL2_3	[15] SYSLBEN	使能系统故障作为电平检测刹车源（写保护）
PWM_BRKCTL2_3	[13] BRKP1LEN	使能引脚BKP1作为电平检测刹车源（写保护）
PWM_BRKCTL2_3	[12] BRKP0LEN	使能引脚BKP0作为电平检测刹车源（写保护）
PWM_BRKCTL2_3	[9] CPO1LBEN	使能ACMP1_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL2_3	[8] CPO0LBEN	使能ACMP0_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL2_3	[7] SYSEBEN	使能系统故障作为边沿检测刹车源（写保护）
PWM_BRKCTL2_3	[5] BRKP1EEN	使能PWMx_BRAKE1引脚作为边沿检测刹车源（写保护）

PWM_BRKCTL2_3	[4] BRKP0EEN	使能PWMx_BRAKE0引脚作为边沿检测刹车源（写保护）
PWM_BRKCTL2_3	[1] CPO1EBEN	使能ACMP1_O 数字输出作为边沿检测刹车源（写保护）
PWM_BRKCTL2_3	[0] CPO0EBEN	使能ACMP0_O数字输出作为边沿检测刹车源（写保护）
PWM_BRKCTL4_5	[19:18] BRKAODD	PWM奇数通道刹车行为选择（写保护）
PWM_BRKCTL4_5	[17:16] BRKAEVEN	PWM偶数通道刹车行为选择（写保护）
PWM_BRKCTL4_5	[15] SYSLBEN	使能系统故障作为电平检测刹车源（写保护）
PWM_BRKCTL4_5	[13] BRKP1LEN	使能引脚BKP1作为电平检测刹车源（写保护）
PWM_BRKCTL4_5	[12] BRKP0LEN	使能引脚BKP0作为电平检测刹车源（写保护）
PWM_BRKCTL4_5	[9] CPO1LBEN	使能ACMP1_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL4_5	[8] CPO0LBEN	使能ACMP0_O数字输出作为电平检测刹车源（写保护）
PWM_BRKCTL4_5	[7] SYSEBEN	使能系统故障作为边沿检测刹车源（写保护）
PWM_BRKCTL4_5	[5] BRKP1EEN	使能PWMx_BRAKE1引脚作为边沿检测刹车源（写保护）
PWM_BRKCTL4_5	[4] BRKP0EEN	使能PWMx_BRAKE0引脚作为边沿检测刹车源（写保护）
PWM_BRKCTL4_5	[1] CPO1EBEN	使能ACMP1_O 数字输出作为边沿检测刹车源（写保护）
PWM_BRKCTL4_5	[0] CPO0EBEN	使能ACMP0_O数字输出作为边沿检测刹车源（写保护）
PWM_SWBRK	[8+n/2] n=0,2,4 BRKLTRGn	PWM电平刹车软件触发（写保护）
PWM_SWBRK	[n/2] n=0,2,4 BRKETRGn	PWM边沿刹车软件触发（写保护）
PWM_INTEN1	[10] BRKLIEN4_5	通道4/5的PWM电平检测刹车中断启用（写保护）
PWM_INTEN1	[9] BRKLIEN2_3	通道2/3的PWM电平检测刹车中断启用（写保护）
PWM_INTEN1	[8] BRKLIEN0_1	通道0/1的PWM电平检测刹车中断启用（写保护）
PWM_INTEN1	[2] BRKEIEN4_5	通道4/5的PWM边沿检测刹车中断启用（写保护）
PWM_INTEN1	[1] BRKEIEN2_3	通道2/3的PWM边沿检测刹车中断启用（写保护）
PWM_INTEN1	[0] BRKEIEN0_1	通道0/1的PWM边沿检测刹车中断启用（写保护）
PWM_INTSTS1	[8+n] n=0,1..5 BRKLIFn	PWM通道n电平检测刹车中断标志（写保护）
PWM_INTSTS1	[n] n=0,1..5 BRKEIFn	PWM通道n边沿检测刹车中断标志（写保护）
ADC_ADCR	[12] RESET	ADC复位（写保护）

6.3.11 UART0_TXD/USCI0_DAT1调制PWM

此芯片支持用UART0_TXD/USCI_DAT0调制PWM通道。用户可设置 MODPWMSEL（SYS_MODCTL[7:4]）来选择要调制的PWM0通道，通过设置MODEN（SYS_MODCTL[0]）来使能调制功能。用户可以在调制PWM之前设置 TXDINV（UART_LINE[8]）来反向UART0_TXD或者设置

DATOINV（UUART_LINECTL[5]）来反向USCI0_DAT1。

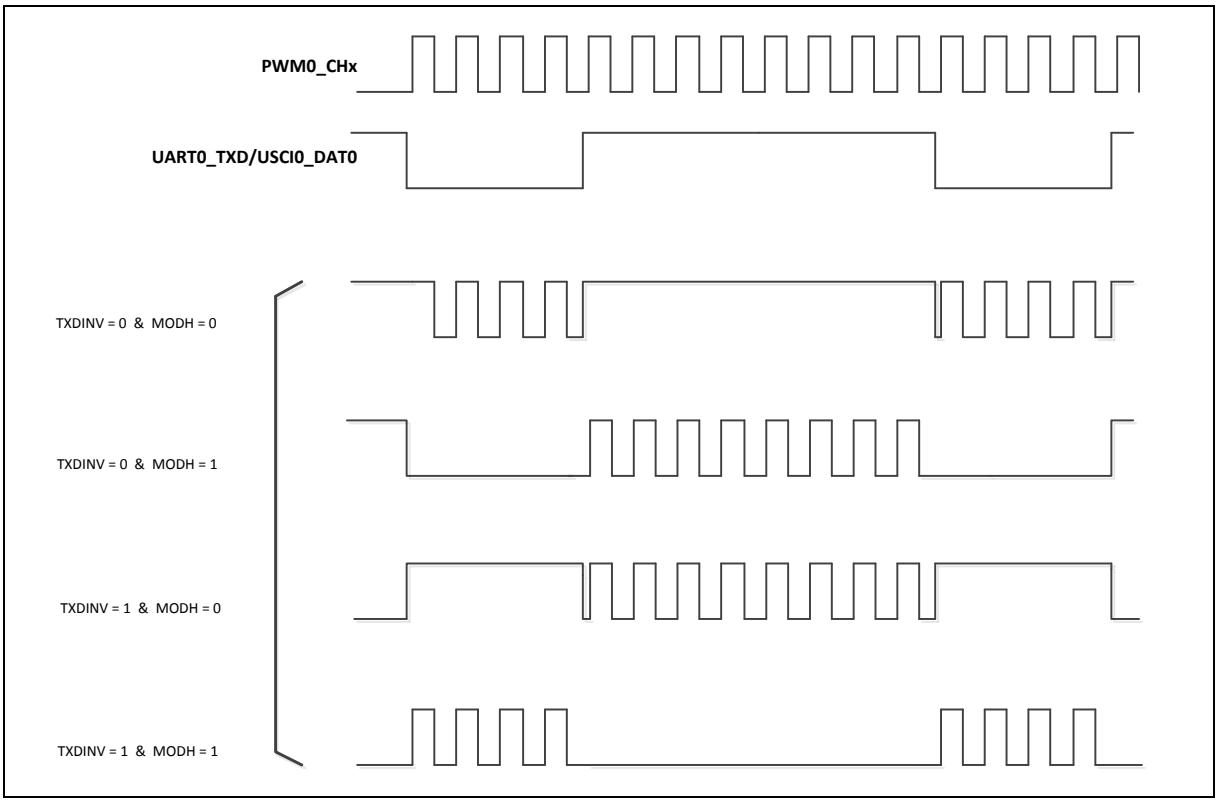


图 6.3-11 UART0_TXD/USCI0_DAT1 通过 PWM 通道调制

6.3.12 寄存器映射

R: 只读，**W**: 只写，**R/W**：可读写

寄存器	偏移	R/W	描述	复位值
SYS基地址： **SYS_BA = 0x4000_0000**				
SYS_PDID	SYS_BA+0x00	R	器件ID寄存器	0xXXXX_XXXX
SYS_RSTSTS	SYS_BA+0x04	R/W	系统复位状态寄存器	0x0000_0043
SYS_IPRST0	SYS_BA+0x08	R/W	外设复位控制寄存器0	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	外设复位控制寄存器1	0x0000_0000
SYS_IPRST2	SYS_BA+0x10	R/W	外设复位控制寄存器2	0x0000_0000
SYS_BODCTL	SYS_BA+0x18	R/W	欠压检测控制寄存器	0x00XX_038X
SYS_PORCTL	SYS_BA+0x24	R/W	上电复位控制寄存器	0x0000_0000
SYS_GPA_MFPL	SYS_BA+0x30	R/W	GPIOA多功能控制寄存器低字节	0x0000_0000
SYS_GPA_MFPH	SYS_BA+0x34	R/W	GPIOA多功能控制寄存器高字节	0x0000_0000
SYS_GPB_MFPL	SYS_BA+0x38	R/W	GPIOB多功能控制寄存器低字节	0x0000_0000
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	GPIOB多功能控制寄存器高字节	0x0000_0000
SYS_GPC_MFPL	SYS_BA+0x40	R/W	GPIOC多功能控制寄存器低字节	0x0000_0000
SYS_GPC_MFPH	SYS_BA+0x44	R/W	GPIOC多功能控制寄存器高字节	0x0000_0000
SYS_GPD_MFPL	SYS_BA+0x48	R/W	GPIOD多功能控制寄存器低字节	0x0000_0000
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	GPIOD多功能控制寄存器高字节	0x0000_0000
SYS_GPE_MFPL	SYS_BA+0x50	R/W	GPIOE多功能控制寄存器低字节	0x0000_0000
SYS_GPE_MFPH	SYS_BA+0x54	R/W	GPIOE多功能控制寄存器高字节	0x0000_0000
SYS_GPF_MFPL	SYS_BA+0x58	R/W	GPIOF多功能控制寄存器低字节	0x0000_00ee
SYS_GPF_MFPH	SYS_BA+0x5C	R/W	GPIOF多功能控制寄存器高字节	0x0000_0000
SYS_GPG_MFPL	SYS_BA+0x60	R/W	GPIOG多功能控制寄存器低字节	0x0000_0000
SYS_GPG_MFPH	SYS_BA+0x64	R/W	GPIOG多功能控制寄存器高字节	0x0000_0000
SYS_GPH_MFPL	SYS_BA+0x68	R/W	GPIOH多功能控制寄存器低字节	0x0000_0000
SYS_GPH_MFPH	SYS_BA+0x6C	R/W	GPIOH多功能控制寄存器高字节	0x0000_0000
SYS_MODCTL	SYS_BA+0xC0	R/W	调制控制寄存器	0x0000_0000
SYS_SRAM_BISTCTL	SYS_BA+0xD0	R/W	系统SRAM BIST测试控制寄存器	0x000X_0000
SYS_SRAM_BISTSTS	SYS_BA+0xD4	R	系统SRAM BIST测试状态寄存器	0x00xx_00xx

SYS_SRAM_INTCTL	SYS_BA+0xDC	R/W	系统SRAM中断使能控制寄存器	0x0000_0000
SYS_SRAM_STATUS	SYS_BA+0xE0	R/W	系统SRAM奇偶校验错误状态寄存器	0x0000_0000
SYS_SRAM_ERRADDR	SYS_BA+0xE4	R	系统SRAM奇偶校验错误地址寄存器	0x0000_0000
SYS_HIRCTRIMCTL	SYS_BA+0xF0	R/W	HIRC校准控制寄存器	0x0008_0000
SYS_HIRCTRIMIEN	SYS_BA+0xF4	R/W	HIRC校准中断使能寄存器	0x0000_0000
SYS_HIRCTRIMSTS	SYS_BA+0xF8	R/W	HIRC校准中断状态寄存器	0x0000_0000
SYS_REGLCTL	SYS_BA+0x100	R/W	寄存器锁控制寄存器	0x0000_0000
SYS_PORDISAN	SYS_BA+0x1EC	R/W	模拟POR禁用控制寄存器	0x0000_0000

6.3.13 寄存器描述

器件ID寄存器（SYS_PDID）

寄存器	偏移	R/W	描述	复位值
SYS_PDID	SYS_BA+0x00	R	器件ID寄存器	0xXXXX_XXXX

[1] 每个物料编号有唯一的默认复位值。

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

位	描述	
[31:0]	**PDID**	**器件ID（只读）** 此寄存器反映设备型号。软件可以读取这个寄存器来识别使用的设备型号。

系统复位状态寄存器（SYS_RSTSTS）

该寄存器为软件提供特定的信息，以此来指示芯片上次复位的复位源。

寄存器	偏移	R/W	描述	复位值
SYS_RSTSTS	SYS_BA+0x04	R/W	系统复位状态寄存器	0x0000_0043

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**CPULKRF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CPURF	**Reserved**	**SYSRF**	**BODRF**	**LVRF**	**WDTRF**	**PINRF**	**PORF**

位	描述	
[31:9]	**Reserved**	保留
[8]	**CPULKRF**	**CPU 锁定复位标志** 0 = 没有CPU锁定复位发生 1 = Cortex®-M0 锁定发生及芯片复位 **注1：**此位写1清0。 **注2：**当CPU锁定发生在ICE连接上后，这个标志竟会被置1但是芯片不会复位。
[7]	**CPURF**	**CPU复位标志** 如果软件写1到CPURST （SYS_IPRST0[1]） 来复位Cortex®- M0内核和Flash内存控制器（FMC），硬件将会自动置位标志位。 0 = 无复位来自CPU 1 = Cortex®-M0 内核和FMC通过设置CPURST为1被复位 **注：**写操作清除该位为0。
[6]	**Reserved**	保留
[5]	**SYSRF**	**系统复位标志** 系统复位标志来自Cortex®- M0内核的"复位信号"置位，以此来表明之前的复位源。 0 = 无复位来自 Cortex®-M0 1 = Cortex®-M0已经因为往Cortex®- M0内核的系统控制寄存器的SYSRESETREQ（AIRCR[2], 应用中断和复位控制寄存器，地址= 0xE000ED0C）位写1而产生复位信号复位系统 **注：**此位写1清0。

位	描述	
[4]	**BODRF**	**BOD复位标志** BOD复位标志由来自欠压检测器的“复位信号”置位，以此来表明之前的复位源。 0 = 无复位来自 BOD 1 = BOD已经产生复位信号复位系统 **注：**此位写1清0。
[3]	**LVRF**	**LVR复位标志** LVR复位标志由来自低电复位控制器的“复位信号”置位，以此来表明之前的复位源。 0 = 无复位来自 LVR 1 = LVR控制器已经产生复位信号复位系统 **注：**此位写1清0。
[2]	**WDTRF**	**WDT复位标志** WDT复位标志由来自看门狗定时器或窗口看门狗定时器的“复位信号”置位，以此来表明之前的复位源 0 = 无复位来自看门狗定时器或窗口看门狗定时器 1 = 看门狗定时器或窗口看门狗定时器已经产生复位信号复位系统 **注1：**此位写1清0。 **注2：**如果系统被WDT超时复位复位，看门狗定时器寄存器RSTF（WDT_CTL[2]）位将被置1；如果系统被WWDT超时复位复位，窗口看门狗定时器寄存器WWDTRF（WWDT_STATUS[1]）位将被置1。
[1]	**PINRF**	**nRESET 引脚复位标志** nRESET引脚复位由来自nRESET引脚的“复位信号”置位，以此来表明之前的复位源。 0 = 无复位来自nRESET引脚 1 = nRESET引脚已经产生复位信号复位系统 **注1：**此位写1清0。
[0]	**PORF**	**POR复位标志** POR复位标志由来自上电复位（POR）控制器或位CHIPRST （SYS_IPRST0[0]）的“复位信号”置位，以此来表明之前的复位源。 0 = 无复位来自POR 或 CHIPRST 1 = 上电复位（POR）或 CHIPRST已经产生复位信号复位系统 **注：**此位写1清0。

外设复位控制寄存器0（SYS_IPRST0）

寄存器	偏移	R/W	描述	复位值
SYS_IPRST0	SYS_BA+0x08	R/W	外设复位控制寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRCRST	**Reserved**		**HDIV_RST**	**EBIRST**	**PDMARST**	**CPURST**	**CHIPRST**

位	描述	
[31:8]	**Reserved**	保留
[7]	**CRCRST**	**CRC计算控制器复位（写保护）** 置该位为1将产生一个复位信号给CRC计算控制器，用户需要写0到该位才能将复位状态释放。 0 = CRC 计算控制器正常运行 1 = CRC 计算控制器复位 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[6:5]	**Reserved**	保留
[4]	**HDIV_RST**	**HDIV控制器复位（写保护）** 置该位为1将产生一个复位信号给硬件除法器，用户需要置该位为0才能将复位状态释放。 0 = 硬件除法器控制器正常运行 1 = 硬件除法器控制器复位 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[3]	**EBIRST**	**EBI控制器复位（写保护）** 置该位为1将产生一个复位信号给EBI，用户需要置该位为0才能将复位状态释放。 0 = EBI控制器正常运行 1 = EBI控制器复位 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[2]	**PDMARST**	**PDMA控制器复位（写保护）** 置该位为1将产生一个复位信号给PDMA，用户需要置该位为0才能将复位状态释放。 0 = PDMA控制器正常运行 1 = PDMA控制器复位 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[1]	**CPURST**	**处理器内核复位一次（写保护）**

		设置该位为1仅复位处理器内核和Flash存储控制器（FMC），该位两个时钟周期后自动清0。 0 = 处理器内核正常运行 1 = 处理器内核一次复位 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[0]	**CHIPRST**	**芯片复位（写保护）** 设置该位为1将复位整个芯片，包括处理器内核和所有外设，该位两个时钟周期后自动清0。 CHIPRST与POR复位一样，整个芯片控制器复位，来自Flash的芯片设定也将重载。 CHIPRST与SYSRESETREQ（AIRCR[2]）的区别，请参考6.3.2章节。 0 = 芯片正常运行 1 = 芯片复位 **注：** 该位写保护，参考SYS_REGLCTL 寄存器。 **注：** 由上电复位复位。

外设复位控制寄存器1（SYS_IPRST1）

设置这些位为1将产生异步复位信号到相应的模块控制器。用户需要将这些位设置为0，以从复位状态释放相应的模块控制器。

寄存器	偏移	R/W	描述	复位值
SYS_IPRST1	SYS_BA+0x0C	R/W	外设复位控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			**ADCRST**	**USBDRST**	**Reserved**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
UART7RST	**UART6RST**	**UART5RST**	**UART4RST**	**UART3RST**	**UART2RST**	**UART1RST**	**UART0RST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SPI0RST**	**QSPI0RST**	**Reserved**		**I2C1RST**	**I2C0RST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACMP01RST	**Reserved**	**TMR3RST**	**TMR2RST**	**TMR1RST**	**TMR0RST**	**GPIORST**	**Reserved**

位	描述	
[31:29]	**Reserved**	保留
[28]	**ADCRST**	**ADC控制器复位** 0 = ADC控制器正常运行 1 = ADC控制器复位
[27]	**USBDRST**	**USBD控制器复位** 0 = USBD控制器正常运行 1 = USBD控制器复位
[26:24]	**Reserved**	保留
[23]	**UART7RST**	**UART7控制器复位** 0 = UART7控制器正常运行 1 = UART7控制器复位
[22]	**UART6RST**	**UART6控制器复位** 0 = UART6控制器正常运行 1 = UART6控制器复位
[21]	**UART5RST**	**UART5控制器复位** 0 = UART5控制器正常运行 1 = UART5控制器复位
[20]	**UART4RST**	**UART4控制器复位** 0 = UART4控制器正常运行 1 = UART4控制器复位
[19]	**UART3RST**	**UART3控制器复位**

		0 = UART3控制器正常运行 1 = UART3控制器复位
[18]	**UART2RST**	**UART2控制器复位** 0 = UART2控制器正常运行 1 = UART2控制器复位
[17]	**UART1RST**	**UART1控制器复位** 0 = UART1控制器正常运行 1 = UART1控制器复位
[16]	**UART0RST**	**UART0控制器复位** 0 = UART0控制器正常运行 1 = UART0控制器复位
[15:14]	**Reserved**	保留
[13]	**SPI0RST**	**SPI0控制器复位** 0 = SPI0控制器正常运行 1 = SPI0控制器复位
[12]	**QSPI0RST**	**QSPI0控制器复位** 0 = QSPI0控制器正常运行 1 = QSPI0控制器复位
[11:10]	**Reserved**	保留
[9]	**I2C1RST**	**I2C1控制器复位** 0 = I2C1控制器正常运行 1 = I2C1控制器复位
[8]	**I2C0RST**	**I2C0控制器复位** 0 = I2C0控制器正常运行 1 = I2C0控制器复位
[7]	**ACMP01RST**	**模拟比较器0/1控制器复位** 0 = 模拟比较器0/1控制器正常运行 1 = 模拟比较器0/1控制器复位
[6]	**Reserved**	保留
[5]	**TMR3RST**	**Timer3控制器复位** 0 = Timer3控制器正常运行 1 = Timer3控制器复位
[4]	**TMR2RST**	**Timer2控制器复位** 0 = Timer2控制器正常运行 1 = Timer2控制器复位
[3]	**TMR1RST**	**Timer1控制器复位** 0 = Timer1控制器正常运行 1 = Timer1控制器复位
[2]	**TMR0RST**	**Timer0控制器复位** 0 = Timer0控制器正常运行

		1 = Timer0控制器复位
[1]	**GPIORST**	**GPIO控制器复位** 0 = GPIO控制器正常运行 1 = GPIO控制器复位
[0]	**Reserved**	保留

外设复位控制寄存器2（SYS_IPRST2）

设置这些位为1将产生异步复位信号到相应的模块控制器。用户需要将这些位设置为0，以从复位状态释放相应的模块控制器。

寄存器	偏移	R/W	描述	复位值
SYS_IPRST2	SYS_BA+0x10	R/W	外设复位控制寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**BPWM1RST**	**BPWM0RST**	**PWM1RST**	**PWM0RST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**USCI1RST**	**USCI0RST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

位	描述	
[31:20]	**Reserved**	保留
[19]	**BPWM1RST**	**BPWM1控制器复位** 0 = BPWM1控制器正常运行 1 = BPWM1控制器复位
[18]	**BPWM0RST**	**BPWM0控制器复位** 0 = BPWM0控制器正常运行 1 = BPWM0控制器复位
[17]	**PWM1RST**	**PWM1控制器复位** 0 = PWM1控制器正常运行 1 = PWM1控制器复位
[16]	**PWM0RST**	**PWM0控制器复位** 0 = PWM0控制器正常运行 1 = PWM0控制器复位
[15:10]	**Reserved**	保留
[9]	**USCI1RST**	**USCI1控制器复位** 0 = USCI1控制器正常运行 1 = USCI1控制器复位
[8]	**USCI0RST**	**USCI0控制器复位** 0 = USCI0控制器正常运行 1 = USCI0控制器复位

[7:0]	**Reserved**	保留

欠压检测控制寄存器（SYS_BODCTL）

部分SYS-BODCTL控制寄存器值由Flash配置初始化，部分位为写保护位。

寄存器	偏移	R/W	描述	复位值
SYS_BODCTL	SYS_BA+0x18	R/W	欠压检测控制寄存器	0x00XX_038X

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**LVRVL**	**Reserved**			**BODVL**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**LVRDGSEL**			**Reserved**	**BODDGSEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LVREN	**BODOUT**	**BODLPM**	**BODIF**	**BODRSTEN**	**Reserved**		**BODEN**

位	描述	
[31:21]	**Reserved**	保留
[20]	**LVRVL**	**LVR低压复位检测器门限电压选择（写保护）** 默认值由Flash控制器用户配置寄存器 LVRLVSEL（CONFIG0 [29]）设置。 0 = 低压复位门限电压1.6V 1 = 低压复位门限电压1.7V **注1：**此位写保护. 参考SYS_REGLCTL 寄存器。 **注2：**此位只用于特殊情况。 **注3：**由上电复位复位。
[19:17]	**Reserved**	保留
[16]	**BODVL**	**欠压检测门限电压选择 （写保护）** 默认值由Flash控制器用户配置寄存器CBOV（CONFIG0 [21]）决定。 0 = 欠压检测门限电压为2.0V 1 = 欠压检测门限电压为2.5V **注1：**此位写保护. 参考SYS_REGLCTL 寄存器。 **注2：**由上电复位复位。
[15]	**Reserved**	保留

位	描述	
[14:12]	**LVRDGSEL**	**LVR 防抖时间选择 （写保护）** 000 = 无防抖 001 = 64 个系统时钟（HCLK） 010 = 128 个系统时钟（HCLK） 011 = 256 个系统时钟（HCLK） 100 = 512 个系统时钟（HCLK） 101 = 1024 个系统时钟（HCLK） 110 = 2048 个系统时钟（HCLK） 111 = 4096 个系统时钟（HCLK） **注：**这些位写保护。 参考SYS_REGLCTL 寄存器。
[11]	**Reserved**	保留
[10:8]	**BODDGSEL**	**BOD防抖时间选择（写保护）** 000 = BOD 防抖时间是LIRC/4采样时钟 001 = 64个系统时钟（HCLK） 010 = 128个系统时钟（HCLK） 011 = 256个系统时钟（HCLK） 100 = 512个系统时钟（HCLK） 101 = 1024个系统时钟（HCLK） 110 = 2048个系统时钟（HCLK） 111 = 4096个系统时钟（HCLK） **注：**这些位写保护。 参考SYS_REGLCTL 寄存器。
[7]	**LVREN**	**低压复位使能位（写保护）** 当输入电源电压低于LVR设定时，LVR功能将复位芯片。LVR功能默认启用。 0 = 低压复位功能被禁用 1 = 低压复位功能启用 **注1：**使能该位后，延时200us，LVR输出稳定，LVR功能将生效（默认）。 **注2：**这些位写保护。 参考SYS_REGLCTL 寄存器。
[6]	**BODOUT**	**欠压检测器输出状态** 0 = 欠压检测器输出状态为0 它表示检测电压高于BODVL设定值或BODEN为0。 1 = 欠压检测器输出状态为1 它表示检测电压低于BODVL设定值。如果BODEN为0，BOD功能禁用，该位始终为0。
[5]	**BODLPM**	**欠压检测器低功耗模式 （写保护）** 0 = BOD运行在正常模式 （默认） 1 = BOD低功耗模式使能 **注1：**在正常模式，BOD耗电约100uA，低功耗模式可以减少功耗到约1/10，但BOD响应速度变慢。 **注2：**这些位写保护。 参考SYS_REGLCTL 寄存器。

位	描述	
[4]	**BODIF**	**欠压检测中断标志** 0 = BOD没有检测到任何电压从VDD 落下或上升过设定的电压的BODVL 1 = 当BOD电路检测到VDD由高到低跨过BODVL设定电压时或者由低到高跨过时，此位置1，若使能了中断，会触发中断 **注：**此位写1清0。
[3]	**BODRSTEN**	**欠压复位使能位（写保护）** 默认值由flash控制器用户配置寄存器CBORST（CONFIG0[20]）位决定。 0 = 欠压中断功能使能 1 = 欠压复位功能使能 **注1：**当同时使能BOD功能（BODEN高）和BOD复位功能（BODRSTEN高）时，当检测到电压低于设定的值（BODOUT高），BOD将产生信号复位芯片。 当同时使能BOD功能（BODEN高）和BOD中断功能（BODRSTEN低），如果BODOUT 为高，BOD将产生中断。BOD中断将保持直到BODEN设置为0。BOD中断可以通过禁用NVIC BOD中断或禁用BOD功能（置BODEN低）打断。 **注2：**此位写保护。参考SYS_REGLCTL 寄存器。 **注3：**由上电复位复位。
[2:1]	**Reserved**	保留
[0]	**BODEN**	**欠压检测功能使能位（写保护）** 默认值由flash控制器用户配置寄存器CBODEN （CONFIG0 [19]）决定. 0 = 欠压检测功能禁用 1 = 欠压检测功能使能 **注1：**此位写保护。参考SYS_REGLCTL 寄存器。 **注2：**由上电复位复位。

上电复位控制寄存器（SYS_PORCTL）

寄存器	偏移	R/W	描述	复位值
SYS_PORCTL	SYS_BA+0x24	R/W	上电复位控制寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**POROFF**	**上电复位使能位 （写保护）** 上电时，POR电路将产生复位信号复位整块芯片功能，电源上的干扰也可能再次激活POR。用户可以往该区域写入0x5AA5禁用内部POR电路来避免不可预期的干扰引起芯片复位。 当该区域写入其他值，或者芯片通过其他复位源（包括：nRESET, Watchdog, LVR 复位, BOD 复位, ICE复位指令和软件芯片复位）复位，POR功能将再次生效。 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。

GPIOA多功能控制寄存器低字节（SYS_GPA_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPA_MFPL	SYS_BA+0x30	R/W	GPIOA多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PA7MFP				**PA6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PA5MFP				**PA4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PA3MFP				**PA2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA1MFP				**PA0MFP**			

位	描述	
[31:28]	**PA7MFP**	PA.7多功能引脚选择
[27:24]	**PA6MFP**	PA.6多功能引脚选择
[23:20]	**PA5MFP**	PA.5多功能引脚选择
[19:16]	**PA4MFP**	PA.4多功能引脚选择
[15:12]	**PA3MFP**	PA.3多功能引脚选择
[11:8]	**PA2MFP**	PA.2多功能引脚选择
[7:4]	**PA1MFP**	PA.1多功能引脚选择
[3:0]	**PA0MFP**	PA.0多功能引脚选择

GPIOA多功能控制寄存器高字节（SYS_GPA_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPA_MFPH	SYS_BA+0x34	R/W	GPIOA多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PA15MFP				**PA14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PA13MFP				**PA12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PA11MFP				**PA10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA9MFP				**PA8MFP**			

位	描述	
[31:28]	**PA15MFP**	PA.15多功能引脚选择
[27:24]	**PA14MFP**	PA.14多功能引脚选择
[23:20]	**PA13MFP**	PA.13多功能引脚选择
[19:16]	**PA12MFP**	PA.12多功能引脚选择
[15:12]	**PA11MFP**	PA.11多功能引脚选择
[11:8]	**PA10MFP**	PA.10多功能引脚选择
[7:4]	**PA9MFP**	PA.9多功能引脚选择
[3:0]	**PA8MFP**	PA.8多功能引脚选择

GPIOB多功能控制寄存器低字节（SYS_GPB_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPB_MFPL	SYS_BA+0x38	R/W	GPIOB多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PB7MFP				**PB6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PB5MFP				**PB4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PB3MFP				**PB2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB1MFP				**PB0MFP**			

位	描述	
[31:28]	**PB7MFP**	PB.7多功能引脚选择
[27:24]	**PB6MFP**	PB.6多功能引脚选择
[23:20]	**PB5MFP**	PB.5多功能引脚选择
[19:16]	**PB4MFP**	PB.4多功能引脚选择
[15:12]	**PB3MFP**	PB.3多功能引脚选择
[11:8]	**PB2MFP**	PB.2多功能引脚选择
[7:4]	**PB1MFP**	PB.1多功能引脚选择
[3:0]	**PB0MFP**	PB.0多功能引脚选择

GPIOB多功能控制寄存器高字节（SYS_GPB_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	GPIOB多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PB15MFP				**PB14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PB13MFP				**PB12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PB11MFP				**PB10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB9MFP				**PB8MFP**			

位	描述	
[31:28]	**PB15MFP**	PB.15多功能引脚选择
[27:24]	**PB14MFP**	PB.14多功能引脚选择
[23:20]	**PB13MFP**	PB.13多功能引脚选择
[19:16]	**PB12MFP**	PB.12多功能引脚选择
[15:12]	**PB11MFP**	PB.11多功能引脚选择
[11:8]	**PB10MFP**	PB.10多功能引脚选择
[7:4]	**PB9MFP**	PB.9多功能引脚选择
[3:0]	**PB8MFP**	PB.8多功能引脚选择

GPIOC多功能控制寄存器低字节（SYS_GPC_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPC_MFPL	SYS_BA+0x40	R/W	GPIOC多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PC7MFP				**PC6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PC5MFP				**PC4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PC3MFP				**PC2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PC1MFP				**PC0MFP**			

位	描述	
[31:28]	**PC7MFP**	PC.7多功能引脚选择
[27:24]	**PC6MFP**	PC.6多功能引脚选择
[23:20]	**PC5MFP**	PC.5多功能引脚选择
[19:16]	**PC4MFP**	PC.4多功能引脚选择
[15:12]	**PC3MFP**	PC.3多功能引脚选择
[11:8]	**PC2MFP**	PC.2多功能引脚选择
[7:4]	**PC1MFP**	PC.1多功能引脚选择
[3:0]	**PC0MFP**	PC.0多功能引脚选择

GPIOC多功能控制寄存器高字节（SYS_GPC_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPC_MFPH	SYS_BA+0x44	R/W	GPIOC多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PC15MFP				**PC14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PC13MFP				**PC12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PC11MFP				**PC10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PC9MFP				**PC8MFP**			

位	描述	
[31:28]	**PC15MFP**	PC.15多功能引脚选择
[27:24]	**PC14MFP**	PC.14多功能引脚选择
[23:20]	**PC13MFP**	PC.13多功能引脚选择
[19:16]	**PC12MFP**	PC.12多功能引脚选择
[15:12]	**PC11MFP**	PC.11多功能引脚选择
[11:8]	**PC10MFP**	PC.10多功能引脚选择
[7:4]	**PC9MFP**	PC.9多功能引脚选择
[3:0]	**PC8MFP**	PC.8多功能引脚选择

GPIOD多功能控制寄存器低字节（SYS_GPD_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPD_MFPL	SYS_BA+0x48	R/W	GPIOD多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PD7MFP				**PD6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PD5MFP				**PD4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PD3MFP				**PD2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD1MFP				**PD0MFP**			

位	描述	
[31:28]	**PD7MFP**	PD.7多功能引脚选择
[27:24]	**PD6MFP**	PD.6多功能引脚选择
[23:20]	**PD5MFP**	PD.5多功能引脚选择
[19:16]	**PD4MFP**	PD.4多功能引脚选择
[15:12]	**PD3MFP**	PD.3多功能引脚选择
[11:8]	**PD2MFP**	PD.2多功能引脚选择
[7:4]	**PD1MFP**	PD.1多功能引脚选择
[3:0]	**PD0MFP**	PD.0多功能引脚选择

GPIOD多功能控制寄存器高字节（SYS_GPD_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	GPIOD多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PD15MFP				**PD14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PD13MFP				**PD12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PD11MFP				**PD10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD9MFP				**PD8MFP**			

位	描述	
[31:28]	**PD15MFP**	PD.15多功能引脚选择
[27:24]	**PD14MFP**	PD.14多功能引脚选择
[23:20]	**PD13MFP**	PD.13多功能引脚选择
[19:16]	**PD12MFP**	PD.12多功能引脚选择
[15:12]	**PD11MFP**	PD.11多功能引脚选择
[11:8]	**PD10MFP**	PD.10多功能引脚选择
[7:4]	**PD9MFP**	PD.9多功能引脚选择
[3:0]	**PD8MFP**	PD.8多功能引脚选择

GPIOE多功能控制寄存器低字节（SYS_GPE_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPE_MFPL	SYS_BA+0x50	R/W	GPIOE多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PE7MFP				**PE6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PE5MFP				**PE4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PE3MFP				**PE2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PE1MFP				**PE0MFP**			

位	描述	
[31:28]	**PE7MFP**	PE.7多功能引脚选择
[27:24]	**PE6MFP**	PE.6多功能引脚选择
[23:20]	**PE5MFP**	PE.5多功能引脚选择
[19:16]	**PE4MFP**	PE.4多功能引脚选择
[15:12]	**PE3MFP**	PE.3多功能引脚选择
[11:8]	**PE2MFP**	PE.2多功能引脚选择
[7:4]	**PE1MFP**	PE.1多功能引脚选择
[3:0]	**PE0MFP**	PE.0多功能引脚选择

GPIOE多功能控制寄存器高字节（SYS_GPE_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPE_MFPH	SYS_BA+0x54	R/W	GPIOE多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PE15MFP				**PE14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PE13MFP				**PE12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PE11MFP				**PE10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PE9MFP				**PE8MFP**			

位	描述	
[31:28]	**PE15MFP**	PE.15多功能引脚选择
[27:24]	**PE14MFP**	PE.14多功能引脚选择
[23:20]	**PE13MFP**	PE.13多功能引脚选择
[19:16]	**PE12MFP**	PE.12多功能引脚选择
[15:12]	**PE11MFP**	PE.11多功能引脚选择
[11:8]	**PE10MFP**	PE.10多功能引脚选择
[7:4]	**PE9MFP**	PE.9多功能引脚选择
[3:0]	**PE8MFP**	PE.8多功能引脚选择

GPIOF多功能控制寄存器低字节（SYS_GPF_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPF_MFPL	SYS_BA+0x58	R/W	GPIOF多功能控制寄存器低字节	0x0000_00ee

31	30	29	28	27	26	25	24
PF7MFP				**PF6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PF5MFP				**PF4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PF3MFP				**PF2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PF1MFP				**PF0MFP**			

位	描述	
[31:28]	**PF7MFP**	PF.7多功能引脚选择
[27:24]	**PF6MFP**	PF.6多功能引脚选择
[23:20]	**PF5MFP**	PF.5多功能引脚选择
[19:16]	**PF4MFP**	PF.4多功能引脚选择
[15:12]	**PF3MFP**	PF.3多功能引脚选择
[11:8]	**PF2MFP**	PF.2多功能引脚选择
[7:4]	**PF1MFP**	PF.1多功能引脚选择
[3:0]	**PF0MFP**	PF.0多功能引脚选择

GPIOF多功能控制寄存器高字节（SYS_GPF_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPF_MFPH	SYS_BA+0x5C	R/W	GPIOF多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PF15MFP				**PF14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PF13MFP				**PF12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PF11MFP				**PF10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PF9MFP				**PF8MFP**			

位	描述	
[31:28]	**PF15MFP**	PF.15多功能引脚选择
[27:24]	**PF14MFP**	PF.14多功能引脚选择
[23:20]	**PF13MFP**	PF.13多功能引脚选择
[19:16]	**PF12MFP**	PF.12多功能引脚选择
[15:12]	**PF11MFP**	PF.11多功能引脚选择
[11:8]	**PF10MFP**	PF.10多功能引脚选择
[7:4]	**PF9MFP**	PF.9多功能引脚选择
[3:0]	**PF8MFP**	PF.8多功能引脚选择

GPIOG多功能控制寄存器低字节（SYS_GPG_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPG_MFPL	SYS_BA+0x60	R/W	GPIOG多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PG7MFP				**PG6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PG5MFP				**PG4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PG3MFP				**PG2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PG1MFP				**PG0MFP**			

位	描述	
[31:28]	**PG7MFP**	PG.7多功能引脚选择
[27:24]	**PG6MFP**	PG.6多功能引脚选择
[23:20]	**PG5MFP**	PG.5多功能引脚选择
[19:16]	**PG4MFP**	PG.4多功能引脚选择
[15:12]	**PG3MFP**	PG.3多功能引脚选择
[11:8]	**PG2MFP**	PG.2多功能引脚选择
[7:4]	**PG1MFP**	PG.1多功能引脚选择
[3:0]	**PG0MFP**	PG.0多功能引脚选择

GPIOG多功能控制寄存器高字节（SYS_GPG_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPG_MFPH	SYS_BA+0x64	R/W	GPIOG多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PG15MFP				**PG14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PG13MFP				**PG12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PG11MFP				**PG10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PG9MFP				**PG8MFP**			

位	描述	
[31:28]	**PG15MFP**	PG.15多功能引脚选择
[27:24]	**PG14MFP**	PG.14多功能引脚选择
[23:20]	**PG13MFP**	PG.13多功能引脚选择
[19:16]	**PG12MFP**	PG.12多功能引脚选择
[15:12]	**PG11MFP**	PG.11多功能引脚选择
[11:8]	**PG10MFP**	PG.10多功能引脚选择
[7:4]	**PG9MFP**	PG.9多功能引脚选择
[3:0]	**PG8MFP**	PG.8多功能引脚选择

GPIOH多功能控制寄存器低字节（SYS_GPH_MFPL）

寄存器	偏移	R/W	描述	复位值
SYS_GPH_MFPL	SYS_BA+0x68	R/W	GPIOH多功能控制寄存器低字节	0x0000_0000

31	30	29	28	27	26	25	24
PH7MFP				**PH6MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PH5MFP				**PH4MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PH3MFP				**PH2MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PH1MFP				**PH0MFP**			

位	描述	
[31:28]	**PH7MFP**	PH.7多功能引脚选择
[27:24]	**PH6MFP**	PH.6多功能引脚选择
[23:20]	**PH5MFP**	PH.5多功能引脚选择
[19:16]	**PH4MFP**	PH.4多功能引脚选择
[15:12]	**PH3MFP**	PH.3多功能引脚选择
[11:8]	**PH2MFP**	PH.2多功能引脚选择
[7:4]	**PH1MFP**	PH.1多功能引脚选择
[3:0]	**PH0MFP**	PH.0多功能引脚选择

GPIOH多功能控制寄存器高字节（SYS_GPH_MFPH）

寄存器	偏移	R/W	描述	复位值
SYS_GPH_MFPH	SYS_BA+0x6C	R/W	GPIOH多功能控制寄存器高字节	0x0000_0000

31	30	29	28	27	26	25	24
PH15MFP				**PH14MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PH13MFP				**PH12MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PH11MFP				**PH10MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PH9MFP				**PH8MFP**			

位	描述	
[31:28]	**PH15MFP**	PH.15多功能引脚选择
[27:24]	**PH14MFP**	PH.14多功能引脚选择
[23:20]	**PH13MFP**	PH.13多功能引脚选择
[19:16]	**PH12MFP**	PH.12多功能引脚选择
[15:12]	**PH11MFP**	PH.11多功能引脚选择
[11:8]	**PH10MFP**	PH.10多功能引脚选择
[7:4]	**PH9MFP**	PH.9多功能引脚选择
[3:0]	**PH8MFP**	PH.8多功能引脚选择

调制控制寄存器（SYS_MODCTL）

寄存器	偏移	R/W	描述	复位值
SYS_MODCTL	SYS_BA+0xC0	R/W	调制控制寄存器	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留
[7:4]	**MODPWMSEL**	**调制器PWM0通道选择** 选择 PWM0 通道来调制UART0_TXD 或 USCI0_DAT1 0000: 用PWM0通道0调制UART0_TXD 0001: 用PWM0通道1调制UART0_TXD 0010: 用PWM0通道2调制UART0_TXD 0011: 用PWM0通道3调制UART0_TXD 0100: 用PWM0通道4调制UART0_TXD 0101: 用PWM0通道5调制UART0_TXD 0110: 保留 0111: 保留 1000: 用PWM0通道0调制USCI0_DAT1 1001: 用PWM0通道1调制USCI0_DAT1 1010: 用PWM0通道2调制USCI0_DAT1 1011: 用PWM0通道3调制USCI0_DAT1 1100: 用PWM0通道4调制USCI0_DAT1 1101: 用PWM0通道5调制USCI0_DAT1 1110: 保留 1111: 保留 **注：**MODEN （SYS_MODCTL[0]）设为1时，该位有效。
[3:2]	**Reserved**	保留
[1]	**MODH**	**在数据位为高时调制** 选择调制脉冲（PWM0）在UART0_TXD或USCI0_DAT1数据为高或低时调制。 0: 调制脉冲在 UART0_TXD 或 USCI0_DAT1 为低时调制 1: 调制脉冲在 UART0_TXD 或 USCI0_DAT1 为高时调制

[0]	**MODEN**	**调制功能使能位** 该位通过使用PWM0通道来实现USCI0（UCI00_DAT0）或UART0（UART0_TXD）调制输出。 0 = 调制功能禁用 1 = 调制功能使能

系统SRAM BIST测试控制寄存器（SYS_SRAM_BISTCTL）

寄存器	偏移	R/W	描述	复位值
SYS_SRAM_BISTCTL	SYS_BA+0xD0	R/W	系统SRAM BIST测试控制寄存器	0x000X_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					SRS2	SRS1	SRS0
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDMABIST	Reserved		USBBIST	Reserved	FMCBIST	Reserved	SRBIST

位	描述	
[31:19]	**Reserved**	保留
[18]	**SRS2**	**SRAM Bank0 段2 BIST选择（写保护）** 该位定义在进行bist测试时是否选择SRAM的bank0段2（0x2001_0000~0x2001_7FFF）。 0 = 当进行bist测试时，SRAM back0 段2未被选择 1 = 当进行bist测试时，SRAM back0 段2被选择 **注1：**此位写保护。请参考SYS_REGLCTL寄存器。 **注2：**当进行bist测试时至少选择一段SRAM。
[17]	**SRS1**	**SRAM Bank0 段1 BIST选择（写保护）** 该位定义在进行bist测试时是否选择SRAM的bank0段1（0x2000_8000~0x2000_FFFF）。 0 = 当进行bist测试时，SRAM back0 段1未被选择 1 = 当进行bist测试时，SRAM back0 段1被选择 **注1：**此位写保护。请参考SYS_REGLCTL寄存器。 **注2：**当进行bist测试时至少选择一段SRAM。
[16]	**SRS0**	**SRAM Bank0 段0 BIST选择（写保护）** 该位定义在进行bist测试时是否选择SRAM的bank0段0（0x2000_0000~0x2000_7FFF）。 0 = 当进行bist测试时，SRAM back0 段0未被选择 1 = 当进行bist测试时，SRAM back0 段0被选择 **注1：**此位写保护。请参考SYS_REGLCTL寄存器。 **注2：**当进行bist测试时至少选择一段SRAM。
[15:8]	**Reserved**	保留
[7]	**PDMABIST**	**PDMA BIST 使能位（写保护）**

		此位使能 PDMA RAM BIST测试。 0 = 系统PDMA BIST禁止 1 = 系统PDMA BIST使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[6:5]	**Reserved**	保留
[4]	**USBBIST**	**USB BIST使能位（写保护）** 此位使能 USB RAM BIST测试。 0 = 系统USB BIST 禁止 1 = 系统USB BIST 使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[3]	**Reserved**	保留
[2]	**FMCBIST**	**FMC缓存BIST使能位（写保护）** 此位使能 CACHE RAM BIST测试。 0 = 系统缓存BIST禁止 1 = 系统缓存BIST 使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[1]	**Reserved**	保留
[0]	**SRBIST**	**SRAM BIST使能位（写保护）** 此位使能 SRAM BIST测试。 0 = 系统SRAM BIST 禁止 1 = 系统SRAM BIST 使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。

系统SRAM BIST测试状态寄存器（SYS_SRAM_BISTSTS）

寄存器	偏移	R/W	描述	复位值
SYS_SRAM_BISTSTS	SYS_BA+0xD4	R	系统SRAM BIST测试状态寄存器	0x00xx_00xx

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PDMAEND	Reserved		USBBEND	Reserved		CRBEND	SRBEND
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDMABISTF	Reserved		USBBEF	Reserved		CRBISTEF	SRBISTEF

位	描述	
[31:24]	**Reserved**	保留
[23]	**PDMAEND**	**PDMA SRAM BIST测试结束** 0 = PDMA SRAM BIST激活 1 = PDMA SRAM BIST测试结束
[22:21]	**Reserved**	保留
[20]	**USBBEND**	**USB SRAM BIST测试结束** 0 = USB SRAM BIST激活 1 = USB SRAM BIST测试结束
[19:18]	**Reserved**	保留
[17]	**CRBEND**	**缓存SRAM BIST测试结束** 0 = 系统缓存RAM BIST激活 1 = 系统缓存RAM BIST测试结束
[16]	**SRBEND**	**系统SRAM BIST测试结束** 0 = 系统SRAM BIST激活 1 = 系统SRAM BIST完成
[15:8]	**Reserved**	保留
[7]	**PDMABISTF**	**PDMA SRAM BIST失败标志** 0 = PDMA SRAM BIST测试通过 1 = PDMA SRAM BIST测试失败
[6:5]	**Reserved**	保留
[4]	**USBBEF**	**USB SRAM BIST失败标志** 0 = USB SRAM BIST测试通过

		1 = USB SRAM BIST 测试失败
[3:2]	**Reserved**	保留
[1]	**CR0BISTEF**	**缓存SRAM BIST失败标志** 0 = 系统缓存RAM BIST测试通过 1 = 系统缓存RAM BIST测试失败
[0]	**SRBISTEF**	**System SRAM BIST失败标志** 0 = 系统SRAM BIST测试通过 1 = 系统SRAM BIST测试失败

系统SRAM奇偶校验错误中断使能控制寄存器（SYS_SRAM_INTCTL）

寄存器	偏移	R/W	描述	复位值
SYS_SRAM_INTCTL	SYS_BA+0xDC	R/W	系统SRAM中断使能控制寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**PERRIEN**	**SRAM奇偶校验错误中断使能位** 0 = SRAM奇偶校验错误中断禁止 1 = SRAM奇偶校验错误中断使能

系统SRAM奇偶校验检测状态寄存器（SYS_SRAM_STATUS）

寄存器	偏移	R/W	描述	复位值
SYS_SRAM_STATUS	SYS_BA+0xE0	R/W	系统SRAM奇偶校验错误状态寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**PERRIF**	**SRAM 奇偶校验错误标志** 该位指示系统SRAM奇偶校验错误发生。该位写1清0。 0 = 无系统SRAM奇偶错误 1 = 发生系统SRAM奇偶错误

系统SRAM奇偶错误地址寄存器（SYS_SRAM_ERRADDR）

寄存器	偏移	R/W	描述	复位值
SYS_SRAM_ERRADDR	SYS_BA+0xE4	R	系统SRAM奇偶校验错误地址寄存器	0x0000_0000

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

位	描述	
[31:0]	**ERRADDR**	**系统SRAM奇偶错误地址** 该寄存器显示系统SRAM奇偶错误位地址。

HIRC校准控制寄存器（SYS_HIRCTRIMCTL）

寄存器	偏移	R/W	描述	复位值
SYS_HIRCTRIMCTL	SYS_BA+0xF0	R/W	HIRC校准控制寄存器	0x0008_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**BOUNDARY**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**REFCKSEL**	**BOUNDEN**	**CESTOPEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RETRYCNT		**LOOPSEL**		**Reserved**		**FREQSEL**	

位	描述	
[31:21]	**Reserved**	保留
[20:16]	**BOUNDARY**	**边界选择** 填入边界范围从 0x1 到 0x1F, 0x0 保留。 **注:** 只有当边界BOUNDEN（SYS_HIRCTRIMCTL[9]） 使能时，该字段才有效。
[15:11]	**Reserved**	保留
[10]	**REFCKSEL**	**参考时钟选择** 0 = HIRC 校准参考时钟来自 LXT （32.768 kHz） 1 = HIRC 校准参考时钟来自内部USB 同步模式 **注1:** HIRC 校准参考时钟支持 LXT 或内部 USB 同步模式取决于芯片规格。请参照第 4.3章节 NuMicro® M031/M032 系列选型表获取详细信息。 **注2:** 如果没有参考时钟 （LXT 或者内部 USB 同步模式） 当RC校准使能时，CLKERIF（SYS_HIRCTRIMCTL[2]） 将会被置1。
[9]	**BOUNDEN**	**边界使能位** 0 = 边界功能禁止 1 = 边界功能使能
[8]	**CESTOPEN**	**时钟错误停止使能位** 0 = 若时钟错误，保持校准继续 1 = 若时钟错误，停止校准
[7:6]	**RETRYCNT**	**校准更新次数限制** 这个值决定了在 HIRC 锁定之前，HIRC计算更新几次。 一旦HIRC锁定，内部校准更新次数计数器会复位。 如果校准值更新计数器达到这个限制值且HIRC频率仍然没有锁定，自动校准操作会禁止且FREQSEL会清0。 00 = 校准重试次数限制在64次

		01 = 校准重试次数限制在128次 10 = 校准重试次数限制在256次 11 = 校准重试次数限制在512次
[5:4]	**LOOPSEL**	**配置几个参考时钟做一次校准计算** 此字段定义校准值计算基于多少个参考时钟。 00 = 4 个参考时钟做一次校准计 01 = 8个参考时钟做一次校准计 10 = 16个参考时钟做一次校准计 11 = 32个参考时钟做一次校准计 **注：**例如，如果 LOOPSEL 设置为 00，自动校准周期将会计算校准值基于不同的4个参考时钟平均频率。
[3:2]	**Reserved**	保留
[1:0]	**FREQSEL**	**校准频率选择** 该区域定义了内部高速RC振荡器0（HIRC0）自动校准的目标频率。 在自动校验运行期间，如果CESTOPEN（SYS_IRCTCTL1[8]）设置为1且检测到时钟错误或者到达校验限制次数，该区域将被自动清00。 00 = 禁止HIRC自动校准功能 01 = 使能HIRC自动校准功能和校准HIRC到48Mhz 10 = 保留 11 = 保留

HIRC校准中断使能寄存器（SYS_HIRCTRIMIEN）

寄存器	偏移	R/W	描述	复位值
SYS_HIRCTRIMIEN	SYS_BA+0xF4	R/W	HIRC校准中断使能寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**CLKEIEN**	**时钟错误中断使能位** 该位控制在自动校准运行期间, 时钟频率不准确时,CPU是否会接收到中断。 如果该位设置为1，且CLKERRIF（SYS_HIRCTSTS[2]）在自动校准运算期间置位，中断将被触发表明时钟频率不准确。 0 = 禁用 CLKERRIF（SYS_HIRCTRIMSTS[2]）状态触发一个中断给 CPU 1 = 使能 CLKERRIF（SYS_HIRCTRIMSTS[2]）状态触发一个中断给 CPU
[1]	**TFALIEN**	**校准失败中断使能位** 该位控制当达到HIRC校验值更新限制次数且HIRC频率未锁定到FREQSEL（SYS_HIRCTRIMSTS [1:0]）设置的目标频率是否触发中断。 在自动校验运算期间，如果该位为高且TFAILIF（SYS_ HIRCTRIMSTS [1]）被设置，将产生中断，表明到达HIRC校验值限制次数。 0 = 禁用 TFAILIF（SYS_HIRCTRIMSTS[1]） 状态触发一个中断给CPU 1 = 使能 TFAILIF（SYS_HIRCTRIMSTS[1]） 状态触发一个中断给CPU
[0]	**Reserved**	保留

HIRC校准中断状态寄存器（SYS_HIRCTRIMSTS）

寄存器	偏移	R/W	描述	复位值
SYS_HIRCTRIMSTS	SYS_BA+0xF8	R/W	HIRC校准中断状态寄存器	0x0000_0000

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

位	描述	
[31:4]	**Reserved**	保留
[3]	**OVBDIF**	**超出边界状态** 当超出边界功能被设置，如果出现超出边界的情形，这个标志将会被设置。 0 = 超出边界情形未发生 1 = 超出边界情形发生 **注:** 写1清除此标志
[2]	**CLKERIF**	**时钟错误中断状态** 当参考时钟（LXT或USB同步信号）和48MHz内部高速RC振荡器（HIRC）之间的频率关系偏差到较大的不合理值时，将设置该位并且指示时钟频率不准确。一旦该位设置为1，自动校准操作停止，如果CESTOPEN（SYS_HIRCTRIMCTL[8]）设置为1，则硬件将自动将FREQSEL（SYS_HIRCTRIMCTL[1:0]）清除为0。 如果设置了该位并且CLKEIEN（SYS_HIRCTIEN[2]）为1,将触发一个中断来通知时钟频率不准确。写1清除该为0。 0 = 时钟频率准确 1 = 时钟频率不准确 **注**：由上电复位复位。
[1]	**TFAILIF**	**校准失败中断状态** 该位表示到达HIRC校准值更新限制次数且HIRC时钟频率未锁定。一旦该位被设置，自动校准运算将停止且FREQSEL（SYS_HIRCTRIMCTL[1:0]）将被硬件自动清0。 如果该位被设置且TFAILIEN（SYS_HIRCIEN[1]）为1,中断将被触发表明到达HIRC校准值更新限制次数。该位写1清除为0。 0 = 未到达HIRC校验值更新限制次数 1 = 到达HIRC校验值更新限制次数且频率未锁定 **注**：由上电复位复位。
[0]	**FREQLOCK**	**HIRC 频率锁定状态** 该位表明HIRC频率锁定。

		该位是一个状态位，不会触发任何中断。 该位写1清0。此位会被自动设置, 如果频率是锁定的且RC_TRIM被使能。 0 = 内部高速RC振荡器频率未锁定在48MHz 1 = 内部高速RC振荡器频率锁定在48MHz **注**：由上电复位复位。

寄存器锁控制寄存器（SYS_REGLCTL）

部分系统控制寄存器需要保护，以避免无意写入和干扰芯片运行。这些系统控制寄存器在通电复位后被保护，直到用户禁用寄存器保护。用户如果想要编程这些保护寄存器，必须通过特定的编程来禁用寄存器保护。禁用寄存器保护的方法是依次往地址为0x4000_0100的寄存器SYS_REGLCTL写入“59h”, “16h” “88h”。 任何不同的数据、不同的写入顺序或者在这三个数据写入的过程中，写其它地址，将打断禁用寄存器保护方法流程。

禁用保护后，用户可以在地址0x4000_0100位0检查保护禁用位,1是保护禁用,0是保护启用.然后,用户可以更新目标受保护寄存器值，然后将任何数据写入地址“0x4000_0100”以启用寄存器保护。

该寄存器被写入禁用/启用寄存器保护,并读取REGLCTL状态。

寄存器	偏移	R/W	描述	复位值
SYS_REGLCTL	SYS_BA+0x100	R/W	寄存器锁控制寄存器	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**REGLCTL**	**寄存器锁控制码 （只写）** 有些寄存器具有写保护功能。写入这些寄存器必须通过将序列值“59h”、“16h”、“88h”写入该字段来禁用受保护函数。完成此序列后，将REGLCTL位设置为1，写保护寄存器可以正常写入 **REGLCTL[0]** **寄存器锁控制禁用索引 （只读）** 0 =写保护寄存器写保护使能，对保护寄存器的任何写操作都被忽略 1 =写保护寄存器写保护禁用

模拟POR禁用控制寄存器（SYS_PORDISAN）

寄存器	偏移	R/W	描述	复位值
SYS_PORDISAN	SYS_BA+0x1EC	R/W	模拟POR禁用控制寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**POROFFAN**	**上电复位使能位 （写保护）** 上电后，用户可以通过将0x5AA5写入该字段来关闭内部模拟POR电路以节省电力。 当该区域设置为其他值，或者芯片被其他复位源复位时，模拟POR电路将再次激活，包括: nRESET, 看门狗, LVR 复位, BOD 复位, ICE 复位命令和软件芯片复位功能。 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。

6.3.14 系统定时器（SysTick）

Cortex®-M0包含一个完整的系统定时器，系统定时器提供一个简单的、24位写入清零、递减、带灵活控制机制的自动装载计数器。该计数器可以用作实时操作系统（RTOS）的滴答定时器或作为一个简单的计数。

当系统定时器使能后，将从SysTick当前值寄存器（SYST_VAL）的值向下递减到0，并在下一个时钟周期，重新加载SysTick重加载值寄存器（SYST_LOAD）的值，然后再随时钟递减。当计数器递减到0，COUNTFLAG状态位将被设置，COUNTFLAG位读时清0。

复位时，SYST_ VAL的值是未知的。使能前，软件应向该寄存器写入值来清0。这样保证定时器会从SYST_LOAD值开始计数而不是一个随机值。

如果SYST_LOAD为0，定时器在重新加载后，将保持值为0。这种机制可以用来在定时器使能后，独立地禁用计数功能。

更多的介绍，请参考“Arm® Cortex®-M0 技术参考手册” 和 “Arm® v6-M 架构参考手册”。

6.3.14.1 系统时钟控制寄存器映射

R: 只读，**W**: 只写，**R/W**

寄存器	偏移	R/W	描述	复位值
SYST 基地址： **SCS_BA = 0xE000_E000**				
SYST_CTRL	SCS_BA+0x10	R/W	系统时钟控制和状态寄存器	0x0000_0000
SYST_LOAD	SCS_BA+0x14	R/W	系统时钟重载值寄存器	0xXXXX_XXXX
SYST_VAL	SCS_BA+0x18	R/W	系统定时器当前值寄存器	0xXXXX_XXXX

6.3.14.2 系统定时器控制寄存器描述

系统定时器控制和状态寄存器（SYST_CTRL）

寄存器	偏移	R/W	描述	复位值
SYST_CTRL	SCS_BA+0x10	R/W	系统定时器控制和状态寄存器	0x0000_0000

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

位	描述	
[31:17]	**Reserved**	保留
[16]	**COUNTFLAG**	**系统滴答计数器标志** 从上次该寄存器被读后，如果定时器计数到0，返回1。 计数从1到0转变时，COUNTFLAG置位 读该位或写当前值寄存器，COUNTFLAG将清0
[15:3]	**Reserved**	保留
[2]	**CLKSRC**	**系统滴答时钟源选择** 0 = 时钟源为（可选）外部参考时钟 1 = 内核时钟用于SysTick
[1]	**TICKINT**	**系统滴答中断使能** 0 = 计数到0不引起SysTick异常而挂起。软件可以用COUNTFLAG标志来判断是否已经发生计数到0的情况 1 = 计数到0将引起SysTick异常而挂起。软件通过写寄存器清除SysTick当前值寄存器将不会引起SysTick挂起
[0]	**ENABLE**	**系统滴答计数器使能** 0 = 计数器禁用 1 = 计数器将通过multi-shot方式运行

系统定时器重载值寄存器（SYST_LOAD）

寄存器	偏移	R/W	描述	复位值
SYST_LOAD	SCS_BA+0x14	R/W	系统定时器重载值寄存器	0xXXXX_XXXX

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

位	描述	
[31:24]	**Reserved**	保留
[23:0]	**RELOAD**	**系统滴答重载值** 当计数器到0时，这个值就装载到当前值寄存器。

系统定时器当前值寄存器（SYST_VAL）

寄存器	偏移	R/W	描述	复位值
SYST_VAL	SCS_BA+0x18	R/W	系统定时器当前值寄存器	0xXXXX_XXXX

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

位	描述	
[31:24]	**Reserved**	保留
[23:0]	**CURRENT**	**系统滴答当前值** 当前计数器值，该值为采样时刻的计数器的值。该计数器不提供读修改写保护功能，该寄存器写入清0。软件往该寄存器写任何值都会将其清0。

6.3.15 可嵌套向量中断控制器（NVIC）

Cortex®-M0 提供一个中断控制器用于异常模式，称之为“嵌套向量中断控制器（NVIC）”, NVIC与处理器内核紧密相连，且提供以下特性:

- 支持嵌套向量中断
- 自动保存和恢复处理器状态
- 简化和确定的中断延时

NVIC 按照优先级处理所有支持的异常。所有异常在“Handler模式”处理。NVIC架构支持32（IRQ[31:0]）个离散的中断，每个中断有4级优先级。所有的中断和大部分异常可以被设置为不同的优先级。当中断发生时，NVIC会比较新的中断的优先级和当前中断的优先级，如果新中断的优先级比当前中断的优先级高，将立即处理新的中断。

当一个中断接受后，中断服务例程（ISR）的起始地址可从内存中的向量表取得。软件不需要决定哪个中断被响应，也不用分配相关ISR的起始地址。当开始地址取得后，NVIC将自动保存包含寄存器“PC, PSR, LR, R0~R3, R12”值的处理器状态到栈中。在ISR结束后，NVIC将从栈中恢复相关寄存器的值，并运行正常状态。因此花费少量且确定的时间处理中断请求。

NVIC支持“Tail Chaining”模式，可以有效的处理背对背中断，即无需保存和恢复当前状态，从而减少结束当前ISR切换到挂起ISR的延迟时间。NVIC还支持“Late Arrival”模式，因此可以提高并发ISRs的效率。当较高优先级中断请求发生在当前ISR开始执行之前（保存处理器状态和获取起始地址阶段），NVIC将立即处理更高优先级的中断，从而提高实时性。

更多的介绍，请参考“Arm® Cortex®-M0 技术参考手册” 和 “Arm® v6-M 架构参考手册”。

6.3.15.1 异常模式和系统中断映射

表 6.3-8列出了M031/M032系列支持的异常模式.与所有中断一样，软件可以为其中一些异常设置4级优先级.最高用户可配置优先级为0，最低优先级为3，所有用户可配置中断的默认优先级为0. 注：优先级0在系统为第4级优先级，排在“Reset”, “NMI” 和 “Hard Fault”这三个系统异常之后。

异常名称	向量号	优先级
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4 ~ 10	保留
SVCall	11	可配置
Reserved	12 ~ 13	保留
PendSV	14	可配置
SysTick	15	可配置
Interrupt（IRQ0 ~ IRQ31）	16 ~ 47	可配置

表 6.3-8 异常模式

向量号	中断号 （中断寄存器对应位）	中断名	中断描述
0 ~ 15	-	-	系统异常
16	0	BODOUT	欠压检测中断
17	1	WDT_INT	看门狗定时器中断
18	2	EINT024	来自EINT0,2,4外部中断
19	3	EINT135	来自EINT1.3.5外部中断
20	4	GPABGH_INT	来自 PA, PB, PG, PH引脚外部中断
21	5	GPCDEF_INT	来自PC, PD, PE, PF引脚外部中断
22	6	PWM0_INT	PWM0中断
23	7	PWM1_INT	PWM1中断
24	8	TMR0_INT	Timer 0中断
25	9	TMR1_INT	Timer 1中断
26	10	TMR2_INT	Timer 2中断
27	11	TMR3_INT	Timer 3中断
28	12	UART02_INT	UART0,2中断
29	13	UART13_INT	UART1,3中断
30	14	SPI0_INT	SPI0中断
31	15	QSPI0_INT	QSPI0中断
32	16	Reserved	保留
33	17	UART57_INT	UART5,7中断
34	18	I2C0_INT	I2C0中断
35	19	I2C1_INT	I2C1中断
36	20	BPWM0_INT	BPWM0中断
37	21	BPWM1_INT	BPWM1中断
38	22	USCI01	USCI0,1中断
39	23	USBD_INT	USB设备中断
40	24	Reserved	保留
41	25	ACMP01_INT	ACMP0和ACMP1中断
42	26	PDMA_INT	PDMA中断
43	27	UART46_INT	UART4,6中断
44	28	PWRWU_INT	芯片从掉电状态唤醒时钟控制中断
45	29	ADC_INT	ADC中断

46	30	CLKFAIL	时钟失败侦测和IRC自动校准中断和SRAM奇偶校验错误中断
47	31	RTC_INT	RTC中断

表 6.3-9 中断号码表

6.3.15.2 向量表

当中断响应时，处理器将自动从内存中的向量表中提取中断服务例程（ISR）的起始地址。对于ARMV6M，向量表基地址固定在0x00000000。向量表包含复位时堆栈指针的初始化值和所有异常处理程序的入口点地址。上一节上的向量号定义与异常处理程序条目关联的向量表中的条目的顺序，如上一节所示。

向量表偏移号	描述
0	SP_main – 主栈指针
向量号	使用向量号标示的异常入口指针

表 7.2-10 向量表格式

6.3.15.3 操作描述

NVIC中断可以通过写使能中断或清使能中断寄存器相关位来使能或禁用。这些寄存器使用写1使能和写1清零机制，通过读取这些寄存器可以读取当前相应中断的使能状态。当中断禁用后，中断声明将使中断挂起，因此中断不被激活。如果在禁用中断时中断被激活，它将保持激活状态直到被复位或异常返回清除。清使能位可以阻止相应中断的新中断被激活。

NVIC中断可以使用互补寄存器对来挂起/取消挂起来使能/禁用中断，这些寄存器分别为设置挂起寄存器和清挂起寄存器。这些寄存器使用写1使能和写1清零机制，通过读取这些寄存器可以读取当前相应中断的挂起状态。清挂起寄存器寄存器不影响激活中断的执行状态。

NVIC中断的优先级可以通过更新32位寄存器的8个位段来设置（每个寄存器支持4个中断）。

与NVIC相关的通用寄存器在系统控制空间的一块内存中设置，下一节将做出描述。

6.3.15.4 NVIC控制寄存器

R: 只读，**W:** 只写，**R/W：可读写**

寄存器	偏移	R/W	描述	复位值
NVIC基地址： **NVIC_BA = 0xE000_E100**				
NVIC_ISER0	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31设置使能控制寄存器	0x0000_0000
NVIC_ICER0	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31清除使能控制寄存器	0x0000_0000
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31设置挂起控制寄存器	0x0000_0000
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31清除挂起控制寄存器	0x0000_0000
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ31激活位寄存器	0x0000_0000
NVIC_IPRn **n=0,1..7**	0xE000E400 +0x4*n	R/W	IRQ0 ~ IRQ31优先级控制寄存器	0x0000_0000
STIR	0xE000EF00	R/W	软件触发中断寄存器	0x0000_0000

IRQ0 ~ IRQ31设置使能控制寄存器（NVIC_ISER0）

寄存器	偏移	R/W	描述	复位值
NVIC_ISER0	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31设置使能控制寄存器	0x0000_0000

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

位	描述	
[31:0]	**SETENA**	**中断设置使能位** NVIC_ISER0寄存器使能中断，并显示哪个中断被使能。 读操作： 0 = 无效 1 = 中断使能 写操作： 0 = 中断禁止 1 = 中断使能

IRQ0 ~ IRQ31清除使能控制寄存器（NVIC_ICER0）

寄存器	偏移	R/W	描述	复位值
NVIC_ICER0	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31清除使能控制寄存器	0x0000_0000

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

位	描述	
[31:0]	**CALENA**	**中断清除使能位** NVIC_ISER0寄存器禁止中断，并显示哪个中断被使能。 读操作： 0 = 无效 1 = 中断禁止 写操作： 0 = 中断禁止 1 = 中断使能

IRQ0 ~ IRQ31设置挂起控制寄存器（NVIC_ISPR0）

寄存器	偏移	R/W	描述	复位值
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31设置挂起控制寄存器	0x0000_0000

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

位	描述	
[31:0]	**SETPEND**	**中断设置挂起** NVIC_ISER0寄存器强制中断进入挂起状态，并显示哪个中断被挂起。 读操作： 0 = 无效 1 = 改变中断状态为挂起 写操作： 0 = 中断未挂起 1 = 中断挂起

IRQ0 ~ IRQ31清除挂起控制寄存器（NVIC_ICPR0）

寄存器	偏移	R/W	描述	复位值
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31清除挂起控制寄存器	0x0000_0000

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

位	描述	
[31:0]	**CALPEND**	**中断清除挂起** NVIC_ISER0寄存器移除中断挂起状态，并显示哪个中断被挂起。 读操作： 0 = 无效 1 = 移除一个中断挂起状态 写操作： 0 = 中断未挂起 1 = 中断挂起

IRQ0 ~ IRQ31激活位寄存器（NVIC_IABR0）

寄存器	偏移	R/W	描述	复位值
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ31激活位寄存器	0x0000_0000

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

位	描述	
[31:0]	**ACTIVE**	**中断激活标志** NVIC_IABR0寄存器指示哪个中断是激活的。 0 = 中断未激活 1 = 中断激活

IRQ0 ~ IRQ31中断优先级寄存器（NVIC_IPRn）

寄存器	偏移	R/W	描述	复位值
NVIC_IPRn n=0,1..7	0xE000E400 +0x4*n	R/W	IRQ0 ~ IRQ31中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_4n_3		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_4n_2		**Reserved**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_4n_1		**Reserved**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_4n_0		**Reserved**					

位	描述	
[31:30]	**PRI_4n_3**	**IRQ_4n+3优先级** 0” 表示最高优先级 “3” 表示最低优先级
[29:24]	**Reserved**	保留
[23:22]	**PRI_4n_2**	**IRQ_4n+2优先级** 0” 表示最高优先级 “3” 表示最低优先级
[21:16]	**Reserved**	保留
[15:14]	**PRI_4n_1**	**IRQ_4n+1优先级** 0” 表示最高优先级 “3” 表示最低优先级
[13:8]	**Reserved**	保留
[7:6]	**PRI_4n_0**	**IRQ_4n+0优先级** 0” 表示最高优先级 “3” 表示最低优先级
[5:0]	**Reserved**	保留

软件触发中断寄存器（STIR）

寄存器	偏移	R/W	描述	复位值
STIR	0xE000EF00	R/W	软件触发中断寄存器	0x0000_0000

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

位	描述	
[31:9]	**Reserved**	保留
[8:0]	**INTID**	**中断ID** 软件写STIR触发一个中断 当SCR的USERSETMPEND 位为 1, 非特权软件可以访问STIR 中断ID触发，在0～31范围内触发。例如，0x03的值指定中断IRQ3

6.3.15.5 非屏蔽中断控制寄存器

R: 只读，**W:** 只写，**R/W**：可读写

寄存器	偏移	R/W	描述	复位值
NMI基地址： **NMI_BA = 0x4000_0300**				
NMIEN	NMI_BA+0x00	R/W	非屏蔽中断使能寄存器	0x0000_0000
NMISTS	NMI_BA+0x04	R	非屏蔽中断状态寄存器	0x0000_0000

非屏蔽中断使能寄存器（NMIEN）

寄存器	偏移	R/W	描述	复位值
NMIEN	NMI_BA+0x00	R/W	非屏蔽中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART1_INT	**UART0_INT**	**EINT5**	**EINT4**	**EINT3**	**EINT2**	**EINT1**	**EINT0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**RTC_INT**	**Reserved**	**CLKFAIL**	**SRAM_PERR**	**PWRWU_INT**	**IRC_INT**	**BODOUT**

位	描述	
[31:16]	**Reserved**	保留
[15]	**UART1_INT**	**UART1 非屏蔽源使能（写保护）** 0 = UART1非屏蔽源禁止 1 = UART1 非屏蔽源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[14]	**UART0_INT**	**UART0 非屏蔽源使能（写保护）** 0 = UART0非屏蔽源禁止 1 = UART0非屏蔽源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[13]	**EINT5**	**PB.7, PD.12 或PF.14脚外部中断非屏蔽中断源使能（写保护）** 0 = PB.7, PD.12 或PF.14脚外部中断非屏蔽中断源禁止 1 = PB.7, PD.12 或PF.14脚外部中断非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[12]	**EINT4**	**PA.8, PB.6或PF.15 脚外部中断非屏蔽中断源使能（写保护）** 0 = PA.8, PB.6或PF.15 脚外部中断非屏蔽中断源禁止 1 = PA.8, PB.6或PF.15 脚外部中断非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[11]	**EINT3**	**PB.2或PC.7 脚外部中断非屏蔽中断源使能（写保护）** 0 = PB.2或PC.7 脚外部中断非屏蔽中断源禁止 1 = PB.2或PC.7 脚外部中断非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[10]	**EINT2**	**PB.3或PC.6 脚外部中断非屏蔽中断源使能（写保护）** 0 = PB.3或PC.6 脚外部中断非屏蔽中断源禁止 1 = PB.3或PC.6 脚外部中断非屏蔽中断源使能

		注： 此位写保护。请参考SYS_REGLCTL寄存器。
[9]	**EINT1**	**PA.7, PB.4或PD.15 脚外部中断非屏蔽中断源使能（写保护）** 0 = PA.7, PB.4或PD.15 脚外部中断非屏蔽中断源禁止 1 = PA.7, PB.4或PD.15 脚外部中断非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[8]	**EINT0**	**PA.6或PB.5 脚外部中断非屏蔽中断源使能（写保护）** 0 = PA.6或PB.5 脚外部中断非屏蔽中断源禁止 1 = PA.6或PB.5 脚外部中断非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[7]	**Reserved**	保留
[6]	**RTC_INT**	**RTC非屏蔽中断源使能（写保护）** 0 = RTC非屏蔽源禁止 1 = RTC非屏蔽源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[5]	**Reserved**	保留
[4]	**CLKFAIL**	**时钟失效检测和IRC自动校准中断非屏蔽中断源使能（写保护）** 0 = 禁止时钟失效检测和IRC自动校准中断非屏蔽中断源 1 = 使能时钟失效检测和IRC自动校准中断非屏蔽中断源 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[3]	**SRAM_PERR**	**SRAM奇偶校验错误非屏蔽中断使能（写保护）** 0 = SRAM奇偶校验错误非屏蔽中断源禁止 1 = SRAM 奇偶校验错误非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[2]	**PWRWU_INT**	**掉电模式下唤醒非屏蔽中断源使能（写保护）** 0 = 掉电模式唤醒非屏蔽中断源禁止 1 = 掉电模式唤醒非屏蔽中断源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[1]	**IRC_INT**	**IRC校准非屏蔽中断源使能（写保护）** 0 = IRC TRIM非屏蔽源禁止 1 = IRC TRIM非屏蔽源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。
[0]	**BODOUT**	**BOD非屏蔽中断源使能（写保护）** 0 = BOD非屏蔽源禁止 1 = BOD非屏蔽源使能 **注：** 此位写保护。请参考SYS_REGLCTL寄存器。

非屏蔽中断状态寄存器（NMISTS）

寄存器	偏移	R/W	描述	复位值
NMISTS	NMI_BA+0x04	R	非屏蔽中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART1_INT	**UART0_INT**	**EINT5**	**EINT4**	**EINT3**	**EINT2**	**EINT1**	**EINT0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**RTC_INT**	**Reserved**	**CLKFAIL**	**SRAM_PERR**	**PWRWU_INT**	**IRC_INT**	**BODOUT**

位	描述	
[31:16]	**Reserved**	保留
[15]	**UART1_INT**	**UART1中断标志（只读）** 0 = UART1未发生中断 1 = UART1发生中断
[14]	**UART0_INT**	**UART0中断标志（只读）** 0 = UART1未发生中断 1 = UART1发生中断
[13]	**EINT5**	**PB.7 或PF.14 脚外部中断标志（只读）** 0 = PB.7 或 PF.14 未发生中断 1 = PB.7 或PF.14 发生中断
[12]	**EINT4**	**PA.8, PB.6 或PF.15引脚外部中断标志（只读）** 0 = PA.8, PB.6 或PF.15 未发生中断 1 = PA.8, PB.6 或PF.15 发生中断
[11]	**EINT3**	**PB.2 或PC.7引脚外部中断标志（只读）** 0 = PB.2或PC.7 未发生中断 1 = PB.2或PC.7 发生中断
[10]	**EINT2**	**PB.3或PC.6引脚外部中断标志（只读）** 0 = PB.3或PC.6 未发生中断 1 = PB.3或PC.6 发生中断
[9]	**EINT1**	**PA.7, PB.4 或PD.15引脚外部中断标志（只读）** 0 = PA.7, PB.4或PD.15 未发生中断 1 = PA.7, PB.4或PD.15 发生中断
[8]	**EINT0**	**PA.6 或PB.5引脚外部中断标志（只读）**

		0 = PA.6或PB.5 未发生中断 1 = PA.6或PB.5 发生中断
[7]	**Reserved**	保留
[6]	**RTC_INT**	**RTC中断标志（只读）** 0 = RTC 未发生中断 1 = RTC 发生中断
[5]	**Reserved**	保留
[4]	**CLKFAIL**	**时钟失效检测或IRC自动校准中断标志（只读）** 0 = 时钟失效检测或IRC自动校准未发生中断 1 = 时钟失效检测或IRC自动校准发生中断
[3]	**SRAM_PERR**	**SRAM奇偶校验错误中断标志（只读）** 0 = SRAM奇偶校验错误未发生中断 1 = SRAM 奇偶校验错误发生中断
[2]	**PWRWU_INT**	**掉电模式下唤醒中断标志（只读）** 0 = 掉电模式下唤醒未发生中断 1 = 掉电模式下唤醒发生中断
[1]	**IRC_INT**	**IRC校准中断标志（只读）** 0 = HIRC校准未发生中断 1 = HIRC校准发生中断
[0]	**BODOUT**	**BOD中断标志（只读）** 0 = BOD 未发生中断 1 = BOD 发生中断

6.3.16 系统控制寄存器

Cortex®-M0状态和运行模式控制由系统控制寄存器管理。通过这些系统控制寄存器可以控制CPUID, Cortex®-M0中断优先级和Cortex®-M0电源管理。

更多的介绍，请参考“*Arm® Cortex®-M0 技术参考手册*” 和 “*Arm® v6-M 架构参考手册*”。

R: 只读，**W:** 只写，**R/W**

寄存器	偏移	R/W	描述	复位值
SCR基地址： **SCS_BA = 0xE000_E000**				
ICSR	SCS_BA+0xD04	R/W	中断控制和状态寄存器	0x0000_0000
VTOR	SCS_BA+0xD08	R/W	向量表偏移寄存器	0x0000_0000
AIRCR	SCS_BA+0xD0C	R/W	应用中断和复位控制寄存器	0xFA05_0000
SCR	SCS_BA+0xD10	R/W	系统控制寄存器	0x0000_0000
SHPR1	SCS_BA+0xD18	R/W	系统中断优先级寄存器1	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	系统中断优先级寄存器2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	系统中断优先级寄存器3	0x0000_0000

中断控制和状态寄存器（ICSR）

寄存器	偏移	R/W	描述	复位值
ICSR	SCS_BA+0xD04	R/W	中断控制和状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	**Reserved**		**PENDSVSET**	**PENDSVCLR**	**PENDSTSET**	**PENDSTCLR**	**Reserved**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISRPREEMPT	**ISRPENDING**	**Reserved**				**VECTPENDING**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECTPENDING				**RETTOBASE**	**Reserved**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**VECTACTIVE**					

位	描述	
[31]	**NMIPENDSET**	**非屏蔽中断设置挂起位** 写操作： 0 = 无效 1 = 将NMI异常状态更改为挂起 读操作： 0 = NMI异常状态无挂起 1 = NMI异常状态挂起 **注：** 由于NMI是最高优先级的异常，通常处理器在检测到对该位的写入为1时就进入NMI异常处理程序。进入处理程序，然后将此位清除为0。这意味着，只有在处理器执行NMI异常处理程序时再次断言NMI信号时，NMI异常处理程序读此位才会返回1。
[30:29]	**Reserved**	保留
[28]	**PENDSVSET**	**PendSV设置挂起位** 写操作： 0 = 无效 1 = 将PendSV异常状态更改为挂起 读操作： 0 = PendSV异常状态无挂起 1 = PendSV异常状态挂起 **注：** 设置PendSV 异常状态的唯一方法是写1到该位。
[27]	**PENDSVCLR**	**PendSV清除挂起位** 写操作： 0 = 无效 1 = 从PendSV异常移除挂起状态 **注：** 该位只写。清除PendSV位，必须同时“写0到PENDSVSET且写1到PENDSVRTC_CAL”。

[26]	**PENDSTSET**	**SysTick异常设置挂起位** 写操作： 0 = 无效 1 =将SysTick异常状态更改为挂起 读操作： 0 = SysTick异常状态无挂起 1 = SysTick异常状态挂起
[25]	**PENDSTCLR**	**SysTick异常清除挂起位** 写操作： 0 = 无效 1 = 移除SysTick异常挂起状态 **注：** 该位只写。清除PENDST位，必须同时“写0到PENDSTSET且写1到PENDSTRTC_CAL”。
[24]	**Reserved**	保留
[23]	**ISRPREEMPT**	**中断抢先位（只读）** 如果设置，则在退出调试停止状态时将执行挂起异常。
[22]	**ISRPENDING**	**中断挂起标志, NMI 和 Faults 除外（只读）** 0 = 中断未挂起 1 = 中断挂起
[21:18]	**Reserved**	保留
[17:12]	**VECTPENDING**	**最高优先级挂起异常号码** 指示挂起使能异常中，最高优先级的异常号 0 = 没有挂起的异常 非0 =挂起使能异常中，最高优先级的异常号 此字段指示的值会被BASEPRI和FAULTMASK寄存器影响，不会被PRIMASK寄存器影响。
[11]	**RETTOBASE**	**抢先的异常标志** 指示是否有抢先的异常 0 = 在执行抢先的中断 1 = 无抢先的中断，或当前中断是唯一活动的中断
[10:6]	**Reserved**	保留
[5:0]	**VECTACTIVE**	**当前活动异常的号码** 0 = 线程模式 非0 = 目前活动的异常号码

向量表偏移寄存器（VTOR）

寄存器	偏移	R/W	描述	复位值
VTOR	SCS_BA+0xD08	R/W	向量表偏移寄存器	0x0000_0000

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

位	描述	
[31:7]	**TBLOFF**	**表偏移位** 选定安全状态的向量表地址
[6:0]	**Reserved**	保留

应用中断和复位控制寄存器（AIRCR）

寄存器	偏移	R/W	描述	复位值
AIRCR	SCS_BA+0xD0C	R/W	应用中断和复位控制寄存器	0xFA05_0000

31	30	29	28	27	26	25	24
VECTORKEY							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VECTORKEY							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ENDIANNESS	**Reserved**				**PRIGROUP**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**SYSRESETREQ**	**VECTCLRACTIVE**	**VECTRESET**

位	描述	
[31:16]	**VECTORKEY**	**寄存器访问密匙** 当要写这个寄存器时，VECTORKEY域必须设置为0x05FA，否则写操作将被忽略。 VECTORKEY域用于防止系统复位或者清除异常状态时误写该寄存器。
[15]	**ENDIANNESS**	**数据连续性** 0 = 小端模式 1 = 大端模式
[14:11]	**Reserved**	保留
[10:8]	**PRIGROUP**	**中断优先级分组** 此字段决定组优先级从子优先级的划分。
[7:3]	**Reserved**	保留
[2]	**SYSRESETREQ**	**系统复位请求** 将该位写入1将导致芯片产生复位信号并指示请求复位。这个位是写的和自清除作为复位序列的一部分。
[1]	**VECTCLRACTIVE**	**异常活动状态清除位** 将此位设置为1将清除固定和可配置异常的所有活动状态信息此位是只写的，只能在内核停止时写入。 **注:**要重新初始化堆栈是调试程序的职责。
[0]	**VECTRESET**	保留

优先级组	二进制	组优先级位	子优先级位	组优先级数	子优先级数
0b000	bxxxxxxx.y	[7:1]	[0]	128	2
0b001	bxxxxxx.yy	[7:2]	[1:0]	64	4
0b010	bxxxxx.yyy	[7:3]	[2:0]	32	8
0b011	bxxxx.yyyy	[7:4]	[3;0]	16	16
0b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
0b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
0b110	bx.yyyyyyy	[7]	[6:0]	2	128
0b111	b.yyyyyyyy	None	[7:0]	1	256

表 6.3-10 优先级分组

系统控制寄存器（SCR）

寄存器	偏移	R/W	描述	复位值
SCR	SCS_BA+0xD10	R/W	系统控制寄存器	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[4]	**SEVONPEND**	**挂起时发送事件** 0 = 只有使能中断或事件可以唤醒处理器，不包括禁用中断在内 1 = 使能事件和所有中断（包括禁用的中断），可以唤醒处理器 当一个事件或中断进入挂起状态时，事件信号从WFE唤醒处理器。如果处理器没在等待事件，事件将会被注册并影响下一个WFE。 执行SEV指令或外部事件同样会唤醒处理器。
[3]	**Reserved**	保留
[2]	**SLEEPDEEP**	**处理器深度睡眠和睡眠模式选择** 控制处理器在低电模式时使用休眠还是深度休眠模式。 0 = 睡眠 1 = 深度睡眠
[1]	**SLEEPONEXIT**	**Sleep-on-exit 使能位** 该位表明从Handler模式返回到Thread模式时，是否退出休眠。 0 =从线程模式返回，不休眠 1 =从ISR返回到线程模式时，进入休眠或深度休眠 置该位为1使能一个中断驱动应用，从而避免返回到一个空的主函数应用。
[0]	**Reserved**	保留

系统中断优先级寄存器1（SHPR1）

寄存器	偏移	R/W	描述	复位值
SHPR1	SCS_BA+0xD18	R/W	系统中断优先级寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_6							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_4							

位	描述	
[31:24]	**Reserved**	保留
[23:16]	**PRI_6**	系统中断优先级6，使用故障UsageFault
[15:8]	**PRI_5**	系统中断优先级5，总线错误BusFault
[7:0]	**PRI_4**	系统中断优先级4，内存错误MemManage

系统中断优先级寄存器2（SHPR2）

寄存器	偏移	R/W	描述	复位值
SHPR2	SCS_BA+0xD1C	R/W	系统中断优先级寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

位	描述	
[31:30]	**PRI_11**	**系统中断优先级11 – SVCall** “0” 代表最高优先级，“3” 代表最低优先级。
[29:0]	**Reserved**	保留

系统中断优先级寄存器3（SHPR3）

寄存器	偏移	R/W	描述	复位值
SHPR3	SCS_BA+0xD20	R/W	系统中断优先级寄存器3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		**Reserved**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

位	描述	
[31:30]	**PRI_15**	**系统中断优先级15 – SysTick** “0” 代表最高优先级，“3” 代表最低优先级。
[29:24]	**Reserved**	保留
[23:22]	**PRI_14**	**系统中断优先级14 – PendSV** “0” 代表最高优先级，“3” 代表最低优先级。

6.4 Flash 存储控制器(FMC)

6.4.1 概述

此芯片内置16/32/64/128/256/512K字节Flash(512K字节的芯片包含2个256K字节BANK0和BANK1)。一个用户配置区，用于系统初始化。引导存储器区(LDROM)，用于在系统编程(ISP)功能，安全保护ROM(SPROM)可以用来隐藏客户程序。M031xG/I和M032xG/I，有4K字节零等待周期高速缓存来提升代码/数据的执行效率。此芯片同样支持在应用编程（IAP）功能，FLASH更新后无需复位芯片就可以切换执行代码。 参考表 6.4-1详细介绍M031/M032系列各芯片的区别。

6.4.2 特性

- 支持安全固件升级的双BANK Flash架构
- 支持OTA功能
- 支持APROM bank擦除
- 支持16/32/64/128/256/512K字节应用程序存储空间(APROM)
- 16/32/64/128K字节型号的芯片，Flash每页512字节
- 256/512K字节型号的芯片，Flash每页2048字节
- 支持2/4/8K字节引导存储器 (LDROM)
- 支持与APROM共用空间，大小可配置的数据Flash
- 支持512/2048字节安全保护存储空间(SPROM)可隐藏客户程序
- 支持12字节用户配置块控制系统初始化
- 片上Flash支持512/2048字节页擦除
- 支持CRC-32校验和计算功能
- 支持在系统编程（ISP）/在应用编程（IAP）来更新片上Flash代码
- 支持高速缓存来提升Flash存储性能和减少电源功耗

章节	子章节	M031xB/C/D/E M032xB/C/D/E	M031xG/I M032xG/I
6.4.4 功能描述	6.4.4.3物理和虚拟地址概念	-	-/●
	6.4.4.4 APROM 重启地址操作模式选项	-	●
	6.4.4.14 *6.4.2.1 高速缓存控制器*	-	●
	6.4.4.15 片内片上Flash存储器编程 64位编程和多字编程	-	●
	6.4.4.17 Flash 全为1校验	-	●
	ISP控制寄存器(FMC_ISPCTL)	-	●

章节	子章节	M031xB/C/D/E M032xB/C/D/E	M031xG/I M032xG/I
	INTEN (FMC_ISPCTL[24])		
寄存器描述	Flash (FMC_FTCTL) CACHEINV (FMC_FTCTL[9])	-	●
	Flash (FMC_FTCTL) BBOFF (FMC_FTCTL[7])	●	-
	Flash (FMC_FTCTL) FOM (FMC_FTCTL[6:4])	000 = 频率小于或等于 48 MHz. 001 =频率小于或等于24 MHz. 其它 = 保留.	000 =频率小于或等于72 MHz. 001 =频率小于或等于12 MHz. 010 =频率小于或等于36 MHz. 011 =频率小于或等于60 MHz. 其它 =频率小于或等于72MHz.
	ISP状态寄存器 FBS (FMC_ISPSTS[30])	-	●
	ISP状态寄存器 (FMC_ISPSTS) INTFLAG (FMC_ISPSTS[8])	-	●
	ISP状态寄存器 (FMC_ISPSTS) PGFF (FMC_ISPSTS[5])	-	●
	ISP 数据 0 寄存器 (FMC_MPDAT0)	-	●
	ISP 数据1寄存器 (FMC_MPDAT1)	-	●
	ISP数据2寄存器(FMC_MPDAT2)	-	●
	ISP数据3寄存器(FMC_MPDAT3)	-	●
	ISP多字编程地址寄存器(FMC_MPADDR)	-	●
	ISP 多字编程状态寄存器 (FMC_MPSTS)	-	●

表 6.4-1 不同芯片 FMC 特性比较表

6.4.3 框图

存储控制器(FMC)包括AHB从接口，flash控制寄存器，高速缓存控制器（仅M031xG/I和M032xG/I），flash初始化控制器，Flash操作控制和片上flash。Flash存储控制器框图如图 6.4-1所示

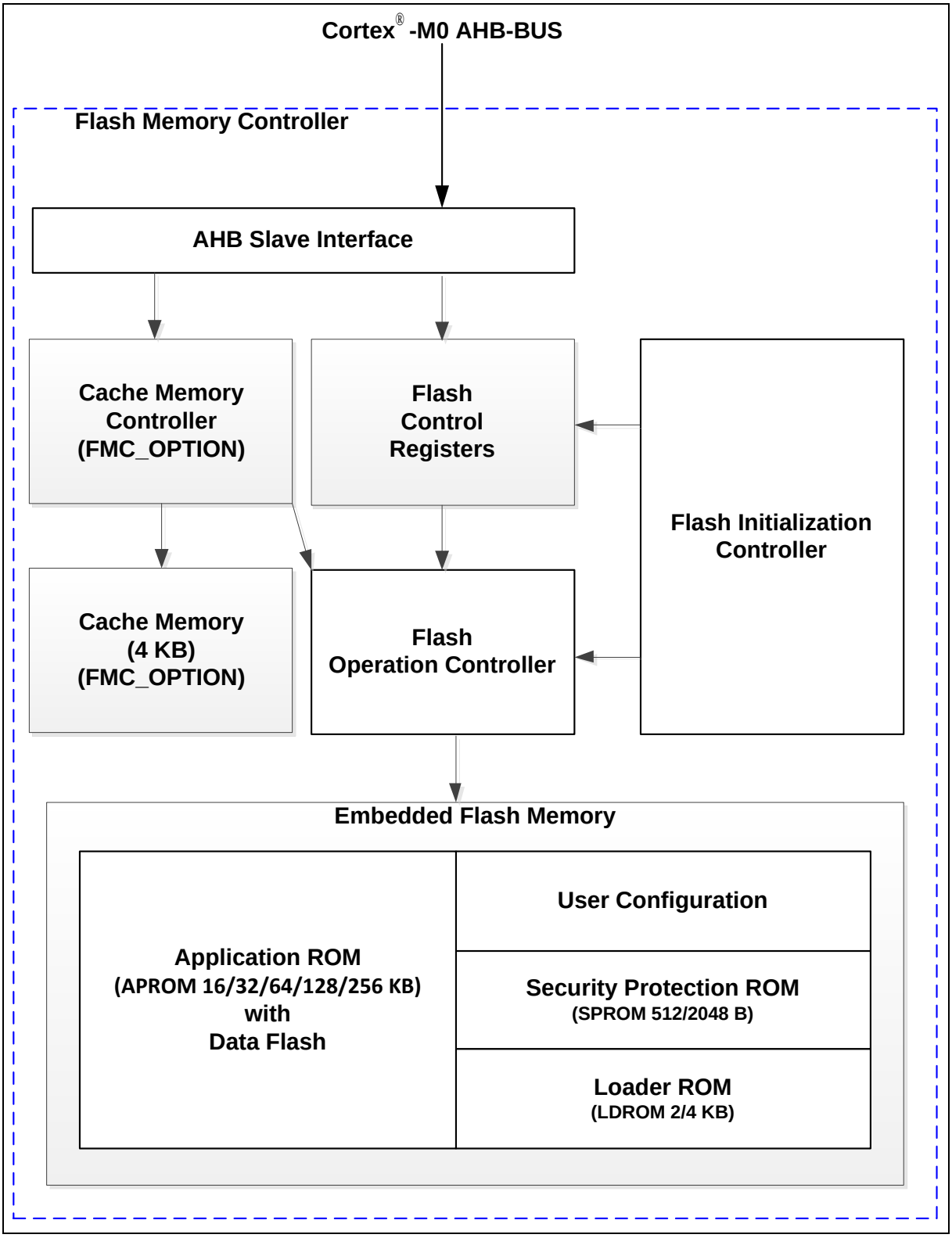


图 6.4-1 16/32/64/128/256KB 型号 Flash 存储器控制框图

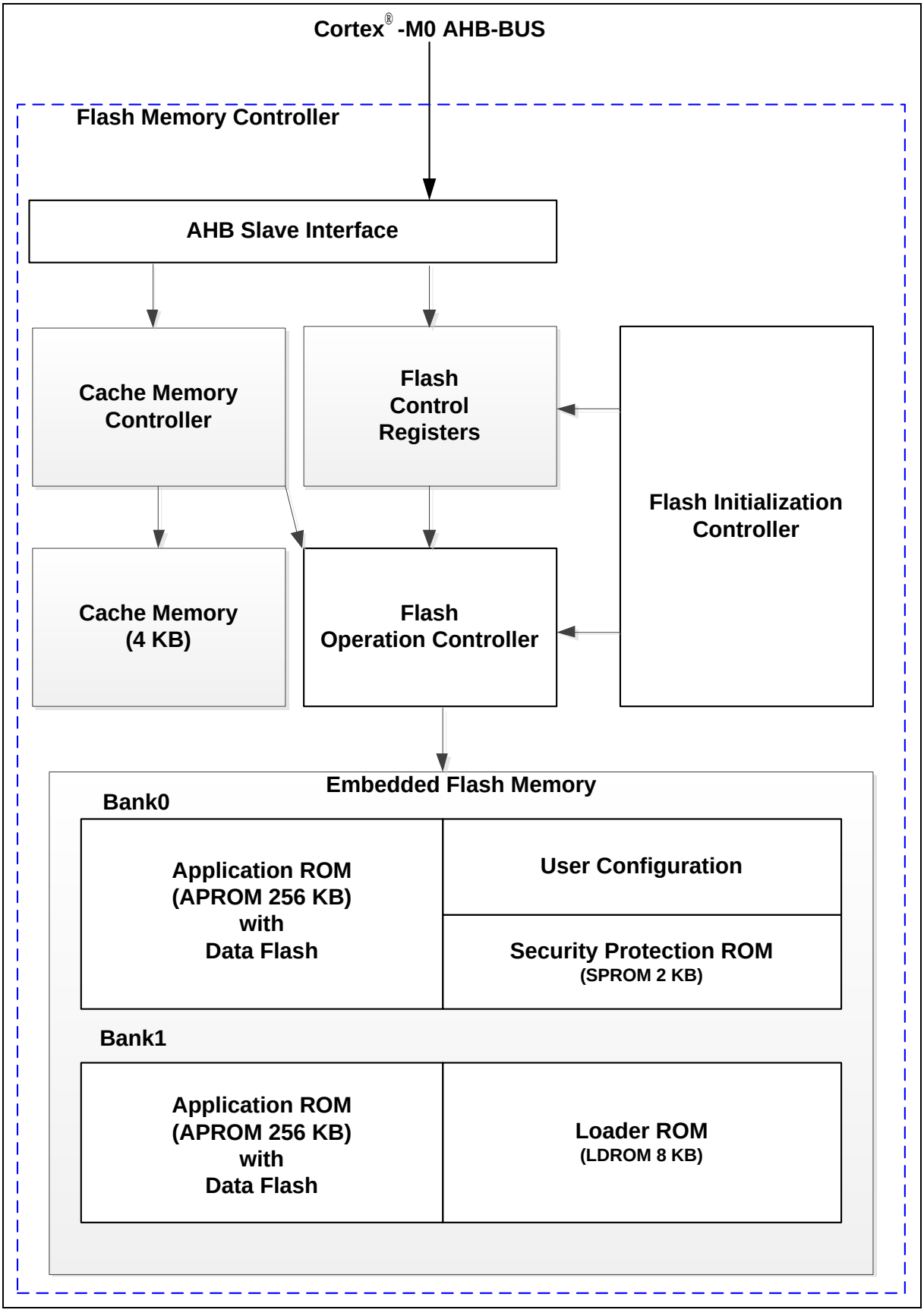

Cortex® -M0 AHB-BUS
Flash Memory Controller
AHB Slave Interface
Cache Memory Controller
Flash Control Registers
Flash Initialization Controller
Cache Memory (4 KB)
Flash Operation Controller
Embedded Flash Memory
Bank0
Application ROM (APROM 256 KB) with Data Flash
User Configuration
Security Protection ROM (SPROM 2 KB)
Bank1
Application ROM (APROM 256 KB) with Data Flash
Loader ROM (LDROM 8 KB)

图 6.4-2 512KB 型号 Flash 存储器控制框图

AHB 从接口

在flash存储控制器中有1个AHB从接口，用于Cortex®-M0 指令与数据的获取以及ISP控制寄存器。

Flash 控制寄存器

所有的ISP控制和状态寄存器都在flash控制寄存器中。详细的寄存器描述会在寄存器介绍章节介绍。

高速缓存控制器

在Cortex®-M0 CPU和片内Flash之间有一个4K字节零等待周期高速缓存。这个高速缓存控制器可以提升指令/数据执行效率和减少片内Flash功耗。

Flash 初始化控制器

当芯片上电或复位，flash初始化控制器将开始自动访问flash，并且检测flash的稳定性，重载用户配置内容到flash控制寄存器用于系统初始化。

Flash 操作控制器

对flash操作，例如计算校验和，flash擦除，flash编程和读flash，有特定的控制时序。Flash操作控制器在收到flash 控制寄存器，高速缓存控制器和flash初始化控制器的请求后， 会产生这些时序。

内置 Flash 存储器

片上flash存储器是用于存储用户应用程序和参数的。它包括了用户配置区，2/4/8K字节的LDROM, 512/2048字节SPROM和16/32/64/128/256/512K字节APROM（数据Flash共享空间）。页擦除的flash大小是512/2048字节，可编程位大小是32位。

6.4.4 功能描述

FMC 功能包括存储器组织，启动选择，IAP，ISP，片上flash编程，和checksum计算。在存储器组织中也介绍了flash存储器映射和系统存储器映射。

6.4.4.1 存储器组织

FMC存储包含了片上flash。片上存储器是可编程的，包括APROM，LDROM，SPROM，数据Flash和用户配置区。地址映射包括flash存储映射和4个地址映射：支持IAP功能的LDROM，不支持IAP功能的LDROM，支持IAP功能的APROM，不支持IAP功能的APROM。

Bank	Flash 存储区	地址范围
0	256K字节APROM	0x00_0000 ~ 0x03_ffff
	用户配置区	0x30_0000 ~ 0x30_000b
	SPROM	0x20_0000 ~ 0x20_07ff
1	256K字节APROM	0x04_0000 ~ 0x07_ffff
	LDROM	0x10_0000 ~ 0x10_1fff

表 6.4-2 双 Bank 区地址范围

6.4.4.2 LDROM，APROM和数据Flash

LDROM是用于通过引导程序，来执行在系统编程(ISP)的功能。LDROM是2/4/8KB的片上flash存储器，地址范围是从0x0010_0000到0x0010_07FF/0x0010_0FFF/0x0010_1FFF。APROM是用户应用程序的

主要存储器。APROM大小是16/32/64/128/256/512K字节。数据Flash是用于存储应用参数的（不是指令）。数据FLASH共享APROM存储空间，大小可配置，数据Flash的基地址由寄存器DFBA(CONFIG1[19:0])设定。所有片上flash存储器，页擦除的大小是512/2048字节。

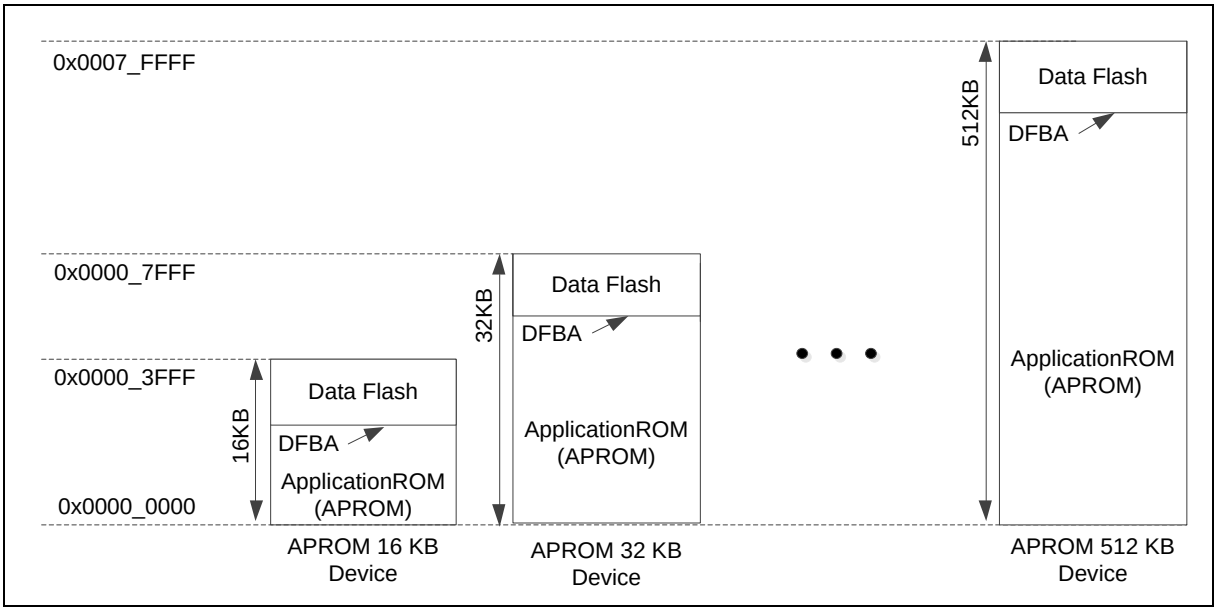


图 6.4-3 数据 Flash 共享 APROM 空间

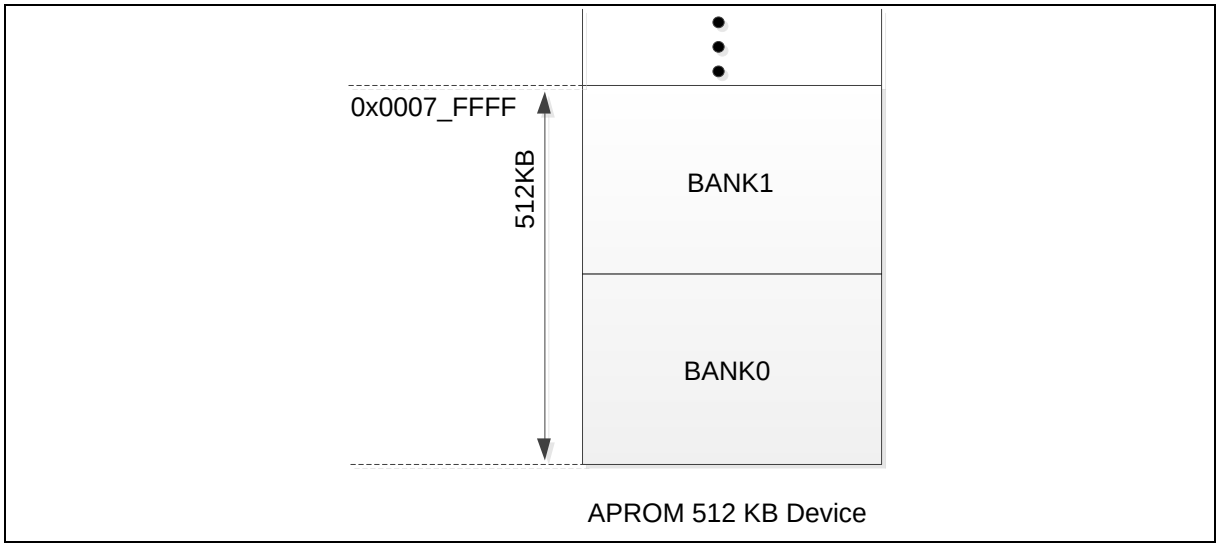


图 6.4-4 双 BANK 区 APROM 示例(512K 字节)

6.4.4.3 物理和虚拟地址概念

在OTA应用，存储空间概念，FMC命令和CPU的角度不同。

- FMC命令的角度，采用物理地址的概念。除了“再映射”命令，所有FMC命令都认为APROM存储器是物理地址。这意味着无论内存再映射状态如何， APROM存储器都是扁平存储模式（flat memory model）。
- 数据Flash视点，采用虚拟地址的概念。

- CPU来说，采用虚拟地址的概念。APROM存储器被分为2个段（BANKS）。在运行阶段，虚拟地址取决于地址操作模式(OP0，OP1)是什么，如图 6.4-5。

6.4.4.4 APROM 重启地址操作模式选项

通过不同地址操作模式来激活APROM重启，使用地址操作模式选择命令0x2c，同时FMC_ISPDAT “0x5AA5_5AA5”和FMC_ISPADDR为”0x0”(地址OP0)或”0x1”(地址OP1)来选择地址操作模式。

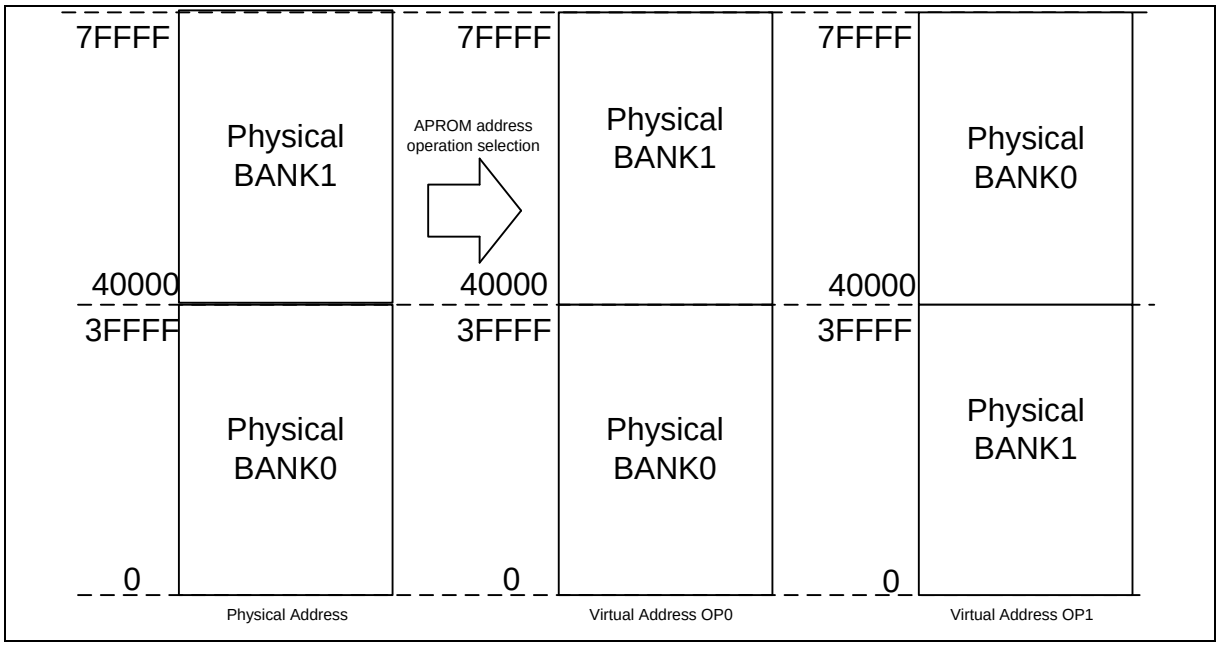


图 6.4-5 地址操作模式

6.4.4.5 用户配置区

用户配置区是内部可编程的配置区域，用于启动选项，比如flash安全锁，启动选择，欠电压电平设置和数据flash基地址。用户配置区的作用类似保险丝用于上电时的缺省设置。在上电的时候，用户配置区设定会加载到相应的控制寄存器。用户可以通过不同的应用需求来设定寄存器。用户配置区可以通过ISP方式更新，位于地址0x0030_0000，有3个32位寄存器(CONFIG0，CONFIG1和CONFIG2)。用户配置所有的更改将在系统重启后生效。

CONFIG0 (地址 = 0x0030_0000)

31	30	29	28	27	26	25	24
CWDTEN[2]	CWDTPDEN	Reserved		CFGXT1	Reserved		
23	22	21	20	19	18	17	16
Reserved		CBOV	CBORST	CBODEN	Reserved		
15	14	13	12	11	10	9	8
Reserved			ICELOCK	Reserved	CIOINI	RSTEXT	RSTWSEL
7	6	5	4	3	2	1	0
CBS		Reserved	CWDTEN[1:0]		Reserved	LOCK	DFEN

位	描述	
[31]	**CWDTEN[2]**	**看门狗定时器硬件使能位** 当看门狗定时器硬件使能功能有效，上电后，看门狗使能位WDTEN (WDT_CTL[7])和看门狗复位使能位RSTEN(WDT_CTL[1])会自动为1。看门狗的时钟源为LIRC，正常模式下LIRC不能被禁用，但是在掉电模式下，LIRC可以通过设定CWDTPDEN=1和LIRCEN=0(CLK_PWRCTL[3]) 禁用。 **CWDTEN[2:0]** 在寄存器 CONFIG0的相应位[31][4][3]。 011 = WDT 看门狗硬件使能功能有效。除芯片进入掉电模式以外，WDT时钟一直开启。当芯片进入掉电模式，如果寄存器CWDTPDEN=0，WDT时钟是一直开启的，如果寄存器CWDTPDEN=1，WDT时钟由寄存器LIRCEN(CLK_PWRCTL[3])控制。请参考寄存器CWDTPDEN的位描述。 111 = WDT 硬件使能功能无效，只有在这种情况下才可以切换WDT时钟源 其它 = WDT 硬件使能功能有效。WDT时钟一直开启。
[30]	**CWDTPDEN**	**看门狗时钟掉电使能** 此位应与CWDTEN共同使用，WDT使能后用户可以通过此位控制系统是否在掉电模式下由WDT唤醒系统；需要唤醒系统时，我们设置CWDTPDEN=0保证WDT在掉电模式下继续工作；不需要唤醒系统时，我们设置CWDTPDEN=1 及 LIRCEN=0让WDT在掉电模式下停止工作。 0 = 当芯片进入掉电模式时，看门狗时钟保持使能 1 = 当芯片进入掉电模式时，看门狗时钟由LIRCEN (CLK_PWRCTL[3])控制. **注:** 该位只有在CWDTEN[2:0]=011时才起作用
[29:28]	**Reserved**	保留.
[27]	**CFGXT1**	**HXT功能选择** 0 = HXT作为外部时钟模式，PF.3管脚配置为外部时钟输入. 1 =HXT配置为晶振模式， PF.2, PF.3配置为外部高速晶振(HXT)管脚. 注：当CFGXT1=0, PF.3 应该设置为GPIO模式，XT1_IN的直流特性跟GPIO是相同的
[26:22]	**Reserved**	保留
[21]	**CBOV**	**欠压电压选择** 00 = 2.0V 01 = 2.5V

[20]	**CBORST**	**欠压复位使能** 0 =上电后，使能欠压复位 1 =上电后，禁用欠压复位.
[19]	**CBODEN**	**欠压检测使能位** 0= 上电后，使能欠压检测 1= 上电后，禁用欠压检测.
[18:13]	**Reserved**	保留.
[12]	**ICELOCK**	**ICE锁定位** 此位用来禁止ICE功能，用户可以用来增加安全级别 0 = ICE 功能禁止 1 = ICE 功能使能
[11]	**Reserved**	保留
[10]	**CIOINI**	**I/O 初始状态选择** 0 = 上电或管脚复位后所有GPIO默认为准双向模式 1 = 上电或管脚复位后所有GPIO默认为三态模式
[9]	**RSTEXT**	**芯片复位时间扩展** 0 = 当芯片从 上电/低压/掉电/管脚复位释放后，复位时间扩展到1s 1 =当芯片从 上电/低压/掉电/管脚复位释放后，复位时间扩展到3.2ms
[8]	**RSTWSEL**	**RST 脚信号宽度选择** 0 = RST管脚去抖宽度2us 1 = RST管脚去抖宽度32us
[7:6]	**CBS**	**芯片启动选择** 当CBS[0] = 0 IAP功能有效，LDROM的基地址映射到0x100000, APROM的基地址映射到0x0. 用户既可以访问APROM也可以访问LDROM，无需启动切换。换句话说，如果是IAP模式，存放在LDROM与APROM的代码可以互相调用。 00=由LDROM启动支持IAP功能 01=由LDROM启动不支持IAP功能 10=由APROM启动支持IAP功能 11=由APROM启动不支持IAP功能 注： 当CBS[0] = 1，只有BS (FMC_ISPCTL[1]) 可以控制启动选择 当CBS[0] = 0，VECMAP (FMC_ISPSTS[23:9])可以将地址0x0~0x1ff重新映射到APROM或者LDROM
[5]	**Reserved**	保留.
[4:3]	**CWDTEN[1:0]**	**看门狗定时器硬件使能位** 详细请参考CWDTEN[2] (CONFIG0[31])
[2]	**Reserved**	保留.
[1]	**LOCK**	**加密锁控制** 0 = 加密FLASH数据 1 = Flash数据除了ALOCK (CONFIG2[7:0]) 是 0x5A外其他都没有加密

[0]	**DFEN**	**数据FLASH使能位** 当DFEN=0时 APROM与数据flash共享，数据flash的基地址由DFBA(CONFIG1[19:0])设定。 0 = 使能数据FLASH 1 = 禁用数据FLASH

CONFIG1 (地址 = 0x0030_0004)

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

位	描述	
[31:20]	**Reserved**	保留.
[19:0]	**DFBA**	**数据Flash基地址** 只有当DFEN (CONFIG0[0])=0时，该寄存器才工作。如果DFEN (CONFIG0[0])=0， 数据Flash基地址由用户定义。因为片上FLASH擦除单位为512或2048字节，所以强制地址bit8-0位或bit10-0为0。

CONFIG2 (地址 = 0x0030_0008)

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

位	描述	
[31:8]	**Reserved**	保留.
[7:0]	**ALOCK**	**高级加密锁控制** 0x5A =如果LOCK (CONFIG0[1])为1，Flash存储器内容未加密 其他 = Flash存储器内容加密 **注:** 执行页擦除或者整片擦除命令后ALOCK将被写为0x5A

6.4.4.6 安全保护存储器(SPROM)

安全保护存储器（SPROM）用于应用程序的保密。SPROM包含512/2048字节，起始地址从0x20_0000~0x20_01FF/0x20_07FF，不支持整片擦除命令。图 6.4-6示SPROM的最后一个字节(地址：0x0020_01FF) 用于标识SPROM中的程序是非安全，DEBUG还是安全模式。SPROM最后一个字节 (地址：0x0020_07FF)用于256/512K字节Flash芯片。

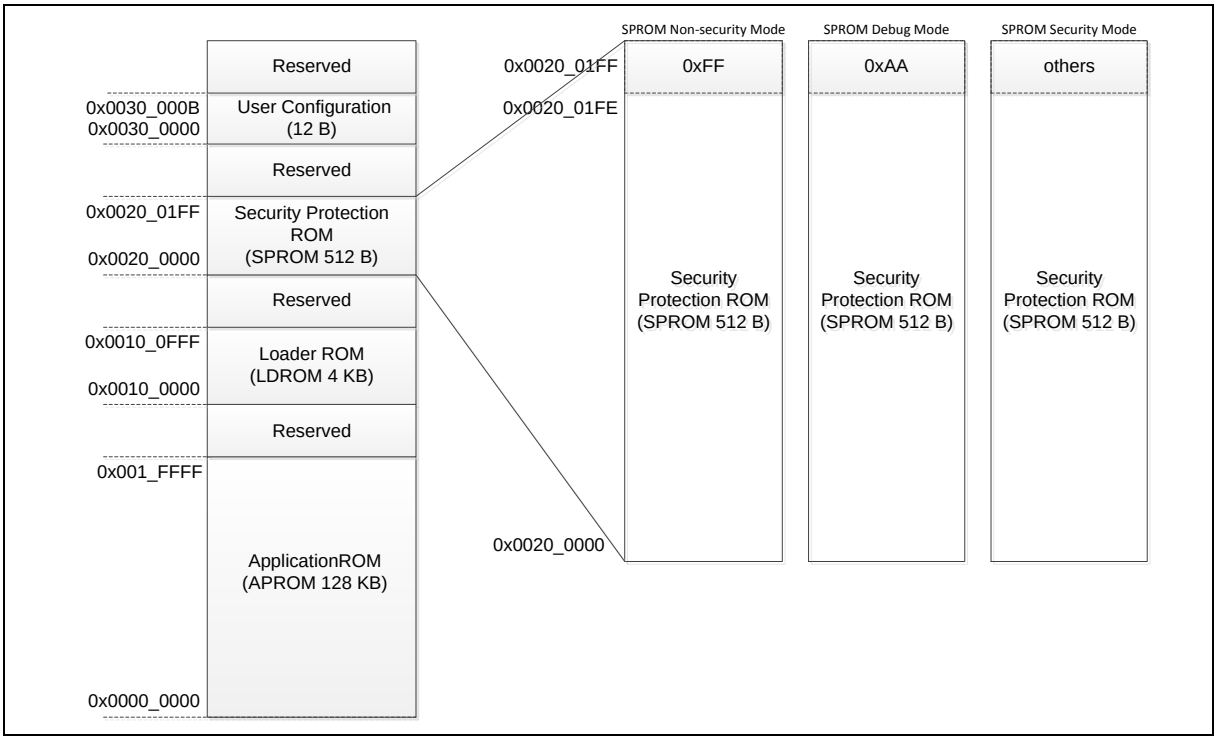


图 6.4-6 16/32/64/128K 字节 Flash 芯片 SPROM 安全模式

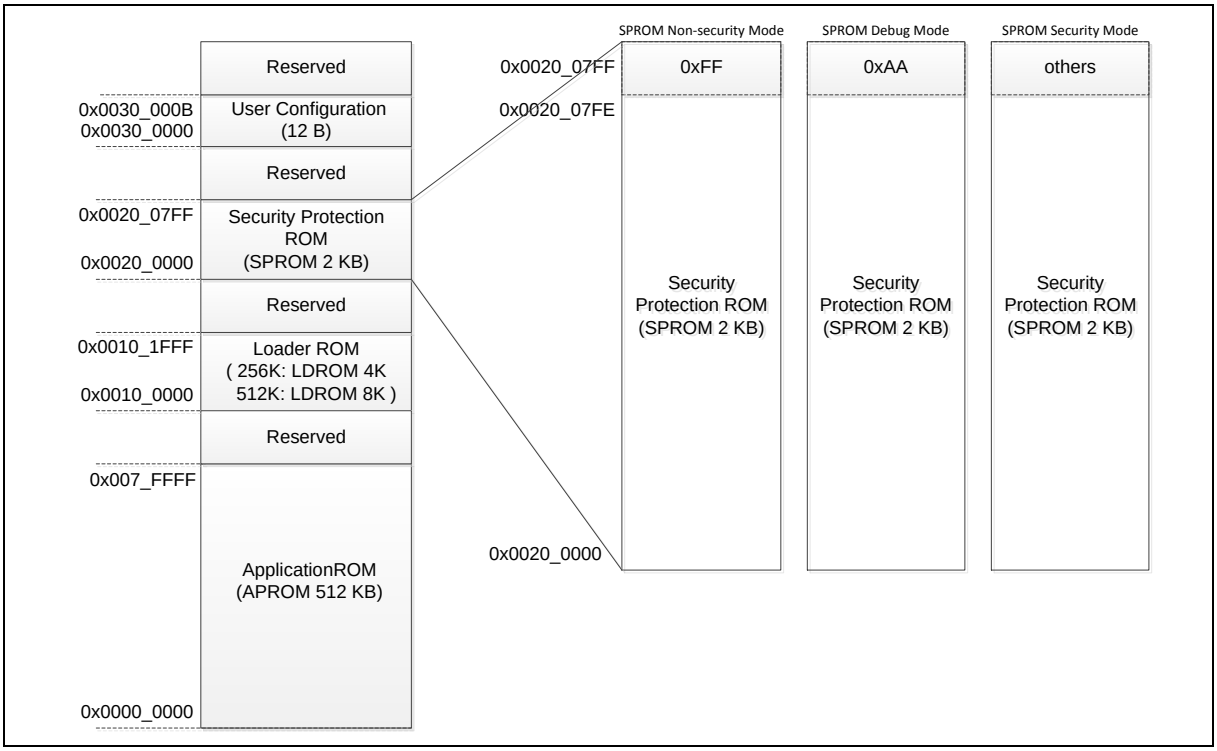


图 6.4-7 256/512K 字节 Flash 芯片 SPROM 安全模式

1. SPROM非安全模式(最后一个字节是0xFF)。SPROM的操作跟APROM，LDROM是一样的。所有的区域都可以被CPU或者ISP读取，也能够通过ISP命令擦除和编程。

2. SPROM在debug安全模式下(最后一个字节是0xAA)。为了方便DEBUG，当Cortex®-M0 ICE接口被连接上时，Flash存储器控制器可以执行SPROM中的程序。SPROM其它的操作与SPROM安全模式下一样。

3. SPROM 安全模式 (最后一个字节不为0xFF 或0xAA)。在安全模式下为了隐藏SPROM中的代码,只有当CPU运行在SPROM区域，CPU才能从SPROM执行取指令和获取数据。否则，CPU的操作收到的数据将全是0。在安全模式下，即使Cortex®-M0 ICE接口被连接上，为了保护SPROM,CPU获取的指令也将全部为0。在这个模式下，SPROM不支持ISP编程和读flash的命令。仅支持用0x0055AA03数据来进行页擦除命令。

4. SCODE(FMC_ISPSTS[31])是SPROM在安全模式的标志位，用于表明SPROM是在安全模式，DEBUG安全模式，还是非安全模式。如果最后一个字节不是0xff，那么在初始化时这个位就置1，当SPROM页擦除完成后这个位被清0。在通常的运行中，为了容易测试SPROM的安全模式，如果SPROM最后一个字节是0xff，用户也能将这个位置1。

6.4.4.7 Flash 存储器映射

M031/M032系列，Flash存储器映射有别于系统存储器映射。当CPU访问FMC存储器获取代码或数据时，使用系统存储器映射。当用户用ISP命令去读、编程、或者擦除FMC存储器时，使用Flash存储器映射。 图 6.4-8 为Flash存储器映射。

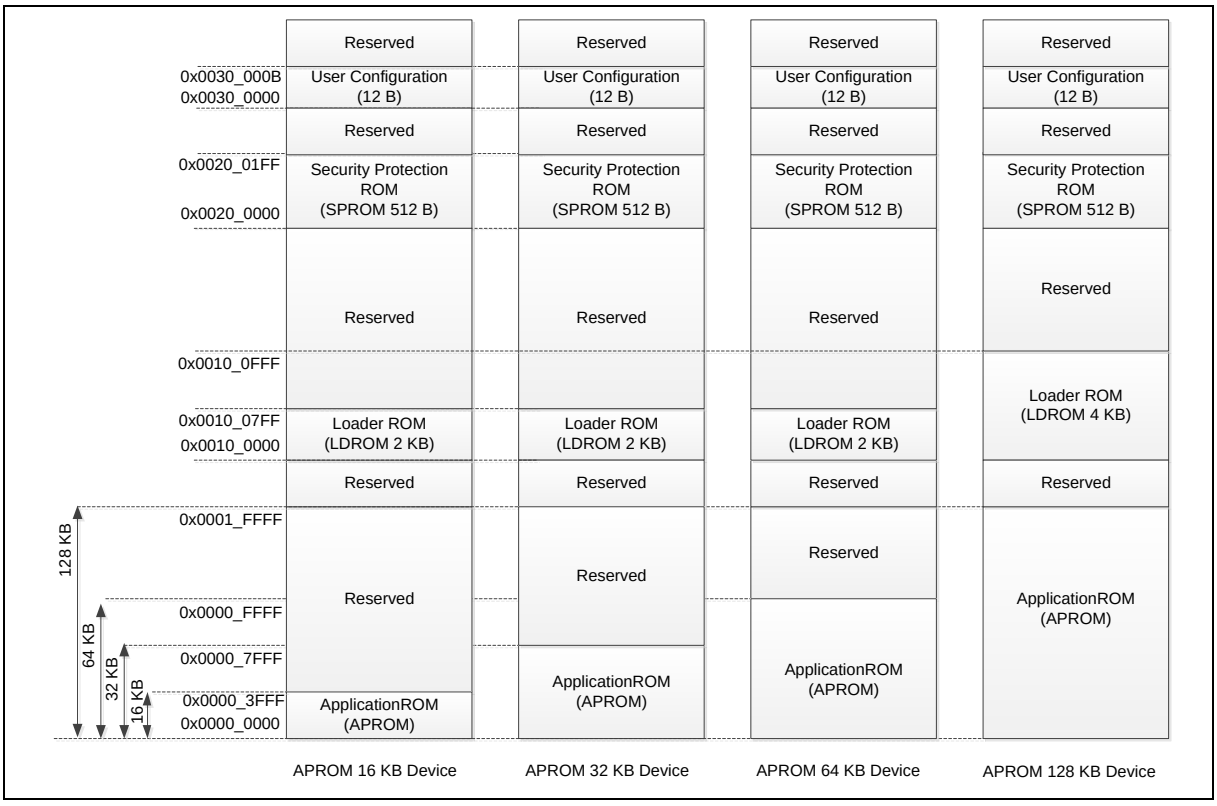


图 6.4-8 16/32/64/128K 字节芯片存储器映射

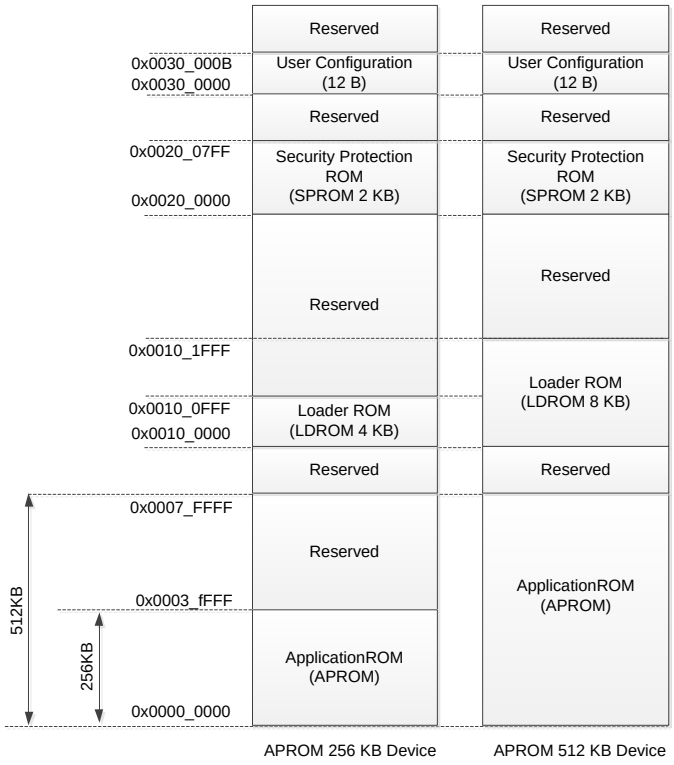


图 6.4-9 256/512K 字节芯片存储器映射

6.4.4.8 支持IAP的系统存储器映射

在 CPU 访问 FMC 存储器获取代码或数据的时候，用到系统内存映射。SPROM(0x0020_0000~0x0020_01FF/0x0020_07FF) 和 LDROM(0x0010_0000~0x0010_07FF/0x0010_0FFF/0x0010_1FFF)地址映射在Flash存储器映射中是相同的。数据Flash共享APROM空间，数据Flash基地址在CONFIG1中定义，CONFIG1的值是在Flash初始化的时候加载到DFBA (Data Flash Base Address Register)中的， DFBA到APROM的最大长度是数据Flash区域，用于Cortex®-M0数据存取，0x0000_0200到(DFBA-1)位置是APROM区域，用于Cortex®-M0 指令存取。

0x0000_0000到0x0000_01FF的地址空间叫做系统存储器向量，CPU启动时APROM，LDROM都可以映射到系统存储向量来启动，芯片带IAP模式启动有两种系统存储器映射：(1)LDROM带IAP；(2)APROM带IAP.

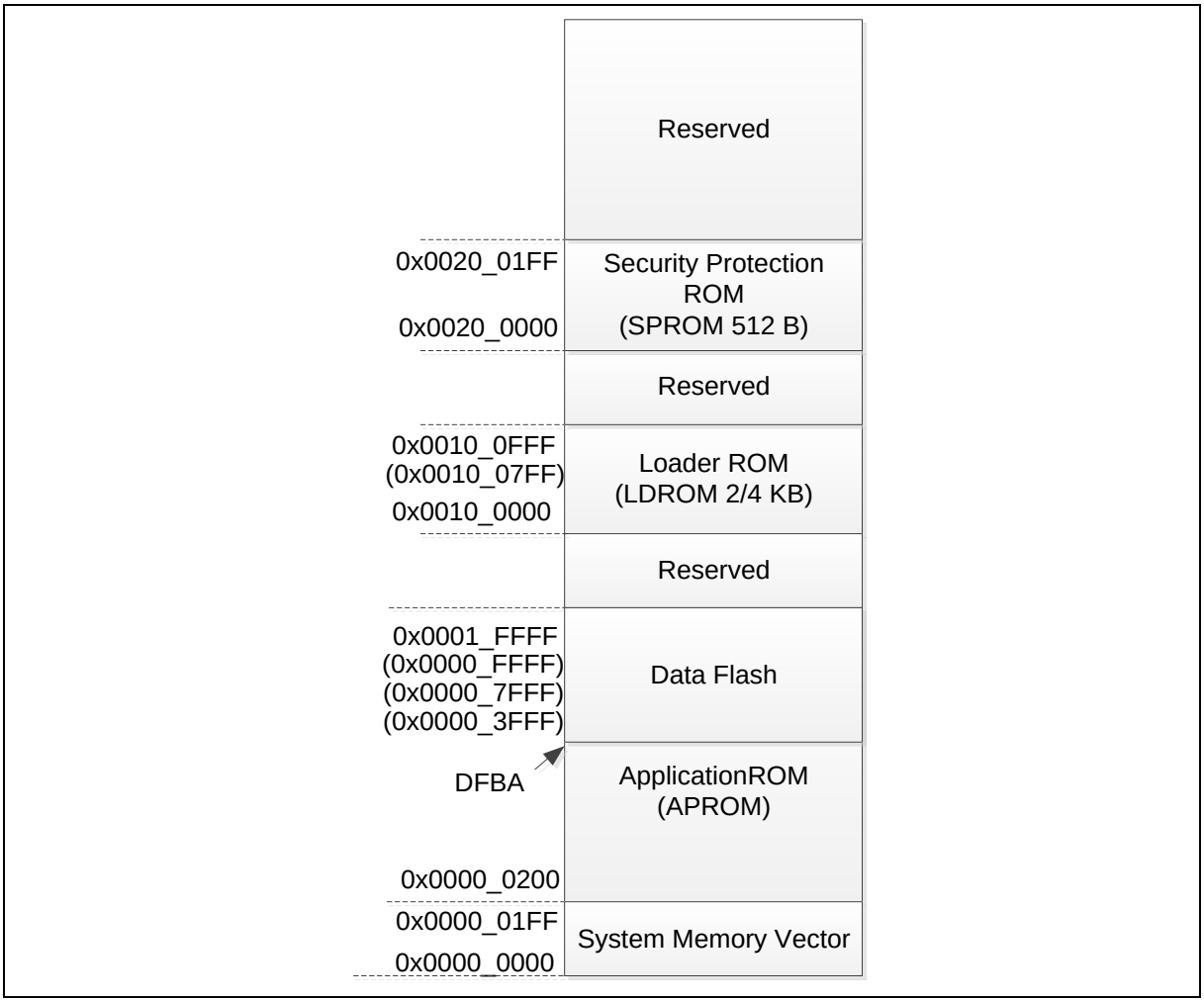


图 6.4-10 16/32/64/128K 字节 Flash 芯片系统存储器映射带 IAP 模式

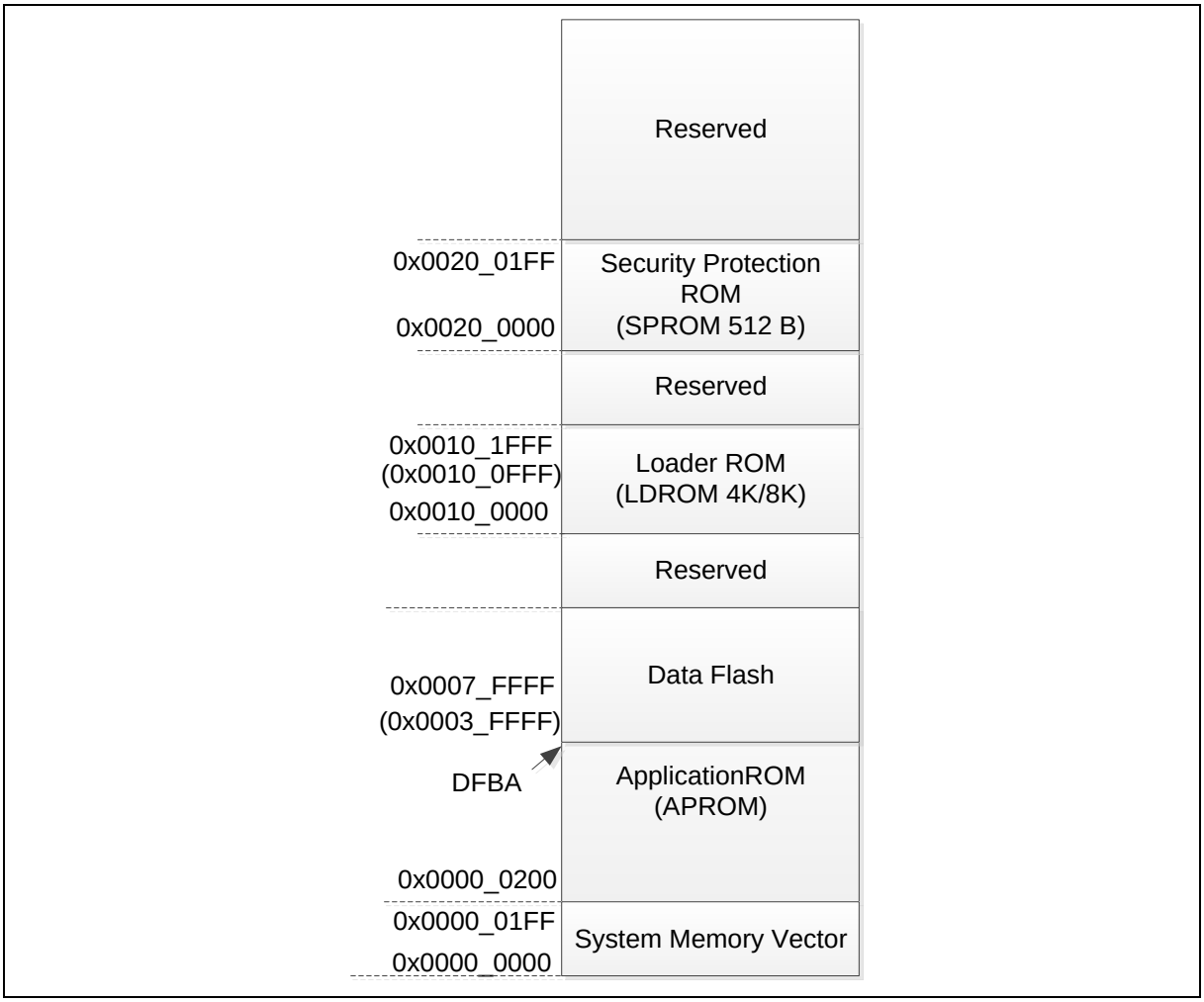


图 6.4-11 256/512K 字节 Flash 芯片系统存储器映射带 IAP 模式

在支持IAP的LDROM模式，{VECMAP[11:0]，9'h000}的默认值是0x100000，LDROM的第一页(0x0010_0000~0x0010_01FF)映射到系统存储向量区，用于Cortex®-M0指令或数据存取。

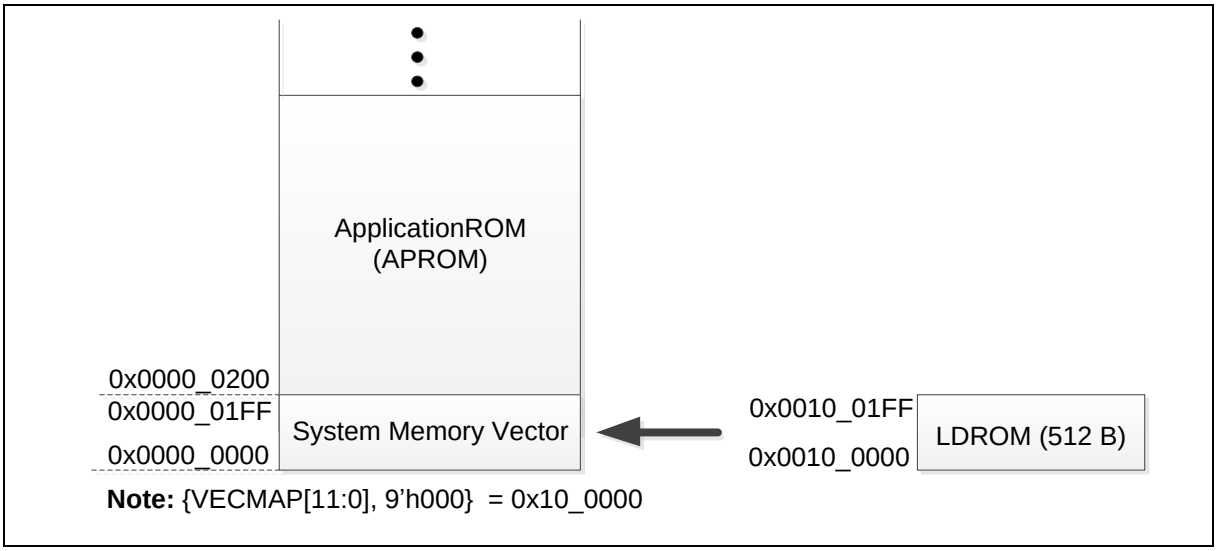


图 6.4-12 LDROM 带 IAP 模式

在支持IAP的APROM模式，{VECMAP[11:0], 9'h000}默认值为0x000000，APROM的第一页(0x0000_0000~0x0000_01FF)映射到系统存储向量区，用于Cortex®-M0指令或数据存取。

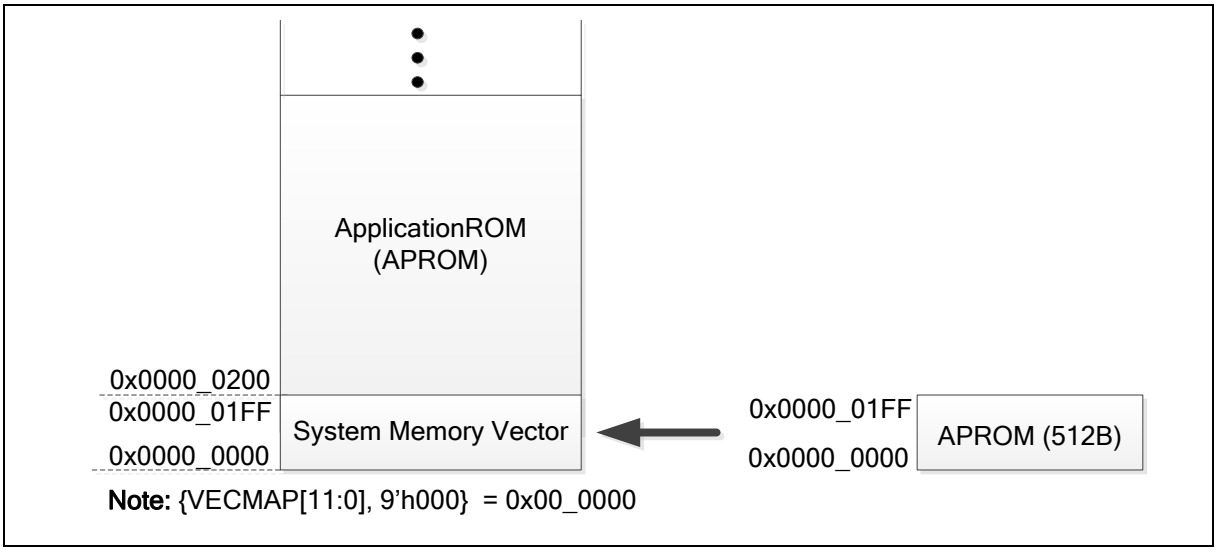


图 6.4-13 APROM 带 IAP 模式

在支持IAP的系统存储映射模式中，当CPU运行时，APROM，LDROM都可以重新映射到系统存储向量区。用户可以写重映射的目标地址到寄存器FMC_ISPADDR，然后通过"向量重映射"指令(0x2E)触发ISP流程。在寄存器VECMAP (FMC_ISPSTS[23:9])，显示了最终的系统存储向量映射地址。

6.4.4.9 不支持IAP功能的系统内存映射

不支持IAP功能的系统存储映射，CPU仍然可以访问SPROM(0x0020_0000~0x0020_01FF/0x0020_07FF)，但不支持系统存储向量映射。在芯片启动时，有两种不支持IAP的系统存储映射：(1)不支持IAP功能的LDROM (2)不支持IAP功能的APROM。在不支持IAP功能的LDROM模式中，LDROM 基地址映射到0x0000_0000，CPU程序不能访问APROM。在不支持IAP功能的APROM中，APROM的基地址映射到0x0000_0000，CPU程序不能访问LDROM。数据Flash与APROM共享，数据Flash的基地址由CONFIG1设定。在Flash初始化期间，CONFIG1的内容被加载到DFBA(数据Flash基地址)。DFBA~0x0000_7FFF是数据Flash区用于Cortex®-M0数据的存取，0x0000_0000~(DFBA-1)是APROM区用于Cortex®-M0的指令存取。

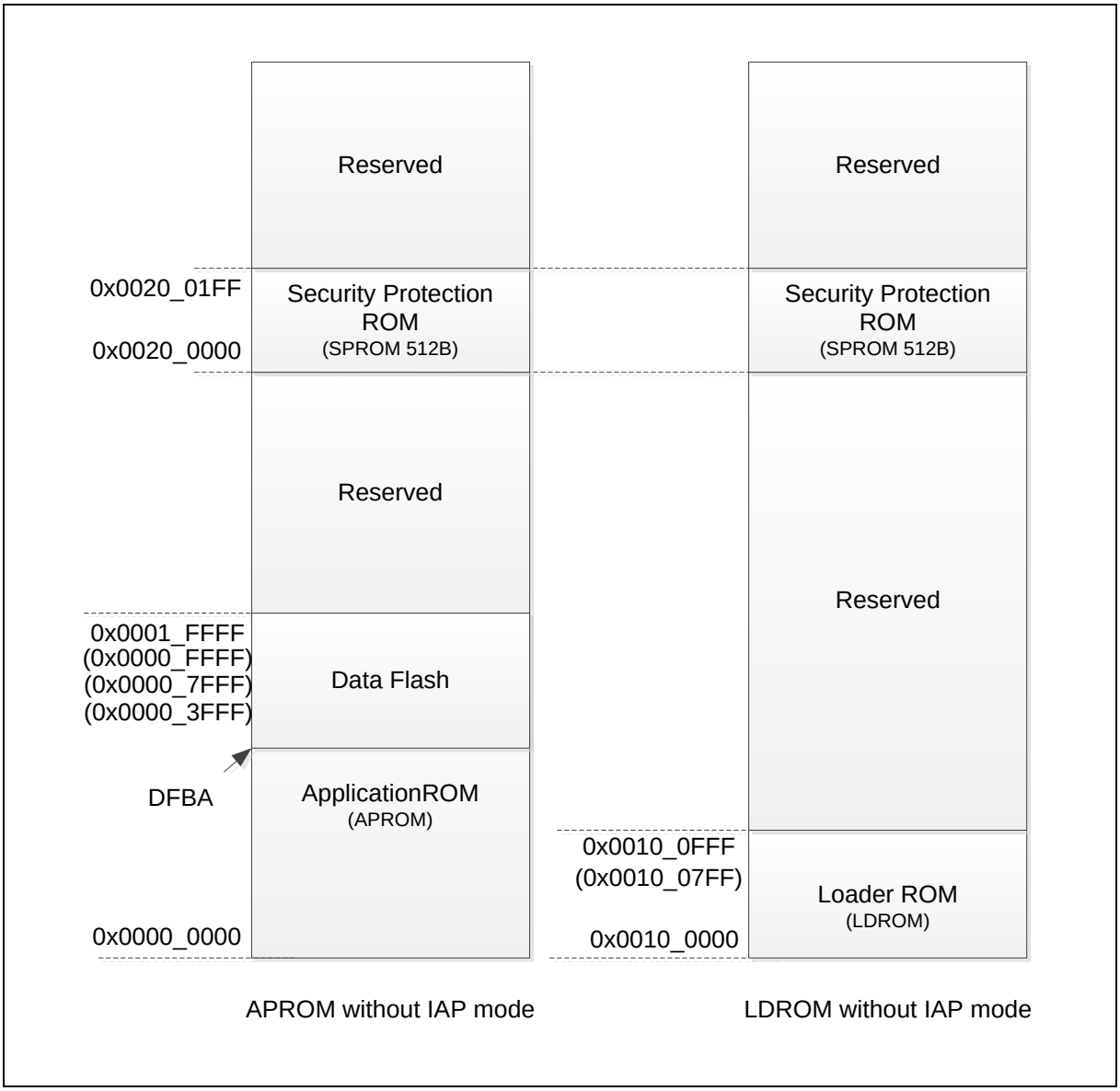


图 6.4-14 16/32/64/128K 字节 Flash 的芯片不带 IAP 模式系统内存映射

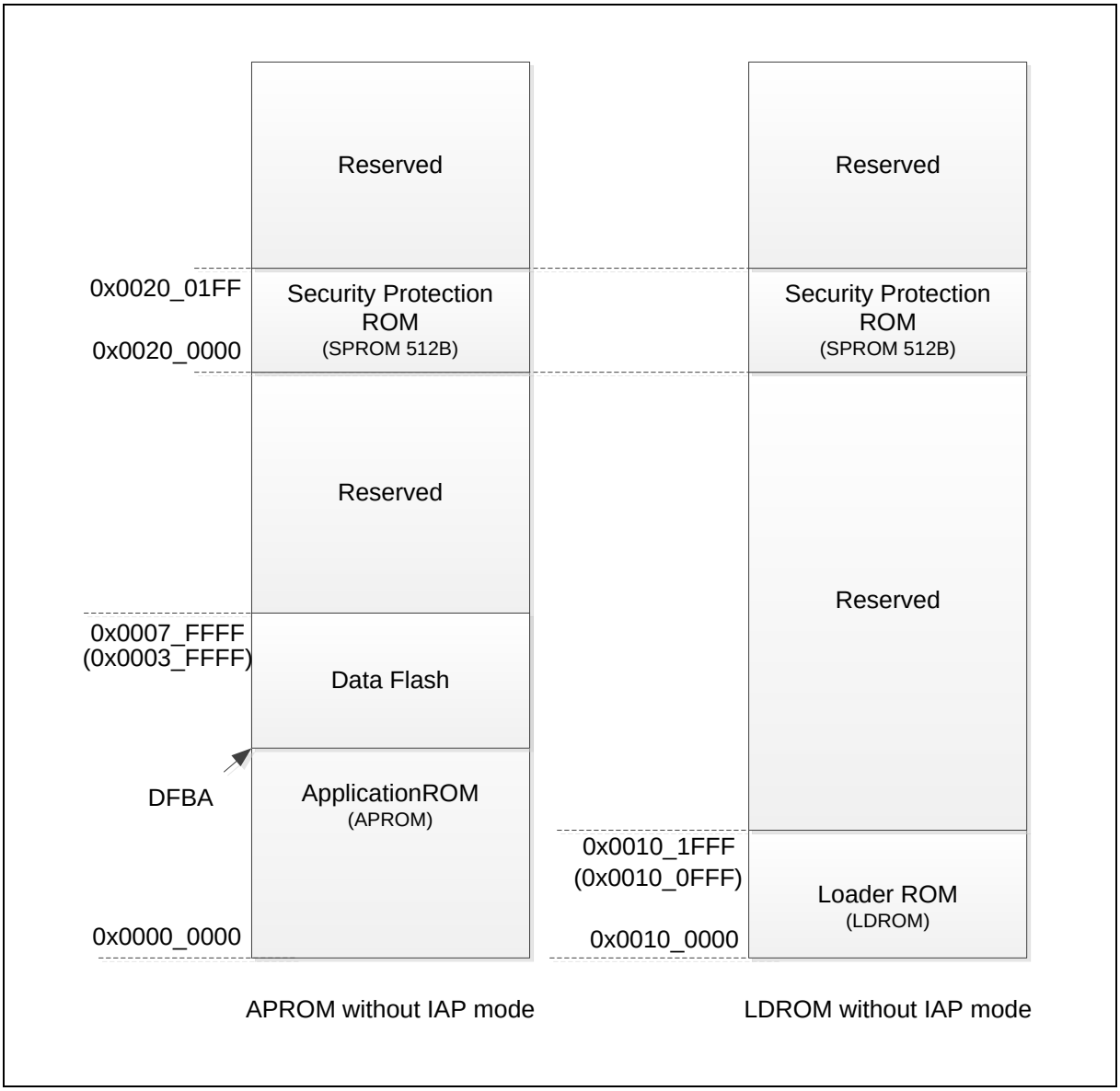


图 6.4-15 256/512K 字节 Flash 的芯片不带 IAP 模式系统内存映射

6.4.4.10 启动选择

M031/M032系列提供了4种启动源，包括支持IAP功能的LDROM，不支持IAP功能的LDROM，支持IAP功能的APROM，不支持IAP功能的APROM。启动来源和系统存储映射由CBS(CONFIG0[7:6])设置。

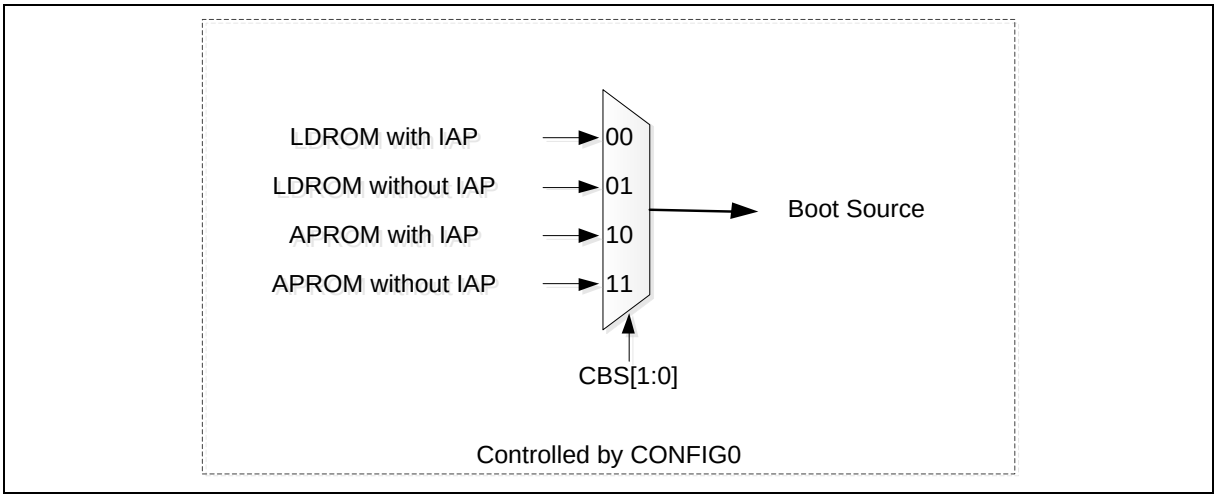


图 6.4-16 启动源选择

CBS[1:0]	启动选择/系统存储器映射	支持向量映射
00	LDROM 带 IAP	支持
01	LDROM 不带IAP	不支持
10	APROM 带 IAP	支持
11	APROM 不带IAP	不支持

表 6.4-3 向量映射支持

6.4.4.11 在应用编程(IAP)

M031/M032系列提供了在应用编程(IAP)功能，用户可以切换APROM，LDROM和SPROM之间的代码。用户可以通过设定芯片的启动位寄存器CBS (CONFIG0[7:6])等于10或00来使能IAP功能。

当支持IAP功能的芯片启动模式使能，任何可以执行的代码(512字节对齐)允许随时映射到系统存储向量区(0x0000_0000~0x0000_01FF)。用户可以改变重映射地址到FMC_ISPADDR，然后用“向量重映射”命令触发ISP流程。

6.4.4.12 在系统编程(ISP)

M031/M032系列支持在系统编程(ISP)模式，可以通过软件控制重新烧写片上Flash。通过固件和多种接口ISP可以在目标板上直接编程不需要将芯片从系统中取下来。接口包括UART，I2C，SPI。

M031系列ISP对片上Flash操作提供如下功能：

- 支持页擦除功能
- 支持Flash APROM bank擦除功能
- 支持数据Flash编程功能
- 支持读数据Flash功能
- 支持读公司ID功能
- 支持读设备ID功能
- 支持读UID功能

- 支持存储器checksum计算功能
- 支持系统存储器向量重映射功能

ISP 命令

ISP 命令	FMC_ISPCMD	FMC_ISPADDR	FMC_ISPDAT FMC_MPDAT0~ MPDAT3
FLASH页擦除	0x22	FLASH存储器的有效地址 它必须512/2048字节地址对齐	N/A
SPROM 页擦除	0x22	0x0020_0000	0x0055_AA03
FLASH APROM BANK擦除	0x23	FLASH APROM存储器的有效地址	N/A
FLASH 32-位编程	0x21	Flash存储器的有效地址	FMC_ISPDAT : 编程数据
FLASH 64-位编程	0x61	Flash存储器的有效地址 必须64位对齐	FMC_ISPDAT :N/A FMC_MPDAT0: LSB 编程数据 FMC_MPDAT1: MSB编程数据 FMC_MPDAT2~FMC_MPDAT3: N/A
FLASH 多字编程	0x27	Flash存储器的有效地址 必须32位对齐	FMC_ISPDAT :N/A FMC_MPDAT0: 第1位编程数据 FMC_MPDAT1: 第2位编程数据 FMC_MPDAT2: 第3位编程数据 FMC_MPDAT3: 第4位编程数据
FLASH 32-位读	0x00	Flash存储器的有效地址 必须字对齐	FMC_ISPDAT: 返回数据
FLASH 64-位读	0x40	Flash存储器的有效地址 必须64位对齐	FMC_ISPDAT: LSB返回数据 FMC_MPDAT0: LSB返回数据 FMC_MPDAT1: MSB返回数据 FMC_MPDAT2~FMC_MPDAT3: N/A
读公司 ID	0x0B	0x0000_0000	FMC_ISPDAT: 0x0000_00DA
APROM 地址操作模式选择	0x2C	0x0000_0000 (地址 OP0) : BANK0 虚拟地址范围 0x00000~0x3FFFF BANK1虚拟地址范围 0x40000~0x7FFFF 0x0000_0001 (地址 OP1) : BANK0虚拟地址范围 0x40000~0x7FFFF BANK1虚拟地址范围 0x00000~0x3FFFF	0x5AA5_5AA5
读Checksum	0x0D	保存“运行Checksum计算值”地址	FMC_ISPDAT: 返回 Checksum
运算Checksum	0x2D	存储器的有效开始地址	FMC_ISPDAT: 大小

		必须512字节地址对齐	必须512字节地址对齐
读 Flash All-One 结果	0x08	保存 “运算 Flash All-One校验” 地址	FMC_ISPDAT: 返回结果 0xA110_0000 : Flash不是全为1 0xA11F_FFFF: Flash是全为1
运算 Flash All-One 校验	0x28	存储器的有效开始地址 必须512字节地址对齐	FMC_ISPDAT: 大小 必须512字节地址对齐
读Unique ID	0x04	0x0000_0000	FMC_ISPDAT: Unique ID 字 0
		0x0000_0004	FMC_ISPDAT: Unique ID 字 1
		0x0000_0008	FMC_ISPDAT: Unique ID 字 2
		0x0000_0070	FMC_ISPDAT[11:0]: 内置 VBG ADC 转换结果
重映射	0x2E	APROM或LDROM的有效地址，必须512字节对齐	N/A

表 6.4-4 ISP 命令列表

ISP 流程

FMC控制器提供了片上flash存储器的读，擦除和编程操作。一些FMC控制器的寄存器是写保护的，所以在设定之前要解锁。

在解锁了保护寄存器之后，用户需要设定FMC_ISPCTL控制寄存器来决定更新LDROM，APROM，SPROM或配置区，然后设定ISPEN (FMC_ISPCTL[0]) 来使能ISP功能。

一旦FMC_ISPCTL寄存器被设置成功，用户可以设定FMC_ISPCMD(参考上述ISP命令列表)来完成相应的操作。设置FMC_ISPADDR作为Flash存储器的目标地址。FMC_ISPDAT可以作为设定编程数据或作为读寄存器命令FMC_ISPCMD的返回数据。

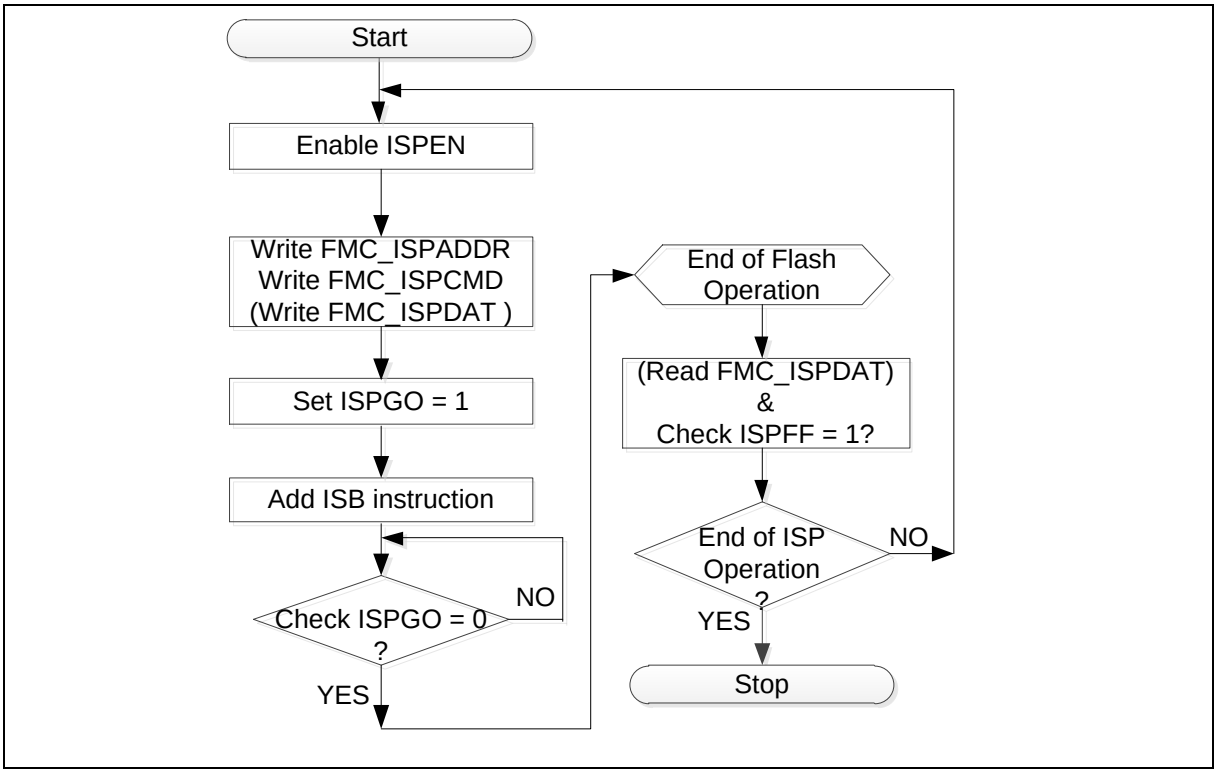


图 6.4-17 ISP 示范流程

最终，设定ISPGO (FMC_ISPTRG[0])寄存器来执行ISP功能。当ISP功能完成以后，ISPGO(FMC_ISPTRG[0])位自动清除。为了确保CPU继续执行指令前ISP功能已经完成，在ISPGO(FMC_ISPTRG[0])位被置1后将使用ISB(指令同步隔离)命令。

ISP完成后，几个错误条件需要检查。如果出现错误，ISP操作就不会开始并且ISP失败标志被置位。ISPFF(FMC_ISPSTS[6])标志只由软件清除，即使ISPFF(FMC_ISPSTS[6])为1，仍然可以触发下一个ISP流程，因此用户在ISP操作完成后需要检查ISPFF(FMC_ISPSTS[6])位，如果被置1要清0。

当ISPGO(FMC_ISPTRG[0])置位，FMC开始执行ISP命令，CPU将会等待ISP完成，在此期间外设仍然保持正常工作，但是任何中断请求都不会响应直到CPU完成ISP操作。当ISP操作完成后ISPGO位将被硬件自动清0。 用户可以通过ISPGO(FMC_ISPTRG[0])位来检查ISP操作是否完成。用户应该在ISPGO(FMC_ISPTRG[0])置1之后加ISB(指令同步隔离)指令，确保ISP操作之后的指令正确执行。

6.4.4.13 用于中断及存储器编程的VECMAP

通过 VECMAP 加速中断处理

IAP模式下，VECMAP功能可以映射512字节SRAM到中断向量区，也就是说可以将所有中断向量放在SRAM中，任何中断产生后，CPU可以零等待状态读取SRAM中的中断函数入口地址加速中断处理。

由于向量映射空间固定512字节，用户必须在映射SRAM到中断向量区之前拷贝所有512字节到SRAM，否则CPU在重映射后可能获取到错误的数据，图 6.4-18展示了一个VECMAP加速中断处理的范例。

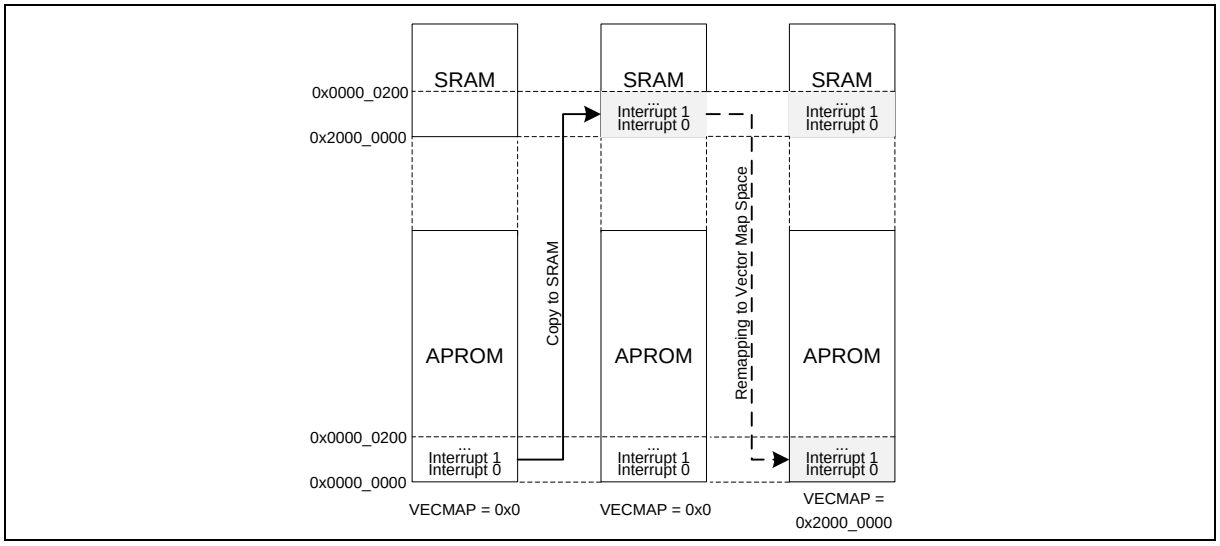


图 6.4-18 VECMAP 加速中断处理

Flash 编程时避免 CPU 停机

当Flash存储控制器（FMC）在工作时，CPU任何访问Flash的操作都会造成CPU停机等待FMC完成操作，如果FMC执行的是页擦除操作可能会造成CPU长时间停机等待。为了避免这种情况，用户在FMC页擦除时要避免访问flash，最简单的方法是在SRAM中执行程序，并且用VECMAP将所有中断都映射到SRAM中。在SRAM中执行程序CPU将不会访问flash取指令，将所有中断映射到SRAM中，CPU也不需要访问flash获取中断向量。

6.4.4.14　高速缓存控制器

高速缓存控制器将保存访问APROM和LDROM的每个Flash数据，以提高下次访问同一地址的速度。高速缓存控制器所采用的地址空间与CPU相同。高速缓存具有监视更改Flash内容的侦听机制，监控ISP相关命令包括：编程命令，擦除命令，和VECMAP命令。与编程相关的ISP命令，如果目标地址的数据已经在高速缓存中，则相应的数据将失效，而其他数据将保持不变。对于与擦除和VECMAP相关的ISP命令，所有高速缓存内容都将无效。由于CPU的地址空间取决于引导模式和VECMAP设置，当VECMAP处于有效状态或者MCU不是APROM启动，CPU和ISP命令（ISPADDR）的地址空间将不同。因此，当VECMAP处于有效状态或者MCU不是APROM启动时，如果需要使用与编程相关的ISP命令，则侦听检测机制无法成功检测出来。在这种情况下，需要手动发送缓存失效命令（通过在FMC_FTCTL中写入CACHEINV）。

6.4.4.15　片上Flash存储器编程

芯片提供32位，64位和多字编程功能加速Flash存储器更新处理。表 6.4-5 列出每种Flash编程功能的FMC 控制寄存器要求：

寄存器	描述	32-位 编程	64-位 编程	多字 编程
FMC_ISPCTL	ISP控制寄存器	●	●	●
FMC_ISPADDR	ISP 地址寄存器	●	●	●
FMC_ISPDAT	ISP数据寄存器	●	N/A	N/A
FMC_ISPCMD	ISP 命令 寄存器	0x21	0x61	0x27
FMC_ISPTRG	ISP 触发寄存器	●	●	●

FMC_ISPSTS	ISP 状态寄存器	●	●	N/A
FMC_MPDAT0	ISP 数据0寄存器	N/A	●	●
FMC_MPDAT1	ISP 数据1寄存器	N/A	●	●
FMC_MPDAT2	ISP 数据2寄存器	N/A	N/A	●
FMC_MPDAT3	ISP 数据3寄存器	N/A	N/A	●
FMC_MPSTS	ISP 多字编程状态	N/A	N/A	●
FMC_MPADDR	ISP 多字编程地址	N/A	N/A	●

表 6.4-5 Flash 编程的 FMC 控制寄存器

64 位编程

NuMicro® M031/M032系列64位编程功能比32位编程要快。FMC_ISPDAT用于32位编程数据寄存器。在64位编程中，有2个编程数据寄存器，一个是FMC_MPDAT0用于LSB字，另一个是FMC_MPDAT1用于MSB字，ISP命令是0x61，其它寄存器同32位编程。图 6.4-19 / 图 6.4-20 为ISP 32位/ 64位编程流程。

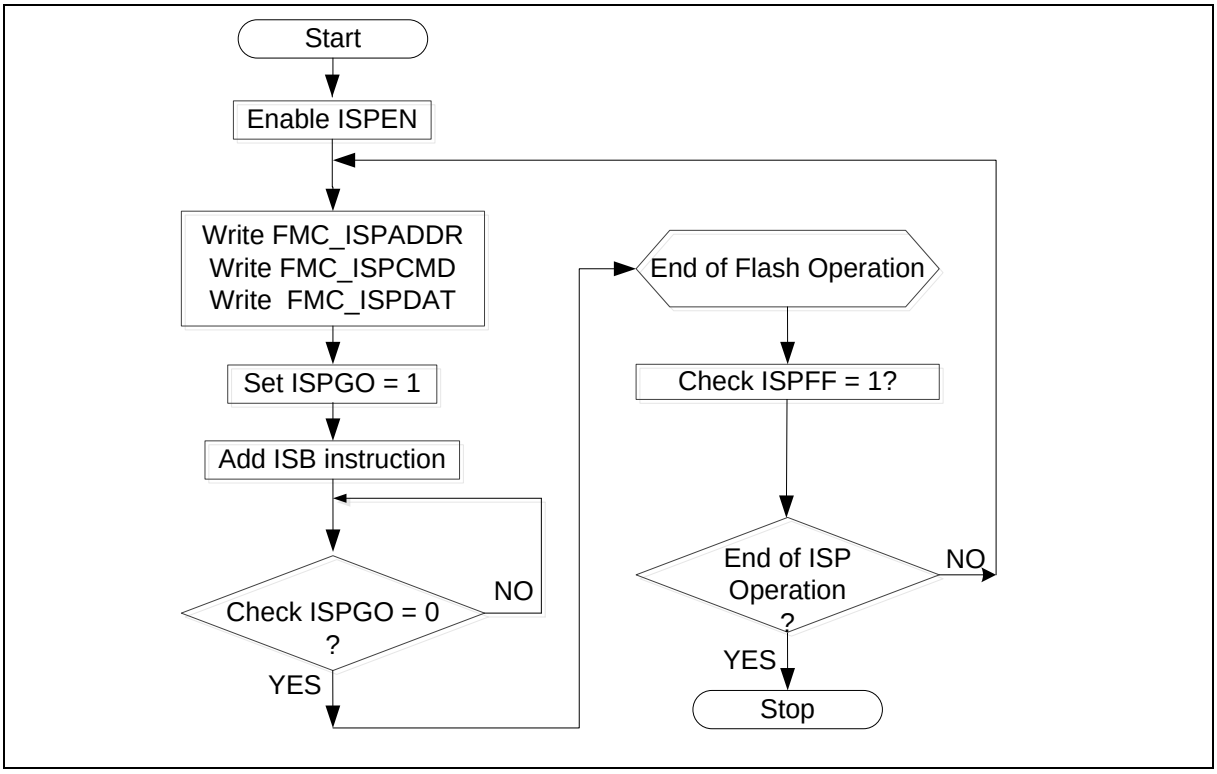


图 6.4-19 ISP 32 位编程流程

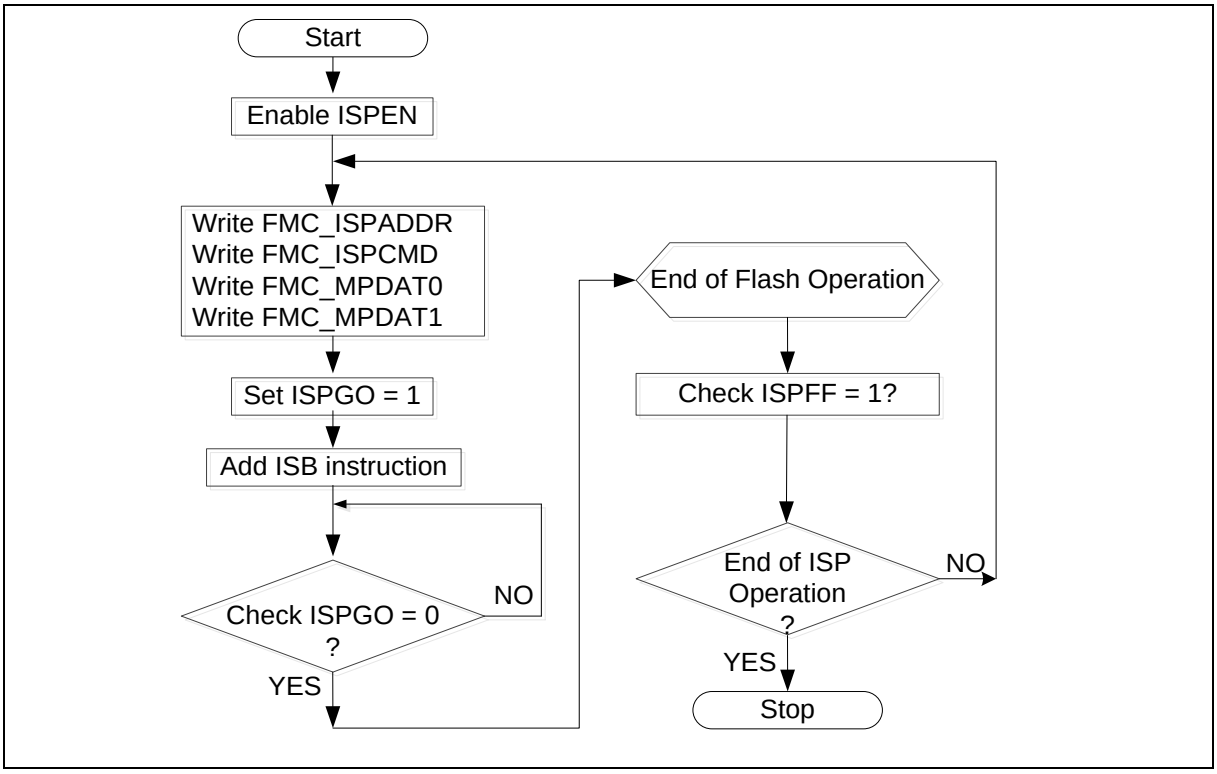


图 6.4-20 ISP 64 位编程流程

多字编程

M031/M032系列支持多字编程功能来提高Flash更新速度。最大编程长度可达到256字节，最小编程长度时8字节(2字)。如果编程多于8个字节，多字编程是最快的。

多字编程操作中，CPU监视编程缓存的空状态，并持续准备下一个数据。多字编程的固件不能放在APROM，LDROM或SPROM中，因为CPU取指令不能在ISP过程中停止。固件应存放在片上SRAM中，以避免CPU停止。多字编程代码必须保证所有异常（中断）处理不能在ISP过程中访问Flash。请注意，多字编程需要HCLK运行最少48MHz。

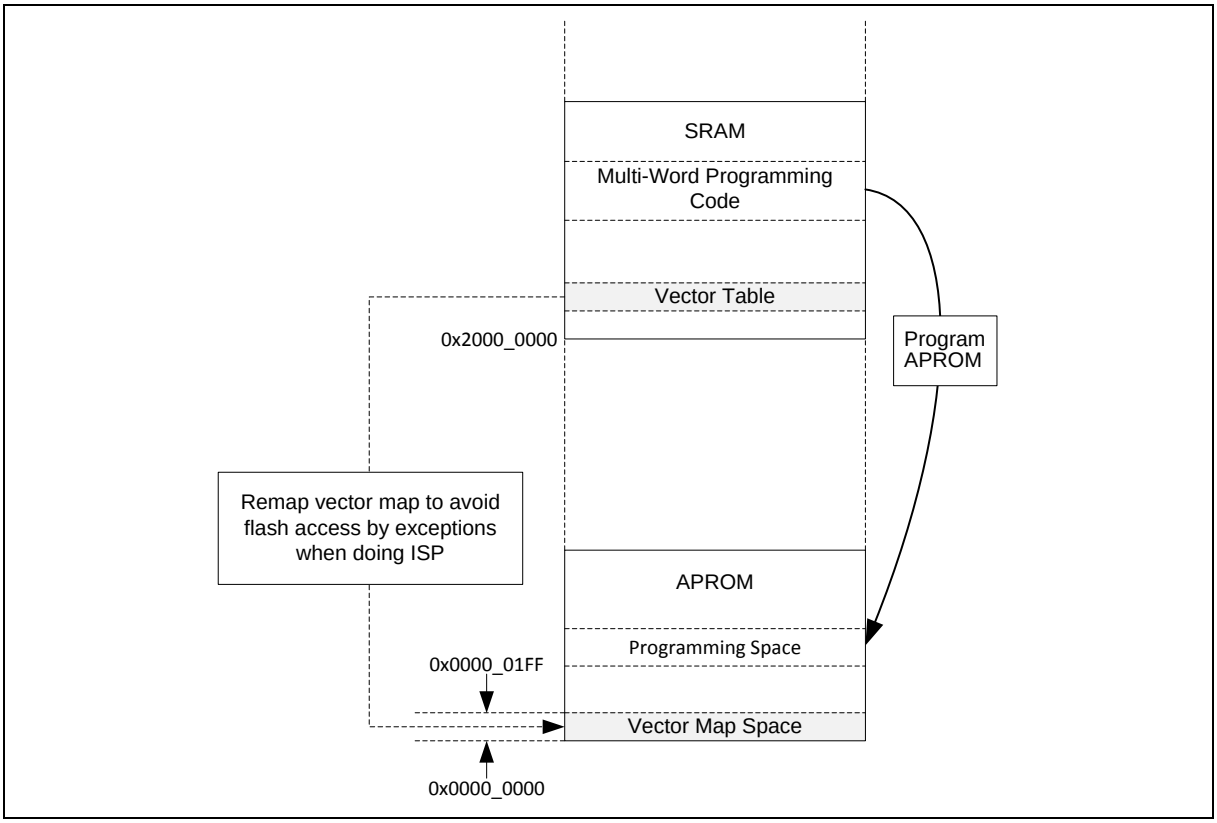


图 6.4-21 多字编程固件运行在 SRAM

多字编程的流程如下。ISP的起始地址（FMC_ISPADDR）是8字节对齐的。FMC_MPDAT0是第一个数据字（偏移0x0），FMC_MPDAT1是第二个数据字（偏移0x04），FMC_MPDAT2是第三个数据字（偏移0x08），FMC_MPDAT3是第四个数据字（偏移0x0C）。如果ISP起始地址FMC_ISPADDR[3]=0，第一个字数据应放在FMC_MPDAT0，第二个字数据应放在FMC_MPDAT1，第三个字数据应放在FMC_MPDAT2，第四个字数据应放在FMC_MPDAT3。如果ISP起始地址FMC_ISPADDR[3]=1，第一个字数据应放在FMC_MPDAT2，第二个字数据应放在FMC_MPDAT3，第三个字数据应放在FMC_MPDAT0，第四个字数据应放在 FMC_MPDAT1。最大的可编程大小是256字节。当FMC控制器在多字编程操作，CPU需要监测缓冲器的状态D3~D0（FMC_MPSTS[7：4]）以及MPBUSY（FMC_MPSTS[0]）来等待缓冲器清空（(D1,D0)=00或(D3,D2)=00)），然后CPU需要及时更新下一次的编程数据（FMC_MPDAT0，FMC_MPDAT1，FMC_MPDAT2和FMC_MPDAT3）。否则，FMC控制器将退出多字编程操作（MPBUSY（FMC_MPSTS[0]）=0）。如果CPU无法及时更新数据 （MPBUSY（FMC_MPSTS[0] ）=0），根据FMC_MPADDR提供的上一次编程地址的信息，CPU需要重新开始一个新的多字编程来延续流程。最后，CPU必须检查ISPFF（FMC_MPSTS[2]）来确认多字操作是否已经成功的完成了。

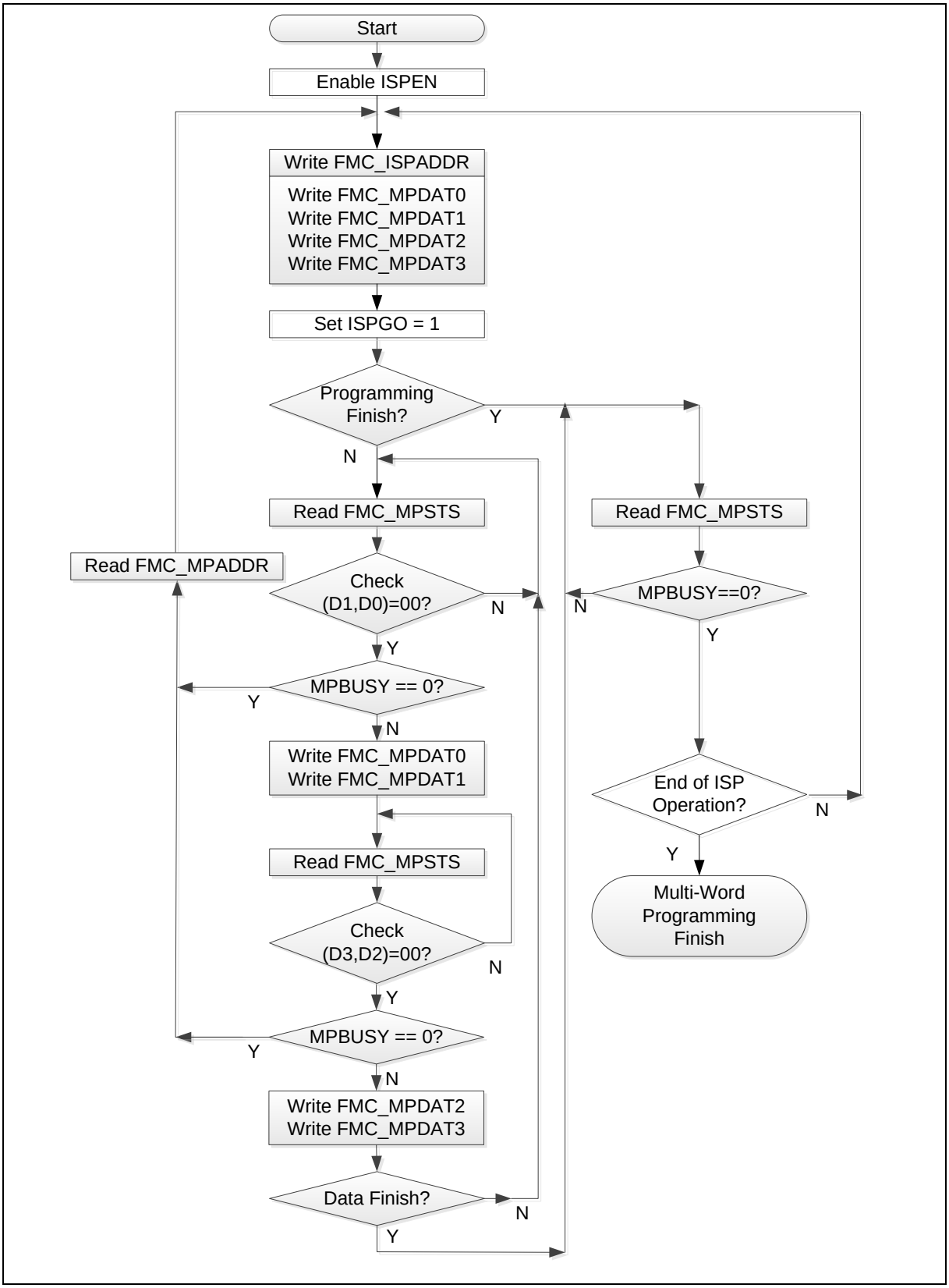


图 6.4-22 多字编程流程

6.4.4.16 CRC32 Checksum 计算

M031/M032系列支持CRC-32 checksum计算功能来帮助用户快速检查存储器内容，包括APROM ，LDROM和 SPROM。CRC-32的多项式如下：

$$\text{CRC-32: } X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

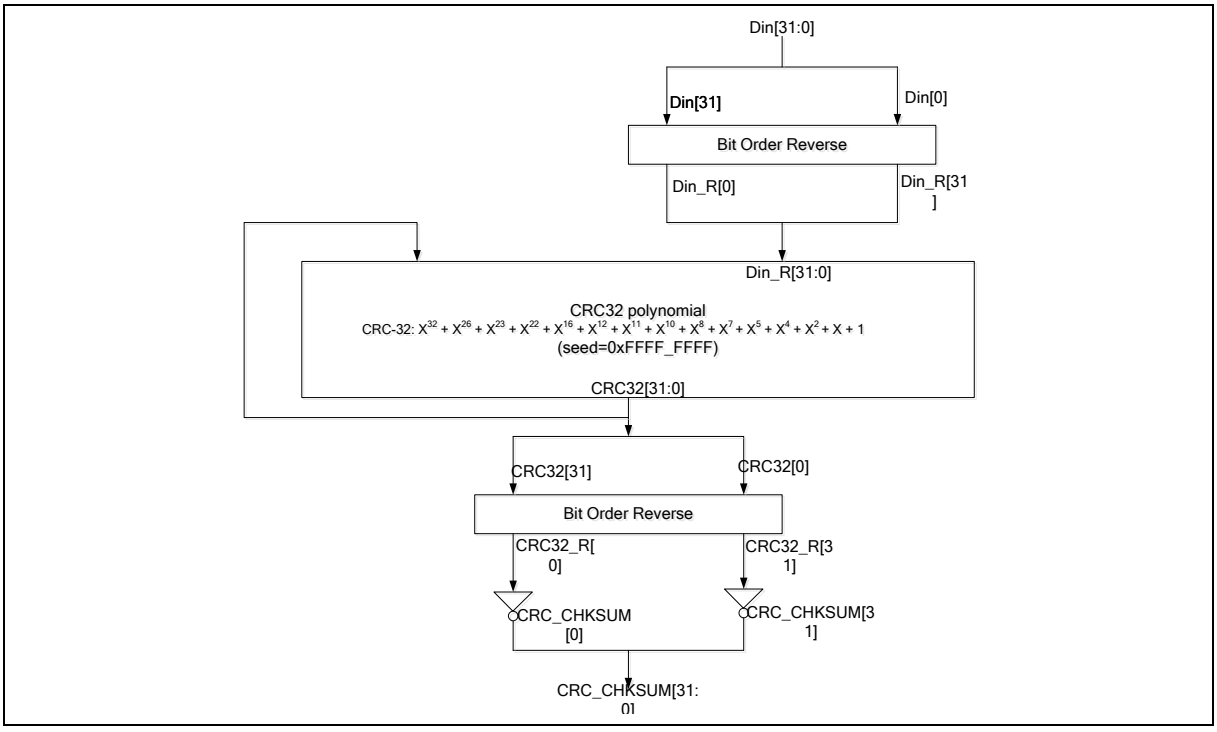


图 6.4-23 CRC-32 Checksum 计算

三步完成这个CRC-32checksum计算：

1. 执行ISP“运行存储器Checksum”操作：用户先设定存储器的开始地址(FMC_ISPADDR)和大小(FMC_ISPDAT)用于计算。地址和大小都要512字节对齐。大小必须是512字节的整数倍，起始地址必须是在APROM,LDROM和SPROM的存储空间。
2. 执行ISP“读存储器Checksum”操作：FMC_ISPADDR保持和步骤一中设置的一样。
3. 读FMC_ISPDAT 获得checksum：checksum结果从FMC_ISPDAT读取。如果checksum是0x0000_0000，有两种可能，（1）checksum还在计算中（2）地址和大小超出了设备的限制。

当SPROM设置为安全模式时，CPU和ISP的读命令不能直接读SPROM中的内容，但用户可以用checksum功能来验证SPROM中的内容是否正确。

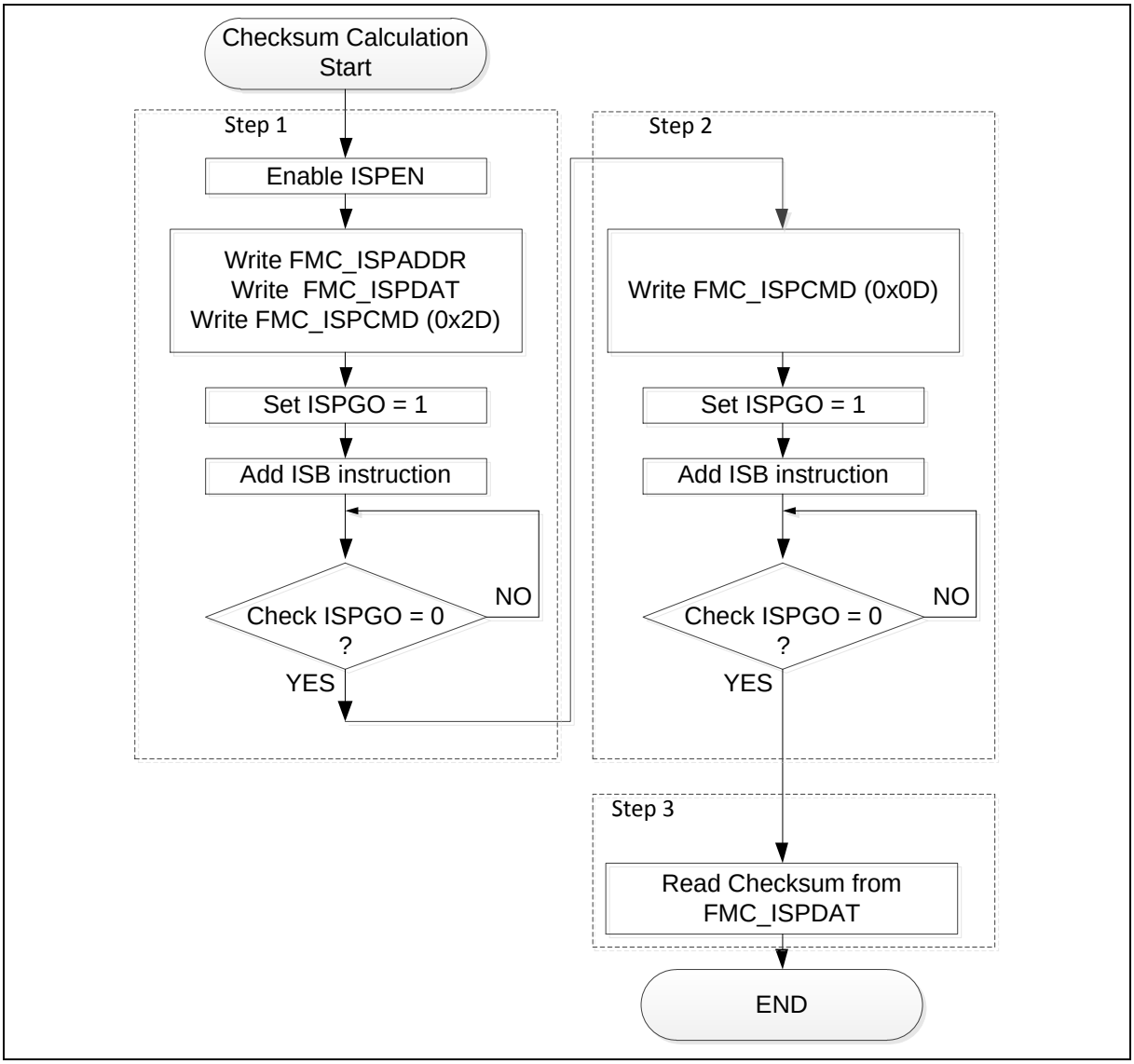


图 6.4-24 CRC-32 Checksum 计算流程

6.4.4.17 Flash 全为1校验

此芯片支持Flash全为1校验功能来帮助用户快速检查APROM和LDROM在Flash擦除操作后一个内存块内容是否为空白。

完成flash全为1的校验有两个步骤：

1. 运行ISP“全是1”命令并检测命令运行完成FMC_ISPSTS[0]为0。
2. 运行ISP“读取全是1”命令并检测命令运行完成FMC_ISPSTS[0]为0。然后检查FMC_ISPDAT内容，如果值为0，重复步骤2。如果指定的Flash区域都为空，FMC_ISPDAT 将为0xA11FFFFF，否则，FMC_ISPDAT为0xA1100000。

在步骤1中，用户需设置内存的起始地址 （FMC_ISPADDR）和大小（FMC_ISPDAT）用于检验，两者都要512字节对齐，大小需≥512字节，起始地址适用于APROM和LDROM。

在步骤2中，FMC_ISPADDR 需与步骤1保持一致。

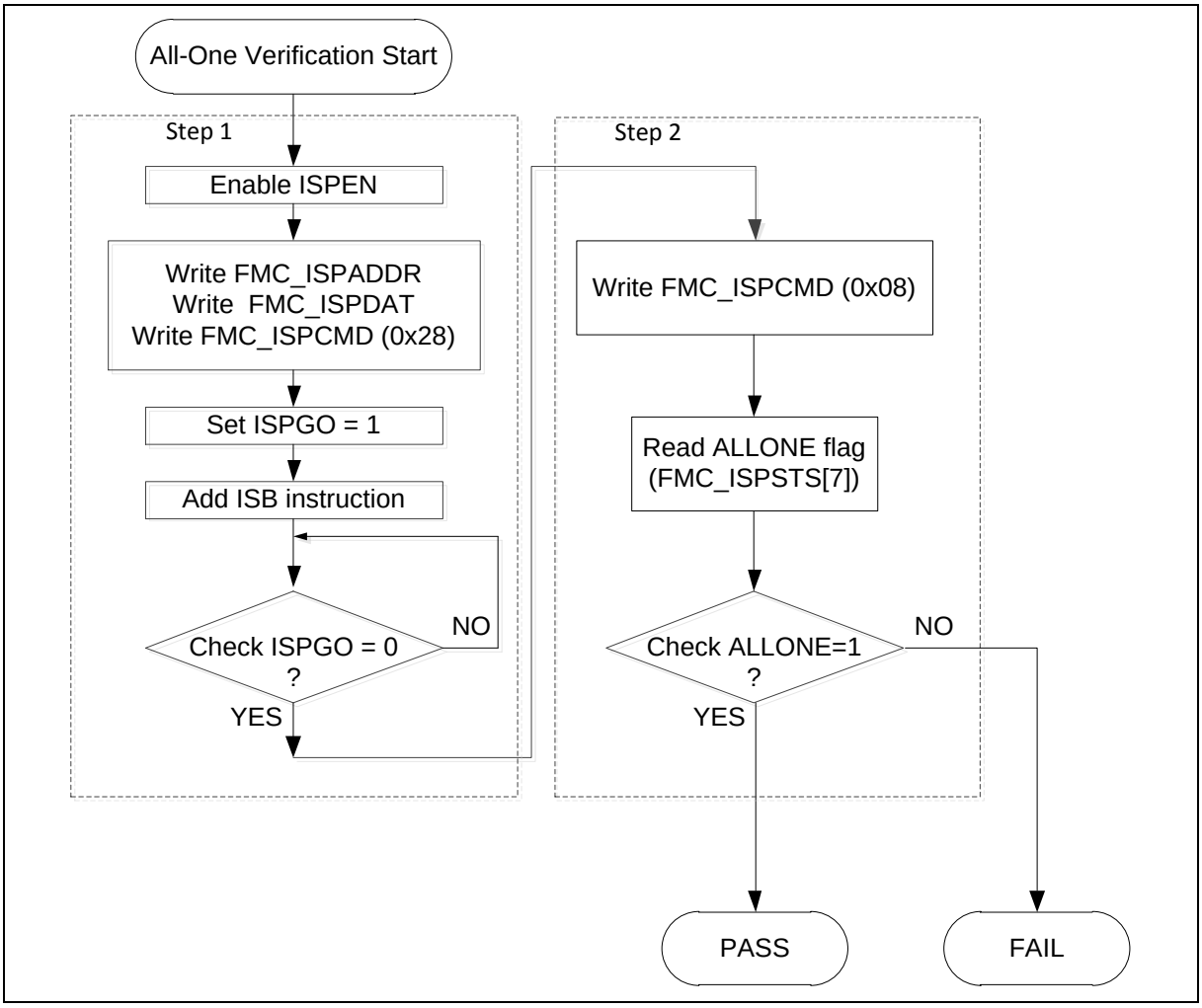


图 6.4-25 全是 1 校验流程

6.4.5 寄存器映射

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移量	R/W	描述	复位值
FMC Base 地址: **FMC_BA = 0x4000_C000**				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP 控制寄存器	0x0000_000X
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP 地址寄存器	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP 数据寄存器	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP 命令寄存器	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP 触发控制寄存器	0x0000_0000
FMC_DFBA	FMC_BA+0x14	R	数据 Flash 基地址	0xXXXX_XXXX
FMC_FTCTL	FMC_BA+0x18	R/W	Flash访问时间控制寄存器	0x0000_0000
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP 状态寄存器	0xX0X0_000X
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP数据0寄存器	0x0000_0000
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP数据1寄存器	0x0000_0000
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP数据2寄存器	0x0000_0000
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP数据3寄存器	0x0000_0000
FMC_MPSTS	FMC_BA+0xC0	R	ISP 多字编程状态寄存器	0x0000_0000
FMC_MPADDR	FMC_BA+0xC4	R	ISP 多字编程地址寄存器	0x0000_0000

6.4.6 寄存器描述

ISP控制寄存器(FMC_ISPCTL)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP控制寄存器	0x0000_000X

31	30	29	28	27	26	25	24
Reserved							**INTEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**ISPFF**	**LDUEN**	**CFGUEN**	**APUEN**	**SPUEN**	**BS**	**ISPEN**

位	描述	
[31:15]	**Reserved**	保留.
[24]	**INTEN**	**ISP 中断使能位（写保护）** 0 = ISP中断禁止 1 = ISP中断使能. **注：**该位是写保护位，参看SYS_REGLCTL 寄存器。使用中断前，需要清除INTFLAG(FMC_ISPSTS[8]) 保证中断在正确时间产生。
[23:7]	**Reserved**	保留.
[6]	**ISPFF**	**ISP失败标志（写保护）** 当ISP遇到下面任何一种情况将会触发硬件置起这个位： 该位需要写1进去才能清掉。 APUEN等于0时，APROM 写APROM LDUEN等于0时，LDROM 写LDROM. CFGUEN等于0时，CONFIG 被擦除或编程 SPUEN等于0时，SPROM被擦除或编程。 在SPROM安全模式下，对SPROM进行编程 ICE连接状况下，LOCK模式，下页擦除命令 在欠压检测时，下擦除或编程命令 目标地址无效，比如超出了有效范围 无效的ISP命令 **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器

[5]	**LDUEN**	**LDROM 更新使能 (写保护)** LDROM 更新使能位. 0 = LDROM不能更新. 1 = LDROM 能更新. **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器
[4]	**CFGUEN**	**CONFIG更新使能 (写保护)** 0 = CONFIG 不能更新 1 = CONFIG 能更新 **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器.
[3]	**APUEN**	**APROM 更新使能（写保护）** 0 = APROM 不能更新，当芯片运行在APROM中 1 = APROM 能更新当芯片运行在APROM中 **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器.
[2]	**SPUEN**	**SPROM 更新使能（写保护）** 0 = SPROM 不能更新. 1 = SPROM 能更新. **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器.
[1]	**BS**	**启动选择（写保护）** 设置/清除该位用于选择下次启动从LDROM/APROM,同时该位也作为芯片启动状态标志，用于检查芯片是从什么地方启动的。除了CPU 复位(RSTS_CPU = 1)及system 复位(RSTS_SYS)外的任何复位发生时，该位由CBS[1] (CONFIG0[7])的反转值初始化。 0 = 启动从APROM. 1 = 启动从LDROM. **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器
[0]	**ISPEN**	**ISP 使能（写保护）** ISP 功能使能位，设置该位使能ISP功能 0 = 禁用ISP功能. 1 = 使能 ISP功能. **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器

ISP地址(FMC_ISPADDR)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP地址寄存器	0x0000_0000

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

位	描述	
[31:0]	**ISPADDR**	**ISP地址** M031/M032系列内嵌FLASH。ISP 32-位操作时，ISPADDR[1:0]必须为00。CRC32 Checksum 计算命令，该寄存器为Flash开始地址，必须512字节对齐 。 16/32/64/128 K字节Flash的芯片： 向量页重映射命令，ISPADR[8:0] 必须全部为0。 256/512 K字节 Flash的芯片： 向量页重映射命令，ISPADR[11:0] 必须全部为0。

ISP数据寄存器(FMC_ISPDAT)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP数据寄存器	0x0000_0000

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

位	描述	
[31:0]	**ISPDAT**	**ISP数据** 在ISP写操作之前，写数据到该寄存器 在ISP读操作之后，读该寄存器中数据. 执行CheckSum计算时，ISPDAT代表内存大小(字节)，512字节对齐。 读ISP CheckSum时，ISPDAT代表CheckSum的计算结果。 ISPDAT = 0x0000_0000，代表(1) 还在CheckSum计算中，(2) CheckSum 计算的内存范围不正确。

ISP 命令 (FMC_ISPCMD)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP命令寄存器	0x0000_0000

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

位	描述	
[31:7]	**Reserved**	保留.
[6:0]	**CMD**	**ISP CMD** ISP 命令如下： 0x00= Flash读. 0x04=读UID. 0x08= 读All One. 0x0B= 读公司ID. 0x0C= 读 设备ID. 0x0D= 读 CRC32 Checksum. 0x21= Flash 32位编程 0x22= Flash 页擦除 0x23= Flash APROM Bank 擦除. 0x27= Flash 多字编程. 0x28= 运行 All One. 0x2C= APROM地址操作模式选择 0x2D= 有效 CRC32 Checksum 计算. 0x2E= 向量重映射. 0x61= Flash 64位编程 其它命令无效

ISP触发控制寄存器(FMC_ISPTRG)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP触发控制寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留.
[0]	**ISPGO**	**ISP 开始触发（写保护）** 写1到该位开始ISP操作，当ISP操作完成后该位由硬件自动清 0 = ISP 操作完成. 1 = ISP 操作中. **注:** 该位是写保护位，参看SYS_REGLCTL 寄存器

数据 Flash基地址寄存器(FMC_DFBA)

寄存器	偏移量	R/W	描述	复位值
FMC_DFBA	FMC_BA+0x14	R	数据Flash基地址	0xXXXX_XXXX

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

位	描述	
[31:0]	**DFBA**	**数据Flash 基地址** 该寄存器是数据Flash的起始地址，只读属性 数据Flash与APROM共用存储空间。从CONFIG1加载寄存器内容。 当DFEN (CONFIG0[0]) =0寄存器生效

Flash访问时间控制寄存器(FMC_FTCTL)

寄存器	偏移量	R/W	描述	复位值
FMC_FTCTL	FMC_BA+0x18	R/W	Flash访问时间控制寄存器	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留.
[9]	**CACHEINV**	**Flash 高速缓存无效（写保护）** 0 = Flash高速缓存结束后失效（默认） 1 = Flash 高速缓存无效 注 1：写1高速缓存开始无效。一旦处理器完成，值将变为0。 注 2：该为是写保护，详情请参看SYS_REGLCTL 寄存器 注 3：仅支持 256/512 K字节Flash芯片。
[8]	**Reserved**	保留.
[7]	**BBOFF**	**Flash Branch缓存禁止控制位（写保护）** 0 = Flash Branch缓存功能使能（默认） 1 = Flash Branch缓存禁止 注1：该为是写保护，详情请参看SYS_REGLCTL 寄存器 注 2：仅支持16/32/64/128 K字节Flash芯片芯片。

[6:4]	**FOM**	**频率选择模式（写保护）** M031/M032 系列支持可调整的flash访问时间来优化不同工作模式下的flash访问周期时间。 16/32/64/128K字节Flash的芯片： 000 = 频率小于或等于 48 MHz 001 =频率小于或等于24 MHz 其它 = 保留. 256/512 K字节Flash的芯片： 000 =频率小于或等于72 MHz 001 =频率小于或等于12 MHz 010 =频率小于或等于36 MHz 011 =频率小于或等于60 MHz Others = 保留 **注：** 该为是写保护，详情请参看SYS_REGLCTL 寄存器。
[3:0]	**Reserved**	保留

ISP状态寄存器(FMC_ISPSTS)

寄存器	偏移量	R/W	描述	复位值
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP状态寄存器	0xX0X0_000X

31	30	29	28	27	26	25	24
SCODE	**FBS**	**VECMAP**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VECMAP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECMAP							**INTFLAG**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ALLONE	**ISPFF**	**PGFF**	**Reserved**		**CBS**		**ISPBUSY**

位	描述	
[31]	**SCODE**	**保护代码起效标志位** 当flash在初始化的时候检测到SPROM的代码保护功能起效时，硬件将该位置1。或者软件将该位置1用于使代码保护功能起效。该位只有在SPROM页擦除的操作下才清掉。 0 = SPROM 禁用代码保护功能 1 = SPROM 启用代码保护功能
[30]	**FBS**	**Flash Bank 选择指示** 该位指示哪个APROM地址模式被选择来启动。 0 = 地址模式OP0被选择启动 1 = 地址模式OP1被选择启动 **注：**仅支持 256/512 K字节Flash芯片。
[29:9]	**VECMAP**	**向量页的映射地址（只读）** 所有对地址 0x0000_0000~0x0000_01FF的访问都被映射到flash存储器或SRAM中，其向量地址存在 {VECMAP[20:0], 9'b0} ~ {VECMAP[20:0], 9'h1FF}中,除了SPROM。 VECMAP [20:19] = 00 系统地址映射到flash 存储器 VECMAP [20:19] = 10 系统地址映射到SRAM存储器 VECMAP [18:12] 将为0
[8]	**INTFLAG**	**ISP 命令完成中断标志** 0 = ISP 没有完成 1 = ISP完成或ISPFF置位 **注：**仅支持 256/512 K字节Flash芯片。
[7]	**ALLONE**	**Flash All-one校验标志** 该位由硬件置位，如果Flash所有位都为1，或如所有位不全为1，清0，在“Run Flash All-One 校验”完成后。该位也能写1清0。 0 = Flash 不是所有都为1在“Run Flash All-One 校验”完成后 1 = Flash所有都为1在“Run Flash All-One 校验”完成后

[6]	**ISPFF**	**ISP 失败标志（写保护位）** 该位是寄存器ISPFF (FMC_ISPCTL[6])的镜像，需向FMC_ISPCTL[6] 或FMC_ISPSTS[6] 写1清零。 当已启动的ISP符合下面的任何一个条件时，该位由硬件置1： (1) APUEN等于0时，APROM 写APROM. (2) LDUEN等于0时，LDROM 写LDROM. (3) CFGUEN等于0时，CONFIG 被擦除或编程. (4)SPUEN等于0时，SPROM被擦除或编程 (5)在SPROM安全模式下，对SPROM进行编程 (6) ICE连接状况下，LOCK模式，下页擦除命令 (7)在欠压检测时，下擦除或编程命令 (8)目标地址无效，比如超出了有效范围 (9) 无效的ISP命令
[5]	**PGFF**	**Flash编程快速校验标志（只读）** 当ISP编程校验数据不匹配，该位置1。该位清除通过ISP Flash、擦除或ISP 读CID操作。 0 = Flash 编程成功 1 = Flash 编程失败。编程数据与Flash存储器数据不同。
[4:3]	**Reserved**	保留.
[2:1]	**CBS**	**配置中设置的启动选择（只读）** 除了CPU 复位 (RSTS_CPU= 1)及系统 复位 (RSTS_SYS)外的任何复位发生时，该位由CBS (CONFIG0[7:6])初始化。 00 = LDROM支持 IAP模式 01 = LDROM不支持 IAP模式 10 = APROM 支持IAP模式 11 = APROM 不支持 IAP模式
[0]	**ISPBUSY**	**ISP 忙标志（只读）** 0 = ISP 操作完成 1 = ISP 操作中

ISP 数据 0 寄存器 (FMC_MPDAT0)

寄存器	偏移量	R/W	描述	复位值
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP 数据0寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT0							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT0							

位	描述	
[31:0]	**ISPDAT0**	**ISP 数据 0** 该寄存器是32位/64位/多字编程的第一个32位数据，也是FMC_ISPDAT镜像，两个为同样的值。

ISP 数据1寄存器 (FMC_MPDAT1)

寄存器	偏移量	R/W	描述	复位值
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP数据1寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT1							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT1							

位	描述	
[31:0]	**ISPDAT1**	**ISP 数据 1** 该寄存器是32位/64位/多字编程的第二个32位数据。

ISP数据2寄存器(FMC_MPDAT2)

寄存器	偏移量	R/W	描述	复位值
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP数据2寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT2							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT2							

位	描述	
[31:0]	**ISPDAT2**	**ISP 数据 2** 该寄存器是32位/64位/多字编程的第三个32位数据。

ISP数据3寄存器(FMC_MPDAT3)

寄存器	偏移量	R/W	描述	复位值
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP数据3寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT3							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT3							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT3							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT3							

位	描述	
[31:0]	**ISPDAT3**	**ISP 数据 3** 该寄存器是32位/64位/多字编程的第四个32位数据。

ISP 多字编程状态寄存器 (FMC_MPSTS)

寄存器	偏移量	R/W	描述	复位值
FMC_MPSTS	FMC_BA+0xC0	R	ISP多字编程状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D3	**D2**	**D1**	**D0**	**Reserved**	**ISPFF**	**PPGO**	**MPBUSY**

位	描述	
[31:8]	**Reserved**	保留.
[7]	**D3**	**ISP 数据3标志（只读）** 当FMC_MPDAT3被写入该位置1，当FMC_MPDAT3编程到Flash完成，该位自动清0。 0 = FMC_MPDAT3寄存器是空，或编程到Flash完成 1 = FMC_MPDAT3 寄存器被写入值，但还未编程到Flash完成
[6]	**D2**	**ISP 数据2标志（只读）** 当FMC_MPDAT2被写入该位置1，当FMC_MPDAT2编程到Flash完成，该位自动清0。 0 = FMC_MPDAT2寄存器是空，或编程到Flash完成 1 = FMC_MPDAT2 寄存器被写入值，但还未编程到Flash完成
[5]	**D1**	**ISP 数据1标志（只读）** 当FMC_MPDAT1被写入该位置1，当FMC_MPDAT1编程到Flash完成，该位自动清0。 0 = FMC_MPDAT1寄存器是空，或编程到Flash完成 1 = FMC_MPDAT1 寄存器被写入值，但还未编程到Flash完成
[4]	**D0**	**ISP 数据0标志（只读）** 当FMC_MPDAT0被写入该位置1，当FMC_MPDAT0编程到Flash完成，该位自动清0。 0 = FMC_MPDAT0寄存器是空，或编程到Flash完成 1 = FMC_MPDAT0 寄存器被写入值，但还未编程到Flash完成
[3]	**Reserved**	保留.

[2]	**ISPFF**	**ISP失败标志（只读）** 该位是ISPFF (FMC_ISPCTL[6])镜像，通过写1到FMC_ISPCTL[6] 或 FMC_ISPSTS[6]来清除该位。 该位硬件置1通过以下条件触发ISP：. APROM写APROM当 APUEN=0 LDROM写LDROM当 LDUEN=0 CONFIG被擦除/编程当 CFGUEN=0 当ICE连接在LOCK模式下，页擦除命令执行 在欠压检测下，擦除编程命令执行 目标地址无效，比如超出了有效范围 无效的ISP命令
[1]	**PPGO**	**ISP多字编程状态（只读）** 0 = ISP多字编程操作无效 1 = ISP多字编程操作正在执行
[0]	**MPBUSY**	**ISP多字编程忙标志（只读）** .写1开始ISP多字编程操作，当ISP多字编程操作结束该位硬件清0。 该位是 ISPGO(FMC_ISPTRG[0])镜像 0 = ISP多字编程操作结束 1 = ISP多字编程操作正在执行

ISP多字编程地址寄存器(FMC_MPADDR)

寄存器	偏移量	R/W	描述	复位值
FMC_MPADDR	FMC_BA+0xC4	R	ISP多字编程地址寄存器	0x0000_0000

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

位	描述	
[31:0]	**MPADDR**	**ISP多字编程地址** 当 ISPGO标志为1，MPADDR是多字编程操作的地址。 当多字编程结束，MPADDR将保存最后的ISP地址。

6.5 通用I/O (GPIO)

6.5.1 概述

此芯片最多有111个GPIO，通过芯片配置，这些I/O可以作为其他功能接口，这111个I/O分别位于8个端口，分别是PA，PB，PC，PD，PE，PF，PG和PH。PA和PB端口有16个I/O，PC端口有15个I/O，PD和PE端口有16个I/O，PF端口有14个I/O，PG端口有10个I/O，PH端口有8个I/O。每个I/O口都可以独立配置管脚模式功能及数据。

每个I/O类型可以通过软件单独配置为输入，推挽输出，开漏输出或准双向模式，芯片复位后，I/O模式由CIOINI (CONFIG0[10]) 决定。

6.5.2 特性

- 4种I/O口模式:
  - 准双向模式
  - 推挽输出模式
  - 开漏输出模式
  - 输入模式
- 可选TTL/Schmitt 触发输入
- I/O 可配置成边沿/电平触发的中断源
- 通过(CONFIG0[10]) 设置，可以配置所有IO口在复位后的默认模式
  - CIOIN = 0，复位后所有GPIO管脚为准双向模式。
  - CIOIN = 1，复位后所有GPIO管脚为高阻输入模式
- 仅在准双向模式，I/O 管脚的内部上拉电阻使能
- 使能管脚中断功能同时也使能了唤醒功能

6.5.3 框图

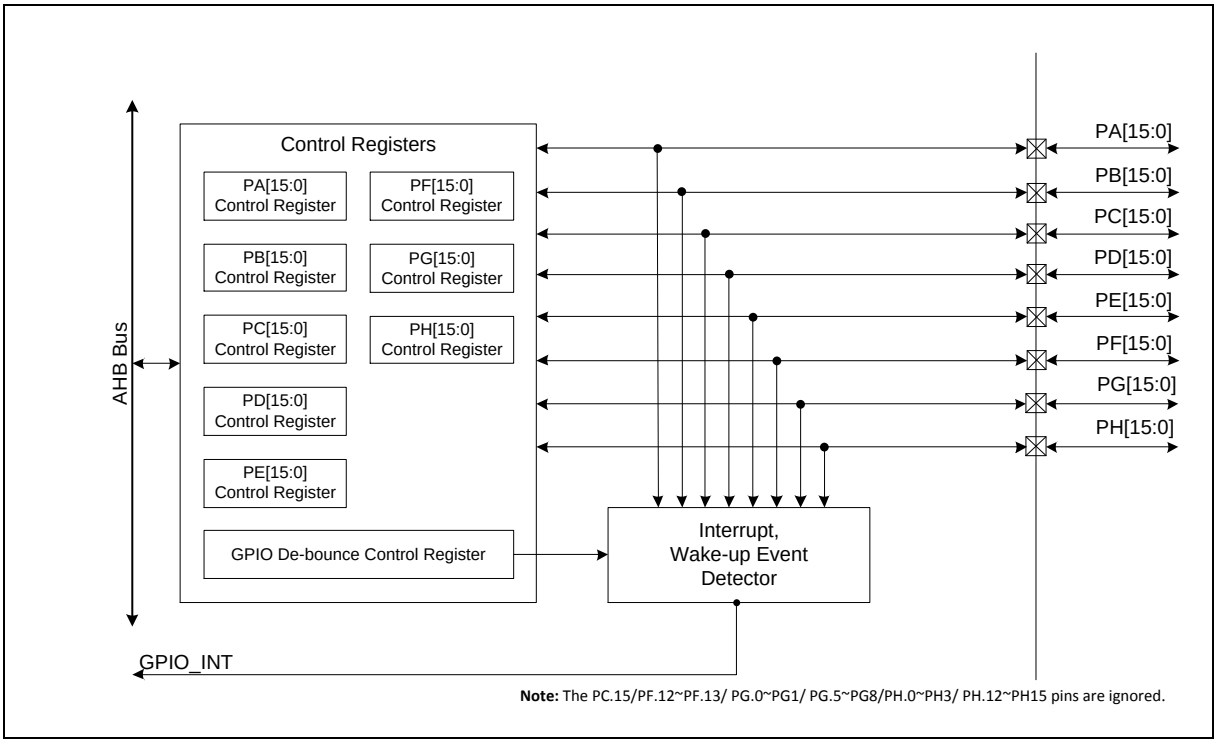


图 6.5-1 GPIO 控制器框图

注: PC.15/PF.12~PF.13/ PG.0~PG1/ PG.5~PG8/PH.0~PH3/ PH.12~PH15 引脚忽略

6.5.4 基本配置

- 复位配置
  - 通过GPIORST SYS_IPRST1[1]复位GPIO
- 管脚配置

组	管脚名称	GPIO	MFP
INT0	INT0	PA.6, PB.5	MFP15
INT1	INT1	PA.7, PB.4, PD.15	MFP15
INT2	INT2	PB.3, PC.6	MFP15
INT3	INT3	PB.2, PC.7	MFP15
INT4	INT4	PB.6	MFP13
		PA.8, PF.15	MFP15
INT5	INT5	PB.7	MFP13
		PD.12, PF.14	MFP15

6.5.5 功能描述

6.5.5.1 输入模式

设置MODEn(Px_MODE[2n+1:2n])为00使Px.n管脚变成输入模式，此时管脚为三态（高阻），没有输出能力。管脚(Px_PIN[n])值反应相应端口的状态。

6.5.5.2 推挽输出模式

图 6.5-2展示了推挽输出电路，设置 MODEn (Px_MODE[2n+1:2n])为01，Px.n 管脚为推挽输出模式，I/O支持数字输出功能，有拉/灌电流能力。DOUT(Px_DOUT[n])相应位的值被送到相应管脚上。

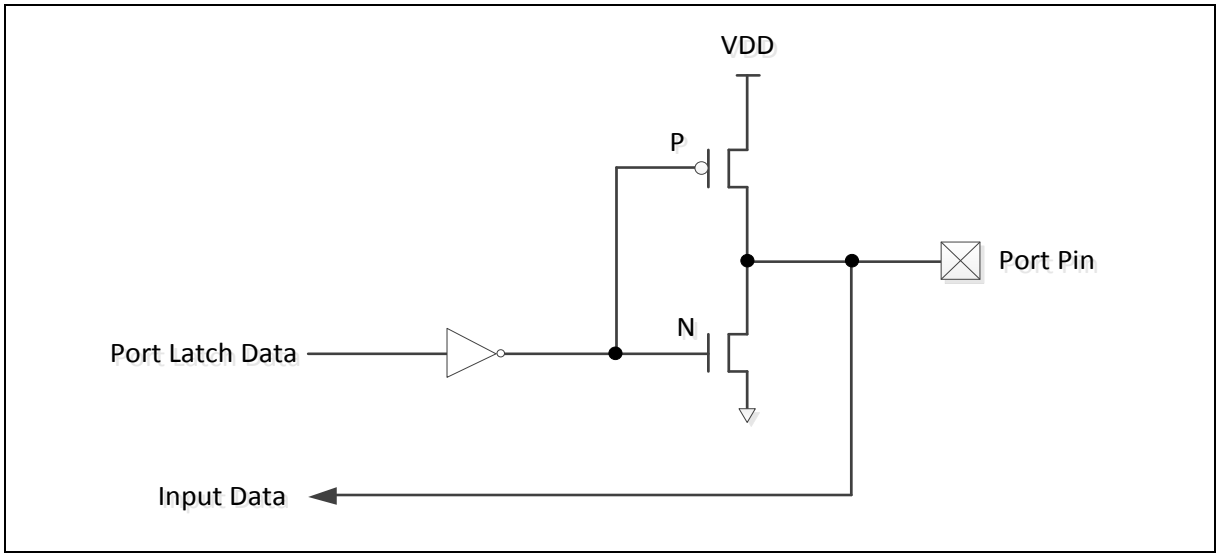


图 6.5-2 推挽输出

6.5.5.3 开漏输出模式

图 6.5-3 开漏输出电路，置MODEn(Px_MODE[2n+1:2n])为10，Px.n管脚为开漏模式且I/O管脚数字输出功能仅支持灌电流，驱动到高电平需要一个外加上拉电阻。如果DOUT (Px_DOUT[n])相应位为‘0’，管脚上输出低。如果DOUT(Px_DOUT[n])相应位为‘1’，该管脚输出为高阻，可以由外部上拉电阻控制。

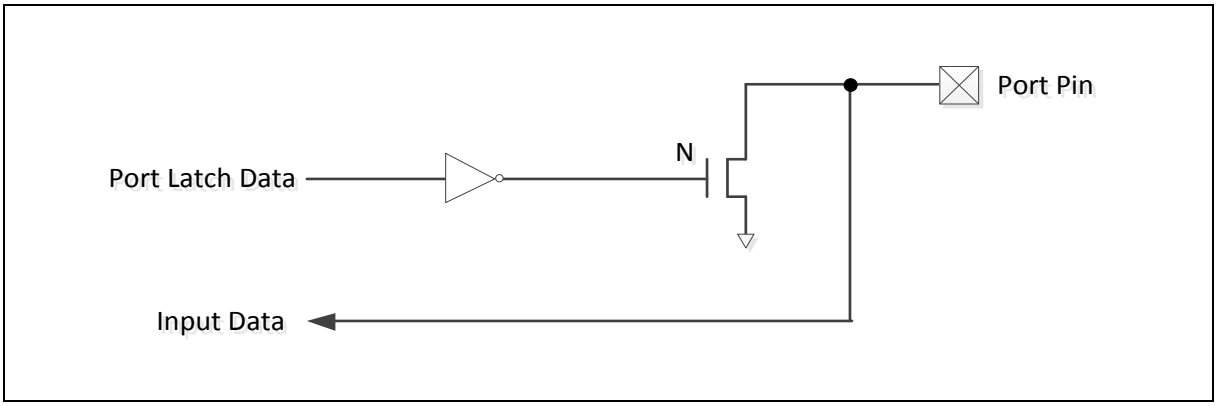


图 6.5-3 开漏输出

6.5.5.4 准双向模式

图 6.5-4展示了准双向模式的电路，设置MODEn(Px_MODE[2n+1:2n])为11，Px.n管脚为准双向模式，

I/O同时支持数字输出和输入功能，但拉电流能力仅达数百uA。要实现数字输入，需要先将DOUT(Px_DOUT[n])相应位置1。若DOUT(Px_DOUT[n])相应位bit[n]为‘0’，管脚上输出为“低”。若DOUT(Px_DOUT[n])相应位bit[n]为‘1’，该管脚将可以检测管脚值。若管脚值为高，没有任何动作，若管脚值为低，在该管脚上将有强输出驱动2个时钟周期的高电平，然后禁止强输出驱动， 其后管脚状态由内部上拉电阻控制。

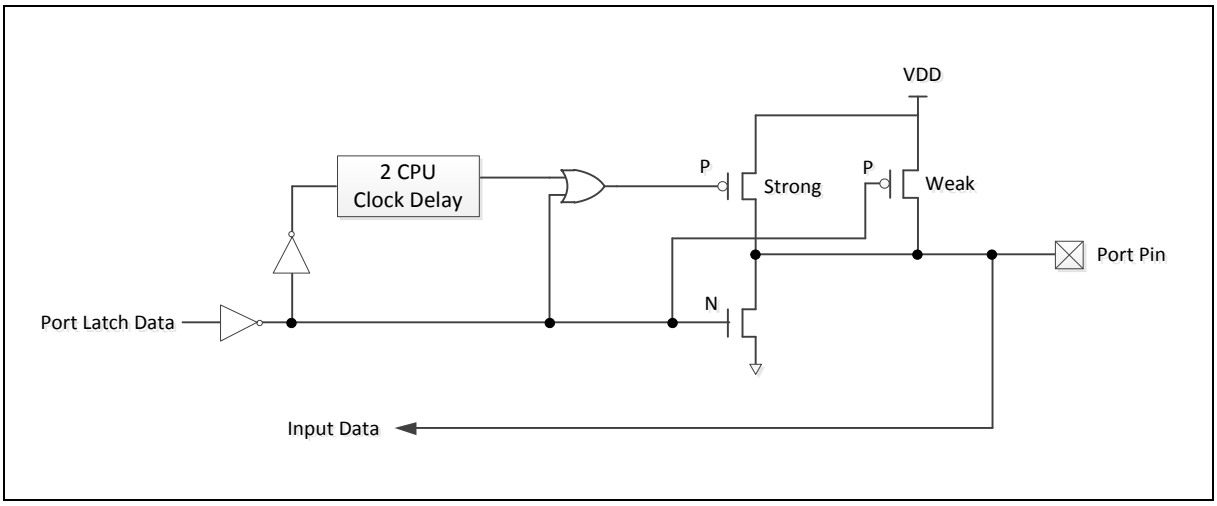


图 6.5-4 准双向 I/O 模式

6.5.5.5 GPIO中断及唤醒功能

每个GPIO管脚都可以通过RHIEN(Px_INTEN[n+16])/FLIEN(Px_INTEN[n])位和TYPE(Px_INTTYPE[n])设置成芯片的中断源。有五种中断触发条件可以设置：低电平触发、高电平触发、下降沿触发和上升沿触发以及上升与下降沿同时触发。当芯片进入空闲/掉电模式时，GPIO也可以唤醒系统。设置GPIO为唤醒触发的条件与GPIO中断触发的条件相同。

6.5.5.6 GPIO去抖动功能

GPIO去抖动功能可以用来采样GPIO管脚中断输入信号，防止噪声误触发中断。去抖动功能只有在边沿触发条件下生效，包括三种类型中断条件：下降沿触发，上升沿触发及双边沿触发，中断触发条件可以通过RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n])和TYPE(Px_INTTYPE[n])寄存器设定。客户使用去抖动功能前先要设置相应GPIO管脚的Px_DBEN寄存器使能去抖动功能。去抖动所用的时钟源可以是HCLK或LIRC(38.4kHz)，通过寄存器DBCLKSRC(Px_DBCTL[4])选择，DBCLKSEL(Px_DBCTL[3:0])用来设定去抖动采样周期。

图 6.5-5展示了GPIO上升沿触发中断，两次有效采样的时间间隔取决于DBCLKSRC(Px_DBCTL[4])和DBCLKSEL(Px_DBCTL[3:0])的值，采样两次产生一个有效数据。对于上升沿设定，如果在设定DBEN(Px_DBEN)前管脚状态为低，那么当管脚产生有效高电平数据时将触发中断；但是，如果设定DBEN(Px_DBEN)前管脚状态为高，那么管脚先要产生有效低电平然后再产生一个有效高电平才会产生中断。对于下降沿触发，图 6.5-6展示了跟上升沿相反的情况。

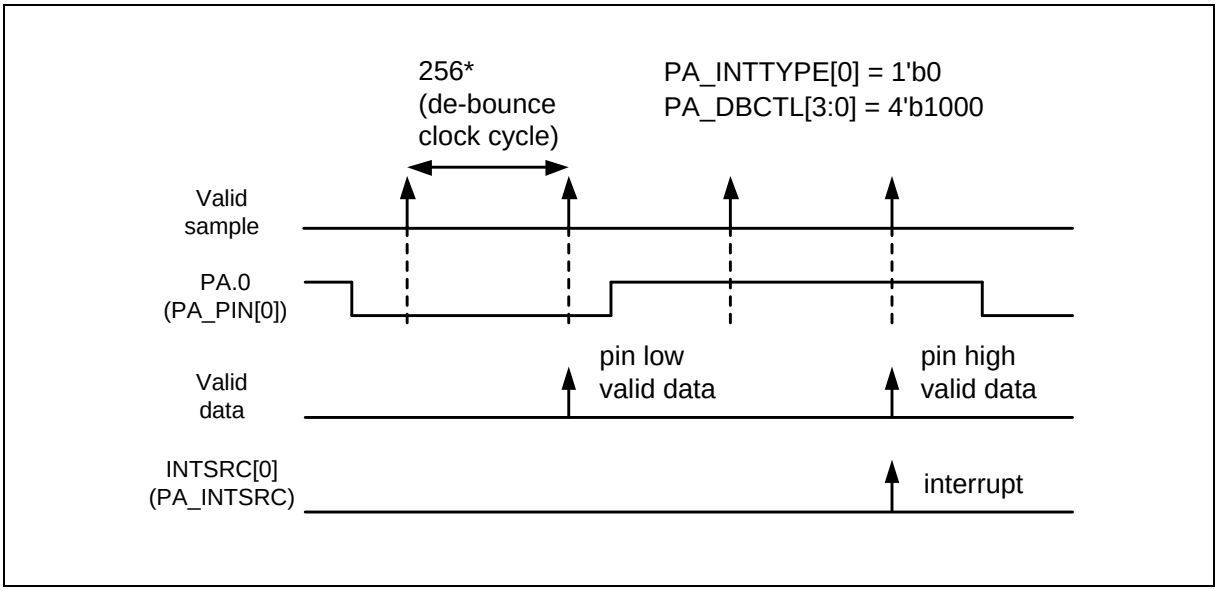


图 6.5-5 GPIO 上升沿触发中断

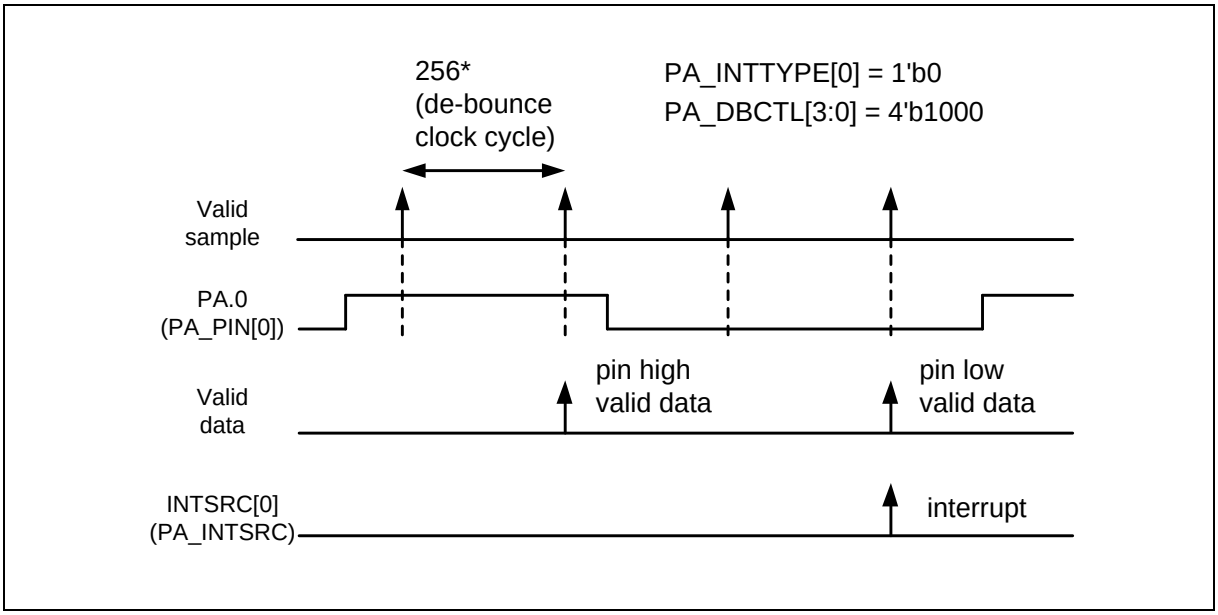


图 6.5-6 GPIO 下降沿触发中断

GPIO去抖动功能在掉电模式下也可以支持，表 6.5-1展示了不同系统状态下对去抖动的支持。通过设置DBENn(Px_DBEN[n]) 及 DBCLKSRC(Px_DBCTL[4])为1，使能掉电模式下去抖动功能，DBCLKSEL (Px_DBCTL[3:0])用来设置GPIO唤醒系统的去抖动时间

系统状态	DBEN	DBCLKSRC	描述
正常 /空闲 模式	0	0	没有去抖动功能
		1	没有去抖动功能
	1	0	去抖动时钟使用HCLK

		1	去抖动时钟使用 LIRC (38.4 kHz)
掉电模式	0	0	没有去抖动功能
		1	没有去抖动功能
	1	0	没有去抖动功能
		1	去抖动时钟使用 LIRC (38.4kHz)

表 6.5-1 去抖动功能设定表

6.5.5.7 GPIO数字输入通道关闭控制

通过设置DINOFF(Px_DINOFF[n+16])用户可以关闭GPIO数字输入通路，从而避免输入电流泄漏，当关闭GPIO数字输入通路，该管脚值PIN(PxPIN[n])为低。与此同时，当IO功能被设置成ADC/ACMP/ext.XTL时GPIO数字输入通路被硬件强制关闭而DIN OFF控制也无效。

6.5.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移量	R/W	描述	复位值
GPIO 基地址: **GPIO_BA = 0x4000_4000**				
PA_MODE	GPIO_BA+0x000	R/W	PA I/O 模式控制	0xXXXX_XXXX
PA_DINOFF	GPIO_BA+0x004	R/W	PA 数字输入通道关闭控制	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	PA 数据输出值	0x0000_FFFF
PA_DATMSK	GPIO_BA+0x00C	R/W	PA 输出数据写屏蔽	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	PA 管脚值	0x0000_XXXX
PA_DBEN	GPIO_BA+0x014	R/W	PA 去抖使能控制寄存器	0x0000_0000
PA_INTTYPE	GPIO_BA+0x018	R/W	PA中断触发类型控制	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	PA 中断使能控制寄存器	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	PA中断源标志	0x0000_XXXX
PB_MODE	GPIO_BA+0x040	R/W	PB I/O 模式控制	0xXXXX_XXXX
PB_DINOFF	GPIO_BA+0x044	R/W	PB 数字输入通道关闭控制	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	PB 数据输出值	0x0000_FFFF
PB_DATMSK	GPIO_BA+0x04C	R/W	PB 输出数据写屏蔽	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	PB 管脚值	0x0000_XXXX
PB_DBEN	GPIO_BA+0x054	R/W	PB 去抖使能控制寄存器	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB中断触发类型控制	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB 中断使能控制寄存器	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	PB中断源标志	0x0000_XXXX
PC_MODE	GPIO_BA+0x080	R/W	PC I/O 模式控制	0xXXXX_XXXX
PC_DINOFF	GPIO_BA+0x084	R/W	PC 数字输入通道关闭控制	0x0000_0000
PC_DOUT	GPIO_BA+0x088	R/W	PC 数据输出值	0x0000_7FFF
PC_DATMSK	GPIO_BA+0x08C	R/W	PC 输出数据写屏蔽	0x0000_0000
PC_PIN	GPIO_BA+0x090	R	PC 管脚值	0x0000_XXXX
PC_DBEN	GPIO_BA+0x094	R/W	PC 去抖使能控制寄存器	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC中断触发类型控制	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC 中断使能控制寄存器	0x0000_0000

PC_INTSRC	GPIO_BA+0x0A0	R/W	PC中断源标志	0x0000_XXXX
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O 模式控制	0xXXXX_XXXX
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD 数字输入通道关闭控制	0x0000_0000
PD_DOUT	GPIO_BA+0x0C8	R/W	PD 数据输出值	0x0000_FFFF
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD 输出数据写屏蔽	0x0000_0000
PD_PIN	GPIO_BA+0x0D0	R	PD 管脚值	0x0000_XXXX
PD_DBEN	GPIO_BA+0x0D4	R/W	PD 去抖使能控制寄存器	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD中断触发类型控制	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD 中断使能控制寄存器	0x0000_0000
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD中断源标志	0x0000_XXXX
PE_MODE	GPIO_BA+0x100	R/W	PE I/O模式控制	0xXXXX_XXXX
PE_DINOFF	GPIO_BA+0x104	R/W	PE数字输入通道关闭控制	0x0000_0000
PE_DOUT	GPIO_BA+0x108	R/W	PE数据输出值	0x0000_FFFF
PE_DATMSK	GPIO_BA+0x10C	R/W	PE输出数据写屏蔽	0x0000_0000
PE_PIN	GPIO_BA+0x110	R	PE管脚值	0x0000_XXXX
PE_DBEN	GPIO_BA+0x114	R/W	PE去抖使能控制寄存器	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE 中断触发类型控制	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE中断使能控制寄存器	0x0000_0000
PE_INTSRC	GPIO_BA+0x120	R/W	PE中断源标志	0x0000_XXXX
PF_MODE	GPIO_BA+0x140	R/W	PF I/O 模式控制	0xXXXX_XXXX
PF_DINOFF	GPIO_BA+0x144	R/W	PF 数字输入通道关闭控制	0x0000_0000
PF_DOUT	GPIO_BA+0x148	R/W	PF 数据输出值	0x0000_CFFF
PF_DATMSK	GPIO_BA+0x14C	R/W	PF 输出数据写屏蔽	0x0000_0000
PF_PIN	GPIO_BA+0x150	R	PF 管脚值	0x0000_XXXX
PF_DBEN	GPIO_BA+0x154	R/W	PF 去抖使能控制寄存器	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF中断触发类型控制	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF 中断使能控制寄存器	0x0000_0000
PF_INTSRC	GPIO_BA+0x160	R/W	PF 中断源标志	0x0000_XXXX
PG_MODE	GPIO_BA+0x180	R/W	PG I/O 模式控制	0xXXXX_XXXX
PG_DINOFF	GPIO_BA+0x184	R/W	PG数字输入通道关闭控制	0x0000_0000
PG_DOUT	GPIO_BA+0x188	R/W	PG数据输出值	0x0000_FE1C

PG_DATMSK	GPIO_BA+0x18C	R/W	PG输出数据写屏蔽	0x0000_0000
PG_PIN	GPIO_BA+0x190	R	PG管脚值	0x0000_XXXX
PG_DBEN	GPIO_BA+0x194	R/W	PG去抖使能控制寄存器	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG中断触发类型控制	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG中断使能控制寄存器	0x0000_0000
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG中断源标志	0x0000_XXXX
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O 模式控制	0xXXXX_XXXX
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH数字输入通道关闭控制	0x0000_0000
PH_DOUT	GPIO_BA+0x1C8	R/W	PH数据输出值	0x0000_0FF0
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH输出数据写屏蔽	0x0000_0000
PH_PIN	GPIO_BA+0x1D0	R	PH管脚值	0x0000_XXXX
PH_DBEN	GPIO_BA+0x1D4	R/W	PH去抖使能控制寄存器	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH中断触发类型控制	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH中断使能控制寄存器	0x0000_0000
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH中断源标志	0x0000_XXXX
GPIO_DBCTL	GPIO_BA+0x440	R/W	中断去抖控制寄存器	0x0000_0020
PAn_PDIO n=0,1..15	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n管脚数据输入/输出寄存器	0x0000_000X
PBn_PDIO n=0,1..15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n管脚数据输入/输出寄存器	0x0000_000X
PCn_PDIO n=0,1..14	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n管脚数据输入/输出寄存器	0x0000_000X
PDn_PDIO n=0,1..15	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n管脚数据输入/输出寄存器	0x0000_000X
PEn_PDIO n=0,1..15	GPIO_BA+0x900+(0x04 * n)	R/W	GPIO PE.n管脚数据输入/输出寄存器	0x0000_000X
PFn_PDIO n=0,1..11,14,15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n管脚数据输入/输出寄存器	0x0000_000X
PGn_PDIO n=2,3,4,9,10..15	GPIO_BA+0x980+(0x04 * n)	R/W	GPIO PG.n 管脚数据输入/输出寄存器	0x0000_000X
PHn_PDIO n=4,5..11	GPIO_BA+0x9C0+(0x04 * n)	R/W	GPIO PH.n 管脚数据输入/输出寄存器	0x0000_000X

6.5.7 寄存器描述

Port A-H I/O模式控制 (Px_MODE)

寄存器	偏移量	R/W	描述	复位值
PA_MODE	GPIO_BA+0x000	R/W	PA I/O 模式控制	0xXXXX_XXXX
PB_MODE	GPIO_BA+0x040	R/W	PB I/O 模式控制	0xXXXX_XXXX
PC_MODE	GPIO_BA+0x080	R/W	PC I/O 模式控制	0xXXXX_XXXX
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O 模式控制	0xXXXX_XXXX
PE_MODE	GPIO_BA+0x100	R/W	PE I/O 模式控制	0xXXXX_XXXX
PF_MODE	GPIO_BA+0x140	R/W	PF I/O 模式控制	0xXXXX_XXXX
PG_MODE	GPIO_BA+0x180	R/W	PG I/O 模式控制	0xXXXX_XXXX
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O 模式控制	0xXXXX_XXXX

31	30	29	28	27	26	25	24
MODE15		MODE14		MODE13		MODE12	
23	22	21	20	19	18	17	16
MODE11		MODE10		MODE9		MODE8	
15	14	13	12	11	10	9	8
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

位	描述	
[2n+1:2n] n=0,1..15	**MODEn**	**端口A-H I/O 管脚[n] 模式控制** 决定GPIO Px.n的I/O 类型 00 = Px.n 管脚为输入模式. 01 = Px.n 管脚为推挽输出模式. 10 = Px.n 管脚为开漏输出模式 11 = Px.n is 管脚为准双向模式 **注1：** 由CIOINI (CONFIG0 [10])可决定该寄存器的初始值，如果CIOINI置 0 芯片上电后所有的IO口都将是准双向模式，默认值为0xFFFFFFFF，如果CIOINI置 1，那么这个默认值就为0x0000_0000 ，芯片上电后所有管脚都为输入模式。 **注2：** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效。

Port A-H数字输入通道关闭控制(Px_DINOFF)

寄存器	偏移量	R/W	描述	复位值
PA_DINOFF	GPIO_BA+0x004	R/W	PA数字输入通道关闭控制	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB数字输入通道关闭控制	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC数字输入通道关闭控制	0x0000_0000
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD数字输入通道关闭控制	0x0000_0000
PE_DINOFF	GPIO_BA+0x104	R/W	PE数字输入通道关闭控制	0x0000_0000
PF_DINOFF	GPIO_BA+0x144	R/W	PF 数字输入通道关闭控制	0x0000_0000
PG_DINOFF	GPIO_BA+0x184	R/W	PG数字输入通道关闭控制	0x0000_0000
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH数字输入通道关闭控制	0x0000_0000

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

位	描述	
[n+16] n=0,1..15	**DINOFF[n]**	**端口A-H管脚[n] 数字通路关闭控制** 每一个位都控制相应Px.n管脚的数字通路是否关闭控。如果输入的是模拟信号，用户可以关闭Px.n数字输入通路以避免输入漏电流 0 = Px.n 数字通路启动. 1 = Px.n 数字通路关闭(输入值为低). **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效
[15:0]	**Reserved**	保留.

Port A-H 数据输出值(Px_DOUT)

寄存器	偏移量	R/W	描述	复位值
PA_DOUT	GPIO_BA+0x008	R/W	PA 数据输出值	0x0000_FFFF
PB_DOUT	GPIO_BA+0x048	R/W	PB 数据输出值	0x0000_FFFF
PC_DOUT	GPIO_BA+0x088	R/W	PC 数据输出值	0x0000_7FFF
PD_DOUT	GPIO_BA+0x0C8	R/W	PD 数据输出值	0x0000_FFFF
PE_DOUT	GPIO_BA+0x108	R/W	PE 数据输出值	0x0000_FFFF
PF_DOUT	GPIO_BA+0x148	R/W	PF 数据输出值	0x0000_CFFF
PG_DOUT	GPIO_BA+0x188	R/W	PG 数据输出值	0x0000_FE1C
PH_DOUT	GPIO_BA+0x1C8	R/W	PH 数据输出值	0x0000_0FF0

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

位	描述	
[31:16]	**Reserved**	保留.
[n] n=0,1..15	**DOUT[n]**	**端口 A-H 管脚[n] 输出值** 当Px.n被设置成推挽输出，开漏输出或准双向模式时，每一个位都对应控制Px.n管脚的状态 0 = Px.n被设置成推挽输出，开漏输出或准双向模式时，Px.n 为低 1 = Px.n被设置成推挽输出或准双向模式时，Px.n 为高 **注：** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15 引脚无效

Port A-H输出数据写屏蔽(Px_DATMSK)

寄存器	偏移量	R/W	描述	复位值
PA_DATMSK	GPIO_BA+0x00C	R/W	PA输出数据写屏蔽	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	PB 输出数据写屏蔽	0x0000_0000
PC_DATMSK	GPIO_BA+0x08C	R/W	PC输出数据写屏蔽	0x0000_0000
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD输出数据写屏蔽	0x0000_0000
PE_DATMSK	GPIO_BA+0x10C	R/W	PE输出数据写屏蔽	0x0000_0000
PF_DATMSK	GPIO_BA+0x14C	R/W	PF输出数据写屏蔽	0x0000_0000
PG_DATMSK	GPIO_BA+0x18C	R/W	PG输出数据写屏蔽	0x0000_0000
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH输出数据写屏蔽	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留.
[n] n=0,1..15	**DATMSK[n]**	**端口 A-H管脚[n] 数据输出写屏蔽** 用于保护相应寄存器DOUT (Px_DOUT[n]). 当设置DATMSK (Px_DATMSK[n])为 ‘1’时，相应DOUT (Px_DOUT[n])位被保护，写信号被屏蔽时，不能向保护位写数据。 0 = 相应的DOUT (Px_DOUT[n]) 位能更新 1 = 相应的DOUT (Px_DOUT[n]) 位被保护，不能对其写数据 **注1:** 该功能只保护相应的DOUT (Px_DOUT[n])位，不保护(PDIO (Pxn_PDIO[0]))相应位 **注2:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15 引脚无效

Port A-H 管脚值(Px_PIN)

寄存器	偏移量	R/W	描述	复位值
PA_PIN	GPIO_BA+0x010	R	PA 管脚值	0x0000_XXXX
PB_PIN	GPIO_BA+0x050	R	PB 管脚值	0x0000_XXXX
PC_PIN	GPIO_BA+0x090	R	PC 管脚值	0x0000_XXXX
PD_PIN	GPIO_BA+0x0D0	R	PD 管脚值	0x0000_XXXX
PE_PIN	GPIO_BA+0x110	R	PE 管脚值	0x0000_XXXX
PF_PIN	GPIO_BA+0x150	R	PF 管脚值	0x0000_XXXX
PG_PIN	GPIO_BA+0x190	R	PG 管脚值	0x0000_XXXX
PH_PIN	GPIO_BA+0x1D0	R	PH 管脚值	0x0000_XXXX

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

位	描述	
[31:16]	**Reserved**	保留.
[n] n=0,1..15	**PIN[n]**	**端口 A-H管脚[n] 管脚数值** 该寄存器中的 每一位表示Px.n管脚相应的实际状态。如果某位为 1，表示对应的管脚状态为高，否则，该状态为低。 **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15 管脚无效

Port A-H 抖使能控制寄存器 (Px_DBEN)

寄存器	偏移量	R/W	描述	复位值
PA_DBEN	GPIO_BA+0x014	R/W	PA去抖使能控制寄存器	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB 去抖使能控制寄存器	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC去抖使能控制寄存器	0x0000_0000
PD_DBEN	GPIO_BA+0x0D4	R/W	PD 去抖使能控制寄存器	0x0000_0000
PE_DBEN	GPIO_BA+0x114	R/W	PE去抖使能控制寄存器	0x0000_0000
PF_DBEN	GPIO_BA+0x154	R/W	PF去抖使能控制寄存器	0x0000_0000
PG_DBEN	GPIO_BA+0x194	R/W	PG去抖使能控制寄存器	0x0000_0000
PH_DBEN	GPIO_BA+0x1D4	R/W	PH去抖使能控制寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[n] n=0,1..15	**DBEN[n]**	**端口 A-H 管脚[n] 输入信号去抖动使能位** DBEN[n]用于使能相应位的去抖动功能。如果输入信号脉冲宽度不能被两个连续的去抖动采样周期所采样，则被视为信号反弹，从而不触发中断. 去抖动时钟源由DBCLKSRC(GPIO_DBCTL [4])控制，一个去抖动周期由DBCLKSEL(GPIO_DBCTL [3:0])控制。 0 = Px.n 去抖动功能禁止 1 = Px.n 去抖动功能使能 去抖动功能只有在边沿触发中断才有效，电平触发中断模式去抖动使能位无效 **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效。

Port A-H中断触发类型控制 (Px_INTTYPE)

Register	Offset	R/W	Description	Reset Value
PA_INTTYPE	GPIO_BA+0x018	R/W	PA 中断触发类型控制	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB 中断触发类型控制	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC 中断触发类型控制	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD中断触发类型控制	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE 中断触发类型控制	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF中断触发类型控制	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG 中断触发类型控制	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH 中断触发类型控制	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[n] n=0,1..15	**TYPE[n]**	**端口 A-F 管脚[n] 边沿或电平检测中断触发类型控制** TYPE (Px_INTTYPE[n]) 位用于控制电平触发或边沿触发中断。若为边沿触发中断，触发源可以有去抖动控制，如果是电平触发中断，输入源由一个HCLK时钟采样并产生中断。 0 = 边沿触发中断 1 = 电平触发中断 如果设置管脚为电平触发模式，则在寄存器RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n])中，只能设置一个电平(高电平或者低电平)；若设置了两个电平都触发中断，则设置被忽略，不会产生中断 去抖动功能对于边沿触发中断有效，对于电平触发中断无效 **注：** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15 引脚无效

Port A-H中断使能控制寄存器(Px_INTEN)

寄存器	偏移量	R/W	描述	复位值
PA_INTEN	GPIO_BA+0x01C	R/W	PA中断使能控制寄存器	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB中断使能控制寄存器	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC中断使能控制寄存器	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD中断使能控制寄存器	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE中断使能控制寄存器	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF中断使能控制寄存器	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG中断使能控制寄存器	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH中断使能控制寄存器	0x0000_0000

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

位	描述	
[n+16] n=0,1..15	**RHIEN[n]**	**端口 A-H管脚[n] 上升沿或高电平中断触发使能位** RHIEN (Px_INTEN[n+16])用于使能相应GPIO Px.n输入的中断。置‘1’也可以使能管脚唤醒功能 当设置RHIEN (Px_INTEN[n+16])位为‘1’: 如果中断是电平触发模式(TYPE (Px_INTTYPE[n]) = 1)，输入Px.n的状态为高电平时，产生中断。 如果中断是边沿触发模式(TYPE (Px_INTTYPE[n]) =0)，输入Px.n的状态由低电平到高电平变化时，产生中断。 0 = 禁用Px.n高电平或由低电平到高电平变化的中断 1 = 使能Px.n高电平或由低电平到高电平变化的中断 **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效
[n] n=0,1..15	**FLIEN[n]**	**端口 A-H 管脚[n] 下降沿或低电平中断触发使能位** FLIEN (Px_INTEN[n])用于使能相应GPIO Px.n]输入的中断。置 ‘1’ 也可以使能管脚唤醒功能 当设置FLIEN (Px_INTEN[n])位为 ‘1’: 如果中断是电平触发模式(TYPE (Px_INTTYPE[n]) = 1),输入Px.n的状态为低电平时，产生中断.

		如果中断是边沿触发模式(TYPE (Px_INTTYPE[n]) = 0)，输入Px.n的状态由高电平到低电平变化时，产生中断。 0 = 禁用Px.n低电平或由高电平到低电平变化的中断 1 =使能Px.n低电平或由高电平到低电平变化的中断 **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效

Port A-H中断源标志(Px_INTSRC)

寄存器	偏移量	R/W	描述	复位值
PA_INTSRC	GPIO_BA+0x020	R/W	PA中断源标志	0x0000_XXXX
PB_INTSRC	GPIO_BA+0x060	R/W	PB中断源标志	0x0000_XXXX
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC中断源标志	0x0000_XXXX
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD中断源标志	0x0000_XXXX
PE_INTSRC	GPIO_BA+0x120	R/W	PE中断源标志	0x0000_XXXX
PF_INTSRC	GPIO_BA+0x160	R/W	PF中断源标志	0x0000_XXXX
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG中断源标志	0x0000_XXXX
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH中断源标志	0x0000_XXXX

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

位	描述	
[31:16]	**Reserved**	保留.
[n] n=0,1..15	**INTSRC[n]**	**中断触发源标志** 写操作： 0 = 无动作. 1 = 清掉相应的未处理中断标志位 读操作： 0 = Px.n 没有中断 1 = Px.n 产生中断. **注:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15引脚无效

中断去抖控制寄存器(GPIO_DBCTL)

寄存器	偏移量	R/W	描述	复位值
GPIO_DBCTL	GPIO_BA+0x440	R/W	中断去抖控制寄存器·	0x0000_0020

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

位	描述	
[31:6]	**Reserved**	保留.
[5]	**ICLKON**	**中断时钟开启 模式** 0 = 边沿侦测电路仅在 I/O管脚对应的RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n])位置1有效 1 = 复位之后所有 I/O边沿侦测电路使能 **注意**：如果没有特定应用考虑，建议关掉时钟以减少耗电
[4]	**DBCLKSRC**	**去抖动时钟源选择** 0 = 去抖动时钟源为HCLK. 1 =去抖动时钟源为内部38.4K 时钟(LIRC)

位	描述	
[3:0]	**DBCLKSEL**	**去抖动采样周期选择** 0000 = 采样中断信号，每1个时钟一次 0001 =采样中断信号，每2个时钟一次 0010 =采样中断信号，每4个时钟一次 0011 =采样中断信号，每8个时钟一次 0100 =采样中断信号，每16个时钟一次 0101 =采样中断信号，每32个时钟一次 0110 =采样中断信号，每64个时钟一次 0111 =采样中断信号，每128个时钟一次 1000 =采样中断信号，每256个时钟一次 1001 =采样中断信号，每2*256个时钟一次 1010 =采样中断信号，每4*256个时钟一次 1011 =采样中断信号，每8*256个时钟一次 1100 =采样中断信号，每16*256个时钟一次 1101 =采样中断信号，每32*256个时钟一次 1110 =采样中断信号，每64*256个时钟一次 1111 =采样中断信号，每128*256个时钟一次

GPIO Px.n管脚数据输入输出寄存器(Pxn_PDIO)

寄存器	偏移量	R/W	描述	复位值
PAn_PDIO n=0,1..15	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n管脚数据输入/输出寄存器	0x0000_000X
PBn_PDIO n=0,1..15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n管脚数据输入/输出寄存器	0x0000_000X
PCn_PDIO n=0,1..14	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n管脚数据输入/输出寄存器	0x0000_000X
PDn_PDIO n=0,1..15	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n管脚数据输入/输出寄存器	0x0000_000X
PEn_PDIO n=0,1..15	GPIO_BA+0x900+(0x04 * n)	R/W	GPIO PE.n 管脚数据输入/输出寄存器	0x0000_000X
PFn_PDIO n=0,1..11,14,15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n 管脚数据输入/输出寄存器	0x0000_000X
PGn_PDIO n=2,3,4,9..15	GPIO_BA+0x980+(0x04 * n)	R/W	GPIO PG.n 管脚数据输入/输出寄存器	0x0000_000X
PHn_PDIO n=4,5..11	GPIO_BA+0x9C0+(0x04 * n)	R/W	GPIO PH.n 管脚数据输入/输出寄存器	0x0000_000X

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

位	描述	
[31:1]	**Reserved**	保留.
[0]	**PDIO**	**GPIO Px.N 管脚数值 输入/输出** 写该位可以控制一个GPIO管脚的输出值 0 = 设置相应GPIO管脚为低 1 = 设置相应GPIO管脚为高 读该寄存器得到GPIO管脚状态 例如：写PA0_PDIO即把值写到DOUT(Px_DOUT[0])位上，读PA0_PDIO即读取PIN(PA_PIN[0])的值。 注**1**：写操作不受DATMSK (Px_DATMSK[n])影响 **注2:** PC.15/PF.12~13/PG.0~1,5~8/PH.0~3,12~15 引脚无效

6.6 PDMA控制器(PDMA)

6.6.1 概述

直接存储器访问(PDMA)控制器用于高速数据传输。PDMA控制器可以从一个地址到另一个地址传输数据，无需CPU介入。这样做的好处是减少CPU的工作量，把节省下的CPU资源做其他应用。PDMA控制器包含9个通道，每个通道支持内存和外设之间的数据传输和内存与内存之间的数据传输。

6.6.2 特性

- 支持9个可独立配置的通道
- 支持两种优先级选择（固定优先级和轮循优先级）
- 支持8位，16位，32位数据传输
- 支持源和目标地址自动增加或者固定，数据宽度支持字节，半字，字
- 支持软件，I^2C，SPI/I^2S，UART，USCI，ADC，PWM和TIMER请求
- 支持Scatter-Gather模式，通过描述表链表执行灵活的数据传输
- 支持单笔和批量传输方式
- 仅通道0和通道1支持Time-out功能

6.6.3 框图

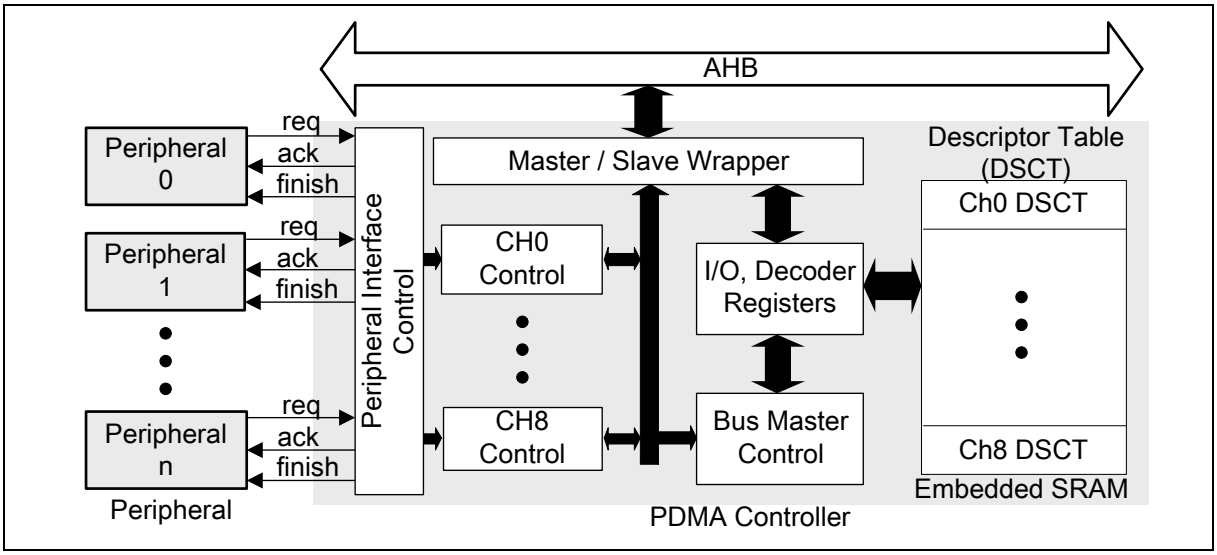


图 6.6-1 PDMA 控制器框图

6.6.4 基本配置

- 时钟源配置
  - 通过PDMACKEN(CLK_AHBCLK[1])使能PDMA的外设时钟
- 复位配置
  - 通过PDMARST (SYS_IPRST0[2])复位PDMA控制器

6.6.5 功能描述

PDMA可以从一个地址到另一个地址传输数据，无需CPU介入。PDMA有9个独立通道，同一时刻只有一个通道工作，因此，PDMA 控制器支持两种通道优先级：固定优先级和调度优先级(round-robin priority)，PDMA控制器通道执行的优先级是从高到低的。PDMA控制器支持两种运行模式：基本模式和Scatter-gather模式。基本模式用于按照一个传输描述表格传输数据。Scatter-gather模式对于每个PDMA都有多个传输描述表格，所以PDMA控制器通过这些表格，实现灵活的数据传输，传输描述表数据结构包含了传送信息，例如：传输源地址，传输目的地址，传输计数，批量传输数据大小，传输类型和操作模式。图 6.6-2是描述表(DSCT)数据结构。

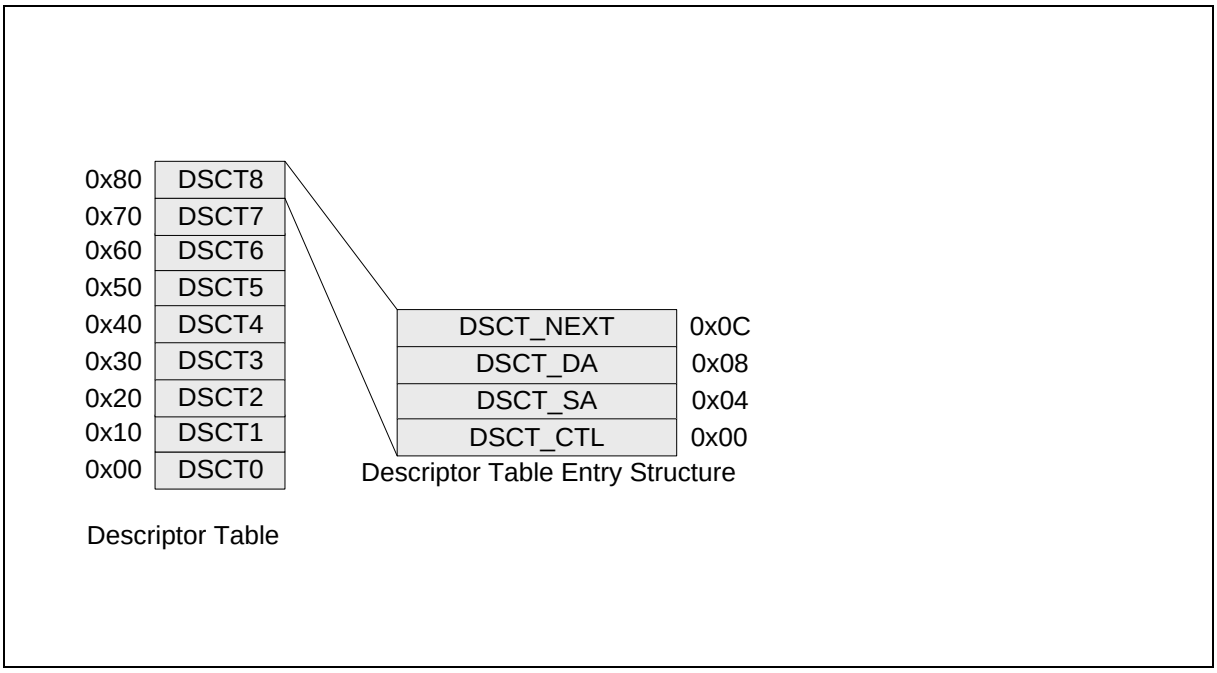


图 6.6-2 描述表数据结构

PDMA 控制器也支持单一和成组数据的传输类型，请求源可以是软件请求，接口请求，内存之间的数据传输是使用软件请求。单一传输的意思是软件或接口准备好传输一个数据(每个数据需要一次请求)，批量传输的意思是软件或接口将传输多个数据(多个数据仅需一次请求)。

6.6.5.1 通道优先级

PDMA控制器支持两级通道优先等级，包括固定优先级和轮循调度优先级(round-robin priority)。固定优先级比轮循调度优先级(round-robin priority)的优先级别更高。如果多路通道设定为固定优先级或调度优先级(round-robin priority)，高通道的优先级别也高。表 6.6-1是简要的优先级排序表。

PDMA_PRISET	通道数	优先级设定	优先级递减顺序
1	8	通道8, 固定优先级	最高
1	7	通道7, 固定优先级	---
---	---	---	---
1	0	通道0, 固定优先级	---

0	8	通道8, 轮循优先级	---
0	7	通道8, 轮循优先级	---
---	---	---	---
0	0	通道0, 轮循优先级	最低

表 6.6-1 通道优先级表

6.6.5.2 PDMA操作模式

PDMA 控制器支持两种操作模式，基本模式和Scatter-Gather模式

基本模式

基本模式用于执行一个描述表格的传输模式。该模式用于内存与内存之间或接口与内存之间或接口与接口之间的数据传输。如果客户想在接口与接口之间传输数据，必须确保请求传输的接口知道对方的数据是否准备好了。PDMA控制器操作模式可以通过寄存器OPMODE(PDMA_DSCTn_CTL[1:0]，n代表PDMA通道数)设定，默认设置是在空闲状态(OPMODE(PDMA_DSCTn_CTL[1:0]) = 0x00)，建议用户在空闲状态下设定描述表。如果操作不是在空闲状态，用户重新配置通道设定可能会引起操作错误。

用户必须填写传输计数寄存器TXCNT(PDMA_DSCTn_CTL[29:16])和传输宽度选择寄存器TXWIDTH(PDMA_DSCTn_CTL[13:12])，目的地址递增大小寄存器DAINC(PDMA_DSCTn_CTL[11:10])，源地址递增大小寄存器SAINC(PDMA_DSCTn_CTL[9:8])，批量大小寄存器BURSIZE(PDMA_DSCTn_CTL[6:4])和传输类型寄存器TXTYPE(PDMA_DSCTn_CTL[2])，那么PDMA控制器将在接收到请求信号后执行传输操作。如果相应的PDMA 中断位INTENn(PDMA_INTEN[11:0])使能，传输完成后将给CPU产生一个中断，操作模式将被自动更新为空闲模式，如图 6.6-3所示。如果软件配置操作模式为空闲状态，PDMA控制器不会执行任何传输，并清除这个操作请求。如果相应的PDMA中断位被使能，完成这个任务后也会给CPU产生中断。

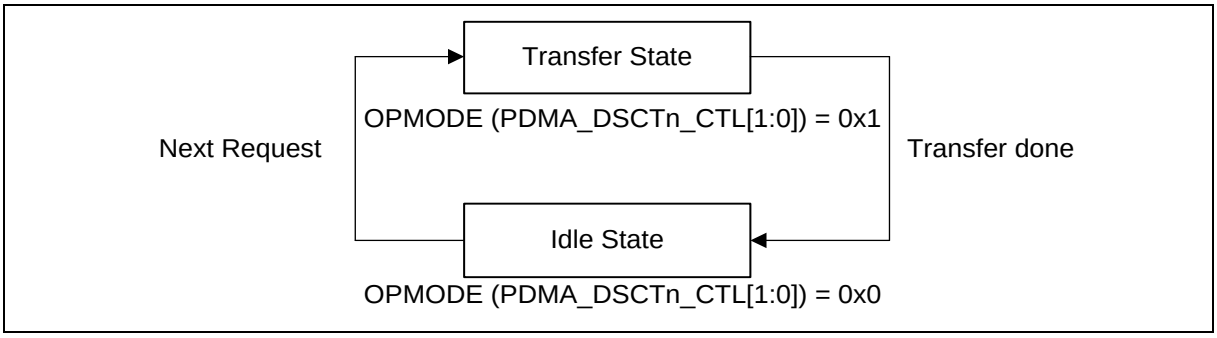


图 6.6-3 基本模式有限状态机

Scatter-Gather 模式

Scatter-Gather模式是一个综合模式，通过如表 6.6-4描述表链表设定，可以实现灵活的数据传输。通过该模式，用户可以实现外设的回环访问，用于多路PDMA任务或系统内存不连续地址的数据搬移。Scatter-Gather模式只需要一次请求就可以完成所有链表所描述的任务，一直到最后一个OPMODE (PDMA_DSCTn_CTL[1:0])为idle状态的链表，这也意味着Scatter-Gather模式可以用来在存储器间传送数据而不需要握手信号。

在Scatter-Gather模式，第一个表用于跳转到下一个表的入口。第一个任务不会做数据搬移操作。如果相应的PDMA 中断位使能，寄存器TBINTDIS (PDMA_DSCTn_CTL[7])=0，完成每个任务后，将给CPU产生一个中断，(任务完成，TBINTDIS位为“0”，会插入相应的寄存器TDIFn (PDMA_TDSTS[11:0])标志，TBINTDIS位为“1”，禁用TDIFn)。

如果触发了通道**8**，在Scatter-Gather(OPMODE (PDMA_DSCTn_CTL[1:0]) =0x2)模式，硬件将把寄存器PDMA_DSCTn_NEXT(连接偏移地址)和PDMA_SCATBA (基地址)相加来获取真正的PDMA任务信息。例如，基地址是0x2000_0000(PDMA_SCATBA寄存器中仅MSB 16位有效)，当前链接偏移地址是0x0000_0100(在寄存器PDMA_DSCTn_NEXT中，仅LSB 16位有效)， 那么，下一个DSCT的入口开始地址是0x2000_0100。

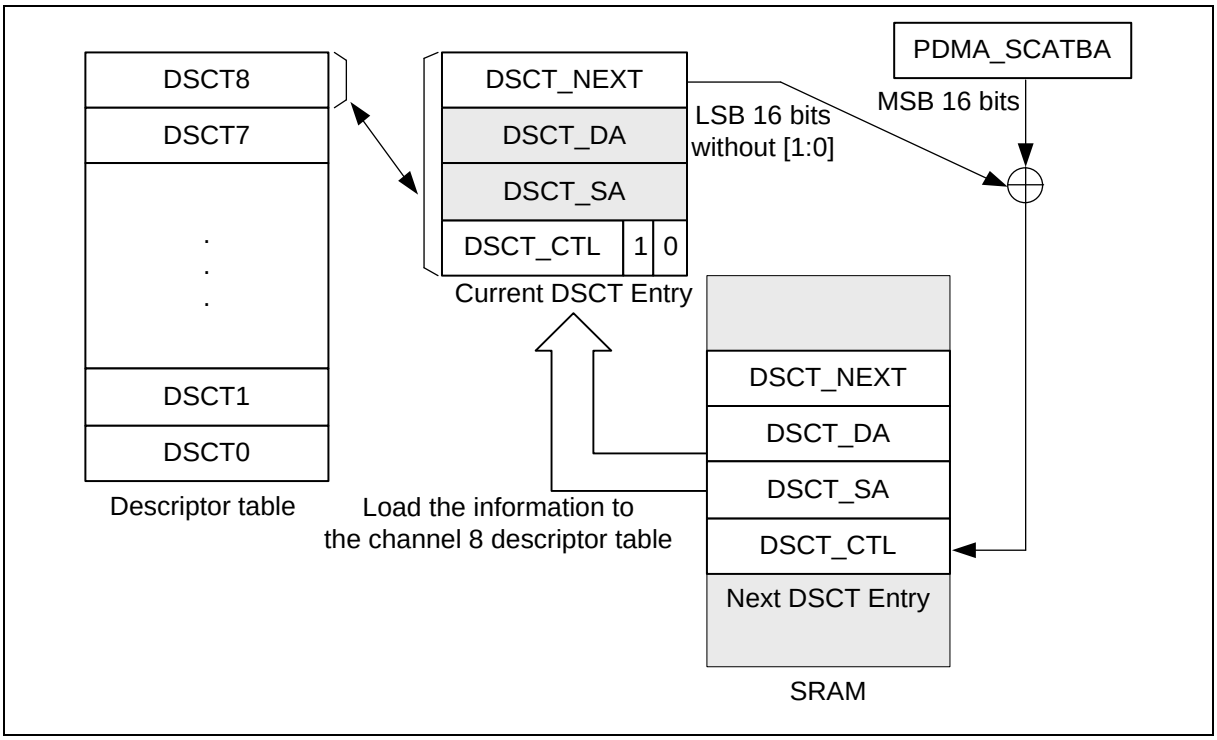


表 6.6-4 描述表链表结构

上述链接列表操作是Scatter-Gather模式的DSCT状态如表 6.6-5。当加载信息结束后，将自动进入搬移数据状态。然而，如果下一个PDMA信息也是Scatter-Gather模式，当前任务结束后，硬件将抓下一个PDMA块信息。Scatter-Gather模式只有在PDMA控制器操作模式切换到基本模式并完成最后一次传输或者直接切换到空闲状态后才会结束。

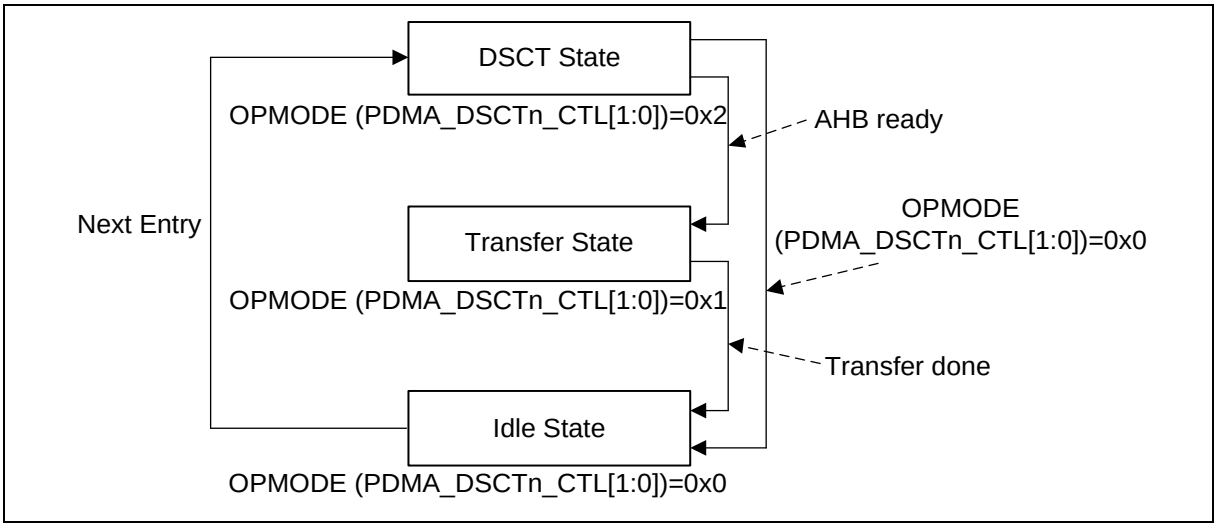


表 6.6-5 Scatter-Gather 模式有限状态机

6.6.5.3 传输类型

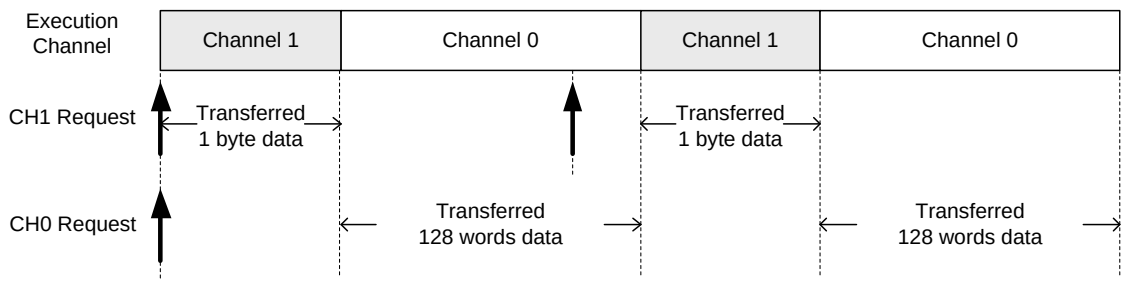
PDMA 控制器支持两种传输模式：单一传输和批量传输模式，通过寄存器TXTYPE(PDMA_DSCTn_CTL[2])设定。

当PDMA控制器运行在单一传输模式，每搬移一个数据需要一次请求，当搬移一次数据，寄存器TXCNT(PDMA_DSCTn_CTL[31:16])会减1，直到寄存器TXCNT(PDMA_DSCTn_CTL[31:16])为0，搬移才会完成。在该模式，寄存器BURSIZE(PDMA_DSCTn_CTL[6:4])不用于控制搬移数据量大小，它的值固定为1。

在批量搬移模式，PDMA控制器搬移TXCNT(PDMA_DSCTn_CTL[31:16])个数据，仅需一次请求。当搬移BURSIZE(PDMA_DSCTn_CTL[6:4])个数据后，寄存器TXCNT(PDMA_DSCTn_CTL[31:16])中的数目会减去BURSIZE。直到TXCNT(PDMA_DSCTn_CTL[31:16])递减为0，搬移数据才完成。该模式仅用于PDMA控制器做内存到内存的数据搬移，如果要做内存到外设或者外设到内存的数据搬移必须用单一模式。

图 6.6-6 是基本传输模式下的单一传输和批量传输举例。在这个例子，通道1使用单一传输模式，TXCNT(PDMA_DSCTn_CTL[31:16])=127，通道0使用批量传输模式，BURSIZE(PDMA_DSCTn_CTL[6:4])=128次传输）并且TXCNT(PDMA_DSCTn_CTL[29:16])=255，操作顺序如下：

1. 通道0与通道1 同时接收到触发信号。
2. 默认通道1优先级高于通道0；PDMA将先装载通道1的描述表并执行，但通道1为单一模式数据大小是1个字节所以PDMA控制器只传输一个字节的数据。
3. 然后，PDMA控制器将切换到通道0，并加载通道0的描述表，因为通道0是批量传输模式传输次数是128，数据宽度是1个字，所以PDMA控制器将传输128个字数据。
4. 当通道0传输完128个字的数据后，通道1接收到另一个传输请求，然后通道0完成传输后PDMA控制器就切换到通道1上并传输下一个1字节的数据。
5. 当通道1传输完第二个1字节的数据后 PDMA控制器又切换到通道0去传输下一个128个字的数据，如果通道1没有再有传素请求，那么PDMA控制器将继续传输通道0的下一个128个字的数据。
6. 当通道0收到2次数据请求完成256个字的数据传输，通道1收到128次数据请求完成128个字节的数据传输，PDMA控制器就完成了传输。



	CH1	CH0
TXCNT (PDMA_DSCTn_CTL[31:16])	127	255
TXWIDTH (PDMA_DSCTn_CTL[13:12])	0 (1 byte)	2 (1 word)
BURSIZE (PDMA_DSCTn_CTL[6:4])	Non-useful	0 (128 transfers)
TXTYPE (PDMA_DSCTn_CTL[2])	1 (Single Transfer)	0 (Burst Transfer)

图 6.6-6 基本模式下单次传输跟批量传输示意图

6.6.5.4 通道 Time-out

只有通道0和通道1支持Time-Out功能。当通道被选择传输到指定的外设中并使能，相应的通道Time-Out TOUTENn (PDMA_TOUTEN[n], n=0,1)也被使能，那么，在通道从外设收到触发信号后，通道的相应Time-Out向上计数器就开始从0开始计数。

Time-Out 计数器的时钟由 HCLK 分频获得，通过设置相应通道的TOUTPSCn(PDMA_TOUTPSC[2+4n:4n], n=0,1) 寄存器。当通道相应的TOUTIENn(PDMA_TOUTIEN[n], n=0,1)寄存器被使能，如果Time-Out计数器从0向上计数到寄存器TOUTPSCn(PDMA_TOUTPSC[2+4n:4n], n=0,1)中设定的值，PDMA控制器将产生一个中断信号。当Time-Out发生时，相应通道的REQTOFn(PDMA_INTSTS[n+8], n=0,1) 寄存器将置位，用以表明通道发生了Time-Out事件。

在计数器计数到TOCn (PDMA_TOC0_1[16(n+1)-1):16n]寄存器中设定的值或者收到触发信号，或者芯片进入到掉电模式，Time-Out计数器就复位到0。

图 6.6-7是一个关于Time-out计数器操作的例子，操作顺序如下描述：

1. 通过设置TOUTEN0(PDMA_TOUTEN[0])为1，使能通道0的time-out功能，但此时time-out计数器还没有开始计数
2. 当收到外设第一个请求后，time-out 计数器开始从0向上计数到TOC0(PDMA_TOC0_1[15:0]) 中设定的值
3. 当收到外设第二个请求后，time-out 计数器复位到0
4. 当time-out计数器计数到5时，通道0的time-out标志(REQTOF0(PDMA_INTSTS[8]))将会被置高。通道0计数器将继续从0到5循环计数。用户可以清掉REQTOF0标志，然后轮循REQTOF0标志来检查下一个time-out是否发生
5. 当time-out功能被禁用时，time-out计数器复位到0。

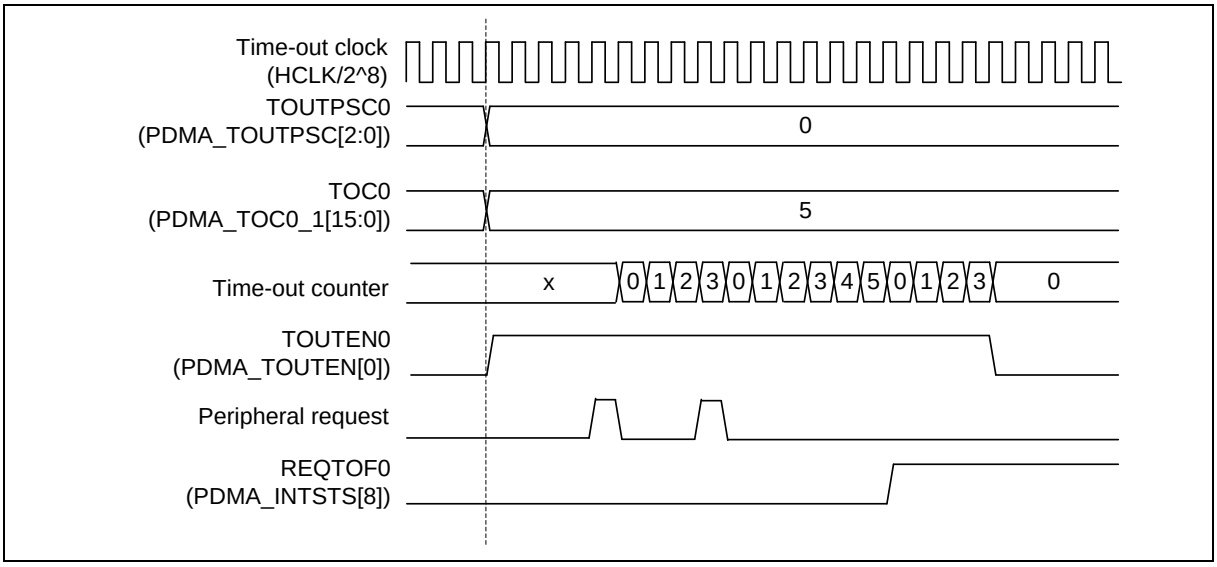


图 6.6-7 PDMA 通道 0 时间溢出计数操作示意图

6.6.6 寄存器映射

R:只读, **W:** 只写, **R/W:** 可读写

寄存器	偏移地址	R/W	模式	复位值
PDMA基地址: **PDMA_BA = 0x4000_8000**				
PDMA_DSCTn_CTL **n = 0,1..8**	PDMA_BA+0x10*n	R/W	PDMA 通道 n描述表控制寄存器	0xXXXX_XXXX
PDMA_DSCTn_SA **n = 0,1..8**	PDMA_BA+0x0004+0x10*n	R/W	PDMA通道n源地址寄存器	0xXXXX_XXXX
PDMA_DSCTn_DA **n = 0,1..8**	PDMA_BA+0x0008+0x10*n	R/W	PDMA通道n目的地址寄存器	0xXXXX_XXXX
PDMA_DSCTn_NEXT **n = 0,1..8**	PDMA_BA+0x000c+0x10*n	R/W	PDMA通道n下一个Scatter-Gather 描述表偏移地址寄存器	0xXXXX_XXXX
PDMA_CURSCATn **n = 0,1..8**	PDMA_BA+0x0100+0x004*n	R	PDMA通道n当前Scatter-Gather 描述表偏移地址寄存器	0xXXXX_XXXX
PDMA_CHCTL	PDMA_BA + 0x400	R/W	PDMA 通道控制寄存器	0x0000_0000
PDMA_PAUSE	PDMA_BA + 0x404	W	PDMA 传输暂停控制寄存器	0x0000_0000
PDMA_SWREQ	PDMA_BA + 0x408	W	PDMA 软件请求寄存器	0x0000_0000
PDMA_TRGSTS	PDMA_BA + 0x40C	R	PDMA 通道请求状态寄存器	0x0000_0000
PDMA_PRISET	PDMA_BA + 0x410	R/W	PDMA 固定优先级设置寄存器	0x0000_0000
PDMA_PRICLR	PDMA_BA + 0x414	W	PDMA 固定优先级清除寄存器	0x0000_0000
PDMA_INTEN	PDMA_BA + 0x418	R/W	PDMA 中断使能寄存器	0x0000_0000
PDMA_INTSTS	PDMA_BA + 0x41C	R/W	PDMA 中断状态寄存器	0x0000_0000
PDMA_ABTSTS	PDMA_BA + 0x420	R/W	PDMA 通道读写目标终止标志寄存器	0x0000_0000
PDMA_TDSTS	PDMA_BA + 0x424	R/W	PDMA 通道传输完成标志寄存器	0x0000_0000
PDMA_ALIGN	PDMA_BA + 0x428	R/W	PDMA 传输对齐状态寄存器	0x0000_0000
PDMA_TACTSTS	PDMA_BA + 0x42C	R	PDMA 传输有效标志寄存器	0x0000_0000
PDMA_TOUTPSC	PDMA_BA + 0x430	R/W	PDMA Time-out 预分频寄存器	0x0000_0000
PDMA_TOUTEN	PDMA_BA + 0x434	R/W	PDMA Time-out 使能寄存器	0x0000_0000
PDMA_TOUTIEN	PDMA_BA + 0x438	R/W	PDMA Time-out 中断使能寄存器	0x0000_0000
PDMA_SCATBA	PDMA_BA + 0x43C	R/W	PDMA Scatter-Gather描述表基地址寄存器	0x2000_0000
PDMA_TOC0_1	PDMA_BA + 0x440	R/W	PDMA 通道0和通道1的Time-Out计数器寄存器	0xFFFF_FFFF
PDMA_CHRST	PDMA_BA + 0x460	R/W	PDMA 通道复位控制寄存器	0x0000_0000
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA请求源选择寄存器0	0x0000_0000

PDMA_REQSEL4_7	PDMA_BA + 0x484	R/W	PDMA请求源选择寄存器1	0x0000_0000
PDMA_REQSEL8	PDMA_BA + 0x488	R/W	PDMA请求源选择寄存器2	0x0000_0000

6.6.7 寄存器描述

描述表控制寄存器(PDMA_DSCTn_CTL)

寄存器	偏移地址	R/W	描述	复位值
PDMA_DSCTn_CTL	PDMA_BA+0x10*n	R/W	通道 n描述表控制寄存器	0xXXXX_XXXX

31	30	29	28	27	26	25	24
TXCNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**TXWIDTH**		**DAINC**		**SAINC**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TBINTDIS	**BURSIZE**			**Reserved**	**TXTYPE**	**OPMODE**	

位	描述	
[31:16]	**TXCNT**	**传输计数** TXCNT代表PDMA传输的请求数，实际的传输数是(TXCNT + 1)最大的传输计数是16384，依据TXWIDTH寄存器设定，每次传输数据单位可以是字节，半-字，或字。 **注:** 当PDMA完成每次数据传输，该寄存器计数会立刻递减。
[14]	**Reserved**	保留.
[13:12]	**TXWIDTH**	**传输宽度选择** 该寄存器用于数据传输宽度设定 00 = 每次操作数据传输宽度是一个字节(8位) 01 = 每次操作数据传输宽度是半-字(16位) 10 = 每次操作数据传输宽度是字(32位) 11 = 保留 注意：PDMA传输源地址(PDMA_DSCT_SA)和PDMA传输目的地址(PDMA_DSCT_DA)应在寄存器TXWIDTH设定中对齐
[11:10]	**DAINC**	**目的地址增加** 该寄存器用于设定目的地址的增量大小 11 = 无增量(固定地址) 其它= 增量的大小依据寄存器TXWIDTH设定
[9:8]	**SAINC**	**源地址增加** 该寄存器用于设定源地址增量的大小 11 = 无增量(固定地址).

位	描述	
		其它 = 增量的大小依据寄存器TXWIDTH设定
[7]	**TBINTDIS**	**禁用表中断** 该寄存器是用于是否使能表中断。当PDMA完成传输任务，如果TBINTDIS位使能，将不会产生中断。 0 = 表中断使能 1 = 禁用表中断 注意: 如果该位是'1'，TEMPTYF 不被设置
[6:4]	**BURSIZE**	**批量大小** 该寄存器用于外设定义的批量传输数据大小 000 = 128 个数据传输 001 = 64 个数据传输 010 = 32 个数据传输 011 = 16 个数据传输 100 = 8 个数据传输 101 = 4 个数据传输 110 = 2 个数据传输 111 = 1 个数据传输 注意：这个寄存器只用于批量传输模式
[3]	**Reserved**	保留.
[2]	**TXTYPE**	**传输类型** 0 = 批量传输类型 1 = 单一传输类型
[1:0]	**OPMODE**	**PDMA 传输类型选择** 00 = 空闲状态：通道传输中断或传输结束，当PDMA完成通道传输，OPMODE会自动清零为空闲状态。 01 = 基本模式：描述表格只有一个任务。当这个任务完成，寄存器PDMA_INTSTS[n]将被插入。 10 = Scatter-Gather 模式：在该模式，用户会把下一个描述表格地址输入寄存器PDMA_DSCT_NEXT，PDMA控制机将忽略这次任务，然后加载执行下一个任务。 11 = 保留 注意: 在填写传输任务到描述表之前，用户必须要检查PDMA_INTSTS[1]确认当前任务是否完成。

源地址开始寄存器(PDMA_DSCTn_SA)

寄存器	偏移地址	R/W	描述	复位值
PDMA_DSCTn_SA	PDMA_BA+0x0004+0x10*n	R/W	PDMA通道n源地址寄存器	0xXXXX_XXXX

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

位	描述	
[31:0]	**SA**	**PDMA 传输源地址寄存器** 该寄存器是32-位PDMA的源地址寄存器

目的地址寄存器(PDMA_DSCTn_DA)

寄存器	偏移地址	R/W	描述	复位值
PDMA_DSCTn_DA	PDMA_BA+0x0008+0x10*n	R/W	PDMA通道n目的地址寄存器	0xXXXX_XXXX

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

位	描述	
[31:0]	**DA**	**PDMA 传输 目标地址寄存器** 该寄存器存放PDMA控制器32位目标地址

下一个Scatter-gather描述表偏移地址(PDMA_DSCTn_NEXT)

寄存器	偏移地址	R/W	描述	复位值
PDMA_DSCTn_NEXT	PDMA_BA+0x000c+0x10*n	R/W	PDMA通道n下一个Scatter-Gather 描述表偏移地址寄存器	0xXXXX_XXXX

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

位	描述	
[31:16]	**EXENEXT**	**PDMA执行下一个描述表偏移** 此处是下一个描述表地址相对于当前执行描述表在系统内存中的偏移量 **注:** 此处写入无效
[15:0]	**NEXT**	**PDMA 下一个描述表偏移** 此处表示下一个描述表地址在系统内存中的偏移 **写操作:** 如果系统内存基地址是0x2000_0000 (PDMA_SCATBA)，下一个描述表地址是0x2000_0100，那么此处必须写0x0100 **读操作:** 如果在scatter-gather模式下操作，最后两个位NEXT[1:0]将变为保留为，表示内存中下一个描述表地址 **注1:** 描述表地址必须字对齐 **注2:** 在填写传输任务到描述表前，用户必须检查描述表的完整性

当前Scatter-gather描述表地址(PDMA_CURSCATn)

寄存器	偏移地址	R/W	描述	复位值
PDMA_CURSCATn	PDMA_BA+0x0100+0x004*n	R	PDMA通道n当前Scatter-Gather 描述表偏移地址寄存器	0xXXXX_XXXX

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

位	描述	
[31:0]	**CURADDR**	**PDMA 当前描述表地址寄存器（只读）** 该寄存器是PDMA控制器的当前32-位描述表地址。 **注**: 该寄存器只读，仅用于Scatter-Gather模式来表示当前描述表地址。

通道控制寄存器(PDMA_CHCTL)

寄存器	偏移地址	R/W	描述	复位值
PDMA_CHCTL	PDMA_BA + 0x400	R/W	PDMA 通道控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**CHEN8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHEN7	**CHEN6**	**CHEN5**	**CHEN4**	**CHEN3**	**CHEN2**	**CHEN1**	**CHEN0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**CHENn**	**PDMA 通道使能位** 该位为1是使能PDMA通道n，如果不使能该位，该通道无法工作。 0 = 禁用 PDMA通道 [n] 1 = 使能 PDMA通道 [n] 注：设置PDMA_PAUSE或PDMA_RESET寄存器将清除该位

PDMA 传输暂停控制(PDMA_PAUSE)

寄存器	偏移地址	R/W	描述	复位值
PDMA_PAUSE	PDMA_BA + 0x404	W	PDMA传输暂停控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**PAUSE8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PAUSE7	**PAUSE6**	**PAUSE5**	**PAUSE4**	**PAUSE3**	**PAUSE2**	**PAUSE1**	**PAUSE0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**PAUSEn**	**PDMA 通道N 传输暂停控制寄存器（只写）** 用户可以暂停PDMA传输，通过设置PAUSEn 位。当用户置PDMA_PAUSE位，将暂停通道N 正在进行的传输，然后清除通道使能位（PDMA_CHCTL [n],n=0,1,…8)，清掉传输请求有效标志（PDMA_TRGSTS [n:0],n=0,1,…8)。如果重新启动暂停的通道，那么将从上次保留的进度继续传输。 0 = 无效. 1 =暂停PDMA 通道N传输.

PDMA 软件请求寄存器(PDMA_SWREQ)

寄存器	偏移地址	R/W	描述	复位值
PDMA_SWREQ	PDMA_BA + 0x408	W	PDMA 软件请求寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**SWREQ8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWREQ7	**SWREQ6**	**SWREQ5**	**SWREQ4**	**SWREQ3**	**SWREQ2**	**SWREQ1**	**SWREQ0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**SWREQn**	**PDMA软件请求寄存器（只写）** 置1该位，产生一个软件请求给PDMA通道n 0 = 无效. 1 = 产生一个软件请求. **注1:** 用户可以通过读寄存器PDMA_TRGSTS来知道哪个通道被使能，使能标志可通过软件请求或外设请求触发。 **注2:** 如果用户没有使能相关的PDMA通道，软件请求会被忽略。

PDMA 通道请求状态寄存器(PDMA_TRGSTS)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TRGSTS	PDMA_BA + 0x40C	R	PDMA 通道请求状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**REQSTS8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSTS7	**REQSTS6**	**REQSTS5**	**REQSTS4**	**REQSTS3**	**REQSTS2**	**REQSTS1**	**REQSTS0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**REQSTSn**	**PDMA 通道N 请求状态寄存器（只读）** 这个标志用来表示通道n是否有软件或外设请求。当PDMA控制器完成通道传输，该位会被自动清除。 0 = PDMA 通道n无请求 1 = PDMA 通道n有请求. **注:** 如果用户通过他们各自的寄存器PDMA_PAUSE 或PDMA_RESET来暂停或者复位PDMA传输，那么该位将在当前的传输完成之后自动清掉。

PDMA 固定优先级设置寄存器 (PDMA_PRISET)

寄存器	偏移地址	R/W	描述	复位值
PDMA_PRISET	PDMA_BA + 0x410	R/W	PDMA 固定优先级设置寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**FPRISET8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPRISET7	**FPRISET6**	**FPRISET5**	**FPRISET4**	**FPRISET3**	**FPRISET2**	**FPRISET1**	**FPRISET0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**FPRISETn**	**PDMA 固定优先级设定寄存器** 设置该位为"1" 来使能固定优先级的等级。 **写操作:** 0 = 无效. 1 = 设定PDMA通道n为固定优先级通道. **读操作:** 0 = PDMA是轮循调度(round-robin)优先级 1 = PDMA是固定优先级 注:该位只设定为固定优先级，通过可以设定寄存器PDMA_PRICLR来清除固定优先级

PDMA 固定优先级清除寄存器(PDMA_PRICLR)

寄存器	偏移地址	R/W	描述	复位值
PDMA_PRICLR	PDMA_BA + 0x414	W	PDMA 固定优先级清除寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**FPRICLR8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPRICLR7	**FPRICLR6**	**FPRICLR5**	**FPRICLR4**	**FPRICLR3**	**FPRICLR2**	**FPRICLR1**	**FPRICLR0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**FPRICLRn**	**PDMA固定优先级清除寄存器（只写）** 设置该位为“1”，清除固定优先级等级。 0 = 无效 1 = 清除PDMA通道N固定优先级设定 **注:** 用户可以读PDMA_PRISET 寄存器来获取通道的优先级

PDMA中断使能寄存器(PDMA_INTEN)

寄存器	偏移地址	R/W	描述	复位值
PDMA_INTEN	PDMA_BA + 0x418	R/W	PDMA中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**INTEN8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN7	**INTEN6**	**INTEN5**	**INTEN4**	**INTEN3**	**INTEN2**	**INTEN1**	**INTEN0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**INTENn**	**PDMA中断使能寄存器** 该寄存器用来使能PDMA通道n的中断 0 = PDMA 通道N中断禁止 1 = PDMA 通道N中断使能 注：中断标志包括time-out，终止，传输完成及对齐。

PDMA中断状态寄存器(PDMA_INTSTS)

寄存器	偏移地址	R/W	描述	复位值
PDMA_INTSTS	PDMA_BA + 0x41C	R/W	PDMA 中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**REQTOF1**	**REQTOF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**ALIGNF**	**TDIF**	**ABTIF**

位	描述	
[31:10]	**Reserved**	保留.
[9]	**REQTOF1**	**PDMA通道1请求Time-out标志** 该标志表示PDMA控制器等待外设发送请求已经超过了一个设定的时间，该时间由PDMA_TOC1 寄存器设定。用户可以写“1”到该位清除这个标志。 0 = 请求没有time-out 1 = 外设请求time-out
[8]	**REQTOF0**	**PDMA通道0请求Time-out标志** 该标志表示PDMA控制器等待外设发送请求已经超过了一个设定的时间，该时间由PDMA_TOC0 寄存器设定。用户可以写“1”到该位清除这个标志。 0 = 请求没有time-out 1 = 外设请求time-out
[7:3]	**Reserved**	保留.
[2]	**ALIGNF**	**传输对齐中断标志（只读）** 0 = PDMA通道源地址及目的地址都符合传输数据宽度设定 1 = PDMA通道源地址或目的地址不符合传输数据宽度设定
[1]	**TDIF**	**传输完成中断标志（只读）** 该位表明PDMA控制器完成了传输任务，用户可以读PDMA_TDSTS 寄存器来判断是那个通道传输完成了 0 = 还没有完成 1 = PDMA 通道完成了传输任务
[0]	**ABTIF**	**PDMA 读/写目标中止中断标志（只读）** 该位指示PDMA有目标终止错误；软件可以读寄存器PDMA_ABTSTS来找出哪个通道存在目标终止错误。 0 = 未收到AHB总线错误响应

位	描述	
		1 = 收到AHB总线错误响应

PDMA通道读写目标终止标志(PDMA_ABTSTS)

寄存器	偏移地址	R/W	描述	复位值
PDMA_ABTSTS	PDMA_BA + 0x420	R/W	PDMA通道读写目标终止标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**ABTIF8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABTIF7	**ABTIF6**	**ABTIF5**	**ABTIF4**	**ABTIF3**	**ABTIF2**	**ABTIF1**	**ABTIF0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**ABTIFn**	**PDMA读/写 目标中止中断状态标志** 该位表示PDMA控制器存在目标中止错误；用户可以写1清除这些位。 0 = 当通道n传输数据时，未收到AHB总线错误响应。 1 = 当通道n传输数据时，收到AHB总线错误响应。

PDMA 通道传输完成标志(PDMA_TDSTS)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TDSTS	PDMA_BA + 0x424	R/W	PDMA 通道传输完成标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**TDIF8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDIF7	**TDIF6**	**TDIF5**	**TDIF4**	**TDIF3**	**TDIF2**	**TDIF1**	**TDIF0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**TDIFn**	**PDMA传输结束标志结束标志寄存器** 该位指示PDMA控制器通道传输数据是否完成，用户可以写1清除这些位。 0 = PDMA 通道传输数据未完成。 1 = PDMA 通道传输数据完成。

PDMA传输对齐状态寄存器(PDMA_ALIGN)

寄存器	偏移地址	R/W	描述	复位值
PDMA_ALIGN	PDMA_BA + 0x428	R/W	PDMA 传输对齐状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**ALIGN8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ALIGN7	**ALIGN6**	**ALIGN5**	**ALIGN4**	**ALIGN3**	**ALIGN2**	**ALIGN1**	**ALIGN0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**ALIGNn**	**传输对齐标志** 0 = PDMA通道源地址及目的地址都符合传输数据宽度设定 1 = PDMA通道源地址或目的地址不符合传输数据宽度设定

PDMA 传输有效标志寄存器(PDMA_TACTSTS)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TACTSTS	PDMA_BA + 0x42C	R	PDMA传输有效标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**TXACTF8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXACTF7	**TXACTF6**	**TXACTF5**	**TXACTF4**	**TXACTF3**	**TXACTF2**	**TXACTF1**	**TXACTF0**

位	描述	
[31:9]	**Reserved**	保留.
[n] n=0,1..8	**TXACTFn**	**PDMA传输有效标志寄存器（只读）** 该位表示对应PDMA通道有效 0 = PDMA 通道传输还没有完成 1 = PDMA 通道有效

PDMA Time-out 预分频寄存器(PDMA_TOUTPSC)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TOUTPSC	PDMA_BA + 0x430	R/W	PDMA Time-out 预分频寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TOUTPSC1**			**Reserved**	**TOUTPSC0**		

位	描述	
[31:7]	**Reserved**	保留.
[6:4]	**TOUTPSC1**	**PDMA 通道 1 Time-out 时钟源预分频位** 000 = PDMA 通道1 time-out 时钟源为HCLK/2^8 001 = PDMA通道1 time-out 时钟源为HCLK/2^9. 010 = PDMA通道1 time-out 时钟源为HCLK/2^{10}. 011 = PDMA通道1 time-out 时钟源为HCLK/2^{11}. 100 = PDMA通道1 time-out 时钟源为HCLK/2^{12}. 101 = PDMA通道1 time-out 时钟源为HCLK/2^{13}. 110 = PDMA通道1 time-out 时钟源为HCLK/2^{14}. 111 = PDMA通道1 time-out 时钟源为HCLK/2^{15}.
[3]	**Reserved**	保留.
[2:0]	**TOUTPSC0**	**PDMA 通道0 Time-out 时钟源预分频位** 000 = PDMA通道0 time-out 时钟源为 HCLK/2^8. 001 = PDMA通道0 time-out 时钟源为HCLK/2^9. 010 = PDMA通道0 time-out 时钟源为HCLK/2^{10}. 011 = PDMA通道0 time-out 时钟源为HCLK/2^{11}. 100 = PDMA通道0 time-out 时钟源为HCLK/2^{12}. 101 = PDMA通道0 time-out 时钟源为HCLK/2^{13}. 110 = PDMA通道0 time-out 时钟源为HCLK/2^{14}. 111 = PDMA通道0 time-out 时钟源为HCLK/2^{15}

PDMA Time-out 使能寄存器(PDMA_TOUTEN)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TOUTEN	PDMA_BA + 0x434	R/W	PDMA Time-out 使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TOUTEN1**	**TOUTEN0**

位	描述	
[31:2]	**Reserved**	保留.
[n] n=0,1	**TOUTENn**	**PDMA Time-out 使能位** 0 = PDMA 通道 n time-out不使能 1 = PDMA 通道 n time-out 使能.

PDMA 中断使能寄存器 (PDMA_TOUTIEN)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TOUTIEN	PDMA_BA + 0x438	R/W	PDMA Time-out 中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TOUTIEN1**	**TOUTIEN0**

位	描述	
[31:2]	**Reserved**	保留.
[n] n=0,1	**TOUTIENn**	**PDMA Time-out中断使能标志位** 0 = PDMA 通道 n time-out 中断禁止 1 = PDMA 通道 n time-out 中断使能

PDMA Scatter-gather描述表基地址寄存器(PDMA_SCATBA)

寄存器	偏移地址	R/W	描述	复位值
PDMA_SCATBA	PDMA_BA + 0x43C	R/W	PDMA Scatter-gather 描述表基地址寄存器	0x2000_0000

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

位	描述	
[31:16]	**SCATBA**	**PDMA Scatter-gather 描述符表基地址寄存器** 在 Scatter-Gather 模式, 该寄存器是用于计算下一个链接地址的基地址 下一个链接地址计算公式是： 下一个链接地址 = PDMA_SCATBA + PDMA_DSCT_FIRST. **注:** 仅在Scatter-Gather 模式有效
[15:0]	**Reserved**	保留.

PDMA Time-out 周期计数器寄存器 0 (PDMA_TOC0_1)

寄存器	偏移地址	R/W	描述	复位值
PDMA_TOC0_1	PDMA_BA + 0x440	R/W	PDMA 通道0和通道1 Time-Out计数器寄存器	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC1							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC0							

位	描述	
[31:16]	**TOC1**	**通道 1 Time-out 计数器** 该寄存器用于控制通道1的time-out周期时间，计算的单位是基于TOUTPSC1 (PDMA_TOUTPSC[6:4])寄存器设置的时钟。Time-out 周期计算的例子可以参看寄存器TOC0 位的描述
[15:0]	**TOC0**	**通道 0 Time-out 计数器** 该寄存器用于控制通道0的time-out周期时间，计算的单位是基于TOUTPSC0(PDMA_TOUTPSC[2:0])寄存器设置的时钟。 Time-out 周期 = (time-out 时钟周期) * (16-bit TOCn),n = 0 ~ 1.

PDMA 通道复位控制寄存器(PDMA_CHRST)

寄存器	偏移地址	R/W	描述	复位值
PDMA_CHRST	PDMA_BA + 0x460	R/W	PDMA 通道复位控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**CH8RST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH7RST	**CH6RST**	**CH5RST**	**CH4RST**	**CH3RST**	**CH2RST**	**CH1RST**	**CH0RST**

位	描述	
[31:9]	**Reserved**	保留.
[8:0]	**CHnRST**	**通道 N 复位控制寄存器** 该位用于控制PDMA 通道复位。 0 = PDMA通道n不复位 1 = 复位PDMA通道n 。

PDMA请求源选择寄存器0 (PDMA_REQSEL0_3)

寄存器	偏移地址	R/W	描述	复位值
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA 请求源选择寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		REQSRC3					
23	22	21	20	19	18	17	16
Reserved		REQSRC2					
15	14	13	12	11	10	9	8
Reserved		REQSRC1					
7	6	5	4	3	2	1	0
Reserved		REQSRC0					

位	描述	
[31:30]	**Reserved**	保留.
[29:24]	**REQSRC3**	**通道 3 请求源选择** 该寄存器定义哪个外设连接到PDMA通道3用户可以通过设定寄存器REQSRC3配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[23:22]	**Reserved**	保留.
[21:16]	**REQSRC2**	**通道2请求源选择** 该寄存器定义哪个外设连接到PDMA通道2.用户可以通过设定寄存器REQSRC2配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[15:14]	**Reserved**	保留.
[13:8]	**REQSRC1**	**通道1请求源选择** 该寄存器定义哪个外设连接到PDMA通道1.用户可以通过设定寄存器REQSRC1配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[7:6]	**Reserved**	保留.
[5:0]	**REQSRC0**	**通道0请求源选择** 该寄存器定义哪个外设连接到PDMA通道0.用户可以通过设定寄存器REQSRC0配置外设。 0 =禁用PDMA 外设请求 1 = 保留. 4 = 通道连接到 UART0_TX. 5 = 通道连接到 UART0_RX. 6 = 通道连接到 UART1_TX. 7 = 通道连接到 UART1_RX. 8 = 通道连接到 UART2_TX. 9 = 通道连接到 UART2_RX.

位	描述	
		10 = 通道连接到 USCI0_TX. 11 = 通道连接到 USCI0_RX. 12 = 通道连接到 USCI1_TX. 13 = 通道连接到 USCI1_RX. 14 = 保留. 15 = 保留. 16 = 通道连接到 QSPI0_TX. 17 = 通道连接到 QSPI0_RX. 18 = 通道连接到 SPI0_TX. 19 = 通道连接到 SPI0_RX. 20 = 通道连接到 ADC_RX. 21 = 通道连接到 PWM0_P1_RX. 22 = 通道连接到 PWM0_P2_RX. 23 = 通道连接到 PWM0_P3_RX. 24 = 通道连接到 PWM1_P1_RX. 25 = 通道连接到 PWM1_P2_RX. 26 = 通道连接到 PWM1_P3_RX. 27 =保留. 28 = 通道连接到 I2C0_TX. 29 = 通道连接到 I2C0_RX. 30 = 通道连接到 I2C1_TX. 31 = 通道连接到 I2C1_RX. 32 = 通道连接到 TMR0. 33 = 通道连接到 TMR1. 34 = 通道连接到 TMR2. 35 = 通道连接到 TMR3. 36 = 通道连接到 UART3_TX. 37 = 通道连接到 UART3_RX. 38 = 通道连接到 UART4_TX. 39 = 通道连接到 UART4_RX. 40 = 通道连接到 UART5_TX. 41 = 通道连接到 UART5_RX. 42 = 通道连接到 UART6_TX. 43 = 通道连接到 UART6_RX. 44 = 通道连接到 UART7_TX. 45 = 通道连接到 UART7_RX. 其他 =保留. 注 1:一个外设不能同时连接到两个通道。 注 2: 当传输是在内存之间是该寄存器无用

PDMA 请求源选择寄存器1 (PDMA_REQSEL4_7)

寄存器	偏移地址	R/W	描述	复位值
PDMA_REQSEL4_7	PDMA_BA + 0x484	R/W	PDMA请求源选择寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		**REQSRC7**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**REQSRC6**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**REQSRC5**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**REQSRC4**					

位	描述	
[31:30]	**Reserved**	保留.
[29:24]	**REQSRC7**	**通道7请求源选择寄存器** 该寄存器定义哪个外设连接到PDMA通道7。用户可以通过设定寄存器REQSRC7配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[23:22]	**Reserved**	保留.
[21:16]	**REQSRC6**	**通道6请求源选择寄存器** 该寄存器定义哪个外设连接到PDMA通道6。用户可以通过设定寄存器REQSRC6配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[15:14]	**Reserved**	保留.
[13:8]	**REQSRC5**	**通道5请求源选择寄存器** 该寄存器定义哪个外设连接到PDMA通道5。用户可以通过设定寄存器REQSRC5配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。
[7:6]	**Reserved**	保留.
[5:0]	**REQSRC4**	**通道4请求源选择寄存器** 该寄存器定义哪个外设连接到PDMA通道4.用户可以通过设定寄存器REQSRC4配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。

PDMA请求源选择寄存器2 (PDMA_REQSEL8)

寄存器	偏移地址	R/W	描述	复位值
PDMA_REQSEL8	PDMA_BA + 0x488	R/W	PDMA请求源选择寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**REQSRC8**					

位	描述	
[31:6]	**Reserved**	保留.
[5:0]	**REQSRC8**	**通道8请求源选择寄存器** 该寄存器定义哪个外设连接到PDMA通道8。用户可以通过设定寄存器REQSRC8配置外设。 **注意:** 通道配置与寄存器REQSRC0相同。请参考REQSRC0说明。

6.7 定时器控制器(TMR)

6.7.1 概述

定时器控制器包含4组32位定时器，定时器0--定时器3。可以让用户很容易的在应用中开发一个定时器应用。定时器能执行的功能有：频率测量，延时，时钟产生，通过外部管脚进行事件计数，通过外部捕获管脚进行脉宽测量。

6.7.2 特性

6.7.2.1 定时器功能特性

- 4 组 32-位定时器，带24位向上计数器和一个8位的预分频计数器
- 每个定时器都可以设置独立的时钟源
- 提供 one-shot, periodic, toggle 和 continuous 四种计数操作模式
- 通过CNT(TIMERx_CNT[23:0])可读取内部24位向上计数器的值
- 支持事件计数功能
- 支持计数USB SOF 信号
- 通过CAPDAT (TIMERx_CAP[23:0])可读取24位捕捉值
- 支持外部管脚捕捉功能，可用于脉宽测量
- 支持外部管脚事件计数，可用于复位24位向上定时器
- 支持ACMP输出信号及LIRC变化触发内部捕获功能
- 如果定时器中断信号产生，支持芯片从空闲/掉电模式唤醒
- 支持定时器0 ~ 定时器3 超时溢出中断或捕捉中断来触发PWM，PDMA，ADC，BPWM功能
- 支持定时器间触发模式

6.7.3 框图

定时器的框图和时钟控制如下所示：

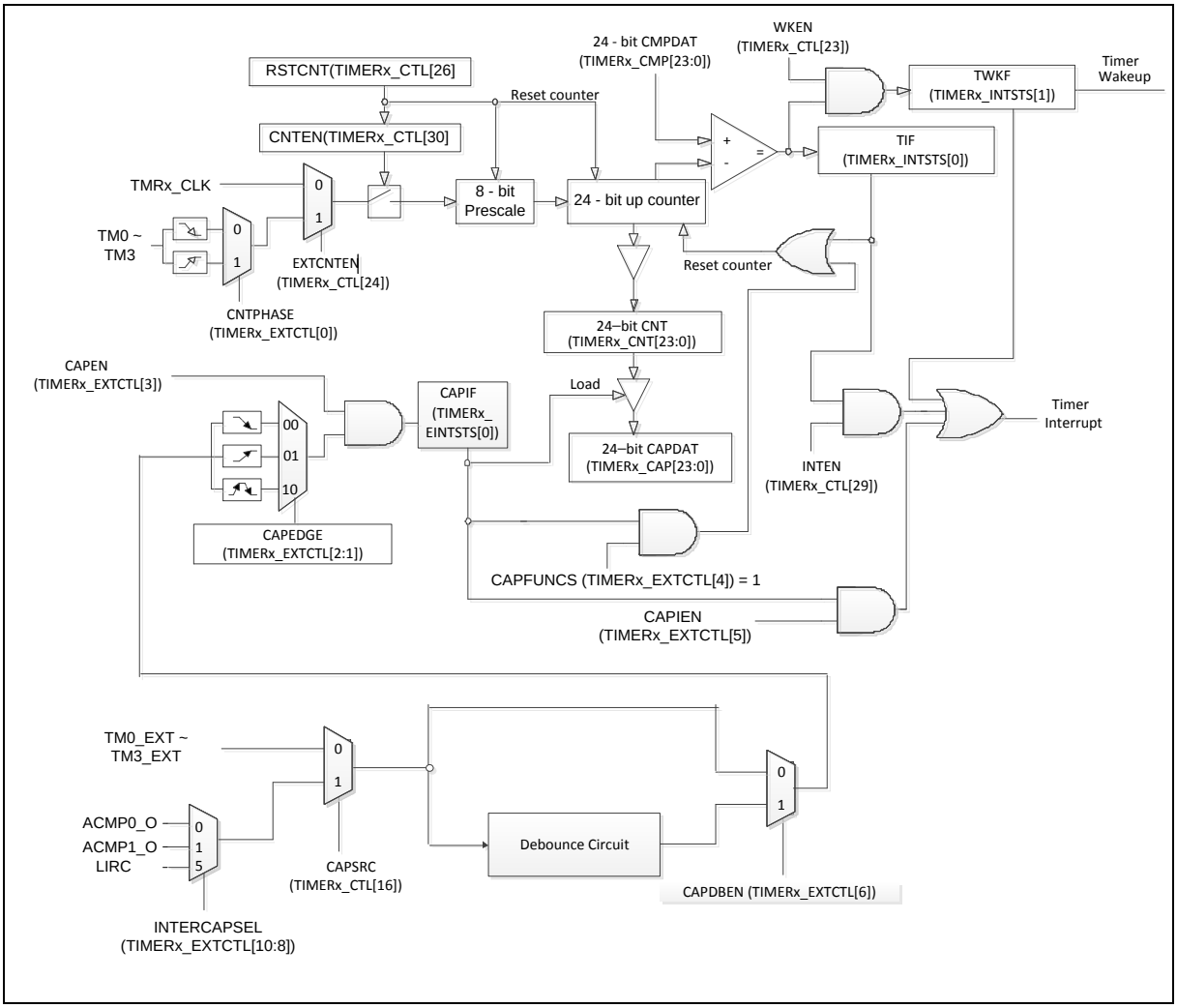


图 6.7-1 定时器框图

6.7.4 基本配置

- 时钟源配置
  - 定时器模式下定时器0~定时器3的时钟源可以由TMRxCKEN(CLK_APBCLK0[5:2])设定
  - 定时器0~定时器3的时钟源选择分别由TMR0SEL(CLK_CLKSEL1[10:8])，TMR1SEL(CLK_CLKSEL1[14:12])，及TMR3SEL(CLK_CLKSEL1[22:20])设定

定时器0~定时器3的时钟控制如图 6.7-2：

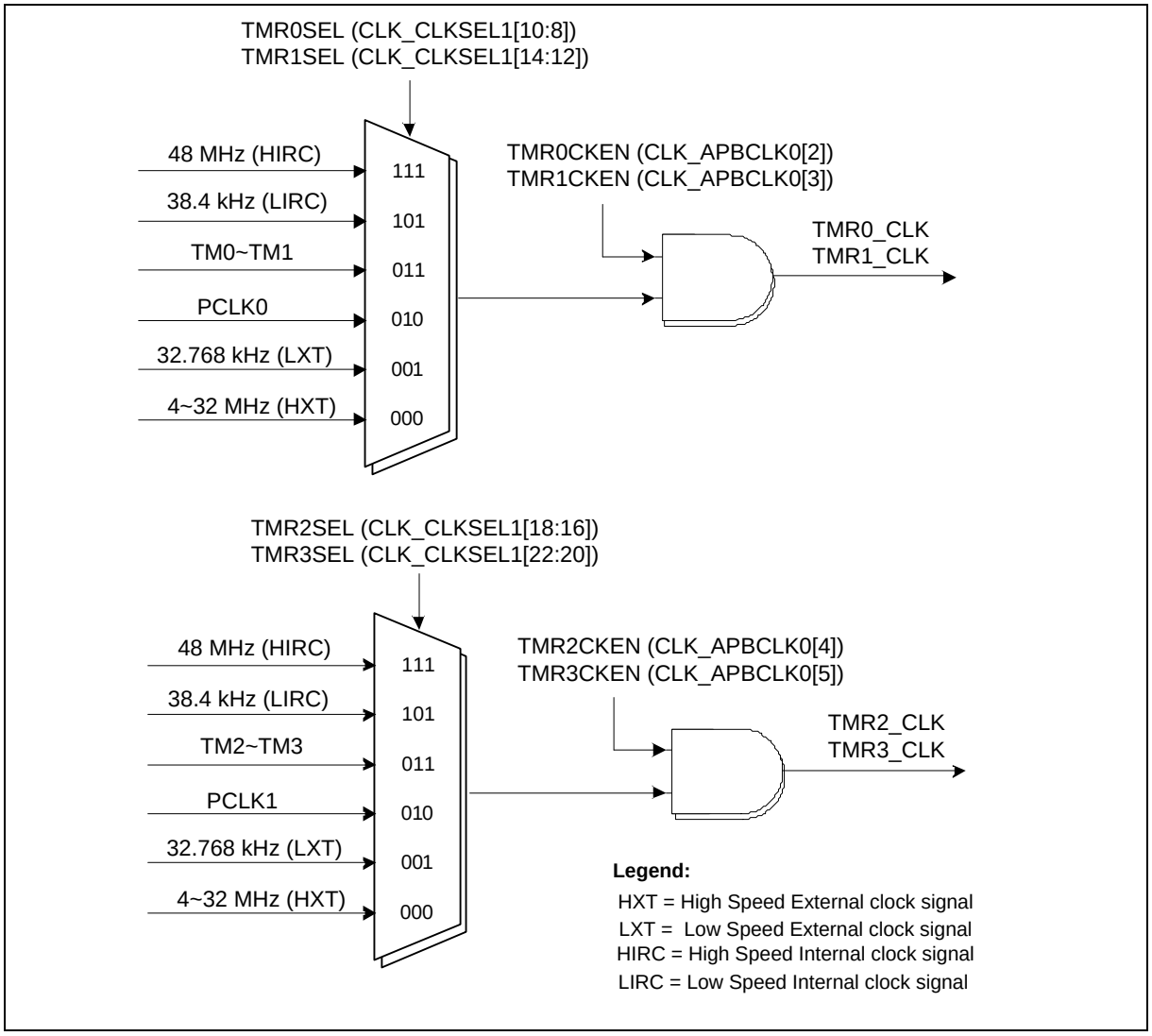


图 6.7-2 定时器时钟源

- 定时器0~定时器3 MFP 配置
  - 定时器0~定时器3的MFP 如表 6.7-1：.

信号名称	管脚名称	MFP
TM0	PB.5, PC.7	MFP14
	PG.2	MFP13

TM1	PB.4, PC.6	MFP14
	PC.14, PG.3	MFP13
TM2	PA.7, PB.3, PD.0	MFP14
	PF.15, PG.4	MFP13
TM3	PA.6, PB.2, PD.15, PF.14	MFP14
	PF.11	MFP13
TM0_EXT	PA.11, PB.15	MFP13
TM1_EXT	PA.10, PB.14	MFP13
TM2_EXT	PA.9, PB.13	MFP13
TM3_EXT	PA.8, PB.12	MFP13

表 6.7-1 定时器 0~定时器 3 MFP 表

6.7.5 功能描述

6.7.5.1 定时器中断标志

定时器控制器支持两个中断标志：一个是TIF(TIMERx_INTSTS[0])标志，该标志当定时器计数器值CNT(TIMERx_CNT[23:0])与定时器比较值CMPDAT(TIMERx_CMP[23:0])相匹配时置位，另一个是CAPIF(TIMERx_EINTSTS[0])标志，该标志当Tx_EXT管脚，LIRC，或ACMP输出的变化与CAPEDGE(TIMERx_EXTCTL[2:1])的设置一致时置位，用户可以通过设置CAPSRC(TIMERx_CTL[16])和INTERCAPSEL(TIMERx_EXTCTL[10:8])来选择捕获源。TWKF(TIMERx_INTSTS[1])显示了定时器的中断唤醒标志，设置WKEN(TIMERx_CTL[23])为1使能定时器唤醒功能。

6.7.5.2 定时器计数模式

定时器控制器提供四种定时器计数模式：单周期（one-shot）,周期（periodic）,反转输出（toggle-output）和连续计数（continuous counting）模式。

6.7.5.3 单周期模式

如果定时器工作在单周期(one-shot)模式(TIMERx_CTL[28:27]为00，且CNTEN(TIMERx_CTL[30])置1)，则定时器的计数器开始计数。一旦CNT(TIMERx_CNT[23:0])计数器的值达到CMPDAT(TIMERx_CMP[23:0])的值时，TIF(TIMERx_INTSTS[0])标志将变为1，CNT的值和CNTEN位将由定时器控制器自动清零，然后定时器停止操作。与此同时，如果INTEN(TIMERx_CTL[29])位使能，则定时器中断信号产生并送到NVIC通知CPU。

用户可以设置ICEDEBUG(TIMERx_CTL[31])为1禁止ICE调试模式响应，它在CPU停止时会影响定时器的计数。

6.7.5.4 周期模式

如果定时器工作在周期(periodic) 模式(TIMERx_CTL[28:27]为01)且CNTEN(TIMERx_CTL[30])置1，则定时器的计数器开始向上计数。一旦CNT(TIMERx_CNT[23:0])计数器的值达到CMPDAT(TIMERx_CMP[23:0])的值时，TIF(TIMERx_INTSTS[0])标志将变为1，CNT的值将由定时器控制器自动清零，然后定时器重新计数。与此同时，如果INTEN(TIMERx_CTL[29])使能，则定时器中断信号产生并送到NVIC通知CPU。在该模式，定时器控制器周期性地操作计数和与CMPDAT的值比较，直到CNTEN位由软件清0。

6.7.5.5 反转模式

如果定时器工作在反转输出(toggle-out)模式(TIMERx_CTL[28:27]为10)且CNTEN(TIMERx_CTL[30])位

置1，则定时器的计数器开始计数。toggle-out模式的计数操作大部分与周期模式是一样的，除了该模式当TIF(TIMERx_INTSTS[0])位设置时，相关的TM0~TM3或TM0_EXT~TM3_EXT管脚会输出反转信号，用户可以通过TGLPINSEL(TIMERx_CTL[22])选择TMx，还是TMx_EXT作为反转输出管脚。因此，管脚T0~T3上的触发输出信号以50%的占空周期反复改变。

6.7.5.6 连续计数模式

如果定时器工作在连续计数(continuous counting)模式(TIMERx_CTL[28:27]为11)且CNTEN(TIMERx_CTL[30])位置1，则定时器的计数器开始计数。一旦CNT(TIMERx_CNT[23:0])的值达到CMPDAT(TIMERx_CMP[23:0])的值时，TIF(TIMERx_INTSTS[0])标志将变为1，但CNT的值继续保持向上计数。与此同时，如果INTEN(TIMERx_CTL[29])使能，则定时器中断信号产生并送到NVIC通知CPU。在该模式，用户可以立刻改变不同的CMPDAT值，而不需要停止定时器计数和重新开始定时器计数。

例如，CMPDAT的值设置为80。当CNT达到80时,TIF标志将被置1，定时器计数器继续计数，而且CNT的值将不回到0，而是继续计数，81,82,83,···到(2^{24} -1)，然后再一次0,1,2,3,···到2^{24} -1，如此往复。接下来，如果软件改变CMPDAT的值为200并且清除TIF标志位，当CNT的值达到200时，TIF标志将再次变为1。最后，软件改变CMPDAT的值为500并且清除TIF标志，当CNT的值达到500时，TIF标志将再次变为1。

在该模式，计数器计数是连续的。所以该操作模式叫做连续计数模式。

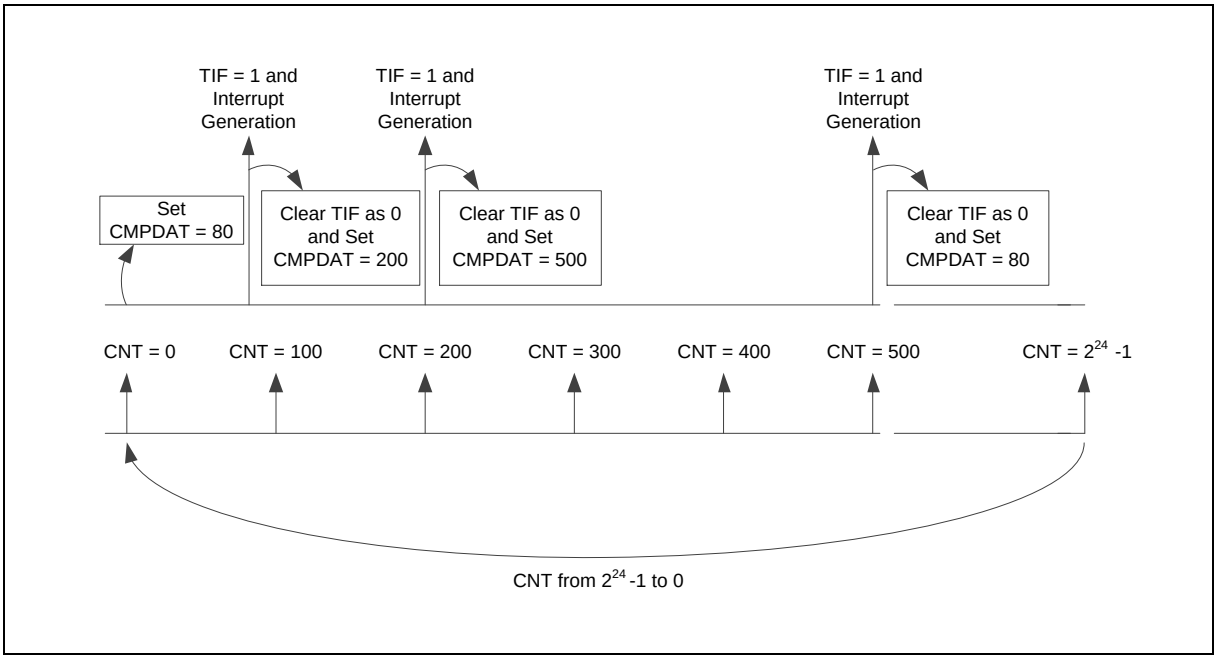


图 6.7-3 连续计数模式

6.7.5.7 事件计数模式

定时器控制器也提供这样的应用，能对输入事件(来自管脚TMx=0~3)计数并将事件的次数反应到CNT(TIMERx_CNT[23:0])的值。也可以称为事件计数功能。该功能下，EXTCNTEN(TIMERx_CTL[24])位需置位并且定时器外设时钟源必须设为PCLK。

如果ECNTSSEL(TIMERx_EXTCTL[16])为0，事件计数源来自外部TMx管脚，软件可以通过CNTDBEN(TIMERx_EXTCTL[7])位来使能或关闭TMx管脚消抖电路。如果TMx管脚的消抖电路关闭，输入事件频率必须少于1/3 PCLK，如果消抖电路打开，输入事件的频率须小于1/8 PCLK，以保证CNT的值是正确的。软件也可以通过设置CNTPHASE(TIMERx_EXTCTL[0])来选择边沿检测TMx管脚的相位。

事件计数模式下，定时器计数操作模式可以设置为单次，周期，和连续计数模式来计算来自TMx管脚的输入事件CNT(TIMERx_CNT[23:0])的值。

如果ECNTSSEL(TIMERx_EXTCTL[16])为1，事件计数源将会是USB设备检测到起始帧(SOF)信号，请参考USB设备规格书。

6.7.5.8 捕获模式

事件捕获功能是当检测到Tx_EXT(x=0~3)，LIRC或ACMP管脚边沿电平有变化时，CNT(TIMERx_CNT[23:0])会送到CAPDAT(TIMERx_CAP[23:0])。在该模式下，需把CAPFUNCS(TIMERx_EXTCTL[4])位设置为0，用来选择Tx_EXT变化时用作事件捕捉功能，而且定时器外设时钟源必须设为PCLK。

如果CAPSRC(TIMERx_CTL[16])为0捕获事件由TMx_EXT管脚触发，用户可以通过CAPDBEN(TIMERx_EXTCTL[6])位来使能或关闭TMx_EXT管脚消抖电路。在TMx_EXT的消抖电路关闭时，TMx_EXT管脚的转变频率必须少于1/3 PCLK，在TMx_EXT的消抖电路打开时，TMx_EXT管脚的转变频率必须少于1/8 PCLK，以保证捕捉功能能够正常工作。用户也可以通过设置CAPEDGE(TIMERx_EXTCTL[2:1])位来选择TMx_EXT管脚的边沿转变检测方式。

在事件捕捉模式，用户不用考虑定时器计数器工作模式的选择，只有当检测到TMx_EXT管脚有边沿变化时捕捉事件才会发生。

设置CAPIEN(TIMERx_EXTCTL[5])用于产生捕获中断，当TMx_EXT管脚信号改变符合设定，CAPIF将被置位。

如果用户不清除CAPIF状态标志，用户应知道此时的定时器会保持TIMERx_CAP寄存器的值不变，且不会保存新的捕捉值。

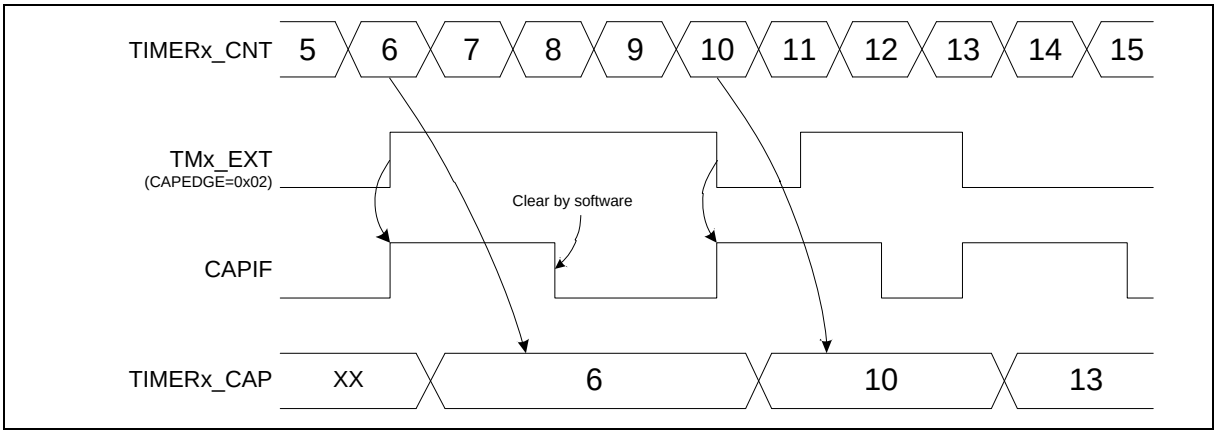


图 6.7-4 外部捕获模式

如果CAPSRC (TIMERx_CTL[16]) 为1，设置INTERCAPSEL(TIMERx_EXTCTL[10:8])选择不同的捕获源，捕获源可以是ACMP0（INTERCAPSEL为000），ACMP1（INTERCAPSEL为001）的内部输出信号或者LIRC（INTERCAPSEL 为101）。

6.7.5.9 复位计数模式

定时器也提供了事件复位计数器功能，在捕获事件发生时用来复位CNT(TIMERx_CNT[23:0])的值。在该模式时CAPFUNCS(TIMERx_EXTCTL[4])位置1，用户还需设置CAPSRC和INTERCAPSEL选择TMx_EXT管脚变化，内部ACMPx信号还是LIRC来触发复位计数。

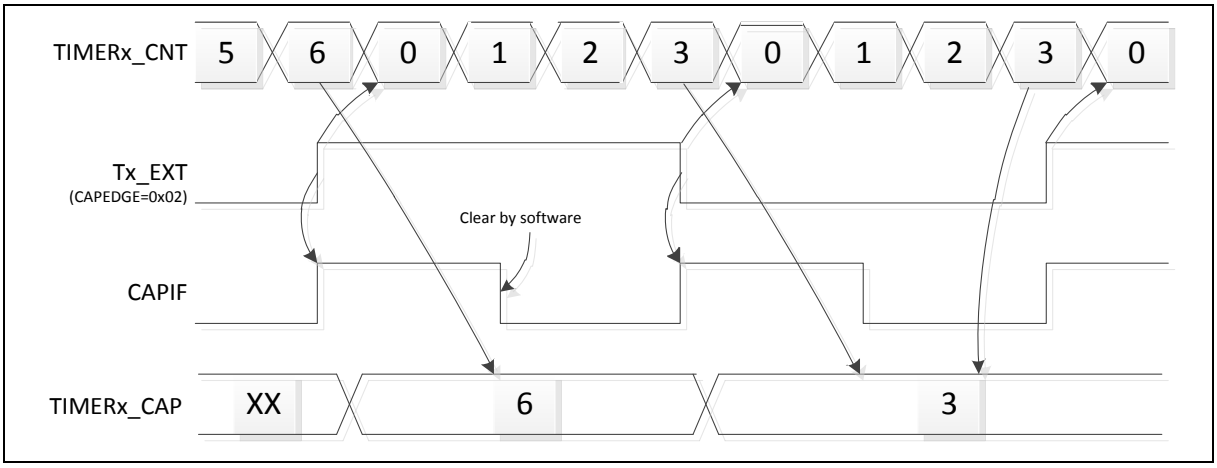


图 6.7-5 复位计数模式

6.7.5.10 定时器触发功能

定时器控制器提供定时器超时溢出中断或捕捉中断来触发PWM，BPWM，PDMA和ADC。如果TRGSSEL(TIMERx_CTL[18])为0，超时溢出中断信号用于触发PWM，BPWM，PDMA和ADC。如果TRGSSEL(TIMERx_CTL[18])为1，捕捉中断信号用于触发PWM，BPWM，PDMA和ADC。

当TRGPWM (TIMERx_CTL[19])被置1，如果定时中断信号产生，定时器控制器将产生一个触发脉冲作为PWM外部时钟源。

当TRGADC(TIMERx_CTL[21])被置1，如果定时中断信号产生，定时器控制器将触发ADC开始转换。

当TRGPDMA(TIMERx_CTL[8])被置1，如果定时中断信号产生，定时器控制器将产生一个PDMA传输请求。

当TRGBPWM(TIMERx_CTL[9]) 被置1，如果定时中断信号产生，定时器控制器将触发BPWM。

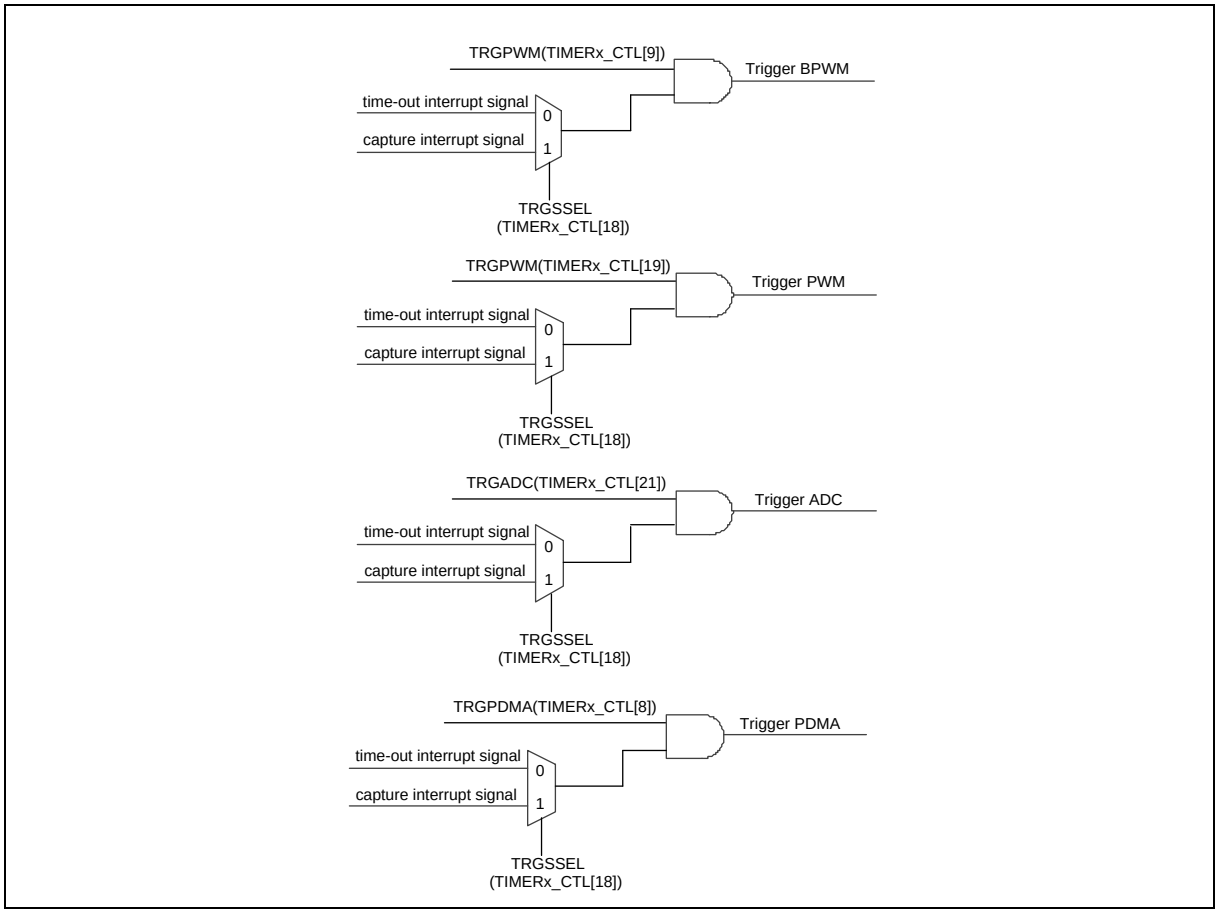


图 6.7-6 内部定时器触发

6.7.5.11 定时器间触发捕获模式

这种模式下，定时器0/2将处于外部事件计数模式，并且会产生内部信号(INTR_TMR_TRG)来触发定时器1/3开始或停止计数。定时器1/3将处于捕获模式，并且通过定时器0/2的计数状态触发定时器1/3开始/停止。

设置定时器0定时器间触发捕获使能，定时器1的触发计数捕获功能会被使能；设置定时器2定时器间触发捕获使能，定时器3的触发计数捕获功能会被使能。

开始触发

当定时器0/2的INTRGEN(TIMERx_CTL[10])置位，并且定时器0/2的24位计数值（CNT）从0变成1的时候将产生 INTR_TMR_TRG信号的上升沿，同时定时器1/3将会立即自动计数。

停止触发

当定时器0/2的CNT值达到CMPADAT值时，定时器0/2将产生INTR_TMR_TRG信号的下降沿，定时器0/2的计数模式将被禁止并且INTRGEN(TIMERx_CTL[10])将被硬件清零，定时器1/3也将会停止计数，同时定时器1/3的CNT值将被保存到CAPDAT(TIMERx_CAP[23:0])。

定时器间触发模式可以用来更精确测量外部事件(TMx)的事件周期，图 6.7-7 说明了定时器0/1定时器间触发捕获模式的时序流程。

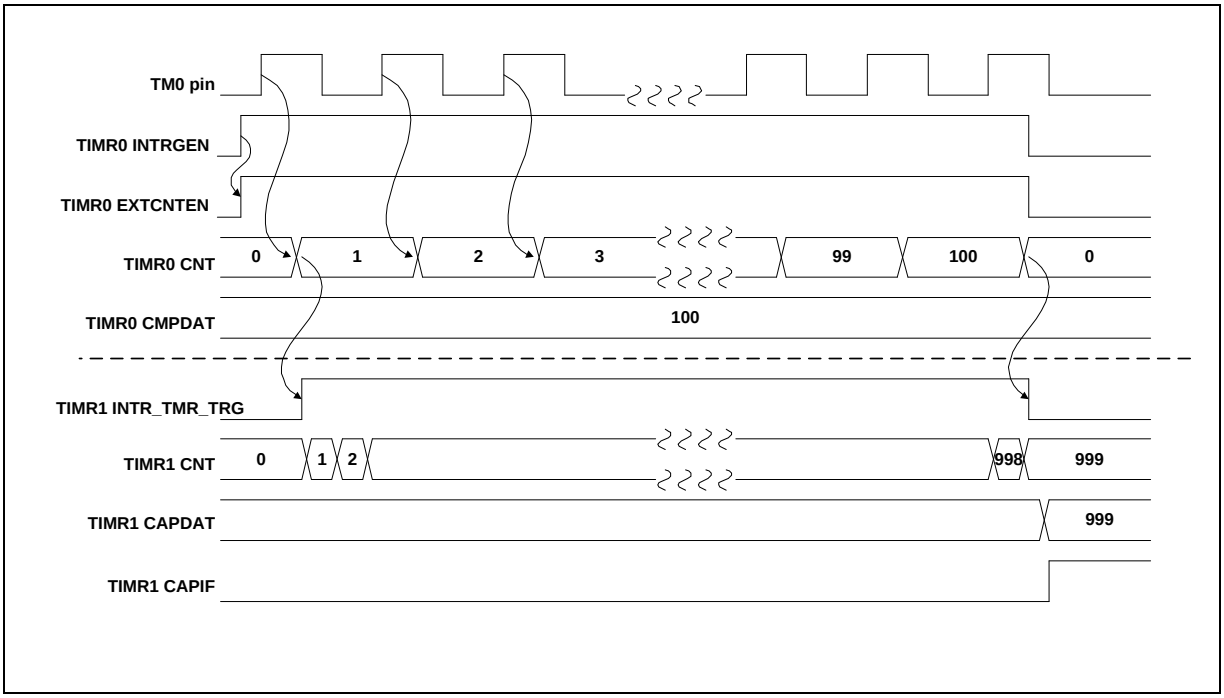


图 6.7-7 定时器间触发捕获时序

用户必须清除定时器1/3 CAPIF才能再次使用定时器间触发捕获功能。

6.7.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移量	R/W	描述	复位值
TIMER基地址: **TMR01_BA = 0x4005_0000** **TMR23_BA = 0x4005_1000**				
TIMER0_CTL	TMR01_BA+0x00	R/W	定时器0 控制寄存器	0x0000_0005
TIMER0_CMP	TMR01_BA+0x04	R/W	定时器0 比较寄存器	0x0000_0000
TIMER0_INTSTS	TMR01_BA+0x08	R/W	定时器0中断状态寄存器	0x0000_0000
TIMER0_CNT	TMR01_BA+0x0C	R	定时器0 数据寄存器	0x0000_0000
TIMER0_CAP	TMR01_BA+0x10	R	定时器0 捕获数据寄存器	0x0000_0000
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	定时器0 外部控制寄存器	0x0000_0000
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	定时器0外部中断状态寄存器	0x0000_0000
TIMER1_CTL	TMR01_BA+0x20	R/W	定时器1 控制寄存器	0x0000_0005
TIMER1_CMP	TMR01_BA+0x24	R/W	定时器1 比较寄存器	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x28	R/W	定时器1 中断状态寄存器	0x0000_0000
TIMER1_CNT	TMR01_BA+0x2C	R	定时器1 数据寄存器	0x0000_0000
TIMER1_CAP	TMR01_BA+0x30	R	定时器1 捕获数据寄存器	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x34	R/W	定时器1 外部控制寄存器	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x38	R/W	定时器1 外部中断状态寄存器	0x0000_0000
TIMER2_CTL	TMR23_BA+0x00	R/W	定时器2 控制寄存器	0x0000_0005
TIMER2_CMP	TMR23_BA+0x04	R/W	定时器2 比较寄存器	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	定时器2 中断状态寄存器	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R	定时器2 数据寄存器	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	定时器2 捕获数据寄存器	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	定时器2 外部控制寄存器	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	定时器2 外部中断状态寄存器	0x0000_0000
TIMER3_CTL	TMR23_BA+0x20	R/W	定时器3 控制寄存器	0x0000_0005
TIMER3_CMP	TMR23_BA+0x24	R/W	定时器3 比较寄存器	0x0000_0000
TIMER3_INTSTS	TMR23_BA+0x28	R/W	定时器3 中断状态寄存器	0x0000_0000
TIMER3_CNT	TMR23_BA+0x2C	R	定时器3 数据寄存器	0x0000_0000

TIMER3_CAP	TMR23_BA+0x30	R	定时器3 捕获数据寄存器	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x34	R/W	定时器3 外部控制寄存器	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x38	R/W	定时器3 外部中断状态寄存器	0x0000_0000

6.7.7 寄存器描述

定时器控制寄存器(TIMERx_CTL)

寄存器	偏移量	R/W	描述	复位值
TIMER0_CTL	TMR01_BA+0x00	R/W	定时器0控制寄存器	0x0000_0005
TIMER1_CTL	TMR01_BA+0x20	R/W	定时器1控制寄存器	0x0000_0005
TIMER2_CTL	TMR23_BA+0x00	R/W	定时器2控制寄存器	0x0000_0005
TIMER3_CTL	TMR23_BA+0x20	R/W	定时器3控制寄存器	0x0000_0005

31	30	29	28	27	26	25	24
ICEDEBUG	**CNTEN**	**INTEN**	**OPMODE**		**RSTCNT**	**ACTSTS**	**EXTCNTEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WKEN	**TGLPINSEL**	**TRGADC**	**Reserved**	**TRGPWM**	**TRGSSEL**	**Reserved**	**CAPSRC**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**INTRGEN**	**TRGBPWM**	**TRGPDMA**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC							

位	描述	
[31]	**ICEDEBUG**	**仿真器 (ICE) 调试模式响应禁止位 （写保护位）** 0 = 仿真器调试模式响应定时器计数。 当调试器调试模式响应时，定时器计数器将被固定住。 1 = 仿真器调试模式响应禁用。 无论调试器调试模式响应与否，定时器计数器将持续计数下去。 **注:** 该位为写保护 位。详情请参考SYS_REGLCTL 寄存器
[30]	**CNTEN**	**定时器计数使能位** 0 = 停止/暂停计数 1 = 开始计数 **注**1：在停止状态，设置CNTEN为 1 将使能 24-位向上计数器从上次停止的计数值继续计数。 **注**2：在 one-shot 模式下(TIMER_CTL[28:27] = 00)，当相应的定时中断标志 TIF (TIMERx_INTSTS[0])产生时，该位由硬件自动清零 **注3**：设置使能/禁止此位需要2 * TMR_CLK周期才能生效，用户可以通过读ACTSTS (TIMERx_CTL[25])确认使能/禁止命令是否完成
[29]	**INTEN**	**定时器中断使能位** 0 = 定时器中断禁止. 1 = 定时器中断使能 **注:** 如果该位被使能，当定时器中断标志TIF 被置1，定时器中断信号将产生并通知CPU

[28:27]	**OPMODE**	**定时器操作模式选择** 00 = 定时器控制器为单次（One-shot）模式 01 = 定时器控制器为周期(Periodic)模式 10 = 定时器控制器为触发输出(Toggle-output)模式 11 = 定时器控制器为连续计数(Continuous Counting)模式
[26]	**RSTCNT**	**定时器计数器复位位** 设置该位将复位24位向上计数器CNT (TIMERx_CNT[23:0])的值，且如果ACTSTS (TIMERx_CTL[25])位为1，则会把CNTEN (TIMERx_CTL[30])强制清0. 0 = 该位写 0 无效 1 =复位8-位预分频计数器，内部24-位向上计数器的值和CNTEN位 **注:** 此位将被自动清零
[25]	**ACTSTS**	**定时器有效状态位（只读）** 该位表明24位向上计数器的状态 0 = 24位向上计数器无效 1 = 24位向上计数器有效 注： 此位当CNT由0变1时有效
[24]	**EXTCNTEN**	**事件计数模式使能位** 该位用来使能外部计数管脚功能 0 = 禁用事件计数器模式 1 = 使能事件计数器模式 注1：当定时器用作事件计数器，该位需要设置成1.并选择PCLK (x=0~1)作为定时器时钟源 注2：当TMR0/TMR2 INTRGEN 设置为1 此位强制为1，当INTRGEN=1且TMR1/TMR3 CAPIF (TIMERx_EINTSTS[0])=1时，此位强制为0
[23]	**WKEN**	**唤醒功能使能位** 如果该位置1，当定时器中断标志TIF (TIMERx_INTSTS[0])被置1，且INTEN (TIMERx_CTL[29])被使能，定时器中断信号将产生一个唤醒触发事件给CPU 0 = 如果定时器中断信号产生，唤醒功能禁止 1 =如果定时器中断信号产生，唤醒功能使能
[22]	**TGLPINSEL**	**Toggle-Output 管脚选择** 0 = Toggle 模式输出到 Tx 脚(定时器事件计数管脚). 1 = Toggle模式输出到Tx_EXT (定时器外部捕捉管脚)
[21]	**TRGADC**	**触发 ADC 使能位** 如果该位被置1，定时器超时溢出中断或捕捉中断将触发ADC。 0 = 定时器中断触发 ADC 禁止. 1 = 定时器中断触发ADC 使能. **注意：** 如果 TRGSSEL (TIMERx_CTL[18]) = 0,超时溢出中断信号将触发ADC. 如果 TRGSSEL (TIMERx_CTL[18]) = 1, 捕捉中断信号将触发 ADC
[20]	**Reserved**	保留.
[19]	**TRGPWM**	**触发PWM使能位** 如果此位置1定时器溢出中断或者捕获中断可以触发PWM 0 = 定时器触发PWM禁止 1 = 定时器触发PWM使能 **注:** 如果 TRGSSEL (TIMERx_CTL[18]) = 0,时间溢出中断信号触发PWM

		如果 TRGSSEL (TIMERx_CTL[18]) = 1,捕获中断信号触发PWM
[18]	**TRGSSEL**	**触发源选择位** 该位用于选择触发源是从超时溢出中断信号还是捕捉中断信号。 0 = 选择超时溢出中断信号触发 PWM，BPWM，PDMA及ADC 1 = 选择捕捉中断信号触发触发 PWM，BPWM，PDMA及ADC
[17]	**Reserved**	保留.
[16]	**CAPSRC**	**捕获引脚源选择** 0 = 捕获功能源于TMx_EXT (x= 0~3)管脚 1 = 捕获功能源于内部ACMP输出或者LIRC，用户可以设置INTERCAPSEL (TIMERx_EXTCTL[10:8])决定内部ACMP输出信号还是LIRC作为定时器捕获源
[15:11]	**Reserved**	保留.
[10]	**INTRGEN**	**定时器间触发模式使能** 设置此位将使能定时器间触发捕获功能 此模式下，定时器0/2将处于事件计数模式，计数外部时钟或事件，同时定时器1/3将处于捕获功能的触发计数模式 0 = 定时器间触发模式禁止 1 = 定时器间触发模式使能 **注:** 对于 Timer1/3此位忽略，读出数据为0
[9]	**TRGBPWM**	**触发 BPWM使能位** 如果设置该位为1，定时器超时中断发生或者捕获中断发生将触发BPWM。 0 = 定时器中断触发BPWM禁止 1 = 定时器中断触发BPWM使能 **注：** 如TRGSSEL (TIMERx_CTL[18]) = 0，超时中断信号将触发 BPWM。如TRGSSEL(TIMERx_CTL[18]) = 1，捕获中断信号将触发 BPWM
[8]	**TRGPDMA**	**触发PDMA使能位** 如果该位置1，定时器溢出中断信号或捕获中断信号触发PDMA 0 =定时器中断触发PDMA禁用 1 =定时器中断出发PDMA使能 **注:** 如果TRGSSEL (TIMERx_CTL[18]) = 0，溢出中断信号触发PDMA 如果 TRGSSEL (TIMERx_CTL[18]) = 1， 捕获中断信号触发PDMA
[7:0]	**PSC**	**预分频计数器** 定时器的输入时钟源或事件计数源被PSC+1预分频，然后再输入到定时器。如果 该位为0，表示不分频。 **注：** 更新预分频器将复位内部8位预分频计数器及24位向上计数器

定时器比较寄存器(TIMERx_CMP)

寄存器	偏移量	R/W	描述	复位值
TIMER0_CMP	TMR01_BA+0x04	R/W	定时器0 比较寄存器	0x0000_0000
TIMER1_CMP	TMR01_BA+0x24	R/W	定时器1 比较寄存器	0x0000_0000
TIMER2_CMP	TMR23_BA+0x04	R/W	定时器2 比较寄存器	0x0000_0000
TIMER3_CMP	TMR23_BA+0x24	R/W	定时器3 比较寄存器	0x0000_0000

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

位	描述	
[31:24]	**Reserved**	保留.
[23:0]	**CMPDAT**	**定时器比较值** CMPDAT是24-位比较寄存器。当内部 24-位向上计数器的值等于CMPDAT的值时，TIF (TIMERx_INTSTS[0] 定时器中断标志)将被置1。 超时溢出周期 = (定时器输入时钟周期) * (8-bit PSC + 1) * (24-bit CMPDAT) **注**1：不能向CMPDAT里写 0x0 或 0x1，否则内核将运行到未知状态 **注**2：当定时器工作在 continuous counting模式，即使软件写一个新的值到CMPDAT，24-位向上计数定时器将保持继续计数。如果定时器工作在其他模式，如果软件写一个新的值到CMPDAT，定时器将使用新比较值并退出当前计数，开始重新计数

定时器中断状态寄存器(TIMERx_INTSTS)

寄存器	偏移量	R/W	描述	复位值
TIMER0_INTSTS	TMR01_BA+0x08	R/W	定时器0中断状态寄存器	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x28	R/W	定时器1中断状态寄存器	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	定时器2中断状态寄存器	0x0000_0000
TIMER3_INTSTS	TMR23_BA+0x28	R/W	定时器3中断状态寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留.
[1]	**TWKF**	**定时器唤醒标志位** 该位表示定时器的中断唤醒标志状态。 0 = 定时器不会引起CPU 唤醒 1 = 如果定时器中断信号产生， CPU 从空闲或掉电模式唤醒 **注:** 该位必须通过软件写1清0
[0]	**TIF**	**定时器中断标志** 当内部 24-位向上计数定时器CNT (TIMERx_CNT[23:0])的值与定时器比较值CMPDAT (TIMERx_CMP[23:0])匹配时，该位会显示其中断状态。. 0 = 无影响 1 = 计数定时器与CMPDAT的值相匹配 **注**：该位写 1 清零

定时器数据寄存器(TIMERx_CNT)

寄存器	偏移量	R/W	描述	复位值
TIMER0_CNT	TMR01_BA+0x0C	R	定时器0 数据寄存器	0x0000_0000
TIMER1_CNT	TMR01_BA+0x2C	R	定时器1 数据寄存器	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R	定时器2 数据寄存器	0x0000_0000
TIMER3_CNT	TMR23_BA+0x2C	R	定时器3 数据寄存器	0x0000_0000

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

位	描述	
[31:24]	**Reserved**	保留.
[23:0]	**CNT**	**定时器数据寄存器** 该位可以反映内部24位定时器计数器的值。 如果EXTCNTEN(TIMERx_CTL[24])为0，用户可以读CNT的值来获取当前24位向上计数器的值。 如果EXTCNTEN(TIMERx_CTL[24])为1，用户可以读CNT的值来获取当前24位事件计数器的值。

定时器捕获数据寄存器(TIMERx_CAP)

寄存器	偏移量	R/W	描述	复位值
TIMER0_CAP	TMR01_BA+0x10	R	定时器0 捕获 数据寄存器	0x0000_0000
TIMER1_CAP	TMR01_BA+0x30	R	定时器1捕获数据寄存器	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	定时器2捕获数据寄存器	0x0000_0000
TIMER3_CAP	TMR23_BA+0x30	R	定时器3捕获数据寄存器	0x0000_0000

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

位	描述	
[31:24]	**Reserved**	保留.
[23:0]	**CAPDAT**	**定时器捕获数据寄存器** 当CAPEN(TIMERx_EXTCTL[3]) 位被置1，CAPFUNCS (TIMERx_EXTCTL[4])位为0，Tx_EXT 管脚的变化与CAPEDGE(TIMERx_EXTCTL[2:1])设定相匹配时，CAPIF (TIMERx_EINTSTS[0])将被置1且当前定时器计数器CNT(TIMERx_CNT[23:0])的值将被自动载入到CAPDAT。

定时器外部控制寄存器(TIMERx_EXTCTL)

寄存器	偏移量	R/W	描述	复位值
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	定时器0 外部控制寄存器	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x34	R/W	定时器1 外部控制寄存器	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	定时器2 外部控制寄存器	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x34	R/W	定时器3 外部控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**ECNTSSEL**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**INTERCAPSEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTDBEN	**CAPDBEN**	**CAPIEN**	**CAPFUNCS**	**CAPEN**	**CAPEDGE**		**CNTPHASE**

位	描述	
[31:17]	**Reserved**	保留.
[16]	**ECNTSSEL**	**事件计数触发源选择** 0 =事件计数输入源于TMx (x= 0~3)管脚 1 =事件计数输入源于 USB 内部 SOF信号
[15:11]	**Reserved**	保留.
[10:8]	**INTERCAPSEL**	**内部捕获源选择** 000 = 捕获功能触发源于ACMP0输出信号 001 =捕获功能触发源于ACMP1输出信号 101 =捕获功能触发源于LIRC输出信号 其他 =保留 **注:** 这些位只有当CAPSRC (TIMERx_CTL[16])为1时有效
[7]	**CNTDBEN**	**定时器外部计数输入管脚防抖动使能位** 0 = Tx (x= 0~3)管脚防抖禁用 1 = Tx (x= 0~3)管脚防抖使能 **注:** 如果该位使能，Tx管脚边沿检测带防抖动电路
[6]	**CAPDBEN**	**定时器外部捕获输入管脚防抖动使能位** 0 = Tx_EXT (x= 0~3)管脚防抖禁用或ACMP输出防抖禁用

		1 = Tx_EXT (x= 0~3)管脚防抖使能或ACMP输出防抖使能 **注**：如果该位使能，Tx_EXT或ACMP输出管脚边沿检测带防抖电路。
[5]	**CAPIEN**	**定时器外部捕获中断使能位** 0 = Tx_EXT (x= 0~3) 管脚，LIRC或ACMP 检测中断禁止. 1 = Tx_EXT (x= 0~3)管脚，LIRC或ACMP 检测中断使能 **注**： CAPIEN 用于使能定时器外部中断。如果 CAPIEN 使能，当 CAPIF (TIMERx_EINTSTS[0])标志设为1时，定时器会产生一个外部捕获中断信号。 例如, 当 CAPIEN = 1, CAPEN = 1, 且 CAPEDGE = 00, Tx_EXT 管脚或ACMP输出的由1到0变化，将 导致CAPIF 被置位，然后中断信号产生并发送到NVIC告知CPU
[4]	**CAPFUNCS**	**捕获功能选择** 0 =外部捕获模式使能. 1 =外部复位模式使能. **注1:** 当 CAPFUNCS为 0, Tx_EXT (x= 0~3) 管脚上的变化将触发保存 24-位定时器(CNT)计数值到CAPDAT域 **注2:** 当 CAPFUNCS为 1, Tx_EXT (x= 0~3) 管脚上的变化将触发保存 24-位定时器(CNT)计数值到CAPDAT域同时CNT值被清零
[3]	**CAPEN**	**定时器外部捕获使能** 该位使能捕获输入功能 0 =捕获输入禁用 1 =捕获输入使能 注：当TMR0/TMR2 INTRGEN使能时TMR1/TMR3 CAPEN 将强制为 1
[2:1]	**CAPEDGE**	**定时器外部捕获管脚边沿检测选择** 00 = Tx_EXT (x= 0~3)管脚，LIRC或ACMPx (x=0~1)上的下降沿将被检测。 01 = Tx_EXT (x= 0~3)管脚，LIRC或ACMPx (x=0~1)上的上升沿将被检测。 10 = Tx_EXT (x= 0~3)管脚，LIRC或ACMPx (x=0~1)上的上升沿或下降沿将被检测。 11 = 保留。
[0]	**CNTPHASE**	**定时器外部计数管脚相位检测选择** 该位表示外部计数管脚Tx (x= 0~3)的相位检测 0 =管脚的下降沿将被统计 1 =管脚的上升沿将被统计

定时器外部中断状态寄存器(TIMERx_EINTSTS)

寄存器	偏移量	R/W	描述	复位值
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	定时器0 外部中断状态寄存器	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x38	R/W	定时器1 外部中断状态寄存器	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	定时器2 外部中断状态寄存器	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x38	R/W	定时器3 外部中断状态寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留.
[0]	**CAPIF**	**定时器外部捕捉中断标志位** 该位表示定时器外部捕捉中断标志状态。. 当TEXEN (TEXCON[3]使能，且TMx_EXT管脚选择为外部捕捉功能，在TMx_EXT管脚边沿转变与TEX_EDGE (TEXCON[2:1]) 设定相匹配时，该标志将被硬件置位。 0 = Tx_EXT (x= 0~3)管脚中断没有发生. 1 = Tx_EXT (x= 0~3)管脚中断发生. **注1**：该位写 1 清零。 **注2:** 当 CAPEN (TIMERx_EXTCTL[3]) 位被置1, CAPFUNCS (TIMERx_EXTCTL[4]) 位 为 0, Tx_EXT (x= 0~3) 管脚的变化与CAPEDGE (TIMERx_EXTCTL[2:1]) 设定匹配, 该位将被硬件置1. **注3:**CPU 清 CAPIF 状态前会检测到一个新的捕捉事件.如果以上条件发生, 定时器将保持TIMERx_CAP的值不变并丢掉新的捕捉值.

6.8 看门狗定时器(WDT)

6.8.1 概述

看门狗定时器用于，当系统运行到一个未知状态时，通过它来使系统复位。这种做法可以预防系统进入到无限期的死循环。此外，看门狗定时器支持系统从空闲/掉电模式唤醒。

6.8.2 特性

- 20位的向上计数器用于WDT超时间隔设置
- 溢出时间间隔(2^4 ~ 2^{20}) 个WDT_CLK 时钟周期可选，如WDT_CLK = 38.4 kHz (LIRC)，那么溢出时间间隔是416us ~ 27.3秒
- 系统复位保持时间(1 / WDT_CLK) * 63
- 支持看门狗定时器复位延时周期, 包括1026、130、18 或 3 个WDT_CLK复位延时周期
- 通过设置CONFIG0中CWDTEN[2:0]位，支持芯片上电或复位条件下看门狗强制打开
- 如果时钟源选择内部低速时钟（LIRC）或LXT时钟，支持看门狗定时器溢出唤醒

6.8.3 框图

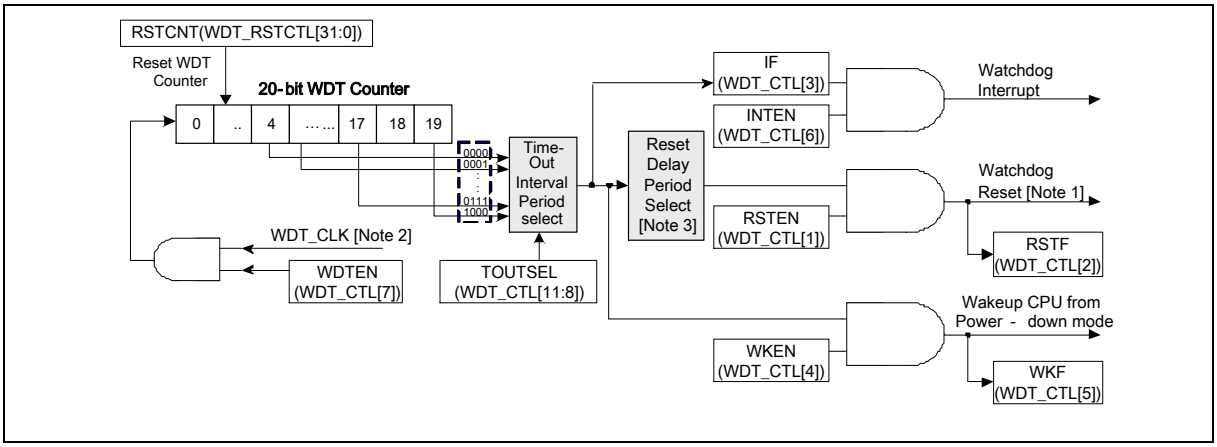


图 6.8-1 看门狗定时器框图

注1: 看门狗定时器复位CPU将持续63个WDT_CLK。

注2: 如果看门狗定时器时钟源选择内部LXT或LIRC时钟源，CPU可以被看门狗定时溢出中断信号唤醒。

注3: 看门狗定时器复位延时周期可以选择3/18/130/1026 个WDT_CLK。

6.8.4 基本配置

- 时钟源配置
  - 通过WDTSEL(CLK_CLKSEL1[1:0])选择WDT 外设时钟源
  - 通过WDTCKEN(CLK_APBCLK0[0])使能WDT外设时钟
  - 通过CWDTEN[2:0] (CWDTEN[2] 是Config0[31]，CWDTEN[1:0]是Config0[4:3])配置在芯片上电或复位后强制使能WDT控制器
  - 如果CWDTEN[2:0]为111，WDT时钟源可以被改变。

WDT时钟控制如 图 6.8-2所示。

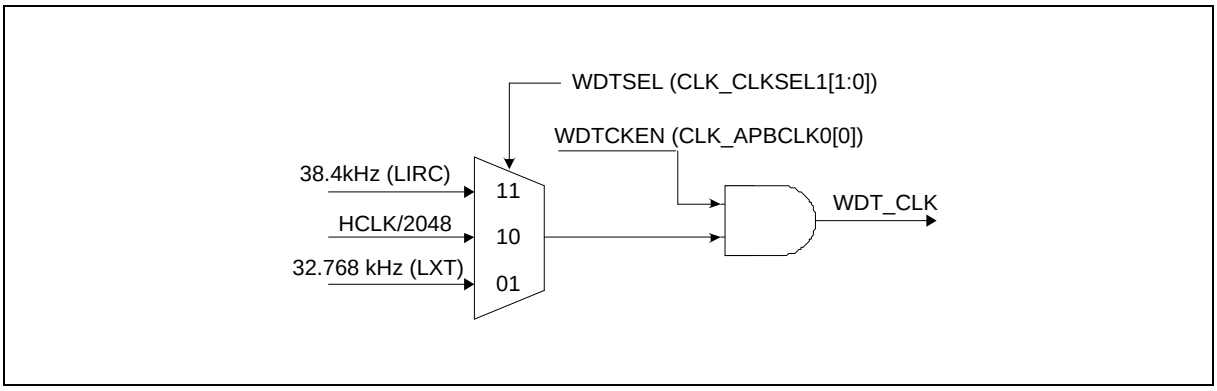


图 6.8-2 看门狗定时器时钟控制

6.8.5 功能描述

看门狗定时器(WDT)包含一个20位的可设置溢出时间间隔的向上计数器。表 6.8-1展示了WDT溢出时间间隔选择，图 6.8-3 展示看门狗定时器溢出时间间隔和复位周期。

6.8.5.1 WDT定时溢出中断

对WDTEN(WDT_CTL[7])置位可以使能WDT功能，然后WDT计数器向上计数。SYNC(WDT_CTL[30])表示使能/禁止WDTEN功能是否完成。通过设置TOUTSEL(WDT_CTL[11:8])，可以选择8个不同的溢出时间间隔。当WDT计数器计到TOUTSEL(WDT_CTL[11:8])设定值，WDT产生溢出中断，然后IF(WDT_CTL[3])标志立即被置1。如果INTEN(WDT_CTL[6])使能，产生WDT超时中断。

6.8.5.2 WDT复位延时周期和复位系统

如果(WDT_CTL[3])标志被置位后，还会有一个固定的延时周期TRSTD，用户应在TRSTD延时计完之前设置WDT_RSTCNT来复位20位的WDT向上计数器的值，以防止产生WDT时间溢出复位信号。此外，用户还应该设置RSTDSEL(WDT_ALTCTL[1:0])选择复位延时周期用于清看门狗定时器，如TRSTD时间计满以后，WDT向上计数器的值仍没有被清除，RSTF(WDT_CTL[2])标志会被置1，若RSTEN(WDT_CTL[1])位使能，芯片立即复位。参考图 6.8-3。TRST复位周期将保持至少63个WDT时钟周期，然后芯片从复位向量地址(0x0000_0000)重新开始执行程序。芯片由WDT定时溢出复位后，RSTF(WDT_CTL[2])标志将会保持1。用户可以通过检查该标志来确认系统是否因WDT定时溢出复位。

TOUTSEL	超时间隔周期 T_{TIS}	复位延时周期 T_{RSTD}
000	$2^4 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
001	$2^6 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
010	$2^8 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
011	$2^{10} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
100	$2^{12} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
101	$2^{14} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
110	$2^{16} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
111	$2^{18} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
1000	$2^{20} * T_{WDT}$	(3/18/130/1026) * T_{WDT}

表 6.8-1 看门狗定时器定时溢出间隔周期选择

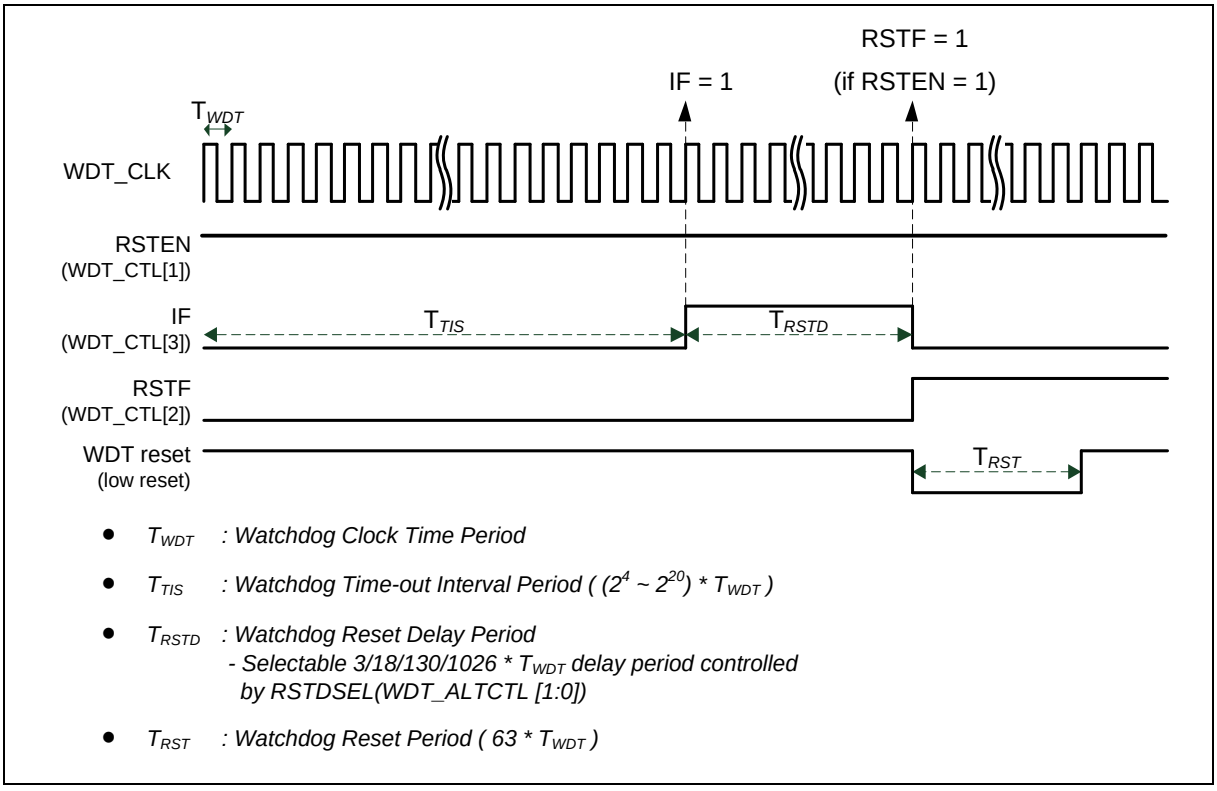


图 6.8-3 看门狗定时器定时溢出间隔和复位周期时序图

6.8.5.3 WDT 唤醒

如果看门狗时钟源选择为LIRC或LXT，如果WKEN(WDT_CTL[4])位被使能，看门狗定时溢出中断产生后，系统能从Power-down模式下被唤醒。注意，用户需在系统进入掉电模式前，设置LXTEN(CLK_PWRCTL[1])或LIRCEN(CLK_PWRCTL[3])选择时钟源，因为当系统进入掉电模式后外设时钟将被禁止。同时，WKF(WDT_CTL[5])标志会被自动置1。用户可以通过查询WKF(WDT_CTL[5])标志知道系统是否是被看门狗定时溢出中断唤醒。

6.8.5.4 WDT ICE 调试

当ICE连接到 MCU，WDT计数器是否还在计数由ICEDEBUG(WDT_CTL[31])决定。ICEDEBUG默认值是0，当CPU被ICE停住，WDT停止计数。如果ICEDEBUG置1，无论CPU是否被ICE停住，WDT将保持计数。

6.8.6 寄存器映射

R:只读, **W**: 只写, **R/W**: 读/写

寄存器	偏移	R/W	描述	复位值
WDT 基地址： **WDT_BA = 0x4004_0000**				
WDT_CTL	WDT_BA+0x00	R/W	WDT 控制寄存器	0x0000_0800
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT 选择控制寄存器	0x0000_0000
WDT_RSTCNT	WDT_BA+0x08	W	WDT 复位计数器寄存器	0x0000_0000

6.8.7 寄存器描述

WDT控制寄存器(WDT_CTL)

寄存器	偏移	R/W	描述	复位值
WDT_CTL	WDT_BA+0x00	R/W	WDT 控制寄存器	0x0000_0800

31	30	29	28	27	26	25	24
ICEDEBUG	**SYNC**	**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**TOUTSEL**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDTEN	**INTEN**	**WKF**	**WKEN**	**IF**	**RSTF**	**RSTEN**	**Reserved**

位	描述	
[31]	**ICEDEBUG**	**ICE调试模式下看门狗计数控制位(写保护)** 0 = ICE 调试模式影响看门狗计数 当ICE让CPU运行停止，看门狗计数器也停止 1 = ICE 调试模式不影响看门狗计数 不论CPU受ICE停止还是运行，看门狗继续计数 **注:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器
[30]	**SYNC**	**WDT 使能控制 SYNC 标志指示 (只读)** 如果用户使能/禁止WDTEN (WDT_CTL[7])，该标志可以指示使能/禁止WDTEN操作是否完成。 0 = 设置 WDTEN 位完成 1 = 设置 WDTEN位正在同步，还未生效 **注:** 执行使能/禁止 WDTEN 位需要2 * WDT_CLK 个时钟周期来生效
[29:12]	**Reserved**	保留
[11:8]	**TOUTSEL**	**看门狗定时器溢出时间间隔选择 (写保护)** 以下四个位用于选择看门狗定时溢出周期 0000 = 2^4 * WDT_CLK 0001 = 2^6 * WDT_CLK 0010 = 2^8 * WDT_CLK 0011 = 2^{10} * WDT_CLK 0100 = 2^{12} * WDT_CLK. 0101 = 2^{14} * WDT_CLK 0110 = 2^{16} * WDT_CLK 0111 = 2^{18} * WDT_CLK 1000 = 2^{20} * WDT_CLK

		注: 该位为写保护位，详情请参考SYS_REGLCTL寄存器
[7]	**WDTEN**	**WDT 使能位(写保护)** 0 = WDT 禁用 (该操作会复位看门狗计数器的值). 1 = WDT 使能 **注1:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器. **注2:** 如果CWDTEN[2:0] (Config0[31] 和 Config0[4:3]的组合) 位的值非111, 该位会被强制置1，且用户不能把它改为0.
[6]	**INTEN**	**WDT溢出中断使能控制(写保护)** 如果该位使能,看门狗定时溢出后会产生中断信号，并告知CPU 0 = 看门狗定时溢出中断禁止 1 = 看门狗定时溢出中断使能 **注:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器.
[5]	**WKF**	**WDT溢出唤醒标志 (写保护)** 该位表明看门狗定时器中断唤醒标志的状态 0 = 看门狗定时器没有导致芯片唤醒 1 = 芯片从Idle或Power-down 模式被唤醒 **注1:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器. **注2:** 该位写1清0
[4]	**WKEN**	**WDT溢出唤醒功能控制位(写保护)** 如果此位被置1, 当IF (WDT_CTL[3])被置1且INTEN(WDT_CTL[6])被使能，看门狗定时溢出中断信号将会触发唤醒MCU 0 = 唤醒触发事件禁止，即便看门狗定时器定时溢出中断发生 1 = 唤醒触发事件使能，如果看门狗定时溢出中断产生，将唤醒MCU **注1:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器 **注2:** MCU能被看门狗定时器定时溢出中断信号唤醒的条件是看门狗定时器时钟源必须为内部38.4 kHz低速RC振荡器(LIRC) 或 LXT
[3]	**IF**	**WDT 超时中断标志** 当看门狗定时器计数器的值达到溢出时间间隔，该位将被置1 0 = 没有发生看门狗定时溢出中断 1 = 有看门狗定时溢出中断发生 **注:** 该位写1清0
[2]	**RSTF**	**WDT 超时复位标志** 该位用来指示系统是否因看门狗定时器定时溢出发生复位 0 = 没有发生看门狗定时溢出复位 1 = 发生了看门狗定时溢出复位 **注:** 该位写1清0
[1]	**RSTEN**	**WDT 溢出复位使能位(写保护)** 如果看门狗计数器的值达到设定值，且指定的WDT复位延时时间已经计完，计数器值还没有被清零，如果该位被设置，则会使能看门狗定时溢出复位 0 = 看门狗定时溢出复位功能禁止 1 =看门狗定时溢出复位功能使能 **注:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器.
[0]	**Reserved**	保留.

WDT选择控制寄存器(WDT_ALTCTL)

寄存器	偏移	R/W	描述	复位值
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT选择控制寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留
[1:0]	**RSTDSEL**	**WDT复位延时选择(写保护)** 当看门狗定时器发生溢出，用户有一个复位延时时间来通过往RSTCNT (WDT_RSTCNT[31:0])写入0x00005aa5将看门狗计数器清零，以防止看门狗溢出复位。 用户也可以设置RSTDSEL值选择适当的看门狗复位延时时间。 00 = WDT复位延时时间是1026 * WDT_CLK 01 = WDT复位延时时间是130 * WDT_CLK 10 = WDT 复位延时时间是18 * WDT_CLK 11 = WDT 复位延时时间是 3 * WDT_CLK **注1:** 该位为写保护位，详情请参考SYS_REGLCTL寄存器 **注2:** 如果看门狗定时器超时复位发生，该寄存器的值将被清零

WDT复位计数器寄存器(WDT_RSTCNT)

寄存器	偏移	R/W	描述	复位值
WDT_RSTCNT	WDT_BA+0x08	W	WDT复位计数器寄存器	0x0000_0000

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

位	描述	
[31:0]	**RSTCNT**	**WDT复位计数器寄存器** 写0x00005AA5 到该域会复位内部20位WDT向上计数器值为0 **注1:** 执行 RSTCNT 复位计数器需要2 * WDT_CLK 周期生效

6.9 窗口看门狗定时器 (WWDT)

6.9.1 概述

窗口看门狗定时器(WWDT)用于在一个窗口时间内执行系统复位，以防止程序在不可预知条件下跑到一个不可控的状态。

6.9.2 特性

- 6位向下计数值(CNTDAT, WWDT_CNT[5:0])和6位比较值(CMPDAT,WWDT_CTL[21:16])，使得窗口周期更加灵活
- 支持4-位值(PSCSEL, WWDT_CTL[11:8])选择看门狗预分频值，预分频计数器最大可达11位
- WWDT计数器在空闲或掉电模式下暂停

6.9.3 框图

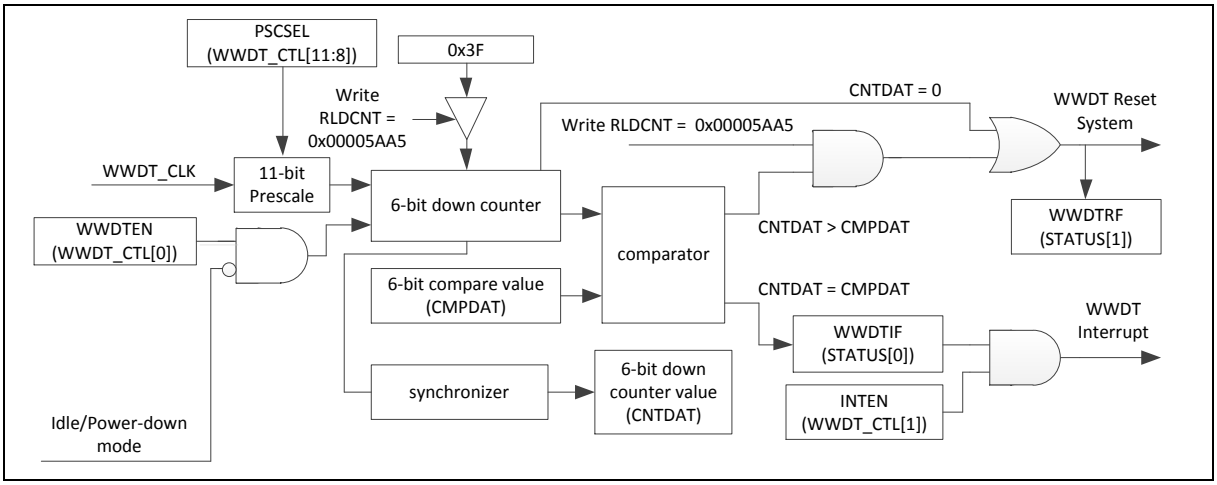


图 6.9-1 WWDT 框图

6.9.4 基本配置

- 时钟源配置
  - 通过WWDTSEL (CLK_CLKSEL1[31:30])选择WWDT外设时钟源
  - 通过WDTCKEN (CLK_APBCLK0[0])使能WWDT外设时钟

WWDT时钟控制如图 6.9-2所示.

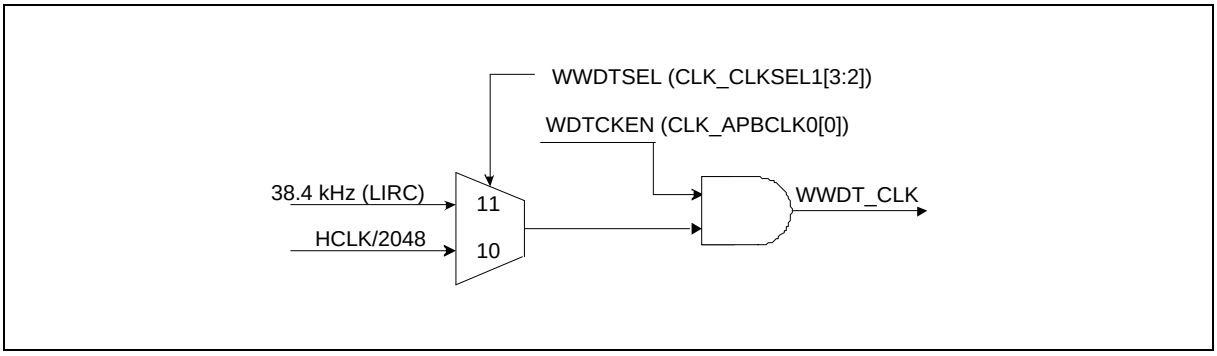


图 6.9-2 WWDT 时钟控制

6.9.5 功能描述

窗口看门狗定时器(WWDT)是一个6位向下计数器，该计数器带一个可选择预分频值，不同的预分频值对应不同的看门狗定时溢出时间。6位窗口看门狗定时器的时钟源可以是系统时钟经2048 (HCLK/2048)分频的时钟，也可以是内部38.4 kHz低速RC振荡时钟源。看门狗的时钟源带一个可选择的11位预分频值，该值可通过PSCSEL (WWDT_CTL[11:8])位来设置选择，对应预分频值如表 6.9-1 所示.

PSCSEL	预分频值	最大定时溢出周期	最大定时溢出间隔 (WWDT_CLK=38.4 kHz)
0000	1	1 * 64 * T_{WWDT}	1.666667 ms
0001	2	2 * 64 * T_{WWDT}	3.33333 ms
0010	4	4 * 64 * T_{WWDT}	6.6667 ms
0011	8	8 * 64 * T_{WWDT}	13.333 ms
0100	16	16 * 64 * T_{WWDT}	26.667 ms
0101	32	32 * 64 * T_{WWDT}	53.333 ms
0110	64	64 * 64 * T_{WWDT}	106.66 ms
0111	128	128 * 64 * T_{WWDT}	213.33 ms
1000	192	192 * 64 * T_{WWDT}	320 ms
1001	256	256 * 64 * T_{WWDT}	426.66 ms
1010	384	384 * 64 * T_{WWDT}	640 ms
1011	512	512 * 64 * T_{WWDT}	853.33 ms
1100	768	768 * 64 * T_{WWDT}	1.28 s
1101	1024	1024 * 64 * T_{WWDT}	1.706 s
1110	1536	1536 * 64 * T_{WWDT}	2.56 s
1111	2048	2048 * 64 * T_{WWDT}	3.413 s

表 6.9-1 WWDT 预分频值选择

6.9.5.1 WWDT 计数

当WWDTEN (WWDT_CTL[0])位被使能, 窗口看门狗向下计数器将会从0x3F向下递减计数到0. 为了防止程序在非用户指定位置关闭窗口看门狗定时器, 窗口看门狗定时器控制寄存器WWDT_CTL在芯片上电或复位后仅可写一次。当WWDTEN (WWDT_CTL[0])位被软件使能以后，除非芯片复位，否则用户不能进行禁止窗口看门狗定时器 (WWDTEN)、 修改计数器预分频周期(PSCSEL)或修改窗口比较值(CMPDAT)的操作。

为了避免在CPU时钟关闭状态WWDT产生系统复位，当CPU进入空闲或掉电模式，WWDT计数器会停止计数。在CPU进入正常模式后，WWDT计数器才会开始下计数。

6.9.5.2 WWDT 比较匹配中断

窗口看门狗定时器向下计数过程中，当窗口看门狗定时器计数值(CNTDAT) 等于比较值CMPDAT，WWDTIF (WWDT_STATUS[0])会被置1， WWDTIF 可以被软件清零; 如果INTEN (WWDT_CTL[1])位被使能, 当WWDTIF 位被硬件置1，就会产生窗口看门狗比较匹配中断。

6.9.5.3 WWDT 复位系统

图 6.9-3为WWDT三种复位和重载动作情况的框图.

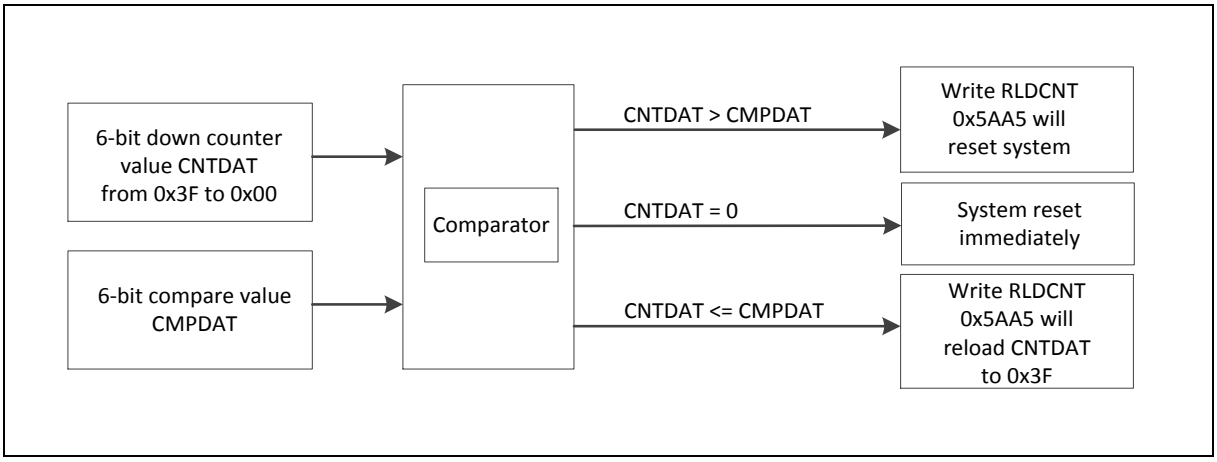


图 6.9-3 WWDT 复位和重载动作

如果当前 CNTDAT (WWDT_CNT[5:0]) 大于 CMPDAT (WWDT_CTL[21:16]) 且用户写0x00005AA5 到 WWDT_RLDCNT 寄存器， WWDT 立即产生复位系统信号，芯片复位。当CNTDAT > CMPDAT ，WWDT重载计数器波形如 图 6.9-4.

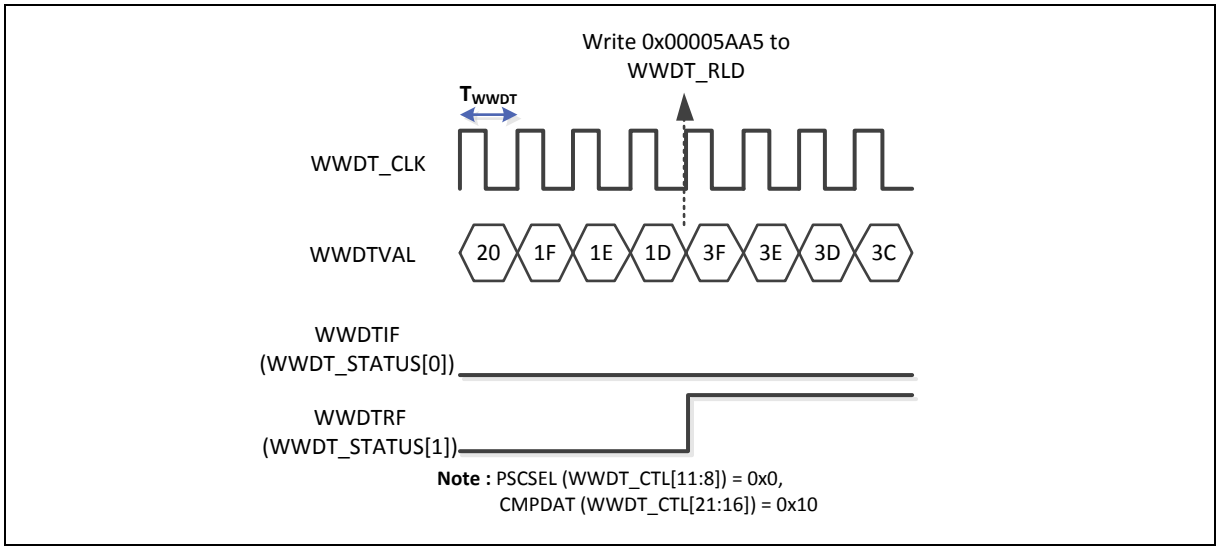


图 6.9-4 当 CNTDAT > CMPDAT WWDT 重载计数器

当 WWDTIF (WWDT_STATUS[0]) 被置位，用户必须通过写0x00005AA5 到 WWDT_RLDCNT，重载WWDT计数器值为0x3F，以防止WWDT计数器计数到0时产生系统复位信号使系统复位。图 6.9-5展示了当CNTDAT < CMPDAT WWDT重载计数器的时序波形。图 6.9-6展示了如果用户在计数器数到0之前还没有写0x00005AA5 到 WWDT_RLDCNT WWDT时，产生复位系统信号(WWDTRF)的时序波形。

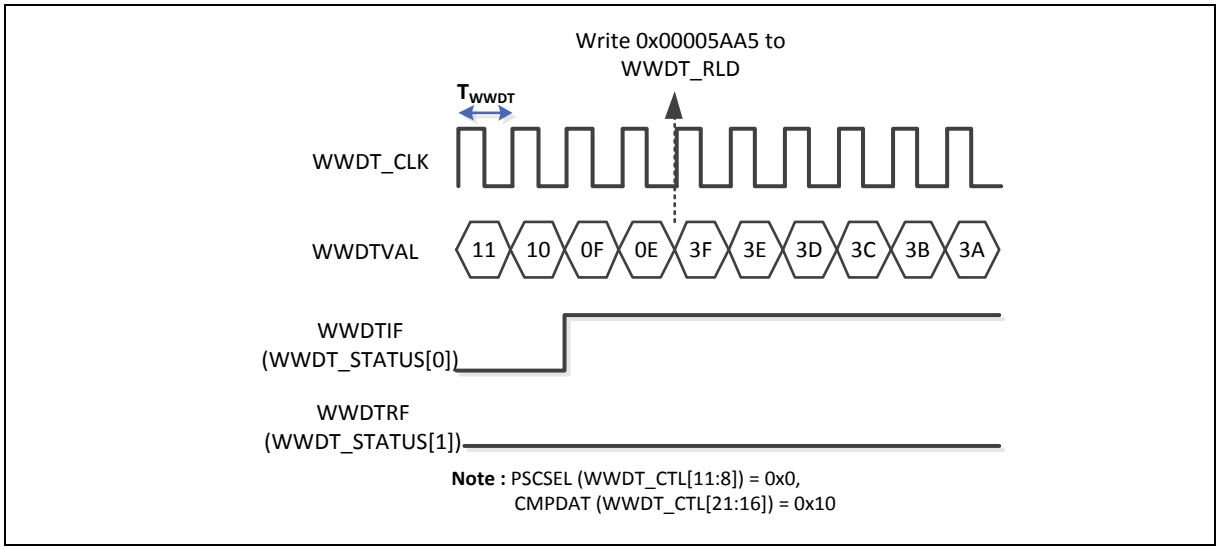


图 6.9-5 当 WWDT_CNT < WINCMP WWDT 重载计数器

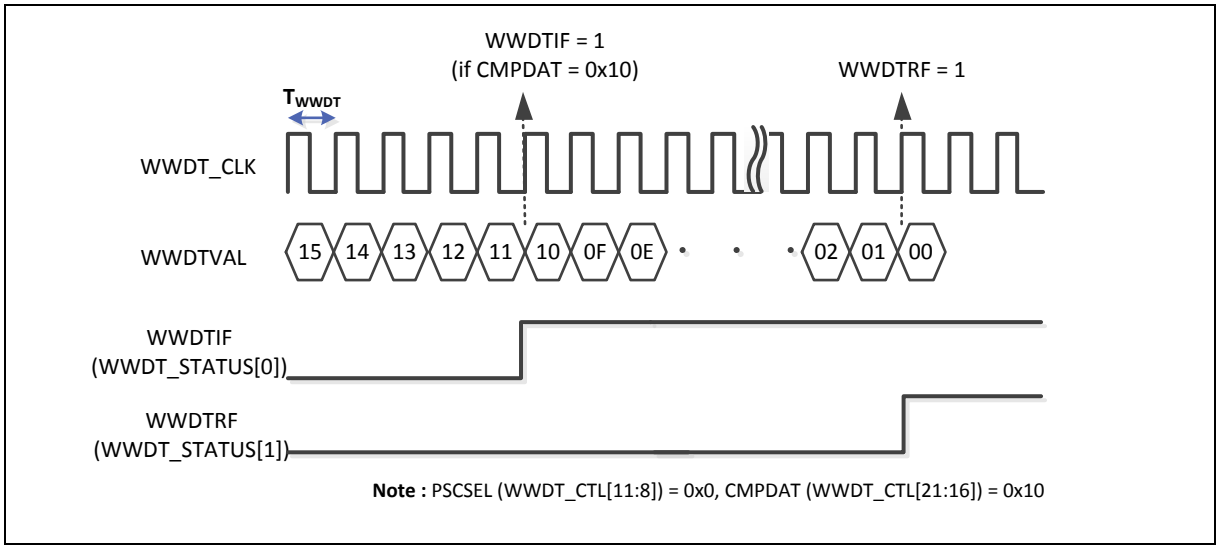


图 6.9-6 WWDT 中断和复位信号

6.9.5.4 WWDT窗口设置限制

当用户对WWDT_RLDCNT寄存器写0x00005AA5来重载WWDT的值到0x3F的时候，大概需要3个窗口看门狗定时器时钟来同步重载命令到实际执行重载操作。这意味着如果用户设置PSCSEL (WWDT_CTL[11:8])的值为0000，计数器预分频值则应该设置为1，CMPDAT (WWDT_CTL[21:16])寄存器的值则必须大于2；否则，当WWDTIF标志产生后，写WWDT_RLDCNT来重载窗口看门狗定时器值为0x3F的操作将无效，WWDTIF (WWDT_STATUS[0])将会产生，窗口看门狗定时器系统复位将发生。WWDT CMPDAT设置限制如表 6.9-2.

如果CMPDATA设置为0x3F或0x0,中断也不会发生。这是因为WWDT计数到0x0就会发生复位,所以当CMPDATA 为 0x0时中断也不会发生。

PSCSEL	预分频值	有效 CMPDAT 值
0000	1	0x3 ~ 0x3E

0001	2	0x2 ~ 0x3E
其它	其它	0x1 ~ 0x3E

表 6.9-2 CMPDAT 设置限制

6.9.5.5 WWDT ICE 调试

当 ICE 连接到 MCU，WWDT 计数器是否计数由ICEDEBUG (WWDT_CTL[31])决定。ICEDEBUG默认值是0，当 CPU 被 ICE停住，WWDT 计数器停止计数。如果ICEDEBUG 置 1, WWDT 计数器会保持计数，无论CPU 被 ICE停住与否。

6.9.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读/写

寄存器	偏移	R/W	描述	复位值
WWDT 基地址: **WWDT_BA = 0x4004_0100**				
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT重载计数器寄存器	0x0000_0000
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT控制寄存器	0x003F_0800
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT状态寄存器	0x0000_0000
WWDT_CNT	WWDT_BA+0x0C	R	WWDT计数器值寄存器	0x0000_003F

6.9.7 寄存器描述

WWDT重载计数器寄存器 (WWDT_RLDCNT)

寄存器	偏移	R/W	描述	复位值
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT重载计数器寄存器	0x0000_0000

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

位	描述	
[31:0]	**RLDCNT**	**WWDT重载计数器寄存器** 对此寄存器写0x00005AA5 将会重载窗口看门狗定时器计数器的值到0x3F. **注:** 用户只能当窗口看门狗计数器的值在0到CMPDAT (WWDT_CTL[21:16])之间才可以写WWDT_RLDCNT寄存器来重载窗口看门狗计数器。如果窗口看门狗计数器的值大于CMPDAT时写WWDT_RLDCNT寄存器，窗口看门狗定时器复位信号会立即产生。

WWDT控制寄存器 (WWDT_CTL)

寄存器	偏移	R/W	描述	复位值
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT控制寄存器	0x003F_0800

注: :此寄存器仅可以在芯片上电或复位后写一次.

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

位	描述	
[31]	**ICEDEBUG**	**ICE调试模式下窗口看门狗计数控制位** 0 =ICE 调试模式下影响窗口看门狗定时器计数，当CPU被ICE暂停后，窗口看门狗向下计数器计数值将保持不变 1 = ICE 调试模式下不影响窗口看门狗定时器计数，不管CPU是否被ICE暂停，窗口看门狗定时器向下计数器将会保持继续计数。
[30:22]	**Reserved**	保留.
[21:16]	**CMPDAT**	**WWDT窗口比较寄存器** 设置此寄存器来调整有效重载窗口. **注：**当前窗口看门狗定时器计数器的值在0到CMPDAT之间时，用户才可以写WWDT_RLDCNT来重载窗口看门狗定时器计数器的值。如果当前窗口看门狗定时器计数器的值大于CMPDAT时用户写WWDT_RLDCNT寄存器，窗口看门狗定时器复位信号会立即产生。
[15:12]	**Reserved**	保留.
[11:8]	**PSCSEL**	**WWDT计数器预分频周期选择** 0000 =预分频为1; 最大定时溢出周期是1 * 64 * WWDT_CLK. 0001 =预分频为2; 最大定时溢出周期是2 * 64 * WWDT_CLK. 0010 =预分频为4; 最大定时溢出周期是4 * 64 * WWDT_CLK. 0011 =预分频为8; 最大定时溢出周期是8 * 64 * WWDT_CLK. 0100 =预分频为16; 最大定时溢出周期是16 * 64 * WWDT_CLK. 0101 =预分频为32; 最大定时溢出周期是32 * 64 * WWDT_CLK. 0110 =预分频为64; 最大定时溢出周期是64 * 64 * WWDT_CLK. 0111 =预分频为128; 最大定时溢出周期是128 * 64 * WWDT_CLK. 1000 =预分频为192; 最大定时溢出周期是192 * 64 * WWDT_CLK. 1001 =预分频为256; 最大定时溢出周期是256 * 64 * WWDT_CLK. 1010 =预分频为384; 最大定时溢出周期是384 * 64 * WWDT_CLK.

		1011 =预分频为512; 最大定时溢出周期是512 * 64 * WWDT_CLK.. 1100 =预分频为768; 最大定时溢出周期是768 * 64 * WWDT_CLK. 1101 =预分频为1024; 最大定时溢出周期是1024 * 64 * WWDT_CLK. 1110 =预分频为1536; 最大定时溢出周期是1536 * 64 * WWDT_CLK. 1111 =预分频为2048; 最大定时溢出周期是2048 * 64 * WWDT_CLK.
[7:2]	**Reserved**	保留.
[1]	**INTEN**	**WWDT中断使能位** 该位被使能后，如果有WWDT计数器比较匹配中断信号产生，该中断就会通知给CPU 0 = WWDT计数器比较匹配中断禁止 1 = WWDT计数器比较匹配中断使能
[0]	**WWDTEN**	**WWDT 使能位** 0 = WWDT计数器停止. 1 = WWDT计数器计数开始

WWDT状态寄存器 (WWDT_STATUS)

寄存器	偏移	R/W	描述	复位值
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT状态寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留.
[1]	**WWDTRF**	**WWDT 超时复位标志** 该位指示系统是否已被窗口看门狗定时器溢出复位 0 =窗口看门狗定时器超时复位没有发生 1 =窗口看门狗定时器超时复位发生 **注:** 该位写1清0
[0]	**WWDTIF**	**WWDT比较匹配中断标志** 该位指示当WWDT 的计数值与CMPDAT (WWDT_CTL[21:16])匹配时，窗口看门狗定时器的中断状态标志 0 =无影响 1 =窗口看门狗定时器计数器的值与CMPDAT寄存器的值匹配 **注:** 该位写1清0

WWDT计数器值寄存器 (WWDT_CNT)

寄存器	偏移	R/W	描述	复位值
WWDT_CNT	WWDT_BA+0x0C	R	WWDT计数器值寄存器	0x0000_003F

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

位	描述	
[31:6]	**Reserved**	保留.
[5:0]	**CNTDAT**	**WWDT计数器值** CNTDAT将会被持续更新，以监视6位向下计数器的值

6.10 实时时钟(RTC)

6.10.1 概述

实时时钟 (RTC) 控制器用于记录实时时间及日历等信息。RTC控制器支持可配置的时间节拍和闹钟定时中断。时间及日历等信息的表示格式为BCD 码。RTC控制器支持RTC时钟源频率补偿功能，可对外接晶振的频率精度进行数字频率补偿。

6.10.2 特性

- 支持时间计数（秒、分、时）和日历计数（日、月、年），用户可以通过访问寄存器RTC_TIME和RTC_CAL查看时间及日历。
- 可设定闹钟时间（秒、分、时）和日历（日、月、年），参看寄存器RTC_TALM 和RTC_CALM。
- 可设定闹钟时间（秒、分、时）和日历（日、月、年）的掩码使能功能，参看RTC_TAMSK和 RTC_CAMSK寄存器。
- 可选择12小时或 24小时制式，参看RTC_CLKFMT寄存器。
- 支持闰年自动识别，参看RTC_LEAPYEAR寄存器
- 支持周内日期计数，参看RTC_WEEKDAY寄存器
- 支持RTC时钟源频率补偿功能，参看寄存器RTC_FREQADJ
- 所有时间、日期的数据格式为 BCD 码
- 支持周期RTC时间节拍中断，提供 8个周期选项供选择，分别为：1/128、1/64、1/32、1/16、1/8、1/4、1/2 及1秒
- 支持 RTC 定时节拍和闹钟定时中断
- 支持1Hz时钟输出
- 支持RTC中断从空闲模式或掉电模式下唤醒芯片

6.10.3 框图

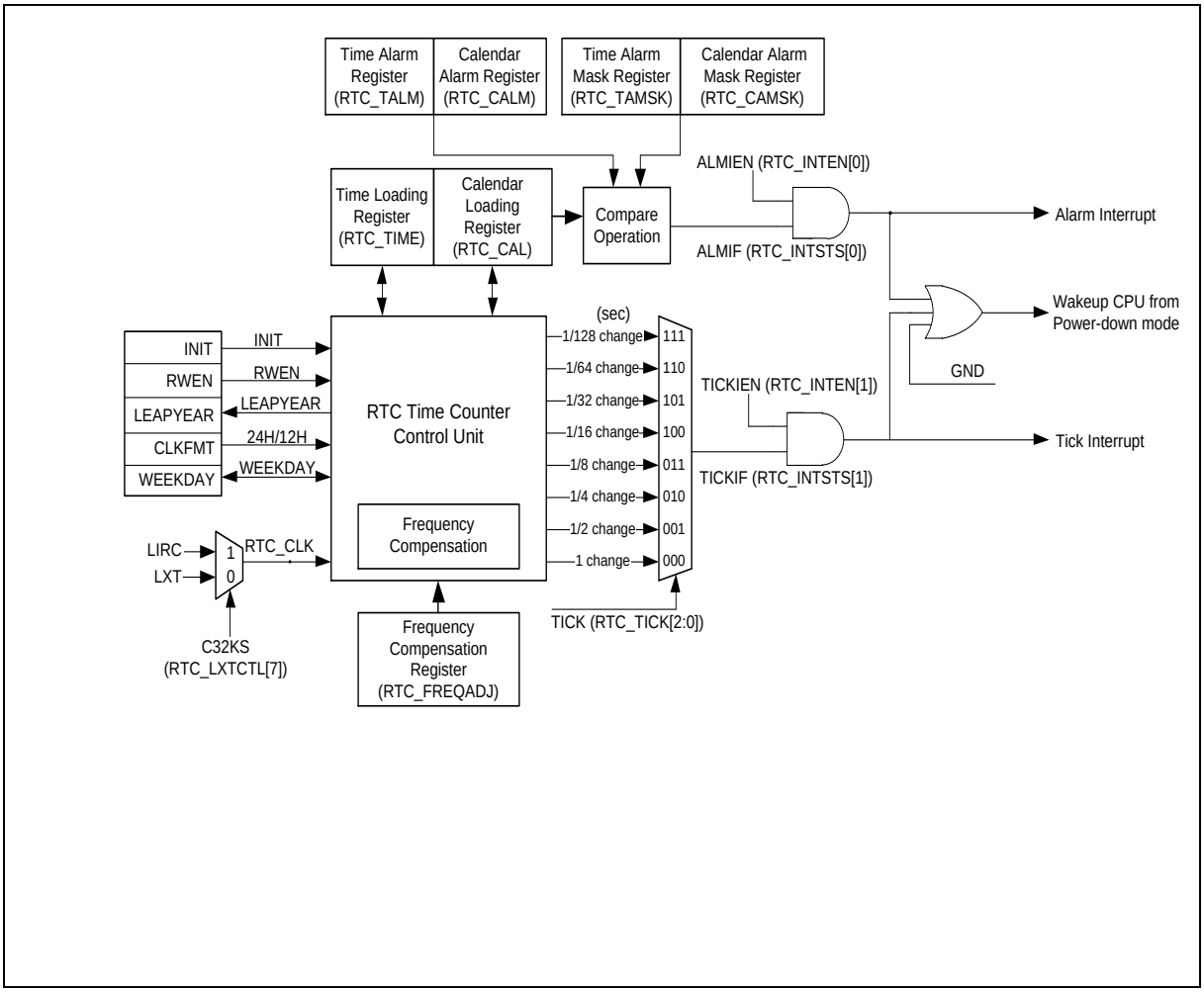


图 6.10-1 RTC 框图

6.10.4 基本配置

- 时钟源配置
  - RTC控制器时钟源通过RTCCKEN (APBCLK0[1])使能， RTC 时间计数器时钟源通过 RTC_LXTCTL[7] 选择LXT 或 LIRC
- 管脚配置

组	管脚名称	GPIO	MFP
X32	X32_OUT	PF.4	MFP10
X32	X32_IN	PF.5	MFP10

6.10.5 功能描述

6.10.5.1 RTC初始化

当 RTC 模块上电后，RTC 处于 reset 状态。用户需要写入数据 (0x a5eb1357) 到RTC 初始化寄存器

INIT(RTC_INIT[31:0])中，让 RTC 离开 reset 状态。一旦INIT寄存器被写入 0xa5eb1357，RTC 将会永远处于Active状态。用户可以读ACTIVE(RTC_INIT[0])来判断RTC是处于active状态或是reset状态。

RTC控制寄存器访问特征如表 6.10-1所示。

寄存器	激活(RTC_INIT[0]) = 0时	激活(RTC_INIT[0]) = 1时
RTC_INIT	读/写（注 1）	只读
RTC_FREQADJ	读/写	读/写
RTC_TIME	无效	读/写
RTC_CAL	无效	读/写
RTC_CLKFMT	无效	读/写
RTC_WEEKDAY	无效	读/写
RTC_TALM	无效	读/写
RTC_CALM	无效	读/写
RTC_INTEN	读/写	读/写
RTC_INTSTS	只读	读/写
RTC_TICK	无效	读/写
RTC_TAMSK	无效	读/写
RTC_CAMSK	无效	读/写

表 6.10-1 RTC 读/写使能

注1： INIT位可以执行写操作，INIT[0]只能执行读操作。

6.10.5.2 频率补偿

RTC_FREQADJ寄存器允许用户对时钟输入做数字补偿。请遵循如下例子和公式写32K晶振的实际频率到RTC_FREQADJ寄存器。如下是高于或低于32768 Hz的补偿例子。

例1:

频率计数器测量：32773.65 Hz

整数部分：32773 => RTC_FREQADJ[12:8] = 0x15,参考INTEGER(RTC_FREQADJ[13:8])获取详细设置值。

小数部分：0.65 X 64 = 41.6(0x2A) => RTC_FREQADJ[5:0]=0x2A

例2:

频率计数器测量：32763.25 Hz

整数部分：32763=> RTC_FREQADJ[12:8] = 0x0B, 参考INTEGER(RTC_FREQADJ[13:8]) 获取详细设置值。

小数部分：0.25 X 64 = 16(0x10) => RTC_FREQADJ[5:0] = 0x10

注：如果补偿没有执行，RTC_FREQADJ 寄存器为默认值(0x0000_1000)。用户可以在生产时通过时钟

输出功能，使用频率计数器测量RTC时钟源。同时，用户可以使用时钟输出功能检查RTC频率补偿的结果。

6.10.5.3 时间和日历计数器

RTC_TIME 和 RTC_CAL 用于加载实时时间和日期。 RTC_TALM 和 RTC_CALM 用于设置闹钟时间和日期。

6.10.5.4 12/24 小时制选择

设置24HEN (RTC_CLKFMT[0]) 选择12/24小时时标制式。当RTC运行在12小时制模式时，RTC_TIME[21]（TENHR[1:0])中的高位代表AM/PM提示符）。（如果RTC_TIME[21]=1，表示目前为PM时间，RTC_TIME[21]=0，表示目前为AM时间）。表 6.10-2为RTC_TIME 12/24小时制时间转换映射表。

注：小时信息写入到RTC_TIME[21:16]，信息格式为BCD格式			
24小时制 (24HEN = 1)		12小时制 (PM 时间 + 0x20) (24HEN = 0) (PM 时间 + 0x20)	
0x00 (AM12)	0x12 (PM12)	0x12 (AM12)	0x32 (PM12)
0x01 (AM01)	0x13 (PM01)	0x01 (AM01)	0x21 (PM01)
0x02 (AM02)	0x14 (PM02)	0x02 (AM02)	0x22 (PM02)
0x03 (AM03)	0x15 (PM03)	0x03 (AM03)	0x23 (PM03)
0x04 (AM04)	0x16 (PM04)	0x04 (AM04)	0x24 (PM04)
0x05 (AM05)	0x17 (PM05)	0x05 (AM05)	0x25 (PM05)
0x06 (AM06)	0x18 (PM06)	0x06 (AM06)	0x26 (PM06)
0x07 (AM07)	0x19 (PM07)	0x07 (AM07)	0x27 (PM07)
0x08 (AM08)	0x20 (PM08)	0x08 (AM08)	0x28 (PM08)
0x09 (AM09)	0x21 (PM09)	0x09 (AM09)	0x29 (PM09)
0x10 (AM10)	0x22 (PM10)	0x10 (AM10)	0x30 (PM10)
0x11 (AM11)	0x23 (PM11)	0x11 (AM11)	0x31 (PM11)

表 6.10-2 12/24 小时制选择

6.10.5.5 星期计数器

RTC控制器提供一周日期寄存器WEEKDAY (RTC_WEEKDAY [2：0])，其值由0至6，用于表示周日至周六。

6.10.5.6 时间周期节拍中断

通过设置TICK（RTC_TICK [2:0]）来选择周期中断，周期中断有8个选项：1/128、1/64、1/32、1/16、1/8、1/4、1/2以及1秒。当TICKIEN (RTC_INTEN[1]) 被置 1，周期中断使能后，MCU根据RTC_TICK[2:0]寄存器内设定的值周期性的发生中断

6.10.5.7 闹钟中断

当RTC_TIME和RTC_CAL中的值等于RTC_TALM和RTC_CALM中的设定值，如果此时闹钟中断已设为使能 (ALMIEN (RTC_INTEN[0])=1)，则闹钟中断标志ALMIF (RTC_INTSTS[0])被置位同时发出闹钟中断请求。

RTC控制器提供时间屏蔽寄存器（RTC_TAMSK）和日历闹钟屏蔽寄存器（RTC_CAMSK），用于屏蔽指定数字并产生周期中断而无需改动每个闹钟中断服务程序中的闹钟匹配条件（匹配条件根据RTC_TALM 和 RTC_CALM中配置决定）。

6.10.5.8 1 Hz时钟输出

RTC控制器提供1Hz时钟输出到CLKO功能管脚。用户可以设置CLK1HZEN (CLK_CLKOCTL[6])为1并使能RTC，1Hz时钟就会从CLKO功能管脚输出。

6.10.5.9 应用指南

1. RTC_TALM、RTC_CALM、RTC_TIME 和 RTC_CAL寄存器中数据格式为BCD 格式。
2. 用户必须保证载入值为有效合理值。例如，RTC_CAL = 201a (年)、13 (月)、00 (日)或RTC_CAL与RTC_WEEKDAY不匹配、等等，这些值均为不合理值。
3. 在RTC_CAL 和 RTC_CALM 中，仅 2 位 BCD 码用于指示“年”。目前仅支持20xy年份，不支持19xy 或 21xy年。
4. 12小时制时间设定范例
   如果RTC时间为PM12:59:30，RTC_TIME设定为:
   1) HOUR:
      RTC_TIME[21:16]: 0x32 (0x12+0x20)，可表示为：TENHR (RTC_TIME[21:20]) = 0x3, HR (RTC_TIME[19:16]) = 0x2。
   2) MIN:
      RTC_TIME[14:8]: 0x59 可表示为：TENMIN (RTC_TIME[14:12]) =0x5, MIN (RTC_TIME[11:8]) = 0x9。
   3) SEC:
      RTC_TIME[6:0]: 0x30 ，可表示为：TENSEC (RTC_TIME[6:4]) = 0x3, SEC (RTC_TIME[3:0]) = 0x0。

表 6.10-3为在内核和电池初次上电后的寄存器值。

寄存器	复位状态
RTC_INIT	0
RTC_CAL	15/8/8 (年/月/日)
RTC_TIME	00:00:00（小时：分钟:秒）
RTC_CALM	00/00/00 (年/月/日)
RTC_TALM	00:00:00（小时：分钟:秒）
RTC_CLKFMT	1（24小时制）
RTC_WEEKDAY	6（星期六）
RTC_INTEN	0

RTC_INTSTS	0
RTC_LEAPYEAR	0
RTC_TICK	0

表 6.10-3 上电后寄存器值

6.10.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读/写

寄存器	偏移	R/W	描述	复位值
RTC基地址: **RTC_BA = 0x4004_1000**				
RTC_INIT	RTC_BA+0x00	R/W	RTC初始化寄存器	0x0000_0000
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC频率补偿寄存器	0x0000_1000
RTC_TIME	RTC_BA+0x0C	R/W	时间载入寄存器	0x0000_0000
RTC_CAL	RTC_BA+0x10	R/W	日历载入寄存器	0x0015_0808
RTC_CLKFMT	RTC_BA+0x14	R/W	时间制式（时标）选择寄存器	0x0000_0001
RTC_WEEKDAY	RTC_BA+0x18	R/W	星期寄存器	0x0000_0006
RTC_TALM	RTC_BA+0x1C	R/W	时间闹钟寄存器	0x0000_0000
RTC_CALM	RTC_BA+0x20	R/W	日历闹钟寄存器	0x0000_0000
RTC_LEAPYEAR	RTC_BA+0x24	R	闰年指示寄存器	0x0000_0000
RTC_INTEN	RTC_BA+0x28	R/W	RTC中断使能寄存器	0x0000_0000
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC中断指示寄存器	0x0000_0000
RTC_TICK	RTC_BA+0x30	R/W	RTC时钟节拍寄存器	0x0000_0000
RTC_TAMSK	RTC_BA+0x34	R/W	RTC时间闹钟屏蔽寄存器	0x0000_0000
RTC_CAMSK	RTC_BA+0x38	R/W	RTC日历闹钟屏蔽寄存器	0x0000_0000
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz晶振控制寄存器	0x0000_0000

6.10.7 寄存器描述

RTC初始化寄存器(RTC_INIT)

寄存器	偏移	R/W	描述	复位值
RTC_INIT	RTC_BA+0x00	R/W	RTC初始化寄存器	0x0000_0000

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

位	描述	
[31:1]	**INIT[31:1]**	**RTC 初始化(只写)** 当 RTC 模块上电后，RTC 处于复位状态。用户需要写入数据 (0xa5eb1357) 到 INIT 使得 RTC 离开复位状态。一旦 INIT 被写入 0xa5eb1357，RTC 将会一直处于非复位状态。 INIT寄存器的这些位只能被写，如果读的话，总是为0。
[0]	**INIT[0]/ACTIVE**	**RTC 激活状态（只读）** 0 = RTC 在复位状态 1 = RTC 在正常激活状态

RTC频率补偿寄存器(RTC_FREQADJ)

寄存器	偏移	R/W	描述	复位值
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC频率补偿寄存器	0x0000_1000

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

位	描述	
[31:13]	**Reserved**	保留
[12:8]	**INTEGER**	**整数部分** 00000 = 检测值的整数部分为32752 00001 =检测值的整数部分为32753 00010 =检测值的整数部分为32754 00011 =检测值的整数部分为32755 00100 =检测值的整数部分为32756 00101 =检测值的整数部分为32757 00110 = 检测值的整数部分为32758 00111 = 检测值的整数部分为32759 01000 = 检测值的整数部分为32760 01001 = 检测值的整数部分为32761 01010 = 检测值的整数部分为32762 01011 = 检测值的整数部分为32763 01100 = 检测值的整数部分为32764 01101 = 检测值的整数部分为32765 01110 = 检测值的整数部分为32766 01111 = 检测值的整数部分为32767 10000 = 检测值的整数部分为32768 10001 = 检测值的整数部分为32769 10010 = 检测值的整数部分为32770 10011 = 检测值的整数部分为32771 10100 = 检测值的整数部分为32772 10101 = 检测值的整数部分为32773 10110 = 检测值的整数部分为32774

		10111 = 检测值的整数部分为32775 11000 = 检测值的整数部分为32776 11001 = 检测值的整数部分为32777 11010 = 检测值的整数部分为32778 11011 = 检测值的整数部分为32779 11100 = 检测值的整数部分为32780 11101 = 检测值的整数部分为32781 11110 = 检测值的整数部分为32782 11111 = 检测值的整数部分为32783
[7:6]	**Reserved**	保留
[5:0]	**FRACTION**	**小数部分** 公式：FRACTION = (检测值小数部分) X 64 **注：**在 FCR的数字必须是16进制格式。

注：对寄存器RTC_FREQADJ、 RTC_TIME、 RTC_CAL或 RTC_WEEKDAY进行写操作后，FREQADJ的计数器都将被复位并开始补偿，RTC时间将重新开始。

RTC 时间载入寄存器 (RTC_TIME)

寄存器	偏移	R/W	描述	复位值
RTC_TIME	RTC_BA+0x0C	R/W	RTC时间载入寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**TENHR**		**HR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TENMIN**			**MIN**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TENSEC**			**SEC**			

位	描述	
[31:22]	**Reserved**	保留
[21:20]	**TENHR**	小时的十位数(0~2) 当 RTC 采用 12 小时制时，RTC_TIME[21](TENHR[1:0] 的高位) 表示 AM/PM 提示符 (如果 IRTC_TIME[21]=1，表示为PM时间)
[19:16]	**HR**	小时的个位数(0~9)
[15]	**Reserved**	保留
[14:12]	**TENMIN**	分钟的十位数(0~5)
[11:8]	**MIN**	分钟的个位数(0~9)
[7]	**Reserved**	保留
[6:4]	**TENSEC**	秒钟的十位数(0~5)
[3:0]	**SEC**	秒钟的个位数(0~9)

注:

1.RTC_TIME 为 BCD 计数方式，RTC 不会对载入值的合理性进行检测；

2. 合理的参数范围列举在括号内；

3. 对寄存器RTC_FREQADJ、 RTC_TIME、 RTC_CAL或 RTC_WEEKDAY进行写操作后，FREQADJ的计数器都将被复位并开始补偿，RTC时间将重新开始。

RTC日历载入寄存器(RTC_CAL)

寄存器	偏移	R/W	描述	复位值
RTC_CAL	RTC_BA+0x10	R/W	RTC日历载入寄存器	0x0015_0808

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TENYEAR				**YEAR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**TENMON**	**MON**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TENDAY**		**DAY**			

位	描述	
[31:24]	**Reserved**	保留
[23:20]	**TENYEAR**	年日历的十位数(0~9)
[19:16]	**YEAR**	年日历的个位数(0~9)
[15:13]	**Reserved**	保留
[12]	**TENMON**	月日历的十位数(0~1)
[11:8]	**MON**	月日历的个位数(0~9)
[7:6]	**Reserved**	保留
[5:4]	**TENDAY**	天日历的十位数(0~3)
[3:0]	**DAY**	天日历的个位数(0~9)

注：

1. RTC_CAL为 BCD 计数格式，RTC 不会检测载入值的合理性；
2. 合理的参数范围列举在括号内；
3. 对寄存器RTC_FREQADJ、 RTC_TIME、 RTC_CAL或 RTC_WEEKDAY进行写操作后，FREQADJ的计数器都将被复位并开始补偿，RTC时间将重新开始。

RTC时间制式选择寄存器(RTC_CLKFMT)

寄存器	偏移	R/W	描述	复位值
RTC_CLKFMT	RTC_BA+0x14	R/W	RTC时间制式选择寄存器	0x0000_0001

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**24HEN**	**24小时/ 12小时时间制式选择** 用于表示RTC_TIME 和 RTC_TALM的时间为24小时或12小时制式 0 =选择12小时制，带AM和PM指示 1 =选择24小时制

RTC星期寄存器(RTC_WEEKDAY)

寄存器	偏移	R/W	描述	复位值
RTC_WEEKDAY	RTC_BA+0x18	R/W	RTC星期寄存器	0x0000_0006

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

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**WEEKDAY**	**星期寄存器** 000 = 星期日 001 = 星期一 010 = 星期二 011 = 星期三 100 = 星期四 101 =星期五 110 =星期六 111 = 保留

注：对寄存器RTC_FREQADJ、 RTC_TIME、 RTC_CAL或 RTC_WEEKDAY进行写操作后，FREQADJ的计数器都将被复位并开始补偿，RTC时间将重新开始。

RTC时间闹钟寄存器(RTC_TALM)

寄存器	偏移	R/W	描述	复位值
RTC_TALM	RTC_BA+0x1C	R/W	RTC时间闹钟寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**TENHR**		**HR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TENMIN**			**MIN**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TENSEC**			**SEC**			

位	描述	
[31:22]	**Reserved**	保留
[21:20]	**TENHR**	闹钟中小时时间的十位数(0~2) 当 RTC采用12-小时制时，RTC_TIME[21](TENHR[1:0]的高位)表示 AM/PM 提示符.(如果RTC_TIME[21]=1,表示为PM时间.)
[19:16]	**HR**	闹钟中小时时间的个位数(0~9)
[15]	**Reserved**	保留
[14:12]	**TENMIN**	闹钟中分钟时间的十位数(0~5)
[11:8]	**MIN**	闹钟中分钟时间的个位数(0~9)
[7]	**Reserved**	保留
[6:4]	**TENSEC**	闹钟中秒钟时间的十位数(0~5)
[3:0]	**SEC**	闹钟中秒钟时间的个位数(0~9)

注:

1. RTC_TALM 为 BCD 计数方式，RTC 不会对载入值的合理性进行检测；
2. 合理的参数范围列举在括号内；

RTC日历闹钟寄存器(RTC_CALM)

寄存器	偏移	R/W	描述	复位值
RTC_CALM	RTC_BA+0x20	R/W	RTC日历闹钟寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TENYEAR				**YEAR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**TENMON**	**MON**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TENDAY**		**DAY**			

位	描述	
[31:24]	**Reserved**	保留
[23:20]	**TENYEAR**	闹钟中年日历的十位数(0~9)
[19:16]	**YEAR**	闹钟中年日历的个位数(0~9)
[15:13]	**Reserved**	保留
[12]	**TENMON**	闹钟中月日历的十位数 (0~1)
[11:8]	**MON**	闹钟中月日历的个位数(0~9)
[7:6]	**Reserved**	保留
[5:4]	**TENDAY**	闹钟中天日历的十位数(0~3)
[3:0]	**DAY**	闹钟中天日历的个位数(0~9)

注：

1. RTC_CALM 为 BCD 计数方式，RTC 不会对载入值的合理性进行检测；
2. 合理的参数范围列举在括号内；

RTC闰年指示寄存器(RTC_LEAPYEAR)

寄存器	偏移	R/W	描述	复位值
RTC_LEAPYEAR	RTC_BA+0x24	R	RTC闰年指示寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**LEAPYEAR**	闰年指示寄存器（只读） 0 = 表示该年非闰年 1 = 表示该年为闰年

RTC中断使能寄存器(RTC_INTEN)

寄存器	偏移	R/W	描述	复位值
RTC_INTEN	RTC_BA+0x28	R/W	RTC中断使能寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留
[1]	**TICKIEN**	**时钟节拍中断使能位** 当时钟节拍中断发生时，设置 TICKIEN 为 1也可以使能芯片唤醒功能 0 = 禁止RTC 时钟节拍中断 1 = 使能RTC 时钟节拍中断
[0]	**ALMIEN**	**闹钟中断使能位** 当RTC闹钟中断发生时，设置ALMIEN 为 1也可以使能芯片唤醒功能 0 = 禁止RTC 闹钟中断 1 = 使能RTC 闹钟中断

RTC中断状态寄存器 (RTC_INTSTS)

寄存器	偏移	R/W	描述	复位值
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC中断状态寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留
[1]	**TICKIF**	**RTC时钟节拍中断标志** 0 = 未发生节拍情况 1 = 发生节拍情况 **注：** 该位写1清0。
[0]	**ALMIF**	**RTC 闹钟中断标志** 0 = 不匹配闹钟条件 1 = 匹配闹钟条件 **注：** 该位写1清0。

RTC时钟节拍寄存器(RTC_TICK)

寄存器	偏移	R/W	描述	复位值
RTC_TICK	RTC_BA+0x30	R/W	RTC时钟节拍寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**TICK**	**时钟节拍寄存器** 该位用来设定产生时钟节拍中断的周期. 000 = 时钟节拍为1秒 001 = 时钟节拍为1/2 秒 010 = 时钟节拍为1/4 秒 011 = 时钟节拍为1/8 秒 100 = 时钟节拍为1/16 秒 101 = 时钟节拍为1/32 秒 110 = 时钟节拍为1/64 秒 111 = 时钟节拍为1/128 秒

RTC时间闹钟屏蔽寄存器(RTC_TAMSK)

寄存器	偏移	R/W	描述	复位值
RTC_TAMSK	RTC_BA+0x34	R/W	RTC时间闹钟屏蔽寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MTENHR**	**MHR**	**MTENMIN**	**MMIN**	**MTENSEC**	**MSEC**

位	描述	
[31:6]	**Reserved**	保留
[5]	**MTENHR**	屏蔽闹钟小时时间的十位数(0~2)
[4]	**MHR**	屏蔽闹钟小时时间的个位数(0~9)
[3]	**MTENMIN**	屏蔽闹钟分钟的十位数(0~5)
[2]	**MMIN**	屏蔽闹钟分钟的个位数(0~9)
[1]	**MTENSEC**	屏蔽闹钟秒钟的十位数(0~5)
[0]	**MSEC**	屏蔽闹钟秒钟的个位数(0~9)

注:

1. RTC_TAMSK 为 BCD 计数方式，RTC 不会对载入值的合理性进行检测；
2. 合理的参数范围列举在括号内；
3. MTENHR/MHR 基于24小时制。

RTC日历闹钟屏蔽寄存器(RTC_CAMSK)

寄存器	偏移	R/W	描述	复位值
RTC_CAMSK	RTC_BA+0x38	R/W	RTC日历闹钟屏蔽寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MTENYEAR**	**MYEAR**	**MTENMON**	**MMON**	**MTENDAY**	**MDAY**

位	描述	
[31:6]	**Reserved**	保留
[5]	**MTENYEAR**	屏蔽闹钟年日历的十位数(0~9)
[4]	**MYEAR**	屏蔽闹钟年日历的个位数(0~9)
[3]	**MTENMON**	屏蔽闹钟月日历的十位数(0~1)
[2]	**MMON**	屏蔽闹钟月日历的个位数(0~9)
[1]	**MTENDAY**	屏蔽闹钟天日历的个位数(0~3)
[0]	**MDAY**	屏蔽闹钟天日历的十位数(0~9)

注:

1. RTC_CALM为 BCD 计数方式，RTC不会对载入值的合理性进行检测；
2. 合理的参数范围列举在括号内。

RTC 32K晶振控制寄存器(RTC_LXTCTL)

寄存器	偏移	R/W	描述	复位值
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz晶振控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C32KS	**Reserved**						

位	描述	
[31:8]	**Reserved**	保留
[7]	**C32KS**	32K时钟源选择： 0=时钟源选择外部LXT 32K时钟 1=时钟源选择内部LIRC 38K时钟
[6:0]	**Reserved**	保留

注:

1. RTC_CALM 为BCD 数字计数器，RTC不会检查载入的数据的正确性。
2. 合理的参数范围列举在括号内。

6.11 基本PWM 发生器和捕获定时器(BPWM)

6.11.1 概述

如图 6.11-1所示，芯片有 2 组 BPWM 发生器，分别为BPWM0 和 BPWM1。每组 BPWM 都支持 6路 BPWM 输出和捕捉输入通道、12 位的预分频器、16 位比较器，6 路通道共享一个16 位的 BPWM 计数器。BPWM 计数器支持向上、向下和上下三种计数方式。BPWM 通过比较比较器和计数器的值来产生事件。这些事件可以用来产生 BPWM 脉冲、BPWM中断和触发ADC开始转换的信号。BPWM 的输出控制单元支持极性输出、独立的管脚屏蔽和三态输出使能。

BPWM 发生器还支持输入捕捉功能，当输入通道有上升沿、下降沿或双边沿信号发生时,锁存 BPWM 计数器的值到对应的寄存器。

6.11.2 特性

6.11.2.1 BPWM 功能特性

- 时钟频率最高可达到最大的 144MHz 频率
- 2 组 BPWM 模块，每组都提供 6 路输出通道
- BPWM 输出/捕捉通道支持独立模式
- 从 1 到 4096 的 12 位预分频器
- 16 位精度的 BPWM 计数器，每个模块都有 1 个 BPWM 计数器
  - 向上、向下及上下的计数器操作模式
- 每个 BPWM 管脚都支持屏蔽功能和三态使能
- 以下事件可以触发中断：
  - BPWM 计数器计数到 0 、周期值或比较器值
- 以下事件可以触发ADC：
  - BPWM 计数器到 0、周期值或比较器值

6.11.2.2 捕捉功能特性

- 多达 12 路 16 位精度的输入捕捉通道
- 支持向上或向下的捕捉条件
- 支持向上或向下的捕捉中断
- 支持向上或向下的捕捉触发计数器重载

6.11.3 框图

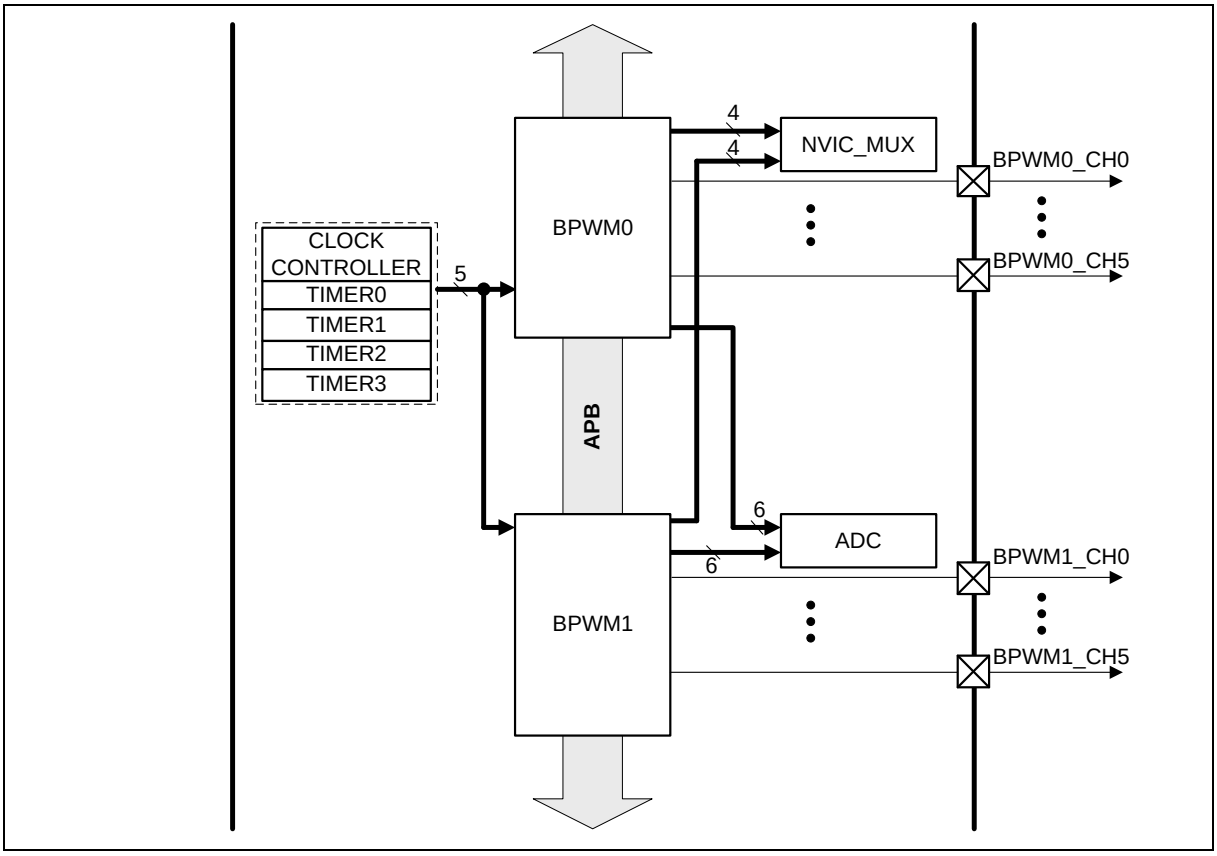


图 6.11-1 BPWM 发生器框图

如图 6.11-3所示，每个 BPWM 发生器只有一个时钟源输入，通过 BPWM_CLK0 的 ECLKSRC0 (BPWM_CLKSRC[2:0]) 寄存器决定BPWM 输出的触发源是BPWM 时钟还是定时器。通常，BPWM0要设置成 BPWM0SEL (CLK_CLKSEL2[8]) 为 1 选择 PCLK0，BPWM1 要设置成 BPWM1SEL (CLK_CLKSEL2[9]) 为 1 选择 PCLK1。

如

图 6.11-2

频率比 PCLK:BPWM时钟	HCLK	PCLK	BPWM Clock	HCLKSEL CLK_CLKSEL0[2:0]	HCLKDIV CLK_CLKDIV0[3:0]	APBnDIV (CLK_PCLKDIVn [2+4n:4n]), N表示0或1	BPWMnSEL (CLK_CLKSEL2[N+8]), N表示0或1
1:1	HCLK	PCLK	PCLK	无关	无关	无关	1
1:2	PLL/ 2	PLL/ 2	PLL	2	1	0	0

和表 6.11-1 BPWM时钟源控制寄存器设置表所示，当需要运行在最大PLL时钟频率时，BPWM0和BPWM1时钟必须通过设置BPWM0SEL (CLK_CLKSEL2[8]) 和 BPWM1SEL (CLK_CLKSEL2[9])为0来选择PLL时钟。

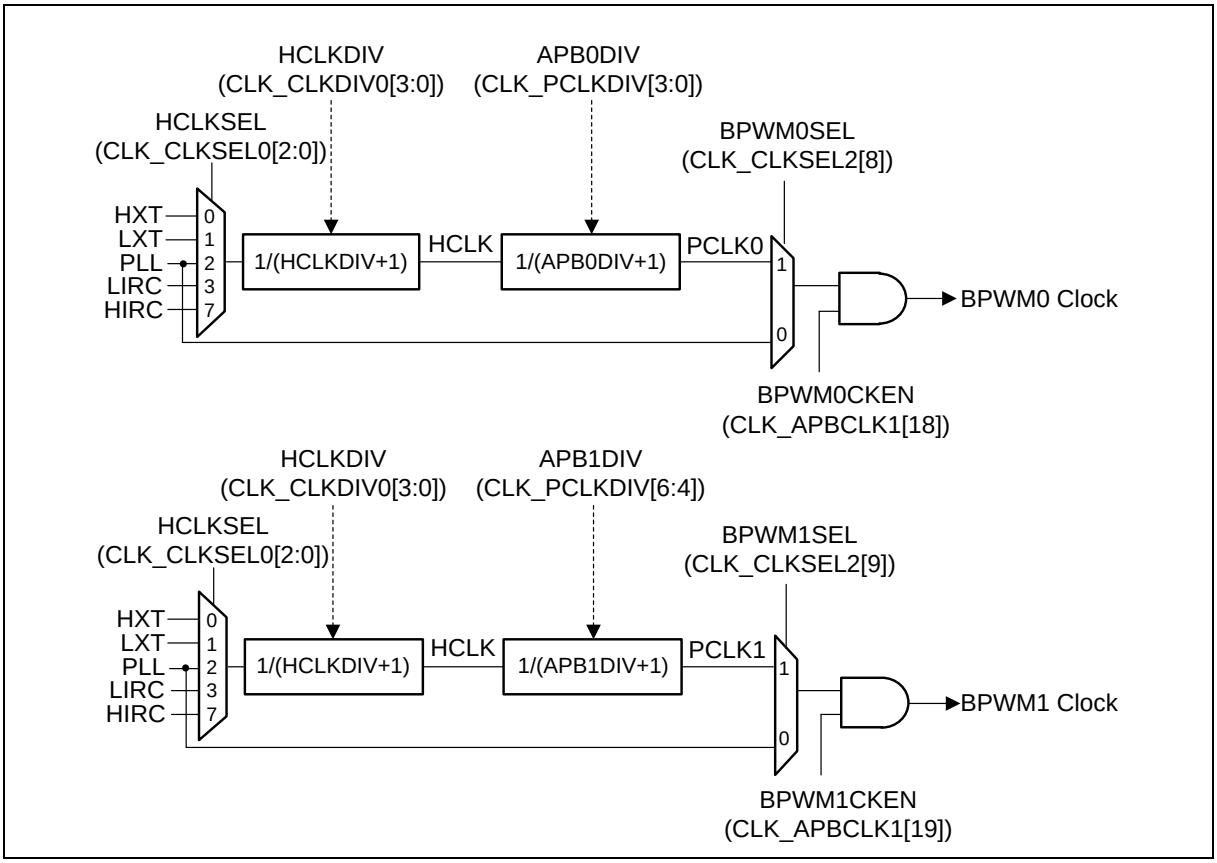


图 6.11-2 BPWM 系统时钟源控制

频率比 PCLK:BPWM时钟	HCLK	PCLK	BPWM Clock	HCLKSEL CLK_CLKSEL0[2:0]	HCLKDIV CLK_CLKDIV0[3:0]	APBnDIV (CLK_PCLKDIVn [2+4n:4n]), N表示0或1	BPWMnSEL (CLK_CLKSEL2[N+8]), N表示0或1
1:1	HCLK	PCLK	PCLK	无关	无关	无关	1
1:2	PLL/ 2	PLL/ 2	PLL	2	1	0	0

表 6.11-1 BPWM 时钟源控制寄存器设置表

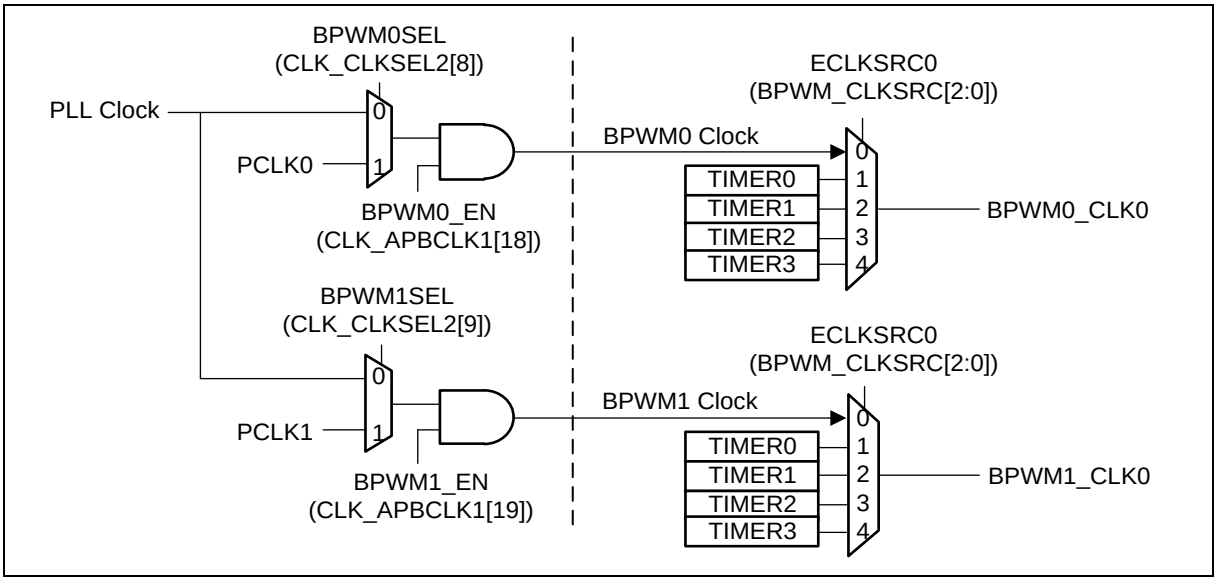


图 6.11-3 BPWM 时钟源控制

图 6.11-4所示是BPWM独立模式的架构，所有的6路通道共享一个计数器。当计数器计数到0、PERIOD (BPWM_PERIOD[15:0])或者比较器值时，事件将会产生。这些事件被传到对应的发生器后，将产生 BPWM 脉冲、中断信号或者触发ADC开始转换的信号。输出控制则用来改变 BPWM 脉冲的输出状态。

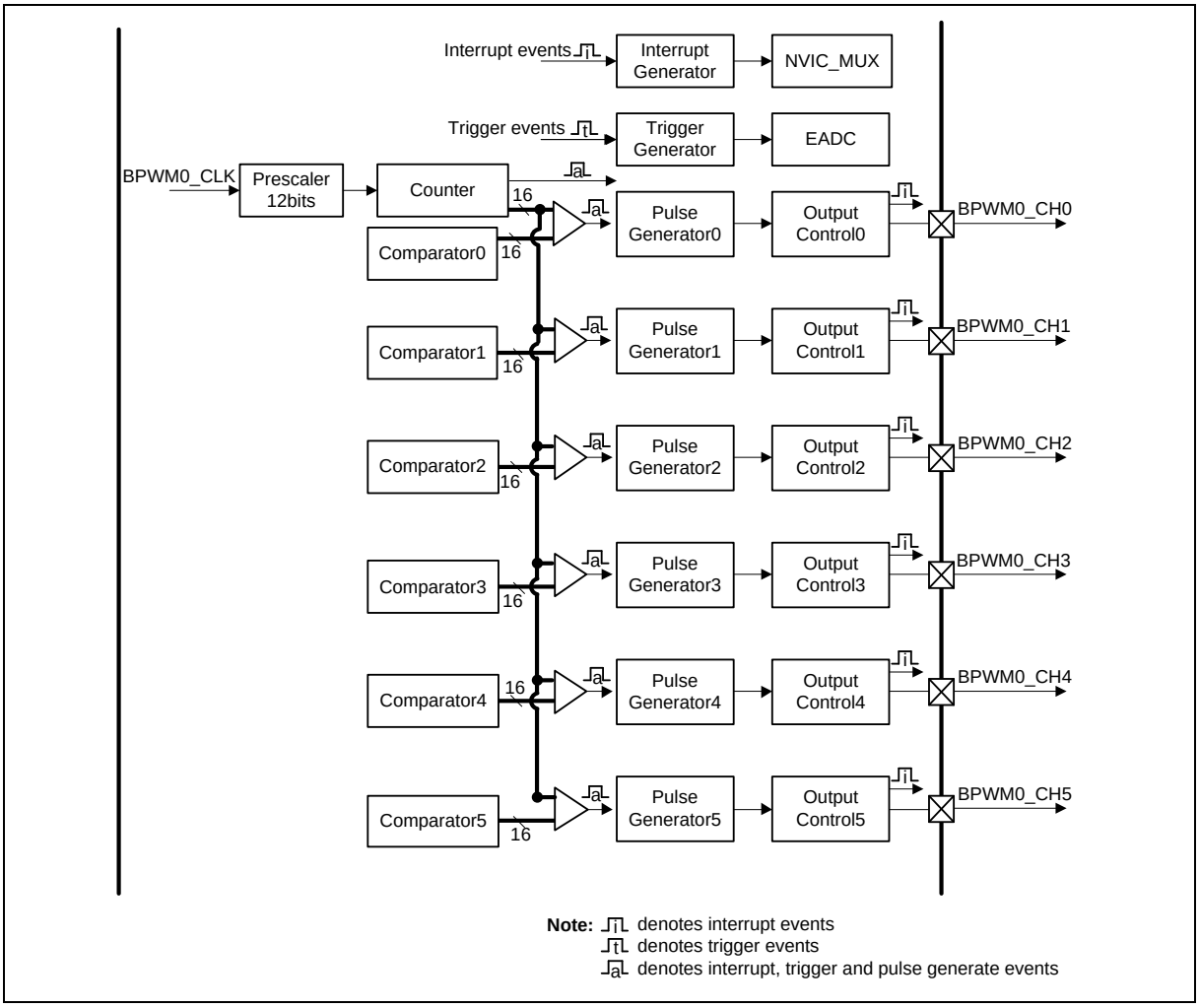


图 6.11-4 BPWM 独立模式框架图

6.11.4 基本配置

6.11.4.1 BPWM0 基本配置

- 时钟源配置
  - 在寄存器BPWM0SEL (CLK_CLKSEL2[8])选择 BPWM0 的时钟源
  - 在寄存器BPWM0CKEN (CLK_APBCLK1[18])使能 BPWM0 外设时钟
- 复位配置
  - 在寄存器BPWM0RST (SYS_IPRST2[18])复位 BPWM0 控制器
- 管脚配置

组	管脚名	GPIO	MFP
BPWM0	BPWM0_CH0	PA.11	MFP9
		PA.0, PG.14	MFP12

		PE.2	MFP13
	BPWM0_CH1	PA.10	MFP9
		PA.1, PG.13	MFP12
		PE.3	MFP13
	BPWM0_CH2	PA.9	MFP9
		PA.2, PG.12	MFP12
		PE.4	MFP13
	BPWM0_CH3	PA.8	MFP9
		PA.3, PG.11	MFP12
		PE.5	MFP13
	BPWM0_CH4	PF.5	MFP8
		PC.13	MFP9
		PA.4, PG.10	MFP12
		PE.6	MFP13
	BPWM0_CH5	PF.4	MFP8
		PD.12	MFP9
		PA.5, PG.9	MFP12
		PE.7	MFP13

6.11.4.2 BPWM1 基本配置

- 时钟源配置
  - 在寄存器BPWM1SEL (CLK_CLKSEL2[9])选择 BPWM1 的时钟源
  - 在寄存器BPWM1CKEN (CLK_APBCLK1[19])使能 BPWM1 的外设时钟
- 复位配置
  - 在寄存器BPWM1RST (SYS_IPRST2[19])复位BPWM1控制器
- 管脚配置

组	管脚名	GPIO	MFP
BPWM1	BPWM1_CH0	PB.11	MFP10
		PF.3	MFP11
		PC.7, PF.0	MFP12
	BPWM1_CH1	PB.10	MFP10
		PF.2	MFP11
		PC.6, PF.1	MFP12
	BPWM1_CH2	PB.9	MFP10
		PA.12	MFP11

		PA.7	MFP12
	BPWM1_CH3	PB.8	MFP10
		PA.13	MFP11
		PA.6	MFP12
	BPWM1_CH4	PB.7	MFP10
		PA.14	MFP11
		PC.8	MFP12
	BPWM1_CH5	PB.6	MFP10
		PA.15	MFP11
		PE.13	MFP12

6.11.5 功能描述

6.11.5.1 BPWM预分频器

BPWM 预分频器用来分频时钟源，预分频器每计数 CLKPSC+1 次， BPWM 计数器则只计数 1 次。预分频的值由 CLKPSC (BPWM_CLKPSC[11:0])的设置决定。图 6.11-5为 BPWM 通道 0 CLKPSC 波形示例图。预分频计数器会在下一个预分频向下计数的开始时重载 CLKPSC 的值。

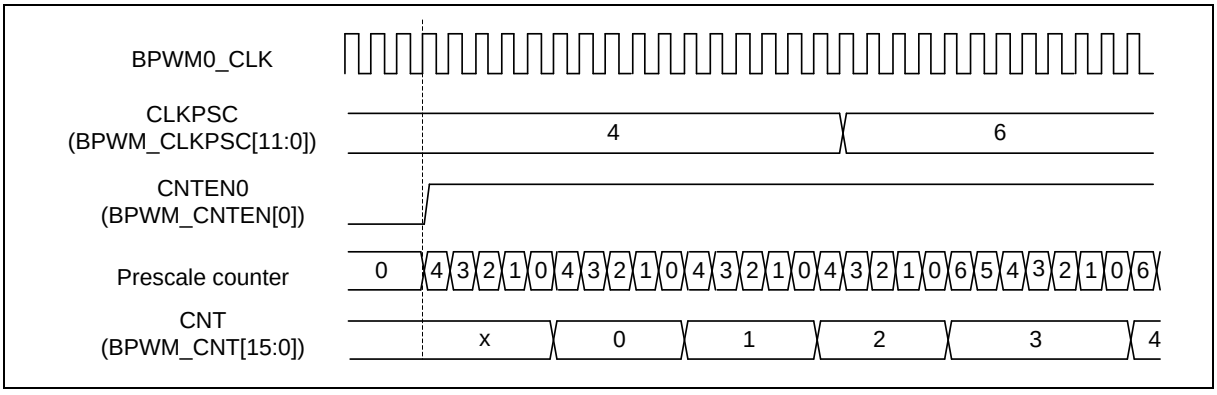


图 6.11-5 BPWM_CH0 CLKPSC 波形

6.11.5.2 BPWM计数器

BPWM有一个计数器,支持 3 种计数操作方式：向上、向下和上下计数。

对于 BPWM 通道 0 ，当分频计数器向下计数到0时, CNTCLR0(BPWM_CNTCLR[0])将会由硬件自动清零，CNTCLR0（BPWM_CNTCLR [0]）则会将CNT（BPWM_CNT [15：0]）清空为0x00。

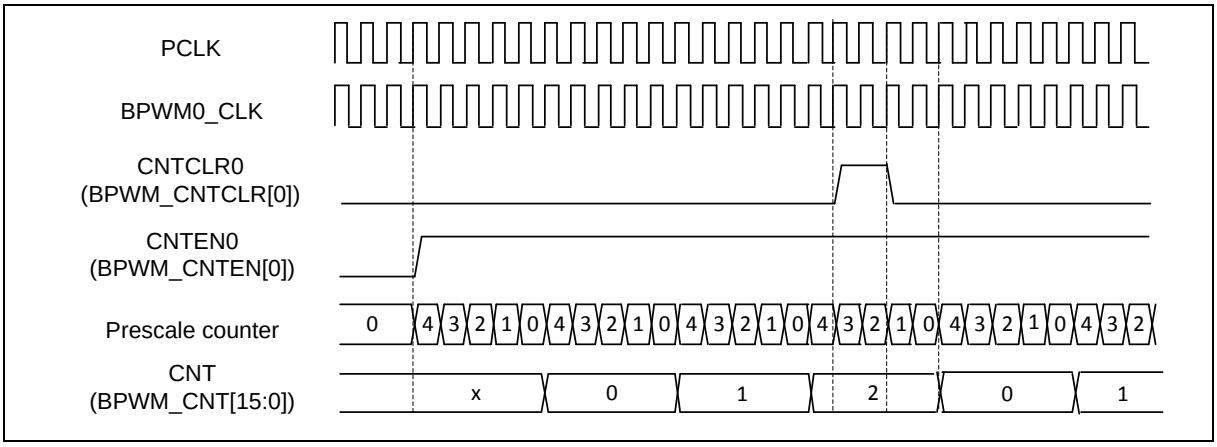


图 6.11-6 BPWM 计数器清空波形

6.11.5.3 向上计数方式

在向上计数操作中，16 位的 BPWM 计数器从 0 开始向上计数到 PERIOD (BPWM_PERIOD) 完成一个 BPWM 周期。从 CNT (BPWM_CNT[15:0]) 寄存器可以读到当前的计数值。当计数器计数到 0 时，BPWM 发生器会产生零点事件；计数器计数到 PERIOD 时则会产生周期点事件。图 6.11-7所示为向上计数方式下的周期时间示例，BPWM 的周期时间 = (PERIOD+1) * (CLKPSC+1) * BPWMx_CLK。

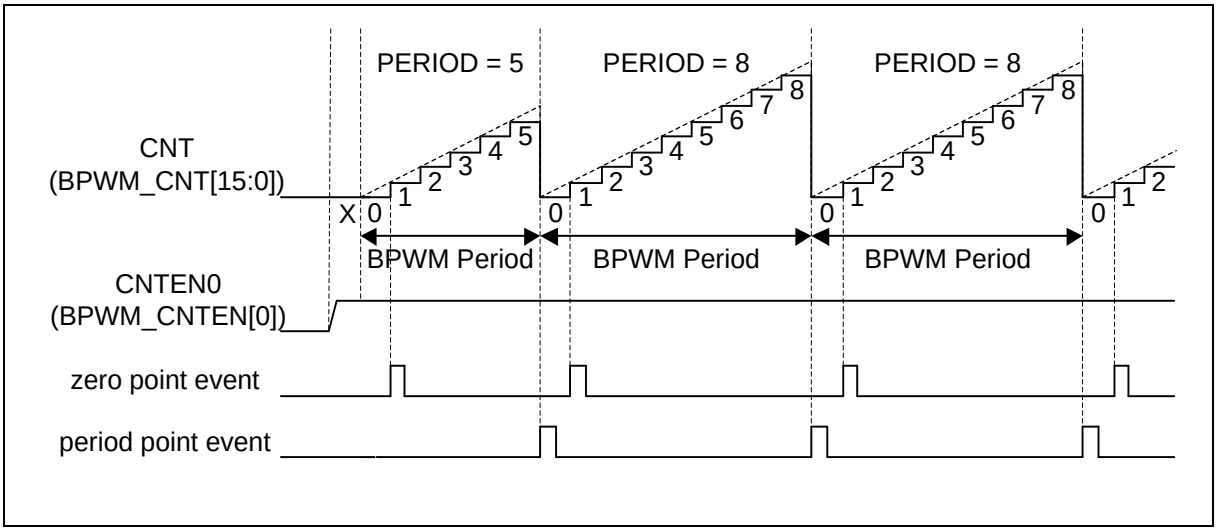


图 6.11-7 BPWM 向上计数模式

6.11.5.4 向下计数方式

在向下计数操作中，16 位的 BPWM 计数器从 PERIOD 开始向下计数到 0 完成一个 BPWM 周期。从 CNT (BPWM_CNT[15:0]) 寄存器可以读到当前的计数值。当计数器计数到 0 时，BPWM 发生器会产生零点事件；计数器计数到 PERIOD 时则会产生周期点事件。图 6.11-8所示为向下计数方式下的周期时间示例，BPWM 的周期时间 = (PERIOD+1) * (CLKPSC+1) * BPWMx_CLK。

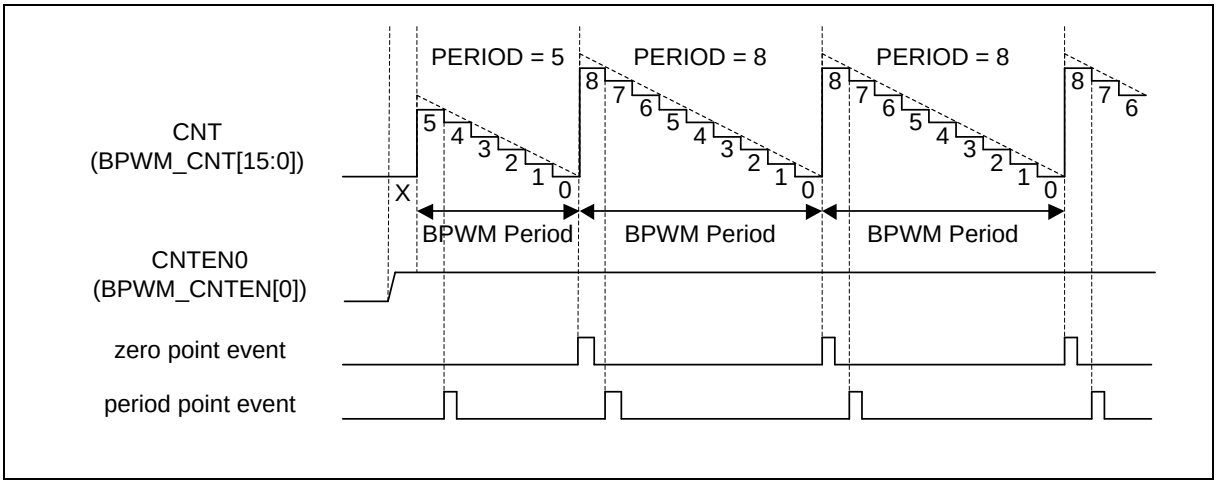


图 6.11-8 BPWM 向下计数方式

6.11.5.5 上下计数方式

在上下计数操作中，16 位的 BPWM 计数器先从 0 开始向上计数到 PERIOD，再从 PERIOD 开始向下计数到 0 完成一个 BPWM 周期。从 CNT (BPWM_CNT[15:0]) 寄存器可以读到当前的计数值。当计数器计数到 0 时，BPWM 发生器会产生零点事件；计数器计数到 PERIOD 时则会产生中点事件。图 6.11-9所示为上下计数方式下的周期时间示例，BPWM 的周期时间 = (2*PERIOD) * (CLKPSC+1) * BPWMx_CLK，寄存器 DIRF (BPWM_CNT[16]) 是计数方向的标志位，向上计数时为 1，向下计数时为 0。

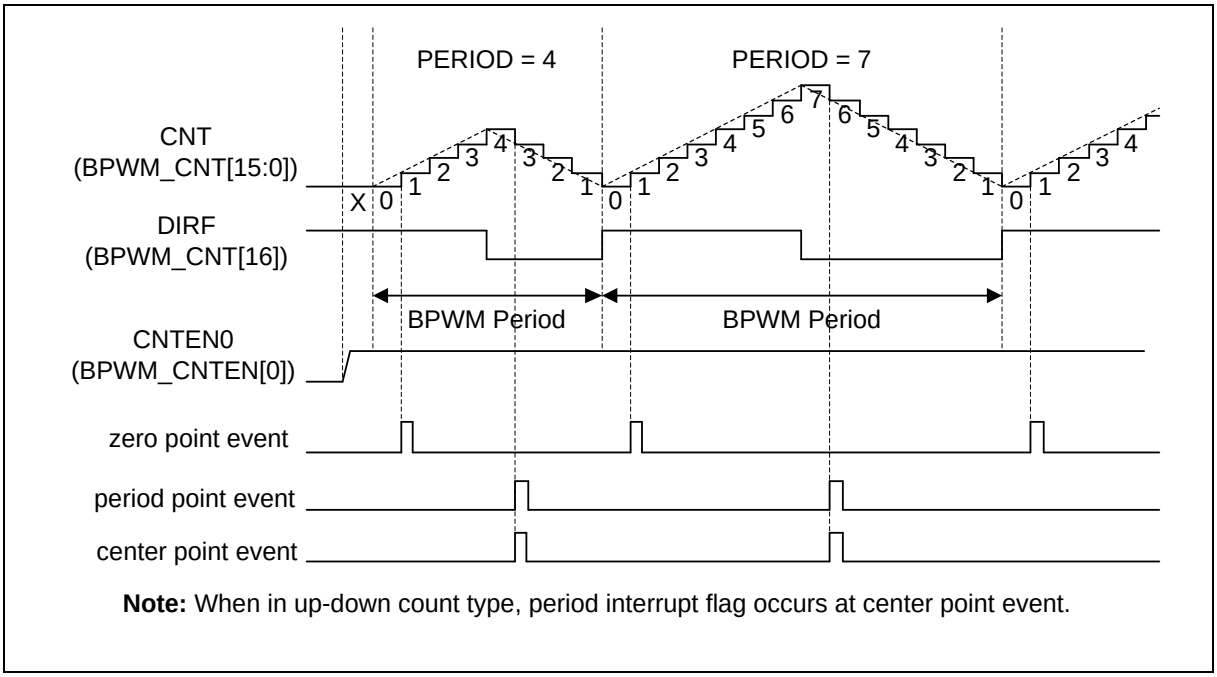


图 6.11-9 BPWM 上下计数方式

6.11.5.6 BPWM比较器

CMPDAT (BPWM_CMPDATn[15:0]) 是 BPWM 通道 n 的基本比较器寄存器，每路通道都有且仅有一个

CMPDAT。CMPDAT 的值不断与计数器的值做比较。当计数器计数到比较器寄存器的值时， BPWM 会产生一个事件并以此产生 BPWM 脉冲、中断或者ADC触发信号。如图 6.11-10所示，在上下计数方式里会有 2 个事件产生。

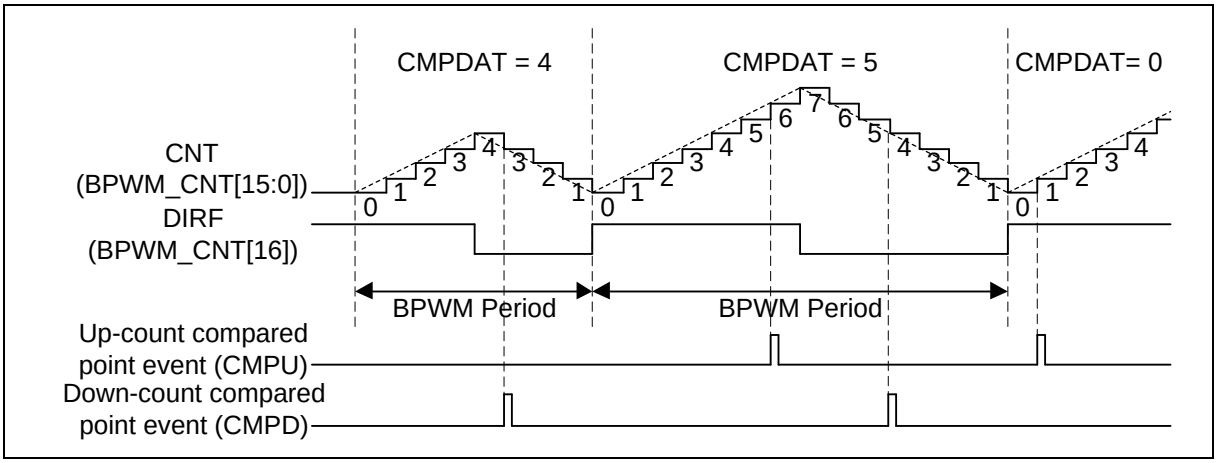


图 6.11-10 上下计数方式时 BPWM CMPDAT 事件

6.11.5.7 周期载入模式

周期载入模式是默认的载入模式。在所有载入模式里它是优先级最低的。当周期结束时，PERIOD 和 CMPDAT 都会被载入到对应的缓存。例如，在向上计数操作里 BPWM 从 0 向上计数到 PERIOD 后，或者向下计数操作里从 PERIOD 向下计数到 0 后，亦或上下计数操作里从 0 计数到 PERIOD 再向下计数到 0 后，PERIOD 和 CMPDAT都会进行重载。

图 6.11-11所示是向上计数方式的重载时序，图示里 PERIOD DATA0 为 PERIOD 的初始值，PERIOD DATA1 为 PERIOD 第一次由软件更新的值，以此类推，CMPDAT 也按这种方式标注。图 6.11-11的具体步骤顺序参见下面的描述。用户可以查看 BPWM 的周期和 CMPU 事件来了解 PERIOD 和 CMPDAT 的更新条件。

1. 在位置 1，软件写 CMPDAT DATA1 到 CMPDAT
2. 在位置 2，PWM 周期结束时，硬件载入 CMPDAT DATA1 到 CMPBUF
3. 在位置 3，软件写 PERIOD DATA1 到 PERIOD
4. 在位置 4，PWM 周期结束时，硬件载入 PERIOD DATA1 到 PBUF
5. 在位置 5，软件写 PERIOD DATA2 到 PERIOD
6. 在位置 6，PWM 周期结束时，硬件载入 PERIOD DATA2 到 PBUF.

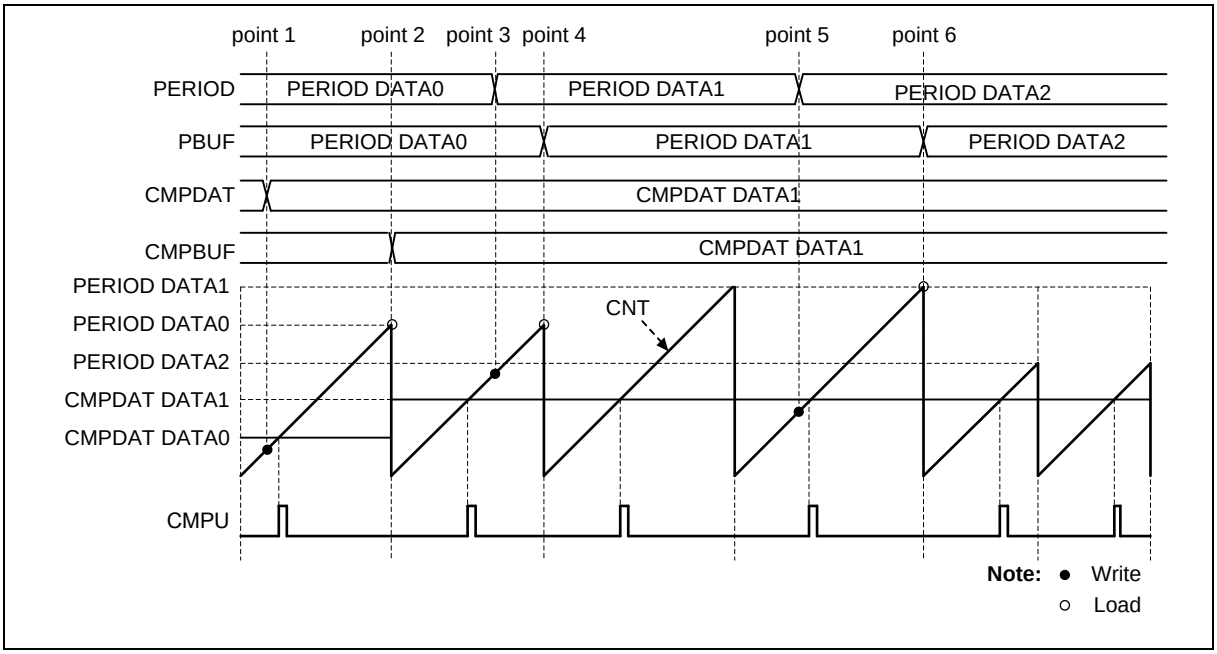


图 6.11-11 向上计数方式下的周期载入模式

6.11.5.8 立即载入模式

如果 BPWM 通道 n 在寄存器IMMLDENn (BPWM_CTL0[21:16])里对应的位置 1 时，当软件更新 PERIOD 或者 CMPDAT 时，硬件会立刻载入 PERIOD 和 CMPDAT 到缓存。如果更新的 PERIOD 值小于当前的计数器值，计数器会一直计数到 0xFFFF，当计数器计数到 0xFFFF 且预分频值计数到 0时，寄存器CNTMAX0(BPWM_STATUS[0])会被置位，然后计数器重新回转计数（wraparound）。立即载入模式拥有最高的优先级。如果 IMMLDENn 被置位，其他针对通道 n 的模式设置都会无效。下面是对图 6.11-12示例的文字说明：

1. 在位置 1 软件写 CMPDAT DATA1，硬件立即重载 CMPDAT DATA1 到 CMPBUF
2. 在位置 2 软件写入一个比当前计数器大的值 PERIOD DATA1，计数器会一直计数到 PERIOD DATA1 的值以完成一个载入周期.
3. 在位置 3 软件写入一个比当前计数器小的值 PERIOD DATA2，计数器会一直计数到最大值 0xFFFF，再从 0 开始回转计数到 PERIOD DATA2 以完成周期载入。

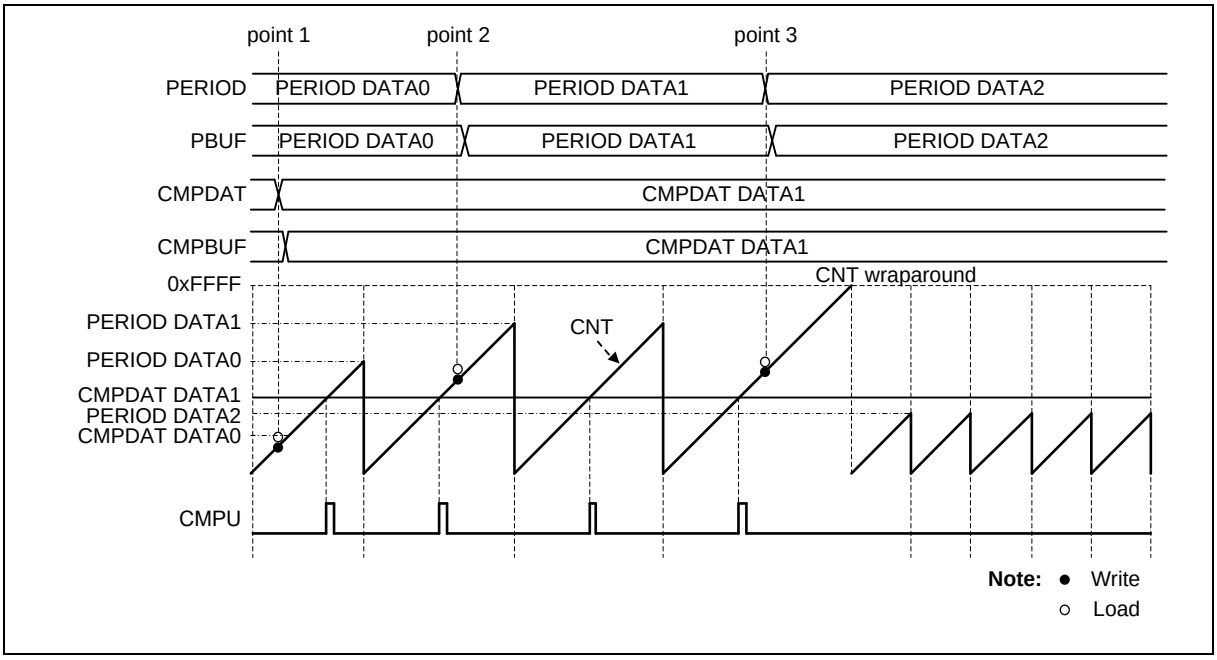


图 6.11-12 立即载入模式下的向上计数

6.11.5.9 中心载入模式

如果 BPWM 通道 n 在寄存器 CTRLDn (BPWM_CTL0[5:0]) 里对应的位置 1 且处于上下计数方式时，CMPDAT 的值会在周期中间被载入到 CMPBUFn，即在计数器计数到 PERIOD 时载入。PERIOD 的载入时间同周期载入模式相同。下面是对图 6.11-13示例的文字说明：

1. 在位置 1 软件写 CMPDAT DATA1
2. 在位置 2 PWM 周期中间位置，硬件载入 CMPDAT DATA1 到 CMPBUF
3. 在位置 3 软件写 PERIOD DATA1
4. 在位置 4 PWM 周期末位置，硬件载入 PERIOD DATA1 到 PBUF
5. 在位置 5 软件写 CMPDAT DATA2
6. 在位置 6 PWM 周期中间位置，硬件载入 CMPDAT DATA2 到 CMPBUF
7. 在位置 7 软件写 PERIOD DATA2
8. 在位置 8 PWM 周期末位置，硬件写 PERIOD DATA2 到 PBUF

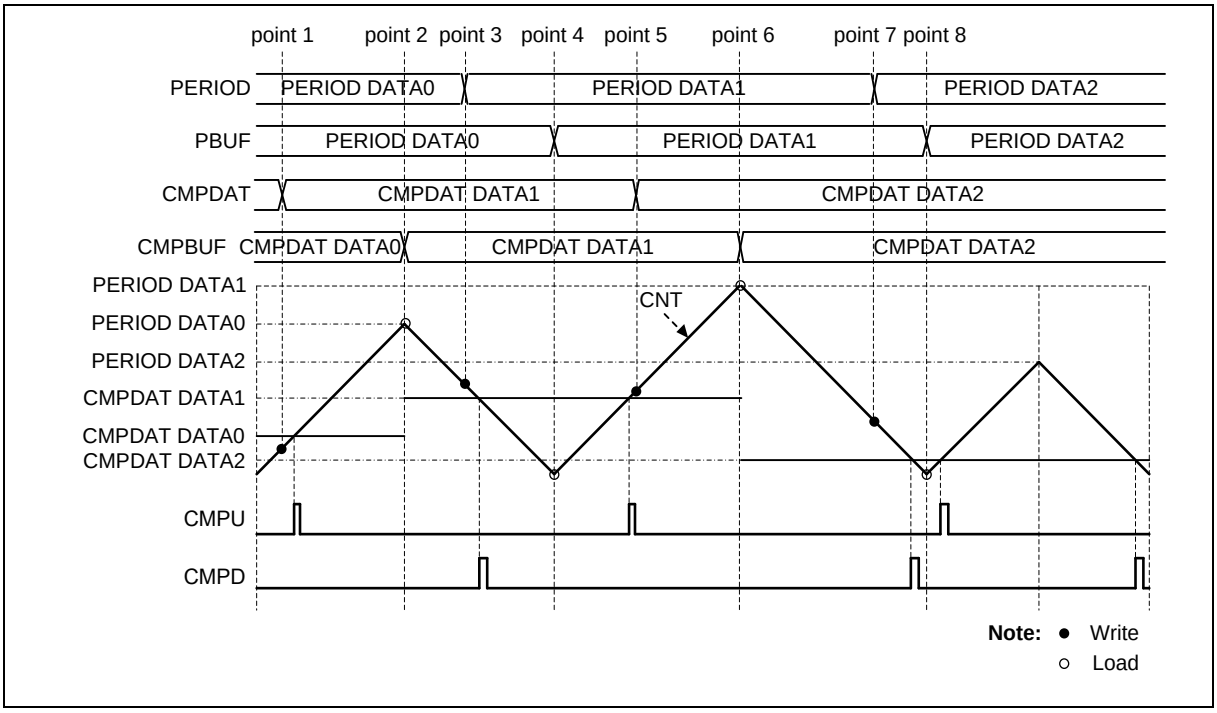


图 6.11-13 上下计数方式下的中间载入模式

6.11.5.10 BPWM脉冲发生器

BPWM 脉冲发生器使用计数器和比较器事件来产生 BPWM 脉冲。对应的事件有：零点、向上计数器方式和向下计数器方式的周期点、上下计数方式的中点以及三种模式下计数器计数到比较器值的位置。对于上下计数方式，有两个计数器计数到比较器值的位置，一是向上计数的过程，二是向下计数的过程。

通过 BPWM_WGCTL0 和 BPWM_WGCTL1 寄存器可以设定每个位置点 的BPWM 波形：无动作 (X)、拉低 (L)、拉高 (H) 或者触发 (T)。如图 6.11-14，通过这些位置点的设置，用户可以轻易的产生不对称的 BPWM 脉冲和其它多样化的波形。图中，比较器 n 用于产生 BPWM 脉冲，n 指通道 0 ~ 5；CMPU 指向上计数时 CNT 计数到 CMPDAT，CMPD 指向下计数时 CNT 计数到 CMPDAT。

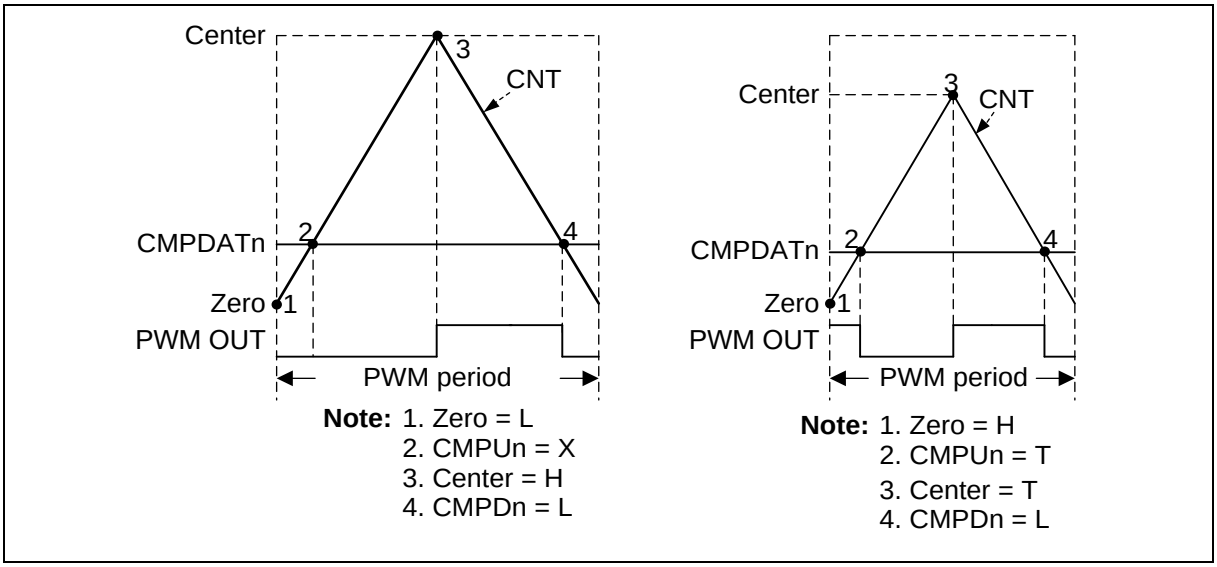


图 6.11-14 BPWM 脉冲发生 (左：不对称脉冲，右：多样化脉冲)

多个事件有时候会发生在同一时刻。这时候，不同的计数方式的事件的优先级如下所示： 表 6.11-2是向上计数方式的优先级、表 6.11-3是向下计数方式的优先级，而表 6.11-4是上下计数方式的优先级。如图 6.11-15所示，通过事件优先级，用户可以轻易产生从 0% 到 100% 占空比的脉冲。

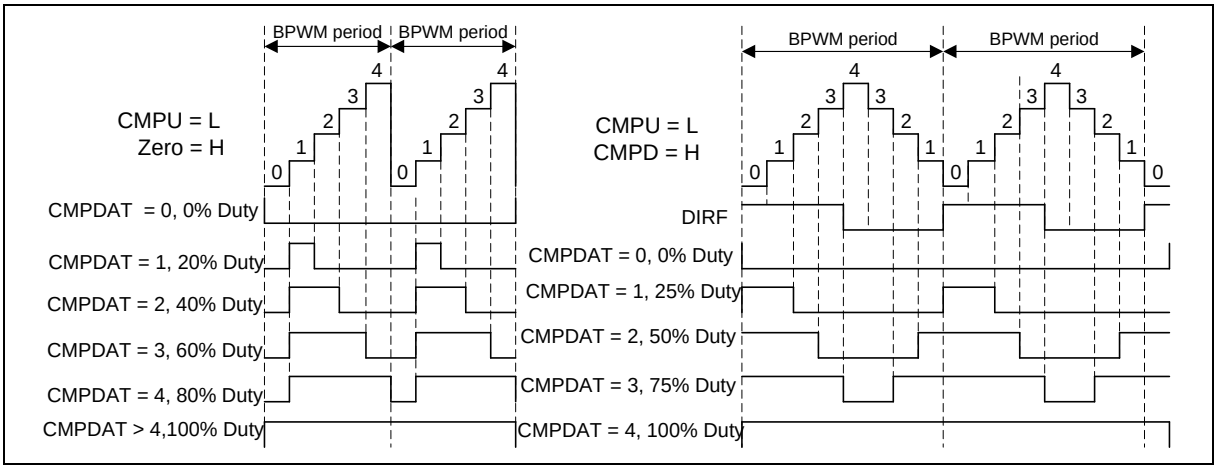


图 6.11-15 BPWM0% 到 100% 脉冲产生 (左:上计数器模式, 右：上下计数器模式)

优先级	向上计数事件
1 (最高)	周期事件(CNT = PERIOD)
2	向上比较事件(CNT = CMPUn)
3 (最低)	零位事件 (CNT = 0)

表 6.11-2 向上计数方式下的 BPWM 脉冲发生事件优先级

优先级	向下计数方式

1 (最高)	零位事件 (CNT = 0)
2	向下比较事件 (CNT = CMPDn)
3 (最低)	周期事件 (CNT = PERIOD)

表 6.11-3 向下计数方式的 BPWM 脉冲发生事件优先级

优先级	向上事件	向下事件
1 (最高)	向上比较事件(CNT = CMPUn)	向下比较事件(CNT = CMPDn)
2(最低)	零点事件(CNT = 0)	周期 (中心) 事件 (CNT =PERIOD)

表 6.11-4 上下计数方式的 BPWM 脉冲发生事件优先级

6.11.5.11 同步功能

如需同时使能 BPWM 和 PWM 计数器，用户需要：先设定 BPWM 同步开始控制寄存器 (BPWM_SSCTL[0]) 以使能希望开始同步计数的计数器通道，接着设定寄存器 SSRC (BPWM_SSCTL[9:8]) 来选择同步开始源，然后再设定 BPWM 同步开始触发寄存器 CNTSEN (BPWM_SSTRG[0])。

6.11.5.12 BPWM输出控制

BPWM 脉冲发生后，可以分三步来控制 BPWM 通道输出。这三个步骤如图 6.11-16所示，分别是：屏蔽、管脚极性和输出使能。

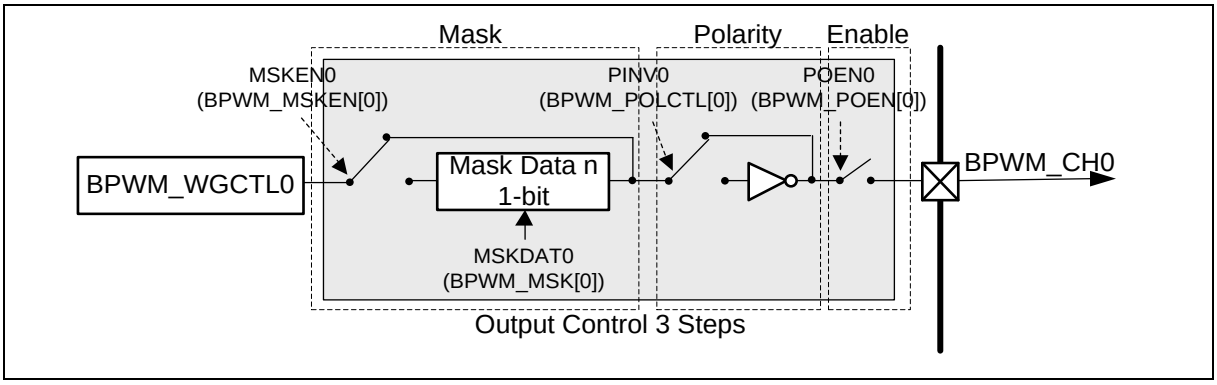


图 6.11-16 BPWM_CH0 输出控制的 3 步

6.11.5.13 BPWM屏蔽输出功能

每个 BPWM 的输出通道都可以通过人为修改 BPWM 屏蔽使能控制寄存器 (BPWM_MSKEN) 和 BPWM 屏蔽数据寄存器 (BPWM_MSK) 的对应位，来输出特定的逻辑电平，该输出可独立于占空比循环比较单元。在驱动类似 BLDC 电机的电力整流电机时，BPWM 的屏蔽位会非常有用。BPWM_MSKEN 寄存器中有 6 位控制位，即MSKENn (BPWM_MSKEN[5:0])，这6位控制位分别决定对应的BPWM 通道输出是否会被修改，这 6 位为高有效。BPWM_MSK 寄存器中有 6 位控制位，即 MSKDATn (BPWM_MSK[5:0]) ，这6位控制位分别决定被 MSKDAT 对应位屏蔽的 BPWM 通道的输出状态。图 6.11-17所示为通过 BPWM 屏蔽功能修改输出的例子。

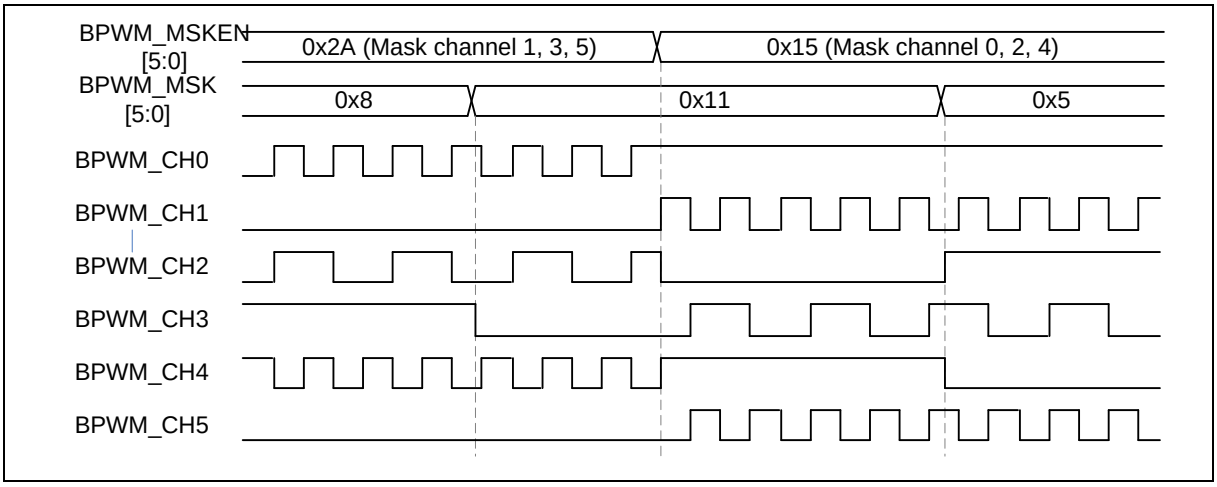
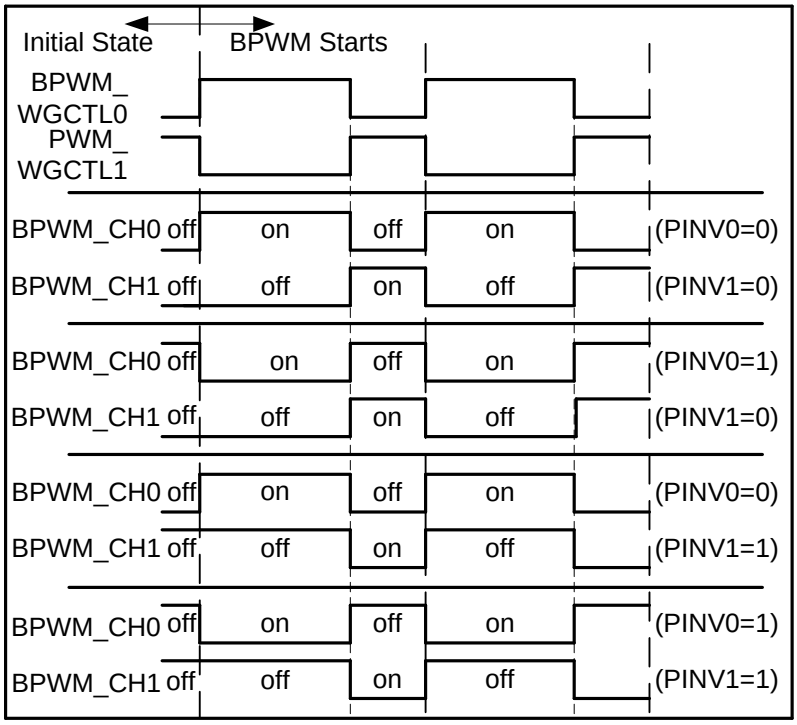


图 6.11-17 屏蔽功能波形说明

6.11.5.14 极性控制

从 BPWM_CH0 到 BPWM_CH5 每个 BPWM 端口都有独立的极性控制模块来配置 BPWM 输出的有效状态的极性。BPWM 默认输出高态有效。也就是说，BPWM 关闭状态为低，打开状态为高。每个独立的 BPWM 通道都可以通过配置 BPWM 阴极极性控制寄存器 (BPWM_POLCTL) 来自由配置有效状态。如图 6.11-18是应用不同极性配置前 BPWM 的初始状态。



Note: PINVx: Negative Polarity control bits; It controls the PWM output initial state and polarity, x denotes 0 or 1.

图 6.11-18 初始状态和极性控制

6.11.5.15 BPWM中断发生器

如图 6.11-19所示，每个 BPWM 都有 2 个独立的中断。

BPWM 中断 (BPWM_INT) 来自于对应的 BPWM 事件。计数器可以产生零点中断标志 ZID0 (BPWM_INTSTS0[0]) 和周期点中断标志 PIF0 (BPWM_INTSTS0[8])。当 BPWM 通道 n 的计数值和 BPWM_CMPDATn 里的比较值相同时，会根据计数的方向触发不同的中断标志。如果发生在向上计数时，向上中断标志 CMPUIFn (BPWM_INSTS0[21:16]) 会被置位；如果发生在向下计数时，向下中断标志 CMPDIFn (BPWM_INSTS0[29:24]) 会被置位。此时如果对应的中断使能位置位，就会产生相应的中断信号。

另一个中断是捕捉中断 (CAP_INT)，该中断在 NVIC 中和 BPWM 中断共用 BPWM_INT 向量。CAP_INT 在以下条件下会被触发：上升沿时，CAPRIFn (BPWM_CAPIF[5:0]) 触发且上升沿捕捉中断使能位 CAPRIENn (BPWM_CAPIEN[5:0]) 置 1；下降沿时，CAPFIFn (BPWM_CAPIF[13:8])触发且下降沿中断使能位 CAPFIENn (BPWM_CAPIEN[13:8]) 置 1。

图 6.11-19所示为 BPWM 中断的架构图。

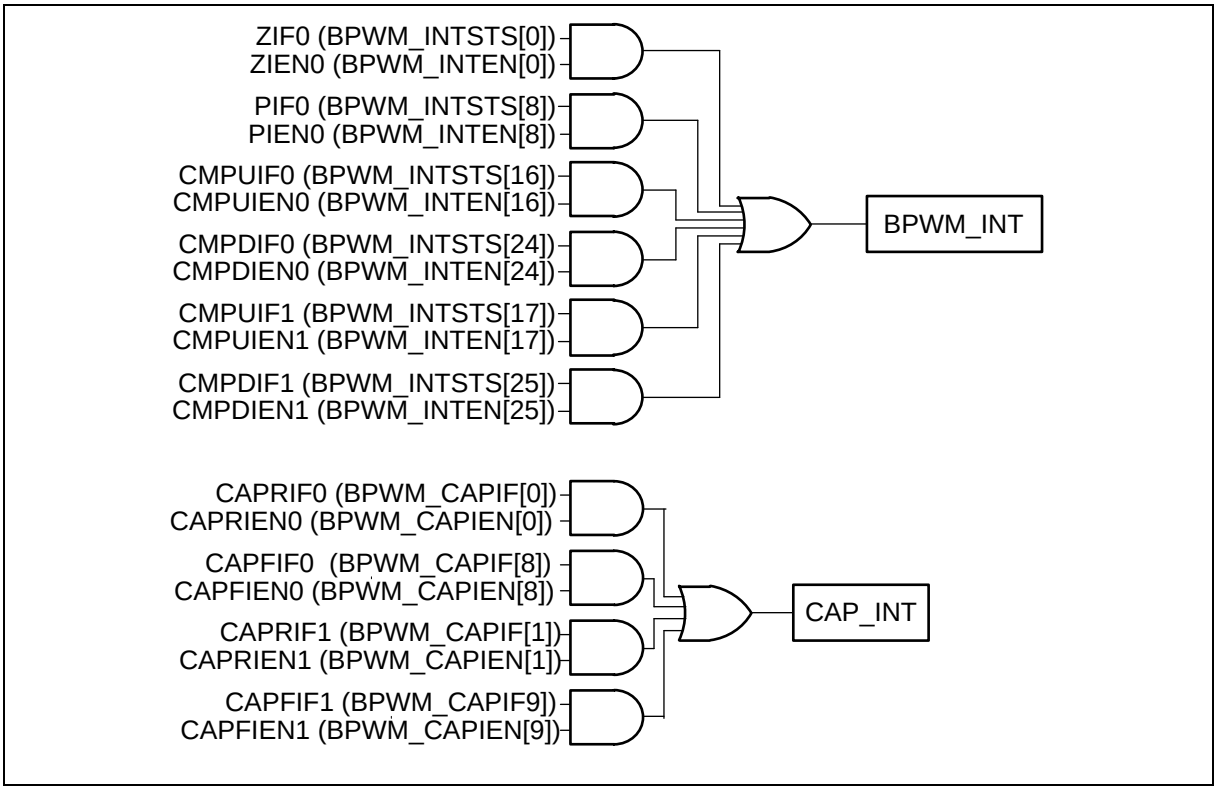


图 6.11-19 BPWM_CH0 和 BPWM_CH1 对中断架构图

6.11.5.16 BPWM触发ADC发生器

BPWM 可以作为ADC转换的触发源。每对 BPWM 通道共享一个触发源。用户可以通过设置 TRGSELn 选择触发源，TRGSELn 指 TRGSEL0、TRGSEL1 …和 TRGESL5，对应位置分别为 BPWM_ADCTS0[3:0]、BPWM_ADCTS0[11:8]、BPWM_ADCTS0[19:16]、BPWM_ADCTS0[27:24]、BPWM_ADCTS1[3:0] 和 BPWM_ADCTS1[11:8]。设置 TRGENn可以使能触发输出到 ADC，TRGENn 指 TRGEN0、TRGEN1 …和 TRGEN5，分别对应到 BPWM_ADCTS0[7]、BPWM_ADCTS0[15]、BPWM_ADCTS0[23]、BPWM_ADCTS0[31]、BPWM_ADCTS1[7] 和 BPWM_ADTS1[15]。其中，n (n = 0,1,…5) 对应 BPWM 通道数。

一对 BPWM 通道有 7 个事件可以选作触发源，图 6.11-20中就以BPWM_CH0 和 BPWM_CH1 为例。

通过设置PERIOD 和 CMPDAT，BPWM可以在不同的时刻触发ADC启动转换。图 6.11-22为上下计数方式下触发ADC的时序波形。

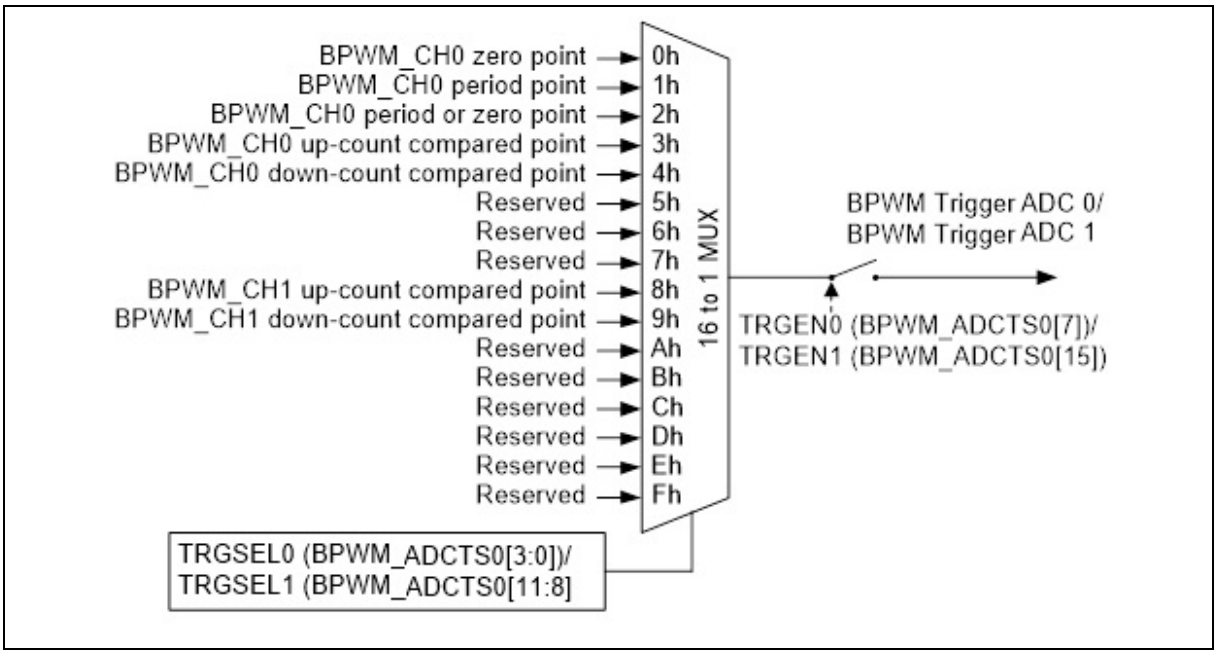


图 6.11-20 BPWM_CH0 和 BPWM_CH1 触发 ADC 源的框架图

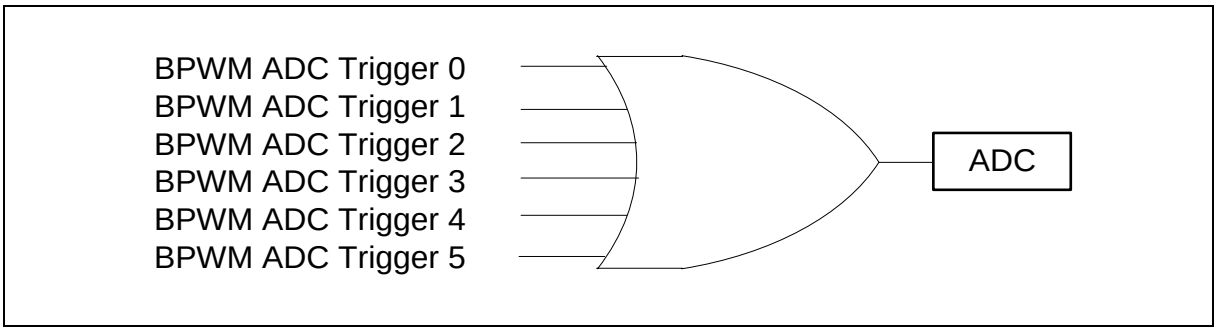


图 6.11-21 BPWM CH0~ CH5 触发 ADC 的框架图

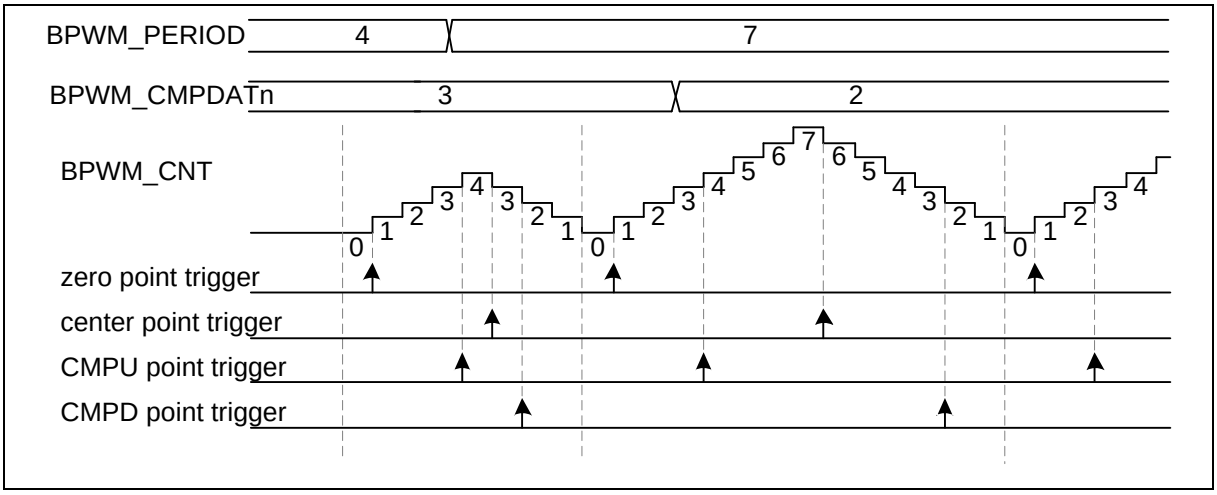


图 6.11-22 上下计数方式下 BPWM 触发 ADC 的时序波形图

6.11.5.17 捕捉操作

输入捕捉通道和 BPWM 输出通道共享相同的管脚和计数器。这里，计数器可以工作在向上或者向下计数方式。捕捉功能会占有 BPWM 计数器，在输入通道有上升沿时锁存 BPWM 计数器到寄存器 RCAPDATn (BPWM_RCAPDATn[15:0])，在输入通道有下降沿时锁存 BPWM 计数器到寄存器 FCAPDATn (BPWM_FCAPDATn[15:0])。捕捉功能在对应的上升沿或者下降沿使能位使能时，上升沿或者下降沿发生时会产生 CAP_INT (使用 BPWM_INT 向量) 中断，其中 CAPRIENn (BPWM_CAPIEN[5:0]) 是上升沿的使能位，CAPFIENn (BPWM_CAPIEN[13:8]) 是下降沿的使能位。当上升沿锁存发生时，对应的 BPWM 计数器会重载 BPWM_PERIODD 的值，这部分由 RCRLDENn 或者 FCRLDENn 决定 (RCRLDENn 和 FCRLDENn 分别位于寄存器 BPWM_CAPCTL[21:16] 和 BPWM_CAPCTL[29:24]) 。值得注意的是，需要使能捕捉通道 n 对应的 CAPINENn (BPWM_CAPINEN[5:0]) 寄存器位来配置管脚的捕捉功能。图 6.11-23是通道 0 的捕捉框图。

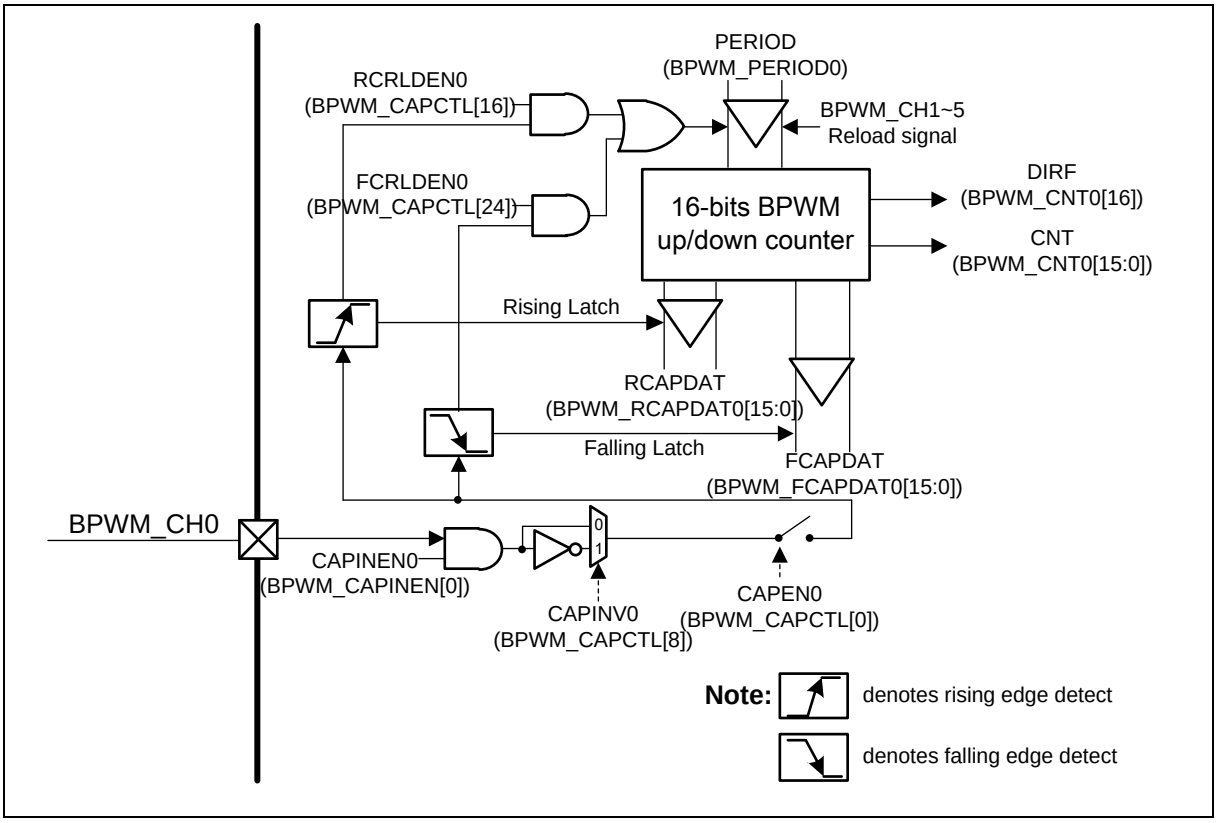


图 6.11-23 BPWM_CH0 捕捉框图

图 6.11-24是捕捉功能的时序示例。此时，捕捉计数器处于 BPWM 向下计数方式，PERIOD 为 8，计数器会从 8 到 0 向下计数。当检测到捕捉管脚的下降沿时，捕捉功能会锁存计数器的值到 BPWM_RCAPDATn。在时序图中，第一次检测到下降沿时，由于 FCRLDENn 被使能，捕捉功能会从 PERIOD 的配置值重载计数器的值。但在第二次时，由于禁止了 FCRLDENn，下降沿就不会引起计数器重载了。在这个例子里，由于使能了 RCRLDENn，计数器的值在上升沿时也会被重载。

另外，对于向上计数方式，计数器的值会被重载为 0 再向上计数到 PERIOD 的值。需要特别注意的是，所有通道共享一个计数器，所以计数器重载的时间点也是由所有通道的重载信号一起控制的。

图 6.11-24还展示了中断和中断标志发生的时序。当通道 n 检测到了上升沿时，对应的 CAPRIFn (BPWM_CAPIF[5:0]) 位由硬件置位。类似的，当通道 n 检测到了下降沿时，对应的 CAPFIFn (BPWM_CAPIF[13:8]) 位由硬件置位。CAPRIFn (BPWM_CAPIF[5:0]) 和 CAPFIFn (BPWM_CAPIF[13:8]) 由软件写 1 清 0。如果 CAPRIFn (BPWM_CAPIF[5:0]) 且 CAPRIENn 使能，捕捉功能会产生一个中断。如果 CAPFIFn (BPWM_CAPIF[13:8]) 置位且 CAPFIENn 使能，同样也会产生一个中断。

图中没有列出的一种情况是：如果在 CAPRIFn (BPWM_CAPIF[5:0]) 刚置位时，上升沿锁存又发生了，过载状态寄存器 CRIFOVn (BPWM_CAPSTS[5:0]) 会由硬件置 1 来表示 CAPRIFn (BPWM_CAPIF[5:0]) 过载了。类似的，如果下降沿锁存再次发生了，同样的情况会发生在过载状态寄存器 CRIFOVn (BPWM_CAPSTS[13:8]) 和 CAPFIFn (BPWM_CAPIF[13:8]) 寄存器。

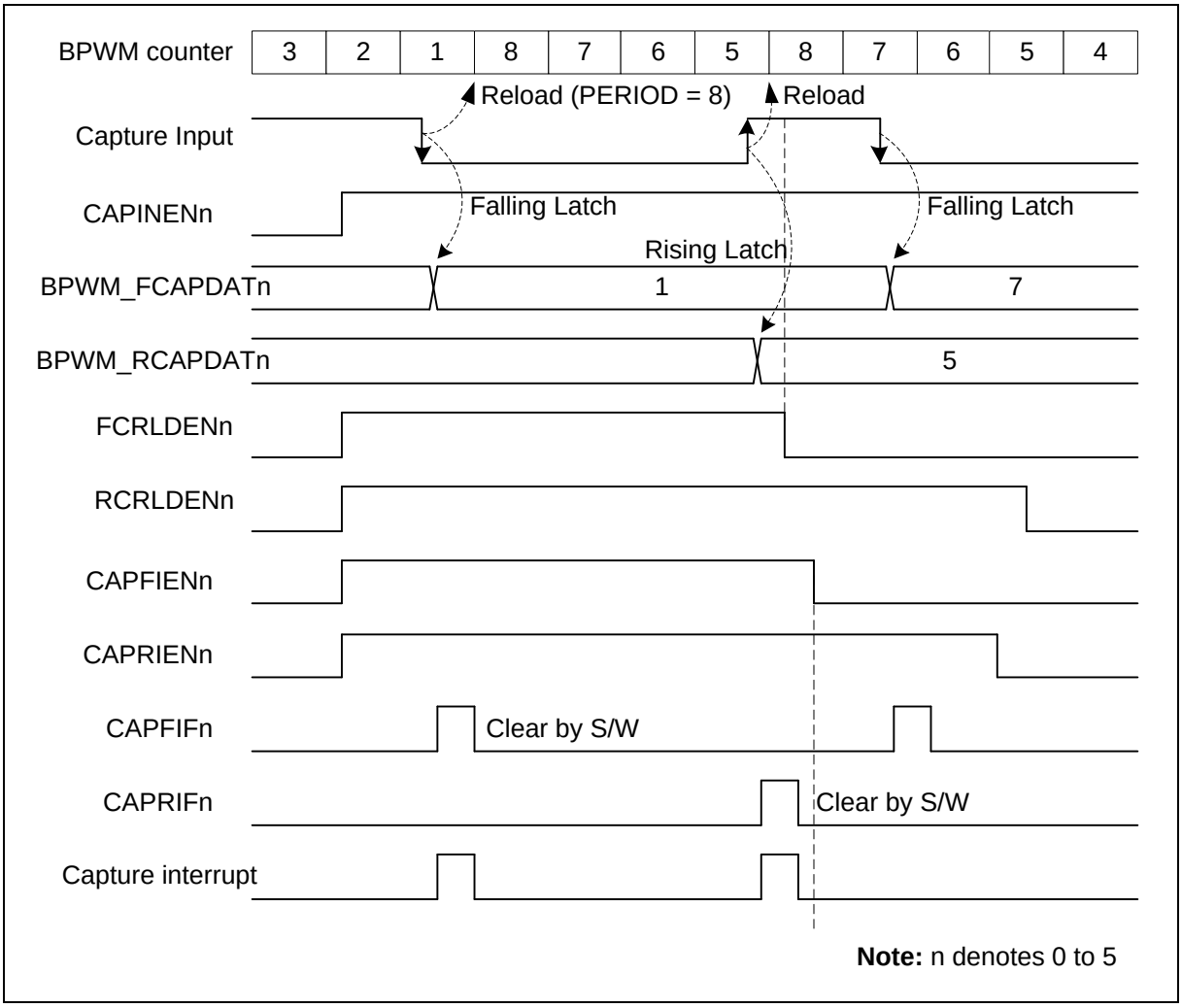


图 6.11-24 捕捉操作波形图

满足以下条件的捕获脉冲可按如下公式计算脉冲宽度：

1. 捕获的正或负脉冲宽度短于计数器周期；

2.计数器以向下计数器类型运行；

3. PWM_CAPCTL寄存器的FCRLDENn和RCRLDENn位设置为1，在下降和上升捕获事件可以重新加载计数器。

对于负脉冲，通道上的低脉冲宽度为：(BPWM_PERIOD + 1 - BPWM_RCAPDATn) 个 BPWM 计数器时间，其中 BPWM 计数器时间为(CLKPSC+1) * BPWMx_CLK 时钟时间。如图 6.11-24，低脉冲宽度为 8+1-5 = 4 个 BPWM 计数器时间。

对于正脉冲，通道上的高脉冲宽度为：(BPWM_PERIOD + 1 - BPWM_ FCAPDATn) 个 BPWM 计数器时间，其中 BPWM 计数器时间为(CLKPSC+1) * BPWMx_CLK 时钟时间。如图 6.11-24，低脉冲宽度为 8+1-7 = 2 个 BPWM 计数器时间。

6.11.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读/写

寄存器	偏移量	R/W	描述	复位值
BPWM基地址: **BPWM0_BA = 0x4005_A000** **BPWM1_BA = 0x4005_B000**				
BPWM_CTL0 **x=0, 1**	BPWMx_BA+0x00	R/W	BPWM 控制寄存器0	0x0000_0000
BPWM_CTL1 **x=0, 1**	BPWMx_BA+0x04	R/W	BPWM 控制寄存器1	0x0000_0000
BPWM_CLKSRC **x=0, 1**	BPWMx_BA+0x10	R/W	BPWM 时钟源寄存器	0x0000_0000
BPWM_CLKPSC **x=0, 1**	BPWMx_BA+0x14	R/W	BPWM 时钟预分频寄存器	0x0000_0000
BPWM_CNTEN **x=0, 1**	BPWMx_BA+0x20	R/W	BPWM 计数器使能寄存器	0x0000_0000
BPWM_CNTCLR **x=0, 1**	BPWMx_BA+0x24	R/W	BPWM 清除计数器寄存器	0x0000_0000
BPWM_PERIOD **x=0, 1**	BPWMx_BA+0x30	R/W	BPWM 周期寄存器	0x0000_0000
BPWM_CMPDAT0 **x=0, 1**	BPWMx_BA+0x50	R/W	BPWM 比较值寄存器0	0x0000_0000
BPWM_CMPDAT1 **x=0, 1**	BPWMx_BA+0x54	R/W	BPWM 比较值寄存器1	0x0000_0000
BPWM_CMPDAT2 **x=0, 1**	BPWMx_BA+0x58	R/W	BPWM 比较值寄存器2	0x0000_0000
BPWM_CMPDAT3 **x=0, 1**	BPWMx_BA+0x5C	R/W	BPWM 比较值寄存器3	0x0000_0000
BPWM_CMPDAT4 **x=0, 1**	BPWMx_BA+0x60	R/W	BPWM 比较值寄存器4	0x0000_0000
BPWM_CMPDAT5 **x=0, 1**	BPWMx_BA+0x64	R/W	BPWM 比较值寄存器5	0x0000_0000
BPWM_CNT **x=0, 1**	BPWMx_BA+0x90	R	BPWM 计数器寄存器	0x0000_0000
BPWM_WGCTL0 **x=0, 1**	BPWMx_BA+0xB0	R/W	BPWM 发生器寄存器0	0x0000_0000
BPWM_WGCTL1 **x=0, 1**	BPWMx_BA+0xB4	R/W	BPWM 发生器寄存器1	0x0000_0000
BPWM_MSKEN **x=0, 1**	BPWMx_BA+0xB8	R/W	BPWM 屏蔽使能寄存器	0x0000_0000

BPWM_MSK x=0, 1	BPWMx_BA+0xBC	R/W	BPWM 屏蔽数据寄存器	0x0000_0000
BPWM_POLCTL x=0, 1	BPWMx_BA+0xD4	R/W	BPWM 管脚极性反转寄存器	0x0000_0000
BPWM_POEN x=0, 1	BPWMx_BA+0xD8	R/W	BPWM 输出使能寄存器	0x0000_0000
BPWM_INTEN x=0, 1	BPWMx_BA+0xE0	R/W	BPWM 中断使能寄存器	0x0000_0000
BPWM_INTSTS x=0, 1	BPWMx_BA+0xE8	R/W	BPWM 中断标志寄存器	0x0000_0000
BPWM_ADCTS0 x=0, 1	BPWMx_BA+0xF8	R/W	BPWM 触发ADC源选择寄存器0	0x0000_0000
BPWM_ADCTS1 x=0, 1	BPWMx_BA+0xFC	R/W	BPWM 触发ADC源选择寄存器1	0x0000_0000
BPWM_SSCTL x=0, 1	BPWMx_BA+0x110	R/W	BPWM 同步开始控制寄存器	0x0000_0000
BPWM_SSTRG x=0, 1	BPWMx_BA+0x114	W	BPWM 同步开始触发寄存器	0x0000_0000
BPWM_STATUS x=0, 1	BPWMx_BA+0x120	R/W	BPWM 状态寄存器	0x0000_0000
BPWM_CAPINEN x=0, 1	BPWMx_BA+0x200	R/W	BPWM 输入捕捉使能寄存器	0x0000_0000
BPWM_CAPCTL x=0, 1	BPWMx_BA+0x204	R/W	BPWM 捕捉控制寄存器	0x0000_0000
BPWM_CAPSTS x=0, 1	BPWMx_BA+0x208	R	BPWM 捕捉状态寄存器	0x0000_0000
BPWM_RCAPDAT0 x=0, 1	BPWMx_BA+0x20C	R	BPWM 上升沿捕捉数据寄存器0	0x0000_0000
BPWM_FCAPDAT0 x=0, 1	BPWMx_BA+0x210	R	BPWM 下降沿捕捉数据寄存器0	0x0000_0000
BPWM_RCAPDAT1 x=0, 1	BPWMx_BA+0x214	R	BPWM 上升沿捕捉数据寄存器1	0x0000_0000
BPWM_FCAPDAT1 x=0, 1	BPWMx_BA+0x218	R	BPWM 下降沿捕捉数据寄存器1	0x0000_0000
BPWM_RCAPDAT2 x=0, 1	BPWMx_BA+0x21C	R	BPWM 上升沿捕捉数据寄存器2	0x0000_0000
BPWM_FCAPDAT2 x=0, 1	BPWMx_BA+0x220	R	BPWM 下降沿捕捉数据寄存器2	0x0000_0000
BPWM_RCAPDAT3 x=0, 1	BPWMx_BA+0x224	R	BPWM 上升沿捕捉数据寄存器3	0x0000_0000
BPWM_FCAPDAT3	BPWMx_BA+0x228	R	BPWM 下降沿捕捉数据寄存器3	0x0000_0000

x=0, 1				
BPWM_RCAPDAT4 x=0, 1	BPWMx_BA+0x22C	R	BPWM 上升沿捕捉数据寄存器4	0x0000_0000
BPWM_FCAPDAT4 x=0, 1	BPWMx_BA+0x230	R	BPWM 下降沿捕捉数据寄存器4	0x0000_0000
BPWM_RCAPDAT5 x=0, 1	BPWMx_BA+0x234	R	BPWM 上升沿捕捉数据寄存器5	0x0000_0000
BPWM_FCAPDAT5 x=0, 1	BPWMx_BA+0x238	R	BPWM 下降沿捕捉数据寄存器5	0x0000_0000
BPWM_CAPIEN x=0, 1	BPWMx_BA+0x250	R/W	BPWM 捕捉中断使能寄存器	0x0000_0000
BPWM_CAPIF x=0, 1	BPWMx_BA+0x254	R/W	BPWM 捕捉中断标志寄存器	0x0000_0000
BPWM_PBUF x=0, 1	BPWMx_BA+0x304	R	BPWM 周期缓存	0x0000_0000
BPWM_CMPBUF0 x=0, 1	BPWMx_BA+0x31C	R	BPWM CMPDAT 0 缓存	0x0000_0000
BPWM_CMPBUF1 x=0, 1	BPWMx_BA+0x320	R	BPWM CMPDAT 1 缓存	0x0000_0000
BPWM_CMPBUF2 x=0, 1	BPWMx_BA+0x324	R	BPWM CMPDAT 2 缓存	0x0000_0000
BPWM_CMPBUF3 x=0, 1	BPWMx_BA+0x328	R	BPWM CMPDAT 3 缓存	0x0000_0000
BPWM_CMPBUF4 x=0, 1	BPWMx_BA+0x32C	R	BPWM CMPDAT 4 缓存	0x0000_0000
BPWM_CMPBUF5 x=0, 1	BPWMx_BA+0x330	R	BPWM CMPDAT 5 缓存	0x0000_0000

6.11.7 寄存器描述

BPWM控制寄存器 0 (BPWM_CTL0)

寄存器	偏移	R/W	描述	复位值
BPWM_CTL0	BPWMx_BA+0x00	R/W	BPWM 控制寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	**DBGHALT**	**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**IMMLDEN5**	**IMMLDEN4**	**IMMLDEN3**	**IMMLDEN2**	**IMMLDEN1**	**IMMLDEN0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CTRLD5**	**CTRLD4**	**CTRLD3**	**CTRLD2**	**CTRLD1**	**CTRLD0**

位	描述	
[31]	**DBGTRIOFF**	**ICE 调试模式应答禁止 (写保护)** 0 = ICE 调试模式下的应答会影响 BPWM 输出。此时，BPWM 管脚会强制设为三态模式 1 = ICE 调试模式下的应答不影响BPWM输出。无论 ICE 调试模式是否应答，BPWM 管脚都会保持输出 **注：**该位写保护，具体请参考 SYS_REGLCTL 寄存器。
[30]	**DBGHALT**	**ICE 调试模式计数器停止(写保护)** 如果使能调试模式下关闭计数器，BPWM 所有计数器都会保持当前值直到退出 ICE 调试模式。 0 =禁止 ICE 调试模式下关闭计数器 1 =使能 ICE 调试模式下关闭计数器 **注：**该位写保护，具体请参考 SYS_REGLCTL 寄存器。
[29:22]	**Reserved**	保留
[16+n] n=0,1..5	**IMMLDENn**	**立即重载使能位** 每一位 n 控制着对应的 BPWM 通道n 0 = 在每个周期结束点重载PERIOD到 PBUF。通过设定 CTRLD 位，CMPDAT 会在中间点或者结束点重载到 CMPBUF。 1 = 软件更新 PERIOD/CMPDAT 时，PERIOD/CMPDAT 会立即重载到 PBUF 和 CMPBUF。 **注：**如果 IMMLDENn 使能， WINLDENn和CTRLDn 无效。
[15:6]	**Reserved**	保留
[n] n=0,1..5	**CTRLDn**	**中心重载** 每一位 n 控制着对应的 BPWM 通道n 上下计数方式时， PERIOD 会在每个周期结束位置重载到 PBUF。CMPDAT 则会在周期中

		间位置重载到 CMPBUF。

BPWM控制寄存器 1 (BPWM_CTL1)

寄存器	偏移	R/W	描述	复位值
BPWM_CTL1	BPWMx_BA+0x04	R/W	BPWM控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**CNTTYPE0**	

位	描述	
[31:2]	**Reserved**	保留
[1:0]	**CNTTYPE0**	**BPWM 计数器规则类型 0** 每一位 n 控制着对应的 BPWM 通道n 00 = 向上计数方式 (支持捕捉模式) 01 = 向下计数方式 (支持捕捉模式) 10 = 上下计数方式 11 = 保留

BPWM时钟源寄存器(BPWM_CLKSRC)

寄存器	偏移	R/W	描述	复位值
BPWM_CLKSRC	BPWMx_BA+0x10	R/W	BPWM时钟源寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**ECLKSRC0**		

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**ECLKSRC0**	**BPWM_CH01 外部时钟源选择** 000 = BPWMx_CLK， x 指 0 或 1 001 = TIMER0 溢出 010 = TIMER1 溢出 011 = TIMER2 溢出 100 = TIMER3 溢出 其他 = 保留

BPWM时钟分频寄存器(BPWM_CLKPSC)

寄存器	偏移	R/W	描述	复位值
BPWM_CLKPSC	BPWMx_BA+0x14	R/W	BPWM时钟分频寄存器	0x0000_0000

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

位	描述	
[31:12]	**Reserved**	保留
[11:0]	**CLKPSC**	**BPWM 计数器时钟分频** BPWM计数器的时钟会被分频器除频，每对 BPWM共享一个 BPWM 计数器时钟分频器，BPWM 计数器时钟需除以 (CLKPSC+ 1)。

BPWM计数器使能寄存器(BPWM_CNTEN)

寄存器	偏移	R/W	描述	复位值
BPWM_CNTEN	BPWMx_BA+0x20	R/W	BPWM 计数器使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**CNTEN0**

位	描述	
[31:1]	**Reserved**	保留
[0]	**CNTEN0**	**BPWM 计数器 0 使能位** 0 = BPWM 计数器和时钟分频器停止运行 1 = BPWM 计数器和时钟分频器开始运行

BPWM清除计数器寄存器(BPWM_CNTCLR)

寄存器	偏移	R/W	描述	复位值
BPWM_CNTCLR	BPWMx_BA+0x24	R/W	BPWM清除计数器寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**CNTCLR0**

位	描述	
[31:1]	**Reserved**	保留
[0]	**CNTCLR0**	**清除 BPWM 计数器控制位 0** 硬件自动清除 0 = 无效 1 = 清除 16 位的 BPWM 计数器到 0000H

BPWM周期寄存器(BPWM_PERIOD)

寄存器	偏移	R/W	描述	复位值
BPWM_PERIOD	BPWMx_BA+0x30	R/W	BPWM周期寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**PERIOD**	**BPWM 周期寄存器** 向上计数模式：该模式下，BPWM 计数器从 0 计数到 PERIOD，再重新从 0 计数 向下计数模式：该模式下，BPWM 计数器从 PERIOD 计数到 0，再重新从 PERIOND 计数 以上，BPWM 周期时间 = (PERIOD+1) * BPWM_CLK 周期 上下计数模式：该模式下，BPWM 计数器从 0 计数到 PERIOD，再向下递减到 0 ，依次循环，BPWM 周期时间= 2 * PERIOD * BPWM_CLK周期

BPWM比较器寄存器0~5 (BPWM_CMPDAT0~5)

寄存器	偏移	R/W	描述	复位值
BPWM_CMPDAT0	BPWMx_BA+0x50	R/W	BPWM 比较值寄存器0	0x0000_0000
BPWM_CMPDAT1	BPWMx_BA+0x54	R/W	BPWM 比较值寄存器1	0x0000_0000
BPWM_CMPDAT2	BPWMx_BA+0x58	R/W	BPWM 比较值寄存器2	0x0000_0000
BPWM_CMPDAT3	BPWMx_BA+0x5C	R/W	BPWM 比较值寄存器3	0x0000_0000
BPWM_CMPDAT4	BPWMx_BA+0x60	R/W	BPWM 比较值寄存器4	0x0000_0000
BPWM_CMPDAT5	BPWMx_BA+0x64	R/W	BPWM 比较值寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**CMPDAT**	**BPWM 比较值寄存器** CMPDAT 用来和 CNTR 比较从而产生 BPWM 波形、中断和触发 ADC 信号 在独立模式下，CMPDAT0~5 代表 6 路 BPWM_CH0~5 的比较点

BPWM计数器寄存器(BPWM_CNT)

寄存器	偏移	R/W	描述	复位值
BPWM_CNT	BPWMx_BA+0x90	R	BPWM计数器寄存器	0x0000_0000

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

位	描述	
[31:17]	**Reserved**	保留
[16]	**DIRF**	**BPWM 方向指示标志 (只读)** 0 = 计数器向下计数 1 = 计数器向上计数
[15:0]	**CNT**	**BPWM 数据寄存器 (只读)** 用户可以从 CNTR寄存器读到当前16位周期计数器的值

BPWM发生器寄存器 0 (BPWM_WGCTL0)

寄存器	偏移	R/W	描述	复位值
BPWM_WGCTL0	BPWMx_BA+0xB0	R/W	BPWM发生器寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**PRDPCTL5**		**PRDPCTL4**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRDPCTL3		**PRDPCTL2**		**PRDPCTL1**		**PRDPCTL0**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**ZPCTL5**		**ZPCTL4**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ZPCTL3		**ZPCTL2**		**ZPCTL1**		**ZPCTL0**	

位	描述	
[31:28]	**Reserved**	保留
[16+2n+1:16+2n] n=0,1..5	**PRDPCTLn**	**BPWM 周期 (中间) 位置控制** 每一位 n 控制着对应的 BPWM 通道n 00 = 无动作 01 = BPWM 周期 (中间) 位置输出低 10 = BPWM 周期 (中间) 位置输出高 11 = BPWM 周期 (中间) 位置输出触发 计数器计数到 (PERIOD+1) 时，BPWM可以控制输出电平。 **注：**该位为BPWM 计数器上下计数方式时的中间位置输出电平控制位
[15:12]	**Reserved**	保留
[2n+1:2n] n=0,1..5	**ZPCTLn**	**BPWM 零位控制** 每一位 n 控制着对应的 BPWM 通道 n 00 = 无动作 01 = BPWM 零位输出低 10 = BPWM 零位输出高 11 = BPWM 零位输出触发 BPWM 可以在 BPWM 计数器计数到 0 时控制输出电平

BPWM发生器寄存器 1(BPWM_WGCTL1)

寄存器	偏移	R/W	描述	复位值
BPWM_WGCTL1	BPWMx_BA+0xB4	R/W	BPWM发生器寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**CMPDCTL5**		**CMPDCTL4**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDCTL3		**CMPDCTL2**		**CMPDCTL1**		**CMPDCTL0**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CMPUCTL5**		**CMPUCTL4**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPUCTL3		**CMPUCTL2**		**CMPUCTL1**		**CMPUCTL0**	

位	描述	
[31:28]	**Reserved**	保留
[16+2n+1:16+2n] n=0,1..5	**CMPDCTLn**	**BPWM 向下比较位置控制** 每一位 n 控制着对应的 BPWM 通道n 00 = 无动作 01 = BPWM 向下比较位置输出低 10 = BPWM 向下比较位置输出高 11 = BPWM 向下比较位置输出触发 BPWM 可以在向下计数到 CMPDAT 时控制输出电平。
[15:12]	**Reserved**	保留
[2n+1:2n] n=0,1..5	**CMPUCTLn**	**BPWM 向上比较位置控制** 每一位 n 控制着对应的 BPWM 通道n 00 = 无动作 01 = BPWM 向上比较位置输出低 10 = BPWM 向上比较位置输出高 11 = BPWM 向上比较位置输出触发 BPWM 可以在向上计数到 CMPDAT 时控制输出电平。

BPWM屏蔽使能寄存器(BPWM_MSKEN)

寄存器	偏移	R/W	描述	复位值
BPWM_MSKEN	BPWMx_BA+0xB8	R/W	BPWM屏蔽使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKEN5**	**MSKEN4**	**MSKEN3**	**MSKEN2**	**MSKEN1**	**MSKEN0**

位	描述	
[31:6]	**Reserved**	保留
[n] n=0,1..5	**MSKENn**	**BPWM 屏蔽使能位** 每一位 n 控制着对应的 BPWM 通道 n 该位使能时 BPWM 输出信号被屏蔽。对应的 BPWM 通道 n 会输出 MSKDATn (BPWM_MSK[5:0]) 的值 0 = BPWM 输出信号未被屏蔽 1 = BPWM 输出信号被屏蔽，输出 MSKDATn 的值

BPWM屏蔽数据寄存器(BPWM_MSK)

寄存器	偏移	R/W	描述	复位值
BPWM_MSK	BPWMx_BA+0xBC	R/W	BPWM屏蔽数据寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKDAT5**	**MSKDAT4**	**MSKDAT3**	**MSKDAT2**	**MSKDAT1**	**MSKDAT0**

位	描述	
[31:6]	**Reserved**	保留
[n] n=0,1..5	**MSKDATn**	**BPWM 屏蔽数据位** 如果对应的管脚屏蔽功能使能，该数据位控制 BPWM 管脚输出状态 每一位 n 控制着对应的 BPWM 通道 n 0 = BPWMn 输出逻辑电平低 1 = BPWMn 输出逻辑电平高

BPWM管脚极性反转寄存器(BPWM_POLCTL)

寄存器	偏移	R/W	描述	复位值
BPWM_POLCTL	BPWMx_BA+0xD4	R/W	BPWM管脚极性反转寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**PINV5**	**PINV4**	**PINV3**	**PINV2**	**PINV1**	**PINV0**

位	描述	
[31:6]	**Reserved**	保留
[n] n=0,1..5	**PINVn**	**BPWM 管脚极性反转控制** 该寄存器控制 BPWM 输出极性的状态 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止BPWM 输出极性反转 1 = 使能BPWM 输出极性反转

BPWM输出使能寄存器(BPWM_POEN)

寄存器	偏移	R/W	描述	复位值
BPWM_POEN	BPWMx_BA+0xD8	R/W	BPWM输出使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**POEN5**	**POEN4**	**POEN3**	**POEN2**	**POEN1**	**POEN0**

位	描述	
[31:6]	**Reserved**	保留
[n] n=0,1..5	**POENn**	**BPWM 管脚输出使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = BPWM 管脚处于三态模式 1 = BPWM 管脚处于输出模式

BPWM中断使能寄存器(BPWM_INTEN)

寄存器	偏移	R/W	描述	复位值
BPWM_INTEN	BPWMx_BA+0xE0	R/W	BPWM中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
23	22	21	20	19	18	17	16
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
15	14	13	12	11	10	9	8
Reserved							PIEN0
7	6	5	4	3	2	1	0
Reserved							ZIEN0

位	描述	
[31:30]	**Reserved**	保留
[24+n] n=0,1..5	**CMPDIENn**	**BPWM 向下计数中断使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止向下计数中断 1 =使能向下计数中断
[23:22]	**Reserved**	保留
[16+n] n=0,1..5	**CMPUIENn**	**BPWM 向上计数中断使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 =禁止向上计数中断 1 =使能向上计数中断
[15:9]	**Reserved**	保留
[8]	**PIEN0**	**BPWM 周期位置中断 0 使能位** 0 = 禁止周期位置中断 1 =使能周期位置中断 **注:**上下计数方式里周期位置指中间位置
[7:1]	**Reserved**	保留
[0]	**ZIEN0**	**BPWM 零位中断 0 使能位** 0 = 禁止零位中断 1 = 使能零位中断

BPWM中断标志寄存器(BPWM_INTSTS)

寄存器	偏移	R/W	描述	复位值
BPWM_INTSTS	BPWMx_BA+0xE8	R/W	BPWM中断标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		**CMPDIF5**	**CMPDIF4**	**CMPDIF3**	**CMPDIF2**	**CMPDIF1**	**CMPDIF0**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**CMPUIF5**	**CMPUIF4**	**CMPUIF3**	**CMPUIF2**	**CMPUIF1**	**CMPUIF0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**PIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ZIF0**

位	描述	
[31:30]	**Reserved**	保留
[24+n] n=0,1..5	**CMPDIFn**	**BPWM 向下比较计数中断标志位** 每一位 n 控制着对应的 BPWM 通道 n BPWM 向下计数到 BPWM_CMPDATn 时，硬件置位该标志。软件可写 1 清零该位。 **注：**如果 CMPDAT 等于 PERIOD，该标志在向下计数方式时无效
[23:22]	**Reserved**	保留
[16+n] n=0,1..5	**CMPUIFn**	**BPWM 向上比较计数中断标志** 每一位 n 控制着对应的 BPWM 通道 n BPWM 向上计数到 BPWM_CMPDATn 时，硬件置位该标志。软件可写 1 清 0 该位。 **注：**如果 CMPDAT 等于 PERIOD，该标志在向上计数方式时无效.
[15:9]	**Reserved**	保留
[8]	**PIF0**	**BPWM 周期位置中断标志 0** BPWM_CH0 向上计数到 BPWM_PERIOD0 时，硬件置位该位。软件可写 1 清 0 该位。
[7:1]	**Reserved**	保留
[0]	**ZIF0**	**BPWM 零位中断标志 0** BPWM_CH0 向上计数到 0 时，硬件置位该位。软件可写 1 清 0 该位。

BPWM触发 ADC源选择寄存器 0 (BPWM_ADCTS0)

寄存器	偏移	R/W	描述	复位值
BPWM_ADCTS0	BPWMx_BA+0xF8	R/W	BPWM触发 ADC源选择寄存器 0	0x0000_0000

31	30	29	28	27	26	25	24
TRGEN3	**Reserved**			**TRGSEL3**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRGEN2	**Reserved**			**TRGSEL2**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN1	**Reserved**			**TRGSEL1**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN0	**Reserved**			**TRGSEL0**			

位	描述	
[31]	**TRGEN3**	BPWM_CH3 触发 ADC使能位
[30:28]	**Reserved**	保留
[27:24]	**TRGSEL3**	**BPWM_CH3 触发 ADC源选择** 0000 = BPWM_CH2 零位 0001 = BPWM_CH2 周期位置 0010 = BPWM_CH2 零或周期位置 0011 = BPWM_CH2 向上计数的 CMPDAT 位置 0100 = BPWM_CH2 向下计数的 CMPDAT 位置 0101 = 保留 0110 = 保留 0111 = 保留 1000 = BPWM_CH3 向上计数的 CMPDAT 位置 1001 = BPWM_CH3 向下计数的 CMPDAT 位置 其他保留
[23]	**TRGEN2**	BPWM_CH2 触发 ADC使能位
[22:20]	**Reserved**	保留
[19:16]	**TRGSEL2**	**BPWM_CH2 触发 ADC源选择** 0000 = BPWM_CH2零位 0001 = BPWM_CH2周期位置 0010 = BPWM_CH2 零位或周期位置 0011 = BPWM_CH2 向上计数的 CMPDAT 位置 0100 = BPWM_CH2 向下计数的 CMPDAT 位置 0101 = 保留 0110 = 保留

		0111 = 保留 1000 = BPWM_CH3 向上计数的 CMPDAT 位置. 1001 = BPWM_CH3 向下计数的 CMPDAT 位置. 其他保留
[15]	**TRGEN1**	BPWM_CH1 触发 ADC使能位
[14:12]	**Reserved**	保留
[11:8]	**TRGSEL1**	**BPWM_CH1 触发 ADC源选择** 0000 = BPWM_CH0 零位 0001 = BPWM_CH0 周期位置 0010 = BPWM_CH0零位或周期位置 0011 = BPWM_CH0 向上计数的 CMPDAT 位置 0100 = BPWM_CH0 向下计数的 CMPDAT 位置 0101 = 保留 0110 = 保留 0111 = 保留 1000 = BPWM_CH1向上计数的 CMPDAT 位置 1001 = BPWM_CH1 向下计数的 CMPDAT 位置 其他保留
[7]	**TRGEN0**	BPWM_CH0 触发 ADC使能位
[6:4]	**Reserved**	保留
[3:0]	**TRGSEL0**	**BPWM_CH0 触发 ADC源选择** 0000 = BPWM_CH0 零位 0001 = BPWM_CH0 周期位置 0010 = BPWM_CH0零位或周期位置 0011 = BPWM_CH0 向上计数的 CMPDAT 位置 0100 = BPWM_CH0 向下计数的 CMPDAT 位置 0101 = 保留 0110 = 保留 0111 = 保留 1000 = BPWM_CH1向上计数的 CMPDAT 位置 1001 = BPWM_CH1 向下计数的 CMPDAT 位置 其他保留

BPWM触发 ADC源选择寄存器 1(BPWM_ADCTS1)

寄存器	偏移	R/W	描述	复位值
BPWM_ADCTS1	BPWMx_BA+0xFC	R/W	BPWM触发ADC源选择寄存器 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN5	**Reserved**			**TRGSEL5**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN4	**Reserved**			**TRGSEL4**			

位	描述	
[31:16]	**Reserved**	保留
[15]	**TRGEN5**	BPWM_CH5 触发 ADC使能位
[14:12]	**Reserved**	保留
[11:8]	**TRGSEL5**	**BPWM_CH5 触发 ADC源选择** 0000 = BPWM_CH4 零位 0001 = BPWM_CH4 周期位置 0010 = BPWM_CH4 零位或周期位置 0011 = BPWM_CH4 向上计数的 CMPDAT 位置 0100 = BPWM_CH4 向下计数的 CMPDAT 位置 0101 = 保留 0110 = 保留 0111 = 保留 1000 = BPWM_CH5 向上计数的 CMPDAT 位置 1001 = BPWM_CH5 向下计数的 CMPDAT 位置 其他保留
[7]	**TRGEN4**	BPWM_CH4 触发ADC使能位
[6:4]	**Reserved**	保留
[3:0]	**TRGSEL4**	**BPWM_CH4 触发 ADC源选择** 0000 = BPWM_CH4 零位 0001 = BPWM_CH4 周期位置 0010 = BPWM_CH4零位或周期位置 0011 = BPWM_CH4 向上计数的 CMPDAT 位置 0100 = BPWM_CH4 向下计数的 CMPDAT 位置

		0101 = 保留 0110 = 保留 0111 = 保留 1000 = BPWM_CH5 向上计数的 CMPDAT 位置 1001 = BPWM_CH5 向下计数的 CMPDAT 位置 其他保留

BPWM同步开始寄存器(BPWM_SSCTL)

寄存器	偏移	R/W	描述	复位值
BPWM_SSCTL	BPWMx_BA+0x110	R/W	BPWM同步开始寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**SSRC**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SSEN0**

位	描述	
[31:10]	**Reserved**	保留
[9:8]	**SSRC**	**BPWM 同步开始源选择** 00 = 同步开始源来自 PWM0 01 = 同步开始源来自 PWM1 10 = 同步开始源来自 BPWM0 11 = 同步开始源来自 BPWM1
[7:1]	**Reserved**	保留
[0]	**SSEN0**	**BPWM 同步开始功能 0 使能位** 同步开始功能使能后，写 BPWM 同步开始使能位 (CNTSEN) 可以使能 BPWM_CH0 计数器使能位 (CNTEN0) 0 = 禁止BPWM 同步开始功能 1 = 使能BPWM 同步开始功能

BPWM同步开始触发寄存器(BPWM_SSTRG)

寄存器	偏移	R/W	描述	复位值
BPWM_SSTRG	BPWMx_BA+0x114	W	BPWM同步开始触发寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**CNTSEN**	**BPWM 计数器同步开始使能位 (只写)** BPWM 计数器同步使能功能可以使 PWM 和 BPWM 通道同时开始计数 如果相关的 BPWM 通道计数器同步开始功能也使能了，向该位写 1 还会使能计数器使能位

BPWM状态寄存器(BPWM_STATUS)

寄存器	偏移	R/W	描述	复位值
BPWM_STATUS	BPWMx_BA+0x120	R/W	BPWM状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**ADCTRG5**	**ADCTRG4**	**ADCTRG3**	**ADCTRG2**	**ADCTRG1**	**ADCTRG0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**CNTMAXF0**

位	描述	
[31:22]	**Reserved**	保留
[16+n] n=0,1..5	**ADCTRGn**	**ADC开始转换状态** 每一位 n 控制着对应的 BPWM 通道 n 0 = 没有 ADC开始转换触发事件发生 1 = 有 ADC开始转换触发事件发生，软件写 1 清 0 该位
[15:1]	**Reserved**	保留
[0]	**CNTMAX0**	**时间基准计数器 0 为 0Xffff时的锁存状态** 0 = 时间基准计数器从未达到最大值 0xFFFF 1 = 时间基准计数器达到过最大值，软件写 1 清 0 该位

BPWM输入捕捉使能寄存器(BPWM_CAPINEN)

寄存器	偏移	R/W	描述	复位值
BPWM_CAPINEN	BPWMx_BA+0x200	R/W	BPWM 输入捕捉使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CAPINEN5**	**CAPINEN4**	**CAPINEN3**	**CAPINEN2**	**CAPINEN1**	**CAPINEN0**

位	描述	
[31:6]	**Reserved**	保留
[n] n=0,1..5	**CAPINENn**	**输入捕捉使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止BPWM 通道输入捕捉通道，BPWM 通道捕捉功能的输入将被视作0 1 = 使能BPWM 通道输入捕捉通道，BPWM 通道捕捉的输入由多功能引脚的状态决定

BPWM捕捉控制寄存器(BPWM_CAPCTL)

寄存器	偏移	R/W	描述	复位值
BPWM_CAPCTL	BPWMx_BA+0x204	R/W	BPWM捕捉控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FCRLDEN5	FCRLDEN4	FCRLDEN3	FCRLDEN2	FCRLDEN1	FCRLDEN0
23	22	21	20	19	18	17	16
Reserved		RCRLDEN5	RCRLDEN4	RCRLDEN3	RCRLDEN2	RCRLDEN1	RCRLDEN0
15	14	13	12	11	10	9	8
Reserved		CAPINV5	CAPINV4	CAPINV3	CAPINV2	CAPINV1	CAPINV0
7	6	5	4	3	2	1	0
Reserved		CAPEN5	CAPEN4	CAPEN3	CAPEN2	CAPEN1	CAPEN0

位	描述	
[31:30]	**Reserved**	保留
[24+n] n=0,1..5	**FCRLDENn**	**下降沿捕捉重载使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止下降沿捕捉重载计数器 1 = 使能下降沿捕捉重载计数器
[23:22]	**Reserved**	保留
[16+n] n=0,1..5	**RCRLDENn**	**上升沿捕捉重载使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止上升沿捕捉重载计数器 1 = 使能上升沿捕捉重载计数器
[15:14]	**Reserved**	保留
[8+n] n=0,1..5	**CAPINVn**	**捕捉反相使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止捕捉源反相 1 = 使能捕捉源反相。从 GPIO 反转输入信号
[7:6]	**Reserved**	保留
[n] n=0,1..5	**CAPENn**	**捕捉功能使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 = 禁止捕捉功能，RCAPDAT/FCAPDAT不会被更新 1 = 使能捕捉功能。当侦测到输入信号为上升或下降沿， BPWM计数器数值将被捕获锁存，并且保存在RCAPDAT(上升锁存)和FCAPDAT (下降锁存)。

BPWM捕捉状态寄存器(BPWM_CAPSTS)

寄存器	偏移	R/W	描述	复位值
BPWM_CAPSTS	BPWMx_BA+0x208	R	BPWM捕捉状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CFIFOV5**	**CFIFOV4**	**CFIFOV3**	**CFIFOV2**	**CFIFOV1**	**CFIFOV0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CRIFOV5**	**CRIFOV4**	**CRIFOV3**	**CRIFOV2**	**CRIFOV1**	**CRIFOV0**

位	描述	
[31:14]	**Reserved**	保留
[8+n] n=0,1..5	**CFIFOVn**	**下降沿捕捉中断标志溢出状态 (只读)** 每一位 n 控制着对应的 BPWM 通道 n 对应的 CAPFIF 为 1 时，如果发生了下降沿锁存，该位会置位。 **注：**用户清除对应的 CAPFIF 时，该位会自动清零。
[7:6]	**Reserved**	保留
[n] n=0,1..5	**CRIFOVn**	**上降沿捕捉中断标志溢出状态 (只读)** 每一位 n 控制着对应的 BPWM 通道 n 对应的 CAPRIF 为 1 时，如果发生了上升沿锁存，该位会置位。 **注：**用户清除对应的 CAPRIF 时，该位会自动清零。

BPWM上升沿捕捉数据寄存器 0~5(BPWM_RCAPDAT 0~5)

寄存器	偏移	R/W	描述	复位值
BPWM_RCAPDAT0	BPWMx_BA+0x20C	R	BPWM 上升沿捕捉数据寄存器0	0x0000_0000
BPWM_RCAPDAT1	BPWMx_BA+0x214	R	BPWM 上升沿捕捉数据寄存器1	0x0000_0000
BPWM_RCAPDAT2	BPWMx_BA+0x21C	R	BPWM 上升沿捕捉数据寄存器2	0x0000_0000
BPWM_RCAPDAT3	BPWMx_BA+0x224	R	BPWM 上升沿捕捉数据寄存器3	0x0000_0000
BPWM_RCAPDAT4	BPWMx_BA+0x22C	R	BPWM 上升沿捕捉数据寄存器4	0x0000_0000
BPWM_RCAPDAT5	BPWMx_BA+0x234	R	BPWM 上升沿捕捉数据寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**RCAPDAT**	**BPWM 上升沿捕捉数据 (只读)** 上升沿捕捉发生时，BPWM 计数器值会被保存到该寄存器

BPWM下降沿捕捉数据寄存器 0~5(BPWM_FCAPDAT 0~5)

寄存器	偏移	R/W	描述	复位值
BPWM_FCAPDAT0	BPWMx_BA+0x210	R	BPWM 下降沿捕捉数据寄存器0	0x0000_0000
BPWM_FCAPDAT1	BPWMx_BA+0x218	R	BPWM 下降沿捕捉数据寄存器1	0x0000_0000
BPWM_FCAPDAT2	BPWMx_BA+0x220	R	BPWM 下降沿捕捉数据寄存器2	0x0000_0000
BPWM_FCAPDAT3	BPWMx_BA+0x228	R	BPWM 下降沿捕捉数据寄存器3	0x0000_0000
BPWM_FCAPDAT4	BPWMx_BA+0x230	R	BPWM 下降沿捕捉数据寄存器4	0x0000_0000
BPWM_FCAPDAT5	BPWMx_BA+0x238	R	BPWM 下降沿捕捉数据寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**FCAPDAT**	**BPWM 下降沿捕捉数据 (只读)** 下降沿捕捉发生时，BPWM 计数器值会被保存到该寄存器

BPWM捕捉中断使能寄存器(BPWM_CAPIEN)

寄存器	偏移	R/W	描述	复位值
BPWM_CAPIEN	BPWMx_BA+0x250	R/W	BPWM捕捉中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CAPFIEN5**	**CAPFIEN4**	**CAPFIEN3**	**CAPFIEN2**	**CAPFIEN1**	**CAPFIEN0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CAPRIEN5**	**CAPRIEN4**	**CAPRIEN3**	**CAPRIEN2**	**CAPRIEN1**	**CAPRIEN0**

位	描述	
[31:14]	**Reserved**	保留
[13:8]	**CAPFIENn**	**BPWM 捕捉下降沿锁存中断使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 =禁止捕捉下降沿锁存中断 1 = 使能捕捉下降沿锁存中断
[7:6]	**Reserved**	保留
[5:0]	**CAPRIENn**	**BPWM 捕捉上升沿锁存中断使能位** 每一位 n 控制着对应的 BPWM 通道 n 0 =禁止捕捉上升沿锁存中断 1 =使能捕捉上升沿锁存中断

BPWM捕捉中断标志寄存器(BPWM_CAPIF)

寄存器	偏移	R/W	描述	复位值
BPWM_CAPIF	BPWMx_BA+0x254	R/W	BPWM 捕捉中断标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CAPFIF5**	**CAPFIF4**	**CAPFIF3**	**CAPFIF2**	**CAPFIF1**	**CAPFIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CAPRIF5**	**CAPRIF4**	**CAPRIF3**	**CAPRIF2**	**CAPRIF1**	**CAPRIF0**

位	描述	
[31:14]	**Reserved**	保留
[8+n] n=0,1..5	**CAPFIFn**	**BPWM 捕捉下降沿锁存中断标志** 该位写 1 清 0，每一位 n 控制着对应的 BPWM 通道 n 0 = 无捕捉下降沿锁存情况发生 1 = 捕捉下降沿锁存发生，该位置 1
[7:6]	**Reserved**	保留
[n] n=0,1..5	**CAPRIFn**	**BPWM 捕捉上升沿锁存中断标志** 该位写 1 清 0，每一位 n 控制着对应的 BPWM 通道 n 0 = 无捕捉上升沿锁存情况发生 1 = 捕捉上升沿锁存发生，该位置 1

BPWM周期缓存(BPWM_PBUF)

寄存器	偏移	R/W	描述	复位值
BPWM_PBUF	BPWMx_BA+0x304	R	BPWM 周期缓存	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**PBUF**	**BPWM 周期缓存 (只读)** 同 PERIOD 有效寄存器一样使用

BPWM比较值寄存器缓存 0~5(BPWM_CMPBUF0~5)

寄存器	偏移	R/W	描述	复位值
BPWM_CMPBUF0	BPWMx_BA+0x31C	R	BPWM CMPDAT 0缓存	0x0000_0000
BPWM_CMPBUF1	BPWMx_BA+0x320	R	BPWM CMPDAT 1缓存	0x0000_0000
BPWM_CMPBUF2	BPWMx_BA+0x324	R	BPWM CMPDAT 2缓存	0x0000_0000
BPWM_CMPBUF3	BPWMx_BA+0x328	R	BPWM CMPDAT 3缓存	0x0000_0000
BPWM_CMPBUF4	BPWMx_BA+0x32C	R	BPWM CMPDAT 4缓存	0x0000_0000
BPWM_CMPBUF5	BPWMx_BA+0x330	R	BPWM CMPDAT 5缓存	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**CMPBUF**	**BPWM 比较器缓存 (只读)** 同 CMP 有效寄存器一样使用

6.12 PWM发生器和捕获定时器 (PWM)

6.12.1 概述

该芯片提供了两路PWM发生器，PWM0和PWM1。每路PWM支持6通道的PWM输出或输入捕获。一个12位的预分频器可将时钟源分频，为配有16位比较器的16位计数器提供灵活的时钟。PWM计数器支持向上，向下，上下计数方式。PWM将比较器与计数器比较来产生事件，这些事件可以用来产生PWM脉冲、中断、ADC开始转换触发信号。

PWM发生器支持两种标准PWM输出模式：独立模式和互补模式，这两种模式的架构互为不同。互补模式中，两个比较器可插入各种12位死区时间的PWM脉宽。PWM输出控制单元，它支持极性输出，独立引脚屏蔽和刹车功能。

PWM发生器还支持输入捕获功能，当输入通道有向上跳变、向下跳变、或者向上向下都有的跳变时，PWM计数器的值将会被锁存到相应的寄存器中。捕获功能还支持通过PDMA把捕获到的数据搬移到指定内存区域。

6.12.2 特性

6.12.2.1 PWM 功能特性

- 支持最大144 MHz时钟频率
- 支持两个PWM模块，每个模块提供6个输出通道
- 支持独立模式的PWM输出/输入捕获
- 支持3组互补通道的互补模式
  - 12位分辨率的死区插入
  - 每个周期两个比较值
- 支持12位从1到4096的预分频
- 支持16位分辨率的PWM计数器
  - 向上，向下和上下计数操作类型
- 每个PWM引脚支持屏蔽功能和三态使能
- 支持刹车功能
  - 刹车源来自引脚和系统安全事件（时钟故障、欠压检测、SRAM奇偶校验错误和CPU锁死）
  - 刹车源引脚噪声滤波器
  - 通过边沿检测刹车源来控制刹车状态直到刹车中断清除
  - 刹车条件解除后通过电平检测刹车源来控制自动恢复功能
- 支持下列事件中断:
  - PWM计数器值为 0、周期值或比较值
  - 满足刹车条件
- 支持下列事件触发ADC:
  - PWM计数器值为0、周期值或比较值

6.12.2.2 捕获功能特性

- 支持12路16位分辨率的输入捕获通道
- 支持上升/下降沿捕获条件
- 支持输入上升/下降沿捕获中断
- 支持带计数器重载选择的上升/下降沿捕获
- 支持PWM 所有通道的PDMA数据搬移功能

6.12.3 框图

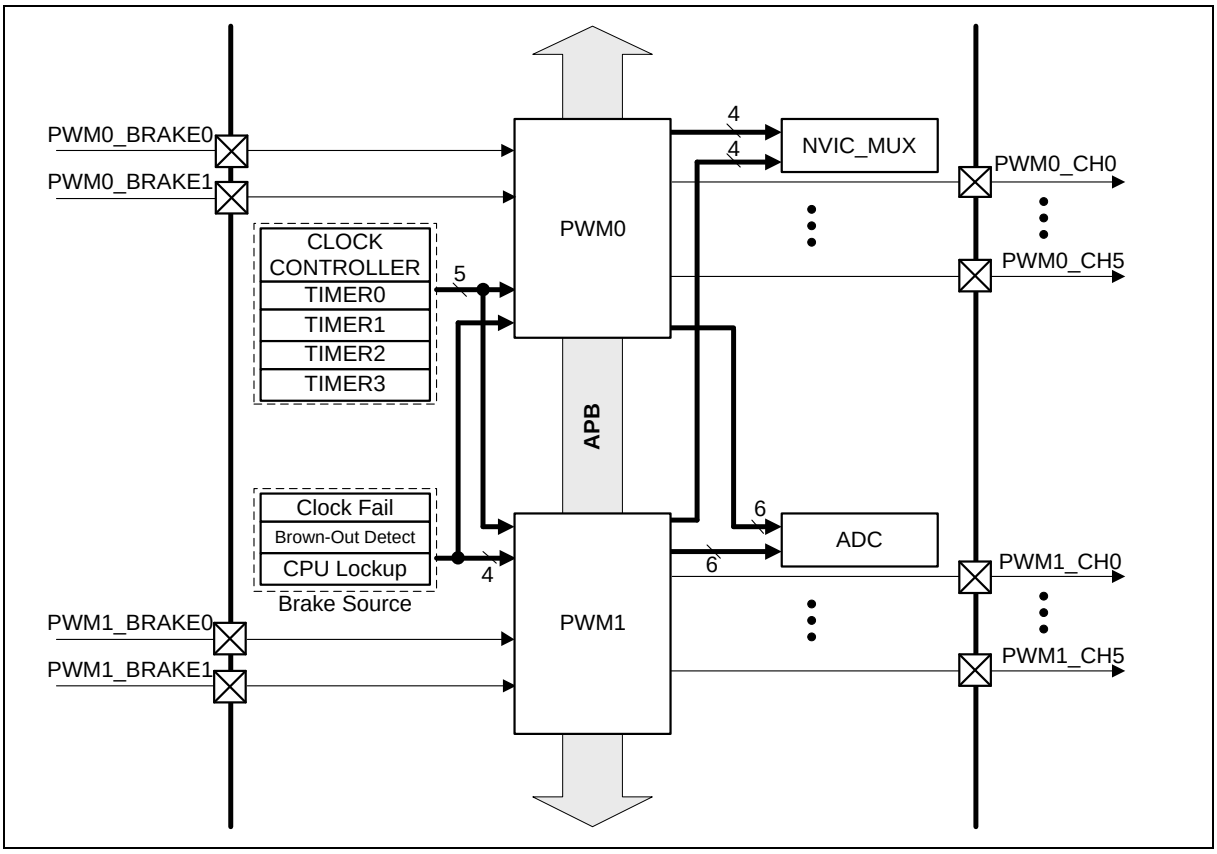


图 6.12-1 PWM 发生器简要方块图

PWM系统时钟可以设为等于或两倍于PCLK的频率，如图 6.12-2所示，寄存器设置细节请参考表 6.12-1。每个PWM发生器为每对通道都分配有对应的输入时钟源，每对通道的时钟源可以选择来自PWM时钟或者4组定时器触发PWM输出如图 6.12-3所示，通过ECLKSRC0(PWM_CLKSRC[2:0])设置PWM_CLK0，ECLKSRC2(PWM_CLKSRC[10:8])设置PWM_CLK2，ECLKSRC4(PWM_CLKSRC[18:16])设置PWM_CLK4。

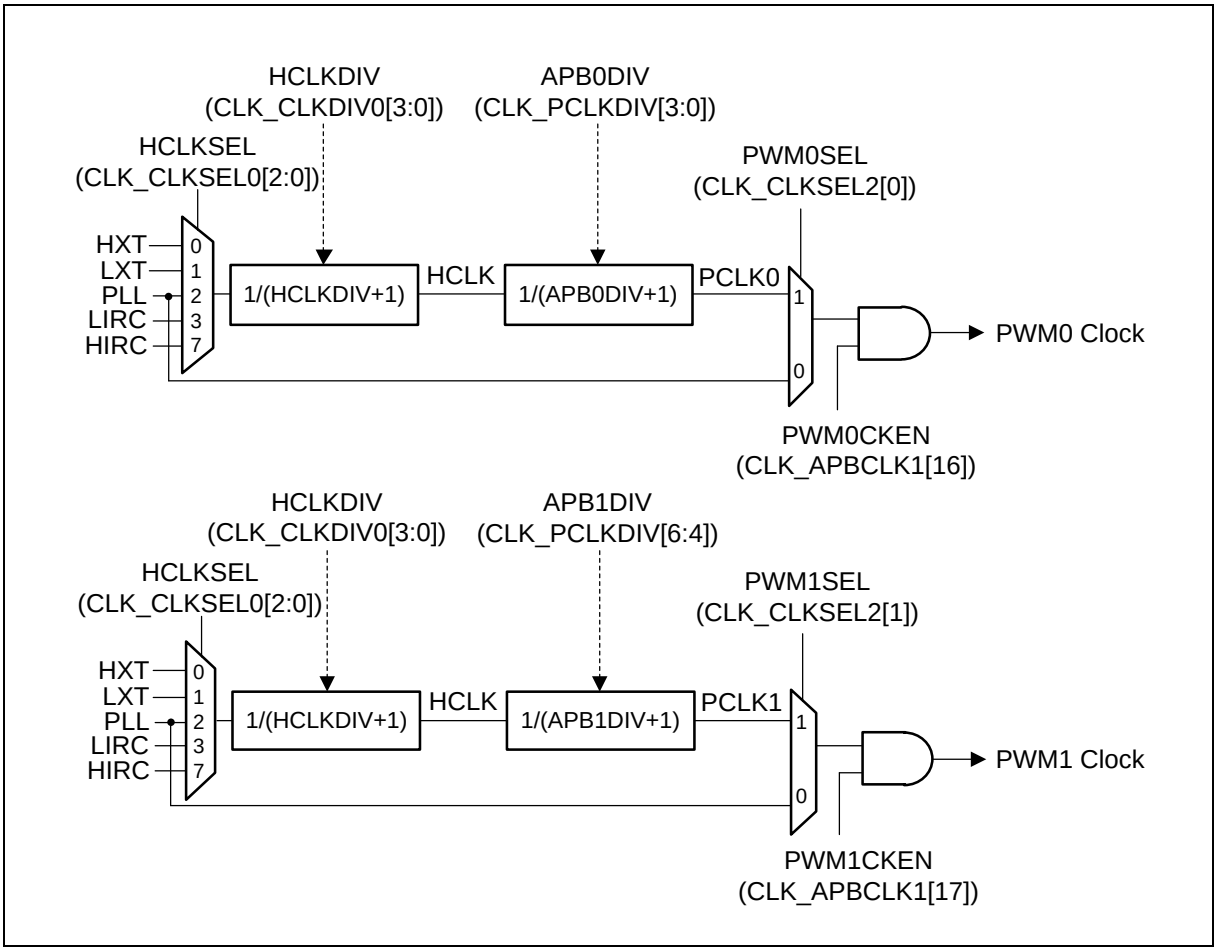


图 6.12-2 PWM 系统时钟源控制

频率比 PCLK:PWM 时钟	HCLK	PCLK	PWM Clock	HCLKSEL CLK_CLKSEL0[2:0]	HCLKDIV CLK_CLKDIV0[3:0]	APBnDIV (CLK_PCLKDIV [2+4n:4n]), N 为 0 或 1	PWMnSEL (CLK_CLKSEL2[N]), N 为 0 或 1
1:1	HCLK	PCLK	PCLK	无关	无关	无关	1
1:2	PLL	PLL/ 2	PLL	2	0	1	0
1:2	PLL/ 2	PLL/ 2	PLL	2	1	0	0

表 6.12-1 PWM 时钟源控制寄存器设置表

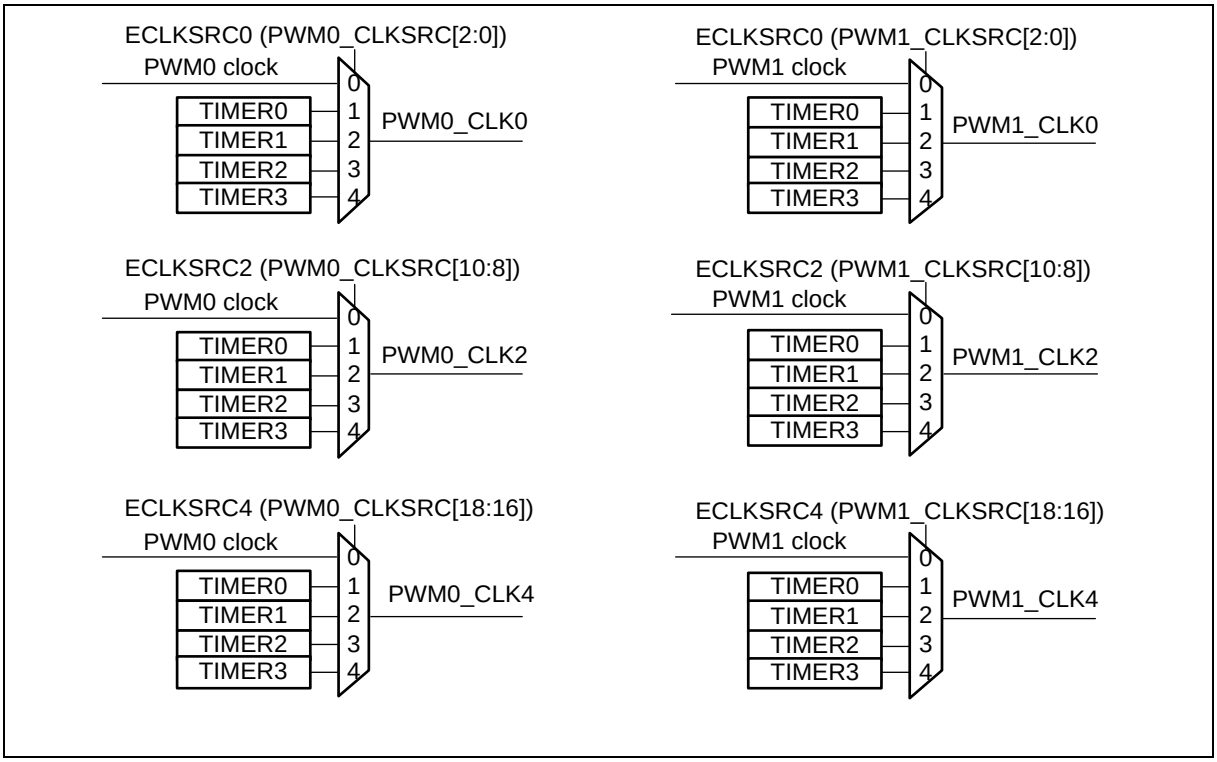


图 6.12-3 PWM 时钟源控制

图 6.12-4和 图 6.12-5分别为PWM独立模式和互补模式的架构示意图。不管是独立模式还是互补模式，每对通道组(PWM_CH0 和PWM_CH1, PWM_CH2 和 PWM_CH3, PWM_CH4 和 PWM_CH5)的计数器都来自相同的时钟源和预分频。当计数器的值等于0、PERIOD(PWM_PERIODn[15:0])或比较器值时，一些事件将会产生。这些事件可以通过相应的发生器来产生PWM脉冲、中断信号、ADC的转换触发信号。PWM输出控制单元是用来改变PWM脉冲输出状态的。输出控制的刹车功能也能产生中断事件。在互补模式中，偶数通道是用奇数通道比较器来产生事件的。

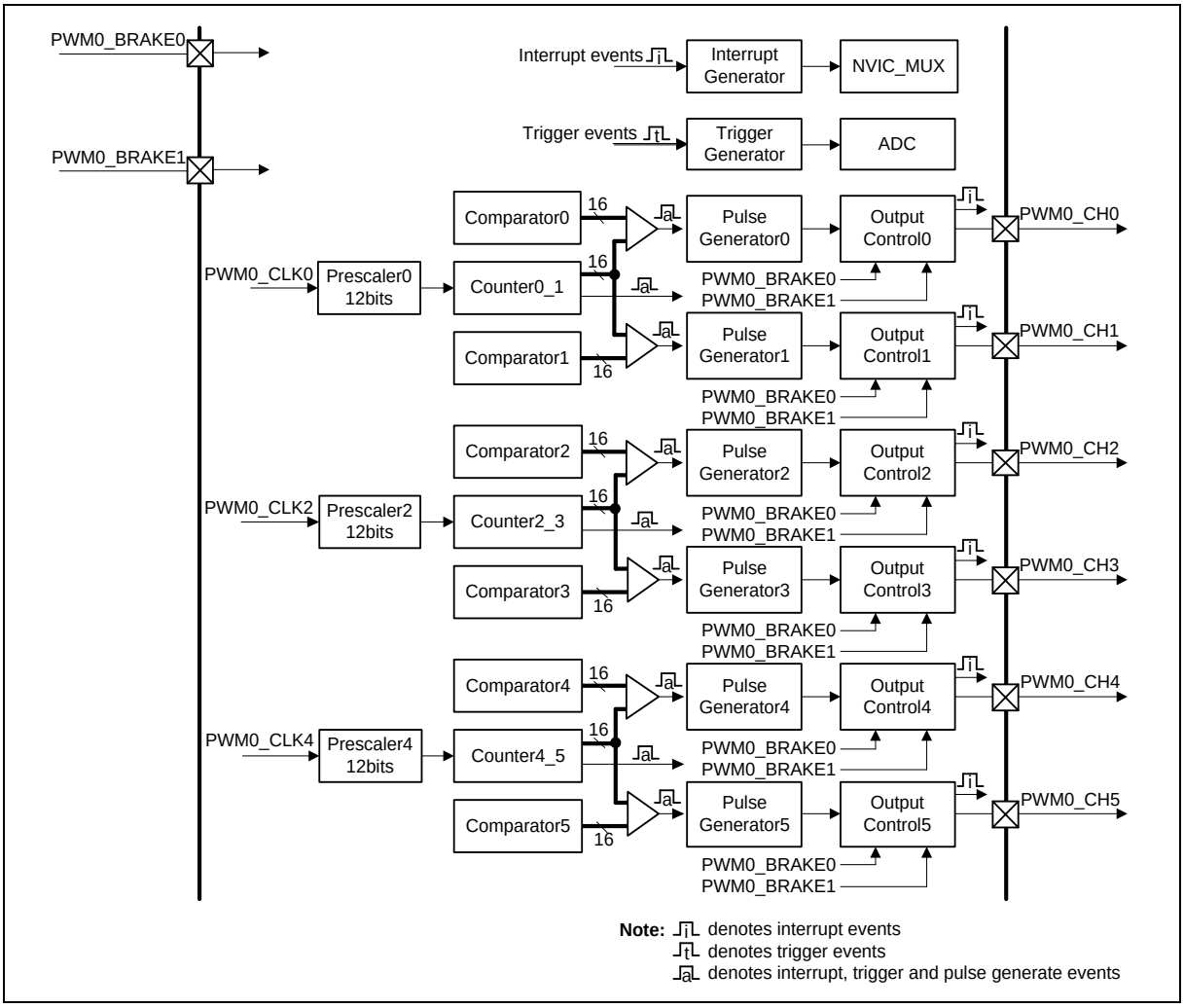


图 6.12-4 PWM 独立模式架构图

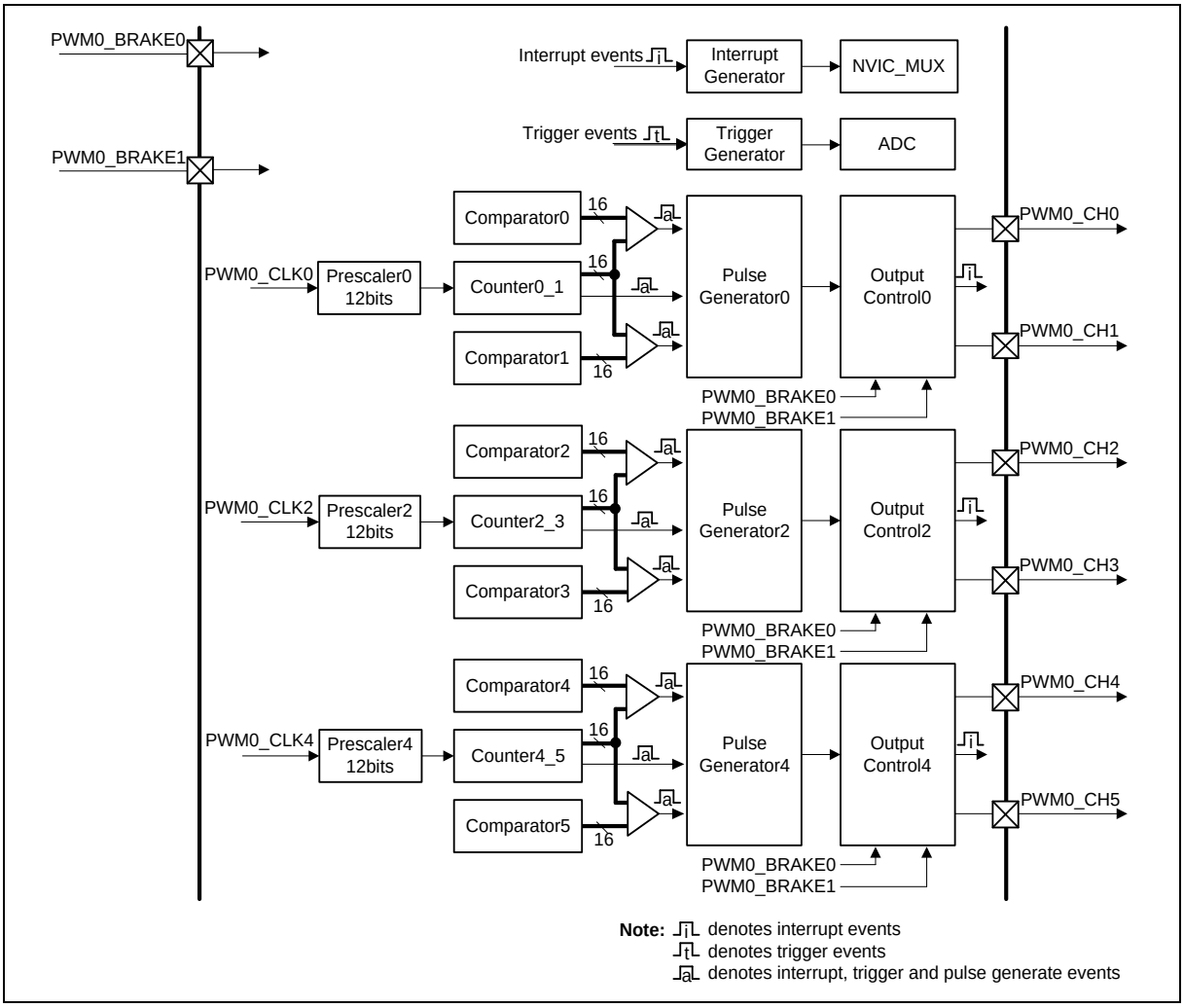


图 6.12-5 PWM 互补模式架构图

6.12.4 基本配置

6.12.4.1 PWM0 基本配置

- 时钟源配置
  - 通过PWM0SEL (CLK_CLKSEL2[0])选择PWM外设时钟源
  - 通过PWM0CKEN CLK_APBCLK1[16]) 使能 PWM0 外设时钟.
- 复位配置
  - 通过PWM0RST SYS_IPRST2[16]复位 PWM0

6.12.4.2 PWM1 基本配置

- 时钟源配置
  - 通过 PWM1SEL (CLK_CLKSEL2[1])选择PWM1外设时钟源
  - 通过 PWM1CKEN (CLK_APBCLK1[17])使能PWM1外设时钟

- 复位配置
  - 通过PWM1RST (SYS_IPRST2[17])复位PWM1

6.12.5 功能描述

6.12.5.1 PWM 预分频器

PWM预分频器用于时钟源分频，预分频器计数CLKPSC+1次，PWM计数器只加一次。用户可以通过CLKPSC (PWM_CLKPSCn[11:0], n = 0, 2, 4)来设置预分频双缓存。图 6.12-6是PWM通道0 预分频波形的一个例子， 预分频计数器会在预分频计数器下一个向下计数开始时重载CLKPSC.

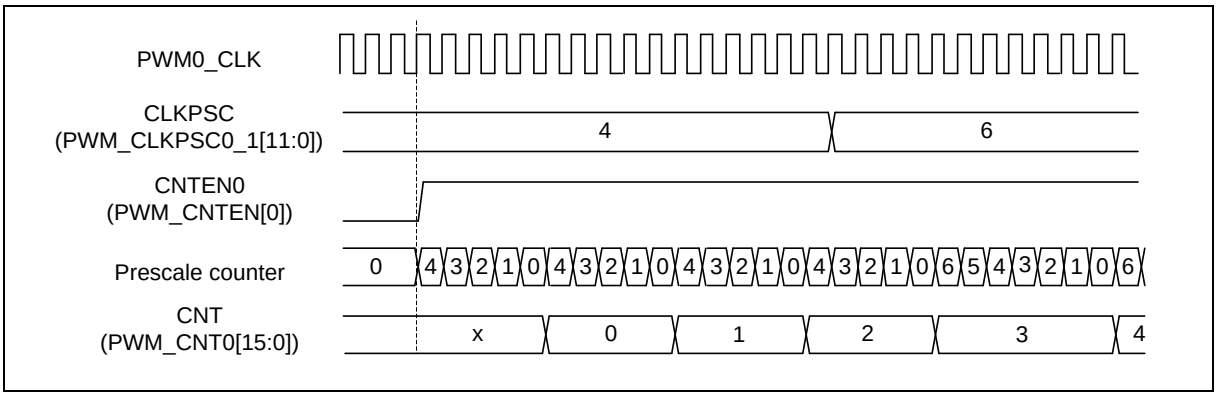


图 6.12-6 向上计数模式下的 PWM0_CH0 的预分频器 CLKPSC 波形

6.12.5.2 PWM计数器

PWM 支持 3 种计数方式操作：向上计数，向下计数，上下计数方式。

对于PWM通道0，CNT(PWM_CNT0[15:0])可以通过置位CNTCLR0 (PWM_CNTCLR[0])来清为0X00。当预分频计数器数到0时，CNT会被清除，CNTCLR0也被硬件自动清0。

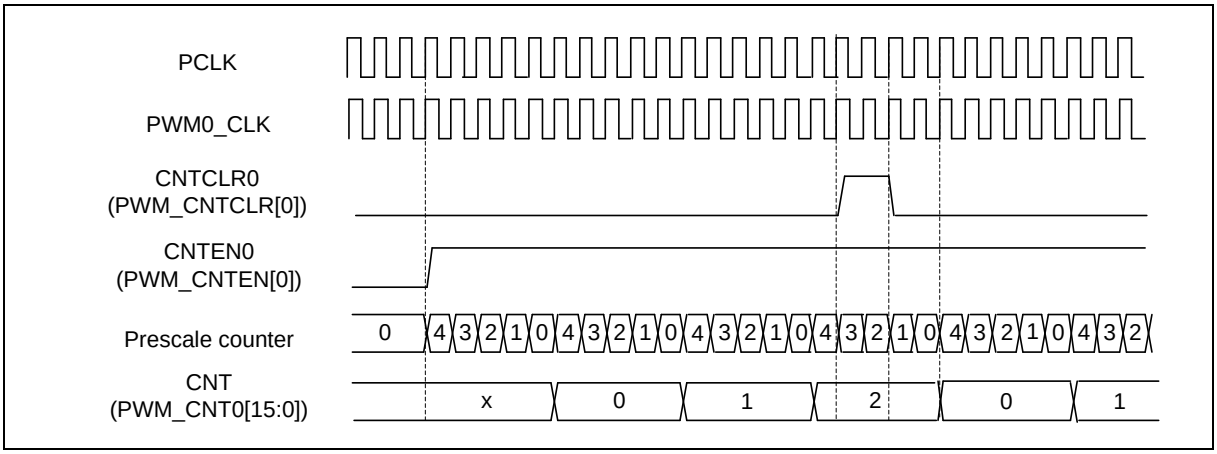


图 6.12-7 当设置清计数器 PWMx 计数器的波形

6.12.5.3 上计数模式

在向上计数操作中，CNTTYPEn (PWM_CTL1[2n+1:2n], n = 0,1..5) 是 0x0。16位PWM计数器从0开始向上计数到PERIOD (PWM_PERIODn[15:0]，其中n表示通道数)来完成一个PWM周期。当前计数值可以从CNT (PWM_CNTn[15:0])读出。当计数器计数到0且预分频计数到0时，将产生零点事件；当计数

器计数到PERIOD且预分频计数到0时，将产生周期点事件。图 6.12-8所示为向上计数器模式的示例。

PWM周期 = (PERIOD+1) *(CLKPSC+1)* PWMx_CLK.

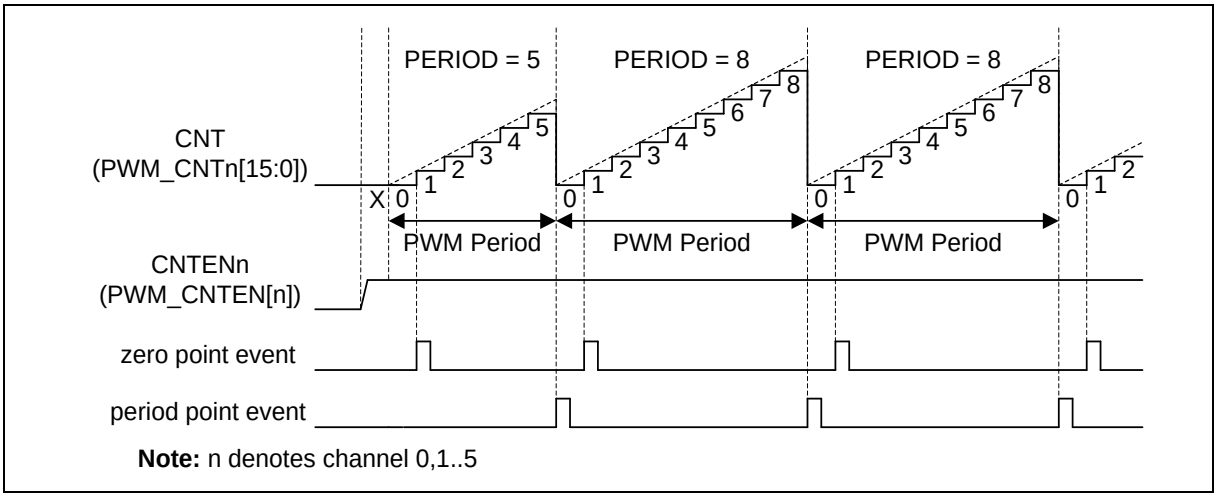


图 6.12-8 PWM 上计数器模式

6.12.5.4 下计数模式

在向下计数方式中，CNTTYPEn (PWM_CTL1[2n+1:2n], n = 0,1..5) 是 0x1，16位PWM计数器从PERIOD开始向下计数到0来完成一个PWM周期。当前计数值可以从CNT(PWM_CNTn[15:0])读出。当计数器计数到0且预分频计数到0时，将产生零点事件；当计数到PERIOD且预分频计数到0时，将产生周期事件，图 6.12-9所示为向下计数器模式的示例：

PWM 周期 = (PERIOD+1) * (CLKPSC+1) * PWMx_CLK.

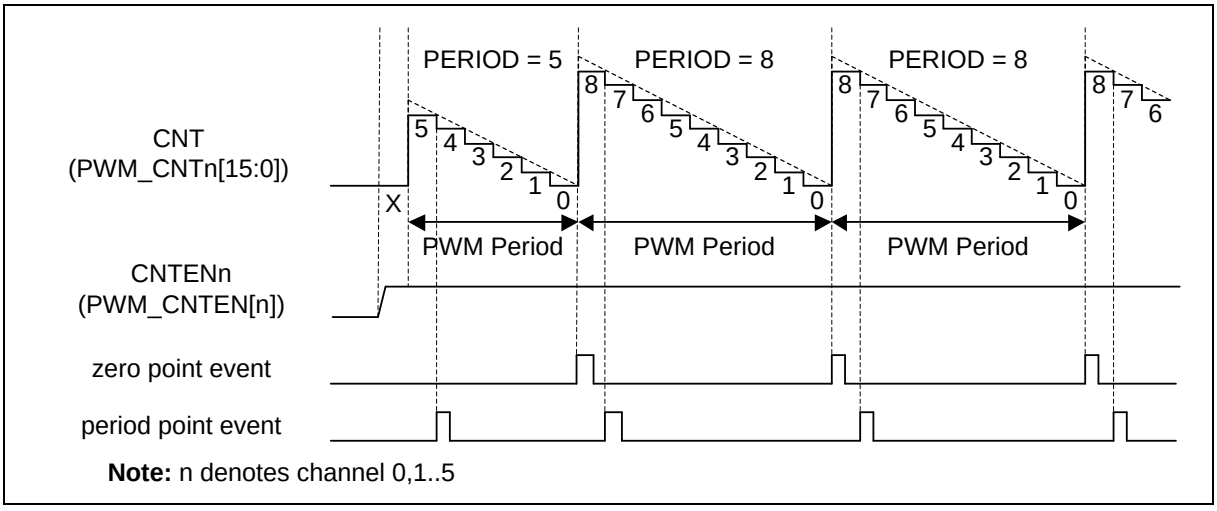


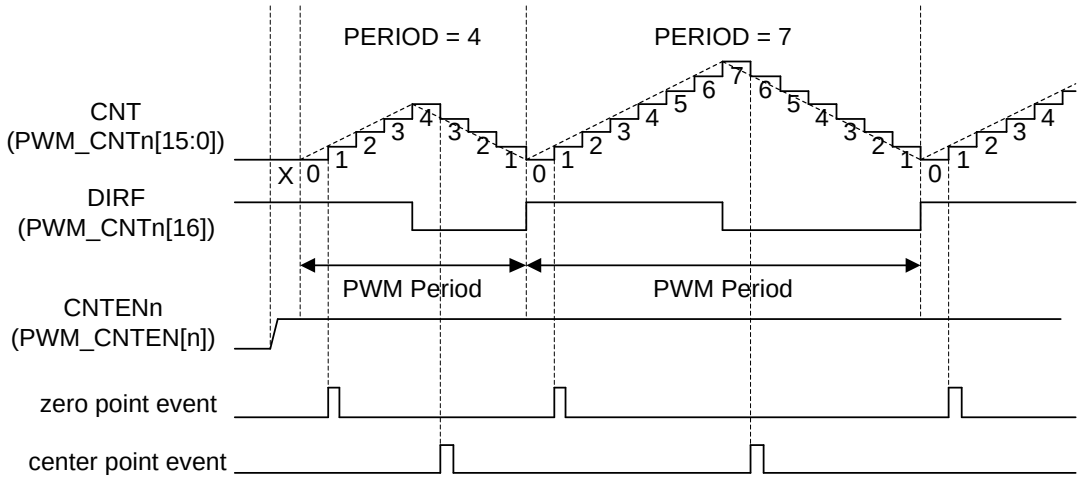
图 6.12-9 PWM 下计数器模式

6.12.5.5 上下计数模式

在上下计数操作中，CNTTYPEn (PWM_CTL1[2n+1:2n], n = 0,1..5) 是 0x2，16位PWM计数器向上计数从0到PERIOD然后又向下计数到0完成一个PWM周期。当前计数值可以从CNT中读出。当计数器计数到0且预分频计数到0时，将产生零点事件；当计数到PERIOD时，将产生中心点事件。图 6.12-10所示为上下计数器模式的示例：

PWM周期 =(2*PERIOD) * (CLKPSC+1) * PWMx_CLK.

DIRF (PWM_CNTn[16]) 位是计数器方向指示标志，高表示向上计数，低表示向下计数.

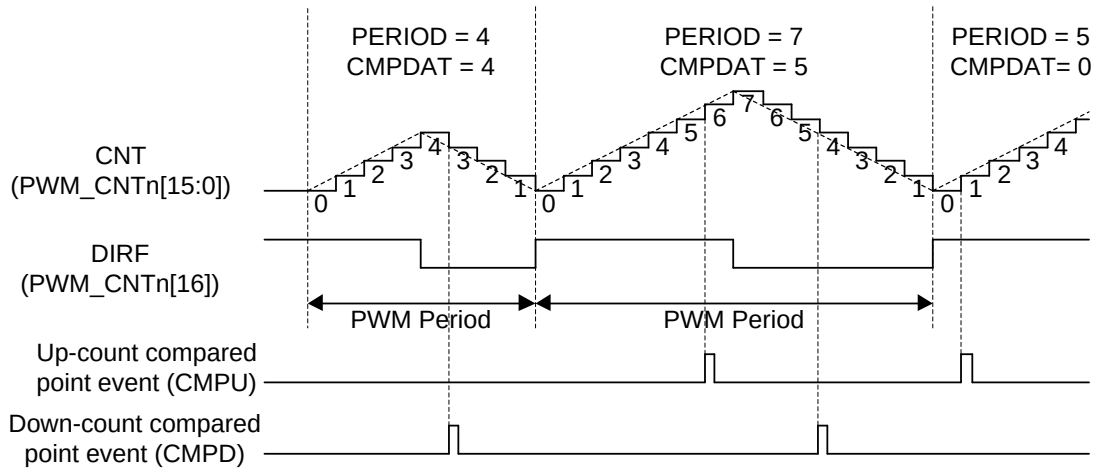


Note1: When in up-down count type, period interrupt flag occurs at center point event.
Note2: n denotes channel 0,1..5

图 6.12-10 PWM 上下计数器模式

6.12.5.6 PWM 比较器

CMPDATn是PWM通道n的基本比较器寄存器。独立模式下，每个通道对应一个比较器，CMPDATn值一直与相应通道的计数器值作比较。互补模式下，每对通道有两个比较器CMPDATn 和 CMPDATm (n = 0,2,4, m = 1,3,5)，两个比较器寄存器的值会一直与偶数通道计数值进行比较，而奇数通道的计数器一直会被忽略。例如，通道0和通道1为互补通道，在互补模式，通道1的比较器持续与通道0的计数器相比较，而不与通道1的计数器比较。当计数器等于CMPDAT0寄存器的值时，PWM产生一个比较点事件，并用该事件产生PWM脉冲、中断或触发ADC。在上下计数方式中，一个PWM 周期将产生两个事件，如图 6.12-11所示， CMPU是上数比较点事件，CMPD是下数比较点事件。



Note1: No CMPU event occurred when CMPDAT equals to PERIOD.
Note2: n denotes channel 0,1..5

图 6.12-11 PWM 在上下计数方式中的比较点事件

6.12.5.7 PWM双缓存

双缓存是用两个缓存来把软件写和硬件操作时序分开。载入值到缓存有三种装载模式：周期装载模式，立即装载模式，中心装载模式。软件设置好寄存器后，硬件会按照装载模式时序将寄存器值装载到缓存寄存器中。硬件的操作是基于缓存寄存器的值。这样做可以避免软硬件不同步时的操作问题。

PWM 提供 PBUF (PWM_PBUFn[15:0]) 作为使 PERIOD 生效的缓存寄存器，CMPBUF (PWM_CMPBUFn[15:0]) 作为使 CMPDAT 生效的缓存寄存器。双缓存用于装载数据时的时序，如下面描述。例如，如图 6.12-12，在周期装载模式，通过软件写PERIOD 、 CMPDAT， 在下个周期时 PWM 将载入新值到它们的缓存 PBUF(PWM_PBUFn[15:0])，CMPBUF (PWM_CMPBUFn[15:0])，而不影响当前计数操作。

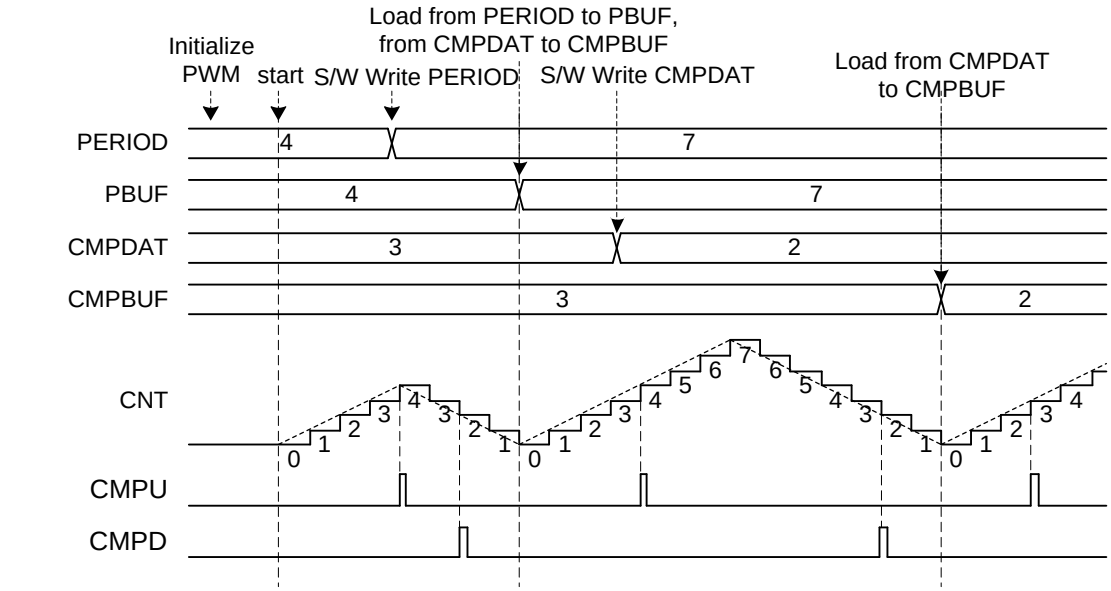


图 6.12-12 PWM 双缓存说明

6.12.5.8 周期装载模式

当立即装载模式和中心装载模式禁止时，IMMLDENn (PWM_CTL0[21:16]) 和 CTRLDn (PWM_CTL0[5:0]) 为0，PWM工作在周期装载模式。在周期装载模式，当每一个周期完成，PERIOD(PWM_PERIODn[15:0]) 和 CMP(PWM_CMPDATn[15:0]) 会全部加载到他们的有效的PBUF和 CMPBUF寄存器中，例如：在向上计数模式PWM计数器从零计数到PERIOD后或在向下计数模式PWM计数器从PERIOD计数到零，或在上下计数模式PWM计数器向上计数从零到PERIOD然后向下计数到零时。

图 6.12-13表示向上计数操作时的周期装载时序，PERIOD DATA0表示PERIOD初始数据，PERIOD DATA1表示通过软件要更新的第一个PERIOD数据，以此类推，CMPDAT同样遵照这规则。以下是图6.12-13的步骤描述。用户可以通过观察PWM周期和CMPU事件知道PERIOD和CMPDAT的更新条件。

1. point1处，软件写CMPDAT DATA1到CMPDAT
2. point 2处，PWM周期结束，硬件装载CMPDAT DATA1 到 CMPBUF
3. point 3处，软件写PERIOD DATA1到 PERIOD
4. point 4处，PWM周期结束，硬件装载CMPDAT DATA1 到 PBUF
5. point 5处，软件写PERIOD DATA2 到 PERIOD
6. point 6处，PWM周期结束，硬件装载PERIOD DATA2到 PBUF

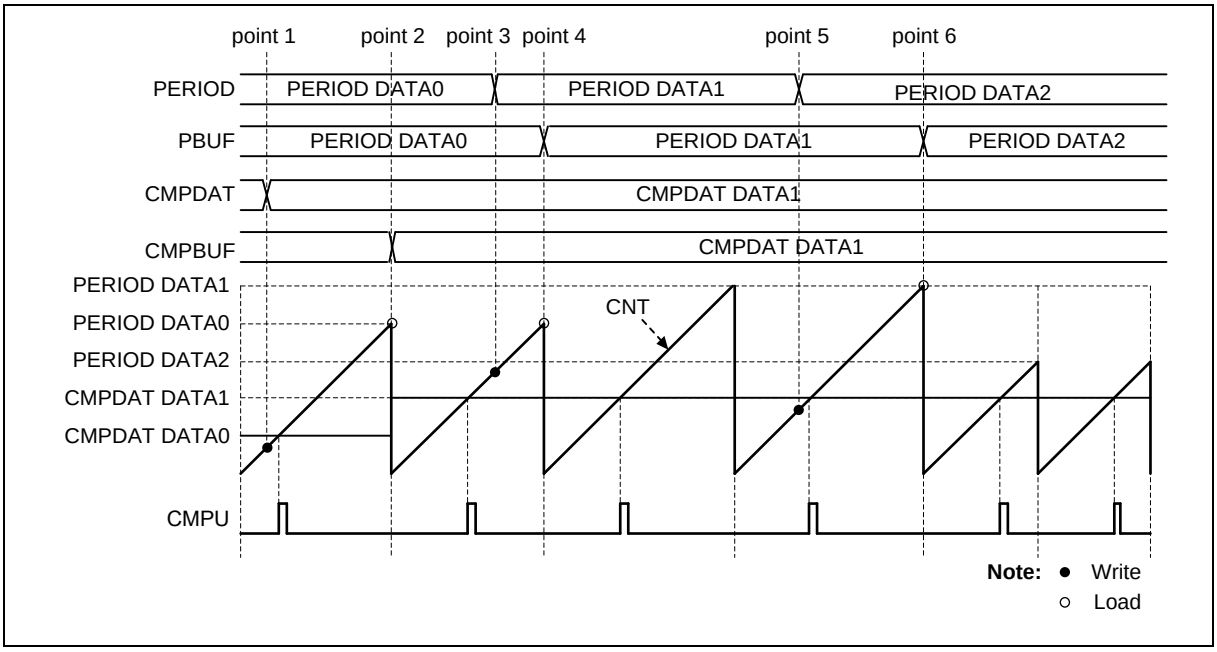


图 6.12-13 向上计数方式的周期装载模式

6.12.5.9 立即装载模式

如果IMMLDENn (PWM_CTL0[21:16])位被置1时，PWM工作在立即装载模式。在立即装载模式下，当用户软 件 改变PERIOD(PWM_PERIODn[15:0]) 或 CMP(PWM_CMPDATn[15:0])后，这些值会马上更新到PBUF (PWM_PBUFn[15:0]) 和 CMPBUF (PWM_CMPBUFn[15:0])。如果PERIOD更新值小于当前计数值，计数器会计数到0xFFFF，当计数器计数到0xFFFF且预分频器计数到0时，标志位CNTMAXF(PWMx_STATUS[5:0])会置位，计数器将会循环计数。立即装载模式有最高优先级，如果IMMLDENn被设置，通道n的其他装载模式将失效。图 6.12-14所示例子的步骤顺序如下描述：

1. 软件写CMPDAT DATA1，硬件在point1处将立即装载CMPDAT DATA1到CMPBUF

2. point 2处，软件写入PERIOD DATA1值大于当前值，计数器将继续计数到PERIOD DATA1来完成装载。
3. point 3处，软件写入PERIOD DATA2值小于当前值，计数器将继续计数到最大值0XFFFF并循环从0开始到PERIOD DATA2来完成这个周期装载。

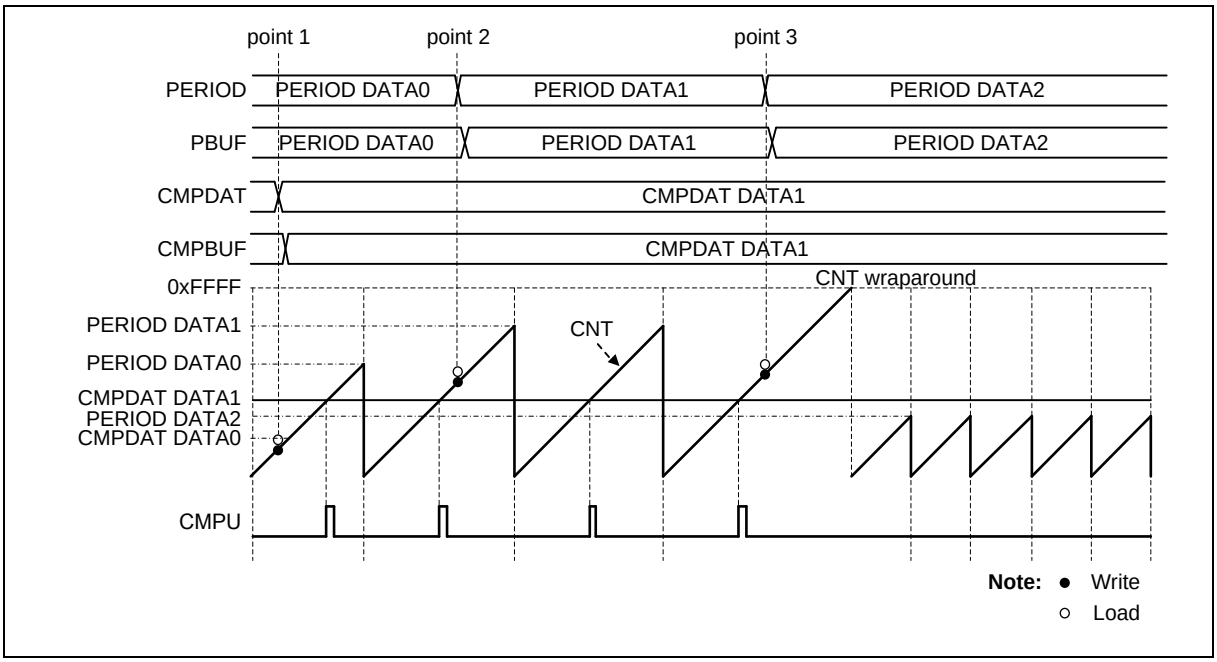


图 6.12-14 向上计数方式的立即装载模式

6.12.5.10 中心装载模式

当CTRLDn (PWM_CTL0[5:0])位被置1，且PWM计数器设置为上下计数模式时，CNTTYPEn (PWM_CTL1[2n+1:2n], n = 0,1..5) 是 0x2，PWM工作在中心装载模式。中心装载模式下，每个周期的中点CMP(PWM_CMPDATn[15:0])值将装载到有效的CMPBUF寄存器，也就是说当当前周期完成，计数器计数到PERIOD点，PERIOD(PWM_PERIODn[15:0])会装载到有效的PBUF。图 6.12-15所示例子的步骤顺序如下描述：

1. point 1处，软件写CMPDAT DATA1。
2. point 2处，PWM周期中心点硬件装载CMPDAT DATA1到CMPBUF
3. point 3处，软件写PERIOD DATA1.
4. point 4处，PWM周期终点硬件装载PERIOD DATA1到PBUF
5. point 5处，软件写CMPDAT DATA2.
6. point 6处，在PWM周期中心点硬件装载CMPDAT DATA2到CMPBUF.
7. point 7处，软件写PERIOD DATA2.
8. point 8处，在PWM周期终点硬件装载PERIOD DATA2到PBUF.

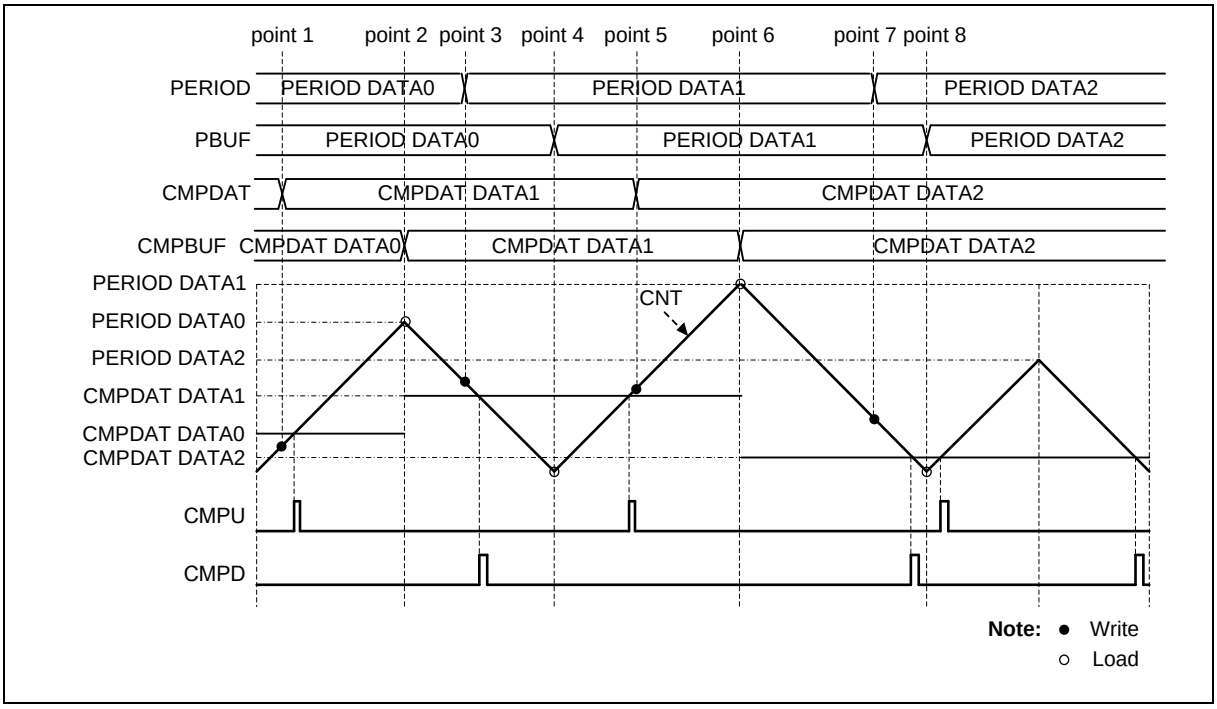


图 6.12-15 上下计数方式的中心装载模式

6.12.5.11 PWM计数器工作模式

PWM计数器支持自动装载模式.

在自动装载模式，应该先写CMPDAT 和 PERIOD，然后置 CNTENn为1来使能PWM预分频器 并 开 始 计 数 。 根 据 不 同 的 装 载 模 式 ， CLKPSC(PWM_CLKPSCn_m[11:0]), PERIOD(PWM_PERIODn[15:0]) 和 CMP(PWM_CMPDATn[15:0])的值将被自动装载到它们的缓存。如果周期PERIOD(PWM_PERIODn[15:0])被设为0， PWM 计数器就会被设为0。

6.12.5.12 PWM 脉冲发生器

PWM脉冲发生器是用计数器和比较器事件来产生PWM脉冲。这些事件包括：向上和向下计数方式的零点或周期点，上下计数方式的中心点，以及三种方式中的计数器值等于比较值。作为上下计数方式，计数器有两个等于比较器的点，一个在向上计数，一个在向下计数。此外，互补模式有两个比较器值与计数器比较，因此在上下计数方式中有四个相等值（向上或向下计数方式下各有两个）。

通过设定PWM_WGCTL0 和 PWM_WGCTL1，每个事件点可以设定PWM波形：不变（X），为低（L），为高（H）或切换（T）。用这些点可以很容易地产生一个不规则的PWM脉冲或可变波形，如图6.12-16中，PWM是互补模式，两个比较器n和m来产生PWM脉冲，n表示偶数通道0，2，4；m表示奇数通道1,3,5，n和m通道是互补组。互补模式以两个通道（(CH0 和 CH1, CH2 和 CH3, CH4 和 CH5）为一组，输出一对互补波形。CMPU 表示当向上计数时 CNT(PWM_CNTn[15:0]) 等于 CMP(PWM_CMPDATn[15:0])，CMPD表示当向下计数时CNT等于CMP。

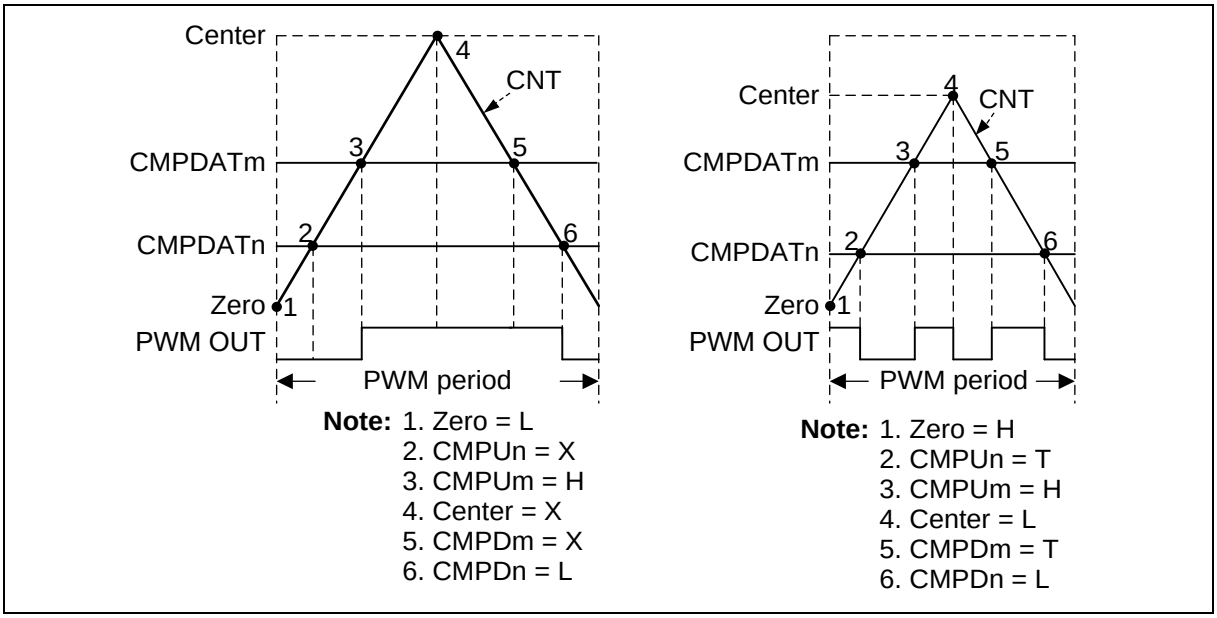


图 6.12-16 PWM 脉冲产生

有时事件会被同时产生，因此，不同计数方式的事件优先级列表如下：向上计数方式(表 6.12-2)，向下计数方式(表 6.12-3)，上下计数方式(表 6.12-4)。通过事件优先级，使用者可以很容易地设计产生出占空比从0% 到 100%的脉冲，如图 6.12-17所示。

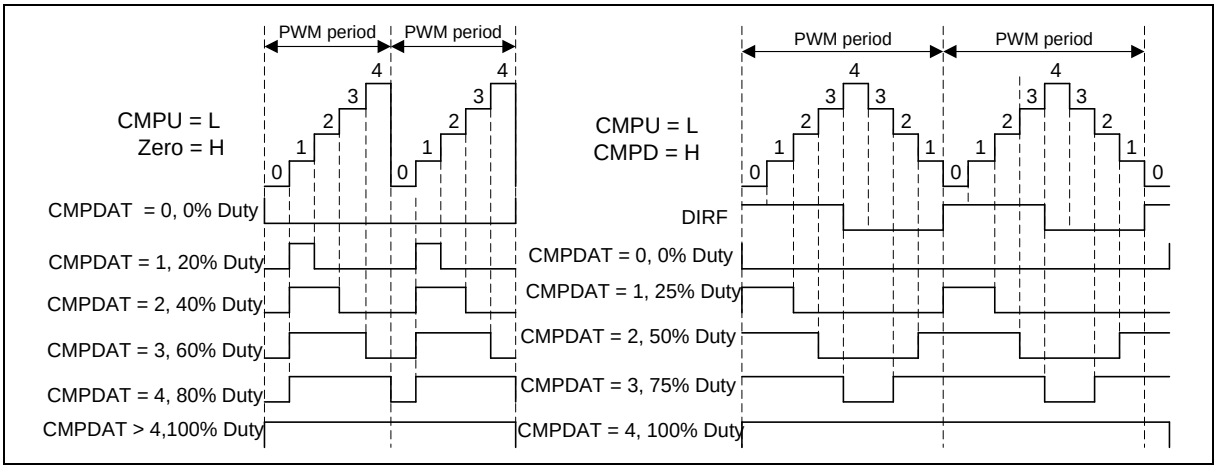


图 6.12-17 PWM 0% 到 100%占空比脉冲产生

优先级	向上事件
1 (最高)	周期事件 (CNT = PERIOD)
2	奇数通道的向上比较事件 (CNT = CMPUm)
3	偶数通道的向上比较事件 (CNT = CMPUn)
4 (最低)	零点事件 (CNT = 0)

表 6.12-2 向上计数器 PWM 脉冲发生事件优先级

优先级	向下事件
1 (最高)	零点事件 (CNT = 0)
2	奇数通道的向下比较事件 (CNT = CMPDm)
3	偶数通道的向下比较事件 (CNT = CMPDn)
4 (最低)	周期点事件(CNT = PERIOD)

表 6.12-3 向下计数器 PWM 脉冲发生优先级

优先级	向上事件	向下事件
1 (最高)	奇数通道的向上比较事件 (CNT = CMPUm	奇数通道的向下比较事件((CNT = CMPDm)
2	偶数通道的向上比较事件(CNT = CMPUn)	偶数通道的向下比较事件(CNT = CMPDn)
3 (最低)	零点事件 (CNT = 0)	周期点 (中点) 事件 (CNT = PERIOD)

表 6.12-4 上下计数器 PWM 脉冲产生事件优先级

6.12.5.13 PWM 输出模式

PWM支持两种输出模式：独立模式，可以应用于直流电机系统；带死区插入的互补模式，可以用于交流感应电机和永磁同步电机。

6.12.5.14 独立模式

PWM输出默认在独立模式，当通道n相应位PWMMODEn (PWM_CTL1[26:24])置0，独立模式被使能。独立模式中6个PWM通道：PWM_CH0, PWM_CH1, PWM_CH2, PWM_CH3, PWM_CH4 和 PWM_CH5 都在各自的周期占空比设置上运行，如图 6.12-18所示.

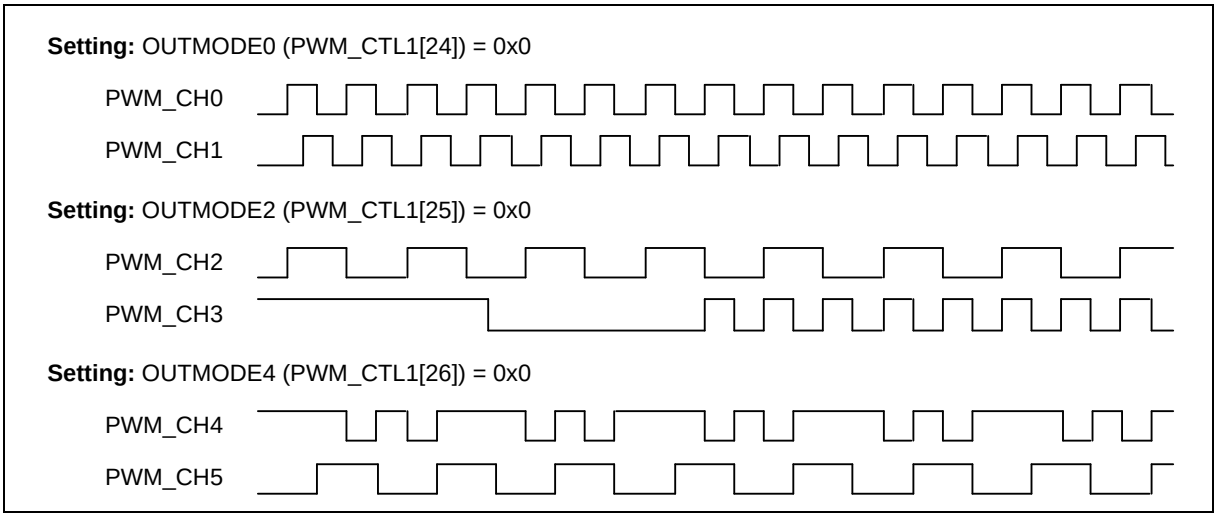


图 6.12-18 PWM 独立模式波形

6.12.5.15 互补模式

当互补通道PWMMODEn (PWM_CTL1[26:24])相应位被置1时，互补模式被使能。互补模式每个PWM模块中共有3个PWM计数器，3对PWM输出引脚。互补模式中，内部奇数通道的PWM信号必须与相应偶数通道的PWM信号互补。PWM_CH1与PWM_CH0互补， PWM_CH3与 PWM_CH2互补，

PWM_CH5与PWM_CH4互补，如图 6.12-19所示.

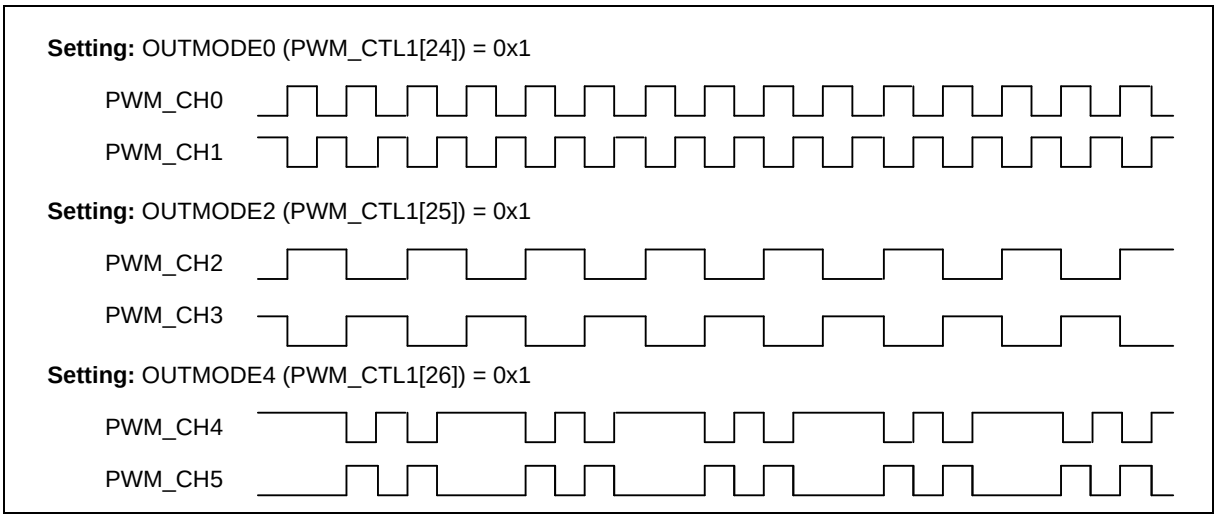


图 6.12-19 PWM 互补模式波形

6.12.5.16 PWM 输出控制

PWM脉冲产生后，需要4到6步设置来控制PWM通道输出。独立模式中，屏蔽、刹车、引脚极性和输出使能4步的设置如图 6.12-20所示. 互补模式中，需要在以上4步前再增加2步：互补通道和死区插入，如图 6.12-21所示.

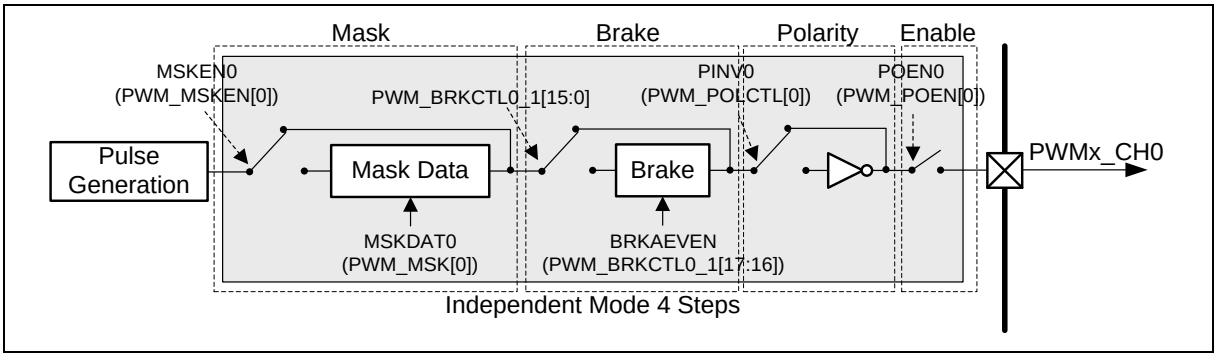


图 6.12-20 PWMx_CH0 独立模式下的输出控制

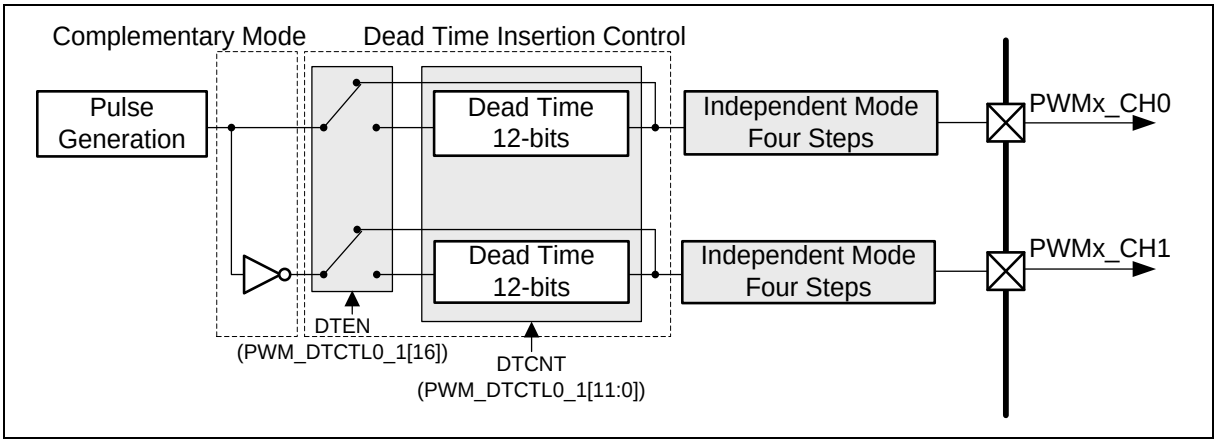


图 6.12-21 PWMx_CH0 和 PWMx_CH1 在互补模式下的输出控制

6.12.5.17 死区插入

互补应用中，在应用于驱动外部设备例如开关电源时，为了防止系统或设备烧坏，互补通道输出常常需要通过死区发生器在输出波形中插入一段低电平来安全驱动设备，这段低电平时段我们称为“死区时间”。因此，死区控制对实现互补系统正确操作非常关键。用户可以通过设置通道n的DTEN (PWM_DTCTLn[16])相应位来使能死区功能，设置DTCNT (PWM_DTCTLn[11:0])来控制死区时间周期。死区时间可以通过以下公式计算：

死区时间 = (DTCNT (PWM_DTCTLn[11:0])+1) * PWMx_CLK 周期

死区插入时钟源可以通过设置DTCKSEL (PWM_DTCTLn_m[24])为1来选择预分频器输出。默认设置中，时钟源为PWM_CLK，即预分频器的输入时钟。死区时间可以通过下面公式计算：

死区时间 = (DTCNT (PWM_DTCTLn[11:0])+1) *

(CLKPSC (PWM_CLKPSCn [11:0])+1)*PWMx_CLK周期

注意PWM_DTCTLn_m 是写保护寄存器.

图 6.12-22是一对PWM信号的死区插入波形.

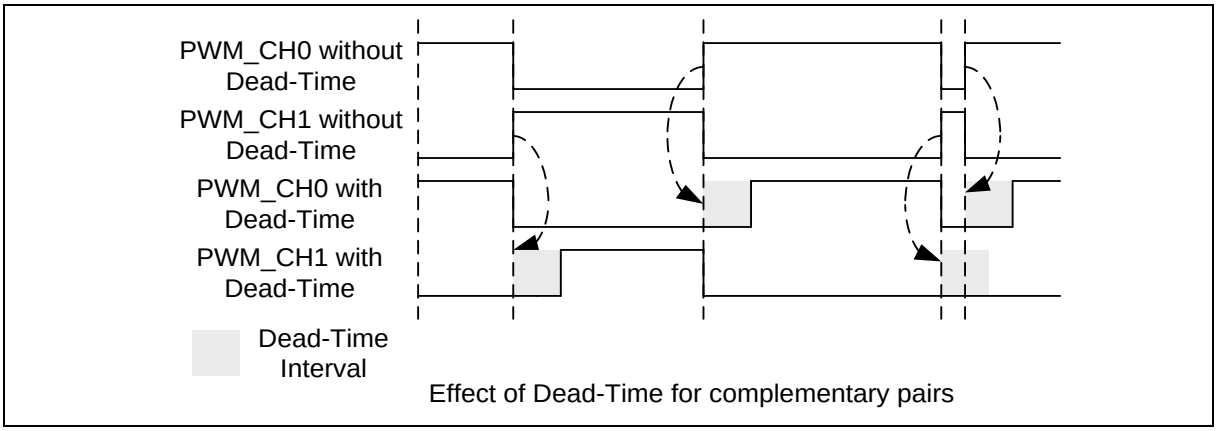


图 6.12-22 死区插入

6.12.5.18 PWM 屏蔽输出功能

用户通过设置PWM屏蔽使能控制寄存器(PWM_MSKEN)和PWM屏蔽数据寄存器(PWM_MSK)，可以对每个PWM的输出实现手动覆盖。通过这些设置，PWM通道的输出可以被强制设定为特定的逻辑状态，无关于比较单元。PWM屏蔽位在用于控制各类电子整流电机(ECM)时，如BLDC电机，非常实用。PWM_MSKEN寄存器有六个位，MSKENn (PWM_MSKEN[5:0]) 。如果MASKENn置高，通道n输出将被覆盖。PWM_MSK寄存器有六个位，MSKDATn (PWM_MSK[5:0])。当通道被覆盖，MSKDATn位的值决定PWM通道n的输出状态。图 6.12-23为如何将PWM屏蔽控制器用于覆盖输出。

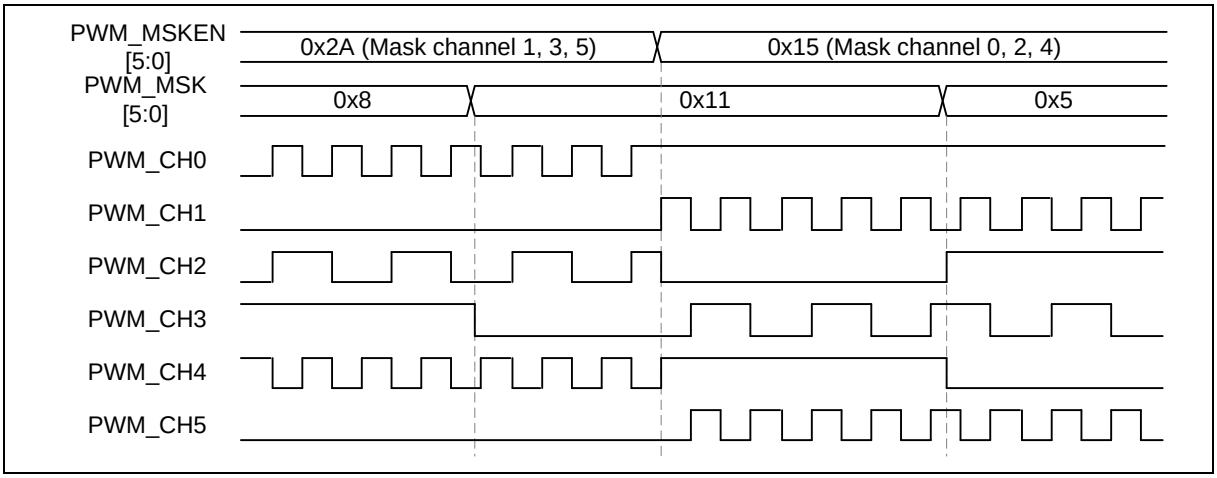


图 6.12-23 屏蔽控制波形图解

6.12.5.19 PWM 刹车

每个PWM模块都有两个外部输入刹车控制信号。用户可以通过BNF寄存器的BKxSRC位选择是否启用PWMx_BRAKEy(x=0,1, y=0,1)引脚作为刹车的输入源。外部信号紧接着会经过一个3位滤波器，进行滤波。用户可以通过BNF寄存器的BRKxNFEN位使能噪音滤波器功能。噪音滤波器的采样时钟可以通过BNF寄存器的BRKxNFSEL位选择以适应不同的噪音特性。通过设置BRKxFCNT，用户可以定义滤波器需要多少次采样时钟周期来确认有效刹车信号。

此外，外部信号可以通过设置BNF寄存器的BRKxPINV(x表示输入引脚0或1)来选择刹车控制信号的极性。设置BRKxPINV位为0，当PWMx_BRAKEy(x=0,1, y=0,1)引脚状态从低到高时，刹车事件就会发生。设置BRKxPINV位为1，当PWMx_BRAKEy引脚状态从高到低时，刹车事件就会发生。

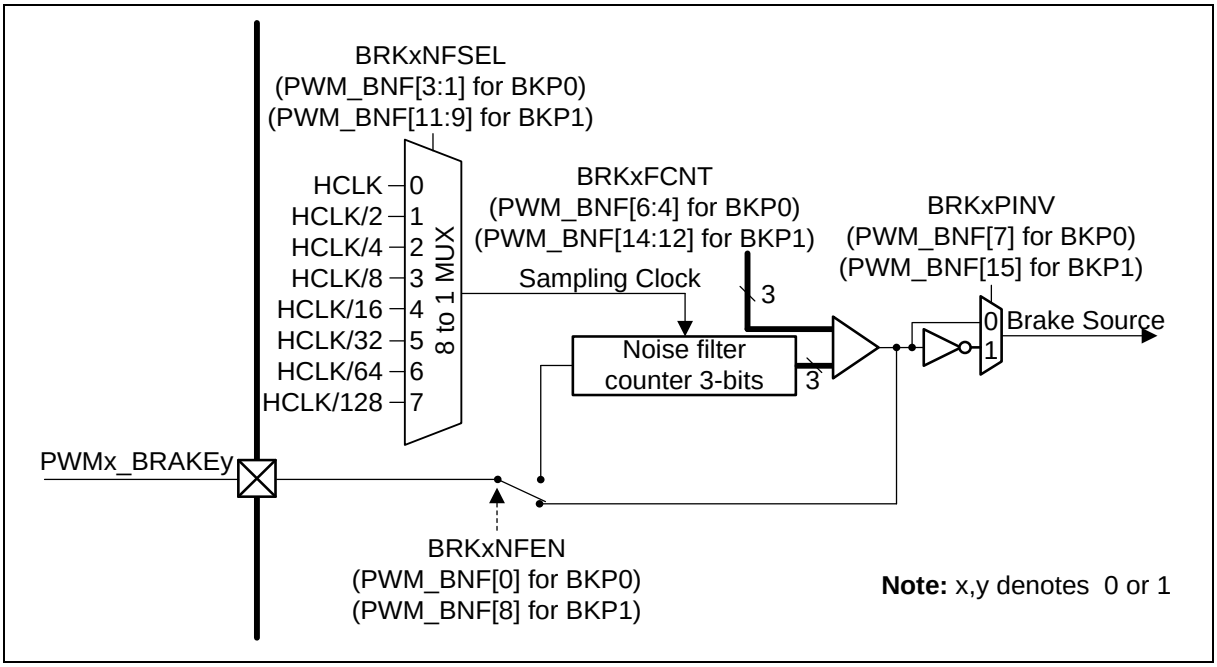


图 6.12-24 刹车噪音滤波器方块图

但是当刹车信号回到高电平，当前PWM周期结束后，PWM输出将变为正常，BRKLSTSn (PWM_INTSTS1[29:24])位会被自动清除。

对于互补模式，一旦刹车事件发生，通常还需要设置一个安全输出状态到互补输出组。

每组互补通道共享一个刹车功能，如图 6.12-25所示. 当刹车事件发生时，用户可以设置偶数通道的BRKAEVEN (PWM_BRKCTL0_1[17:16])和奇数通道的BRKAODD (PWM_BRKCTL0_1[19:18]) ，来确保通道输出的安全状态。刹车功能支持两种刹车检测：边沿检测和电平检测。当边沿检测发现刹车信号并且BRKEIENn_m (PWM_INTEN1[2:0])被使能时，刹车功能产生BRK_INT中断。这个中断需要用软件清除，且在中断清除后BRKESTSn (PWM_INTSTS1[21:16])刹车状态将一直保持直到下一个PWM周期开始。利用电平检测功能，刹车功能还可以用另一种方式操作。一旦电平检测发现刹车信号并且BRKLIENn_m(PWM_INTEN1[10:8])被使能，刹车功能将产生BRK_INT中断。但在电平刹车源恢复到高电平，并且在刹车条件已消失、中断未清除的PWM周期中通过“低电平检测”后，BRKLSTSn (PWM_INTSTS1[29:24])刹车状态将自动恢复到正常输出。

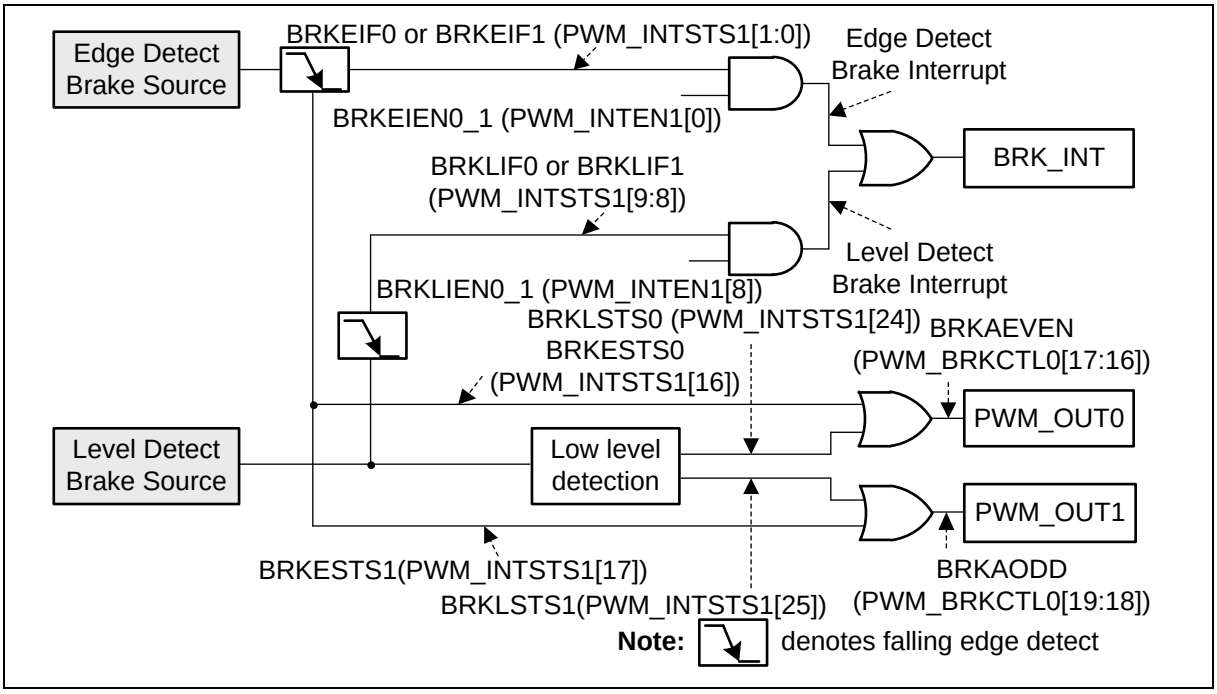


图 6.12-25 PWMx_CH0 和 PWMx_CH1 通道组刹车方块图

图 6.12-26为PWM_CH0 和 PWM_CH1通道组的边沿检测示例。示例中，边沿检测到两次刹车事件。当刹车事件发生，BRKEIF0 和 BRKEIF1都被置1 并且BRKESTS0 和 BRKESTS1也被置1，表明PWM_CH0 和 PWM_CH1处于刹车状态。当第一个事件发生，软件需写1清BRKEIF0。然后在下一个PWM周期开始时硬件清除BRKESTS0。此时即使刹车事件依旧存在，PWM_CH0仍然输出正常波形。第二个事件也触发相同标志，但此时软件需写1去清BRKEIF1，此后在下个PWM周期开始PWM_CH1正常输出。

相对于边沿检测例子，图 6.12-27为PWM_CH0 和 PWM_CH1通道组的电平检测波形示例。示例中，BRKLIF0 和 BRKLIF1只可以表示刹车事件已发生。在下一个PWM周期开始时不管BRKLIF0 和 BRKLIF是什么状态，BRKLSTS0 和 BRKLSTS1的刹车状态都会被自动恢复。

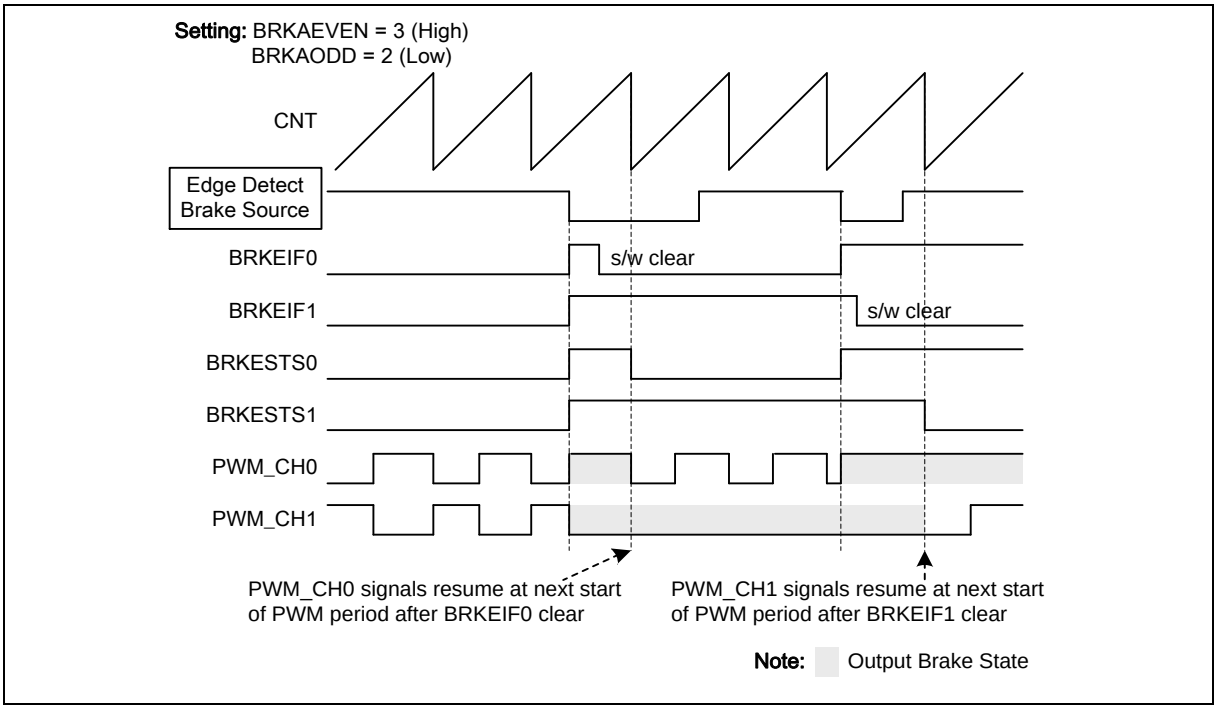


图 6.12-26 PWMx_CH0 和 PWMx_CH1 通道组的边沿检测波形

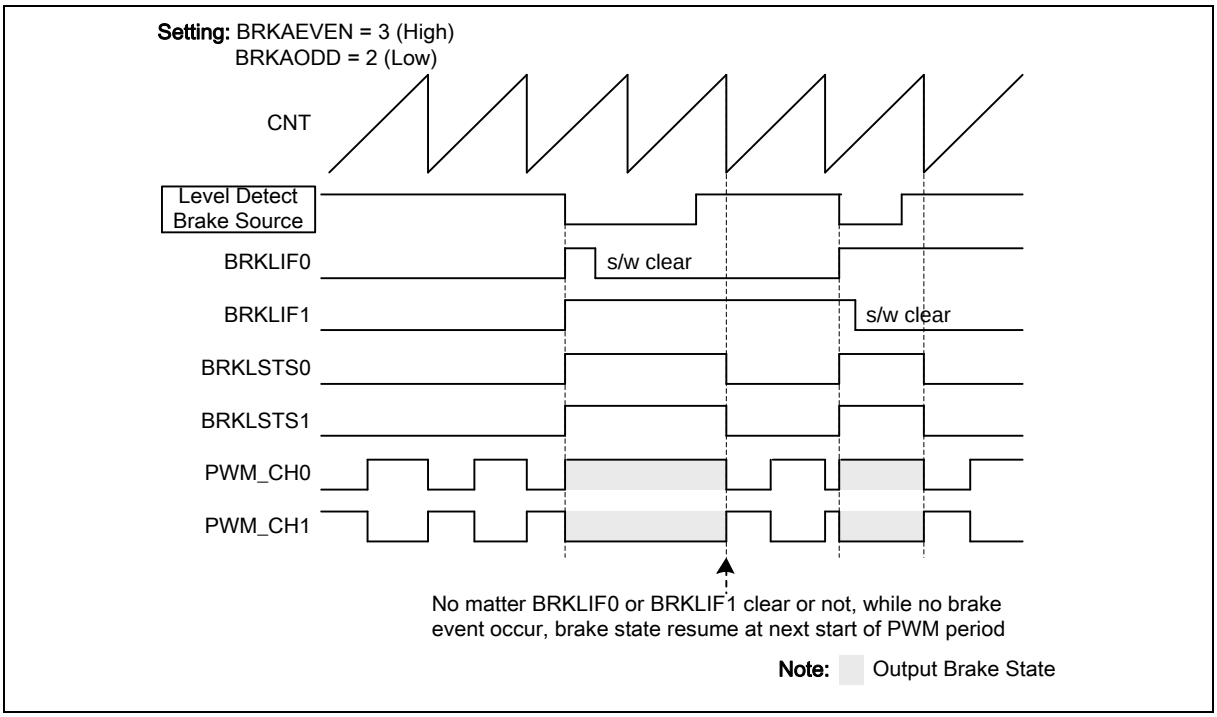


图 6.12-27 PWMx_CH0 和 PWMx_CH1 通道组的电平检测波形

两种检测器检测六个同样的刹车源：两个从外部输入的信号，两个来自模拟比较器(ACMP)，一个来自系统故障和一个来自软件触发，如图 6.12-28所示. 仅当内部ACMP0_O 或 ACMP1_O信号从低到高，ACMP刹车源才会被侦测到。

在以上描述的几个刹车源中，来自系统故障刹车源可以被设定为几种不同的系统故障条件，这些故障条

件包括时钟故障、欠压侦测、SRAM奇偶校验错误和内核锁死。 图 6.12-29显示通过设置相应使能位，系统故障条件可以作为一个系统故障刹车源发送给PWM刹车。

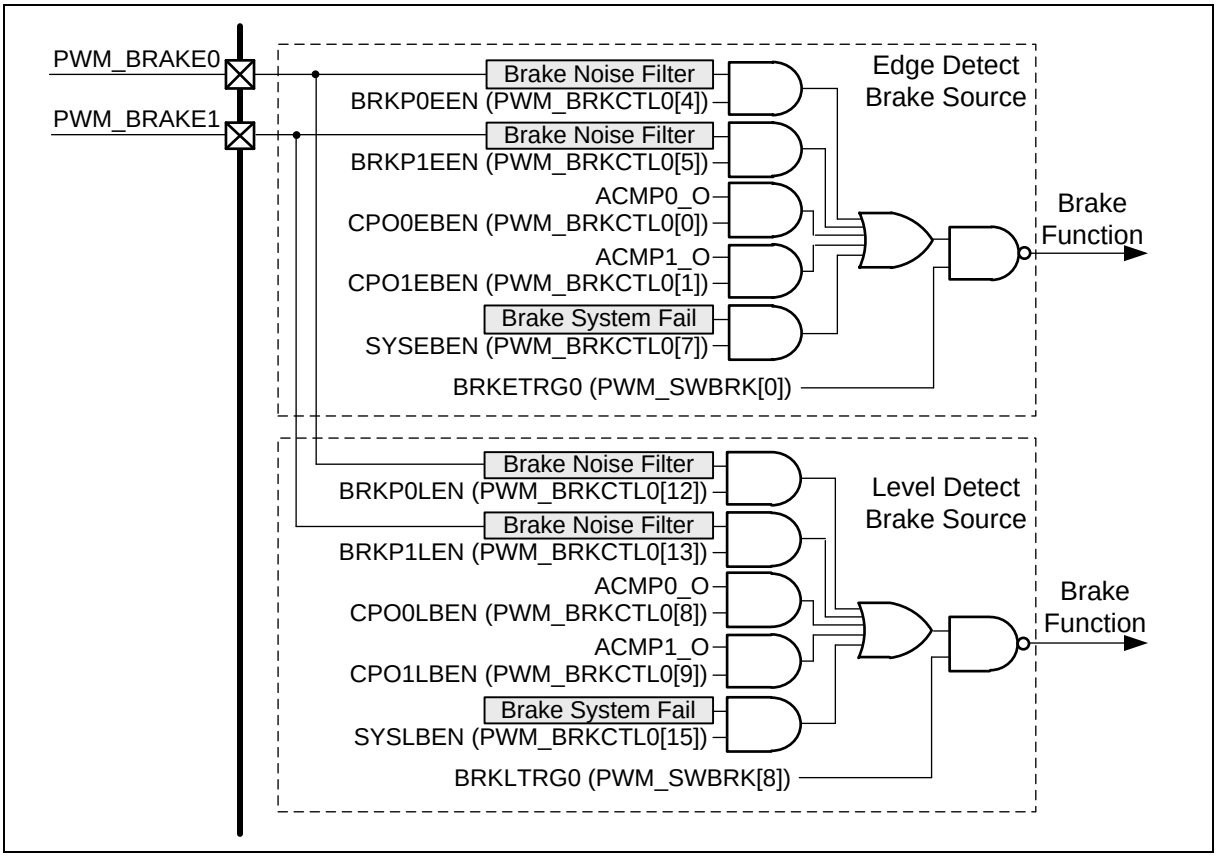


图 6.12-28 刹车源框图

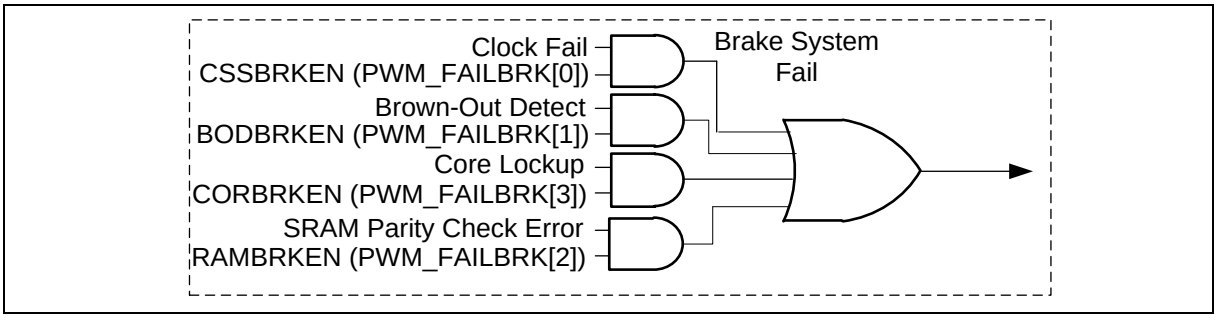
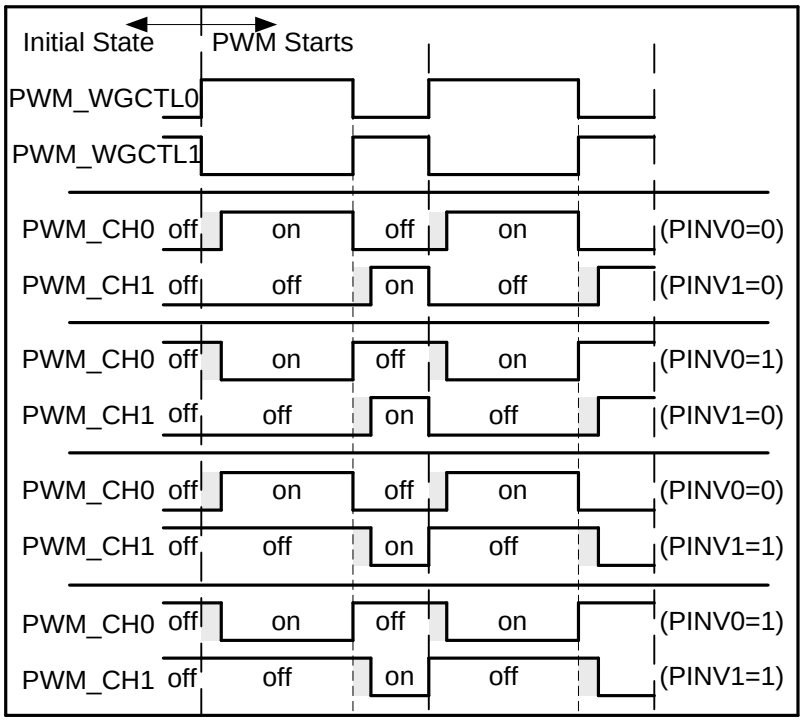


图 6.12-29 系统故障刹车框图

6.12.5.20 极性控制

每个PWM端口，从PWM_CH0 到 PWM_CH5，都有一个独立的极性控制模块来配置PWM输出的极性状态。默认PWM输出是高有效，这意味着PWM关闭状态时输出为低，打开时输出为高。对每个独立的PWM通道，PWM的极性可以通过PWM负极控制寄存器(PWM_POLCTL)来设置。图 6.12-30展示的是PWM不同极性设置开始前的初始状态。



Dead-time insertion; It is only effective in complementary mode
Note: PINVx: Negative Polarity control bits; It controls the PWM output initial state and polarity, x denotes 0 or 1.

图 6.12-30 带上升沿死区插入初始状态和极性控制

6.12.5.21 同步启动功能

当SSEN0 (PWM_SSCTL[0])置位，同步启动功能将被使能。用户可以通过SSRC (PWM_SSCTL[9:8])选择同步源是PWM0或PWM1，一旦使能同步启动功能，且CNTSEN (PWM_SSTRG)置1，所选择的PWM通道（包含每个PWM的通道0~通道5）将同时开始计数。注：置位CNTSEN (PWM_SSTRG)也将置位计数器使能位(CNTENn, n 为 0 ~ 5)以开始计数。

6.12.5.22 PWM 中断发生器.

每个PWM发生器有三个独立中断，如图 6.12-31所示.

第一个PWM中断(PWM_INT)来自PWM互补组事件。计数器可以产生零点中断标志ZIFn (PWM_INTSTS0[5:0])和周期点中断标志PIFn (PWM_INTSTS0[13:8])。当PWM通道n 计数器值等于PWM_CMPDATn比较器值的时候，根据计数方向将触发不同的中断标志。如发生在向上计数则向上中断标志CMPUIFn (PWM_INTSTS0[21:16])被置1，如果发生在相反方向则向下计数中断标志CMPDIFn (PWM_INTSTS0[29:24])被置1。如果相应的中断使能位置1，事件将触发产生中断信号。

第二个中断是捕获中断(CAP_INT)，在NVIC 中共享PWM_INT中断。当CRLIFn (PWM_CAPIF[5:0])被触发，并且捕获上升沿中断使能位CAPRIENn (PWM_CAPIEN[5:0])置1， CAP_INT中断就可以产生。或者在信号下降沿中，捕获下降沿中断使能位CAPFIENn (PWM_CAPIEN[13:8])置1，CFLIFn (PWM_CAPIF[13:8])标志可以被触发。

最后是刹车中断(BRK_INT)，在PWM刹车章节有BRK_INT详细描述。

图 6.12-31表示PWM中断架构。

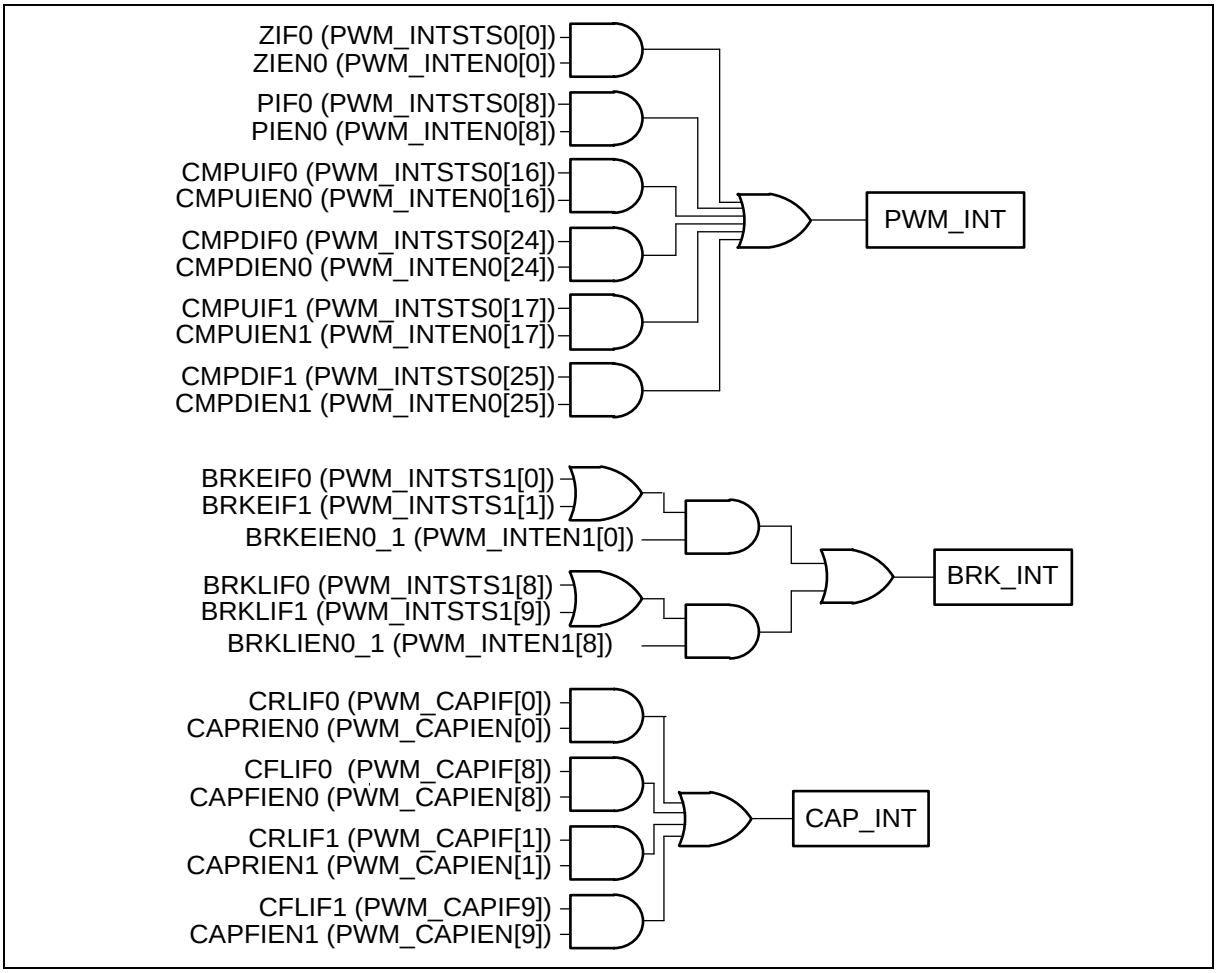


图 6.12-31 PWM_CH0 和 PWM_CH1 组中断架构图

6.12.5.23 PWM 触发ADC发生器

PWM可以作为一个ADC转换触发源。每一个PWM通道组都共享同样的触发源。设置TRGSELn来选择触发源，TRGSELn 有 TRGSEL0，TRGSEL1…TRGSEL5 ，它们分别位于 PWM_ADCTS0[3:0], PWM_ADCTS0[11:8], PWM_ADCTS0[19:16],PWM_ADCTS0[27:24],PWM_ADCTS1[3:0] 和 PWM_ADTS1[11:8]。设置TRGENn来使能触发输出到 ADC，TRGENn 是 TRGEN0，TRGEN1，…，TRGEN5 ，他们分别位于 PWM_ADCTS0[7], PWM_ADCTS0[15], PWM_ADCTS0[23], PWM_ADCTS0[31], PWM_ADCTS1[7] 和PWM_ADCTS1[15]。数字n(n = 0,1, ..,5)表示PWM通道数。

一对通道可以有7个PWM事件触发源可选择，图 6.12-32所示为PWM_CH0和PWM_CH1的范例。通过设置PERIOD和CMPDAT，PWM可以在不同时刻触发ADC开始转换 。 图 6.12-33是在上下计数模式触发ADC的时序图。

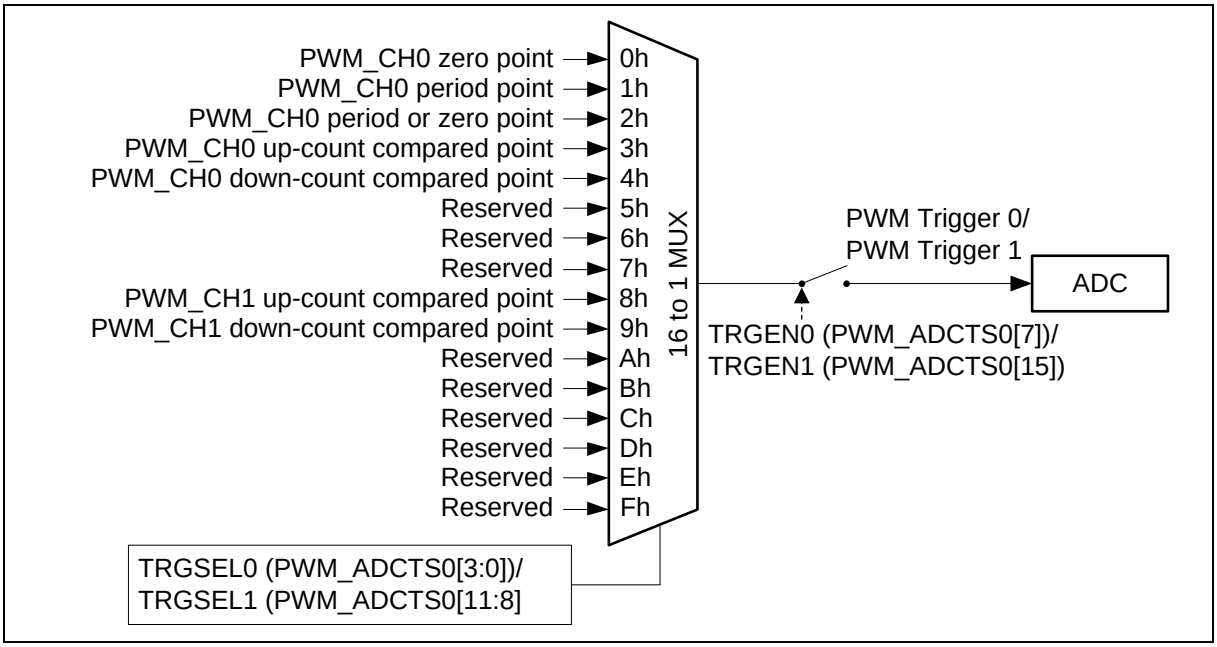


图 6.12-32 PWMx_CH0 和 PWMx_CH1 组触发 ADC 框图

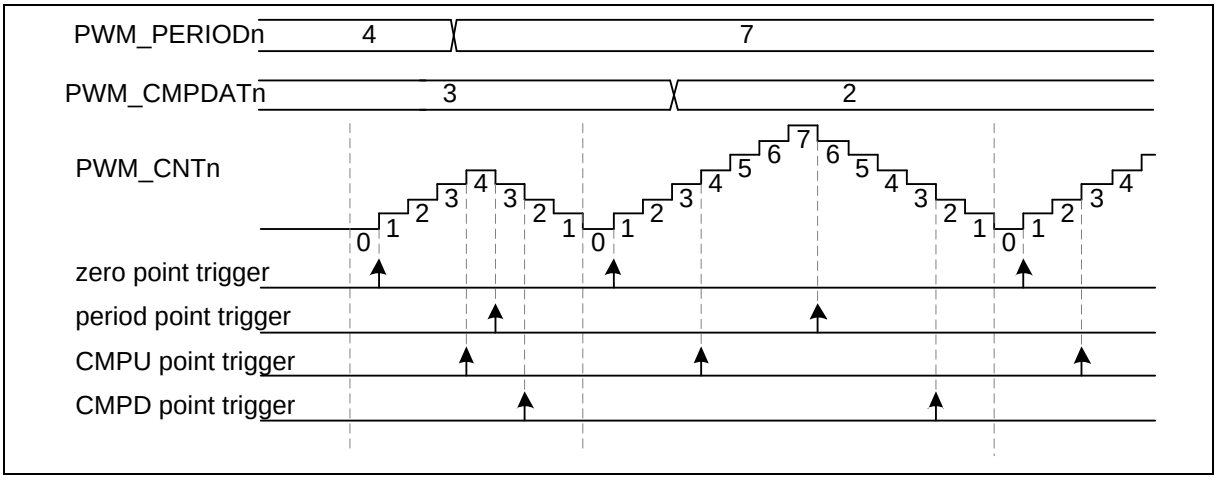


图 6.12-33 上下计数方式 PWM 触发 ADC 的时序波形

6.12.5.24 捕获操作

捕获输入通道和PWM输出通道共用引脚和计数器。计数器可以是上或下计数方式。如果输入通道有上升沿或下降沿跳变时，捕获功能将PWM计数器值分别锁存到RCAPDATn (PWM_RCAPDATn[15:0]) 或FCAPDATn (PWM_FCAPDATn[15:0]) 寄存器。如果上升沿或下降沿锁存发生并且相应通道n的上升沿或下降沿中断使能位被设置，即CAPRIENn (PWM_CAPIEN[5:0])设置上升沿中断使能或CAPFIENn (PWM_CAPIEN[13:8])设置下降沿中断使能，捕获功能将产生一个中断CAP_INT (使用PWM_INT向量)。当上升沿或下降沿锁存发生，相应的PWM计数器是否重载PWM_PERIODn值，取决于RCRLDENn或 FCRLDENn 的设置，RCRLDENn 和 FCRLDENn 分别位于 PWM_CAPCTL[21:16] 和PWM_CAPCTL[29:24]。注：相应的GPIO引脚必须通过使能相应的捕获通道n来配置它的捕获功能，捕获通道使能设置位为CAPINENn (PWM_CAPINEN[5:0])。图 6.12-34是通道0的捕获方块图。

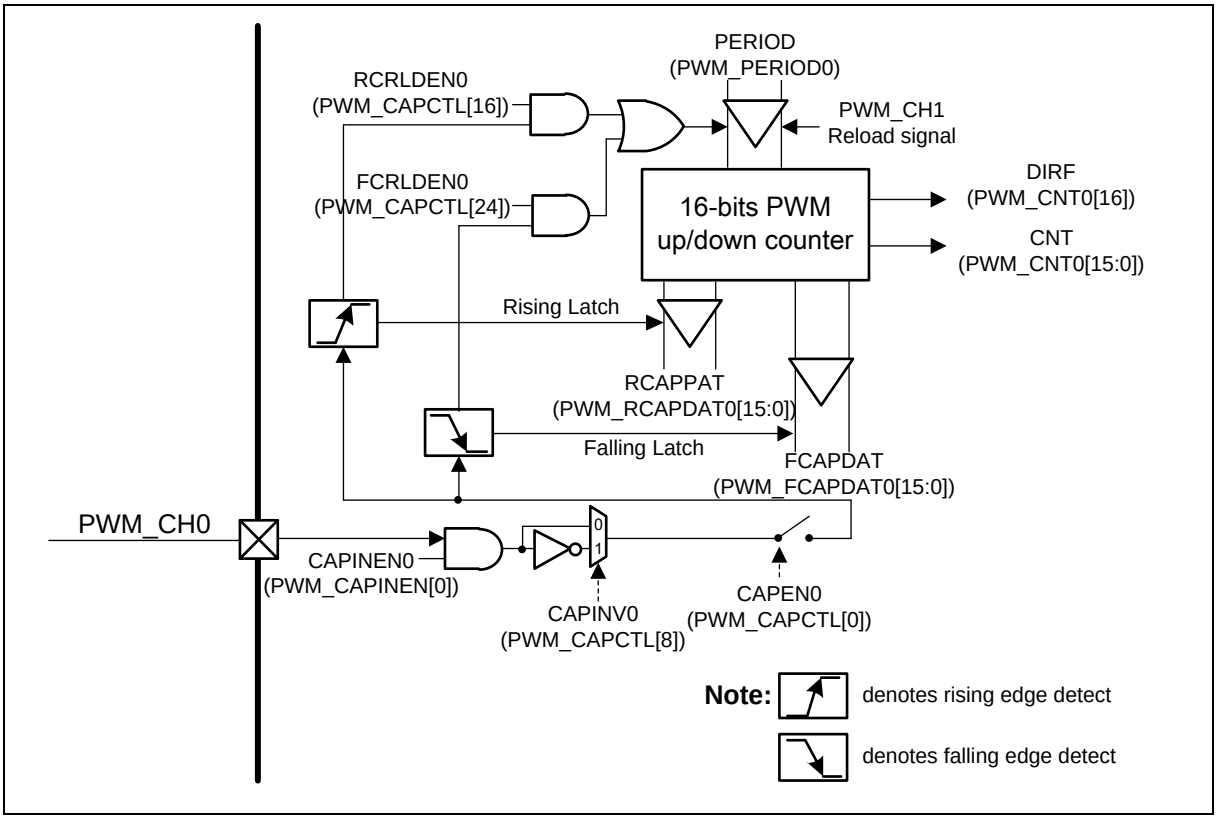


图 6.12-34 PWM_CH0 捕获框图

图 6.12-35为捕获功能时序。在这个例子中，捕获计数器被设置成PWM下计数器类型，而且周期PERIOD被设置成8，所以这个计数器计数的方向是从上到下的，从8到0。当输入捕获引脚检测到一个下降沿时，捕获功能把计数器数值锁存到PWM_FCAPDATn中。当输入捕获引脚检测到一个上升沿，它锁存计数器数值到PWM_RCAPDATn中。在这个时序框图中，最开始检测到的是一个下降沿信号，因为使能了FCRLDENn，捕获器就会重新加载计数器的数值，数值为周期值（PERIOD）。但是在第二次下降沿时，计数器就不会重新被加载，这是因为关闭了FCRLDENn。在这个例子中，计数器在捕获到上升沿时也被重新加载了，原因是RCRLDENn也被使能了。

另外，这种情况如果是设为向上计数方式，计数器将重载零并向上计数到PERIOD值。

图 6.12-35为中断和中断标志产生的时序例子。当上升沿在通道n被检测到时，相应位CRLIFn (PWM_CAPIF[5:0])将被硬件置位。同样，通道n检测到下降沿时，相应位CFLIFn (PWM_CAPIF[13:8])被硬件置位。CRLIFn和CFLIFn可以通过软件写1清除。如果CRLIFn被置位并且CAPRIENn被使能，捕获功能将产生一个中断。如果CFLIFn被置位并且CAPFIENn被使能，捕获功能也将产生一个中断。

在本图没有描述的一个情况是：当CRLIF已经被置位了，如果上升锁存再次发生了，运行状态寄存器CRLIFOVn (PWM_CAPSTS[5:0])将通过硬件被置起，来指示CAPRIF超载。同理，当下降锁存再次发生，对于中断标志CFLIF和超载状态CFLIFOVn (BPWM_CAPSTS[13:8])，也会发生相同的硬件操作。

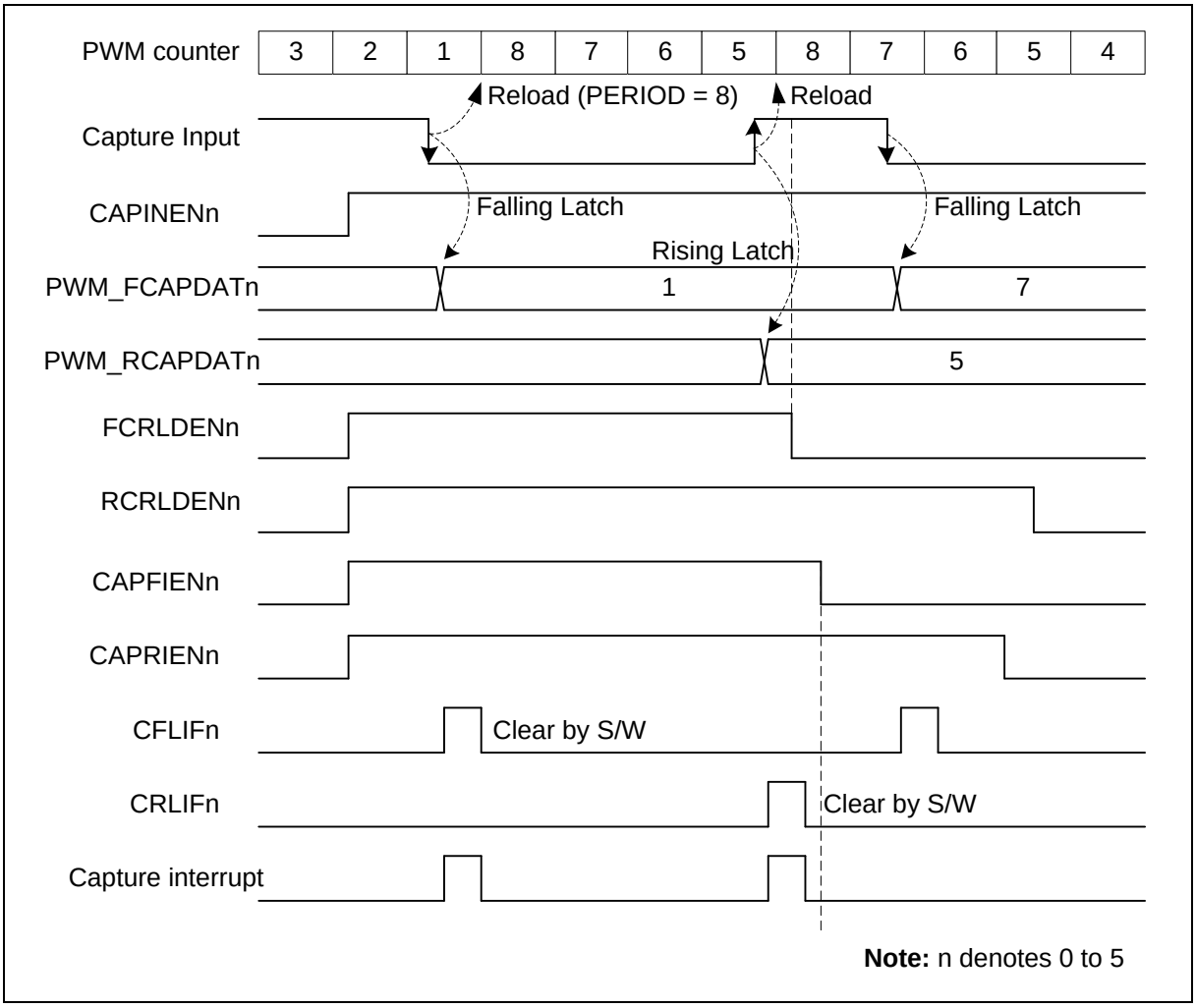


图 6.12-35 捕获操作波形

捕获脉冲宽度如果匹配下述条件，可以通过下述公式计算：

1. 捕获正或负脉冲宽度小于一个计数器周期.
2. 计数器运行在向下计数模式.
3. 通过设置 PWM_CAPCTL 寄存器的 FCRLDENn 和 RCRLDENn 位为 1，计数器可以通过下捕获或上捕获事件重载。

对于负脉冲的情况，通道低电平脉冲宽度为(PWM_PERIODn + 1 - PWM_RCAPDATn)个PWM计数器时间。这里一个PWM计数器时间为(CLKPSC+1) * PWMx_CLK。在图 6.12-35, 低脉冲的宽度是8+1-5 = 4个PWM计数器时间。

对于正脉冲的情况，通道高电平脉冲的宽度为(PWM_PERIODn + 1 - PWM_FCAPDATn) 个PWM计数器时间。这里一个WPM计数器时间为(CLKPSC+1) * PWMx_CLK。 图 6.12-35, 高电平的宽度是8+1-7 = 2个PWM计数器时间。

6.12.5.25 捕获 PDMA 功能

当PWM工作在捕获模式， PWM 模块支持PDMA 数据搬移功能。(**注:** 如果不支持PDMA功能，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节). 当相应的PDMA 使能位 CHENn_m (CHEN0_1 在 PWM_PDMACTL[0], CHEN2_3 在 PWM_PDMACTL[8] ， CHEN4_5 在

PWM_PDMACTL[16], n和m 代表互补通道)被使能, 捕获事件发生后，捕获模块将会产生一个到PDMA控制器的请求。PDMA模块读取了捕获模块CAPBUF (PWM_PDMACAPn_m[15:0], n, m代表互补通道)寄存器的值，并搬移寄存器数据到内存后，PDMA 控制器将会给捕获模块返回一个应答。通过设置CAPMODn_m (CAPMOD0_1 在 PWM_PDMACTL[2:1], CAPMOD2_3 在 PWM_PDMACTL[10:9]，CAPMOD4_5 在 PWM_PDMACTL[18:17]), PDMA可以搬移上升沿或下降沿或两个边沿捕获到的数据到内存。当使用PDMA 来搬移上升沿、下降沿的数据时，不要忘记设置CAPORDn_m (CAPORD0_1 在 PWM_PDMACTL[3], CAPORD2_3 在 PWM_PDMACTL[11] ，CAPORD4_5 在 PWM_PDMACTL[19])寄存器来设定搬移数据的顺序(下降沿捕获数据在先还是上升沿捕获数据在先). 互补通道共用一个PDMA通道。因此，需要设置CHSELn_m (CHSEL0_1 (PWM_PDMACTL[4]), CHSEL2_3 (PWM_PDMACTL[12])， CHSEL4_5 (PWM_PDMACTL[20])) 来确定是选择通道n还是通道m作为PDMA 通道。

图 6.12-36是捕获PDMA的波形。在该例子当中， CHSEL0_1 (PWM_PDMACTL[4]) 被设置为0。因此PDMA将会选择通道0 作为捕获数据搬移通道。. CAPMOD0_1 (PWM_PDMACTL[2:1]) 被设为3，表明上升沿和下降沿捕获到的数据都会被搬移到内存。CAPORD0_1 (PWM_PDMACTL[1]) 被置1， 先搬移上升沿捕获到的数据，然后再搬移下降沿数据。如图 6.12-36所示, CAPRIF0 CRLIF0 和 CAPFIF0 CFLIF0信号的最后一部分有些重叠。PWM_RCAPDAT0 的值11 将被装载到PWM_PDMACAP0_1来等待搬移，但是PWM_FCAPDAT0 的值6则不会被搬移。PWM_PDMACAP0_1 保存了将通过PDMA搬移到内存的数据。图中HWDATA表示PDMA正在搬移的数据。

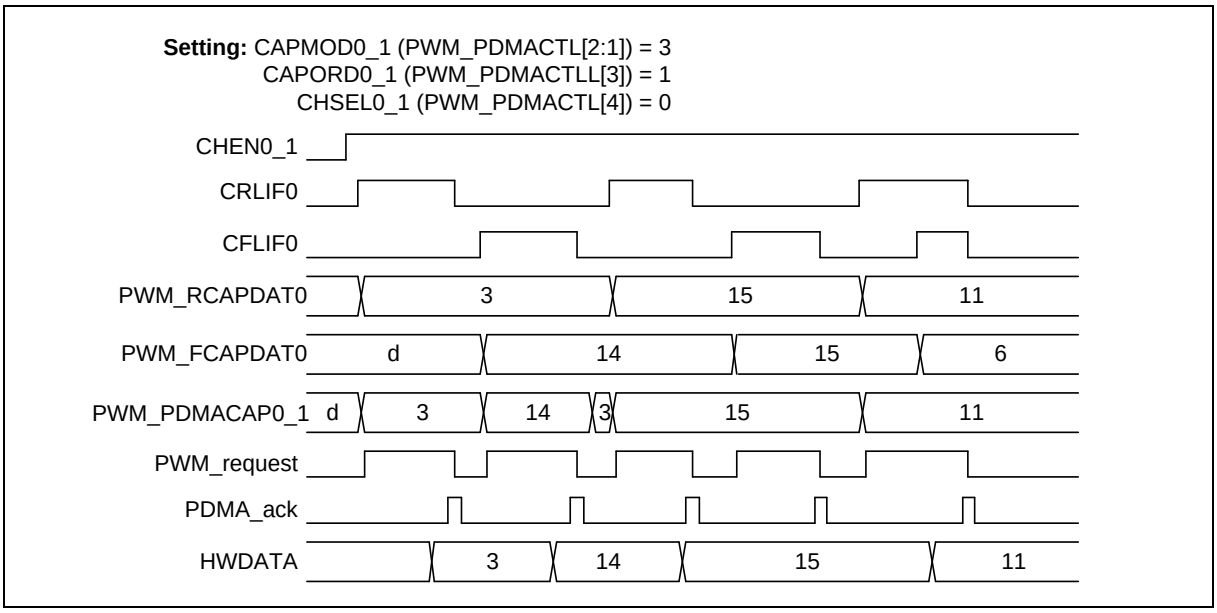


图 6.12-36 通道 0 捕获 PDMA 操作波形

6.12.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读/写

寄存器	偏移	R/W	描述	复位值
PWM 基地址: **PWM0_BA = 0x4005_8000** **PWM1_BA = 0x4005_9000**				
PWM_CTL0 **x=0, 1**	PWMx_BA+0x00	R/W	PWM控制寄存器 0	0x0000_0000

PWM_CTL1 **x=0, 1**	PWMx_BA+0x04	R/W	PWM控制寄存器 1	0x0000_0000
PWM_CLKSRC **x=0, 1**	PWMx_BA+0x10	R/W	PWM时钟源寄存器	0x0000_0000
PWM_CLKPSC0_1 **x=0, 1**	PWMx_BA+0x14	R/W	PWM时钟预分频寄存器0/1	0x0000_0000
PWM_CLKPSC2_3 **x=0, 1**	PWMx_BA+0x18	R/W	PWM时钟预分频寄存器2/3	0x0000_0000
PWM_CLKPSC4_5 **x=0, 1**	PWMx_BA+0x1C	R/W	PWM时钟预分频寄存器4/5	0x0000_0000
PWM_CNTEN **x=0, 1**	PWMx_BA+0x20	R/W	PWM计数器使能寄存器	0x0000_0000
PWM_CNTCLR **x=0, 1**	PWMx_BA+0x24	R/W	PWM清计数器寄存器	0x0000_0000
PWM_PERIOD0 **x=0, 1**	PWMx_BA+0x30	R/W	PWM周期寄存器0	0x0000_0000
PWM_PERIOD2 **x=0, 1**	PWMx_BA+0x38	R/W	PWM周期寄存器2	0x0000_0000
PWM_PERIOD4 **x=0, 1**	PWMx_BA+0x40	R/W	PWM周期寄存器4	0x0000_0000
PWM_CMPDAT0 **x=0, 1**	PWMx_BA+0x50	R/W	PWM比较器寄存器0	0x0000_0000
PWM_CMPDAT1 **x=0, 1**	PWMx_BA+0x54	R/W	PWM比较器寄存器1	0x0000_0000
PWM_CMPDAT2 **x=0, 1**	PWMx_BA+0x58	R/W	PWM比较器寄存器2	0x0000_0000
PWM_CMPDAT3 **x=0, 1**	PWMx_BA+0x5C	R/W	PWM比较器寄存器3	0x0000_0000
PWM_CMPDAT4 **x=0, 1**	PWMx_BA+0x60	R/W	PWM比较器寄存器4	0x0000_0000
PWM_CMPDAT5 **x=0, 1**	PWMx_BA+0x64	R/W	PWM比较器寄存器5	0x0000_0000
PWM_DTCTL0_1 **x=0, 1**	PWMx_BA+0x70	R/W	PWM 死区控制寄存器 0/1	0x0000_0000
PWM_DTCTL2_3 **x=0, 1**	PWMx_BA+0x74	R/W	PWM 死区控制寄存器 2/3	0x0000_0000
PWM_DTCTL4_5 **x=0, 1**	PWMx_BA+0x78	R/W	PWM 死区控制寄存器 4/5	0x0000_0000
PWM_CNT0 **x=0, 1**	PWMx_BA+0x90	R	PWM计数器寄存器 0	0x0000_0000
PWM_CNT2	PWMx_BA+0x98	R	PWM计数器寄存器 2	0x0000_0000

x=0, 1				
PWM_CNT4 **x=0, 1**	PWMx_BA+0xA0	R	PWM计数器寄存器 4	0x0000_0000
PWM_WGCTL0 **x=0, 1**	PWMx_BA+0xB0	R/W	PWM发生寄存器0	0x0000_0000
PWM_WGCTL1 **x=0, 1**	PWMx_BA+0xB4	R/W	PWM发生寄存器1	0x0000_0000
PWM_MSKEN **x=0, 1**	PWMx_BA+0xB8	R/W	PWM屏蔽使能寄存器	0x0000_0000
PWM_MSK **x=0, 1**	PWMx_BA+0xBC	R/W	PWM屏蔽数据寄存器	0x0000_0000
PWM_BNF **x=0, 1**	PWMx_BA+0xC0	R/W	PWM刹车噪声滤波器寄存器	0x0000_0000
PWM_FAILBRK **x=0, 1**	PWMx_BA+0xC4	R/W	PWM系统故障刹车控制寄存器	0x0000_0000
PWM_BRKCTL0_1 **x=0, 1**	PWMx_BA+0xC8	R/W	PWM刹车边沿检测控制寄存器0/1	0x0000_0000
PWM_BRKCTL2_3 **x=0, 1**	PWMx_BA+0xCC	R/W	PWM刹车边沿检测控制寄存器2/3	0x0000_0000
PWM_BRKCTL4_5 **x=0, 1**	PWMx_BA+0xD0	R/W	PWM刹车边沿检测控制寄存器4/5	0x0000_0000
PWM_POLCTL **x=0, 1**	PWMx_BA+0xD4	R/W	PWM引脚极性反转寄存器	0x0000_0000
PWM_POEN **x=0, 1**	PWMx_BA+0xD8	R/W	PWM输出使能寄存器	0x0000_0000
PWM_SWBRK **x=0, 1**	PWMx_BA+0xDC	W	PWM软件刹车控制寄存器	0x0000_0000
PWM_INTEN0 **x=0, 1**	PWMx_BA+0xE0	R/W	PWM中断使能寄存器0	0x0000_0000
PWM_INTEN1 **x=0, 1**	PWMx_BA+0xE4	R/W	PWM中断使能寄存器1	0x0000_0000
PWM_INTSTS0 **x=0, 1**	PWMx_BA+0xE8	R/W	PWM中断标志寄存器0	0x0000_0000
PWM_INTSTS1 **x=0, 1**	PWMx_BA+0xEC	R/W	PWM中断标志寄存器1	0x0000_0000
PWM_ADCTS0 **x=0, 1**	PWMx_BA+0xF8	R/W	PWM触发ADC源选择寄存器0	0x0000_0000
PWM_ADCTS1 **x=0, 1**	PWMx_BA+0xFC	R/W	PWM触发ADC源选择寄存器1	0x0000_0000
PWM_SSCTL **x=0, 1**	PWMx_BA+0x110	R/W	PWM同步开始控制寄存器	0x0000_0000

PWM_SSTRG x=0, 1	PWMx_BA+0x114	W	PWM同步开始触发寄存器	0x0000_0000
PWM_STATUS x=0, 1	PWMx_BA+0x120	R/W	PWM状态寄存器	0x0000_0000
PWM_CAPINEN x=0, 1	PWMx_BA+0x200	R/W	PWM捕获输入使能寄存器	0x0000_0000
PWM_CAPCTL x=0, 1	PWMx_BA+0x204	R/W	PWM捕获控制寄存器	0x0000_0000
PWM_CAPSTS x=0, 1	PWMx_BA+0x208	R	PWM捕获状态寄存器	0x0000_0000
PWM_RCAPDAT0 x=0, 1	PWMx_BA+0x20C	R	PWM上升沿捕获数据寄存器0	0x0000_0000
PWM_FCAPDAT0 x=0, 1	PWMx_BA+0x210	R	PWM下降沿捕获数据寄存器0	0x0000_0000
PWM_RCAPDAT1 x=0, 1	PWMx_BA+0x214	R	PWM上升沿捕获数据寄存器1	0x0000_0000
PWM_FCAPDAT1 x=0, 1	PWMx_BA+0x218	R	PWM下降沿捕获数据寄存器1	0x0000_0000
PWM_RCAPDAT2 x=0, 1	PWMx_BA+0x21C	R	PWM上升沿捕获数据寄存器2	0x0000_0000
PWM_FCAPDAT2 x=0, 1	PWMx_BA+0x220	R	PWM下降沿捕获数据寄存器2	0x0000_0000
PWM_RCAPDAT3 x=0, 1	PWMx_BA+0x224	R	PWM上升沿捕获数据寄存器3	0x0000_0000
PWM_FCAPDAT3 x=0, 1	PWMx_BA+0x228	R	PWM下降沿捕获数据寄存器3	0x0000_0000
PWM_RCAPDAT4 x=0, 1	PWMx_BA+0x22C	R	PWM上升沿捕获数据寄存器4	0x0000_0000
PWM_FCAPDAT4 x=0, 1	PWMx_BA+0x230	R	PWM下降沿捕获数据寄存器4	0x0000_0000
PWM_RCAPDAT5 x=0, 1	PWMx_BA+0x234	R	PWM上升沿捕获数据寄存器5	0x0000_0000
PWM_FCAPDAT5 x=0, 1	PWMx_BA+0x238	R	PWM下降沿捕获数据寄存器5	0x0000_0000
PWM_PDMACTL x=0, 1	PWMx_BA+0x23C	R/W	PWM PDMA控制寄存器	0x0000_0000
PWM_PDMACAP0_1 x=0, 1	PWMx_BA+0x240	R	PWM捕获通道01 PDMA寄存器	0x0000_0000
PWM_PDMACAP2_3 x=0, 1	PWMx_BA+0x244	R	PWM捕获通道23 PDMA寄存器	0x0000_0000
PWM_PDMACAP4_5	PWMx_BA+0x248	R	PWM捕获通道45 PDMA寄存器	0x0000_0000

x=0, 1				
PWM_CAPIEN x=0, 1	PWMx_BA+0x250	R/W	PWM捕获中断使能寄存器	0x0000_0000
PWM_CAPIF x=0, 1	PWMx_BA+0x254	R/W	PWM捕获中断标志寄存器	0x0000_0000
PWM_PBUF0 x=0, 1	PWMx_BA+0x304	R	PWM PERIOD0缓存	0x0000_0000
PWM_PBUF2 x=0, 1	PWMx_BA+0x30C	R	PWM PERIOD2缓存	0x0000_0000
PWM_PBUF4 x=0, 1	PWMx_BA+0x314	R	PWM PERIOD4缓存	0x0000_0000
PWM_CMPBUF0 x=0, 1	PWMx_BA+0x31C	R	PWM CMPDAT0缓存	0x0000_0000
PWM_CMPBUF1 x=0, 1	PWMx_BA+0x320	R	PWM CMPDAT1缓存	0x0000_0000
PWM_CMPBUF2 x=0, 1	PWMx_BA+0x324	R	PWM CMPDAT2缓存	0x0000_0000
PWM_CMPBUF3 x=0, 1	PWMx_BA+0x328	R	PWM CMPDAT3缓存	0x0000_0000
PWM_CMPBUF4 x=0, 1	PWMx_BA+0x32C	R	PWM CMPDAT4缓存	0x0000_0000
PWM_CMPBUF5 x=0, 1	PWMx_BA+0x330	R	PWM CMPDAT5缓存	0x0000_0000

6.12.7 寄存器描述

PWM控制寄存器 0 (PWM_CTL0)

寄存器	偏移	R/W	描述	复位值
PWM_CTL0	PWMx_BA+0x00	R/W	PWM控制寄存器 0	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	**DBGHALT**	**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**IMMLDEN5**	**IMMLDEN4**	**IMMLDEN3**	**IMMLDEN2**	**IMMLDEN1**	**IMMLDEN0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CTRLD5**	**CTRLD4**	**CTRLD3**	**CTRLD2**	**CTRLD1**	**CTRLD0**

位	描述	
[31]	**DBGTRIOFF**	**ICE调试模式禁止位(写保护)** 0 = ICE 调试模式影响PWM输出 ICE 调试模式期间PWM引脚将强制作为三态模式。 1= 禁止ICE 调试模式影响PWM输出。 PWM引脚将保持正常输出不管是否在ICE调试模式 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[30]	**DBGHALT**	**ICE 调试模式计数暂停(写保护)** 如果 计数暂停被使能，PWM所有计数器将保持当前值直到退出ICE调试模式。 0 = 禁止ICE 调试模式计数暂停 1 = 使能ICE 调试模式计数暂停 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[29:22]	**Reserved**	保留.
[n+16] n=0,1…5	**IMMLDENn**	**立即装载使能位** 每位n控制相应PWM通道n 0 = 每个周期结束时PERIOD将被载入到PBUF。通过设定CTRLD位， CMPDAT将在每个周期的终点或者中心点被载入到CMPBUF。 1= 当软件更新 PERIOD/CMPDAT ， PERIOD/CMPDAT 将 立 即分 别 被载 入 到 PBUF和CMPBUF。 注：如果IMMLDENn使能，WINLDENn 和 CTRLDn无效
[15:6]	**Reserved**	保留.
[n] n=0,1…5	**CTRLDn**	**中心重载使能位** 在上-下计数方式，在每个周期终点PERIOD将被载入到PBUF。每个周期中心点CMPDAT将载入到CMPBUF。

PWM控制寄存器 1 (PWM_CTL1)

寄存器	偏移	R/W	描述	复位值
PWM_CTL1	PWMx_BA+0x04	R/W	PWM控制寄存器 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					OUTMODE4	OUTMODE2	OUTMODE0
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						CNTTYPE4	
7	6	5	4	3	2	1	0
Reserved		CNTTYPE2		Reserved		CNTTYPE0	

位	描述	
[31:27]	**Reserved**	保留.
[26:24]	**OUTMODEn**	**PWM 输出模式** 每位n控制相应PWM通道n 0=PWM独立模式 1=PWM互补模式 **注**：每对通道的这些位必须都设置为相同模式。
[23:10]	**Reserved**	保留.
[9:8]	**CNTTYPE4**	**PWM计数器类型4** 这两位控制通道5和通道4 00 = 向上计数类型（在捕获模式下也支持）. 01 = 向下计数类型（在捕获模式下也支持）. 10 = 上下计数类型. 11 = 保留.
[7:6]	**Reserved**	保留.
[5:4]	**CNTTYPE2**	**PWM计数器类型2** 这两位控制通道3和通道2 00 = 向上计数类型（在捕获模式下也支持）. 01 = 向下计数类型（在捕获模式下也支持）. 10 = 上下计数类型. 11 = 保留.
[3:2]	**Reserved**	保留.
[1:0]	**CNTTYPE0**	**PWM计数器类型0** 这两位控制通道1和通道0

		00 = 向上计数类型（在捕获模式下也支持）. 01 = 向下计数类型（在捕获模式下也支持）. 10 = 上下计数类型. 11 = 保留.

PWM时钟源寄存器 (PWM_CLKSRC)

寄存器	偏移	R/W	描述	复位值
PWM_CLKSRC	PWMx_BA+0x10	R/W	PWM时钟源寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**ECLKSRC4**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ECLKSRC2**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**ECLKSRC0**		

位	描述	
[31:19]	**Reserved**	保留.
[18:16]	**ECLKSRC4**	**PWM_CH45外部时钟源选择** 000 = PWMx_CLK, x 为 0 或 1. 001 = TIMER0 溢出. 010 = TIMER1 溢出. 011 = TIMER2 溢出. 100 = TIMER3 溢出. 其它 = 保留.
[15:11]	**Reserved**	保留.
[10:8]	**ECLKSRC2**	**PWM_CH23外部时钟源选择** 000 = PWMx_CLK, x 为 0 或 1. 001 = TIMER0 溢出. 010 = TIMER1 溢出. 011 = TIMER2 溢出. 100 = TIMER3 溢出. 其它 = 保留.
[7:3]	**Reserved**	保留.
[2:0]	**ECLKSRC0**	**PWM_CH01外部时钟源选择** 000 = PWMx_CLK, x 为 0 或 1. 001 = TIMER0 溢出. 010 = TIMER1 溢出. 011 = TIMER2 溢出. 100 = TIMER3 溢出.

		其它 = 保留..

PWM时钟预分频寄存器 0_1, 2_3, 4_5 (PWM_CLKPSC0_1, 2_3, 4_5)

寄存器	偏移	R/W	描述	复位值
PWM_CLKPSC0_1	PWMx_BA+0x14	R/W	PWM时钟预分频寄存器0/1	0x0000_0000
PWM_CLKPSC2_3	PWMx_BA+0x18	R/W	PWM时钟预分频寄存器2/3	0x0000_0000
PWM_CLKPSC4_5	PWMx_BA+0x1C	R/W	PWM时钟预分频寄存器4/5	0x0000_0000

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

位	描述	
[31:12]	**Reserved**	保留.
[11:0]	**CLKPSC**	**PWM计数器时钟预分频** PWM计数器时钟由时钟预分频器决定。每个PWM组共享一个PWM寄存器时钟预分频器。PWM计数器时钟源被(CLKPSC+ 1)除频。

PWM计数器使能寄存器 (PWM_CNTEN)

寄存器	偏移	R/W	描述	复位值
PWM_CNTEN	PWMx_BA+0x20	R/W	PWM计数器使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**CNTEN4**	**Reserved**	**CNTEN2**	**Reserved**	**CNTEN0**

位	描述	
[31:5]	**Reserved**	保留.
[4]	**CNTEN4**	**PWM计数器使能位4** 0 = PWM计数器和时钟分频器停止工作. 1 = PWM计数器和时钟分频器开始工作.
[3]	**Reserved**	保留.
[2]	**CNTEN2**	**PWM计数器使能位2** 0 = PWM计数器和时钟分频器停止工作. 1 = PWM计数器和时钟分频器开始工作.
[1]	**Reserved**	保留.
[0]	**CNTEN0**	**PWM计数器使能位0** 0 = PWM计数器和时钟分频器停止工作. 1 = PWM计数器和时钟分频器开始工作.

PWM清计数器寄存器 (PWM_CNTCLR)

寄存器	偏移	R/W	描述	复位值
PWM_CNTCLR	PWMx_BA+0x24	R/W	PWM清计数器寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**CNTCLR4**	**Reserved**	**CNTCLR2**	**Reserved**	**CNTCLR0**

位	描述	
[31:5]	**Reserved**	保留.
[4]	**CNTCLR4**	**清PWM计数器控制位 4** 此位由软件自动清零. 0 = 无影响. 1 = 清16位PWM计数器到0000H.
[3]	**Reserved**	保留.
[2]	**CNTCLR2**	**清PWM计数器控制位 2** 此位由软件自动清零. 0 = 无影响. 1 = 清16位PWM计数器到0000H.
[1]	**Reserved**	保留.
[0]	**CNTCLR0**	**清PWM计数器控制位 0** 此位由软件自动清零. 0 = 无影响. 1 = 清16位PWM计数器到0000H.

PWM周期寄存器0, 2, 4 (PWM_PERIOD0, 2, 4)

寄存器	偏移	R/W	描述	复位值
PWM_PERIOD0	PWMx_BA+0x30	R/W	PWM周期寄存器0	0x0000_0000
PWM_PERIOD2	PWMx_BA+0x38	R/W	PWM周期寄存器2	0x0000_0000
PWM_PERIOD4	PWMx_BA+0x40	R/W	PWM周期寄存器4	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**PERIOD**	**PWM周期寄存器** 向上计数模式： 此模式下，PWM计数器从0计数到PERIOD，然后从0重新开始计数。 向下计数模式： 此模式下，PWM计数器从PERIOD计数到0，然后从PERIOD重新开始计数。 PWM周期 = (PERIOD+1) * PWM_CLK 周期. 上下计数模式：这个模式PWM计数器从0计数到PERIOD然后递减到0，并重复。 PWM周期 = 2 * PERIOD * PWM_CLK 周期.

PWM比较器寄存器0~5 (PWM_CMPDAT0~5)

寄存器	偏移	R/W	描述	复位值
PWM_CMPDAT0	PWMx_BA+0x50	R/W	PWM比较器寄存器0	0x0000_0000
PWM_CMPDAT1	PWMx_BA+0x54	R/W	PWM比较器寄存器1	0x0000_0000
PWM_CMPDAT2	PWMx_BA+0x58	R/W	PWM比较器寄存器2	0x0000_0000
PWM_CMPDAT3	PWMx_BA+0x5C	R/W	PWM比较器寄存器3	0x0000_0000
PWM_CMPDAT4	PWMx_BA+0x60	R/W	PWM比较器寄存器4	0x0000_0000
PWM_CMPDAT5	PWMx_BA+0x64	R/W	PWM比较器寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**CMP**	**PWM比较寄存器** CMP用于与CNTR比较来产生PWM波形，中断和触发ADC. 独立模式，CMPDAT0~5作为6个独立PWM_CH0~5比较点。 互补模式， CMPDAT0，2，4作为第一比较点，而CMPDAT1，3，5作为第二比较点，对应于相应的三个互补组PWM_CH0和PWM_CH1，PWM_CH2和PWM_CH3，PWM_CH4 和PWM_CH5。

PWM 死区控制寄存器 0_1, 2_3, 4_5 (PWM_DTCTL0_1, 2_3, 4_5)

寄存器	偏移	R/W	描述	复位值
PWM_DTCTL0_1	PWMx_BA+0x70	R/W	PWM 死区控制寄存器 0/1	0x0000_0000
PWM_DTCTL2_3	PWMx_BA+0x74	R/W	PWM 死区控制寄存器 2/3	0x0000_0000
PWM_DTCTL4_5	PWMx_BA+0x78	R/W	PWM 死区控制寄存器 4/5	0x0000_0000

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

位	描述	
[31:25]	**Reserved**	保留.
[24]	**DTCKSEL**	**死区时钟选择(写保护)** 0 = 死区时钟源来自 PWM_CLK. 1 =死区时钟源来自预分频器输出. **注:** 该位写保护，详情请参考REGWRPROT寄存器
[23:17]	**Reserved**	保留.
[16]	**DTEN**	**使能PWM组(PWM_CH0, PWM_CH1) (PWM_CH2, PWM_CH3) (PWM_CH4, PWM_CH5) 死区插入(写保护)** 死区插入只有当该组互补PWM使能才启用。如果死区插入未启用，该互补组输出引脚没有任何延时。 0 = 禁止引脚组死区插入。 1 = 使能引脚组死区插入。 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[15:12]	**Reserved**	保留.
[11:0]	**DTCNT**	**死区计数器 (写保护)** 死区时间可以根据以下公式计算: DTCKSEL=0: 死区时间 = (DTCNT[11:0]+1) * PWM_CLK 周期. DTCKSEL=1: 死区时间= (DTCNT[11:0]+1) * PWM_CLK 周期 * (CLKPSC+1). **注:** 该位写保护，详情参考SYS_REGLCTL寄存器

PWM计数器寄存器 0, 2, 4 (PWM_CNT0, 2, 4)

寄存器	偏移	R/W	描述	复位值
PWM_CNT0	PWMx_BA+0x90	R	PWM计数器寄存器 0	0x0000_0000
PWM_CNT2	PWMx_BA+0x98	R	PWM计数器寄存器 2	0x0000_0000
PWM_CNT4	PWMx_BA+0xA0	R	PWM计数器寄存器 4	0x0000_0000

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

位	描述	
[31:17]	**Reserved**	保留.
[16]	**DIRF**	**PWM方向标志(只读)** 0 =向下计数 1 =向上计数
[15:0]	**CNT**	**PWM数据寄存器(只读)** 用户可以读CNTR以知道16位周期计数器当前值。

PWM发生寄存器0 (PWM_WGCTL0)

寄存器	偏移	R/W	描述	复位值
PWM_WGCTL0	PWMx_BA+0xB0	R/W	PWM发生寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**PRDPCTL5**		**PRDPCTL4**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRDPCTL3		**PRDPCTL2**		**PRDPCTL1**		**PRDPCTL0**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**ZPCTL5**		**ZPCTL4**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ZPCTL3		**ZPCTL2**		**ZPCTL1**		**ZPCTL0**	

位	描述	
[31:28]	**Reserved**	保留.
[17+2n:16+2n] n=0,1..5	**PRDPCTLn**	**PWM周期（中心）点输出控制** 00 =输出状态不变 01 = PWM周期（中心）点输出低 10 = PWM周期（中心）点输出高 11 = PWM周期（中心）点输出翻转 **注1:** 当PWM 计数器计数到 (PERIODn+1)时，PWM可以控制输出电平. **注2:**当PWM计数器运行在上下计数器类型，该位为中心控制点.
[15:12]	**Reserved**	保留.
[1+2n:2n] n=0,1..5	**ZPCTLn**	**PWM零点输出控制** 00 =输出状态不变 01 = PWM零点输出低 10 = PWM零点输出高 11 = PWM零点输出翻转. **注:** 当PWM计数器计数到零点时，可以控制PWM输出电平.

PWM发生寄存器1 (PWM_WGCTL1)

寄存器	偏移	R/W	描述	复位值
PWM_WGCTL1	PWMx_BA+0xB4	R/W	PWM发生寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**CMPDCTL5**		**CMPDCTL4**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDCTL3		**CMPDCTL2**		**CMPDCTL1**		**CMPDCTL0**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CMPUCTL5**		**CMPUCTL4**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPUCTL3		**CMPUCTL2**		**CMPUCTL1**		**CMPUCTL0**	

位	描述	
[31:28]	**Reserved**	保留.
[17+2n:16+2n] n=0,1..5	**CMPDCTLn**	**PWM向下比较点输出控制** 00 =输出状态不变 01 = PWM向下比较点输出低 10 = PWM向下比较点输出高 11 =向下比较点输出翻转 **注1:** 当寄存器向下计数到CMPDAT,可控制PWM输出电平 **注2:** 在互补模式，CMPDCTL 1,3,5的设置是无效的，其输出参照CMPDCTL 0,2,4 。
[15:12]	**Reserved**	保留.
[1+2n:2n] n=0,1..5	**CMPUCTLn**	**PWM向上比较点输出控制** 00 =输出状态不变 01 = PWM 向上比较点输出低 10 = PWM 向上比较点输出高 11 = PWM向上比较点输出翻转 **注1:** 当寄存器向上计数到CMPDAT,PWM可控制输出电平 **注2:** 在互补模式，CMPDCTL 1,3,5的设置是无效的，其输出参照CMPDCTL 0,2,4 。

PWM屏蔽使能寄存器 (PWM_MSKEN)

寄存器	偏移	R/W	描述	复位值
PWM_MSKEN	PWMx_BA+0xB8	R/W	PWM屏蔽使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKEN5**	**MSKEN4**	**MSKEN3**	**MSKEN2**	**MSKEN1**	**MSKEN0**

位	描述	
[31:6]	**Reserved**	保留.
[n] n=0,1..5	**MSKENn**	**PWM屏蔽使能寄存器** 当该位使能，PWM输出信号将被屏蔽。相应的PWM通道n将输出MSKDATn (PWM_MSK[5:0])数据 0 =输出信号不屏蔽 1 = PWM输出信号被屏蔽并输出MSKDATn数据.

PWM屏蔽数据寄存器 (PWM_MSK)

寄存器	偏移	R/W	描述	复位值
PWM_MSK	PWMx_BA+0xBC	R/W	PWM屏蔽数据寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MSKDAT5**	**MSKDAT4**	**MSKDAT3**	**MSKDAT2**	**MSKDAT1**	**MSKDAT0**

位	描述	
[31:6]	**Reserved**	保留.
[n] n=0,1..5	**MSKDATn**	**PWM屏蔽数据位** 如果相应屏蔽功能使能，该数据位控制PWMn输出引脚状态。每位n控制相应PWM通道n 0 = 输出逻辑低到PWM通道n 1 = 输出逻辑高到PWM通道n

PWM刹车噪声滤波器寄存器 (PWM_BNF)

寄存器	偏移	R/W	描述	复位值
PWM_BNF	PWMx_BA+0xC0	R/W	PWM刹车噪声滤波器寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**BK1SRC**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**BK0SRC**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BRK1PINV	**BRK1FCNT**			**BRK1NFSEL**			**BRK1NFEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRK0PINV	**BRK0FCNT**			**BRK0NFSEL**			**BRK0NFEN**

位	描述	
[31:25]	**Reserved**	保留.
[24]	**BK1SRC**	**刹车1引脚源选择** 对于 PWM0 的设置: 0 = 刹车 1 引脚源来自 PWM0_BRAKE1. 1 = 刹车 1 引脚源来自PWM1_BRAKE1. 对于 PWM1 的设置: 0 =刹车 1 引脚源来自PWM1_BRAKE1. 1 =刹车 1 引脚源来自PWM0_BRAKE1.
[23:17]	**Reserved**	保留.
[16]	**BK0SRC**	**刹车 0引脚源选择** 对于 PWM0 的设置: 0 = 刹车 0 引脚源来自 PWM0_BRAKE0. 1 = 刹车 0 引脚源来自PWM1_BRAKE0. 对于 PWM1 的设置: 0 =刹车 0 引脚源来自PWM1_BRAKE0. 1 =刹车 0 引脚源来自PWM0_BRAKE0.
[15]	**BRK1PINV**	**刹车1引脚反向** 0 = PWMx_BRAKE1引脚状态传到负边沿检测器. 1 = PWMx_BRAKE1引脚反向状态传到负边沿检测器..
[14:12]	**BRK1FCNT**	**刹车 1 边沿检测器滤波器** 该位控制刹车1滤波计数器从0到BRK1FCNT计数
[11:9]	**BRK1NFSEL**	**刹车1边沿检测滤波器时钟选择** 000 =滤波器时钟= HCLK.

		001 =滤波器时钟HCLK/2. 010 =滤波器时钟= HCLK/4. 011 =滤波器时钟= HCLK/8. 100 =滤波器时钟= HCLK/16. 101 =滤波器时钟= HCLK/32. 110 =滤波器时钟= HCLK/64. 111 =滤波器时钟= HCLK/128.
[8]	**BRK1NFEN**	**刹车1噪音滤波器使能** 0 = 禁止PWM刹车1噪音滤波器 1 = 使能PWM刹车1噪音滤波器.
[7]	**BRK0PINV**	**刹车0引脚反向** 0 = PWMx_BRAKE0引脚状态传到负边沿检测器. 1 = PWMx_BRAKE0引脚反向状态传到负边沿检测器..
[6:4]	**BRK0FCNT**	**刹车0边沿检测滤波器计数** 寄存器位控制刹车0滤波计数器从0到BRK0FCNT计数
[3:1]	**BRK0NFSEL**	**刹车0边沿检测滤波器时钟选择** 000 ： 滤波器时钟= HCLK. 001 ： 滤波器时钟= HCLK/2. 010 ： 滤波器时钟= HCLK/4. 011 ： 滤波器时钟= HCLK/8. 100 ： 滤波器时钟= HCLK/16. 101 ： 滤波器时钟= HCLK/32. 110 ： 滤波器时钟= HCLK/64. 111 ： 滤波器时钟= HCLK/128.
[0]	**BRK0NFEN**	**PWM刹车0噪音滤波器使能** 0 = 禁止PWM刹车0噪音滤波器 1 = 使能PWM刹车0噪音滤波器

PWM系统故障刹车控制寄存器 (PWM_FAILBRK)

寄存器	偏移	R/W	描述	复位值
PWM_FAILBRK	PWMx_BA+0xC4	R/W	PWM系统故障刹车控制寄存器	0x0000_0000

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

位	描述	
[31:4]	**Reserved**	保留.
[3]	**CORBRKEN**	**内核锁死检测触发PWM刹车功能0使能** 0 =禁止通过内核锁死检测触发刹车功能 1 =使能通过内核锁死检测触发刹车功能
[2]	**RAMBRKEN**	SRAM奇偶校验错误检测触发PWM刹车功能0使能： 0 =禁止通过SRAM奇偶校验错误检测触发刹车功能 1 =使能通过SRAM奇偶校验错误检测触发刹车功能
[1]	**BODBRKEN**	**欠压检测触发PWM刹车功能0使能** 0 =禁止BOD触发刹车功能 1 =使能BOD触发刹车功能
[0]	**CSSBRKEN**	**时钟安全系统检测触发PWM刹车功能0使能** 0 =禁止通过CSS检测触发刹车功能 1 =使能通过CSS检测触发刹车功能

PWM刹车边沿检测控制寄存器0_1, 2_3, 4_5 (PWM_BRKCTL0_1, 2_3, 4_5)

寄存器	偏移	R/W	描述	复位值
PWM_BRKCTL0_1	PWMx_BA+0xC8	R/W	PWM刹车边沿检测控制寄存器0/1	0x0000_0000
PWM_BRKCTL2_3	PWMx_BA+0xCC	R/W	PWM刹车边沿检测控制寄存器2/3	0x0000_0000
PWM_BRKCTL4_5	PWMx_BA+0xD0	R/W	PWM刹车边沿检测控制寄存器4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**BRKAODD**		**BRKAEVEN**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SYSLBEN	**Reserved**	**BRKP1LEN**	**BRKP0LEN**	**Reserved**		**CPO1LBEN**	**CPO0LBEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SYSEBEN	**Reserved**	**BRKP1EEN**	**BRKP0EEN**	**Reserved**		**CPO1EBEN**	**CPO0EBEN**

位	描述	
[31:20]	**Reserved**	保留.
[19:18]	**BRKAODD**	**PWM 奇数通道刹车行为选择(写保护)** 00 = PWMx的刹车事件不影响奇数通道输出。 01 = 当PWMx的刹车事件发生，PWM奇数通道输出三态。 10 =当PWMx的刹车事件发生，PWM奇数通道输出低电平。 11 =当PWMx的刹车事件发生，PWM奇数通道输出高电平。 **注:** 这些位写保护，详情参考SYS_REGLCTL寄存器
[17:16]	**BRKAEVEN**	**PWM偶数通道刹车行为选择 (写保护)** 00 = PWMx的刹车事件不影响偶数通道输出。 01 = 当PWMx的刹车事件发生，PWM偶数通道输出三态。 10 =当PWMx的刹车事件发生，PWM偶数通道输出低电平。 11 =当PWMx的刹车事件发生，PWM偶数通道输出高电平。 **注:** 这些位写保护，详情参考SYS_REGLCTL寄存器
[15]	**SYSLBEN**	**使能系统故障作为电平检测刹车源 (写保护)** 0 =禁止系统故障条件作为电平检测刹车源 1 =使能系统故障条件作为电平检测刹车源 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[14]	**Reserved**	保留.
[13]	**BRKP1LEN**	**使能引脚BKP1作为电平检测刹车源(写保护)**

		0 =.禁止PWMx_BRAKE1引脚作为电平检测刹车源 1 =使能PWMx_BRAKE1引脚作为电平检测刹车源 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[12]	**BRKP0LEN**	**使能引脚BKP0作为电平检测刹车源(写保护)** 0 = 禁止PWMx_BRAKE0引脚作为电平检测刹车源 1 = 使能PWMx_BRAKE0引脚作为电平检测刹车源 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[11:10]	**Reserved**	保留.
[9]	**CPO1LBEN**	**使能 ACMP1_O 数字输出作为电平检测刹车源 (写保护)** 0 = ACMP1_O 作为电平检测刹车源禁止 1 = ACMP1_O 作为电平检测刹车源使能 **注:** 该寄存器写保护，详情参考SYS_REGLCTL寄存器
[8]	**CPO0LBEN**	**使能ACMP0_O数字输出作为电平检测刹车源(写保护)** 0 = ACMP0_O作为电平检测刹车源禁止. 1 = ACMP0_O作为电平检测刹车源使能 **注:** 该寄存器写保护，详情参考SYS_REGLCTL寄存器
[7]	**SYSEBEN**	**使能系统故障作为边沿检测刹车源 (写保护)** 0 =系统故障条件作为沿检测刹车源禁止. 1 = 系统故障条件作为沿检测刹车源使能. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[6]	**Reserved**	保留.
[5]	**BRKP1EEN**	**使能PWMx_BRAKE1 引脚作为边沿检测刹车源(写保护)** 0 = BKP1 1引脚作为边沿检测刹车源禁止. 1 = BKP1 1引脚作为边沿检测刹车源使能. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[4]	**BRKP0EEN**	**使能PWMx_BRAKE0引脚作为边沿检测刹车源 (写保护)** 0 = BKP0引脚作为边沿检测刹车源禁止. 1 = BKP0引脚作为边沿检测刹车源使能. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[3:2]	**Reserved**	保留.
[1]	**CPO1EBEN**	**使能ACMP1_O 数字输出作为边沿检测刹车源 (写保护)** 0 = ACMP1_O 作为边沿检测刹车源禁止 1 = ACMP1_O 作为边沿检测刹车源使能. **注:** 该寄存器写保护，详情参考SYS_REGLCTL寄存器
[0]	**CPO0EBEN**	**使能ACMP0_O数字输出作为边沿检测刹车源 (写保护)** 0 = ACMP0_O 作为边沿检测刹车源禁止. 1 = ACMP0_O 作为边沿检测刹车源使能 **注:** 该寄存器写保护，详情参考SYS_REGLCTL寄存器

PWM 引脚极性反转控制(PWM_POLCTL)

寄存器	偏移	R/W	描述	复位值
PWM_POLCTL	PWMx_BA+0xD4	R/W	PWM引脚极性反转寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PINV5	PINV4	PINV3	PINV2	PINV1	PINV0

位	描述	
[31:6]	**Reserved**	保留.
[n] n=0,1..5	**PINVn**	**PWM引脚极性反转控制** 该寄存器控制PWM输出的极性状态。 0 =禁止PWMx_CHn输出极性反转 1 =使能PWMx_CHn输出极性反转

PWM输出使能寄存器 (PWM_POEN)

寄存器	偏移	R/W	描述	复位值
PWM_POEN	PWMx_BA+0xD8	R/W	PWM输出使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**POEN5**	**POEN4**	**POEN3**	**POEN2**	**POEN1**	**POEN0**

位	描述	
[31:6]	**Reserved**	保留.
[n] n=0,1..5	**POENn**	**PWM引脚输出使能** 0 = PWM引脚在三态模式. 1 = PWM引脚在输出模式.

PWM软件刹车控制寄存器 (PWM_SWBRK)

寄存器	偏移	R/W	描述	复位值
PWM_SWBRK	PWMx_BA+0xDC	W	PWM软件刹车控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**BRKLTRG4**	**BRKLTRG2**	**BRKLTRG0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**BRKETRG4**	**BRKETRG2**	**BRKETRG0**

位	描述	
[31:11]	**Reserved**	保留.
[8+n/2] n=0,2,4	**BRKLTRGn**	**PWM电平刹车软件触发 (只写) (写保护)** 写1到该位将触发电平刹车，并将PWM_INTSTS1寄存器的BRKLIFn置1. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[7:3]	**Reserved**	保留.
[n/2] n=0,2,4	**BRKETRGn**	**PWM边沿刹车软件触发(只写) (写保护)** 写1到该位将触发边沿刹车，并将PWM_INTSTS1寄存器的BRKEIFn置1. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器

PWM中断使能寄存器0 (PWM_INTEN0)

寄存器	偏移	R/W	描述	复位值
PWM_INTEN0	PWMx_BA+0xE0	R/W	PWM中断使能寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
23	22	21	20	19	18	17	16
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
15	14	13	12	11	10	9	8
Reserved			PIEN4	Reserved	PIEN2	Reserved	PIEN0
7	6	5	4	3	2	1	0
Reserved			ZIEN4	Reserved	ZIEN2	Reserved	ZIEN0

位	描述	
[31:30]	**Reserved**	保留.
[24+n] n=0,1..5	**CMPDIENn**	**PWM比较向下计数中断使能** 0 =禁止比较向下计数中断 1 =使能比较向下计数中断 **注:** 在互补模式，CMPDIEN 1,3,5的设置是无效的，实际功能参照CMPDIEN 0,2,4的设置
[23:22]	**Reserved**	保留.
[16+n] n=0,1..5	**CMPUIENn**	**PWM比较向上计数中断使能** 每一位n控制对应的PWM通道n 0 =禁止比较向上计数中断 1 =使能比较向上计数中断 **注:** 在互补模式，CMPUIEN 1,3,5的设置是无效的，实际功能参照CMPUIEN 0,2,4的设置
[15:13]	**Reserved**	保留.
[12]	**PIEN4**	**PWM 周期点中断使能位4** 0 =禁止周期点中断 1 =使能周期点中断 **注:** :上下计数方式周期点指的是中心点。
[11]	**Reserved**	保留.
[10]	**PIEN2**	**PWM 周期点中断使能位2** 0 =禁止周期点中断 1 =使能周期点中断 **注:** :上下计数方式周期点指的是中心点。
[9]	**Reserved**	保留.

[8]	**PIEN0**	**PWM 周期点中断使能位0** 0 =禁止周期点中断 1 =使能周期点中断 **注:** :上下计数方式周期点指的是中心点。
[7:5]	**Reserved**	保留.
[4]	**ZIEN4**	**PWM 0点中断使能位 4** 0 = 0点中断禁用. 1 = 0点中断使能. **注:** 在互补模式奇数通道读数总是0.
[3]	**Reserved**	保留.
[2]	**ZIEN2**	**PWM 0点中断使能位 2** 0 = 0点中断禁用. 1 = 0点中断使能. **注:** 在互补模式奇数通道读数总是0.
[1]	**Reserved**	保留.
[0]	**ZIEN0**	**PWM 0点中断使能位 0** 0 = 0点中断禁用. 1 = 0点中断使能. **注:** 在互补模式奇数通道读数总是0.

PWM中断使能寄存器1 (PWM_INTEN1)

寄存器	偏移	R/W	描述	复位值
PWM_INTEN1	PWMx_BA+0xE4	R/W	PWM中断使能寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**BRKLIEN4_5**	**BRKLIEN2_3**	**BRKLIEN0_1**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**BRKEIEN4_5**	**BRKEIEN2_3**	**BRKEIEN0_1**

位	描述	
[31:11]	**Reserved**	保留.
[10]	**BRKLIEN4_5**	**PWM通道4/5电平检测刹车中断使能 (写保护)** 0 =禁止通道4/5电平检测刹车中断 1 =使能通道4/5电平检测刹车中断 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[9]	**BRKLIEN2_3**	**PWM通道2/3电平检测刹车中断使能 (写保护)** 0 =禁止通道2/3电平检测刹车中断 1 =使能通道2/3电平检测刹车中断 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[8]	**BRKLIEN0_1**	**PWM 通道0/1电平检测刹车中断使能 (写保护)** 0 =禁止通道0/1电平检测刹车中断 1 =使能通道0/1电平检测刹车中断 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[7:3]	**Reserved**	保留.
[2]	**BRKEIEN4_5**	**PWM通道4/5边沿检测刹车中断使能 (写保护)** 0 =禁止通道4/5边沿检测刹车中断 1 =使能通道4/5边沿检测刹车中断. **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[1]	**BRKEIEN2_3**	**PWM通道2/3边沿检测刹车中断使能(写保护)** 0 =禁止通道2/3边沿检测刹车中断 1 =使能通道2/3边沿检测刹车中断 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器

[0]	**BRKEIEN0_1**	**PWM通道0/1边沿检测刹车中断使能(写保护)** 0 = 禁止通道0/1边沿检测刹车中断 1 = 使能通道0/1边沿检测刹车中断 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器

PWM中断标志寄存器0 (PWM_INTSTS0)

寄存器	偏移	R/W	描述	复位值
PWM_INTSTS0	PWMx_BA+0xE8	R/W	PWM中断标志寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		**CMPDIF5**	**CMPDIF4**	**CMPDIF3**	**CMPDIF2**	**CMPDIF1**	**CMPDIF0**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**CMPUIF5**	**CMPUIF4**	**CMPUIF3**	**CMPUIF2**	**CMPUIF1**	**CMPUIF0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**PIF4**	**Reserved**	**PIF2**	**Reserved**	**PIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**ZIF4**	**Reserved**	**ZIF2**	**Reserved**	**ZIF0**

位	描述	
[31:30]	**Reserved**	保留.
[24+n] n=0,1..5	**CMPDIFn**	**PWM比较向下计数中断标志** 当PWM计数器向下计数到PWM_CMPDATn，标志被硬件置1。软件写1到该位将清标志。 **注:** 在互补模式，通道 0, 2, 4.的CMPDIF就是CMPDIF 1,3,5 的CMPDIF
[23:22]	**Reserved**	保留.
[21:16]	**CMPUIFn**	**PWM比较向上计数中断标志** 当PWM计数器向上计数到PWM_CMPDATn，标志被硬件置1。软件写1到该位将清标志. **注:** 在互补模式，通道 0, 2, 4.的CMPUIF就是CMPDIF 1,3,5 的CMPUIF
[15:13]	**Reserved**	保留.
[12]	**PIF4**	**PWM 周期点中断标志4** 当PWM_CH4计数到PWM_PERIOD4，该位被硬件置1. **注:** 该位写1清0
[11]	**Reserved**	保留.
[10]	**PIF2**	**PWM 周期点中断标志2** 当PWM_CH2计数到PWM_PERIOD2，该位被硬件置1. **注:** 该位写1清0
[9]	**Reserved**	保留.
[8]	**PIF0**	**PWM 周期点中断标志0** 当PWM_CH0计数到PWM_PERIOD0，该位被硬件置1. **注:** 该位写1清0
[7:5]	**Reserved**	保留.

[4]	**ZIF4**	**PWM零点中断标志4** 当PWM_CH4计数到0，该位被硬件置1. **注:** 该位写1清0
[3]	**Reserved**	保留.
[2]	**ZIF2**	**PWM零点中断标志2** 当PWM_CH2计数到0，该位被硬件置1. **注:** 该位写1清0
[1]	**Reserved**	保留.
[0]	**ZIF0**	**PWM零点中断标志0** 当PWM_CH0计数到0，该位被硬件置1.. **注:** 该位写1清0

PWM中断标志寄存器1 (PWM_INTSTS1)

寄存器	偏移	R/W	描述	复位值
PWM_INTSTS1	PWMx_BA+0xEC	R/W	PWM中断标志寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		BRKLSTS5	BRKLSTS4	BRKLSTS3	BRKLSTS2	BRKLSTS1	BRKLSTS0
23	22	21	20	19	18	17	16
Reserved		BRKESTS5	BRKESTS4	BRKESTS3	BRKESTS2	BRKESTS1	BRKESTS0
15	14	13	12	11	10	9	8
Reserved		BRKLIF5	BRKLIF4	BRKLIF3	BRKLIF2	BRKLIF1	BRKLIF0
7	6	5	4	3	2	1	0
Reserved		BRKEIF5	BRKEIF4	BRKEIF3	BRKEIF2	BRKEIF1	BRKEIF0

位	描述	
[31:30]	**Reserved**	保留.
[24+n] n=0,1..5	**BRKLSTSn**	**PWM通道n电平检测刹车状态 (只读)** 0 = PWM通道n电平检测刹车状态被释放。 1 =当PWM通道n电平检测到任何一个已被使能的刹车源有个下降沿时，该位被置1表示PWM通道N处在刹车状态。 **注:** 该位只读并被硬件自动清除。当已使能刹车源恢复到高电平，直到当前周期完成后PWM输出才会释放刹车状态。PWM波形将在下个完整的PWM周期开始输出。
[23:22]	**Reserved**	保留.
[16+n] n=0,1..5	**BRKESTSn**	**PWM通道n边沿检测刹车状态(只读)** 0 = PWM通道n边沿检测刹车状态释放 1 =当PWM通道n边沿刹车检测到任何被使能的刹车源有一个下降沿时，该位被置1表示通道N处在刹车状态. **注:** 该位只读并被硬件自动清除. 当边沿检测中断标志清除， PWM将释放刹车状态直到PWM周期结束。PWM波形将从下个PWM周期开始输出。T
[15:14]	**Reserved**	保留.
[8+n] n=0,1..5	**BRKLIFn**	**PWM 通道n电平检测刹车中断标志 (写保护)** 0 =PWM通道n电平检测刹车事件没发生 1 =当PWM通道n电平检测中断事件发生，该位被置1，该位写1清除。 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器
[7:6]	**Reserved**	保留.
[n] n=0,1..5	**BRKEIFn**	**PWM 通道n边沿检测刹车中断标志 (写保护)** 0 = PWM通道n边沿检测刹车事件没发生 1 =当PWM通道n边沿检测中断事件发生，该位被置1，该位写1清除。 **注:** 该位写保护，详情参考SYS_REGLCTL寄存器

PWM触发ADC源选择寄存器0 (PWM_ADCTS0)

寄存器	偏移	R/W	描述	复位值
PWM_ADCTS0	PWMx_BA+0xF8	R/W	PWM触发ADC源选择寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
TRGEN3	**Reserved**			**TRGSEL3**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRGEN2	**Reserved**			**TRGSEL2**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN1	**Reserved**			**TRGSEL1**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN0	**Reserved**			**TRGSEL0**			

位	描述	
[31]	**TRGEN3**	**PWM_CH3触发ADC使能位** 0 = PWM_CH3触发ADC禁止 1 = PWM_CH3触发ADC使能
[30:28]	**Reserved**	保留.
[27:24]	**TRGSEL3**	**PWM_CH3触发ADC源选择** 0000 = PWM_CH2零点. 0001 = PWM_CH2 周期点. 0010 = PWM_CH2 零或周期点. 0011 = PWM_CH2向上计数到CMPDAT点. 0100 = PWM_CH2向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH3向上计数到CMPDAT点. 1001 = PWM_CH3向下计数到CMPDAT点. 其它 = 保留.
[23]	**TRGEN2**	**PWM_CH2触发ADC使能位** 0 = PWM_CH2触发ADC禁止 1 = PWM_CH2触发ADC使能
[22:20]	**Reserved**	保留.
[19:16]	**TRGSEL2**	**PWM_CH2触发ADC源选择** 0000 = PWM_CH2零点. 0001 = PWM_CH2 周期点. 0010 = PWM_CH2 零或周期点.

		0011 = PWM_CH2向上计数到CMPDAT点. 0100 = PWM_CH2向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH3向上计数到CMPDAT点. 1001 = PWM_CH3向下计数到CMPDAT点. 其它 = 保留.
[15]	**TRGEN1**	**PWM_CH1触发ADC使能位** 0 = PWM_CH1触发ADC禁止 1 = PWM_CH1触发ADC使能
[14:12]	**Reserved**	保留.
[11:8]	**TRGSEL1**	**PWM_CH1触发ADC源选择** 0000 = PWM_CH0零点. 0001 = PWM_CH0 周期点. 0010 = PWM_CH0 零或周期点. 0011 = PWM_CH0向上计数到CMPDAT点. 0100 = PWM_CH0向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH1向上计数到CMPDAT点. 1001 = PWM_CH1向下计数到CMPDAT点. 其它 = 保留.
[7]	**TRGEN0**	**PWM_CH0触发ADC使能位** 0 = PWM_CH0触发ADC禁止 1 = PWM_CH0触发ADC使能
[6:4]	**Reserved**	保留.
[3:0]	**TRGSEL0**	**PWM_CH0触发ADC源选择** 0000 = PWM_CH0零点. 0001 = PWM_CH0 周期点. 0010 = PWM_CH0 零或周期点. 0011 = PWM_CH0向上计数到CMPDAT点. 0100 = PWM_CH0向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH1向上计数到CMPDAT点. 1001 = PWM_CH1向下计数到CMPDAT点. 其它 = 保留.

PWM触发ADC源选择寄存器1 (PWM_ADCTS1)

寄存器	偏移	R/W	描述	复位值
PWM_ADCTS1	PWMx_BA+0xFC	R/W	PWM触发ADC源选择寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN5	**Reserved**			**TRGSEL5**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN4	**Reserved**			**TRGSEL4**			

位	描述	
[31:16]	**Reserved**	保留.
[15]	**TRGEN5**	**PWM_CH5触发ADC使能位** 0 = PWM_CH5触发ADC禁止 1 = PWM_CH5触发ADC使能
[14:12]	**Reserved**	保留.
[11:8]	**TRGSEL5**	**PWM_CH5触发ADC源选择** 0000 = PWM_CH4零点. 0001 = PWM_CH4 周期点. 0010 = PWM_CH4 零或周期点. 0011 = PWM_CH4向上计数到CMPDAT点. 0100 = PWM_CH4向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH5向上计数到CMPDAT点. 1001 = PWM_CH5向下计数到CMPDAT点. 其它 = 保留.
[7]	**TRGEN4**	**PWM_CH4触发ADC使能位** 0 = PWM_CH4触发ADC禁止 1 = PWM_CH4触发ADC使能
[6:4]	**Reserved**	保留.
[3:0]	**TRGSEL4**	**PWM_CH4触发ADC源选择** 0000 = PWM_CH4零点.

		0001 = PWM_CH4 周期点. 0010 = PWM_CH4 零或周期点. 0011 = PWM_CH4向上计数到CMPDAT点. 0100 = PWM_CH4向下计数到CMPDAT点. 0101 = 保留. 0110 = 保留. 0111 = 保留. 1000 = PWM_CH5向上计数到CMPDAT点. 1001 = PWM_CH5向下计数到CMPDAT点. 其它 = 保留.

PWM同步开始控制寄存器 (PWM_SSCTL)

寄存器	偏移	R/W	描述	复位值
PWM_SSCTL	PWMx_BA+0x110	R/W	PWM同步开始控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**SSRC**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**SSEN4**	**Reserved**	**SSEN2**	**Reserved**	**SSEN0**

位	描述	
[31:10]	**Reserved**	保留.
[9:8]	**SSRC**	**PWM同步开始源选择位** 00 = 同步开始源来自PWM0. 01 =同步开始源来自PWM1. 10 = 保留. 11 = 保留.
[7:5]	**Reserved**	保留.
[4]	**SSEN4**	**PWM 同步开始功能使能位4** 当同步开始功能使能，PWM_CH4计数器使能位(CNTEN4)可以通过写PWM同步开始触发位(CNTSEN)来使能。 0 = PWM 同步开始功能禁止. 1 = PWM 同步开始功能使能.
[3]	**Reserved**	保留.
[2]	**SSEN2**	**PWM 同步开始功能使能位2** 当同步开始功能使能，PWM_CH2计数器使能位(CNTEN2)可以通过写PWM同步开始触发位(CNTSEN)来使能。 0 = PWM 同步开始功能禁止. 1 = PWM 同步开始功能使能.
[1]	**Reserved**	保留.
[0]	**SSEN0**	**PWM 同步开始功能使能位0** 当同步开始功能使能，PWM_CH0计数器使能位(CNTEN0)可以通过写PWM同步开始触发位(CNTSEN)来使能。 0 = PWM 同步开始功能禁止.

		1 = PWM 同步开始功能使能.

PWM同步开始触发寄存器 (PWM_SSTRG)

寄存器	偏移	R/W	描述	复位值
PWM_SSTRG	PWMx_BA+0x114	W	PWM同步开始触发寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留.
[0]	**CNTSEN**	**PWM计数器同步开始使能(只写)** PWM计数器同步使能功能用于使所选择的PWM通道 (包括 PWM0_CHx 和 PWM1_CHx)同时开始计数 。 如果相关PWM通道计数器同步开始功能被使能，该位置1也会置位计数器使能位(CNTENn, n 代表通道0 到 5)。

PWM状态寄存器 (PWM_STATUS)

寄存器	偏移	R/W	描述	复位值
PWM_STATUS	PWMx_BA+0x120	R/W	PWM状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**ADCTRG5**	**ADCTRG4**	**ADCTRG3**	**ADCTRG2**	**ADCTRG1**	**ADCTRG0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CNTMAX4**		**Reserved**	**CNTMAX2**	**Reserved**	**CNTMAX0**

位	描述	
[31:22]	**Reserved**	保留.
[16+n] n=0,1..5	**ADCTRGn**	**ADC开始转换状态** 0 =表示没有ADC开始转换触发事件发生. 1 =表示有ADC开始转换触发事件发生. **注:** 该位软件写1清0.
[15:5]	**Reserved**	保留.
[4]	**CNTMAX4**	**时基计数器4等于0xFFFF 锁存标志** 0 = 表示时基计数器 从未达到最大值 0xFFFF. 1 = 表示时基计数器达到其最大值。 **注:** 该位软件写1清0.
[3]	**Reserved**	保留.
[2]	**CNTMAX2**	**时基计数器2等于0xFFFF 锁存标志** 0 = 表示时基计数器 从未达到最大值 0xFFFF. 1 = 表示时基计数器达到其最大值。 **注:** 该位软件写1清0.
[1]	**Reserved**	保留.
[0]	**CNTMAX0**	**时基计数器0等于0xFFFF 锁存标志** 0 = 表示时基计数器 从未达到最大值 0xFFFF. 1 = 表示时基计数器达到其最大值。 **注:** 该位软件写1清0.

PWM捕获输入使能寄存器 (PWM_CAPINEN)

寄存器	偏移	R/W	描述	复位值
PWM_CAPINEN	PWMx_BA+0x200	R/W	PWM捕获输入使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CAPINEN5**		**CAPINEN4**	**CAPINEN3**	**CAPINEN2**	**CAPINEN1**	**CAPINEN0**

位	描述	
[31:6]	**Reserved**	保留.
[n] n=0,1..5	**CAPINENn**	**捕获输入使能** 0 =禁止PWM输入捕获，PWM通道捕获功能的输入总是被认为是0 1 =使能PWM输入捕获，PWM通道捕获功能的输入来自相关的复用引脚

PWM捕获控制寄存器 (PWM_CAPCTL)

寄存器	偏移	R/W	描述	复位值
PWM_CAPCTL	PWMx_BA+0x204	R/W	PWM捕获控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FCRLDEN5	FCRLDEN4	FCRLDEN3	FCRLDEN2	FCRLDEN1	FCRLDEN0
23	22	21	20	19	18	17	16
Reserved		RCRLDEN5	RCRLDEN4	RCRLDEN3	RCRLDEN2	RCRLDEN1	RCRLDEN0
15	14	13	12	11	10	9	8
Reserved		CAPINV5	CAPINV4	CAPINV3	CAPINV2	CAPINV1	CAPINV0
7	6	5	4	3	2	1	0
Reserved	CAPEN5		CAPEN4	CAPEN3	CAPEN2	CAPEN1	CAPEN0

位	描述	
[31:30]	**Reserved**	保留.
[24+n] n=0,1..5	**FCRLDENn**	**下降沿捕获重载使能** 0 =禁止下降沿捕获重载计数器 1 =使能下降沿捕获重载计数器
[23:22]	**Reserved**	保留.
[16+n] n=0,1..5	**RCRLDENn**	**上升沿捕获重载使能** 0 =禁止上升沿捕获重载计数器 1 =使能上升沿捕获重载计数器
[15:14]	**Reserved**	保留.
[8+n] n=0,1..5	**CAPINVn**	**捕获反向使能** 0 =禁止捕获源反向 1 =使能捕获源反向，将来自GPIO的信号反向
[7:6]	**Reserved**	保留.
[n] n=0,1..5	**CAPENn**	**捕获功能使能** 0 =禁止捕获功能。RCAPDAT/FCAPDAT不会更新 1= 使能捕获功能。当检测到输入信号的上升/下降沿时锁存PWM计数器值并保存到RCAPDAT（上升沿锁存）和FCAPDAT（下降沿锁存）

PWM捕获状态寄存器 (PWM_CAPSTS)

寄存器	偏移	R/W	描述	复位值
PWM_CAPSTS	PWMx_BA+0x208	R	PWM捕获状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CFLIFOV5**	**CFLIFOV4**	**CFLIFOV3**	**CFLIFOV2**	**CFLIFOV1**	**CFLIFOV0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CRLIFOV5**	**CRLIFOV4**	**CRLIFOV3**	**CRLIFOV2**	**CRLIFOV1**	**CRLIFOV0**

位	描述	
[31:14]	**Reserved**	保留.
[8+n] n=0,1..5	**CFLIFOVn**	**下降沿捕获锁存中断标志溢出状态 (只读)** 如果相应的CFLIF是1，且下降沿锁存事件发生，此标志为1。. **注:** 当用户清相应的CFLIF，该位将自动清零.
[7:6]	**Reserved**	保留.
[n] n=0,1..5	**CRLIFOVn**	**上升沿捕获锁存中断标志溢出状态(只读)** 如果相应的CRLIF是1，且上升沿捕获事件发生，此标志为1。 **注:** 当用户清相应的CRLIF，该位将自动清零.

PWM上升沿捕获数据寄存器0~5 (PWM_RCAPDAT 0~5)

寄存器	偏移	R/W	描述	复位值
PWM_RCAPDAT0	PWMx_BA+0x20C	R	PWM上升沿捕获数据寄存器0	0x0000_0000
PWM_RCAPDAT1	PWMx_BA+0x214	R	PWM上升沿捕获数据寄存器1	0x0000_0000
PWM_RCAPDAT2	PWMx_BA+0x21C	R	PWM上升沿捕获数据寄存器2	0x0000_0000
PWM_RCAPDAT3	PWMx_BA+0x224	R	PWM上升沿捕获数据寄存器3	0x0000_0000
PWM_RCAPDAT4	PWMx_BA+0x22C	R	PWM上升沿捕获数据寄存器4	0x0000_0000
PWM_RCAPDAT5	PWMx_BA+0x234	R	PWM上升沿捕获数据寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**RCAPDAT**	**PWM上升沿捕获数据寄存器(只读)** 当上升沿捕获条件发生，PWM计数器值将被保存到该寄存器。

PWM下降沿捕获数据寄存器0~5 (PWM_FCAPDAT 0~5)

寄存器	偏移	R/W	描述	复位值
PWM_FCAPDAT0	PWMx_BA+0x210	R	PWM下降沿捕获数据寄存器0	0x0000_0000
PWM_FCAPDAT1	PWMx_BA+0x218	R	PWM下降沿捕获数据寄存器1	0x0000_0000
PWM_FCAPDAT2	PWMx_BA+0x220	R	PWM下降沿捕获数据寄存器2	0x0000_0000
PWM_FCAPDAT3	PWMx_BA+0x228	R	PWM下降沿捕获数据寄存器3	0x0000_0000
PWM_FCAPDAT4	PWMx_BA+0x230	R	PWM下降沿捕获数据寄存器4	0x0000_0000
PWM_FCAPDAT5	PWMx_BA+0x238	R	PWM下降沿捕获数据寄存器5	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**FCAPDAT**	**PWM下降沿捕获数据寄存器(只读)** 当下降沿捕获条件发生，PWM计数器值将被保存到该寄存器。

PWM PDMA控制寄存器 (PWM_PDMACTL)

寄存器	偏移	R/W	描述	复位值
PWM_PDMACTL	PWMx_BA+0x23C	R/W	PWM PDMA控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**CHSEL4_5**	**CAPORD4_5**	**CAPMOD4_5**		**CHEN4_5**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**CHSEL2_3**	**CAPORD2_3**	**CAPMOD2_3**		**CHEN2_3**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**CHSEL0_1**	**CAPORD0_1**	**CAPMOD0_1**		**CHEN0_1**

位	描述	
[31:21]	**Reserved**	保留.
[20]	**CHSEL4_5**	**选择通道 4/5 用于PDMA传输** 0 = 通道4. 1 = 通道5. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[19]	**CAPORD4_5**	**捕获通道 4/5 上升沿/下降沿 顺序** 设置该位用于确定，当CAPMOD4_5 =11时，是PWM_RCAPDAT4/5 还是PWM_FCAPDAT4/5作为第一次捕获的数据通过PDMA开始搬移到内存。 0 = PWM_FCAPDAT4/5是先捕获并搬移到内存的数据 1 = PWM_RCAPDAT4/5是先捕获并搬移到内存的数据. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[18:17]	**CAPMOD4_5**	**选择 PWM_RCAPDAT4/5 或 PWM_FCAPDAT4/5 来做PDMA 数据搬移** 00 = 保留. 01 = PWM_RCAPDAT4/5. 10 = PWM_FCAPDAT4/5. 11 = PWM_RCAPDAT4/5 和 PWM_FCAPDAT4/5. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[16]	**CHEN4_5**	**通道4/5 PDMA 使能** 0 = 通道 4/5 PDMA功能禁止 1 =通道4/5 PDMA 功能使能，用于通道4/5 捕获数据并搬移到内存. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[15:13]	**Reserved**	保留.

[12]	**CHSEL2_3**	**选择 通道 2/3用于PDMA传输** 0 =通道2. 1 =通道3. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[11]	**CAPORD2_3**	**获通道 2/3 上升沿/下降沿 顺序** 设置该位用于确定，当CAPMOD2_3 =11时，是PWM_RCAPDAT2/3 还是PWM_FCAPDAT2/3作为第一次捕获的数据通过PDMA开始搬移到内存。 0 = PWM_FCAPDAT2/3是先捕获并搬移到内存的数据 1 = PWM_RCAPDAT2/3是先捕获并搬移到内存的数据 **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[10:9]	**CAPMOD2_3**	**选择 PWM_RCAPDAT2/3 或 PWM_FCAODAT2/3来做PDMA 数据搬移** 00 = 保留. 01 = PWM_RCAPDAT2/3. 10 = PWM_FCAPDAT2/3. 11 = PWM_RCAPDAT2/3 和 PWM_FCAPDAT2/3. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[8]	**CHEN2_3**	**通道2/3 PDMA 使能** 0 = 通道 2/3 PDMA功能禁止 1 = 通道2/3功能使能，用于通道2/3捕获数据并搬移到内存 **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[7:5]	**Reserved**	保留.
[4]	**CHSEL0_1**	**选择 通道0/1来做PDMA 数据搬移** 0 =通道0. 1 =通道1. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[3]	**CAPORD0_1**	**获通道0/1上升沿/下降沿 顺序** 设置该位用于确定，当CAPMOD0_1 =11时，是PWM_RCAPDAT0/1还是PWM_FCAPDAT0/1作为第一次捕获的数据通过PDMA开始搬移到内存。 0 = PWM_FCAPDAT0/1是先捕获并搬移到内存的数据 1 = PWM_RCAPDAT0/1是先捕获并搬移到内存的数据. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[2:1]	**CAPMOD0_1**	**选择 PWM_RCAPDAT0/1 或 PWM_FCAPDAT0/1来做PDMA 数据搬移** 0 = 保留. 01 = PWM_RCAPDAT0/1. 10 = PWM_FCAPDAT0/1. 11 = PWM_RCAPDAT0/1 和 PWM_FCAPDAT0/1. **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节
[0]	**CHEN0_1**	**通道0/1 PDMA 使能**

		0 = 通道0/1 PDMA功能禁止. 1 = 通道0/1 PDMA功能使能，用于通道0/1捕获数据并搬移到内存 **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节

PWM 捕获通道 0_1, 2_3, 4_5 PDMA 寄存器 (PWM_PDMACAP 0_1, 2_3, 4_5)

寄存器	偏移	R/W	描述	复位值
PWM_PDMACAP0_1	PWMx_BA+0x240	R	PWM捕获通道01 PDMA寄存器	0x0000_0000
PWM_PDMACAP2_3	PWMx_BA+0x244	R	PWM捕获通道23 PDMA寄存器	0x0000_0000
PWM_PDMACAP4_5	PWMx_BA+0x248	R	PWM捕获通道45 PDMA寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**CAPBUF**	**PWM捕获 PDMA 寄存器(只读)** 该寄存器用于作为一个 缓存，用于通过PDMA.来传输PWM 捕获到的上升沿或下降沿数据到内存 **注:** 如果PDMA功能不支持，该位无效，详情请参考NuMicro® M031/M032系类选型指南的3.2章节

PWM 捕获中断使能寄存器(PWM_CAPIEN)

寄存器	偏移	R/W	描述	复位值
PWM_CAPIEN	PWMx_BA+0x250	R/W	PWM捕获中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CAPFIEN5**	**CAPFIEN4**	**CAPFIEN3**	**CAPFIEN2**	**CAPFIEN1**	**CAPFIEN0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CAPRIEN5**	**CAPRIEN4**	**CAPRIEN3**	**CAPRIEN2**	**CAPRIEN1**	**CAPRIEN0**

位	描述	
[31:14]	**Reserved**	保留.
[8+n] n=0,1..5	**CAPFIENn**	**PWM下降沿捕获锁存中断使能** 0 =.禁止下降沿锁存中断 1 = 使能下降沿锁存中断
[7:6]	**Reserved**	保留.
[n] n=0,1..5	**CAPRIENn**	**PWM上升沿捕获锁存中断使能** 0 =禁止上升沿锁存中断 1 =使能上升沿锁存中断

PWM捕获中断标志寄存器 (PWM_CAPIF)

寄存器	偏移	R/W	描述	复位值
PWM_CAPIF	PWMx_BA+0x254	R/W	PWM捕获中断标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**CFLIF5**	**CFLIF4**	**CFLIF3**	**CFLIF2**	**CFLIF1**	**CFLIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CRLIF5**	**CRLIF4**	**CRLIF3**	**CRLIF2**	**CRLIF1**	**CRLIF0**

位	描述	
[31:14]	**Reserved**	保留.
[8+n] n=0,1..5	**CFLIFn**	**PWM下降沿捕获锁存中断标志** 0 =没有下降沿捕获锁存条件发生。 1 =下降沿捕获锁存条件发生，该标志被置高。 **注1:** 当带PDMA运行捕获功能， PDMA 搬移完数据后，CAPIF对应通道的CFLIF位将被硬件清除。 **注2:** 该位写1清0
[7:6]	**Reserved**	保留.
[n] n=0,1..5	**CRLIFn**	**PWM上升沿捕获锁存中断标志** 0 =没有上升沿捕获锁存条件发生。 1 =上升沿捕获锁存条件发生，该标志被置高。 **注1:** 当带PDMA运行捕获功能，PDMA 搬移完数据后，CAPIF对应通道的CFLIF位将被硬件清除。 **注2:** 该位写1清0

PWM周期寄存器缓存0, 2, 4 (PWM_PBUF0, 2, 4)

寄存器	偏移	R/W	描述	复位值
PWM_PBUF0	PWMx_BA+0x304	R	PWM PERIOD0 缓存	0x0000_0000
PWM_PBUF2	PWMx_BA+0x30C	R	PWM PERIOD2 缓存	0x0000_0000
PWM_PBUF4	PWMx_BA+0x314	R	PWM PERIOD4 缓存	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**PBUF**	**PWM 周期寄存器缓存 (只读)** 用作PERIOD有效寄存器.

PWM比较器寄存器缓存0~5 (PWM_CMPBUF0~5)

寄存器	偏移	R/W	描述	复位值
PWM_CMPBUF0	PWMx_BA+0x31C	R	PWM CMPDAT0 缓存	0x0000_0000
PWM_CMPBUF1	PWMx_BA+0x320	R	PWM CMPDAT1 缓存	0x0000_0000
PWM_CMPBUF2	PWMx_BA+0x324	R	PWM CMPDAT2 缓存	0x0000_0000
PWM_CMPBUF3	PWMx_BA+0x328	R	PWM CMPDAT3 缓存	0x0000_0000
PWM_CMPBUF4	PWMx_BA+0x32C	R	PWM CMPDAT4 缓存	0x0000_0000
PWM_CMPBUF5	PWMx_BA+0x330	R	PWM CMPDAT5 缓存	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留.
[15:0]	**CMPBUF**	**PWM比较器寄存器缓存 (只读)** 用作 CMP 有效寄存器.

6.13 UART接口控制器 (UART)

6.13.1 概述

该芯片提供8路通用异步收发器(UART)。UART控制器作为标准速度的UART还支持流控功能。UART控制器的接收过程是把外设的串行数据转为并行数据，发送过程是把CPU的并行数据转成串行数据发送出去。每个UART通道支持10种类型的中断。UART控制器还支持IrDA，SIR，RS-485，单线功能模式和波特率自动测量功能。

6.13.2 特征

- 全双工，异步通讯
- 独立的接收/发送FIFO（16/16字节或1/1字节），用于数据装载
- 支持硬件自动流控制
- 接收缓存触发等级的数据长度可设
- 每个通道波特率可单独设置
- 支持nCTS、输入数据、接收数据FIFO达到阈值以及RS-485地址匹配(AAD模式)这几种情况下的唤醒功能，(仅UART0/UART1/UART4/UART5支持接收数据FIFO达到阈值和RS-485地址匹配(AAD模式)的唤醒功能)
- 支持8位接收缓存超时溢出检测功能
- 通过设置寄存器DLY(UART_TOUT[15:8])，可配置两个数据之间（从上一个stop位到下一个start位）的传送时间间隔
- 支持自动波特率测量和波特率补偿功能
  - 支持9600 bps下，UART_CLK可选择LXT(仅UART0/UART1/UART4/UART5支持)
- 支持break error、frame error、parity error和收/发缓冲区溢出检测等功能
- 可编程串行接口特性
  - 数据位长度可设为5~8位
  - 可编程校验，包括奇、偶、无校验位或固定校验位生成和检测
  - 可设置停止位长度为1位、1.5位或2位
- 支持IrDA SIR功能模式
  - 标准模式下支持3/16位宽功能
- 支持RS-485模式
  - 支持RS-485 9位模式
  - 支持软硬件控制nRTS管脚，用于控制RS-485传送方向
- 支持PDMA传输功能
- 支持单线功能模式

UART 特性	UART0/UART1/UART4/UART5	UART2/UART3/UART6/UART7	USCI-UART
FIFO	16 字节	1 字节	TX: 1字节 RX: 2字节

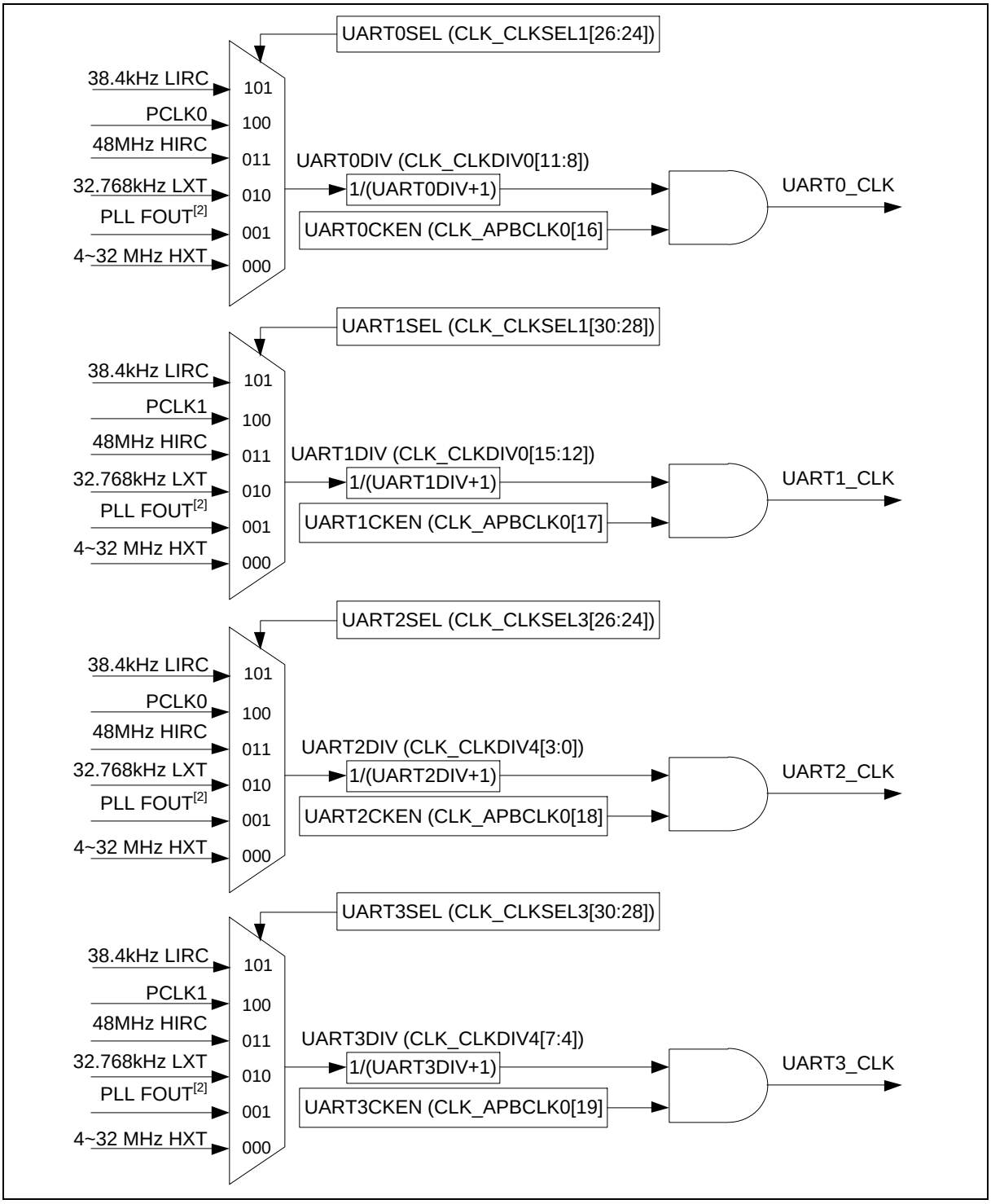
自动流控制(CTS/RTS)	√	√	√
IrDA	√	√	-
LIN	-	-	-
RS-485 功能模式	√	√	√
nCTS 唤醒	√	√	√
接收数据唤醒	√	√	√
接收数据达到设定阈值唤醒	√	-	-
RS-485地址匹配(AAD模式)唤醒	√	-	-
波特率自动测量	√	√	√
停止位长度	1, 1.5, 2 位	1, 1.5, 2 位	1, 2 位
字长	5, 6, 7, 8 位	5, 6, 7, 8 位	6~13 位
偶/奇校验	√	√	√
固定位	√	√	-
注: √=支持			

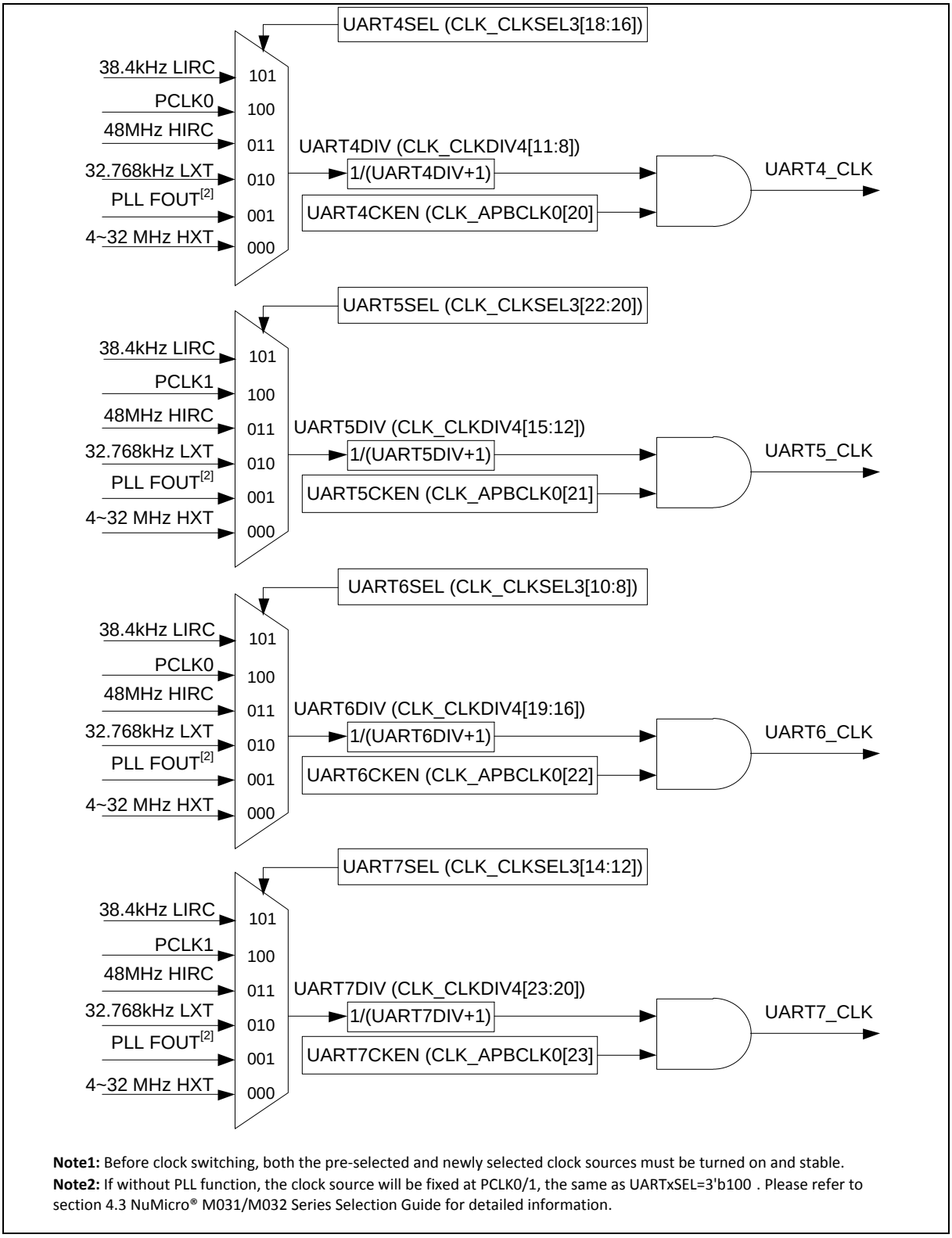
表 6.13-1 NuMicro® M031/M032 系列 UART 特征

6.13.3 框图

UART时钟控制和模块框图分别如图 6.13-1 和图 6.13-2所示 。

注:UARTx_CLK的频率不能大于30倍的HCLK。


UART0SEL (CLK_CLKSEL1[26:24])
38.4kHz LIRC
PCLK0
48MHz HIRC
32.768kHz LXT
PLL FOUT[2]
4~32 MHz HXT
101
100
011
010
001
000
UART0DIV (CLK_CLKDIV0[11:8])
1/(UART0DIV+1)
UART0CKEN (CLK_APBCLK0[16]
UART0_CLK
UART1SEL (CLK_CLKSEL1[30:28])
38.4kHz LIRC
PCLK1
48MHz HIRC
32.768kHz LXT
PLL FOUT[2]
4~32 MHz HXT
101
100
011
010
001
000
UART1DIV (CLK_CLKDIV0[15:12])
1/(UART1DIV+1)
UART1CKEN (CLK_APBCLK0[17]
UART1_CLK
UART2SEL (CLK_CLKSEL3[26:24])
38.4kHz LIRC
PCLK0
48MHz HIRC
32.768kHz LXT
PLL FOUT[2]
4~32 MHz HXT
101
100
011
010
001
000
UART2DIV (CLK_CLKDIV4[3:0])
1/(UART2DIV+1)
UART2CKEN (CLK_APBCLK0[18]
UART2_CLK
UART3SEL (CLK_CLKSEL3[30:28])
38.4kHz LIRC
PCLK1
48MHz HIRC
32.768kHz LXT
PLL FOUT[2]
4~32 MHz HXT
101
100
011
010
001
000
UART3DIV (CLK_CLKDIV4[7:4])
1/(UART3DIV+1)
UART3CKEN (CLK_APBCLK0[19]
UART3_CLK



Note1: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.
Note2: If without PLL function, the clock source will be fixed at PCLK0/1, the same as UARTxSEL=3'b100 . Please refer to section 4.3 NuMicro® M031/M032 Series Selection Guide for detailed information.

图 6.13-1 UART 时钟控制框图

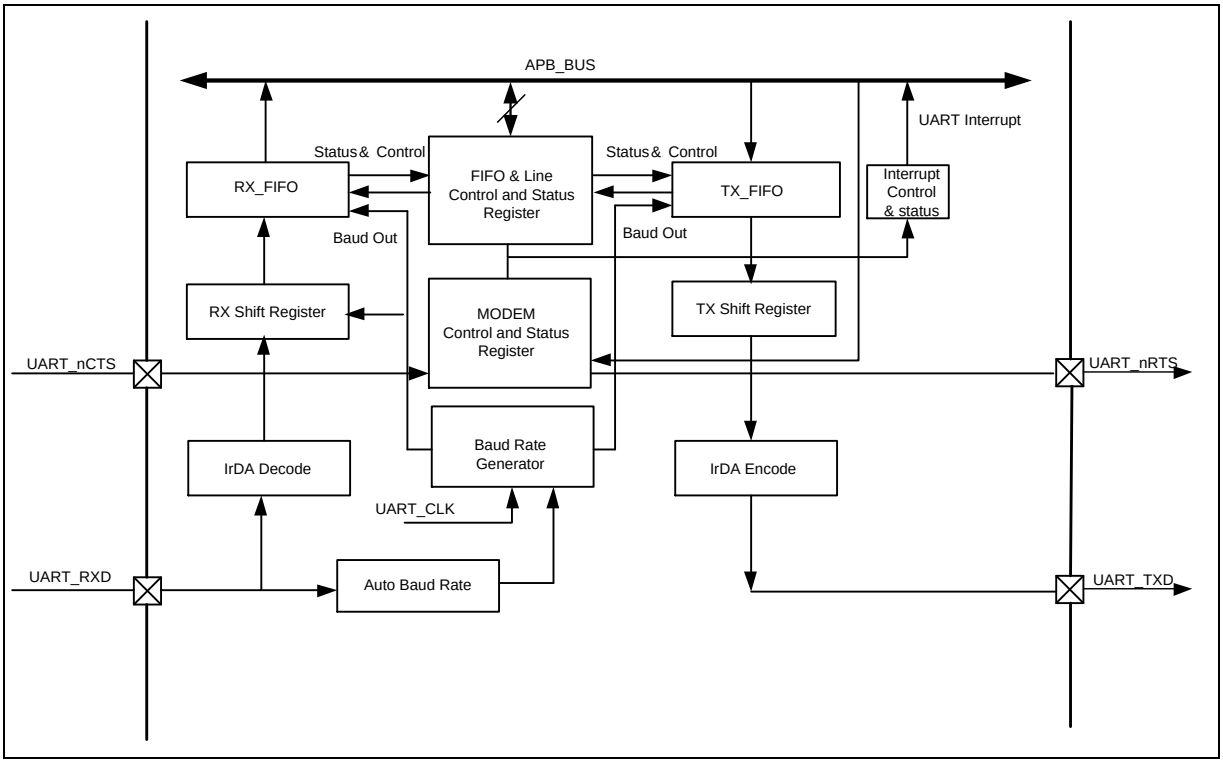


图 6.13-2 UART 模块框图

每个模块功能详细描述如下：

TX_FIFO

发送口带有一个16字节的FIFO缓冲区以减少CPU中断的频率。

RX_FIFO

接收口带有一个16字节的FIFO缓冲区（另加3个出错位，即BIF(UART_FIFOSTS[6])、FEF(UART_FIFOSTS[5])、PEF(UART_FIFOSTS[4])）以减少CPU中断的频率。

TX 移位寄存器

该模块用于控制把并行数据串行输出。

RX 移位寄存器

该模块用于控制把串行数据并行输入。

Modem 控制和状态寄存器

该寄存器用于控制到Modem或数传机（或类似于Modem的外设）的接口。

波特率发生器

该模块通过把输入的外部时钟除频得到期望的波特率。详情请参考波特率公式。

IrDA 编码

该模块为IrDA编码控制模块。

IrDA 解码

该模块为IrDA解码控制模块。

FIFO&Line 控制和状态寄存器

该寄存器组包括用于传输和接收的FIFO控制寄存器(UART_FIFO)、FIFO状态寄存器(UART_FIFOSTS)

和Line控制寄存器(UART_LINE)。定时溢出寄存器(UART_TOUT)配置时间溢出中断条件。

波特率自动检测

该模块用于自动检测波特率。

中断控制和状态寄存器

该模块一共有10种类型的中断，包括接收数据有效中断(RDAINT)、发送保持寄存器为空中断(THERINT)、发送器为空中断(TXENDINT)、接收Line状态中断(parity error 或 framing error 或 break error) (RLSINT)、MODEM状态中断(MODEMINT)、接收缓存超时中断(RXTOINT)、缓存错误中断(BUFERRINT)、唤醒中断(WKINT)、自动波特率中断(ABRINT)和单线模式位错误检测中断。中断使能寄存器(UART_INTEN)用于使能或禁止相应中断，中断状态寄存器(UART_INTSTS)显示哪些中断正在发生

中断	描述
RDAINT	接收数据有效中断。
THERINT	发送保持寄存器为空中断。
TXENDINT	发送器为空中断。
RLSINT	接收Line状态中断(parity error 或 framing error 或 break error)。
MODEMINT	MODEM状态中断。
RXTOINT	接收缓存超时中断。
BUFERRINT	缓存错误中断。
WKINT	唤醒中断。
ABRINT	自动波特率中断。
SWBEINT	单线模式位错误检测中断。

表 6.13-2 UART 中断

6.13.4 基本配置

UART0基本配置如下：

- 时钟源配置
  - UART0外设时钟源通过UART0SEL(CLK_CLKSEL1[26:24])设置
  - UART0控制器时钟预分频通过UART0DIV(CLK_CLKDIV0[11:8])来设置.
  - 使能UART0外设时钟通过UART0CKEN(CLK_APBCLK0[16]).
- 复位UART0控制器通过UART0RST(SYS_IPRST1[16]).

UART1基本配置如下：

- 时钟源配置
  - UART1外设时钟源通过UART1SEL(CLK_CLKSEL1[30:28])来设置.
  - UART1控制器时钟预分频通过UART1DIV(CLK_CLKDIV0[15:12])来设置.
  - 使能UART1外设时钟通过UART1CKEN(CLK_APBCLK0[17]).

- 复位UART1控制器通过UART1RST(SYS_IPRST1[17]).

UART2基本配置如下：

- 时钟源配置
  - UART2外设时钟源通过UART2SEL(CLK_CLKSEL3[26:24])来设置
  - UART2控制器时钟预分频通过UART2DIV(CLK_CLKDIV4[3:0])来设置
  - 使能UART2外设时钟通过UART2CKEN(CLK_APBCLK0[18]) .
- 复位UART2控制器通过UART2RST(SYS_IPRST1[18]).

UART3基本配置如下：

- 时钟源配置
  - UART3外设时钟源通过UART3SEL(CLK_CLKSEL3[30:28])来设置
  - UART3控制器时钟预分频通过UART3DIV(CLK_CLKDIV4[7:4])来设置
  - 使能UART3外设时钟通过UART3CKEN(CLK_APBCLK0[19]).
- 复位UART3控制器通过UART3RST(SYS_IPRST1[19]).

UART4基本配置如下：

- 时钟源配置
  - UART4外设时钟源通过UART4SEL(CLK_CLKSEL3[18:16])来设置
  - UART4控制器时钟预分频通过UART4DIV(CLK_CLKDIV4[11:8])来设置
  - 使能UART4外设时钟通过UART4CKEN(CLK_APBCLK0[20]).
- 复位UART4控制器通过UART4RST(SYS_IPRST1[20]).

UART5基本配置如下：

- 时钟源配置
  - UART5外设时钟源通过UART5SEL(CLK_CLKSEL3[22:20])来设置
  - UART5控制器时钟预分频通过UART5DIV(CLK_CLKDIV4[15:12])来设置
  - 使能UART5外设时钟通过UART5CKEN(CLK_APBCLK0[21]).
- 复位UART5控制器通过UART5RST(SYS_IPRST1[21]).

UART6基本配置如下：

- 时钟源配置
  - UART6外设时钟源通过UART6SEL(CLK_CLKSEL3[10:8])来设置
  - UART6控制器时钟预分频通过UART6DIV(CLK_CLKDIV4[19:16])来设置
  - 使能UART6外设时钟通过UART6CKEN(CLK_APBCLK0[22]).

- 复位UART6控制器通过UART6RST(SYS_IPRST1[22]).

UART7基本配置如下：

- 时钟源配置
  - UART7外设时钟源通过UART7SEL(CLK_CLKSEL3[14:12])来设置
  - UART7控制器时钟预分频通过UART7DIV(CLK_CLKDIV4[23:20])来设置
  - 使能UART7外设时钟通过UART7CKEN(CLK_APBCLK0[23]).
- 复位UART7控制器通过UART7RST(SYS_IPRST1[23]).

UART 接口控制器引脚描述如表 6.13-3:

引脚	类型	描述
UARTx_TXD	输出	UARTx发送
UARTx_RXD	输入	UARTx接收
UARTx_nCTS	输入	UARTx modem清零发送
UARTx_nRTS	输出	UARTx modem请求发送

表 6.13-3 UART 接口控制器引脚

6.13.5 功能描述

UART控制器支持三种功能模式，即UART、IrDA和RS-485模式。用户可以通过对UART_FUNCSEL设置选择功能。三种功能模式将在下面的章节中描述。

6.13.5.1 UART控制器波特率发生器

UART控制器包含一个可编程波特率发生器，其通过分频器对输入时钟源分频而得到收发数据所需的串行时钟。表 6.13-4列出UART波特率在各种条件下的公式，表 6.13-5和表 6.13-6列出UART波特率参数和寄存器设置。IrDA功能模式时，波特率发生器必须是模式0。寄存器UART_BAUD一栏中将更详细地介绍寄存器内容。波特率发生器共有三种设定模式：设定UART_BAUD[29:28]=00为模式0，设定UART_BAUD[29:28]=10为模式1，设定UART_BAUD[29:28]=11为模式2。

模式	BAUDM1	BAUDM0	波特率公式
模式0	0	0	UART_CLK/[16*(BRD+2)].
模式1	1	0	UART_CLK/[(EDIVM1+1)*(BRD+2)],EDIVM1必须>=8.
模式2	1	1	UART_CLK/(BRD+2) 如果UART_CLK<=3*HCLK,BRD必须>=8. 如果UART_CLK>3*HCLK,BRD必须>=3*N–1. N为大于或等于UART_CLK /HCLK的最小整数 例如, 如果3*HCLK<UART_CLK=<4*HCLK,BRD必须>=11. 如果4*HCLK<UART_CLK=<5*HCLK,BRD必须>=14. (如果UART_CLK来源于LXT,BRD大于或等于1)

表 6.13-4 UART 控制器波特率计算公式

UART外设时钟=12MHz			
波特率	模式0	模式1	模式2
921600	不支持	不推荐	BRD=11
460800	不推荐	BRD=0,EDIVM1=13	BRD=24
230400	不推荐	BRD=2,EDIVM1=13	BRD=50
115200	不推荐	BRD=6,EDIVM1=13	BRD=102
57600	BRD=11	BRD=14,EDIVM1=13	BRD=206
38400	BRD=18	BRD=22,EDIVM1=13	BRD=311
19200	BRD=37	BRD=123,EDIVM1=5	BRD=623
9600	BRD=76	BRD=123,EDIVM1=10	BRD=1248
4800	BRD=154	BRD=248,EDIVM1=10	BRD=2498

表 6.13-5 UART 控制器波特率参数设置示范表

UART外设时钟=12MHz			
波特率	UART_BAUD值		
	模式0	模式1	模式2
921600	不支持	不推荐	0x3000_000B
460800	不推荐	0x2D00_0000	0x3000_0018
230400	不推荐	0x2D00_0002	0x3000_0032
115200	不推荐	0x2D00_0006	0x3000_0066
57600	0x0000_000B	0x2D00_000E	0x3000_00CE
38400	0x0000_0012	0x2D00_0016	0x3000_0137
19200	0x0000_0025	0x2500_007B	0x3000_026F
9600	0x0000_004C	0x2A00_007B	0x3000_04E0
4800	0x0000_009A	0x2A00_00F8	0x3000_09C2

表 6.13-6 UART 控制器波特率寄存器设置示范表

6.13.5.2 UART控制器波特率补偿

UART控制器支持波特率补偿功能，用于补偿每位的精度。由于每一位都由BRCOMDEC位(UART_BRCOMP[31])来定义正负补偿，所以补偿的精度为UART模块时钟的一半。如果BRCOMPDEC(UART_BRCOMP[31])=0,每一位为正向补偿,被补偿位的时长会超过一个模块时钟长度。如果BRCOMPDEC(UART_BRCOMP[31])=1,每一位为负向补偿,被补偿位的时长会少于一个模块时钟长度。

BRCOMP[8:0](UART_BRCOMP[8:0])总共有9位，可以由用户自由定义对应的位是否被补偿。BRCOMP[7:0]用来配置UART_DAT[7:0]的补偿，而BRCOMP[8]用来定义校验位。

例如:

1. UART外设时钟=32.768K，波特率为9600

波特率为9600,UART外设时钟为32.768K→3.413时钟/位

如果波特率分频设为1(3时钟/位),每位的误差是-0.413时钟，设置BRCOMPDEC=0,

位	名	总误差	BRCOMP补偿	最终误差
0	开始位	-0.413	x	-0.413
1	UART_DAT[0]	-0.826(-0.413-0.413)	1	0.174
2	UART_DAT[1]	-0.239(0.174-0.413)	0	-0.239
3	UART_DAT[2]	-0.652(-0.239-0.413)	1	0.348
4	UART_DAT[3]	-0.065(0.348-0.413)	0	-0.065
5	UART_DAT[4]	-0.478(-0.065-0.413)	0	-0.478
6	UART_DAT[5]	-0.891(-0.478-0.413)	1	0.109
7	UART_DAT[6]	-0.304(0.109-0.413)	0	-0.304
8	UART_DAT[7]	-0.717(-0.304-0.413)	1	0.283

9	校验位	-0.130(0.283-0.413)	0	-0.13

表 6.13-7 波特率补偿例表 1

所以BRCOMP(UART_BRCOMP[8:0])设置为9’b010100101=0xa5.

2. UART外设时钟=32.768K，波特率为4800

波特率为4800,UART设置为32.768K→6.827时钟/位

如果波特率分频设为5(7时钟/位),每位误差为0.173时钟，BRCOMPDEC=1,

位	名	总误差	BRCOMP补偿	最终误差
0	Start	0.173	x	0.173
1	UART_DAT[0]	0.346(0.173+0.173)	0	0.346
2	UART_DAT[1]	0.519(0.346+0.173)	1	-0.481
3	UART_DAT[2]	-0.308(-0.481+0.173)	0	-0.308
4	UART_DAT[3]	-0.135(-0.308+0.173)	0	-0.135
5	UART_DAT[4]	-0.038(-0.135+0.173)	0	0.038
6	UART_DAT[5]	0.211(0.038+0.173)	0	0.211
7	UART_DAT[6]	0.384(0.211+0.173)	0	0.384
8	UART_DAT[7]	0.557(0.384+0.173)	1	-0.443
9	Parity	-0.270(-0.443+0.173)	0	-0.270

表 6.13-8 波特率补偿示表 2

所以BRCOMP(UART_BRCOMP[8:0])设置为9’b010000010=0x82.

UART控制器自动波特率功能模式

自动波特率功能可以自动测量从UART RX管脚输入数据的波特率。当自动波特率测量结束，测量结果将会放置在BRD(UA_BAUD[15:0])中。BAUDM1(UART_BAUD[29])和BAUDM0(UART_BAUD[28])都被自动设置为1。在自动波特率检测帧中，UART RX数据从起始位到第一个上升沿的时间由$2^{ABRDBITS}$位时间设定。

从起始位到第一个上升沿之间的$2^{ABRDBITS}$位时间通过设定ABRDBITS(UART_ALTCTL[20:19])计算得出。通过设置ABRDEN(UART_ALTCTL[18])可以使能自动波特率检测功能。在初始阶段，UART RX保持为1。一旦检测到下降沿，即为接收到起始位。自动波特率计数器被重启并开始计数。当检测到第一个上升沿时，自动波特率计数器将停止计数。然后，自动波特率计数器数值除以ABRDBITS(UART_ALTCTL[20:19])的结果将会自动载入到BRD(UART_BAUD[15:0])中，而且ABRDEN(UART_ALTCTL[18])会被清零。自动波特率测量如图 6.13-3。一旦自动波特率测量结束，ABRDIF(UART_FIFOSTS[1])会被置位。当自动波特率计数器溢出，ABRDTOIF(UART_FIFOSTS[2])将会被置位。ABRDIF(UART_FIFOSTS[1])或ABRDTOIF(UART_FIFOSTS[2])导致自动波特率标志ABRIF(UART_ALTCTL[17])产生。如果ABRIEN(UART_INTEN[18])使能，ABRIF(UART_ALTCTL[17])导致自动波特率中断ABRINT(UART_INTSTS[31])产生。

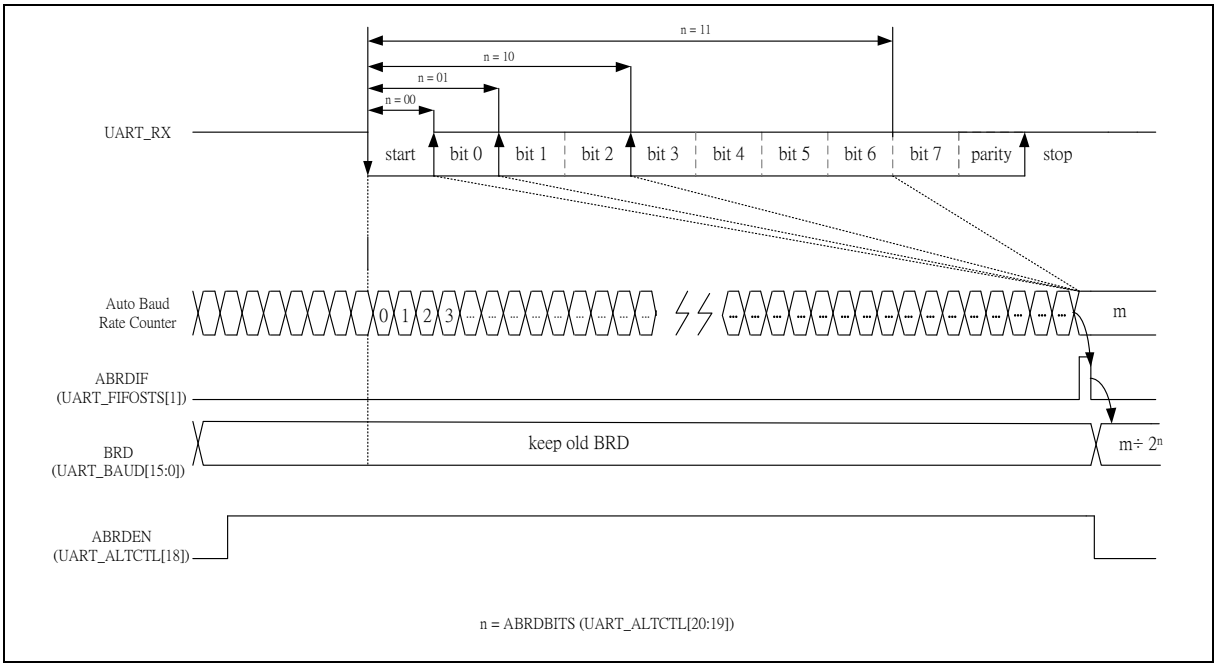


图 6.13-3 自动波特率测量

6.13.5.3 编程顺序范例:

1. 设置ABRDBITS(UART_ALTCTL[20:19])用于决定UART RX数据从起始到第一个上升沿时间用的$2^{ABRDBITS}$位时间
2. 设置ABRIEN(UART_INTEN[18])用于使能自动波特率检测功能中断
3. 时钟ABRDEN(UART_ALTCTL[18])用于使能自动波特率检测功能
4. 当ABRDIF(UART_FIFOSTS[1])为1，表示自动波特率测量结束
5. 执行UART发送和接收的动作
6. 当ABRDTOIF(UART_FIFOSTS[2])为1，表示自动波特率计数器溢出
7. 返回第三步

6.13.5.4 UART控制器发送延时时间

UART控制器可以通过设置DLY(UART_TOUT[15:8])来控制传输过程两个数据帧之间即上一个数据帧的停止位和下一个数据帧的起始位之间的间隔。单位是波特。延时操作如下图 6.13-4.

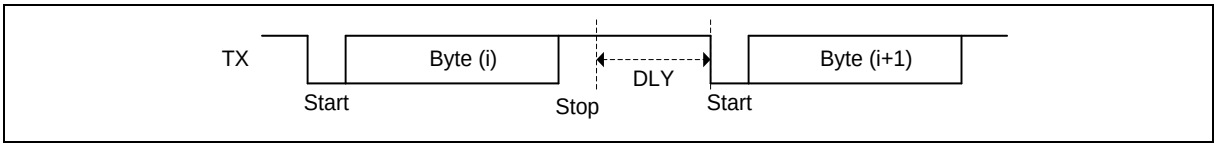


图 6.13-4 发送延时操作

6.13.5.5 UART控制器FIFO控制和状态

UART内置一个16字节的发送FIFO(TX_FIFO)和一个16字节的接收FIFO(RX_FIFO),通讯中,使用这些收发FIFO可以减少对CPU的中断次数。CPU在操作过程中任何时间都可以读取UART的状态。报告的状态信息包括UART正在执行的传输操作的条件，以及如果接收数据具有奇偶校验、帧错误或break错误时发生的3种错误条件(奇偶性错误、帧错误、break中断)。UART、IrDA和RS-485模式支持FIFO控制和状态

功能。

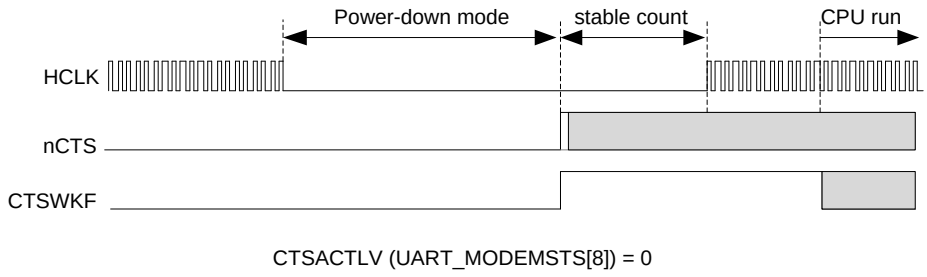
6.13.5.6 UART控制器唤醒功能

UART控制器支持系统唤醒功能。唤醒功能包括nCTS唤醒和接收数据唤醒、接收FIFO数据达到阈值唤醒、RS-485地址匹配(AAD模式)唤醒和接收数据FIFO阈值超时唤醒。CTSWKF(UART_WKSTS[0])、DATWKF(UART_WKSTS[1])、RFRTWKF(UART_WKSTS[2])、RS485WKF(UART_WKSTS[3])或TOUTWKF(UART_WKSTS[4])可以引发产生唤醒中断标志WKIF(UART_INTSTS[6])。如果WKIEN(UART_INTEN[6])使能，唤醒中断标志WKIF(UART_INTSTS[6])引发产生唤醒中断WKINT(UART_INTSTS[14])。

nCTS引脚唤醒:

当系统处于掉电模式且WKCTSEN(UART_WKCTL[0])置位，nCTS引脚可以唤醒系统。如果WKCTSEN(UART_WKCTL[0])使能，nCTS引脚上的跳变会产生nCTS唤醒标志CTSWKF(UART_WKSTS[0])。nCTS唤醒如图 6.13-5和图 6.13-6.

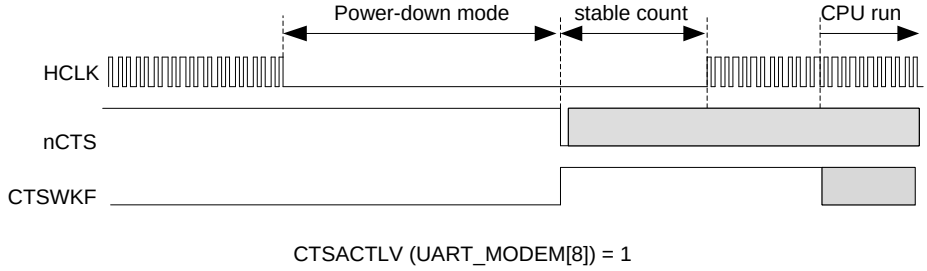
nCTS 唤醒示例 1(nCTS 发送由低到高)



注:Stable count指HCLK恢复稳定的时钟数.

图 6.13-5 UART nCTS 唤醒示例 1

nCTS 唤醒示例 2(nCTS 发送由高到低)



注:Stable count指HCLK恢复稳定的时钟数.

图 6.13-6 UART nCTS 唤醒示例 2

输入数据唤醒

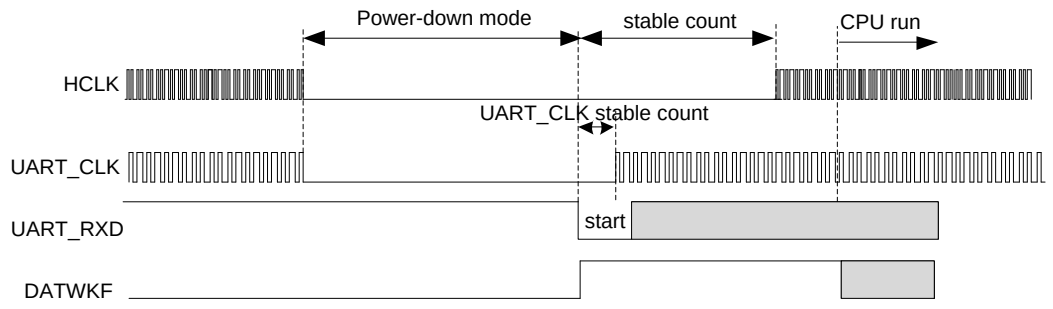
当系统处于掉电模式且WKDATEN(UART_WKCTL[1])置位，输入数据(UART_RXD)引脚上的跳变可以唤醒系统。为了能在系统唤醒后正确接收数据，需要设置STCOMP(UART_DWKCOMP[15:0])位域。STCOMP里的各位代表系统从掉电模式唤醒后，可以接收第1个位(开始位)需要的时钟数

输入数据唤醒系统后，输入的数据会被存储到FIFO内。如果使能了WKDATEN(UART_WKCTL[1])，输入数据(UART_RXD)引脚上的跳变会触发输入数据唤醒标志DATWKF(UART_WKSTS[1])输入数据唤醒如图 6.13-7.

注 1:UART控制器的时钟源需要选为HIRC且开始位的补偿时间需为91.129us。也就是说，

STCOMP(UART_DWKCOMP[15:0])可以设为0x1116。

注2:BRD(UART_BAUD[15:0])的值需要大于STCOMP(UART_DWKCOMP[15:0])。



注1: Stable count指HCLK恢复稳定的时钟数.

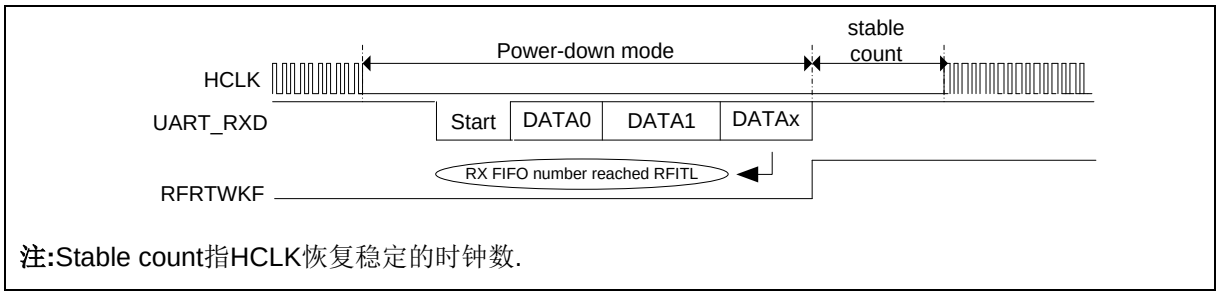
注2:UART_CLK稳定数指UART时钟源恢复稳定的计数.

图 6.13-7 UART 数据唤醒

接收数据 FIFO 达到阈值唤醒

设定WKRFRTEN(UART_WKCTL[2])可以使能接收数据FIFO达到阈值唤醒功能。在掉电模式，当RX FIFO 里接收到的数据达到了阈值设定 RFITL(UART_FIFO[7:4])，就会唤醒系统。如果WKRFRTEN(UART_WKCTL[2])使能，RXFIFO接收到了RFITL(UART_FIFO[7:4])设定个数的值后会触发接收数据FIFO达到阈值标志RFRTWKF(UART_WKSTS[2])。接收数据FIFO达到阈值唤醒如图6.13-8.

注:为了接收数据，掉电模式下UART时钟源需要设为LXT



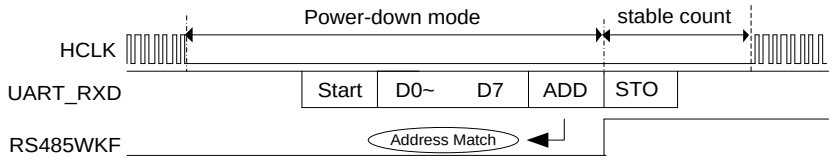
注:Stable count指HCLK恢复稳定的时钟数.

图 6.13-8 UART 接收数据 FIFO 达到阈值唤醒

RS-485 地址匹配(AAD 模式)唤醒

设定WKRFRTEN(UART_WKCTL[2])和WKRS485EN(UART_WKCTL[3])使能RS-485地址匹配唤醒功能。该功能需要在ADDRDEN(UART_ALTCTL[15])置1，处于RS-485自动地址检测(AAD)模式下使用。在掉电模式下，检测到的地址位与ADDRMV(UART_ALTCTL[31:24])相匹配或者RXFIFO里接收到的数据达到了阈值设定RFITL(UART_FIFO[7:4])都会唤醒系统。如果使能WKRS485EN(UART_WKCTL[3])，检测到的地址位与ADDRMV(UART_ALTCTL[31:24])相匹配时会触发RS485地址匹配(AAD模式)标志位RS485WKF(UART_WKSTS[3])。图 6.13-9是地址检测(AAD)模式唤醒示例.

注:为了接收数据，掉电模式下UART时钟源需要设为LXT



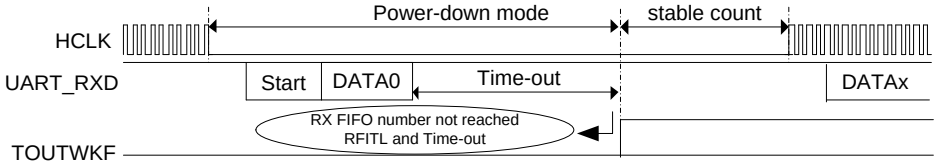
注:Stable count指HCLK恢复稳定的时钟数.

图 6.13-9 UART RS-485 AAD 模式地址匹配唤醒

数据接收 FIFO 超时唤醒

设置WKRFRTEN(UART_WKCTL[2])和WKTOUTEN(UART_WKCTL[4])使能数据接收FIFO超时唤醒功能。位TOCNTEN(UART_INTEN[11])使能接收缓存超时计数器。掉电模式下，RXFIFO里接收到的数据没有达到设定阈值RFITL(UART_FIFO[7:4])时，且超时计数器计数到TOIC(UART_TOUT[7:0])会唤醒系统。如果WKTOUTEN(UART_WKCTL[4])使能，超时计数器计数到TOIC(UART_TOUT[7:0])的值会触发数据接收FIFO阈值超时唤醒标志位TOUTWKF(UART_WKSTS[4])。图 6.13-10是数据接收超时唤醒.

注:为了接收数据，掉电模式下UART时钟源需要设为LXT或LIRC.



注:Stable count指HCLK恢复稳定的时钟数.

图 6.13-10 UART 数据接收 FIFO 阈值超时唤醒

6.13.5.7 UART控制器中断和状态

每个UART控制器都有10种类型中断:

- 接收数据有效中断(RDAINT)
- 发送保持寄存器空中断(THERINT)
- 发送完成中断(TXENDIF)
- Line接收状态中断(RLSINT)
  - 间隔中断标志(BIF)
  - 帧错误标志(FEF)
  - 校验错误标志(PEF)
  - RS-485地址字节检测标志(ADDRDETF)
- MODEM状态中断(MODEMINT)
  - 检测nCTS状态改变标志(CTSDETF)
- 接收缓存超时中断(RXTOINT)
- 缓存错误中断(BUFERRINT)
  - TX溢出错误中断标志(TXOVIF)

- RX溢出错误中断标志(RXOVIF)

- 唤醒中断(WKINT)
  - nCTS唤醒标志(CTSWKF)
  - 输入数据唤醒标志(DATWKF)
  - 接收数据FIFO达到阈值唤醒标志(RFRTWKF)
  - RS-485地址匹配(AAD模式)唤醒标志(RS485WKF)
  - 数据接收FIFO阈值超时唤醒标志(TOUTWKF)
- 自动波特率中断(ABRINT)
  - 自动波特率检测中断标志(ABRDIF)
  - 自动波特率检测超时中断标志(ABRDTOIF)
- 单线位错误检测中断(SWBEINT)

表 6.13-9所示是各中断源和标志位。当中断使能位使能，中断标志位触发，就会发生中断。用户需要在中断发生后清除中断标志。

中断源	中断指示	中断使能位	中断标志	标志触发条件	清除标志方法
接收数据可用中断	RDAINT	RDAIEN	RDAIF	N/A	读UART_DAT
发送保持寄存器空中断	THERINT	THREIEN	THREIF	N/A	写UART_DAT
发送完成中断	TXENDINT	TXENDIEN	TXENDIF	N/A	写UART_DAT
Line接收状态中断	RLSINT	RLSIEN	RLSIF	RLSIF=BIF	写‘1’到BIF
				RLSIF=FEF	写‘1’到FEF
				RLSIF=PEF	写‘1’到PEF
				RLSIF=ADDRDETF	写‘1’到ADDRDETF
MODEM状态中断	MODEMINT	MODEMIEN	MODEMIF	MODEMIF=CTSDETF	写‘1’到CTSDETF
接收缓存超时中断	RXTOINT	RXTOIEN	RXTOIF	N/A	读UART_DAT
缓存错误中断	BUFERRINT	BUFERRIEN	BUFERRIF	BUFERRIF=TXOVIF	写‘1’到TXOVIF
				BUFERRIF=RXOVIF	写‘1’到RXOVIF
唤醒中断	WKINT	WKIEN	WKIF	WKIF=CTSWKF	写‘1’到CTSWKF
				WKIF=DATWKF	写‘1’到DATWKF
				WKIF=RFRTWKF	写‘1’到RFRTWKF
				WKIF=RS485WKF	写‘1’到RS485WKF
				WKIF=TOUTWKF	写‘1’到TOUTWKF
自动波特率中断	ABRINT	ABRIEN	ABRIF	ABRIF=ABRDIF	写‘1’到ABRDIF
				ABRIF=ABRDTOIF	写‘1’到ABRDTOIF

单线位错误检测中断	SWBEINT	SWBEIEN	SWBEIF	N/A	写‘1’到SWBEIF

表 6.13-9 UART 控制器中断源和标志位列表

6.13.5.8 UART功能模式

UART控制器提供UART功能(用户须设置FUNCSEL(UART_FUNCSEL[1:0])设置为’00’来使能UART功能模式)。

UART为全双工异步通讯接口。收发各包含一个16字节的FIFO缓冲区。用户可以设置接收时的FIFO触发阈值以及定时溢出检测时间。发送数据帧间(即从上一帧停止位到下一帧起始位)时间间隔通过DLY(UART_TOR[15:8])位可设。UART支持硬件自动流控功能，且nRTS流控触发电平可设。如果RX FIFO中数据字节数大于或等于RTSTRGLV(UART_FIFO[19:16])，nRTS将被释放

UART 线控功能

UART控制器通过设置UART_LINE寄存器支持串行接口全部特性。可以通过对UART_LINE寄存器设置数据位和停止位长度以及校验位。表 6.13-10 和表 6.13-11列出了UART数据位和停止位长度的设置以及UART校验位的设置。

NSB (UART_LINE[2])	WLS (UART_LINE[1:0])	数据长度(Bit)	停止位长度(Bit)
0	00	5	1
0	01	6	1
0	10	7	1
0	11	8	1
1	00	5	1.5
1	01	6	2
1	10	7	2
1	11	8	2

表 6.13-10 UART 线控的数据位和停止位长度设置

校验类型	SPE (UART_LINE[5])	EPE (UART_LINE[4])	PSS (UART_LINE[7])	PBE (UART_LINE[3])	描述
No Parity	x	x	x	0	无奇偶校验位输出
Parity source from UART_DAT	x	x	1	1	软件产生并验证校验位
Odd Parity	0	0	0	1	奇校验位的计算方法是把数据流中的所有的1相加，使得包括校验位在内,1的总数为奇数个
Even Parity	0	1	0	1	偶校验位的计算方法是把数据流中的所有的1相加，使得包括校验位在内,1的总数为偶数个
Forced Mask Parity	1	0	0	1	奇偶校验位总是逻辑1 不管数据位中1的个数是奇数还是偶数，奇偶校验

					位永远都是逻辑1
Forced Space Parity	1	1	0	1	奇偶校验位总是逻辑0 不管数据位中1的个数是奇数还是偶数，奇偶校验位永远都是逻辑0

表 6.13-11 UART 线控校验位设置

UART 自动流控制功能功能

UART支持自动流控功能，该功能用到两根信号线nCTS(清零发送)和nRTS(请求发送)来控制UART与外部设备(如Modem)间的数据传输。当自动流控使能后，只有等到UART对外部设备发出有效的nRTS信号后才允许接收数据，否则不接收。当RX FIFO接收到数据的数量等于RTSTRGLV(UART_FIFO[19:16])位的值后，nRTS信号会被取消。当UART检测到外部设备给nCTS信号脚有效信号后，UART开始发送数据，否则UART不会发送数据。流控控制的框图参见图 6.13-11.

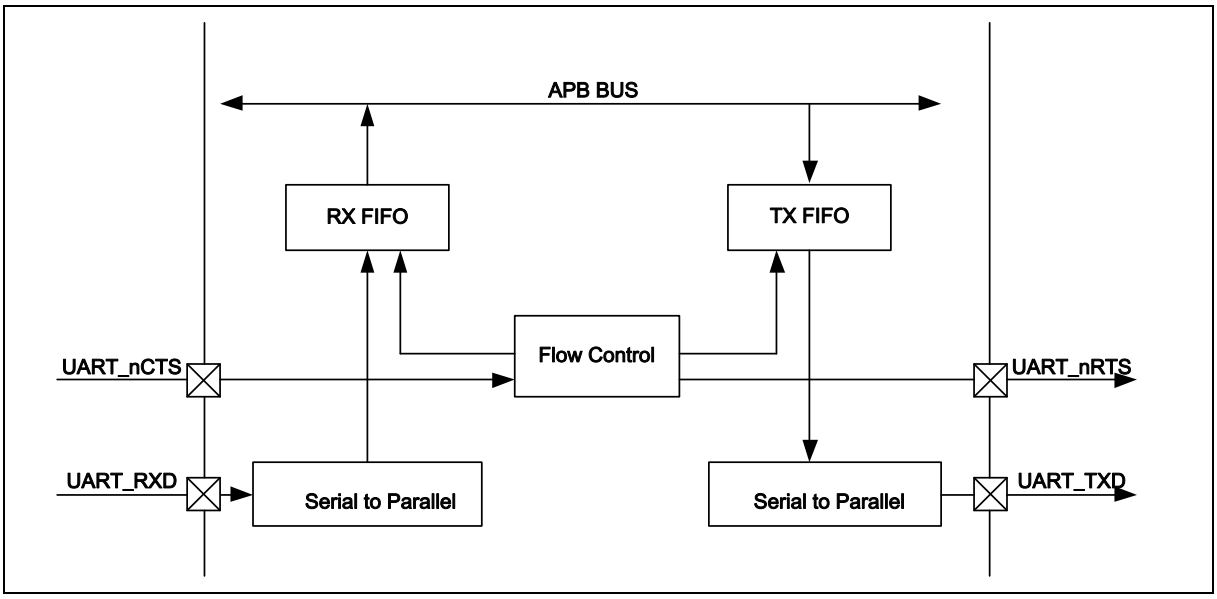


图 6.13-11 自动流控框图

图 6.13-12展示了UART nCTS自动流控功能。用户必须要先设置ATOCTSEN(UART_INTEN[13])以使能nCTS自动流控功能。CTSACTLV(UART_MODEMSTS[8])位可以设置CTS脚输入的有效状态。当nCTS脚上任何电平变化将导致CTSDETF(UART_MODEMSTS[0])位被置1，然后TX FIFO将自动将数据发送到TX脚,并传送出去。

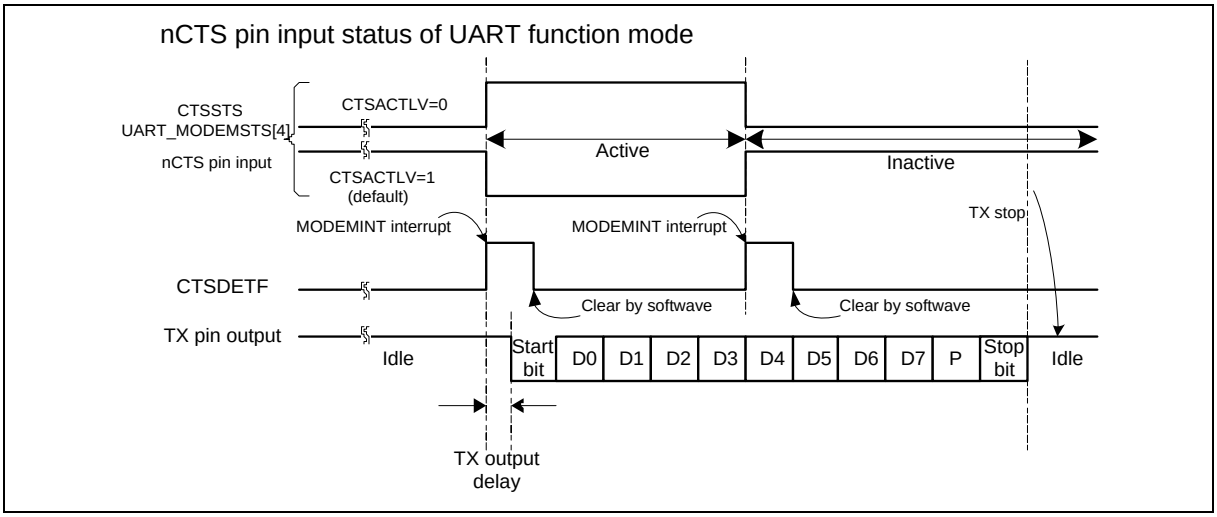


图 6.13-12 UART nCTS 自动流控使能

图 6.13-13所示，UART nRTS自动流控模式(ATORTSEN(UART_INTEN[12])=1)中，nRTS的触发阈值由UART FIFO控制寄存器的RTSTRGLV(UART_FIFO[19:16])控制。

设置RTSACTLV(UART_MODEM[9])可以控制nRTS引脚正常输出或反向，信号来自内部的nRTS。用户可以读RTSSTS(UART_MODEM[13])来获取nRTS脚输出电压的真实逻辑状态。

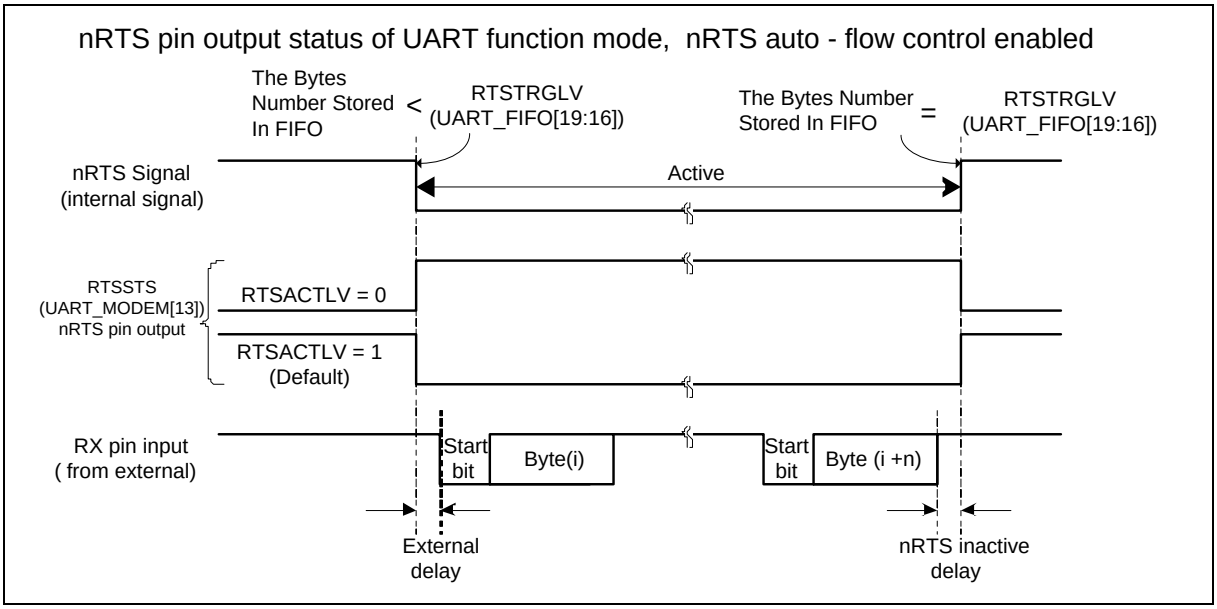


图 6.13-13 UART nRTS 自动流控功能使能

图 6.13-14，在软件模式下(ATORTSEN(UART_INTEN[12])=0)，软件改动RTS(UART_MODEM[1])控制位来实现nRTS流控。

设置RTSACTLV(UART_MODEM[9])可以控制nRTS引脚正常输出或反向，该信号来自RTS(UART_MODEM[1])。用户可以读RTSSTS(UART_MODEM[13])位来获取nRTS引脚真实输出电平状态。

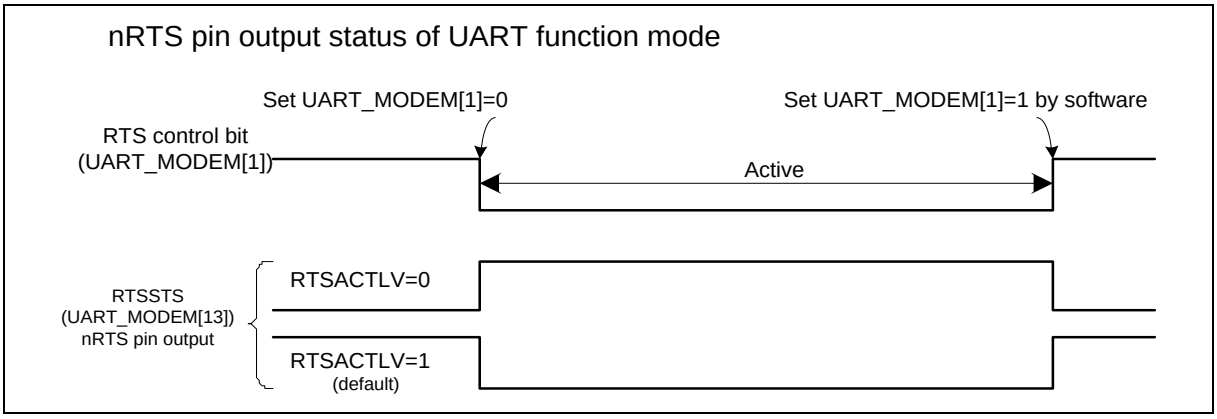


图 6.13-14 UART nRTS 软件控制的自动流控

6.13.5.9 IrDA功能模式

UART控制器也提供串行IrDA(SIR串行红外)功能，(用户必须设置UART_FUNCSEL[1:0]为'10'来使能IrDA功能)。SIR规范定义了一个短距离红外异步串行传输模式，它包括一个起始位、8个数据位和一个停止位。最大速率115.2kbps。IrDA SIR模块包含一个IrDA SIR协议编/解码器。IrDA SIR协议是半双工工作模式，所以它不能同时收发数据。IrDA SIR物理层规定了发送与接收数据的时间上至少10ms的时间间隔，该延迟特性需通过软件来完成。

IrDA模式下，BAUDM1(UART_BAUD[29])需为0

波特率=Clock/(16*(BRD+2))，这里BRD(UART_BAUD[15:0])是UA_BAUD寄存器中定义的波特率分频器。

注:IrDA主机和从机间波特率不能相差超过±5%.

图 6.13-15展示了IrDA控制模块框图。

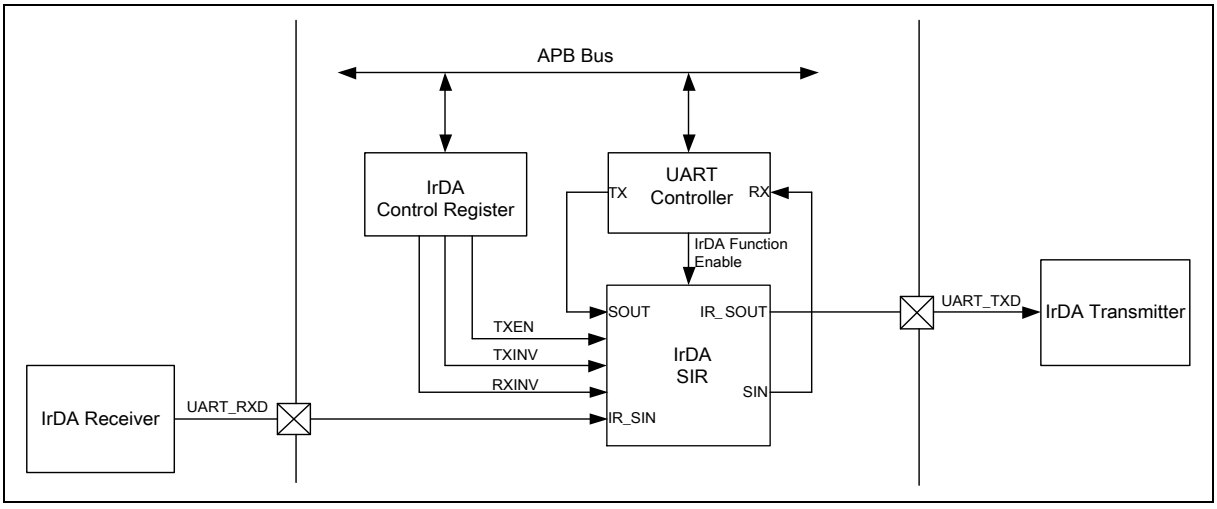


图 6.13-15 IrDA 控制模块框图

IrDA SIR 发送编码

IrDA SIR传送编码调制采用Non-Return-to Zero (NRZ)编码，数据流编码后从UART接口输出。IrDA SIR物理层指定使用归零反向调制编码(Return-to-Zero,反转(RZI))，用一个红外光脉冲代表逻辑0，被调制后的脉冲信号流发送到外部输出驱动器和红外线发光二极管上。

在正常模式下，传输脉冲的宽度为3/16波特率周期.

IrDA SIR 接收解码

IrDA SIR接收解码IrDA SIR接收解码器对输入引脚的(Return-to-Zero,Inverted(RZI))串行位流进行解调，并输出NRZ串行位流到UART接收数据输入端。

当解码器输入端为低时，表明接收到一个起始位。在空闲状态里，解码器输入端通常为高。正常操作时，RXINV(UART_IRDA[6])设置为1，TXINV(UART_IRDA[5])设置为0

IrDA SIR 操作

IrDA SIR编码/解码提供UART数据流和半双工串行SIR之间的转换。图 6.13-16是IrDA编码/解码波形图：

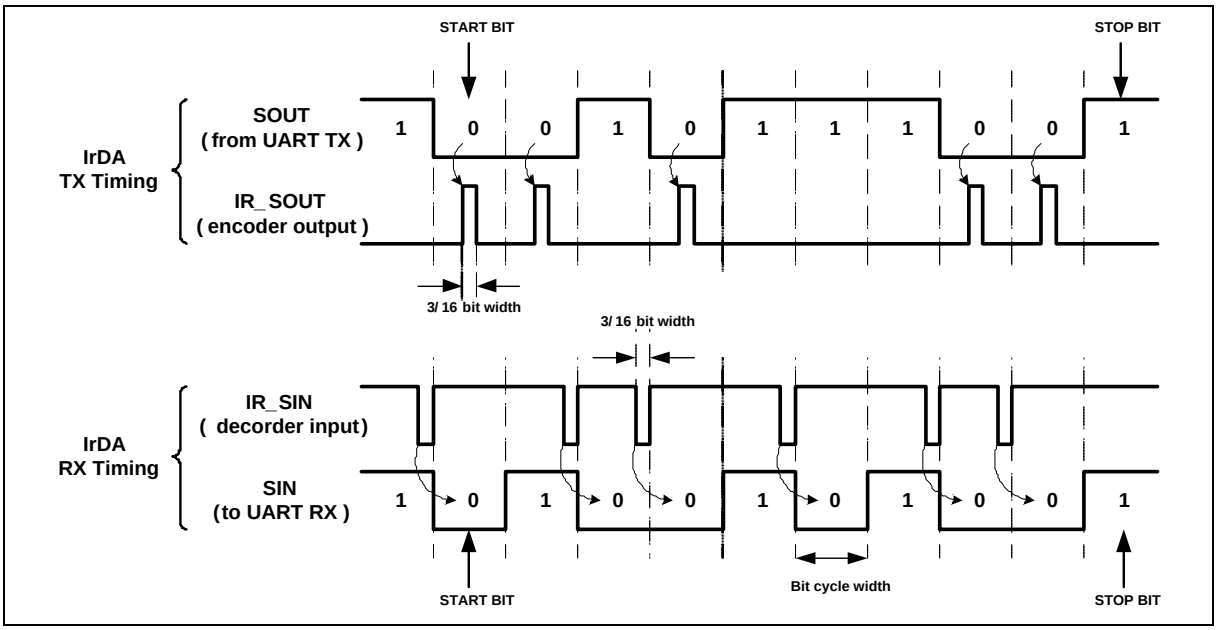


图 6.13-16 IrDA TX/RX 时序图

6.13.5.10 RS-485功能模式

UART控制器另一个可选择的功能是RS-485功能(用户必须设置UART_FUNCSEL[1:0]='11'来使能RS-485功能)，方向控制则由异步串口的nRTS脚来控制。RS-485收发器的驱动控制是通过nRTS控制信号来驱动的。RS-485模式下的RX和TX大多数特性与UART相同。

RS-485模式，控制器可以配置成RS-485可寻址的从机模式，RS-485主机发送可通过设置检验位(第9位)为1标识地址特性。对于数据字符，检验位设置为0。设置寄存器UART_LINE控制第9位(PBE,EPE和SPE置位时，第9位发送0；PBE和SPE置位，EPE清零时，第9位发送1)。

该控制器支持三种操作模式：RS-485普通多点操作模式(NMM)，RS-485自动地址识别模式(AAD)和RS-485自动方向控制模式(AUD)，可通过UART_ALTCTL寄存器的设置选择其中一种工作模式，通过设置DLY(UART_TOUT[15:8])可以设置上一个停止位与下一个开始位之间的延迟时间

RS-485 普通多点操作模式(NMM)

RS-485普通多点操作模式(RS485NMM(UART_ALTCTL[8])=1)，首先,软件决定在检测到地址字节之前的数据是否保存到RX FIFO。如果软件想忽略在检测到地址字节之前的任何数据，可以设置RXOFF(UART_FIFO[8])，使能RS485NMM(UART_ALTCTL[8])，这样接收器将忽略所有数据除非一个地址字节被检测到(bit9=1)，地址字节将被保存到RX FIFO。如果软件想接收地址字节前的任何数据，可以禁止RXOFF(UART_FIFO[8])，然后使能RS485NMM(UART_ALTCTL[8])，这样就能接收任何数据。

如果检测到地址字节(第9位为1)，会产生一个中断到CPU，软件可以通过设置RXOFF(UART_FIFO[8])，决定是否使能或禁用接收器接收数据。如果使能接收器，就会接收所有字节数据并存储到RX-FIFO。如果设置RXOFF(UART_FIFO[8])禁用接收器，会忽略所有接收到的字节数据，直到检测到下一个地址字节。当检测到地址字节后，控制器将清除RXOFF(UART_FIFO[8])且地址字节数据将存储到RX-FIFO

RS-485 自动地址识别工作模式(AAD)

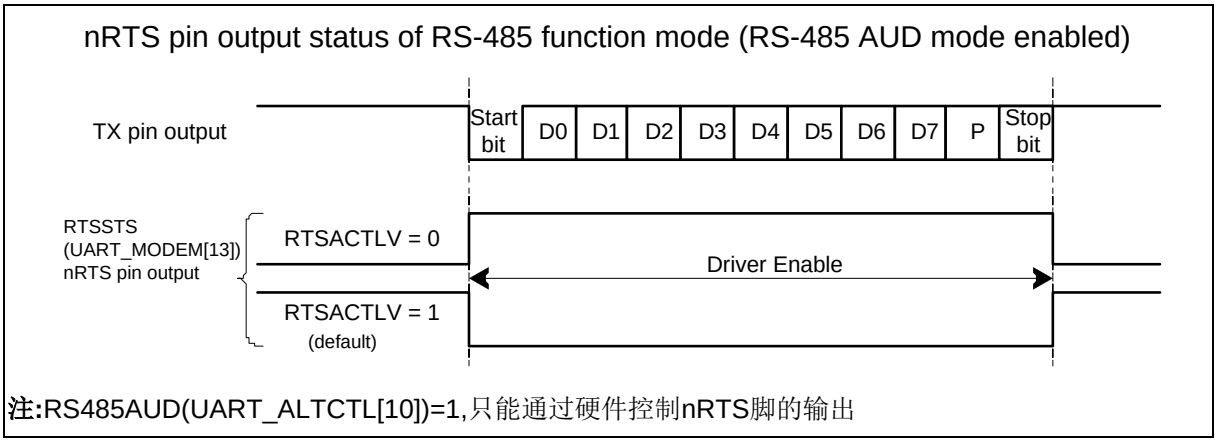
RS-485自动地址识别模式(RS485AAD(UART_ALTCTL[9])=1)，接收器在检测到地址字节(第9位为1)并且地址字节数据与ADDRMV(UART_ALTCTL[31:24])的值相匹配之前将忽略所有数据。然后匹配的地址字节数据将存储在RX-FIFO。之后所有接收到的字节数据都将被接收并存储于RX-FIFO，直到出现与ADDRMV(UART_ALTCTL[31:24])的值不匹配的地址字节。

RS-485 自动方向控制功能(AUD)

RS-485控制器的另一个功能是RS-485自动方向控制功能(RS485AUD(UART_ALTCTL[10)=1)。RS-485通过nRTS驱动控制异步串口的控制信号，使能RS-485驱动器。nRTS连接到RS-485驱动器，设置nRTS线为高(逻辑1)使能RS-485驱动器。设置nRTS为低(逻辑0)，使驱动器进入tri-state状态。用户通过设置寄存器UART_MODEM中的RTSACTLV改变nRTS驱动电平。

图 6.13-17展示了AUD模式下RS-485 nRTS 驱动电平。nRTS脚在TX数据发送阶段自动驱动收发器。

设置RTSACTLV(UART_MODEM[9])可以控制nRTS脚的输出电平。用户可以通过读RTSSTS(UART_MODEM[13])来获取nRTS脚上实际的输出逻辑电平。

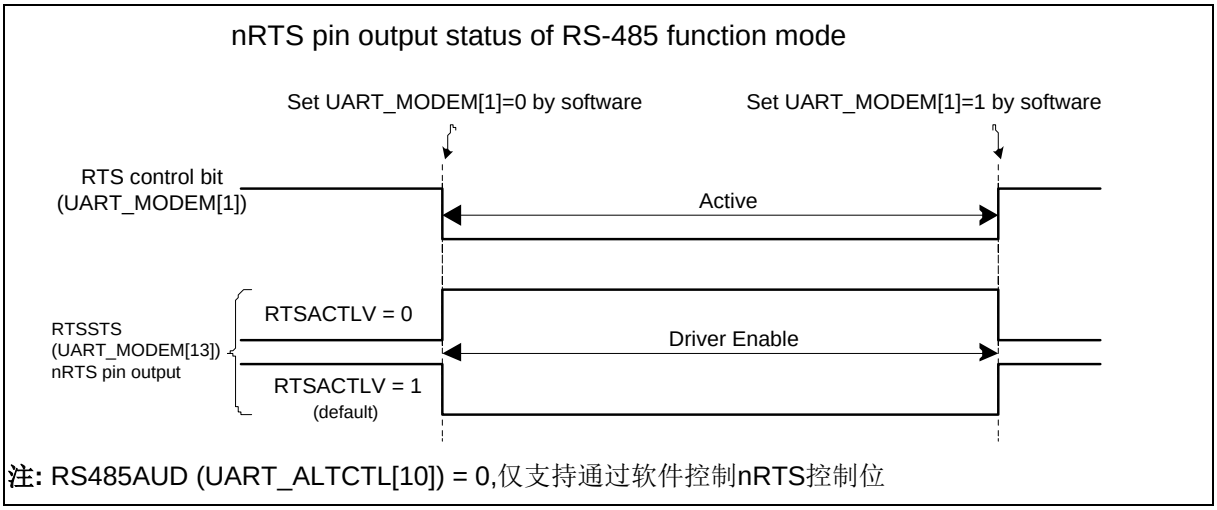


注:RS485AUD(UART_ALTCTL[10])=1,只能通过硬件控制nRTS脚的输出

图 6.13-17 RS-485 自动方向控制模式下 nRTS 引脚驱动电平

图 6.13-18展示了通过软件控制(RS485AUD(UART_ALTCTL[10])=0)模式下，RS-485 nRTS脚的驱动电平。nRTS驱动电平通过设置RTS(UART_MODEM[1])来控制。

设置RTSACTLV(UART_MODEM[9])可以控制nRTS脚的输出与RTS(UART_MODEM[1])设置值是否相反。用户可以读RTSSTS(UART_MODEM[13])来获取nRTS脚上实际的逻辑电平。RS-485的帧结构参见图 6.13-19.

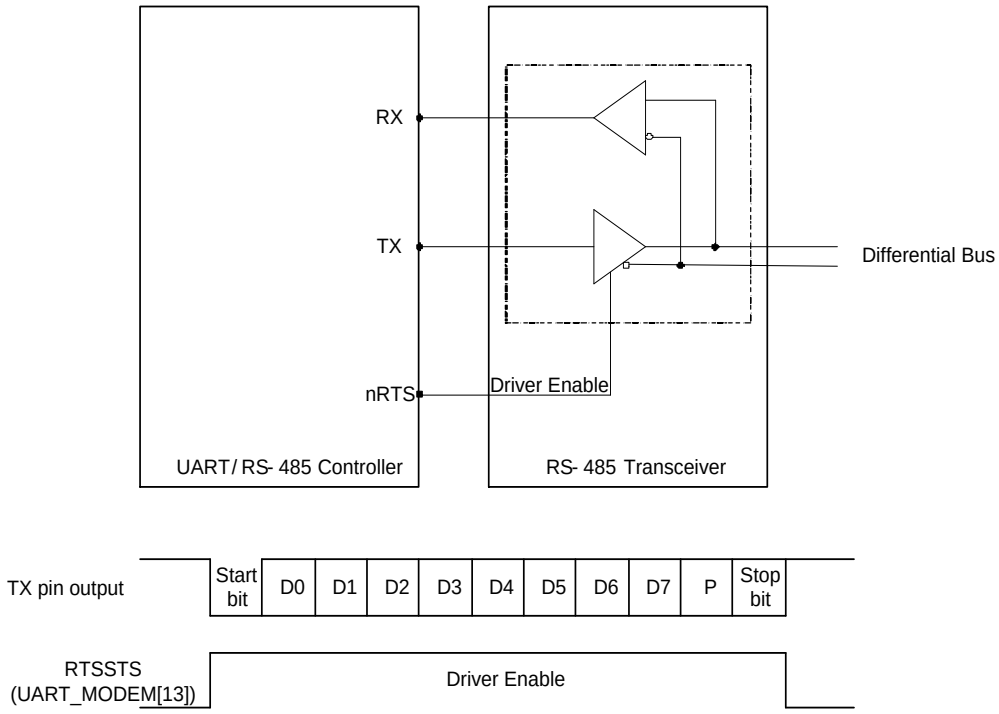


注: RS485AUD (UART_ALTCTL[10]) = 0,仅支持通过软件控制nRTS控制位

图 6.13-18 RS-485 nRTS 软件控制时的驱动电平

编程流程示例:

1. 设置UART_FUNCSEL中的FUN_SEL选择RS-485功能.
2. 设置RXOFF(UART_FIFO[8])，使能或禁用RS-485接收器
3. 设置RS485NMM(UART_ALTCTL[8])或RS485AAD(UART_ALTCTL[9])模式..
4. 如果选择RS485AAD(UART_ALTCTL[9])模式，ADDRMV(UART_ALTCTL[31:24])需设置成自动地址匹配值
5. 设置RS485_AUD(UA_ALT_CSR[10])来决定是否为自动方向控制)



注: RS485AUD (UART_ALTCTL[10]) 需为 1, 且 RTSACTLV (UART_MODEM[9]) 需为 0。

图 6.13-19 RS-485 帧结构

6.13.5.11 UART单线半双工模式

UART控制器提供在UART功能模式下的单线半双工功能（设置UART_FUNCSEL[2:0]为'100'来使能UART单线功能)。单线总线空闲时保持RX状态。通过将数据写入TX缓冲区DAT(UART_DAT[7:0])，单线总线立即将总线RX状态转变到TX状态。传输结束后，单线总线状态从TX状态回到RX状态。

为了减少总线冲突问题，UART控制器支持流控制和位错误检测。nRTS在总线为保持TX状态时不激活。UART默认状态为RX 模式，并且UART只会在ATOCTSEN (UART_INTEN[13])使能，nCTS 被取消激活状态后，才转换为TX模式发送数据。在TX状态期间，UART控制器将监视总线状态。如果总线状态不等于UART控制器TX状态，则设置SWBEIF(UART_INTSTS[16])。

注1:在写入TX缓存数据前,总线状态可以通过读RXIDLE(UART_FIFOSTS[29])来检测是否为空闲状态。总线冲突可能导致RX接收错误数据。

注2:单线不支持自动流控制，因为自动流控制会在TX发送时自动激活nRTS。

6.13.5.12 PDMA传输功能

UART控制器支持PDMA传输功能.

设定PDMA相关参数并设定UART_DAT为PDMA目标地址后，当TXPDMAEN(UART_INTEN[14])为1，UART控制器会请求PDMA控制器自动开始PDMA传输。

设定PDMA相关参数并设定UART_DAT为PDMA为源地址后，当RXPDMAEN(UART_INTEN[15])为1，UART控制器会请求PDMA控制器在RX FIFO缓存有数据时，自动开始PDMA传输。

注:如果配置了STOPn(PDMA_STOP[n])寄存器停止UART RXPDMA任务，且UART还未完成接收，UART控制器会完成传输并存储当前的接收数据到接收缓存。读寄存器RXEMPTY(UART_FIFOSTS[14])可以知道当前接收缓存有无有效数据。

6.13.6 寄存器映射

R:只读,**W**:只写,**R/W**:读写

寄存器	偏移	R/W	描述	复位值
UART 基地址: **UARTx_BA = 0x4007_0000 + (0x1000 * x)** **x=0,1,2,3,4,5,6,7**				
UART_DAT **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x00	R/W	UART接收/发送缓存寄存器	未定义
UART_INTEN **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x04	R/W	UART中断使能寄存器	0x0000_0000
UART_FIFO **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x08	R/W	UART FIFO控制寄存器	0x0000_0101
UART_LINE **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x0C	R/W	UART线控寄存器	0x0000_0000
UART_MODEM **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x10	R/W	UART Modem控制寄存器	0x0000_0200
UART_MODEMSTS **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x14	R/W	UART Modem状态寄存器	0x0000_0110
UART_FIFOSTS **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x18	R/W	UART FIFO状态寄存器	0xB040_4000
UART_INTSTS **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x1C	R/W	UART中断状态寄存器	0x0040_0002
UART_TOUT **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x20	R/W	UART超时寄存器	0x0000_0000
UART_BAUD **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x24	R/W	UART波特率分频器	0x0F00_0000
UART_IRDA **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x28	R/W	UART IrDA控制寄存器	0x0000_0040
UART_ALTCTL **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x2C	R/W	UART选择控制/状态寄存器	0x0000_000C
UART_FUNCSEL **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x30	R/W	UART功能选择寄存器	0x0000_0000
UART_BRCOMP **x=0,1,4,5**	UARTx_BA+0x3C	R/W	UART波特率补偿寄存器	0x0000_0000
UART_WKCTL **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x40	R/W	UART唤醒控制寄存器	0x0000_0000
UART_WKSTS **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x44	R/W	UART唤醒状态寄存器	0x0000_0000
UART_DWKCOMP **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x48	R/W	UART输入数据唤醒补偿寄存器	0x0000_0000

6.13.7 寄存器描述

UART接收/发送缓存寄存器(UART_DAT)

寄存器	偏移	R/W	描述	复位值
UART_DAT x=0,1,2,3,4,5,6,7	UARTx_BA+0x00	R/W	UART接收/发送缓存寄存器	未定义

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

位	描述	
[31:9]	**Reserved**	保留.
[8]	**PARITY**	**校验位接收/发送缓存** 写操作： 写该位，校验位会被存入发送FIFO。 如果PBE(UART_LINE[3])和PSS(UART_LINE[7])置位，UART控制器会通过UART_TXD发送DAT(UART_DAT[7:0])及该位 如果PBE(UART_LINE[3])和PSS(UART_LINE[7])使能，可以读这位获得校验位 **注:**仅PBE(UART_LINE[3])和PSS(UART_LINE[7])置位时，该位有效.
[7:0]	**DAT**	**数据接收/发送缓存** 写操作： 写数据到该寄存器，数据将会保存到发送FIFO。UART控制器将会通过UART_TXD把存放在FIFO中最前面的数据发送出去 读操作： 读该寄存器,UART将返回从接收FIFO中接收到的8位数据

UART中断使能寄存器(UART_INTEN)

寄存器	偏移	R/W	描述	复位值
UART_INTEN x=0,1,2,3,4,5,6,7	UARTx_BA+0x04	R/W	UART中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**TXENDIEN**	**Reserved**			**ABRIEN**	**Reserved**	**SWBEIEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXPDMAEN	**TXPDMAEN**	**ATOCTSEN**	**ATORTSEN**	**TOCNTEN**	**Reserved**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**WKIEN**	**BUFERRIEN**	**RXTOIEN**	**MODEMIEN**	**RLSIEN**	**THREIEN**	**RDAIEN**

位	描述	
[31:23]	**Reserved**	保留.
[22]	**TXENDIEN**	**发送完成中断使能位** 使能TXENDIEN(UART_INTEN[22]), 若TXENDIF(UART_INTSTS[22])置位，(TX FIFO(UART_DAT)为空且最后一字节的停止位已发送，则会产生发送完成中断TXENDINT(UART_INTSTS[30])。 0=禁止发送完成中断 1=使能发送完成中断
[21:19]	**Reserved**	保留.
[18]	**ABRIEN**	**自动波特率中断使能位** 0=禁止自动波特率中断 1=使能自动波特率中断
[17]	**Reserved**	保留.
[16]	**SWBEIEN**	**单线位错误检测中断使能位** 设置这个位，当单线位错误检测SWBEIF(UART_INTSTS[16])被置位时，单线半双工位错误检测中断SWBEINT(UART_INTSTS[24])将产生。 0=单线位错误检测中断禁止 1=单线位错误检测中断使能 **注:**当FUNCSEL(UART_FUNCSEL[2:0])选择为单线模式时，此位有效。
[15]	**RXPDMAEN**	**RX PDMA 使能位** 该位可以使能或禁止RX PDMA功能. 0=禁止RX PDMA. 1=使能RX PDMA. **注:**如果RLSIEN(UART_INTEN[2])为使能并且HWRLSINT(UART_INTSTS[26])=1，RLS(接收线状态)中断将产生。如果RLS中断是因为分隔错误标志BIF(UART_FIFOSTS[6])、帧错误标志FEF(UART_FIFO[5])或是奇偶校验错误标志PEF(UART_FIFOSTS[4])产生，UART

		PDMA接收请求将被停止。写“1”到错误标志BIF、FEF和PEF，分别清除分隔错误标志BIF、帧错误标志FEF和奇偶校验错误标志PEF后，UART PDMA接收请求将继续进行
[14]	**TXPDMAEN**	**TX PDMA使能位** 该位可以使能或禁止TX PDMA 功能. 0=禁止TX PDMA. 1=使能TX PDMA. **注:**如果RLSIEN(UART_INTEN[2])位使能并且HWRLSINT(UART_INTSTS[26])=1，RLS(接收线状态)中断将产生。如果RLS中断是因为break错误标志BIF(UART_FIFOSTS[6])、帧错误标志FEF(UART_FIFO[5])或是奇偶校验错误标志PEF(UART_FIFOSTS[4])产生，UART PDMA接收请求将被停止。写“1”到错误标志BIF、FEF和PEF，分别清除break错误标志BIF、帧错误标志FEF和奇偶校验错误标志PEF后，UART PDMA接收请求将继续进行。
[13]	**ATOCTSEN**	**nCTS自动流控使能位** 0=禁止nCTS自动流控. 1=使能nCTS自动流控. **注:**当nCTS自动流控使能后，当nCTS有效，UART会发送数据到外部设备(UART将不会发送数据到外部设备直到nCTS有效).
[12]	**ATORTSEN**	**nRTS自动流控使能位** 0=禁止nRTS自动流控. 1=使能nRTS自动流控. **注:**当nRTS自动流控使能后，如果RX FIFO中字节的数量等于RTSTRGLV(UART_FIFO[19:16])，UART会自动禁止nRTS信号
[11]	**TOCNTEN**	**接收缓存超时计数器使能位** 0=禁止接收缓存超时计数器 1=使能接收缓存超时计数器
[10:7]	**Reserved**	保留.
[6]	**WKIEN**	**唤醒中断使能位** 0=禁止唤醒中断 1=使能唤醒中断.
[5]	**BUFERRIEN**	**缓存错误中断使能位** 0=禁止缓存错误中断. 1=使能缓存错误中断.
[4]	**RXTOIEN**	**RX超时中断使能位** 0=禁止RX超时中断. 1=使能RX超时中断
[3]	**MODEMIEN**	**Modem状态中断使能位** 0=禁止Modem状态中断. 1=使能Modem状态中断.
[2]	**RLSIEN**	**接收Line状态中断使能位** 0=禁止接收Line状态中断. 1=使能接收Line状态中断.
[1]	**THREIEN**	**发送保持寄存器空中断使能位** 0=禁止发送保持寄存器空中断. 1=使能发送保持寄存器空中断

[0]	**RDAIEN**	接收数据可用中断使能位 0=禁止接收数据可用中断. 1=使能接收数据可用中断.

UART FIFO 控制寄存器(UART_FIFO)

寄存器	偏移	R/W	描述	复位值
UART_FIFO x=0,1,2,3,4,5,6,7	UARTx_BA+0x08	R/W	UART FIFO控制寄存器	0x0000_0101

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

位	描述	
[31:20]	**Reserved**	保留.
[19:16]	**RTSTRGLV**	**nRTS自动流控触发阈值** 0000=nRTS触发阈值为1字节. 0001=nRTS触发阈值为4字节. 0010=nRTS触发阈值为8字节. 0011=nRTS触发阈值为14字节. 其他 = 保留. **注:** 该区域用于自动 nRTS 流控制
[15:9]	**Reserved**	保留.
[8]	**RXOFF**	**接收禁止位** 是否禁用接收 (置 1 禁用接收) 0 =使能接收 1 =禁用接收 **注：** 该位用于 RS-485 普通多点模式,需要在 RS485NMM (UART_ALTCTL [8]) 被设置之前设置
[7:4]	**RFITL**	**RX FIFO 中断触发阈值** 当 FIFO 接收字节数等于 RFITL 后， RDAIF (UART_INTSTS[0]) 将被置位 (如果 RDAIEN (UART_INTEN [0]) 使能，将产生中断)。 0000 = RX FIFO 触发中断阈值为 1 字节 0001 = RX FIFO 触发中断阈值为 4 字节 0010 = RX FIFO 触发中断阈值为 8 字节 0011 = RX FIFO 触发中断阈值为 14 字节 其它 = 保留
[3]	**Reserved**	保留.

[2]	**TXRST**	**TX 域软件复位** 当 TXRST (UART_FIFO[2]) 置位， 发送 FIFO 和 TX 内部状态机中的所有数据将被清除. 0 = 无效 1 = 复位 TX 内部状态机和指针. **注 1:** 最少 3 个 UART 外设时钟后，该位自动清 0 . **注 2:** 设置该位前，需要先等待TXEMPTYF (UART_FIFOSTS[28]) 被置位
[1]	**RXRST**	**RX 域软件复位** 当 RXRST (UART_FIFO[1]) 置位， 接收 FIFO 和 RX 内部状态机中的所有数据将被清除. 0 = 无效 1 = 复位 RX 内部状态机和指针. **注 1**: 最少 3 个 UART 外设时钟后，该位自动清 0 **注 2**: 设置该位前，需要先等待RXIDLE (UART_FIFOSTS[29]) 被置位
[0]	**Reserved**	保留.

UART 线控寄存器(UART_LINE)

寄存器	偏移	R/W	描述	复位值
UART_LINE x=0,1,2,3,4,5,6,7	UARTx_BA+0x0C	R/W	UART 线控寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**RXDINV**	**TXDINV**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSS	**BCB**	**SPE**	**EPE**	**PBE**	**NSB**	**WLS**	

位	描述	
[31:10]	**Reserved**	保留.
[9]	**RXDINV**	**RX 数据反转** 0 = 禁止接收数据信号反转. 1 = 使能接收数据信号反转. **注 1:** 设置该位前，需要先置位 TXRXDIS (UART_FUNCSEL[3])，再等 TXRXACT (UART_FIFOSTS[31]) 被清除。完成配置后，清除 TXRXDIS (UART_FUNCSEL[3]) 以触发 UART 控制器 **注2:** 该位仅在 FUNCSEL (UART_FUNCSEL[1:0]) 配置为 UART, LIN 或 RS485 模式下有效
[8]	**TXDINV**	**TX 数据反转** 0 = 禁止发送数据信号反转. 1 = 使能发送数据信号反转. **注 1:** 设置该位前，需要先置位 TXRXDIS (UART_FUNCSEL[3])，再等 TXRXACT (UART_FIFOSTS[31]) 被清除。完成配置后，清除 TXRXDIS (UART_FUNCSEL[3]) 以触发 UART 控制器 **注2:** 该位仅在 FUNCSEL (UART_FUNCSEL[1:0]) 配置为 UART, LIN 或 RS485 模式下有效
[7]	**PSS**	**校验位源选择** 校验位可以由软件或是自动产生并校验 0 = 校验位由EPE (UART_LINE[4]) 和 SPE (UART_LINE[5])的设置决定并自动检查. 1 = 校验位由软件产生并校验. **注1:** 仅 PBE (UART_LINE[3]) 置位时，该位有效. **注2:** 如果 PSS 为 0，校验位自动发送并检查。如果 PSS 为 1，发送出去的校验位可由 PARITY (UART_DAT[8]) 设置，读 PARITY (UART_DAT[8]) 可得校验位
[6]	**BCB**	**Break 控制位** 0 = 禁止Break 控制

		1 = 使能Break 控制 **注:** 当该位被置逻辑 1，串行数据输出 (TX) 将强制到 Spacing 状态 (logic 0)。该位仅作用于 TX ，对传输逻辑不起作用
[5]	**SPE**	**Stick 校验使能位** 0 = 禁止Stick 校验. 1 = 使能Stick 校验. **注:** 如果 PBE (UART_LINE[3]) 和 EPE (UART_LINE[4]) 为逻辑 1，校验位发送和检验值为逻辑 0。如果 PBE (UART_LINE[3]) 是1 ，EPE (UART_LINE[4]) 是 0，则校验位发送和检验值为 1。
[4]	**EPE**	**偶校验使能位** 0 = 逻辑 1 的奇数数目在每个字节中被发送和检验 1 = 逻辑 1 的偶数数目在每个字中被发送和检验 **注:**该位只在 PBE (UART_LINE[3]) 置位时有效。
[3]	**PBE**	**校验使能位** 0 = 禁止生成校验位 1 = 使能生成校验位 **注:** 每一个发送字符中都产生校验位，对每一个传进来的数据进行校验位检测
[2]	**NSB**	**停止位数目** 0= 当发送数据时，产生 1 个“ STOP 位” 1= 当发送数据，选择 5-位 字长度时，产生 1.5 “ STOP 位” 当选择 6-, 7- 和 8-位字长度时，产生 2 个“STOP 位”.
[1:0]	**WLS**	**字长选择** 该域选择 UART 的字长. 00 = 5 位. 01 = 6 位. 10 = 7 位. 11 = 8 位.

UART Modem控制寄存器(UART_MODEM)

寄存器	偏移	R/W	描述	复位值
UART_MODEM x=0,1,2,3,4,5,6,7	UARTx_BA+0x10	R/W	UART Modem 控制寄存器	0x0000_0200

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

位	描述	
[31:14]	**Reserved**	保留.
[13]	**RTSSTS**	**nRTS引脚状态 (只读)** 该位的值对应于 nRTS 引脚的输出电平 0 = nRTS 引脚为低电平逻辑状态 1 = nRTS 引脚为高电平逻辑状态.
[12:10]	**Reserved**	保留
[9]	**RTSACTLV**	**nRTS 脚的有效电平** 该位定义了 nRTS 输出引脚的有效电平 0 = nRTS 引脚输出为高电平有效. 1 = nRTS 引脚输出为低电平有效.(默认) **注1:** UART 的功能模式请参考图图 6.13-13 和 图 6.13-14 **注2:** RS-485 功能模式请参考图 6.13-17 和 图 6.13-18 **注3:** 设置该位前，需要先置位 TXRXDIS (UART_FUNCSEL[3])，再等 TXRXACT (UART_FIFOSTS[31]) 被清除。完成配置后，清除 TXRXDIS (UART_FUNCSEL[3]) 以触发 UART 控制器
[8:2]	**Reserved**	保留
[1]	**RTS**	**nRTS (请求发送) 信号控制** 该位直接控制内部 nRTS 脚信号是否有效，然后使用 RTSACTLV 位的配置驱动 nRTS 脚输出 0 = nRTS 信号有效. 1 = nRTS 信号无效. **注1:** UART 功能模式下，当 nRTS 自动流控被使能后，nRTS 信号控制位无效 **注2:** RS-485 模式下，当 RS-485 自动方向模式 (AUD) 被使能后，nRTS 信号控制位无效 **注3:** 支持单线模式

[0]	**Reserved**	保留.

UART Modem状态寄存器(UART_MODEMSTS)

寄存器	偏移	R/W	描述	复位值
UART_MODEMSTS **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x14	R/W	UART Modem状态寄存器	0x0000_0110

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

Bits	Description	
[31:9]	**Reserved**	保留.
[8]	**CTSACTLV**	**nCTS 引脚有效电平** 该位定义了 nCTS 输入引脚脚的有效电平 0 = nCTS 输入脚高电平有效 1 = nCTS 输入脚低电平有效.(默认) **注:** 设置该位前，需要先置位 TXRXDIS (UART_FUNCSEL[3])，再等 TXRXACT (UART_FIFOSTS[31]) 被清除。完成配置后，清除 TXRXDIS (UART_FUNCSEL[3]) 以触发 UART 控制器
[7:5]	**Reserved**	保留.
[4]	**CTSSTS**	**nCTS 引脚状态 (只读)** 该位对应于 nCTS 引脚输入逻辑状态 0 = nCTS 引脚输入状态为低电平 1 = nCTS 引脚输入状态为高电平 **注:** 当 UART 控制器外设时钟被使能，且 nCTS 多功能引脚被使能，该位才有效。
[3:1]	**Reserved**	保留.
[0]	**CTSDETF**	**检测到 nCTS 状态改变标志** 如果 nCTS 输入脚上有电平变化，该位将被置位，如果 MODEMIEN (UART_INTEN [3]) 位被 置位，将会产生 Modem 中断。 0 = nCTS 输入引脚状态无变化. 1 = nCTS 输入引脚状态已变化. **注:** 该位写 1 清 0 .

UART FIFO状态寄存器(UART_FIFOSTS)

寄存器	偏移	R/W	描述	复位值
UART_FIFOSTS x=0,1,2,3,4,5,6,7	UARTx_BA+0x18	R/W	UART FIFO状态寄存器	0xB040_4000

31	30	29	28	27	26	25	24
TXRXACT	**Reserved**	**RXIDLE**	**TXEMPTYF**	**Reserved**			**TXOVIF**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXFULL	**TXEMPTY**	**TXPTR**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXFULL	**RXEMPTY**	**RXPTR**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BIF**	**FEF**	**PEF**	**ADDRDETF**	**ABRDTOIF**	**ABRDIF**	**RXOVIF**

位	描述	
[31]	**TXRXACT**	**TX 和 RX 有效状态 (只读)** 该位反映 TX 和 RX 是否有效 0 = TX 和 RX 无效. 1 = TX 和 RX 有效.(默认) **注:** 当 TXRXDIS (UART_FUNCSEL[3]) 置位，TX 和 RX 都空闲时，该位清 0，UART 控制器此时无法发送或接收数据。其他情况，该位均为 1.
[30]	**Reserved**	保留.
[29]	**RXIDLE**	**RX 空闲状态 (只读)** RX 空闲时，硬件自动置位该位. 0 = RX 忙. 1 = RX 空闲 (默认)
[28]	**TXEMPTYF**	**发送空标志 (只读)** 当 TX FIFO (UART_DAT) 为空,并且最后一个字节的 STOP 位也已经被发送完毕，该位被硬件置 1. 0 = TX FIFO 不为空或最后一个字节的 STOP 位还没有发送. 1 = TX FIFO 为空而且最后一个字节的 STOP 位已经发送 **注:** 当 TX FIFO 不为空或最后一个字节的 STOP 位还没有被发送完毕，那么该位将被自动清零
[27:25]	**Reserved**	保留.
[24]	**TXOVIF**	**TX 溢出错误中断标志 (只读)** 如果 TX FIFO (UART_DAT) 满，如果再向 UART_DAT 写入数据，将会导致此位被置 1.. 0 = TX FIFO 未溢出 1 = TX FIFO 已溢出.. **注:** 该位写 1 清 0 .
[23]	**TXFULL**	**发送 FIFO 满 (只读)** 此位用于指示 TX FIFO 是否已满..

		0 = TX FIFO 未满 1 = TX FIFO 已满. **注:** 当 TX FIFO Buffer 中的字节数量等于 16,此位将被置 1.否则被硬件清零.
[22]	**TXEMPTY**	**发送 FIFO 空 (只读)** 此位指示 TX FIFO 是否为空。 0 = TX FIFO 不为空. 1 = TX FIFO 为空. **注:** 当 TX FIFO 中的最后一个字节被发送到发送移位寄存器，硬件将把此位置 1.当写入数据到 UART_DAT 中, (TX FIFO 不为空)， 此位被清零
[21:16]	**TXPTR**	**TX FIFO 指针 (只读)** 此位指示 TX FIFO 缓冲区指针位置。当 CPU 写一个字节到 UART_DAT 寄存器，TXPTR 将累加 1.当 TX FIFO 发送一个字节到发送移位寄存器中，TXPTR 指针将减 1. TXPTR 显示的最大值是 15。当 TX FIFO 缓冲区所填充数据数量达到 16，TXFULL 将被置 1，TXPTR 显示为0，如果 TX FIFO 中发送了一个字节到发送移位寄存器，TXFULL 位将自动被清零，TXPTR 则显示为 15。
[15]	**RXFULL**	**接收 FIFO 满 (只读)** 该位指示 RX FIFO 是否已满。 0 = RX FIFO 未满 1 = RX FIFO 已满 **注:** 当 RX FIFO 缓冲区中的数据数量等于 16 后，此位将被置 1，否则被硬件清零。
[14]	**RXEMPTY**	**接收 FIFO 空 (只读)** 此位指示 RX FIFO 是否为空。 0 = RX FIFO 不为空. 1 = RX FIFO 为空. **注:** 当 RX FIFO 中最后一个字节被 CPU 读取后，硬件将对此位置 1，UART 接收到新数据后此位将被清零。
[13:8]	**RXPTR**	**RX FIFO 指针 (只读)** 此位指示 RX FIFO 缓冲区指针。当 UART 从外部设备接收到一个字节，RXPTR 将累加 1.当 RX FIFO 的数据被 CPU 读取一个字节，RXPTR 将递减 1. RXPTR 显示的最大值是 15。当 RX FIFO 的数据达到 16，RXFULL 位将被置 1，RXPTR 显示 0，而RX FIFO 当中的数据被 CPU 读取一个后，RXFULL 将被清零，RXPTR则显示为15。
[7]	**Reserved**	保留.
[6]	**BIF**	**Break 中断标志位 (只读)** 每当接收到的数据输入 (RX) 维持在 “空状态” (电平 0) 的时间长于一个全字的传输时间 (即开始位 + 数据位 + 校验位 + 停止位 的总时间)，该位置 1。当 CPU 向该位写 1，该位重置。 0 = 没有 Break 中断产生. 1 = 有 Break 中断产生. **注:** 该位写 1 清 0 .
[5]	**FEF**	**帧错误标志 (只读)** 每当接收到的字符没有有效的 “停止位”(即：跟在最后一位数据位或奇偶校验位后的数据为 0) 该位置 1。 0 = 没有帧错误产生 1 = 有帧错误产生.

		注: 该位写 1 清 0 .
[4]	**PEF**	**奇偶校验错误标志 (只读)** 每当接收到的字符没有有效的奇偶校验位，该位置 1。当 CPU 向该位写 1，该位重置。 0 = 没有奇偶校验错误产生 1 = 有奇偶校验错误产生 **注:** 该位写 1 清 0 .
[3]	**ADDRDETF**	**RS-485 地址位检测标志 (只读)** 0 = 接收到的数据没有地址位标记 (bit 9 ='0'). 1 = 接收到的数据有地址位标记 (bit 9 ='1'). **注 1:** 此位用于 RS-485 功能模式，且 ADDRDEN (UART_ALTCTL[15])) 位被置 1 使能地址检测模式 **注 2:** 该位写 1 清 0 .
[2]	**ABRDTOIF**	**自动波特率检测超时中断 (只读)** 0 = 自动波特率计数器没有溢出. 1 = 自动波特率计数器溢出. **注:** 该位写 1 清 0 .
[1]	**ABRDIF**	**自动波特率检测中断** 当自动波特率检测完成时该位将被置为 "1". 0 = 自动波特率检测还没有完成. 1 = 自动波特率检测已经完成 **注:** 该位写 1 清 0 .
[0]	**RXOVIF**	**RX 溢出错误标志 (只读)** 当 RX FIFO 溢出时被置 1 如果接收到的数据数量大于 RX_FIFO (UART_DAT) 16 字节，该位将被置位. 0 = RX FIFO 未溢出. 1 = RX FIFO 溢出 **注:** 该位写 1 清 0 .

UART中断状态寄存器(UART_INTSTS)

寄存器	偏移	R/W	描述	复位值
UART_INTSTS x=0,1,2,3,4,5,6,7	UARTx_BA+0x1C	R/W	UART 中断状态寄存器	0x0040_0002

31	30	29	28	27	26	25	24
ABRINT	**TXENDINT**	**HWBUFEINT**	**HWTOINT**	**HWMODINT**	**HWRLSINT**	**Reserved**	**SWBEINT**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**TXENDIF**	**HWBUFEIF**	**HWTOIF**	**HWMODIF**	**HWRLSIF**	**Reserved**	**SWBEIF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**WKINT**	**BUFERRINT**	**RXTOINT**	**MODEMINT**	**RLSINT**	**THREINT**	**RDAINT**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**WKIF**	**BUFERRIF**	**RXTOIF**	**MODEMIF**	**RLSIF**	**THREIF**	**RDAIF**

位	描述	
[31]	**ABRINT**	**自动波特率中断指示 (只读)** ABRIEN (UART_INTEN[18]) 和 ABRIF (UART_ALTCTL[17]) 都为 1，该位置 1 . 0 = 无自动波特率中断产生. 1 = 有自动波特率中断产生.
[30]	**TXENDINT**	**发送空中断指示器 (只读)** TXENDIEN (UART_INTEN[22]) 和 TXENDIF (UART_INTSTS[22]) 都为 1，该位置 1. 0 = 无发送空中断发生. 1 = 有发送空中断发生.
[29]	**HWBUFEINT**	**PDMA 模式缓存错误中断 (只读)** BUFERRIEN (UART_INTEN[5]) 和 HWBUFEIF (UART_INTSTS[21]) 都为 1，该位置 1. 0 = PDMA 模式下，无缓存错误中断发生. 1 = PDMA 模式下，有缓存错误中断发生.
[28]	**HWTOINT**	**PDMA 模式 RX 超时中断指示 (只读)** RXTOIEN (UART_INTEN[4]) 和 HWTOIF (UART_INTSTS[20]) 都为 1,该位被置 1. 0 = PDMA 模式时,无 RX 超时中断产生 1 = PDMA 模式时,有 RX 超时中断产生
[27]	**HWMODINT**	**PDMA 模式 MODEM 状态中断指示 (只读)** MODEMIEN (UART_INTEN[3]) 和 HWMODIF (UART_INTSTS[19]) 都为 1，该位置 1. 0 = PDMA 模式下，无 MODEM 状态中断发生. 1 = PDMA 模式下，有 MODEM 状态中断发生.
[26]	**HWRLSINT**	**PDMA 模式 RLS 中断指示 (只读)** RLSIEN (UART_INTEN[2]) 和 HWRLSIF (UART_INTSTS[18]) 都为 1，该位置 1. 0 = PDMA 模式下，无 RLS 中断发生.

		1 = PDMA 模式下，有 RLS 中断发生.
[25]	**Reserved**	保留.
[24]	**SWBEINT**	**单线位错误检测中断指示位（只读）** 如果SWBEIEN (UART_INTEN[16]) 和 SWBEIF (UART_INTSTS[16]) 都被置1，该位被置1. 0 = 没有单线位错误检测中断产生 1 = 单线位错误检测中断产生
[23]	**Reserved**	保留
[22]	**TXENDIF**	**发送空中断标志** 该位在 TX FIFO (UART_DAT) 空且最后一字节的停止位已发送出去 (TXEMPTYF (UART_FIFOSTS[28]) 置位) 时置位。如果使能了 TXENDIEN (UART_INTEN[22])，发送空中断触发 0 = 未触发发送空中断标志. 1 = 已触发发送空中断标志 **注:**该位在 TX FIFO 为非空或最后一字节的发送还未完成时由硬件自动清空
[21]	**HWBUFEIF**	**PDMA 模式缓存错误中断标志 (只读)** 当 TX 或 RX FIFO 溢出 (即 (TXOVIF (UART_FIFOSTS [24]) 或 RXOVIF (UART_FIFOSTS[0]) 被置 1),该位被置 1.当 BERRIF (UART_INTSTS[5]) 被置位,传送有可能出错. 如果 BFERRIEN (UART_INTEN [5]) 为使能,缓存错误中断将产生 0= PDMA 模式下,无缓存错误中断标志产生 1= PDMA 模式下,有缓存错误中断标志产生 **注**:当 TXOVIF (UART_FIFOSTS[24]) 和 RXOVIF (UART_FIFOSTS[0]) 都被清零时,该位也会被清零.
[20]	**HWTOIF**	**PDMA 模式 RX 超时中断标志 (只读)** 当 RX FIFO 不为空而且持续没有变化,当超时计数器达到 TOIC (UART_TOUT[7:0]) 时,该位被置 1.如果 RXTOIEN (UART_INTEN [4]) 为使能,超时溢出中断将产生 0= PDMA 模式下,无超时溢出中断标志产生 1= PDMA 模式下,有超时溢出中断标志产生 **注**:该位只读,用户可以通过读 UART_DAT (RX 为活动状态) 将该位清零.
[19]	**HWMODIF**	**PDMA 模式 MODEM 中断标志 (只读)** 当 nCTS 引脚状态发生变化, (CTSDETF (UART_CTSDETF[0] =1)) 时,该位被置 1.如果 MODEMIEN (UART_INTEN [3]) 位使能, Modem 中断将产生 0= PDMA 模式下,无 Modem 中断标志产生 1= PDMA 模式下,有 Modem 中断标志产生 **注**:该位只读,当写 1 到 CTSDETF (UART_CTSDETF [0]) 时, UART_CTSDETF (US_MSR[0]) 将被清零,该位也将被清零.
[18]	**HWRLSIF**	**PDMA 模式 RLS 中断标志 (只读)** 当 RX 接收数据出现校验错误、帧错误或 Break 错误,即至少 BIF (UART_FIFOSTS[6]),、FEF (UART_FIFOSTS[5]) 和或PEF (UART_FIFOSTS[4]) 中的一位为 1,该位被置 1. 如果 RLSIEN (UART_INTEN [2])) 为使能, RLS 中断将产生 0= PDMA 模式时,无 RLS 中断标志产生 1= PDMA 模式时,有 RLS 中断标志产生 **注1**: 在 RS-485 模式,该域包括"地址检测字节 (bit9 = '1')" **注2**: 在 UART 模式,该位只读,当 BIF (UART_FIFOSTS[6])、FEF (UART_FIFOSTS[5]) 和 PEF (UART_FIFOSTS[4]) 都被清零时,该位也将被清零. **注 3**: 在 RS-485 模式,该位只读,当 BIF (UART_FIFOSTS[6]) 、 FEF

		(UART_FIFOSTS[5])、PEF (UART_FIFOSTS[4]) 和 ADDRDETF (UART_FIFOSTS[3]) 都被清零时,该位也将被清零.
[17]	**Reserved**	保留
[16]	**SWBEIF**	**单线位错误检测中断标志** 在单线模式下，单线总线状态不与UART控制器同处于TX 状态时，该位置1. 0 = 没有单线位错误检测中断标志产生. 1 = 单线位错误检测中断标志产生 **注1:** 当 FUNCSEL (UART_FUNCSEL[2:0]) 选择为UART 单线模式，该位激活. **注2:** 写1清除该位
[15]	**Reserved**	保留.
[14]	**WKINT**	**UART 唤醒中断指示 (只读)** WKIEN (UART_INTEN[6]) 和 WKIF (UART_INTSTS[6])]) 都为 1，该位置 1. 0 = 无 UART 唤醒中断发生. 1 = 有 UART 唤醒中断发生.
[13]	**BUFERRINT**	**缓存错误中断指示 (只读)** BUFERRIEN (UART_INTEN[5]) 和 BUFERRIF (UART_ INTSTS[5]) 都为 1，该位置 1. 0 = 无缓存错误中断发生. 1 = 有缓存错误中断发生.
[12]	**RXTOINT**	**RX 超时中断指示 (只读)** RXTOIEN (UART_INTEN[4]) 和 RXTOIF (UART_INTSTS[4]) 都为 1，该位置 1. 0 = 无 RX 超时中断发生. 1 = 有 RX 超时中断发生.
[11]	**MODEMINT**	**MODEM 状态中断 (只读)** MODEMIEN (UART_INTEN[3]) 和 MODEMIF (UART_INTSTS[3]) 都为 1，该位置 1. 0 = 无 MODEM 状态中断发生. 1 = 有 MODEM 状态中断发生.
[10]	**RLSINT**	**RLS 中断指示 (只读)** RLSIEN (UART_INTEN[2]) 和 RLSIF (UART_INTSTS[2]) 都为 1，该位置 1. 0 = 无 RLS 中断发生. 1 = 有 RLS 中断发生.
[9]	**THREINT**	**发送保持寄存器空中断指示 (只读)** THREIEN (UART_INTEN[1]) 和 THREIF (UART_INTSTS[1]) 都为 1，该位置 1. 0 = 无发送保持寄存器空中断发生. 1 = 有发送保持寄存器空中断发生.
[8]	**RDAINT**	**RDA 中断指示 (只读)** RDAIEN (UART_INTEN[0]) 和 RDAIF (UART_INTSTS[0]) 都为 1，该位置 1. 0 = 无 RDA 中断发生. 1 = 有 RDA 中断发生.
[7]	**Reserved**	保留.
[6]	**WKIF**	**UART 唤醒中断标志 (只读)** TOUTWKF (UART_WKSTS[4]) 、 RS485WKF (UART_WKSTS[3]) 、 RFRTWKF (UART_WKSTS[2])、DATWKF (UART_WKSTS[1]) 或 CTSWKF (UART_WKSTS[0])])

		为 1，该位置 1. 0 = 无 UART 唤醒中断标志. 1 = 有 UART 唤醒中断标志. **注:** 写 1到TOUTWKF, RS485WKF, RFRTWKF, DATWKF 和 CTSWKF清0后，该位清 0.
[5]	**BUFERRIF**	**缓存错误中断标志 (只读)** TX FIFO 或 RX FIFO 溢出 (TXOVIF (UART_FIFOSTS[24]) 或 RXOVIF (UART_FIFOSTS[0]) 置位)，该位置位. 传输错误时，BUFERRIF (UART_INTSTS[5]) 置位. 如果使能了 BUFERRIEN (UART_INTEN [5]), 会产生缓存错误中断 0 = 无缓存错误中断产生. 1 = 有缓存错误中断产生. **注:** 写 1 到 RXOVIF (UART_FIFOSTS[0]) 和 TXOVIF (UART_FIFOSTS[24]) 清空 RXOVIF (UART_FIFOSTS[0]) 和 TXOVIF (UART_FIFOSTS[24]) 后，该位清 0
[4]	**RXTOIF**	**RX 超时中断标志 (只读)** 如果 RX FIFO 为非空且无其他事件时，当超时计数器计数到 TOIC (UART_TOUT[7:0]) 时，该位置位。此时如果使能了 RXTOIEN (UART_INTEN [4])，会产生 RX 超时中断。 0 = 无 RX 超时中断标志产生. 1 = 有 RX 超时中断标志产生.. **注:** 该位只读，读 UART_DAT (RX 有效时) 清 0 该位.
[3]	**MODEMIF**	**MODEM 中断标志 (只读)** 如果 nCTS 引脚电平发生改变 (CTSDETF (UART_MODEMSTS[0]) = 1)，该位置位。此时如果使能了 MODEMIEN (UART_INTEN [3])，会产生 Modem 中断 0 = 无 Modem 中断标注产生. 1 = 有 Modem 中断标注产生. **注:** 该位只读，写 1 到 CTSDETF (UART_MODEMSTS[0]) 清 0 CTSDETF 后，该位清 0
[2]	**RLSIF**	**RLS 中断标志 (只读)** 当 RX 接收数据出现校验错误、帧错误或 Break 错误,即至少 BIF (UART_FIFOSTS[6])、FEF (UART_FIFOSTS[5]) 和 PEF (UART_FIFOSTS[4]) 中的一位为 1,该位被置 1. 如果 RLSIEN (UART_INTEN [2])) 为使能, RLS 中断将产生 0= 无 RLS 中断标志产生 1= 有 RLS 中断标志产生 **注1:** 在 RS-485 模式, 当“接收检测任一接收到的地址字节字符 (bit9 = ‘1’) 位”时， 该位也会置1，同时， ADDRDETF (UART_FIFOSTS[3])位也被置1 **注 2:** 该位只读,当 BIF (UART_FIFOSTS[6]) 、FEF (UART_FIFOSTS[5]) 和 PEF (UART_FIFOSTS[4]) 都被清零时,该位也将被清零. **注 3:** 在 RS-485 模式,该位只读,当 BIF (UART_FIFOSTS[6]) 、 FEF (UART_FIFOSTS[5]) 、 PEF (UART_FIFOSTS[4]) 和 ADDRDETF (UART_FIFOSTS[3]) 都被清零时,该位也将被清零.
[1]	**THREIF**	**发送保持寄存器空中断标志 (只读)** 当 TX FIFO 中最后数据传输到发送移位寄存器时，该位置位。如果 THREIEN (UART_INTEN[1]) 使能，THRE 中断将产生。 0 =没有 THRE 中断标志产生 1 =有 THRE 中断标志产生. **注:** 该位只读，当写数据到 UART_DAT (TX FIFO 非空) 时，该位将被清除
[0]	**RDAIF**	**接收数据可用中断标志 (只读)** 当 RX FIFO 中的数据数量等于 RFITL 时，RDAIF (UART_INTSTS[0]) 将被置位。如果 RDAIEN (UART_INTEN [0]) 使能，RDA 中断将产生。

		0 =没有 RDA 中断标志产生. 1 =有 RDA 中断标志产生. **注:** 该位只读，当 RX FIFO 的未读数据低于阈值 (RFITL (UART_FIFO[7:4]) 时，该位将被清除

UART 超时寄存器(UART_TOUT)

寄存器	偏移	R/W	描述	复位值
UART_TOUT x=0,1,2,3,4,5,6,7	UARTx_BA+0x20	R/W	UART超时寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:8]	**DLY**	**TX 延迟时间值** 该域用于设置上一次停止位和下一次开始位之间的传输延迟时间
[7:0]	**TOIC**	**超时中断比较器** 使能 TOCNTEN (UART_INTEN[11]) 超时计数功能后，每当 RX FIFO 接收到一个新的数据字，定时溢出计数器都会重置并开始计数 (计数时钟 = 波特率)。一旦超时计数器的内容等于超时中断比较器 (TOIC (UART_TOUT[7:0]))，如果此时 RXTOIEN (UART_INTEN [4]) 使能，则接收超时中断 (RXTOINT (UART_INTSTS[12])) 产生。接收到新的数据或 RX FIFO 为空将把RXTOIF (UART_INTSTS[4])清零。为了避免接收超时中断在接收到一个字符就立即产生，TOIC 的值必须设置在 40 到 255 之间。例如，如果 TOIC 为 40，当 UART 传输设置为 1 位停止位且无奇偶校验位时，在 4 个字符时间长度后还没收到数据，超时中断将产生。

UART 波特率分频寄存器(UART_BAUD)

寄存器	偏移	R/W	描述	复位值
UART_BAUD x=0,1,2,3,4,5,6,7	UARTx_BA+0x24	R/W	UART 波特率分频寄存器	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		**BAUDM1**	**BAUDM0**	**EDIVM1**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BRD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRD							

位	描述	
[31:30]	**Reserved**	保留.
[29]	**BAUDM1**	**波特率模式选择位 1** 该位为波特率模式选择位 1。UART 提供三种波特率计算方式。 该位和 BAUDM0 (UART_BAUD[28]) 组合选择波特率计算方式。 详细描述在表 6.13-4中有介绍 **注：**IrDA 模式时必须选择模式 0
[28]	**BAUDM0**	**波特率模式选择位 0** 该位为波特率模式选择位 0。UART 提供三种波特率计算方式。 该位和BAUDM1 (UART_BAUD[29]) 组合选择波特率计算方式。详细描述在表 6.13-4中有介绍.
[27:24]	**EDIVM1**	**波特率模式 1 的扩展分频** 该域用于波特率计算模式 1，对于波特率计算模式 0 和模式 2 无效。详细描述在表 6.13-4 中有介绍.
[23:16]	**Reserved**	保留.
[15:0]	**BRD**	**波特率分频** 该域用于波特率分频。详细描述在表 6.13-4中有介绍.

UART IrDA 控制寄存器(UART_IRDA)

寄存器	偏移	R/W	描述	复位值
UART_IRDA x=0,1,2,3,4,5,6,7	UARTx_BA+0x28	R/W	UART IrDA控制寄存器	0x0000_0040

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

位	描述	
[31:7]	**Reserved**	保留.
[6]	**RXINV**	**IrDA 接收信号反转** 0 = 不反转接收信号. 1 = 反转接收信号 (默认) **注 1:** 设置该位前，需要置位 TXRXDIS (UART_FUNCSEL[3]) 再等 TXRXACT (UART_FIFOSTS[31]) 清除。配置完成后，清除 TXRXDIS (UART_FUNCSEL[3]) 以启用 UART 控制器 **注2:** 该位在 FUNCSEL (UART_FUNCSEL[1:0]) 选择为 IrDATA 功能时有效
[5]	**TXINV**	**IrDA 发送信号反转** 0 = 不反转发送信号. 1 = 反转发送信号 (默认) **注 1:** 设置该位前，需要置位 TXRXDIS (UART_FUNCSEL[3]) 再等 TXRXACT (UART_FIFOSTS[31]) 清除。配置完成后，清除 TXRXDIS (UART_FUNCSEL[3]) 以启用 UART 控制器 **注2:** 该位在 FUNCSEL (UART_FUNCSEL[1:0]) 选择为 IrDATA 功能时有效
[4:2]	**Reserved**	保留.
[1]	**TXEN**	**IrDA 接收/发送选择使能位** 0 = IrDA 禁止发送和使能接收(默认) 1 = IrDA 使能发送和禁止接收
[0]	**Reserved**	保留.

UART选择控制/状态 寄存器(UART_ALTCTL)

寄存器	偏移	R/W	描述	复位值
UART_ALTCTL x=0,1,2,3,4,5,6,7	UARTx_BA+0x2C	R/W	UART 选择控制/状态寄存器	0x0000_000C

31	30	29	28	27	26	25	24
ADDRMV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**ABRDBITS**		**ABRDEN**	**ABRIF**	**Reserved**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRDEN	**Reserved**				**RS485AUD**	**RS485AAD**	**RS485NMM**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LINTXEN	**LINRXEN**	**Reserved**		**BRKFL**			

位	描述	
[31:24]	**ADDRMV**	**地址匹配值** 该位包括 RS-485 地址匹配值. **注：**该域用于 RS-485 自动地址检测模式
[23:21]	**Reserved**	保留
[20:19]	**ABRDBITS**	**自动波特率检测位长** 00 = 1-位时长，从起始位到第一个 上升沿。 输入数据格式应该为 0x01。 01 = 2-位时长，从起始位到第一个 上升沿。 输入数据格式应该为 0x02。 10 = 4-位时长，从起始位到第一个 上升沿。 输入数据格式应该为 0x08。 11 = 8-位时长，从起始位到第一个 上升沿。 输入数据格式应该为 0x80。 **注:**计算的位数包括起始位。
[18]	**ABRDEN**	**自动波特率检测使能位** 0 = 禁止自动波特率检测功能. 1 = 使能自动波特率检测功能 **注：**自动检测结束后，该位将被自动清零。
[17]	**ABRIF**	**自动波特率中断标志 (只读)** 当自动波特率检测结束，或者自动波特率计数器发生溢出，如果 ABRIEN (UART_INTEN[18]) 为 1，那么自动波特率中断将会产生. **注：**该位为只读位，但可以通过写“1”到 ABRDTOIF (UART_FIFOSTS[2]) 和 ABRDIF (UART_FIFOSTS[1]) 将该位清零。
[16]	**Reserved**	保留
[15]	**ADDRDEN**	**RS-485 地址检测使能位** 该位用于使能 RS-485 地址检测模式. 0 =禁止地址检测模式.

		1 =使能地址检测模式. **注：**该位适用于 各种 RS-485 操作模式
[14:11]	**Reserved**	保留
[10]	**RS485AUD**	**RS-485 自动方向模式 (AUD)** 0 =RS-485 自动方向操作模式 (AUD) 禁止 1 =RS-485 自动方向操作模式 (AUD) 使能 **注：**仅在 RS-485_AAD 或 RS-485_NMM 操作模式有效
[9]	**RS485AAD**	**RS-485 自动地址检测操作模式 (AAD)** 0 =禁止RS-485 自动地址检测模式 (AAD) 1 =使能RS-485 自动地址检测模式 (AAD) **注：**在 RS-485_NMM 操作模式下无效
[8]	**RS485NMM**	**RS-485 普通多点操作模式 (NMM)** 0 =禁止RS-485 普通多点操作模式 (NMM) 1 =使能RS-485 普通多点操作模式 (NMM) **注：**在 RS-485_AAD 操作模式下无效.
[7]	**LINTXEN**	**LIN TX Break 模式使能** 0 = 禁止LIN TX Break 模式 1 = 使能LIN TX Break 模式 **注：**当 TX 间隔域传输操作完成后，该位将被自动清除.
[6]	**LINRXEN**	**LIN RX 使能** 0 = 禁止LIN RX 模式 1 = 使能LIN RX 模式
[5:4]	**Reserved**	保留
[3:0]	**BRKFL**	**UART LIN 间隔域长度** 该域表示一个 4-位 LIN TX 间隔域数量 **注1:** 该间隔域长度为 BRKFL + 1 **注2:** 根据 LIN 规范, 复位值是 0xC (间隔域长度= 13).

UART功能选择寄存器(UART_FUNCSEL)

寄存器	偏移	R/W	描述	复位值
UART_FUNCSEL **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x30	R/W	UART功能选择寄存器	0x0000_0000

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

位	描述	
[31:4]	**Reserved**	保留.
[3]	**TXRXDIS**	**TX 和 RX 禁止位** 置位该位以禁止 TX 和 RX. 0 = TX 和 RX 使能. 1 = TX 和 RX 禁止. **注:** 该位置位后，TX 和 RX 不会立刻关闭。TX 和 RX 在完成当前任务后再关闭。关闭 TX 和 RX 后，TXRXACT (UART_FIFOSTS[31]) 清 0
[2:0]	**FUNCSEL**	**功能选择** 000 = UART 功能. 010 = IrDA 功能. 011 = RS-485 功能. 100 = UART 单线 功能. 其它 = 保留.

UART 波特率补偿寄存器(UART_BRCOMP)

寄存器	偏移	R/W	描述	复位值
UART_BRCOMP **x=0,1,3,5**	UARTx_BA+0x3C	R/W	UART 波特率补偿寄存器	0x0000_0000

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

位	描述	
[31]	**BRCOMPDEC**	**波特率补偿方向** 0 = 对应位正向 (加 1 个模块时钟) 补偿 1 = 对应位负向 (减 1 个模块时钟) 补偿.
[30:9]	**Reserved**	保留
[8:0]	**BRCOMP**	**波特率补偿模式** 这 9 位用于定义相关的位是否被补偿，BRCOMP[7:0] 用来定义 UART_DAT[7:0] 的补偿，BRCOM[8] 则是校验位

UART 唤醒控制寄存器(UART_WKCTL)

寄存器	偏移	R/W	描述	复位值
UART_WKCTL x=0,1,2,3,4,5,6,7	UARTx_BA+0x40	R/W	UART 唤醒控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**WKTOUTEN**	**WKRS485EN**	**WKRFRTEN**	**WKDATEN**	**WKCTSEN**

位	描述	
[31:5]	**Reserved**	保留.
[4]	**WKTOUTEN**	**数据接收 FIFO 达到阈值超时唤醒系统功能使能位** 0 =数据接收 FIFO 达到阈值超时唤醒系统功能禁止 1 = 数据接收 FIFO 达到阈值超时唤醒系统功能使能, **注1:**当系统在power-down模式，接收数据FIFO达到阈值超时将把系统从power-down状态唤醒。 **注2:** WKRFRTEN (UART_WKCTL[2]) 置 1 时，建议使能该位 **注3:** UART0 ，UART1，UART4和 UART5此位有效.
[3]	**WKRS485EN**	**RS-485 地址匹配 (AAD 模式) 唤醒使能位** 0 = RS-485 地址匹配 (AAD 模式) 唤醒系统功能禁止. 1 = RS-485 地址匹配 (AAD 模式) 唤醒系统功能使能, **注1:**当系统在power-down模式，RS-485 地址匹配将把系统从power-down状态唤醒。 **注2:** 该位在 RS-485 自动地址检测 (AAD 模式) ，ADDRDEN (UART_ALTCTL[15]) 置 1 时使用 **注3:** UART0 ，UART1，UART4和 UART5此位有效.
[2]	**WKRFRTEN**	**数据接收 FIFO 达到阈值唤醒使能位** 0 = 数据接收 FIFO 达到阈值唤醒系统功能禁止. 1 = 数据接收 FIFO 达到阈值唤醒系统功能使能， **注1：** 数据接收 FIFO 达到阈值唤醒系统功能可以将处于掉电模式的系统唤醒 **注2:** UART0 ，UART1，UART4和 UART5此位有效.
[1]	**WKDATEN**	**输入数据唤醒使能位** 0 = 输入数据唤醒系统功能禁止. 1 = 输入数据唤醒系统功能使能， **注：** 输入数据唤醒系统功能可以将处于掉电模式的系统唤醒

[0]	**WKCTSEN**	**nCTS 唤醒使能位** 0 = nCTS 唤醒系统功能禁止 1 = nCTS 唤醒系统功能使能， **注：**当系统处于掉电模式，外部的nCTS变化可以将处于掉电模式的系统唤醒

UART 唤醒状态寄存器(UART_WKSTS)

寄存器	偏移	R/W	描述	复位值
UART_WKSTS x=0,1,2,3,4,5,6,7	UARTx_BA+0x44	R/W	UART 唤醒状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**TOUTWKF**	**RS485WKF**	**RFRTWKF**	**DATWKF**	**CTSWKF**

位	描述	
[31:5]	**Reserved**	保留
[4]	**TOUTWKF**	**接收数据 FIFO 阈值超时唤醒标志** 掉电的芯片被接收数据 FIFO 阈值超时唤醒时，该位置位 0 = 芯片处于掉电模式. 1 = 掉电的芯片被接收数据 FIFO 阈值超时唤醒. **注1:** WKTOUTEN (UART_WKCTL[4]) 使能, 接收数据 FIFO 阈值超时唤醒导致该位置 1 . **注2:** 该位写 1 清 0. **注3:** UART0，UART1，UART4 和 UART5此位有效.
[3]	**RS485WKF**	**RS-485 地址匹配 (AAD 模式) 唤醒标志** 掉电的芯片被 RS-485 地址匹配 (AAD 模式) 唤醒时，该位置位 0 = 芯片处于掉电模式. 1 = 掉电的芯片被 RS-485 地址匹配 (AAD 模式) 唤醒 **注1:** WKRS485EN (UART_WKCTL[3]) 使能, RS-485 地址匹配 (AAD 模式) 唤醒导致该位置 1 **注2:** 该位写 1 清 0. **注3:** UART0，UART1，UART4 和 UART5此位有效.
[2]	**RFRTWKF**	**接收数据 FIFO 达到阈值唤醒标志** 掉电的芯片被接收数据 FIFO 达到阈值唤醒时，该位置位 0 = 芯片处于掉电模式. 1 = 掉电的芯片被接收数据 FIFO 达到阈值唤醒 **注 1:** WKRFRTEN (UART_WKCTL[2]) 使能, 接收数据 FIFO 达到阈值唤醒导致该位置 1 **注 2:** 该位写 1 清 0. **注3:** UART0，UART1，UART4 和 UART5此位有效.

[1]	**DATWKF**	**输入数据唤醒标志** 掉电的芯片被输入数据唤醒时，该位置位. 0 = 芯片处于掉电模式. 1 = 掉电的芯片被输入数据唤醒. **注1:** WKDATEN (UART_WKCTL[1]) 使能，输入数据唤醒导致该位置 1 **注2:** 该位写 1 清 0.
[0]	**CTSWKF**	**nCTS 唤醒标志** 掉电的芯片被nCTS唤醒时，该位置 1 . 0 = 芯片处于掉电模式. 1 = 掉电的芯片被 nCTS 唤醒. **注1:** WKCTSEN (UART_WKCTL[0]) 使能，nCTS 唤醒导致该位置 1 **注2:** 该位写 1 清 0.

UART 输入数据唤醒补偿寄存器(UART_DWKCOMP)

寄存器	偏移	R/W	描述	复位值
UART_DWKCOMP **x=0,1,2,3,4,5,6,7**	UARTx_BA+0x48	R/W	UART 输入数据唤醒补偿寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**STCOMP**	**开始位补偿值** 这些位代表在芯片从掉电模式被唤醒到 UART 控制器可以接收第一个开始位需要的 UART_CLK 数目。 **注:** 仅在 WKDATEN (UART_WKCTL[1]) 置位时有效.

6.14 SPI接口(SPI)

6.14.1 概述

SPI接口是全双工同步串行数据通讯接口，可做为主机或从机，用 4 线双向通讯。M031 包含 1 组 SPI 控制器，用于对从外围设备接收到的数据执行串并转换，并对发送到外围设备的数据进行并串转换。 每个SPI控制器可以配置为主设备或从设备，并支持PDMA功能存取数据缓冲区。 每个SPI控制器还支持I2S模式来连接外部音频编解码器。

6.14.2 特征

- SPI 模式
  - 1组 SPI 控制器
  - 支持主机模式和从机模式
  - 传输位长可为 8 ~ 32位
  - 提供独立的4级32位（或8级16位）收发FIFO缓存，实际数据位长由SPI的设置决定
  - 支持高位优先（MSB）或低位优先(LSB)时序
  - 支持字节重排功能
  - 支持字节或字暂停模式
  - 总线时钟主机模式最高可到24 MHz ，从机模式最高可到16 MHz (当芯片工作在 V_{DD} = 1.8~3.6V)
  - 支持一数据通道半双工传输
  - 支持只接收模式
  - 支持 PDMA 传输
- I^2S 模式
  - 通过SPI 控制器支持一路 I^2S
  - 外接音频 CODEC
  - 支持主机/从模式
  - 可处理 8，16，24 和 32 位数据宽度
  - 支持单声道和双声道格式
  - 支持 PCM A，PCM B，I²S 和 MSB 对齐的数据格式
  - 每组控制器提供 2 个 4 级 FIFO 缓存，一个用于发送，另一个用于接收
  - 支持两路PDMA 请求，一个用于发送，另一个用于接收
  - 当缓存数量超过边界值（边界值可编程设定），将产生中断。

6.14.3 框图

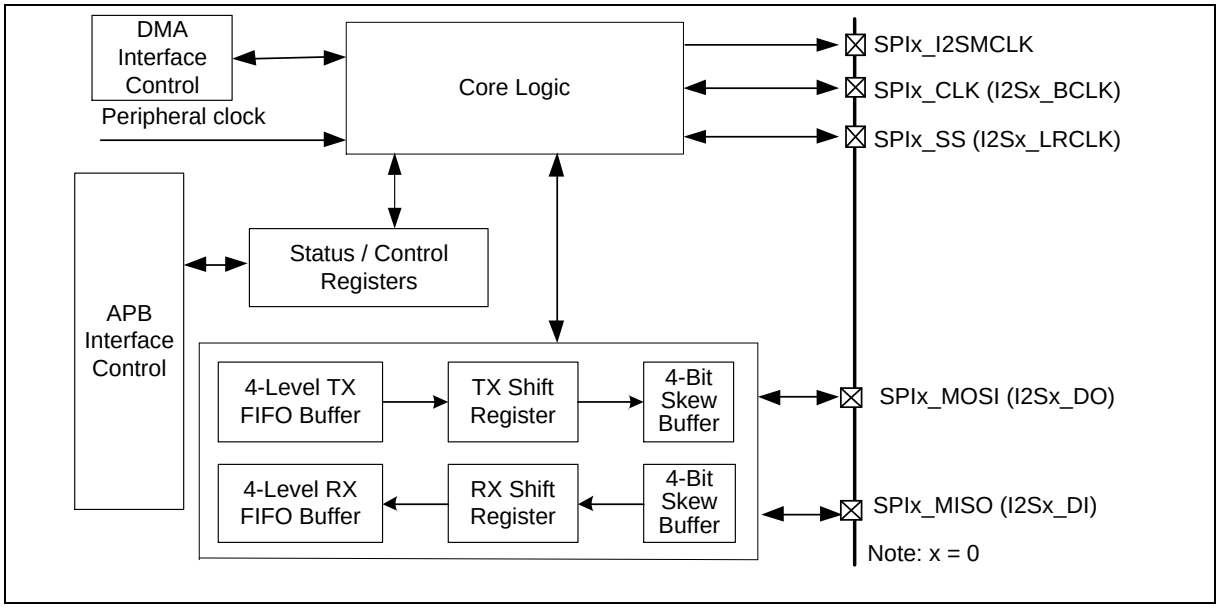


图 6.14-1 SPI 框图

TX FIFO 缓存:

发送 FIFO为 4 级深度、32 位宽、先进先出的缓存寄存器。数据写入寄存器 SPIx_TX 就是写入发送 FIFO。SPI 模式，如果数据长度为 8~16 位，则发送 FIFO 可以设为 8 级深度。

RX FIFO 缓存:

接收 FIFO 为4 级深度、32 位宽、先进先出的缓存寄存器。接收控制会把数据存到该缓存。软件可通过读 SPIx_RX 得到该 FIFO 数据。SPI 模式时，如果数据长度为 8~16 位，接收 FIFO 可以设为 8 级深度。

TX 移位寄存器:

发送移位寄存器是一个 32 位的寄存器。发送数据从 TX FIFO 缓存加载到这里，并一位一位的移到斜移缓存中。

RX 移位寄存器:

接收移位寄存器是一个 32 位的寄存器。接收数据从斜移缓存一位一位的移到这里，当一个事务完成时再载入到 RX FIFO。

斜移缓存:

斜移缓存是一个 4 级 1 位的缓存。总共有 2 个斜移缓存，分别在发送端和接收端。在接收端，斜移缓存用于把数据从 SPI 总线移位到 Rx 移位寄存器。在发送端，斜移缓存用于把数据从 Tx 移位寄存器移位到 SPI 总线。

6.14.4 基本配置

6.14.4.1 SPI0基本配置

- 时钟配置
  - 寄存器 SPI0SEL (CLK_CLKSEL2[5:4]) 配置 SPI0 的时钟

- 寄存器 SPI0CKEN (CLK_APBCLK0[13]) 使能 SPI0 的时钟

- 复位配置
  - 寄存器 SPI0RST (SYS_IPRST1[13]) 复位 SPI0 控制器

SPI/I^{2}S (SPI0) 接口控制器引脚描述如下:

引脚	SPI 模式描述	I^{2}S 模式描述
SPIx_SS	SPI 从机片选	I^{2}S 左右通道时钟 (I2Sx_LRCLK)
SPIx_CLK	SPI 时钟	I^{2}S 位时钟 (I2Sx_BCLK)
SPIx_MISO	SPI 主机输入或从机输出	I^{2}S 数据输入 (I2Sx_DI)
SPIx_MOSI	SPI 主机输出或从机输入	I^{2}S 数据输出 (I2Sx_DO)
SPIx_I2SMCLK	不可用	I^{2}S 主时钟输出

表 6.14-1 SPI/I^{2}S 接口引脚描述(SPI0)

6.14.5 功能描述

6.14.5.1 术语

SPI 外设时钟和 SPI 总线时钟

SPI控制器需要外设时钟来驱动SPI逻辑单元来执行数据传输。SPI外设时钟速率决定于时钟分频器(SPIx_CLKDIV)和时钟源的设置，时钟源可以设置为HXT、HIRC、PLL输出时钟或PCLK。寄存器CLK_CLKSEL2的SPIxSEL位决定SPI外设时钟的时钟源。寄存器DIVIDER (SPIx_CLKDIV[8:0])的设定值决定时钟速率计算的分频值。

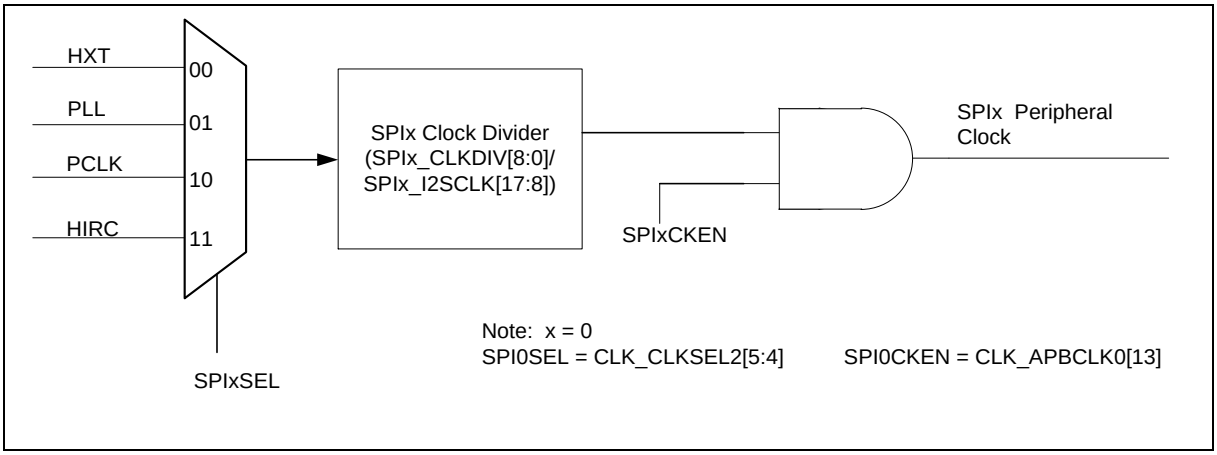


图 6.14-2 SPI 外设时钟

在主机模式下，SPI总线的时钟频率等于外设的时钟速率。在通常情况下，SPI总线时钟表示为SPI时钟。在从机模式下，SPI总线时钟由片外主机设备提供。不论工作于主机还是从机模式，SPI外设的时钟频率不能快于系统时钟速率。如果外设的时钟源不同于其中的一个系统时钟，SPI外设时钟不管是主机模式还是从机模式都必须不能超过系统时钟频率。

在I2S模式下，外设时钟速率等于I2S的位时钟速率，I2S的位时钟速率由SPIx_I2SCLK寄存器设定。

主机/从机模式

SPI控制器可通过设置SLAVE(SPIx_CTL[18])配置成主机或从机模式，与片外的SPI从机或者主机进行通信。在SPI传输中，HALFDPX (SPIx_CTL[14]) 用于选择全双工或半双工。主机模式与从机模式的应用框图如下：

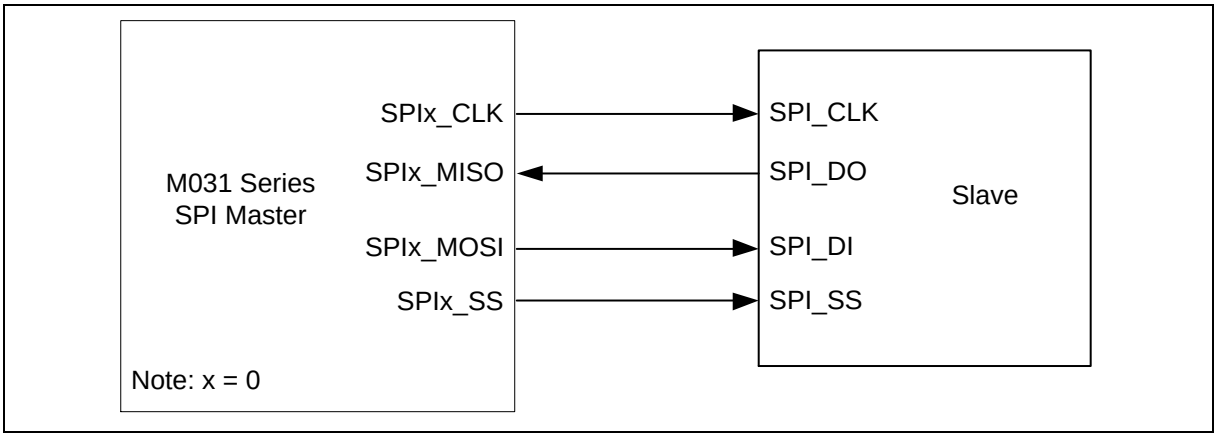


图 6.14-3 SPI 全双工主机模式应用框图

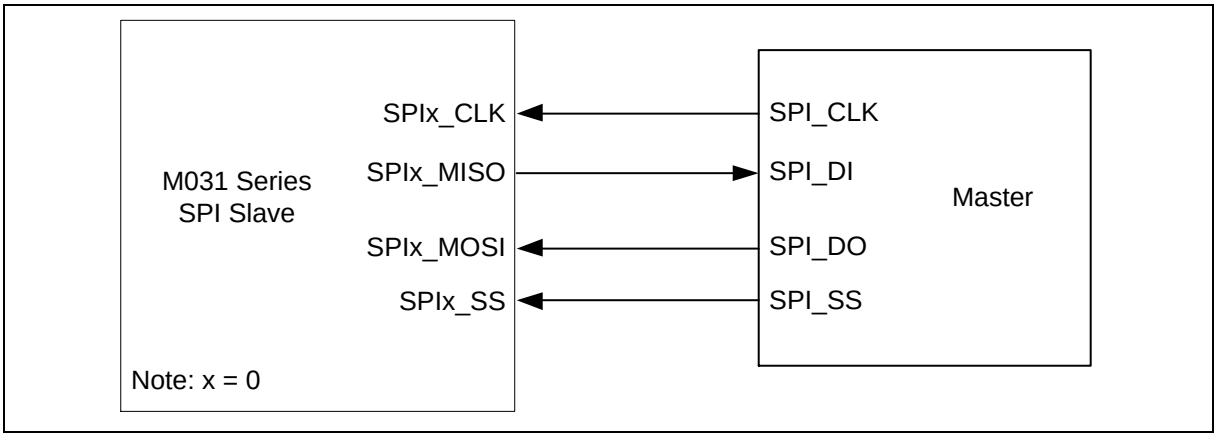


图 6.14-4 SPI 全双工从机模式应用框图

从机选择

在主机模式下，该SPI控制器能通过从机片选输出脚SPIx_SS来驱动片外从机设备。在从机模式，片外主机设备通过SPIx_SS输入端口驱动从机片选信号到SPI控制器。在从机片选激活边沿到第一个SPI时钟输入应多于3个SPI从机外设时钟周期。

在主机/从机模式下，从机选择信号的有效状态可以通过修改寄存器SSACTPOL(SPIx_SSCTL[2])来设定为低有效或高有效。从机片选条件取决于所连设备的类型。在从机模式下，为了区别从机片选信号的无效状态，在两次数据传输之间，从机片选信号的无效周期必须大于或等于3个外设时钟周期。

时序条件

CLKPOL (SPIx_CTL[3])定义了SPI时钟线空闲时的状态。如果CLKPOL = 1，SPI时钟线在空闲时输出高电平，如果CLKPOL = 0，则SPI时钟线在空闲时输出低电平。

TXNEG (SPIx_CTL[2])定义了数据是在SPI时钟的下降沿还是上升沿传输出去。RXNEG (SPIx_CTL[1])定义了数据在SPI时钟的下降沿还是上升沿接收。

注意:TXNEG和RXNEG的设置是相斥的，换话说就是发送和接收数据的时钟沿设置不能相同。

传输/接收位长度

数据字的位长由DWIDTH (SPIx_CTL[12:8])来定义。对于发送和接收，数据字的位长可配置，最多可达32位。

当SPI控制器完成一个数据字传输时，例如：接收或发送完DWIDTH(SPIx_CTL[12:8])中定义的数据位长的数据字时，传输中断标志将被置1。

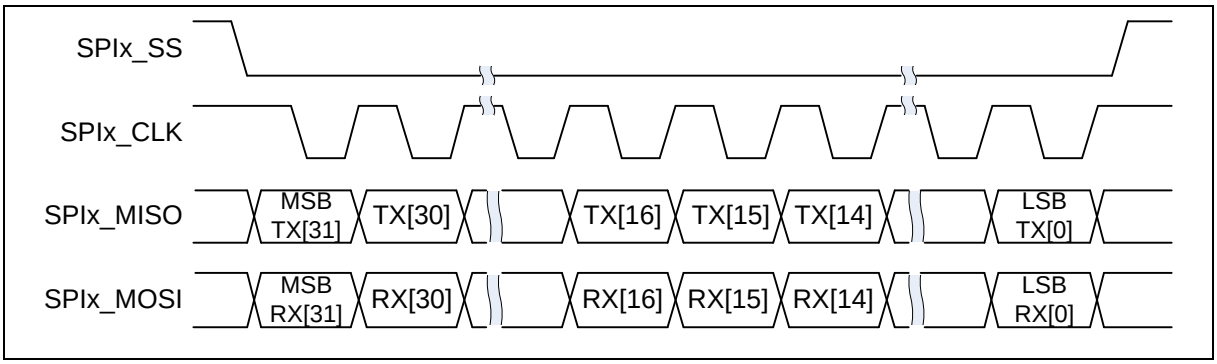


图 6.14-5 在一个数据字 32-位长度

LSB/MSB 优先

LSB (SPIx_CTL[13])定义一个数据字中位传输顺序。如果设定LSB(SPIx_CTL[13])为1，传输顺序为LSB优先。第0位首先将被传输。如果清LSB(SPIx_CTL[13])为0，传输顺序是MSB优先

休眠间隔

在主机模式下，SUSPITV (SPIx_CTL[7:4])提供在两个连续数据字之间，可配置为0.5～15.5个SPI时钟周期的休眠间隔。休眠间隔是指从前一个数据字的最后一个时钟沿到下一个数据字的第一个时钟沿的时间间隔。SUSPITV的默认值是0x3(3.5 SPI总线时钟周期)。

6.14.5.2 自动从机选择

在主机模式下，如果AUTOSS(SPIx_SSCTL[3])设置为1，从机选择信号将会自动产生，并根据SS (SPIx_SSCTL[0])是否使能，将从机选择信号输出到SPIx_SS管脚上。当SPI把数据写入FIFO，启动数据传输时，从机选择信号将由SPI控制器自动设置为有效状态。当SPI总线空闲时，从机选择信号将变为无效状态。如果SPI总线不是空闲的，例如：发送FIFO，发送移位寄存器或发送斜移缓存不空，如果SUSPITV (SPIx_CTL[7:4])的值大于或等于3，从机选择信号在两次数据字传输之间将被设置为无效状态。

在主机模式下，如果SUSPITV的值小于3且AUTOSS被设置为1，在两个连续数据字传输之间，从机选择信号将保持有效。

如果AUTOSS被清零时，从机选择输出信号将由通过SS设置决定。从机选择输出信号的激活电平由SSACTPOL (SPIx_SSCTL[2])来定义。

从机片选信号有效边沿与SPI总线的第一个时钟边沿间隔时间为1个SPI总线时钟周期。SPI总线的最后一个时钟与从机选择信号无效边沿的间隔时间为1.5个SPI总线时钟周期。

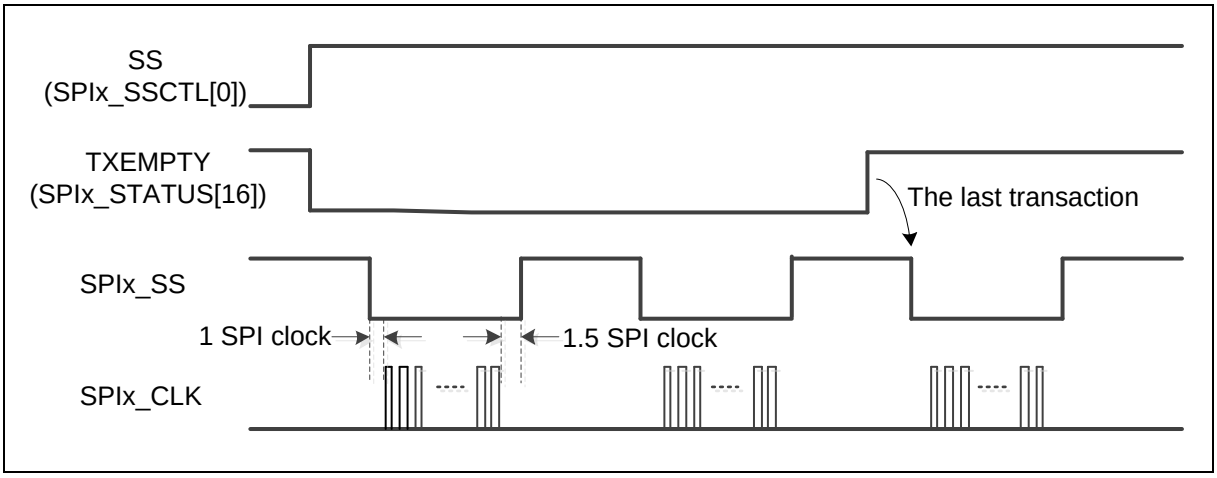


图 6.14-6 自动从机选择(SSACTPOL = 0, SUSPITV > 0x2)

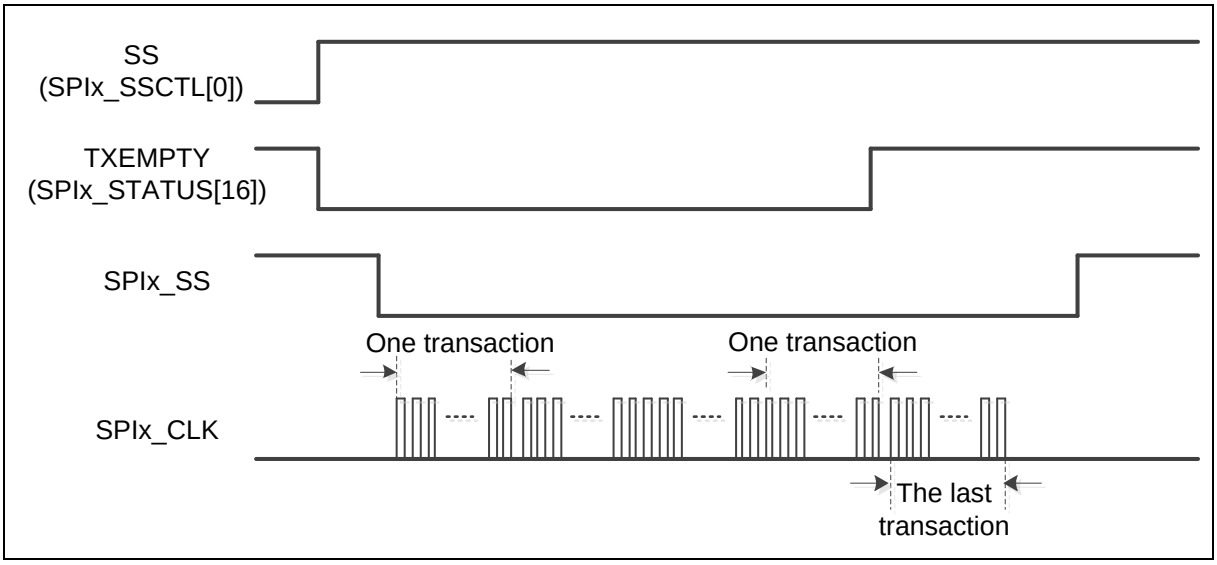


图 6.14-7 自动从机选择(SSACTPOL = 0, SUSPITV < 0x3)

6.14.5.3 重排序和休眠功能

当传输设置为 MSB 优先 (LSB = 0) 且使能REORDER (SPIx_CTL[19])，在32-位模式(DWIDTH = 0)时，存储在 TX 缓存与 RX 缓存的数据将重新按 [BYTE0, BYTE1, BYTE2, BYTE3] 的顺序重排列，数据将以 BYTE0, BYTE1, BYTE2, 和 BYTE3 的顺序进行发送/接收。如果DWIDTH设为24-位传输模式，存储在 TX 缓存与 RX 缓存的数据将重新按 [unknown byte, BYTE0, BYTE1, BYTE2] 的顺序重排列。SPI 控制器将按照 BYTE0, BYTE1, BYTE2 的顺序发送/接收数据，每个字节MSB优先发送/接收。16-位模式的规则与上面相同。字节重排序功能只在DWIDTH为16, 24和32 位时适用。

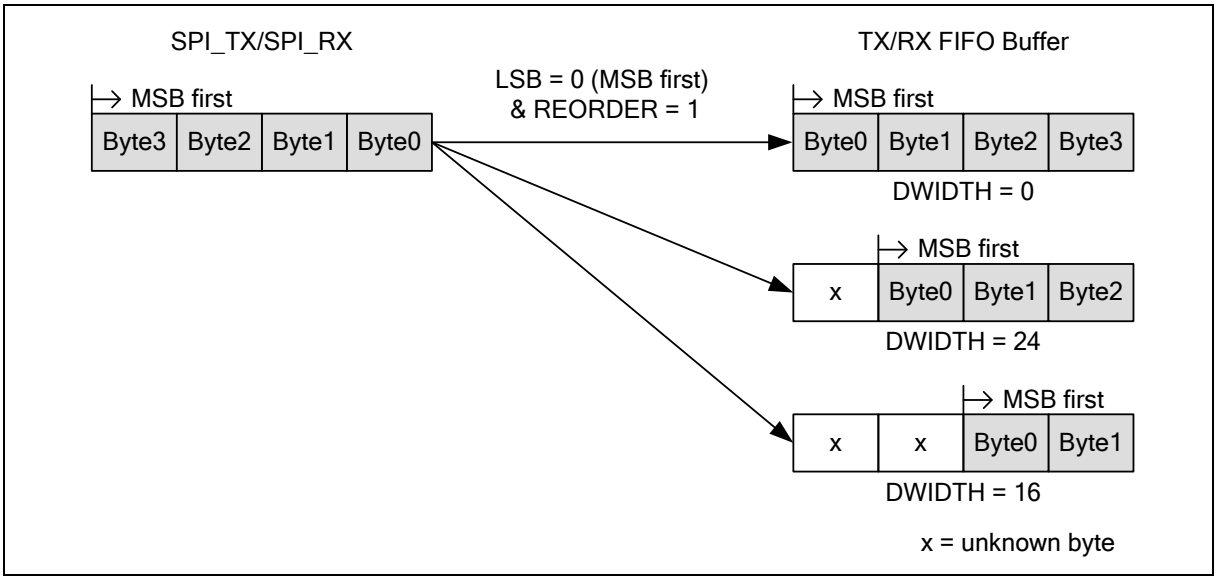


图 6.14-8 字节重排序功能

主机模式下，如果REORDER (SPIx_CTL[19])被设为1，硬件将会在两个连续传输字节之间插入一个0.5 ~ 15.5个SPI时钟周期的休眠间隔。休眠间隔时间由SUSPITV(SPIx_CTL[7:4])设置。

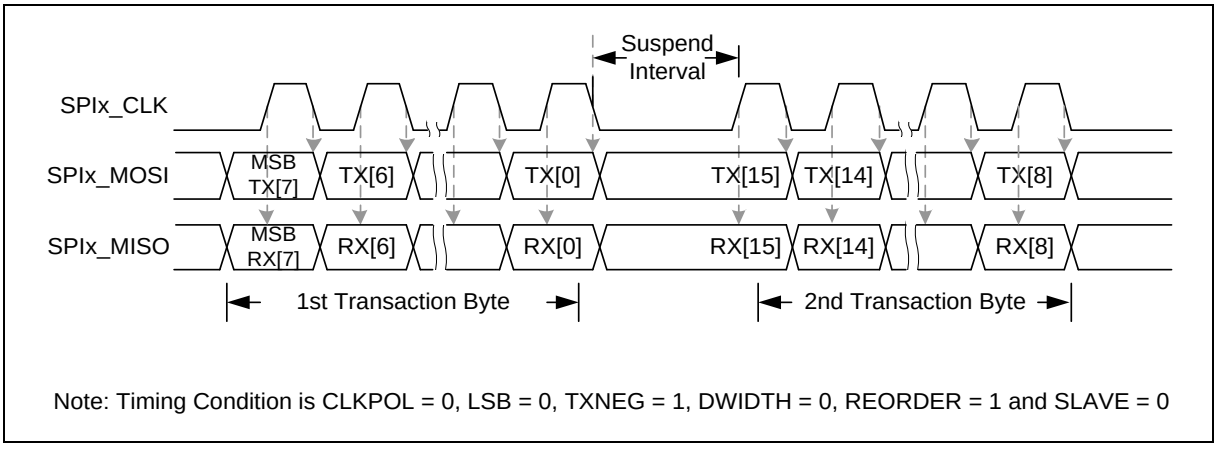


图 6.14-9 字节休眠时序图

6.14.5.4 半双工通讯

通过设置HALFDPX (SPIx_CTL[14])位，SPI控制器能工作在半双工模式。在半双工模式，仅有一根数据线用于接收或发送数据，通过DATDIR (SPIx_CTL[20])来定义传输方向。在半双工模式下，SPIx_MISO管脚是空闲的，可用做其他功能，并能当做GPIO用。使能或禁止HALFDPX (SPIx_CTL[14])控制位将自动同时改变TXFBCLR (SPIx_FIFOCTL[9]) 和RXFBCLR (SPIx_FIFOCTL[8])位.

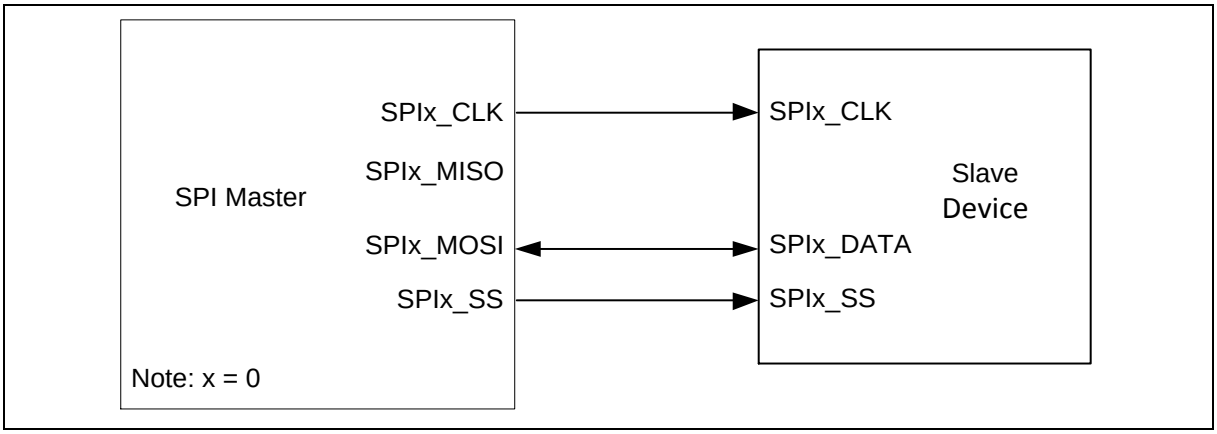


图 6.14-10 SPI 半双工主机模式应用框图

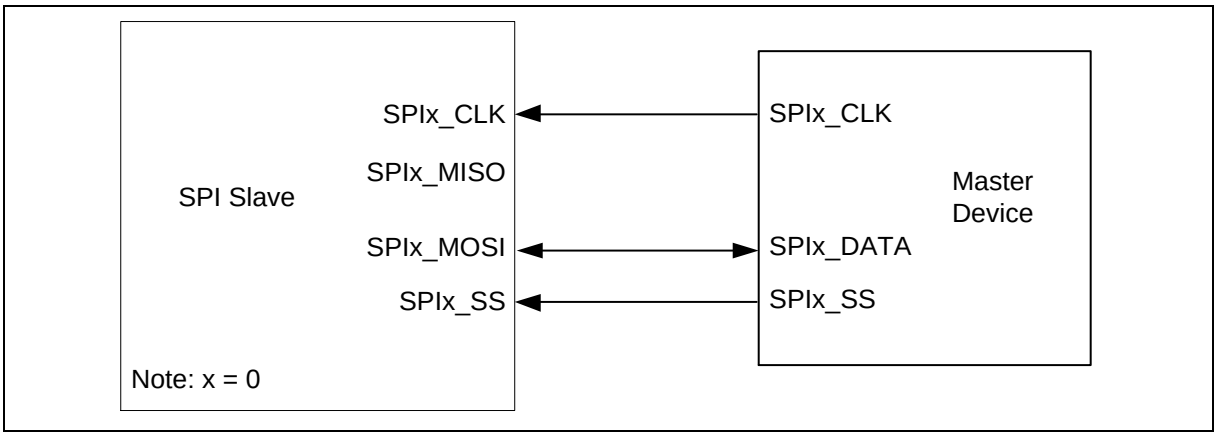


图 6.14-11 SPI 半双工从机模式应用框图

6.14.5.5 只接收模式

通过设置RXONLY (SPIx_CTL[15])，SPI主机设备可工作在只接收模式。只要只接收模式被使能，SPI主机设备产生SPI总线时钟连续从从机设备接收数据。在只接收模式，如果AUTOSS (SPIx_SSCTL[3])被使能，SPI 主机将保持从机片选信号有效。

SPI 主机设备的SPIx_MOSI 管脚不用于通讯，可以配置为GPIO。在只接收模式， 由于SPI总线时钟一直在工作，BUSY (SPIx_STATUS[0]) 状态也一直会被保持。同时，进入这模式还会自动产生TXFBCLR (SPIx_FIFOCTL[9]) 和 RXFBCLR (SPIx_FIFOCTL[8])。当用户使能这种模式后，经过6个周期的外设时钟后SPI总线时钟就会开始输出，已经被写入到传输FIFO缓冲区的数据将被加载到发送移位寄存器，并发送出去。

当用户使能RXONLY (SPIx_CTL[15])后，由DWIDTH (SPIx_CTL[12:8])指定数据位长的SPI RX数据将被接收到RX FIFO中，SPI 时钟会一直传送给SPI 从机设备直到RX FIFO满了为止。

对于位长为8~16位的数据，SPI主设备会输出SPI总线时钟给SPI从设备，并且当RX FIFO计数器RXCNT (SPIx_STATUS[27:24])小于或等于6时接收从设备发送的数据。

对于位长为17~32位的数据，SPI主设备会输出SPI总线时钟给SPI从设备，并且当RX FIFO计数器RXCNT (SPIx_STATUS[27:24])小于或等于2时接收从设备发送的数据。

6.14.5.6 PDMA传输功能

SPI 控制器支持PDMA 传输功能

当TXPDMAEN (SPIx_PDMACTL[0])设置为1时，控制器将请求PDMA控制器来启动PDMA自动发送处理。

当RXPDMAEN (SPIx_PDMACTL[1])设置为1时，控制器将请求PDMA控制器自动接收处理。当接收FIFO缓存有数据时，SPI控制器将请求PDMA控制器自动处理。

PDMA传输完成后，用户需要禁止TXPDMAEN/RXPDMAEN。在再次使能TXPDMAEN/RXPDMAEN前，用户需重新设置PDMA控制寄存器。

注意: SPI 只支持单一PDMA (读/写)请求, 不支持批量PDMA请求。

6.14.5.7 FIFO缓冲区操作

SPI控制器配备了4 个32-位宽的发送和接收FIFO缓存。存放在发送FIFO缓存的数据通过发送控制逻辑进行读取和发送。如果发送FIFO缓存满了，TXFULL (SPIx_STATUS[17])会被置1。当SPI传输逻辑单元抽出发送FIFO缓存的最后一个数据，发送FIFO缓存就为空了，TXEMPTY (SPIx_STATUS[16])位被置1。注意TXEMPTY(SPIx_STATUS[16])标志被置1时，最后一笔传输还在进行。在主机模式，当FIFO缓存写入数据或者SPI总线上有任何事务， BUSY (SPIx_STATUS[0])位被设置为1(如：从机片选信号激活和SPI控制器在从机模式正在接收数据)。当传送缓存为空且当前事务已经完成，该位将被清0。因此，软件应该检查BUSY (SPIx_STATUS[0]) 位的状态以确认SPI是否已经空闲。

接收控制逻辑存储SPI接收到的数据到接收FIFO缓存。有FIFO相关的状态位，像RXEMPTY (SPIx_STATUS[8]) 和RXFULL (SPIx_STATUS[9])，来表明当前接收FIFO缓存的状态。

发送和接收的阀值可以通过设置TXTH (SPIx_FIFOCTL[30:28]) 和RXTH (SPIx_FIFOCTL[26:24])来设定。当存储在发送FIFO缓存的有效数据计数小于或等于TXTH (SPxI_FIFOCTL[30:28])设置时，TXTHIF (SPIx_STATUS[18]) 位会被置1。当存储在接收FIFO缓存的有效数据计数大于RXTH (SPIx_FIFOCTL[26:24])设定，RXTHIF (SPIx_STATUS[10])位会被置1。

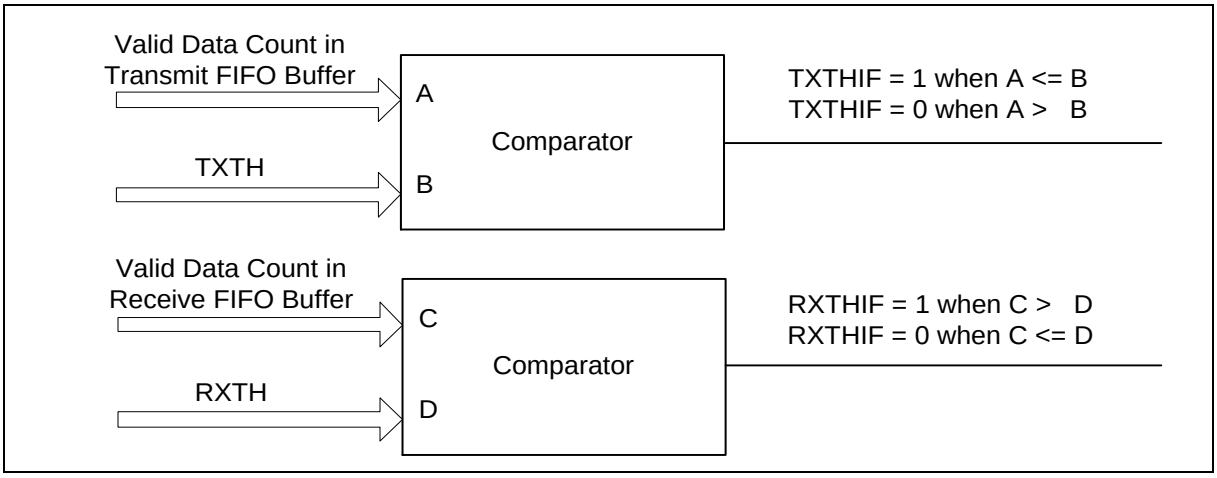


图 6.14-12 FIFO 阀值比较器

在主机模式，当第一个数据写入SPIx_TX寄存器时，TXEMPTY(SPIx_STATUS[16]) 标志将会被清 0。在1个APB时钟周期和6个外设时钟周期后，发送将开始。用户可以立即写下一个数据到SPIx_TX寄存器。SPI 控制器将会在两个连续的数据字传输之间插入一个休眠间隔，休眠间隔的长度由SUSPITV (SPIx_CTL[7:4]) 的设定值决定。如果SUSPITV (SPIx_CTL[7:4])等于0，SPI控制器可以执行连续发送。只要TXFULL(SPIx_STATUS[17])为0，用户可以不断的写入数据到SPIx_TX寄存器。

当用户设置DWIDTH (SPIx_CTL[12:8])为8位~16位时，FIFO的结构将被自动配置为8个16位宽度的发送和接收FIFO缓存区。

图 6.14-13中例 1所示为TXEMPTY (SPIx_STATUS[16])的更新条件和FIFO缓存、移位寄存器和斜移缓存之间的关系。当Data 0 写入FIFO缓存时，TXEMPTY (SPIx_STATUS[16])位被清0。Data 0 将由内核

逻辑加载到移位寄存器中，并且TXEMPTY (SPIx_STATUS[16])将被置1。移位寄存器中Data 0将逐位移入斜移缓存传输直到传输完成。

图 6.14-13中例 2所示为当FIFO缓存中有8个数据时，更新TXFULL (SPIx_STATUS[17])的条件。当TXFULL = 1时，下个数据Data 8 不能被写入FIFO缓存。

图 6.14-14 中例 1所示为17~32位数据长度时，TXEMPTY (SPIx_STATUS[16]) 的更新条件和FIFO缓存、移位寄存器和斜移缓存之间的关系。

图 6.14-14 中例 2所示当FIFO缓存中有4个数据时TXFULL (SPIx_STATUS[17]) 的更新条件。当TXFULL = 1时，下个数据Data5不能被写入FIFO缓存。

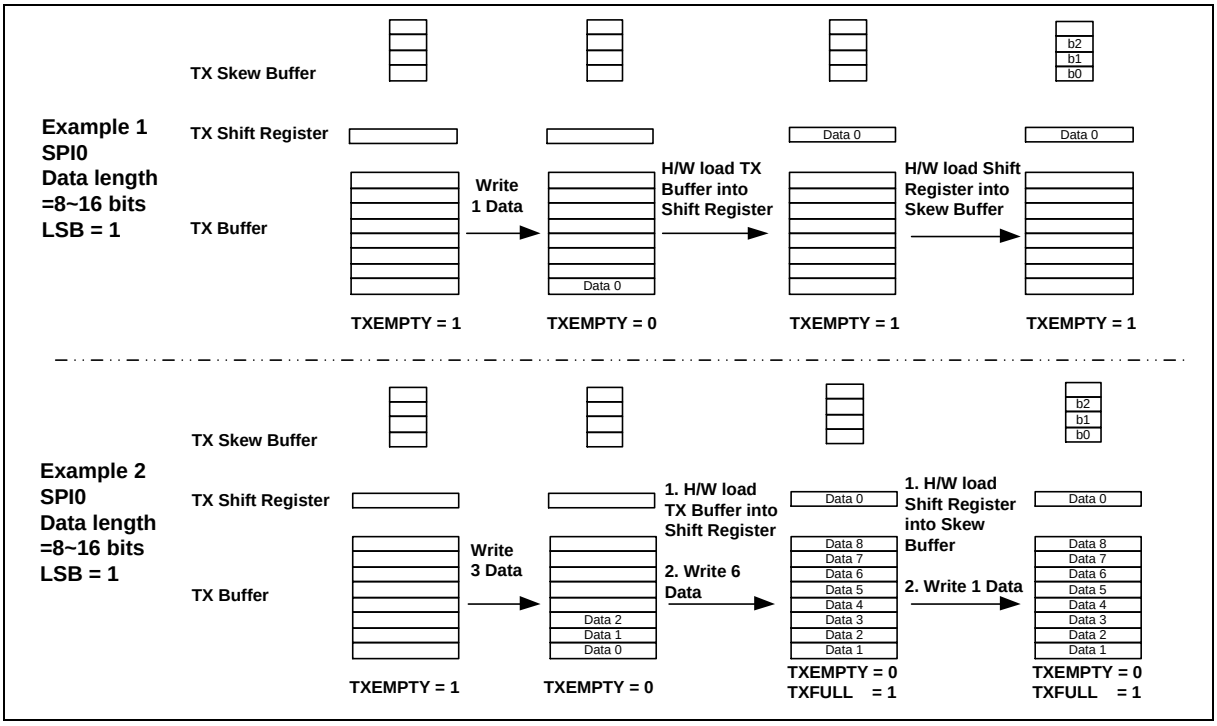


图 6.14-13 8~16 位数据长度的发送 FIFO 缓存示例

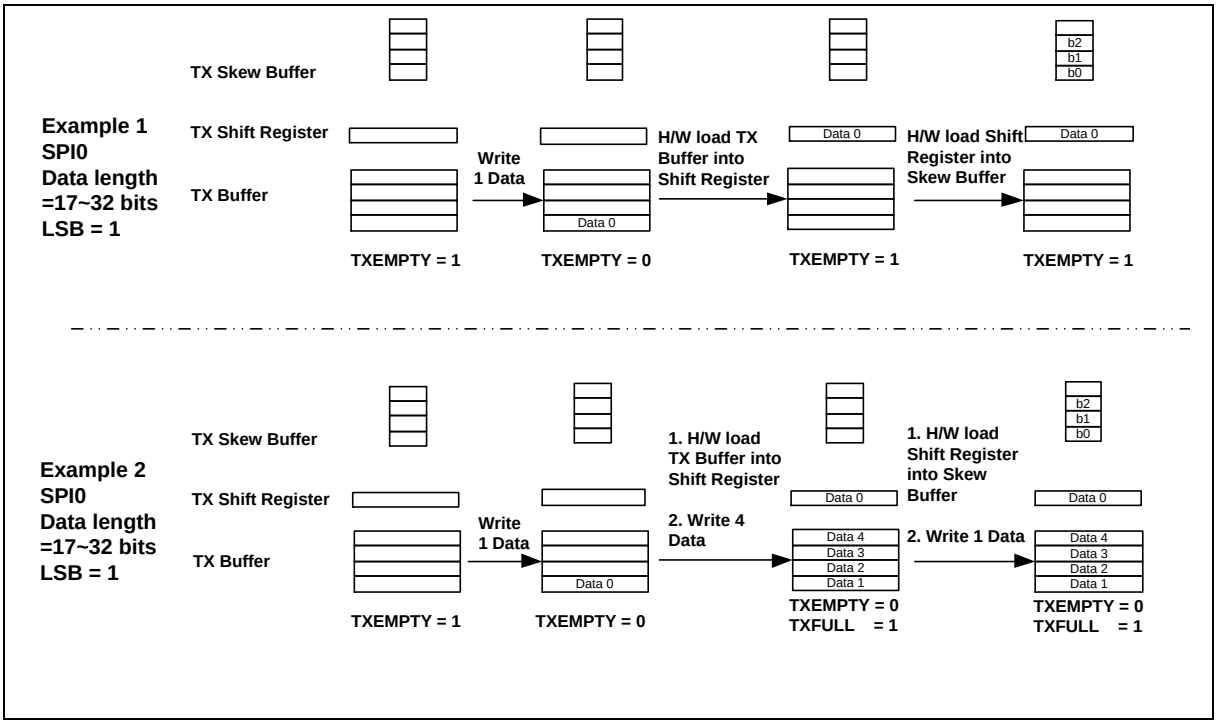


图 6.14-14 17~32 位数据长度的发送 FIFO 缓存示例

如果要发送的数据更新及时，接下来的传输数据将会自动被触发。如果在所有数据传输完成之后，SPIx_TX 寄存器没有被更新，则传输停止。

在主机模式接收操作中，串行数据从SPIx_MISO管脚接收并被存储在接收FIFO缓存。

接收数据(Data 0's b0, b1, …b31)通过串行时钟(SPIx_CLK)先存到斜移缓存，然后再逐位移到移位寄存器。当接收数据位达到DWIDTH (SPIx_CTL[12:8])的值时，内部逻辑将移位寄存器中的数据加载到FIFO缓存。.

当接收FIFO缓存有未读数据时，RXEMPTY (SPIx_STATUS[8]) 将清0 (如图 6.14-15 接收FIFO缓存示例中的示例1)。只要是RXEMPTY (SPIx_STATUS[8])为0，用户可以通过SPIx_RX寄存器来读取接收的数据。如果接收FIFO缓存有8个未读数据，RXFULL(SPIx_STATUS[9])将被置1 (如图 6.14-15接收FIFO缓存示例中示例2)。

当接收FIFO缓存有未读数据时，RXEMPTY (SPIx_STATUS[8]) 将清0 (如图 6.14-16接收FIFO缓存示例中的示例1)。只要是RXEMPTY (SPIx_STATUS[8])为0，用户可以通过SPIx_RX寄存器来读取接收的数据。如果接收FIFO缓存有4个未读数据，RXFULL(SPIx_STATUS[9])将被置1 (如图 6.14-16接收FIFO缓存示例中示例2)。

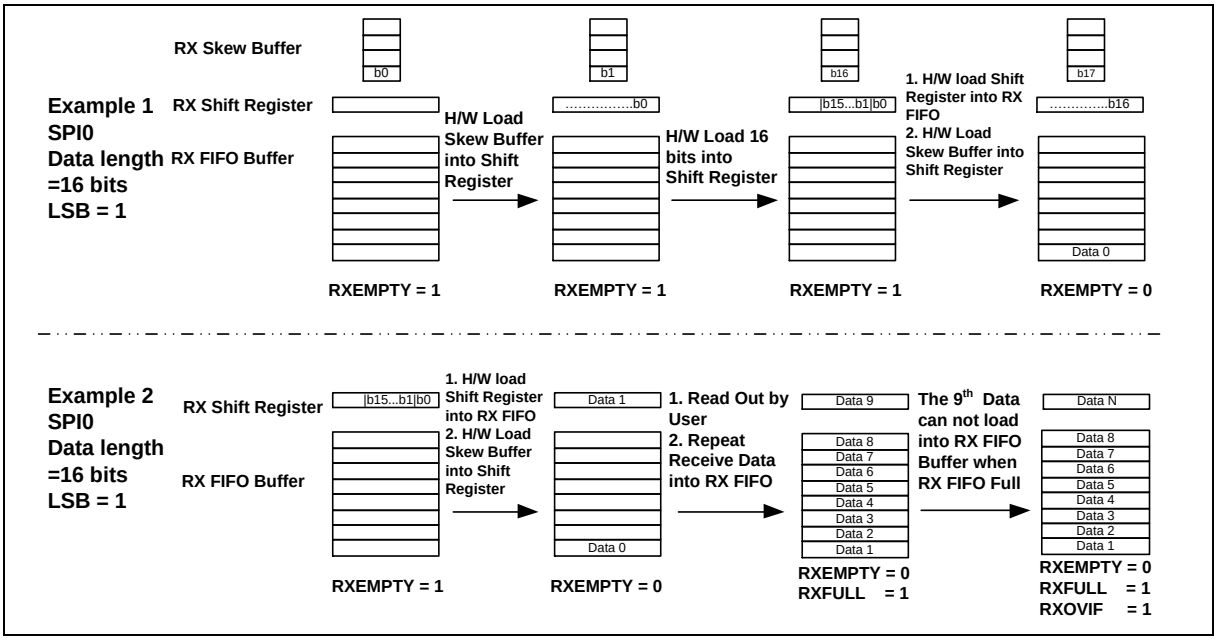


图 6.14-15 接收 FIFO 缓存示例对于 16 位数据长度

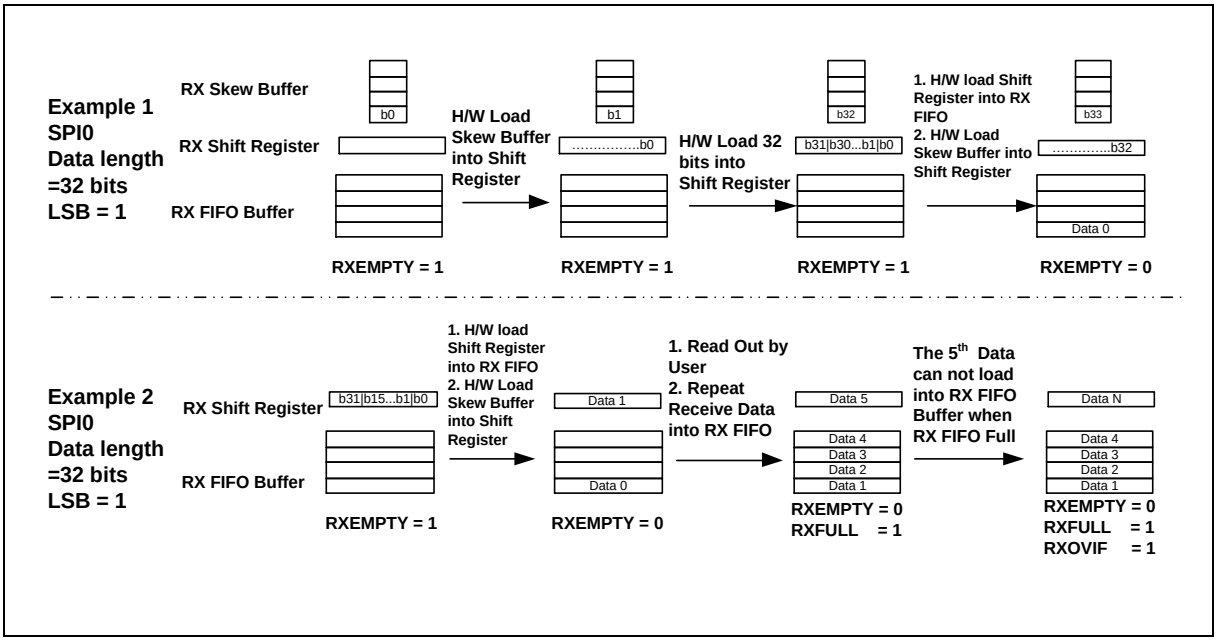


图 6.14-16 接收 FIFO 缓存示例对于 32 位数据长度

在从机模式发送操作中，当软件写数据到SPIx_TX 寄存器时，数据将会被加载到发送FIFO缓存，而且TXEMPTY (SPIx_STATUS[16]) 标志也将被设置为 0。当从设备从主机接收到时钟信号时，发送操作将会开始。只要TXFULL (SPIx_STATUS[17]) 标志为 0，用户就可以写数据到SPIx_TX 寄存器。在所有数据都被 SPI 发送逻辑单元发送出去后，而且用户没有再更新SPIx_TX寄存器，TXEMPTY (SPIx_STATUS[16]) 标志将会被设置为 1。

当从机片选信号有效时，如果没有任何数据写入SPIx_TX寄存器，发送下溢标志TXUFIF (SPIx_STATUS[19]) 将被设置为 1。输出的数据在本次传输中将会通过设置TXUFPOL (SPIx_FIFOCTL[6])保留，直到从机片选信号为无效状态。当传送下溢事件发生时，从机溢出运行标志

SLVURIF (SPIx_STATUS[7])将置为1，同时SPIx_SS进入无效状态。

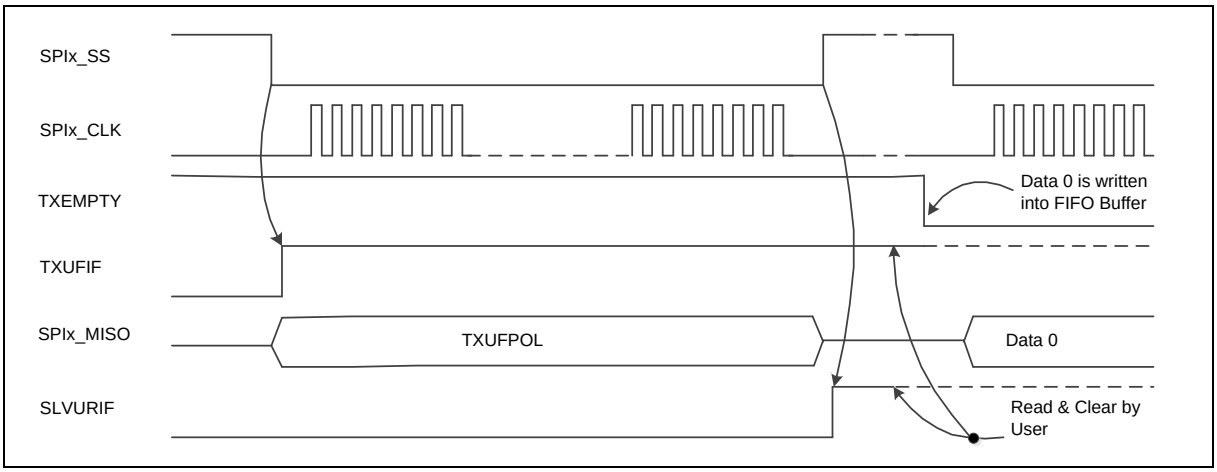


图 6.14-17 TX 下溢事件和从机溢出运行事件

在从机模式接收操作中，串行数据从SPIx_MOSI管脚接收并存储到SPIx_RX寄存器。接收机制类似于主机模式的接收操作。如果接收FIFO缓存包含4个未读数据，RXFULL (SPIx_STATUS[9])将设置为1，RXOVIF (SPIx_STATUS[11])也将设置为1。如果在SPIx_MOSI管脚上有更多串行数据接收，接下来的数据将被丢掉(参照接收FIFO缓存例图)。当从机片选线进入无效状态时，如果接收到位数数据与DWIDTH (SPIx_CTL[12:8])设定的不一致，SLVBEIF (SPIx_STATUS[6])将设置为1。

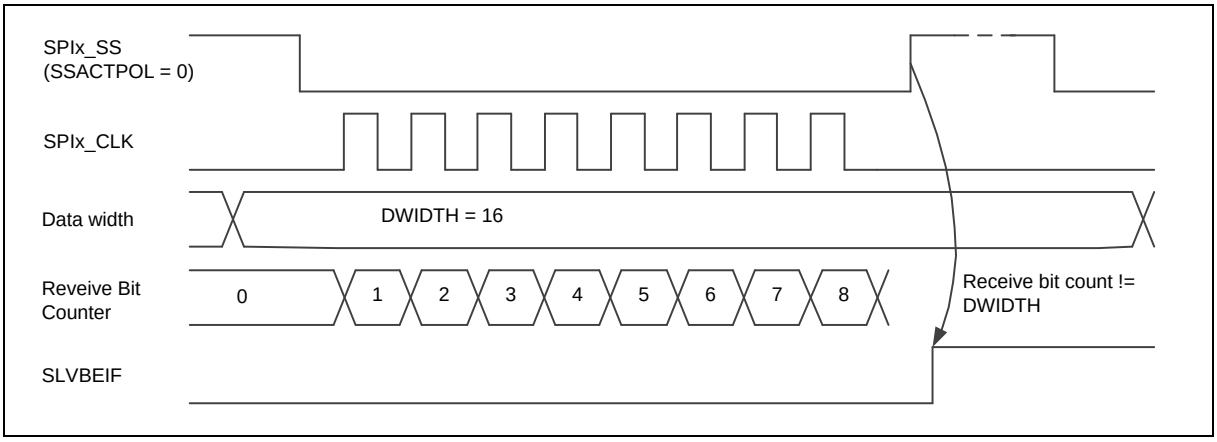


图 6.14-18 从机模式位个数错误

控制器带接收超时功能。当接收FIFO不是空并且接收FIFO在主机模式超过64个SPI时钟周期或在从机模式超过576个SPI外设时钟周期没有读操作, 接收超时发生并且RXTOIF (SPIx_STATUS[12]) 被置1，当通过用户读接收FIFO, 超时状态将自动被清除。

6.14.5.8 中断

- SPI单元传输中断

  当SPI控制器完成一个单元传输，单元传输中断标志UNITIF (SPIx_STATUS[1])将会被设为1。如果单元传输中断使能位UNITIEN (SPIx_CTL[17]) 被置位，则单元传输中断事件将给CPU产生中断请求。单位传输中断标志位只能写1清零。

- SPI从机片选有效/无效中断

  在从机模式，当SPIEN (SPIx_CTL[0])和SLAVE (SPIx_CTL[18])都被设置为1，且从机片选

信号进入有效或无效状态时，从机片选有效或无效中断标志SSACTIF (SPIx_STATUS[2])和SSINAIF (SPIx_STATUS[3])将被置1。如果中断使能位SSINAIEN (SPIx_SSCTL[13])或SSACTIEN (SPIx_SSCTL[12])设置为1，SPI控制器将发生一个中断。

- 从机位计数错误中断

  在从机模式，当从机片选信号线进入无效状态时，如果发送或接收到位数据个数与DWIDTH (SPIx_CTL[12:8])设置的不一致，SLVBEIF (SPIx_STATUS[6])将被设置为1。未传输完成的数据将在发送和接收移位寄存器中弃除。如果中断使能SLVBEIEN (SPIx_SSCTL[8])位设置为1，SPI控制器将发生一个中断。

  注意:如果从机片选信号激活，但是没有任何串行时钟输入，当从机片选信号进入无效状态时，SLVBEIF (SPIx_STATUS[6])也将被置1。

- TX下溢中断

  在SPI从机模式，如果没有任何数据写入SPIx_TX寄存器，当从机选择信号激活时，TXUFIF (SPIx_STATUS[19])将被置1。如果中断使能位TXUFIEN (SPIx_FIFOCTL[7])设置为1，SPI控制器将产生发送下溢中断。

  注意:如果在SPI从机模式下溢事件发生时,有两个条件使SPI从机模式返回到空闲状态，然后准备下一次传输： (1)设置 TXRST 为1 (2) 从机片选信号改变为无效状态。

- 从机发送溢出运行中断

  如果TX 溢出事件发生, 当SPIx_SS 变为无效状态，SLVURIF (SPIx_STATUS[7])将被置1。如果中断使能位SLVURIEN (SPIx_SSCTL[9])设置为1，SPI控制器将发生发送下溢中断。

- 接收溢出中断

  在从机模式，如果接收FIFO缓冲区已有4个未读数据，SPI中的RXFULL (SPIx_STATUS[9])标志将会被置1。如果从SPI总线上接收到更多串行数据并且多余的数据将会被丢弃，RXOVIF (SPIx_STATUS[11]) 标志将被置1。如果中断使能位RXOVIEN (SPIx_FIFOCTL[5])设置为1，SPI控制器将产生接收溢出中断。

- 接收FIFO超时中断

  如果在FIFO里有一个接收到的数据，在主机模式下用户超过64个SPI外设时钟周期没有去读取，或者从机模式下超过576个SPI外设时钟周期没有去读取，如果接收超时中断使能位RXTOIEN (SPIx_FIFOCTL[4])设置为1，则会向系统发出一个接收超时中断。

- 发送 FIFO 中断

  在FIFO模式，如果发送FIFO缓冲区的有效数据计数少于或等于TXTH (SPIx_FIFOCTL[30:28])的设定值，发送FIFO中断标志TXTHIF (SPIx_STATUS[18])会被置1。如果发送FIFO中断使能位TXTHIEN (SPIx_FIFOCTL[3])设置为1，则SPI控制器会向系统产生一个发送FIFO中断。

- 接收 FIFO 中断

  在 FIFO 模式，如果接收FIFO缓冲区的有效数据计数大于RXTH (SPIx_FIFOCTL[26:24])的设定值，接收FIFO中断标志RXTHIF (SPIx_STATUS[10])会被置1。如果接收FIFO中断使能位RXTHIEN (SPIx_FIFOCTL[2])设置为1，SPI控制器将会向系统产生一个接收FIFO中断。

6.14.5.9 I²S模式

SPI0控制器支持I2S模式、支持PCM模式A、PCM模式B和MSB对齐和I2S数据格式。音频通道的位宽由WDWIDTH (SPIx_I2SCTL[5:4])设定。传输顺序一般为MSB(最高位优先)。数据在时钟的上升沿时读取下降沿时传送。

在I²S数据格式中，在音频通道的第二个时钟最高位(MSB)被送出。I2Sx_LRCLK信号表示正在传输的音频通道。

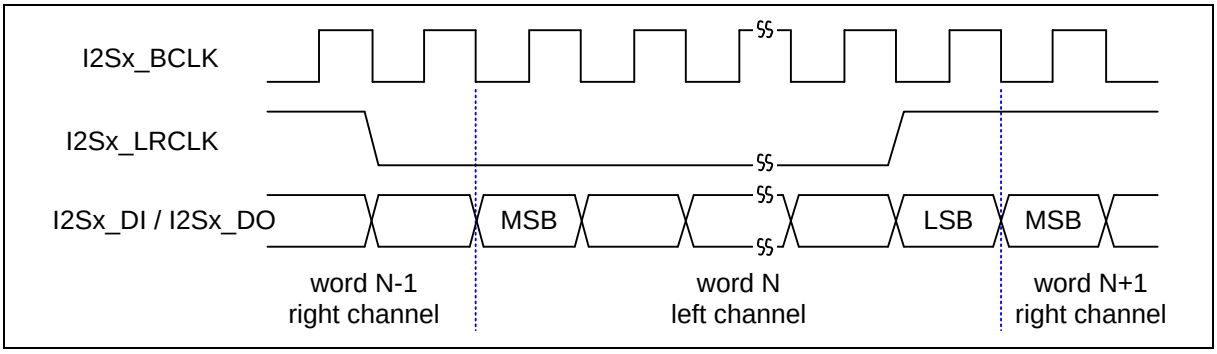


图 6.14-19 I²S 数据格式时序图

在MSB对齐数据格式，在音频通道的第一个时钟最高位(MSB)被送出。

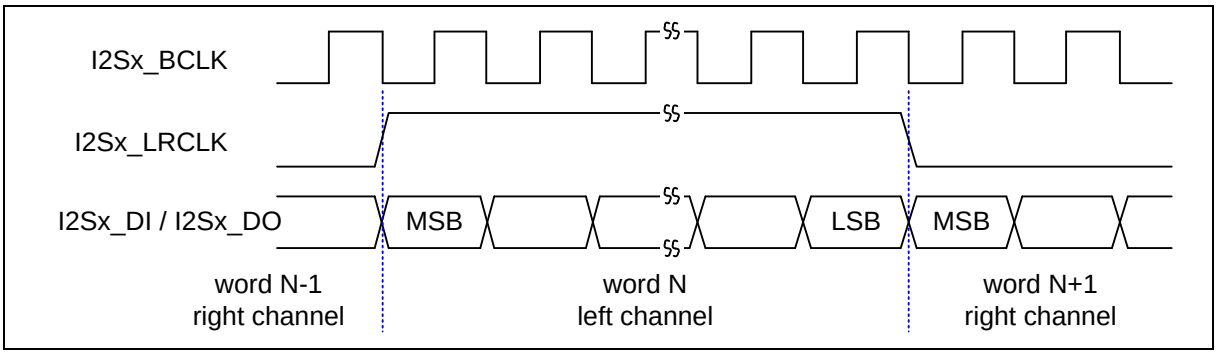


图 6.14-20 MSB 对齐数据格式时序图

I2Sx_LRCLK 信号也支持 PCM 模式A 和 PCM 模式B。在PCM模式，I2Sx_LRCLK信号表示音频通道的开始。.

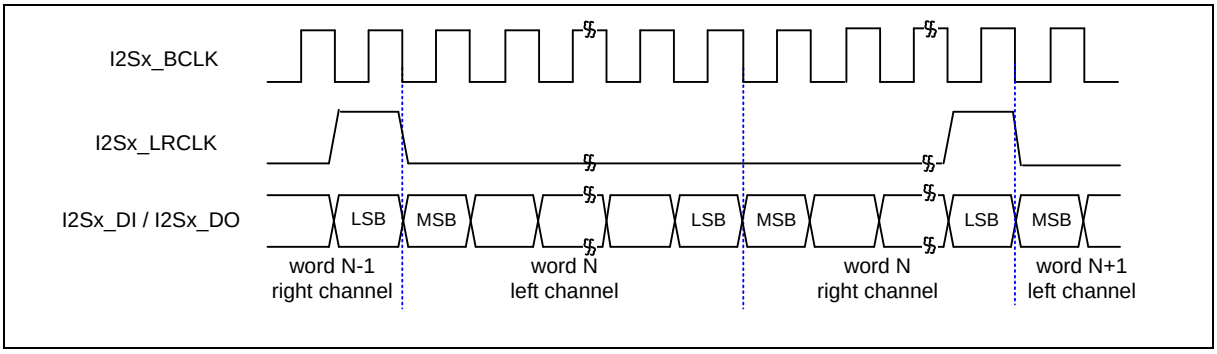


图 6.14-21 PCM 模式 A 时序图

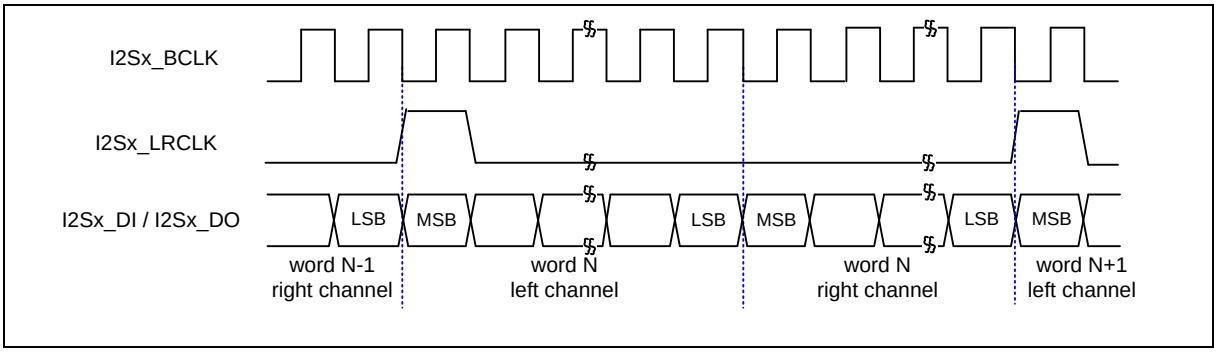


图 6.14-22 PCM 模式 B 时序图

6.14.5.10 I²S模式 FIFO 操作

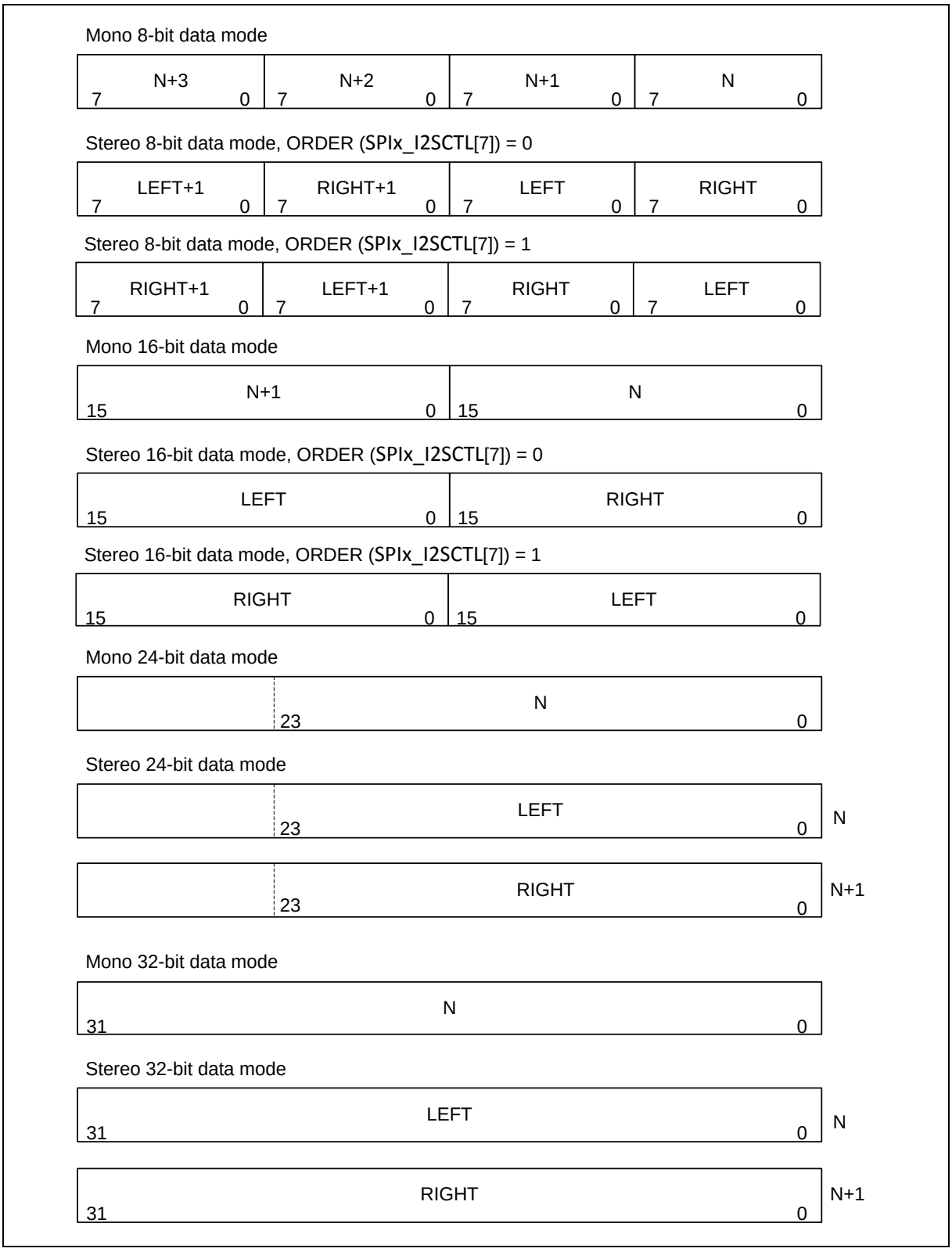


图 6.14-23 各种 I²S 模式的 FIFO 内容

6.14.5.11 I²S / PCM 主机模式和单声道模式虚拟数据数

I²S / PCM主机在开始发送TX数据给外部从机设备之前，我们需要设置控制寄存器I2SEN (SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能和写数据到TX FIFO。主机发送虚拟数据（数据为0）给外部从机设备之后，主机才将发送TX FIFO的数据给外部从机设备。表 6.14-2 表示在主机单声道模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8 位	0
16位	0
24位	1
32位	1

表 6.14-2 主机模式和单声道模式虚拟数据数

6.14.5.12 主机模式和立体声模式虚拟数据数

I²S / PCM主机在开始发送TX数据给外部从机设备之前，我们需要设置控制寄存器I2SEN (SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能和写数据到TX FIFO。主机发送虚拟数据（数据为0）给外部从机设备之后，主机才将发送TX FIFO的数据给外部从机设备。表 6.14-3表示在主机立体声模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8位	0
16位	0
24位	1
32位	1

表 6.14-3 主机模式和立体声模式虚拟数据数

6.14.5.13 I²S从机模式和单声道模式虚拟数据数

I²S / PCM从机在开始发送TX数据给外部主机设备之前，我们需要设置控制寄存器I2SEN (SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能，SLAVE模式(SPIx_I2SCTL[8])和写数据到TX FIFO。从机发送虚拟数据（数据为0）给外部主机设备之后，从机才将发送TX FIFO的数据给外部主机设备。表 6.14-4表示在I²S从机单声道模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8位	3
16位	2
24位	2
32位	2

表 6.14-4 I²S 从机模式和单声道模式虚拟数据数

6.14.5.14 PCM从机模式和单声道模式虚拟数据数

I²S / PCM从机在开始发送TX数据给外部主机设备之前，我们需要设置控制寄存器I2SEN

(SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能，SLAVE模式(SPIx_I2SCTL[8])和写数据到TX FIFO。从机发送虚拟数据（数据为0）给外部主机设备之后，从机才将发送TX FIFO的数据给外部主机设备。表 6.14-5表示在PCM从机单声道模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8位	2
16位	1
24位	1
32位	1

表 6.14-5 PCM 从机模式和单声道模式虚拟数据数

6.14.5.15 I²S从机模式和立体声模式虚拟数据数

I²S / PCM从机在开始发送TX数据给外部主机设备之前，我们需要设置控制寄存器I2SEN (SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能，SLAVE模式(SPIx_I2SCTL[8])和写数据到TX FIFO。从机发送虚拟数据（数据为0）给外部主机设备之后，从机才将发送TX FIFO的数据给外部主机设备。表 6.14-6表示在I²S从机立体声模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8位	3
16位	2
24位	2
32位	2

表 6.14-6 I²S 从机模式和立体声模式虚拟数据数

6.14.5.16 PCM从机模式和立体声模式虚拟数据数

I²S / PCM从机在开始发送TX数据给外部主机设备之前，我们需要设置控制寄存器I2SEN (SPIx_I2SCTL[0])使能，TXEN (SPIx_I2SCTL[1])使能，SLAVE模式(SPIx_I2SCTL[8])和写数据到TX FIFO。从机发送虚拟数据（数据为0）给外部主机设备之后，从机才将发送TX FIFO的数据给外部主机设备。表 6.14-7表示在PCM从机立体声模式虚拟数据的个数和给左声道+右声道虚拟数据量的单位。

数据宽度 (SPIx_I2SCTL[5:4])	虚拟数据数 (单位 = L 声道 + R 声道)
8位	2
16位	1
24位	1
32位	1

表 6.14-7 PCM 从机模式和立体声模式虚拟数据数

6.14.6 时序图

从机片选信号的有效状态可以由SSACTPOL (SPIx_SSCTL[2])的设置定义。SPI时钟在空闲状态可以通过CLKPOL (SPIx_CTL[3]) 配置为高电平或低电平。传输字长度在DWIDTH (SPIx_CTL[12:8])中定义，发送/接收数据是以MSB 还是 LSB 优先由 LSB 位(SPIx_CTL[13]) 定义。用户也可以通过设置

TXNEG/RXNEG (SPIx_CTL[2:1])来选择发送/接收数据时SPI时钟的边沿。关于SPI主机/从机操作四种时序图以及相关设置如下图所示。

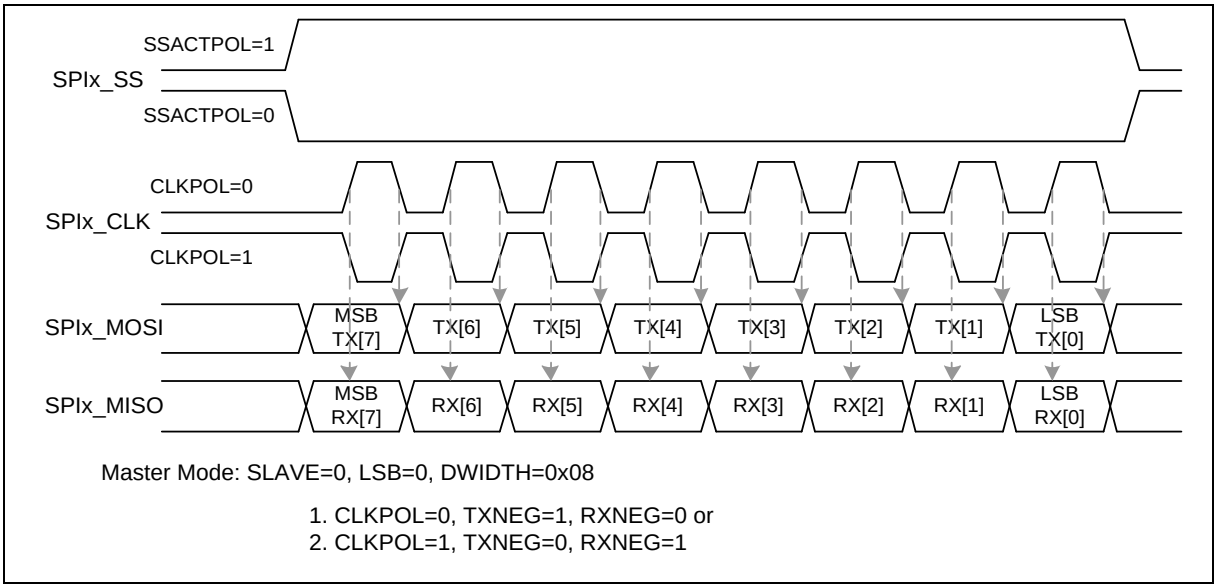


图 6.14-24 主机模式下的 SPI 时序

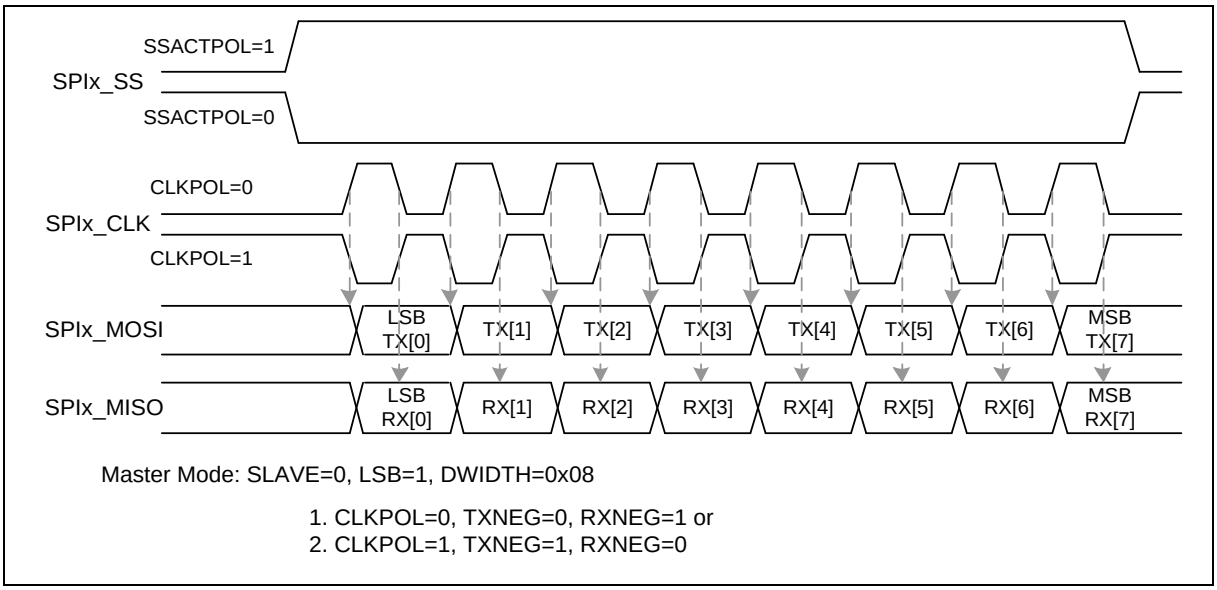


图 6.14-25 主机模式下的 SPI 时序(交替 SPIx_CLK 时钟相位)

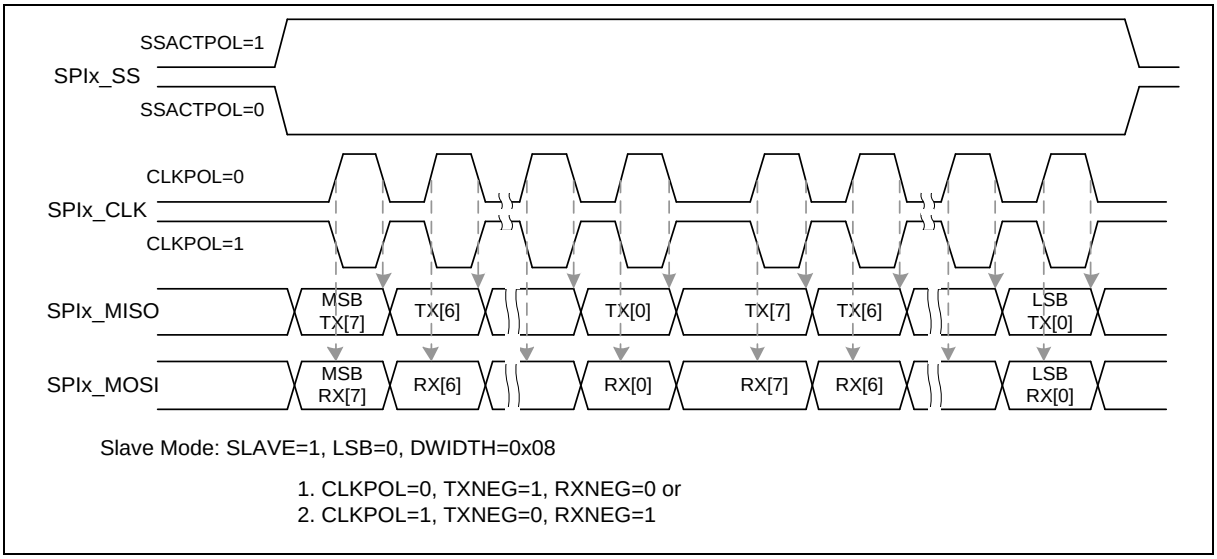


图 6.14-26 从机模式下的 SPI 时序

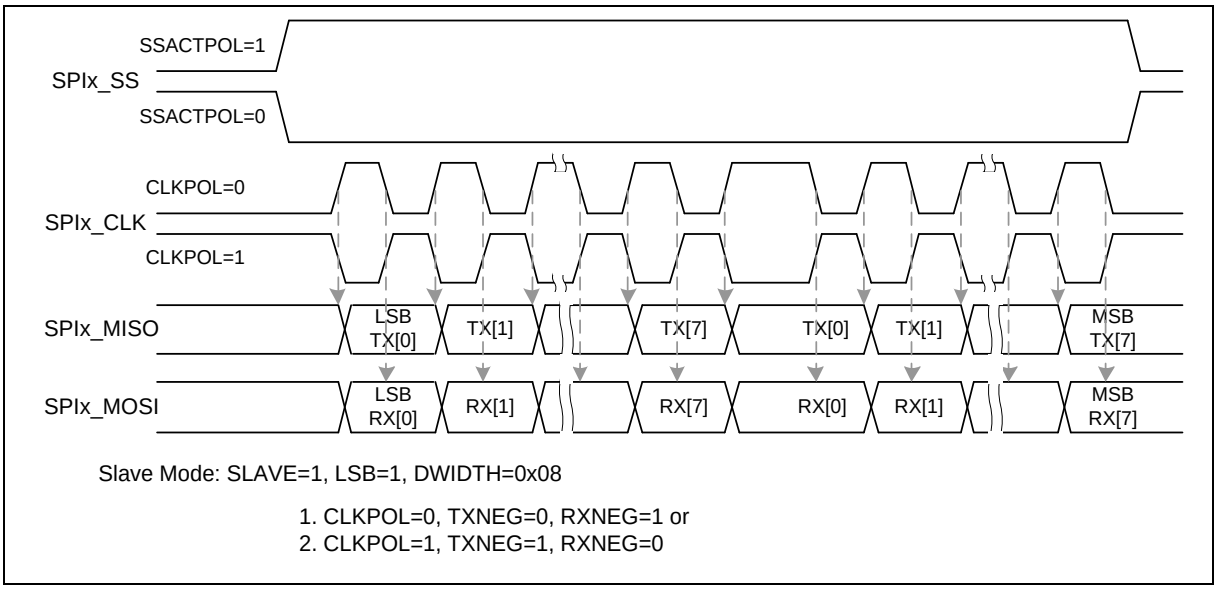


图 6.14-27 从机模式下的 SPI 时序(交替 SPIx_CLK 时钟相位)

6.14.7 编程例程

例 1:

The SPI控制器设定为全双工主机模式去访问片外从机设备遵循如下规范：

- 数据在SPI总线时钟上升沿锁存
- 数据在SPI总线时钟下降沿传输
- MSB 优先传输
- SPI总线时钟空闲状态下为低电平
- 每次发送/接收只有一个字节
- 使用SPI从机片选管脚和一个片外从机相连。从机片选信号为低电平有效.

操作流程如下：

1. 设置DIVIDER (SPIx_CLKDIV[8:0]) 来确定SPI时钟输出频率。
2. 写一个合适的值到SPIx_SSCTL 寄存器来对主机相关配置进行设定：
   1) 清除AUTOSS (SPIx_SSCTL[3])为0，来禁止自动从机选择功能。
   2) 清除SSACTPOL (SPIx_SSCTL[2])为0，来设置从机片选信号有效电平为低电平。
   3) 设置SS (SPIx_SSCTL[0])为1，来使能从机片选信号以激活片外从机设备。
3. 通过设置SPIx_CTL寄存器来控制SPI 主机的行为。
   1) 通过设置SLAVE (SPIx_CTL[18])为0，将 SPI 控制器设为主机设备。
   2) 通过设置CLKPOL (SPIx_CTL[3])为0，将SPI时钟的空闲状态设为低电平。
   3) 通过设置TXNEG (SPIx_CTL[2])为1，选择数据在SPI总线时钟的下降沿传输。
   4) 通过设置RXNEG (SPIx_CTL[1])为0，选择数据在SPI总线时钟的上升沿锁存。
   5) 通过设置DWIDTH 位域 (SPIx_CTL[12:8] = 0x08)，来设定一个字传输的长度为8位。
   6) 通过设置LSB (SPIx_CTL[13])为0，来设定MSB传输优先。
4. 设置SPIEN (SPIx_CTL[0]) 为1来使能SPI接口的数据传输。
5. 如果 SPI 主机想要发送(写)一个字的数据到片外从机设备，则将所要发送到数据写入SPIx_TX寄存器。
6. 等待 SPI 中断发生，如果中断使能位UNITIEN (SPIx_CTL[17])设置为1）或轮询传输单元中断标志UNITIF (SPIx_STATUS[1])。
7. 从寄存器SPIx_RX中读出接收到的一个字节数据。
8. 重复步骤 5) 继续其他数据传输或设置SS(SPIx_SSCTL[0])为 0 来停止片外从机设备。

例 2:

SPI 控制器作为全双工从机设备，和片外主机设备相连，外设主机设备通过SPI 接口与片上 SPI 从机通信。过程如下：

- 数据在SPI总线时钟上升沿锁存
- 数据在SPI总线时钟下降沿传输
- LSB 先传输
- SPI总线时钟空闲状态为高电平
- 每次发送/接收一个字节
- 从机片选信号为高电平有效

操作流程如下：

1. 将从机模式的相应配置写入合适值到SPIx_SSCTL 寄存器。
2. 通过设置SSACTPOL (SPIx_SSCTL[2]) 为1，来选择高电平作为从机片选信号的有效输入。
3. 给SPIx_CTL 寄存器写入相关的值来控制 SPI 从机的行为
   1) 通过设置SLAVE (SPIx_CTL[18]) 为1 将 SPI 控制器设为从机设备。
   2) 通过设置CLKPOL (SPIx_CTL[3])为1 将SPI时钟的空闲状态设为高电平。
   3) 通过设置TXNEG (SPIx_CTL[2])为1 选择数据在SPI总线时钟的下降沿传输。

4) 通过设置RXNEG (SPIx_CTL[1])为0 选择数据在SPI总线时钟的上升沿锁存。

5) 通过设置DWIDTH 位域 (SPIx_CTL[12:8] = 0x08)来设定一个字传输的位长度为8位。

6) 通过设置LSB (SPIx_CTL[13])为1来设置LSB优先传输。

4. 设置SPIEN (SPIx_CTL[0])为1，等待片外主机设备提供给从机片选触发输入和SPI时钟输入来启动数据传输。

5. 如果 SPI 从机要发送(被读取)一个字节的数据到片外主机，则将所要发送的数据写入到SPIx_TX寄存器。

6. 如果 SPI 从机只是要从片外主机接收(被写入)一字节数据，用户不需要关心什么数据将被传输，SPIx_TX 寄存器也不需要用户去更新。

7. 等待 SPI 中断发生（如果中断使能位UNITIEN (SPIx_CTL[17])设置为1）或轮询传输单元中断标志UNITIF (SPIx_STATUS[1])。

8. 从寄存器SPIx_RX中读取接收到的一个字节数据。

9. 重复步骤 5) 继续其他数据传输或停止数据传输。

6.14.8 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读写

寄存器	偏移	R/W	描述	复位值
SPI 基地址: **SPIx_BA = 0x4006_1000**				
SPIx_CTL	SPIx_BA+0x00	R/W	SPI控制寄存器	0x0000_0034
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI 时钟除频器	0x0000_0000
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI 从机选择控制寄存器	0x0000_0000
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA 控制寄存器	0x0000_0000
SPIx_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO 控制寄存器	0x2200_0000
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI 状态寄存器	0x0005_0110
SPIx_TX	SPIx_BA+0x20	W	SPI 数据传输寄存器	0x0000_0000
SPIx_RX	SPIx_BA+0x30	R	SPI 数据接收寄存器	0x0000_0000
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I^2S 控制寄存器	0x0000_0000
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I^2S 时钟分频控制寄存器	0x0000_0000
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I^2S 状态寄存器	0x0005_0100

6.14.9 寄存器描述

SPI 控制寄存器 (SPIx_CTL)

寄存器	偏移	R/W	描述	复位值
SPIx_CTL	SPIx_BA+0x00	R/W	SPI 控制寄存器	0x0000_0034

注: 不支持 I^2S 模式。

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**DATDIR**	**REORDER**	**SLAVE**	**UNITIEN**	**Reserved**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXONLY	**HALFDPX**	**LSB**	**DWIDTH**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPITV				**CLKPOL**	**TXNEG**	**RXNEG**	**SPIEN**

位	描述	
[31:21]	**Reserved**	保留。
[20]	**DATDIR**	**数据端口方向控制** 在半双工和双线，四线传输，该位用于选择数据输入/输出方向。 0 = SPI 数据输入方向. 1 = SPI 数据输出方向.
[19]	**REORDER**	**字节重排序功能使能位** 0 =禁止字节重排序功能. 1 =使能字节重排序功能，在每两个字节之间插入一个字节休眠间隔。字节休眠间隔时间取决于SUSPITV 的设置。 **注意:**字节重排序仅在DWIDTH 被定义为 16 位，24 位和 32 位时有效.
[18]	**SLAVE**	**从机模式控制位** 0 = 主机模式 1 = 从机模式
[17]	**UNITIEN**	**单元传输中断使能位** 0 = 禁止SPI单元传输中断 1 = 使能SPI单元传输中断
[16]	**Reserved**	保留.
[15]	**RXONLY**	**只接收模式使能位 (仅主机支持)** 该位仅仅在主机模式有效。在只接收模式，SPI主机设备产生SPI总线时钟连续从从机设备接收数据并且挂起BUSY 状态. 0 = 只接收模式禁止. 1 = 只接收模式使能.

[14]	**HALFDPX**	**SPI 半双工传输使能位** 该位用来选择SPI全双工或半双工传输。DATDIR (SPIx_CTL[20])位用来设置半双工传输时的数据方向. 0 = SPI 全双工传输. 1 = SPI 半双工传输.
[13]	**LSB**	**优先发送LSB** 0 = MSB, 具体发送/接收寄存器的哪一位首先被发送/接收，取决于DWIDTH 的设定值。 1 = LSB, SPI TX寄存器的位 0，首先被发送到 SPI 数据输出管脚，从 SPI 数据输入管脚上接收到的第一个数据位将会被放置到 RX 寄存器 (SPIx_RX的位0) LSB 的位置.
[12:8]	**DWIDTH**	**数据宽度** 该位域指定在一次发送/接收事务中，有多少个数据位将会被传输。最小位长是8位，最多可以达到 32 位。 DWIDTH = 0x08 … 8 位 DWIDTH = 0x09 … 9位 …… DWIDTH = 0x1F … 31位 DWIDTH = 0x00 … 32位 **注:**SPI 模式，此位决定TX/RX FIFO深度配置。同时，改变这个位将让硬件自动清除TX/RX FIFO 。
[7:4]	**SUSPITV**	**休眠间隔(仅主机模式支持)** 该四位用来配置在一次数据传输过程中连续两个发送/接收数据字之间的休眠间隔。休眠间隔是从当前数据字的最后一个时钟边沿到接下来的传输数据字的第一个边沿时钟。缺省值是0x3. 休眠间隔的周期可以根据下面公式获得： (SUSPITV [3:0] + 0.5) * SPICLK时钟周期 例: SUSPITV = 0x0 … 0.5 SPICLK 时钟周期 SUSPITV = 0x1 … 1.5 SPICLK 时钟周期 …… SUSPITV = 0xE … 14.5 SPICLK 时钟周期 SUSPITV = 0xF … 15.5 SPICLK 时钟周期
[3]	**CLKPOL**	**时钟极性** 0 = SPI总线时钟空闲状态的电平为低. 1 = SPI总线时钟空闲状态的电平为高
[2]	**TXNEG**	**在下降沿发送** 0 = 在 SPI总线时钟的上升沿改变发送数据输出信号 1 = 在 SPI总线时钟的下降沿改变发送数据输出信号
[1]	**RXNEG**	**在下降沿接收** 0 = 在 SPI总线时钟的上升沿锁存接收数据输入信号. 1 = 在 SPI总线时钟的下降沿锁存接收数据输入信号.

[0]	**SPIEN**	**SPI 传输控制使能位** 在主机模式下，如果在FIFO缓存有数据，在该位设置为1后，开始传输。在从机模式下，当该位设置为1时，该设备已准备好接收数据。 0 = 禁止控制传输 1 = 使能控制传输 **注意:**在更改SPIx_CTL, SPIx_CLKDIV, SPIx_SSCTL 和 SPIx_FIFOCTL寄存器的配置之前，用户需对SPIEN (SPIx_CTL[0])清0，并且确定SPIENSTS (SPIx_STATUS[15])的值为0。

SPI除频寄存器(SPIx_CLKDIV)

寄存器	偏移	R/W	描述	复位值
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI时钟除频寄存器	0x0000_0000

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

位	描述	
[31:9]	**Reserved**	保留..
[8:0]	**DIVIDER**	**时钟除频** 该域的值是频率除频器值，用于产生外设时钟f_{spi_eclk}，和SPI主机的SPI总线时钟。 可根据下列公式获得所期望的频率： $f_{spi_eclk} = \frac{f_{spi_clock_src}}{(DIVIDER+1)}$ $f_{spi_clock_src}$ 是外设的时钟源，在时钟控制寄存器CLK_CLKSEL2 里定义。 **注1:** 不支持 I^2S 模式. **注2:** 在释放SPI IP软件复位和设置这个时钟分频寄存器的时钟间隔必须大于或等于5个外设时钟周期。

注:因为外设时钟频率必须低于或等于系统频率，用户应该仔细设置除频器的值。

SPI 从机选择寄存器(SPIx_SSCTL)

寄存器	偏移	R/W	描述	复位值
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI 从机选择控制寄存器	0x0000_0000

注意: 不支持 I^2S 模式.

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SSINAIEN**	**SSACTIEN**	**Reserved**		**SLVURIEN**	**SLVBEIEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**AUTOSS**	**SSACTPOL**	**Reserved**	**SS**

位	描述	
[31:14]	**Reserved**	保留.
[13]	**SSINAIEN**	**从机片选无效中断使能位** 0 = 禁止从机片选无效中断 1 = 使能从机片选无效中断
[12]	**SSACTIEN**	**从机片选有效中断使能位** 0 = 禁止从机片选有效中断 1 = 使能从机片选有效中断
[11:10]	**Reserved**	保留.
[9]	**SLVURIEN**	**从机模式发送溢出运行中断使能位** 0 = 禁止从机模式发送溢出运行中断 1 = 使能从机模式发送溢出运行中断
[8]	**SLVBEIEN**	**从机模式位计数错误中断使能位** 0 = 禁止从机模式位计数错误中断 1 = 使能从机模式位计数错误中断
[7:4]	**Reserved**	保留.
[3]	**AUTOSS**	**自动从机片选功能使能位(仅支持主机模式)** 0 = 禁止自动从机片选功能。此时，通过SS (SPIx_SSCTL[0])来设置从机片选信号无效或激活状态。 1 = 使能自动从机片选功能
[2]	**SSACTPOL**	**从机片选有效极性位** 该位定义从机片选信号(SPIx_SS)有效的极性 0 = 从机片选信号 SPIx_SS低有效 1 = 从机片选信号 SPIx_SS高有效

[1]	**Reserved**	保留.
[0]	**SS**	**从机片选控制位 (仅主机模式)** 如果AUTOSS位设置为0 0 = 设置SPIx_SS线为无效状态 1 = 设置SPIx_SS线为激活状态 如果AUTOSS位设置为1 0 = 保持SPIx_SS线为无效状态 1 = 在数据传输期间，SPIx_SS线将自动进入激活状态，同样在其它时间将自动进入无效状态。SPIx_SS的激活状态在SSACTPOL (SPIx_SSCTL[2])中指定。

SPI PDMA 控制寄存器(SPIx_PDMACTL)

寄存器	偏移	R/W	描述	复位值
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA 控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留.
[2]	**PDMARST**	**PDMA 复位** 0 = 无效. 1 = 复位SPI控制器的PDMA 控制逻辑。该位会自动清0。
[1]	**RXPDMAEN**	**接收PDMA 使能位** 0 = 禁止PDMA接收功能 1 = 使能PDMA接收功能
[0]	**TXPDMAEN**	**发送PDMA 使能位** 0 = 禁止PDMA发送功能. 1 = 使能PDMA发送功能 **注意:** 在SPI主机模式支持全双工传输，如果发送和接收PDMA功能都使能，接收PDMA功能不能在发送PDMA功能之前使能。用户可以先使能发送PDMA功能或者同时使能两个功能。

SPI FIFO 控制寄存器(SPIx_FIFOCTL)

寄存器	偏移	R/W	描述	复位值
SPIx_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO 控制寄存器	0x2200_0000

31	30	29	28	27	26	25	24
Reserved	**TXTH**			**Reserved**	**RXTH**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**TXFBCLR**	**RXFBCLR**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXUFIEN	**TXUFPOL**	**RXOVIEN**	**RXTOIEN**	**TXTHIEN**	**RXTHIEN**	**TXRST**	**RXRST**

位	描述	
[31]	**Reserved**	保留.
[30:28]	**TXTH**	**发送 FIFO 阈值** 如果发送FIFO缓存的有效数据个数小于或者等于TXTH的设置，TXTHIF将会被设置为1， 否则TXTHIF将会被设置为 0。 该位域的MSB只有在SPI模式8 ~16数据长度时才有意义。
[27]	**Reserved**	保留.
[26:24]	**RXTH**	**接收 FIFO阈值** 如果接收FIFO缓存的有效数据个数大于RXTH的设置，RXTHIF 将会被设置为 1， 否则RXTHIF 将会被设置为 0。该位域的MSB只有在SPI模式8 ~16数据长度时才有意义。
[23:10]	**Reserved**	保留.
[9]	**TXFBCLR**	**发送FIFO 缓存清除位** 0 = 无影响 1 = 清除发送FIFO指针，TXFULL位将被清零而且TXEMPTY位也将被置1。该位设置为1后，在一个系统时钟后，由硬件自动清零。 **注意:** 发送移位寄存器不会被清除。
[8]	**RXFBCLR**	**接收FIFO缓存清除位** 0 = 无影响 1 = 清除接收FIFO指针，RXFULL位将被清零而且RXEMPTY位将被设置1。 该位设置为1后，在一个系统时钟后，由硬件自动清零。 **注意:** 接收移位寄存器不会被清除。
[7]	**TXUFIEN**	**发送下溢中断使能位** 该位用来使能下溢中断。在从机模式，当发送下溢事件发生时，TXUFIF (SPIx_STATUS[19])位将被置1。 0 = 禁止从机发送下溢中断

		1 = 使能从机发送下溢中断
[6]	**TXUFPOL**	**TX下溢数据极性** 0 = 在从机模式，如果有发送下溢事件发生，SPI数据输出保持为0。 1 = 在从机模式，如果有发送下溢事件发生，SPI数据输出保持为1。 **注意 1:**当从机片选信号激活时，如果发送FIFO没有任何数据,发送下溢事件发生。 **2:** 在I^2S 模式，这位应该设置为0.. **3**. 当发送下溢事件发生, 虽然TX FIFO不为空，SPxI_MISO管脚状态将由设置决定。存储在TX FIFO 中的数据将在下一次传输数据的时候通过SPIx_MISO管脚送出.
[5]	**RXOVIEN**	**接收FIFO 溢出中断使能位** 0 = 禁止接收FIFO 溢出中断 1 = 使能接收FIFO 溢出中断
[4]	**RXTOIEN**	**从机接收超时中断使能位** 0 = 禁止接收超时中断 1 = 使能接收超时中断
[3]	**TXTHIEN**	**发送FIFO 阀值中断使能位** 0 = 禁止发送FIFO 阀值中断 1 = 使能发送FIFO 阀值中断
[2]	**RXTHIEN**	**接收FIFO阀值中断使能位** 0 = 禁止接收FIFO阀值中断 1 = 使能接收FIFO阀值中断
[1]	**TXRST**	**发送复位 (仅SPI模式支持)** 0 = 无影响 1 =复位发送FIFO指针和发送电路。TXFULL位将被清零，TXEMPTY位将被置1。该位设置为1后，大约经过3个系统时钟周期和2个外设时钟周期，硬件将该位清零。如果用户想确认复位是否已经完成，可以读取TXRXRST (SPIx_STATUS[23])位来确认。 **注意:** 如果在 SPI 从机模式下 TX 下溢事件发生, 该位能使 SPI 返回到空闲状态
[0]	**RXRST**	**接收复位 (仅SPI模式支持)** 0 = 无影响 1 =复位接收 FIFO 指针和接收电路。RXFULL 位将被清零，RXEMPTY 位将被置1。该位设置为1后，大约经过3个系统时钟周期和2个外设时钟周期，硬件将该位清零。如果用户想确认复位是否已经完成，可以读取 TXRXRST (SPIx_STATUS[23])位来确认。

SPI 状态寄存器(SPIx_STATUS)

寄存器	偏移	R/W	描述	复位值
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI 状态寄存器	0x0005_0110

注意: 不支持 I^2S 模式

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
23	22	21	20	19	18	17	16
TXRXRST	Reserved			TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
SPIENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	Reserved	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

位	描述	
[31:28]	**TXCNT**	**发送 FIFO 数据个数 (只读)** 该位域表明发送FIFO缓存的有效数据个数。
[27:24]	**RXCNT**	**接收 FIFO 数据个数 (只读)** 该域表明接收 FIFO 缓存有效数据个数..
[23]	**TXRXRST**	**发送或接收复位状态(只读)** 0 = TXRST 或 RXRST 的复位功能已经完成 1 = TXRST 或 RXRST 正在复位 **注意:** TXRST 和 RXRST的复位操作都需要3个系统时钟周期+2个外设时钟周期。用户可以检查该状态位来检测复位功能是否完成。
[22:20]	**Reserved**	保留.
[19]	**TXUFIF**	**发送下溢中断标志** 当发送下溢事件发生时，该位被设置为1，数据输出管脚的状态取决于TXUFPOL的设置。 0 = 无影响 1 = 当从机片选信号有效时，发送FIFO和发送移位寄存器中没有数据。 **注意 1:** 该位通过写1清除。 **注意 2:** 当从机片选信号有效时，如果复位从机的发送电路，在3个系统时钟周期和2个外设时钟周期后，此时复位操作已经完成，该位将被设置为1。
[18]	**TXTHIF**	**发送FIFO阀值中断标志(只读)** 0 = 发送FIFO缓存中的有效数据个数大于TXTH设置的值。 1 = 发送FIFO缓存中的有效数据个数少于或等于TXTH设置的值
[17]	**TXFULL**	**发送FIFO缓存满标志 (只读)** 0 = 发送FIFO 缓存未满 1 = 发送FIFO 缓存已满
[16]	**TXEMPTY**	**发送FIFO缓存空标志 (只读)**

		0 = 发送FIFO 缓存不空 1 = 发送FIFO 缓存已空
[15]	**SPIENSTS**	**SPI 使能状态位(只读)** 0 = SPI控制器被禁止. 1 = SPI控制器被使能. **注意:** SPI外设时钟与系统时钟不同步。为了确保SPI控制器已经被禁止，该位指示了SPI控制器的真实状态。
[14:13]	**Reserved**	保留
[12]	**RXTOIF**	**接收超时中断标志** 0 = 没有接收FIFO超时事件 1 = 接收 FIFO 缓存非空并且主机模式下超过64个SPI时钟周期或从机模式下超过576个SPI外设时钟周期，接收FIFO缓存上没有读操作。当接收FIFO缓存被软件读取，超时状态会自动清0。 **注意:** 该位通过写1清除
[11]	**RXOVIF**	**接收FIFO溢出中断标志** 当接收FIFO缓存已经满了，接下来的数据将被丢失，同时该位被设置为1. 0 = 接收FIFO没有溢出 1 = 接收FIFO溢出 **注意:** 该位通过写1清除
[10]	**RXTHIF**	**接收FIFO阀值中断标志(只读)** 0 = 接收FIFO缓存的有效数据个数少于或等于RXTH设置的值 1 = 接收FIFO缓存的有效数据个数大于RXTH设置的值
[9]	**RXFULL**	**接收FIFO 缓存满标志(只读)** 0 = 接收 FIFO 缓存未满 1 = 接收 FIFO 缓存已满
[8]	**RXEMPTY**	**接收FIFO 缓存空标志 (只读)** 0 = 接收 FIFO 缓存不空 1 = 接收 FIFO 缓存为空
[7]	**SLVURIF**	**从机模式发送下溢中断标志** 在从机模式，如果发送下溢事件发生，而且从机片选线进入无效状态，该中断标志将被设置为1。 0 = 未发生从机发送下溢事件 1 = 发生从机发送下溢事件 **注意:** 该位通过写1清除
[6]	**SLVBEIF**	**从机模式位计数错误中断标志** 在从机模式，当从机片选信号线进入无效状态时，如果位计数与DWIDTH不一致，该中断标志将被设置为1。 0 = 未发生从机模式位计数错误事件 1 = 发生从机模式位计数错误事件 **注意:** 如果从机片选激活，但是没有总线时钟输入，当从机片选进入无效状态时，SLVBCEIF也将被设置为1.该位通过写1清除。
[5]	**Reserved**	保留.
[4]	**SSLINE**	**从机片选线总线状态(只读)**

		0 = 从机片选线状态为0 1 = 从机片选线状态为1 **注意:** 该位只在从机模式有效。如果SSACTPOL (SPIx_SSCTL[2])设置为0，而且SSLINE的值为1，SPI从机片选处于无效状态。
[3]	**SSINAIF**	**从机片选无效中断标志** 0 = 从机片选无效中断被清除或没有发生 1 = 发生了从机片选无效中断 **注意:** 该位只在从机模式有效，该位通过写1清除。
[2]	**SSACTIF**	**从机片选激活中断标志** 0 = 从机片选激活中断被清除或未发生 1 = 发生了从机片选激活中断 **注意:** 该位只在从机模式有效，该位通过写1清除。
[1]	**UNITIF**	**单元传输中断标志** 0 = 该位清除后，没有事务完成 1 = SPI 控制器已经完成一个单元传输 **注意:** 该位通过写1清除
[0]	**BUSY**	**忙状态(只读)** 0 = SPI 控制器处于空闲状态 1 = SPI 控制器处于忙状态 下面列出忙状态的条件： a. (SPIx_CTL[0] = 1) 和 TXEMPTY = 0. b. 在主机模式， (SPIx_CTL[0]) = 1 和 TXEMPTY = 1但是当前传输还没有完成. c. 在主机模式， (SPIx_CTL[0]) = 1 和 RXONLY = 1. d. SPI 从机模式时， (SPIx_CTL[0] = 1)且当从机片选激活时，SPI内部逻辑有串行时钟输入。 **e.** SPI 从机模式时， (SPIx_CTL[0] = 1)且传输缓存或传输移位寄存器不为空，即使从机片选无效

SPI 数据发送寄存器(SPIx_TX)

寄存器	偏移	R/W	描述	复位值
SPIx_TX	SPIx_BA+0x20	W	SPI 数据传输寄存器	0x0000_0000

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

位	描述	
[31:0]	**TX**	**数据发送寄存器** 数据发送寄存器把数据传送到4级发送FIFO缓存。数据的有效长度根据SPI模式时DWIDTH (SPIx_CTL[12:8])的设置或I2S模式时WDWIDTH (SPIx_I2SCTL[5:4])的设置。 在SPI模式，如果DWIDTH为0x08 ，则TX[7:0]位将会被发送。如果DWIDTH为0x00，则SPI 控制器会执行32位数据传输。 在I^2S 模式，如果 WDWIDTH (SPIx_I2SCTL[5:4]) 为 0x2，音频通道数据宽度是24位，则TX[23:0]被发送. 如果WDWIDTH为0x0, 0x1,或 0x3，可以参考I^2S 模式下FIFO 数据发送规范。 **注意**:在主机模式，在数据写入到该寄存器后，SPI控制器在1个APB时钟周期和6个外设时钟周期后开始传输数据

SPI 数据接收寄存器(SPIx_RX)

寄存器	偏移	R/W	描述	复位值
SPIx_RX	SPIx_BA+0x30	R	SPI 数据接收寄存器	0x0000_0000

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

位	描述	
[31:0]	**RX**	**数据接收寄存器（只读）** SPI 控制器有 4 级 FIFO 缓存。从 SPI 数据输入引脚接收到的数据会被保存到该寄存器。如果 RXEMPTY (SPIx_STATUS[8] 或 SPIx_I2SSTS[8]) 不为 1，软件可以通过读该寄存器获取 FIFO 缓存的数据

I²S 控制寄存器(SPIx_I2SCTL)

寄存器	偏移	R/W	描述	复位值
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I²S控制寄存器	0x0000_0000

注: 该寄存器不支持 SPI 模式

31	30	29	28	27	26	25	24
SLVERRIEN	**Reserved**	**FORMAT**		**Reserved**		**LZCIEN**	**RZCIEN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXLCH	**Reserved**					**LZCEN**	**RZCEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MCLKEN	**Reserved**						**SLAVE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ORDER	**MONO**	**WDWIDTH**		**MUTE**	**RXEN**	**TXEN**	**I2SEN**

位	描述	
[31]	**SLVERRIEN**	**从机模式位时钟丢失中断使能位** 如果该位为1且位时钟丢失产生中断 0 = 中断禁止. 1 = 中断使能.
[30]	**Reserved**	保留.
[29:28]	**FORMAT**	**数据格式选择** 00 = I²S 数据格式 01 = MSB 对齐数据格式 10 = PCM 模式 A. 11 = PCM 模式 B.
[27:26]	**Reserved**	保留
[25]	**LZCIEN**	**左声道过零检测中断使能位** 如果该位置 1，左声道检测到过零事件时发生中断 0 = 关闭中断 1 = 使能中断
[24]	**RZCIEN**	**右声道过零检测中断使能位** 如果该位置 1，右声道检测到过零事件时发生中断 0 = 关闭中断 1 = 使能中断
[23]	**RXLCH**	**右声道接收使能位** I²S 控制器处于单声道模式 (MONO = 1) 时，如果 RXLCH 为 0，I²S 控制器接收右声道的数据；如果 RXLCH 为 1，I²S 控制器接收左声道的数据。 0 = 单声道模式下，接收右声道的数据 1 = 单声道模式下，接收左声道的数据

[22:18]	**Reserved**	保留
[17]	**LZCEN**	**左声道过零检测使能位** 如果该位为 1，当左声道数据信号位改变或下一笔数据位全为 0 ，SPIx_I2SSTS 寄存器的 LZCIF 会被置 1。该功能仅在发送操作下有效。 0 = 左声道过零检测关闭 1 = 左声道过零检测使能
[16]	**RZCEN**	**右声道过零检测使能位** 如果该位为 1，当右声道数据信号位改变或下一笔数据位全为 0 ，SPIx_I2SSTS 寄存器的 RZCIF 会被置 1。该功能仅在发送操作下有效。 0 = 右声道过零检测关闭 1 = 右声道过零检测使能
[15]	**MCLKEN**	**主时钟使能位** 如果 MCLKEN 置 1，I²S 控制球会在 SPIx_I2SMCLK 引脚产生主时钟以供外部音频设备使用 0 = 主时钟关闭 1 = 主时钟使能
[14:9]	**Reserved**	保留
[8]	**SLAVE**	**从机模式** I²S 可以工作在主机模式或从机模式。主机模式时，I2Sx_BCLK 和 I2Sx_LRCLK 引脚都为输出模式，而且可以通过该芯片为音频解码芯片发送位时钟。从机模式时，I2Sx_BCLK 和 I2Sx_LRCLK 引脚都为输入模式，而且 I2Sx_BCLK 和 I2Sx_LRCLK 的信号都来自外部的音频解码芯片。 0 = 主机模式 1 = 从机模式
[7]	**ORDER**	**FIFO 内双通道数据顺序** 0 = 左声道位于高字节 1 = 左声道位于低字节
[6]	**MONO**	**单通道数据** 0 = 双通道的数据格式 1 = 单通道的数据格式
[5:4]	**WDWIDTH**	**字宽度** 00 = 数据宽度为 8 位 01 = 数据宽度为 16 位 10 = 数据宽度为 24 位 11 = 数据宽度为 32 位
[3]	**MUTE**	**发送静音使能位** 0 = 发送数据来自缓存 1 = 发送数据为 0
[2]	**RXEN**	**接收使能位** 0 = 数据接收关闭 1 = 数据接收使能

[1]	**TXEN**	**发送使能位** 0 = 数据发送关闭 1 = 数据发送使能
[0]	**I2SEN**	**I^2S 控制器使能位** 0 = I^2S 模式禁止 1 = I2S 模式使能 **注:** 1. 主机模式下使能该位后，I2Sx_BCLK 将开始输出 2. 用户在更改寄存器 SPIx_I2SCTL, SPIx_I2SCLK, 和 SPIx_FIFOCTL 前，需要清 0 I2SEN (SPIx_I2SCTL[0]) 并确认 I2SENSTS (SPIx_I2SSTS[15]) 为 0。

I²S时钟分频控制寄存器(SPIx_I2SCLK)

寄存器	偏移	R/W	描述	复位值
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I²S时钟分频控制寄存器	0x0000_0000

注: 该寄存器不支持 SPI 模式

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

位	描述	
[31:18]	**Reserved**	保留.
[17:8]	**BCLKDIV**	**位时钟除频** 主机模式下位时钟由 I²S 控制器产生。位时钟和 f_{BCLK} 有以下关系： $f_{BCLK} = \frac{f_{i2s_clock_src}}{2 \times (\mathrm{BCLKDIV} + 1)}$ 其中 $f_{i2s_clock_src}$ 为 I²S 外设时钟源，由寄存器 CLK_CLKSEL2 定义。 从机模式下，该域用来定义外设时钟的频率，公式为： $f_{i2s_clock_src} \div \left(\frac{\mathrm{BCLKDIV}}{2} + 1\right)$. I²S 从机模式外设时钟频率必须等于或大于 6 倍输入位时钟的频率。 **注:** 软件复位释放SPI IP和设置这个时钟分频寄存器的时间间隔必须大于或等于5个外设时钟周期。
[7]	**Reserved**	保留.

[6:0]	**MCLKDIV**	**主时钟除频** MCLKEN 为 1 时， I²S 控制器会产生时钟到外部音频设备。主时钟，f_{MCLK} 由下列公式决定： MCLKDIV >= 1,. $f_{MCLK} = \frac{f_{i2s_clock_src}}{2 \times \mathrm{MCLKDIV}}$ MCLKDIV = 0,. $f_{MCLK} = f_{i2s_clock_src}$ 其中， $f_{i2s_clock_src}$ 为 I²S 外设时钟频率，由 CLK_CLKSEL2 决定。通常来说，主时钟的频率为采样时钟频率的 256 倍。

注: 谨慎配置 **BCLKDIV**， 频率不能超过系统时钟

I²S 状态寄存器(SPIx_I2SSTS)

寄存器	偏移	R/W	描述	复位值
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I²S状态寄存器	0x0005_0100

注: 该寄存器不支持 SPI 模式

31	30	29	28	27	26	25	24
Reserved	**TXCNT**			**Reserved**	**RXCNT**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXRXRST	**SLVERRIF**	**LZCIF**	**RZCIF**	**TXUFIF**	**TXTHIF**	**TXFULL**	**TXEMPTY**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
I2SENSTS	**Reserved**		**RXTOIF**	**RXOVIF**	**RXTHIF**	**RXFULL**	**RXEMPTY**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**RIGHT**	**Reserved**			

位	描述	
[31]	**Reserved**	保留
[30:28]	**TXCNT**	**发送 FIFO 数据计数 (只读)** 该位表示发送 FIFO 缓存的有效数据个数
[27]	**Reserved**	保留
[26:24]	**RXCNT**	**接收 FIFO 数据计数 (只读)** 该位表示接收 FIFO 缓存的有效数据个数
[23]	**TXRXRST**	**TX 或 RX 复位状态 (只读)** 0 = TXRST 或 RXRST 复位完成 1 = TXRST 或 RXRST 复位中 **注:** TXRST 或 RXRST 复位需要花费 3 个系统时钟周期 + 2 个外设时钟周期。用户可以检查该位来判断复位是否完成。
[22]	**SLVERRIF**	保留
[21]	**LZCIF**	**左通道过零中断标志位** 0 = 左通道未发生过零事件 1 = 左通道已发生过零事件
[20]	**RZCIF**	**右通道过零中断标志位** 0 = 右通道未发生过零事件 1 = 右通道已发生过零事件
[19]	**TXUFIF**	**发送 FIFO 下溢中断标志** 如果在发送 FIFO 缓存为空， FIFO 缓存里也没有数据写入，此时如果仍有总线时钟输入，该位置 1 **注:** 该位写 1 清 0
[18]	**TXTHIF**	**发送 FIFO 阈值中断标志 (只读)**

		0 = 传送 FIFO 缓存中的有效数据个数大于 TXTH 设置的值 1 = 传送 FIFO 缓存中的有效数据个数小于或等于 TXTH 设置的值 **注:** 如果 TXTHIEN = 1 且 TXTHIF = 1, SPI/I^{2}S 控制器会产生一个 SPI 中断请求
[17]	**TXFULL**	**发送 FIFO 缓存满标志位 (只读)** 0 = 发送 FIFO 缓存未满 1 = 发送 FIFO 缓存已满
[16]	**TXEMPTY**	**发送 FIFO 缓存空标志位 (只读)** 0 = 发送 FIFO 缓存非空 1 = 发送 FIFO 缓存已空
[15]	**I2SENSTS**	**I^{2}S 使能状态 (只读)** 0 = SPI/I^{2}S 控制逻辑关闭 1 = SPI/I^{2}S 控制逻辑使能 **注:** SPI/I2S 外设时钟与系统时钟不同步。为了确保 SPI/I2S 控制器已经被禁止，该位指示了 SPI/I2S 控制器的真实状态。
[14:13]	**Reserved**	保留
[12]	**RXTOIF**	**接收超时中断标志** 0 = 无接收 FIFO 超时中断事件 1 = 接收 FIFO 缓存非空且主机模式下超过 64 个 SPI 时钟周期或从机模式下超过 576 个 SPI 引擎时钟周期，接收 FIFO 缓存上没有读操作。当接收 FIFO 缓存被软件读取，超时状态会自动清 0。 **注:** 该位写 1 清 0
[11]	**RXOVIF**	**接收 FIFO 下溢中断标志** 如果接收 FIFO 缓存已满，之后接收的数据会被丢弃且该位置 1。 **注:** 该位写 1 清 0
[10]	**RXTHIF**	**接收 FIFO 阈值中断标志 (只读)** 0 = 接收 FIFO 缓存的有效数据个数少于或等于 RXTH 设置的值 1 = 接收 FIFO 缓存的有效数据个数大于 RXTH 设置的值 **注:** 如果 RXTHIEN = 1 且 RXTHIF = 1, SPI/I2S 控制器会产生一个 SPI 中断请求
[9]	**RXFULL**	**接收 FIFO 缓存满标志 (只读)** 0 = 接收 FIFO 缓存未满 1 = 接收 FIFO 缓存已满
[8]	**RXEMPTY**	**接收 FIFO 缓存空标志 (只读)** 0 = 接收 FIFO 缓存非空 1 = 接收 FIFO 缓存已空
[7:5]	**Reserved**	保留
[4]	**RIGHT**	**右声道 (只读)** 该位指示当前传输的数据是哪一个声道 0 = 左声道 1 = 右声道
[3:0]	**Reserved**	保留

6.15 QSPI接口 (QSPI)

6.15.1 概述

QSPI接口是全双工同步串行数据通讯接口，可做为主机或从机，用4线进行双向通讯。M031包含1组QSPI控制器用于执行从外设接收到的数据串行到并行转换和发送给外设数据的并行到串行转换。

QSPI支持全双工的2位传输模式，也支持双I/O和四I/O传输模式。QSPI支持PDMA传输功能。

6.15.2 特征

- 支持一组QSPI控制器
- 支持主机模式和从机模式
- 总线时钟主机模式最高可到24MHz,从机模式最高可到16MHz(当芯片工作在Vdd=1.8~3.6V)
- 支持2位传输模式
- 支持双I/O和四I/O传输模式
- 传输位长可为8~32位
- 提供独立的8级深度的发送和接收FIFO缓存
- 支持高位（MSB）或者低位(LSB)在前传输顺序
- 支持字节重排功能
- 支持字节或字暂停模式
- 支持PDMA传输
- 支持三线模式，无从机片选的双向接口
- 支持单数据通道半双工传输
- 支持只接收模式

6.15.3 框图

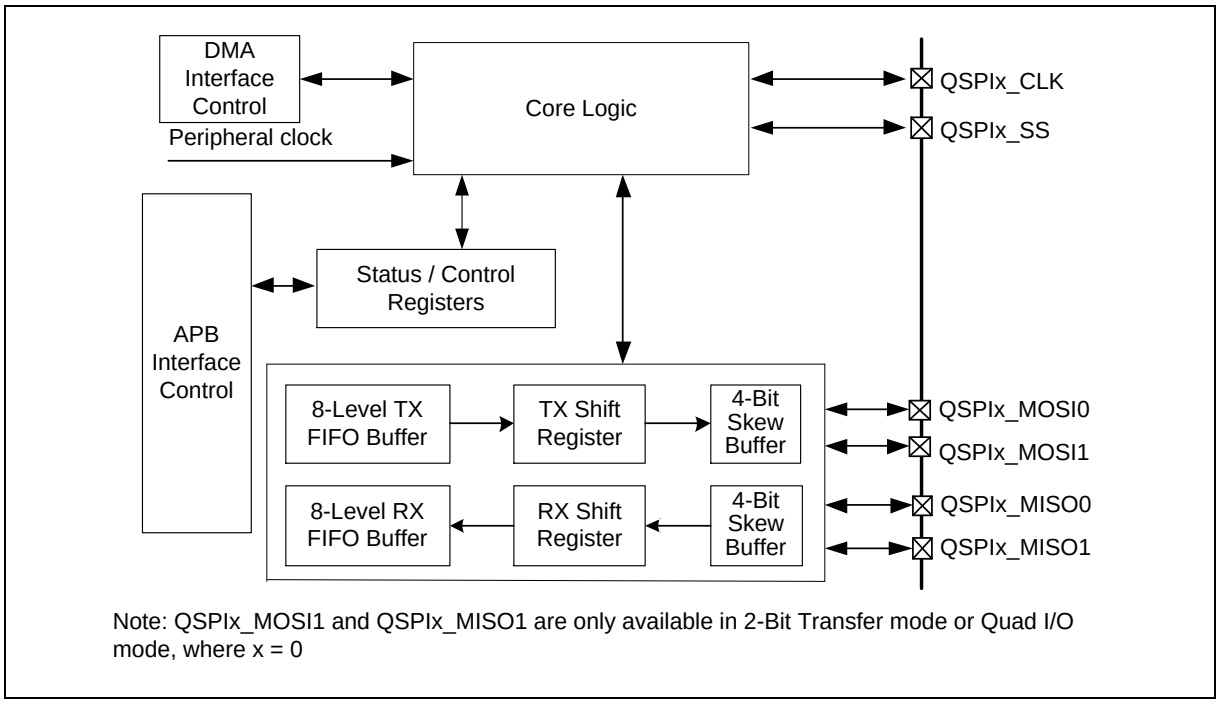


图 6.15-1 QSPI 框图

TX FIFO 缓存:

发送FIFO8级深度，32位宽，先进先出。数据通过软件写入QSPIx_TX寄存器后再存入FIFO缓存。

RX FIFO 缓存:

接收FIFO8级深度，32位宽，先进先出。接收控制逻辑器保存接收数据在FIFO缓存，软件通过读取QSPIx_RX寄存器可以得到FIFO缓存数据。

TX 移位寄存器:

发送移位寄存器是一个32位的寄存器。发送数据首先从TX FIFO缓存加载到这里，并一位一位的移到斜移缓存。

RX 移位寄存器:

接收移位寄存器是一个32位的寄存器。接收数据从斜移缓存一位一位的移到这里，当一个事务完成时再载入到RX FIFO。

斜移缓存:

斜移缓存是一个4级1位的缓存。总共有2个斜移缓存，分别在发送和接收端。在接收端，斜移缓存用于把数据从QSPI总线移位到RX移位寄存器。在发送端，斜移缓存用于把数据从TX移位寄存器移位到QSPI总线。

6.15.4 基本配置

6.15.4.1 QSPI0基本配置

- 时钟配置
  - 寄存器QSPI0SEL(CLK_CLKSEL2[3:2])配置QSPI0的时钟
  - 寄存器QSPI0CKEN(CLK_APBCLK0[12])使能QSPI0的时钟
- 复位配置
  - 寄存器QSPI0RST(SYS_IPRST1[12])复位QSPI0控制器.
- 管脚配置

组	管脚名	GPIO	MFP
QSPI0	QSPI0_CLK	PA.2, PH.8	MFP3
		PC.2	MFP4
		PF.2	MFP5
		PC.14	MFP6
	QSPI0_MISO0	PA.1, PE.1	MFP3
		PC.1	MFP4
	QSPI0_MISO1	PA.5, PH.10	MFP3
		PC.5	MFP4
		PB.1	MFP10
	QSPI0_MOSI0	PA.0, PE.0	MFP3
		PC.0	MFP4
	QSPI0_MOSI1	PA.4, PH.11	MFP3
		PC.4	MFP4
		PB.0	MFP10
	QSPI0_SS	PA.3, PH.9	MFP3
		PC.3	MFP4

6.15.5 功能描述

6.15.5.1 术语

QSPI 外设时钟和QSPI 总线时钟

QSPI控制器需要外设时钟来驱动QSPI逻辑单元执行数据传输。QSPI时钟决定于时钟分频器(QSPIx_CLKDIV)和时钟源的设置，时钟源可以设置为HXT, PLL, PCLK 或HIRC。寄存器CLK_CLKSEL2的QSPIxSEL位选择QSPI的时钟源。寄存器DIVIDER(QSPIx_CLKDIV[8:0])的值决定时钟的分频值。

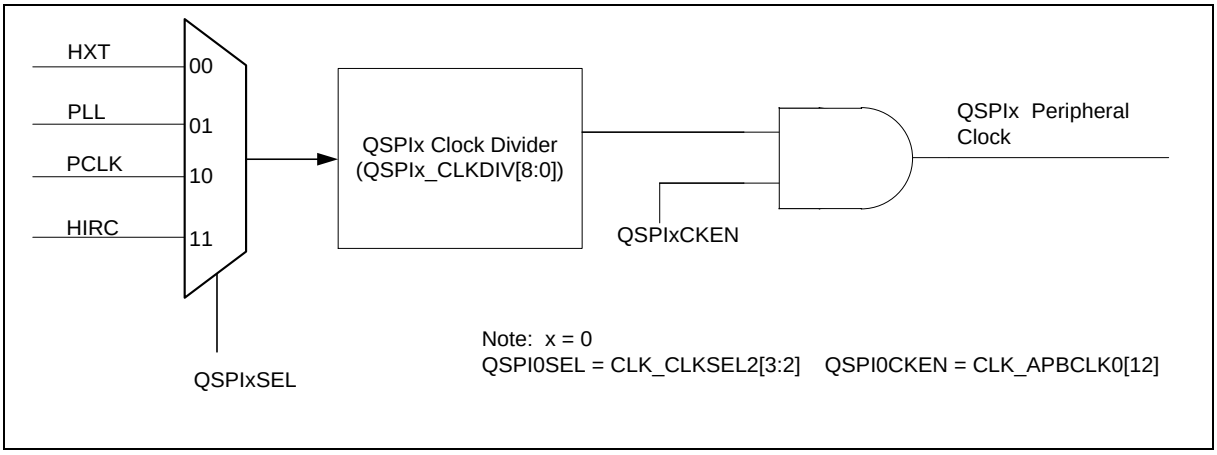


图 6.15-2 QSPI 外设时钟

QSPI主机模式下，QSPI总线的时钟频率等于外设的时钟速率。通常QSPI总线时钟就表示QSPI时钟。从机模式下，QSPI总线时钟由主设备提供。无论是主机模式还是从机模式，QSPI外设时钟都不能快于系统时钟。

主机/从机模式

QSPI控制器可以通过寄存器SLAVE(QSPIx_CTL[18])设成主机模式或从机模式。QSPI传送过程中，通过HALFDPX(QSPIx_CTL[14])可以切换全双工模式或半双工模式。主机模式和从机模式的应用框图如下所示：

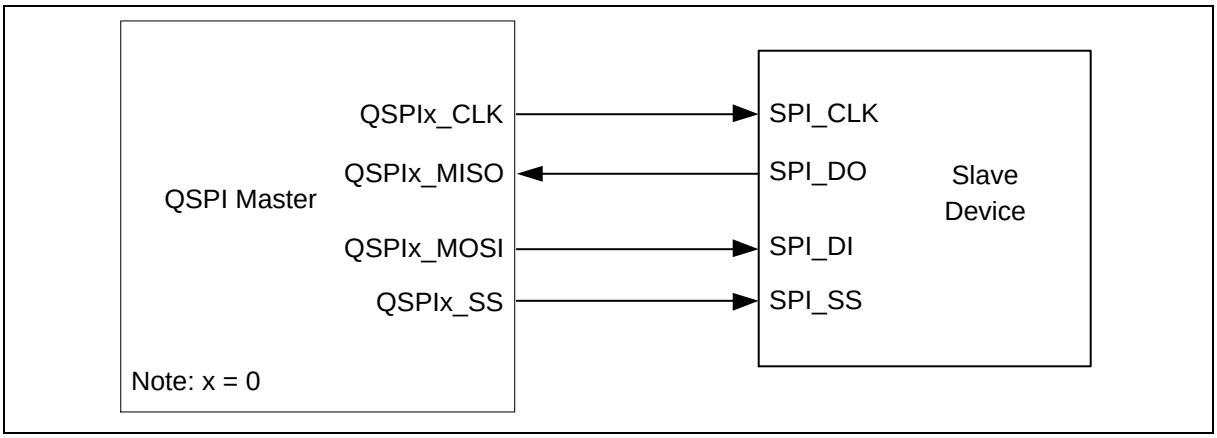


图 6.15-3 QSPI QSPI 全双工主机模式应用框图

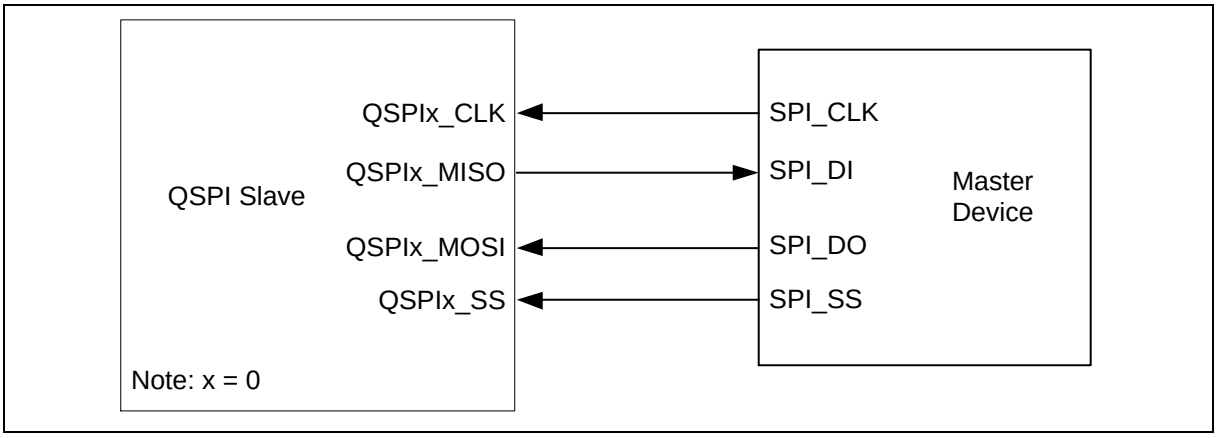


图 6.15-4 QSPI QSPI 全双工从机模式应用框图

从机选择

主机模式下，QSPI通过从机片选QSPIx_SS（输出）使能从机设备。从机模式，其它主机通过QSPIx_SS（输入）使能QSPI控制器。片选有效边沿到第一个QSPI时钟输入应多于3个SPI外设时钟周期。

在主机模式，从机片选信号可通过寄存器SSACTPOL(QSPIx_SSCTL[2])设定为低有效或高有效。从机片选信号的有效状态根据所连接的设备类型决定。在从机模式，从机片选信号的无效时间必须大于等于3个外设时钟周期

时序条件

CLKPOL(QSPIx_CTL[3])定义QSPI时钟空闲时的电平。如果CLKPOL=1，QSPI时钟空闲时输出高电平，如果CLKPOL=0，则QSPI时钟空闲时输出低电平。

TXNEG(QSPIx_CTL[2])定义数据是在QSPI时钟的下沿发送还是上沿发送。RXNEG(QSPIx_CTL[1])定义数据在QSPI时钟的下沿接收还是上沿接收。

注: TXNEG和RXNEG的设置是互斥的，收发不能都配置为上沿或都是下沿。

发送/接收位长

位长由DWIDTH(QSPIx_CTL[12:8])来配置，最多可为32位。

当QSPI完成一次DWIDTH(QSPIx_CTL[12:8])定义的位长的收/发时，传输中断标志UNITIF(QSPIx_STATUS[1])将被置1

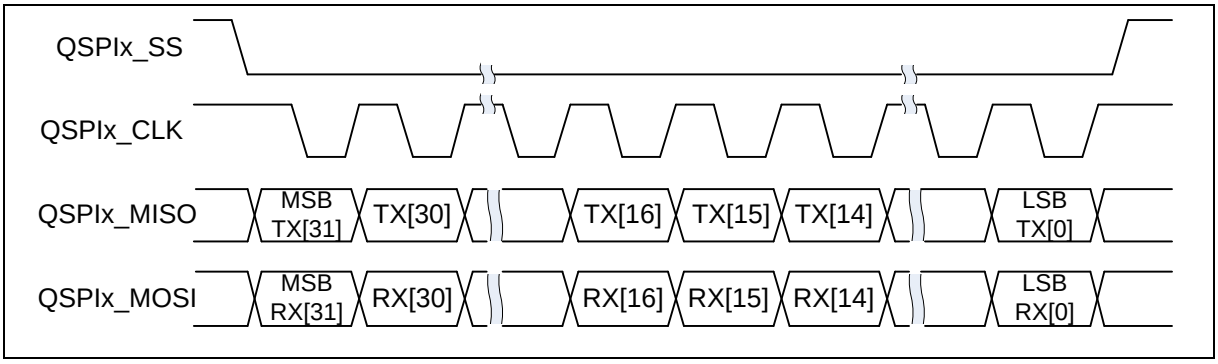


图 6.15-5 一次 32-位的传输事务

LSB/MSB 在前

LSB(QSPIx_CTL[13]) 定义传输顺序。LSB(QSPIx_CTL[13])=1 时，低位会先传输。若(QSPIx_CTL[13])=0，则先传高位。

休眠间隔

主机模式下，SUSPITV(QSPIx_CTL[7:4])可配置在两次连续传输之间，插入0.5~15.5个QSPI时钟周期的空闲：空闲时间的计算是从前一次传输的最后一个时钟沿，到下一次传输的第一个时钟沿。SUSPITV的默认值是0x3(3.5个QSPI时钟周期)。

6.15.5.2 自动从机选择

主机模式下，如果AUTOSS(QSPIx_SSCTL[3])置位，从机选择信号根据SS(QSPIx_SSCTL[0])自动输出到QSPIx_SS管脚上。当数据写入FIFO数据传输启动，从机选择信号自动设置为有效状态。当QSPI总线空闲时，从机选择信号将自动变为无效状态。QSPI总线不空闲时（比如：TX FIFO、TX移位寄存器或TX斜移缓存不空），在SUSPITV(QSPIx_CTL[7:4])的值大于或等于3的条件下，从机选择信号在两次传输之间将被置为无效状态。

主机模式下，如果SUSPITV的值小于3且AUTOSS置1时，两次连续传输之间，从机选择信号将保持有效。

如果AUTOSS被清零，从机选择输出信号的有效电平由SSACTPOL(QSPIx_SSCTL[2])来定义。

从机片选信号有效边沿，到第一个QSPI时钟边沿的间隔为1个QSPI总线时钟周期。QSPI总线的最后一个时钟，到从机选择信号无效边沿的间隔为1.5个QSPI总线时钟周期。

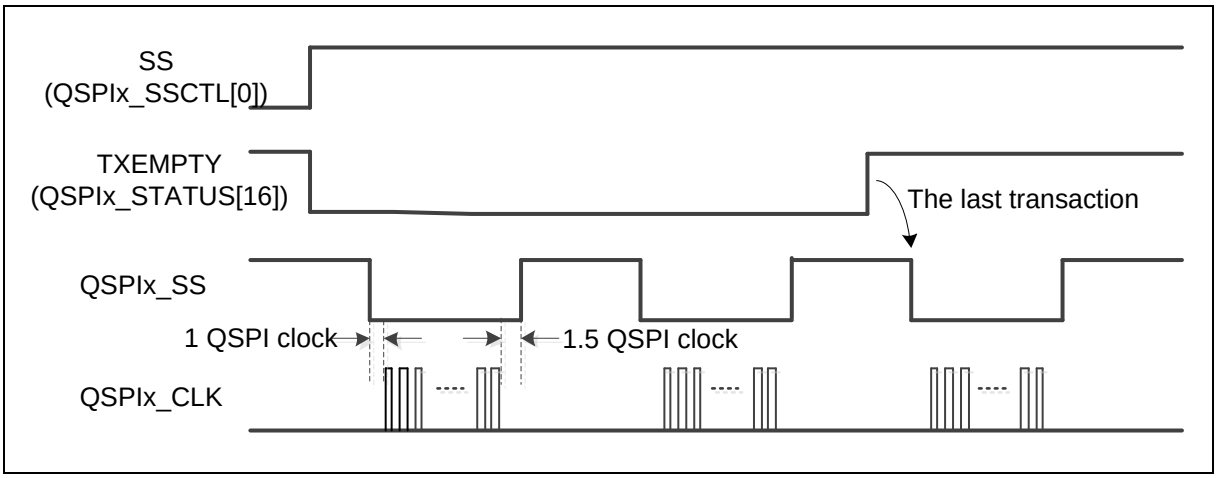


图 6.15-6 自动从机选择(SSACTPOL = 0, SUSPITV > 0x2)

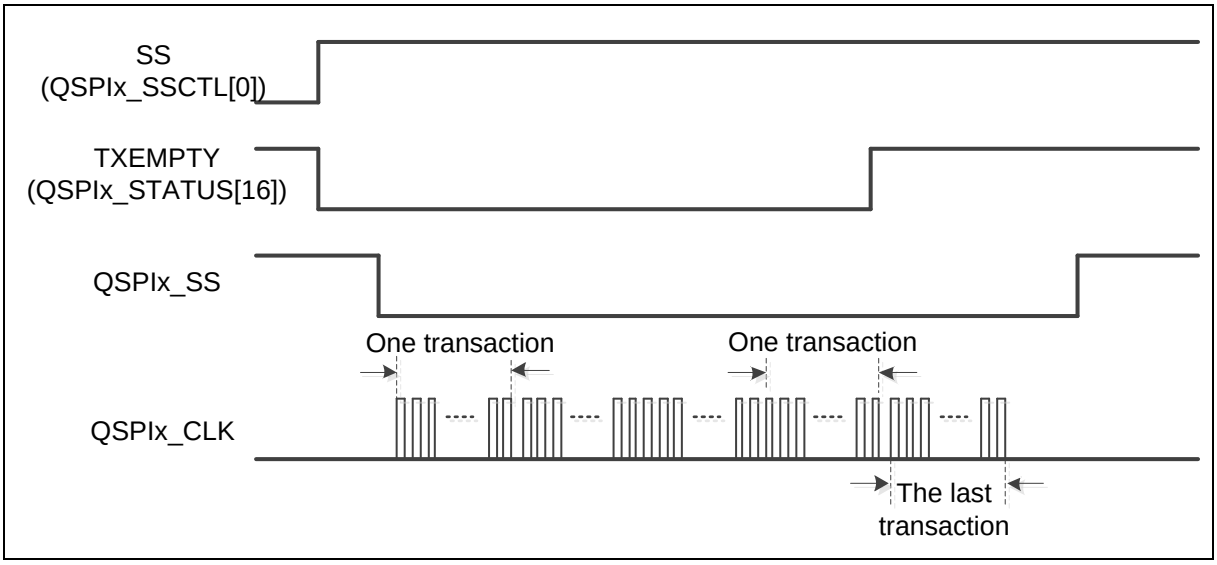


图 6.15-7 自动从机选择(SSACTPOL = 0, SUSPITV < 0x3)

6.15.5.3 字节重排和挂起功能

当传输设置为MSB优先(LSB=0)且使能了REORDER(QSPIx_CTL[19])，在32位传输模式(DWIDTH=0)下，存储在TX缓存与RX缓存的数据将按[BYTE0,BYTE1,BYTE2,BYTE3]的顺序重新排列。数据发送/接收的顺序为BYTE0,BYTE1,BYTE2,和BYTE3。如果DWIDTH设为24位传输模式，存储在TX缓存与RX缓存的数据将按[未知字节,BYTE0,BYTE1,BYTE2]的顺序重新排列。QSPI控制器将按照BYTE0,BYTE1,BYTE2的顺序发送/接收数据，每个字节MSB优先发送/接收。16位传输模式的规则与上面相同。字节重排序功能只适用于DWIDTH为16,24,和32位时。

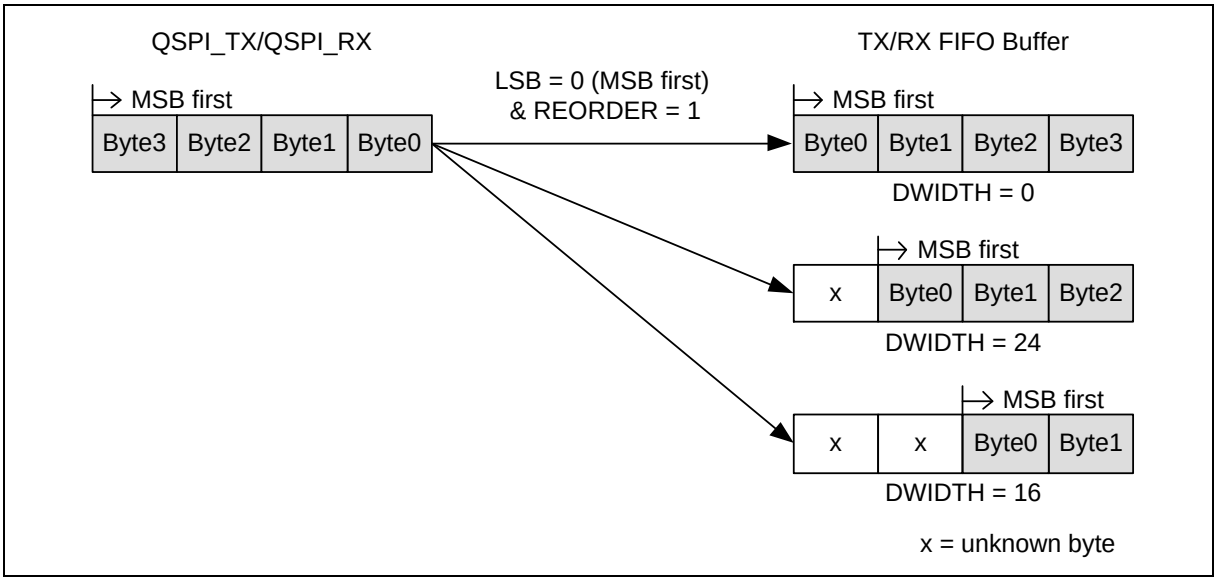


图 6.15-8 字节重排功能

主机模式下，如果REORDER(QSPIx_CTL[19])为1，两次连续传输字节之间将插入0.5~15.5个QSPI时钟周期的挂起间隔。挂起间隔时间由SUSPITV(QSPIx_CTL[7:4])设置。

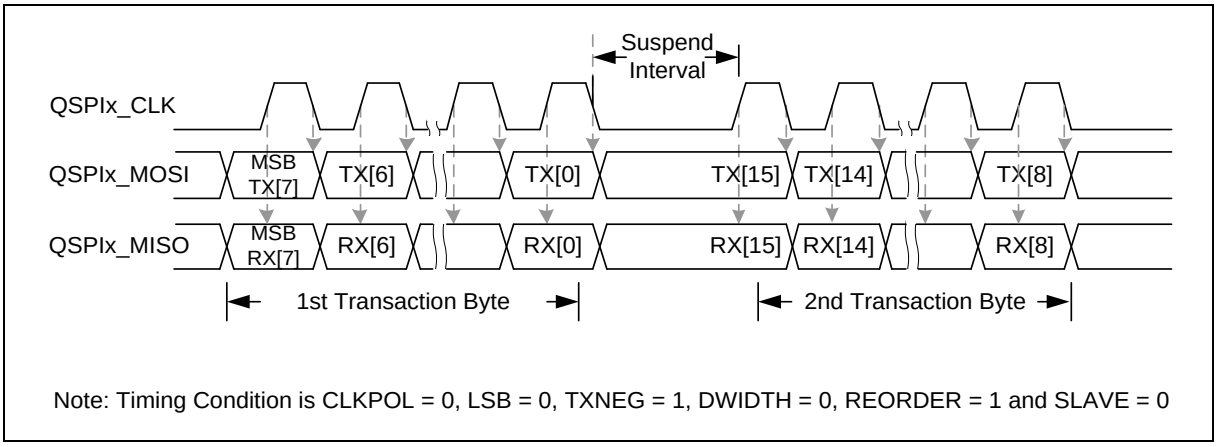


图 6.15-9 字节挂起波形图

6.15.5.4 半双工通讯

设置HALFDPX(QSPIx_CTL[14])位可以使能QSPI工作在半双工模式。半双工模式下，仅有一条数据线进行发送或接收，方向由DATDIR(QSPIx_CTL[20])配置。半双工模式下，没有用到的QSPIx_MISO管脚可以配置成GPIO供其他功能使用。使能或关闭HALFDPX(QSPIx_CTL[14])控制位将会同时产生TXFBCLR(QSPIx_FIFOCTL[9])和RXFBCLR(QSPIx_FIFOCTL[8])。

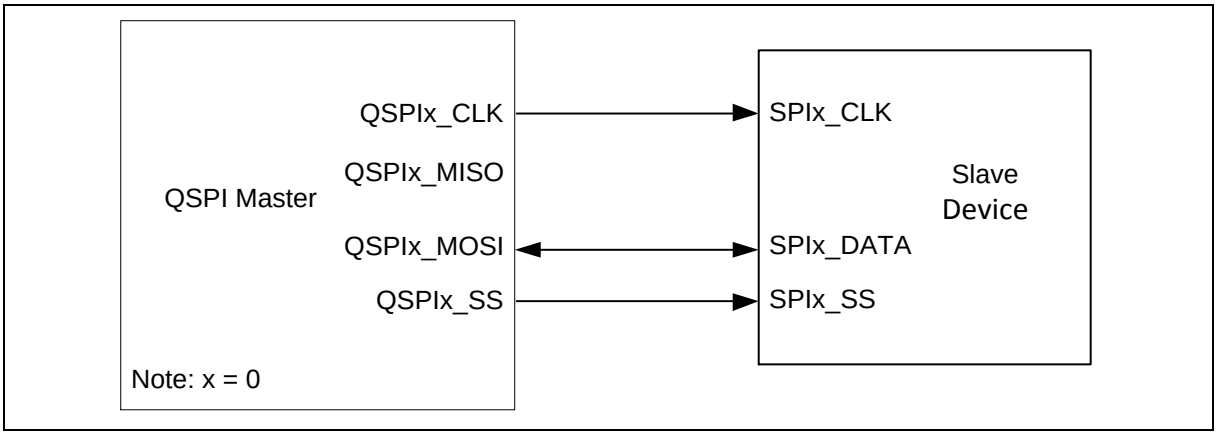
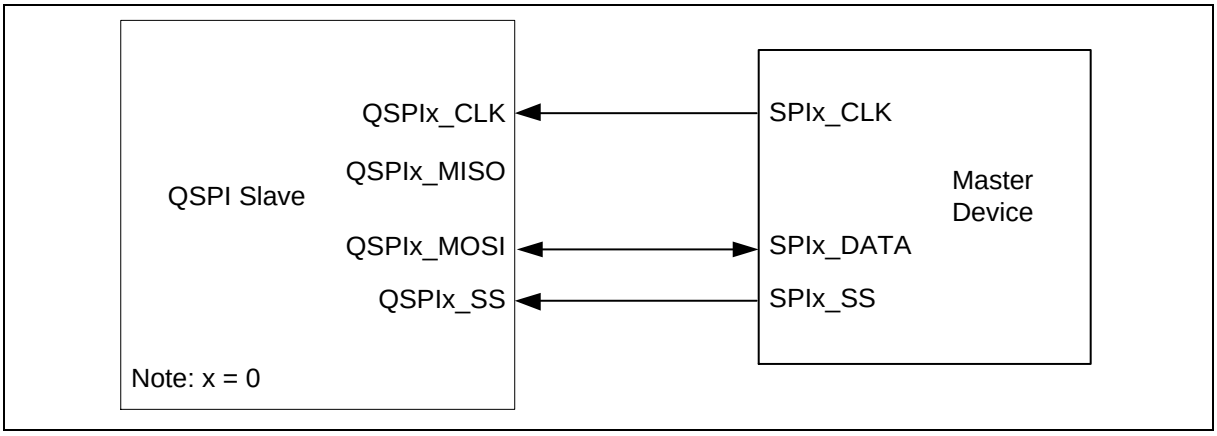


图 6.15-10 QSPI 半双工主机模式应用框图



涂 6.15-11 QSPI 半双工从机模式应用框图

6.15.5.5 *只接收模式*

对于QSPI主机，可配置RXONLY(QSPIx_CTL[15])让其工作在只接收模式，QSPI主机一直不断地产生QSPI总线时钟。该模式下，如果使能了AUTOSS(QSPIx_SSCTL[3])，QSPI主机将一直保持从机片选信号有效。

在只接收模式下，没有用到的QSPIx_MOSI管脚可以配置成GPIO供其他功能使用。由于QSPI总线时钟一直存在，BUSY(QSPIx_STATUS[0])将会被置位。该模式下会同时产生TXFBCLR(QSPIx_FIFOCTL[9])和RXFBCLR(QSPIx_FIFOCTL[8])。使能该模式后，QSPI总线时钟会在6个外设时钟周期之后发送出去。该模式下，写入发送FIFO的数据也会被载入发送移位寄存器发送出去。

当设置RXONLY(QSPIx_CTL[15])使能，QSPIRX接收到DWIDTH(QSPIx_CTL[12:8])设置的数据位数据后将存放到RX FIFO，同时QSPI时钟会一直发送给SPI从机直到RX FIFO满。

2位传输模式TWOBIT(QSPIx_CTL[16])使能，QSPI主机发送时钟给从机，并接收RX数据直到RX FIFO满。在从RX FIFO中读出RX数据后，如果RX FIFO计数器RXCNT小于或等于4，QSPI主机将继续发送时钟给SPI从机并接收RX数据。

6.15.5.6 *从机三线模式*

SLV3WIRE(QSPIx_SSCTL[4])置1就使能了从机3线模式，QSPI控制器在无从机选择信号时正常工作。SLV3WIRE(QSPIx_SSCTL[4])仅在从机模式有效。仅需三个管脚：QSPIx_CLK,QSPIx_MISO,和QSPIx_MOSI，QSPI控制器在无从机选择信号时正常工作，QSPIx_SS脚可配置成GPIO。在与QSPI主机通讯时，当SLV3WIRE(QSPIx_SSCTL[4])设为1时，在SPIEN(QSPIx_CTL[0])置1后就开始准备发送和接收数据。

6.15.5.7 *PDMA 传输功能*

QSPI控制器支持PDMA传输功能。

当TXPDMAEN(QSPIx_PDMACTL[0])置1时，控制器会请求PDMA控制发送过程。

当RXPDMAEN(QSPIx_PDMACTL[1])置1时，将启动PDMA接收过程。当接收FIFO有数据时，控制器会请求PDMA接收数据。

注:QSPI只支持单一PDMA请求(读/写),不支持PDMA批量请求。

6.15.5.8 *2位传输模式*

TWOBIT(QSPIx_CTL[16])为1时，使能2位传输模式，QSPI进行全双工数据传输，可同时收发两位数据。

主机模式下，存储在QSPIx TX FIFO的偶数位数据(TX Data(n))通过QSPIx_MOSI0管脚发送，存储在QSPIx TX FIFO的奇数位(TX Data(n+1))通过QSPIx_MOSI1管脚发送。同时，从QSPIx_MOSI0管脚接收到的偶数位数据将优先于从QSPIx_MOSI1管脚接收到的奇数位数据写入接收FIFO。

从机模式下，存储在QSPIx TX FIFO的偶数位和奇数位将分别从QSPIx_MISO0管脚和QSPIx_MISO1管脚发送。同时，从QSPIx_MISO0管脚接收到的偶数位和从QSPIx_MISO1管脚接收到的奇数位将分别写入QSPI_RX寄存器。FIFO缓存的数据顺序与主机模式是一样的。

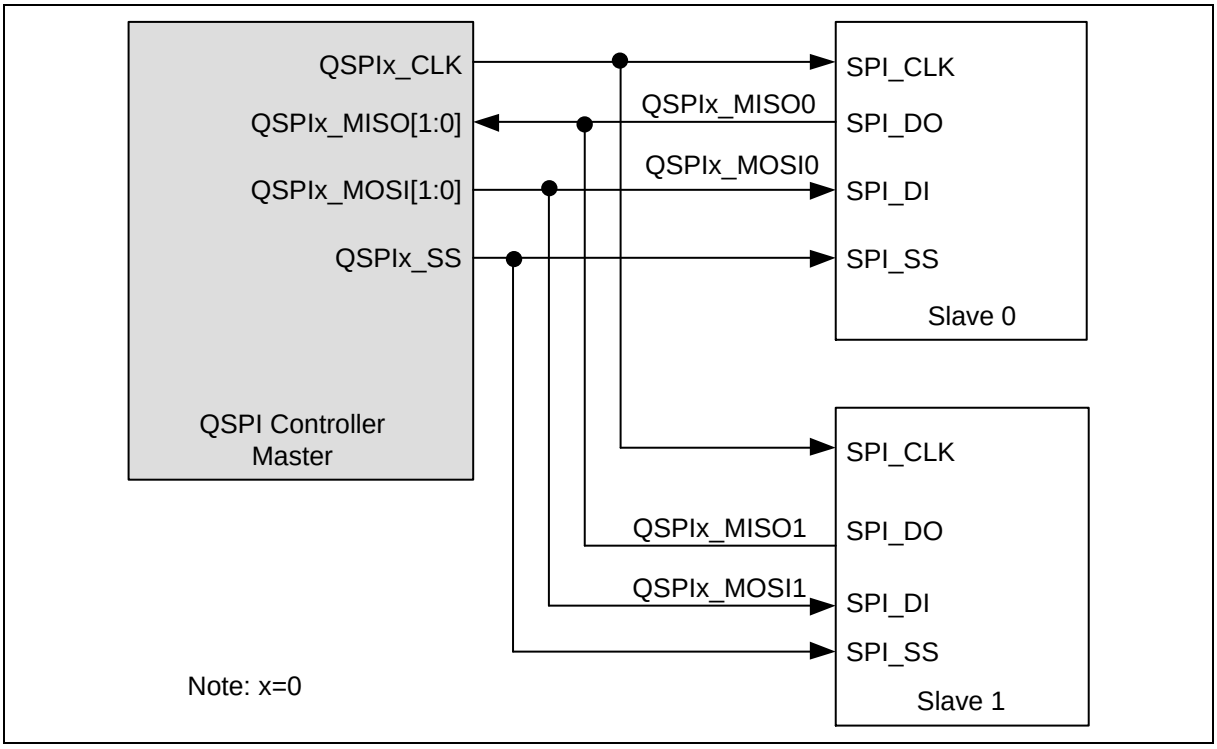


图 6.15-12 2 位传输模式系统架构

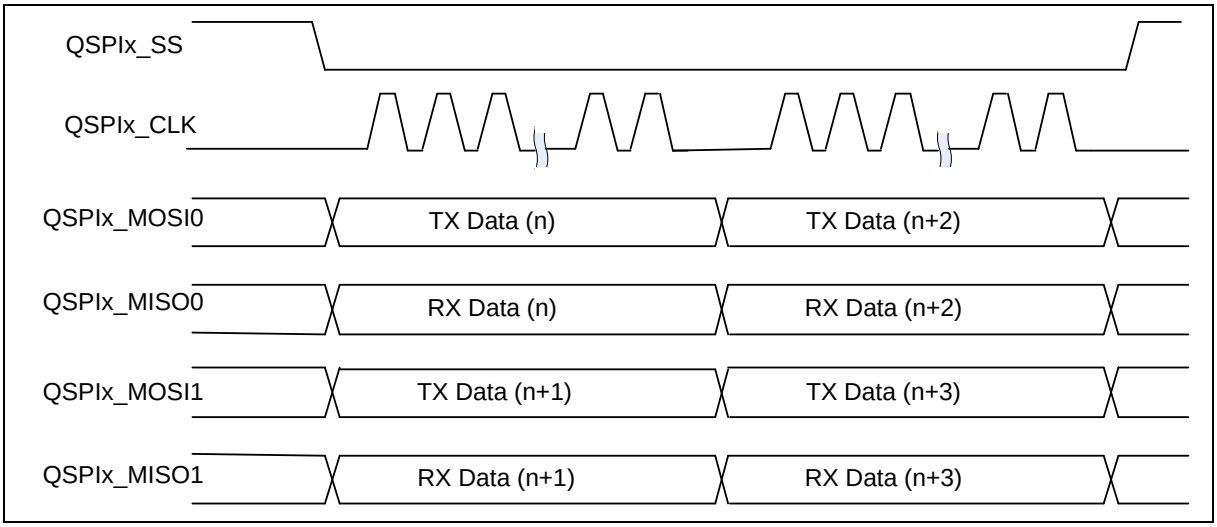


图 6.15-13 2 位传输模式的时序图(主机模式)

6.15.5.9 双I/O模式

DUALIOEN((QSPIx_CTL[21])置1，QSPI配置为双I/O传输。许多通用SPI Flash支持双I/O传输模式。DATDIR(QSPIx_CTL[20])用来定义传输数据的方向。DATDIR=1时，为双路发送，反之为双路接收。该功能支持8,16,24,和32位长度。

从机三线模式或字节重排功能时，不支持双I/O模式。

针对双I/O模式，如果DUALIOEN(QSPIx_CTL[21])和DATDIR(QSPIx_CTL[20])都设置成1，则QSPIx_MISO0输出偶数位，QSPIx_MISO0输出奇数位。如果DUALIOEN(QSPIx_CTL[21])置1，DATDIR(QSPIx_CTL[20])置0，QSPIx_MISO0和QSPIx_MOSI0都为输入端口。

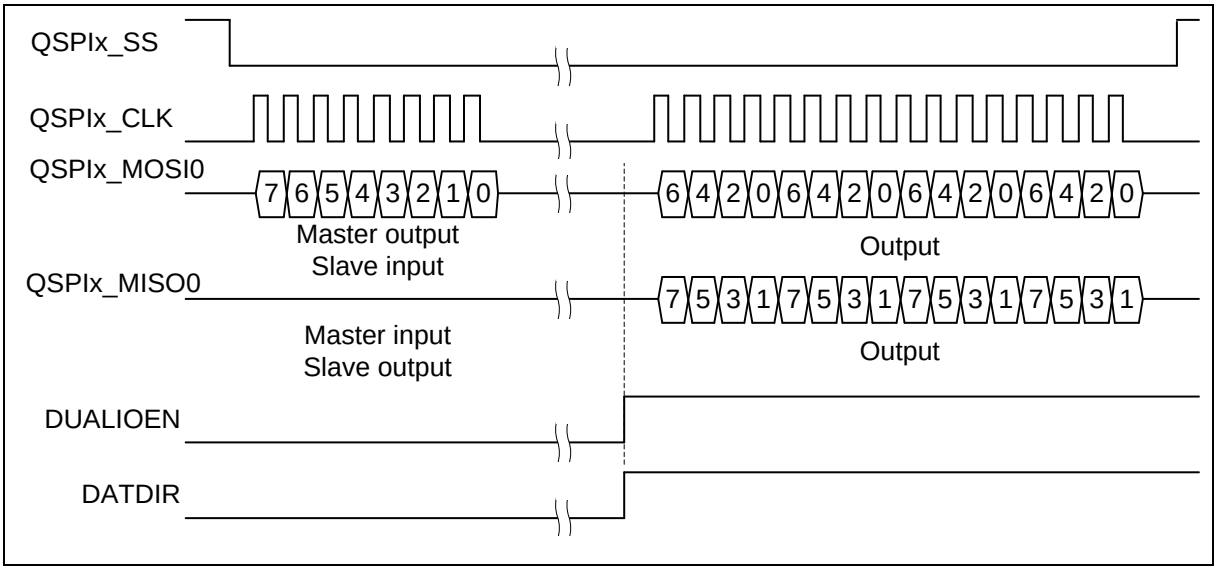


图 6.15-14 双输出模式的位时序

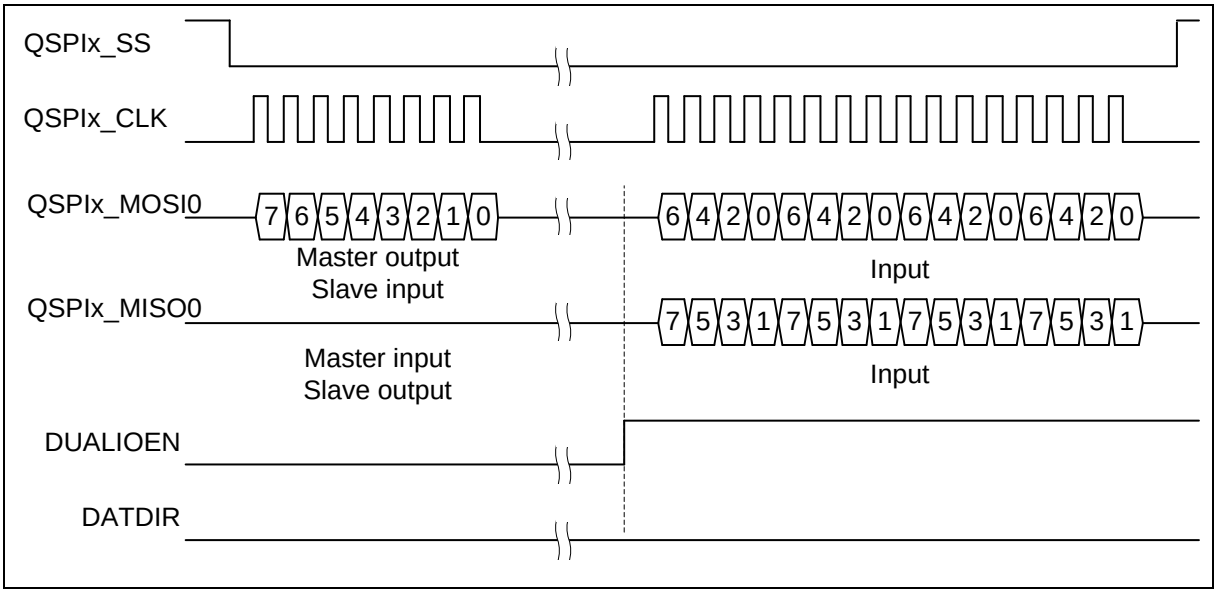


图 6.15-15 双输入模式的位时序

6.15.5.10 四I/O模式

当QUADIOEN (QSPIx_CTL[22]) 设置为1时，QSPI控制器也支持4 I/O模式。许多通用的SPI Flash支持4 I/O传输。DATDIR 位(QSPIx_CTL[20])用于定义传输方向，当DATDIR(QSPIx_CTL[20])=1，为发送，反之为接收。该功能支持8,16,24,和32位长度。

从机三线模式或字节重排功能时，不支持4 I/O模式。DUALIOEN(QSPIx_CTL[21]) 和QUADIOEN(QSPIx_CTL[22])不能同时设置为1。

对于四 I/O模式，如果QUADIOEN(QSPIx_CTL[22]) 和DATDIR(QSPIx_CTL[20]) 都设成1，则QSPIx_MOSI0和QSPIx_MOSI1输出偶数位，QSPIx_MOSI0和QSPIx_MOSI1输出奇数位。如果QUADIOEN(QSPIx_CTL[22]) 置 1，DATDIR(QSPIx_CTL[20]) 置 0，QSPIx_MISO0,QSPIx_MISO1,QSPIx_MOSI0和QSPIx_MOSI1都是输入。

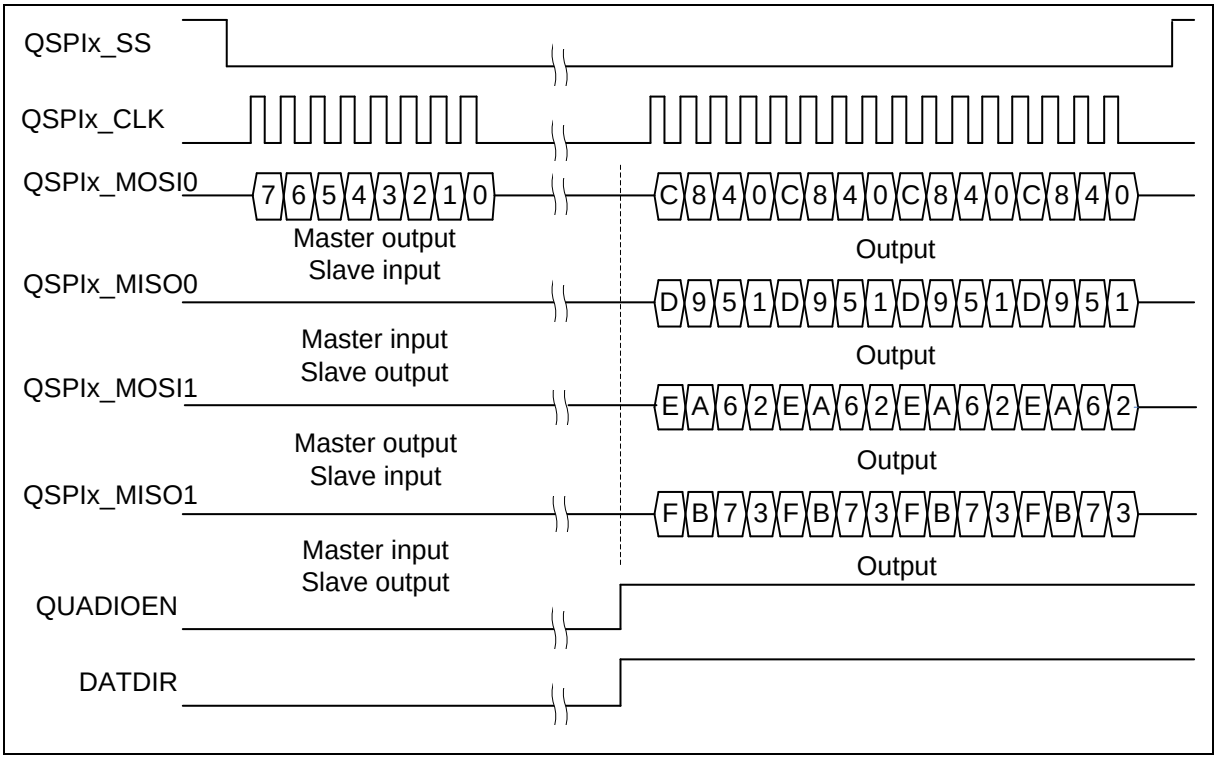


图 6.15-16 四 I/O 输出模式位顺序

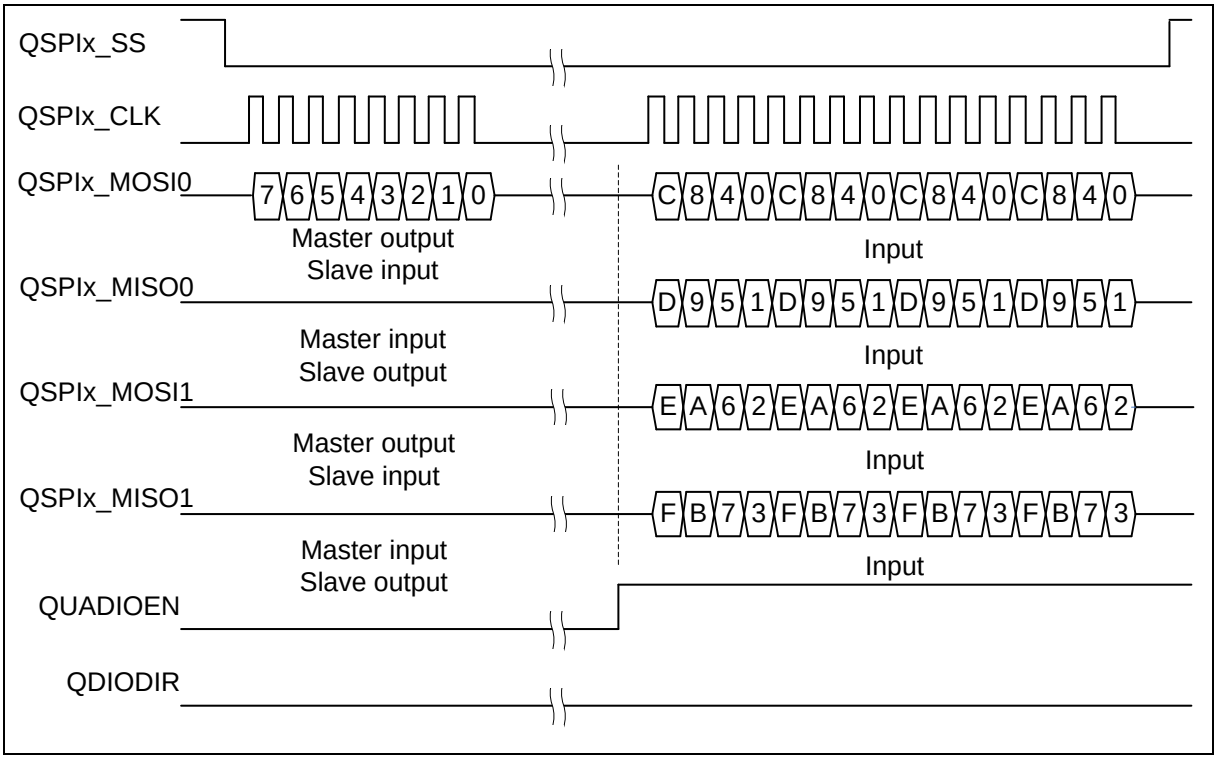


图 6.15-17 四 I/O 输入模式位顺序

6.15.5.11 FIFO缓存操作

QSPI控制器配备了32位宽的发送和接收FIFO缓存。存放在发送FIFO缓存的数据会通过发送控制逻辑进行读取和发送。如果发送FIFO缓存满了，TXFULL(QSPIx_STATUS[17])会被置1。当QSPI传输逻辑单元抽出发送FIFO缓存的最后一个数据，发送FIFO缓存就为空了，TXEMPTY(QSPIx_STATUS[16])位被置1。注意TXEMPTY(QSPIx_STATUS[16])标志在最后一笔数据传输还在进行时就被置1了。在主机模式，当FIFO缓存写入数据或者QSPI总线上有任何事务，BUSY(QSPIx_STATUS[0])位被设置为1。(如：从机片选信号激活和QSPI控制器在从机模式正在接收数据)。当传送缓存为空且当前事务已经完成后，该位将设置为0。因此，软件可以检查BUSY(QSPIx_STATUS[0])位的状态以确认QSPI是否已经空闲。

接收控制逻辑存储QSPI接收到的数据存放在接收FIFO缓存。有FIFO相关的状态位，像RXEMPTY(QSPIx_STATUS[8])和RXFULL(QSPIx_STATUS[9])，来表明当前FIFO缓存的状态。

发送和接收的阀值可以通过设置TXTH(QSPIx_FIFOCTL[30:28])和RXTH(QSPIx_FIFOCTL[26:24])来设定。当存储在发送FIFO缓存的有效数据数小于或等于TXTH(QSPIx_FIFOCTL[30:28])的设定值时，TXTHIF(QSPIx_STATUS[18]) 位会被置1。当存储在接收FIFO缓存的有效数据数大于RXTH(QSPIx_FIFOCTL[26:24])的设定值时，RXTHIF(QSPIx_STATUS[10])位会被置1。

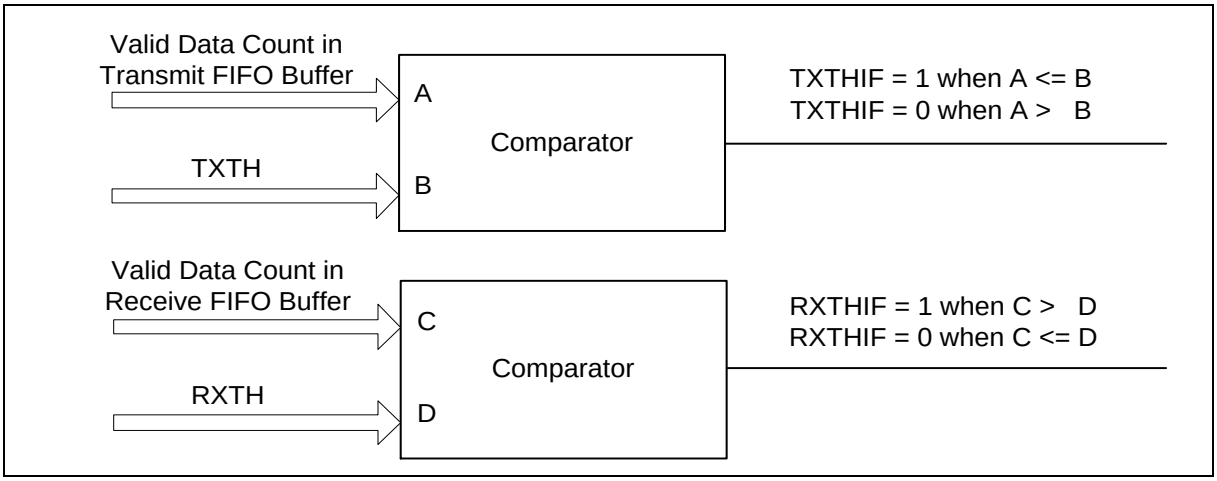


图 6.15-18 FIFO 阈值比较器

在主机模式，当第一个数据写入QSPIx_TX寄存器时，TXEMPTY(QSPIx_STATUS[16])标志将会被清除为0。在1个APB时钟周期和6个外设时钟周期后，发送将开始。用户可以立即写下一个数据到QSPIx_TX寄存器。QSPI控制器将会在两个连续的事务之间插入一个休眠间隔，休眠间隔的长度由SUSPITV(QSPIx_CTL[7:4])的设定值决定。如果SUSPITV(QSPIx_CTL[7:4])等于0，QSPI控制器可以执行连续发送。只要TXFULL(QSPIx_STATUS[17])为0，用户就可以向QSPIx_TX寄存器写入新的数据。

如图 6.14-13的例1所示，该图指明了TXEMPTY(QSPIx_STATUS[16])的更新条件以及FIFO缓存、移位寄存器和斜移缓存之间的关系。当Data0写入FIFO缓存时，TXEMPTY(QSPIx_STATUS[16])位被设置为0。Data0将由内核逻辑加载到移位寄存器，此时，TXEMPTY(QSPIx_STATUS[16])将设置为1。移位寄存器中Data0将逐位移入斜移缓存传输直到传输完成。

在例2中，该图为当FIFO缓存中有8个数据时，更新TXFULL(QSPIx_STATUS[17])的条件。当TXFULL=1时，下一数据Data9不能被写入FIFO缓存。

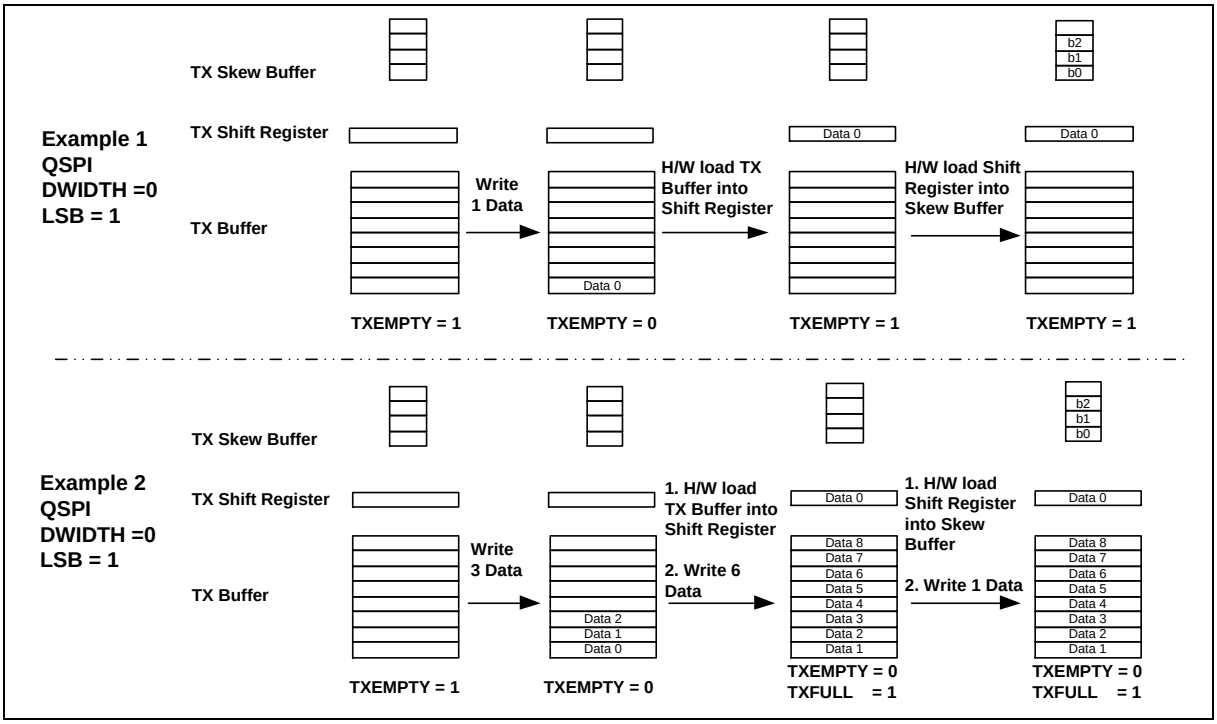


图 6.15-19 发送 FIFO 缓存示例

如果要发送的数据更新及时，接下来的事务将会被自动触发。如果在所有数据传输完成之后，QSPIx_TX寄存器没有被更新，则传输停止。

在主机模式接收操作中，串行数据从QSPIx_MISO管脚接收并被存储在接收FIFO缓存。

接收数据(Data0's b0,b1,…b31)通过串行时钟(QSPIx_CLK)先存到斜移缓存，然后再逐位移到移位寄存器。当接收数据位达到DWIDTH(QSPIx_CTL[12:8])的值时，内部逻辑将移位寄存器中的数据加载到FIFO缓存。当接收FIFO缓存有未读数据时(参见图 6.15-20示例1)，RXEMPTY(QSPIx_STATUS[8])将清0。只要RXEMPTY(QSPIx_STATUS[8])为0，用户就可以通过QSPIx_RX寄存器来读取接收的数据。如果接收FIFO缓存包含8个未读数据，RXFULL(QSPIx_STATUS[9])将设置为1 (详见图 6.15-20 的示例2).

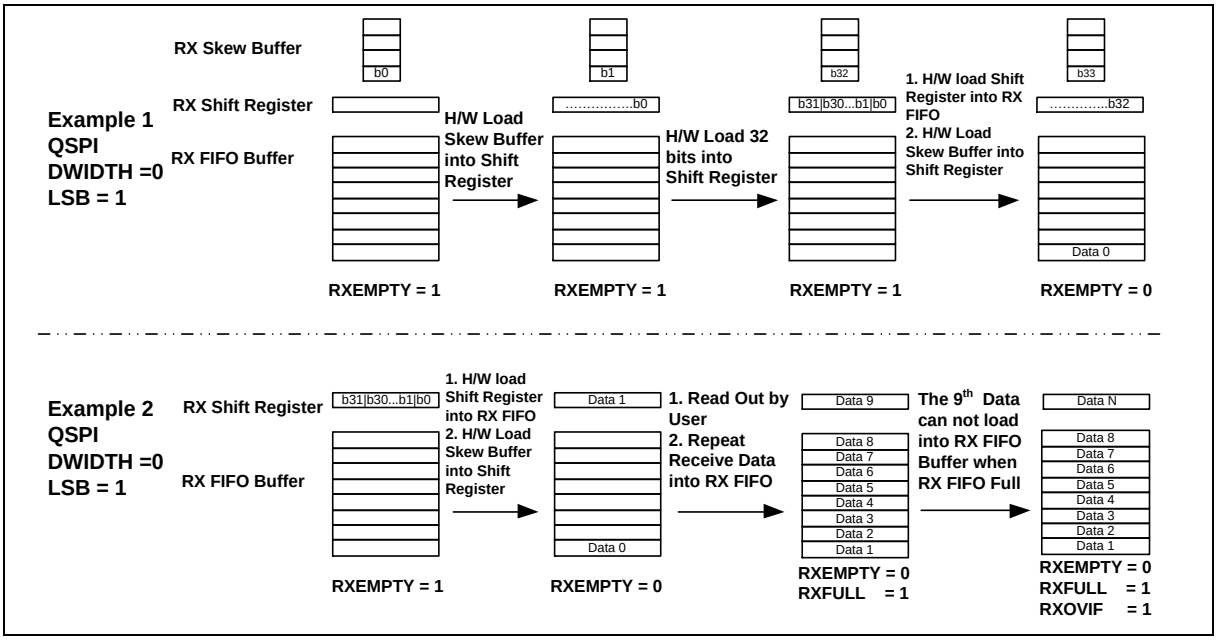


图 6.15-20 接收 FIFO 缓存示例

在从机模式中，当软件写数据到QSPIx_TX寄存器时，数据将会被加载到发送FIFO缓存，同时TXEMPTY(QSPIx_STATUS[16])标志也将被设置为0。当从设备从主机接收到时钟信号时，发送操作将会开始。只要TXFULL(QSPIx_STATUS[17])标志为0，用户就可以写数据到QSPIx_TX寄存器。所有数据都被QSPI发送逻辑单元发送出去后，而且软件没有再更新QSPIx_TX寄存器TXEMPTY(QSPIx_STATUS[16])标志将会被设置为1。

当从机片选信号有效时，如果没有任何数据写入QSPIx_TX寄存器，发送下溢标志TXUFIF(QSPIx_STATUS[19])将被设置为1。输出的数据在本次传输中将会通过设置TXUFPOL(QSPIx_FIFOCTL[6])保留，直到从机片选信号为无效状态。当传送下溢事件发生时，从机溢出运行标志SLVURIF(QSPIx_STATUS[7])将置为1。同时QSPIx_SS进入无效状态。

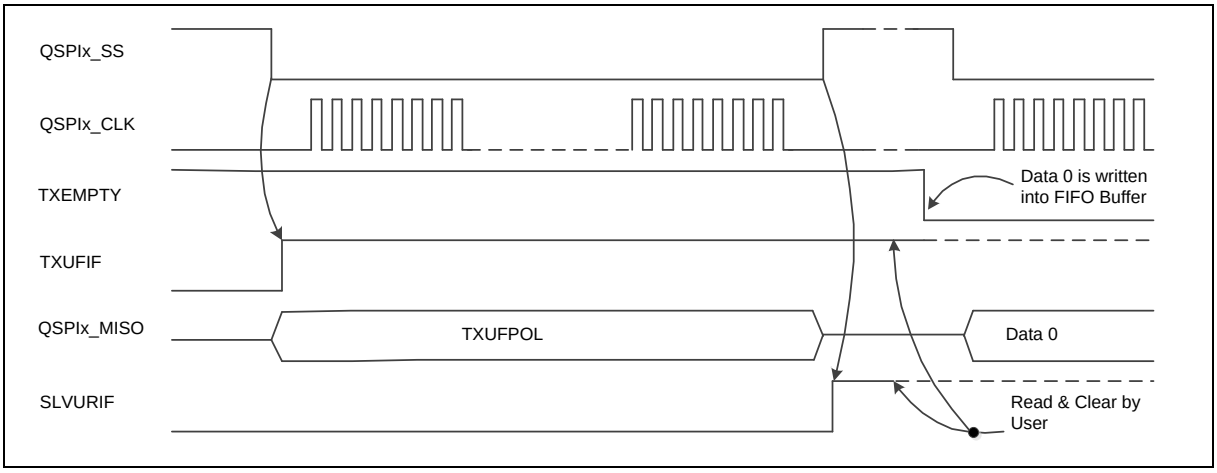


图 6.15-21 TX 下溢事件和从机溢出事件

在2位传输模式，在2个数据已经写入TX FIFO后，传送数据将被加载到移位寄存器。该模式同时使用两个移位寄存器和两个4级的斜移缓存。2位传送模式的详细时序，请参考2位传输模式的相关介绍.

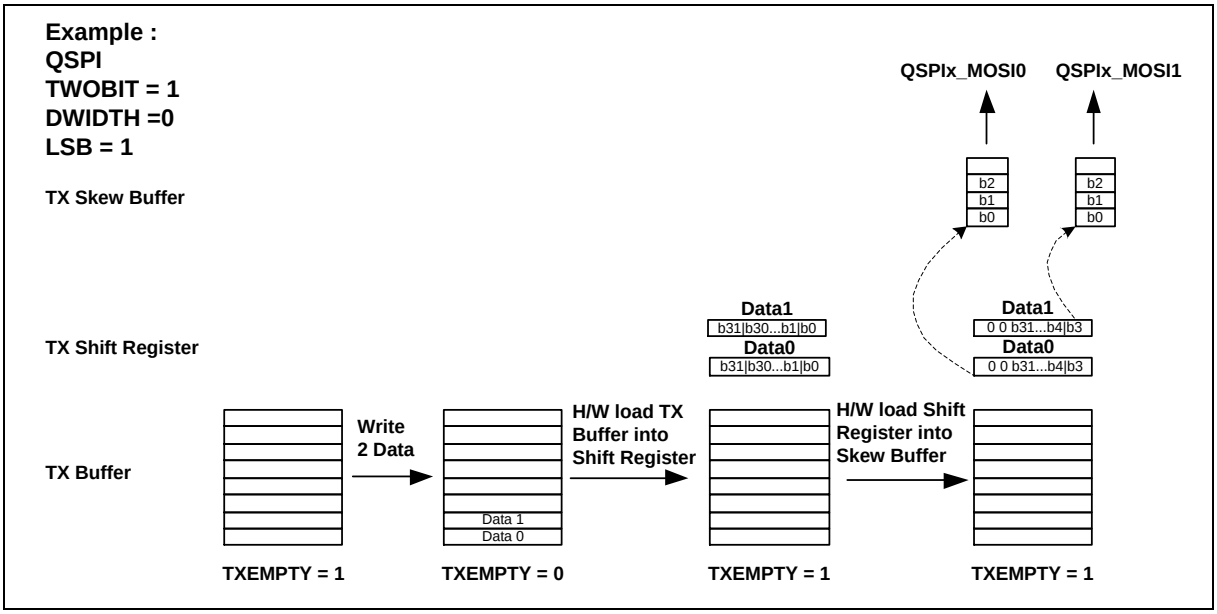


图 6.15-22 2 位传输模式 FIFO 缓存示例

在从机三线模式，如果在QSPI总线时钟出现前的3个外设时钟周期内，把数据写入发送FIFO缓存，则开始的2位数据是不可预知的(会保持上次传输最后一位的电平)。其它位则被TXUFPOL(QSPIx_FIFOCTL[6])保存。写入的数据将在下一次传输时发送。

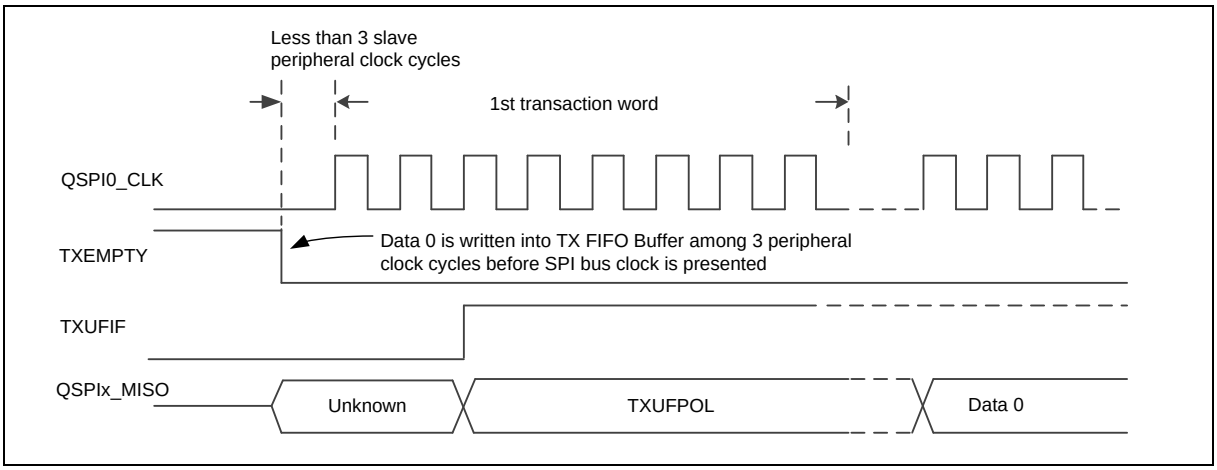


图 6.15-23 TX 下溢事件(QSPI0 从机三线模式使能)

在从机模式接收操作中，串行数据从QSPIx_MOSI管脚接收并存储到QSPIx_RX寄存器。接收机制类似于主机模式的接收操作。如果接收FIFO缓存包含8个未读数据，RXFULL(QSPIx_STATUS[9])将被置为1，RXOVIF(QSPIx_STATUS[11])也会被置为1。如果在QSPIx_MOSI管脚上有更多串行数据要接收，接下来的数据将被丢掉(参看接收FIFO缓存的示例图)。当从机片选线进入无效状态时，如果接收到位数数据与DWIDTH(QSPIx_CTL[12:8])设定的不一致，SLVBEIF(QSPIx_STATUS[6])将设置为1。

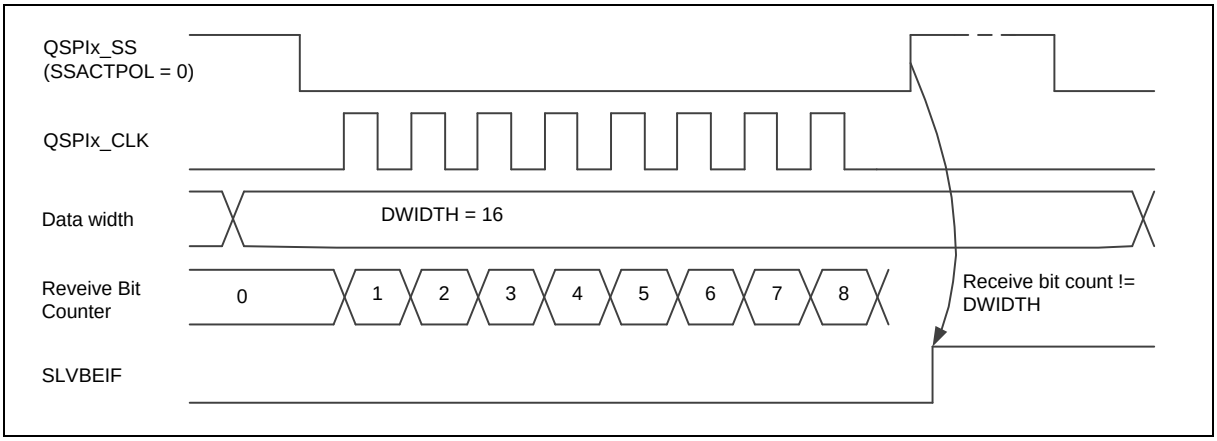


图 6.15-24 从机模式位计数错误

当从机选择信号有效且SLVTOCNT(QSPIx_SSCTL[31:16])的值不为0时，在串行时钟输入后，QSPI控制器逻辑的从机超时计数器开始计数。在一次事务完成后或者SLVTOCNT设置为0后，该计数器清除。如果超时计数器的值在该事务完成前大于或等于SLVTOCNT的值，从机超时事件发生，SLVTOIF(QSPIx_STATUS[5])将设置为1。

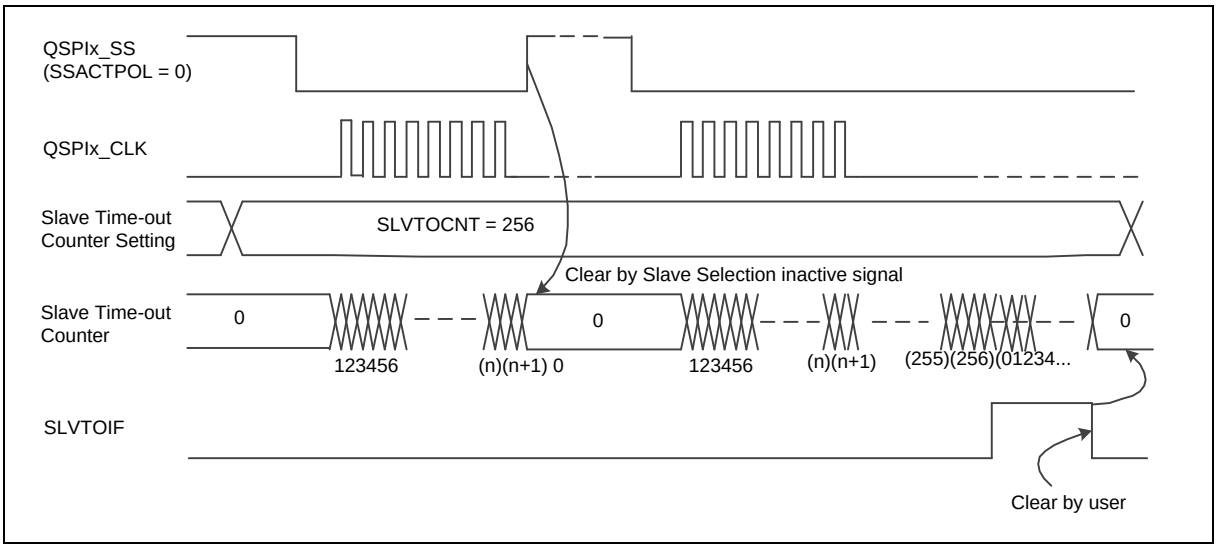


图 6.15-25 从机超时事件

控制器内置了一个接收超时功能模块。当接收FIFO非空，而且在主机模式时接收FIFO超过64个QSPI外设时钟周期或者从机模式时超过576个QSPI外设时钟周期没有读操作，则接收超时发生，RXTOIF(QSPIx_STATUS[12])置1。当接收FIFO有用户读取时，超时状态将自动清除。

6.15.5.12 中断

- QSPI 单元传输中断

  当QSPI控制器完成一个单元传输，单元传输中断标志UNITIF(QSPIx_STATUS[1])将会被设为1。如果单元传输中断使能位UNITIEN(QSPIx_CTL[17])被置位，则单元传输中断事件将给CPU产生中断请求。单位传输中断标志位只能写1清零。

- QSPI 从机片选有效/无效中断

  从机模式下，如果SPIEN(QSPIx_CTL[0])和SLAVE(QSPIx_CTL[18])置1，从机片选信号进

入有效/无效状态，从机片选有效/无效中断标志SSACTIF(QSPIx_STATUS[2])和SSINAIF(QSPIx_STATUS[3])会被置位。此时如果SSINAIEN(QSPIx_SSCTL[13])或SSACTIEN(QSPIx_SSCTL[12])为1，QSPI控制器会产生中断。

- 从机超时中断

  QSPI在从机模式时，用户可以通过从机超时功能识别是否有串行时钟输入但超过了SLVTOCNT(QSPIx_SSCTL[31:16])中定义的从机外设时钟周期个数、传输事务还没有完成的情况。

  当从机片选信号有效时，而且SLVTOCNT(QSPIx_SSCTL[31:16])的值不为0，在串行时钟输入后，QSPI控制器逻辑的从机超时计数器开始计数。在一次事务完成后或者SLVTOCNT(QSPIx_SSCTL[31:16])被置为0，该计数器将被清除。如果在一次事务传输完前，超时计数器的值大于或等于SLVTOCNT(QSPIx_SSCTL[31:16])的值，则从机超时事件发生，同时SLVTOIF(QSPIx_STATUS[5])将设置为1。如果SLVTOIEN(QSPIx_SSCTL[5])为1，SPI控制器将发生一个中断。

- 从机位计数错误中断

  在从机模式，当从机片选信号线进入无效状态时，如果发送或接收到位数据个数与DWIDTH(QSPIx_CTL[12:8])设置的不一致，SLVBEIF(QSPIx_STATUS[6])将被设置为1。未传输完成的数据将在TX和RX移位寄存器中弃除。如果SLVBEIEN(QSPIx_SSCTL[8])位为1，QSPI控制器将发生一个中断。

  注:如果从机片选信号激活，但是没有任何串行时钟输入，当从机片选信号进入无效状态时，SLVBEIF(QSPIx_STATUS[6])也将设置为1。

- TX 下溢中断

  在QSPI从机模式，如果没有任何数据写入QSPIx_TX寄存器，当从机选择信号激活时，TXUFIF(QSPIx_STATUS[19])将设置为1。如果TXUFIEN(QSPIx_FIFOCTL[7])为1，QSPI控制器将发生发送下溢中断。

  注:QSPI从机模式发生下溢事件后，有两种方法可以使其恢复空闲状态并进入下个传输：(1)置TXRST为1；(2)当SLV3WIRE=0，从机片选信号切为无效状态。

- 从机TX下溢运行中断

  当QSPIx_SS进入无效状态时，如果有TX下溢事件发生，SLVURIF(QSPIx_STATUS[7])将被设置为1。如果SLVURIEN(QSPIx_SSCTL[9])为1，QSPI控制器将发生发送溢出运行中断。

  注:在从机三线模式，从机片选信号被认为是一直有效的，用户需要去查看TXUFIF(QSPIx_STATUS[19])位来确定是否有发生TX下溢事件。

- 接收溢出中断

  在从机模式，如果接收FIFO缓存已有8个未读数据，RXFULL(QSPIx_STATUS[9])和RXOVIF(QSPIx_STATUS[11])标志将会被设置为1。如果从QSPI总线上接收到更多串行数据，多余的数据将会丢失。如果RXOVIEN(QSPIx_FIFOCTL[5])为1，QSPI控制器将发生接收溢出中断。

- 接收FIFO超时中断

  如果在FIFO里有一个接收到的数据，在主机模式下用户超过64个QSPI外设时钟周期没有去读取，或者从机模式下超过576个QSPI外设时钟周期没有去读取，如果接收超时中断使能位RXTOIEN(QSPIx_FIFOCTL[4])有设置为1，则会向系统发出一个RX超时中断。

- 发送FIFO中断

在FIFO模式，如果发送FIFO缓存的有效数据少于或等于TXTH(QSPIx_FIFOCTL[30:28])的设定值，发送FIFO中断标志TXTHIF(QSPIx_STATUS[18])会被置1。如果发送FIFO中断位TXTHIEN(QSPIx_FIFOCTL[3])为1，则QSPI控制器会向系统产生一个发送FIFO中断。

- 接收FIFO中断

  在FIFO模式，如果接收FIFO缓存的有效数据大于RXTH(QSPIx_FIFOCTL[26:24])的设定值，接收FIFO中断标志RXTHIF(QSPIx_STATUS[10])会被设置为1。如果接收FIFO中断位RXTHIEN(QSPIx_FIFOCTL[2])有设置为1，QSPI控制器将会向系统产生一个接收FIFO中断。

6.15.6 时序图

从机选择信号的有效状态可以由SSACTPOL(QSPIx_SSCTL[2])配置。串行时钟的空闲状态可以通过CLKPOL(QSPIx_CTL[3])配置为高电平或低电平。传输字段长度在DWIDTH(QSPIx_CTL[12:8])中定义，发送/接收数据是以MSB还是LSB优先由LSB位(QSPIx_CTL[13])定义。用户也可以通过TXNEG/RXNEG(QSPIx_CTL[2:1])来选择发送/接收数据时串行时钟的边沿。四种QSPI主机/从机操作时序图以及相关设置如下图：

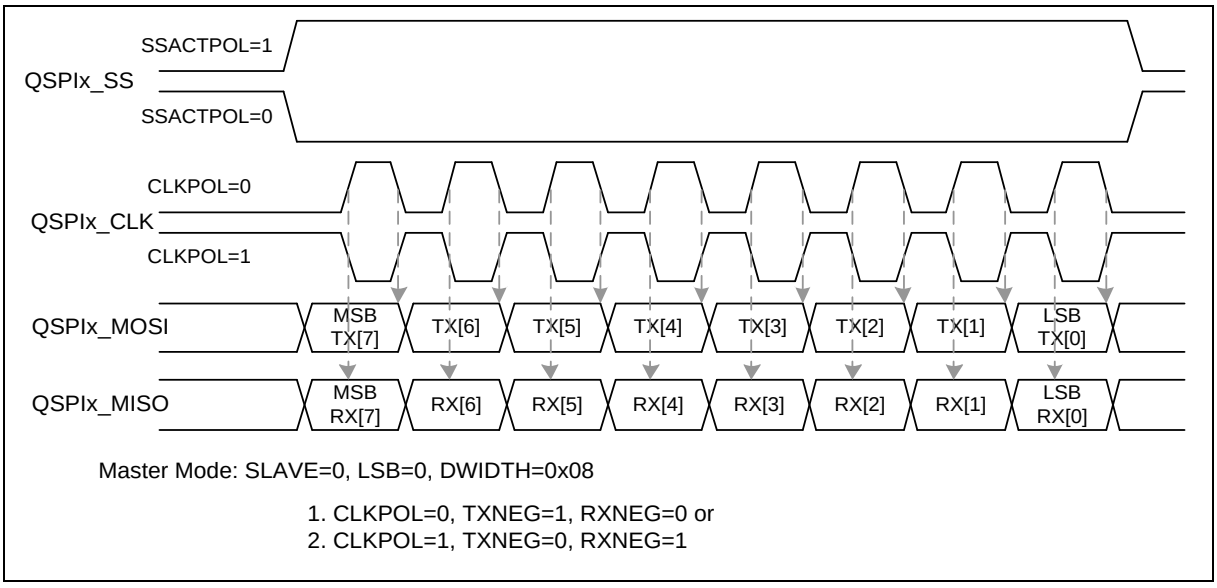


图 6.15-26 QSPI 主机模式时序

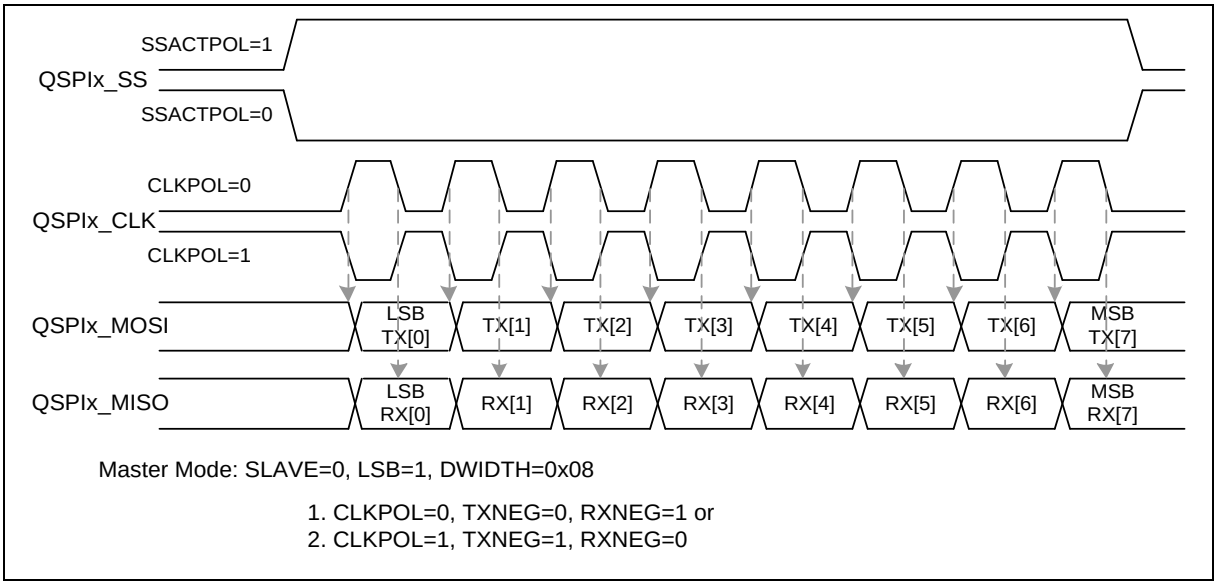


图 6.15-27 QSPI 主机模式时序(QSPIx_CLK 交替相位)

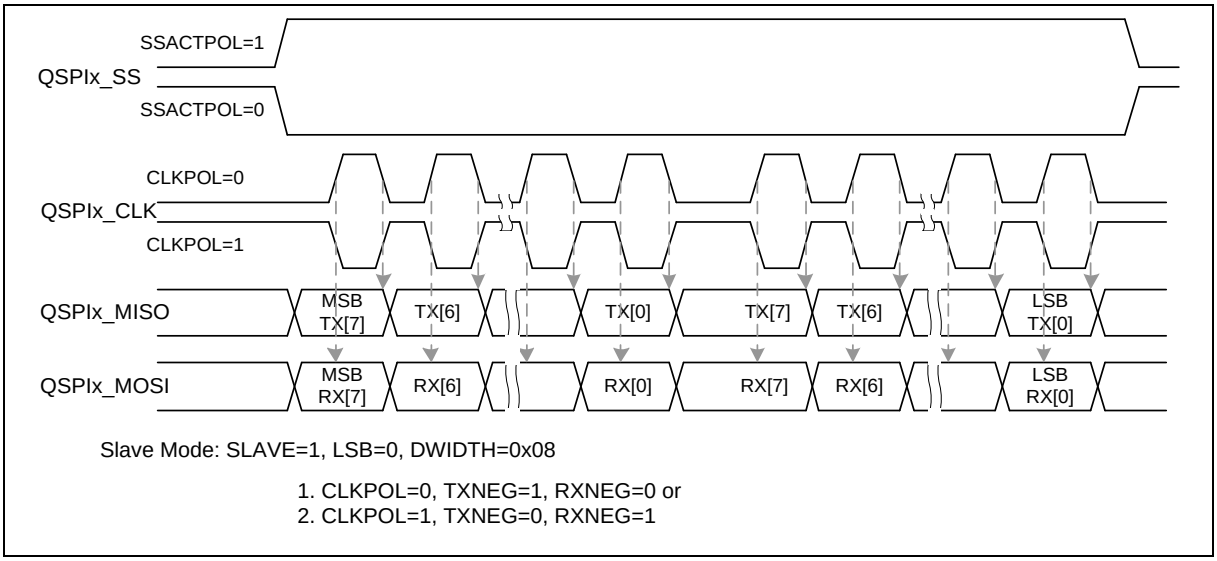


图 6.15-28 QSPI 从机模式时序

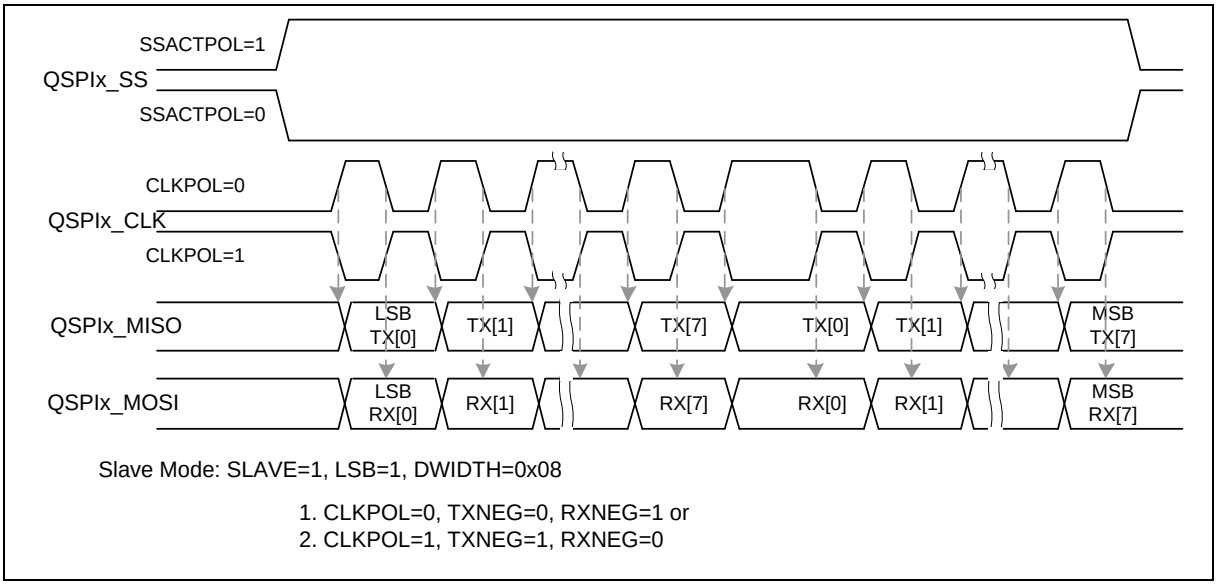


图 6.15-29 QSPI 从机模式时序(QSPIx_CLK 交替相位)

6.15.7 编程示例

例 1:

QSPI配置为全双工主机时序如下：

- QSPI时钟上沿锁存数据
- QSPI时钟下沿发送数据
- MSB优先
- QSPI时钟空闲时处于低电平
- 每次只收发一个字节
- 片选信号低电平有效。

操作流程如下：

1. 在寄存器DIVIDER(QSPIx_CLKDIV[8:0])配置QSPI输出时钟频率。
2. 写寄存器QSPIx_SSCTL配置QSPI主机相关模式：
   1) 清AUTOSS(QSPIx_SSCTL[3]) 为0关闭自动从机选择功能。
   2) 清SSACTPOL(QSPIx_SSCTL[2]) 为0设置片选信号低电平有效。
   3) 置SS(QSPIx_SSCTL[0])为1使能从机片选信号，激活片外从机设备。
3. 写寄存器QSPIx_CTL控制QSPI主机行为：
   1) 设SLAVE(QSPIx_CTL[18])为0配置QSPI为主机设备。
   2) 清CLKPOL(QSPIx_CTL[3])为0强制QSPI空闲状态为低电平。
   3) 置TXNEG(QSPIx_CTL[2])为1配置QSPI总线时钟下沿传输数据。
   4) 清RXNEG(QSPIx_CTL[1])为 0配置QSPI总线时钟上沿锁存数据。
   5) 设置传输长度DWIDTH为8位(QSPIx_CTL[12:8]=0x08)。

6) 清LSB(QSPIx_CTL[13]) 为0配置MSB传输优先。

4. 置SPIEN(QSPIx_CTL[0])位1使能数据传输。
5. QSPI主机想要写一个字节的数据到片外设备时，将数据写到QSPIx_TX寄存器即可。
6. 等待QSPI中断发生(如果UNITIEN(QSPIx_CTL[17])为1)，或者轮询单元传输中断标志UNITIF(QSPIx_STATUS[1])。
7. 从QSPIx_RX读接收到的数据。
8. 重复步骤5继续其他数据的传输，或者配置SS(QSPIx_SSCTL[0])为0让片选信号无效以关闭片外从机设备。

例 2:

QSPI为全双工从机。片外主机按下面的规则通讯：

- QSPI时钟上沿锁存数据
- QSPI时钟下沿发送数据
- LSB优先
- QSPI时钟空闲时处于高电平
- 每次只收发一个字节
- 片选信号高电平有效。

操作流程如下：

1. 写寄存器QSPIx_SSCTL配置为从机模式。
2. 置SSACTPOL(QSPIx_SSCTL[2])为1配置片选信号高电平有效。
3. 写QSPIx_CTL配置从机行为
   1) 设SLAVE(QSPIx_CTL[18])为1配置QSPI控制球为从机设备。
   2) 置CLKPOL(QSPIx_CTL[3])为1选择QSPI空闲状态为高电平。
   3) 置TXNEG(QSPIx_CTL[2])为1配置QSPI总线时钟下降沿传输数据.
   4) 清RXNEG(QSPIx_CTL[1]) 为0配置QSPI总线时钟上升沿锁存数据.
   5) 设置传输长度DWIDTH为8位(QSPIx_CTL[12:8]=0x08)。
4. 置LSB(QSPIx_CTL[13])为1配置LSB传输优先。
5. 置SPIEN(QSPIx_CTL[0])为1。等待片外主机设备的片选信号和QSPI时钟输入来启动数据传输。
6. 如果QSPI从机想要发送一个字节的数据到片外主机，写数据到QSPIx_TX寄存器即可。
7. 如果QSPI从机想要从片外主机接收一个字的数据，且用户不关心什么数据被传输，则软件不需要去更新QSPIx_TX寄存器。
8. 等待QSPI中断发生(如果UNITIEN(QSPIx_CTL[17])为1)，或者轮询单元传输中断标志UNITIF(QSPIx_STATUS[1])。
9. 从QSPIx_RX寄存器读接收到的数据。
10. 回到7继续下个字节发送或停止发送。

6.15.8 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读写

寄存器	偏移	R/W	描述	复位值
QSPI 基地址: **QSPIx_BA = 0x4006_0000**				
QSPIx_CTL	QSPIx_BA+0x00	R/W	QSPI控制寄存器	0x0000_0034
QSPIx_CLKDIV	QSPIx_BA+0x04	R/W	QSPI时钟分频寄存器	0x0000_0000
QSPIx_SSCTL	QSPIx_BA+0x08	R/W	QSPI从机选择控制寄存器	0x0000_0000
QSPIx_PDMACTL	QSPIx_BA+0x0C	R/W	QSPI PDMA控制寄存器	0x0000_0000
QSPIx_FIFOCTL	QSPIx_BA+0x10	R/W	QSPI FIFO控制寄存器	0x4400_0000
QSPIx_STATUS	QSPIx_BA+0x14	R/W	QSPI状态寄存器	0x0005_0110
QSPIx_TX	QSPIx_BA+0x20	W	QSPI数据发送寄存器	0x0000_0000
QSPIx_RX	QSPIx_BA+0x30	R	QSPI 数据接收寄存器	0x0000_0000

6.15.9 寄存器描述

QSPI 控制寄存器 (QSPIx_CTL)

寄存器	偏移	R/W	描述	复位值
QSPIx_CTL	QSPIx_BA+0x00	R/W	QSPI 控制寄存器	0x0000_0034

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**QUADIOEN**	**DUALIOEN**	**DATDIR**	**REORDER**	**SLAVE**	**UNITIEN**	**TWOBIT**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXONLY	**HALFDPX**	**LSB**	**DWIDTH**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPITV				**CLKPOL**	**TXNEG**	**RXNEG**	**SPIEN**

位	描述	
[31:23]	**Reserved**	保留。
[22]	**QUADIOEN**	**四I/O模式使能位** 0=四I/O模式关闭 1=四I/O模式使能
[21]	**DUALIOEN**	**双I/O模式使能位** 0=双I/O模式关闭 1=双I/O模式使能
[20]	**DATDIR**	**数据方向控制** 该位用来选则半双工模式和双/四I/O模式下数据输入/输出方向 0=QSPI数据输入 1=QSPI数据输出
[19]	**REORDER**	**字节重排功能使能位** 0=字节重排功能禁止 1=字节重排功能使能.每两个字节之间会被插入一个休眠间隔，休眠间隔的时间取决于SUSPITV的设置 **注:**仅在DWIDTH为16，24和32位时，该功能有效。
[18]	**SLAVE**	**从机模式控制** 0=主机模式. 1=从机模式
[17]	**UNITIEN**	**单元传输中断使能位** 0=QSPI单元传输中断禁止 1=QSPI单元传输中断使能

[16]	**TWOBIT**	**2位传输模式使能位** 0=2位传输模式关闭 1=2位传输模式使能 **注:**当2-位传输模式使能时，第一个串行传输位数据来自第一个FIFO缓存数据，第二个串行传输位数据来自第二个FIFO缓存数据。同样，第一个接收位数据存储到第一个FIFO缓存，第二个接收位数据存放到第二个FIFO缓存。
[15]	**RXONLY**	**只接收模式使能位(仅主机)** 该只用于主机模式。在只接收模式，QSPI主机总线时钟会一直保持以接收数据，BUSY标志会一直置位。 0=只接收模式关闭 1=只接收模式使能
[14]	**HALFDPX**	**QSPI半双工传输使能位** 该位用来切换QSPI传输全双工模式和半双工模式。半双工模式下DATDIR(QSPIx_CTL[20])用来选择数据传输方向。 0=QSPI工作在全双工模式 1=QSPI工作在半双工模式
[13]	**LSB**	**LSB优先发送** 0=MSB,具体发送/接收寄存器的哪一位首先被发送/接收，取决于DWIDTH的设定值。 1=LSB,QSPITX寄存器的低位，首先被发送到QSPI数据输出管脚，从QSPI数据输入管脚上接收到的第一个数据位将会被放置到RX寄存器(QSPI_RX的位0)LSB的位置。
[12:8]	**DWIDTH**	**数据宽度** 该位用来标示每次传输时会有多少位被发送/接收。最低为8位，最高为32位。 DWIDTH=0x08….8位 DWIDTH=0x09….9位 …… DWIDTH=0x1F….31位 DWIDTH=0x00….32 位
[7:4]	**SUSPITV**	**休眠间隔 (仅主机)** 该四位用来配置在一次数据传输过程中连续两个发送/接收事务之间的休眠间隔。休眠间隔是从当前事务字的最后一个时钟边沿到接下来的事务的第一个边沿时钟。默认值是0x3.休眠间隔的周期可以根据下面公式获得： (SUSPITV[3:0]+0.5)*QSPICLK时钟周期 例: SUSPITV=0x0….0.5QSPICLK时钟周期 SUSPITV=0x1….1.5QSPICLK时钟周期 …… SUSPITV=0xE….14.5QSPICLK时钟周期 SUSPITV=0xF….15.5QSPICLK时钟周期
[3]	**CLKPOL**	**时钟极性** 0=QSPI总线空闲时默认低 1=QSPI总线空闲时默认高
[2]	**TXNEG**	**下降沿发送** 0=在QSPI总线时钟的上升沿发送数据 1=在QSPI总线时钟的下降沿发送数据

[1]	**RXNEG**	**下降沿接收** 0=在QSPI总线时钟的上升沿锁存数据 1=在QSPI总线时钟的下降沿锁存数据
[0]	**SPIEN**	**QSPI传输控制使能位** 在主机模式下，FIFO缓存有数据时，该位设置为1后，开始传输。在从机模式下，该位置为1时，该设备已准备好接收数据。 0=禁止控制传输 1=使能控制传输 **Note:** **注:**在更改QSPIx_CTL、QSPIx_CLKDIV、QSPIx_SSCTL和QSPIx_FIFOCTL寄存器的配置前，用户需将QSPIEN(SPIx_CTL[0])清零，并且确定QSPIENSTS(QSPI_STATUS[15])的值为0。

QSPI 时钟分频寄存器(QSPIx_CLKDIV)

寄存器	偏移	R/W	描述	复位值
QSPIx_CLKDIV	QSPIx_BA+0x04	R/W	QSPI 时钟分频寄存器	0x0000_0000

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

位	描述	
[31:9]	**Reserved**	保留.
[8:0]	**DIVIDER**	**时钟除频** 该域的值是产生外设时钟，f_{Qspi_eclk}和QSPI主机的总线时钟的除频器的值。频率计算公式如下： $f_{qspi_eclk} = \frac{f_{qspi_clock_src}}{(DIVIDER+1)}$ 其中， $f_{Qspi_clock_src}$ 为外设时钟源，定义于时钟控制寄存器 CLK_CLKSEL2 中 **注:**在是否QSPI IP 软件复位和设置这个时钟除频寄存器之间的间隔时间必须大于或等于8个外设时钟周期。

注:谨慎配置**DIVIDER**，QSPI时钟频率不能大于系统时钟频率.

QSPI 从机选择控制寄存器(QSPIx_SSCTL)

寄存器	偏移	R/W	描述	复位值
QSPIx_SSCTL	QSPIx_BA+0x08	R/W	QSPI 从机选择控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
SLVTOCNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SLVTOCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SSINAIEN**	**SSACTIEN**	**Reserved**		**SLVURIEN**	**SLVBEIEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**SLVTORST**	**SLVTOIEN**	**SLV3WIRE**	**AUTOSS**	**SSACTPOL**	**Reserved**	**SS**

位	描述	
[31:16]	**SLVTOCNT**	**从机模式超时周期** 在从机模式，当在从机片选信号有效期间有总线时钟输入时，这些位表示超时周期。超时计数器的时钟源是从机外设时钟。如果该位值为 0，表示禁止从机模式超时功能。
[15:14]	**Reserved**	保留
[13]	**SSINAIEN**	**从机片选无效中断使能位** 0=从机片选无效中断禁止 1=从机片选无效中断使能
[12]	**SSACTIEN**	**从机片选有效中断使能位** 0=从机片选有效中断禁止 1=从机片选有效中断使能
[11:10]	**Reserved**	保留
[9]	**SLVURIEN**	**从机模式TX下溢中断使能位** 0=从机模式TX下溢中断禁止 1=从机模式TX下溢中断使能
[8]	**SLVBEIEN**	**从机模式位计数错误中断使能位** 0=从机模式位计数错误中断禁止 1=从机模式位计数错误中断使能
[7]	**Reserved**	保留
[6]	**SLVTORST**	**从机模式超时复位控制** 0=从机模式超时事件发生时，TX和RX控制电路不会复位 1=从机模式超时事件发生时，TX和RX控制电路由硬件复位
[5]	**SLVTOIEN**	**从机模式超时中断使能位** 0=从机模式超时中断禁止

		1=从机模式超时中断使能
[4]	**SLV3WIRE**	**从机三线模式使能位** 只有QSPI支持从机三线模式。从机三线模式中，QSPI控制器可以工作在三线接口，QSPIx_CLK,QSPIx_MISO和QSPIx_MOSI。 0=四线双向接口 1=三线双向接口
[3]	**AUTOSS**	**自动从机片选功能使能位(仅主机)** 0=自动从机片选功能禁止，从机片选信号由SS(QSPIx_SSCTL[0])控制 1=自动从机片选功能使能
[2]	**SSACTPOL**	**从机片选有效极性** 该位定义从机片选信号(QSPIx_SS)有效的极性 0=片选信号QSPIx_SS低电平有效 1=片选信号QSPIx_SS高电平有效
[1]	**Reserved**	保留
[0]	**SS**	**从机片选控制位(仅主机)** 如果AUTOSS为0, 0=设置QSPIx_SS为无效状态 1=设置QSPIx_SS为有效状态 如果AUTOSS置1, 0=保持QSPIx_SS为无效状态 1=进行数据传输时，QSPIx_SS会自动切换到有效状态；无数据传输时则会进入无效状态。QSPIx_SS的有效状态定义于SSACTPOL(QSPIx_SSCTL[2])。

QSPI PDMA 控制寄存器(QSPIx_PDMACTL)

寄存器	偏移	R/W	描述	复位值
QSPIx_PDMACTL	QSPIx_BA+0x0C	R/W	QSPI PDMA 控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**PDMARST**	**PDMA复位** 0=无效 1=复位QSPI控制器的PDMA控制逻辑。该位自动清0。
[1]	**RXPDMAEN**	**PDMA接收使能位** 0=PDMA接收禁止 1=PDMA接收使能
[0]	**TXPDMAEN**	**PDMA发送使能位** 0=PDMA发送禁止 1=PDMA发送使能 **注:**在QSPI主机模式支持全双工传输，如果发送和接收PDMA功能都使能，接收PDMA功能不能在发送PDMA功能之前使能。用户可以先使能发送PDMA功能或者同时使能两个功能。

QSPI FIFO 控制寄存器(QSPIx_FIFOCTL)

寄存器	偏移	R/W	描述	复位值
QSPIx_FIFOCTL	QSPIx_BA+0x10	R/W	QSPI FIFO 控制寄存器	0x4400_0000

31	30	29	28	27	26	25	24
Reserved	**TXTH**			**Reserved**	**RXTH**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**TXFBCLR**	**RXFBCLR**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXUFIEN	**TXUFPOL**	**RXOVIEN**	**RXTOIEN**	**TXTHIEN**	**RXTHIEN**	**TXRST**	**RXRST**

位	描述	
[31]	**Reserved**	保留
[30:28]	**TXTH**	**发送FIFO阈值** 如果发送FIFO缓存的有效数据个数小于或者等于TXTH的设置，TXTHIF将会被设置为1，否则TXTHIF将会被设置为0。
[27]	**Reserved**	保留
[26:24]	**RXTH**	**接收FIFO域值** 如果接收FIFO缓存的有效数据个数大于RXTH的设置，RXTHIF将会被设置为1，否则RXTHIF将会被设置为0。
[23:10]	**Reserved**	保留
[9]	**TXFBCLR**	**清发送FIFO缓存** 0=无效 1=清除发送FIFO指针，TXFULL位将清除为0而且TXEMPTY位也将设置为1。该位置1后，在一个系统时钟后，由硬件清0。 **注:**TX移位寄存器不会清空
[8]	**RXFBCLR**	**清接收FIFO缓存** 0=无效 1=清除接收FIFO指针，RXFULL位将清除为0而且RXEMPTY位将设置为1。该位设置为1后，在一个系统时钟后，由硬件清0。 **注:**RX移位寄存器不会清空
[7]	**TXUFIEN**	**TX下溢中断使能位** 从机模式下TX下溢中断事件发生时，TXUFIF(QSPIx_STATUS[19])会被置1。该位用来使能TX下溢中断。 0=从机TX下溢中断禁止 1=从机TX下溢中断使能

[6]	**TXUFPOL**	**TX下溢数据极性** 0=在从机模式，如果有发送下溢事件发生，QSPI数据输出保持为0。 1=在从机模式，如果有发送下溢事件发生，QSPI数据输出保持为1。 **注:** 1.从机片选信号有效时如果没有任何数据发送则会发生下溢事件。 2.TX下溢事件发生时，QSPIx_MISO管脚状态将由该位决定而不管TXFIFO是否为空。存储在TX FIFO的数据会在下一个传输帧通过QSPIx_MISO发送出去。
[5]	**RXOVIEN**	**接收FIFO溢出中断使能位** 0=接收FIFO溢出中断禁止 1=接收FIFO溢出中断使能
[4]	**RXTOIEN**	**从机接收超时中断使能位** 0=从机接收超时中断禁止 1=从机接收超时中断使能
[3]	**TXTHIEN**	**发送FIFO阈值中断使能位** 0= TX FIFO阈值中断禁止 1=TX FIFO阈值中断使能
[2]	**RXTHIEN**	**接收FIFO阈值中断使能位** 0=RX FIFO阈值中断禁止 1=RX FIFO阈值中断使能
[1]	**TXRST**	**发送复位** 0=无效 1=复位发送FIFO指针和发送电路。TXFULL位将被清0，TXEMPTY位将被置1。该位置1后，大约经过3个系统时钟周期和2个外设时钟周期，硬件会将该位清除为0。如果用户想确认复位是否已经完成，可以读取TXRXRST(QSPIx_STATUS[23])来确认。 **注:**如果从机模式下发生TX下溢事件发生，该位也可以用来使QSPI返回空闲状态。
[0]	**RXRST**	**接收复位** 0=无效 1=复位接收FIFO指针和接收电路。RXFULL位将被清除为0，RXEMPTY位将被设置为1。该位置1后，大约经过3个系统时钟周期和2个外设时钟周期，硬件将该位清除为0。如果用户想确认复位是否已经完成，可以读TXRXRST(QSPI_STATUS[23])来确认。

QSPI 状态寄存器(QSPIx_STATUS)

寄存器	偏移	R/W	描述	复位值
QSPIx_STATUS	QSPIx_BA+0x14	R/W	QSPI 状态寄存器	0x0005_0110

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXRXRST	Reserved			TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPIENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SLVURIF	SLVBEIF	SLVTOIF	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

位	描述	
[31:28]	**TXCNT**	**发送FIFO数据个数(只读)** 该位域表明发送FIFO缓存的有效数据个数。
[27:24]	**RXCNT**	**接收FIFO数据个数(只读)** 该域表明接收FIFO缓存有效数据个数..
[23]	**TXRXRST**	**TX或RX复位状态(只读)** 0=TXRST或RXRST的复位功能已经完成 1=TXRST或RXRST正在复位 **注意:**TXRST和RXRST的复位操作都需要3个系统时钟周期+2个外设时钟周期。用户可以检查该状态位来检测复位功能是否完成。
[22:20]	**Reserved**	保留
[19]	**TXUFIF**	**TX下溢中断标志** 当发送下溢事件发生时，该位被设置为1，数据输出管脚的状态取决于TXUFPOL的设置。 0=无影响 1=当从机片选信号有效时，发送FIFO和发送移位寄存器中没有数据。 **注1**:该位写1清0。 **注2**:当从机片选信号有效时，如果复位从机的发送电路，在3个系统时钟周期和2个外设时钟周期后，此时复位操作已经完成，该位将被设置为1。
[18]	**TXTHIF**	**发送FIFO阈值中断标志(只读)** 0=发送FIFO缓存中的有效数据个数大于TXTH设置的值 1=发送FIFO缓存中的有效数据个数少于或等于TXTH设置的值
[17]	**TXFULL**	**发送FIFO缓存满标志(只读)** 0=发送FIFO缓存未满 1=发送FIFO缓存已满
[16]	**TXEMPTY**	**发送FIFO缓存空标志(只读)**

		0=发送FIFO缓存不空 1=发送FIFO缓存已空
[15]	**SPIENSTS**	**QSPI使能状态位(只读)** 0=禁止QSPI控制器. 1=使能QSPI控制器. **注意:**QSPI外设时钟与系统时钟不同步。为了确保QSPI控制器已经被禁止，该位指示了QSPI控制器的真实状态。
[14:13]	**Reserved**	保留
[12]	**RXTOIF**	**接收超时中断标志** 0=没有接收FIFO超时事件 1=接收FIFO缓存非空且主机模式下，超过64个QSPI时钟周期或从机模式下超过576个QSPI引擎时钟周期，接收FIFO缓存上没有读操作。当接收FIFO缓存被软件读取，超时状态会自动清0。 **注:**该位写1清0
[11]	**RXOVIF**	**接收FIFO溢出中断标志** 当接收FIFO缓存已经满了，接下来的数据将被丢弃，同时该位被设置为1. 0=接收FIFO没有溢出 1=接收FIFO溢出 **注:**该位写1清0
[10]	**RXTHIF**	**接收FIFO阀值中断标志(只读)** 0=接收FIFO缓存的有效数据个数少于或等于RXTH设置的值 1=接收FIFO缓存的有效数据个数大于RXTH设置的值
[9]	**RXFULL**	**接收FIFO缓存满标志(只读)** 0=接收FIFO缓存未满 1=接收FIFO缓存已满
[8]	**RXEMPTY**	**接收FIFO缓存空标志(只读)** 0=接收FIFO缓存不空 1=接收FIFO缓存为空
[7]	**SLVURIF**	**从机模式发送下溢中断标志** 在从机模式，如果发送下溢事件发生，而且从机片选线进入无效状态，该中断标志将被设置为1。 0=未发生从机发送下溢事件 1=发生从机发送下溢事件 **注:**该位写1清0
[6]	**SLVBEIF**	**从机模式位计数错误中断标志** 在从机模式，当从机片选信号线进入无效状态时，如果位计数与DWIDTH不一致，该中断标志将被设置为1。 0=未发生从机模式位计数错误事件 1=发生从机模式位计数错误事件 **注:**如果从机片选激活，但是没有总线时钟输入，当从机片选进入无效状态时，SLVBEIF也将被设置为1.该位写1清0。
[5]	**SLVTOIF**	**从机超时中断标志** 当从机片选信号激活和SLVTOCNT的值不为0时，在检测到总线时钟时，QSPI控制器逻辑的从机超时计数器开始计数。在事务完成之前，如果超时计数器的值大于或等于

		SLVTOCNT(QSPIx_SSCTL[31:16])里的值。将发生从机超时中断事件。 0=从机超时未发生 1=发生从机超时 **注:**该位写1清0
[4]	**SSLINE**	**从机片选线总线状态(只读)** 0=从机片选线状态为0 1=从机片选线状态为1 **注:**该位只在从机模式有效。如果SSACTPOL(QSPIx_SSCTL[2])设置为0，而且SSLINE的值为1，QSPI从机片选处于无效状态。
[3]	**SSINAIF**	**从机片选无效中断标志** 0=从机片选无效中断被清除或没有发生 1=发生了从机片选无效中断 **注:**该位只在从机模式有效，该位写1清0。
[2]	**SSACTIF**	**从机片选激活中断标志** 0=从机片选激活中断被清除或未发生 1=发生了从机片选激活中断 **注:**该位只在从机模式有效，该位写1清0。
[1]	**UNITIF**	**单元传输中断标志** 0=该位清除后，没有事务完成 1=QSPI控制器已完成一个单元传输 **注:**该位写1清0。
[0]	**BUSY**	**忙状态(只读)** 0=QSPI控制器处于空闲状态 1=QSPI控制器处于忙状态. 如下是总线忙条件： ● SPIEN (QSPIx_CTL[0])=1且TXEMPTY=0. ● 对于QSPI主机模式，QSPIx_CTL[0]=1且TXEMPTY=1但是当前传输尚未完成。 ● 对于QSPI主机模式，QSPIx_CTL[0]=1和RXONLY=1. ● 对于QSPI从机模式，QSPIx_CTL[0]=1，从机选择有效时仍有串口时钟输入. ● 对于QSPI从机模式，QSPIx_CTL[0]=1，从机选择无效时发送缓存或发送移位寄存器仍非空.

QSPI 数据发送寄存器(QSPIx_TX)

寄存器	偏移	R/W	描述	复位值
QSPIx_TX	QSPIx_BA+0x20	W	QSPI 数据发送寄存器	0x0000_0000

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

位	描述	
[31:0]	**TX**	**数据发送寄存器** 数据发送寄存器会把要发送的数据传入4级的发送FIFO缓存。该寄存器的有效位数在QSPI模式下取决于DWIDTH(QSPIx_CTL[12:8])。 QSPI模式下，如果DWIDTH为0x08，TX[7:0]位会被发送。如果DWIDTH为0x00，QSPI控制器将发送32位。 **注:**主机模式下，QSPI控制器写该寄存器后，1个APB时钟加6个外设时钟后才会开始QSPI总线时钟的传输。

QSPI 数据接收寄存器(QSPIx_RX)

寄存器	偏移	R/W	描述	复位值
QSPIx_RX	QSPIx_BA+0x30	R	QSPI 数据接收寄存器	0x0000_0000

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

位	描述	
[31:0]	**RX**	**数据接收寄存器（只读）** QSPI控制器有8级FIFO缓存。从QSPI数据输入管脚接收到的数据会被保存到该寄存器。如果RXEMPTY(QSPIx_STATUS[8]不为1，软件可以通过读该寄存器获取FIFO缓存的数据。 该寄存器只读。

6.16 I²C 串行接口控制器(I²C)

6.16.1 概述

I²C为双线，双向串行总线，通过简单有效的连线方式实现器件间的数据交换。I²C标准是多主机总线，包括冲突检测和仲裁，以防止在两个或多个主机同时尝试控制总线时发生数据损坏。

有2组I²C控制器，都支持掉电唤醒功能。

6.16.2 特性

I²C通过SDA 及SCL两条线与连接在总线上的器件传输信息，总线的主要特征有：

- 支持最多2组I²C接口
- 支持主机/从机模式
- 主从机之间双向数据传输
- 总线支持多主机 (无中心主机)
- 支持标准模式 (100 kbps), 快速模式 (400 kbps) 和快速加模式(1 Mbps)
- 多主机间同时传输数据仲裁，避免总线上串行数据损坏
- 总线采用串行同步时钟，可实现设备之间以不同的速率传输
- 内建14位超时溢出定时器，当I²C总线中止且定时器超时溢出时，产生I²C中断
- 可配置不同时钟以适用于可变速率控制
- 支持7位地址模式
- 支持多地址识别（ 4组从机地址带mask 选项）
- 支持掉电唤醒功能
- 支持带有一个缓冲的PDMA
- 支持建立/保持时间可编程
- 支持总线管理 (兼容SM/PM) 功能

章节	子章节	M031xB/C/D/E M032xB/C/D/E	M031xG/I M032xG/I
6.16.5功能描述	6.16.5.2操作模式 - 总线管理 (SMBus/PMBus 兼容) - 设备识别 – 从机地址 - 总线协议 - 地址解析协议(ARP) - 接收命令和数据应答控制 -主机通知协议 - 总线管理警报 - 包错误检查 -超时 -总线管理超时		•

章节	子章节	M031xB/C/D/E M032xB/C/D/E	M031xG/I M032xG/I
	- 总线时钟低电平超时 - 总线空闲侦测		
	I2C总线管理控制寄存器(I2C_BUSCTL)		•
	I2C总线管理定时器控制寄存器(I2C_BUSTCTL)		•
	I2C总线管理状态寄存器(I2C_BUSSTS)		•
	I2C包错误检查字节数寄存器(I2C_PKTSIZE)		•
	I2C包错误检查字节值寄存器(I2C_PKTCRC)		•
	I2C总线管理定时器寄存器(I2C_BUSTOUT)		•
	I2C总线管理时钟低定时器寄存器(I2C_CLKTOUT)		•

表 6.16-1 不同芯片 I^{2}C 特性比较表

6.16.3 框图

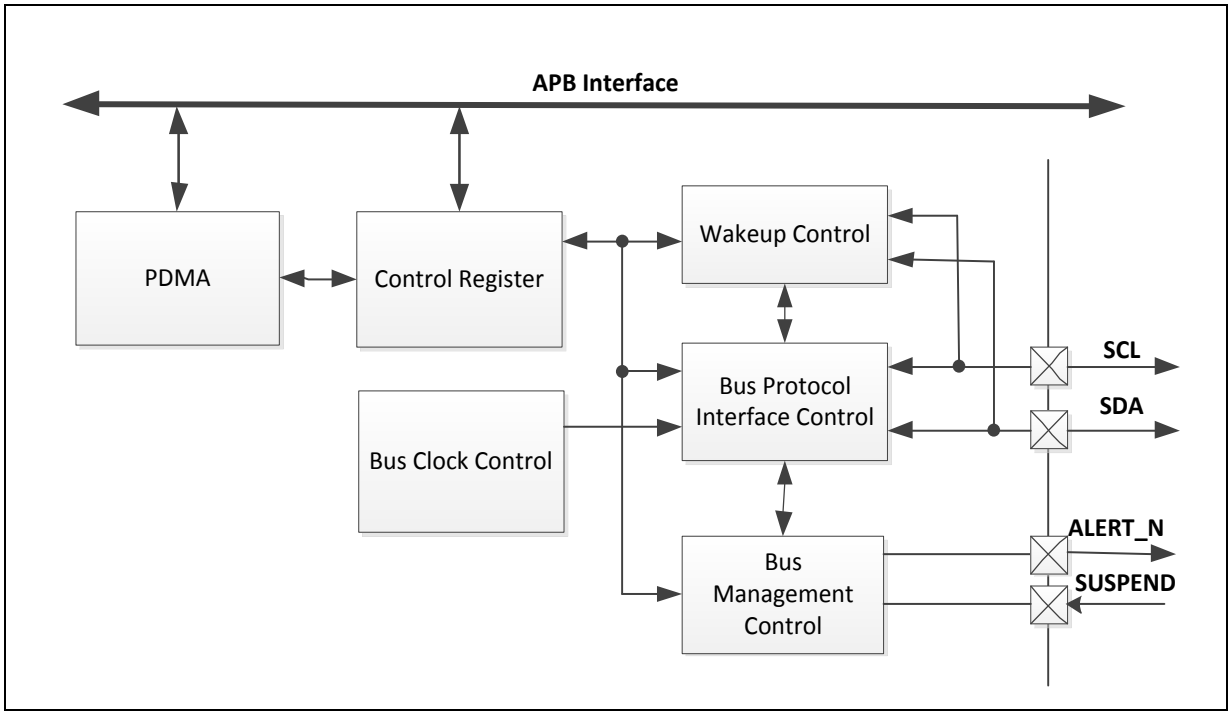


图 6.16-1 I^{2}C 控制器框图

6.16.4 基本配置

6.16.4.1 I2C0基本配置

- 时钟源配置
  - 使能I2C0时钟的控制位是I2C0CKEN (CLK_APBCLK0 [8])
- 复位配置
  - 复位I2C0控制器的控制位是I2C0RST(SYS_IPRST1 [8])

6.16.4.2 I2C1基本配置

- 时钟源配置
  - 使能I2C1时钟的控制位是I2C1CKEN (CLK_APBCLK0 [9])
- 复位配置
  - 复位I2C1控制器的控制位是I2C1RST(SYS_IPRST1 [9])

6.16.5 功能描述

在I²C总线上，数据通过时钟线SCL和数据线SDA在主从机间逐一字节同步传送。每个字节数据长度是8位，一个SCL 时钟脉冲传输一个数据位，数据由最高位MSB开始传输，每个字节传输后跟随一个应答位，每个位在SCL为高时采样。因此，SDA 线可能在SCL为低时改变，但在SCL为高时必须保持稳定。当SCL为高时，SDA线上的跳变视为一个命令(START或STOP)。更多关于I²C总线时序的细节请参考下图。

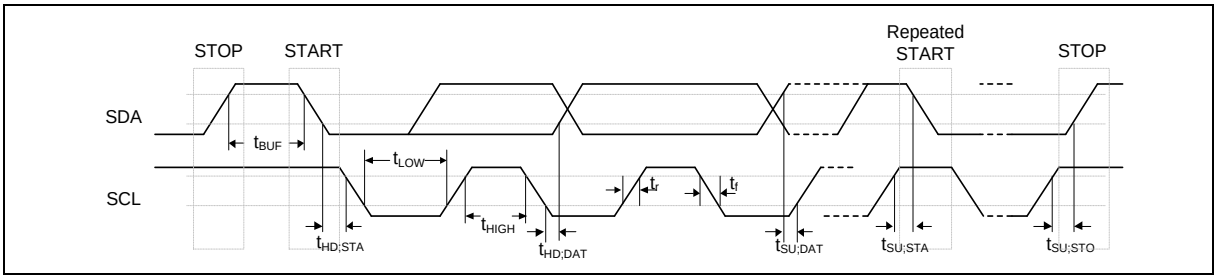


图 6.16-2 I²C 总线时序

片上I²C外设提供了一个符合I²C总线规范的串行接口。I²C端口自动处理字节传输。通过设置寄存器I2C_CTL0的I2CEN为'1'可使能该端口。I²C硬件接口通过数据线SDA和时钟线SCL两个引脚连到I²C总线。当I/O 引脚作为I²C端口使用时，用户必须事先设定I/O引脚为I²C功能。

注：SDA和SCL两个管脚需要上拉电阻，因为这个两个管脚为开漏脚。

6.16.5.1 I²C 协议

标准I²C 协议如图6.16-3所示，通常标准通讯有以下4部分：

- 起始信号(START) 或者重复起始信号(Repeated START)
- 从机地址传输和R/W 位传输
- 数据传输
- 产生STOP信号

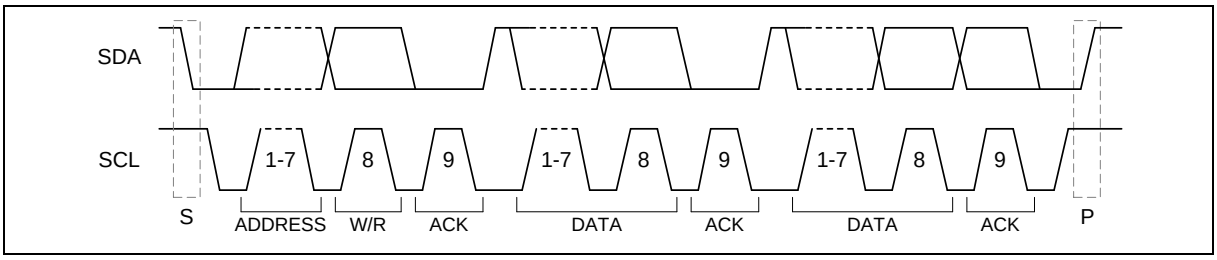


图 6.16-3 I²C 协议

- 起始（START）或重复起始（Repeated START）信号

当总线处于释放/空闲状态下，说明没有主机设备占用总线（SDA与SCL线同时为高），主机可以通过发送起始(START)信号来发起传输过程。起始信号，通常表示为“S”位，当SCL线为高时，SDA 线上信号由高至低变化，就被定义为起始信号(START)。起始信号(START)表示一个新的数据传输的开始。

主机发送完地址字节（地址和读/写位）后，可以发送任何数量的字节数据并带一个停止信号。也可以用另一个起始信号替代停止信号，随后是地址(包含读写位)和更多的数据。这个起始信号叫做重复起始(Repeated START)。这个定义可以用来发送任意个起始信号，它的目的是在不释放总线情况下，对一个或多个设备能读/写操作，而不让操作被打断。用这个方法主机跟其他从机或同一个从机在不同方向传输（例如，写设备到读设备）不用释放总线。

- STOP信号

主机可以通过产生一个停止信号来终止数据通信。停止信号，通常表示为P位，当SCL线为高时，SDA线上信号由低至高变化，就被定义为停止信号。

起始（START），重复起始（Repeat START）和停止（STOP）信号波形图如图6.16-4所示。

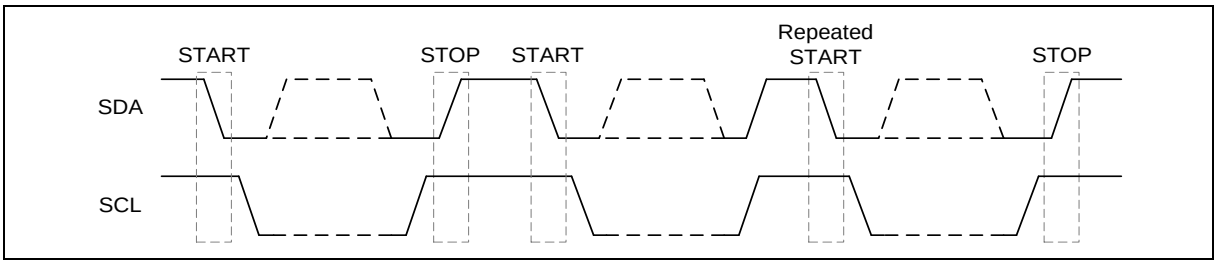


图 6.16-4 起始（START）和停止（STOP）信号条件

- 从机地址传输

（重复）起始信号后，主机发送从机地址(SLA)以确定通信的目标设备。从机地址由一个字节组成，传输一个地址字节后，接收到正确地址的从机一定会应答。

因此，从机设备地址可以设置，从机设备可以匹配接收地址。如果匹配，从机通过(SDA = 0)给出应答(ACK)信号，从机通过(SDA = 1)给出不应答(NACK)信号。除了编程地址匹配外，当从机能够处理相应请求，另外一个地址也应该有应答回复。

- 数据传输

当从机设备地址和R/W位被成功识别到，就可以根据R/W位所决定的方向按一字节一字节方式进行数据传输。每个字节传输完后，紧接着的第9个SCL时钟周期会有一个应答(ACK)信号位。如果从机上产生无应答信号(NACK)，主机可以产生停止信号来中止数据传输或者产生重复起始信号开始新一轮数据传输。

当主机作为接收设备时，发生无应答信号(NACK)给从机，则从机将释放SDA线，让主机产生停止信号或重复起始信号。图6.16-5和图6.16-6为位发送和应答信号波形图。

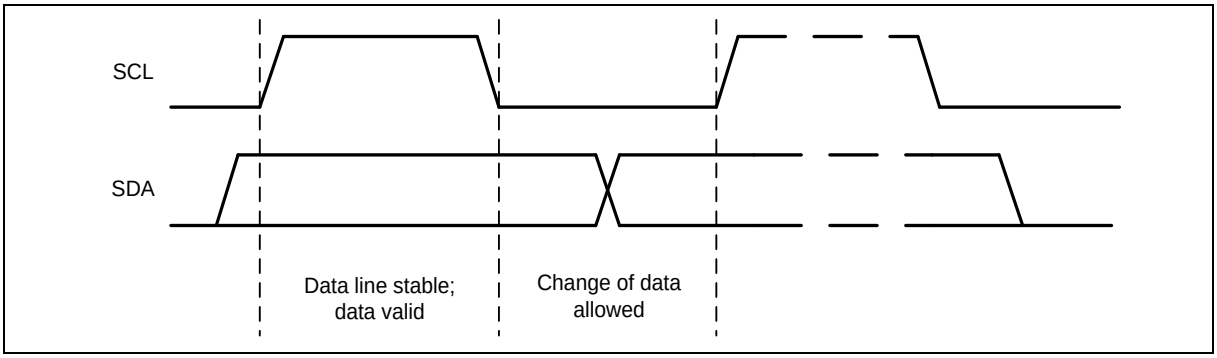


图 6.16-5 在 I²C 总线上位传输

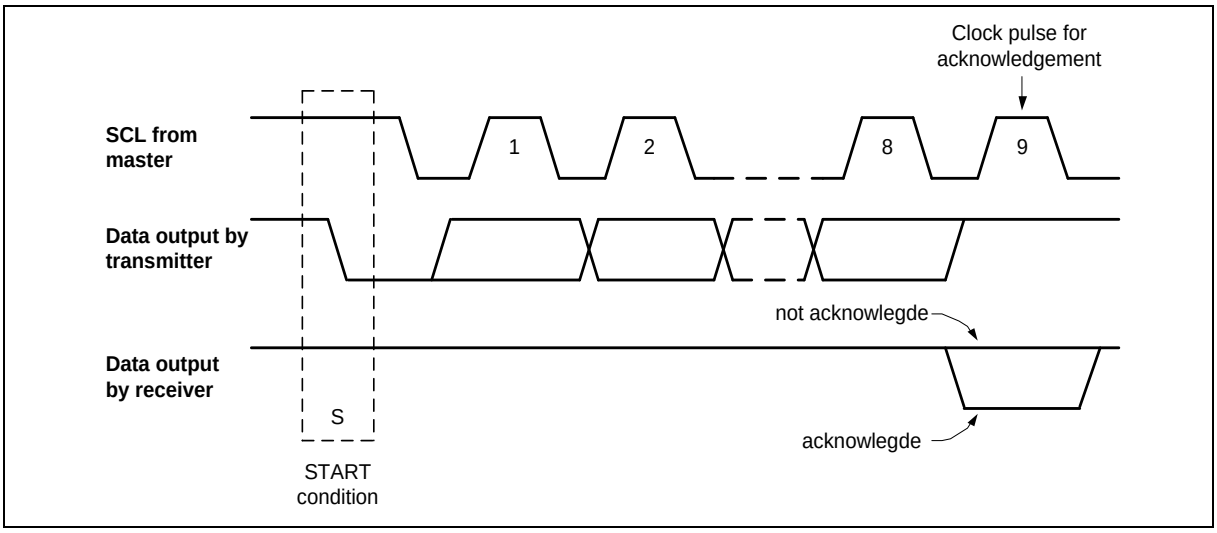


图 6.16-6 在 I²C 总线上应答

- 在 I²C 总线上数据传输

  图6.16-7表示主机向从机写7位数据。主机发出一个7位地址和1位写指示，表示主机想要传送数据给从机。从机回复应答给主机之后，主机继续传输数据。

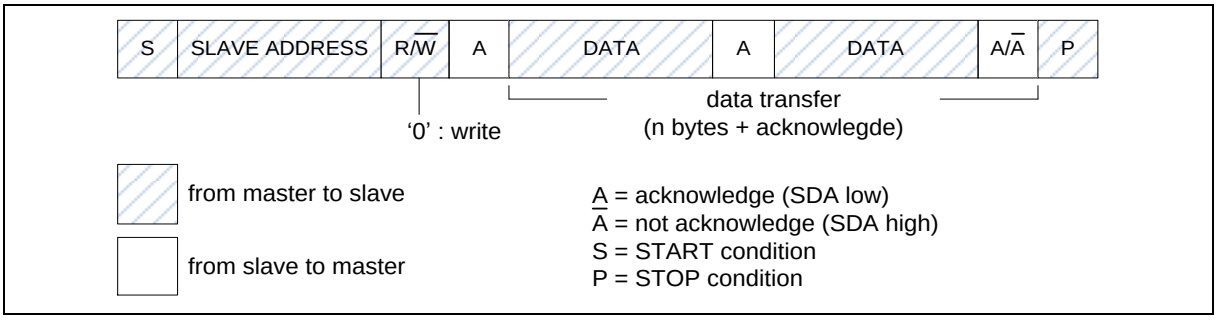


图 6.16-7 主机向从机发送 7 位数据

图6.16-8表示主机向从机读取数据。主机发7位地址寻址和1位读指示，表示主机要向从机读取数据，从机返回应答给主机后，就开始给主机传输数据。

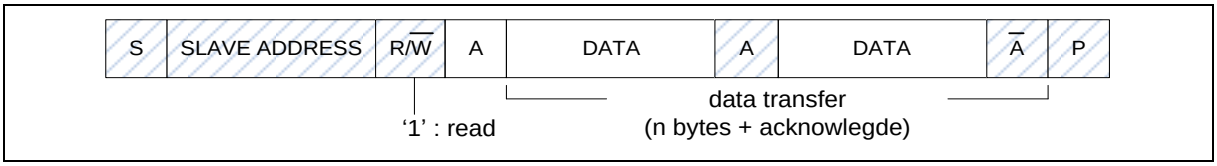


图 6.16-8 主机从从机读取 7 位数据

6.16.5.2 操作模式

片上I²C端口支持三种工作模式：主机模式、从机模式和广播呼叫模式。

应用中，I²C 端口可以作为主机和从机。在从机模式，I2C端口硬件会查找自身从机地址和广播呼叫地址，如果这两个地址的任一个被检测到，并且从机想要从主机接收或向主机发送数据(通过设置AA位)，应答脉冲将会在第9个时钟发出，此时，如果中断使能，则主机和从机设备上都会发生一次中断请求。在微控制器想要成为总线主机时，在进入主机模式之前，硬件需等待到总线空闲，以保证合理的从机动作不会被打断。在主机模式下，如果总线仲裁丢失，I2C端口立即切换到从机模式，并可以在同一次串行传输过程中检测自身的从机地址。

为控制I²C总线的各种模式传输，用户需要按照寄存器I2C_STATUS的当前状态码来设置I2C_CTL0、I2C_DAT寄存器。换句话说，每个I2C总线动作都要检查I2C_STATUS0寄存器的当前状态，然后再设置I2C_CTL0、I2C_DAT寄存器执行总线动作。最后，通过I2C_STATUS0检查响应状态。

在寄存器I2C_CTL0[3]的SI标志清除后，寄存器I2C_CTL0的这些位STA、STO和AA用来控制I²C硬件的下一个状态。当完成一个新的动作，I2C_STATUS0的状态代码将被更新，I2C_CTL寄存器的SI标志将被设置。如果I²C中断控制位INTEN (I2C_CTL0 [7])被设置，新状态代码对应的动作或者软件将在中断服务程序中被执行。

图6.16-9表示当前I²C状态码是0x08，然后通过设置I2C_DATA=SLA+W和(STA,STO,SI,AA) = (0,0,1,x)发送地址到I²C总线。如果总线上有从机匹配相应的地址且返回ACK，I2C_STATUS0状态码更新为0x18。

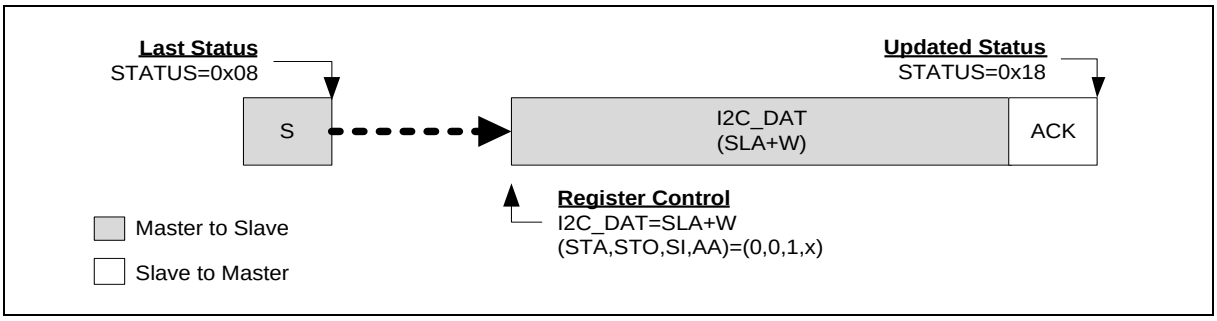


图 6.16-9 根据当前 I²C 状态控制 I²C 总线

主机模式

图6.16-10和图6.16-11,展示了I²C主机模式所有可能的协议。用户需要遵循恰当的流程来实现I²C协议。

换句话说，用户可以发送一个起始信号到总线，I²C总线将设置成主机传输模式 (图6.16-10) 或主机接收模式 (图6.16-11) 起始信号发送成功后新的状态码将是0x08。起始信号后，用户可以发送从机地址，读/写位，数据和重复起始，停止来执行I²C协议。

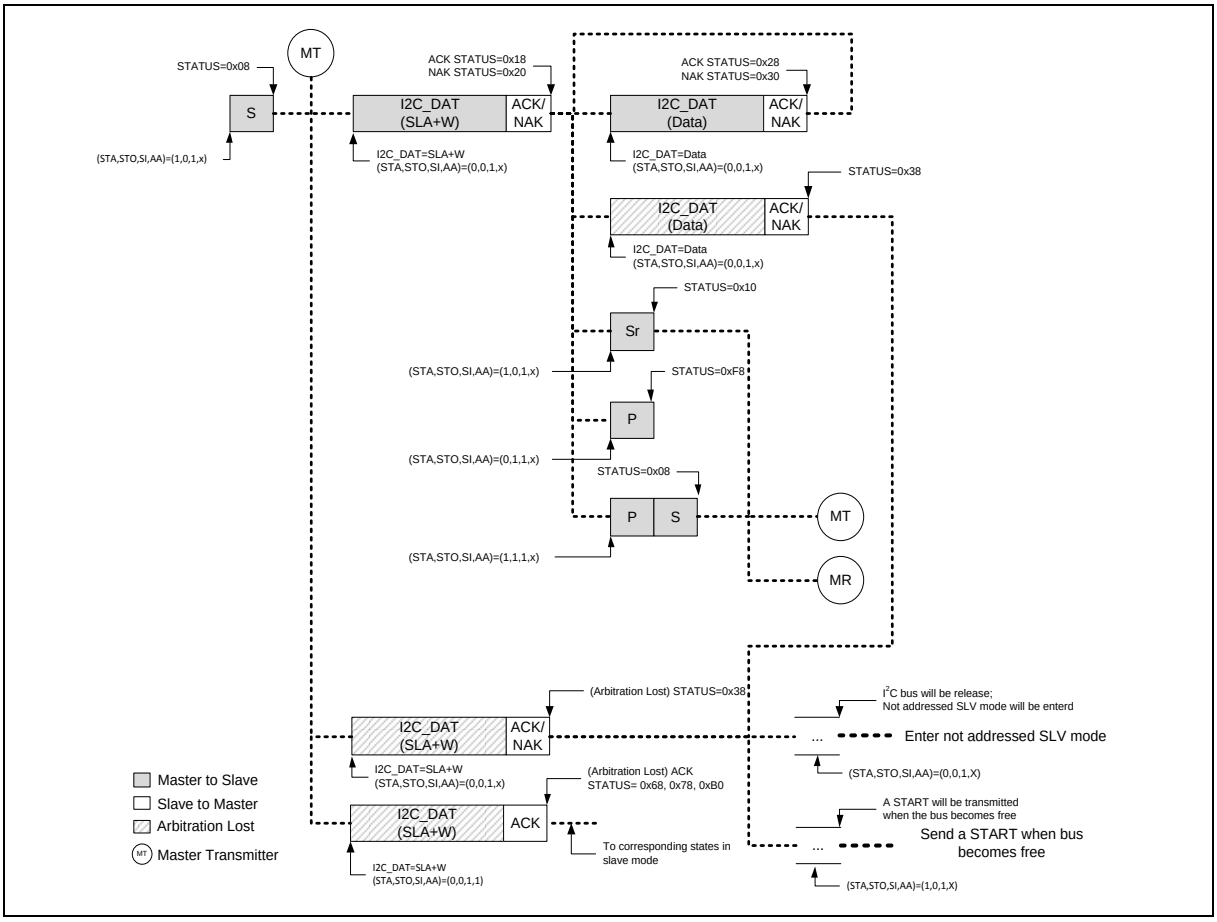


图 6.16-10 主机传输模式控制流程

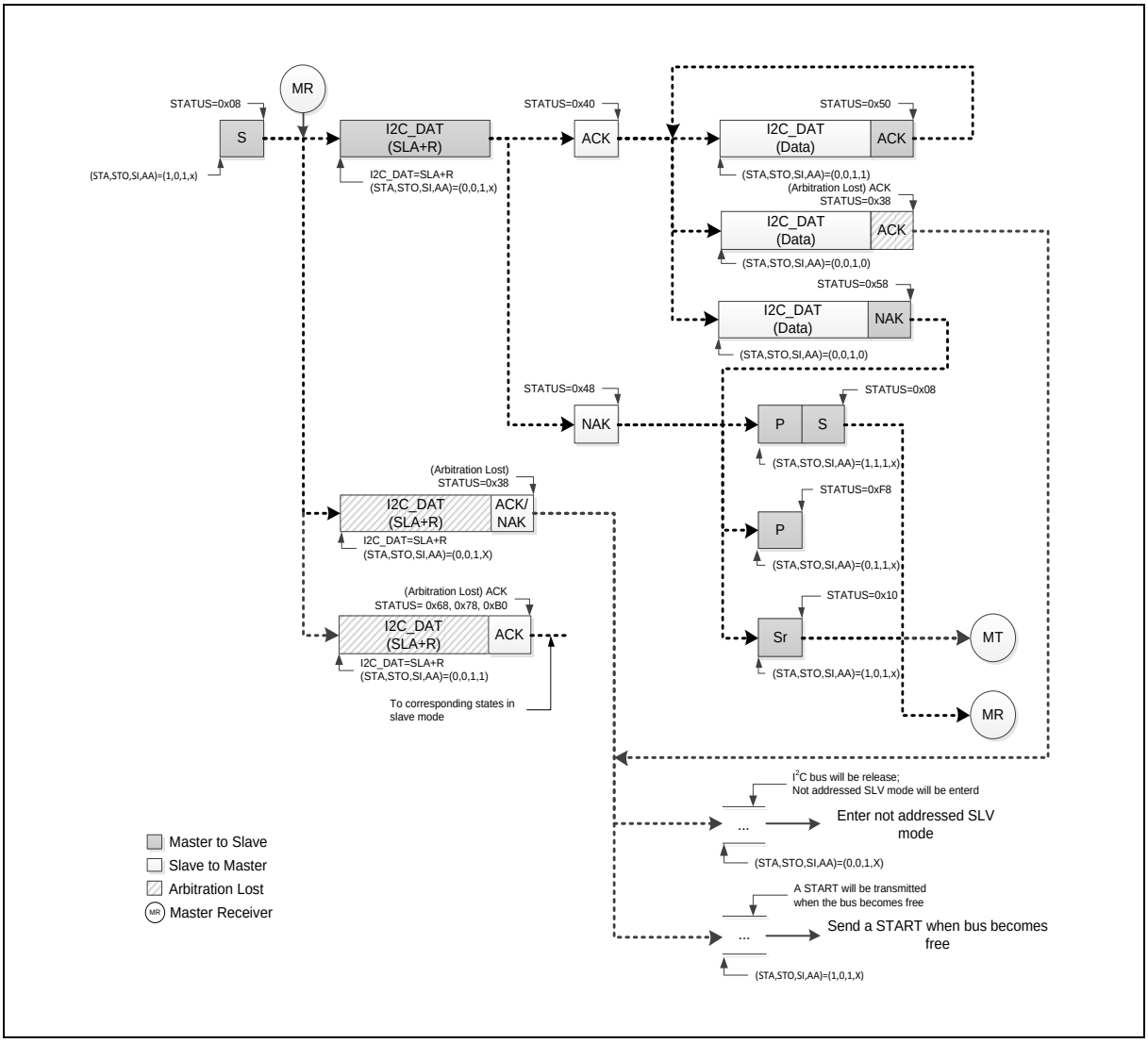


图 6.16-11 主机接收模式控制流程

如果I2C在主机模式并且仲裁丢失，状态码将置为0x38。当状态码为0x38时，当总线空闲时，用户可以设置(STA, STO, SI, AA) = (1, 0, 1, X)发送起始信号来重新开始主机操作。否则，用户可以设置(STA, STO, SI, AA) = (0, 0, 1, X)来释放总线，并进入无地址从机模式。

从机模式

复位后默认情况下，I^2C不会被寻址，并且不会识别I^2C总线上的地址。用户可以通过I2C_ADDRn(n=0~3)设置从机地址和设置 (STA, STO, SI, AA) = (0, 0, 1, 1)来让I^2C识别主机发送的地址。图6.16-12所示为I2C从机模式的所有可能的流程。用户需要遵循如图6.16-12的流程来实现他们的I^2C协议。

如果在主机模式总线仲裁丢失，I^2C端口立即切换到从机模式，并且在同一串行传输中识别自有的从机地址。如果在仲裁丢失后识别到地址是SLA+W（主机想写数据到从机），状态码是0x68。如果在仲裁丢失后识别到地址是SLA+R（主机向从机读数据），状态码是0xB0。

注： I^2C通信期间，在从机模式下，当对SI标志写‘1’清除时，SCL 时钟将被释放。

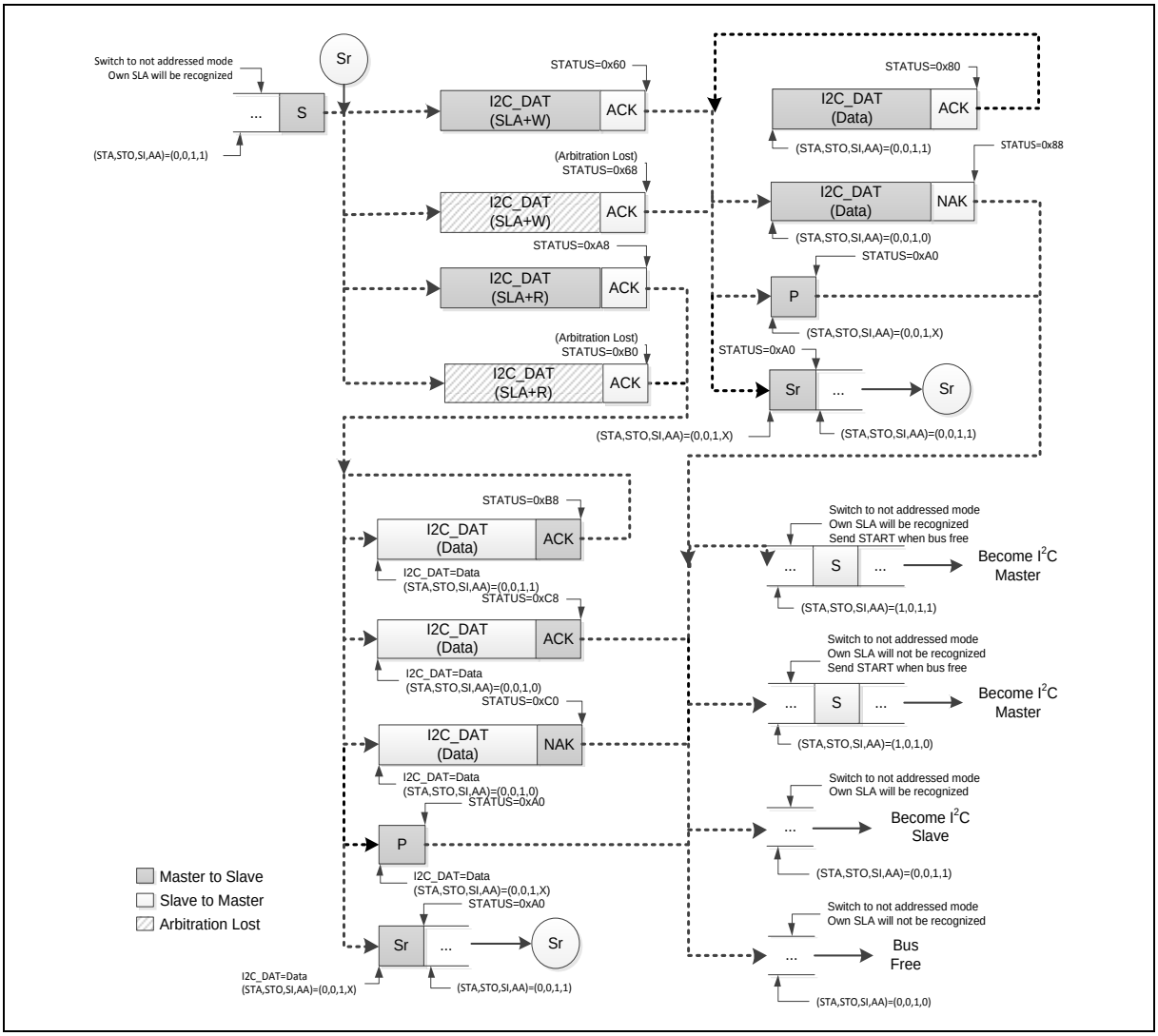


图 6.16-12 从机模式控制流程

如果I2C在可寻址从机模式接收数据时，收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循上图状态码是0x88的操作。

如果I2C在可寻址从机模式发送数据时却收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循上图状态码是0xC8的操作。

注：从机获得0x88, 0xC8, 0xC0 和0xA0状态后,从机可以切换到无地址模式，自身SLA不会被辨识。如果进入这种状态，从机不再接收主机任何信号或地址。在这时候，I2C总线进入空闲状态了。

广播呼叫(GC)模式

如果 GC位(I2C_ADDR0 [0]) 被设置, I^2C端口硬件将响应广播呼叫地址(0x00).用户可以通过清GC位来禁止广播呼叫功能。当I^2C在从机模式时且GC位被设置，可以接收主机地址(0x00)的广播呼叫，将遵循广播模式状态。

当地址匹配，广播呼叫模式能被唤醒，注意默认地址为0x00，但是用户必须设置除0x00以外的地址。

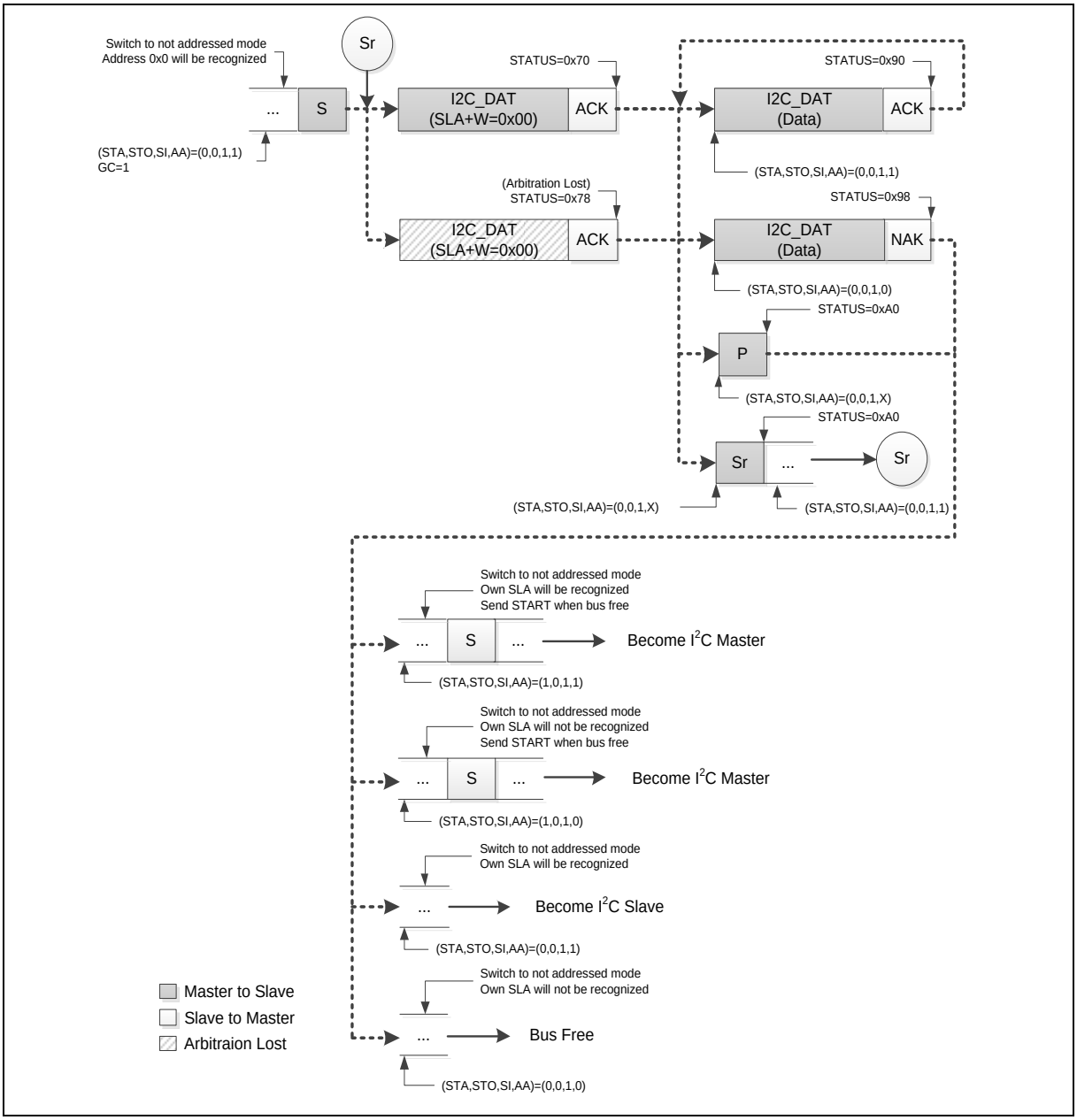


图 6.16-13 GC 模式

如果I2C在广播呼叫模式接收数据时，却收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循上图上状态码是0x98的流程处理。

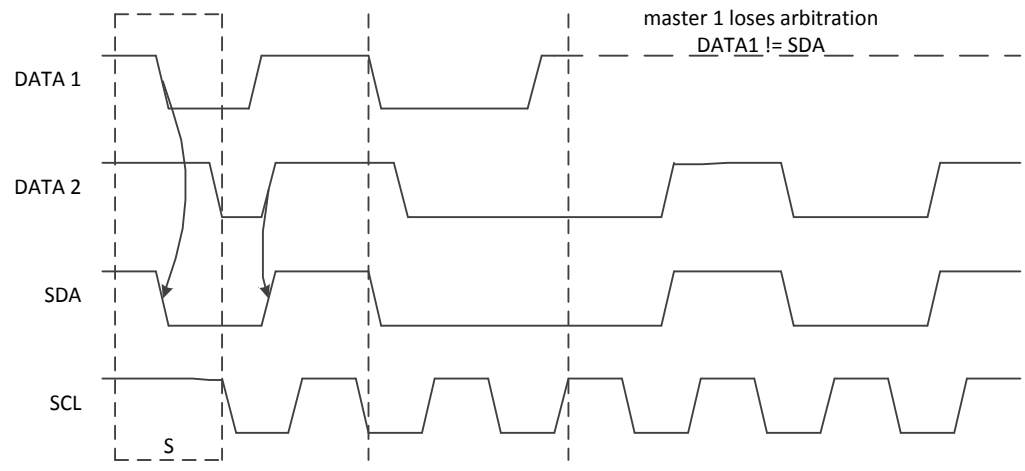
注：从机获得0x98和0xA0状态后，从机会切换到无地址模式并且自身SLA将不被辨识。如果进入这种状态，从机不再接收主机的任何信号或地址。在这时候，I2C总线进入空闲状态了。

多主机模式

在一些应用中，一个I^2C总线上有两个或多个主机同时访问从机，并有可能同时在传送数据。I^2C支持多主机模式，包含有冲突检测和仲裁，防止数据损坏。

如果两个主机同时发命令，通过仲裁来决定哪个优先并继续发命令。仲裁是在SCL为高时在SDA上执行的。每一个主机都会检测总线上的SDA信号是否符合它产生的SDA信号。如果检测到总线上SDA为低，

但应该为高，则这个主机将失去仲裁。设备在仲裁丢失后会产生SCL脉冲直到本字节结束，然后释放总线进入从机模式。仲裁可以一直进行到所有数据被传输完。这样意味着多主机系统中主机必须监控总线和做相应的处理。



Arbitration procedure of two masters

图 6.16-14 仲裁丢失

- 当I2C_STATUS = 0x38代表接收到一个仲裁丢失。仲裁丢失事件可能发生在发送起始位，数据位或者停止位间。用户可以在总线空闲时设置(STA, STO, SI, AA) = (1, 0, 1, X)来再次发送起始信号或者设置(STA, STO, SI, AA) = (0, 0, 1, X)发送停止信号来返回到无地址从机模式。用户可以通过ONBUSY (I2C_STATUS1 [8])查询总线空闲状态。
- 当I2C_STATUS0= 0x00，接收到一个“总线错误”，为了从错误总线恢复到I2C总线，STO应被设置且SI应将被清0，然后对STO清0来释放总线。
  - 设置(STA, STO, SI, AA) = (0, 1, 1, X)停止当前传输。
  - 设置(STA, STO, SI, AA) = (0, 0, 1, X)释放总线。

总线管理 (SMBus/PMBus 兼容)

该部分相关内容只适合支持总线管理特性的I^2C接口。

简介

总线管理是个I^2C接口，通过总线管理各种设备可以相互通信以及与系统的其它部分通信。它是基于I^2C的操作规范。总线管理为系统和电源管理相关任务提供控制总线。

该外设与SMBUS规范2.0版本(*http://smbus.org/specs/*)以及PMBUS规范1.2版(*http://pmbus.org/*)相兼容。

系统管理总线规范涉及到3个类型的设备。

- 从设备，可以接收数据或响应命令。
- 主设备，可以产生命令，产生时钟和结束传输。
- 特定主设备的主机，可以给系统CPU提供主接口，主机必须是主-从设备而且必须支持SMbus主机识别协议，系统里只允许有一个主机。

总线管理设备基于I^2C规范2.1版。

设备识别 – 从机地址

任何存在总线管理的从机设备都有一个唯一的地址称为从机地址。下面的地址为保留而且禁止使用或者赋予任何总线管理设备。(详细请参看SMBus规范)。

从设备地址 位 7-1	读/写位 位 0	说明
0000 000	0	广播呼叫地址
0000 000	1	起始字节
0000 001	X	CBUS 地址
0000 010	X	不同总线格式保留地址
0000 011	X	保留将来使用
0000 1XX	X	保留将来使用
0101 000	X	保留访问总线主机
0110 111	X	保留访问总线默认地址
1111 0XX	X	10位从机寻址
1111 1XX	X	保留将来使用
0001 000	X	SMBus 主机
0001 100	X	SMBus警报相应地址
1100 001	X	SMBus设备默认地址

表 6.16-2 SMBus 保留地址

总线协议

任何设备都可能有11种命令协议，设备可以使用11种协议中某一或者所有来通信。这些协议为Quick Command, Send Byte, Receive Byte, Write Byte, Write Word, Read Byte, Read Word, Process Call, Block Read, Block Write和Block Write-Block Read Process Call。这些协议必须由软件来执行(协议的更详细内容请参看SMBus规范2.0版本或更新版本)。

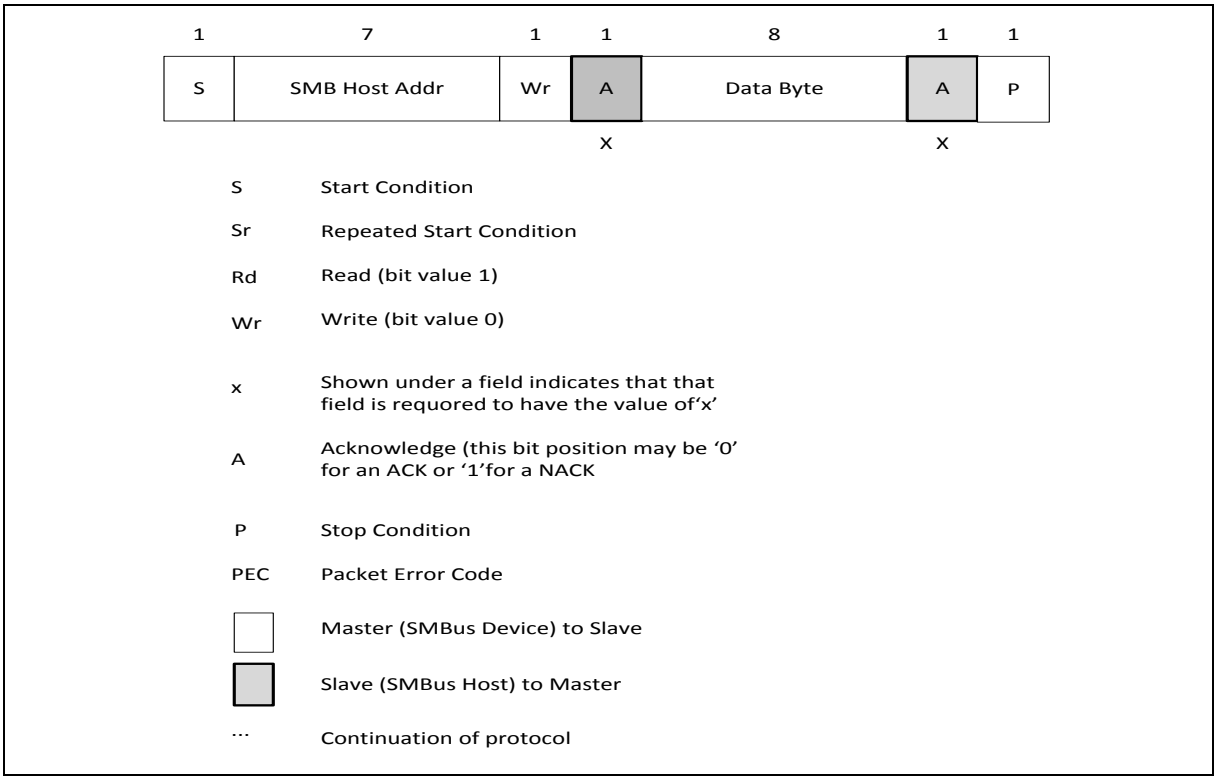


图 6.16-15 总线管理主要元素包协议图

地址解析协议(ARP)

从机地址冲突，总线管理可以通过动态给从设备赋予唯一的新地址解决。为了提供地址识别机制来区分各个设备，每个设备必须分配一个唯一设备标识符(UDID)，由软件分配128位数字。

该设备支持地址解析协议(ARP)。通过设置BUSEN (I^2C_BUSCTL[7])、BMDEN (I^2C_BUSCTL[2])和ALERTEN (I^2C_BUSCTL[4])来使能总线管理设备默认地址(0b1100 001)。ARP命令必须由用户来执行，支持ARP的从机模式也可以执行仲裁。

接收命令和数据应答控制

总线管理接受器必须能够NACK每个收到的命令或数据。为了允许从机模式下的ACK控制，必须通过使能ACKMEN位(I^2C_BUSCTL[0])来设置成从设备字节控制模式。

主机通知协议

为防止未知设备未知信息的格式进入总线管理主机控制器，仅支持使用写字协议更改的协议。此协议将标准写字协议的命令码改为改变的报警设备地址。

该外设支持主机通知协议，通过设置BUSEN (I^2C_BUSCTL[7])、BMHEN (I^2C_BUSCTL[3])和ALERTEN (I^2C_BUSCTL[4])实现。在该情况下，主机将会响应总线管理主机地址(0b0001000)。该协议工作时设备作为主机，而主机作为从机。

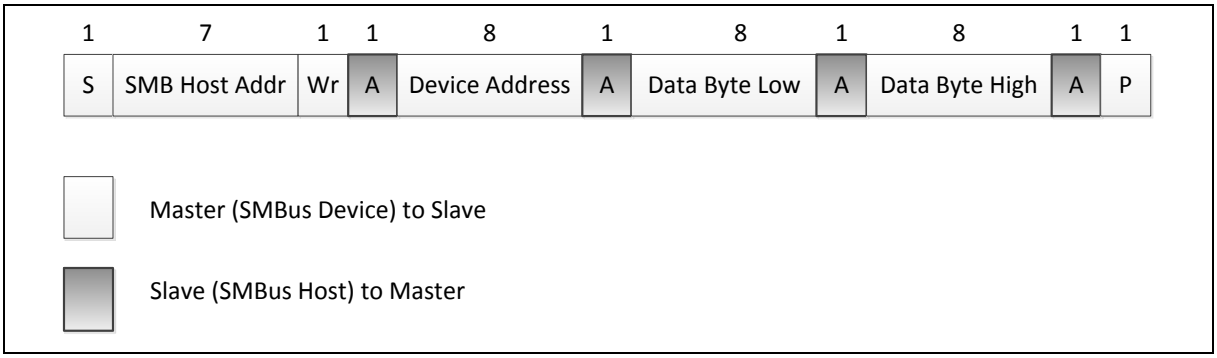


图 6.16-167 位可寻址的设备与主机通信

总线管理警报

总线管理支持可选警报信号。如果需要，从设备可以通过总线管理警报管脚向主机发信号请求通话。主机会进入中断并且向所有总线上的设备发送警报响应地址(0b000100），只有拉低总线管理警报管脚的设备才会应答警报响应地址。

当配置成为从设备(BMHEN=0)时，总线管理警报管脚可以通过设置寄存器(I2C_BUSCTL[4]) ALERTEN 位拉低电平，与此同时警报响应地址(ARA)也使能了。

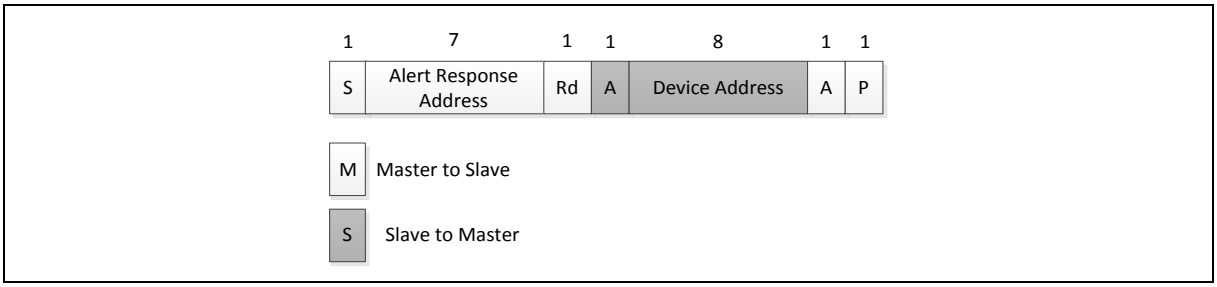


图 6.16-177 位可寻址的设备对警报响应地址的响应

配置成为主机(BMHEN=1)时，当总线管理警报管脚检测到下降沿且ALERTEN=1时，警报标志(I^2C_BUSSTS[3])置位。当ALERTEN=0时，即使外部总线管理警报管脚为低电平，警报线还是认为为高。如果总线管理警报管脚不需要，ALERTEN = 0时，总线管理警报管脚可用作标准的GPIO口。

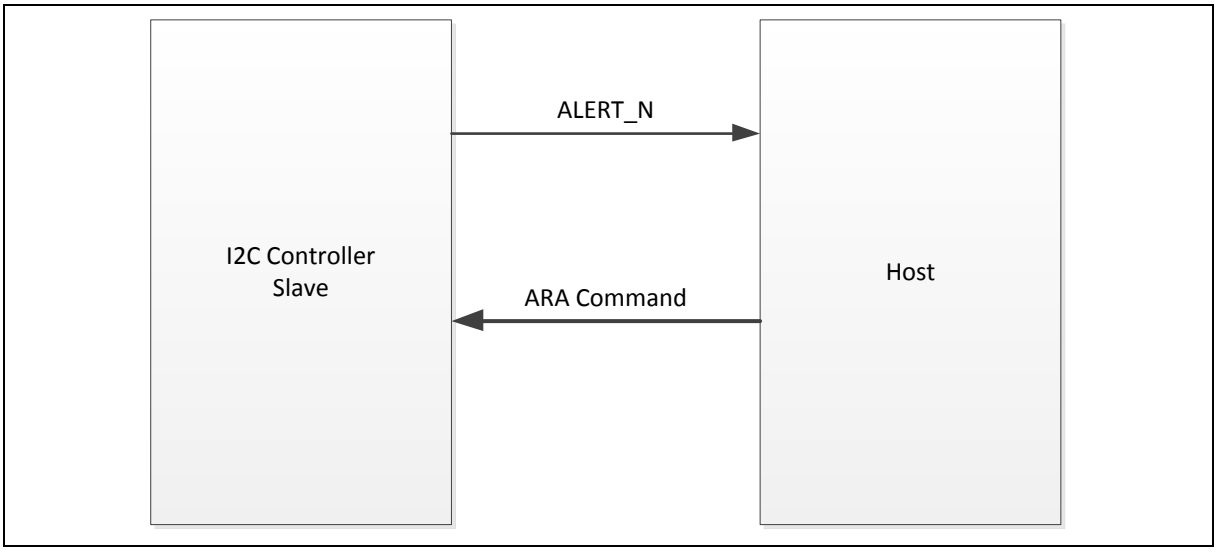


图 6.16-18 总线管理报警功能

包错误检查

包错误检查机制有在SMBus规范里介绍，是用来改进通信的可靠性和健壮性。包错误检查通过每次信息传输的最后增加一个错误包编码(PEC)来实现。PEC是通过使用C(x) = x8 + x2 + x + 1 CRC-8多项式来对所有信息字节(包括地址和读/写位)来计算。

当PECEN位(I^2C_BUSCTL[1])使能时，外设内嵌一个硬件PEC计算器，当收到字节数据与硬件计算的PEC值不匹配时，允许自动发送一个无应答信号。PEC计算的值也可以通过I^2C_PKTCRC读回。

超时

外设内嵌了硬件计数器为了与SMBus规范2.0版本里的3个超时相兼容。

总线管理超时：

总线忙时SCLK低电平超时条件

$T_{Time\text{-}out}$ = (BUSTO(I^2C_BUSTOUT[7:0]) +1) x 16x1024 (14-bit) x T_{PCLK} (如果 TOCDIV4 = 0).

= (BUSTO(I^2C_BUSTOUT[7:0])+1) x 16x1024 (14-bit) x 4 x T_{PCLK} (如果 TOCDIV4 = 1)

总线空闲条件(SCLK和SDA为高)

$T_{Time\text{-}out}$ = (BUSTO(I^2C_BUSTOUT[7:0])+ 1) x 4 x T_{PCLK}

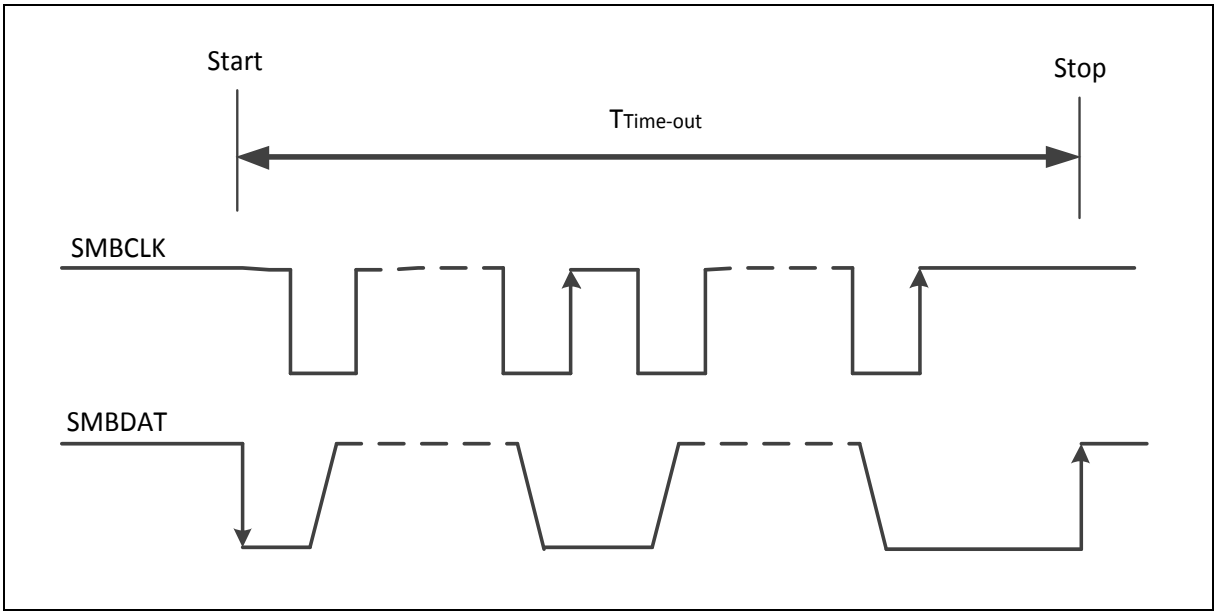


图 6.16-19 总线管理超时时序

总线时钟低电平超时：

在主机模式，侦测主机低电平延时时间($T_{LOW:MEXT}$)。

在从机模式，侦测从机低电平延时时间($T_{LOW:SEXT}$)。

$T_{TLOW:EXT}$ = (CLKTO (I^2C_CLKTOUT[7:0])+1) x 16x1024 (14-bit) x T_{PCLK} (如果 TOCDIV4= 0).

= (CLKTO (I^2C_CLKTOUT[7:0])+1) x 16x1024 (14-bit) x 4 x T_{PCLK} (如果 TOCDIV4= 1)

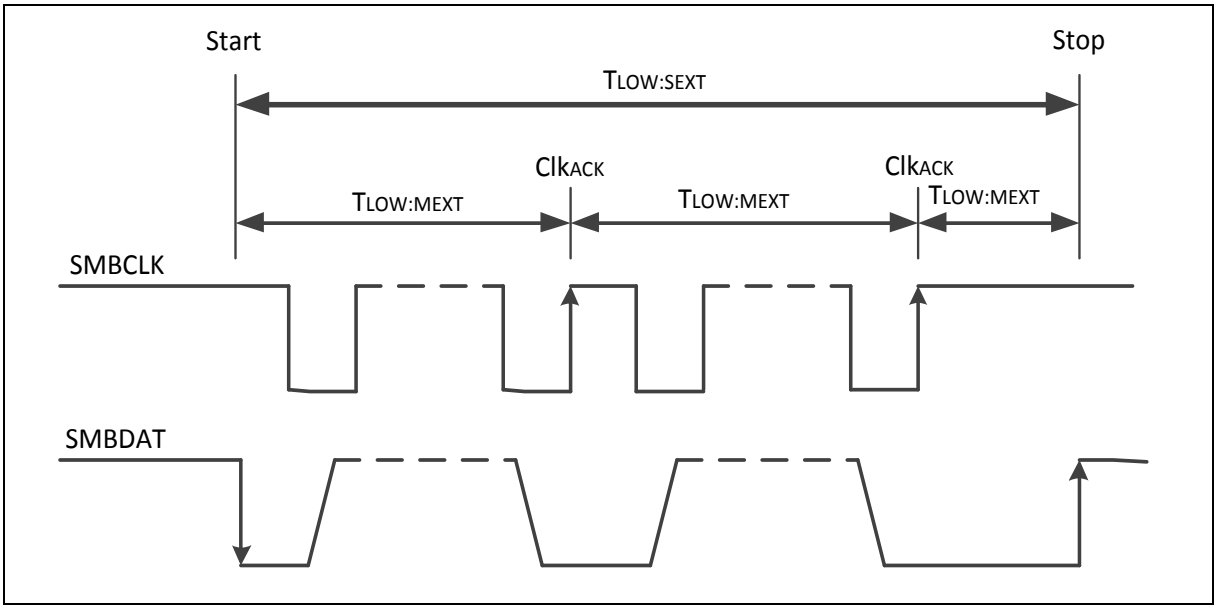


图 6.16-20 总线时钟低电平超时时序

总线空闲侦测

如果侦测到时钟和数据线上信号高电平时间已经T_{IDLE}大于$T_{HIGH,MAX}$时，主机可以认为总线已经空闲。时序参数包括了一个主机已被动态加入到总线，可能没有检测到SMBCLK和SMBDAT线上状态的变化条件。在这种情况下，主机必须等待足够长的时间，以确保传输不在进行中。外设支持硬件空闲检测。

6.16.5.3 PDMA 传输功能

I^2C控制器支持PDMA传输功能。

当TXPDMAEN (I2C_CTL1 [0]) 被置1, I^2C控制器将自动请求PDMA控制器开始PDMA传输处理。

当 RXPDMAEN (I2C_CTL1 [1]) 被置1， I^2C控制器将开始PDMA接收处理，当有数据被写进接收缓冲。

当I^2C 进入PDMA模式，大部分中断状态将会被屏蔽。除让总线错误或不应答或停止中断(0x20, 0x30, 0x38, 0x48, 0x58, 0x00, 0xA0, 0xC0, 0x88 和0x98)中断发生外，其他中断都被屏蔽。

I2C 控制器仅在主机传输模式时可以设置PDMASTR (I2C_CTL1 [8])。如果PDMASTR 被清零， PDMA 传输完成和缓冲空后，I^2C 自动发送停止（STOP）信号。如果PDMASTR被置1， PDMA 传输完成和缓冲空后，SI将被置1并且I2C总线进入时钟延展状态。

6.16.5.4 可编程设置和保持时间

为了确保正确数据设置和保持时间，时序必须被配置。通过设置HTCTL (I2C_TMCTL[24:16])去配置保持时间和 STCTL (I2C_TMCTL[8:0])去配置设置时间。

延时时序参照外设时钟 (PCLK)。当设备延展主设备时钟时，设置和保持配置值将不会受时钟延展的影响。

用户应该注意设置和保持时间配置的限制，时序设置必须遵循I^2C协议。一旦设置时间配置大于设计限制，意思是如果设置时间设置使SCL输出少于3个PCLKs，I^2C 控制器不能正常工作，由于SCL必须3个采样周期。一旦保持时间配置大于I^2C时钟限制，I^2C将发生总线错误。建议用户设置时序前根据波特率和I^2C 协议计算合适的时序。表6.16-3列出了I^2C 波特率和PCLK之间的关系，表示一个I^2C时钟周期包含多少个PCLKs。在设计中，设置和保持时间配置甚至能设置一些极限值，但是用户必须按照 I^2C 协议标准。

I²C波特率 PCLK	100k	200k	400k	800k	1200k
12MHz	120	60	30	15	10
24MHz	240	120	60	30	20
48MHz	480	240	120	60	40
72MHz	720	360	180	90	60

表 6.16-3 波特率和 PCLK 之间的关系

设置时间错误调整例子，假设一个SCL周期包含5 PCLKs 并且设置 STCTL [5:0] (I2C_TMCTL[8:0])为 3，延时3个PCLKs为设置时间配置，设置时间配置限制：ST_{limit} = (I2C_CLKDIV[7:0]+1) X 2 – 6。

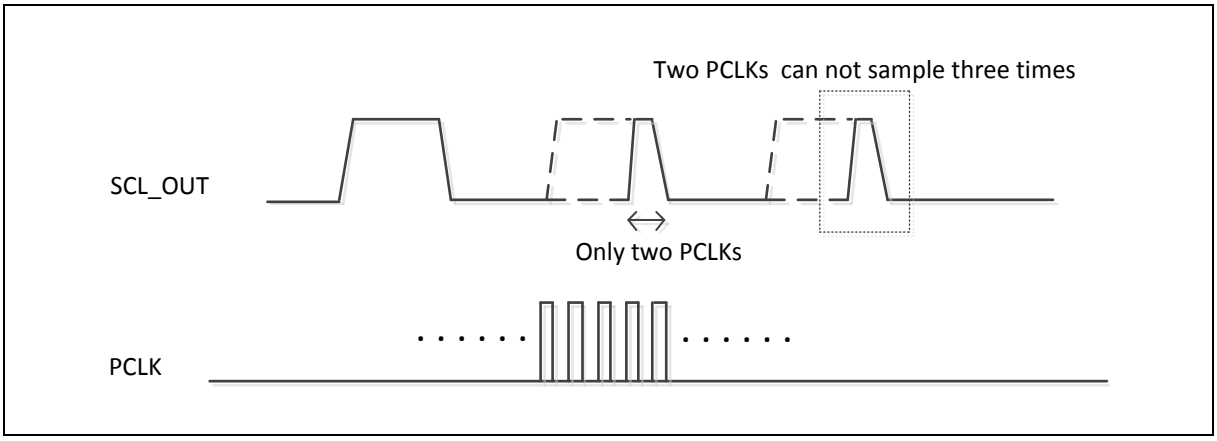


图 6.16-21 设置时间错误调整

为了保持时间错误调整例子, 我们使用 I²C 波特率 = 1200k和PCLK = 72MHz，SCL 高电平/低电平占空比 = 60 PCLK。当我们设置HTCTL (I2C_TMCTL[24:16])为61和 STCTL (I2C_TMCTL[8:0]) 为 0，然后 SDA 输出延时将超过 SCL 高电平的时间促使总线错误。保持时间设置限值：HT_{limit} = (I2C_CLKDIV[7:0]+1) X 2 – 9。

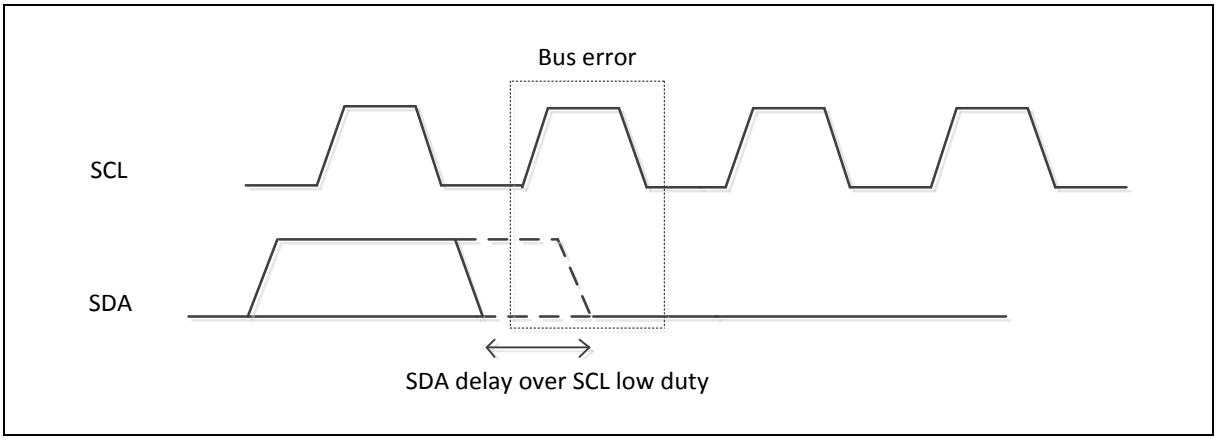


图 6.16-22 保持时间错误调整

6.16.5.5 I²C 协议寄存器

通过下列15个特殊功能寄存器来控制I²C端口：I2C_CTL0 (控制寄存器)、I2C_STATUS0 (状态寄存器)

、I2C_DAT (数据寄存器)、I2C_ADDRn (地址寄存器, n=0~3)、I2C_ADDRMSKn (地址掩码寄存器, n=0~3)、I2C_CLKDIV (时钟除频器)、I2C_TOCTL (超时控制寄存器)、I2C_WKCTL(唤醒控制寄存器)和 I2C_WKSTS(唤醒状态寄存器)。

地址寄存器(I2C_ADDR)

I^{2}C端口内建4个从机地址寄存器I2C_ADDRn (n=0~3)。当I2C作为主机时，这四个寄存器的内容不相关。在从机模式下，位字段ADDR(I2C_ADDRn [7:1]) 必须装载芯片自己的从机地址。当I2C_ADDRn地址与接收到的从机地址匹配时I^{2}C硬件会应答。

I2C 端口支持“广播呼叫”功能。当GC位(I2C_ADDR0 [0]) 被设置，I2C端口硬件将应答广播呼叫地址(00H)。清GC位可禁用“广播呼叫”功能。

当GC位被置位且I2C处于从机模式时，主机发出广播呼叫地址到I2C总线后，从机可以通过地址00H接收广播呼叫地址，然后它将在广播呼叫GC模式的状态下工作。

从机地址掩码寄存器(I2C_ADDRMSK)

I^{2}C总线控制器带有四个地址掩码寄存器I2C_ADDRMSKn (n=0~3) 支持多地址识别。当在地址掩码寄存器中的位被置1，意味着收到的对应地址位是不比较的（"无关的"）。如果位写0则收到相应地址位应跟地址寄存器对应位完全一样才算匹配成功。

数据寄存器(I2C_DAT)

该寄存器包含一个准备发送或刚接收到的一个字节的串行数据。当不在字节移位处理过程时，CPU 可以直接读写访问I2C_DAT[7:0]。当I2C处于一个确定的状态并且串行中断标志(SI) 被置1，I2C_DAT[7:0]中的数据保持稳定。当数据被移出，总线上的数据同时被移入。I2C_DAT [7:0]的内容总是总线上传递的最后一个字节。

应答位由I^{2}C硬件控制，不能被CPU访问。串行数据在SCL线上串行时钟脉冲的上升沿移入I2C_DAT [7:0] 。当一个字节被移入I2C_DAT [7:0]，则I2C_DAT [7:0] 中的串行数据是可用的，应答位(ACK或NACK) 在第9个时钟脉冲由控制逻辑返回。为了在发送数据时监控总线的状态，当发送I2C_DAT [7:0]到总线时，总线数据将同时被移入I2C_DAT[7:0]。在发送数据过程中，串行数据在SCL时钟脉冲的下降沿从I2C_DAT[7:0] 移出，在SCL时钟脉冲的上升沿数据移入I2C_DAT [7:0]。图6.16-23为I^{2}C数据移位方向。

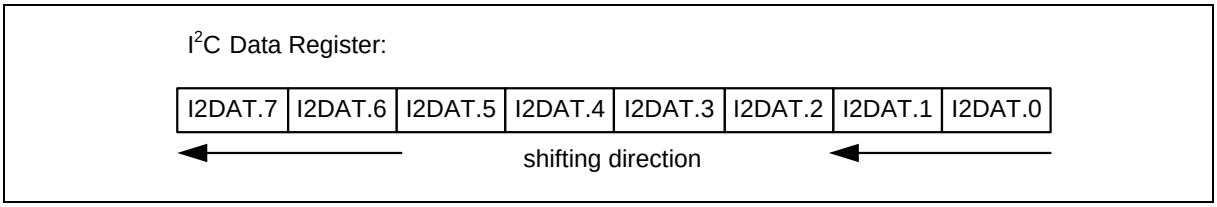


图 6.16-23 I^{2}C 数据移位方向

控制寄存器(I2C_CTL0)

CPU可以直接读或写寄存器I2C_CTL0 [7:0]，通过设置I2CEN (I2C_CTL0 [6])为1使能I^{2}C端口，内部状态由I2C_CT0L和I^{2}C逻辑硬件控制。

有两个位会受硬件影响：当I2C硬件请求中断时SI被置位，当总线上产生停止信号时，STO位被清零，当I2CEN=0时，STO也会被清零。

一旦有新的状态码产生并存储在I2C_STATUS0，I^{2}C中断标志位SI(I2C_CTL0 [3])将被自动置位。如果此时使能中断INTEN (I2C_CTL0 [7])位被置位，I^{2}C中断将产生。I2C_STATUS0[7:0]用来存储内部状态码，SI被软件清除前内容一直保持。

状态寄存器(I2C_STATUS0)

I2C_STATUS0 [7:0] 是一个8位只读寄存器。I2C_STATUS [7:0]共有26个可能的状态码。所有状态码由表6.16-.列出。当I2C_STATUS [7:0] 的值为0xF8时，没有串行中断请求。所有其他的I2C_STATUS [7:0]的值对应I^{2}C的状态。当进入其中任一状态时，就会产生状态中断请求(SI=1)。在SI被硬件置位或SI被软件复位一个PCLK周期后，有效状态码在I2C_STATUS0 [7:0] 中被更新。

此外，状态码0x00表示总线错误，这现象是I^{2}C格式帧中起始或停止条件发生在不正确的位置。总线错误会发生在一个地址字节、一个数据字节或一个应答位的串行传输中。从总线错误恢复时STO应被置位，SI应被清除，进入无地址从机模式，然后STO清除释放总线并等待下一次传输。出现总线错误操作期间I^{2}C总线不能识别停止条件。

主机模式		从机模式	
STATUS	描述	STATUS	描述
0x08[1]	主机起始	0xA0	从机发送重新起始或停止
0x10[1]	主机重新起始	0xA8[1]	从机发送地址ACK
0x18[1]	主机发送地址ACK	0xB8[1]	从机发送数据ACK
0x20	主机发送地址NACK	0xC0	从机发送数据NACK
0x28[1]	主机发送数据ACK	0xC8[1]	从机发送最后数据ACK
0x30	主机发送数据NACK	0x60[1]	从机接收地址 ACK
0x38	主机仲裁丢失	0x68[1]	从机接收仲裁丢失
0x40[1]	主机接收地址ACK	0x80[1]	从机接收数据 ACK
0x48	主机接收地址NACK	0x88	从机接收数据NACK
0x50[1]	主机接收数据ACK	0x70[1]	广播模式地址ACK
0x58	主机接收数据NACK	0x78[1]	广播模式仲裁丢失
0x00	总线错误	0x90[1]	广播模式数据 ACK
		0x98	广播模式数据 NACK
		0xB0[1]	地址传输仲裁丢失
0xF0	如果BMDEN =1和ACKMEN位使能，在从机接收条件下， I2C_STATUS0固定为0xF0。		
0xF8	总线释放 **注：**主/从模式的“0xF8”状态，都不会产生中断。 **注 [1]：**PDMA模式没有中断。		

表 6.16-4 I^{2}C 状态码描述

时钟除频寄存器(I2C_CLKDIV)

当I^{2}C在主机模式下，I^{2}C数据的波特率由DIVIDER(I2C_CLKDIV [7:0])寄存器设定；在从机模式下时不需要。在从机模式下，I^{2}C将自动与主机I^{2}C设备时钟频率同步。在从机模式，系统时钟频率必须大于I2C总线最大时钟频率的20倍。

I^{2}C数据波特率的设定：I^{2}C数据波特率= (系统时钟) / (4x ((I2C_CLKDIV[7:0] +1))。如果系统时钟 = 16 MHz，I2C_CLKDIV [7:0] = 40 (0x28)，那么I^{2}C数据波特率= 16 MHz/ (4x (40 +1)) = 97.5 Kbits/sec。

超时控制寄存器(I2C_TOCTL)

系统提供一个14位超时计数器来处理当I2C总线锁死时的情况。当超时计数器计数功能使能后，计数器

开始计数直至溢出(TOIF=1)并向CPU产生I2C中断或者清除TOCEN为0停止计数。当超时计数器使能，对SI标志置高会使计数器复位，再对SI清零会重新开始计数。如果I2C总线锁死，会使I2C_STATUS0及SI标志不再更新，该14位超时计数器会发生溢出并且产生I2C中断通知CPU。置TOCDIV4 (I2C_TOCTL[1])为1超时计数周期扩大4倍。参照图6.16-2414位超时计数器，用户可以写1 清TOIF为0。

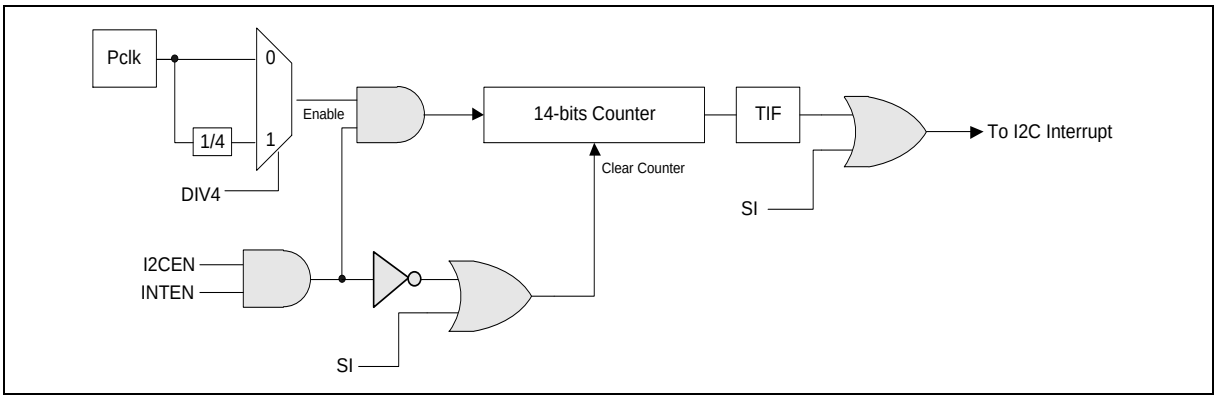


图 6.16-24 I^{2}C 超时计数模块框图

唤醒控制寄存器(I2C_WKCTL)

当芯片进入掉电模式并且设置WKEN(I2C_WKCTL [0])为1，其他I^{2}C主机可以通过寻址I^{2}C设备(从机设备)唤醒我们的芯片，进入休眠模式之前，用户必须设置好相关寄存器。在掉电模式下，地址匹配帧应答位完成。当设备地址匹配成功并且应答位完成，控制器将延长SCL低电平时间，然后I^{2}C控制器继续运行。如果NHDBUSEN (I2C_WKCTL [7])被置1，控制器将不再延长SCL低电平时间。注意：当控制器将不再延长SCL低电平时间，传输或接收数据将立即执行。如果数据传输或接收完成，当事件未被清除，用户必须复位I^{2}C控制器并且再执行原来的操作。

唤醒状态寄存器((I2C_WKSTS)

当系统被其他的I^{2}C主机设备唤醒时，WKIF被置位表示该事件发生，用户需要写“1”来清除此位。

当芯片被设备地址寄存器(I2C_ADDRn)的某个地址匹配唤醒，用户核对WKAKDONE (I2C_WKSTS [1])位是否被置1，来确认这地址完成应答。在掉电模式下，WKAKDONE位表示地址完成应答。当地址被匹配设备的从地址和应答完成，控制器将延长SCL低电平时间。SCL延长直到WKAKDONE被用户清除。如果SCL的频率速度低并且系统已经被地址匹配帧唤醒，用户核对WKAKDONE位来确定这个帧是否传输完成然后进入唤醒程序。注意用户不能通过清零WKAKDONE位来清WKIF位。

控制器发送地址之前，WRSTSWK (I2C_WKSTS [2])位记录读/写命令. 用户能通过读这位的状态来去准备下一个传输数据(WRSTSWK = 0) 或系统通过地址匹配帧唤醒后，等待存储输入数据的时间(WRSTSWK = 1) 。注意当写“1”到WKAKDONE (I2C_WKSTS [1])位，WRSTSWK (I2C_WKSTS [2])位被清零。

系统当通过其他I^{2}C主设备被唤醒, WKIF被置位表示有事件。用户需要写“1”清零这位。

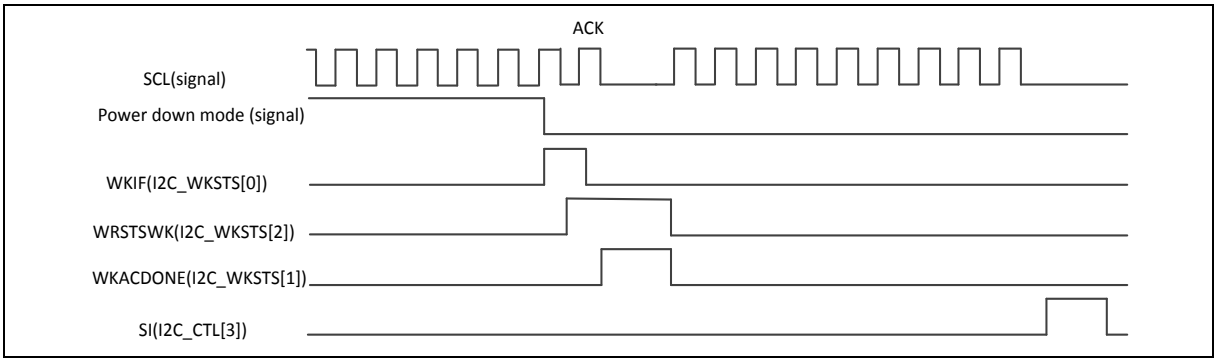


图 6.16-25 I²C 唤醒相关信号波形

I²C 控制寄存器 1 (I2C_CTL1)

对于PDMA功能, TXPDMAEN (I2C_CTL1 [0]) 和 RXPDMAEN (I2C_CTL1 [1]) 可以被设置操作，也可以设置 PDMARST (I2C_CTL1 [2]) 去复位PDMA控制器。

I²C 时序配置控制寄存器((I2C_TMCTL)

为了配置设置/保持时间，根据实际需求来设置寄存器HTCTL (I2C_TMCTL[24:16]) 和STCTL (I2C_TMCTL[8:0])。

总线管理控制寄存器 (I²C_BUSCTL)

SM总线管理控制事件在该寄存器定义。它包括手动控制应答 (ACKMEN (I²C_BUSCTL[0]))，包错误检测使能(PECEN (I²C_BUSCTL[1]))，设备默认地址使能 (BMDEN(I²C_BUSCTL[2])) 或者主机功能使能 (BMHEN (I²C_BUSCTL[3]))。警报和挂起功能可以由ALERTEN (I²C_BUSCTL[4])，SCTLOSTS (I²C_BUSCTL[5]))和SCTLOEN (I²C_BUSCTL[6])来设置。

系统总线管理允许通过BUSEN(I²CBUSCTL[7])位进行控制。BUSTOUT(I²CBUSCTL[9])用于计算有效总线上时钟低电平时间超时和总线空闲中的空闲周期。

可以通过控制PECTXEN(I²C_BUSCTL[8])位来传送或者接收计算PEC(如果PECEN置位)的值。

该寄存器有个特别的位ACKM9SI (I²C_BUSCTL[11])。当ACKMEN使能时，在第8个时钟输入时，有SI中断产生，用户此时可以读取数据和状态寄存器。当SI中断清除时第8个时钟总线被释放，此时，如果ACKM9SI位设置为1，在第9个时钟周期时会有另一个SI中断事件发生，这样就知道传输帧完成后的总线状态。

对于PEC控制流程则是设置PECDIEN (I²C_BUSCTL[13])、BCDIEN (I²C_BUSCTL[12]) 或 PECCLR (I²C_BUSCTL[10])。

I²C 总线管理定时器控制寄存器(I2C_BUSTCTL)

设置TORSTEN(I²C_BUSTCTL[4])，CLKTOIEN(I²C_BUSTCTL[3])，BUSTOIEN (I²C_BUSTCTL[2]), CLKTOEN (I²C_BUSTCTL[1])和BUSTOEN (I²C_BUSTCTL[0])来配置总线超时或时钟低定时器超时。

I²C 总线管理状态寄存器(I2C_BUSSTS)

对于PEC控制流程，需要监控PECDONE (I²C_BUSSTS[7])、BCDONE (I²C_BUSSTS[1])或PECERR (I²C_BUSSTS[2])。

监控SCTLDIN (I²C_BUSSTS[4])来观察SUSCON输入状态。

I²C 字节数寄存器 (I2C_PKTSIZE)

当 PECEN 位 (I²C_BUSCTL[1]) 使能时，I²C 控制器将计算总线上数据的 PEC 值。PLDSIZE

(I^2C_PKTSIZE[8:0])用来定义总线上数据需要传输的个数。当计数值达到PLDSIZE时，且PECTXEN (I^2C_BUSCTL[8])有设置，最后的PEC值将会发送或者自动接收。

I^2C PEC 值寄存器(I2C_PKTCRC)

该寄存器为I^2C总线上传输数据计算出来的PECCRC (I^2C_PKTCRC[7:0])值，详细信息由SM总线的PEC部分说明。

I^2C 总线管理定时器和 I^2C 时钟低定时器寄存器(I2C_BUSTOUT/ I2C_CLKTOUT)

这两个寄存器的定义在SM总线的超时章节描述。

6.16.5.6 EEPROM随机读例子

当使用I^2C从EEPROM读取数据, 通过下面的步骤来配置I^2C相关寄存器：

1. 设置 I2C0多功能管脚设置成SCL和SDA管脚，多功能引脚配置参考基本配置；
2. 使能 I2C0 APB 时钟，时钟配置参考基本配置；
3. 设置 I2C0RST=1来复位I2C0控制器然后设置I2C0控制器到正常操作模式。复位控制器配置参考基本配置；
4. 通过设置寄存器I2C_CTL0的I2CEN=1使能 I2C0 控制器；
5. 通过在I2C_CLKDIV寄存器写I2C0时钟分频值；
6. 在系统 “NVIC” 控制寄存器来使能系统I2C0 IRQ；
7. 设置寄存器I2C_CTL0的INTEN为1使能I2C0中断；
8. 设置I^2C0地址寄存器“I2C_ADDR0~I2C_ADDR3”。

随机读操作是访问EEPROM的其中一种方法。这个方法是允许主机访问EEPROM的任何一个地址。图6.16-26展示对EEPROM随机读取操作。

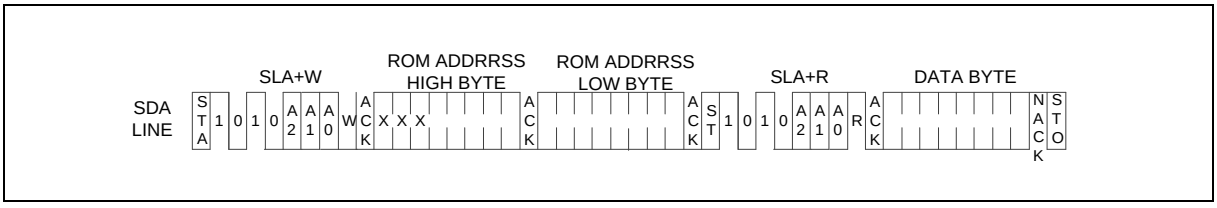


图 6.16-26 EEPROM 随机读

图6.16-27展示怎样使用I2C控制器执行EEPROM随机读取操作协议。

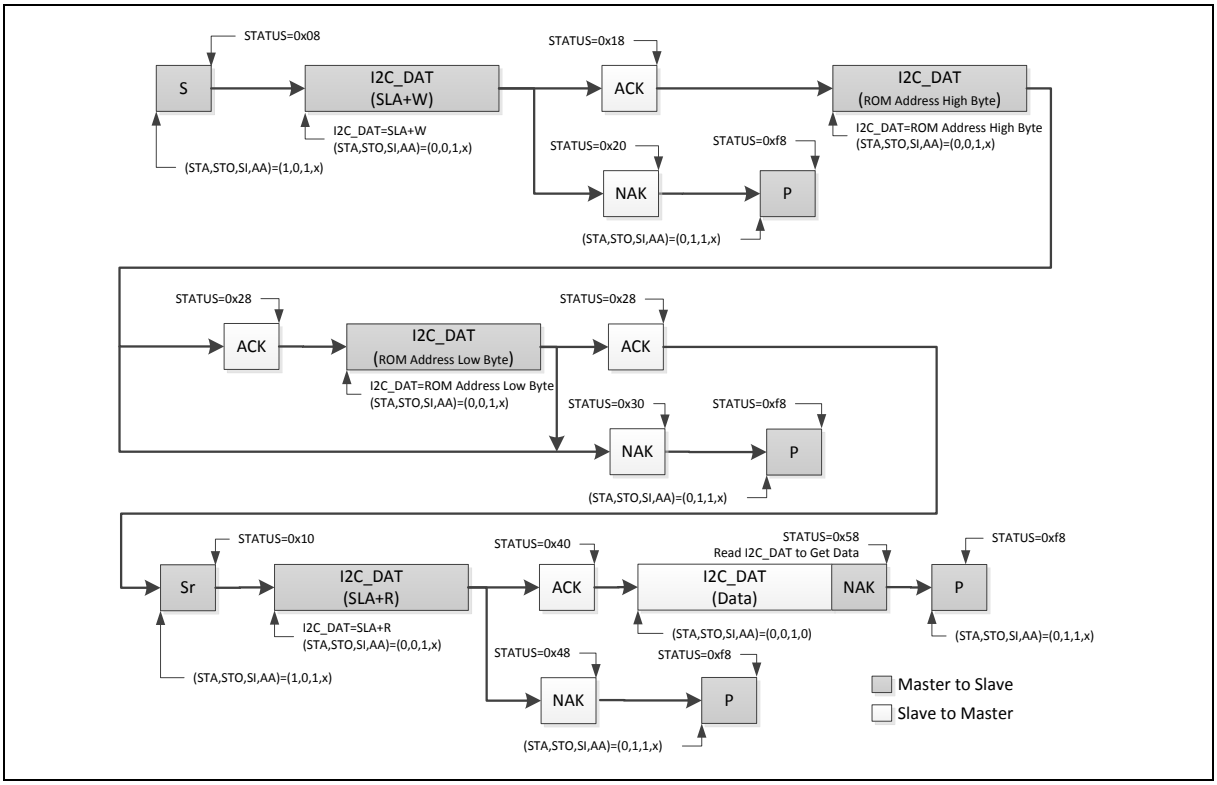


图 6.16-27 随机读的协议

I²C控制器作为主机发送起始信号到总线，然后发送SLA+W (从机地址 + 写位)到EERPOM，跟着由两个字节数据地址来设置EEPROM被读的地址。最后，重复起始信号后接着发送SLA+R从EEPROM去读取数据。

6.16.6 寄存器映射

R:只读, **W:** 只写, **R/W:**读/写

寄存器	偏移	R/W	描述	复位值
I²C基地址: **I2Cn_BA = 0x4008_0000 + (0x1000 *n)** **n= 0,1**				
I2C_CTL0	I2Cn_BA+0x00	R/W	I²C控制寄存器0	0x0000_0000
I2C_ADDR0	I2Cn_BA+0x04	R/W	I²C从机地址寄存器0	0x0000_0000
I2C_DAT	I2Cn_BA+0x08	R/W	I²C数据寄存器	0x0000_0000
I2C_STATUS0	I2Cn_BA+0x0C	R	I²C状态寄存器0	0x0000_00F8
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I²C时钟除频寄存器	0x0000_0000
I2C_TOCTL	I2Cn_BA+0x14	R/W	I²C超时控制寄存器	0x0000_0000
I2C_ADDR1	I2Cn_BA+0x18	R/W	I²C从机地址寄存器1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I²C从机地址寄存器2	0x0000_0000

I2C_ADDR3	I2Cn_BA+0x20	R/W	I^{2}C从机地址寄存器3	0x0000_0000
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I^{2}C从机地址掩码寄存器0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I^{2}C从机地址掩码寄存器1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I^{2}C从机地址掩码寄存器2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I^{2}C从机地址掩码寄存器3	0x0000_0000
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I^{2}C唤醒控制寄存器	0x0000_0000
I2C_WKSTS	I2Cn_BA+0x40	R/W	I^{2}C唤醒状态寄存器	0x0000_0000
I2C_CTL1	I2Cn_BA+0x44	R/W	I^{2}C控制寄存器1	0x0000_0000
I2C_STATUS1	I2Cn_BA+0x48	R/W	I^{2}C状态寄存器1	0x0000_0000
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I2C 时序配置控制寄存器	0x0000_0000
I2C_BUSCTL	I2Cn_BA+0x50	R/W	I^{2}C总线管理控制寄存器	0x0000_0000
I2C_BUSTCTL	I2Cn_BA+0x54	R/W	I^{2}C总线管理定时器控制寄存器	0x0000_0000
I2C_BUSSTS	I2Cn_BA+0x58	R/W	I^{2}C总线管理状态寄存器	0x0000_0000
I2C_PKTSIZE	I2Cn_BA+0x5C	R/W	I^{2}C包错误检查字节数寄存器	0x0000_0000
I2C_PKTCRC	I2Cn_BA+0x60	R	I^{2}C包错误检查字节值寄存器	0x0000_0000
I2C_BUSTOUT	I2Cn_BA+0x64	R/W	I^{2}C总线管理定时器寄存器	0x0000_0005
I2C_CLKTOUT	I2Cn_BA+0x68	R/W	I^{2}C总线管理时钟低定时器寄存器	0x0000_0005

6.16.7 寄存器描述

I²C 控制寄存器 (I2C_CTL0)

寄存器	偏移	R/W	描述	复位值
I2C_CTL0	I2Cn_BA+0x00	R/W	I²C控制寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**Reserved**	

位	描述	
[31:8]	**Reserved**	保留
[7]	**INTEN**	**使能中断** 1 = 使能 I²C 中断 0 = 禁用 I²C 中断
[6]	**I2CEN**	**I²C 控制器使能位** 设置使能 I²C 串行功能控制器。当I2CEN =1，I²C 串行功能使能。SDA和SCL对应的多功能管脚功能必须先设置为 I²C 功能。 1 = I2C控制器使能 0 = I2C控制器禁用
[5]	**STA**	**I²C起始控制位** 设置 STA 为 1，进入主机模式，如果总线处于空闲状态，I²C 硬件会送出START或重复START 条件。
[4]	**STO**	**I²C 停止控制位** 在主机模式，设置STO来传送一个 STOP 条件到总线，然后I²C控制器将会检查总线状况，如果检测到一个 STOP 状况，这个标志会被硬件自动清除。
[3]	**SI**	**I²C 中断标志** 当一个新的I²C 状态出现在寄存器 I2C_STATUS 0时，SI 标志由硬件置位，并且如果INTEN (I2C_CTL0 [7])位被置位，则产生I²C 中断请求。SI 必须由软件通过向该位写 '1' 清零。 如果在从机读模式且ACKMEN置位，SI标志在第8个时钟周期置位，此时用户确认应答位，在第9个时钟周期时，用户可以从数据缓冲器中读取数据。
[2]	**AA**	**应答控制位** 当AA=1先于地址或数据接收，在SCL线上的应答时钟脉冲期间将返回一个应答(SDA上为低电平)，有两种情况：1.) 从机正在应答主机发送的地址；2.) 接收设备正在应答发送设备发送的数据。当 AA = 0 先于地址或数据接收，则在SCL线上的应答时钟脉冲期间将返回一个

		非应答（SDA上为高电平）。
[1:0]	**Reserved**	保留

I²C数据寄存器 (I2C_DAT)

寄存器	偏移	R/W	描述	复位值
I2C_DAT	I2Cn_BA+0x08	R/W	I²C数据寄存器	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**DAT**	**I²C数据** Bit [7:0] 8 位存放I²C 串行端口8位传输/接收数据

I²C 状态寄存器 (I2C_STATUS0)

寄存器	偏移	R/W	描述	复位值
I2C_STATUS0	I2Cn_BA+0x0C	R	I²C状态寄存器0	0x0000_00F8

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**STATUS**	**I²C状态** 低3位有效位一直为0。高5位有效位包括状态码。有28种可能的状态码。当I2C_STATUS的内容为0xF8时，没有中断产生。所有其他的 I2C_STATUS0的值对应I²C 的状态。当进入其中任一状态时，就会产生状态中断请求 (SI=1)。在 SI 被硬件置位或SI 被软件复位后一个机器周期，有效状态码在 I2C_STATUS0中被更新。 此外，00H 状态表示总线错误。当在帧结构中START或STOP条件出现不正确的位置总线错误发生。不正确的位置比如是在串行传输地址字节、数据字节或应答位期间。

I²C时钟除频器(I2C_CLKDIV)

寄存器	偏移	R/W	描述	复位值
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I²C时钟除频器	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留.
[9:0]	**DIVIDER**	**I²C时钟分频位** I²C数据波特率 = (系统时钟) / (4x (I2C_CLKDIV +1)) **注：**I2C_CLKDIV的最小值是 4。

I²C超时控制器寄存器(I2C_TOCTL)

寄存器	偏移	R/W	描述	复位值
I2C_TOCTL	I2Cn_BA+0x14	R/W	I²C超时控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**TOCEN**	**TOCDIV4**	**TOIF**

位	描述	
[31:3]	**Reserved**	保留.
[2]	**TOCEN**	**超时计数器使能位** 当使能该位，则14位溢出定时器将会在SI清零后开始计数。设置SI位为高将会复位计数器，在SI位清零之后，计数器会重新开始计数。 1 = 超时计数器使能 0 = 超时计数器禁用
[1]	**TOCDIV4**	**溢出定时器输入时钟除以 4** 1 = 溢出定时器输入时钟使能 0 = 溢出定时器输入时钟禁
[0]	**TOIF**	**定时溢出标志** 当超时发生时，该位由 H/W置位，如果此时I²C的中断使能位INTEN置为1，则可引发 CPU的中断。 **注意：** 写 1 清除该位。

I²C从机地址寄存器0(ADDR0)

寄存器	偏移	R/W	描述	复位值
I2C_ADDR0	I2Cn_BA+0x04	R/W	I²C从机地址寄存器0	0x0000_0000

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

位	描述	
[31:11]	**Reserved**	保留
[7:1]	**ADDR**	**I²C地址** 主机模式下，该寄存器内容没有意义。从机模式下，高七位作为芯片本身的地址。如果地址符合，I²C 硬件将会自动应答。 **注：**软件设置为0x00，地址不能被使用。
[0]	**GC**	**广播功能** 0 = 广播功能禁用 1 = 广播功能使能

I²C从地址寄存器 (ADDRx)

寄存器	偏移	R/W	描述	复位值
I2C_ADDR1	I2Cn_BA+0x18	R/W	I²C从地址寄存器1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I²C从地址寄存器2	0x0000_0000
I2C_ADDR3	I2Cn_BA+0x20	R/W	I²C从地址寄存器3	0x0000_0000

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

位	描述	
[31:11]	**Reserved**	保留.
[7:1]	**ADDR**	**I²C 地址位** 主机模式下，该寄存器内容没有意义。从机模式下，高七位作为芯片本身的地址。如果地址符合，I²C 硬件将会自动应答。 **注：**软件设置为0x00，地址不能被使用
[0]	**Reserved**	保留

I²C从机地址掩码寄存器(ADDRMSKx)

寄存器	偏移	R/W	描述	复位值
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I²C从机地址掩码寄存器0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I²C从机地址掩码寄存器1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I²C从机地址掩码寄存器2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I²C从机地址掩码寄存器3	0x0000_0000

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

位	描述	
[31:11]	**Reserved**	保留.
[7:1]	**ADDRMSK**	**I²C 地址掩码位** 0 =掩码禁用（接收到的相应地址必须完全符合地址寄存器） 1 =掩码使能（接收到的相应地址位不予辨识） I²C总线控制器有4个地址掩码寄存器，支持多地址识别。当地址掩码寄存器的某位被置 ’1’，表示接收到的地址的相应位可忽略。如果该位被置 ‘0’，则表示接收到的地址的相应位必须和地址寄存器中的位完全一致。
[0]	**Reserved**	保留

I²C 唤醒控制寄存器 (I2C_WKCTL)

寄存器	偏移	R/W	描述	复位值
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I²C唤醒控制寄存器	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留
[7]	**NHDBUSEN**	**I²C不保持总线使能位** 0 = I²C唤醒后保持总线 1= I²C唤醒后不保持总线使能 **注意：**当WKIF事件未清除，I²C 控制器反映，可能导致错误数据传输或接收。如果数据传输或接收完成，当 WKIF 事件未清除，用户必须复位 I2C控制器并且再执行原来的操作。
[6:1]	**Reserved**	保留
[0]	**WKEN**	**I²C唤醒功能使能位** 0 = I²C 唤醒功能禁止 1 = I²C 唤醒功能使能

I²C唤醒状态寄存器(I2C_WKSTS)

寄存器	偏移	R/W	描述	复位值
I2C_WKSTS	I2Cn_BA+0x40	R/W	I²C唤醒状态寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留.
[2]	**WRSTSWK**	**地址唤醒帧读/写状态位（只读）** 0 =地址匹配唤醒帧写命令 1 = 地址匹配唤醒帧读命令 **注：**当软件写1清零 WKAKDONE位(I2C_WKSTS[1])，这位也被清零。
[1]	**WKAKDONE**	**地址唤醒帧应答位完成** 0 = 地址唤醒帧应答位没完成 1 = 在掉电模式地址唤醒帧应答位已完成 **注：**在清零 WKIF位时，这位不能被清零，该位软件写1清0。
[0]	**WKIF**	**I2C唤醒标志** 当芯片从掉电模式下由I2C唤醒时，该位设置为1，该位软件写1清0。

I²C控制寄存器1 (I2C_CTL1)

寄存器	偏移	R/W	描述	复位值
I2C_CTL1	I2Cn_BA+0x44	R/W	I²C控制寄存器1	0x0000_0000

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

位	描述	
[31:7]	**Reserved**	保留.
[8]	**PDMASTR**	**PDMA 延时位** 0 = PDMA传输完成后I²C自动发送STOP信号(仅主机发送) 1 =如果 SI 未被清零，传输完成后，I²C SCL 电平时间自动被延长 (仅主机发送)
[7:3]	**Reserved**	保留
[2]	**PDMARST**	**PDMA 复位** 0 = 无效 1 = 复位 PDMA 控制器. 这位将被自动清零
[1]	**RXPDMAEN**	**PDMA 接收通道使能** 0 = 接收PDMA 功能禁止 1 = 接收PDMA 功能使能
[0]	**TXPDMAEN**	**PDMA 传输通道使能** 0 = 传输PDMA功能禁止 1 = 传输PDMA功能使能

I²C状态寄存器1 (I2C_STATUS1)

寄存器	偏移	R/W	描述	复位值
I2C_STATUS1	I2Cn_BA+0x48	R/W	I²C状态寄存器1	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留.
[8]	**ONBUSY**	**总线忙（只读）** 表明在总线上有通信. 当检测到开始信号自动置位。当检测到停止信号自动清零。 0 = 总线空闲IDLE (SCLK 和 SDA 都是高). 1 = 总线忙
[7:0]	**Reserved**	保留

I²C 时序配置控制寄存器 (I2C_TMCTL)

寄存器	偏移	R/W	描述	复位值
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I²C 时序配置控制寄存器	0x0000_0000

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

位	描述	
[31:25]	**Reserved**	保留
[24:16]	**HTCTL**	**保持时序配置寄存器** 在传输模式，这位域用于在SCL下降沿和SDA上升沿之间产生延时时间。 延时保持时间相当于外设时钟的个数= HTCTL x PCLK
[15:9]	**Reserved**	保留
[8:0]	**STCTL**	**建立时序配置寄存器** 在传输模式，这位域用于在SDA下降沿SCL升沿之间产生延时时间。 延时保持时间相当于外围时钟的个数= STCTL x PCLK **注：**建立时间需要使SCL输出大于3个PCLKs。

I²C总线管理控制寄存器(I2C_BUSCTL)

寄存器	偏移	R/W	描述	复位值
I2C_BUSCTL	I2Cn_BA+0x50	R/W	I²C总线管理控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**PECDIEN**	**BCDIEN**	**ACKM9SI**	**PECCLR**	**TIDLE**	**PECTXEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BUSEN	**SCTLOEN**	**SCTLOSTS**	**ALERTEN**	**BMHEN**	**BMDEN**	**PECEN**	**ACKMEN**

位	描述	
[31:14]	**Reserved**	保留
[13]	**PECDIEN**	**包错误检测字节传输完成中断使能位** 0 = 禁用PEC传输完成中断 1 = 使能PEC传输完成中断 **注：**该位在 PECEN =1时使用。
[12]	**BCDIEN**	**包错误检测字节计数完成中断使能位** 0 = 禁用计数完成中断 1 = 使能计数完成中断 **注：**该位在 PECEN =1时使用。
[11]	**ACKM9SI**	**手工应答使能额外SI中断位** 0 = 当BUSEN =1 和 ACKMEN =1时，在第9个时钟周期没有SI中断 1 = 当BUSEN =1 和 ACKMEN =1时，在第9个时钟周期有SI中断
[10]	**PECCLR**	**在重新起始时清除PEC** 当PECEN设置为1时，PEC的计算开始。当检测到STA或STO位时清除。当发生REPEAT START时，PECCLR位用来使能清除PEC计算。 0 = 发生“Repeat Start”功能时禁止清除PEC计算 1 = 发生“Repeat Start”功能时使能清除PEC计算
[9]	**TIDLE**	**空闲状态时间检查** BUSTOUT用来计算有效总线上时钟低的时间和空闲总线上空闲的周期。该位用来定义使能哪个条件被使能。 0 = BUSTOUT用来计算有效总线上时钟低的时间。 1 = BUSTOUT用来计算空闲总线上空闲的周期。 **注：**BUSY (I2C_BUSSTS[0])位表示当前总线的状态。
[8]	**PECTXEN**	**包错误检查字节发送/接收使能位**

		0 = 没有PEC传送 1 = 请求PEC传输 **注：**当ACKMEN =0时，在从机模式该位无影响。
[7]	**BUSEN**	**总线使能位** 0 = 禁止系统管理功能 1 = 使能系统管理功能 **注：**当该位使能，内部14位计数器用来计算时钟低条件的事件时间。
[6]	**SCTLOEN**	**挂起或控制管脚输出使能位** 0 = SUSCON管脚输入 1 = 输出使能SUSCON管脚有效
[5]	**SCTLOSTS**	**挂起/控制数据输出状态** 0 = SUSCON管脚输出为低 1 = SUSCON管脚输出为高
[4]	**ALERTEN**	**总线管脚警报使能位** 从模式(BMHEN =0) 0 = 释放BM_ALERT管脚为高而且禁止警报响应地址头：如果BMDEN和ACKMEN都使能，NACK后接着就是0001100x。 1 = 驱动BM_ALERT管脚为低而且使能警报响应地址头：如果BMDEN和ACKMEN都使能，ACK后接着就是0001100x。 主模式(BMHEN =1) 0 = BM_ALERT管脚不支持 1 = BM_ALERT管脚支持
[3]	**BMHEN**	**总线管理主机使能位** 0 = 禁止主机功能 1 = 使能主机功能
[2]	**BMDEN**	**总线管理设备默认地址使能位** 0 = 禁止设备默认地址。当检测地址0’b1100001x且BMDEN和ACKMEN都使能，设备响应NACK。 1 = 使能设备默认地址。当检测地址0’b1100001x且BMDEN和ACKMEN都使能，设备响应ACK。
[1]	**PECEN**	**包错误检查计算使能位** 0 = 禁止包错误检查计算 1 = 使能包错误检查计算 **注：**当I^2C进入掉电模式，在唤醒之后如果需要PEC计算该位应该使能。
[0]	**ACKMEN**	**手动应答控制使能位** 为了从机接收包括命令和数据应许ACK控制，从机字节控制模式必须通过设置ACKMEN位使能。 0 = 禁止从机字节控制 1 = 使能从机字节控制。用户通过接收的数据在第9位可以响应ACK或NACK。当字节被接收时，第8个和第9个SCLK脉冲之间的SCLK低信号拉长。 **注：**如果BMDEN =1且该位使能，在从接收模式下I2C_STATUS的信息将固定在0xF0。

I²C总线管理定时器控制寄存器(I2C_BUSTCTL)

寄存器	偏移	R/W	描述	复位值
I2C_BUSTCTL	I2Cn_BA+0x54	R/W	I²C总线管理定时器控制寄存器	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[4]	**TORSTEN**	**超时复位使能位** 0 = 禁用I²C状态机复位 1 = 使能I²C状态机复位(时钟和数据线将被放为高)
[3]	**CLKTOIEN**	**延长时钟超时中断使能位** 0 = 禁止延长时间中断 1 = 使能延长时间中断
[2]	**BUSTOIEN**	**超时中断使能位** BUSY =1 0 = 禁止SCLK低超时中断 1 = 使能SCLK低超时中断 BUSY =0 0 = 禁止总线空闲超时中断 1 = 使能总线空闲超时中断
[1]	**CLKTOEN**	**累积时钟低超时使能位** 0 = 禁止累积时钟低电平超时检测 1 = 使能累积时钟低电平超时检测 主机，计算从START到ACK周期 从机，计算从START到STOP周期
[0]	**BUSTOEN**	**总线超时使能位** 0 = 禁止总线时钟低电平超时检测 1 = 使能总线时钟低电平超时检测(总线时钟为低电平大于T时间 (BIDLE=0) 或者高电平大与T时间 (BIDLE =1)

I²C总线管理状态寄存器(I2C_BUSSTS)

寄存器	偏移	R/W	描述	复位值
I2C_BUSSTS	I2Cn_BA+0x58	R/W	I²C总线管理状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PECDONE	**CLKTO**	**BUSTO**	**SCTLDIN**	**ALERT**	**PECERR**	**BCDONE**	**BUSY**

位	描述	
[31:6]	**Reserved**	保留
[7]	**PECDONE**	**PEC字节发送/接收完成** 0 = PECEN置位时表明PEC发送/接收还未完成 1 = PECEN置位时表明PEC发送/接收已完成 **注：**该位软件写1清0。
[6]	**CLKTO**	**时钟累积低电平超时状态** 0 = 未发生时钟累积低电平超时 1 = 有发生时钟累积低电平超时 **注：**该位软件写1清0。
[5]	**BUSTO**	**总线超时状态** 0 = 未发生超时或未发生外部时钟超时 1 = 有发生超时或有发生外部时钟超时 在总线忙时该位指示时钟低电平总数发生超时事件，别的方式就指示总线空闲超时事件发生。 **注：**该位软件写1清0。
[4]	**SCTLDIN**	**总线挂起或控制信号输入状态(只读)** 0 = SUSCON管脚输入状态为0 1 = SUSCON管脚输入状态为1
[3]	**ALERT**	**SMBus警报状态** 从模式(BMHEN =0) 0 = 指示SMALERT管脚状态为低 1 = 指示SMALERT管脚状态为高 主模式(BMHEN =1) 0 = 无SMBALERT事件

		1 = 当BMHEN = 1(SMBus主机配置)和ALERTEN=1时，指示在SMALERT管脚上有检测到SMBALERT事件(下降沿)发生 **注：** 1. SMALERT管脚是开漏管脚，使用时需要上拉电阻；2. 该位软件写1清0。
[2]	**PECERR**	**接收PEC错误状态位** 0 = 指示PEC值等于接收到PEC数据包 1 = 指示PEC值与接收到PEC数据包不匹配 **注：** 该位软件写1清0。
[1]	**BCDONE**	**字节数传送/接收完成状态位** 0 = 当PECEN置位时，传送/接收还没有完成 1 = 当PECEN置位时，传送/接收已经完成 **注：** 该位软件写1清0。
[0]	**BUSY**	**总线忙状态位(只读)** 指示总线上通信的进程。当检测到起始条件时该位由硬件置位。当检测到停止条件时由硬件清除。 0 = 总线处于空闲状态(SCLK 和 SDA 都为高) 1 = 总线处于忙状态

I²C包错误检查字节数寄存器(I2C_PKTSIZE)

寄存器	偏移	R/W	描述	复位值
I2C_PKTSIZE	I2Cn_BA+0x5C	R/W	I²C包错误检查字节数寄存器	0x0000_0000

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

位	描述	
[31:9]	**Reserved**	保留
[8:0]	**PLDSIZE**	**传输字节个数** 当PECEN置位时，指示一次事务中传送或接收数据的字节个数。最大传送或接受字节个数为256个字节。 **注：**字节计数包括地址、命令码和数据帧

I²C包错误检查字节值寄存器(I2C_PKTCRC)

寄存器	偏移	R/W	描述	复位值
I2C_PKTCRC	I2Cn_BA+0x60	R	I²C包错误检查字节值寄存器	0x0000_0000

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**PECCRC**	**包错误价差字节值** 该字节指示在传送或接收字节数据后，包错误检查内容通过使用C(x) = X8 + X2 + X + 1计数。该位域只读。

I²C总线管理定时器寄存器(I2C_BUSTOUT)

寄存器	偏移	R/W	描述	复位值
I2C_BUSTOUT	I2Cn_BA+0x64	R/W	I²C总线管理定时器寄存器	0x0000_0005

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**BUSTO**	**总线管理超时时间值** 该位域指示总线空闲或时钟低电平超时时间值 **注：**如果用户想修改BUSTOUT的值，先使能BUSEN(I2C_BUSCTL[7])，然后把TORSTEN (I2C_BUSTCTL[4])位必须设置为1，接着再清零。

I²C总线管理时钟低定时器寄存器(I2C_CLKTOUT)

寄存器	偏移	R/W	描述	复位值
I2C_CLKTOUT	I2Cn_BA+0x68	R/W	I²C总线管理时钟低定时器寄存器	0x0000_0005

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

位	描述	
[31:8]	**Reserved**	保留
[7:0]	**CLKTO**	**总线时钟低电平时间** 该位域用来配置累积时钟延长时间 **注：**如果用户想要修改CLKLTOUT的值，先使能BUSEN，然后把TORSTEN需设置为1，接着再清零。

6.17 USCI –通用串行控制接口控制器(USCI)

6.17.1 概述

通用串行控制接口(USCI)是一种灵活的接口模块，它集中了几种串行通信协议。用户可以配置该控制器作为UART、SPI或是I^2C功能协议。

6.17.2 特性

控制器可以依据应用需求进行单独的配置。主要支持以下几种协议：

- UART
- SPI
- I^2C

6.17.3 框图

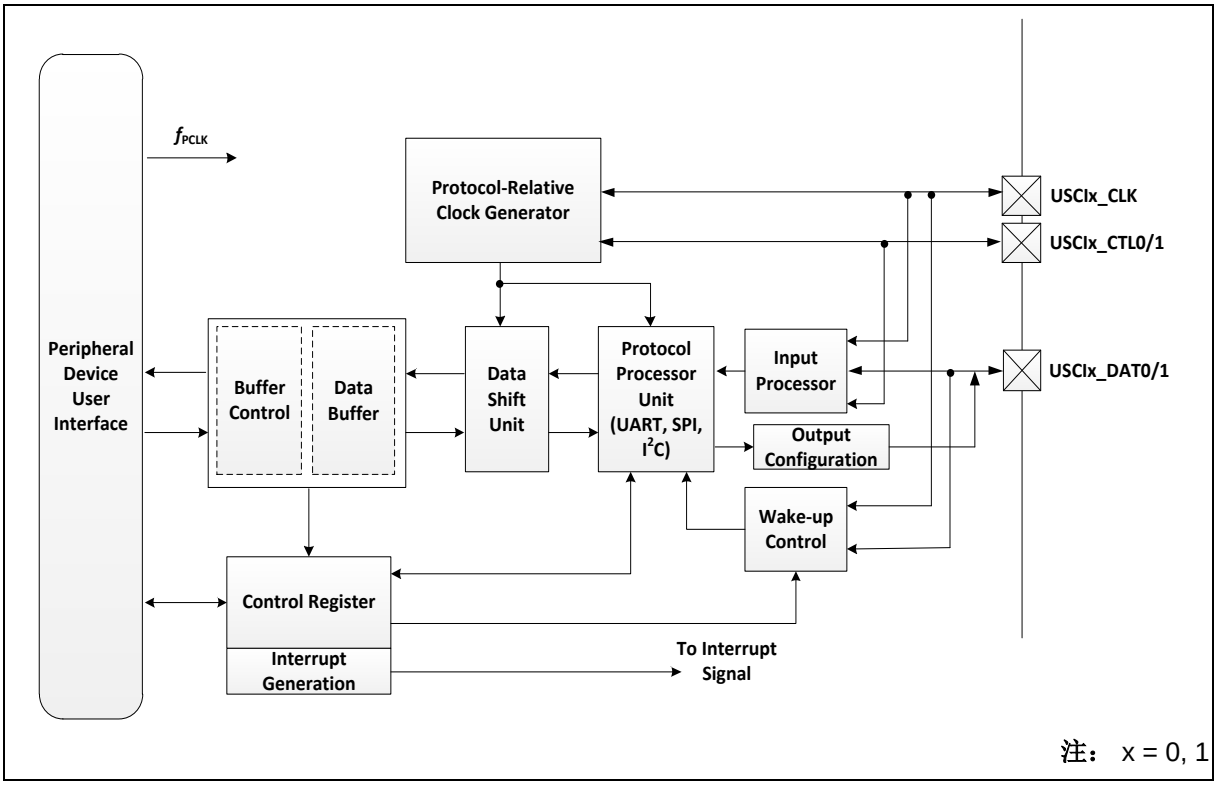


图 6.17-1USCI 框图

6.17.4 功能描述

USCI控制器的结构如图6.17-1USCI。输入信号被输入处理器处理。数据缓冲和数据移位单元支持数据传输。每一个具体协议功能被协议处理单元处理。时序和具体协议的时间事件控制信号被协议相关时钟发生器处理。所有具体协议事件被中断产生单元处理。具体协议唤醒功能被唤醒控制器单元处理。

USCI具有三种协议分别是: UART、SPI、和 I^2C，可以通过FUNMODE (USCI_CTL [2:0])选择。在改变协议前FUNMODE必须设置为0。

6.17.4.1 I/O 处理器

输入信号

所有输入阶段提供类似的特性设置，可以用于所有协议。

表6.17-1为每种选择的协议列出了相关的输入信号。每种输入信号由输入处理器处理加工。比如信号反相选择控制或数字输入滤波。

选择协议		UART	SPI	I^2C
串行总线时钟输入	USCIx_CLK	-	SPI_CLK	SCL
控制输入	USCIx_CTL0	nCTS	SPI_SS	-
	USCIx_CTL1	-	-	-
数据输入	USCIx_DAT0	RX	SPI_MOSI_0	SDA
	USCIx_DAT1	-	SPI_MISO_0	-

表 6.17-1 不同协议的输入信号

具体协议的描述见相关的协议章节。

一般输入结构

数据输入结构和控制信号包含反相器、数字滤波器和边沿检测（仅数据信号）。

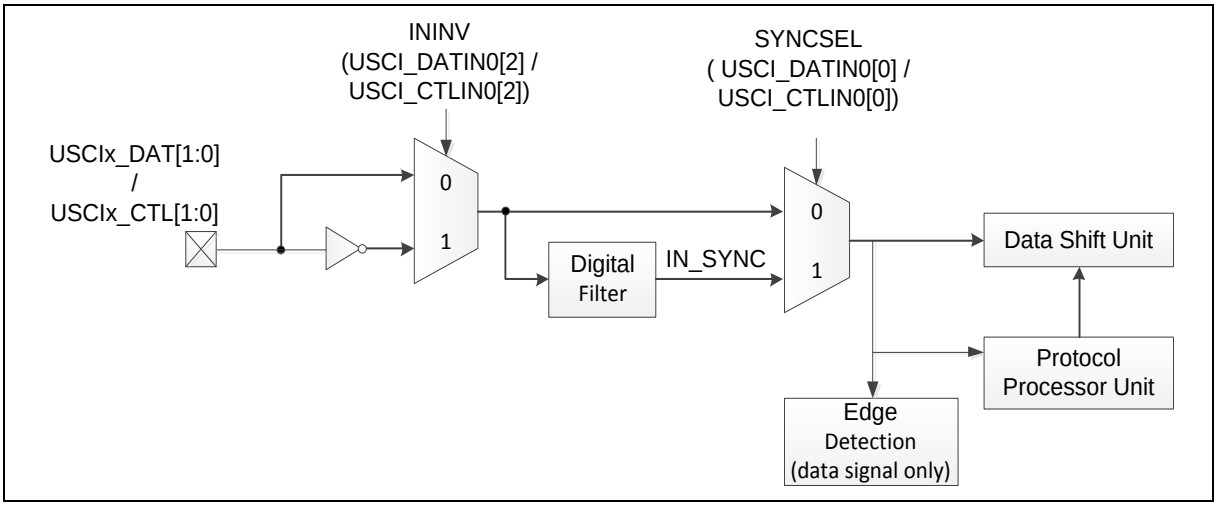


图 6.17-2 输入调节 USCIx_DAT[1:0] 和 USCIx_CTL[1:0]

USCIx_CLK输入结构类似于USCIx_CTL[1:0]输入结构，除了不支持反相功能。

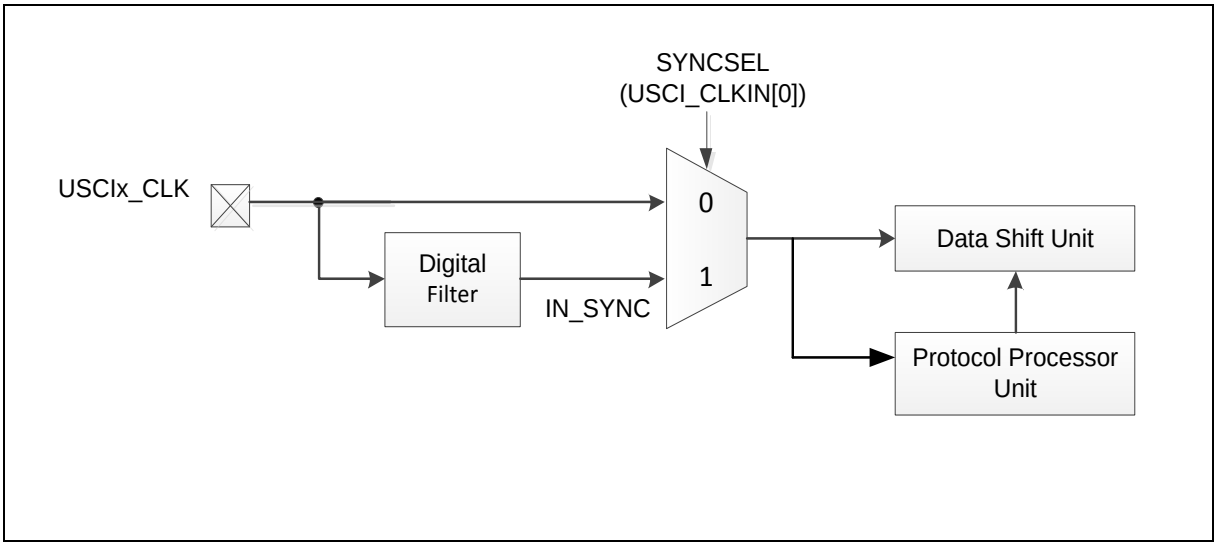


图 6.17-3 输入调节 USCIx_CLK

控制、时钟和数据输入结构的配置分别在USCI_CTLIN0, USCI_CLKIN 和 USCI_DATIN0寄存器中。EDGEDET (USCI_DATIN0[4:3]) 被用来选择边沿检测条件。在 UART 模式 EDGEDET (USCI_DATIN0[4:3])必须设置为2'b10。通过设置可编程的边沿检测及触发信号，可以指示发生了某些事件。

设置ININV (USCI_DATIN0[2] / USCI_CTLIN0[2])寄存器可以反转输入信号的极性以适应数据移位单元及协议状态机的需求。

如果SYNCSEL (USCI_DATIN0[0] / USCI_CTLIN0[0] / USCI_CLKIN[0])被设置为0，由于同步和过滤，输入信号的路径不会包含任何延时。输入信号上的噪声也可能造成信号同步（IN_SYNC被fPCLK同步）；如果SYNCSEL = 1，输入信号会有一个2-3倍fPCLK时钟周期的延时滤波。

输出信号

表6.17-2为每种选择的协议列出了相关的输出信号，实际使用的输出数目依据所选的协议。

选择协议		UART	SPI	I^2C
串行总线时钟输出	USCIx_CLK	-	SPI_CLK	SCL
控制输出	USCIx_CTL0	-	SPI_SS	-
	USCIx_CTL1	nRTS	-	-
数据输出	USCIx_DAT0	-	SPI_MOSI_0	SDA
	USCIx_DAT1	TX	SPI_MISO_0	-

表 6.17-2 不同协议输出信号

具体协议的描述见相关的协议章节。

6.17.4.2 数据缓冲

USCI控制器的数据处理是建立在数据移位单元(DSU)和缓冲结构基础上。数据移位和缓存寄存器都是16

位的宽度。数据移位单元输入包含移位数据、串行总线时钟和移位控制。传输输出引脚可以是USCIx_DAT0 引脚或USCIx_DAT1引脚，这取决于选择的协议。

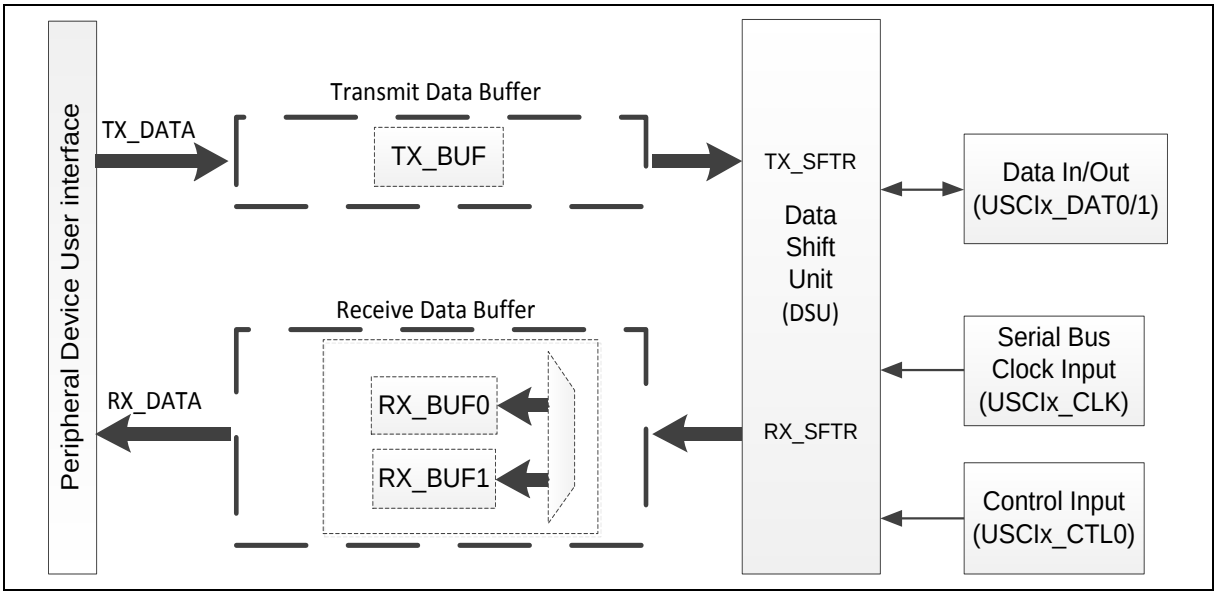


图 6.17-4 数据缓冲框图

数据处理操作包含：

- 外设用户接口(APB)被用于处理数据、中断、状态和控制信息。
- 发送器包含发送移位寄存器(TX_SFTR)和发送数据缓冲(TX_BUF)。TXFULL / TXEMPTY (USCI_BUFSTS[9:8]) 和 TXENDIF (USCI_PROTSTS[2])被用来指示发送的状态。
- 接收器包含接收移位寄存器(RX_SFTR)和双缓冲接收结构(RX_BUF0, RX_BUF1)。在双缓冲结构，用户不用关心接收顺序，如果用户没有及时读取USCI_RXDAT寄存器的数据，接收到两次数据会被保持住。

数据存取结构

数据存取结构包括对接收数据读存取和对传送数据写存取。接收到的数据被保存在接收缓冲RX_BUF0和RX_BUF1中。用户不用关心接收顺序。通过读USCI_RXDAT寄存器来存取接收缓冲区。先接收到数据被先读出，然后在USCI_RXDAT可以查看下一个接收到的数据，可以再一次被读出。

通过写传送寄存器USCI_TXDAT来把发送数据加载到TX_BUF。

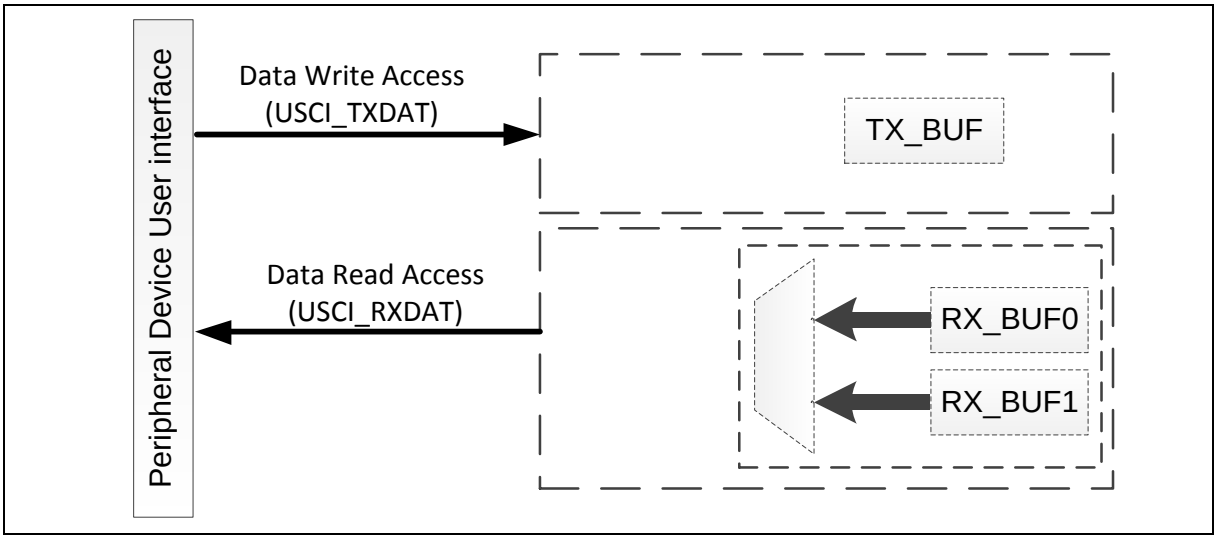


图 6.17-5 数据存取结构

传送数据路径

传送数据路径建立在16位宽度的传送移位寄存器(TX_SFTR)和传送缓冲TX_BUF。数据传输参数比如数据字长通常是通过线控制寄存器USCI_LINECTL来控制发送和接收的。

传送缓冲

传送移位寄存器不能被用户直接存取。在数值存到发送缓冲TX_BUF时会被自动的更新。如果当前传送数据完成则新数据将会被传送。

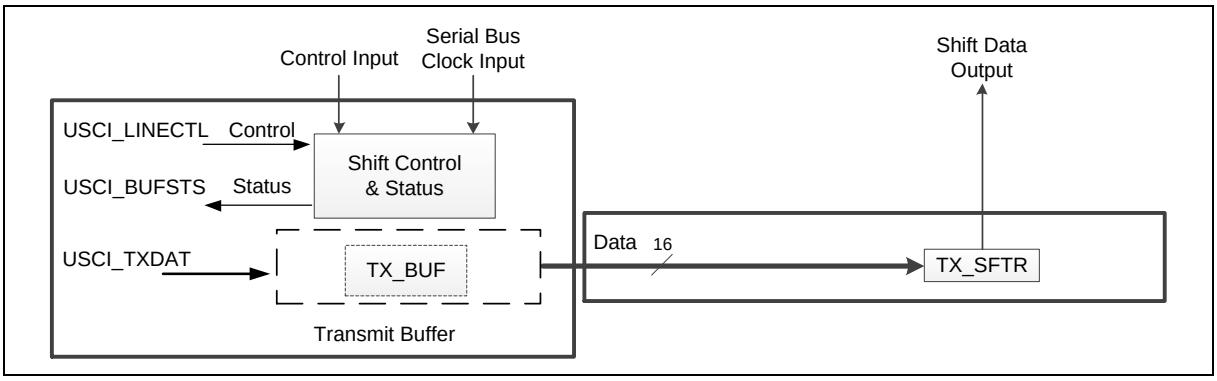


图 6.17-6 传送数据路径

传送数据的有效性

TXEMPTY (USCI_BUFSTS[8])的状态表明在传送缓冲(TX_BUF)中传送数据是有效的，还是没有准备好。TXSTIF (USCI_PROTSTS[1])在每次数据的启动过程时指示。

- 如果USCI控制器是主机，数据传输只能将有效数据载入到传送缓冲(TX_BUF)来启动。在这种情况下，传送移位寄存器会加载传送缓冲的内容。

  注：数据传输启动由主机决定。

- 如果USCI控制器是从机，由主机请求数据传输，从机的启动必须独立于传送缓冲(TX_BUF)的状态。如果一次数据的传输由主机发起，传送移位寄存器需依据具体协议控制信号加载有效的数据。

  注：从机不能自己启动但是必须有回应。

- 从传送缓冲到数据移位单元加载数据时序取决于协议的配置。
- **UART:**在正常操作下，如果TXEMPTY = 0，在缓冲中数据字被开始传送。在自动流控下，当TXEMPTY = 0 和 USCIx_CTL0在有效状态时，在缓冲中数据字的传送开始启动。
- **SPI:** 在主机模式下，当TXEMPTY (USCI_BUFSTS[8])为0时数据传送将会启动。在从机模式下，当从机选择信号处在启用状态和USCIx_CLK引脚出现时钟信号时，数据传送将会启动。
- **I^{2}C:**如果TXEMPTY = 0，在缓冲中数据字被传送。
- 如果TXEMPTY (USCI_BUFSTS [8]) = 0缓冲中的传输数据会开始传送。当下一次有效的新的传送启动时，传送缓冲的内容(TX_BUF)不能被新的数据重新写入。如果TX_BUF的内容必须改变，在更新数据前用户必须设置TXRST (USCI_BUFCTL [16])为1去清除TX_BUF的内容。而且当传送缓冲(TX_BUF)被新数据更新时，TXEMPTY (USCI_BUFSTS [8])会被自动的清除。当一次传送正在进行时，TX_BUF可以加载新的数据。用户必须在新的传送前更新TX_BUF。

接收数据路径

接收数据路径基于16位宽度的接收移位寄存器(RX_SFTR)和传送缓冲RX_BUF0 和 RX_BUF1。数据传输参数比如数据字长或移位方向通常是通过线控制寄存器USCI_LINECTL来控制发送和接收的。寄存器USCI_BUFSTS监控USCI_RXDAT数据的有效性。

接收缓冲

接收移位寄存器不能被用户直接存取。但是如果接收到一个完整数据字或数据帧，则该数据会被自动地载入到接收缓冲中。缓冲的接收数据字可以通过寄存器USCI_RXDAT被读出。

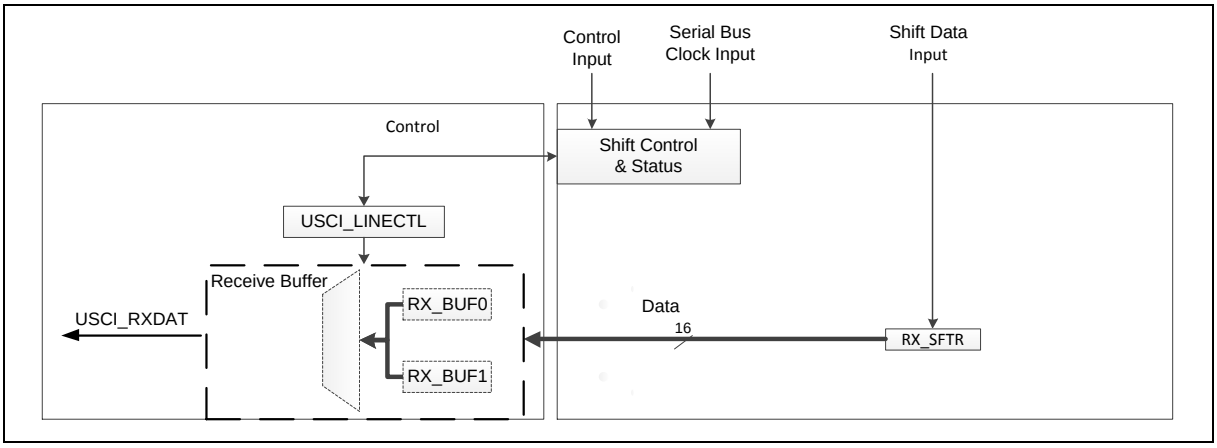


图 6.17-7 接收数据路径

6.17.4.3 端口方向控制

SPI协议是半双工配置，数据端口是双向的。端口方向控制通过专用的硬件接口来控制引脚方向。

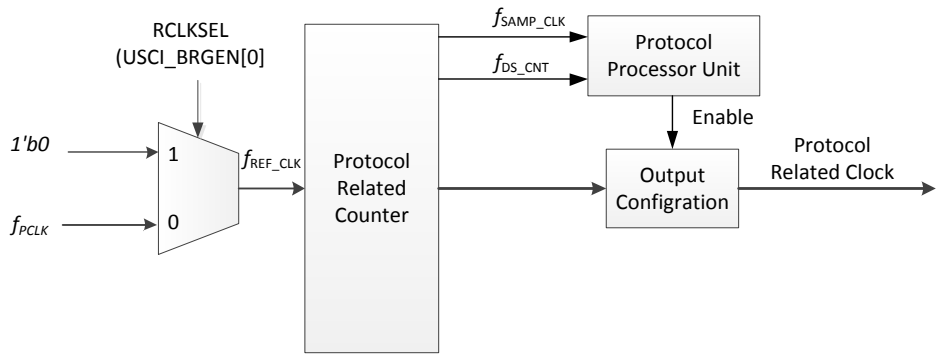
通过PORTDIR (USCI_TXDAT[16])来控制选中引脚的方向。当用户写USCI_TXDAT寄存器时，传送数据和它的端口方向同时被操作。

6.17.4.4 协议控制和状态

相关协议的控制和状态信息在协议控制寄存器USCI_PROTCTL和协议状态寄存器USCI_PROTSTS中。这些寄存器之间共享可用的协议。因此，这些寄存器每个位的意义在不同协议间会有所不同。详细信息参考每个协议的相关寄存器。

6.17.4.5 相关协议的时钟发生器

USCI控制器包含相关协议时钟发生器，由寄存器USCI_BRGEN控制。当USCI_BRGEN寄存器被写入时会产生复位。相关协议时钟发生器结构如下所示。



注：参考**基本时钟分频计数器**章节获取f_{SAMP_CLK}

图 6.17-8 相关协议时钟发生器

协议相关的计数器包含基本时钟分频计数器和时序测量计数器。它基于一个分频器，为不同协议提供所需的频率。主要包含：

- 基本时钟分频计数器提供协议的相关时钟信号和其它相关协议信号(fSAMP_CLK和fDS_CLK)
- 时序测量计数器提供时间间隔测量，比如，UART协议的波特率侦测。
- 相关协议时钟发生器输出信号输出到相应引脚上（比如SPI的USCIx_CLK）

基本时钟分频计数器

基本时钟分频计数器是来自对f_{REF_CLK2}，f_{REF_CLK}，f_{DIV_CLK}，f_{SCLK} ,和f_{SAMP_CLK}的分频。这些分频器的频率由PTCLKSEL (USCI_BRGEN [1]), CLKDIV (USCI_BRGEN [25:16]), SPCLKSEL (USCI_BRGEN [3:2])控制。

基本时钟分频计数器用来生成相关协议时序信号。

$$f_{\mathrm{DIV_CLK}} = f_{REF_CLK} \times \frac{1}{\mathrm{CLKDIV} + 1} \text{ if PTCLKSEL} = 0$$

$$f_{\mathrm{DIV_CLK}} = f_{REF_CLK} \times \frac{1}{(\mathrm{CLKDIV} + 1) \times 2} \text{ if PTCLKSEL} = 1$$

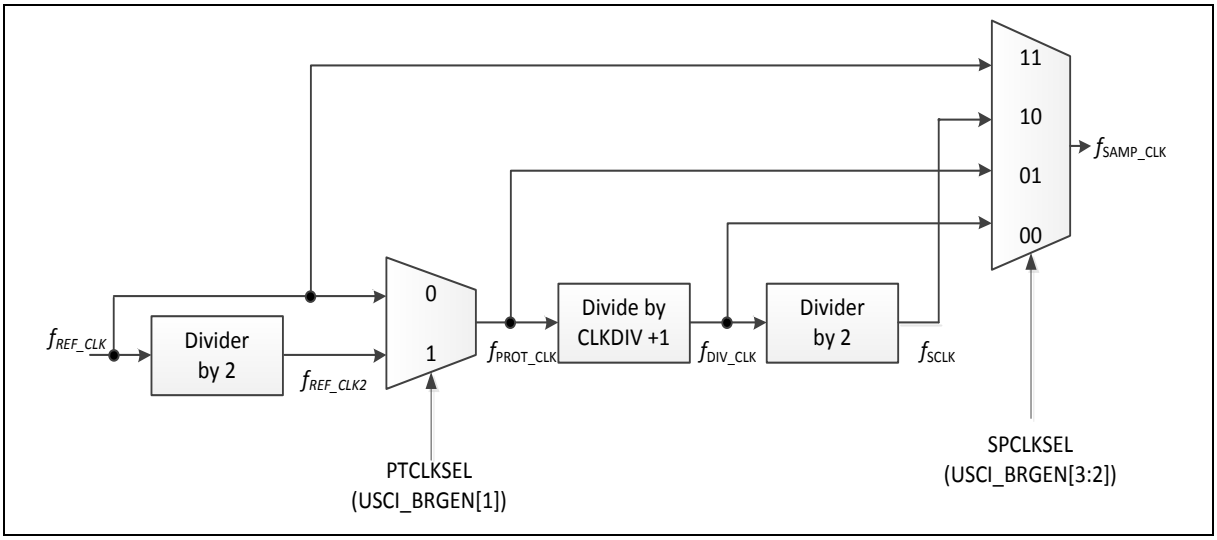


图 6.17-9 基本时钟分频计数器

时序测量计数器

时序测量计数器是用来对时间间隔的测量，通过TMCNTEN (USCI_BRGEN [4]) = 1使能。当TMCNTSRC (USCI_BRGEN [5])被置1，定时器时钟来自f_{DIV_CLK}，否则来自f_{PROT_CLK}。因此在串行数据接收和发送的同时定时器可以执行时序测量。定时器计数的相关协议信号来自f_{PROT_CLK}或 f_{DIV_CLK}。当它达到用户指定的值时会停止计数。

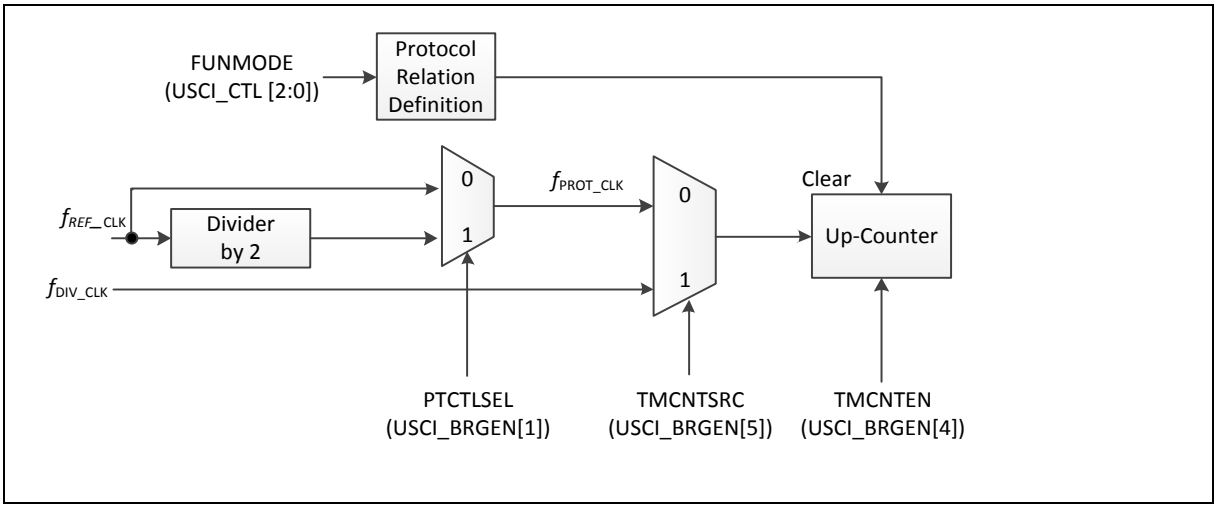


图 6.17-10 时序测量计数器框图

时序测量计数器是用来执行超时功能或自动波特率机制。该功能取决于所选用的协议，如下所示。

- UART: 时序测量计数器被用作自动波特率侦测
- SPI: 时序测量计数器被用作从机超时周期计数
- I^2C: 时序测量计数器被用作超时时钟周期

采样时间计数器

采样时间计数器与决定协议时序的协议相关计数器是有关联的。比如移位控制信号或位时序是建立在输入频率f_{SAMP_CLK} 基础上的。采样时间计数器为具体的协议产生时间间隔。采样频率f_{PDS_CNT} 周期是通过选择输入频率f_{SAMP_CLK} 和可编程预分频值(PDSCNT (USCI_BRGEN [9:8]))给定的。这就意味着采样时

间取决于所选的协议。详细协议信息请参考相关的章节。

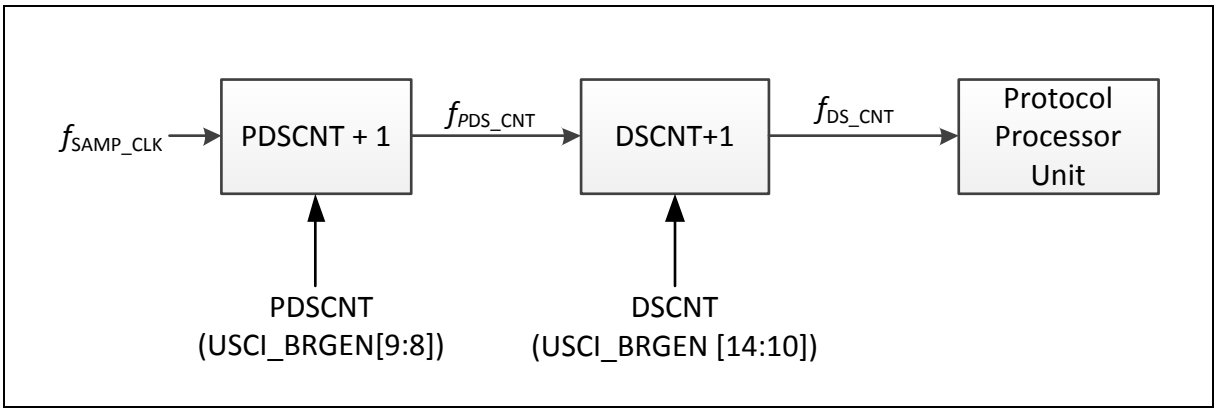


图 6.17-11 采样时间计数器

6.17.4.6 数据传输事件和中断

数据传输事件是建立在一个数据字的发送和接收基础上的。相关的标志在寄存器USCI_PROTSTS中。所有事件可以被单独使能，用于产生中断。如果FUNMODE (USCI_CTL [2:0])被置0，USCI会被禁用。当FUNMODE (USCI_CTL [2:0])被设置为某个协议端口时，内部状态会被所选的协议的逻辑硬件所控制。

- 传送起始中断事件来表明一个数据字传送已经启动：

  当数据被载入到发送移位寄存器时一个传送起始中断发生。由TXSTIF (USCI_PROTSTS [1])标志指示，如果中断使能将会产生传送起始中断。

- 传送结束中断事件表明一个数据字传送已经完成：

  当在移位寄存器的发送数据已经完成时传送结束中断事件发生。由TXENDIF (USCI_PROTSTS [2])标志指示，如果中断使能将会产生传送结束中断。当移位控制设置（字长度、移位方向等等）时该事件同时可以指示当前数据字传输被内部“冻结”。在UART和I^2C模式下，依据TXEMPTY (USCI_BUFSTS [8])和协议相关内部信号传送结束中断事件来确定传送数据的有效性。

- 接收起始事件表明一个数据字接收已经启动：

  当检测到接收时钟边沿在新数据字的第一个位发生改变并且接收使能的时候，接收起始事件发生。由RXSTIF (USCI_PROTSTS [3])标志指示，如果中断使能将会产生接收起始中断。

- 接收事件表明一个数据字接收已经完成：

  当一个新接收的字在接收缓冲中完成时，接收事件发生。由RXENDIF (USCI_PROTSTS [4])标志指示，如果中断使能将会产生接收中断。

- 数据丢失事件表明最新接收到的数据丢失：

  如果在寄存器USCI_RXDAT (旧的数据来自 RX_BUF0 或 RX_BUF1)的可用的数据没有被读出那么接收缓冲会溢出，新来的数据会丢失掉，该事件发生。由RXOVIF (USCI_BUFSTS[3])标志指示，如果中断使能将会产生协议中断

事件产生和中断结构如下图6.17-12.

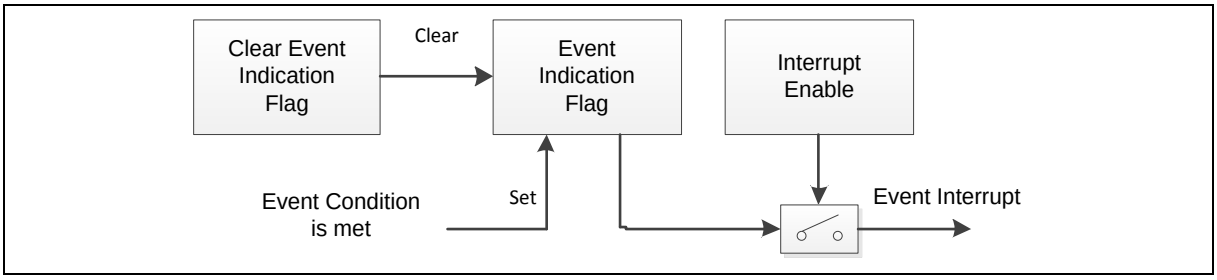


图 6.17-12 事件和中断结构

每个中断使能是通过设置USCI_INTEN [4:1]中的RXENDIEN, RXSTIEN, TXENDIEN, 和 TXSTIEN。这些事件包括接收结束中断事件、接收起始中断事件、传送结束中断事件和传送起始中断事件。对于具体的中断在每个协议中断使能寄存器中指定。

如果一个定义条件发生，此事件被侦测到并且事件指示标志自动被设置。该标志保持置位直到被软件清除。如果对应中断使能，在事件被侦测到时会产生中断。

寄存器、位和位域表明传输事件和控制产生下表6.17-3所示的USCI中断。

事件	指示标志	指示清除	中断使能
发送开始中断标志	TXSTIF (USCI_PROTSTS [1])	由软件写1到 USCI_PROTSTS对应的中断位清除	TXSTIEN (USCI_INTEN [1])
发送结束中断标志	TXENDIF (USCI_PROTSTS [2])		TXENDIEN (USCI_INTEN [2])
接收开始中断标志	RXSTIF (USCI_PROTSTS [3])		RXSTIEN (USCI_INTEN [3])
接收结束中断标志	RXENDIF (USCI_PROTSTS [4])		RXENDIEN (USCI_INTEN [4])

表 6.17-3 数据传输事件和中断处理

6.17.4.7 具体协议事件和中断

这些事件与在对应协议的章节描述的具体协议动作有关。相关的标志位在寄存器USCI_PROTSTS中。对于共同协议的中断，所有事件可以单独的使能。

事件	指示标志	指示清除	中断使能
UART模式下的协议	USCI_PROTSTS [17:16] 和USCI_PROTSTS [11:5]	由软件写1到 USCI_PROTSTS对应的中断位清除	USCI_PROTIEN[2:1]
SPI模式下的协议	USCI_PROTSTS [9:8], USCI_PROTSTS [6:5]		USCI_PROTIEN [3:0]
I^2C 模式下的协议	USCI_PROTSTS [13:8], USCI_PROTSTS [5]		USCI_PROTIEN [6:0]

表 6.17-4 具体协议事件和中断处理

6.17.4.8 唤醒

相关协议唤醒功能信息在唤醒控制寄存器(USCI_WKCTL)和唤醒状态寄存器(USCI_WKSTS)中。在可用的协议之间可以共享。因此，这些寄存器的位域代表的意义在不同协议之间会有所不同。

6.17.4.9 PDMA

USCI支持PDMA传输。当PDMAEN (USCI_PDMACTL[3])置1时，PDMA功能使能。

当TXPDMAEN (USCI_PDMACTL [1])置1时，控制器会产生请求到PDMA控制器来自动地启动PDMA传输。

当RXPDMAEN (USCI_PDMACTL [2]) 置1时，控制器会启动PDMA接收进程。当数据接收到FIFO缓冲时，USCI会自动地产生请求到PDMA控制器。

在UART功能下，如果有错误事件（包括帧错误、奇偶校验错误和break侦测），RXPDMAEN的请求会被清除并保持住。用户需要读出当前的数据，下一个数据的RXPDMAEN请求将发送给PDMA模块。

6.18 USCI – UART模式

6.18.1 概述

异步串行通道UART能处理异步数据帧的接收和发送。接收过程是把外设的串行数据转为并行数据，发送过程是把CPU的并行数据转成串行数据发送出去。接收和发送帧是独立的，他们可以在不同时刻启动。

UART控制器也提供流控功能，有两个条件可以唤醒系统。

6.18.2 特性

- 有一个发送缓冲和两个接收缓冲来存放数据
- 支持硬件流控功能
- 支持可编程的波特率发生器
- 支持9位数据传输 (支持 9位 RS-485)
- 波特率发生器内建捕获事件为波特率检测提供了可行性
- 支持PDMA功能
- 支持唤醒功能 (仅数据和nCTS 唤醒)

6.18.3 框图

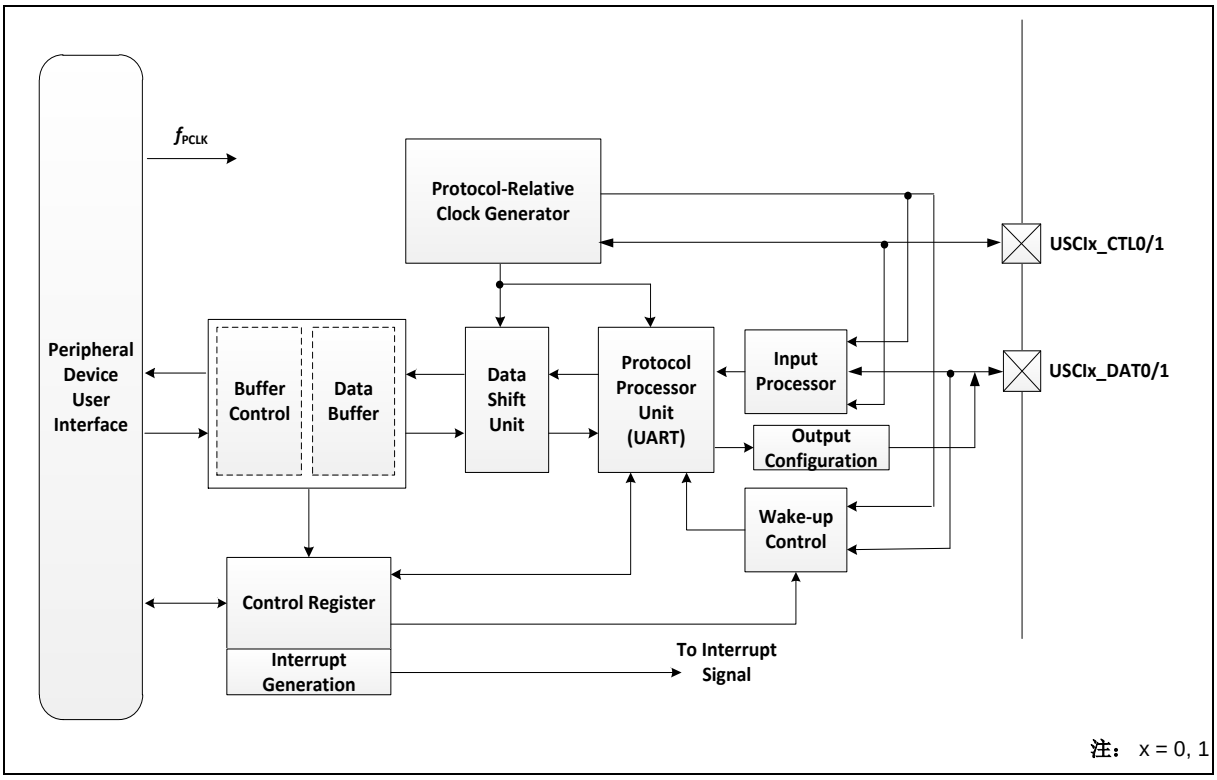


图 6.18-1USCI-UART 模式框图

6.18.4 基本配置

6.18.4.1 USCI0_UART基本配置

- 时钟源配置
  - 在USCI0CKEN（CLK_APBCLK1[8]）使能USCI0时钟
  - 在USPI_CTL[2:0]寄存器使能USCI0_UART功能, USPI_CTL[2:0]=3’b010
- 复位配置
  - 在USCI0RST（SYS_IPRST2[8]）复位USCI0控制器

6.18.4.2 USCI1_UART基本配置

- 时钟源配置
  - 在USCI1CKEN (CLK_APBCLK1[9])使能USCI1外设时钟
  - 在USPI_CTL[2:0]寄存器使能USCI1_UART功能, USPI_CTL[2:0]=3’b010
- 复位配置
  - 在USCI1RST (SYS_IPRST2[9])复位USCI1控制器

6.18.5 功能描述

6.18.5.1 USCI共同功能描述

具体信息参考6.17.4章节。

6.18.5.2 信号描述

UART半双工连接的特点是在发送和接收之间使用单根连接线。接收输入信号(RXD)由输入USCIx_DAT0处理，发送输出(TXD) 信号由USCIx_DAT1处理。

对于全双工通信，每个传输方向需要一根独立的通信线。图6.18-2展示的是UART模块A和UART模块B之间用点对点全双工通信的连接。

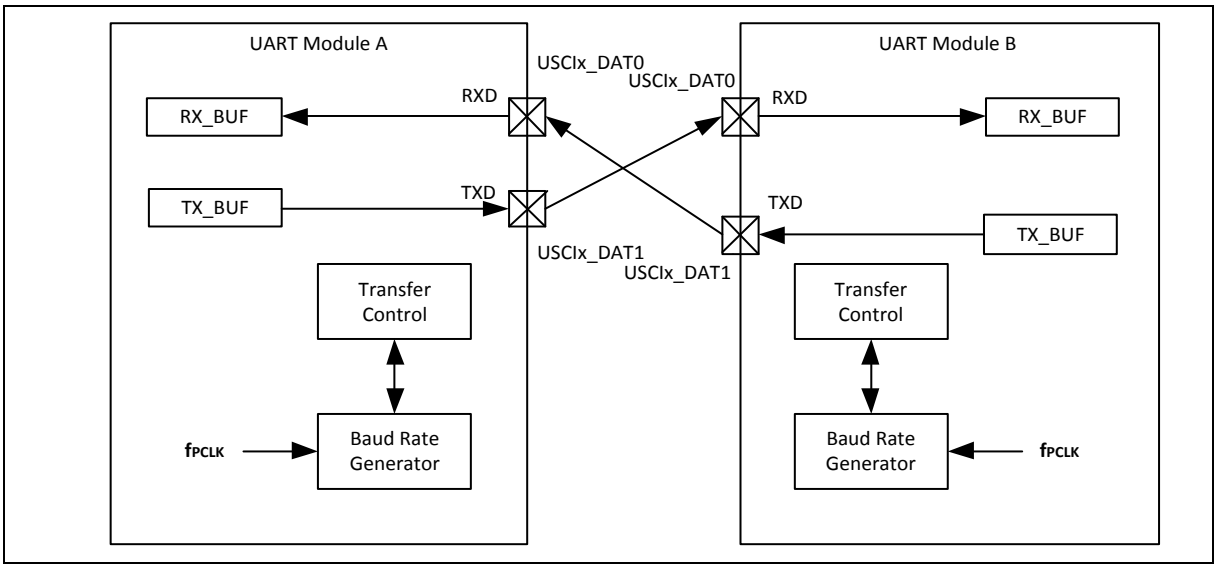


图 6.18-2 UART 全双工通信的 UART 信号连接

输入信号

对于UART协议，表6.18-1中列出了输入信号的数目。每种输入信号由输入处理器处理加工。比如信号反相选择控制或数字输入滤波。他们可以依据在协议中的意义进行分类，见表6.18-1。

选择协议		UART
控制输入	USCIx_CTL0	nCTS
	USCIx_CTL1	X
数据输入	USCIx_DAT0	RX
	USCIx_DAT1	X

表 6.18-1 协议输入信号

输出信号

对于UART协议，每个相关协议输出信号都是可用的。使用输出的数目实际是取决于所选的协议。他们可以依据在协议中的意义进行分类。

选择协议		UART
控制输出	USCIx_CTL0	X
	USCIx_CTL1	nRTS
数据输出	USCIx_DAT0	X
	USCIx_DAT1	TX

表 6.18-2UART 协议的输出信号

6.18.5.3 帧格式

标准UART 帧如图6.18-3所示，包含:

- 信号为1的空闲时间
- 信号为0的一个起始帧位（SOF）
- 6~13 位数据
- 一个校验位(P), 可选择奇校验或偶校验。还可以选择是否需要该位。
- 信号为1的一或两位的停止位

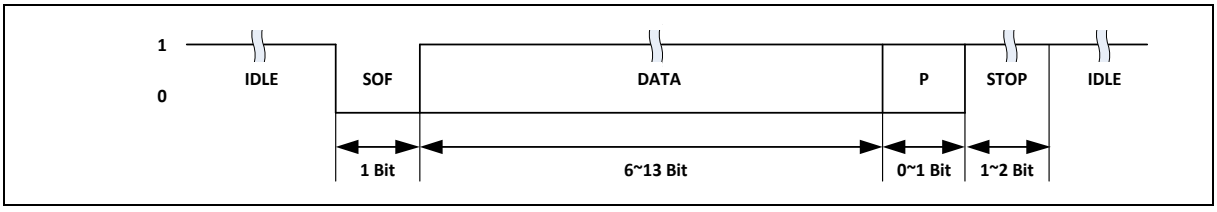


图 6.18-3 UART 标准帧格式

协议的特定位(SOF, P, STOP)由UART协议状态机自动处理，不会在接收和发送缓冲时出现在数据中。

起始位

接收器输入信号USCIx_DAT0检测是以一个下降沿开始的。当接收器处于空闲或位于上一个停止位采样点后，检测到下降沿则认为是一个SOF位。为了增强抗噪声能力，下降沿触发SOF位时序后如果采样位的值为1，那么先前下降沿被当作是噪声并且接收器会再次进入空闲。

数据域

数据域的长度（数据位数目）可以通过DWIDTH (UUART_LINECTL[11:8])编程。取值范围在6到13数据位。

注：在UART协议中，为了先传输LSB需设置LSB (UUART_LINECTL[0])为1。

校验位

UART可以在发送的时候生成校验位，在接收的时候检查校验位。通过PARITYEN (UUART_PROTCTL[1]) 和 EVENPARITY (UUART_PROTCTL[2])来选择校验的类型（无校验、奇校验或偶校验）。如果禁用校验功能，UART帧不会包含校验位。为了一致性，所有通信的设备必须配置成一样的校验类型。

在数据域的最后数据位之后，如果校验功能使能，发送器会自动地计算校验位并发出去。接收器解释该位作为接收校验并且和内部计算的相比较。校验检测和帧检测（停止位）在协议状态寄存器(UUART_PROTSTS)中被监测。该寄存器包含监测相关协议状态和相关协议错误指示(FRMERR, PARITYERR)这些位。

停止位

UART帧是以1或2位信号为1的停止位结束的（电平同空闲时电平）。停止位数目由STOPB (UUART_PROTCTL[0])位编程。在停止位过后一个新的起始位可以被直接的传输。

传输状态指示

RXBUSY (UUART_PROTSTS[10])指示接收状态。

通过RXBUSY位监测接收状态。在起始帧位开始到最后的停止位结束期间RXBUSY被置位。

6.18.5.4 操作模式

为了操作UART协议，必须考虑下面的问题：

选择 UART 模式

通过设置FUNMODOE (USCI_CTL[2:0]) 为010B选择UART协议，通过设置PROTEN (USCI_PROTCTL[31])为1使能UART协议。注意在改变协议前必须设置FUNMODE为0，强烈建议在使能UART协议前配置好UART的所有参数。

引脚连接

在UART协议中USCIx_DAT0引脚被用作接收数据输入信号(RX)。输入信号的属性在UUART_DATIN0中配置。对于起始位侦测，建议设置EDGEDET (UUART_DATIN0[4:3])为10B。

在UART协议中USCIx_DAT1引脚被用作发送数据输出信号(TX)。输出信号的属性在UUART_LINECTL中配置。

在UART协议中USCIx_CTL0引脚被用作UART清除发送信号 (nCTS)。输入控制信号的属性在UUART_CTLIN0中配置。

在UART协议中USCIx_CTL1引脚被用作UART请求发送信号(nRTS)。输出控制信号的属性在UUART_LINECTL中配置。

位时序配置

需要的波特率设置必须被选择，包括波特率发生器和位时序在内。

帧格式配置

字长度、停止位数目和校验模式必须依据应用需求来设定，这些都是通过UUART_LINECTL和UUART_PROTCTL寄存器来编程。如果应用需要，数据输入和输出信号可以反转。通过设置LSB (UUART_LINECTL[0])为1来决定先传送LSB。

6.18.5.5 位时序

在UART模式下，每个数据位被分成若干采样时间，以便根据应用需要来调整采样点，每个位的采样时间个数通过DSCNT (UUART_BRGEN[14:10])来设定，每个位的采样时间长度通过PDSCNT (UUART_BRGEN[9:8])设定

在图6.18-4所示的情况下，一个位被分为16个采样时间DSCNT(UUART_BRGEN[14:10]) = 15，建议每个数据位的采样时间个数大于4

每个位的采样点固定在1/2采样时间个数，采样数据会取平均

发送接收会用同样的位时序设定（采样时间个数），因为他们使用相同的硬件电路

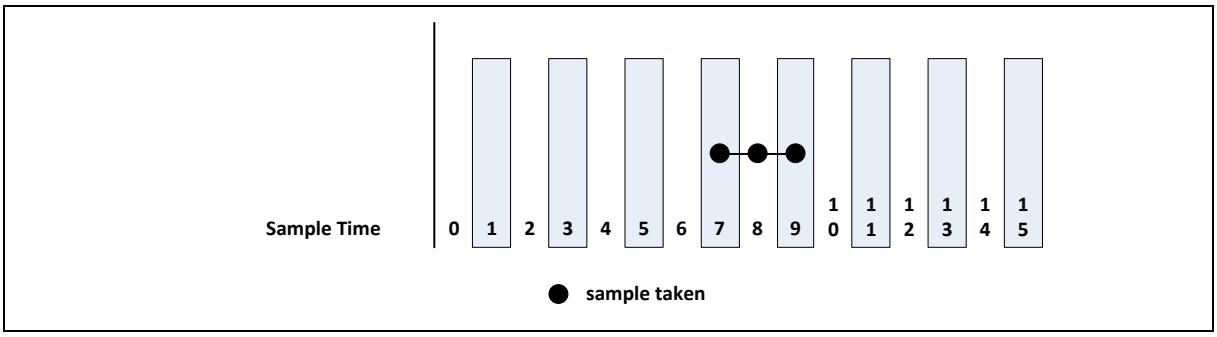


图 6.18-4 UART 位时序 (数据采样时间)

6.18.5.6 波特率发生器

在UART模式下波特率取决于数据采样每位时间的数目和他们的时序。波特率设定只能在发送器和接收器空闲的时候改变。在RCLKSEL, SPCLKSEL, PDSCNT, 和 DSCNT中设置波特率：

RCLKSEL (UUART_BRGEN [0])

定义输入频率 f_{REF_CLK}

SPCLKSEL (UUART_BRGEN[3:2])

定义采样时钟fSAM_CLK的时钟源

PDSCNT (UUART_BRGEN [9:8])

定义数据采样时间的长度 (由 f_{REF_CLK} 经 1, 2, 3, 或 4分频)

DSCNT (UUART_BRGEN [14:10])

定义一个数据位的采样时间数目

标准波特率设定是由 RCLKSEL = 0 (f_{REF_CLK} = f_{PCLK}), PTCLKSEL = 0 (f_{PROT_CLK} = f_{REF_CLK}) 和 SPCLKSEL = 2'b00 (f_{SAMP_CLK} = f_{DIV_CLK})决定. 在这些条件下，波特率值如下:

$$f_{\mathrm{UART}} = f_{REF_CLK} \times \frac{1}{\mathrm{CLKDIV}+1} \times \frac{1}{\mathrm{PDSCNT}+1} \times \frac{1}{\mathrm{DSCNT}+1}$$

为了产生更慢的频率, 可通过以下公式选择除2 PTCLKSEL = 1 ($f_{PROT_CLK} = f_{REF_CLK2}$), 如下:

$$f_{\mathrm{UART}} = \frac{f_{REF_CLK}}{2} \times \frac{1}{\mathrm{CLKDIV}+1} \times \frac{1}{\mathrm{PDSCNT}+1} \times \frac{1}{\mathrm{DSCNT}+1}$$

如果 SPCLKSEL = 2'b10 ($f_{SAMP_CLK} = f_{SCLK}$), 和 RCLKSEL = 0 ($f_{REF_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT_CLK} = f_{REF_CLK}$). 波特率如下:

$$f_{\mathrm{UART}} = f_{REF_CLK} \times \frac{1}{\mathrm{CLKDIV}+1} \times \frac{1}{2} \times \frac{1}{\mathrm{PDSCNT}+1} \times \frac{1}{\mathrm{DSCNT}+1}$$

UART波特率会有一些误差。表6.18-3列出了使用不同晶振的误差百分比。

HCLK 源	PCLK 源	期望波特率	CLKDIV (UUART_BRGEN[25:16])	DSCNT (UUART_BRGEN[14:10])	PDSCNT	实际波特率	误差百分率
12MHz	HCLK	115200	0xC	0x7	0x0	115384	0.16%
12MHz	HCLK	9600	0x7C	0x9	0x0	9600	0%
12MHz	HCLK	2400	0x1F3	0x9	0x0	2400	0%

表 6.18-3 波特率关系

注： {SPCLKSEL, PTCLKSEL, RCLKSEL = 2'b0,1'b0,1'b0}

6.18.5.7 自动波特率侦测

UART支持自动波特率检测功能，可以识别输入信号(USCIx_DAT0)的波特率，并自动调整波特率时钟寄存器CLKDIV (UUART_BRGEN[25:16])，以匹配输入信号的时序。根据时序测量计数器部分的描述，时序测量计数器会测量输入信号(USCIx_DAT0)并把计数值在每个信号下降沿保存到BRDETITV (UUART_PROTCTL [24:16])中

当ABREN (PROTOCOL[6])被使能，模板数据0x55用来做波特率自动识别的信号，输入信号下降沿触发波特率计数开始，在下一个下降沿将计数的值存储到BRDETITV (UUART_PROTCTL [24:16])。建议使用f_{DIV_CLK} (TMCNTSRC (UUART_BRGENC[5]) =1)作为计数时钟源

波特率自动检测完成后（输入信号0x55的第四个下降沿），CLKDIV (UUART_BRGEN[25:16])会根据BRDETITV (UUART_PROTCTL [25:16])的值自动修正，如果用户希望正确接收下一个数据帧，最好将CLKDIV (UUART_BRGEN[25:16]) 和 DSCNT (UUART_BRGEN[14:10])设为相同的值，该值应该处于0xF到0x5之间，因为DSCNT定义了每个位的采样次数而此时PDSCNT (UUART_BRGEN[9:8])=0x0

波特率自动检测过程中，ABRDETIF (UUART_PROTSTS[9]) 和 BRDETITV (UUART_PROTCTL [24:16])在输入信号的每个下降沿都会更新，模板数据0x55不会被接收到输入缓冲中。ABREN位将在输入信号的第四个下降沿被硬件清零，用户可以通过读ABREN的状态了解自动波特率检测是否完成。

如果CLKDIV 和 DSCNT在自动波特率检测过程中没有被设置为相同的值，需要用户根据BRDETITV 和 CLKDIV去计算合适的值

如果输入信号的波特率很慢，时序测量计数器不能计算正确的信号周期，ABERRSTS (UUART_PROTSTS[11])会显示一个错误状态，这时用户需要修改CLKDIV的值并要求主机重新发送0x55模板数据

根据时序测量计数器的限制，BRDETITV能够支持的最大计数值为0x1FE，UART自动波特率控制如图6.18-5所示。

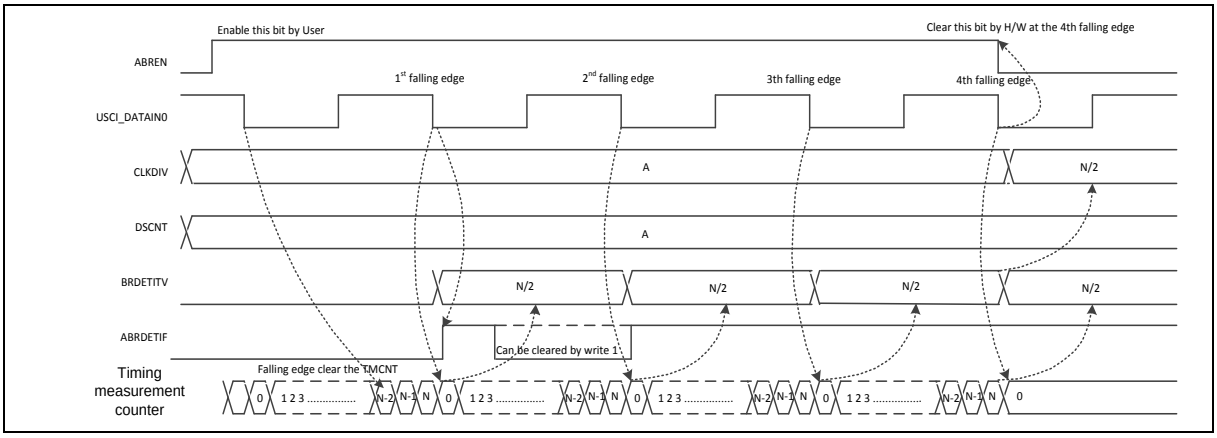


图 6.18-5 UART 自动波特率控制

6.18.5.8 自动流控

UART支持硬件自动流量控制，接收缓存指示标志RXFULL (UUART_BUFSTS[1])控制nRTS，当接收缓存满了(RXFULL = 1)时，nRTS被清零

UART同时也支持发送器的nCTS，nCTS置位发送数据。

6.18.5.9 RS-485 支持

UART控制器可以作为RS-485主机，通过设置校验位（第9位）为1可以表示地址，校验位为0表示数据。当STICKEN (UUART_PROTCTL[26])=1，软件可以通过每个数据的第15位来控制校验位(PARITYEN (UUART_PROTCTL[1])。例如如果STICKEN=1，发送数据顺序为：0x8015, 0x8033, 0x0055, 0x0033 及 0x80AA，发送出的数据为：0x15, 0x33, 0x55 0x33, 0xAA，他们的校验位分别为：1, 1, 0, 0 , 1

UART 控制器也可以作为 RS-485 从机，当 PARITYEN (UUART_PROTCTL[1]), EVENPARITY (UUART_PROTCTL[2]) 及 STICKEN(UUART_PROTCTL[26]) 都被置位时，协议错误位 PARITYERR(UUART_PROTSTS[5])可以作为地址位检测，如果PARITYERR=1 表示检测到地址否则就是数据

6.18.5.10 唤醒功能

UART模式下的USCI控制器支持唤醒功能，唤醒源包括接收到数据和nCTS管脚信号，下面详细描述每个唤醒源：

(a) 输入数据唤醒

当系统进入掉电模式，并且WKEN (UUART_WKCTL [0]) 和 DATWKEN (UUART_PROTCTL[9])都被置位，数据输入管脚的电平变化会唤醒系统。如果希望唤醒后能够收到这个唤醒数据，必须设置WAKECNT (UUART_PROTCTL[14:11])寄存器，它表示系统唤醒时，数据起始位期间有多少个f_{PDS_CLK}时钟周期。输入数据唤醒如图6.18-6所示。

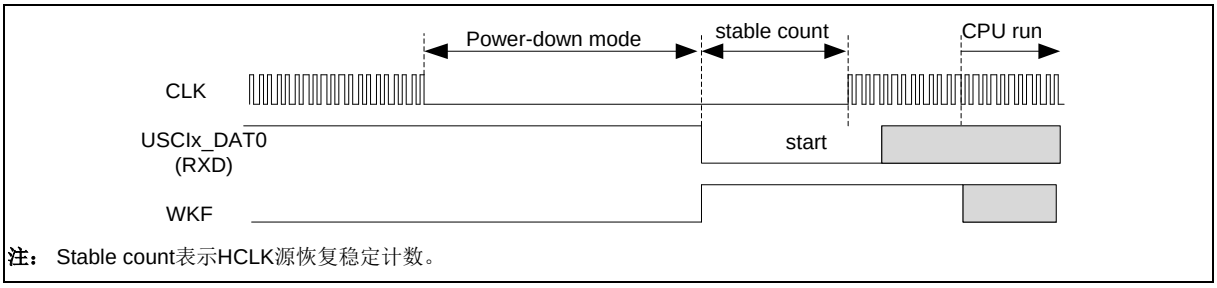


图 6.18-6 输入数据唤醒

(b) nCTS 引脚唤醒

当系统处在掉电模式下时并且WKEN (UUART_WKCTL [0]) 和 CTSWKEN (UUART_PROTCTL[10])被设置，nCTS引脚的变化可以唤醒系统。nCTS唤醒如图6.18-7和图6.18-8.

情况 1(nCTS 传输从低变高):

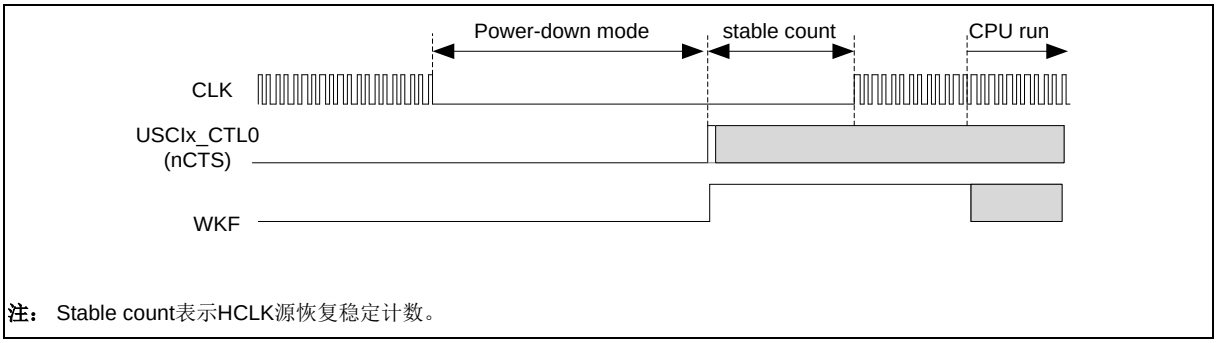


图 6.18-7nCTS 唤醒情况 1

情况 2 (nCTS 传输从高变低):

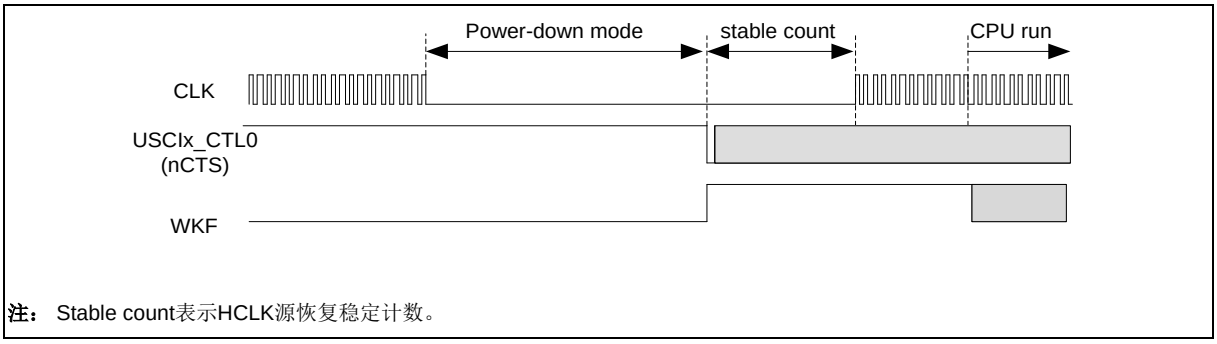


图 6.18-8nCTS 唤醒情况 2

6.18.5.11 中断事件

UART为协议事件和数据传输事件提供了中断。描述如下：

协议中断事件

在UART模式下，可以产生下面相关协议事件并且可以产生协议中断。

请注意UUART_PROTSTS位不会被硬件自动地清除，必须通过软件清除，这样是为了监控新到来的事件。

接收器线状态

协议相关的错误标志FRMERR (UUART_PROTSTS[6]) 及 PARITYERR (UUART_PROTSTS[5])，在每次接收数据时都会在其状态寄存器中更新

UART模式下，奇偶校验错误标志为零，表示没有奇偶校验错误，帧错误标志为零表示没有帧错误，这些标志在每个数据帧都会更新

在收到的数据，校验位，停止位都为零时，Break错误标志(BREAK (UUART_PROTSTS[7]))置位

当产生校验错误，帧错误及break错误标志时（BREAK, FRMERR, PARITYERR (UUART_PROTSTS[7:5])），如果RLSIEN (UUART_PROTIEN[2])=1，则会产生相关错误中断

自动波特率侦测

自动波特率中断标志ABRDETIF (UUART_PROTSTS [9])=1，表示时序测量计数器捕获到2个数据位时长的计数，如果ABRIEN (UUART_PROTIEN[1])=1 控制器会产生中断

自动波特率检测功能，在模板数据0x55的第一个下降沿触发，并开始计数一直计到下一个下降沿，整个模板数据接收过程中会产生4次中断，用户将捕获的数据除以2得到的就是每个数据位周期内的采样时间个数。用户可以读出BRDETITV (UUART_PROTCTL[24:16]) 的值然后写入CLKDIV (UUART_BRGEN[25:16])中

数据传输中断处理

数据传输中断指示相关UART帧处理事件。

发送起始中断

发送数据的起始位发出后TXSTIF (UUART_PROTSTS[1]) 置1，在缓存模式，这是可以向UUART_TXDAT写入数据的最早时间点，如果此时TXSTIEN (UUART_INTEN[1])=1，控制器将会产生中断

发送结束

该中断表示发送缓存中的数据完全发送完毕了，数据结束位发送完成后TXENDIF (UUART_PROTSTS[2])=1，如果此时TXENDIEN (UUART_INTEN[2])=1，将会产生中断

接收起始中断

当控制器接收到数据起始位的采样点，RXSTIF (UUART_PROTSTS[3])置位，如果此时RXSTIEN (UUART_INTEN[3])=1 控制器将产生中断

接收帧完成

该中断表示接收器完整接收到一个数据帧，RXENDIF (UUART_PROTSTS[4])置位，如果此时RXENDIEN (UUART_INTEN[4])=1，控制器会产生中断

6.18.5.12 编程范例

下面步骤是被用作配置UART协议设置和数据传输

1. 设置 FUNMODE (UUART_CTL[2:0]) 为 0x2 来选择 UART 协议
2. 写波特率寄存器UUART_BRGEN来选择期望的波特率
   - 设置SPCLKSEL (UUART_BRGEN[3:2]), PTCLKSEL (UUART_BRGEN[1]) 和 RCLKSEL (UUART_BRGEN[0]) 来选择时钟源

- 配置 CLKDIV (UUART_BRGEN[25:16]), DSCNT (UUART_BRGEN[14:10]) 和PDSCNT (UUART_BRGEN[9:8]) 来确定波特率分频

3. 写线控制寄存器UUART_LINECTL 和协议控制寄存器UUART_PROTCTL来配置传输数据格式和UART协议设置
   - 在 DWIDTH (UUART_LINECTL[11:8])中编程数据域长度
   - 通过设置 EVENPARITY (UUART_PROTCTL[2]) 和 PARITYEN (UUART_PROTCTL[1])来使能校验位和确定校验类型
   - 通过设置 STOPB (UUART_PROTCTL[0])配置停止位长度
   - UART协议下，使能 LSB (UUART_LINECTL[0])来选择先传输LSB
   - 设置 EDGEDET (UUART_DATIN0[4:3]) 为“0x2” 来选择下降沿作为接收起始位侦测
4. 设置 PROTEN (UUART_PROTCTL[31]) 为 1 来使能 UART 协议
5. 发送和接收数据
   - 写发送数据寄存器UUART_TXDAT 来发送数据
   - 等待 TXSTIF(USCIROTSTS[1]) 直到被置位，然后用户可以写入下次发送数据到 UUART_TXDAT
   - 当 TXENDIF(UUART_PROTSTS[2]) 被置位时，发送缓冲为空并且最后数据的停止位已经被发送。
   - 如果RXENDIF(UUART_PROTSTS[4])被置位时，数据帧已经完成。用户可以通过读接收数据寄存器UUART_RXDAT获得该数据。

6.18.6 寄存器映射

R: 只读, **W:**只写, **R/W:**读/写

寄存器	偏移	R/W	描述	复位值
USCI_UART基地址: **UUARTn_BA = 0x400D_0000 + (0x1000*n)** **n= 0, 1**				
UUART_CTL	UUARTn_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000
UUART_INTEN	UUARTn_BA+0x04	R/W	USCI 中断使能寄存器	0x0000_0000
UUART_BRGEN	UUARTn_BA+0x08	R/W	USCI波特率发生器寄存器	0x0000_3C00
UUART_DATIN0	UUARTn_BA+0x10	R/W	USCI输入数据信号配置寄存器0	0x0000_0000
UUART_CTLIN0	UUARTn_BA+0x20	R/W	USCI输入控制信号配置寄存器0	0x0000_0000
UUART_CLKIN	UUARTn_BA+0x28	R/W	USCI输入时钟信号配置寄存器	0x0000_0000
UUART_LINECTL	UUARTn_BA+0x2C	R/W	USCI线控制寄存器	0x0000_0000
UUART_TXDAT	UUARTn_BA+0x30	W	USCI发送数据寄存器	0x0000_0000
UUART_RXDAT	UUARTn_BA+0x34	R	USCI接收数据寄存器	0x0000_0000
UUART_BUFCTL	UUARTn_BA+0x38	R/W	USCI发送/接收缓冲控制寄存器	0x0000_0000
UUART_BUFSTS	UUARTn_BA+0x3C	R	USCI发送/接收缓冲状态寄存器	0x0000_0101
UUART_PDMACTL	UUARTn_BA+0x40	R/W	USCI PDMA控制寄存器	0x0000_0000
UUART_WKCTL	UUARTn_BA+0x54	R/W	USCI唤醒控制寄存器	0x0000_0000
UUART_WKSTS	UUARTn_BA+0x58	R/W	USCI唤醒状态寄存器	0x0000_0000
UUART_PROTCTL	UUARTn_BA+0x5C	R/W	USCI协议控制寄存器	0x0000_0000
UUART_PROTIEN	UUARTn_BA+0x60	R/W	USCI协议中断使能寄存器	0x0000_0000
UUART_PROTSTS	UUARTn_BA+0x64	R/W	USCI协议状态寄存器	0x0000_0000

6.18.7 寄存器描述

USCI控制寄存器(UUART_CTL)

寄存器	偏移	R/W	描述	复位值
UUART_CTL	UUARTn_BA+0x00	R/W	USCI控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**FUNMODE**	**功能模式** 该位是选择USCI控制器的协议。选择无效或保留的将禁用USCI。当两个协议之间切换时，在选择新协议前USCI必须先禁用。同时，在用户写000到FUNMODE，USCI会被复位。 000 = 禁用USCI，所有协议相关的状态机器被设置为空闲状态。 001 = 选择SPI 协议 010 = 选择UART 协议 100 = 选择I^2C 协议 其他保留

USCI中断使能寄存器(UUART_INTEN)

寄存器	偏移	R/W	描述	复位值
UUART_INTEN	UUARTn_BA+0x04	R/W	USCI中断使能寄存器	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[4]	**RXENDIEN**	**接收结束中断使能位** 该位是使能发生接收结束事件时候产生中断 0 =禁用接收结束中断 1 = 使能接收结束中断
[3]	**RXSTIEN**	**接收起始中断使能位** 该位是使能发生接收起始事件时候产生中断 0 =禁用接收起始中断 1 = 使能接收起始中断
[2]	**TXENDIEN**	**发送结束中断使能位** 该位是使能发生发送结束事件时候产生中断 0 =禁用发送结束中断 1 = 使能发送结束中断
[1]	**TXSTIEN**	**发送起始中断使能位** 该位是使能发生发送起始事件时候产生中断 0 =禁用发送起始中断 1 = 使能发送起始中断
[0]	**Reserved**	保留

USCI波特率发生器寄存器 (UUART_BRGEN)

寄存器	偏移	R/W	描述	复位值
UUART_BRGEN	UUARTn_BA+0x08	R/W	USCI波特率发生器寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						**CLKDIV**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLKDIV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DSCNT**					**PDSCNT**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TMCNTSRC**	**TMCNTEN**	**SPCLKSEL**		**PTCLKSEL**	**RCLKSEL**

位	描述	
[31:26]	**Reserved**	保留
[25:16]	**CLKDIV**	**时钟分频器** 该位是定义协议时钟频率f_{PROT_CLK}和时钟分频器频率f_{DIV_CLK} (f_{DIV_CLK}= f_{PROT_CLK}/ (CLKDIV+1)) 之间的比率 **注：**在UART功能下，当自动波特率功能(ABREN(UUART_PROTCTL[6])) 使能时，会在输入数据0x55的第四个下降沿时被硬件更新。修改值是在位5和位6之间的时间平均值。用户可以使用修改的CLKDIV和新的BRDETITV (UUART_PROTCTL[24:16])来计算精确的波特率
[15]	**Reserved**	保留
[14:10]	**DSCNT**	**采样计数器的分母** 该位定义采样时钟f_{SAMP_CLK} 的分频比。 分频f_{DS_CNT} = f_{PDS_CNT} / (DSCNT+1). **注：**在UART模式下，DSCNT最大值是0xF，建议设置4以上来保证接收数据的采样时正确的值。
[9:8]	**PDSCNT**	**采样计数器预分频器** 该位定义的是对来自采样时钟f_{SAMP_CLK}的分频比 分频f_{PDS_CNT} = f_{SAMP_CLK} / (PDSCNT+1).
[7:6]	**Reserved**	保留
[5]	**TMCNTSRC**	**时序测量计数器时钟源选择** 0 = 来自f_{PROT_CLK}. 1 =来自f_{DIV_CLK}.
[4]	**TMCNTEN**	**时序测量计数器使能位** 该位是使能10位时序测量计数器 0 =禁用时序测量计数器 1 =使能时序测量计数器

[3:2]	**SPCLKSEL**	**采样时钟源选择** 该域用于对协议采样时钟(f_{SAMP_CLK}) 的时钟源选择 00 = f_{SAMP_CLK}被选择到f_{DIV_CLK}. 01 = f_{SAMP_CLK}被选择到f_{PROT_CLK}. 10 = f_{SAMP_CLK}被选择到f_{SCLK}. 11 = f_{SAMP_CLK}被选择到f_{REF_CLK}.
[1]	**PTCLKSEL**	**协议时钟源选择** 该位用于选择协议时钟(f_{PROT_CLK})的源信号 0 = 参考时钟f_{REF_CLK}. 1 = f_{REF_CLK2} (频率是f_{REF_CLK}一半).
[0]	**RCLKSEL**	**参考时钟源选择** 该位是选择参考时钟(f_{REF_CLK})源信号 0 = 外设设备时钟f_{PCLK}. 1 = 保留

USCI输入数据信号配置寄存器0 (UUART_DATIN0)

寄存器	偏移	R/W	描述	复位值
UUART_DATIN0	UUARTn_BA+0x10	R/W	USCI输入数据信号配置寄存器0	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[4:3]	**EDGEDET**	**输入信号边沿侦测模式** 该域是选择输入信号数据信号边沿启用触发事件 00 =禁用启用触发事件 01 =上升沿启用输入数据信号触发事件 10 = 下降沿启用输入数据信号触发事件 11 = 双边沿启用输入数据信号触发事件 **注：**在UART功能模式，建议该域设置为0x2。
[2]	**ININV**	**输入信号反相选择** 该域是使能对输入异步信号的反相 0 =对异步输入信号不反相 1 = 对异步输入信号反相
[1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入信号同步选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入

USCI输入控制信号配置寄存器0 (UUART_CTLIN0)

寄存器	偏移	R/W	描述	复位值
UUART_CTLIN0	UUARTn_BA+0x20	R/W	USCI输入控制信号配置寄存器0	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**ININV**	**输入信号反相选择** 该域是定义使能对输入异步信号输入反相 0 =对异步输入信号不反相 1 = 对异步输入信号反相
[1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入同步信号选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入

USCI输入时钟信号配置寄存器(UUART_CLKIN)

寄存器	偏移	R/W	描述	复位值
UUART_CLKIN	UUARTn_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入同步信号选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入

USCI线控制寄存器(UUART_LINECTL)

寄存器	偏移	R/W	描述	复位值
UUART_LINECTL	UUARTn_BA+0x2C	R/W	USCI 线控制寄存器	0x0000_0000

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

位	描述	
[31:12]	**Reserved**	保留
[11:8]	**DWIDTH**	**传输的字长** 该域定义发送和接收的数据字长。在数据缓冲中数据总是右对齐。USCI支持4到16位的字长 0x0: 数据字包含位于位[15：0]的16位 0x1: 保留 0x2: 保留 0x3: 保留 0x4: 数据字包含位于位[3：0]的4位 0x5: 数据字包含位于位[4：0]的5位 ... 0xF: 数据字包含位于位[14：0]的15位 **注：**在UART协议中，长度可以配置为6∽13位。
[7]	**CTLOINV**	该位定义了内部控制信号和输出控制信号之间的关系 0 = 无影响 1 =在输出前控制信号被反相 **注：**在UART模式下，控制信号就是nRTS信号
[6]	**Reserved**	保留
[5]	**DATOINV**	**数据输出反相选择** 该位定义了内部移位数据值和USCIx DAT1引脚输出数据信号之间的关系 0 = USCIx_DAT1的值等于数据移位寄存器 1 = USCIx_DAT1的值为数据移位寄存器的反相
[4:1]	**Reserved**	保留
[0]	**LSB**	**LSB优先传输选择** 0 =MSB，哪一位先发送/接收，取决于DWIDTH的设置

		1 =LSB，数据缓冲的位0被先发送/接收

USCI发送数据寄存器(UUART_TXDAT)

寄存器	偏移	R/W	描述	复位值
UUART_TXDAT	UUARTn_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**TXDAT**	发送数据 软件可以写16位发送数据到该域用来发送

USCI接收数据寄存器(UUART_RXDAT)

寄存器	偏移	R/W	描述	复位值
UUART_RXDAT	UUARTn_BA+0x34	R	USCI接收数据寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**RXDAT**	**接收的数据** 该域值监控存储在接收数据缓冲的接收到的数据 **注：** RXDAT[15:13] 表明BREAK, FRMERR 和 PARITYERR (USCI_PROTSTS[7:5])相同帧状态

USCI发送/接收缓冲控制寄存器(UUART_BUFCTL)

寄存器	偏移	R/W	描述	复位值
UUART_BUFCTL	UUARTn_BA+0x38	R/W	USCI发送/接收缓冲控制寄存器	0x0000_0000

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

位	描述	
[31:18]	**Reserved**	保留
[17]	**RXRST**	**接收复位** 0 = 无影响 1 =复位接收相关计数器、状态机、接收移位寄存器的内容和数据缓冲。 **注1:** 1个PCLK周期后自动被清除 **注2:**在该位被置1前，建议检查RXBUSY (UUART_PROTSTS[10])
[16]	**TXRST**	**发送复位** 0 = 无影响 1 =复位发送相关计数器、状态机、发送移位寄存器的内容和数据缓冲。 **注：** 1个PCLK周期后自动被清除
[15]	**RXCLR**	**清接收缓冲** 0 = 无影响 1 = 接收缓冲被清除（填充被清除并且输出指针被设置为输入指针的值）仅当缓冲没有参与数据通信时被使用。 **注：** 1个PCLK周期后自动被清除
[14]	**RXOVIEN**	**接收缓冲溢出错误中断使能控制** 0 =禁用接收溢出中断 1 =使能接收溢出中断
[13:8]	**Reserved**	保留
[7]	**TXCLR**	**清发送缓冲** 0 = 无影响 1 = 发送缓冲被清除（填充被清除并且输出指针被设置为输入指针的值）仅当缓冲没有参与数据通信时被使用。 **注：** 1个PCLK周期后自动被清除

[6:0]	**Reserved**	保留

USCI发送/接收缓冲状态寄存器(UUART_BUFSTS)

寄存器	偏移	R/W	描述	复位值
UUART_BUFSTS	UUARTn_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101

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

位	描述	
[31:10]	**Reserved**	保留
[9]	**TXFULL**	**发送缓冲满指示** 0 =发送缓冲还没满 1 =发送缓冲满
[8]	**TXEMPTY**	**发送缓冲空指示（只读）** 0 =发送缓冲非空 1 =发送缓冲为空
[7:4]	**Reserved**	保留
[3]	**RXOVIF**	**接收缓冲溢出错误中断状态** 该位表示接收缓冲溢出错误事件被侦测到。如果RXOVIEN (UUART_BUFCTL[14])使能，会启用相应的中断请求 0 = 接收缓冲溢出错误事件没有被侦测到 1 = 接收缓冲溢出错误事件被侦测到 **注：**软件写1清除该位
[2]	**Reserved**	保留
[1]	**RXFULL**	**接收缓冲满指示** 0 =接收缓冲还没满 1 =接收缓冲满
[0]	**RXEMPTY**	**接收缓冲空指示** 0 =接收缓冲非空 1 =接收缓冲为空

USCI PDMA 控制寄存器(UUART_PDMACTL)

寄存器	偏移	R/W	描述	复位值
UUART_PDMACTL	UUARTn_BA+0x40	R/W	USCI PDMA 控制寄存器	0x0000_0000

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

位	描述	
[31:4]	**Reserved**	保留
[3]	**PDMAEN**	**PDMA模式使能位** 0 = 禁用PDMA功能 1 = 使能PDMA功能
[2]	**RXPDMAEN**	**PDMA接收通道有效** 0 = 禁用PDMA接收功能 1 = 使能PDMA接收功能
[1]	**TXPDMAEN**	**PDMA发送通道有效** 0 = 禁用PDMA发送功能 1 = 使能PDMA发送功能
[0]	**PDMARST**	**PDMA 复位** 0 = 无影响 1 =复位USCI的PDMA控制逻辑。该位会自动地被清0

USCI唤醒控制寄存器(UUART_WKCTL)

寄存器	偏移	R/W	描述	复位值
UUART_WKCTL	UUARTn_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**PDBOPT**	**掉电阻塞选择** 0 = 如果用户试图在协议正在传输的时候执行WFI来进入掉电模式，MCU会停止传输并且立即进入掉电模式 1 =如果用户试图在协议正在传输的时候执行WFI来进入掉电模式，传输会继续进行并且MCU立即进入空闲模式
[1]	**Reserved**	保留
[0]	**WKEN**	**唤醒使能位** 0 =禁用唤醒功能 1 =使能唤醒功能

USCI唤醒状态寄存器(UUART_WKSTS)

寄存器	偏移	R/W	描述	复位值
UUART_WKSTS	UUARTn_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**WKF**	**唤醒标志** 当芯片被从掉电模式唤醒，该位置1.软件写1清除该位。

USCI协议控制寄存器– UART (UUART_PROTCTL)

寄存器	偏移	R/W	描述	复位值
UUART_PROTCTL	UUARTn_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	**Reserved**	**BCEN**	**Reserved**	**Reserved**	**STICKEN**	**Reserved**	**BRDETITV**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BRDETITV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**WAKECNT**				**CTSWKEN**	**DATWKEN**	**Reserved**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**ABREN**	**RTSAUDIREN**	**CTSAUTOEN**	**RTSAUTOEN**	**EVENPARITY**	**PARITYEN**	**STOPB**

位	描述	
[31]	**PROTEN**	**UART协议使能位** 0 = 禁用UART 协议 1 = 使能UART 协议
[30]	**Reserved**	保留
[29]	**BCEN**	**发送 Break 控制使能位** 0 =禁用发送Break控制 1 =使能发送Break控制 **注：**当该位被置1，串行数据输出（TX）被强制为逻辑0状态。该位仅在TX线上有效，对发送逻辑单元没有影响
[27]	**Reserved**	保留
[26]	**STICKEN**	**Stick 校验使能位** 0 =禁用Stick校验 1 =使能Stick校验 **注：**详细信息请参考RS-485章节
[25]	**Reserved**	保留
[24:16]	**BRDETITV**	**波特率侦测间隔** 该位指明从机在一个位时间内计算波特率需要多少时钟周期（该时钟由TMCNTSRC (UUART_BRGEN [5])决定）。总线的顺序应该是1和0这样交替（比如，输入数据格式应该是0x55）。用户可以读该值去了解当前总线输入波特率，不管ABRDETIF (USCI_PROTSTSL[9])何时置位。 **注：**软件写0到BRDETITV.清除为0。
[15]	**Reserved**	保留
[14:11]	**WAKECNT**	**唤醒计数器** 该位指明当设备被从掉电模式下唤醒时从机获得第一个位及起始位需要多少个时钟周期（该

		时钟是f_{PDS_CNT}）。
[10]	**CTSWKEN**	**nCTS唤醒模式使能位** 0 = 禁用nCTS 唤醒模式 1 =使能 nCTS唤醒模式
[9]	**DATWKEN**	**数据唤醒模式使能位** 0 = 禁用数据唤醒模式 1 =使能数据唤醒模式
[6]	**ABREN**	**自动波特率侦测使能位** 0 =禁用自动波特率侦测 1 =使能自动波特率侦测 **注：**当自动波特率侦测操作结束时硬件清除该位。如果ARBIEN (UUART_PROTIEN [1])使能，将产生相关中断ABRDETIF (UUART_PROSTS[9])
[5]	**RTSAUDIREN**	**nRTS自动方向使能位** 当nRTS自动方向使能，如果TX缓冲为空，UART自动启用nRTS信号 0 =禁用nRTS自动方向控制 1 =使能nRTS自动方向控制 **注 1:**该位是用在RS485下nRTS自动方向控制 **注 2:**该位仅在RTSAUTOEN没有置位情况下有作用
[4]	**CTSAUTOEN**	**nCTS自动流控使能位** 当nCTS自动流控使能时，当nCTS输入启用时UART会发送数据到外部设备（如nCTS输入没有启用UART不会发送数据到设备） 0 =禁用 nCTS 自动流控 1 = 使能nCTS 自动流控
[3]	**RTSAUTOEN**	**nRTS 自动流控使能位** 当nRTS自动流控使能时，如果接收缓冲没有满(RXFULL (UUART_BUFSTS[1] =1)，UART会解除nRTS信号。 0 =禁用 nRTS 自动流控 1 = 使能nRTS 自动流控 **注 :**该位仅在RTSAUDIREN没有置位情况下有作用
[2]	**EVENPARITY**	**偶校验使能位** 0 =发送和检测每一个字中逻辑1是奇数。 1 =发送和检测每一个字中逻辑1是偶数。 **注：**仅当PARITYEN置位时该位有作用
[1]	**PARITYEN**	**校验使能位** 在UART帧中，该位定义校验位使能 0 =禁用校验位 1 =使能校验位
[0]	**STOPB**	**停止位** 在UART帧中，该位定义停止位数目 0 = 1位停止位 1 = 2位停止位

USCI协议中断使能寄存器– UART (UUART_PROTIEN)

寄存器	偏移	R/W	描述	复位值
UUART_PROTIEN	UUARTn_BA+0x60	R/W	USCI 协议中断使能寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**RLSIEN**	**接收线状态中断使能** 0 =禁用接收线状态中断 1 =使能接收线状态中断 **注：** UUART_PROTSTS[7:5]反映当前接收线状态中断事件
[1]	**ABRIEN**	**自动波特率中断使能位** 0 =禁用自动波特率中断 1 =使能自动波特率中断
[0]	**Reserved**	保留

USCI协议状态寄存器 – UART (UUART_PROTSTS)

寄存器	偏移	R/W	描述	复位值
UUART_PROTSTS	UUARTn_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**CTSLV**	**CTSSYNCLV**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**ABERRSTS**	**RXBUSY**	**ABRDETIF**	**Reserved**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BREAK	**FRMERR**	**PARITYERR**	**RXENDIF**	**RXSTIF**	**TXENDIF**	**TXSTIF**	**Reserved**

位	描述	
[31:18]	**Reserved**	保留
[17]	**CTSLV**	**nCTS 引脚状态 (只读)** 该位被用作监控nCTS引脚输入的当前状态 0 = nCTS引脚为低电平 1 = nCTS引脚为高电平
[16]	**CTSSYNCLV**	**nCTS 同步电平状态 (只读)** 该位被用作反映内部同步nCTS信号当前状态 0 = 内部同步 nCTS 为低 1 =内部同步 nCTS 为高
[15:12]	**Reserved**	保留
[11]	**ABERRSTS**	**自动波特率错误状态** 当自动波特率侦测计数器溢出时该位被置位。当自动波特率计数器溢出时，用户需要修改CLKDIV (UUART_BRGEN[25:16])值并且使能ABREN (UUART_PROTCTL[6])来再次侦测正确的波特率 0 =自动波特率计数器没有溢出 1 =自动波特率计数器溢出 **注 1:**.该位在ABRDETIF相同时间被置位。 **注 2:** 写1到ABRDETIF 或 ABERRSTS，清除该位
[10]	**RXBUSY**	**RX 总线状态标志 (只读)** 该位反映接收器忙状态 0 = 接收器处于空闲状态. 1 =接收器处于忙状态.

[9]	**ABRDETIF**	**自动波特率中断标志** 当自动波特率侦测在输入数据下降沿中间完成时，该位被置位。如果ABRIEN (UUART_PROTCTL[6])置位，会产生自动波特率中断。当输入数据为0x55时，该位会被设置4次并且在输入总线下次下降沿之前被清除。 0 =自动波特率功能没有完成 1 =自动波特率功能已完成 **注：**该位可以写1清除
[8]	**Reserved**	保留
[7]	**BREAK**	**Break 标志** 每当接收数据输入（RX）保持在逻辑0状态，该状态持续时间超过了一整个字的传输（也就是：起始位+数据位+停止位的时间），该位被置1. 0 =没有Break产生 1 =接收总线上出现Break状态 **注：**写1到BREAK, FRMERR 和 PARITYERR中该位被清除
[6]	**FRMERR**	**帧错误标志** 每当接收字符没有有效的停止位时，该位被置1.（也就是，仅更随在最后数据位或是校验位之后的停止位被侦测到是逻辑0） 0 =没有帧错误产生 1 =有帧错误发生 **注：**写1到BREAK, FRMERR 和 PARITYERR中该位被清除
[5]	**PARITYERR**	**校验错误标志** 每当接收字符没有有效的校验位时，该位被置1. 0 =没有校验错误 1 = 有校验错误发生 **注：**写1到BREAK, FRMERR 和 PARITYERR中该位被清除
[4]	**RXENDIF**	**接收结束中断标志** 0 =没有接收结束中断状态发生 1 = 有接收结束中断状态发生 **注：**写1清除该位
[3]	**RXSTIF**	**接收起始中断标志** 0 =没有接收起始中断状态发生 1 = 有接收起始中断状态发生 **注：**写1清除该位
[2]	**TXENDIF**	**发送结束中断标志** 0 =没有发送结束中断状态发生 1 = 有发送结束中断状态发生 **注：**写1清除该位
[1]	**TXSTIF**	**发送起始中断标志** 0 =没有发送起始中断状态发生 1 = 有发送起始中断状态发生 **注1:**写1清除该位 **注 2:**当没有数据在发送缓冲时用户可以用此标志位来加载下次发送数据
[0]	**Reserved**	保留

6.19 USCI - SPI模式

6.19.1 概述

USCI控制器的SPI协议适用于同步串行数据通信并且支持全双工传输。支持主机和从机操作4线接口模式。USCI控制器的SPI模式在接收外设数据时执行的是串行到并行的转换，在发送数据到外设时执行的是并行到串行的转换。设置FUNMODE (USPI_CTL[2:0]) = 0x1，选择SPI模式。

通过设置SLAVE (USPI_PROTCTL[0])来选择SPI协议运行在主机或是从机模式。下图展示了主机和从机模式的应用。

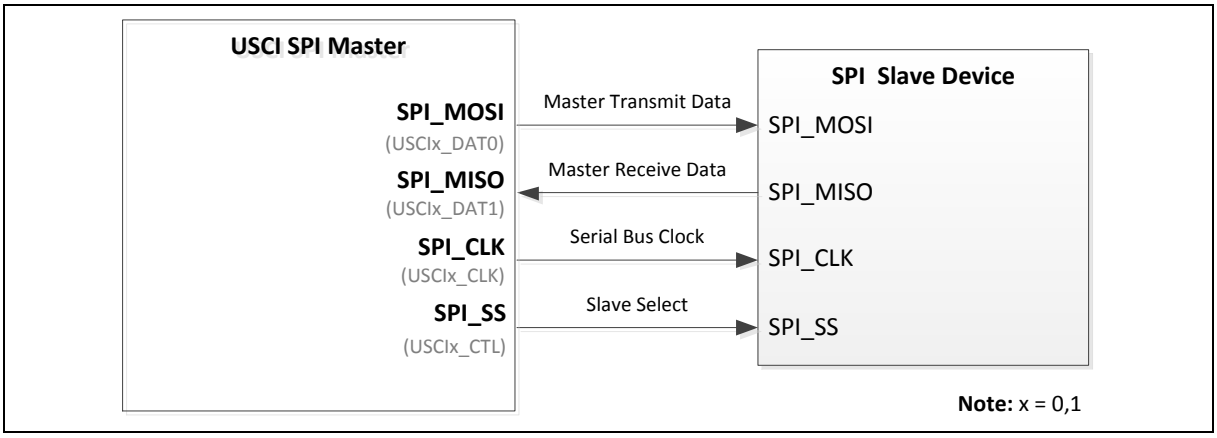


图 6.19-1SPI 主机模式应用框图

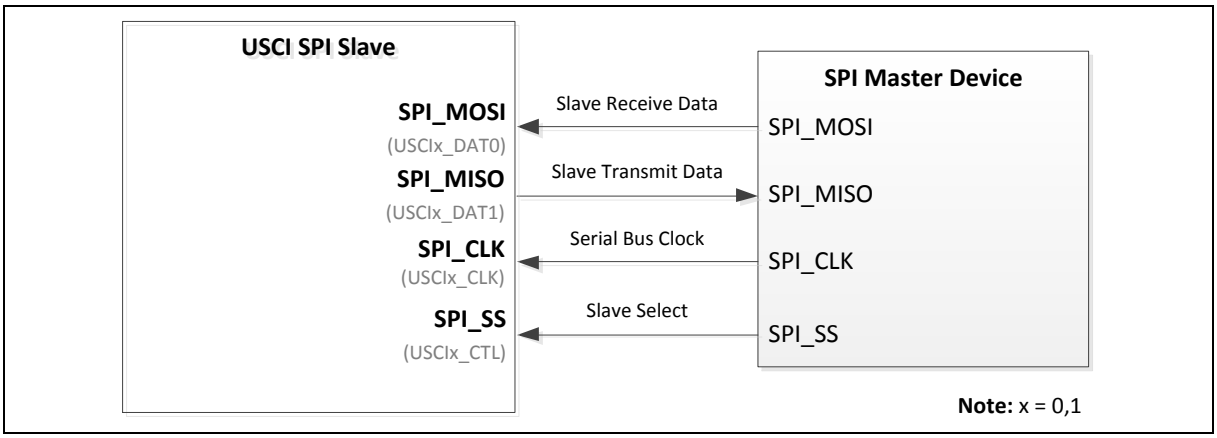


图 6.19-2SPI 从机模式应用框图

6.19.2 特性

- 支持主机或从机模式操作（最大频率—主机== f_{PCLK} / 2, 从机<f_{PCLK} / 5）
- 传输字的长度在4到16位可编程
- 支持一组发送缓冲和两组接收缓冲
- 支持MSB优先或是LSB优先传输
- 支持字传输暂停功能

- 支持PDMA传输
- 支持3-线，没有从机选择信号
- 从机模式下支持片选信号唤醒功能
- 支持单数据通道的半双工传输

6.19.3 框图

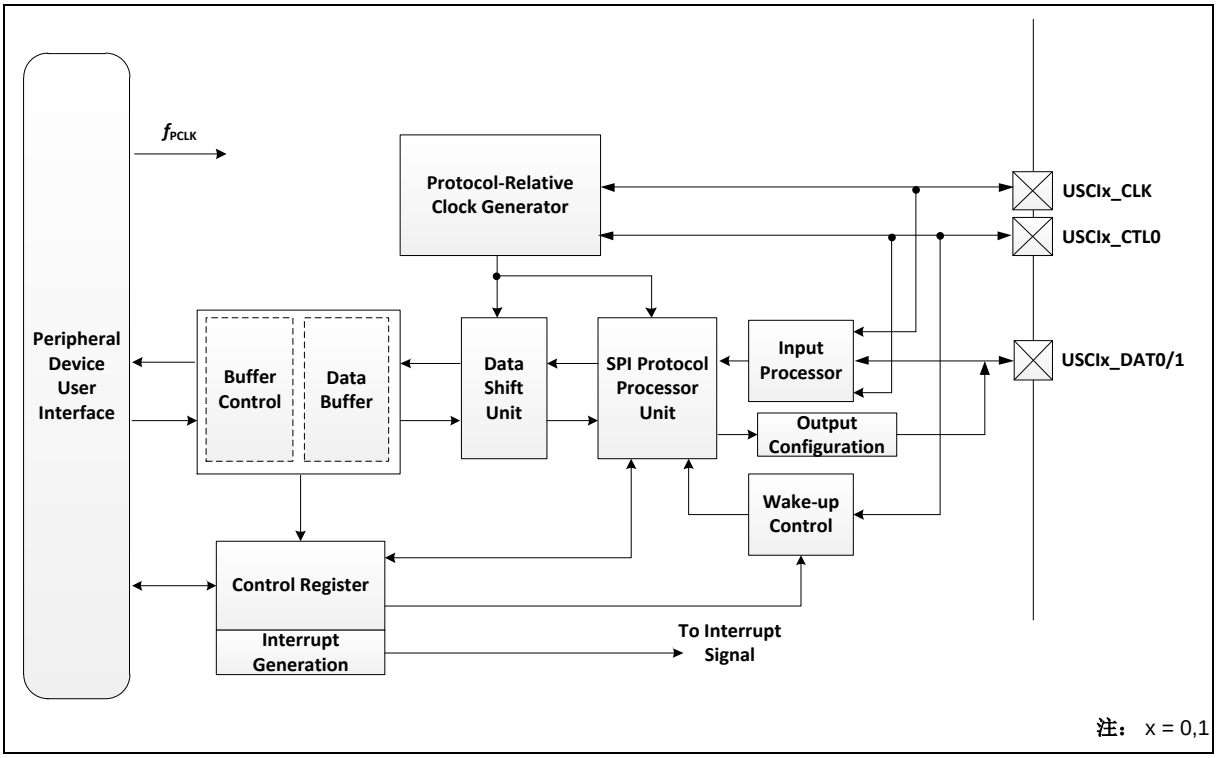


图 6.19-3USCI SPI 模式框图

6.19.4 基本配置

6.19.4.1 USCI0 SPI基本配置

- 时钟源配置
  - 在USCI0CKEN（CLK_APBCLK1[8]）使能USCI0时钟
  - 在USPI_CTL[2:0]寄存器使能USCI0_SPI功能, USPI_CTL[2:0]=3’b001
- 复位配置
  - 在USCI0RST（SYS_IPRST2[8]）复位USCI0控制器

6.19.4.2 USCI1 SPI基本配置

- 时钟源配置
  - 在USCI1CKEN (CLK_APBCLK1[9])使能USCI1外设时钟
  - 在USPI_CTL[2:0]寄存器使能USCI1_SPI功能, USPI_CTL[2:0]=3’b001
- 复位配置

- 在USCI1RST (SYS_IPRST2[9])复位USCI1控制器

6.19.5 功能描述

6.19.5.1 USCI共同功能描述

具体信息参考6.17.4章节。

6.19.5.2 信号描述

工作在主机模式下的设备控制一次数据传输的开始和结束，并且SPI总线时钟和片选信号也是由主机设备产生。片选信号反映了一次数据的起始和结束，主机可以使用它使能对从机的发送或是接收操作。SPI通信信号如下：

SPI 模式	接收数据	发送数据	串行总线时钟	从机选择
全双工 SPI 主机	SPI_MISO (USCIx_DAT1)	SPI_MOSI (USCIx_DAT0)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)
全双工 SPI 从机	SPI_MOSI (USCIx_DAT0)	SPI_MISO (USCIx_DAT1)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)
半双工 SPI 主机/从机	SPI_MOSI (USCIx_DAT0)	SPI_MOSI (USCIx_DAT0)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)

SPI 通信信号

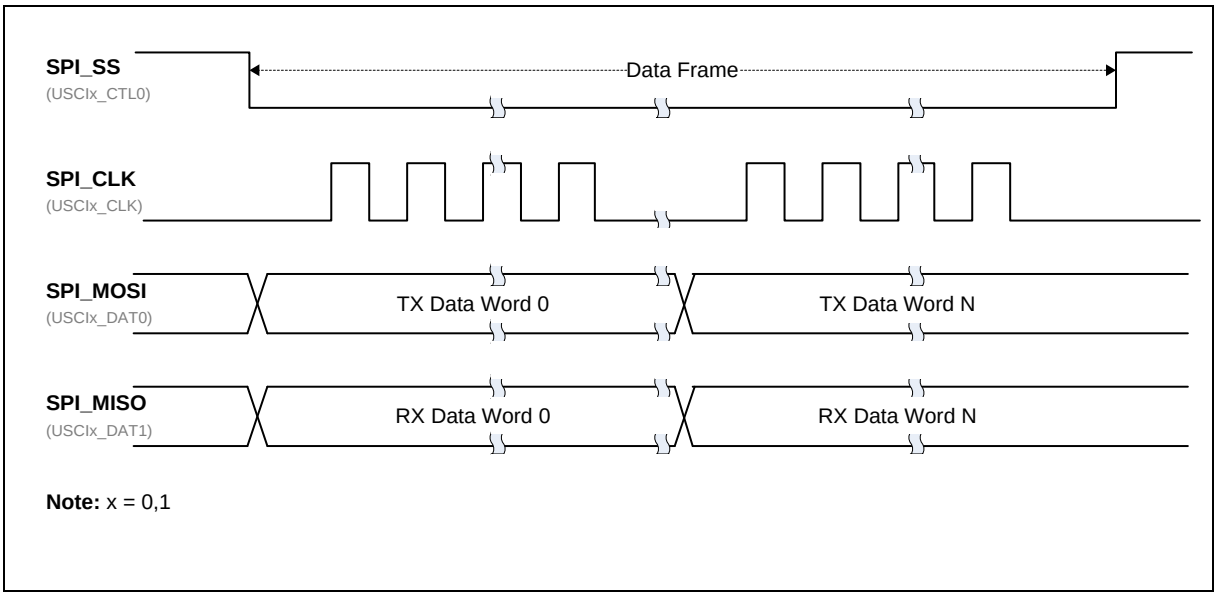


图 6.19-44-线全双工 SPI 通信信号 (主机模式)

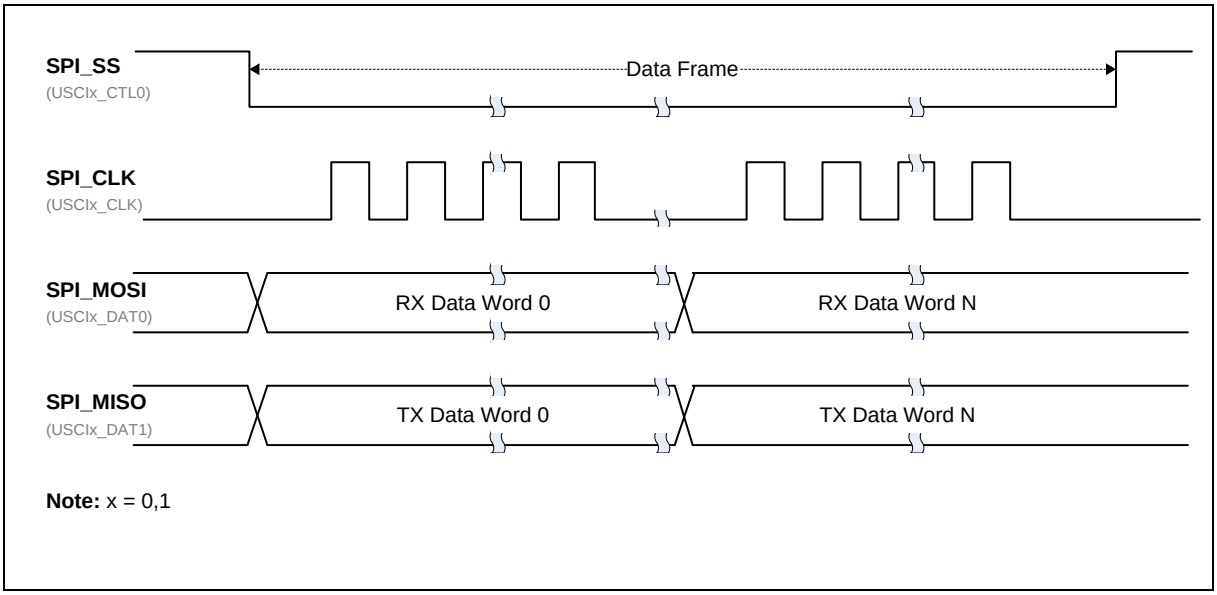


图 6.19-54 线全双工 SPI 通信信号 (从机模式)

6.19.5.3 串行总线时钟配置

USCI控制器需要外设时钟驱动USCI的逻辑单元执行数据传输。外设时钟频率等于PCLK频率。

在主机模式下，SPI总线时钟频率是由相关协议时钟发生器决定的。SPI总线时钟就是SPI时钟。SPI时钟频率是f_{SAMP_CLK}的一半，f_{SAMP_CLK}是由SPCLKSEL (USPI_BRGEN[3:2])决定的。相关协议时钟发生器的详细内容请参考参6.17.4。

在从机模式下，SPI总线时钟由外部主机提供；外设时钟频率f_{PCLK}必须是SPI总线时钟频率的5倍以上（即SPI总线时钟频率<1/5外设f_{PCLK}时钟频率）

在SPI协议下，SCLKMODE (USPI_PROTCTL[7:6])定义了总线的空闲状态及收发数据时时钟边沿的变化。主从机要有相同的SCLKMODE设定。串行总线时钟的四种模式配置如下：

SCLKMODE [1:0]	SPI 时钟空闲状态	发送时序	接收时序
0x0	低	下降沿	上升沿
0x1	低	上升沿	下降沿
0x2	高	上升沿	下降沿
0x3	高	下降沿	上升沿

表 6.19-1 串行总线时钟配置

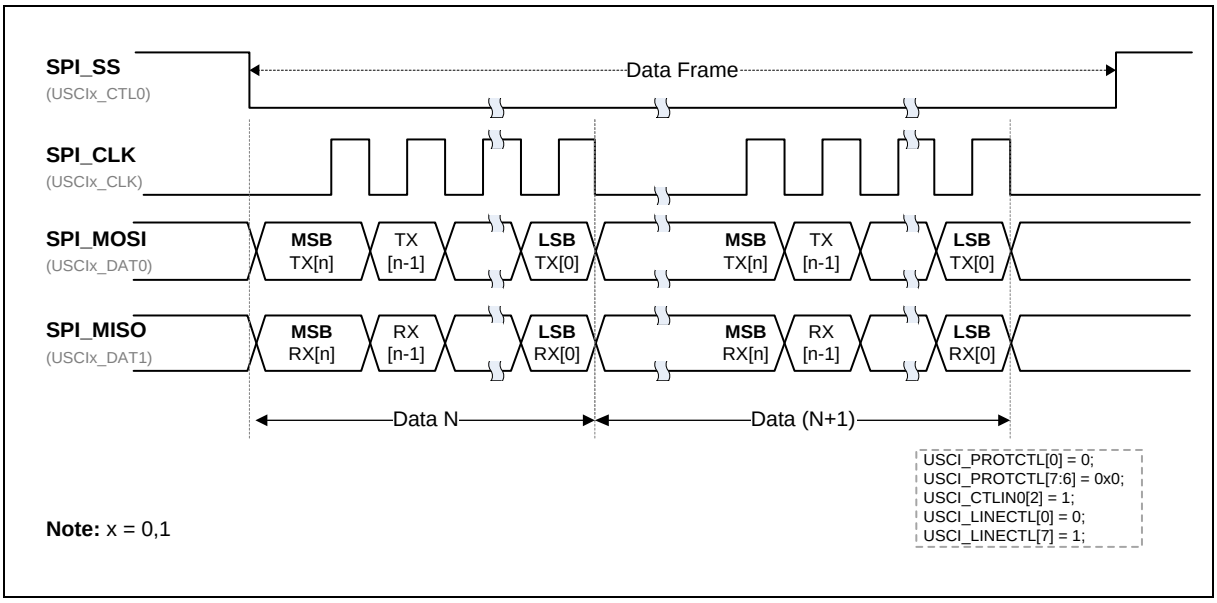


图 6.19-6 不同时钟模式下的 SPI 通信(SCLKMODE=0x0)

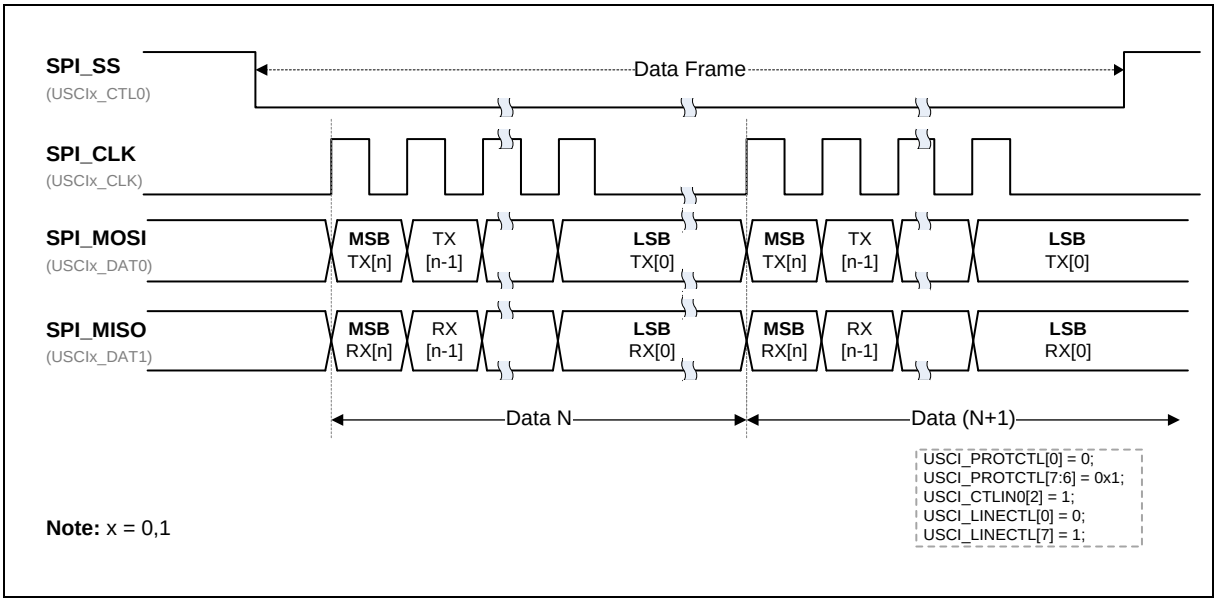


图 6.19-7 不同时钟模式下的 SPI 通信(SCLKMODE=0x1)

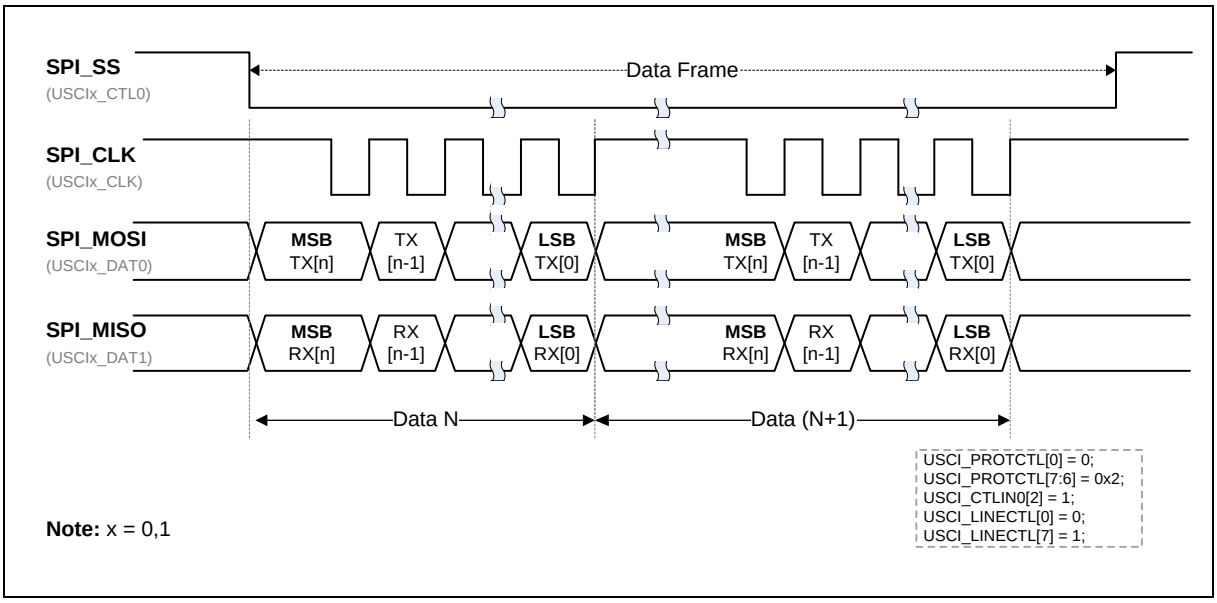


图 6.19-8 不同时钟模式下的 SPI 通信(SCLKMODE=0x2)

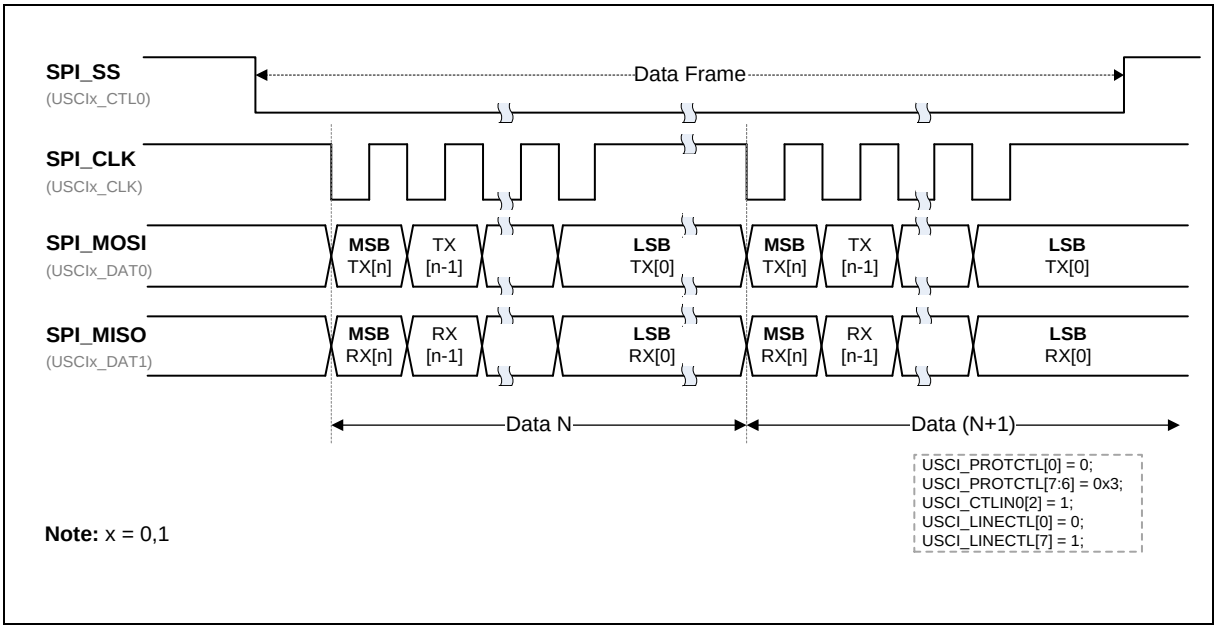


图 6.19-9 不同时钟模式下的 SPI 通信(SCLKMODE=0x3)

6.19.5.4 从机选择信号

SPI协议的从机片选信号默认是高电平启用。在SPI主机模式下，USCI控制器可以通过从机选择引脚SPI_SS (USCIx_CTL0)来驱动控制外部SPI 从机设备。在SPI从机模式下，接收片选信号可以通过ININV (USPI_CTLIN0[2])反相。

如果外部SPI主机设备的片选信号是低电平有效，那么从机设备的ININV (USPI_CTLIN0[2])的设置应该设置为1来反相输入控制信号。如果USCI工作在SPI主机模式下，对于那些片选信号低电平有效的外部SPI从机设备来说，输出从机选择反相CTLOINV (USPI_LINECTL[7])也要置1。

从机选择有效边沿与第一个SPI时钟输入边沿之间要超过2个USCI外设时钟周期。

为了正确地检测到帧结束，在连续两帧之间SPI从机输入片选信号必须保持无效状态时间至少在2个USCI外设时钟周期。

6.19.5.5 发送和接收数据

在USCI控制器SPI协议下，接收/发送数据位长度可以在DWIDTH (USPI_LINECTL[11:8])中定义，在SPI通信中发送和接收数据长度最大可以配置为16位。

LSB 位 (USPI_LINECTL[0])定义传输数据位的顺序。如果LSB位被置1，那么传输的数据顺序是LSB优先。如果LSB位被清0，那么传输的数据顺序是MSB优先。

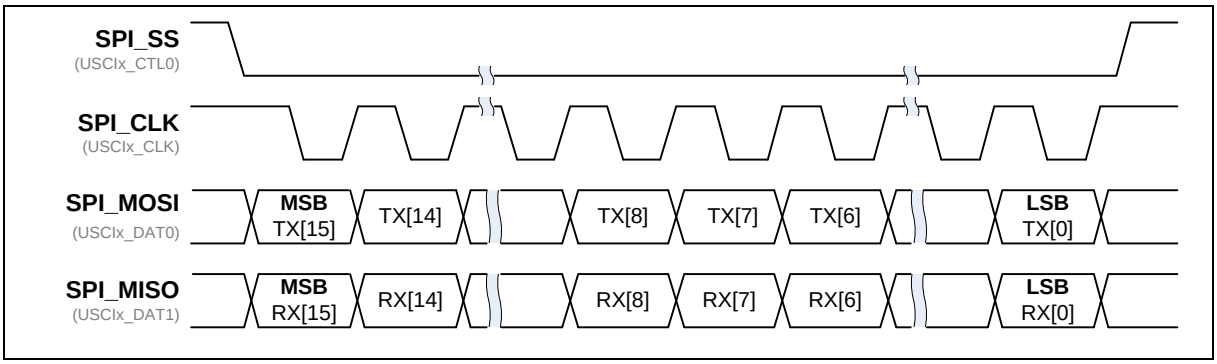


图 6.19-10 MSB 优先格式下 16 位数据长度的一次传输

6.19.5.6 字暂停

主机模式下两个连续字传输之间可以由SUSPITV (USPI_PROTCTL[11:8])提供一个0.5 ~ 15.5 SPI时钟周期可编程的暂停间隔的功能。暂停间隔指的是之前传输的最后一个时钟边沿到下次传输的第一个时钟边沿之间的间隔。SUSPITV (USPI_PROTCTL[11:8])默认值是0x3(3.5 SPI 时钟周期)。

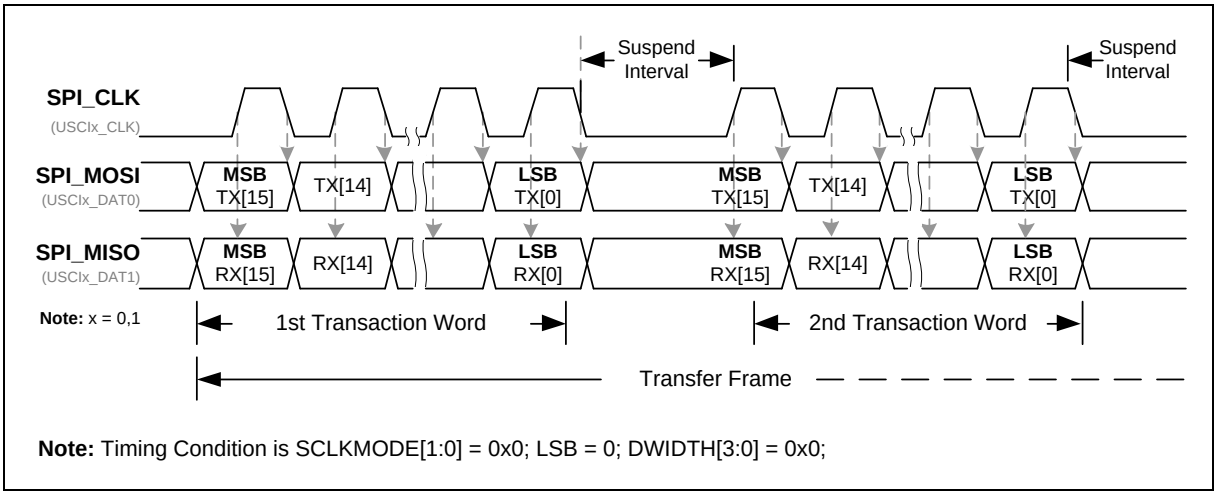


图 6.19-11 两组传输字间的暂停

6.19.5.7 自动片选功能

AUTOSS (USCI_PROTCTL[3]) 被用作SPI主机模式来使能自动片选功能。如果AUTOSS (USPI_PROTCTL[3])被置位，片选信号会自动地产生并且SS (USPI_PROTCTL[2])设置值不会影响输出从机选择USCIx_CTL0线。也就意味着当向发送缓冲写入数据启动SPI数据传输的时候，片选信号会变成有效状态。在所有传输结束或是如果SUSPITV (USPI_PROTCTL[11:8])大于等于3时，一次字传输完

成之后，片选信号会失效。

如果AUTOSS 位 (USPI_PROTCTL[3])被清除，通过设置/清除SS (USPI_PROTCTL[2])来使片选信号USCIx_CTL0引脚有效/无效。内部片选信号启用电平是高，CTLOINV (USPI_LINECTL[7])可以被用作反相片选信号。

在SPI主机模式，如果SUSPITV (USPI_PROTCTL[11:8])值少于3并且AUTOSS (USPI_PROTCTL[3])被置1，在连续两字传输之间片选信号保持有效状态。

在SPI从机模式，在两组连续传输之间为了识别片选信号的无效状态，接收片选信号无效的周期必须大于2个外设时钟周期。

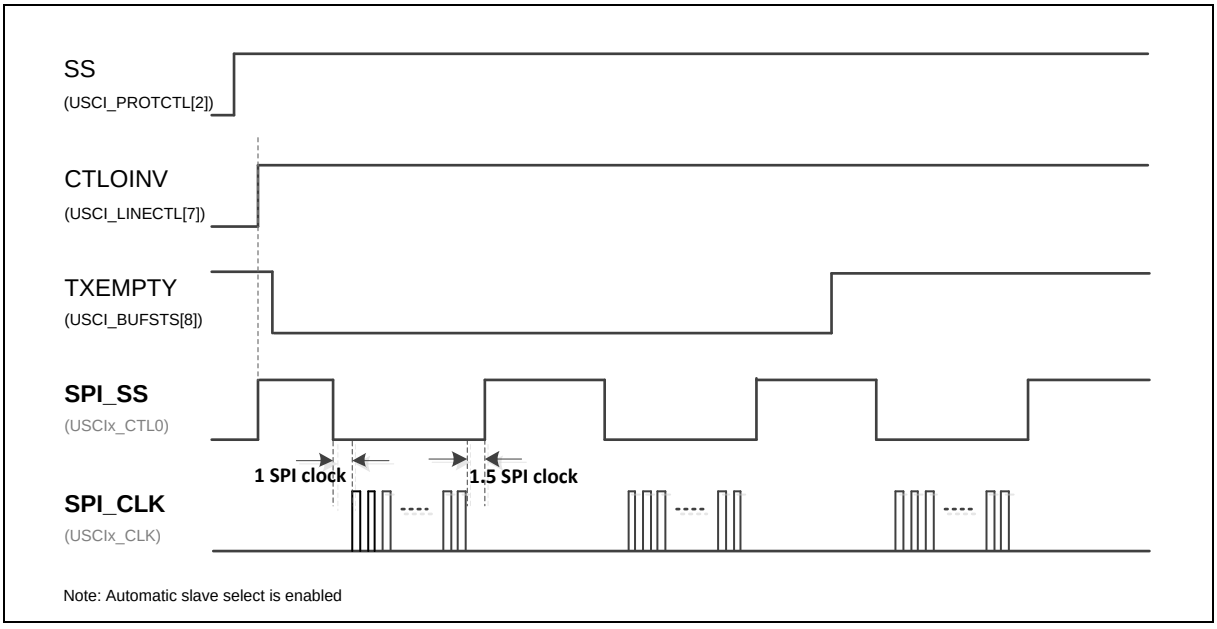


图 6.19-12 自动从机选择(SUSPITV ≧ 0x3)

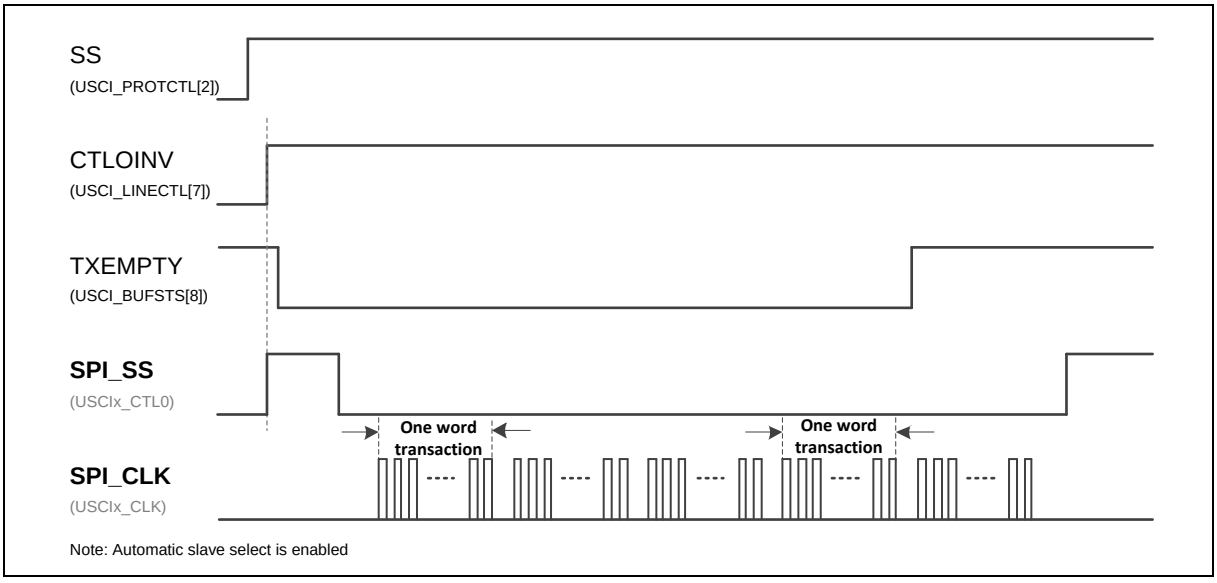


图 6.19-13 自动从机选择(SUSPITV < 0x3)

6.19.5.8 从机3线模式

当通过软件设置SLV3WIRE (USPI_PROTCTL[1])使能从机3线模式的时候，USCI SPI通信在从机模式下可以在没有片选信号下工作。仅三个引脚SPI_CLK (USCIx_CLK), SPI_MOSI (USCIx_DAT0), 和SPI_MISO (USCIx_DAT1)被用作与SPI主机通信。当SLV3WIRE (USPI_PROTCTL[1])被置1时，在通过设置FUNMODE(USPI_CTL [2:0])为0x1使能SPI协议之后SPI从机就准备好发送/接收数据。

6.19.5.9 数据传输模式

USCI控制器支持全双工SPI传输和单数据通道半双工SPI传输。

- 全双工SPI传输

  在全双工SPI传输下，有两根数据引脚。一根用来发送数据，另外一根用来接收数据。因此数据发送和接收可以同时进行。

  SCLKMODE (USPI_PROTCTL[7:6])定义了数据输出信号USCIx_DAT0的发送时序；根据寄存器的设定，数据会在时钟上升沿或下降沿变化；数据脚（USCIx_DAT0）输出的电平会保持在最后一个数据的最后一位状态，直到下一个数据时钟沿变化时才会更改

- 单数据通道半双工SPI传输

  在单数据通道半双工SPI传输下，只有一根数据引脚用于数据传输。因此数据发送和接收不能同时进行。数据移位方向由PORTDIR (USPI_TXDAT[16])决定。更多信息参考寄存器描述。

  单数据通道半双工SPI传输类似于全双工SPI协议。所有传输数据时序等同全双工SPI传输。

  图6.19-14是一个输出数据通道和一个输入数据通道于外部设备进行半双工传输的框图。

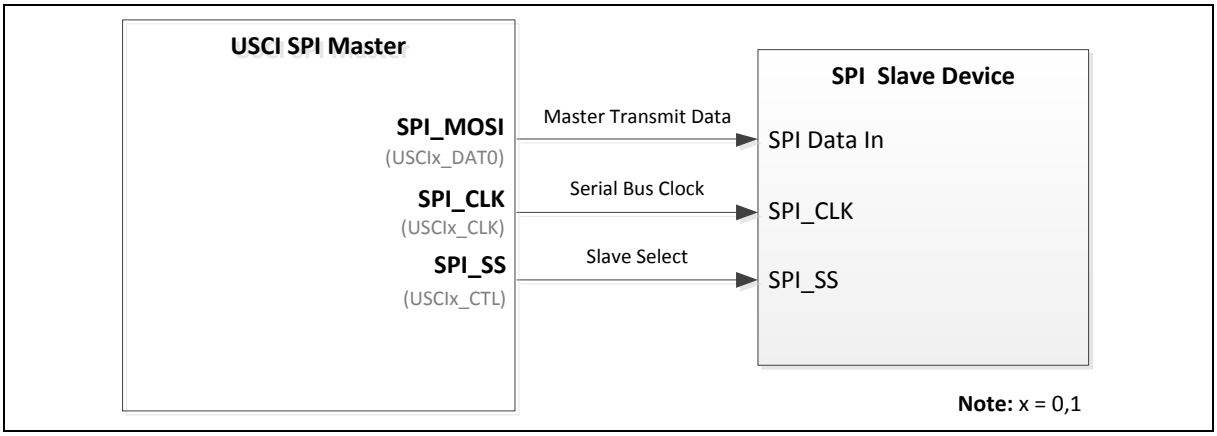


图 6.19-14 一个输出数据通道半双工通信 (SPI 主机模式)

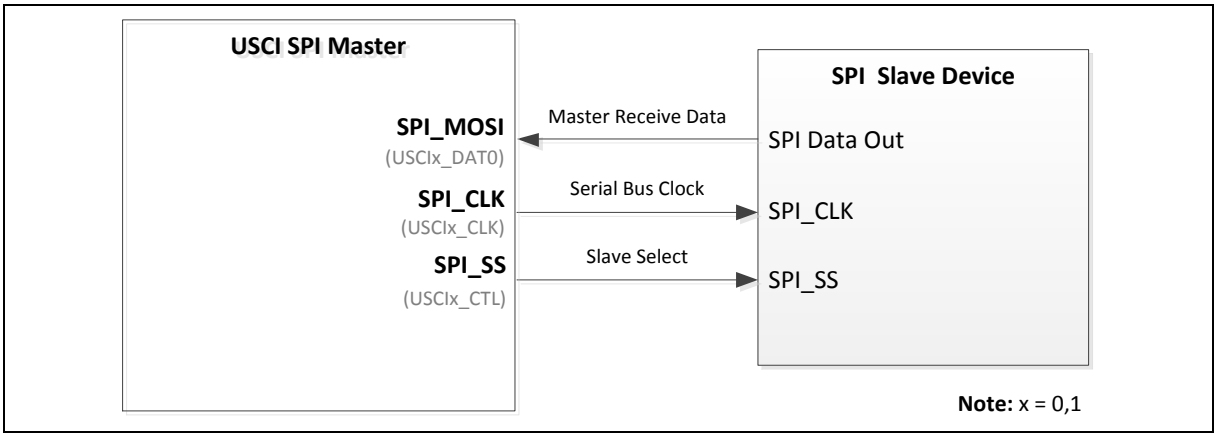


图 6.19-15 一个输入数据通道半双工通信(SPI 主机模式)

单数据通道半双工传输模式是通过设置TSMSEL[2:0] (USPI_PROTCTL[14:12])和PORTDIR (USPI_TXDAT[16])来配置的。当TSMSEL (USPI_PROTCTL[14:12])设置为0x4时，单数据通道半双工传输模式被选择。PORTDIR (USPI_TXDAT[16])被用作定义相应传输数据的方向。当PORTDIR位被设置为0时，USCI控制器发送相应数据到外部SPI设备。当PORTDIR位被设置为1时，USCI控制器从外部SPI设备读相应的数据。

比如，在单数据通道半双工传输模式下，如果PORTDIR=0，USCI SPI用USCIx_DAT0引脚发送数据；如果PORTDIR=1，USCI SPI用USCIx_DAT0引脚接收数据。

6.19.5.10 中断

数据传输中断

- 发送起始中断

  一次数据发送的第一个数据位的起始之后，中断事件TXSTIF (USPI_PROTSTS[1])置位。该位可以写1清除。

- 发送结束中断

  存储在发送缓冲的最后发送数据的最后数据位结束之后，中断事件TXENDIF (USPI_PROTSTS[2])置位。该位可以写1清除。

- 接收起始中断

  一次接收数据的第一个数据位的起始之后，中断事件RXSTIF (USPI_PROTSTS[3])置位。该位可以写1清除。

- 接收结束中断

  接收数据的最后数据位结束之后，中断事件RXENDIF (USPI_PROTSTS[4])置位。该位可以写1清除。

相关协议中断

- SPI从机选择中断

  在SPI从机模式下，有从机选择有效或无效中断标志，SSACTIF (USPI_PROTSTS[9])和SSINAIF (USPI_PROTSTS[8])，当SLAVE (USPI_PROTCTL [0])置1并且从机检测到片选信号有效或无效时，将会置1。如果SSINAIEN (USPI_PROTIEN[0]) 或 SSACTIEN (USPI_PROTIEN[1])置1，SPI控制器会产生一个中断。SPI内部从机选择信号是高有效，用户可以通过设置ININV (USPI_CTLIN0[2])反转从机选择信号的极性满足低有效的设备。

- 从机超时中断

在SPI从机模式下，从机超时功能可以使用户知道数据传输没有完成突然没有了时钟信号的错误；此功能通过设置SLVTOCNT (USPI_PROTCTL[25:16])，使用时序测量计数器来测量超时周期，时序测量计数器的频率可以通过TMCNTSRC (USPI_BRGEN[5])选择

SPI 从 机 模 式 下 ， TMCNTEN (USPI_BRGEN[4])=1 且 SLVTOCNT (USPI_PROTCTL[25:16])不为0，时序测量计数器使能；此计数器在接收数据的第一个时钟开始计数，收到下一个时钟计数器清零并重新计数，最终计数器将在数据接收完成后清零并停止，如果计数值等于或大于SLVTOCNT (USPI_PROTCTL[25:16])设定的值，就会产生超 时 中 断 事 件 SLVTOIF (USPI_PROTSTS[5])=1 ， 如 果 此 时 SLVTOIEN (USPI_PROTIEN[2])=1控制器将产生中断

相关缓冲中断

如果USCI控制器中有发送/接收缓冲，相关缓冲中断有效。

- 接收缓冲上溢出中断

  如果接收缓冲上溢出事件发生，RXOVIF (USPI_BUFSTS[3])会被置1。可以写1清除。

- 发送缓冲下溢出中断

  如果发送缓冲下溢出事件发生，TXUDRIF (USPI_BUFSTS[11])会被置1。可以写1清除。

6.19.5.11 时序图

USCI SPI协议的从机片选信号默认值是高电平有效，可以设置CTLOINV (USPI_LINECTL[7])反相。

串行时钟的空闲状态和被用于发送/接收数据的串行总线时钟边沿可以通过设置SCLKMODE (USPI_PROTCTL[7:6])来配置。传输数据字的位宽度由DWIDTH (USPI_LINECTL[11:8])决定，数据位传输顺序由LSB (USPI_LINECTL[0])决定。主机/从机操作和相关设置如下面四组SPI时序图所示。

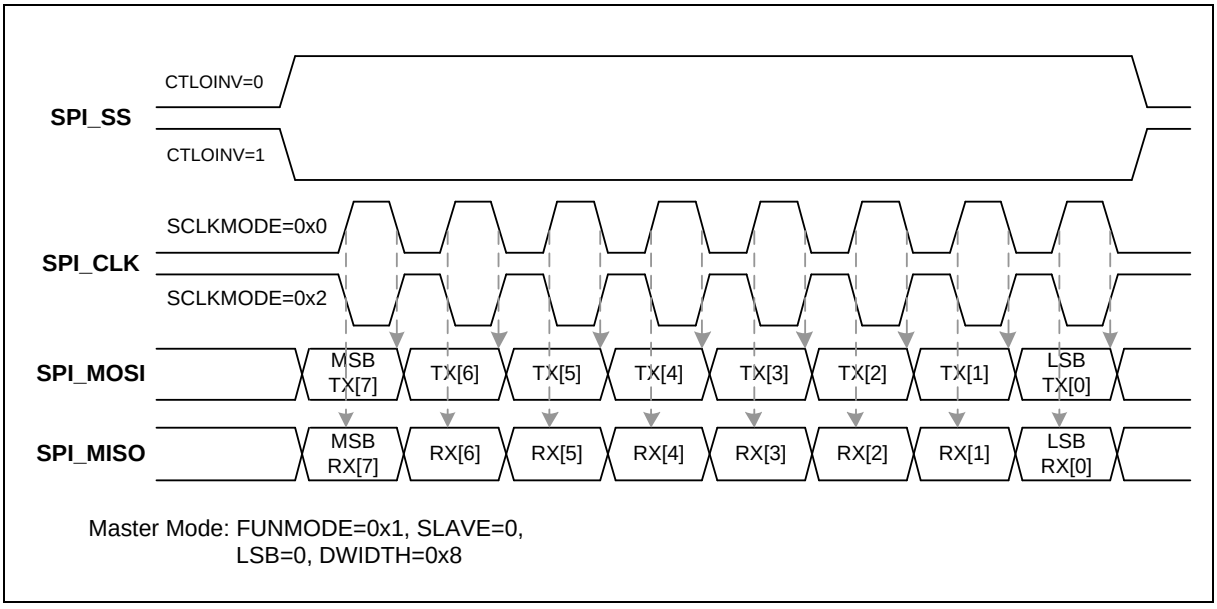


图 6.19-16 主机模式下的 SPI 时序

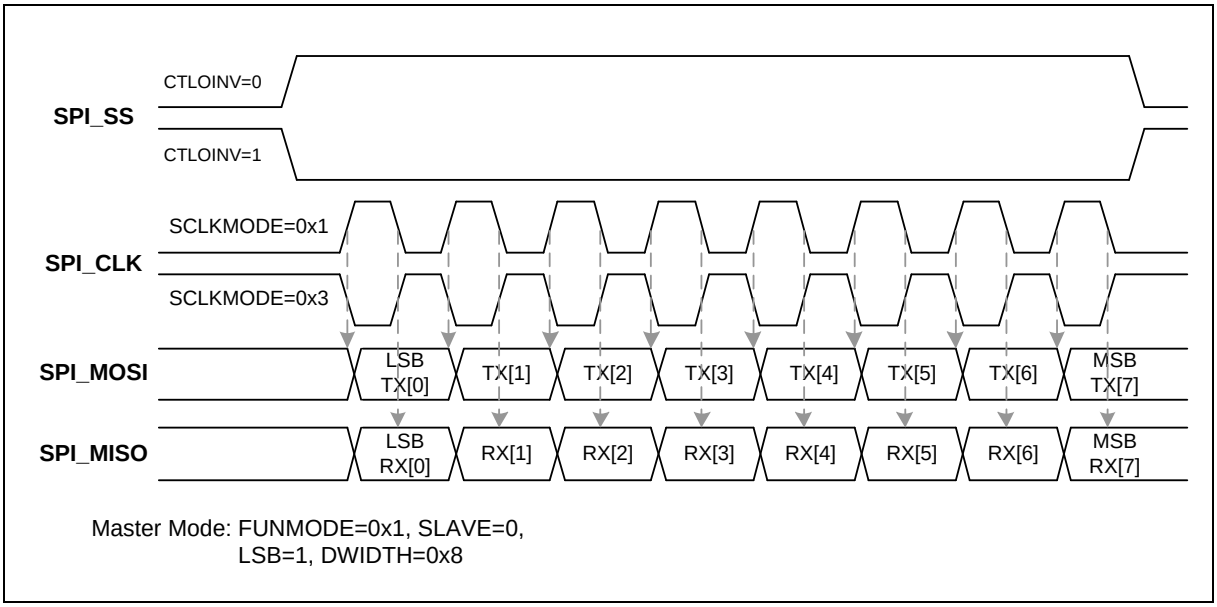


图 6.19-17 主机模式下的 SPI 时序(串行总线时钟反相)

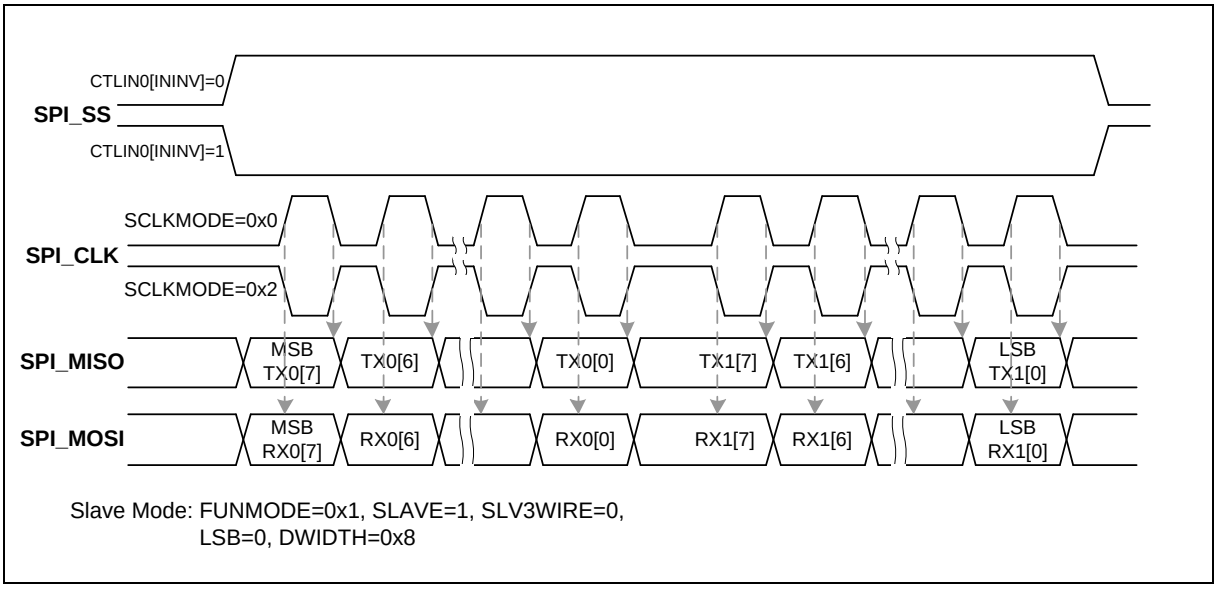


图 6.19-18 从机模式下的 SPI 时序

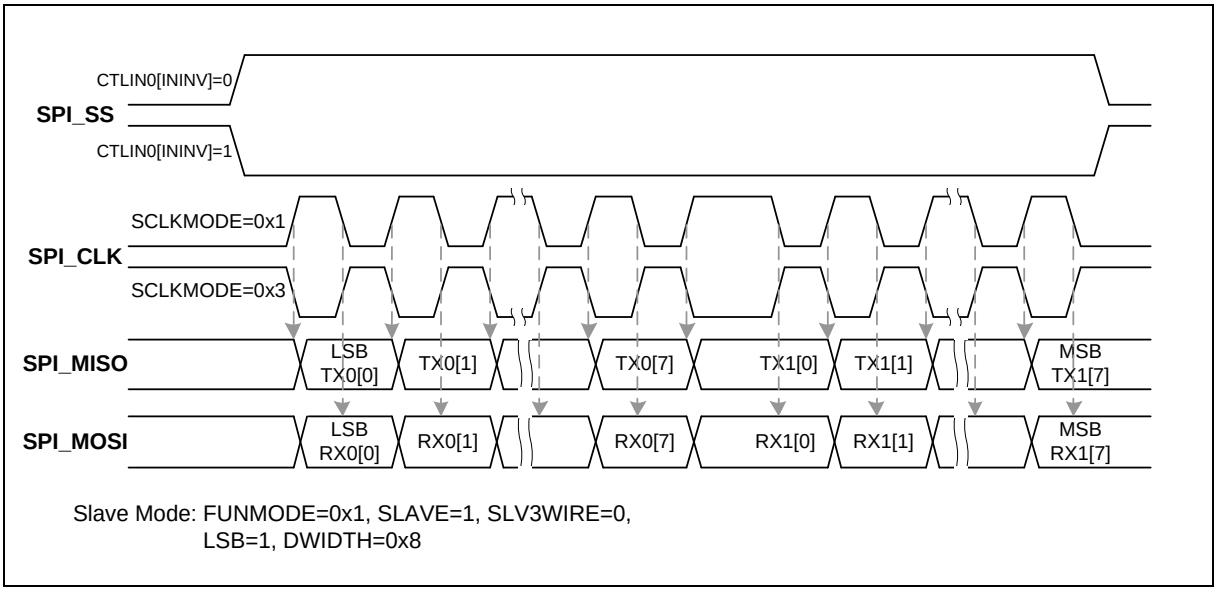


图 6.19-19 从机模式下的 SPI 时序(串行总线时钟反相)

6.19.5.12 编程流程

该章节描述了USCI SPI数据传输的编程流程

主机模式:

1. 设置CLK_APBCLK1寄存器，使能USCI外设时钟
2. 通过设置相关多功能控制寄存器配置用户指定的引脚作为USCI功能引脚
3. 设置FUNMODE (USPI_CTL[2:0])为1来选择SPI模式
4. 设置USPI_BRGEN寄存器确定SPI总线时钟频率
5. 依据用户应用需求，配置设置如下
   - CTLOINV (USPI_LINECTL[7]):如果从机片选信号低电平有效，设置该位为1，否则设置为0
   - DWIDTH (USPI_LINECTL[11:8]): 数据宽度设置
   - LSB (USPI_LINECTL[0]): LSB优先还是MSB 优先
   - TSMSEL (USPI_PROTCTL[14:12]):全双工SPI传输或一个通道半双工SPI传输
   - SCLKMODE (USPI_PROTCTL[7:6]):确定时钟时序
   - AUTOSS (USPI_PROTCTL[3]):是否使能自动从机片选功能
   - SLAVE (USPI_PROTCTL[0]):主机模式设置为0
   - 设置 PROTEN (USPI_PROTCTL[31])为 1 来使能 SPI 协议
6. 如果禁用(AUTOSS=0)自动从机片选功能，在数据传输前设置SS (USPI_PROTCTL[2])为1；设置SS为0来取消从机选择信号
7. 写USPI_TXDAT寄存器来触发SPI传输。在半双工SPI传输中，数据引脚方向通过设置PORTDIR (USPI_TXDAT[16])来确定
8. 只要RXEMPTY (USPI_BUFSTS[0])为0，用户就可以通过读USPI_RXDAT寄存器来获取接收的数据。

从机模式：

1. 设置CLK_APBCLK1寄存器，使能USCI外设时钟
2. 通过设置相关多功能控制寄存器配置用户指定的引脚作为USCI功能引脚
3. 设置FUNMODE (USPI_CTL[2:0])为1来选择SPI模式
4. 依据用户应用需求，配置设置如下
   - ININV (USPI_CTLIN0[2]): 如果从机片选信号启用时低电平，设置该位为1，否则设置为0
   - DWIDTH (USPI_LINECTL[11:8]): 数据宽度设置
   - LSB (USPI_LINECTL[0]): LSB优先还是MSB 优先
   - TSMSEL (USPI_PROTCTL[14:12]): 全双工SPI传输或一个通道半双工SPI传输
   - SCLKMODE (USPI_PROTCTL[7:6]): 确定时钟时序
   - SLAVE (USPI_PROTCTL[0]): 从机模式设置为1
5. 设置 PROTEN (USPI_PROTCTL[31])为 1 来使能 SPI 协议.
6. 写USPI_TXDAT寄存器来触发SPI传输。在半双工SPI传输中，数据引脚方向通过设置PORTDIR (USPI_TXDAT[16])来确定
7. 只要RXEMPTY (USPI_BUFSTS[0])为0，用户就可以通过读USPI_RXDAT寄存器来获取接收的数据。只要TXFULL (USPI_BUFSTS[9])为0，下次发送的数据就可以写入到USPI_TXDAT寄存器中。

6.19.5.13 唤醒功能

在SPI模式下USCI控制器支持唤醒系统功能。SPI协议中唤醒源是输入片选信号的转变。

6.19.6 寄存器映射

R:只读, **W:**只写, **R/W:**读/写

寄存器	偏移	R/W	描述	复位值
USCI_SPI基地址: **USPIn_BA = 0x400D_0000 + (0x1000 * n)** **n= 0,1**				
USPI_CTL	USPIn_BA+0x00	R/W	USCI控制寄存器	0x0000_0000
USPI_INTEN	USPIn_BA+0x04	R/W	USCI中断使能寄存器	0x0000_0000
USPI_BRGEN	USPIn_BA+0x08	R/W	USCI波特率发生器寄存器	0x0000_3C00
USPI_DATIN0	USPIn_BA+0x10	R/W	USCI输入数据信号配置寄存器0	0x0000_0000
USPI_CTLIN0	USPIn_BA+0x20	R/W	USCI输入控制信号配置寄存器0	0x0000_0000
USPI_CLKIN	USPIn_BA+0x28	R/W	USCI输入时钟信号配置寄存器	0x0000_0000
USPI_LINECTL	USPIn_BA+0x2C	R/W	USCI线控制寄存器	0x0000_0000
USPI_TXDAT	USPIn_BA+0x30	W	USCI发送数据寄存器	0x0000_0000
USPI_RXDAT	USPIn_BA+0x34	R	USCI接收数据寄存器	0x0000_0000
USPI_BUFCTL	USPIn_BA+0x38	R/W	USCI发送/接收缓冲控制寄存器	0x0000_0000
USPI_BUFSTS	USPIn_BA+0x3C	R	USCI发送/接收缓冲状态寄存器	0x0000_0101
USPI_PDMACTL	USPIn_BA+0x40	R/W	USCI PDMA控制寄存器	0x0000_0000
USPI_WKCTL	USPIn_BA+0x54	R/W	USCI唤醒控制寄存器	0x0000_0000
USPI_WKSTS	USPIn_BA+0x58	R/W	USCI唤醒状态寄存器	0x0000_0000
USPI_PROTCTL	USPIn_BA+0x5C	R/W	USCI协议控制寄存器	0x0000_0300
USPI_PROTIEN	USPIn_BA+0x60	R/W	USCI协议中断使能寄存器	0x0000_0000
USPI_PROTSTS	USPIn_BA+0x64	R/W	USCI协议状态寄存器	0x0000_0000

6.19.7 寄存器描述

USCI控制寄存器(USPI_CTL)

寄存器	偏移	R/W	描述	复位值
USPI_CTL	USPIn_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**FUNMODE**	**功能模式** 该域是选择USCI控制器的协议。选择不可用或保留的协议将禁用USCI。当两个协议之间切换时，在选择新协议前USCI必须先禁用。在用户写000到FUNMODE时，USCI会被复位。 000 = 禁用USCI，所有协议相关的状态机器被设置为空闲状态。 001 = 选择SPI 协议 010 = 选择UART 协议 100 = 选择I^2C 协议 **注：**其它值保留

USCI中断使能寄存器 (USPI_INTEN)

寄存器	偏移	R/W	描述	复位值
USPI_INTEN	USPIn_BA+0x04	R/W	USCI 中断使能寄存器	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[4]	**RXENDIEN**	**接收结束中断使能位** 该位是使能发生接收结束事件时产生中断 0 =禁用接收结束中断 1 = 使能接收结束中断
[3]	**RXSTIEN**	**接收起始中断使能位** 该位是使能发生接收起始事件时产生中断 0 =禁用接收起始中断 1 = 使能接收起始中断
[2]	**TXENDIEN**	**发送结束中断使能位** 该位是使能发生发送结束事件时产生中断 0 =禁用发送结束中断 1 = 使能发送结束中断
[1]	**TXSTIEN**	**发送起始中断使能位** 该位是使能发生发送起始事件时产生中断 0 =禁用发送起始中断 1 = 使能发送起始中断
[0]	**Reserved**	保留

USCI波特率发生器寄存器(USPI_BRGEN)

寄存器	偏移	R/W	描述	复位值
USPI_BRGEN	USPIn_BA+0x08	R/W	USCI 波特率发生器寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						**CLKDIV**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLKDIV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TMCNTSRC**	**TMCNTEN**	**SPCLKSEL**		**PTCLKSEL**	**RCLKSEL**

位	描述	
[31:26]	**Reserved**	保留
[25:16]	**CLKDIV**	**时钟分频器** 该域是定义协议时钟频率 f_{PROT_CLK} 和时钟分频器频率 f_{DIV_CLK} (f_{DIV_CLK}= f_{PROT_CLK}/(CLKDIV+1)).之间的比率
[15:6]	**Reserved**	保留
[5]	**TMCNTSRC**	**时序测量计数器时钟源选择** 0 = 来自f_{PROT_CLK}. 1 =来自f_{DIV_CLK}.
[4]	**TMCNTEN**	**时序测量计数器使能位** 该位是使能10位时序测量计数器 0 =禁用时序测量计数器 1 =使能时序测量计数器
[3:2]	**SPCLKSEL**	**采样时钟源选择** 该域用于对协议采样时钟(f_{SAMP_CLK}) 的时钟源选择 00 = f_{DIV_CLK}. 01 = f_{PROT_CLK}. 10 = f_{SCLK}. 11 = f_{REF_CLK}.
[1]	**PTCLKSEL**	**协议协议时钟源选择** 该位用于选择协议时钟源(f_{PROT_CLK}) 0 = 参考时钟f_{REF_CLK}. 1 = f_{REF_CLK2} (频率是f_{REF_CLK}一半)
[0]	**RCLKSEL**	**参考时钟源选择** 该位选择参考时钟源(f_{REF_CLK})

		0 = 外设设备时钟f_{PCLK}. 1 = 保留

USCI输入数据信号配置寄存器(USPI_DATIN0)

寄存器	偏移	R/W	描述	复位值
USPI_DATIN0	USPIn_BA+0x10	R/W	USCI输入数据信号配置寄存器0	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**ININV**	**输入信号反相选择** 该域是使能对输入异步信号的反相 0 =对异步输入信号不反相 1 = 对异步输入信号反相 **注：** 在 SPI协议下,我们建议该位设置为0。
[1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入信号同步选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入 **注：** 在 SPI协议下,我们建议该位设置为0。

USCI输入控制信号配置寄存器(USPI_CTLIN0)

寄存器	偏移	R/W	描述	复位值
USPI_CTLIN0	USPIn_BA+0x20	R/W	USCI输入控制信号配置寄存器0	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**ININV**	**输入信号反相选择** 该域是定义使能对输入异步信号输入反相 0 =对异步输入信号不反相 1 = 对异步输入信号不反相
[1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入同步信号选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入 **注：** 在 SPI协议下,我们建议该位设置为0。

USCI输入时钟信号配置寄存器(USPI_CLKIN)

寄存器	偏移	R/W	描述	复位值
USPI_CLKIN	USPIn_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**SYNCSEL**	**输入同步信号选择** 该位是选择异步输入（带反相的）信号或是同步（带过滤的）信号被用作数据移位单元的输入 0 =异步信号作为信号被用作数据移位单元的输入 1 = 同步信号作为信号被用作数据移位单元的输入 **注：**在 SPI协议下,我们建议该位设置为0。

USCI线控制寄存器(USPI_LINECTL)

寄存器	偏移	R/W	描述	复位值
USPI_LINECTL	USPIn_BA+0x2C	R/W	USCI 线控制寄存器	0x0000_0000

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

位	描述	
[31:12]	**Reserved**	保留
[11:8]	**DWIDTH**	**传输的字长** 该域定义发送和接收的数据字长。在数据缓冲中数据总是右对齐。USCI支持4到16位的字长 0x0:在数据字中包含16位，位于[15:0]这些位中 0x1: 保留 0x2: 保留 0x3: 保留 0x4:在数据字中包含4位，位于[3:0]这些位中 0x5:在数据字中包含5位，位于[4:0]这些位中 ... 0xF:在数据字中包含15位，位于[14:0]这些位中
[7]	**CTLOINV**	**控制信号输出反相选择** 该位定义了内部控制信号和输出控制信号之间的关系 0 = 无影响 1 =在输出前控制信号被反相 **注：**在不同协议中控制信号有不同的定义。SPI协议中，控制信号是从机片选信号。
[6]	**Reserved**	保留
[5]	**DATOINV**	**数据输出反相选择** 该位定义了内部移位数据值和USCIx_DAT0/1引脚输出数据信号之间的关系 0 = 数据输出电不反相 1 =数据输出电反相
[4:1]	**Reserved**	保留
[0]	**LSB**	**LSB优先传输选择** 0 =MSB，哪一位先发送/接收，取决于DWIDTH的设置

		1 =LSB，数据缓冲的位0被先发送/接收

USCI发送数据寄存器(USPI_TXDAT)

寄存器	偏移	R/W	描述	复位值
USPI_TXDAT	USPIn_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000

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

位	描述	
[31:17]	**Reserved**	保留
[16]	**PORTDIR**	**端口方向控制** 该位域仅在USCI工作在SPI的半双工传输下有效。它被用来定义数据端口引脚的方向。当软件写USPI_TXDAT寄存器时，发送数据和端口方向同时被处理。 0 =数据引脚被配置成输出模式 1 =数据引脚被配置成输入模式
[15:0]	**TXDAT**	**发送数据** 软件可以写16位发送数据到该域用来发送，为了避免发送数据写溢出，用户必须在写发送数据到该域之前检查TXEMPTY (USPI_BUFSTS[8])状态。

USCI接收数据寄存器(USPI_RXDAT)

寄存器	偏移	R/W	描述	复位值
USPI_RXDAT	USPIn_BA+0x34	R	USCI 接收数据寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**RXDAT**	**接收的数据** 该域值监控存储在接收数据缓冲的接收到的数据

USCI发送/接收缓冲控制寄存器 (USPI_BUFCTL)

寄存器	偏移	R/W	描述	复位值
USPI_BUFCTL	USPIn_BA+0x38	R/W	USCI 发送/接收缓冲控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**RXRST**	**TXRST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXCLR	**RXOVIEN**	**Reserved**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXCLR	**TXUDRIEN**	**Reserved**					

位	描述	
[31:18]	**Reserved**	保留
[17]	**RXRST**	**接收复位** 0 = 无影响 1 =复位接收相关计数器、状态机、接收移位寄存器的内容和数据缓冲。 **注：** 1个PCLK周期后自动被清除
[16]	**TXRST**	**发送复位** 0 = 无影响 1 =复位发送相关计数器、状态机、发送移位寄存器的内容和数据缓冲。 **注：** 1个PCLK周期后自动被清除
[15]	**RXCLR**	**清接收缓冲** 0 = 无影响 1 = 接收缓冲被清除。仅当缓冲没有参与数据通信时被使用。 **注：** 1个PCLK周期后自动被清除
[14]	**RXOVIEN**	**接收缓冲溢出中断使能控制** 0 =禁用接收上溢出中断 1 =使能接收上溢出中断
[13:8]	**Reserved**	保留
[7]	**TXCLR**	**清发送缓冲** 0 = 无影响 1 = 发送缓冲被清除。仅当缓冲没有参与数据通信时被使用。 **注：** 1个PCLK周期后自动被清除
[6]	**TXUDRIEN**	**从机发送下溢出中断使能位** 0 =禁用发送下溢出中断

		1 =使能发送下溢出中断
[5:0]	**Reserved**	保留

USCI发送/接收缓冲状态寄存器(USPI_BUFSTS)

寄存器	偏移	R/W	描述	复位值
USPI_BUFSTS	USPIn_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**TXUDRIF**	**Reserved**	**TXFULL**	**TXEMPTY**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**RXOVIF**	**Reserved**	**RXFULL**	**RXEMPTY**

位	描述	
[31:12]	**Reserved**	保留
[11]	**TXUDRIF**	**发送缓冲下溢出中断状态** 该位表示发送缓冲下溢出事件被侦测到。如果TXUDRIEN (USPI_BUFCTL[6])使能，会启用相应的中断请求。软件写1清除该位。 0 = 发送缓冲下溢出事件未被侦测到 1 = 发送缓冲下溢出事件被侦测到
[10]	**Reserved**	保留
[9]	**TXFULL**	**发送缓冲满指示** 0 =发送缓冲还没满 1 =发送缓冲满
[8]	**TXEMPTY**	**发送缓冲空指示** 0 =发送缓冲非空 1 =发送缓冲为空，可进行下次数据发送
[7:4]	**Reserved**	保留
[3]	**RXOVIF**	**接收缓冲上溢出中断状态** 该位表示接收缓冲上溢出事件被侦测到。如果RXOVIEN (USPI_BUFCTL[14])使能，会启用相应的中断请求。软件写1清除该位。 0 = 接收缓冲上溢出事件未被侦测到 1 = 接收缓冲上溢出错件被侦测到
[2]	**Reserved**	保留
[1]	**RXFULL**	**接收缓冲满指示** 0 =接收缓冲还没满 1 =接收缓冲满

[0]	**RXEMPTY**	**接收缓冲空指示** 0 =接收缓冲非空 1 =接收缓冲为空

USCIPDMA控制寄存器(USPI_PDMACTL)

寄存器	偏移	R/W	描述	复位值
USPI_PDMACTL	USPIn_BA+0x40	R/W	USCI PDMA 控制寄存器	0x0000_0000

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

位	描述	
[31:4]	**Reserved**	保留
[3]	**PDMAEN**	**PDMA模式使能位** 0 = 禁用PDMA功能 1 = 使能PDMA功能 注： I^2C 不支持 PDMA 功能
[2]	**RXPDMAEN**	**PDMA接收通道有效** 0 = 禁用PDMA接收功能 1 = 使能PDMA接收功能
[1]	**TXPDMAEN**	**PDMA发送通道有效** 0 = 禁用PDMA发送功能 1 = 使能PDMA发送功能
[0]	**PDMARST**	**PDMA 复位** 0 = 无影响 1 =复位USCI的PDMA控制逻辑。该位会自动地被清0

USCI唤醒控制寄存器(USPI_WKCTL)

寄存器	偏移	R/W	描述	复位值
USPI_WKCTL	USPIn_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2]	**PDBOPT**	**掉电阻塞选择** 0 = 如果用户试图在协议正在传输的时候执行WFI来进入掉电模式，MCU会停止传输并且立即进入掉电模式 1 =如果用户试图在协议正在传输的时候执行WFI来进入掉电模式，传输会继续进行并且MCU立即进入空闲模式
[1]	**Reserved**	保留
[0]	**WKEN**	**唤醒使能位** 0 =禁用唤醒功能 1 =使能唤醒功能

USCI唤醒状态寄存器(USPI_WKSTS)

寄存器	偏移	R/W	描述	复位值
USPI_WKSTS	USPIn_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**WKF**	**唤醒标志** 当芯片被从掉电模式唤醒，该位置1，软件写1清除该位。

USCI协议控制寄存器– USPI_PROTCTL (SPI)

寄存器	偏移	R/W	描述	复位值
USPI_PROTCTL	USPIn_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0300

31	30	29	28	27	26	25	24
PROTEN	**Reserved**		**TXUDRPOL**	**Reserved**		**SLVTOCNT**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SLVTOCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TSMSEL**			**SUSPITV**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCLKMODE		**Reserved**		**AUTOSS**	**SS**	**SLV3WIRE**	**SLAVE**

位	描述	
[31]	**PROTEN**	**SPI协议使能位** 0 = 禁用SPI协议 1 = 使能SPI协议
[30:29]	**Reserved**	保留
[28]	**TXUDRPOL**	**发送下溢出数据极性 (仅从机模式)** 该位定义在没有数据可发送时发送数据电平 0 =如果TX下溢出事件发生，输出数据电平为0. 1 =如果TX下溢出事件发生，输出数据电平为1.
[27:26]	**Reserved**	保留
[25:16]	**SLVTOCNT**	**从机模式超时周期(仅从机模式)** 在从机模式，该域被用作从机超时周期。该域表示在输入SCLK的两个边沿之间有多少时钟周期（时钟由TMCNTSRC, USPI_BRGEN[5]选择）产生从机超时事件。写0x0到该位域禁用从机超时功能。 范例：SLVTOCNT为 0x0A 和 TMCNTSRC (USPI_BRGEN[5])为1 1，表示如果SPI总线时钟引脚超过（10+1）个f_{DIV_CLK}周期没有改变，将会有超时事件发生。
[15]	**Reserved**	保留
[14:12]	**TSMSEL**	**传输数据模式选择** 该位域定义发送和接收数据是如何移进和移出的。 TSMSEL = 000b: 全双工 SPI. TSMSEL = 100b: 半双工 SPI. 其它值保留 **注：**改变该位域的值将会自动地产生TXRST 和 RXRST来清除TX/RX数据缓冲。
[11:8]	**SUSPITV**	**暂停间隔(仅主模式)** 该四位用来配置在一次数据传输过程中连续两个发送/接收事务之间的暂停间隔。暂停间隔

		是从当前传输字的最后一个时钟边沿到接下来的传输字的第一个边沿时钟。默认值是 0x3. 暂停间隔的周期可以根据下面公式获得： (SUSPITV [3:0] + 0.5) * SPI_CLK时钟周期 例: SUSPITV = 0x0 … 0.5 SPI_CLK 时钟周期 SUSPITV = 0x1 … 1.5 SPI_CLK 时钟周期 …… SUSPITV = 0xE … 14.5 SPI_CLK 时钟周期 SUSPITV = 0xF … 15.5 SPI_CLK 时钟周期
[7:6]	**SCLKMODE**	**串行总线时钟模式** 该位域定义SCLK空闲状态、数据传输和接收边沿 MODE0 =空闲状态时是低电平。下降沿时发送数据，上升沿时接收数据。 MODE1 =空闲状态时是低电平。上升沿时发送数据，下降沿时接收数据。 MODE2 =空闲状态时是高电平。上升沿时发送数据，下降沿时接收数据。 MODE3 =空闲状态时是高电平。下降沿时发送数据，上升沿时接收数据。
[5:4]	**Reserved**	保留
[3]	**AUTOSS**	**自动从机片选功能使能(仅主模式)** 0 =通过设置SS (USPI_PROTCTL[2])值来控制从机片选信号。 1 = 从机片选信号自动地产生。当发送/接收启动时从机片选信号会被SPI控制启用，在每次发送/接收结束之后片选信号取消
[2]	**SS**	**从机选择控制(仅主模式)** 如果AUTOSS位被清除，设置该位为1将启用从机片选信号，设置该位为0将取消从机片选信号。 如果AUTOSS功能使能(AUTOSS = 1),该位的设置值不会影响从机片选信号的当前的状态。 **注：** SPI 协议下，内部从机片选信号是高有效。
[1]	**SLV3WIRE**	**从机3-线模式选择(仅从机模式)** 从机模式下，SPI协议在3-线接口（及没有片选信号）下工作。 0 = 4-线接口 1 = 3-线接口
[0]	**SLAVE**	**从机模式选择** 0 = 主机模式 1 = 从机模式

USCI协议中断使能寄存器– USPI_PROTIEN (SPI)

寄存器	偏移	R/W	描述	复位值
USPI_PROTIEN	USPIn_BA+0x60	R/W	USCI协议中断使能寄存器	0x0000_0000

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

位	描述	
[31:5]	**Reserved**	保留
[3]	**SLVBEIEN**	**从机模式位计数错误中断使能控制** 如果在从机模式下数据传输由从机超时或是从机片选取消事件终止，这样一来发送/接收的数据位计数就会与DWIDTH (USPI_LINECTL[11:8])设置值不匹配。位计数错误事件发生。 0 =禁用从机模式位计数错误中断 1 =使能从机模式位计数错误中断
[2]	**SLVTOIEN**	**从机超时中断使能控制** SPI协议下，使能该中断以防从机超时事件 0 =禁用从机超时中断 1 =使能从机超时中断
[1]	**SSACTIEN**	**从机选择有效中断使能位** 如果从机片选信号变为有效状态，该位使能/禁止产生从机片选中断 0 =从机选择有效中断发生禁用 1 =从机选择有效中断发生使能
[0]	**SSINAIEN**	**从机选择无效中断使能位** 如果从机片选信号变为无效状态，该位使能/禁用产生从机片选中断 0 =从机选择无效中断发生禁用 1 =从机选择无效中断发生使能

USCI协议状态寄存器– USPI_PROTSTS (SPI)

寄存器	偏移	R/W	描述	复位值
USPI_PROTSTS	USPIn_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**SLVUDR**	**BUSY**	**SSLINE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**SSACTIF**	**SSINAIF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**SLVBEIF**	**SLVTOIF**	**RXENDIF**	**RXSTIF**	**TXENDIF**	**TXSTIF**	**Reserved**

位	描述	
[31:19]	**Reserved**	保留
[18]	**SLVUDR**	**从机模式发送下溢出状态（只读）** 在从机模式下，如果没有可用的发送数据在缓冲中，同时输入串行总线时钟使发送数据继续移出，这时该位就会被置1。该位置位与字传输当前移出数据是否已切换到TXUDRPOL (USPI_PROTCTL[28])没有关系。 0 =没有发生从机发送下溢出事件 1 =发生从机发送下溢出事件
[17]	**BUSY**	**忙状态（只读）** 0 = SPI 处在空闲状态 1 = SPI 处在忙状态 下面列出了总线忙的状况： a. USPI_PROTCTL[31] = 1 和TXEMPTY = 0. b. 对于SPI主机模式，TXEMPTY = 1但是当前传输还没完成 c. 对于SPI从机模式，USPI _PROTCTL[31] = 1并且在从机片选有效时有串行时钟输入到SPI内核逻辑单元 **d.** 对于SPI从机模式，USPI _PROTCTL[31] = 1并且即使在从机片选无效状况下发送缓冲或是发送移位寄存器为非空
[16]	**SSLINE**	**从机选择线总线状态（只读）** 仅在从模式下有效。该位是用来监控总线上输入从机片选信号当前的状态。 0 =从机选择线状态为0 1 =从机选择线状态为1
[15:10]	**Reserved**	保留
[9]	**SSACTIF**	**从机选择有效中断标志(仅从机模式)** 该位表示内部从机片选信号变为有效状态。软件写1清除该位。

		0 =从机片选信号没有改变成有效状态 1 =从机片选信号改变成有效状态 **注：** 内部从机片选信号有效状态时为高电平。
[8]	**SSINAIF**	**从机选择取消中断标志(仅从机模式)** 该位表示内部从机片选信号变为无效状态。软件写1清除该位。 0 =从机片选信号没有改变成无效状态 1 =从机片选信号改变成无效状态 **注：** 内部从机片选信号有效状态时为高电平。
[7]	**Reserved**	保留
[6]	**SLVBEIF**	**从机位计数错误中断标志(仅从机模式)** 0 =没有发生从机位计数错误事件 1 = 发生从机位计数错误事件 **注：** 写1清除该位
[5]	**SLVTOIF**	**从机超时中断标志(仅从机模式)** 0 =没有发生从机超时事件 1 = 发生从机超时事件 **注：** 写1清除该位
[4]	**RXENDIF**	**接收结束中断标志** 0 =没有接收结束中断状态发生 1 = 有接收结束中断状态发生 **注：** 写1清除该位
[3]	**RXSTIF**	**接收起始中断标志** 0 =没有接收起始中断状态发生 1 = 有接收起始中断状态发生 **注：** 写1清除该位
[2]	**TXENDIF**	**发送结束中断标志** 0 =没有发送结束中断状态发生 1 = 有发送结束中断状态发生 **注：** 写1清除该位
[1]	**TXSTIF**	**发送起始中断标志** 0 =没有发送起始中断状态发生 1 = 有发送起始中断状态发生 **注：** 写1清除该位
[0]	**Reserved**	保留

6.20 USCI - I²C模式

6.20.1 概述

在I²C总线上，数据通过时钟线SCL和数据线SDA在主从机间逐一字节同步传送。每个字节数据长度是8位. 一个SCL 时钟脉冲传输一个数据位，数据由最高位MSB 开始传输，每个字节传输后跟随一个应答位，每个位在SCL为高时采样；因此，SDA线只能在SCL为低时改变，在SCL为高时必须保持稳定。当SCL为高时，SDA 线上的跳变视为一个命令(START 或STOP)。更多关于I²C总线时序的细节请参考图6.20-1。

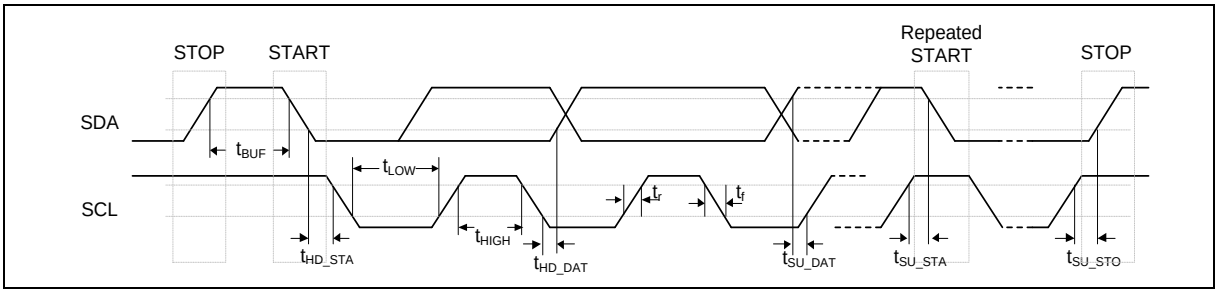


图 6.20-1 I²C 总线时序

片上I²C外设提供了一个符合I²C总线规范的串行接口。I²C端口自动处理字节传输。通过设置寄存FUNMODE (UI2C _CTL [2:0]) = 100B选择I²C模式。USCI硬件接口通过数据线SDA 和时钟线 SCL两个引脚连到I²C总线。当I/O 引脚作为I²C 端口使用时，用户必须事先设定I/O引脚为I²C功能。

注：SDA 和 SCL两个引脚需要上拉电阻，因为这个两个引脚在USCI被选作I²C模式下为开漏脚.

6.20.2 特性

- 支持主机/从机模式
- 支持7位地址模式和10位地址模式
- 支持标准模式 (100 kbps)或快速模式 (400 kbps)
- 支持多主机总线
- 支持一个发送缓冲和两个接收缓冲
- 支持10位总线超时
- 支持总线监控模式
- 支持通过接收'START'信号或地址匹配掉电唤醒
- 支持建立/保持时间可编程
- 支持多地址识别（ 2组从机地址带mask 选项）

6.20.3 框图

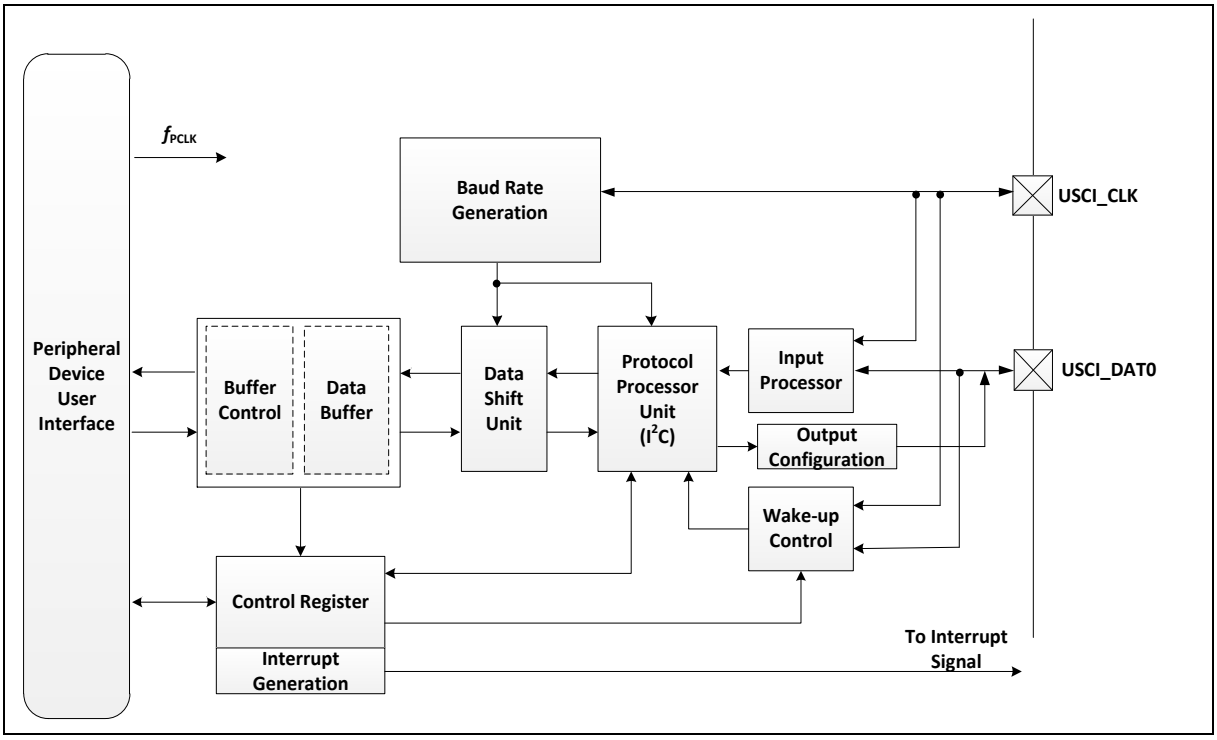


图 6.20-2USCI I²C 模式框图

6.20.4 基本配置

6.20.4.1 USCI0_I2C基本配置

- 时钟源配置
  - 在USCI0CKEN（CLK_APBCLK1[8]）使能USCI0时钟
  - 在USPI_CTL[2:0]寄存器使能USCI0_I2C功能, USPI_CTL[2:0]=3’b100
- 复位配置
  - 在USCI0RST（SYS_IPRST2[8]）复位USCI0控制器

6.20.4.2 USCI1_I2C基本配置

- 时钟源配置
  - 在USCI1CKEN (CLK_APBCLK1[9])使能USCI1外设时钟
  - 在USPI_CTL[2:0]寄存器使能USCI1_I2C功能, USPI_CTL[2:0]=3’b100
- 复位配置
  - 在USCI1RST (SYS_IPRST2[9])复位USCI1控制器

6.20.5 功能描述

6.20.5.1 START 或 Repeated START 信号

标准I²C 协议如图6.20-3，通常标准通讯有以下4部分：

- (START) 或者重复起始信号(Repeated START)
- 从机地址传输和R/W 位传输
- 数据传输
- 停止信号(STOP)

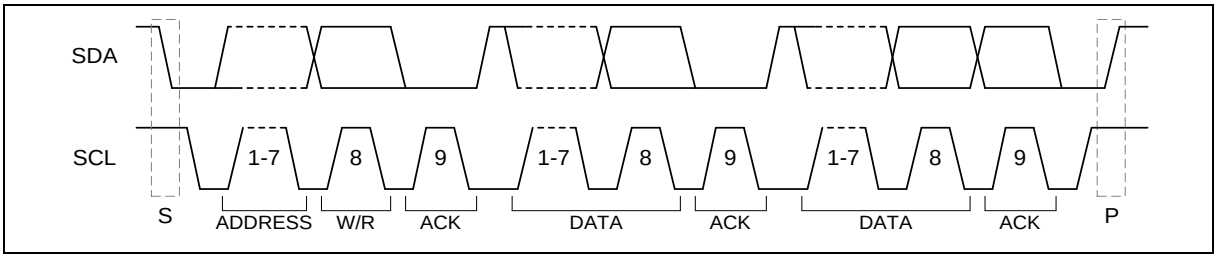


图 6.20-3 I²C 协议

当总线处于释放/空闲状态下，说明没有主机设备占用总线（SDA与SCL线同时为高），主机可以通过发送起始(START)信号来发起传输过程。起始信号，通常表示为"S"位，当SCL线为高时，SDA 线上信号由高至低变化，就被定义为起始信号。起始信号表示一个新的数据传输的开始。

重复起始信号通常标记为"Sr"，主机通过该方法与另一个从机或不同传输方向（例如：从向设备写到从设备读）的同一个从机通讯，而不需要释放总线。

6.20.5.2 停止信号

主机可以通过产生一个停止信号来终止数据通信。停止信号，通常表示为P位，当SCL 线为高时，SDA 线上信号由低至高变化，就被定义为停止信号。从停止信号到开始信号这段时间为总线空闲。

图6.20-4为起始，重复起始和停止信号的波形。

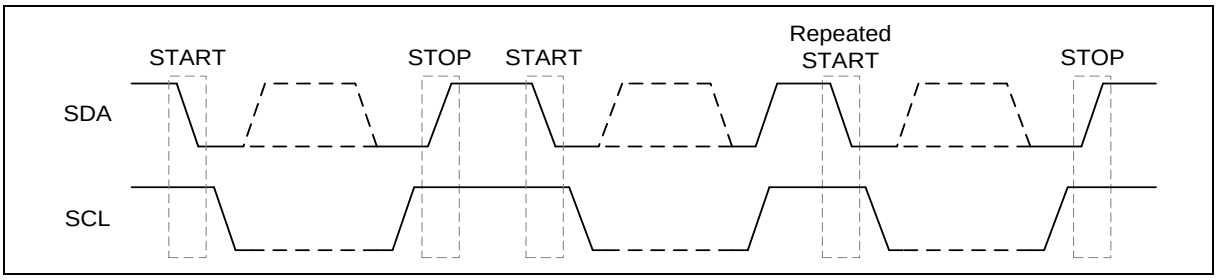


图 6.20-4 START 和 STOP 状态

6.20.5.3 从机地址传输

在一个起始（或重复起始）信号后，主机发送一个从机地址来确认通信的目标设备。起始地址有一个或两个字节组成（针对7位或10位地址情况）。地址字节过后，从机会对传输的地址作应答回应。

因此，从机地址可以被编程并且与接收到的地址比较。一旦匹配，从机会回一个应答(SDA = 0).如果与目标不匹配则无应答(SDA = 1)。除了编程地址匹配之外，如果从机有能力处理相应的请求，那么另一个地址值（00H）也必须有应答。地址为00H是广播地址，可以被应答。

为了可选择应答不同的地址值，控制机制是按如下实现的：

- 如果GCFUNC 位 (UI2C_PROTCTL [0])被置位，I^{2}C端口硬件就会响应广播地址（00H）。清除GC位禁用广播功能
- I^{2}C端口装备了一个设备地址寄存器UI2C _DEVADDRn (n = 0~1)。在7位地址模式下，首先接收到的7位地址会与编程的从机地址(UI2C _DEVADDRn [6:0])相比较。如果匹配，从机会回应答。
- 对于10位地址模式，当ADDR10EN (UI2C _PROTCTL [4])被置位时，如果从机地址是1111 0XXB，XX位用于和UI2C _DEVADDR [9:8]比较，检测到地址匹配时将会产生应答。从机等待第二个地址字节然后与UI2C _DEVADDR [7:0]比较，依据收到的10位地址来决定是否产生应答。用户必须处理好保留的的地址（详细描述请参考I^{2}C规格书）。仅支持地址1111 0XXB。在这些条件下，当地址匹配时，SLASEL (UI2C _PROTSTS [14])会被置位。SLASEL (UI2C _PROTSTS [14])会被一个（重复）起始或停止条件自动的清除。
- I^{2}C端口有两组地址掩码寄存器I2C_ADDRMSKn (n = 0~1)，这样就能支持多地址识别。当地址掩码寄存器中位设置为1的时候，表示不关心接收的地址，如果设置为0，表示接收相应地址位必须和地址寄存器是一样的。

6.20.5.4 数据传输

当从机设备地址和R/W位被成功识别到，就可以根据R/W 位所决定的方向按一字节一字节方式进行数据传输。每个字节传输完后，紧接着的第9个SCL 时钟周期会有一个应答信号位。如果从机上产生无应答信号(NACK)，主机可以产生停止信号来中止数据传输或者产生重复起始信号开始新一轮数据传输。

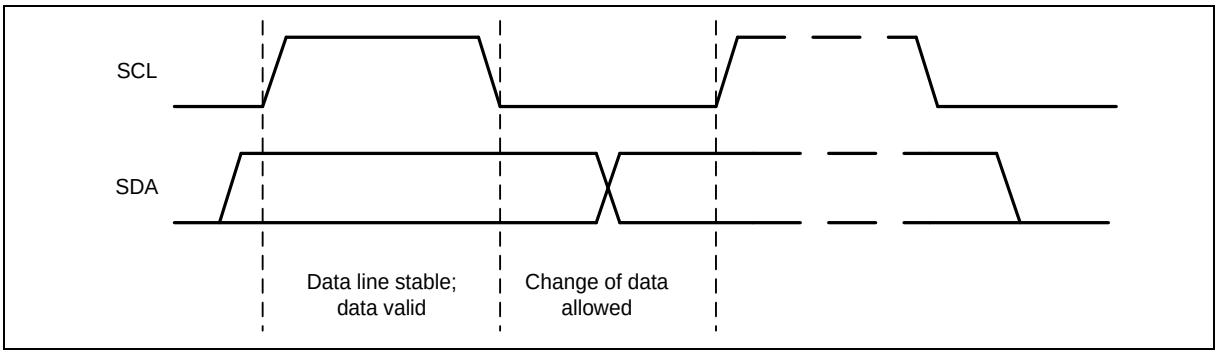


图 6.20-5I^{2}C 总线上的位传输

当主机作为接收设备时，发生无应答信号(NACK)，则从机将释放SDA 线，让主机产生停止信号或重复起始信号。

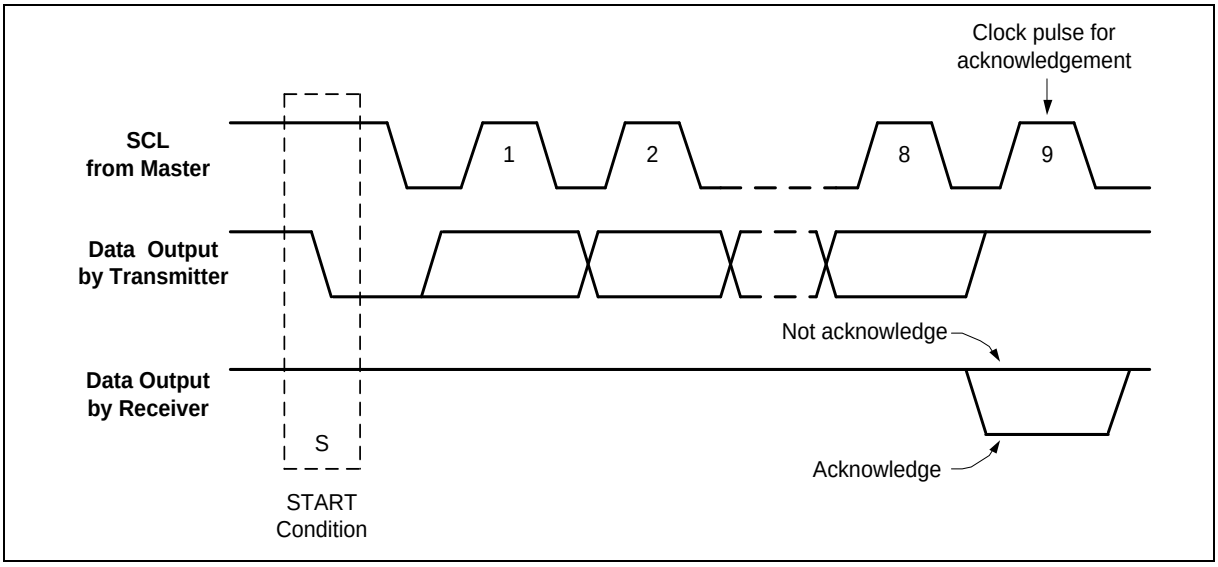


图 6.20-6 总线上的应答信号

6.20.5.5 时钟波特率位

当I^{2}C在主机模式下时，I^{2}C的波特率由UI2C _BRGEN寄存器决定，在从机模式下不用管。从机模式下，I^{2}C会自动识别来自主机的时钟的任何频率。在RCLKSEL, SPCLKSEL, PDSCNT和DSCNT中定义波特率

- RCLKSEL (UI2C_BRGEN [0])

  定义输入频率f_{REF_CLK}

- SPCLKSEL (UI2C_BRGEN[3:2])

  定义采样时钟f_{SAMP_CLK}的时钟源

- PDSCNT (UI2C_BRGEN [9:8])

  定义数据采样时间的长度(由f_{REF_CLK}经 1, 2, 3, 或 4分频)

- DSCNT (UI2C_BRGEN [14:10])

  定义采样时间每位时间的数目

  标准设定是由RCLKSEL = 0 (f_{REF_CLK} = f_{PCLK}), PTCLKSEL = 0 (f_{PROT_CLK} = f_{REF_CLK}) 和 SPCLKSEL = 2'b00 (f_{SAMP_CLK} = f_{DIV_CLK})决定。在这些条件下，波特率值如下：

$$f_{I2C} = f_{REF_CLK} \times \frac{1}{\text{CLKDIV}+1} \times \frac{1}{\text{PDSCNT}+1} \times \frac{1}{\text{DSCNT}+1}$$

  为了产生更慢的频率，设置PTCLKSEL = 1 (f_{PROT_CLK} = f_{REF_CLK2})选择除2，如下：

$$f_{I2C} = \frac{f_{REF_CLK}}{2} \times \frac{1}{\text{CLKDIV}+1} \times \frac{1}{\text{PDSCNT}+1} \times \frac{1}{\text{DSCNT}+1}$$

  如果SPCLKSEL = 2'b10 (f_{SAMP_CLK} = f_{SCLK}), 和 RCLKSEL = 0 (f_{REF_CLK} = f_{PCLK}), PTCLKSEL = 0 (f_{PROT_CLK} = f_{REF_CLK})，波特率如下：

$$f_{I2C} = f_{REF_CLK} \times \frac{1}{\text{CLKDIV}+1} \times \frac{1}{2} \times \frac{1}{\text{PDSCNT}+1} \times \frac{1}{\text{DSCNT}+1}$$

6.20.5.6 字节拉伸

如果设备被选用为主机/从机发送模式，在需要发送数据的时候发送缓冲TXDAT没有有效的数据被传输，设备就会在ACK位前拉低SCL到0。在软件写1到PTRG (UI2C _PROTCTL[5])中后等待周期结束。

6.20.5.7 多主机仲裁

在一些应用中，一个I^2C总线上有多个主机同时存取从机，并有可能同时在传送数据。I^2C是支持多主机模式，并包含有冲突检测和仲裁，防止数据损坏。

如果两个主机同时发命令，通过仲裁来决定哪个优先并继续发命令。仲裁是在SCL为高时在SDA上执行的。每一个主机都会检测总线上的SDA信号是否符合它产生的SDA信号。如果检测到总线上SDA为低，但应该为高，则这个主机将失去仲裁。设备在仲裁丢失后会产生SCL脉冲直到本字节结束，然后释放总线进入从机模式。仲裁可以一直进行到所有数据被传输完。这样意味着多主机系统中主机必须监控总线和做相应的处理。

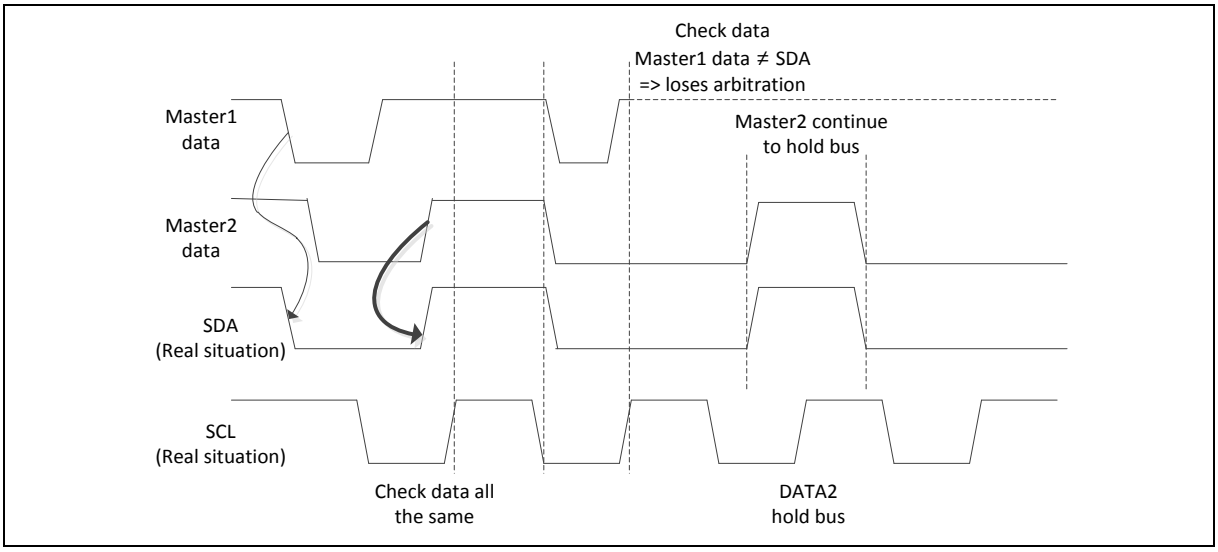


图 6.20-7 仲裁丢失

在地址和数据发送期间，主机发送器会对每个数据位在SCL的上升沿进行检查，如果发送的数据等于在SDA线上读到的数据，主机则继续占有总线，如果不等（发送值=1但是读的值是0），主机失去发送权利。该事件可由ARBLOIF (UI2C _PROTSTS [11])中断标志指示，如果使能ARBLOIEN (UI2C _PROTIEN [4])则会产生协议中断。

当传输仲裁丢失，软件必须再次初始化完整帧，以第一个地址字节加上“START”条件开始一个新的主机传输。在ACK位，也会进行仲裁。如果主机仲裁丢失并且匹配设备地址时，那么主机会转变成从机角色。

6.20.5.8 传输链

I2C总线协议在仲裁和从机传输阶段需要一种位响应机制，因此在传输时会加入一些延时，限制达到最大波特率，这主要取决于总线特性（总线负载、模块频率等等）。

移位时钟SCL由主机设备产生，输出到线上，经过输入阶段和输入滤波，然后边沿可以被检测到，SDA数据信号可以产生。SDA信号通过输出阶段到主机接收器部分。在被采样之前，其经过了输入阶段和输入滤波。

这个完整的循环（包括所有时间设置以获得稳定信号）必须在SCL变化前完成，这些延时在以f_{PCLK}或f_{PROT_CLK}时钟计算波特率时必须考虑在内，我们建议采用f_{PCLK}时钟

6.20.5.9 非应答和错误条件

在一个非应答(NACKIF (UI2C _PROTSTS [10]))或一个错误(ERRIF(UI2C _PROTSTS [12]))状况下，不会有进一步的发送产生。在用合适的值再一次配置传输（通过写TXDAT）使前一个事件起作用之前，用户软件不用使传输缓冲无效和禁用传输。

6.20.5.10 I^2C协议中断事件

下面协议相关的事件是在I^2C模式下产生的，也可以产生协议中断。

请注意UI2C _PROTSTS中的位不会被硬件自动地清除，必须用软件清除这是为了监控新事件的到来。

- 在帧的正确的位置处接收到起始条件(STARIF (UI2C _PROTSTS [8]))
- 在帧的正确的位置处停止条件被传输(STORIF (UI2C _PROTSTS [9]))
- 主机仲裁丢失(ARBLOIF (UI2C _PROTSTS [11]))
- 从机读请求(SLAREAD (UI2C _PROTSTS [15]))
- 应答接收(ACKIF (UI2C _PROTSTS [13]))
- 非应答接收(NACKIF (UI2C _PROTSTS [10]))
- 起始条件没有在帧期望的位置(ERRIF (UI2C _PROTSTS [12]))
- 停止条件没有在帧期望的位置(ERRIF (UI2C _PROTSTS [12]))

6.20.5.11 I2C操作

为了操作I^2C协议，必须考虑到下面的问题：

选择I^2C 模式

建议在FUNMODE (UI2C _CTL [2:0]) = 000B期间配置好I2C的所有参数且保持不变。I2C控制流程必须在FUNMODE (UI2C _CTL [2:0]) = 000B期间完成，避免输入意外的边沿信号。然后，可以通过设置FUNMODE (UI2C _CTL [2:0]) = 100B使能I2C模式。

步骤1. 设置 FUNMODE (UI2C_CTL [2:0]) = 000B

步骤2. 设置 FUNMODE (UI2C_CTL [2:0]) = 100B

引脚连接

引脚被用作SDA和SCL时必须由USCI控制器配置成开漏模式来支持I^2C总线的线与结构

注：*使能备用输出端口功能的步骤只能是在I^2C使能之后，这是为了避免意想不到的脉冲输出*

位时序配置

在标准模式(100 kBit/s) 下模块的最小频率是2MHz,然而在快速模式(400 kBit/s)最小频率是10MHz。此外，被用于消除脉冲的数字滤波阶段达到50ns,滤波器的频率需有20MHz。如果另一个I2C参与者使SCL低电平阶段拉长，SCL变高将不确定，最大为$1/f_{PROT_CLK}$。请注意SCL最大频率是SAMP_CLK/2并且必须设置SPCLKSEL (UI2C _BRGEN [3:2])为0选择$f_{SAMP_CLK} = f_{DIV_CLK}$.

数据格式配置

数据格式必须配置为8数据位(DWIDTH (UI2C _LINECTL [11:8]) = 8)并且MSB优先(LSB (UI2C _LINECTL [0]) = 0).最后UI2C _LINECTL必须设置为0x800

控制流程

片上I^2C端口支持三种模式主机模式、从机模式和广播模式。

应用中，I²C 端口可以作为主机和从机。在从机模式，I²C端口硬件会查找自身从机地址和广播呼叫地址，如果这两个地址的任一个被检测到，并且从机想要从主机接收或向主机发送数据(通过设置AA位)，应答脉冲将会在第9个时钟发出，此时，如果中断使能，则主机和从机设备上都会发生一次中断请求。当微控制器想要成为总线主机时，在进入主机模式之前，硬件需等待到总线空闲，以保证合理的从机动作不会被打断。在主机模式下，如果总线仲裁丢失，I²C 端口立即切换到从机模式，并可以在同一次串行传输过程中检测自身从机地址。

为控制I²C总线的各种模式传输，用户需要按照寄存器UI2C _PROTSTS的当前状态码来设置UI2C _PROTCTL, UI2C _PROTIEN, TXDAT 寄存器。换句话说，每个I²C总线动作都要检查UI2C _PROTSTS寄存器的当前状态，然后再设置UI2C _PROTCTL, UI2C _PROTIEN, TXDAT寄存器执行总线动作。最后，通过UI2C _PROTSTS检查响应状态。

在中断标志清除后，寄存器USCI_PROTCTL 的 STA、STO 和 AA 用来控制 I2C 硬件的下一个状态。当完成一个新的动作，UI2C _PROTSTS的状态将被更新。如果I²C中断控制位UI2C _PROTIEN被设置，新状态适当的动作和软件分支可以在中断服务例程完成。

图6.20-8显示当前I²C STARIF (UI2C _PROTSTS [8])被硬件置1，然后设置TXDAT = SLA+W (从机地址 + 写位), (PTRG, STA, STO, AA) = (1, 0, 0, x)来发送地址到I²C总线，写1到STARIF (UI2C _PROTSTS [8])清标志。如果在总线上一个从机地址匹配并且应答ACK，ACKIF (UI2C _PROTSTS [13])会被更新。

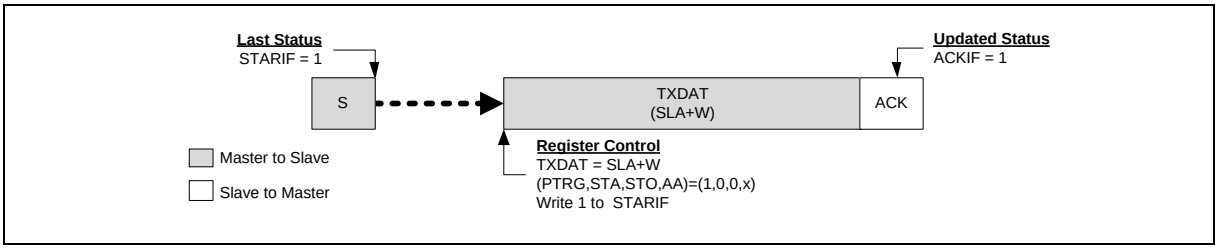


图 6.20-8 依据 I²C 状态控制 I²C 总线

总线上的数据传输

图6.20-9表示7位地址情况下主机向从机传输数据。主机发出一个7位地址和1位写指示，表示主机想要传送数据给从机。从机回复应答给主机之后，主机继续传输数据。

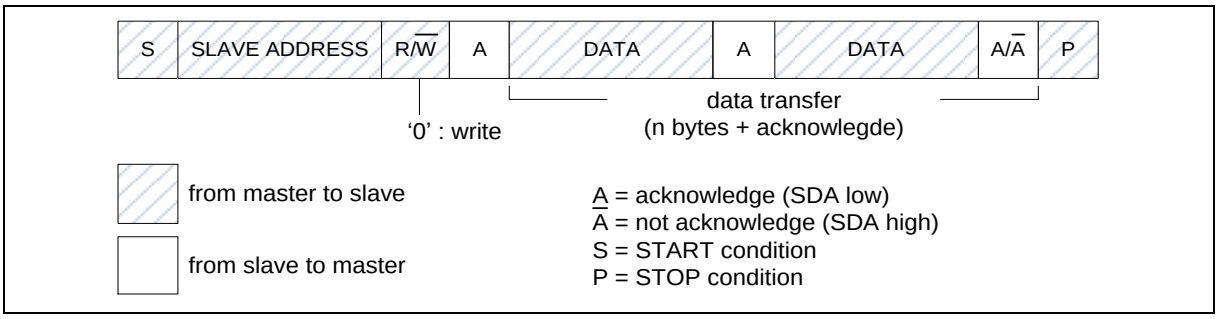


图 6.20-9 7 位地址情况下主机向从机传输数据

图6.20-10表示7位地址情况下主机向从机读取数据。主机发7位地址寻址和1位读指示，表示主机要向从机读取数据，从机返回应答给主机后，就开始给主机传输数据。

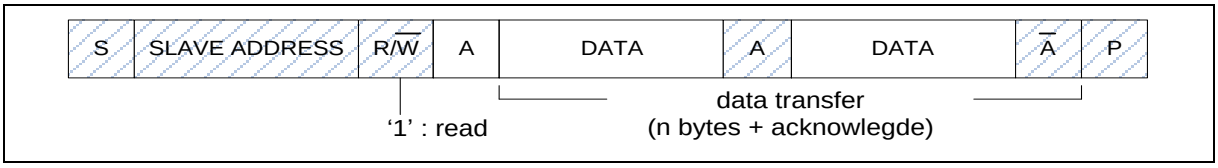


图 6.20-10 7 位地址情况下主机向从机读取数据

图6.20-11表示10位地址情况下主机向从机传输数据。主机发送10位地址。首字节是有10位地址指示(5'b11110)和2位地址再加上写指示，第二字节是剩下的8位地址。在第二字节之后，主机继续传输数据。注意7位和10位地址设备可以工作在同一总线上。

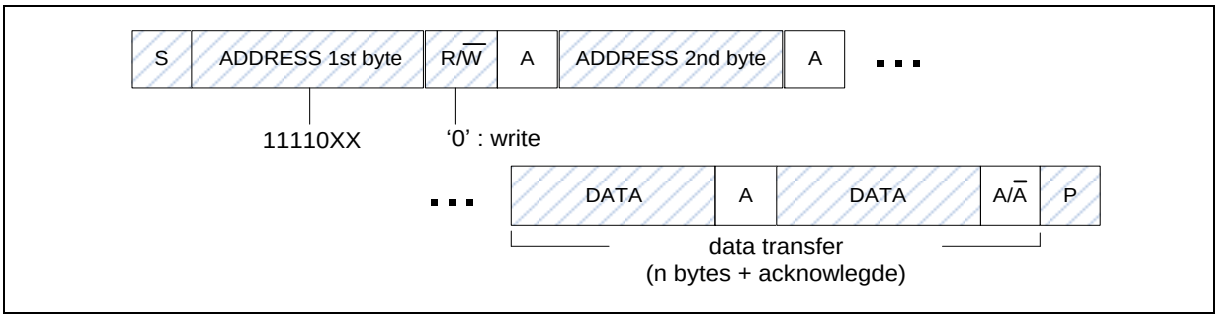


图 6.20-1110 位地址情况下主机向从机传输数据

图6.20-12表示10位地址情况下主机向从机读取数据。首先主机发送10位地址给从机，在主机传输带有读指示的第一个字节之后。在带有读指示的第一个字节之后，从机就开始给主机传输数据。

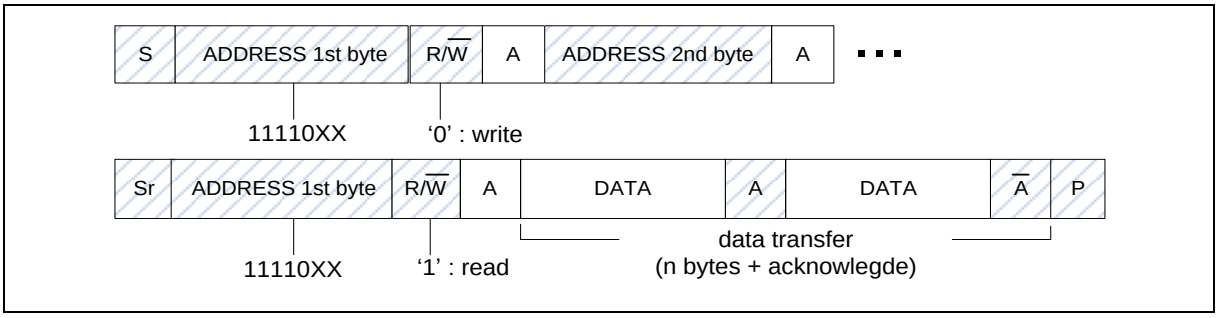


图 6.20-1210 位地址情况下主机向从机读取数据

主机模式

图6.20-13和图6.20-14是I^2C主机模式所有可能的协议展示。用户需要遵循恰当的流程来实现I^2C协议。

换句话说，用户可以发送一个起始信号到总线，I^2C总线将设置成主机传输(MT)模式(图6.20-13)或主机接收(MR)模式((图6.20-14)。起始信号设置成功后新的状态被设置STARIF (UI2C _PROTSTS [8])。起始信号后，用户可以发送从机地址，读/写位，数据和重复起始，停止来执行I^2C协议。

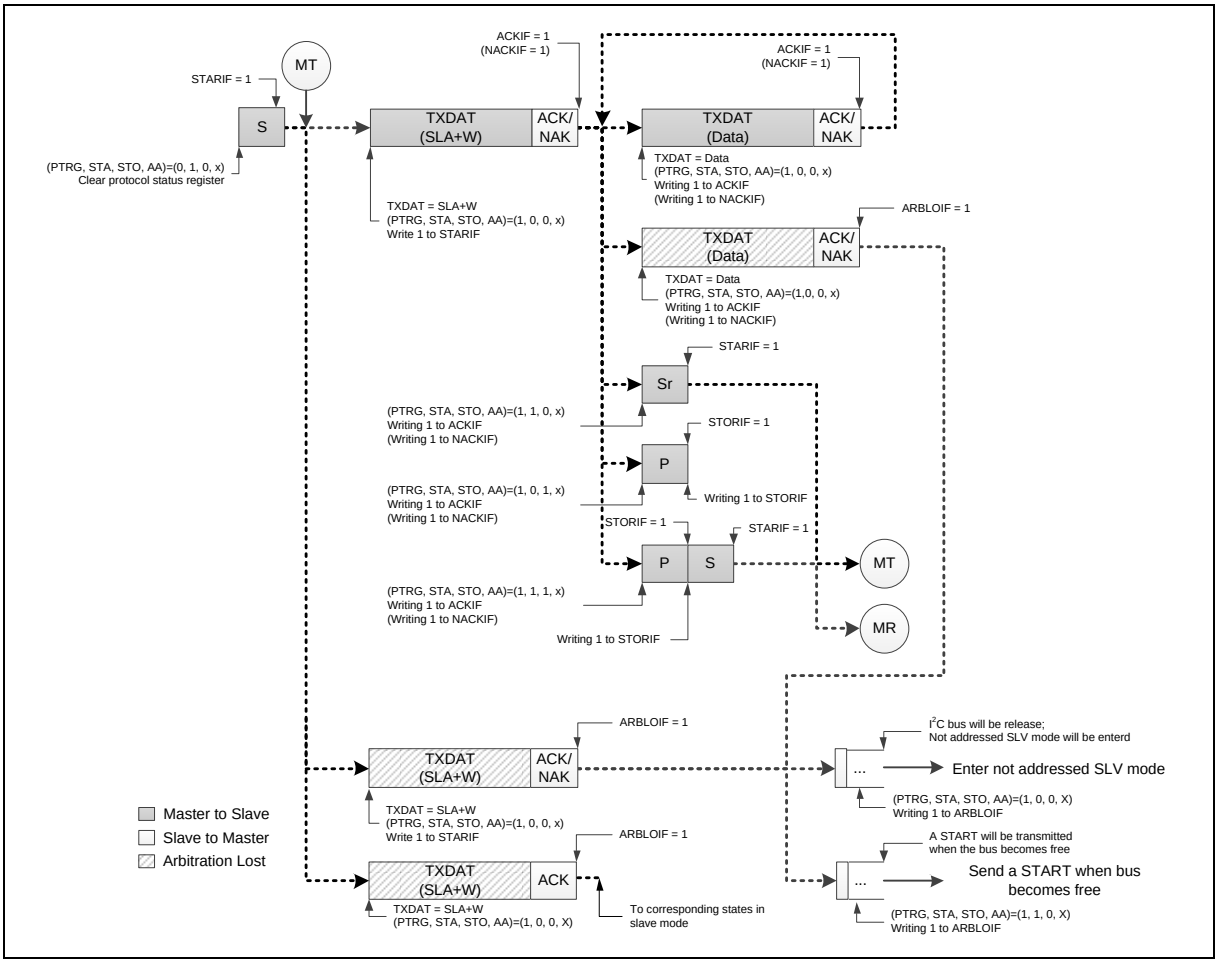


图 6.20-13 7 位地址主机发送模式控制流程

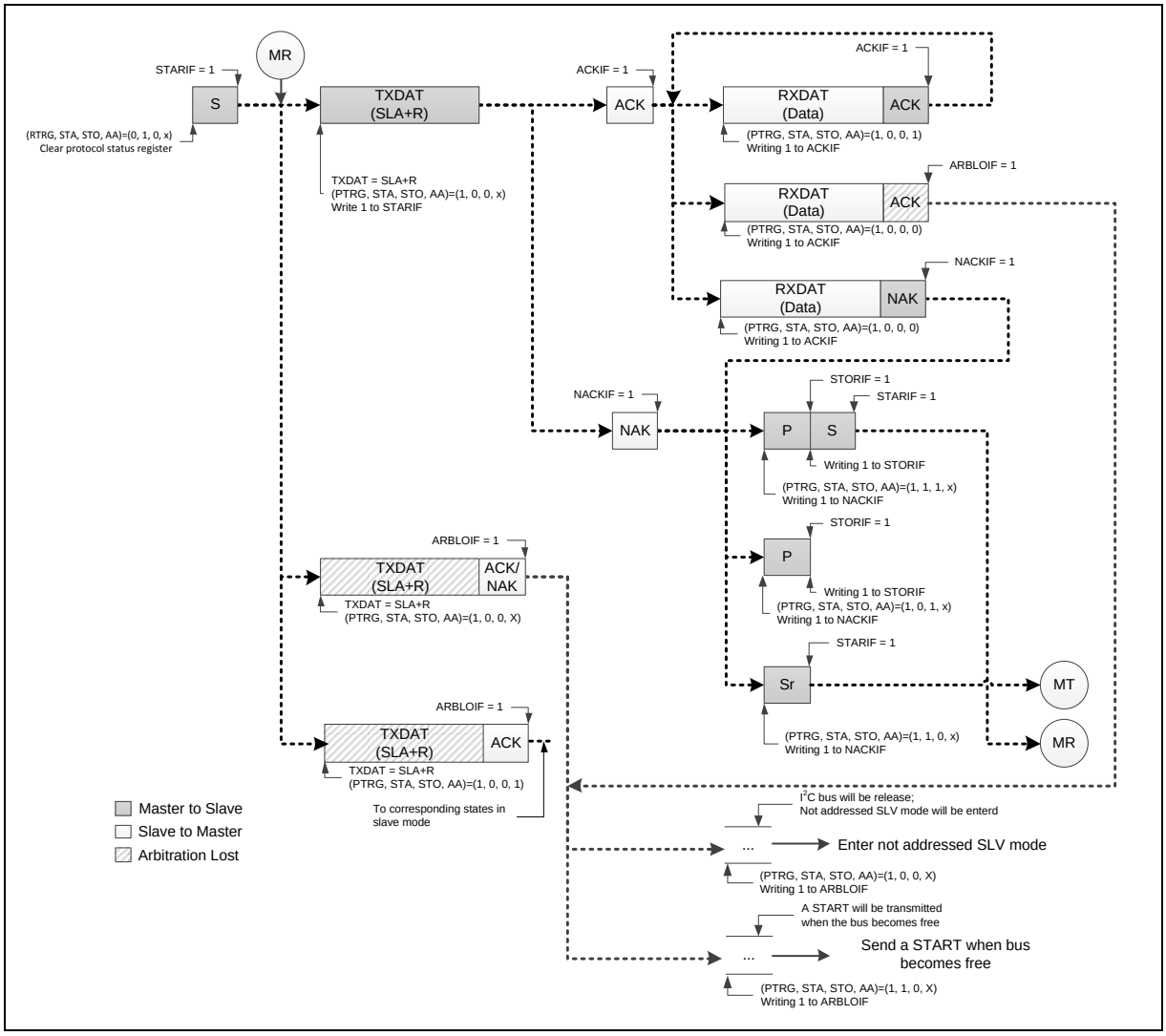


图 6.20-147 位地址主机接收模式控制流程

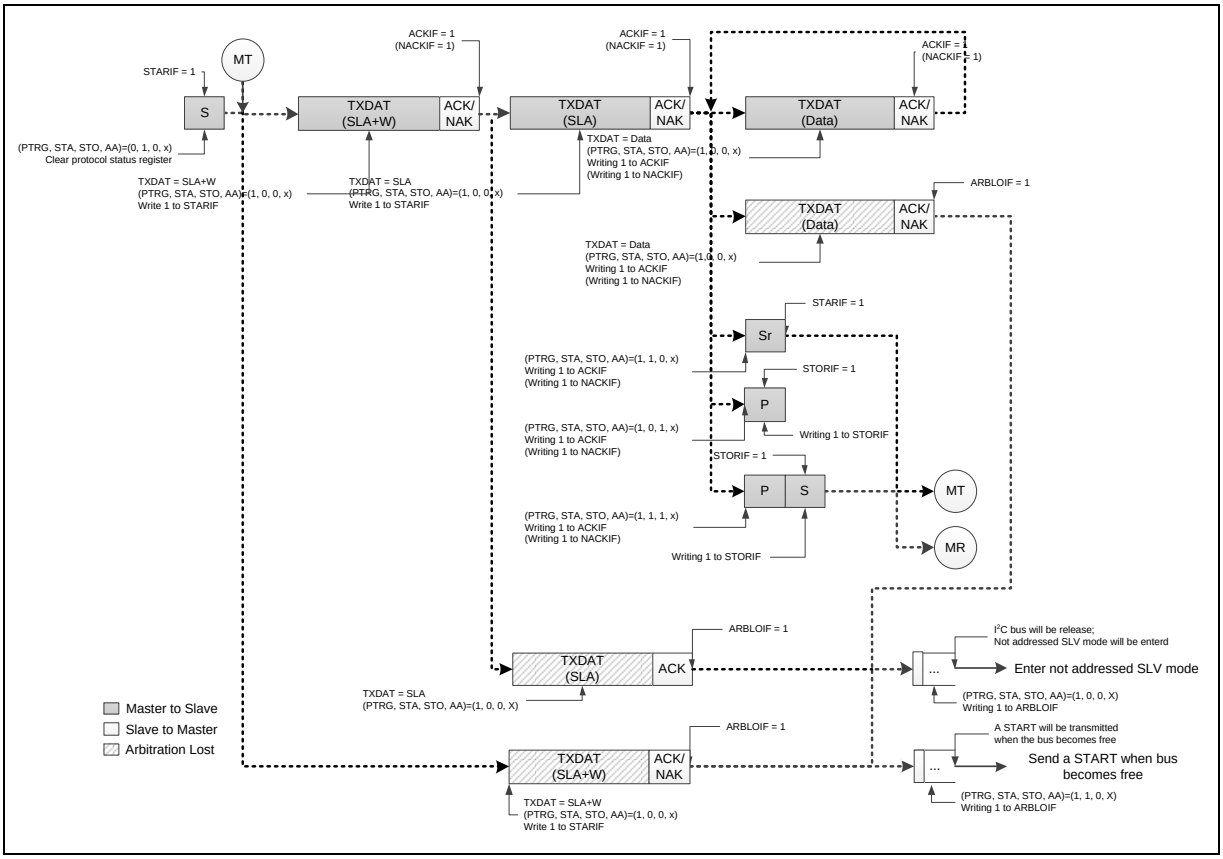


图 6.20-15 10 位地址主机发送模式控制流程

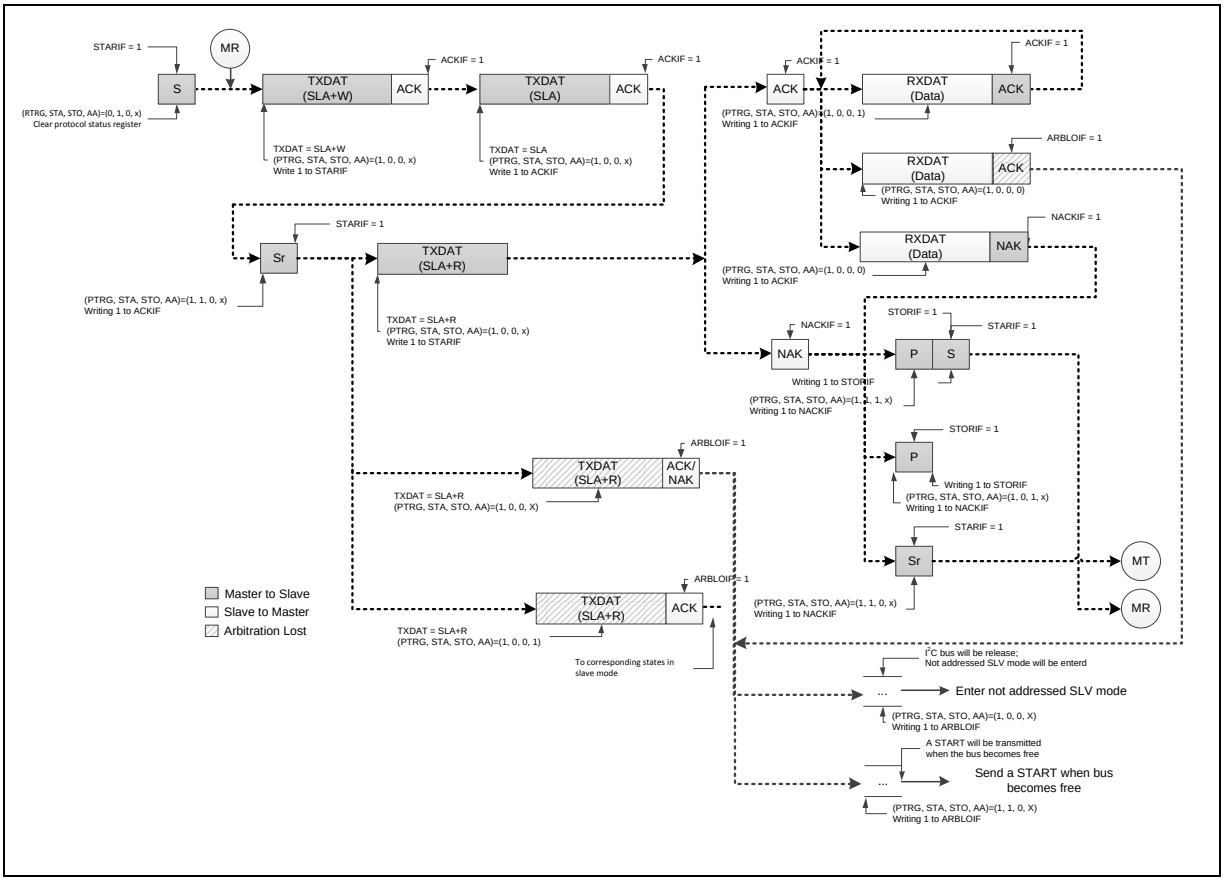


图 6.20-16 10 位地址主机接收模式控制流程

如果I²C在主机模式并且仲裁丢失，ARBLOIF (UI2C _PROTSTS [11])位被置位。当总线空闲时，用户可以写1到ARBLOIF (UI2C _PROTSTS [11])并且设置(PTRG, STA, STO, AA) = (1, 1, 0, X)发送起始信号来重新开始主机操作。否则，用户可以写1到ARBLOIF (UI2C _PROTSTS [11])并且设置(PTRG, STA, STO, AA) = (1, 0, 0, X)来释放总线，并进入无地址从机模式。

从机模式

复位后默认情况下，I²C不会被寻址，并且不会识别I²C总线上的地址。用户可以通过UI2C _DEVADDRn设置从机地址和设置(PTRG, STA, STO, AA) = (1, 0, 0, 1)来让I²C识别主机发送的地址。图6.20-17所示为I²C从机模式的所有可能的流程。用户需要遵循（图6.20-17）的流程来实现他们的I²C协议。

如果在主机模式总线仲裁丢失，I²C端口立即切换到从机模式，并且在同一串行传输中识别自有的从机地址。如果在仲裁丢失后识别到地址是SLA+W（主机想写数据到从机）或是SLA+R（主机向从机读数据），ARBLOIF会被置1。

I2C控制器支持两个从地址匹配标志，分别是UI2C _ADMAT [1：0]寄存器中的ADMAT0和ADMAT1。每个控制寄存器代表使用哪个地址，并设置1通知软件。

注：I²C通信期间，在从机模式下，当对PTRG (UI2C _PROTCTL [5])标志写‘1’时，SCL时钟将被释放。

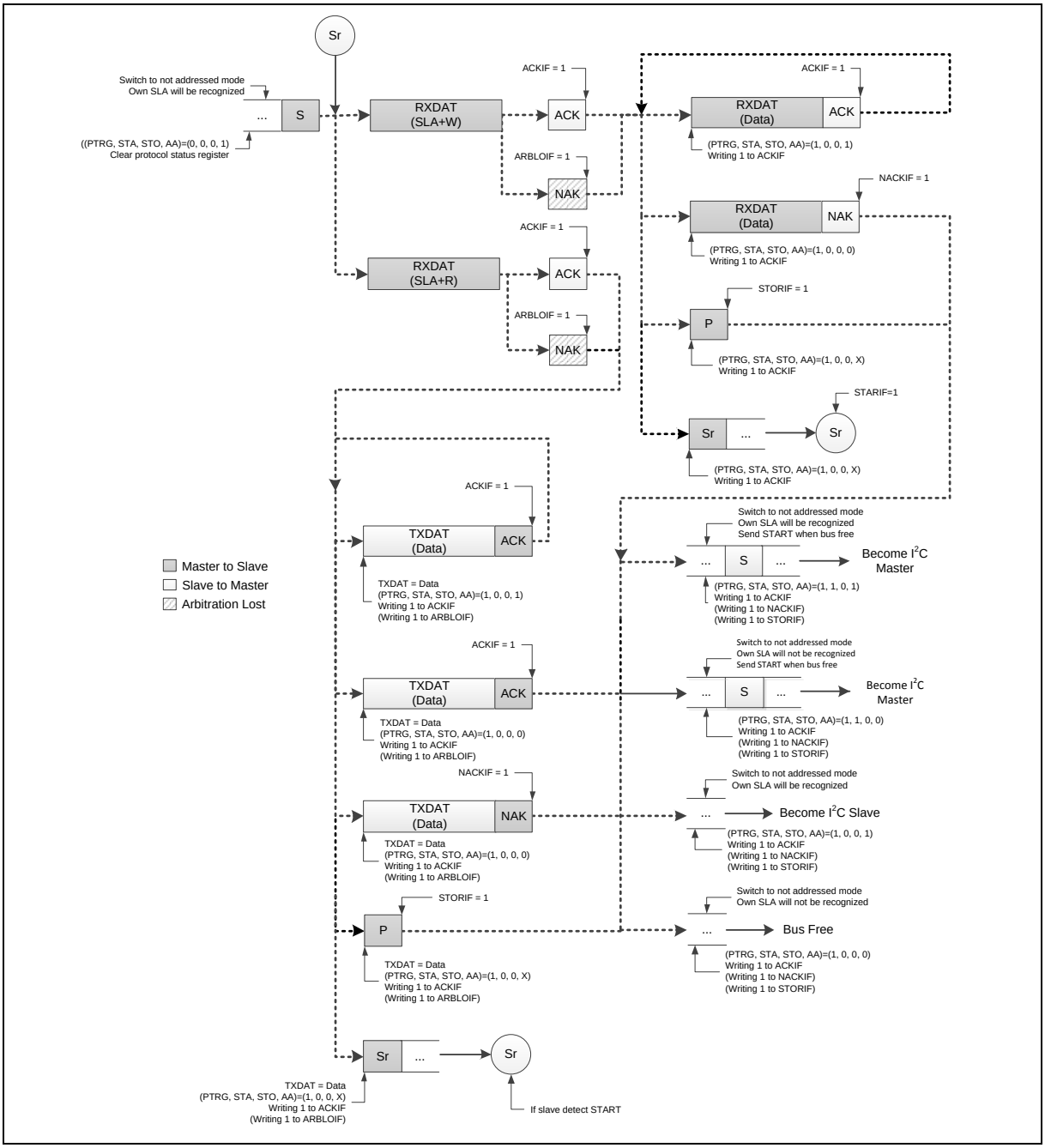


图 6.20-17 7 位地址从机模式控制流程

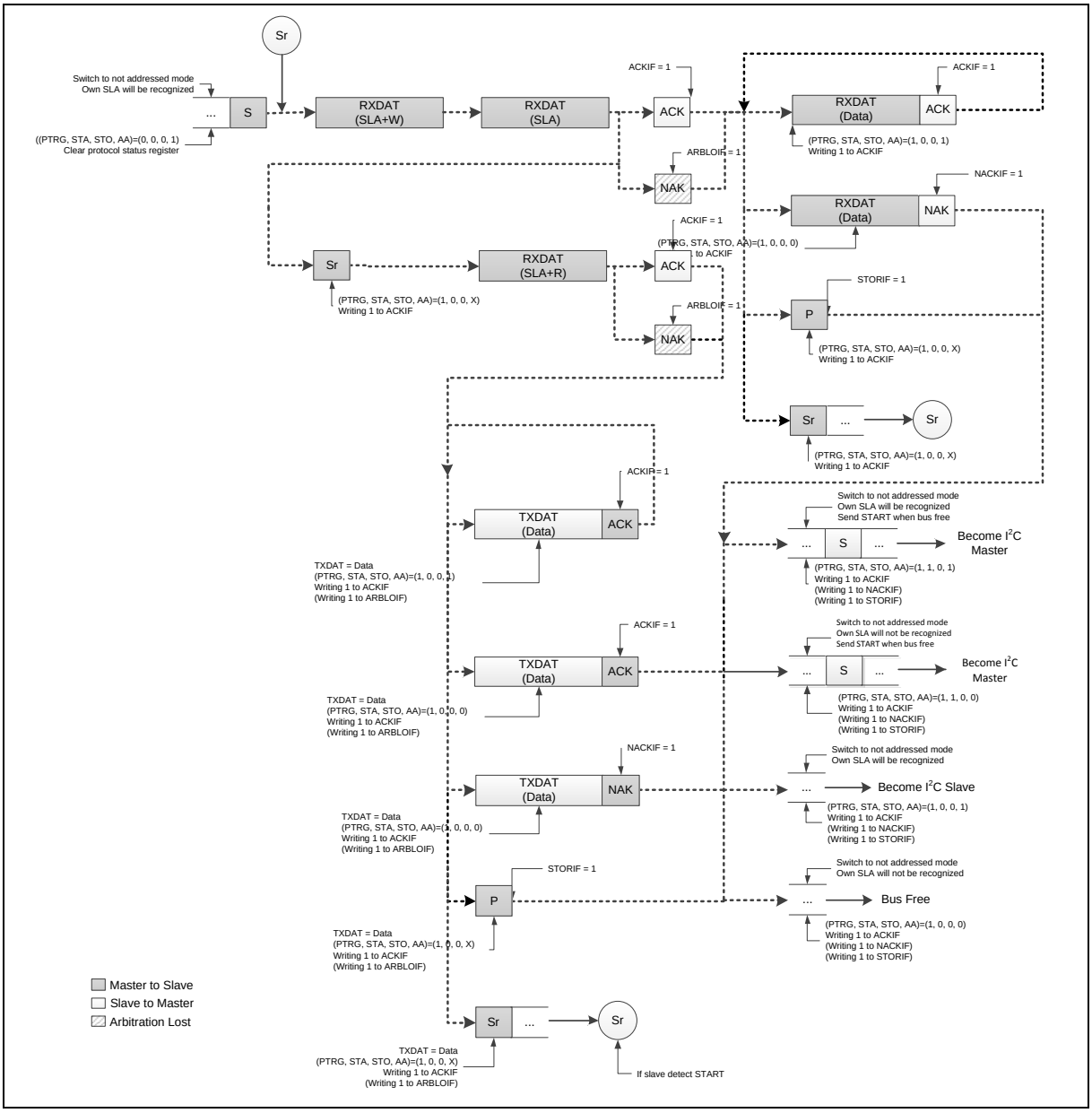


图 6.20-18 10 位地址从机模式控制流程

如果I^{2}C在可寻址从机模式发送和接收数据时，收到停止或重复起始信号，STORIF (UI2C _PROTSTS [9]) 或 STARIF (UI2C _PROTSTS [8])会被置位。当STARIF (UI2C _PROTSTS [8])被设置时，用户可以遵循如上图NACKIF (UI2C _PROTSTS [10])的操作。

注：从机获得NACKIF (UI2C _PROTSTS [10]) 和起始/停止符号包括 STARIF (UI2C _PROTSTS [8]) 和 STORIF (UI2C _PROTSTS [9])中断状态标志后，从机可以切换到无地址模式，自身SLA不会被辨识。如果进入这种状态，从机不再接收主机任何信号或地址。在这种状态， I^{2}C需要设置FUNMODE (UI2C _CTL [2:0]) = 000B离开这个状态。

广播呼叫模式 (GC)

如果GCFUNC 位 (UI2C _PROTCTL [0])被设置，I^{2}C端口硬件将响应广播呼叫地址(0x00)。用户可以通过清GC位来禁止广播呼叫功能。当I^{2}C在从机模式时且GC位被设置，可以接收主机地址(0x00)的广播呼叫，将遵循广播模式状态。

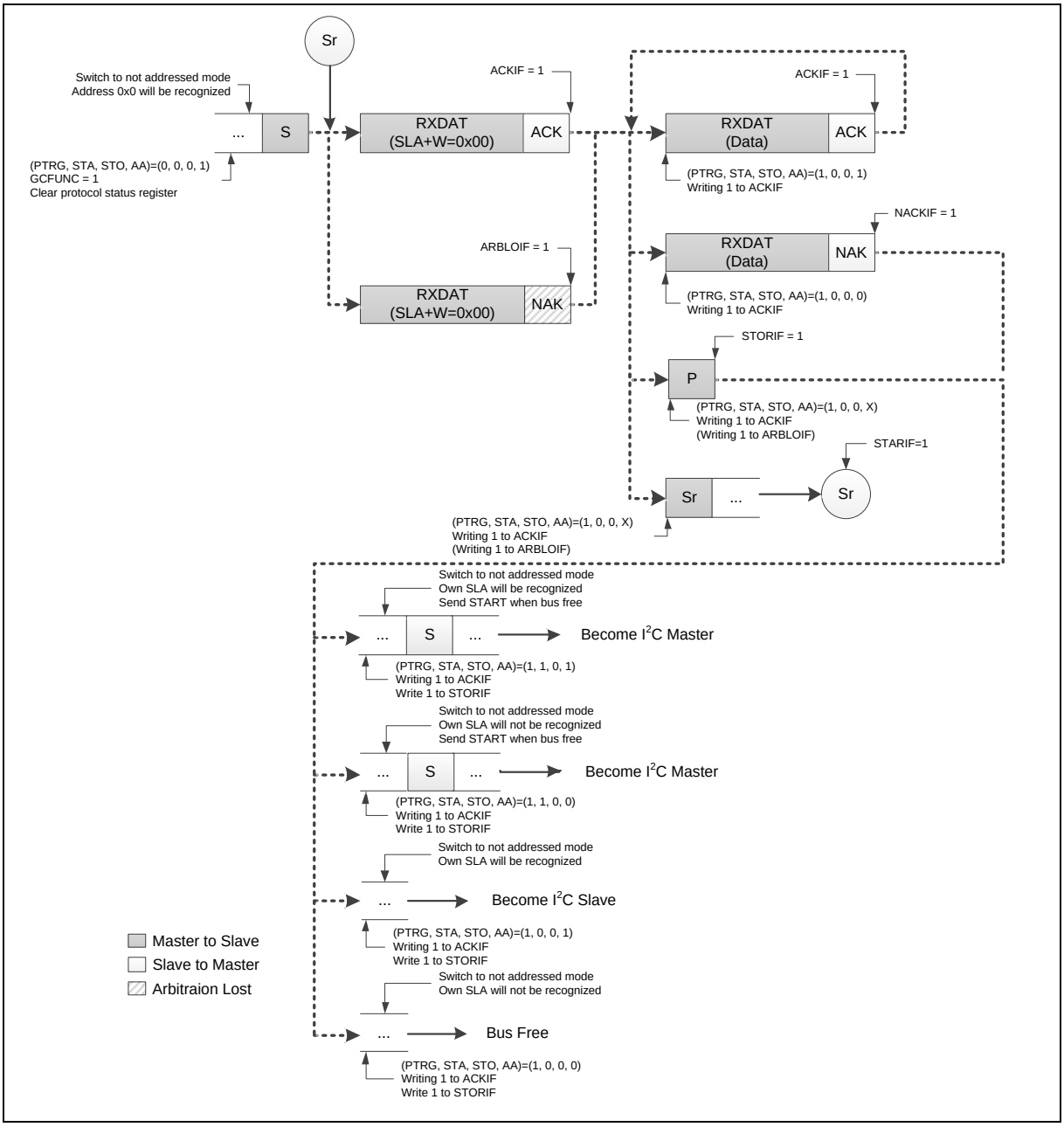


图 6.20-19 7 位地址的 GC 模式

如果I^{2}C在广播呼叫模式接收数据时收到停止或重复起始信号， STORIF (UI2C _PROTSTS [9]) 或STARIF (UI2C _PROTSTS [8])会被置位。当得知STORIF (UI2C _PROTSTS [9]) 或STARIF (UI2C _PROTSTS [8])被置位时，用户可以遵循如上图NACKIF (UI2C _PROTSTS [10])的流程处理。

注：从机获得NACKIF (UI2C _PROTSTS [10]) 和起始/停止符号包括 STARIF (UI2C _PROTSTS [8]) 和STORIF (UI2C _PROTSTS [9])中断状态标志后，从机可以切换到无地址模式，自身SLA不会被辨识。如果进入这种状态，从机不再接收主机任何信号或地址。在这种状态， I^{2}C需要设置FUNMODE (UI2C _CTL [2:0]) = 000B离开这个状态。

协议功能描述

监视器模式

当I²C进入监视器模式，在每次帧接收后设备总是返回NACK给主机即使是地址匹配。此外，设备会存储接收到的任何数据包括地址、命令码和数据。

监视器模式下的中断

当MONEN (UI2C _PROTCTL [9])被设置，所有中断处理与正常一样。注意当有初始化START时会发生第一个中断，这个与I²C从机不一样，但是其它中断是一样的。

地址匹配侦测之后，中断会在每次数据接收之后（如同从机模式控制流程一样）或是在从机读传输模块信任的每个字节传输之后产生。在第二个情况中，数据寄存器实际上会包含总线上一些其它被主机寻址的从机发送的数据。如果用户想监控其它设备，用户可以设置地址掩码和监视器。

如果监视器没有时间响应中断，当SCLOUTEN (UI2C _PROTCTL [8])被置1时SCL信号会被拉低。当SCLOUTEN (UI2C _PROTCTL [8])被置1时，用户必须设置PTRG (UI2C _PROTCTL [5])释放总线。如果SCLOUTEN (UI2C _PROTCTL [8])没有被置1，用户就不需要置PTRG (UI2C _PROTCTL [5])为1。

当设备地址匹配时，但是设备会响应NACK，该地址会被接收到缓冲中并且会产生NACK中断。

接下来的所有中断，处理器可能需要读数据寄存器来看总线上实际传输的是什么。

监视器模式下仲裁丢失

在监视器模式下，I²C模块不会响应由总线主机产生的ACK信息请求。总线上的其它从机会去响应。软件应该知道在模块处在监视器模式下，不能响应检测到的任何仲裁丢失情况。

可编程的建立和保持时间

为了保证一个正确数据的建立和保持时间，则时序必须可以配置。通过编程HTCTL (UI2C _TMCTL[24:16])来配置保持时间，通过编程STCTL (UI2C _TMCTL[8:0])来配置建立时间

延时时序参考外设时钟(PCLK).当设备拉住主机时钟时，建立和保持时间的配置值不会被影响。

用户需要集中精力在建立和保持时间配置的限制上，时序设置必须遵从I²C协议。一旦建立时间配置大于设计限制，也就是如果设置的建立时间让SCL输出小于三个PCLK，由于SCL采样为三次，那么I²C控制器就不能正常工作。一旦保持时间配置大于I²C时钟的限制，I²C将会发送总线错误。在设置时序前建议用户结合传输速率和协议计算好合理的时序。表6.20-1显示了I²C传输速率与PCLK之间的关系，表格呈现的数是一个时钟周期包含多少PCLK。建立和保持时间的配置哪怕在我们设计中可编程一些极限值，但是用户必须要遵从I²C的标准协议。

I²C 传输速率 PCLK	100k	200k	400k	800k	1200k
12MHz	120	60	30	15	10
24MHz	240	120	60	30	20
48MHz	480	240	120	60	40
72MHz	720	360	180	90	60

表 6.20-1 I²C 传输速率与 PCLK 之间的关系

对于建立时间错误调整范例，我们假设一个SCL周期包含10个PCLK，且设置STCTL (UI2C _TMCTL[8:0]) 为3拉伸三个PCLK用于建立时间设置。建立时间设置限制：ST_{limit} = (UI2C _BRGEN[25:16]+1) - 6.

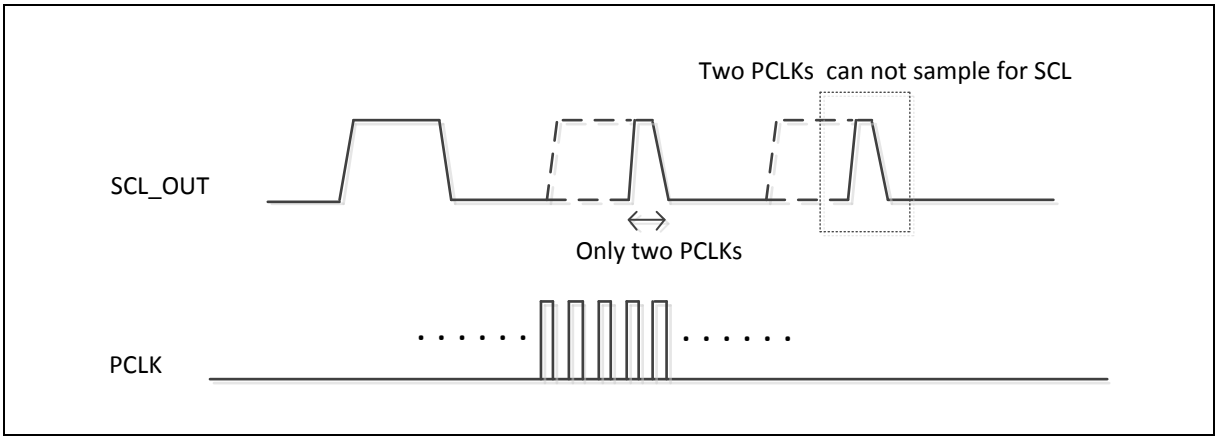


图 6.20-20 建立时间错误调整

对于保持时间错误调整范例，我们使用I²C传输速率=1200K和PCLK=72MHz，SCL的高电平/低电平周期=60PCLK。当我设置HTCTL (UI2C _TMCTL[24:16])为63和STCTL (UI2C _TMCTL[8:0])为0时，然后SDA输出延时将会超过SCL为高的周期，产生总线错误。保持时间设置限制：HT_{limit} = (UI2C _BRGEN[25:16]+1) - 9.

注：保持时间仅仅在主机模式下工作，在从机模式下，USCI-I2C HTCTL (UI2C_TMCTL[24:16])设置为0。

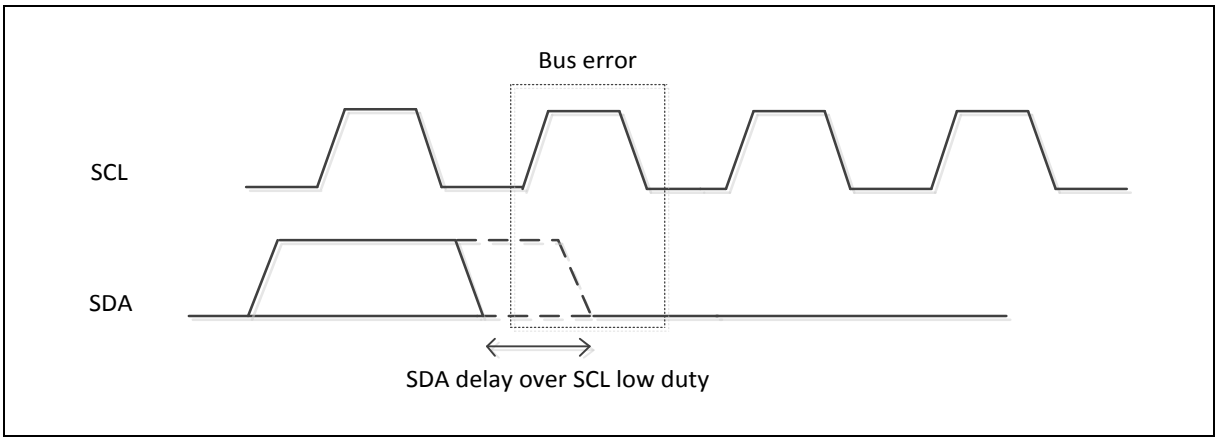


图 6.20-21 保持时间错误调整

***I²C** 超时功能*

系统提供一个10位超时的计数器TOCNT (UI2C _PROTCTL [25:16])来处理当I2C总线锁死时的情况。当超时计数器计数功能使能后，计数器开始上计数直至等于TOCNT (UI2C _PROTCTL [25:16])并向CPU产生I²C中断，清除TOIEN (UI2C _PROTIEN [0])为0停止计数。用户需要写1清TOIF(UI2C _PROTSTS[5])为0。当超时计数器使能，对TOIF写1会使计数器复位，TOIF被清除后重新开始计数。对于超时计数器TOCNT (UI2C _PROTCTL [25:16])、])，T_{TOCNT} = (TOCNT (UI2C _PROTCTL [25:16]) +1) x32 (5位) x T_{PCLK}，请参考图6.20-22。请注意时间计数器时钟源TMCNTSRC (UI2C _ BRGEN [5])必须设置为0。

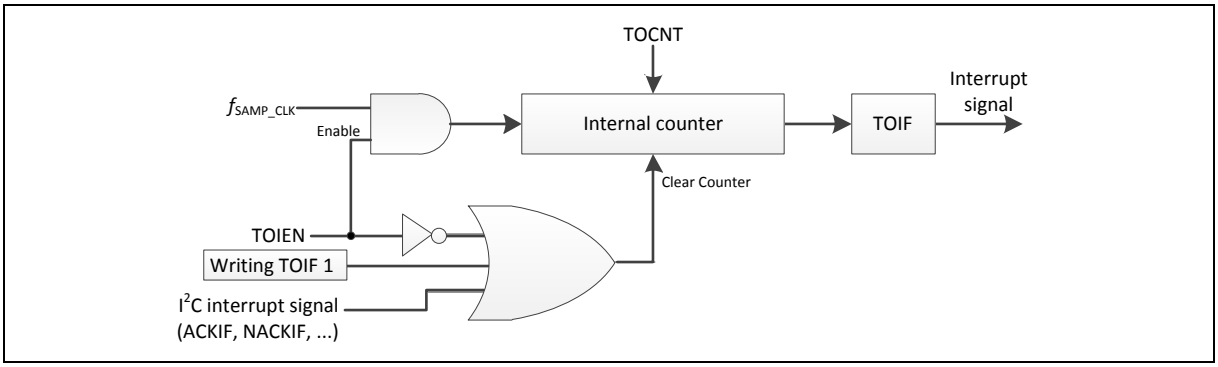


图 6.20-22 I²C 超时计数模块框图

唤醒功能

当芯片进入掉电模式并且WKEN (WKCTL[0])被设置为1时，其它I²C主机可以通过寻址I²C设备唤醒芯片，在进入睡眠前必须配置相关设置。地址匹配帧的ACK位阶段是在掉电情况下完成的。当地址与设备地址匹配并且ACK周期完成，控制器会拉低SCL。直到用户清除WKAKDONE位，SCL释放。如果SCL的频率是低速并且系统从地址匹配帧中唤醒，用户需要检查该位来确认该帧传输已经完成然后去处理唤醒程序。因此，当芯片被设备地址寄存器(UI2C _DEVADDRn)其中之一的地址匹配唤醒时，用户需要检查WKAKDONE (UI2C _PROTSTS [16])是否被置1来确认地址字节已完成。WKAKDONE位表明在掉电时候ACK阶段已完成。请注意用户必须在清除WKAKDONE为0之后清除WKUPIF。

WRSTSWK (UI2C _PROTSTS [17])位用来记录在地址匹配唤醒帧中的读/写命令。在系统被地址匹配帧唤醒后，用户读该位的状态来准备下次传输数据(WRSTSWK = 0)或是等待到来的数据 (WRSTSWK = 1)及时地被存储。

当系统被其他的I²C主机设备唤醒时，WKF(UI2C _WKSTS[0])被置位表示该事件发生。用户需要写“1”来清除此位。

EEPROM 随机读取范例

通过下面的步骤来配置USCI0_I²C相关寄存器，来使用I²C从EEPROM读取数据。

1. 将USCI0_I2C多功能引脚设置成SCL和SDA引脚，多功能引脚配置参考基本配置。
2. 使能USCI0 APB时钟。
3. 通过设置USCI0RST=1来复位I²C0控制器，然后通过设置USCI0RST=0让USCI控制器回归正常操作，复位控制器配置参考基本配置。
4. 通过设置寄存器UI2C _CTL中的FUNMODE =100使能USCI0_I²C控制器。
5. 通过在UI2C _BRGEN寄存器写USCI0_I²C时钟分频值。
6. 在“NVIC_ISR”寄存器中来设置USCI0 IRQ。
7. 设置寄存器UI2C _PROTIEN的ACKIEN, ERRIEN, ARBLOIEN, NACKIEN, STORIEN, STARIEN和TOIEN使能I²C中断
8. 设置USCI地址寄存器“UI2C_DEVADDR0 ~ UI2C_DEVADDR1”。

随机读操作是存取EEPROM的其中一种方法。这个方法是允许主机存取EEPROM的任何一个地址。图6.20-23显示对EEPROM随机读取操作。

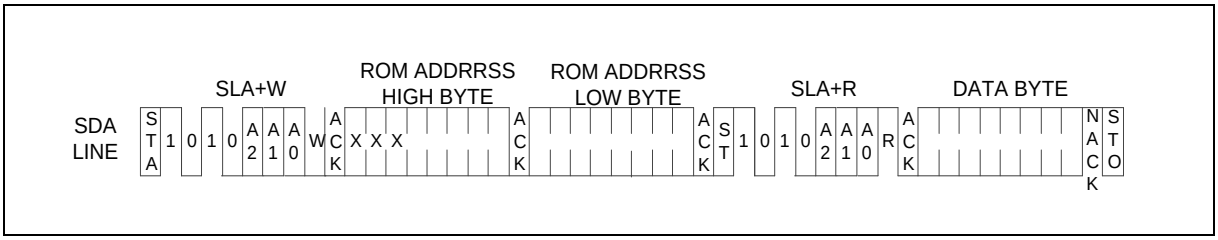


图 6.20-23 EEPROM 随机读取

图6.20-24显示怎样使用I²C控制器执行EEPROM随机读取操作协议

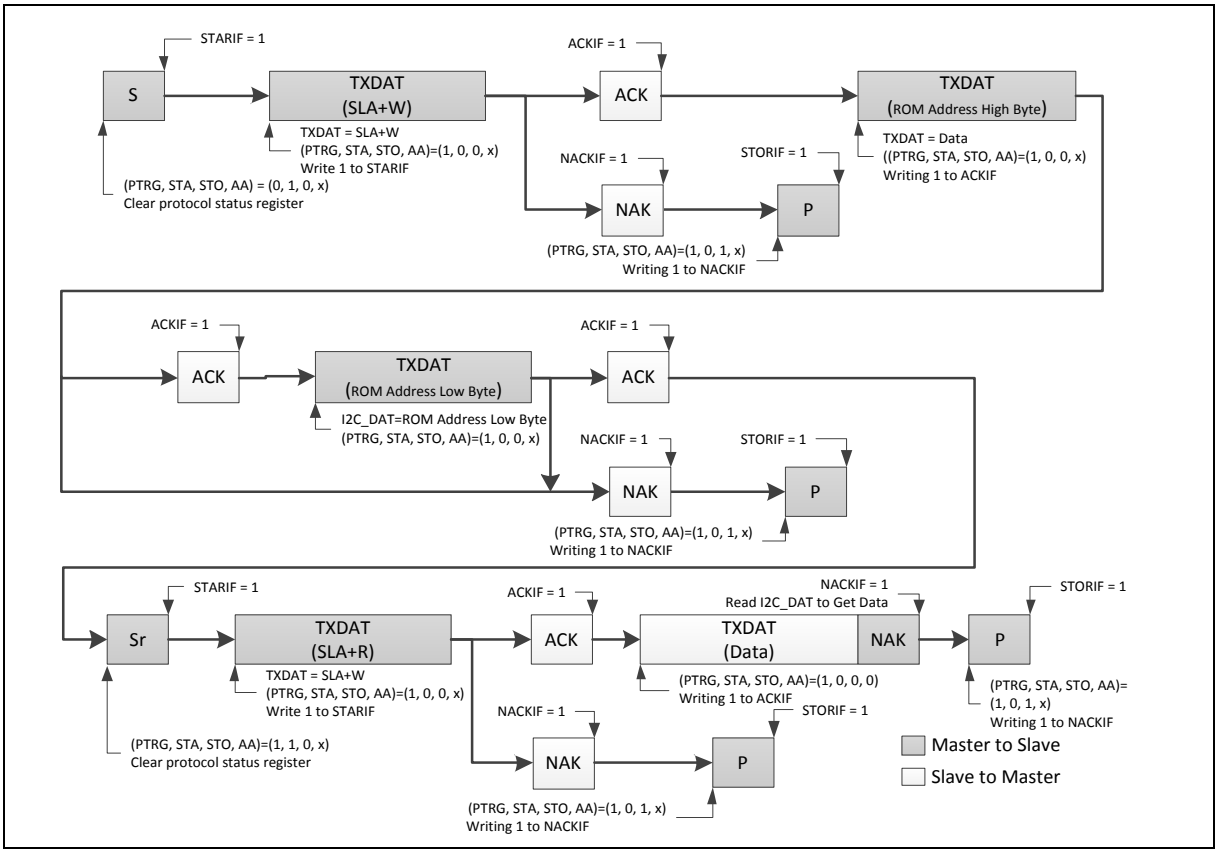


图 6.20-24 随机读取协议

I²C控制器作为主机发送起始信号到总线，然后发送SLA+W (从机地址 + 写位)到EERPOM，跟着由两个字节数据地址来设置EEPROM被读的地址。最后，重复起始信号跟着SLA+R被发送来向EEPROM读取数据。

6.20.6 寄存器映射

R:只读, **W:**只写, **R/W:读/写**

寄存器	偏移	R/W	描述	复位值
UI2C_I2C基地址: **UI2Cn_BA = 0x400D_0000 + (0x1000 * n)** **n= 0, 1**				
UI2C_CTL	UI2Cn_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000
UI2C_BRGEN	UI2Cn_BA+0x08	R/W	USCI 波特率发生器寄存器	0x0000_3C00
UI2C_LINECTL	UI2Cn_BA+0x2C	R/W	USCI 线控制寄存器	0x0000_0000
UI2C_TXDAT	UI2Cn_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000
UI2C_RXDAT	UI2Cn_BA+0x34	R	USCI 接收数据寄存器	0x0000_0000
UI2C_DEVADDR0	UI2Cn_BA+0x44	R/W	USCI 设备地址寄存器0	0x0000_0000
UI2C_DEVADDR1	UI2Cn_BA+0x48	R/W	USCI 设备地址寄存器1	0x0000_0000
UI2C_ADDRMSK0	UI2Cn_BA+0x4C	R/W	USCI 设备地址掩码寄存器0	0x0000_0000
UI2C_ADDRMSK1	UI2Cn_BA+0x50	R/W	USCI 设备地址掩码寄存器1	0x0000_0000
UI2C_WKCTL	UI2Cn_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000
UI2C_WKSTS	UI2Cn_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000
UI2C_PROTCTL	UI2Cn_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000
UI2C_PROTIEN	UI2Cn_BA+0x60	R/W	USCI 协议中断使能寄存器	0x0000_0000
UI2C_PROTSTS	UI2Cn_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000
UI2C_ADMAT	UI2Cn_BA+0x88	R/W	I^2C从属匹配地址寄存器	0x0000_0000
UI2C_TMCTL	UI2Cn_BA+0x8C	R/W	I^2C时序配置控制寄存器	0x0000_0000

6.20.7 寄存器描述

USCI 控制寄存器(UI2C_CTL)

寄存器	偏移	R/W	描述	复位值
UI2C_CTL	UI2Cn_BA+0x00	R/W	USCI控制寄存器	0x0000_0000

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

位	描述	
[31:3]	**Reserved**	保留
[2:0]	**FUNMODE**	**功能模式** 该域是选择USCI控制器的协议。无效或保留的协议将禁用USCI。当两个协议之间切换时，在选择新协议前USCI必须先禁用。用户写000到FUNMODE时，USCI会被复位。 000 = 禁用 USCI，所有协议相关的状态机器被设置为空闲状态。 001 = 选择SPI 协议 010 = 选择 UART 协议 100 = 选择I^2C 协议 **注：**其它值保留

USCI 波特率发生器寄存器(UI2C_BRGEN)

寄存器	偏移	R/W	描述	复位值
UI2C_BRGEN	UI2Cn_BA+0x08	R/W	USCI波特率发生器寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						**CLKDIV**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLKDIV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DSCNT**					**PDSCNT**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TMCNTSRC**	**TMCNTEN**	**SPCLKSEL**		**PTCLKSEL**	**RCLKSEL**

位	描述	
[31:26]	**Reserved**	保留
[25:16]	**CLKDIV**	**时钟分频器** 该域是定义协议时钟频率f_{PROT_CLK}和时钟分频器频率f_{DIV_CLK} (f_{DIV_CLK}= f_{PROT_CLK}/ (CLKDIV+1)) 之间的比率
[15]	**Reserved**	保留
[14:10]	**DSCNT**	**采样计数器的分母** 该域定义采样时钟f_{SAMP_CLK} 的分频比。 分频f_{DS_CNT} = f_{PDS_CNT} / (DSCNT+1). **注：**在UART模式下，DSCNT最大值是0xF，建议设置4以上来保证接收数据的采样时正确的值。
[9:8]	**PDSCNT**	**采样计数器预分频器** 该域定义的是对来自采样时钟f_{SAMP_CLK}的分频比 分频f_{PDS_CNT} = f_{SAMP_CLK} / (PDSCNT+1).
[7:6]	**Reserved**	保留
[5]	**TMCNTSRC**	**时序测量计数器时钟源选择** 0 = 来自f_{PROT_CLK}. 1 =来自f_{DIV_CLK}.
[4]	**TMCNTEN**	**时序测量计数器使能位** 该位使能10位时序测量计数器 0 =禁用时序测量计数器 1 =使能时序测量计数器
[3:2]	**SPCLKSEL**	**采样时钟源选择** 该域用于对协议采样时钟(f_{SAMP_CLK}) 的时钟源选择 00 = f_{SAMP_CLK} = f_{DIV_CLK}.

		01 = f_{SAMP_CLK} = f_{PROT_CLK}. 10 = f_{SAMP_CLK} = f_{SCLK}. 11 = f_{SAMP_CLK} = f_{REF_CLK}
[1]	**PTCLKSEL**	**协议时钟源选择** 该位用于选择协议时钟源(f_{PROT_CLK}) 0 = 参考时钟f_{REF_CLK}. 1 = f_{REF_CLK2} (频率是f_{REF_CLK}一半)
[0]	**RCLKSEL**	**参考时钟源选择** 该位是选择参考时钟源(f_{REF_CLK}) 0 = 外设设备时钟f_{PCLK}. 1 = 保留

USCI线控制寄存器(UI2C_LINECTL)

寄存器	偏移	R/W	描述	复位值
UI2C_LINECTL	UI2Cn_BA+0x2C	R/W	USCI线控制寄存器	0x0000_0000

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

位	描述	
[31:12]	**Reserved**	保留
[11:8]	**DWIDTH**	**传输的字长** 该域定义发送和接收的数据字长。在数据缓冲中数据总是右对齐。USCI支持4到16位的字长 0x0:在数据字中包含16位，位于[15:0]这些位中 0x1: 保留 0x2: 保留 0x3: 保留 0x4:在数据字中包含4位，位于[3:0]这些位中 0x5:在数据字中包含5位，位于[4:0]这些位中 ... 0xF:在数据字中包含15位，位于[14:0]这些位中
[7:1]	**Reserved**	保留
[0]	**LSB**	**LSB优先传输选择** 0 =MSB，哪一位先发送/接收，取决于DWIDTH的设置 1 =LSB，数据缓冲中的位0被先发送/接收

USCI发送数据寄存器(UI2C_TXDAT)

寄存器	偏移	R/W	描述	复位值
UI2C_TXDAT	UI2Cn_BA+0x30	W	USCI发送数据寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**TXDAT**	发送数据 软件可以写16位发送数据到该域用来发送

USCI接收数据寄存器(UI2C_RXDAT)

寄存器	偏移	R/W	描述	复位值
UI2C_RXDAT	UI2Cn_BA+0x34	R	USCI接收数据寄存器	0x0000_0000

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

位	描述	
[31:16]	**Reserved**	保留
[15:0]	**RXDAT**	**接收的数据** 该域值监控存储在接收数据缓冲的接收到的数据 **注:**在I^2C协议下，RXDAT[12:8]表示不同的传输条件，该条件定义在I^2C中。

USCI 设备地址寄存器(UI2C_DEVADDR)

寄存器	偏移	R/W	描述	复位值
UI2C_DEVADDR0	UI2Cn_BA+0x44	R/W	USCI设备地址寄存器0	0x0000_0000
UI2C_DEVADDR1	UI2Cn_BA+0x48	R/W	USCI设备地址寄存器1	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留
[9:0]	**DEVADDR**	**设备地址** 在I^2C协议下，该位域包含可编程的从机地址。如果第一个接收到地址字节是1111 0AAX$_B$，AA位会与DEVADDR[9:8]比较检查地址地址匹配，X是读/写位。第二个地址字节会和DEVADDR[7:0]比较。 **注 1:**当 I^2C工作在7位地址模式时， DEVADDR [9:7]必须设置为3'b000 **注 2:**当软件设置10'h000时，地址不能被使用

USCI 设备地址掩码寄存器(UI2C_ADDRMSK) –仅针对I²C

寄存器	偏移	R/W	描述	复位值
UI2C_ADDRMSK0	UI2Cn_BA+0x4C	R/W	USCI设备地址掩码寄存器0	0x0000_0000
UI2C_ADDRMSK1	UI2Cn_BA+0x50	R/W	USCI设备地址掩码寄存器1	0x0000_0000

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

位	描述	
[31:10]	**Reserved**	保留
[9:0]	**ADDRMSK**	**USCI设备地址掩码** 0 = 掩码禁用 (接收的相关地址位必须完全与地址寄存器一致) 1 = 使能掩码 (不用关心接收相关地址位) 2组地址掩码寄存器使得USCI支持多地址识别。当在地址掩码寄存器中的位被置1，意味着不用关心接收的相关地址位。如果设置为0，意味着接收的相关地址位必须和地址寄存器一致。

USCI 唤醒控制寄存器(UI2C_WKCTL)

寄存器	偏移	R/W	描述	复位值
UI2C_WKCTL	UI2Cn_BA+0x54	R/W	USCI唤醒控制寄存器	0x0000_0000

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

位	描述	
[31:2]	**Reserved**	保留
[1]	**WKADDREN**	**地址匹配唤醒使能位** 0 = 芯片根据接收‘START’ 信号唤醒 1 = 芯片根据地址匹配唤醒
[0]	**WKEN**	**唤醒使能位** 0 =禁用唤醒功能 1 =使能唤醒功能

USCI 唤醒状态寄存器(UI2C_WKSTS)

寄存器	偏移	R/W	描述	复位值
UI2C_WKSTS	UI2Cn_BA+0x58	R/W	USCI唤醒状态寄存器	0x0000_0000

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

位	描述	
[31:1]	**Reserved**	保留
[0]	**WKF**	**唤醒标志** 当芯片被从掉电模式唤醒，该位置1，软件写1清除该位。

USCI协议控制寄存器– I²C (UI2C_PROTCTL)

寄存器	偏移	R/W	描述	复位值
UI2C_PROTCTL	UI2Cn_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	**Reserved**					**TOCNT**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**MONEN**	**SCLOUTEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**PTRG**	**ADDR10EN**	**STA**	**STO**	**AA**	**GCFUNC**

位	描述	
[31]	**PROTEN**	**I²C 协议使能位** 0 =禁用 I²C 协议 1 = 使能I²C 协议
[30:26]	**Reserved**	保留
[25:16]	**TOCNT**	**超时时钟周期** 当每个中断被清除的时候，该位域表示需要多少时钟周期（时钟源由TMCNTSRC (UI2C _BRGEN [5])选择）。当TOCNT大于0时超时使能。 **注：**在I²C模式下，TMCNTSRC (UI2C _BRGEN [5]) 必须设置为0
[15:10]	**Reserved**	保留
[9]	**MONEN**	**监视器模式使能位** 该位使能监视器模式。在监视器模式下，SDA会以高阻态输入。这样就可以阻止I²C模块输出任何数据（包括ACK）到I²C数据总线上。 0 =禁用监视器模式 1 =使能监视器模式 **注：**取决于SCLOUTEN位的状态，SCL输出被强制为高，阻止模块控制I²C时钟线
[8]	**SCLOUTEN**	**SCL输出使能位** 该位使能监视器SCL被拉低。监视器会拉低SCL直到有时间响应I²C中断 0 =由于开漏机制，SCL输出强制为高 1 = I²C模块像正常操作扮演从机角色，I²C保持时钟线为低直到有时间去清除I²C中断。
[7:6]	**Reserved**	保留

[5]	**PTRG**	**I²C 协议触发 (只写)** 当新状态出现在UI2C _PROTSTS寄存器中时，如果相关中断使能位被设置，将会产生中断请求。在相关中断位被置1之后必须软件写1到该位，然后继续执行I²C协议功能直到STOP启用或是PROTIEN被禁用。 0 = I²C拉伸禁用，并且I²C协议功能继续执行 1 = I²C拉伸启用
[4]	**ADDR10EN**	**位地址功能使能位** 0 =禁用10位地址匹配功能 1 =使能10位地址匹配功能
[3]	**STA**	**I²C 起始控制位** 设置 STA 为 1，进入主机模式，如果总线处于空闲状态，I²C 硬件会送出 START或重复 START 条件到总线
[2]	**STO**	**I²C 停止控制位** 在主机模式，设置 STO来传送一个 STOP 条件到总线，然后 I²C 硬件将会检查总线状况，如果检测到一个 STOP 状况，这个标志会被硬件自动清除。在从机模式下，当总线错误(UI2C _PROTSTS.ERRIF = 1)的时候，设置STO复位I²C硬件来进入“无地址”从机模式
[1]	**AA**	**应答控制位** 当 AA=1 先于地址或数据接收，在SCL线上的应答时钟脉冲期间将返回一个应答(SDA上为低电平)，有两种情况：1.) 从机正在应答主机发送的地址。2.) 接收设备正在应答发送设备发送的数据。当 AA = 0 先于地址或数据接收，则在SCL线上的应答时钟脉冲期间将返回一个非应答（SDA上为高电平）。
[0]	**GCFUNC**	**广播呼叫功能** 0 = 广播呼叫功能禁用 1 = 广播呼叫功能使能

USCI协议中断使能寄存器– I²C (UI2C_PROTIEN)

寄存器	偏移	R/W	描述	复位值
UI2C_PROTIEN	UI2Cn_BA+0x60	R/W	USCI协议中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**ACKIEN**	**ERRIEN**	**ARBLOIEN**	**NACKIEN**	**STORIEN**	**STARIEN**	**TOIEN**

位	描述	
[31:7]	**Reserved**	保留
[6]	**ACKIEN**	**应答中断使能控制** 如果应答被主机检测到，该位是使能产生应答中断 0 =禁用应答中断 1 =使能应答中断
[5]	**ERRIEN**	**错误中断使能控制** 如果一个I²C错误条件被检测到(由 ERR (UI2C _PROTSTS [16])指示)，该位使能产生错误中断 0 =禁用错误中断 1 =使能错误中断
[4]	**ARBLOIEN**	**仲裁丢失中断使能控制** 如果仲裁丢失事件被检测到，该位使能产生仲裁丢失中断 0 =禁用仲裁丢失中断 1 =使能仲裁丢失中断
[3]	**NACKIEN**	**非应答中断使能控制** 如果非应答被主机检测到，该位使能产生非应答中断 0 =禁用非应答中断 1 =使能非应答中断
[2]	**STORIEN**	**停止条件接收中断使能控制** 如果停止条件被检测到，该位使能产生停止条件中断 0 =禁用停止条件中断 1 =使能停止条件中断
[1]	**STARIEN**	**起始条件接收中断使能控制** 如果起始条件被检测到，该位使能产生起始条件中断

		0 =禁用起始条件中断 1 =使能起始条件中断
[0]	**TOIEN**	**超时中断使能控制** 在I^2C协议下，该位使能产生超时事件中断 0 =禁用超时中断 1 =使能超时中断

USCI 协议状态寄存器– I²C (UI2C_PROTSTS)

寄存器	偏移	R/W	描述	复位值
UI2C_PROTSTS	UI2Cn_BA+0x64	R/W	USCI协议状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**ERRARBLO**	**BUSHANG**	**WRSTSWK**	**WKAKDONE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SLAREAD	**SLASEL**	**ACKIF**	**ERRIF**	**ARBLOIF**	**NACKIF**	**STORIF**	**STARIF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**ONBUSY**	**TOIF**	**Reserved**				

位	描述	
[31:20]	**Reserved**	保留
[19]	**ERRARBLO**	**仲裁丢失错误** 该位表示由于不能被输入处理器过滤的噪声引起总线仲裁丢失。当ERRARBLO被置位时I²C会发送起始条件。因此从机模式下无需关心。 0 =总线传输状态正常 1 =总线传输仲裁丢失错误 **注：**该位没有中断信号，并且当START条件出现时会被硬件自动清除
[18]	**BUSHANG**	**总线挂起** 该位指示总线挂起状态，当SCL保持高时且参考源是f_{SAMP_CLK}，4位计数器会计数。挂起计数器会计数到溢出并且在SDA位低时该位被置位。计数器由SCL下降沿信号复位。 0 =总线传输正常 1 =总线传输挂起 **注：**该位没有中断信号，并且当START条件出现时会被硬件自动清除
[17]	**WRSTSWK**	**地址帧中的读/写状态位** 0 =写命令被记录到地址匹配唤醒帧中 1 =读命令被记录到地址匹配唤醒帧中
[16]	**WKAKDONE**	**唤醒地址帧应答位完成** 0 =地址匹配帧ACK位阶段没有完成 1 =掉电状况下地址匹配帧ACK位阶段已完成 **注：**当WKUPIF被置位时该位不会释放
[15]	**SLAREAD**	**从机读请求状态** 该位表示从机读请求被检测到 0 =没有检测到从机R/W位是1 1 =检测到从机R/W位是1

		注：该位没有中断信号，它会被硬件自动清除
[14]	**SLASEL**	**从机选择状态** 该位指示设备被选作从机 0 =设备没有被选作从机 1 =设备被选作从机 **注：**该位没有中断信号，它会被硬件自动清除
[13]	**ACKIF**	**应答接收中断标志** 主机模式下该位表示应答位已被接收。如果UI2C _PROTCTL.ACKIEN = 1会产生中断 0 =应答没有被接收 1 =应答被接收 软件写1清除
[12]	**ERRIF**	**错误中断标志** 当START或STOP条件出现在信息帧的不合理位置时该位用于指示发生总线错误。不合理位置比如在地址字节、数据字节或应答位这些串行传输期间。如果UI2C _PROTCTL.ERRIEN = 1会产生中断 0 = I^2C错误没有被检测到 1 = I^2C 错误被检测到 软件写1清除 **注：**从机模式该位被置位时，用户必须写1到STO寄存器是设备进入“无地址”从机模式
[11]	**ARBLOIF**	**仲裁丢失中断标志** 该位表示发送仲裁丢失。如果UI2C _PROTCTL.ARBLOIEN = 1会产生中断 0 =没有发生仲裁丢失 1 =仲裁丢失 软件写1清除
[10]	**NACKIF**	**非应答接收中断标志** 主机模式下该位表示非应答位已被接收。如果UI2C _PROTCTL.NACKIEN = 1会产生中断 0 =非应答没有被接收 1 =非应答被接收 软件写1清除
[9]	**STORIF**	**停止条件接收中断标志** 该位表示停止条件在I^2C总线上已被检测到。如果UI2C _PROTCTL.STORIEN = 1会产生中断 0 =停止条件没有被检测到 1 =停止条件被检测到 软件写1清除 **注：**当显示从机RX模式下被置位
[8]	**STARIF**	**起始条件接收中断标志** 该位表示主机模式下起始条件或是重复起始条件在I^2C总线上已被检测到。然而在从机模式下表示重复起始条件被检测到。 如果UI2C _PROTCTL.STARIEN = 1会产生中断 0 =起始条件还没有被检测到 1 =停止条件已被检测到 软件写1清除

[7]	**Reserved**	保留
[6]	**ONBUSY**	**总线忙** 表明总线上正在进行数据传输。当START条件被检测到时被硬件置位。当STOP被检测到时被硬件清除 0 = 总线空闲 (SCLK 和 SDA 都为高). 1 = 总线忙
[5]	**TOIF**	**超时中断标志** 0 =没有发生超时中断 1 =没有发生超时中断 **注：**软件写1清除
[4:0]	**Reserved**	保留

USCI 从机匹配地址寄存器(UI2C_ADMAT)

寄存器	偏移	R/W	描述	复位值
UI2C_ADMAT	UI2Cn_BA+0x88	R/W	I^2C从机匹配地址寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**ADMAT1**	**ADMAT0**

位	描述	
[31:2]	**Reserved**	保留
[1]	**ADMAT1**	**USCI地址1匹配状态寄存器** 当地址1匹配时，该位将被设置为1，软件可以写入1来清除该位
[0]	**ADMAT0**	**USCI地址0匹配状态寄存器** 当地址0匹配时，该位将被设置为1，软件可以写入1来清除该位。

USCI 时序配置控制寄存器(UI2C_TMCTL)

寄存器	偏移	R/W	描述	复位值
UI2C_TMCTL	UI2Cn_BA+0x8C	R/W	I^2C 时序配置控制寄存器	0x0000_0000

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

位	描述	
[31:25]	**Reserved**	保留
[24:16]	**HTCTL**	**保持时间配置控制寄存器** 该域是发送模式下用来在SCL下降沿和SDA边沿之间产生延时时序。 延时保持时间外设时钟数= HTCTL x f_{PCLK}。 **注：**保持时间功能仅仅在主机模式下可配置，在从机模式下，该位应设置为0。
[15:9]	**Reserved**	保留
[8:0]	**STCTL**	**建立时间配置控制寄存器** 该域是发送模式下用来在SDA边沿和SCL上升沿之间产生延时时序。 延时建立时间外设时钟数= STCTL x f_{PCLK}。

6.21 外部总线接口(EBI)

6.21.1 概述

该芯片配备了一个外部总线接口(EBI)，以供外部设备使用。为节省外部设备与芯片的连接线，EBI支持地址与数据总线多路复用。EBI支持两组片选，可连接两个各自有不同时序设定的外部设备。

6.21.2 特性

- 支持两路存储bank
- 为每个bank 提供了有极性控制的片选引脚
- 每个bank访问空间达1MB,实际外部可寻址空间取决于芯片封装引脚
- 支持 8位/16位 数据宽度
- 16位数据宽度下支持字节写
- 支持地址/数据复用模式
- 每个存储bank支持时序参数独立调整
- 支持LCD i80接口模式
- 支持PDMA模式
- 支持基于HCLK所产生的不同频率的总线基本时钟
- 支持可配置空闲周期以用于不同的访问情况：写指令完成的空闲时间（W2X）和连读的空闲时间（R2R）
- 支持地址总线和数据总线分开模式

		M031xB/C/D/E	M031xG/I
	M032xC/D	M032xE	M032xG/I
6.21.5.3 EBI数据宽度连接-地址总线和数据总线分开模式			•
6.21.5.4 EBI 操作控制--连续数据访问模式			•

表 6.21-1 EBI 特性比较表

6.21.3 框图

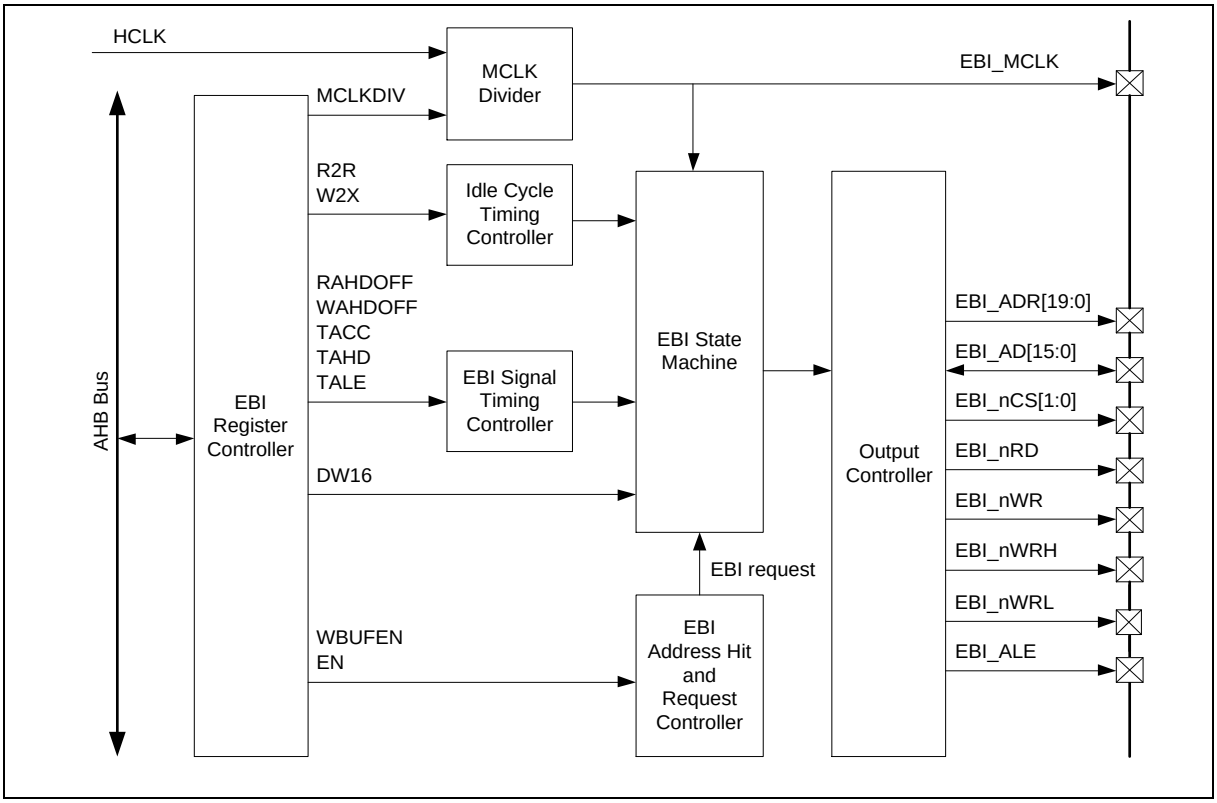


图 6.21-1 EBI 框图

6.21.4 基本配置

- 时钟源配置
  - 通过设置EBICKEN(CLK_AHBCLK[3])使能EBI 控制器时钟
- 复位配置
  - 通过设置EBIRST(SYS_IPRST0[3]) 复位 EBI控制器
- 管脚配置

组	管脚名称	GPIO	MFP
EBI	EBI_AD0	PC.0, PG.9	MFP2
	EBI_AD1	PC.1, PG.10	MFP2
	EBI_AD2	PC.2, PG.11	MFP2
	EBI_AD3	PC.3, PG.12	MFP2
	EBI_AD4	PC.4, PG.13	MFP2
	EBI_AD5	PC.5, PG.14	MFP2
	EBI_AD6	PA.6, PD.8	MFP2
	EBI_AD7	PA.7, PD.9	MFP2

	EBI_AD8	PC.6, PE.14	MFP2
	EBI_AD9	PC.7, PE.15	MFP2
	EBI_AD10	PD.3, PD.13, PE.1	MFP2
	EBI_AD11	PC.14, PD.2, PE.0	MFP2
	EBI_AD12	PB.15, PD.1, PH.8	MFP2
	EBI_AD13	PB.14, PD.0, PH.9	MFP2
	EBI_AD14	PB.13, PH.10	MFP2
	EBI_AD15	PB.12, PH.11	MFP2
	EBI_ADR0	PB.5, PH.7	MFP2
	EBI_ADR1	PB.4, PH.6	MFP2
	EBI_ADR2	PB.3, PH.5	MFP2
	EBI_ADR3	PB.2, PH.4	MFP2
	EBI_ADR4	PC.12	MFP2
	EBI_ADR5	PC.11	MFP2
	EBI_ADR6	PC.10	MFP2
	EBI_ADR7	PC.9	MFP2
	EBI_ADR8	PB.1	MFP2
	EBI_ADR9	PB.0	MFP2
	EBI_ADR10	PC.13, PE.8	MFP2
	EBI_ADR11	PE.9, PG.2	MFP2
	EBI_ADR12	PE.10, PG.3	MFP2
	EBI_ADR13	PE.11, PG.4	MFP2
	EBI_ADR14	PE.12, PF.11	MFP2
	EBI_ADR15	PE.13, PF.10	MFP2
	EBI_ADR16	PB.11, PC.8, PF.9	MFP2
	EBI_ADR17	PB.10, PF.8	MFP2
	EBI_ADR18	PB.9, PF.7	MFP2
	EBI_ADR19	PB.8, PF.6	MFP2
	EBI_ALE	PA.8, PE.2	MFP2
	EBI_MCLK	PA.9, PE.3	MFP2
	EBI_nCS0	PD.12, PD.14, PF.3	MFP2
		PF.6	MFP7
		PB.7	MFP8
	EBI_nCS1	PD.11, PF.2	MFP2
		PB.6	MFP8

	EBI_nRD	PA.11, PE.5	MFP2
	EBI_nWR	PA.10, PE.4	MFP2
	EBI_nWRH	PB.6	MFP2
	EBI_nWRL	PB.7	MFP2

6.21.5 功能描述

6.21.5.1 EBI地址匹配区域

EBI 映射地址分布在0x6000_0000 ~ 0x601F_FFFF，内存空间为2M字节。系统请求在EBI 的内存空间内时，相应的EBI芯片选择信号有效，EBI状态机开始运行

Chip Select	Address Mapping
EBI_nCS0	0x6000_0000 ~ 0x600F_FFFF
EBI_nCS1	0x6010_0000 ~ 0x601F_FFFF

表 6.21-2 EBI 地址映射

为了映射整个EBI 内存空间， 8位数据宽度设备需20位地址， 16位数据宽度设备需19位地址。

小于20位地址的设备， EBI 将映射设备到镜像空间。例如，一个18位EBI地址设备，EBI (Bank0/EBI_nCS0) 将外设同时映射到0x6000_0000 ~0x6003_FFFF, 0x6004_0000 ~ 0x6007_FFFF, 0x6008_0000 ~ 0x600B_FFFF 和 0x600C_0000 ~0x600F_FFFF.

6.21.5.2 EBI 数据宽度连接-地址与数据总线复用模式

EBI 支持地址总线和数据总线复用。对于地址总线和数据总线分开的外部设备，与设备的连接需要额外的逻辑单元锁存地址。这样，管脚EBI_ALE需要连接到锁存器上锁存地址值。管脚EBI_AD为锁存器的输入，锁存器的输出连接到外部设备的地址总线上。

对于16位设备， EBI_AD[15:0]是地址与数据复用线， EBI_ADR [18:16]仅为地址，直接与16位设备连接。当EBI 数据宽度设置为16位， EBI_ADR[19]无作用。对于8位设备，仅EBI_AD [7:0]地址与数据复用，EBI_AD[15:8] 和 EBI_ADR[19:16]为地址，直接与8位设备连接图 6.21-2 16位 EBI 数据宽度与16位设备的连接，图 6.21-3描述8位 EBI 数据宽度与8位设备的连接

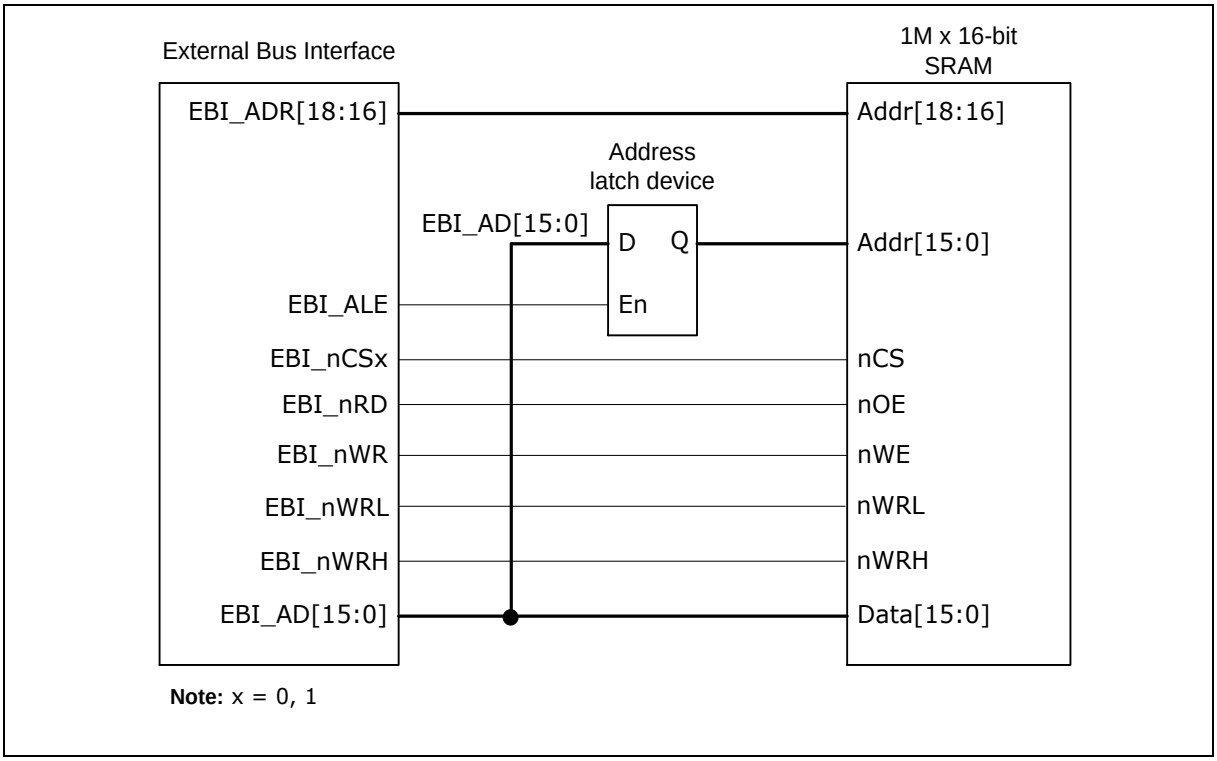


图 6.21-2 16 位 EBI 数据宽度与 16 位设备的连接

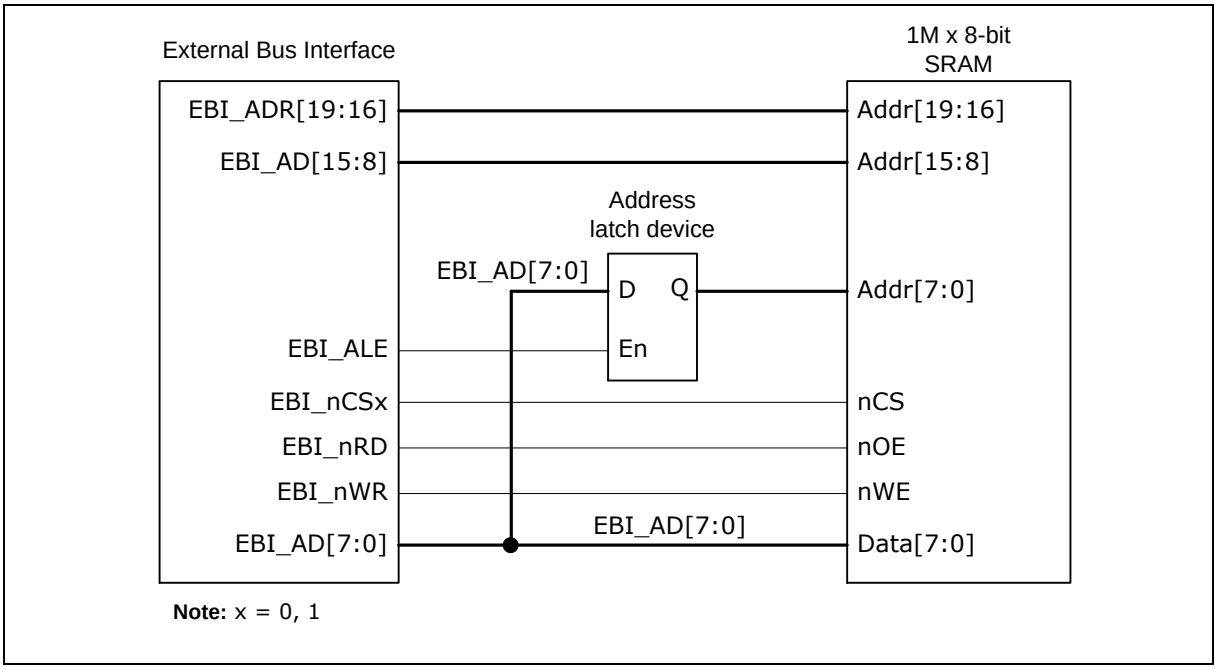


图 6.21-3 8 位 EBI 数据宽度与 8 位设备的连接

当系统访问数据宽度大于EBI 的数据宽度时， EBI 将访问一次以上。例如，如果系统通过EBI设备请求32位数据，当EBI为8位数据宽度时， EBI将4次访问总线

6.21.5.3 EBI数据宽度连接-地址总线和数据总线分开模式

EBI支持地址和数据总线分开模式，用户可以通过设置ADSEPEN (EBI_CTLx[3])使能此模式。当分开模式使能，EBI_AD专门用做数据总线直接接到设备的数据总线，EBI_ADR是专用的地址总线

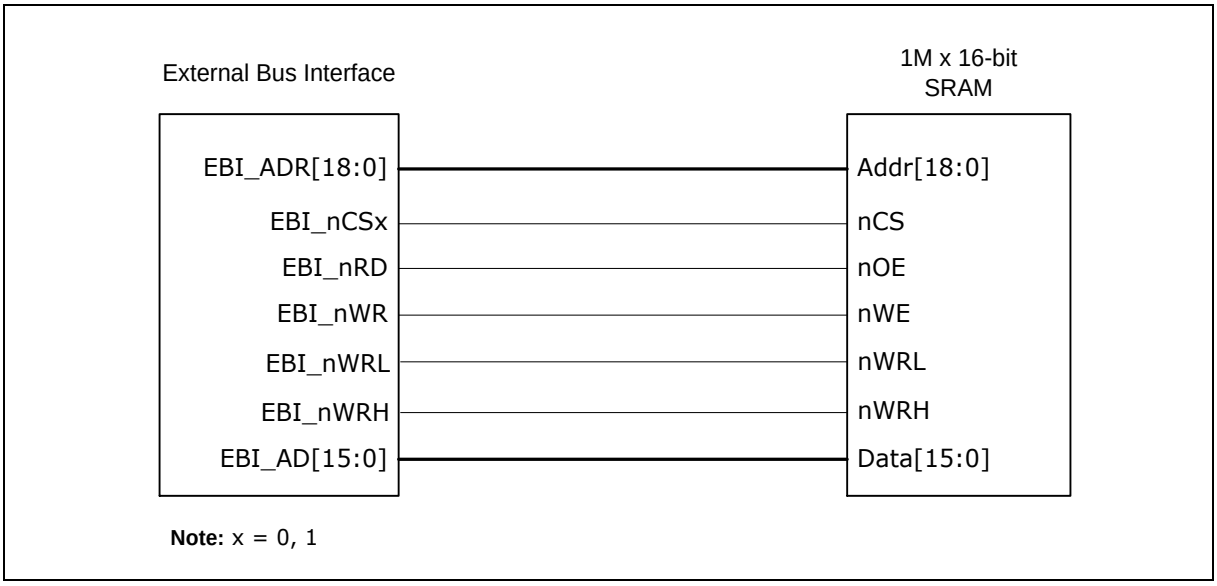


图 6.21-4 16 位 EBI 数据宽度与 16 位设备的分开模式连接

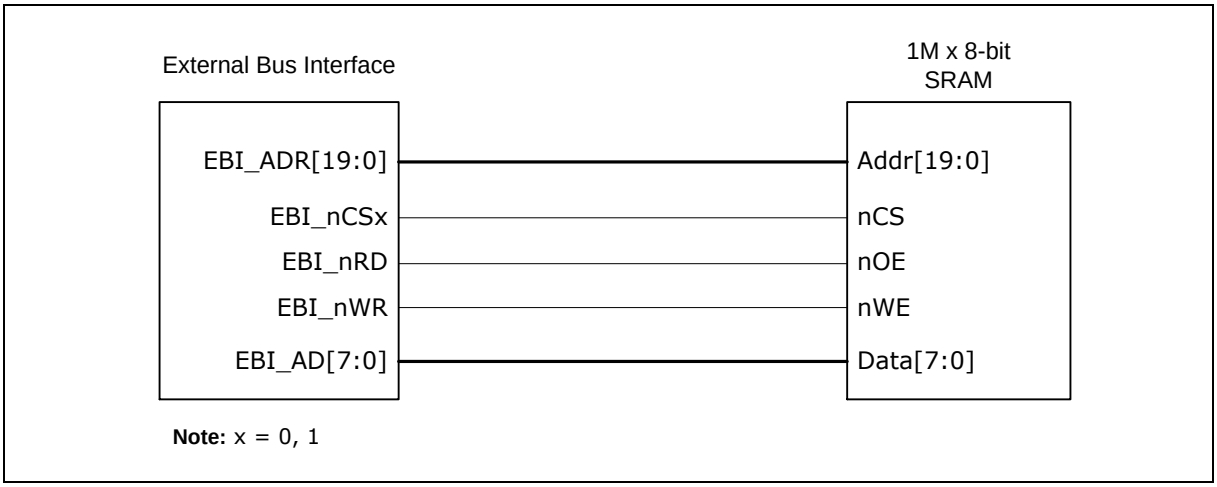


图 6.21-5 8 位 EBI 数据宽度与 8 位设备的分开模式连接

6.21.5.4 EBI操作控制

MCLK 控制

EBI工作时，所有EBI信号与EBI_MCLK同步。以低工作频率连接到外部设备时， EBI_MCLK可以通过寄存器MCLKDIV (EBI_CTLx[10:8])最多分频到HCLK/128，因此，该芯片能适配宽范围频率的EBI设备。如果EBI_MCLK设置为HCLK/1，则EBI信号与EBI_MCLK的正边沿同步，否则与EBI_MCLK的负边沿同步

操作与访问时序控制

开始访问时，片选(EBI_nCS0 、EBI_nCS1和EBI_nCS2)置低并等待一个EBI_MCLK时间 (tASU)以使地址稳定. 然后 EBI_ALE置高并保持一段时间(tALE)用于地址锁存. 地址锁存后，EBI_ALE置低并等待一个EBI_MCLK时间(tLHD)，再插入一个EBI_MCLK时间(tA2D)用于总线转换(地址到数据)。读时EBI_nRD置低，写时EBI_nWR置低。在保持访问时间(tACC)后EBI_nRD或EBI_nWR置高完成读或写。之后， EBI 信号在数据保持时间(tAHD)之后片选置高，一个总线周期结束.

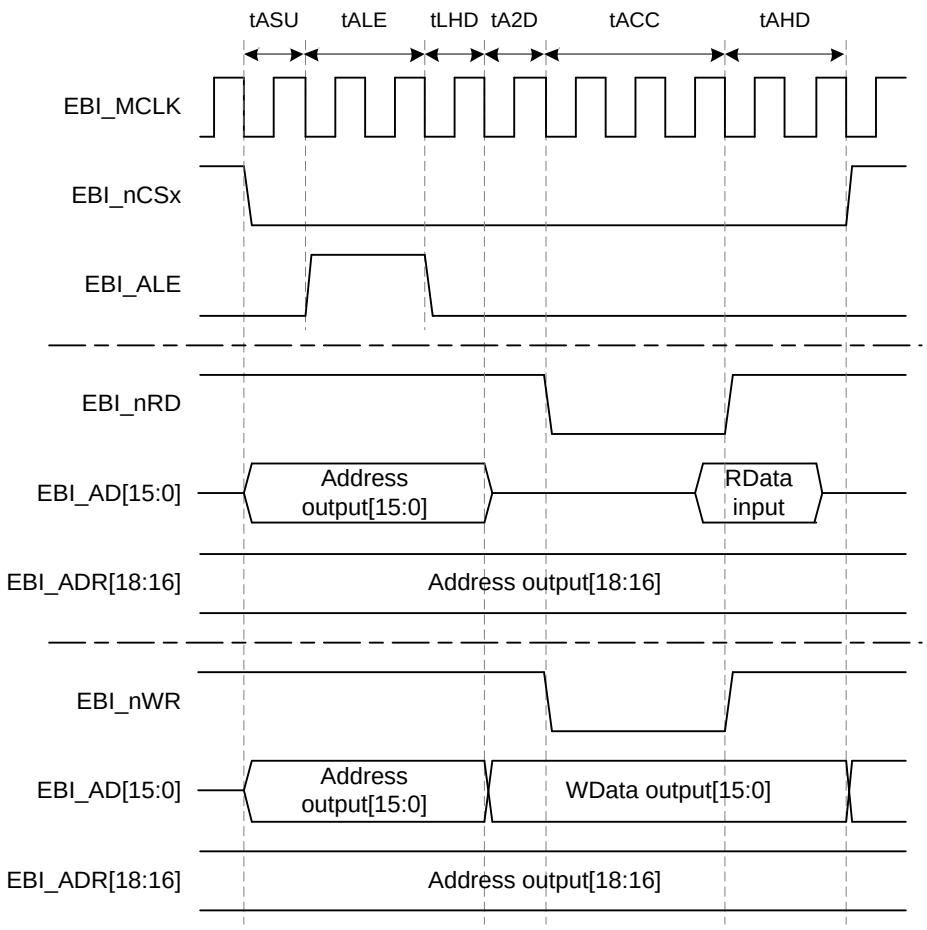
EBI 控制器提供灵活的时序控制. tASU, tLHD 和tA2D固定为1个EBI_MCLK时间。 tAHD通过寄存器设定TAHD (EBI_TCTLx[10:8]) 可以在 1~8 个 EBI_MCLK 周期内调节， tACC 通过寄存器TACC(EBI_TCTLx[7:3]) 可以在 1~32 个 EBI_MCLK 周期内调节， tALE 通过寄存器TALE(EBI_CTL0[18:16] 可以在1~8个EBI_MCLK周期内调节。一些外设可以支持数据访问零保持时间， EBI 控制器可以略过tAHD, 通过寄存器设定WAHDOFF (EBI_TCTLx[23]) 和 RAHDOFF (EBI_TCTLx[22]) 提高访问速度

对每个芯片的片选，除了tALE 只能由EBI_CTL0控制，其他的寄存器设定， EBI会提供单独的寄存器用于时序控制

参数	值	单位	描述
tASU	1	MCLK	地址锁存建立时间 .
tALE	1 ~ 8	MCLK	ALE 高周期. 通过寄存器TALE (EBI_CTL0[18:16])控制.
tLHD	1	MCLK	地址锁存保持时间.
tA2D	1	MCLK	地址到数据的延时 (总线转换时间).
tACC	1 ~ 32	MCLK	数据访问时间. 通过寄存器 TACC (EBI_TCTLx[7:3])控制.
tAHD	1 ~ 8	MCLK	数据访问保持时间. 通过寄存器 TAHD (EBI_TCTLx[10:8])控制.
IDLE	0 ~ 15	MCLK	空闲周期. 通过寄存器 R2R (EBI_TCTLx[27:24]) 和 W2X (EBI_TCTLx[15:12])控制.

表 6.21-3 时序控制参数

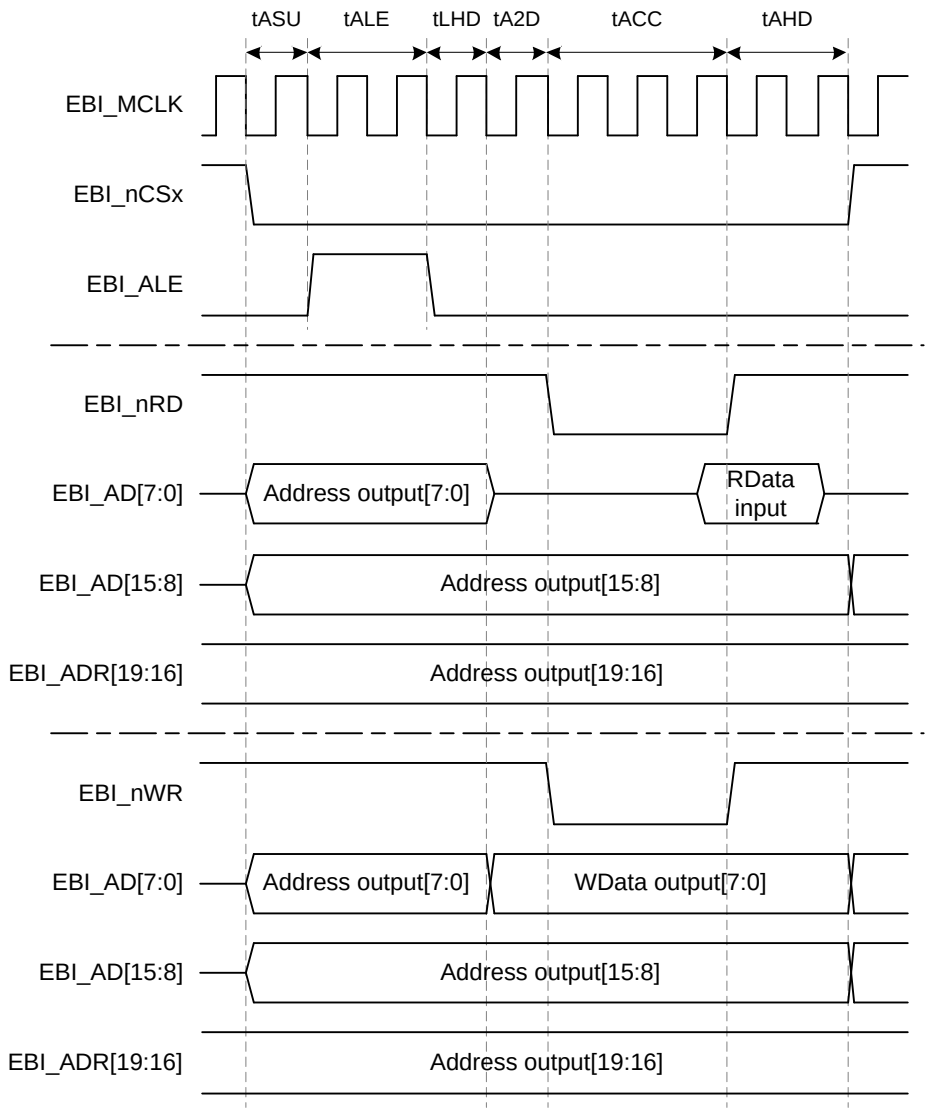
图 6.21-6是一个设置16位数据宽度的例子。在该例中，EBI_AD 总线用作地址[15:0]和数据[15:0]。当EBI_ALE置高，EBI_AD为地址输出。在地址锁存后，EBI_ALE置低并且在读取访问操作时，EBI_AD总线转换成高阻以等待设备输出数据，或用于写数据输出



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1

图 6.21-6 16 位数据宽度的时序控制波形

图 6.21-7是一个设置8位数据宽度的例子。8位和16位数据宽度的不同之处在于EBI_AD[15:8].在8位数据宽度的设置中，EBI_AD[15:8]总为地址[15:8]输出，因此外部锁存只需8位宽度。

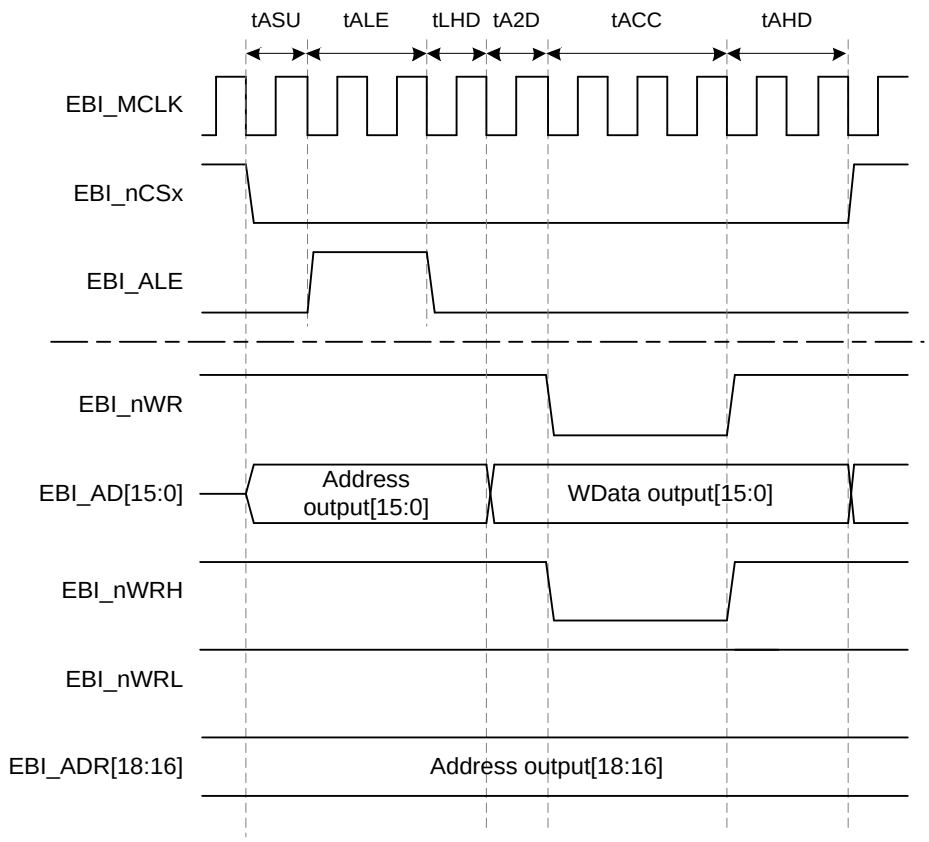


Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1

图 6.21-7 8 位数据宽度的时序控制波形

字节访问

当连接16位设备时EBI支持字节访问。16位数据总线上用引脚EBI_nWRH 和 EBI_nWRL指示高字节使能和低字节使能。图 6.21-8 是展示用EBI_nWRH指示在EBI_AD[15:8]上8位数据宽度的写操作



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1

图 6.21-8 16 位数据模式字节写的时序控制波形

插入空闲周期

当EBI连续访问时，如果器件访问时间较慢，可能会有总线冲突。EBI控制器支持额外空闲周期以解决该问题。在空闲周期，所有EBI的控制信号无效. 图 6.21-9 表示 空闲周期.

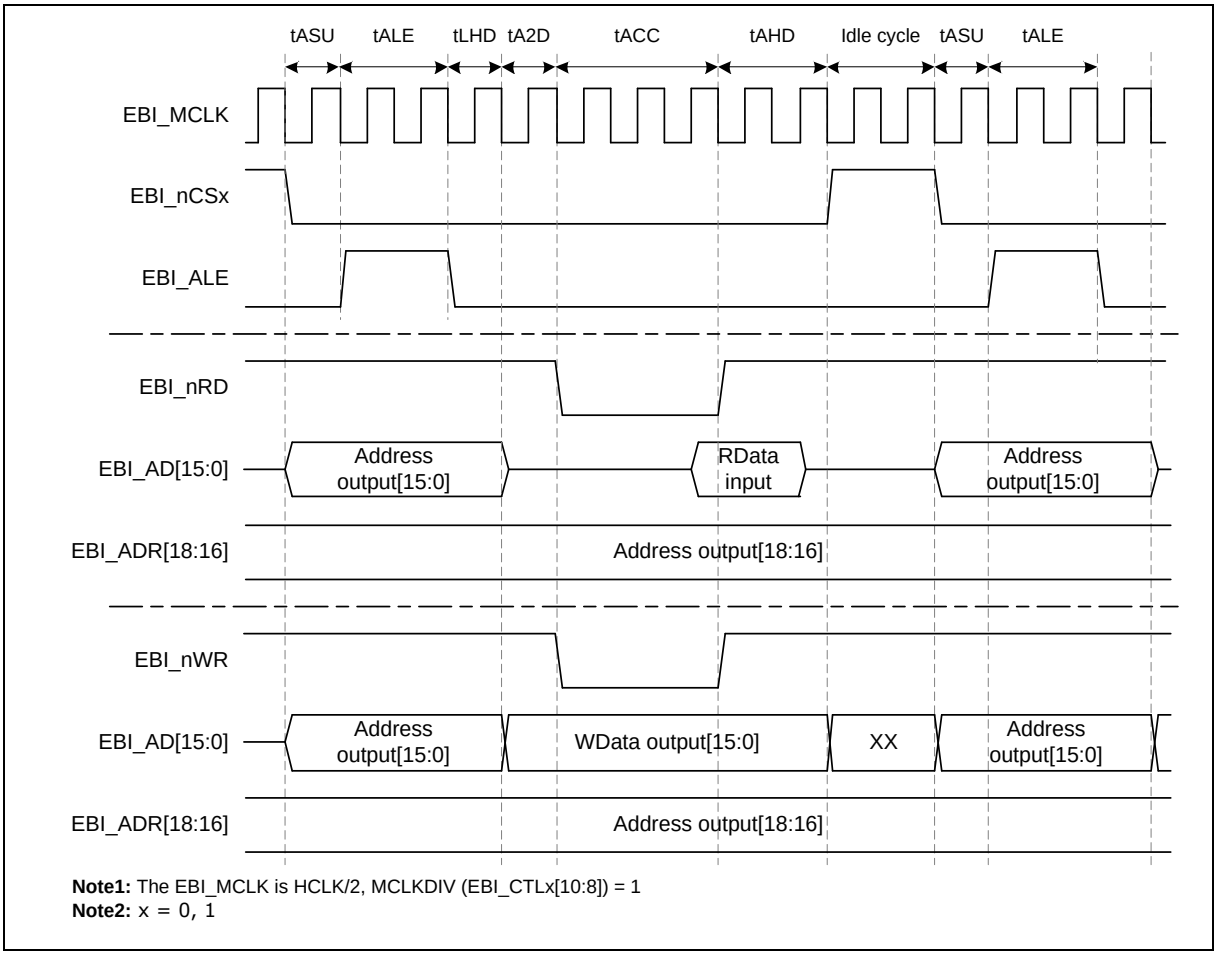


图 6.21-9 插入空闲周期的时序控制波形

以下两个条件，EBI可插入空闲周期:

1. 写访问之后
2. 读访问完之后与下一个读访问之前（R2R 空闲周期）

通过设定寄存器W2X (EBI_TCTLx[15:12]) 和R2R (EBI_TCTLx[27:24]), 空闲周期可设定在0~15 EBI_MCLK.

片选极性控制

EBI 支持片选极性控制以便连接不同的外部设备。CSPOLINV (EBI_CTLx[2]) 置0 ，片选(EBI_nCSx)低电平有效。这意味着在EBI_nCSx低电平时访问外部设备。CSPOLINV (EBI_CTLx[2]) 置1， 片选(EBI_nCSx)高电平有效。这意味着当在EBI_nCSx高电平时访问外部设备。 .

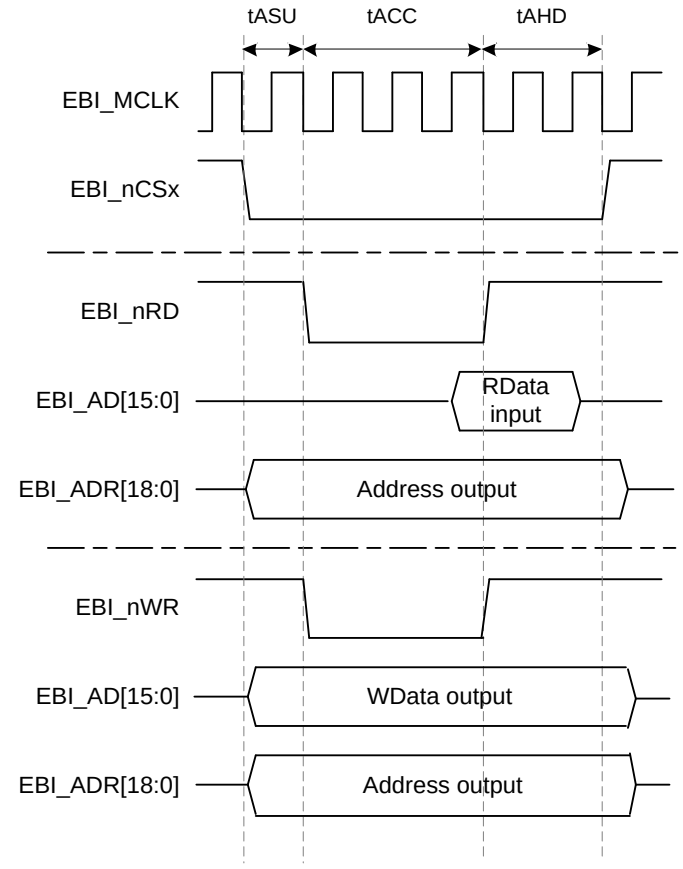
写缓存

当用户通过EBI总线写一个数据到外设时，EBI控制器将立刻开始写动作，CPU是保持状态，直到EBI写动作结束。用户可以使能写缓存功能来提高CPU和EBI的访问效率。当EBI写缓存功能使能后，在EBI向外设写数据期间，CPU还可以持续执行其他指令。例外的情况，当EBI执行写动作时，如果CPU通过EBI执行另一个数据访问，CPU会是保持状态

用户可以通过设置 WBUFEN (EBI_CTL0[24])使能写缓存

地址数据分开模式

当EBI设置为分开模式，tALE, tLHD, tA2D周期失效，EBI_AD和EBI_ADR被分别用作专门的数据和地址总线



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1

图 6.21-10 16 位数据宽度分开模式下的时序控制波形

连续数据访问模式

对于需要更快的数据访问且不需要地址控制的接口，EBI支持连续数据访问。每个bank，用户可以通过设置CACCESS (EBI_CTLX[4])使能该模式。当EBI设置为连续数据访问模式，tASU, tALE, tLHD周期会被忽略，且EBI可以在一个读或写命令内连续地访问数据。在每个访问命令之间会有无效的周期时间，

时序波形如图 6.21-11.

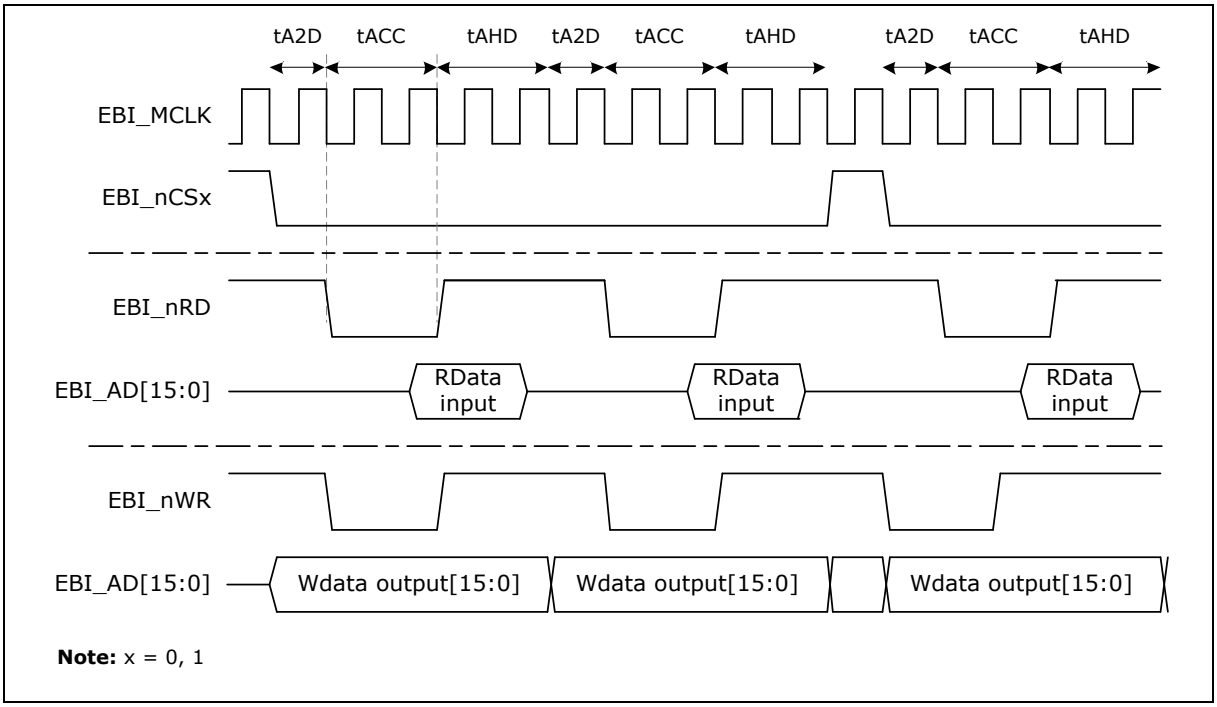


图 6.21-11 连续数据访问模式的时序控制波形

6.21.6 寄存器表

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移	R/W	描述	复位值
EBI基地址: **EBI_BA = 0x4001_0000**				
EBI_CTL0	EBI_BA+0x00	R/W	外部总线接口Bank0控制寄存器	0x0000_0000
EBI_TCTL0	EBI_BA+0x04	R/W	外部总线接口Bank0时序控制寄存器	0x0000_0000
EBI_CTL1	EBI_BA+0x10	R/W	外部总线接口Bank1控制寄存器	0x0000_0000
EBI_TCTL1	EBI_BA+0x14	R/W	外部总线接口Bank1时序控制寄存器	0x0000_0000

6.21.7 寄存器描述

外部总线接口控制寄存器(EBI_CTLx)

寄存器	偏移	R/W	描述	复位值
EBI_CTL0	EBI_BA+0x00	R/W	外部总线接口Bank0控制寄存器	0x0000_0000
EBI_CTL1	EBI_BA+0x10	R/W	外部总线接口Bank1控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reversed							WBUFEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reversed					TALE		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reversed					MCLKDIV		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reversed			CACCESS	ADSEPEN	CSPOLINV	DW16	EN

位	描述	
[31:25]	**Reserved**	保留.
[24]	**WBUFEN**	**EBI 写缓存使能位** 0 = EBI 写缓存禁用. 1 = EBI 写缓存使能. **注:** 该位仅在EBI_CTL0 寄存器中有效.
[23:19]	**保留**	保留.
[18:16]	**TALE**	**ALE扩展时间** 控制EBI_ALE 高电平时间 tALE = (TALE+1)*EBI_MCLK. **注:** 该位仅在EBI_CTL0寄存器中有效.
[15:11]	**保留**	保留.
[10:8]	**MCLKDIV**	**外部输出时钟分频器** MCLK 频率设置如下: 000 = HCLK/1. 001 = HCLK/2. 010 = HCLK/4. 011 = HCLK/8. 100 = HCLK/16. 101 = HCLK/32. 110 = HCLK/64. 111 = HCLK/128.

[7:5]	**保留**	保留.
[4]	**CACCESS**	**连续数据访问模式** 当连续访问模式使能，为了连续传输数据需要，tASU、tALE和tLHD周期会被忽略 0 = 连续数据访问模式禁用. 1 = 连续数据访问模式使能.
[3]	**ADSEPEN**	**EBI地址/数据总线分开模式使能位** 0 =地址/数据总线分开模式禁止. 1 =地址/数据总线分开模式使能
[2]	**CSPOLINV**	**片选管脚极性翻转** 该位定义了片选脚(EBI_nCS)有效电平. 0 = 片选脚 (EBI_nCS) 低有效. 1 = 片选脚 (EBI_nCS) 高有效.
[1]	**DW16**	**EBI 16位数据宽度选择** 该位定义了EBI 数据时8位还是16位 0 = EBI 数据为8位. 1 = EBI 数据为16位.
[0]	**EN**	**EBI 使能位** 该位使能EBI功能. 0 = EBI 功能禁用. 1 = EBI 功能使能.

外部总线接口时序控制寄存器(EBI_TCTLx)

寄存器	偏移	R/W	描述	复位值
EBI_TCTL0	EBI_BA+0x04	R/W	外部总线接口bank0 时序控制寄存器	0x0000_0000
EBI_TCTL1	EBI_BA+0x14	R/W	外部总线接口bank1 时序控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留				R2R			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WAHDOFF	RAHDOFF	保留					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
W2X				Reversed	TAHD		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TACC					保留		

位	描述	
[31:30]	保留	保留.
[27:24]	**R2R**	**读-读之间的空闲状态周期** 该位定义了读-读之间的空闲状态周期数. R2R 空闲状态周期 = (R2R * EBI_MCLK). **注:** 当读动作完成，下一个动作也是读时，插入读-读(R2R)空闲状态周期后，EBI_nCS回空闲状态
[23]	**WAHDOFF**	**EBI 写时，访问保持时间禁用** 0 = 在EBI 写期间，使能数据访问保持时间 1 = 在EBI 写期间，禁用数据访问保持时间
[22]	**RAHDOFF**	**EBI读时，访问保持时间禁用** 0 = 在EBI 读期间，使能数据访问保持时间 1 = 在EBI 读期间，禁用数据访问保持时间
[21:16]	保留	保留.
[15:12]	**W2X**	**写之后的空闲状态周期** W2X 空闲状态周期. W2X 空闲状态周期 = (W2X * EBI_MCLK). **注:**当写动作完成，插入W2X空闲状态周期后，EBI_nCS回空闲状态
[11]	保留	保留.
[10:8]	**TAHD**	**EBI 数据访问保持时间** TAHD 定义数据访问保持时间 (tAHD). tAHD = (TAHD +1) * EBI_MCLK.
[7:3]	**TACC**	**EBI 数据访问时间**

		TACC 定义数据访问时间 (tACC). tACC = (TACC +1) * EBI_MCLK.
[2:0]	保留	保留.

6.22 USB 2.0 全速设备控制器(USBD)

6.22.1 概述

本器件带一组USB 2.0全速设备端收发器。支持控制/批量/中断/同步四种传输类型。

在此设备控制器中，有两个主要接口，APB总线和USB总线。USB总线来自于USB 硬件收发器。CPU可以通过APB总线来设置相应控制寄存器。控制器中有512字节的内部SRAM作为数据缓冲区。对于IN或OUT 传输， CPU通过APB或SIE对SRAM读写。用户要先通过寄存器(USBD_BUFSEGx)对每一个端点在SRAM中设置有效起始地址。

该控制器共有8个端点。每个端点可独立配置成输入或输出端点。所有传输模式包括控制/批量/中断/同步四种模式都通过这一模块传输。端点控制模块也用于管理数据流同步，端点状态，当前起始地址，处理状态和每个端点数据缓冲状态。

控制器中有四个不同的中断事件：唤醒事件，插拔事件，USB事件如:IN ACK，OUT ACK和BUS事件如：挂起，恢复等。以上任何事件都会产生中断，用户只需在中断状态寄存器(USBD_INTSTS)中查找相关事件标志，就可以知道发生了哪种中断，然后查找相关的USB端点状态寄存器USBD_EPSTS0就可知道在这个端点中发生了何种中断事件。

在这个USB控制器中也支持软件断开连接功能。这个功能用于仿真设备从主机断开的过程。如果SE0 位(USBD_SE0)被置位，USB控制器将强迫把USB_D+ 和 USB_D-拉到低电平，从而禁止USB功能。SE0位被清零后，主设备将再次枚举所插入的USB设备。

详细内容请参考通用串行总线规范修订2.0

6.22.2 特性

- 兼容USB2.0 全速规范
- 提供1个中断向量包含5种中断事件（起始帧，唤醒，插拔，USB和总线 ）
- 支持 控制/批量/中断/同步 传输类型
- 支持总线空闲3ms以上总线切换到挂起功能
- 支持8个通讯端点提供控制/批量/中断/同步传输类型，和最大512字节数据缓冲区
- 提供远程唤醒功能

章节	-	M031xB/C/D/E	M031xG/I
	M032xC/D	M032xE	M032xG/I
6.22.7 寄存器描述 USB配置寄存器 (USB_CFGx) DSQSYNC 输出令牌传输	•	-	-

6.22.3 框图

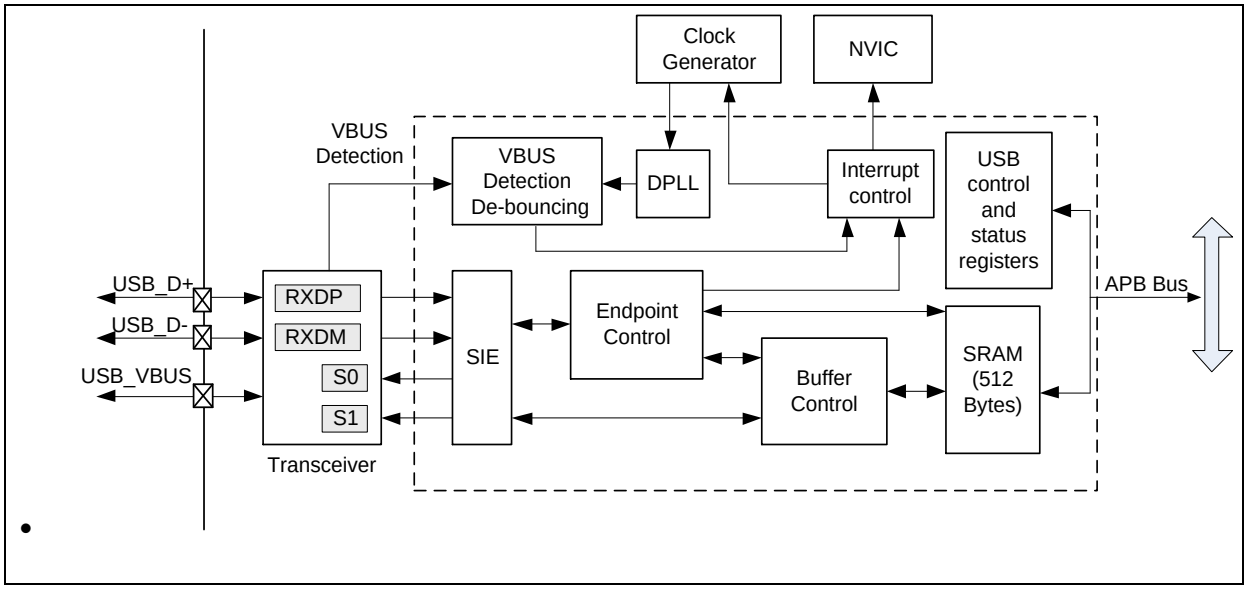


图 6.22-1 USB 框图

6.22.4 基本配置

USBD时钟源来自PLL，用户在使能USB设备控制器前必须要先设置好PLL相关的配置。USBDCKEN (CLK_APBCLK0[27])位使能USBD时钟，USBDIV (CLK_CLKDIV0[7:4])设置4位时钟预分频用来产生合适的USBD时钟

6.22.4.1 USB 2.0 全速设备基本配置

- 时钟源配置
  - 设置 PLL 控制器 (CLK_PLLCTL) 或者使能HIRC48M.
  - 在USBDIV(CLK_CLKDIV0[7:4])中选择USBD 外设时钟分频数目
  - 在 USBDCKEN (CLK_APBCLK0[27])中使能USBD外设时钟
- 复位配置
  - 在USBDRST (SYS_IPRST1[27])中复位USBD控制器

6.22.5 功能描述

6.22.5.1 串行接口引擎(SIE)

SIE是设备控制器的前端，用于处理USB协议。SIE的主要工作是向上发送信号到事务处理层，处理的事务包括：

- 包识别和事务排序
- SOP, EOP, RESET, RESUME信号检测与产生
- 时钟/数据的分离
- NRZI数据编解码和位填充
- CRC校验的生成和检测 (仅 Token 和 Data)

- 包 ID (PID) 产生和检查/解码
- 串-并/并-串 转换

6.22.5.2 端点控制

控制器中总共有8个端点。每个端点都可以配置成控制，批量，中断或同步传输模式。这四种模式所对应的传输过程都是通过这个模块来完成。该控制器也用于管理数据流同步，端点状态控制，当前端点起始地址，当前事务状态，以及每个端点的数据缓冲区状态

6.22.5.3 数字锁相环 (DPLL)

USB数据的传输比特率是12MHz，DPLL使用来自时钟控制器的48MHz时钟源来锁定RXDP和RXDM上的输入数据。12M的比特率时钟也是由DPLL转换而来

6.22.5.4 VBUS去抖检测

USB设备有可能经常在主机上被插拔。当USB设备从主机上拔下后，为了监测到它的状态，设备控制器提供了一个硬件去抖USB VBUS检测中断，来避免USB插拔时的抖动问题。USB设备插拔操作10ms后会产生一个VBUS检测中断。用户可以通过读USBD_VBUSDET寄存器的内容知道USB设备的插拔状态。VBUSDET标志反应了USB BUS 没有消抖情况下的当前状态。如果VBUSDET为1，表示USB线被插入。如果用户轮询该标志来检测USB的状态，需要通过软件来做消抖

6.22.5.5 中断控制

USB控制器带有一个USB中断向量，该中断包含了四个中断事件（唤醒事件，插拔事件，USB事件，BUS事件）。其中唤醒中断事件(NEVWK)发生在当系统从Power-down低功耗模式中唤醒时，（低功耗模式在Power-down控制寄存器CLK_PWRCTL做了定义）。插拔中断事件(VBUSDET)用于USB设备插拔检测。USB中断事件用于告知MCU产生了一些USB请求，如IN ACK, OUT ACK等。BUS中断事件用来告知MCU产生了总线事件，如挂起，恢复等。当需要用到这些中断时，必须在USB设备控制器的中断使能控制寄存器(USBD_INTEN)中打开相应位。

当系统在Power- down模式下唤醒后，如果20 ms内没有其它USB中断事件发生，则发生唤醒中断。系统进入Power-down后，如果USB唤醒功能使能，USB_VBUS,USB_D+ 和 USB_D-上的任何变化可以唤醒MCU。如果这个变化不是有意而为的，那么只有唤醒中断会发生。若 USB 唤醒超过 20 ms 后，如果没有其他 USB 中断事件发生，则唤醒中断将发生。图 6.22-2为唤醒中断的控制流程。

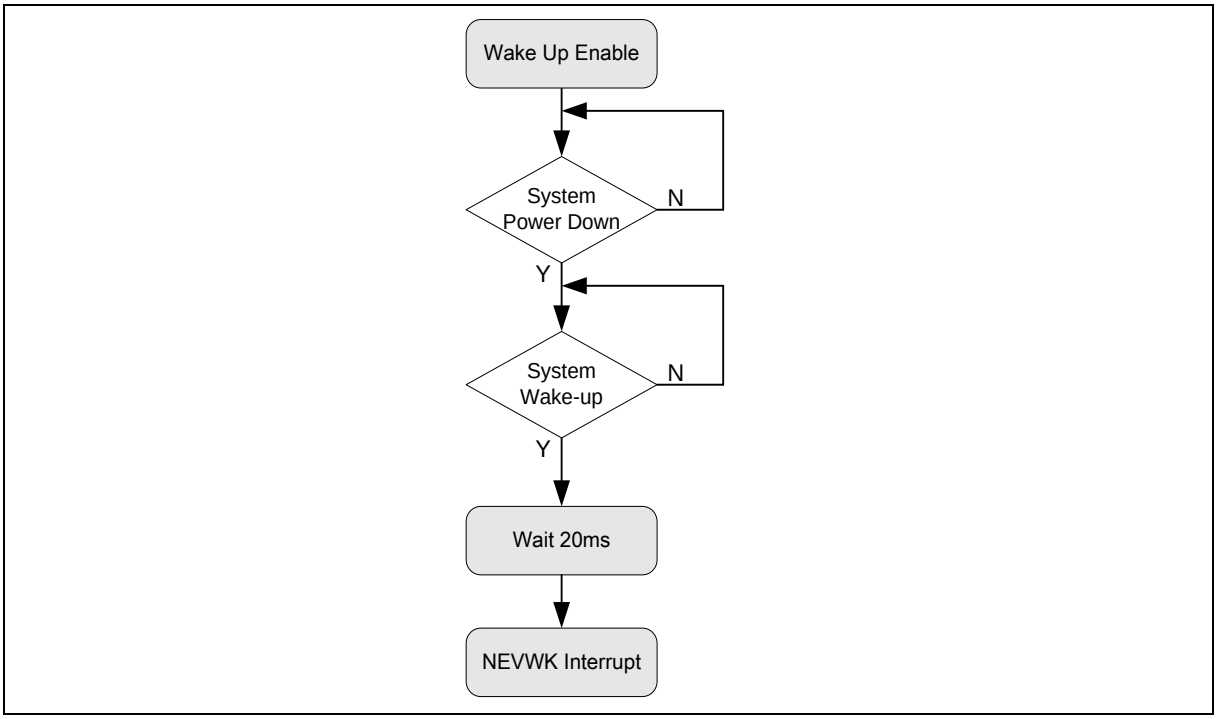


图 6.22-2 NEVWK 中断操作流程

USB中断事件用于告知MCU USB总线上所发生的事件，MCU可以读取EPSTS位(USBD_EPSTS0)和EPEVT7~0 (USBD_INTSTS[31:0])的内容，来知道USB当前的状态以便进行下一步操作。

和USB中断事件一样，BUS中断事件是用于告知MCU一些BUS事件。比如USB复位，挂起，超时溢出以及总线恢复等。用户可以读USBD_ATTR寄存器内容，从而知道USB的总线的状态。

6.22.5.6 省电

在一些特殊情况下，比如挂起，用户可以写0到USBD_ATTR[4]位来手动禁止PHY来省电。

6.22.5.7 缓存控制

USB控制器中总共有512字节的SRAM，8个端点可以共享这些缓存。在USB模块功能使能前，应先在缓冲段寄存器配置每个端点的有效起始地址。”缓冲区控制”模块就是用于控制每个端点的有效起始地址和它所分配的SRAM缓冲区间大小(在USBD_MXPLDx寄存器定义)。

图 6.22-3 描述了 根据USBD_BUFSEGx 和 USBD_MXPLDx寄存器中所定义的，各个端点缓冲区起始地址和大小。如果USBD_BUFSEG0被设置为0x08h, USBD_MXPLD0被设为0x40h,那么端点0 所分配的缓冲区的大小就是从USBD_BA+0x108h开始，到USBD_BA+0x148h结束。

注意：USBD 的SRAM起始地址是从USBD_BA+0x100h开始

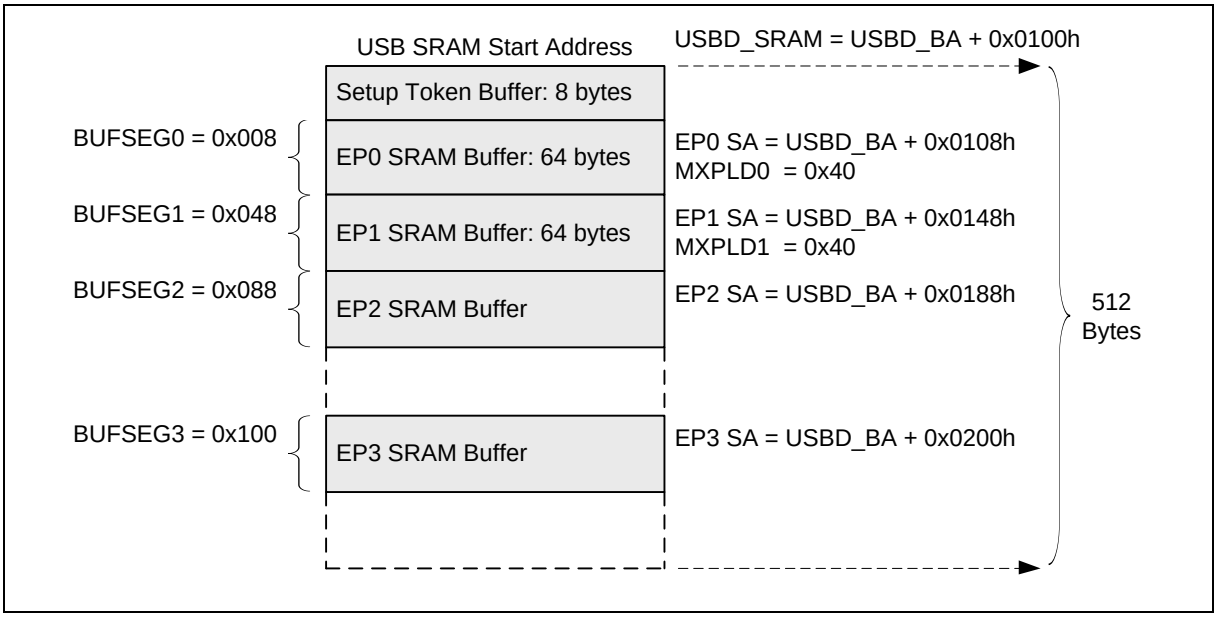


图 6.22-3 端点 SRAM 架构

6.22.5.8 与USB外设通信处理

用户可以使用中断或轮询USBD_INTSTS寄存器来监测USB的数据通信。当通信发生时，USBD_INTSTS寄存器被硬件置位，并向CPU产生一个中断请求(如果相关中断打开)，或者也可以不使用中断方式，用轮询USBD_INTSTS寄存器的相应位的方法来获取事件信息。以下是使用中断方式的控制流程。

当USB主机向设备控制器请求数据时，用户需要预先把相关数据放到指定的端点缓存。填充完数据后，用户需要写实际数据长度到USBD_MXPLDx这个寄存器当中。一旦这个寄存器数据被写入，内部信号“In_Rdy”信号会被设置，当接收到主机发来的IN token信号后，缓冲区数据会被立即传送出去。需要注意在指定数据发送完成后，“In_Rdy”信号会由硬件自动清除。

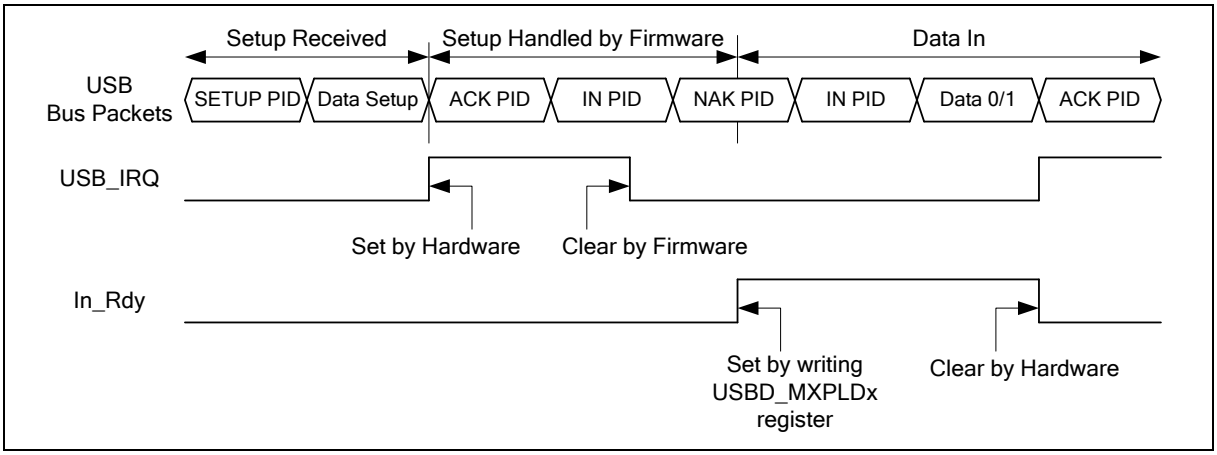


图 6.22-4 Setup 事务后接着是 Data IN 事务

相应的，当USB主机想要发送数据到从设备控制器的输出端点时，硬件会把数据填充到指定的端点缓冲区。通信完成后，硬件会在端点对应的USBD_MXPLDx寄存器中自动记录数据长度，并清除“Out_Rdy”

信号。这将会避免硬件在用户没有取走当前数据时又接收到下一笔数据。一旦用户处理了这次通信时，由软件写入特定的寄存器USBD_MXPLDx，以再次产生“Out_Rdy”信号来接收下一次通信。

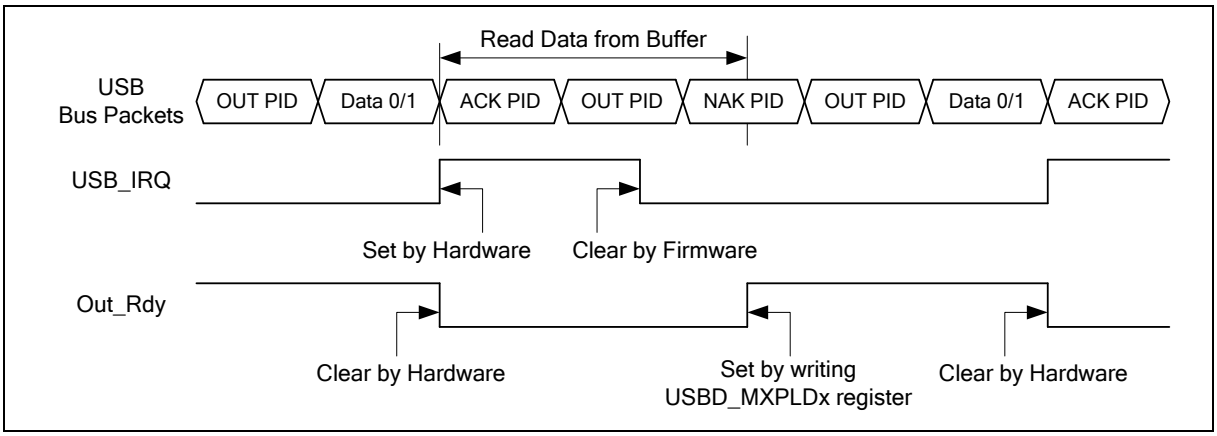


图 6.22-5 数据输出传输

6.22.5.9 链路电源管理 (LPM)

电源管理(LPM)就是类似于挂起/恢复功能，但是电源状态过渡延时只有10ms（取代了USB2.0挂起/恢复大于20ms的延时）

新的快速机制用于根集线器下USB总线从使能状态（L0）到新休眠状态（L1）的转换。L0和L1状态详见表 6.22-1，寄存器 USBD_ATTR & USBD_LPMATTR可以让用户了解LPM机制的当前电源状态

• **LPM 状态**	• **描述**
• L0(运行)	• 在这个状态，端口被使能可以进行信号的传输。L0下的端口不是在激活状态就是在空闲状态。所谓激活就是正在发送或是接收数据。这种状态下开始帧包（SOF）由主机产生，其速率与客户设备一致。
• L1(休眠)	• L1类似于下面的L2使用，但是还是有一点差别。进入L1需要向hub或是主机发出请求。一个LPM事务下发到设备。仅当在设备相应ACK握手后请求事务才能发生。通过远程唤醒、恢复信号、复位信号或是断开连接离开L1状态。连接的设备如L2一样但是L1不会从VBUS消耗多少功耗。当在L1时，主机或是设备可以初始化复位信号。尽管信号恢复如同L2，信号周期和L1到L0过渡延时更短。
• L2(挂起)	• 该状态就是USB2.0挂起状态。向hub或是主机端口发送命令进入L2。设备发现挂起条件是观察到3ms非活动状态。结果状态不是低速就是全速空闲。L2会从VBUS消耗很多功耗。通过远程唤醒、恢复信号、复位信号或是断开连接离开L2状态。
• L3(关闭)	• 这种状态下，端口不能进行数据传输。相当于断电，断开连接和禁用。

表 6.22-1 USB 链路电源管理 (Lx)状态

状态转化过程请参考图 6.22-6，更多的USB链路电源管理（LPM）信息请参考USB2.0链路电源管理ECN

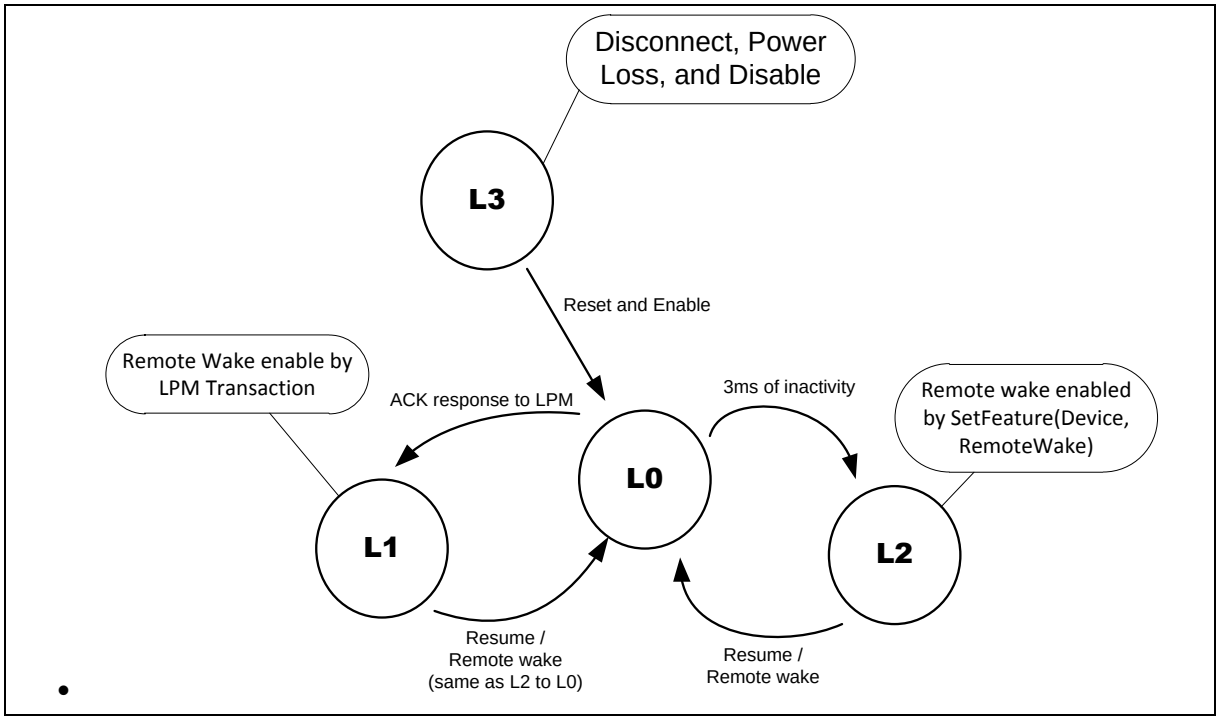


图 6.22-6 LPM 状态转换图

6.22.6 寄存器表

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移	R/W	描述	复位值
USBD基地址: **USBD_BA = 0x400C_0000**				
USBD_INTEN	USBD_BA+0x000	R/W	USB设备中断使能寄存器	0x0000_0000
USBD_INTSTS	USBD_BA+0x004	R/W	USB 设备中断事件状态寄存器	0x0000_0000
USBD_FADDR	USBD_BA+0x008	R/W	USB 设备功能地址寄存器	0x0000_0000
USBD_EPSTS	USBD_BA+0x00C	R	USB 设备端点状态寄存器	0x0000_0000
USBD_ATTR	USBD_BA+0x010	R/W	USB设备总线状态和属性寄存器	0x0000_0040
USBD_VBUSDET	USBD_BA+0x014	R	USB 设备VBUS 检测寄存器	0x0000_0000
USBD_STBUFSEG	USBD_BA+0x018	R/W	SETUP 令牌包缓存段寄存器	0x0000_0000
USBD_EPSTS0	USBD_BA+0x020	R	USB设备端点状态寄存器0	0x0000_0000
USBD_LPMATTR	USBD_BA+0x088	R	USB LPM 属性寄存器	0x0000_0000
USBD_FN	USBD_BA+0x08C	R	USB 帧号寄存器	0x0000_0XXX
USBD_SE0	USBD_BA+0x090	R/W	USB 设备驱动SE0控制寄存器	0x0000_0001
USBD_BUFSEG0	USBD_BA+0x500	R/W	端点0缓存段寄存器	0x0000_0000
USBD_MXPLD0	USBD_BA+0x504	R/W	端点0最大载荷寄存器	0x0000_0000
USBD_CFG0	USBD_BA+0x508	R/W	端点0 配置寄存器	0x0000_0000
USBD_CFGP0	USBD_BA+0x50C	R/W	端点0设置 Stall和清除In/Out 准备控制寄存器	0x0000_0000
USBD_BUFSEG1	USBD_BA+0x510	R/W	端点1 缓存段寄存器	0x0000_0000
USBD_MXPLD1	USBD_BA+0x514	R/W	端点1最大荷载寄存器	0x0000_0000
USBD_CFG1	USBD_BA+0x518	R/W	端点1配置寄存器	0x0000_0000
USBD_CFGP1	USBD_BA+0x51C	R/W	端点1 设置Stall和清除In/Out准备控制寄存器	0x0000_0000
USBD_BUFSEG2	USBD_BA+0x520	R/W	端点2 缓存段寄存器	0x0000_0000
USBD_MXPLD2	USBD_BA+0x524	R/W	端点2最大载荷寄存器	0x0000_0000
USBD_CFG2	USBD_BA+0x528	R/W	端点2配置寄存器	0x0000_0000
USBD_CFGP2	USBD_BA+0x52C	R/W	端点2设置Stall和清除In/Out准备控制寄存器	0x0000_0000
USBD_BUFSEG3	USBD_BA+0x530	R/W	端点3缓存段寄存器	0x0000_0000
USBD_MXPLD3	USBD_BA+0x534	R/W	端点3最大载荷寄存器	0x0000_0000
USBD_CFG3	USBD_BA+0x538	R/W	端点3配置寄存器	0x0000_0000

USBD_CFGP3	USBD_BA+0x53C	R/W	端点3设置Stall和清除In/Out准备控制寄存器	0x0000_0000
USBD_BUFSEG4	USBD_BA+0x540	R/W	端点 4缓存段寄存器	0x0000_0000
USBD_MXPLD4	USBD_BA+0x544	R/W	端点4最大载荷寄存器	0x0000_0000
USBD_CFG4	USBD_BA+0x548	R/W	端点4配置寄存器	0x0000_0000
USBD_CFGP4	USBD_BA+0x54C	R/W	端点4 设置Stall和清除In/Out 准备控制寄存器	0x0000_0000
USBD_BUFSEG5	USBD_BA+0x550	R/W	端点5缓存段寄存器	0x0000_0000
USBD_MXPLD5	USBD_BA+0x554	R/W	端点5最大载荷寄存器	0x0000_0000
USBD_CFG5	USBD_BA+0x558	R/W	端点5配置寄存器	0x0000_0000
USBD_CFGP5	USBD_BA+0x55C	R/W	端点5设置Stall和清除In/Out 准备控制寄存器	0x0000_0000
USBD_BUFSEG6	USBD_BA+0x560	R/W	端点6缓存段寄存器	0x0000_0000
USBD_MXPLD6	USBD_BA+0x564	R/W	端点6最大载荷寄存器	0x0000_0000
USBD_CFG6	USBD_BA+0x568	R/W	端点6配置寄存器	0x0000_0000
USBD_CFGP6	USBD_BA+0x56C	R/W	端点6设置Stall和清除In/Out 准备控制寄存器	0x0000_0000
USBD_BUFSEG7	USBD_BA+0x570	R/W	端点7缓存段寄存器	0x0000_0000
USBD_MXPLD7	USBD_BA+0x574	R/W	端点7最大载荷寄存器	0x0000_0000
USBD_CFG7	USBD_BA+0x578	R/W	端点7配置寄存器	0x0000_0000
USBD_CFGP7	USBD_BA+0x57C	R/W	端点7设置Stall和清除In/Out 准备控制寄存器	0x0000_0000

内存类别	地址	大小	描述
USBD_BA = 0x400C_0000			
USBD_SRAM	USBD_BA+0x100 ~ USBD_BA+0x2FF	512 字节	SRAM 用于整个端点缓存区 . 详细内容参考段0 端点SRAM结构和描述

6.22.7 寄存器描述

USB中断使能寄存器(USBD_INTEN)

寄存器	偏移	R/W	描述	复位值
USBD_INTEN	USBD_BA+0x000	R/W	USB 设备中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INNAKEN	保留						**WKEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留			**SOFIEN**	**NEVWKIEN**	**VBDETIEN**	**USBIEN**	**BUSIEN**

位	描述	
[31:16]	保留	保留.
[15]	**INNAKEN**	**收到IN Token 时，激活的 NAK功能及状态** 0 =当从设备收到 IN token 应答 NAK 时，IN NAK 状态不会被更新到端点状态寄存器USBD_EPSTS0，所以不会产生USB中断事件。 1 = 当从设备收到 IN token 时应答 NAK，IN NAK 的状态被更新到端点状态寄存器USBD_EPSTS0，并发生 USB 中断事件。
[14:9]	保留	保留.
[8]	**WKEN**	**唤醒功能使能使能位** 0 = USB 唤醒功能禁用. 1 = USB 唤醒功能使能.
[7:5]	保留	保留.
[4]	**SOFIEN**	**起始帧中断使能位** 0 = SOF 中断禁用. 1 = SOF 中断使能 .
[3]	**NEVWKIEN**	**USB 无事件唤醒中断使能位** 0 = 无事件唤醒中断禁用. 1 = 无事件唤醒中断使能.
[2]	**VBDETIEN**	**VBUS 检测中断使能位** 0 = VBUS 检测中断禁用. 1 = VBUS 检测中断使能.
[1]	**USBIEN**	**USB 事件中断使能位** 0 = USB事件中断禁用 .

		1 = USB 事件中断使能.
[0]	**BUSIEN**	**Bus事件中断使能位** 0 = BUS 事件中断禁用. 1 = BUS 事件中断使能.

USB中断事件状态寄存器 (USBD_INTSTS)

寄存器	偏移	R/W	描述	复位值
USBD_INTSTS	USBD_BA+0x004	R/W	USB设备中断事件状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
SETUP	保留						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EPEVT7	**EPEVT6**	**EPEVT5**	**EPEVT4**	**EPEVT3**	**EPEVT2**	**EPEVT1**	**EPEVT0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留			**SOFIF**	**NEVWKIF**	**VBDETIF**	**USBIF**	**BUSIF**

位	描述	
[31]	**SETUP**	**Setup 事件状态** 0 = 没有 Setup 事件. 1 = Setup 事件发生，通过写1到USBD_INTSTS[31]清零
[30:24]	保留	保留.
[23]	**EPEVT7**	**端点 7的USB 事件状态** 0 = 端点7没有事件发生 1 = 端点7有USB事件发生 ，通过检查USBD_EPSTS0[31:28] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[23]或者USBD_INTSTS[1]来清零该位。
[22]	**EPEVT6**	**端点 6的USB 事件状态** 0 =端点6没有事件发生. 1 =端点6有USB事件发生 ，通过检查USBD_EPSTS0[27:24] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[22]或者USBD_INTSTS[1]来清零该位。
[21]	**EPEVT5**	**端点 5的USB 事件状态** 0 =端点5没有事件发生. 1 =端点5有USB事件发生 ，通过检查USBD_EPSTS0[23:20] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[21]或者USBD_INTSTS[1]来清零该位。
[20]	**EPEVT4**	**端点 4的USB 事件状态** 0 =端点4没有事件发生. 1 =端点4有USB事件发生 ，通过检查USBD_EPSTS0[19:16] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[20]或者USBD_INTSTS[1]来清零该位。
[19]	**EPEVT3**	**端点 3的USB 事件状态** 0 =端点3没有事件发生. 1 =端点3有USB事件发生 ，通过检查USBD_EPSTS0[15:12] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[19]或者USBD_INTSTS[1]来清零该位。

[18]	**EPEVT2**	**端点 2的USB 事件状态** 0 =端点2没有事件发生. 1 =端点2有USB事件发生 ，通过检查USBD_EPSTS0[11:8] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[18]或者USBD_INTSTS[1]来清零该位。
[17]	**EPEVT1**	**端点 1的USB 事件状态** 0 =端点1没有事件发生. 1 =端点1有USB事件发生 ，通过检查USBD_EPSTS0[7:4] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[17]或者USBD_INTSTS[1]来清零该位。
[16]	**EPEVT0**	**端点 0的USB 事件状态** 0 =端点0没有事件发生. 1 =端点0有USB事件发生 ，通过检查USBD_EPSTS0[3:0] 来确定是那种USB事件发生，通过写1到USBD_INTSTS[16]或者USBD_INTSTS[1]来清零该位。
[15:5]	**保留**	保留.
[4]	**SOFIF**	**起始帧中断状态** 0 = 没有SOF事件发生. 1 = SOF 事件发生，通过写1到USBD_INTSTS[4]清零
[3]	**NEVWKIF**	**无事件唤醒中断状态** 0 = NEVWK 事件没有发生. 1 = 无事件唤醒中断发生，通过写1到USBD_INTSTS[3].清零该位
[2]	**VBDETIF**	**VBUS检测中断状态** 0 = 没有USB插入/拔出 事件发生. 1 = 有USB插入/拔出事件发生，通过写1到USBD_INTSTS[2].清零该位
[1]	**USBIF**	**USB 事件中断状态** USB事件包括在总线上的SETUP Token, IN Token, OUT ACK, ISO IN, 或 ISO OUT 事件 0 = 没有USB事件发生. 1 = USB事件发生，检查EPSTS0~7[2:0] 可以知道是何种USB事件发生，通过写1到USBD_INTSTS[1] 或者EPSTS0~7 和SETUP (USBD_INTSTS[31])清零该位。
[0]	**BUSIF**	**BUS中断事件状态** BUS中断事件说明总线上发生了挂起或者恢复的功能 0 = 没有总线事件发生. 1 = 发生总线事件，检查USBD_ATTR[3:0] 可以知道是何种总线事件发生，通过写1到USBD_INTSTS[0]清零该位

USB设备功能地址寄存器(USBD_FADDR)

7位有效数据用于设置USB总线上的设备地址

寄存器	偏移	R/W	描述	复位值
USBD_FADDR	USBD_BA+0x008	R/W	USB设备功能地址寄存器	0x0000_0000

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

位	描述	
[31:7]	保留	保留.
[6:0]	**FADDR**	USB 设备功能地址

USB端点状态寄存器(USBD_EPSTS)

寄存器	偏移	R/W	描述	复位值
USBD_EPSTS	USBD_BA+0x00C	R	USB设备端点状态寄存器	0x0000_0000

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

位	描述	
[31:8]	保留	保留.
[7]	**OV**	**超负荷** 用于指示接收的数据长度是否超出最大负荷范围 如果接收的数据超出了最大负荷数，多出的数据将被忽略掉 0 = 没有超负荷. 1 = 主机发送的数据超过了MXPLD设置的最大范围，或者Setup 数据超过了8个字节.
[6:0]	保留	保留.

USB总线状态和属性寄存器(USBD_ATTR)

寄存器	偏移	R/W	描述	复位值
USBD_ATTR	USBD_BA+0x010	R/W	USB设备总线状态和属性寄存器	0x0000_0040

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留		**L1RESUME**	**L1SUSPEND**	**LPMACK**	**BYTEM**	**PWRDN**	**DPPUEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USBEN	保留	**RWAKEUP**	**PHYEN**	**TOUT**	**RESUME**	**SUSPEND**	**USBRST**

位	描述	
[31:14]	保留	保留.
[13]	**L1RESUME**	**LPM L1 恢复 (只读)** 0 = 总线没有 LPM L1 状态恢复. 1 = LPM L1 从挂起状态恢复.
[12]	**L1SUSPEND**	**LPM L1 挂起 (只读)** 0 = 总线没有LPM L1 状态挂起 . 1 = 该位在LPM命令进入L1状态被成功接收和应答的时候由硬件置位。
[11]	**LPMACK**	**LPM Token** 应答使能位 NYET/ACK仅在一个成功的LPM事务后被返回也就是在EXT token 和LPM token和有效的bLinkState = 0001 (L1)被接收时没有出错，否则ERROR 和 STALL分别会自动地被返回 0= 有效的 LPM Token 不会被应答 1= 有效的 LPM Token 会被应答
[10]	**BYTEM**	**CPU** 存取 **USB SRAM** 大小模式选择 0 = 字模式: CPU到 USB SRAM 的数据传输必须以字为单位 1 = 字节模式: CPU到 USB SRAM 的数据传输必须以字节为单位
[9]	**PWRDN**	**PHY收发器掉电控制，低有效** 0 = PHY收发器相关电路掉电 1 = PHY收发器相关电路上电
[8]	**DPPUEN**	**USB_DP 管脚上拉电阻使能位** 0 = USB_D+管脚上的上拉电阻禁用 . 1 = USB_D+ 管脚上的上拉电阻使能.
[7]	**USBEN**	**USB 控制器使能位** 0 = USB控制器禁用 1 = USB 控制器使能.
[6]	保留	保留.

[5]	**RWAKEUP**	**远程唤醒** 0 =USB总线从K状态释放 . 1 =强迫USB总线到K (USB_D+ low, USB_D-: high) 状态, 用于远程唤醒
[4]	**PHYEN**	**PHY 收发器使能位** 0 = PHY 收发器功能禁用. 1 = PHY收发器功能使能 .
[3]	**TOUT**	**时间溢出状态 (只读)** 当USB设备控制器收到设置令牌或者发送令牌，USB控制器就保持在J状态等待数据包，如果等待的时间超过了18个位数据长度的时间，TOUT标志就会产生。 0 = 没有时间溢出 1 = 超过18个位计数时间没有总线回应.
[2]	**RESUME**	**恢复状态 (只读)** 0 = 总线没有恢复. 1 = 总线从挂起状态恢复
[1]	**SUSPEND**	**挂起状态 (只读)** 0 = 总线没有挂起. 1 = 总线空闲超过3ms，有可能是从设备拔下或者主机进入睡眠模式
[0]	**USBRST**	**USB 复位状态 (只读)** 0 = 总线没有复位. 1 = 总线有复位，此时SE0 (single-ended 0) 超过2.5us.

USB设备VBUS检测寄存器(USBD_VBUSDET)

寄存器	偏移	R/W	描述	复位值
USBD_VBUSDET	USBD_BA+0x014	R	USB 设备VBUS检测寄存器	0x0000_0000

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

位	描述	
[31:1]	保留	保留
[0]	**VBUSDET**	**设备VBUS 检测** 0 = 控制器没有连接USB 主机. 1 = 控制设备连接USB主机.

USB SETUP 令牌缓存段寄存器(USBD_STBUFSEG)

寄存器	偏移	R/W	描述	复位值
USBD_STBUFSEG	USBD_BA+0x018	R/W	SETUP令牌缓存段寄存器	0x0000_0000

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

位	描述	
[31:9]	保留	保留
[8:3]	**STBUFSEG**	**SETUP 令牌缓存段 设置位** 用于指示 SETUP 令牌包在USB设备SRAM中的起始地址，它的有效起始地址是USBD_SRAM 地址 + {STBUFSEG[8:3], 3'b000} 此处USBD_SRAM 地址 = USBD_BA+0x100h. 注意:该内容只适用于SETUP令牌包.
[2:0]	保留	保留.

USB端点状态寄存器 0 (USBD_EPSTS0)

寄存器	偏移	R/W	描述	复位值
USBD_EPSTS0	USBD_BA+0x020	R	USB设备端点状态寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS7				**EPSTS6**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EPSTS5				**EPSTS4**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EPSTS3				**EPSTS2**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPSTS1				**EPSTS0**			

位	描述	
[31:28]	**EPSTS7**	**端点7状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束 .
[27:24]	**EPSTS6**	**端点6状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.
[23:20]	**EPSTS5**	**端点5状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束

[19:16]	EPSTS4	端点4状态 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.
[15:12]	EPSTS3	**端点3状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.
[11:8]	EPSTS2	**端点2状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.
[7:4]	EPSTS1	**端点1状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.
[3:0]	EPSTS0	**端点0状态** 该位用于指示该端点的当前状态 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = 同步传输结束.

USB LPM属性寄存器(USBD_LPMATTR)

寄存器	偏移	R/W	描述	复位值
USBD_LPMATTR	USBD_BA+0x088	R	USB LPM属性寄存器	0x0000_0000

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

位	描述	
[31:9]	保留	保留
[8]	**LPMRWAKUP**	**LPM 远程唤醒** 该位包含接收的最后ACK LPM Token的远程唤醒值
[7:4]	**LPMBESL**	**LPM 最大力度服务延时** 该位包含接收的最后ACK LPM Token的BESL值
[3:0]	**LPMLINKSTS**	**LPM 链路状态** 该位包含接收的最后ACK LPM Token的链路状态

USB帧号寄存器(USBD_FN)

寄存器	偏移	R/W	描述	复位值
USBD_FN	USBD_BA+0x08C	R	USB帧号寄存器	0x0000_0XXX

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

位	描述	
[31:11]	**保留**	保留
[10:0]	**FN**	**帧号** 该域包含了最近一次收到的SOF包的11位的帧号 .

USB驱动 SE0 寄存器(USBD_SE0)

寄存器	偏移	R/W	描述	复位值
USBD_SE0	USBD_BA+0x090	R/W	USB设备驱动SE0 控制寄存器	0x0000_0001

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留							**SE0**

位	描述	
[31:1]	保留	保留
[0]	**SE0**	**在USB总线上驱动单端0控制位** 当USB_D+ 和 USB_D-都拉低时，为单端0 (SE0)。 0 = 正常操作 1 = 强制USB PHY 收发器驱动SE0.

USB缓存段寄存器(USB_BUFSEGx)

寄存器	偏移	R/W	描述	复位值
USBD_BUFSEG0	USBD_BA+0x500	R/W	端点 0 缓存段寄存器	0x0000_0000
USBD_BUFSEG1	USBD_BA+0x510	R/W	端点1 缓存段寄存器	0x0000_0000
USBD_BUFSEG2	USBD_BA+0x520	R/W	端点2 缓存段寄存器	0x0000_0000
USBD_BUFSEG3	USBD_BA+0x530	R/W	端点3 缓存段寄存器	0x0000_0000
USBD_BUFSEG4	USBD_BA+0x540	R/W	端点4 缓存段寄存器	0x0000_0000
USBD_BUFSEG5	USBD_BA+0x550	R/W	端点5 缓存段寄存器	0x0000_0000
USBD_BUFSEG6	USBD_BA+0x560	R/W	端点6 缓存段寄存器	0x0000_0000
USBD_BUFSEG7	USBD_BA+0x570	R/W	端点7 缓存段寄存器	0x0000_0000

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

位	描述	
[31:9]	保留	保留
[8:3]	**BUFSEG**	端点缓存段 用于指示每个端点在USB SRAM 的偏移地址，端点的有效起始地址是 USBD_SRAM 地址 + { BUFSEG[8:3], 3'b000} 此处USBD_SRAM 地址 = USBD_BA+0x100h. 请参考段 章节 0.关于端点SRAM的结构和相关描述
[2:0]	保留	保留

USB最大负荷设置寄存器(USB_MXPLDx)

寄存器	偏移	R/W	描述	复位值
USBD_MXPLD0	USBD_BA+0x504	R/W	端点0 最大负荷寄存器	0x0000_0000
USBD_MXPLD1	USBD_BA+0x514	R/W	端点1最大负荷寄存器	0x0000_0000
USBD_MXPLD2	USBD_BA+0x524	R/W	端点2最大负荷寄存器	0x0000_0000
USBD_MXPLD3	USBD_BA+0x534	R/W	端点3最大负荷寄存器	0x0000_0000
USBD_MXPLD4	USBD_BA+0x544	R/W	端点4最大负荷寄存器	0x0000_0000
USBD_MXPLD5	USBD_BA+0x554	R/W	端点5最大负荷寄存器	0x0000_0000
USBD_MXPLD6	USBD_BA+0x564	R/W	端点6最大负荷寄存器	0x0000_0000
USBD_MXPLD7	USBD_BA+0x574	R/W	端点7最大负荷寄存器	0x0000_0000

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

位	描述	
[31:9]	保留	保留
[8:0]	**MXPLD**	最大有效负荷位 定义了发送到主机(IN token)的实际数据长度，或 从主机接收到数据 (OUT token)的实际长度. 也用于指示IN token时做好了发送数据准备，或OUT token时做好了接收数据准备. 1) 当CPU写入值到该寄存器后, 对 IN token, MXPLD寄存器的值用于指示要发送的数据长度，且已做好发送准备。 对OUT token,设备已经做好了从主机接收数据的准备。MXPLD的值就表示从主机所接收数据的最大长度。 (2) 当CPU读该寄存器时, 对IN token, MXPLD 的值表示要发送到主机的数据长度 对OUT token, MXPLD 的值表示从主机接收到的实际数据长度 注意**:** 一旦 MXPLD 的值被写入, 收到IN/OUT token后.数据包可以立即收发

USB配置寄存器 (USB_CFGx)

寄存器	偏移	R/W	描述	复位值
USBD_CFG0	USBD_BA+0x508	R/W	端点 0 配置寄存器	0x0000_0000
USBD_CFG1	USBD_BA+0x518	R/W	端点 1配置寄存器	0x0000_0000
USBD_CFG2	USBD_BA+0x528	R/W	端点 2 配置寄存器	0x0000_0000
USBD_CFG3	USBD_BA+0x538	R/W	端点 3 配置寄存器	0x0000_0000
USBD_CFG4	USBD_BA+0x548	R/W	端点 4 配置寄存器	0x0000_0000
USBD_CFG5	USBD_BA+0x558	R/W	端点 5 配置寄存器	0x0000_0000
USBD_CFG6	USBD_BA+0x568	R/W	端点 6 配置寄存器	0x0000_0000
USBD_CFG7	USBD_BA+0x578	R/W	端点 7 配置寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留						**CSTALL**	保留
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DSQSYNC	**STATE**		**ISOCH**	**EPNUM**			

位	描述	
[31:10]	保留	保留.
[9]	**CSTALL**	**清 STALL 响应位** 0 =在setup阶段禁止设备清除STALL 1 = 在setup阶段应许设备清除STALL
[8]	保留	保留.
[7]	**DSQSYNC**	**数据时序同步位** 0 = DATA0 PID. 1 = DATA1 PID. **IN Token 传输:** 该位用于指定在接下来的IN token 传输过程是DATA0还是DATA1 PID. 在IN token过程，该位会基于硬件自动反转 **OUT Token 传输:** 该位用于指定在接下来的OUT token 传输过程是DATA0还是DATA1 PID. 在IN token过程，该位会基于硬件自动反转
[6:5]	**STATE**	**端点 状态**

		00 = 端点禁止. 01 = 输出端点. 10 = 输入端点. 11 = 未定义
[4]	**ISOCH**	**同步端点** 该位用于设置端点为同步端点，无握手信号 0 =不是同步端点 . 1 = 同步端点.
[3:0]	**EPNUM**	**端点号** 该位用于定义当前端点的端点号

USB扩展配置寄存器(USB_CFGPx)

寄存器	偏移	R/W	描述	复位值
USBD_CFGP0	USBD_BA+0x50C	R/W	端点 0 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP1	USBD_BA+0x51C	R/W	端点 1 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP2	USBD_BA+0x52C	R/W	端点 2 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP3	USBD_BA+0x53C	R/W	端点 3设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP4	USBD_BA+0x54C	R/W	端点 4设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP5	USBD_BA+0x55C	R/W	端点 5 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP6	USBD_BA+0x56C	R/W	端点 6 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000
USBD_CFGP7	USBD_BA+0x57C	R/W	端点 7 设置 Stall 和清In/Out 准备控制寄存器	0x0000_0000

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

位	描述	
[31:2]	保留	保留.
[1]	**SSTALL**	**设置 STALL** 0 = 禁止设备响应STALL. 1 = 设置设备自动响应STALL.
[0]	**CLRRDY**	**清除准备位** USBD_ MXPLDx寄存器被设置后，表示该端点准备好可以发送或接收数据。如果用户想在传输开始前关闭传输，需要设置该位为1来进行关闭，该位会自动清零。 对IN token，写 ‘1’ 清除 IN token 时发送数据到 USB 的准备信号 对OUT token，写 ‘1’ 清除 OUT token 时从 USB 接收数据的准备信号 该位只能写 1，读数据返回值总是 0。

6.23 CRC控制器 (CRC)

6.23.1 概述

CRC循环冗余发生器使用4种常见的多项式CRC-CCITT, CRC-8, CRC-16, 和 CRC-32 执行 CRC 计算

6.23.2 特性

- 支持4种常见的多项式 CRC-CCITT, CRC-8, CRC-16, 和CRC-32
  - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
  - CRC-8: $X^8 + X^2 + X + 1$
  - CRC-16: $X^{16} + X^{15} + X^2 + 1$
  - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 可编程种子值
- 对于输入数据和CRC 校验和，支持可编程的位顺序反转设定
- 对于输入数据和CRC 校验和，支持可编程的补码设定
- 支持 8/16/32-位 数据宽度
  - 8-位写模式: 1-AHB 时钟周期操作
  - 16-位写模式: 2-AHB 时钟周期操作
  - 32-位写模式: 4-AHB 时钟周期操作
- 支持使用 PDMA 写数据去执行CRC操作

6.23.3 框图

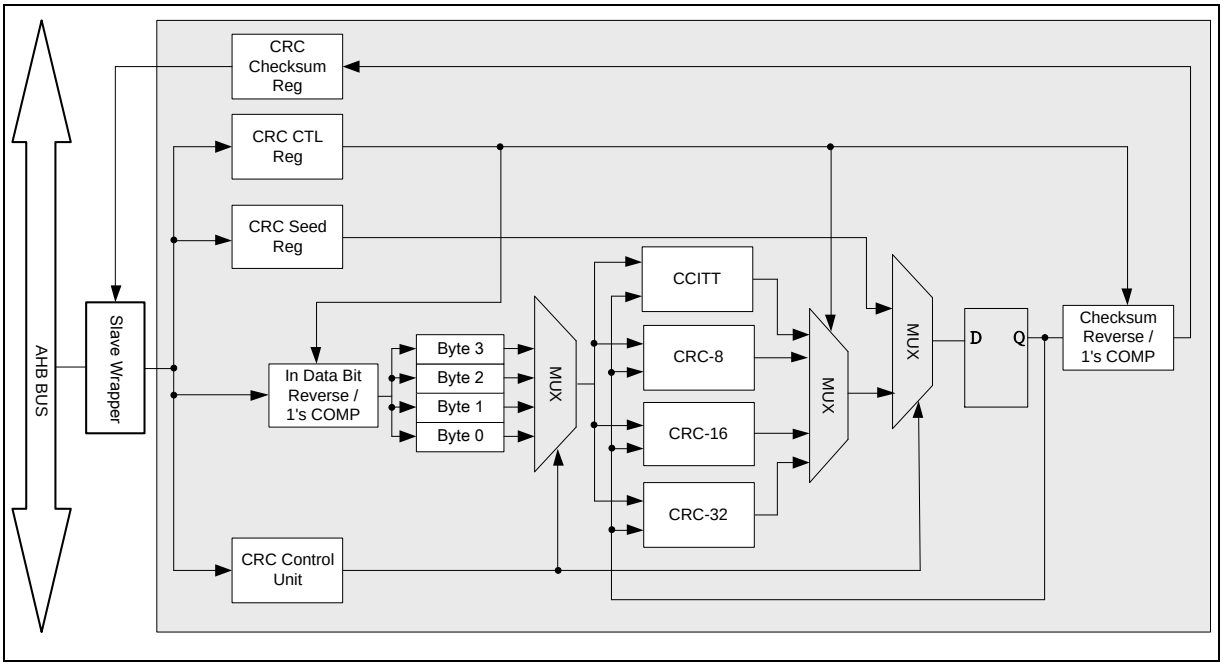


图 6.23-1 CRC 产生器框图

6.23.4 基本配置

- 时钟源配置
  - 在CRCCKEN (CLK_AHBCLK[7])中使能CRC外设时钟
- 复位配置
  - 在CRCRST (SYS_IPRST0[7])中复位CRC控制器 .

6.23.5 功能描述

CRC 计算可选择4种常见多项式：CRC-CCITT, CRC-8, CRC-16 和 CRC-32；通过 CRCMODE[1:0] (CRC_CTL[31:30])选择CRC多项式

下面是编程流程示例.

1. 设置CRCEN (CRC_CTL[0] CRC通道使能位)使能CRC发生器
2. CRC计算初始化设定.
   - 设置CHKSFMT (CRC_CTL[27])配置CRC校验和补码
   - 设置CHKSREV (CRC_CTL[25]) 配置CRC校验和位顺序反转，功能框图请参考图 6.23-2 图 6.23-2校验和位反转功能框图
   - 设置DATFMT (CRC_CTL[26]配置 CRC写数据补码 。
   - 设置DATREV (CRC_CTL[24]配置CRC写数据每个位字节的顺序反转，详见功能框图 6.23-3.
3. 执行 CHKSINIT (CRC_CTL[1]从CRC_SEED寄存器中载入校验和初值
4. 写数据到CRC_DAT 寄存器计算CRC校验和

5. 通过读CRC_CHECKSUM 寄存器获取CRC校验和结果

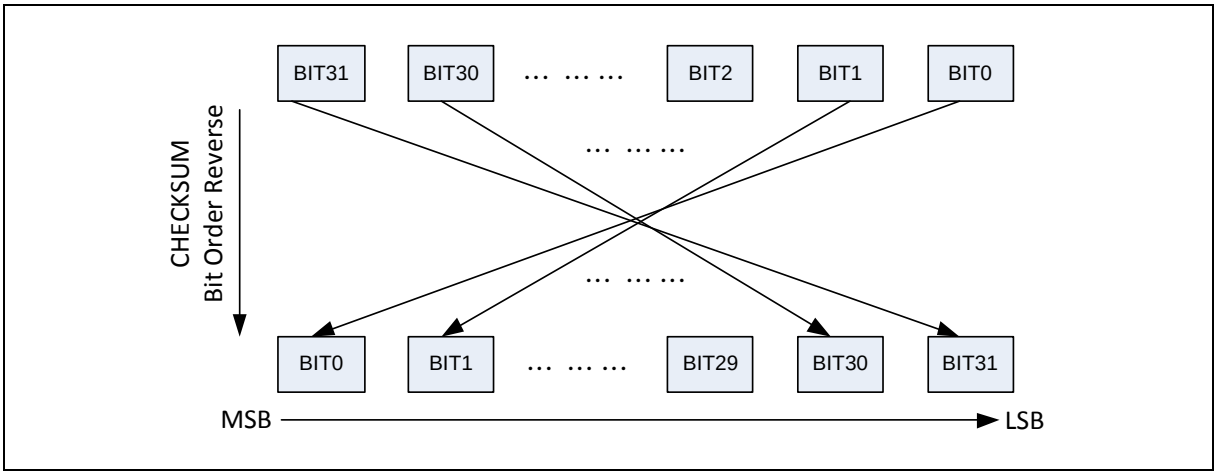


图 6.23-2 校验和位反转功能框图

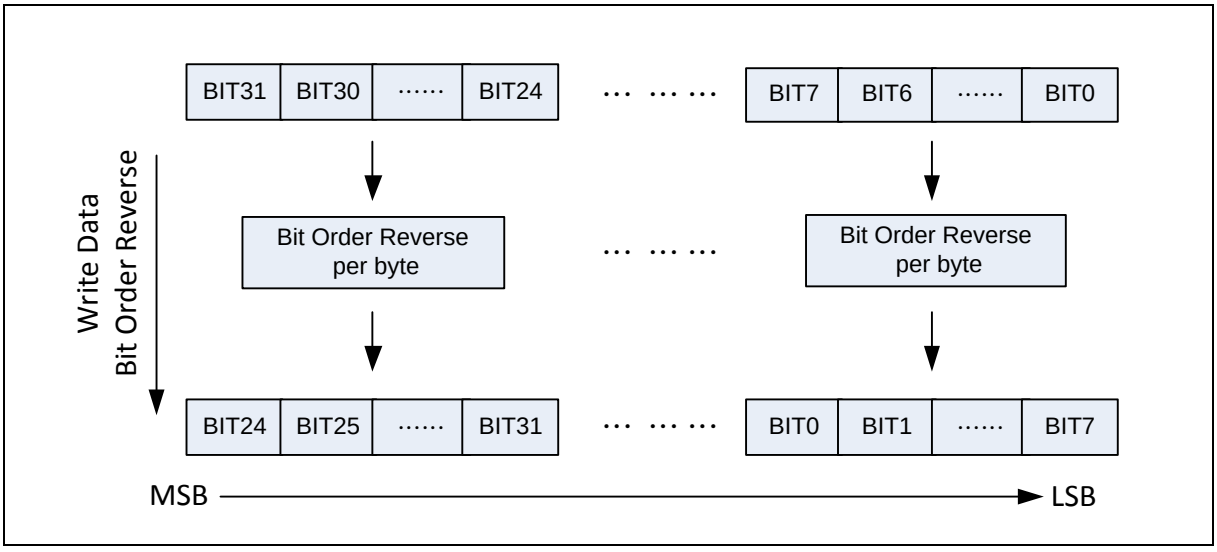


图 6.23-3 写数据位顺序反转功能框图

6.23.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读写

寄存器	偏移	R/W	描述	复位值
CRC 基地址: **CRC_BA = 0x4003_1000**				
CRC_CTL	CRC_BA+0x00	R/W	CRC 控制寄存器	0x2000_0000
CRC_DAT	CRC_BA+0x04	R/W	CRC 写数据寄存器	0x0000_0000
CRC_SEED	CRC_BA+0x08	R/W	CRC 种子寄存器	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC 校验和寄存器	0xFFFF_FFFF

6.23.7 寄存器描述

CRC 控制寄存器 (CRC_CTL)

寄存器	偏移	R/W	描述	复位值
CRC_CTL	CRC_BA+0x00	R/W	CRC 控制寄存器	0x2000_0000

31	30	29	28	27	26	25	24
CRCMODE		**DATLEN**		**CHKSFMT**	**DATFMT**	**CHKSREV**	**DATREV**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留						**CHKSINIT**	**CRCEN**

位	描述	
[31:30]	**CRCMODE**	**CRC 多项式模式** 这里选择CRC操作多项式模式. 00 = CRC-CCITT 多项式模式. 01 = CRC-8 多项式模式. 10 = CRC-16 多项式模式. 11 = CRC-32 多项式模式.
[29:28]	**DATLEN**	**CPU 写数据长度** 这里设置写数据长度. 00 = 8位数据长度模式. 01 = 16位数据长度模式. 1x = 32位数据长度模式. 注意：当写数据长度是8-bit 模式， CRC_DAT 寄存器的有效数据是DATA[7:0]位；如果写数据长度是 16-bit 模式, CRC_DAT寄存器的有效数据是 DATA[15:0].
[27]	**CHKSFMT**	**校验和补码** 该位用于使能CRC_CHECKSUM寄存器中的校验和补码功能. 0 = 禁用CRC校验和补码功能. 1 = 使能CRC校验和补码功能.
[26]	**DATFMT**	**写数据补码** 该位用于使能写数据到CRC_DAT寄存器中的补码功能 0 =CRC写数据补码功能禁用. 1 = CRC写数据补码功能使能.

[25]	**CHKSREV**	**校验和位顺序反转** 该位用于使能CRC_CHECKSUM 寄存器中的校验和结果的位顺序反转功能 0 = 禁用CRC校验和位顺序反转. 1 = 使能CRC校验和位顺序反转. **注:** 如果校验和的结果是0xDD7B0F2E, 位顺序反转后CRC校验和是0x74F0DEBB.
[24]	**DATREV**	**写数据位顺序反转** 该位用于使能写数据到CRC_DAT寄存器每个字节的位顺序反转功能 0 = CRC写数据位顺序反转禁用. 1 = CRC写数据位顺序反转使能 (每字节). **注:** 如果写数据是0xAABBCCDD, 位顺序反转后CRC写数据是0x55DD33BB.
[23:2]	**保留**	保留.
[1]	**CHKSINIT**	**校验和初始化** 0 = 无影响. 1 = 通过自动载入 CRC_SEED 寄存器的值到CRC_CHECKSUM 寄存器，初始化校验和的值 **注:** 该位自动清零.
[0]	**CRCEN**	**CRC 通道使能位** 0 = 无影响. 1 = CRC 操作使能.

CRC 写数据寄存器 (CRC_DAT)

寄存器	偏移	R/W	描述	复位值
CRC_DAT	CRC_BA+0x04	R/W	CRC 写数据寄存器	0x0000_0000

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

位	描述	
[31:0]	**DATA**	**CRC写数据位** 用户可以通过CPU模式，或者使用PDMA功能直接写数据到这个寄存器执行CRC操作 **注:** 当数据长度是8-bit 模式， CRC_DAT 寄存器的有效数据是DATA[7:0]位; 如果写数据长度是 16-bit 模式, CRC_DAT寄存器的有效数据是 DATA[15:0].

CRC 种子寄存器 (CRC_SEED)

寄存器	偏移	R/W	描述	复位值
CRC_SEED	CRC_BA+0x08	R/W	CRC 种子寄存器	0xFFFF_FFFF

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

位	描述	
[31:0]	**SEED**	**CRC 种子值** 该位表示CRC 种子值. **注：** 在执行了 CHKSINIT (CRC_CTL[1])，该值将重新载入作为校验和的初始值 (CRC_CHECKSUM 寄存器)

CRC 校验和寄存器 (CRC_CHECKSUM)

寄存器	偏移	R/W	描述	复位值
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC 校验和寄存器	0xFFFF_FFFF

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

位	描述	
[31:0]	**CHECKSUM**	**CRC 校验和结果** 这些位表示CRC校验和的结果.

6.24 硬件除法器(HDIV)

6.24.1 概述

硬件除法器 (HDIV) 在高性能的应用中很有用，该硬件除法器是有符号的整数除法器，可以输出商和余数

6.24.2 特性

- 有符号 (2的补码)整数计算
- 32-位被除数，16-位除数的计算能力
- 32-位商和32-位余数的输出（16位余数加上符号扩展到32位 ）
- 除0警告标志
- 写除数触发计算

6.24.3 基本配置

在使用硬件除法器之前，除法器的时钟必须使能，使能硬件除法器需要置AHBCLK[4]的HDIV_EN 为1

6.24.4 功能描述

使用硬件除法器，首先需要设置被除数，然后设置除数。在除数写入之后，硬件除法器将自动触发计算。包含商和余数的计算结果可以通过DIVQUO 和 DIVREM寄存器读。用户可以在写入除数一个时钟周期后读取商和余数

如果除数是0，DIVSTS的DIV0标志将被置0。

被除数为32位有符号整数，除数为16位有符号整数。商为32位有符号整数，余数为16位有符号整数。

图 6.24-1展示了硬件除法器的操作流程。为了计算X/Y，CPU需要写X到DIVIDEND 寄存器，然后写Y到DIVISOR 。在DIVISOR 写入之后，CPU可以读DIVQUO 和 DIVREM寄存器获取计算结果。

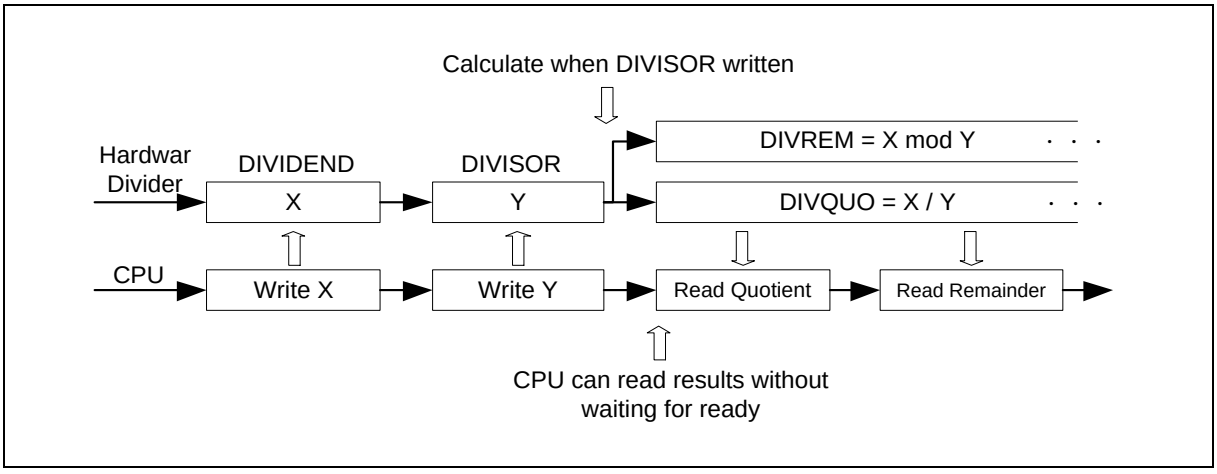


图 6.24-1 硬件除法器操作流程

6.24.5 寄存器表

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移	R/W	描述	复位值
HDIV基地址: **HDIV_BA = 0x4001_4000**				
DIVIDEND	HDIV_BA+0x00	R/W	被除数源寄存器	0x0000_0000
DIVISOR	HDIV_BA+0x04	R/W	除数源寄存器	0x0000_FFFF
DIVQUO	HDIV_BA+0x08	R/W	商结果寄存器	0x0000_0000
DIVREM	HDIV_BA+0x0C	R/W	余数结果寄存器	0x0000_0000
DIVSTS	HDIV_BA+0x10	R	除法器状态寄存器	0x0000_0001

6.24.6 寄存器描述

被除数源寄存器 (DIVIDEND)

寄存器	偏移	R/W	描述	复位值
DIVIDEND	HDIV_BA+0x00	R/W	被除数源寄存器	0x0000_0000

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

位	描述	
[31:0]	**DIVIDEND**	**被除数源** 该寄存器是在除法器计算之前给予被除数

除数源寄存器 (DIVISOR)

寄存器	偏移	R/W	描述	复位值
DIVISOR	HDIV_BA+0x04	R/W	除数源寄存器	0x0000_FFFF

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

位	描述	
[31:16]	**保留**	保留
[15:0]	**DIVISOR**	**除数源** 该寄存器是在除法器计算之前给予除数 **注:** 当写该寄存器后，硬件除法器开始计算

商结果寄存器 (DIVQUO)

寄存器	偏移	R/W	描述	复位值
DIVQUO	HDIV_BA+0x08	R/W	商结果寄存器	0x0000_0000

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

位	描述	
[31:0]	**QUOTIENT**	**商结果** 该位用于保存计算完成后除法器的商结果

余数结果寄存器 (DIVREM)

寄存器	偏移	R/W	描述	复位值
DIVREM	HDIV_BA+0x0C	R/W	余数结果寄存器	0x0000_0000

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

位	描述	
[31:16]	**REMAINDER[31:16]**	**REMAINDER[15:0]符号扩展** 硬件除法器的余数是16位的带符号整数(REMAINDER[15:0])通过符号扩展到32位整数
[15:0]	**REMAINDER[15:0]**	**余数结果** 该寄存器用于保存计算完成后除法器的余数结果

除法器状态寄存器 (DIVSTS)

寄存器	偏移	R/W	描述	复位值
DIVSTS	HDIV_BA+0x10	R	除法器状态寄存器	0x0000_0001

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留						**DIV0**	保留

位	描述	
[31:2]	保留	保留.
[1]	**DIV0**	**除数为0警告** 0 = 除数不为0. 1 = 除数为0. **注:** DIV0 标志用于表示除数为0的情况，当DIVISOR被写入时更新。
[0]	保留	保留.

6.25 模数转换器(ADC)

6.25.1 概述

包含一个12位逐次逼近型模数转换器(SAR A/D转换器)，包含16个输入通道。A/D转换器支持四种操作模式：单次模式、Burst 模式、单周期扫描模式和连续扫描模式。A/D转换器可以通过软件、外部STADC引脚(STADC), 定时器0~3溢出脉冲触发，PWM或BPWM触发转换

6.25.2 特性

- 操作电压: 1.8V~3.6V.
- 模拟输入电压: 0 ~ AV_{DD}.
- 支持VREF管脚输入的外部参考电压.
- 12-位解析度和10-位精度保证.
- 16路单端输入或8对差分输入
- 最大ADC外设时钟频率48MHz.
- 高达2 MSPS 采样率.
- 扫描使能通道
- 门限电压检测
- 4种操作模式 ：
  - 单一模式:在指定通道执行一次 A/D 转换
  - Burst 模式: A/D 转换器连续的采样转换指定通道并将结果存储到FIFO中
  - 单周期扫描模式: A/D转换器执行一次指定的几个通道的AD采集转换，其采集转换顺序是从编号最小的通道到编号最大的通道 。.
  - 连续扫描模式: A/D 转换器连续的执行单周期扫描模式直到软件停止A/D转换。
- A/D 转换可有一下几种方式触发:
  - 软件写 1 到 ADST 位.
  - 外部管脚 (STADC).
  - 定时器 0~3 溢出脉冲.
  - PWM触发
  - BPWM触发.
- 每个通道的转换结果都存到对应的数据寄存器中，并且带有有效和覆盖指示。
- 转换结果可以和一个特殊的值比较，用户可以选择在与比较寄存器值相等时是否产生中断
- 支持延长采样时间功能 (0~255 ADC 时钟).
- 一个内部band-gap 电压 (VBG)通道.
- 一个内部上拉/下拉电路通道
- 支持 PDMA 传输模式.
- 支持校准模式.

- 支持浮空检测功能

注1: ADC 采样率 = (ADC 外设时钟频率) / (总共 ADC 转换周期)

注2: 如果内部band-gap 电压通道起效, 最大采样率为300 k SPS.

注3: ADC 时钟频率必须小于等于PCLK .

	M032xC/D	M031xB/C/D/E M032x/E	M031xG/I M032xG/I
6.25.5.11 PWM 触发	-	●	●
6.25.5.12 BPWM触发	●	-	●
6.25.5.17 Floating检测功能	●	-	●

表 6.25-1 ADC 特性比较表

6.25.3 框图

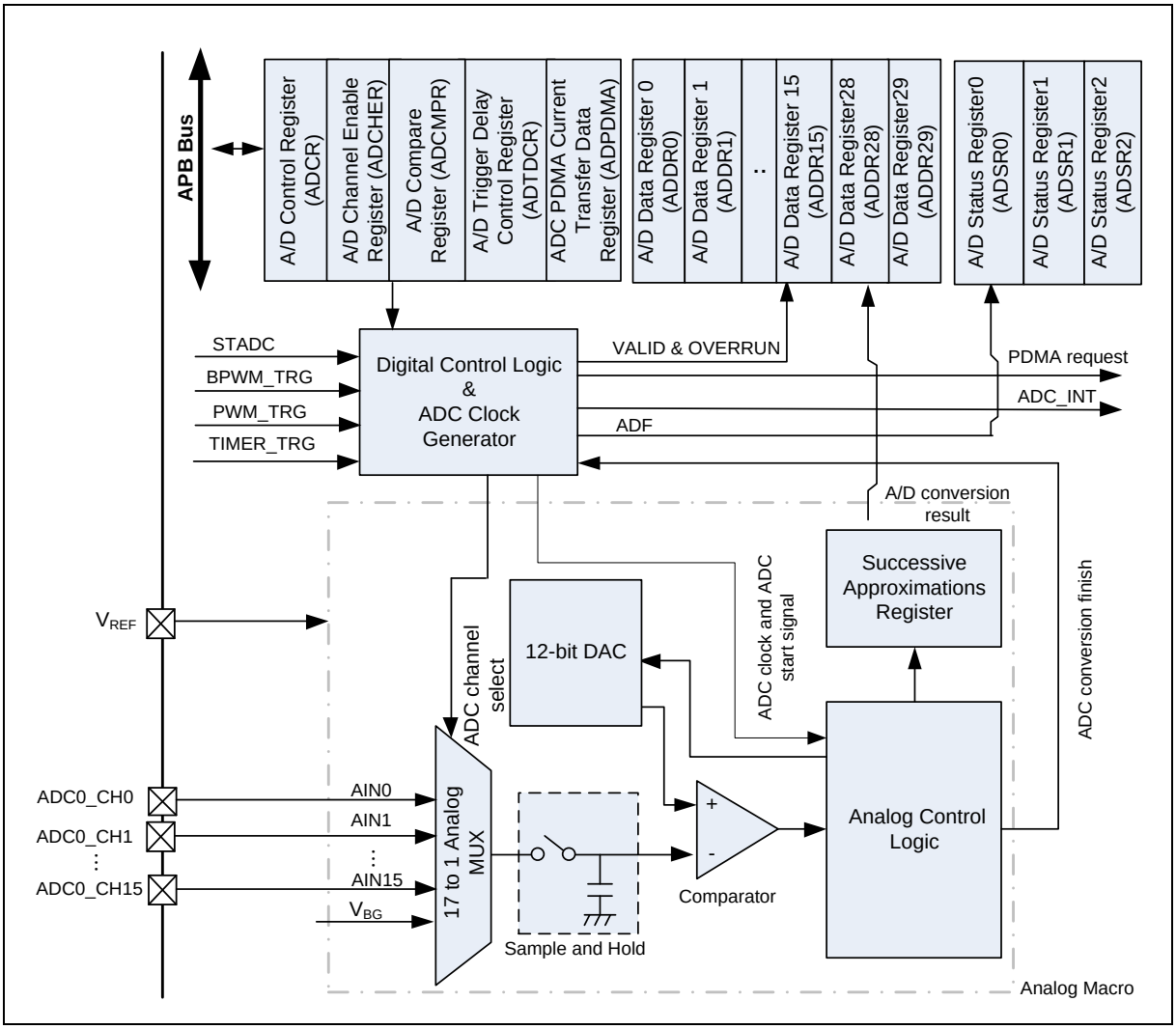


图 6.25-1 AD 控制器框图

6.25.4 基本配置

- 时钟源配置
  - 通过ADCSEL(CLKSEL2[21:20])选择ADC外设时钟源.
  - 通过ADCDIV (CLKDIV0[23:16])选择ADC外设时钟除频系数 .
  - 通过ADCCKEN (CLK_APBCLK0[28])使能ADC外设时钟 .
- 复位配置
  - 通过ADCRST (SYS_IPRST1[28])复位ADC控制器 .
- 管脚配置

组	管脚名称	GPIO	MFP
ADC0	ADC0_CH0	PB.0	MFP1
	ADC0_CH1	PB.1	MFP1
	ADC0_CH2	PB.2	MFP1
	ADC0_CH3	PB.3	MFP1
	ADC0_CH4	PB.4	MFP1
	ADC0_CH5	PB.5	MFP1
	ADC0_CH6	PB.6	MFP1
	ADC0_CH7	PB.7	MFP1
	ADC0_CH8	PB.8	MFP1
	ADC0_CH9	PB.9	MFP1
	ADC0_CH10	PB.10	MFP1
	ADC0_CH11	PB.11	MFP1
	ADC0_CH12	PB.12	MFP1
	ADC0_CH13	PB.13	MFP1
	ADC0_CH14	PB.14	MFP1
	ADC0_CH15	PB.15	MFP1
	ADC0_ST	PF.5	MFP11
		PC.13, PD.12	MFP14
		PC.1, PG.15	MFP15

6.25.5 功能描述

A/D转换器采用逐次逼近转换方式，转换结果为12位数据。 ADC有四种工作模式：单一模式，Burst模式，单周期扫模式和连续模式。当用户需要改变操作模式或模拟输入通道，为了避免误动作，软件必须提前将ADST(ADCR[11])位清0

6.25.5.1 ADC外设时钟产生器

最大采样率为2 MSPS. 通过ADCSEL (CLKSEL2[21:20])ADC有4个时钟源选择，通过8位预分频器和下面的公式可以得到ADC外设时钟频率。

ADC 外设时钟频率 = (ADC 外设时钟源频率) / (ADCDIV+1); 这里 8-位 ADCDIV 是寄存器CLKDIV0[23:16].

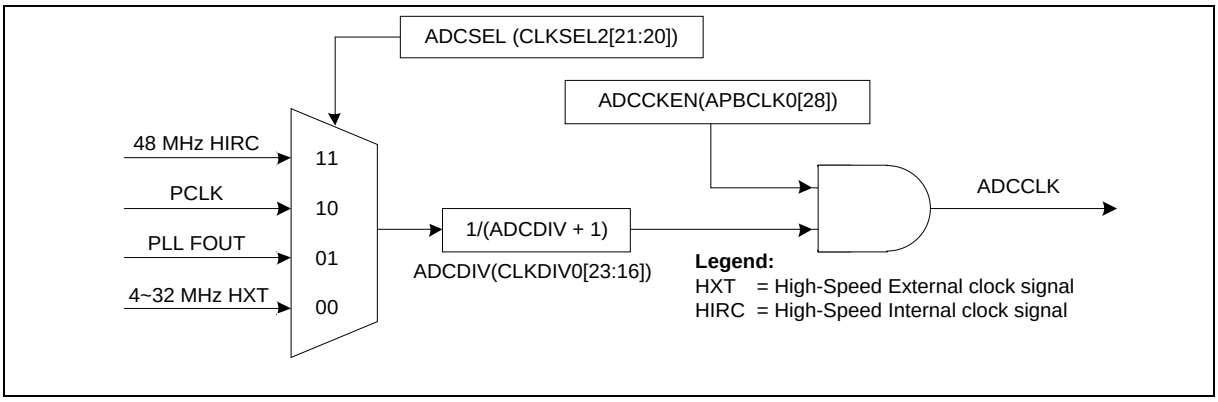


图 6.25-2 ADC 外设时钟控制

6.25.5.2 单次模式

在单一模式, A/D 转换只在指定通道执行一次 。操作流程如下:

1. 当ADCR 寄存器的ADST位被软件置1或者外部触发，A/D 转换开始.
2. 当A/D转换结束，这个结果将存在相应通道的ADC数据寄存器中。
3. A/D转换完成后，ADSR0寄存器的ADF位将被置1 ，如果ADCR寄存器的ADIE位被置1，ADC中断将会产生。
4. 在A/D转换期间ADST位一直保持为1.当A/D转换结束,ADST位自动清除为0，A/D转换器进入空闲状态 。

注1: 在单一模式下，如果软件使能了多个通道，则编号最小的被选中转换其它使能的通道将被忽略。

注2: 在ADC转换完成前ADST位被清0，那么ADC不能完成当前的转换，BUSY为将被立即清0，A/D转换器直接进入空闲状态

单周期模式示例时序图 6.25-3.

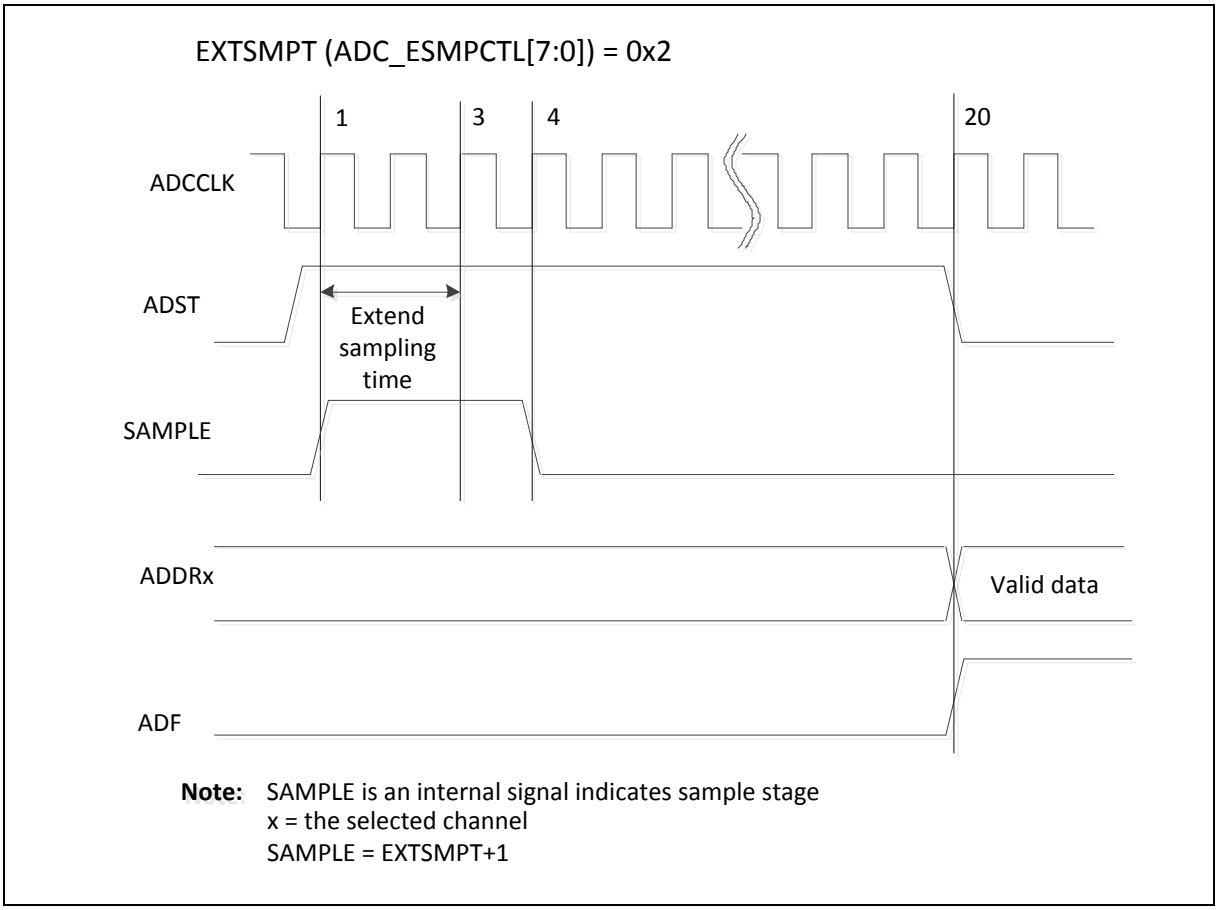


图 6.25-3 单次模式转换时序图

6.25.5.3 Burst模式

在Burst 模式, A/D 转换器连续的采样和转换指定的单一通道并将结果存储到FIFO中（最多8个采样）操作流程如下：

1. 当软件将ADCR寄存器的ADST位置1，或者外部有触发输入，被使能的编号最小的通道开始A/D转换。
2. 当指定通道的A/D转换完成，其结果按顺序传输到FIFO中，只能通过ADC数据寄存器0才可访问
3. 当在FIFO中多于4个采样结果，ADSR0寄存器的ADF位就会置1.如果同时ADCR寄存器的ADIE位被置 1，在A/D转换结束后一个ADC中断将会产生
4. 在ADST位保持1时步骤 2 到 3 一直重复。当ADST位被清0，ADC不能完成当前的转换并且A/D转换器直接进入空闲状态。

Burst 模式的一个示例时序图在图 6.25-4.

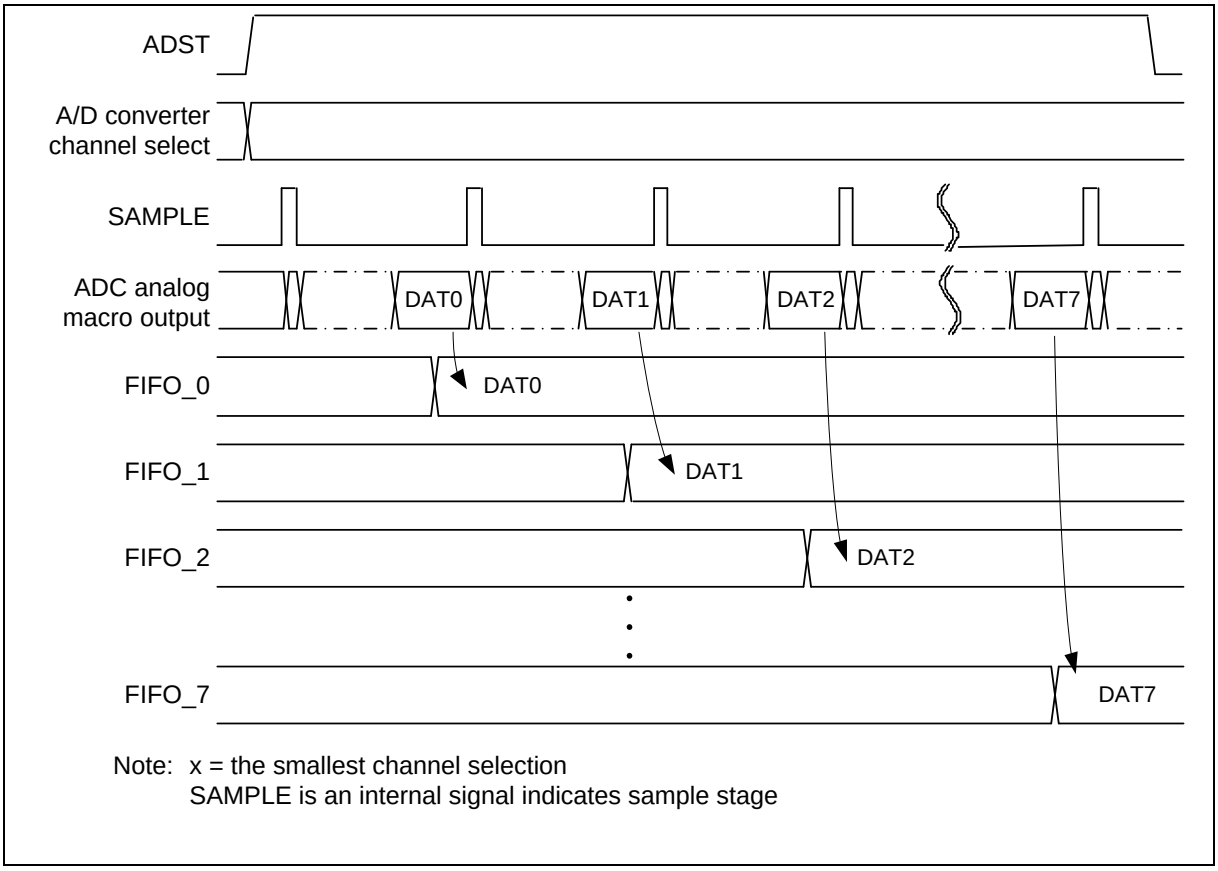


图 6.25-4 Burst 模式转换时序图

注1: 如果在Burst 模式软件使能了超过一个通道，只有编号最小的通道被转换其它通道将忽略。

注2: 用户通过重复读取ADC数据寄存器0（ADDR0）获取转换结果，直到VALIDF（ADSR0[8]）变为0. 例如，如果有4个转换结果再FIFO中，那就需要读取ADC数据寄存器（ADDR0）4次获取所有的转换结果。

6.25.5.4 单周期扫描模式

在单周期扫描模式，A/D转换器按编号从小到大采样和转换所有指定的通道一次。操作流程如下：

1. 当通过软件将ADCR寄存器的ADST位置1或者外部触发输入，被使能的编号最小的通道开始A/D转换。
2. 当被使能的每个通道都完成了A/D转换，其结果会按顺序传输到相应通道的ADC数据寄存器
3. 当使能的所有通道都转换完成，ADSR0寄存器的ADF位将置1，如果ADC中断使能，ADC中断将产生。
4. 在ADC完成了一个周期的转换后，ADST自动清0并且A/D转换器进入空闲状态。如果在所有使能的通道转换完成前ADST位被清0，当前的ADC转换将不能完成，A/D转器直接进入空闲状态

使能通道（0,2,3,7）单周期扫描模式的一个示例时序图如图 6.25-5.

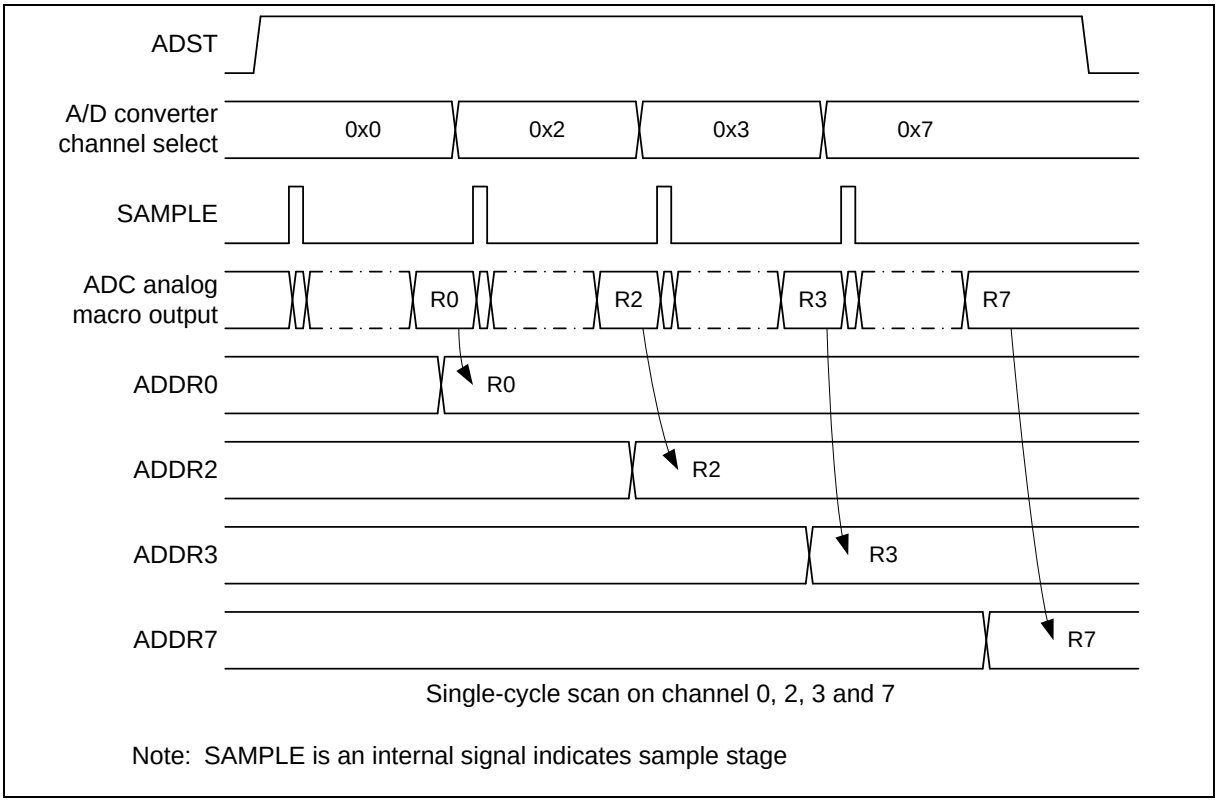


图 6.25-5 使能通道单周期扫描模式时序图

6.25.5.5 A/D 延长采样时间

当 ADC 采样率较高时, 如果模拟电路的输出阻抗很高，导致充电时间过长，模拟输入电压的采样时间可能不够。用户可以对每个采样模块通过写 EXTSMPT (ADC_ESMPCTL [7:0]) 延长采样时间。 延长采样时间的范围是 0 ~255 ADC 时钟

6.25.5.6 连续扫描模式

在连续扫描模式下，A/D转换器循环转换ADCHER 寄存器的CHEN位使能的通道（最多16个通道），操作流程如下：

1. 当ADCR寄存器的ADST位通过软件置1或者外部触发输入，A/D开始转换被使能的编号最小的通道
2. 当A/D转换完成使能的每一个通道，每个通道的转换结果都存储在对应的ADC数据寄存器中。
3. 当A/D转换器按顺序转换完所有使能的通道，ADSR0寄存器的ADF位将被置1.如果ADC中断功能被使能，将产生ADC中断。如果软件没有清零ADST位，被使能的编号最小的通道将再次开始转换。
4. 在ADST位保持为1期间，步骤2~3 将重复进行。当ADST被清0，ADC不能完成当前的转换并且A/D转换器直接进入空闲状态。

使能通道（0,2,3）连续扫描模式的一个示例时序图如图 6.25-6.

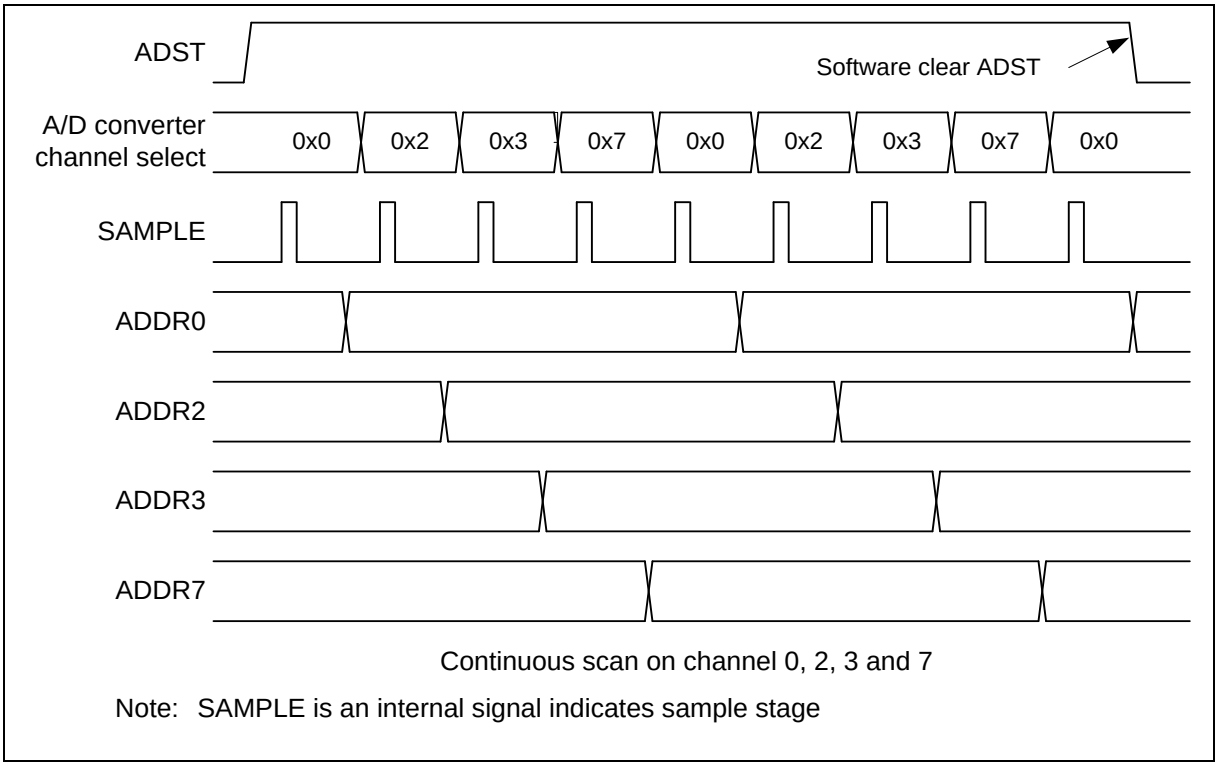


图 6.25-6 使能通道连续扫描模式时序图

6.25.5.7 校准模式

为了减少电子随机噪音的影响，ADC执行校准来获得测量的平均偏移量。然后，在正常操作下，数字模块把校准得到的校准字应用于内部 ADC 电容网络，以消除偏移电压。

在校准模式

1. 设定CALEN(ADC_ADCALR[0]) 位为 1.
2. 当ADCR寄存器的ADST位通过软件置1，校准开始
3. 校准完成后，CALIF标志(ADC_ADCALSTSR[0]) 将被置1
4. 校准模式完成后，模式将自动切换到普通模式.

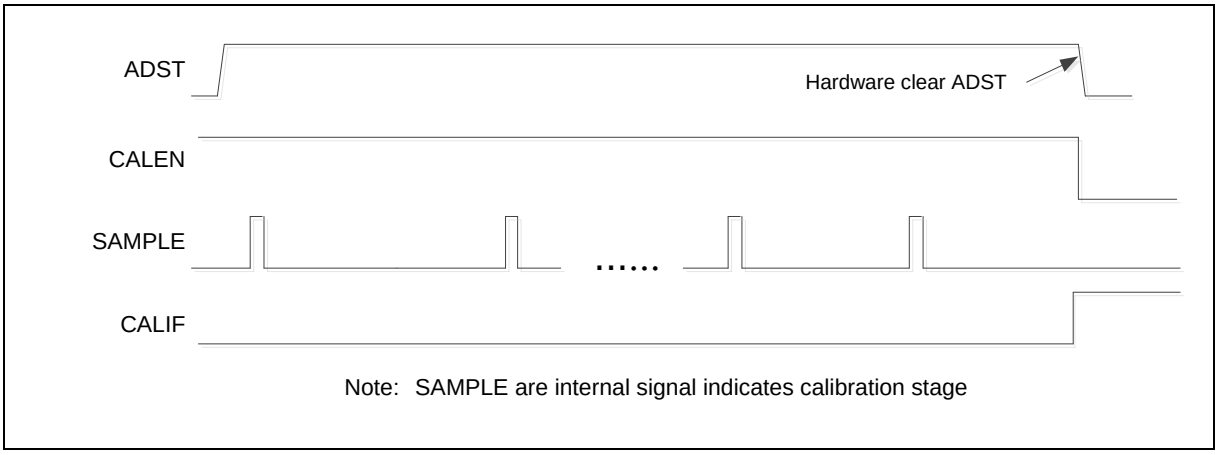


图 6.25-7 16 次平均校准模式

6.25.5.8 *外部触发输入*

在单次扫描模式，可以通过外部引脚触发A/D转换。ADCR 寄存器的TRGEN位设置为1使能ADC外部触发功能，设置TRGS(ADCR[5:4]) 为2'b00选择外部触发输入为STADC引脚。软件可以通过设置TRGCOND(ADCR[7:6]) 选择触发条件为下降/上升沿或低/高电平。如果选择电平触发，STADC引脚必须保持指定状态至少8个PCLK。在第9个PCLK，ADST位将被置1且开始转换。在电平触发模式下，如果外部触发输入保持有效状态，转换将持续进行。只有在外部触发条件消失， 转换才会停止。如果选择边沿触发条件，高和低状态都至少持续4个PLCK。如果脉冲满足不了该条件，将被忽略。

注: 使能ADC外设时钟后，用户须在至少4个PCLK之后方能使能外部触发功能或使能ADC.

6.25.5.9 *内部参考电压*

Band-gap 参考电压(V_{BG})是一个内部的固定参考电压不受外部供电的影响。V_{BG}输出是通过内部连接到ADC通道的多路转接器和模拟比较器(ACMP)的负极输入端，因此AV_{DD}可以通过V_{BG} 的ADC转换结果计算出来

如果AV_{DD}直接接到供电电源，V_{BG}可以用来计算AV_{DD}电压。相比较分压电路使用V_{BG}可以减少外部元件及电流消耗

由于制程的原因每个芯片的V_{BG}稍有不同，因此AV_{DD}的计算会有一点偏差。M031/M032系列在出厂时内置了AV_{DD}=3072mV时V_{BG}的A/D转换结果，用户可以通过这个值及当前V_{BG} A/D转换结果精准计算出AV_{DD}的电压值，相比只是用V_{BG}典型值计算会有更高的精度

对于用电池供电的应用，用户可以根据下面步骤计算出比较精确的电池电压：

1. 使用ISP命令- 读唯一ID码(FMC_ISPCMD = 0x04)读取内置的V_{BG} ADC结果($ADC_{VBG_Built\text{-}In}$，FMC_ISPDAT[11:0])
2. 将V_{BG}作为ADC输入通道(CHEN [29])，然后开始A/D转换获得当前V_{BG}A/D转换结果(ADC_{VBG})
3. 使用$ADC_{VBG_Built\text{-}In}$, ADC_{VBG}及下面公式获得AV_{DD}

   AV_{DD} = 3072 mV * $ADC_{VBG_Built\text{-}In}$ / ADC_{VBG}

例如，如果ADC_{VBG} 是 0x61C (1564)，$ADC_{VBG_Built\text{-}In}$ 是 0x68D (1677)，AV_{DD}可以如下计算：3072 mV * 1677 / 1564 = 3293 mV

6.25.5.10 定时器触发

ADC可以用4个定时器作为触发源，当ADCR寄存器的TRGEN位被置为高使能ADC外部硬件触发功能，TRGS(ADCR[5:4])位为2'b01就选中了外部硬件触发输入源为定时器触发。更多定时器触发情况的详细描述请参阅寄存器TIMER0_CTL ~ TIMER3_CTL

6.25.5.11 PWM 触发

所有的操作模式都可以通过PWM触发A/D转换。当ADCR寄存器的TRGEN位被置为高使能ADC外部硬件触发功能，TRGS(ADCR[5:4])位为2'b11就选中了外部硬件触发输入源为PWM触发。

6.25.5.12 BPWM 触发

所有的操作模式都可以通过BPWM触发A/D转换。当ADCR寄存器的TRGEN位被置为高使能ADC外部硬件触发功能，TRGS(ADCR[5:4])位为2'b10就选中了外部硬件触发输入源为BPWM触发

6.25.5.13 比较模式监控转换结果

M031/M032系列的ADC转换器提供了2个比较寄存器，ADCMPRx(x=0,1),用于监控AD最多支持两路指定通道转换结果。通过软件设定CMPCH (ADCMPRx[7:3])选择要监控的通道。CMPCOND (ADCMPRx[2])位用于设定比较条件。当转换结果小于CMPD (ADCMPRx[27:16])中指定的值，如果CMPCOND位被清0那么内部匹配计数器将加1.当转换结果大于或者等于CMPD (ADCMPRx[27:16])中指定的值，如果CMPCOND位为1，那么内部匹配计数器将加1.当通过CMPCH(ADCMPRx[7:3])指定的通道完成转换，比较动作将被自动触发一次。当比较的结果与设定值相匹配，比较匹配计数器将加1，否则比较匹配计数器将清0.当匹配计数器达到(CMPMATCNT+1)设定的值，那么CMPF0/1 (ADSR0[1]/[2])位将被置1，如果CMPIE (ADCMPRx[1])位被置1那么ADC中断将产生。在扫描模式下软件可以用它来监控外部输入管脚的电压。详细逻辑框图如图 6.25-8.

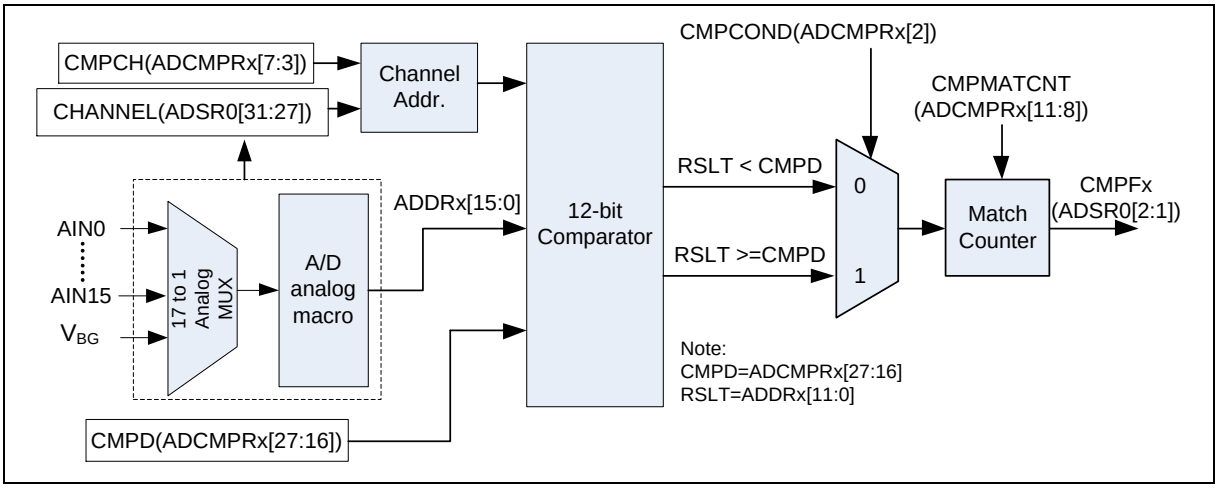


图 6.25-8 A/D 转换结果监控逻辑框图

6.25.5.14 比较窗口模式

ADC控制器支持比较窗口模式。用户能设定CMPWEN (ADCMPR0[15])使能该功能。如果用户使能了该功能，只有当两路转换结果监控逻辑比较条件都匹配，那么CMPF0 (ADSR0[1])将被置1，CMPF1 (ADSR0[2])将总是为0.比较窗口的范围是在CMPD (ADCMPR0[27:16])和CMPD (ADCMPR1[27:16])。

6.25.5.15 PDMA 传输模式

当A/D转换完成，转换结果将被装载到ADDRx(x=0~15, 29)寄存器并且VALID(ADDRx[17])位将被置1.如果PTEN(ADCR[9])位置1，ADC控制器将产生一个PDMA请求。用户能用PDMA去传输转换结果到指定

的内存空间中而不需要CPU的介入。PDMA操作的源地址是固定的在ADPDMA,无论是哪个通道被选中。如果ADC操作模式是burst模式，单周期扫描模式或者连续扫描模式当PDMA正在传输转换结果时，ADC将继续转换下一个选中的通道。用户能通过读ADPDMA寄存器监控当前的PDMA传输数据。如果ADC完成了选中通道的转换但该通道的上一次转换结果还没有被PDMA传输出去，那么相应通道的OVERRUN(ADSR2[31:0])位将被置1，并且上一次的ADC转换结果将被新的ADC转换结果覆盖。PDMA将传输最新一次的选中通道的数据到用户指定的目标地址。

6.25.5.16 中断源

ADC中断有4个中断源。当某个ADC转换模式结束其转换时，A/D转换结束标志ADF(ADSR0[0])将被置1。CMPF0(ADSR0[1]) 和CMPF1(ADSR0[2])是比较功能的比较标志。当转换结果满足ADC_CMP0/1的设定,相应的标志会置1. CALIF(ADCALSTSR[0])是校准功能的标志。当校准完成时该标志将置1.

当某一个标志， ADF, CMPF0 ，CMPF1和CALIF 变为1并且相应的中断使能位（ADCR 寄存器的ADIE，ADCMPR0/1寄存器的CMPIE和ADCAL寄存器的CALIE）被置1，ADC中断就会产生。软件可以清除这些标志来取消中断请求。

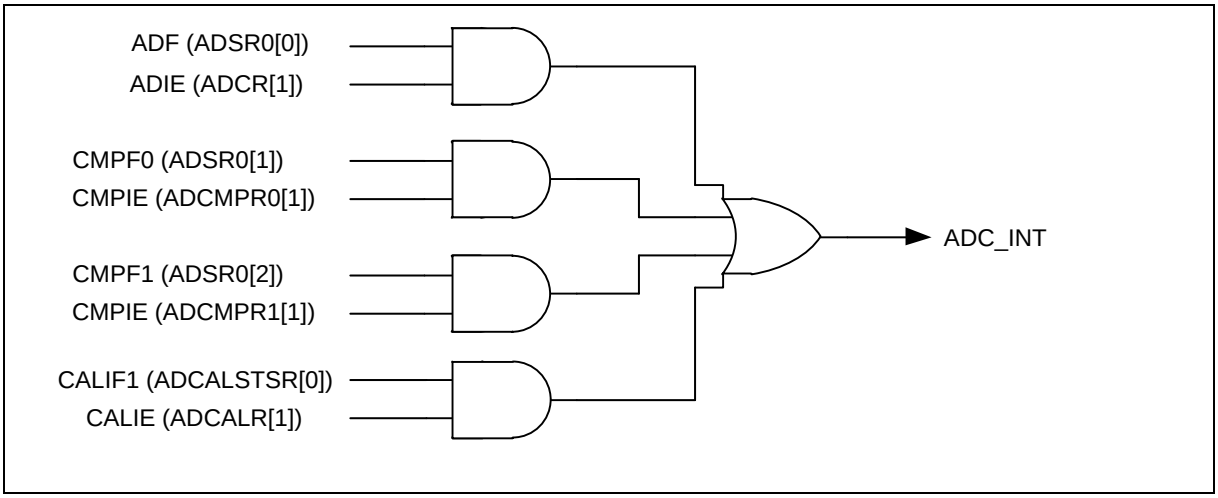


图 6.25-9 A/D 控制器中断

6.25.5.17 浮空检测功能

ADC内部通道可以用来检测管脚连接状态，设置FDETCHEN (ADC_CFDCTL[8])为1打开内部通道，假设V_{REF} 是3.0V：

1. 使能一个外部通道并设置输入电压为1.2V
2. 设置FDETCHEN 及 PRECHEN (ADC_CFDCTL[0]) 为 1.
3. 等待一段时间（根据电阻值）开始转换
4. 如果转换结果接近V_{REF}的值，则检测到此外部通道断开状态

6.25.5.18 差分模式

ADC控制器支持模拟全差分模式，全差分SAR转换输入的差分信号。这种情况下两个动态输入信号必须是互补的或者说是相反的，这时VINP和VINM管脚信号必须相差180°相位，以固定的共模电压为中心，例如VREF/2。全差分ADC有一个输入共模电压（VCM）范围规范，VCM定义为输入信号的平均电压VCM = (VINP + VINM) / 2.，绝大多数全差分输入SAR ADC禁止输入共模电压浮动超过中间值(V_{REF} /2)的10%

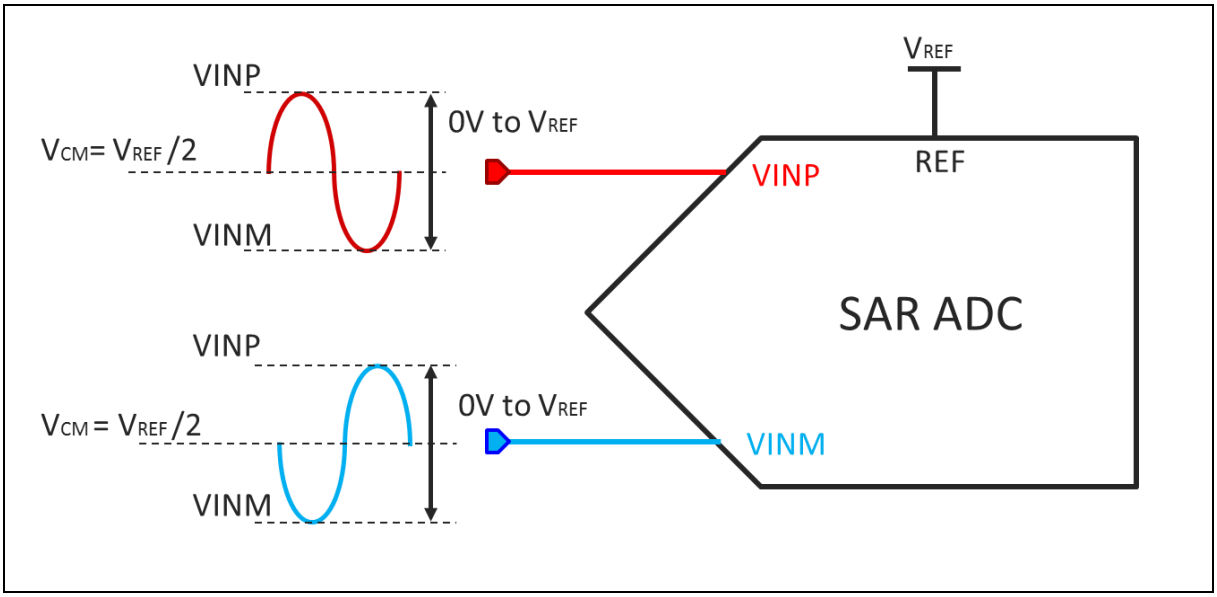


图 1.1-16 全差分 ADC 输入

如果用户使能DIFFEN (ADC_ADCR[10])将使能差分模式，差分输入信号对如表 6.25-2

差分模拟输入通道对	ADC 模拟输入	
	V_{plus}	V_{minus}
0	ADC_CH0	ADC_CH1
1	ADC_CH2	ADC_CH3
2	ADC_CH4	ADC_CH5
3	ADC_CH6	ADC_CH7
4	ADC_CH8	ADC_CH9
5	ADC_CH10	ADC_CH11
6	ADC_CH12	ADC_CH13
7	ADC_CH14	ADC_CH15

表 6.25-2 ADC 差分模式通道选择

差分输入模式下，两个AD通道只需要使能偶数通道CHEN (ADC_ADCHER[31:0])，转换结果将放置在相应使能的通道数据寄存器中，如果DMOF (ADC_ADCR [31]) = 1转换结果将以2的补码格式存放

6.25.6 寄存器映射

R: 只读, **W**: 只写, **R/W**: 读写.

寄存器	偏移	R/W	描述	复位值
ADC 基地址: **ADC_BA = 0x4004_3000**				
ADC_ADDR0	ADC_BA+0x00	R	ADC 数据寄存器 0	0x0000_0000
ADC_ADDR1	ADC_BA+0x04	R	ADC数据寄存器 1	0x0000_0000
ADC_ADDR2	ADC_BA+0x08	R	ADC数据寄存器 2	0x0000_0000
ADC_ADDR3	ADC_BA+0x0C	R	ADC数据寄存器 3	0x0000_0000
ADC_ADDR4	ADC_BA+0x10	R	ADC数据寄存器 4	0x0000_0000
ADC_ADDR5	ADC_BA+0x14	R	ADC数据寄存器 5	0x0000_0000
ADC_ADDR6	ADC_BA+0x18	R	ADC数据寄存器 6	0x0000_0000
ADC_ADDR7	ADC_BA+0x1C	R	ADC数据寄存器 7	0x0000_0000
ADC_ADDR8	ADC_BA+0x20	R	ADC数据寄存器 8	0x0000_0000
ADC_ADDR9	ADC_BA+0x24	R	ADC数据寄存器 9	0x0000_0000
ADC_ADDR10	ADC_BA+0x28	R	ADC数据寄存器 10	0x0000_0000
ADC_ADDR11	ADC_BA+0x2C	R	ADC数据寄存器 11	0x0000_0000
ADC_ADDR12	ADC_BA+0x30	R	ADC数据寄存器 12	0x0000_0000
ADC_ADDR13	ADC_BA+0x34	R	ADC数据寄存器 13	0x0000_0000
ADC_ADDR14	ADC_BA+0x38	R	ADC数据寄存器 14	0x0000_0000
ADC_ADDR15	ADC_BA+0x3C	R	ADC数据寄存器 15	0x0000_0000
ADC_ADDR29	ADC_BA+0x74	R	ADC数据寄存器 29	0x0000_0000
ADC_ADCR	ADC_BA+0x80	R/W	ADC 控制寄存器	0x0000_0000
ADC_ADCHER	ADC_BA+0x84	R/W	ADC 通道使能寄存器	0x0000_0000
ADC_ADCMPR0	ADC_BA+0x88	R/W	ADC 比较寄存器 0	0x0000_0000
ADC_ADCMPR1	ADC_BA+0x8C	R/W	ADC 比较寄存器 1	0x0000_0000
ADC_ADSR0	ADC_BA+0x90	R/W	ADC 状态寄存器 0	0x0000_0000
ADC_ADSR1	ADC_BA+0x94	R	ADC状态寄存器 1	0x0000_0000
ADC_ADSR2	ADC_BA+0x98	R	ADC状态寄存器 2	0x0000_0000
ADC_ESMPCTL	ADC_BA+0xA0	R/W	ADC 延长采样时间控制寄存器	0x0000_0000
ADC_CFDCTL	ADC_BA+0xA4	R/W	ADC通道浮空检测控制寄存器	0x0000_0000

ADC_ADPDMA	ADC_BA+0x100	R	ADC PDMA当前传输数据寄存器	0x0000_0000
ADC_ADCALR	ADC_BA+0x180	R/W	ADC 校准模式寄存器	0x0000_005C
ADC_ADCALSTSR	ADC_BA+0x184	R/W	ADC 校准状态寄存器	0x0000_0000

6.25.7 寄存器描述

ADC 数据寄存器 (ADC_ADDRx x = 0~15, 29)

寄存器	偏移	R/W	描述	复位值
ADC_ADDR0	ADC_BA+0x00	R	ADC 数据寄存器 0	0x0000_0000
ADC_ADDR1	ADC_BA+0x04	R	ADC数据寄存器 1	0x0000_0000
ADC_ADDR2	ADC_BA+0x08	R	ADC数据寄存器 2	0x0000_0000
ADC_ADDR3	ADC_BA+0x0C	R	ADC数据寄存器3	0x0000_0000
ADC_ADDR4	ADC_BA+0x10	R	ADC数据寄存器 4	0x0000_0000
ADC_ADDR5	ADC_BA+0x14	R	ADC数据寄存器 5	0x0000_0000
ADC_ADDR6	ADC_BA+0x18	R	ADC数据寄存器 6	0x0000_0000
ADC_ADDR7	ADC_BA+0x1C	R	ADC数据寄存器 7	0x0000_0000
ADC_ADDR8	ADC_BA+0x20	R	ADC数据寄存器 8	0x0000_0000
ADC_ADDR9	ADC_BA+0x24	R	ADC数据寄存器 9	0x0000_0000
ADC_ADDR10	ADC_BA+0x28	R	ADC数据寄存器 10	0x0000_0000
ADC_ADDR11	ADC_BA+0x2C	R	ADC数据寄存器 11	0x0000_0000
ADC_ADDR12	ADC_BA+0x30	R	ADC数据寄存器 12	0x0000_0000
ADC_ADDR13	ADC_BA+0x34	R	ADC数据寄存器 13	0x0000_0000
ADC_ADDR14	ADC_BA+0x38	R	ADC数据寄存器 14	0x0000_0000
ADC_ADDR15	ADC_BA+0x3C	R	ADC数据寄存器 15	0x0000_0000
ADC_ADDR29	ADC_BA+0x74	R	ADC数据寄存器 29	0x0000_0000

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

位	描述	
[31:18]	保留	保留.
[17]	**VALID**	**有效标志 (只读)** 当相应通道转换完成该位置1. 读取ADDR寄存器转换结果后该位硬件清0. 0 = RSLT 中的数据无效. 1 = RSLT 中的数据有效.
[16]	**OVERRUN**	**覆盖标志 (只读)** 如果新的转换结果数据加载到RSLT寄存器而旧的转换数据还没有读取，OVERRUN位将置1.读取ADD寄存器转换结果后硬件清除该位 0 = RSLT中的数据没有覆盖. 1 = RSLT 中的数据被覆盖.
[15:0]	**RSLT**	**A/D 转换结果 (只读)** 该域包含ADC转换结果.

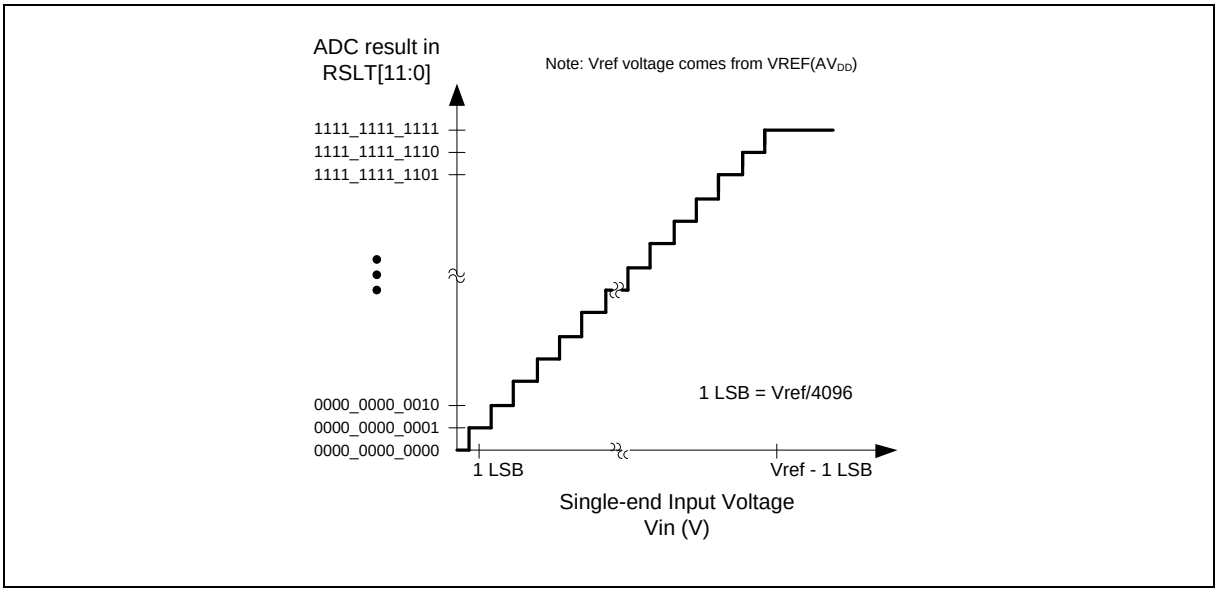


图 6.25-10 ADC 单端输入转换结果映射框图

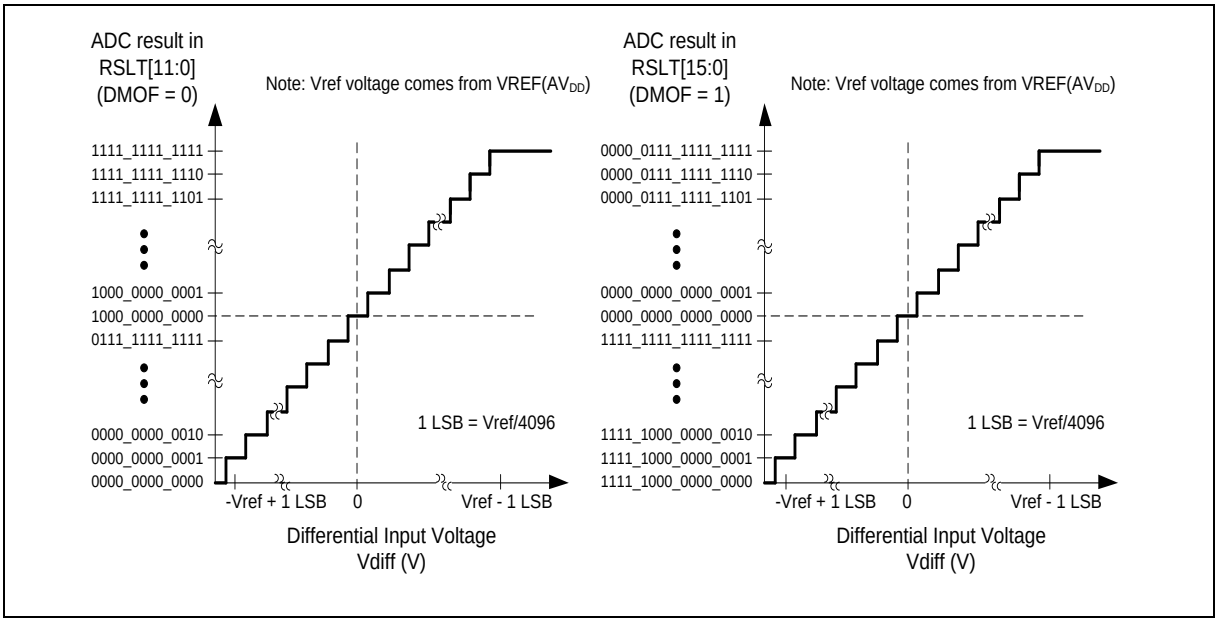


图 6.25-11 ADC 差分输入转换结果映射框图

ADC 控制寄存器 (ADC_ADCR)

寄存器	偏移	R/W	描述	复位值
ADC_ADCR	ADC_BA+0x80	R/W	ADC 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
DMOF	保留						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留			**RESET**	**ADST**	**DIFFEN**	**PTEN**	**TRGEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGCOND		**TRGS**		**ADMD**		**ADIE**	**ADEN**

位	描述	
[31]	**DMOF**	**差分输入模式输入格式** 如果用户使能差分输入模式，转换结果可以用标准二进制格式（无符号格式）或者2进制补码格式（有符合格式） 0 = A/D 转换结果用无符号格式（标准二进制）填入ADDRx 寄存器的RSLT中。 1 = A/D 转换结果用2二进制补码格式填入ADDRx 寄存器的RSLT中。
[30:13]	保留	保留.
[12]	**RESET**	**ADC 复位 (写保护)** 如果用户写该位，ADC模拟模块将复位，模块中校准数据将删除，但ADC控制器寄存器中的值将保留 **注:** 该位硬件清0.
[11]	**ADST**	**A/D 转换开始或者校准开始** ADST 位置1有下列4种方式：软件设定，外部引脚STADC， PWM 触发和定时器触发 。单周期扫描模式，单次模式下和校准模式，在转换结束后， ADST 将被硬件自动清 0 。在连续扫描模式和Burst 模式， A/D 转换将一直进行直到软件向该位写0或芯片复位. 0 = 转换停止，A/D转换器进入空闲状态 1 = 转换开始或校准开始 **注1:** 当ADST由1变为0，ADC模块将复位到初始化状态。在模块复位到初始化状态后，用户需要等待2个ADC时钟再置1该位开始下一次转换。 **注2:** 只有当CALEN (ADC_ADCALR[0]) = 1，校准才开始 。
[10]	**DIFFEN**	**差分输入模式控制** 差分输入电压 (V_{diff}) = V_{plus} - V_{minus}. V_{plus} 和 V_{minus} 的关系是V_{plus} + V_{minus} = V_{ref}. 差分输入配对通道x的Vplus来自ADC0_CHy引脚， Vminus来自ADC0_CHz引脚，x = 0,1..7，y = 2 * x，z = y + 1.

		0 = 单端模拟输入模式. 1 = 差分模拟输入模式. **注:** 在差分输入模式下，只需要在ADCHER寄存器中使能两个相应通道的偶数通道。转换结果将被放置到使能通道的相应数据寄存器.
[9]	**PTEN**	**PDMA 传输使能位** 当A/D 转换完成是，转换的结果被存放到ADDR0~15, ADDR29，软件可使能该位以便产生PDMA 数据传输请求. 0 = PDMA 数据传输禁用 1 = PDMA 传输ADDR0~15, ADDR29中数据使能 **注:** 当 PTEN=1,软件必须置ADIE=0 禁用中断
[8]	**TRGEN**	**外部触发使能位** 使能或禁止外部STADC 引脚，PWM，BPWM和定时器触发A/D转换。如果使能外部触发，则ADST位通过选中的硬件触发源来置1 0 = 禁止外部触发 1 = 使能外部触发. **注:** ADC外部触发功能只有在单周期扫描模式下才可以使用.
[7:6]	**TRGCOND**	**外部触发条件** 该2位决定外部引脚STADC触发条件为电平或边沿触发。该信号必须保持至少8 PCLKs 的稳定状态用于电平触发， 4 PCLKs 的高和低状态用于边沿触发. 00 =低电平. 01 = 高电平. 10 = 下降沿. 11 = 上升沿.
[5:4]	**TRGS**	**硬件触发源** 00 = A/D 转换由外部STADC管脚触发. 01 = 定时器0 ~ 定时器3 溢出脉冲触发. 10 = A/D 转换由BPWM触发 11 = A/D 转换由PWM触发. **注:**改变TRGS前，软件应该清TRGEN和 ADST位为0.
[3:2]	**ADMD**	**A/D 转换器操作模式控制** 00 = 单次转换. 01 = Burst 转换 . 10 = 单周期扫描. 11 = 连续扫描. **注1:** 当改变操作模式是，软件应首先清除 ADST 位. **注2:**在Burst 模式, A/D 转换结果数据总是在数据寄存器 0中.
[1]	**ADIE**	**A/D 中断使能位** 当ADIE位被设为 1，A/D转换结束将产生中断. 0 = 禁止 A/D中断. 1 = 使能A/D中断.
[0]	**ADEN**	**A/D 转换器使能位** 0 = 禁止A/D 转换. 1 = 使能 A/D转换. **注:**开始A/D 转换功能时，该位需置1。该位为 0将关闭 A/D转换器模拟电路的电源.

ADC 通道使能寄存器 (ADC_ADCHER)

寄存器	偏移	R/W	描述	复位值
ADC_ADCHER	ADC_BA+0x84	R/W	ADC 通道使能寄存器	0x0000_0000

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

位	描述	
[31:0]	**CHEN**	**模拟输入通道使能控制** 设置ADCHER [15:0] 相应位为1，使能相应的模拟输入通道 15 ~ 0。如果DIFFEN 位设为1，则只需要使能偶数通道. 此外, 设置ADCHER [29]使能内部band-gap电压通道，其它位保留. 0 = 禁用. 1 = 使能. **注 1：** 当选择内部 band-gap电压 (CHEN[29])作为模拟输入时，最大采样率将低于300k SPS

ADC 比较寄存器0/1 (ADC_ADCMPR0/1)

寄存器	偏移	R/W	描述	复位值
ADC_ADCMPR0	ADC_BA+0x88	R/W	ADC 比较寄存器 0	0x0000_0000
ADC_ADCMPR1	ADC_BA+0x8C	R/W	ADC 比较寄存器 1	0x0000_0000

31	30	29	28	27	26	25	24
保留				CMPD			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPWEN	保留			CMPMATCNT			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPCH					CMPCOND	CMPIE	CMPEN

位	描述	
[31:28]	保留	保留.
[27:16]	**CMPD**	**比较数值** 此 12位数值将和指定通道的转换结果相比较. **注:** CMPD 应以无符号格式的存储 (直接二进制格式).
[15]	**CMPWEN**	**比较窗口模式使能位** 0 = 比较窗口模式禁用. 1 = 比较窗口模式使能. **注:** 该位仅ADCMPR0寄存器中定义 .
[14:12]	保留	保留.
[11:8]	**CMPMATCNT**	**比较匹配计数** 当指定A/D 通道转换结果和比较条件CMPCOND相匹配时，内部匹配计数器将加1，否则 内部匹配计数器将清 0。当内部计数器的值达到设定值 (CMPMATCNT +1)时，CMPFx 位将被置位
[7:3]	**CMPCH**	**比较通道选择** 00000 = 选择通道0转换结果用于比较 00001 =选择通道1转换结果用于比较. 00010 =选择通道2转换结果用于比较. 00011 =选择通道3转换结果用于比较. 00100 =选择通道4转换结果用于比较. 00101 =选择通道5转换结果用于比较. 00110 =选择通道6转换结果用于比较. 00111 =选择通道7转换结果用于比较. 01000 =选择通道8转换结果用于比较.

		01001 =选择通道9转换结果用于比较. 01010 =选择通道10转换结果用于比较. 01011 =选择通道11转换结果用于比较. 01100 =选择通道12转换结果用于比较. 01101 =选择通道13转换结果用于比较. 01110 =选择通道14转换结果用于比较. 01111 =选择通道15转换结果用于比较. 11100 =浮空检测通道转换结果用于比较 11101 =选择band-gap电压转换结果用于比较 其它 = 保留.
[2]	**CMPCOND**	**比较条件** 0 = 设置比较条件为当12-位 A/D 转换结果小于12-位 CMPD，内部匹配计数器加1. 1 = 设置比较条件为当12-位 A/D 转换结果大于或等于12-位 CMPD ，内部匹配计数器加1。 **注:** 当内部匹配计数器的值达到 (CMPMATCNT +1)， CMPFx 将置位
[1]	**CMPIE**	**比较中断使能位** 如果使能比较功能且比较条件满足CMPCOND 和 CMPMATCNT的设定， CMPFx 位置位，同时，如果CMPIE置 1，将产生中断. 0 = 比较功能中断禁用 . 1 = 比较功能中断使能.
[0]	**CMPEN**	**比较使能位** 若该位置 1，当A/D转换器完成指定通道的转换并将数据存到ADDR寄存器时，将自动进行CMPD(ADCMPRx[27:16])与该转换值的比较. 0 = 比较功能禁用. 1 = 比较功能使能.

ADC 状态寄存器 0 (ADC_ADSR0)

寄存器	偏移	R/W	描述	复位值
ADC_ADSR0	ADC_BA+0x90	R/W	ADC 状态寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
CHANNEL					保留		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							**OVERRUNF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							**VALIDF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BUSY	保留				**CMPF1**	**CMPF0**	**ADF**

位	描述	
[31:27]	**CHANNEL**	**当前转换通道 (只读)** 当 BUSY=1，此位域表示当前正在转换中的通道编号。当 BUSY=0，此位域表示下一个将进行转换的通道编号
[26:17]	保留	保留.
[16]	**OVERRUNF**	**溢出标志 (只读)** 如果任何一个(ADDRx[16]) 的OVERRUN位置 位，此标志位置 1。 **注:** 当ADC 是 burst 模式且 FIFO溢出时， 此标志位将置 1.
[15:9]	保留	保留.
[8]	**VALIDF**	**数据有效标志 (只读)** 如果任何一个(ADDRx[17]) 的VALID 位置位，此标志位置 1。 **注:** 当 ADC 是 burst 模式且任一转换结果有效，此标志位将置1
[7]	**BUSY**	**BUSY/IDLE (只读)** 该位是 ADCR寄存器的ADST位的镜像。 0 = A/D 转换器空闲状态. 1 = A/D 转换器忙碌.
[6:3]	保留	保留.
[2]	**CMPF1**	**比较标志 1** 当 A/D 转换器所选择的的通道转换结果和ADCMPR1寄存器所设定的比较条件相匹配，该位置1；写 1清除此标志位 0 = ADDR转换结果和 ADCMPR1寄存器设定的比较条件不匹配 1 = ADDR转换结果和 ADCMPR1寄存器设定的比较条件匹配
[1]	**CMPF0**	**比较标志 0** 当 A/D 转换器所选择的的通道转换结果和ADCMPR0寄存器所设定的比较条件相匹配，该位置1；写 1清除此标志位 0 = ADDR转换结果和 ADCMPR0寄存器设定的比较条件不匹配 1 = ADDR转换结果和 ADCMPR0寄存器设定的比较条件匹配

[0]	**ADF**	**A/D 转换结束标志** 用于指示A/D 转换已结束标志位。软件写1 清除该标志. 以下三种条件ADF置1: 1. 单次模式下， A/D转换结束. 2. 单周期扫描模式和连续扫描模式下，在所有指定通道的A/D转换结束. 3. Burst 模式下，FIFO中有超过4 个采样数据.

ADC 状态寄存器 1 (ADC_ADSR1)

寄存器	偏移	R/W	描述	复位值
ADC_ADSR1	ADC_BA+0x94	R	ADC 状态寄存器1	0x0000_0000

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

位	描述	
[31:0]	**VALID**	**数据有效标志位 （只读）** VALID[29, 15:0] 是ADDR29[17], ADDR15[17]~ ADDR0[17]中VALID位的镜像。其它位保留. **注:** 当 ADC是 burst 模式且任一转换结果有效，VALID[29, 15:0]位将置1.

ADC 状态寄存器2 (ADC_ADSR2)

寄存器	偏移	R/W	描述	复位值
ADC_ADSR2	ADC_BA+0x98	R	ADC 状态寄存器2	0x0000_0000

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

位	描述	
[31:0]	**OVERRUN**	**溢出标志 (只读)** OVERRUN[29, 15:0] 是ADDR29[16], ADDR15[16] ~ ADDR0[16] 中OVERRUN位的镜像。其它位保留. **注:** 当 ADC 是 burst 模式且 FIFO 溢出，, OVERRUN[29, 15:0] 位将置 1.

ADC 延长采样时间控制寄存器 (ADC_ESMPCTL)

寄存器	偏移	R/W	描述	复位值
ADC_ESMPCTL	ADC_BA+0xA0	R/W	ADC延长采样时间控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**23**	**22**	**19**	**18**	**17**	**16**
保留							
15	**14**	**15**	**14**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EXTSMPT							

位	描述	
[31:8]	保留	保留.
[7:0]	**EXTSMPT**	**ADC 采样时间延长** 当ADC 转换在高转换速率的情况下，如果输入通道的负载比较高，那么模拟输入电压的采样时间有可能不够，用户可以在触发源到来之后延长ADC采样时间以获得足够的采样时间。 延长的采样时间范围位0~255个ADC 时钟

ADC 通道浮空检测控制寄存器(ADC_CFDCTL)

寄存器	偏移	R/W	描述	复位值
ADC_CFDCTL	ADC_BA+0xA4	R/W	ADC 通道浮空检测控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**23**	**22**	**19**	**18**	**17**	**16**
保留							
15	**14**	**15**	**14**	**11**	**10**	**9**	**8**
保留							**FDETCHEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留						**DISCHEN**	**PRECHEN**

位	描述	
[31:9]	保留	保留.
[8]	**FDETCHEN**	浮空检测通道使能位 0 = 浮空检测通道禁止 1 = 浮空检测通道使能 **注:** 如果FDETCHEN使能，内部通道一直打开
[7:2]	保留	保留.
[1]	**DISCHEN**	放电使能 0 = 通道放电禁止 1 = 通道放电使能 **注:** 如果PRECHEN 及 DISCHEN同时使能，模拟输入电压是1/2 V_{REF}
[0]	**PRECHEN**	预充电使能 0 = 通道预充电禁止 1 = 通道预充电使能 **注:**如果PRECHEN 及 DISCHEN同时使能，模拟输入电压是1/2 V_{REF}

ADC PDMA 当前传输数据寄存器 (ADC_ADPDMA)

寄存器	偏移	R/W	描述	复位值
ADC_ADPDMA	ADC_BA+0x100	R	ADC PDMA 当前传输数据寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**23**	**22**	**19**	**18**	**17**	**16**
保留						**CURDAT**	
15	**14**	**15**	**14**	**11**	**10**	**9**	**8**
CURDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURDAT							

位	描述	
[31:18]	保留	保留.
[17:0]	**CURDAT**	**ADC PDMA 当前传输数据寄存器 (只读)** PDMA 传输，读该寄存器可以监测当前PDMA传输数据. 当前 PDMA传输的数据可以是 ADDR0 ~ ADDR15 和 ADDR29中的内容.

ADC 校准模式寄存器 (ADC_ADCALR)

寄存器	偏移	R/W	描述	复位值
ADC_ADCALR	ADC_BA+0x180	R/W	ADC 校准模式寄存器	0x0000_005C

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

位	描述	
[31:2]	保留	保留.
[1]	**CALIE**	**校准中断使能位** 如果校准功能使能并且校准完成，CALIF位将被置位，同时，如果CALIE位被置1，校准中断将产生。 0 = 校准功能中断禁用 . 1 = 校准功能中断使能.
[0]	**CALEN**	**校准功能使能位** 0 = 校准功能禁用. 1.= 校准功能功能使能. **注:** 如果芯片下电，校准功能将再次执行

ADC 校准状态寄存器 (ADC_ADCALSTSR)

寄存器	偏移	R/W	描述	复位值
ADC_ADCALSTSR	ADC_BA+0x184	R/W	ADC 校准状态寄存器	0x0000_0000

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

位	描述	
[31:1]	保留	保留
[0]	**CALIF**	**校准完成中断标志** 如果校准完成该标志将置1，该位写1清0

6.26 模拟比较器控制器(ACMP)

6.26.1 概述

该芯片提供两个比较器。当正端输入大于负端输入时，比较器输出逻辑1，否则输出0。当比较器输出值改变时，可以配置每个比较器发生中断

6.26.2 特性

- 模拟输出电压范围: 0 ~ AV_{DD} (AV_{DD} 管脚电压)
- 支持迟滞功能
- 支持唤醒功能
- 正端和负端输入源可选
- ACMP0 支持 :
  - 正端输入源有4个多路复用I/0管脚:
    - ACMP0_P0, ACMP0_P1, ACMP0_P2, 或 ACMP0_P3
  - 3 个负端输出源:
    - ACMP0_N
    - 比较参考电压 (CRV)
    - 内部band-gap 电压 (VBG)
- ACMP1 支持
  - 正端输入源有4个多路复用I/0管脚:
    - ACMP1_P0, ACMP1_P1, ACMP1_P2, 或ACMP1_P3
  - 3个负端输出源:
    - ACMP1_N
    - 比较参考电压(CRV)
    - 内部band-gap 电压 (VBG)
- 所有比较器共享一个ACMP中断向量
- 当比较结果改变时产生中断（中断条件可编程）
- 支持刹车事件触发和PWM循环控制
- 支持窗口比较模式和窗口锁存模式
- 支持校准功能

章节	子章节	M031xB/C/D/E M032xB/C/D/E	M031xG/I M032xG/I
功能描述	6.26.5.7 校准功能	-/-/-/●	●

表 6.26-1 不同芯片校准功能特性比较表

6.26.3 框图

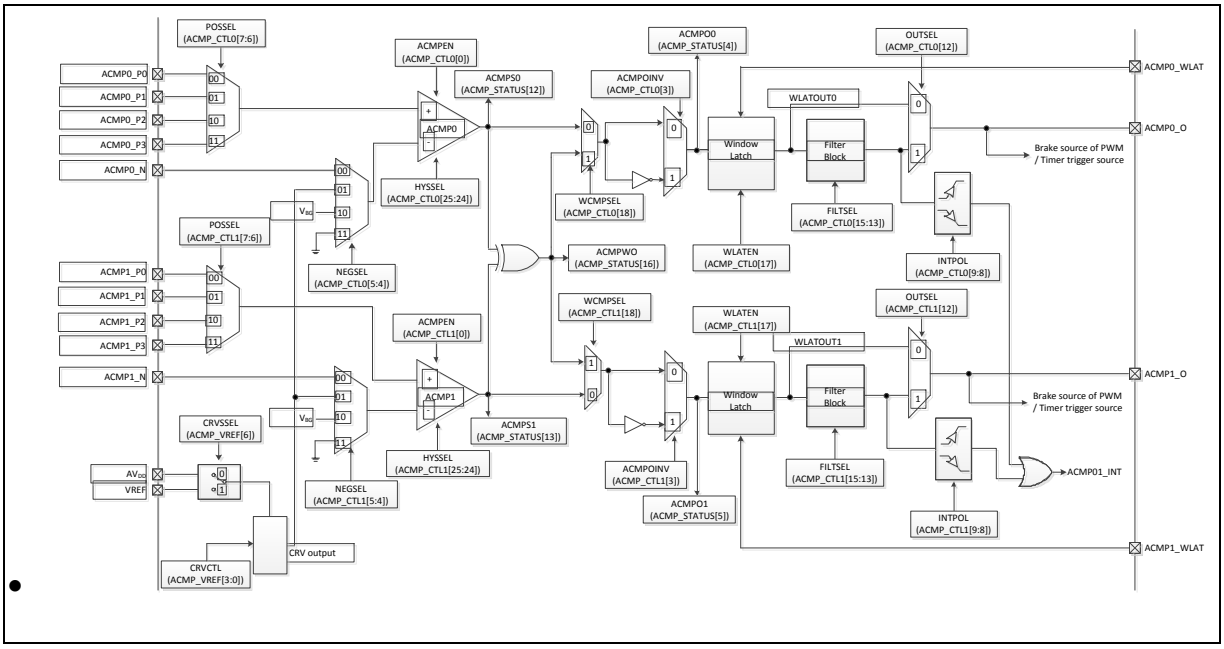


图 6.26-1 模拟比较器框图

6.26.4 基本配置

6.26.4.1 ACMP0 基本配置

- 时钟源配置
  - 通过ACMP01CKEN (CLK_APBCLK0[7])使能ACMP0外设时钟.
- 复位配置
  - 通过ACMP01RST (SYS_IPRST1[7])复位ACMP0控制器.
- 管脚配置

组	管脚名	GPIO	MFP
ACMP0	ACMP0_N	PB.3	MFP1
	ACMP0_O	PC.1, PC.12	MFP14
		PB.7	MFP15
	ACMP0_P0	PA.11	MFP1
	ACMP0_P1	PB.2	MFP1
	ACMP0_P2	PB.12	MFP1
	ACMP0_P3	PB.13	MFP1
	ACMP0_WLAT	PA.7	MFP13

6.26.4.2 ACMP1 基本配置

- 时钟源配置
  - 通过ACMP01CKEN (CLK_APBCLK0[7])使能ACMP1外设时钟 .
- 复位配置
  - 通过ACMP01RST (SYS_IPRST1[7])复位ACMP1控制器.
- 管脚配置

组	管脚名	GPIO	MFP
ACMP1	ACMP1_N	PB.5	MFP1
	ACMP1_O	PC.0, PC.11	MFP14
		PB.6	MFP15
	ACMP1_P0	PA.10	MFP1
	ACMP1_P1	PB.4	MFP1
	ACMP1_P2	PB.12	MFP1
	ACMP1_P3	PB.13	MFP1
	ACMP1_WLAT	PA.6	MFP13

6.26.5 功能描述

6.26.5.1 迟滞功能

模拟比较器支持迟滞功能，使比较器有稳定的输出跳变(参考图 6.26-2) 。如果比较器输出 0，比较器直到正端比负端高于一个门限电压后才会输出 1. 同样的，如果比较器输出 1 ，直到正端比负端低于一个门限电压后才会输出0。

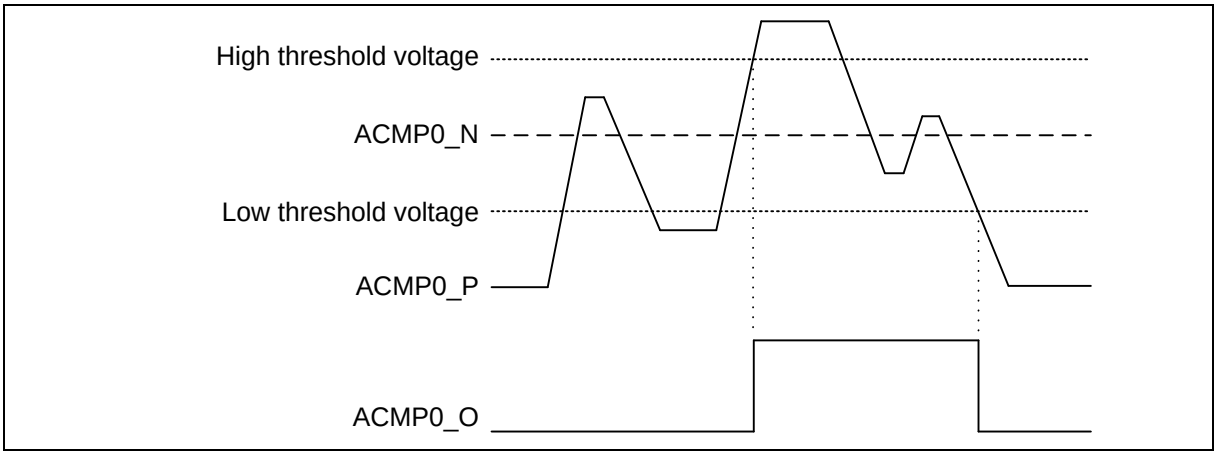


图 6.26-2 ACMP0 比较器迟滞功能

6.26.5.2 窗口锁定功能

图 6.26-3展示了比较器在窗口锁存模式的操作。通过置WLATEN (ACMP_CTL0/1[17])为1使能窗口锁存模式。当窗口锁存模式使能，ACMP0/1_WLAT管脚被用于控制WLATOUT0/1输出。当ACMP0/1_WLAT为高，ACMPO0/1会输出到WLATOUT0/1。当ACMP0/1_WLAT为低，WLATOUT0/1

将保持之前的状态不变。

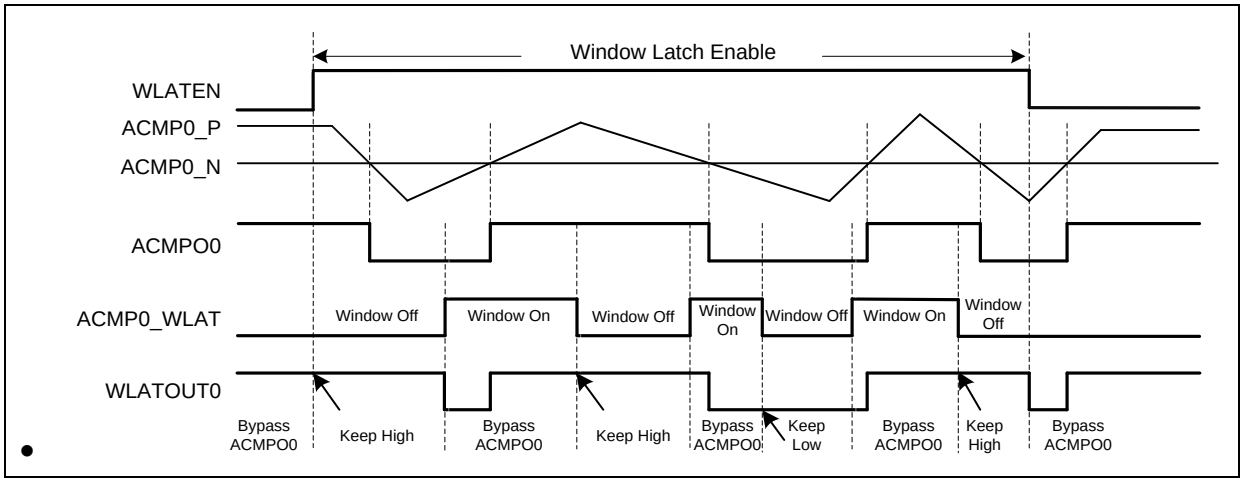


图 6.26-3 窗口锁定模式

6.26.5.3 滤波器功能

模拟比较器带有滤波功能以避免比较器输出不稳定的状态。通过设置FILTSEL (ACMP_CTL0[15:13], ACMP_CTL1[15:13])，比较器输出将通过连续的PCLKs采样，经过较长的采样时钟后比较器的输出就会更加稳定，但同时灵敏度就会降低。

图 6.26-4展示了一个ACMP0经过FILTSEL = 3 (4 PCLK)滤波功能的示例。在这个示例中，比较结果通过PCLK采样。在输出到ACMPO0之前所有的结果都必须保持4个PCLK时钟。如果比较中的结果小于4个PCLK时钟就会被滤掉。

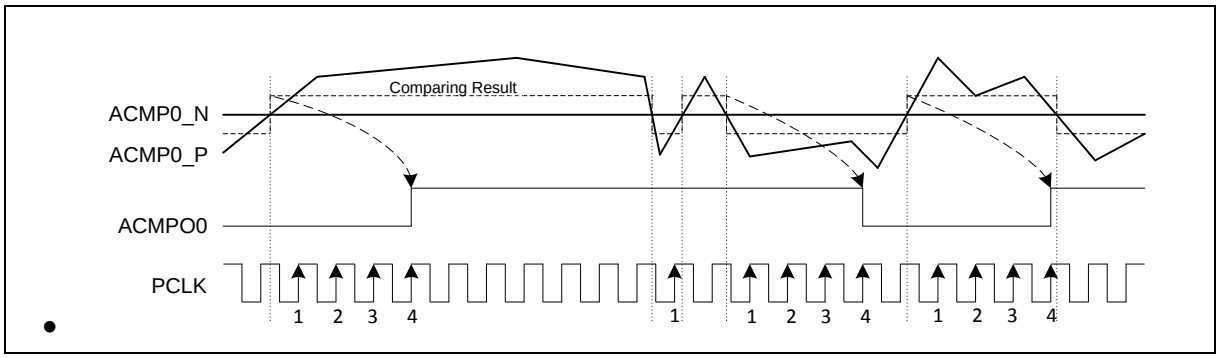


图 6.26-4 滤波器功能的例子

6.26.5.4 中断源

ACMP0和ACMP1的输出分别反映到ACMPO0 (ACMP_STATUS[4]) 和 ACMPO1 (ACMP_STATUS[5])。当他们被窗口锁存器和滤波功能处理后，最后输出的信号就可用于中断。参考图 6.26-5 如果ACMP_CTL0/1寄存器的ACMPIE位被置1，中断使能。如果ACMPO0/1输出状态的改变与INTPOL (ACMP_CTL0/1[9:8])设定相符，将产生比较器中断并且相应的标志ACMPIF0 (ACMP_STATUS[0]) 和 ACMPIF1 (ACMP_STATUS[1])将被置1.中断标志写1清0.

如果ACMP唤醒功能被使能，通过ACMP中断可以将系统从掉电模式下唤醒，WKIF (ACMP_STATUS[8], ACMP_STATUS[9])将被置1并产生中断。

图 6.26-5 展示了来自于ACMPIF或WKIF的ACMP中断，通过ACMPIE使能或禁止。

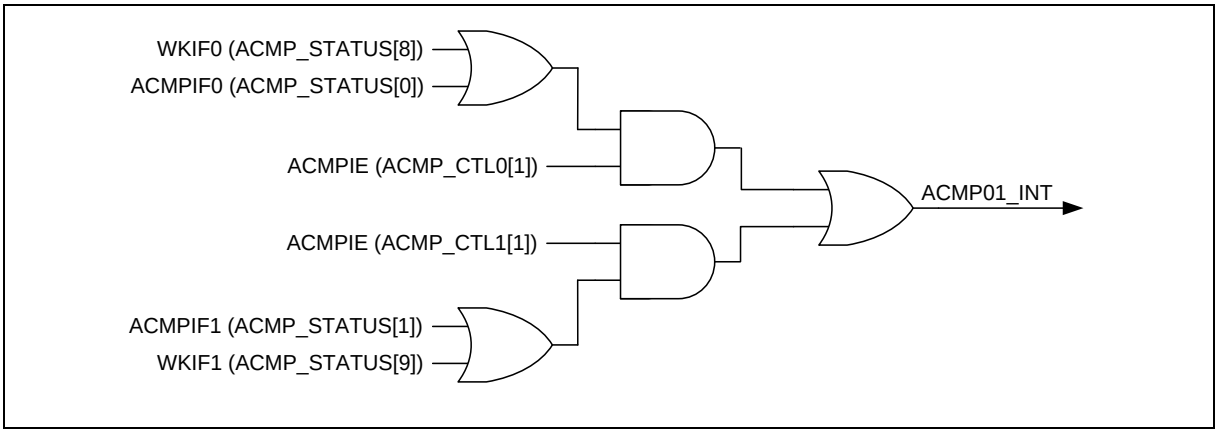


图 6.26-5 比较器控制器中断

6.26.5.5 比较器参考电压(CRV)

比较器参考电压 (CRV) 模块负责产生比较器参考电压。CRV 模块包含电阻梯和模拟开关。用户通过CRVCTL(ACMP_VREF[3:0]) 设置 CRV 输出电压。可以设置 NEGSEL (ACMP_CTL0[5:4]) ，ACMP_CTL1[5:4])选择CRV输出电压作为比较器的负端输入电压。图 6.26-6 展示比较器参考电压的框图

当NEGSEL (ACMP_CTL0[5:4], ACMP_CTL1[5:4])没有被选择，电阻梯可以通过硬件禁用以减少CRV模块的电源消耗。电阻梯的参考电压可以来自于AV_{DD}管脚或VREF管脚。

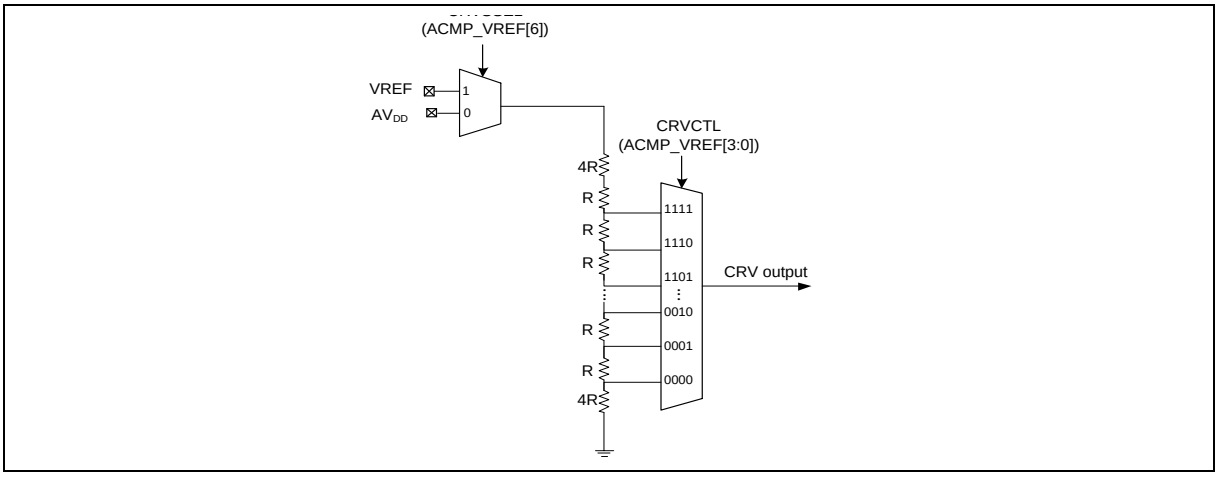


图 6.26-6 比较器参考电压框图

6.26.5.6 窗口比较模式

比较器提供窗口比较模式。设置WCMPSEL (ACMP_CTL0/1[18])为1使能窗口比较器模式，用户可监控某个电压是否在特定范围内。用户可以连接特定的模拟电压源到两个比较器的正输入端或是负输入端。监测电压范围的上、下限取决于两个比较器的另外的输入端。如果一个比较器的输出为低，而另一个比较器的输出为高这就意味着一个比较器监测的是上限，另一个比较器监测的是下限。用户可以通过ACMPWO (ACMP_STATUS[16])直接监控模拟输入电压，如果 ACMPWO 是高，意味着输入电压在上

限和下限的范围之内, 叫做模拟电压在窗口里。

图 6.26-7给出窗口比较模式的例子，该例程中窗口比较模式被选中，用户选择每个比较器四路正输入端中的一路，然后在芯片外部将该两组比较器的输入连接到一起。

如果 ACMPS0 输出高 且 ACMPS1 输出低, 意味着电压源在上限和下限之内，叫做电压源在窗口里，否则，电压源在窗口外。

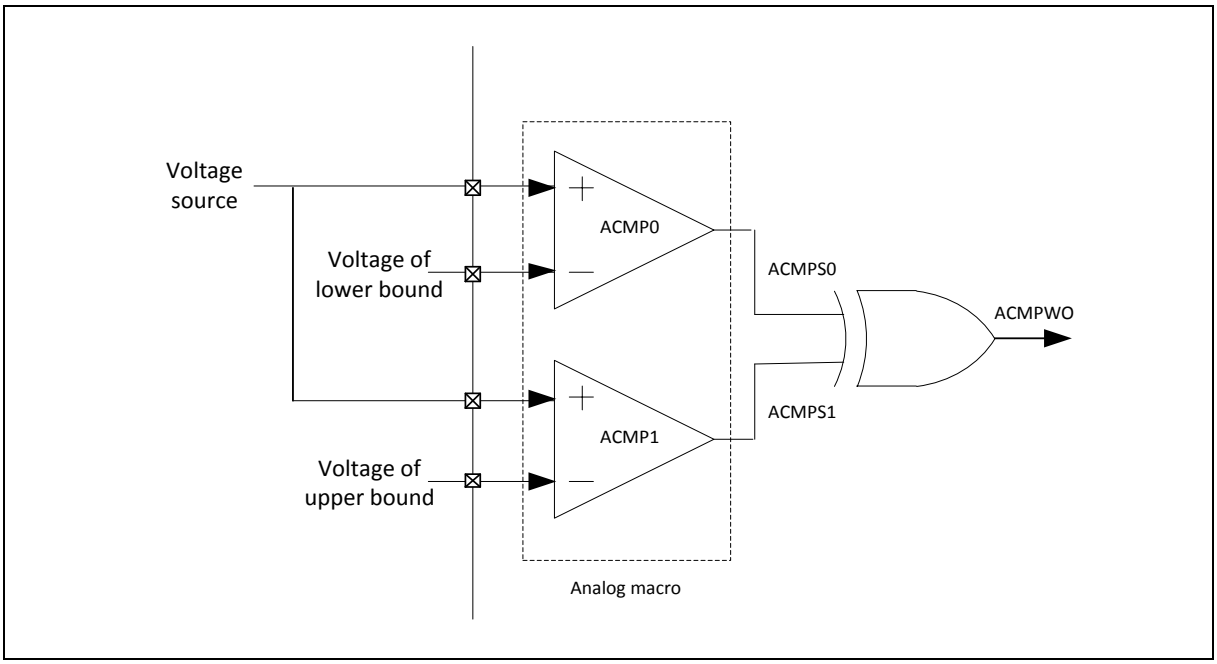


图 6.26-7 窗口比较模式例子

比较器窗口输出 (ACMPWO)在 ACMP_STATUS[16]， 窗口比较逻辑的真值表如表 6.26-2.

• ACMPS0	• ACMPS1	• ACMPWO
• 0	• 0	• 0
• 0	• 1	• 1
• 1	• 1	• 0
• 1	• 0	• 1

表 6.26-2 窗口比较逻辑真值表

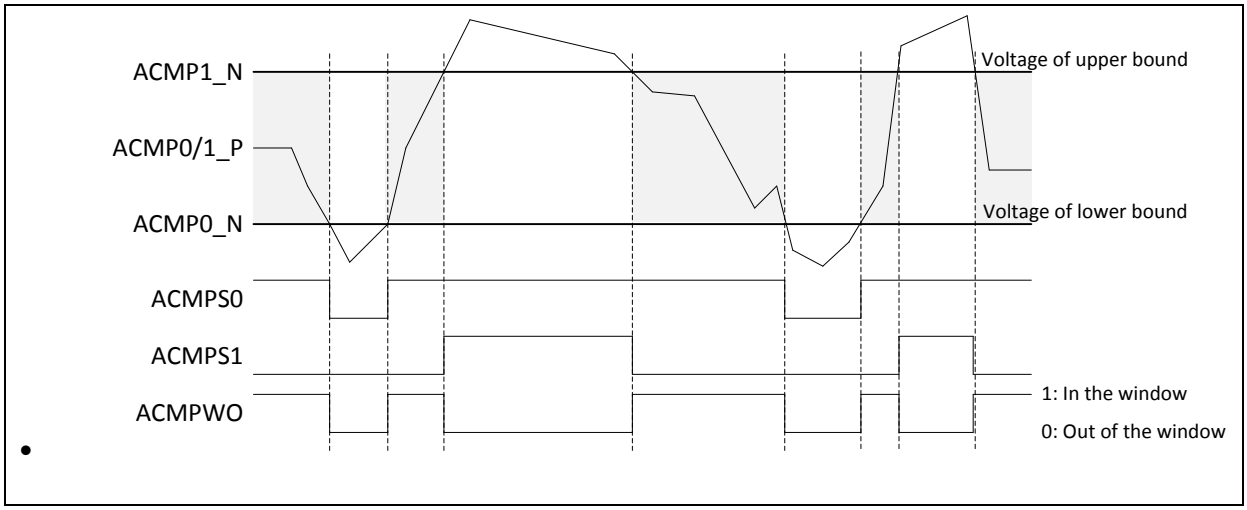


图 6.26-8 窗口比较模式例子

如图 6.26-8所示，如果ACMPWO为1，这就意味着正端输入电压在窗口内，否则就是在窗口外。因此ACMPWO可以被用作监控外部模拟管脚的电压转变。更进一步，ACMPWO 也可以用做窗口锁存，过滤功能和ACMP中断.

注意负端输入必须是不同的电压源，否则该功能就没有任何意义。

6.26.5.7 校准功能

两个比较器都有自己的3位校准位，可以用来校准偏移误差，偏移误差来自于NMOS或PMOS类型的差分输入阶段，设置CALTRG0 (ACMP_CALCTL[0]), CALTRG1 (ACMP_CALCTL[1])开启校准功能

轨到轨的共模输入是通过并联一个NMOS和PMOS的差分输入对达成的。校准后读取ACMP_CALSR寄存器可以监控校准状态，包括校准完成状态，NMOS,PMOS校准结果。简介一个例子：一旦ACMPEN (ACMP_CTLx[0], x=0,1)使能，硬件将自动加载默认校准值补偿偏移电压，因此用户可以直接使用比较器不用每次使用都去校准，如果用户重新校准，新校准的值将更新默认值。每次设置ACMPEN，用户必须设置CALTRG0 (ACMP_CALCTL[0]) 或 CALTRG1 (ACMP_CALCTL[1])启动校准功能并获得新的校准值，否则比较器将使用旧的校准值

6.26.6 寄存器表

R: 只读, **W**: 只写, **R/W**: 可读写

寄存器	偏移	R/W	描述	复位值
ACMP基地址: **ACMP01_BA = 0x4004_5000**				
ACMP_CTL0	ACMP01_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000
ACMP_CTL1	ACMP01_BA+0x04	R/W	模拟比较器1控制寄存器	0x0000_0000
ACMP_STATUS	ACMP01_BA+0x08	R/W	模拟比较器状态寄存器	0x0000_0000
ACMP_VREF	ACMP01_BA+0x0C	R/W	模拟比较器参考电压控制寄存器	0x0000_0000
ACMP_CALCTL	ACMP01_BA+0x10	R/W	模拟比较器校准控制寄存器	0x0000_00f0
ACMP_CALSR	ACMP01_BA+0x14	R	模拟比较器校准状态寄存器	0x0000_0000

6.26.7 寄存器描述

模拟比较器0控制寄存器 (ACMP_CTL0)

寄存器	偏移	R/W	描述	复位值
ACMP_CTL0	ACMP01_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留					**WCMPSEL**	**WLATEN**	**WKEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FILTSEL			**OUTSEL**	保留		**INTPOL**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POSSEL		**NEGSEL**		**ACMPOINV**	保留	**ACMPIE**	**ACMPEN**

位	描述	
[31:19]	保留	保留.
[18]	**WCMPSEL**	**窗口比较模式选择** 0 = 窗口比较模式禁用. 1 = 窗口比较模式选中.
[17]	**WLATEN**	**窗口锁存模式使能位** 0 = 窗口锁存模式禁用. 1 = 窗口锁存模式使能.
[16]	**WKEN**	**掉电唤醒使能位** 0 = 唤醒功能禁用. 1 = 唤醒功能使能 .
[15:13]	**FILTSEL**	**比较器输出滤波采样时长选择** 000 = 滤波功能禁用. 001 = ACMP0 输出经过1 个连续的 PCLKs采样. 010 = ACMP0输出经过2个连续的 PCLKs采样. 011 = ACMP0输出经过4个连续的 PCLKs采样. 100 = ACMP0输出经过8 个连续的 PCLKs采样. 101 = ACMP0输出经过16 个连续的 PCLKs采样. 110 = ACMP0输出经过32个连续的 PCLKs采样. 111 = ACMP0输出经过64 个连续的 PCLKs采样.
[12]	**OUTSEL**	**比较器输出选择** 0 = 比较器 0 输出到 ACMP0_O管脚不经过滤波. 1 = 比较器 0 输出到 ACMP0_O管脚经过滤波..
[11:10]	保留	保留.

[9:8]	**INTPOL**	**中断条件极性选择** ACMPIF0 当下列输出条件被检测到时设为1. 00 = 上升沿或下降沿. 01 = 上升沿. 10 = 下降沿. 11 = 保留.
[7:6]	**POSSEL**	**比较器正端输入选择** 00 = 从 ACMP0_P0输入. 01 = 从 ACMP0_P1输入. 10 = 从 ACMP0_P2输入. 11 = 从 ACMP0_P3输入.
[5:4]	**NEGSEL**	**比较器负端输入选择** 00 = ACMP0_N 管脚. 01 = 内部比较器参考电压 (CRV). 10 = Band-gap 电压. 11 = 保留.
[3]	**ACMPOINV**	**比较器输出反相** 0 = 比较器 0 输出反相禁止. 1 = 比较器 0 输出反相使能 .
[2]	**保留**	保留.
[1]	**ACMPIE**	**比较器中断使能位** 0 = 比较器0中断禁用. 1 = 比较器0中断使能。如果WKEN (ACMP_CTL0[16]) 置1，唤醒中断功能也将使能。
[0]	**ACMPEN**	**比较器使能位** 0 = 比较器0禁用. 1 = 比较器0使能 .

模拟比较器 1 控制寄存器 (ACMP_CTL1)

寄存器	偏移	R/W	描述	复位值
ACMP_CTL1	ACMP01_BA+0x04	R/W	模拟比较器1 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留					**WCMPSEL**	**WLATEN**	**WKEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FILTSEL			**OUTSEL**	保留		**INTPOL**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POSSEL		**NEGSEL**		**ACMPOINV**	保留	**ACMPIE**	**ACMPEN**

位	描述	
[31:19]	保留	保留
[18]	**WCMPSEL**	**窗口比较模式选择** 0 = 窗口比较模式禁用. 1 = 窗口比较模式选中.
[17]	**WLATEN**	**窗口锁存模式使能位** 0 = 窗口锁存模式禁用. 1 = 窗口锁存模式使能.
[16]	**WKEN**	**掉电唤醒使能位** 0 = 唤醒功能禁用. 1 = 唤醒功能使能 .
[15:13]	**FILTSEL**	**比较器输出滤波采样时长选择** 000 = 滤波功能禁用. 001 = ACMP1输出经过1 个连续的 PCLKs采样. 010 = ACMP1输出经过2个连续的 PCLKs采样. 011 = ACMP1输出经过4个连续的 PCLKs采样. 100 = ACMP1输出经过8 个连续的 PCLKs采样. 101 = ACMP1输出经过16 个连续的 PCLKs采样. 110 = ACMP1输出经过32个连续的 PCLKs采样. 111 = ACMP1输出经过64 个连续的 PCLKs采样.
[12]	**OUTSEL**	**比较器输出选择** 0 = 比较器 1 输出到 ACMP1_O管脚不经过滤波. 1 = 比较器 1 输出到 ACMP1_O管脚经过滤波...
[11:10]	保留	保留.

位	描述	
[9:8]	**INTPOL**	**中断条件极性选择** ACMPIF1 当下列输出条件被检测到时设为1. 00 = 上升沿或下降沿. 01 = 上升沿. 10 = 下降沿. 11 = 保留.
[7:6]	**POSSEL**	**比较器正端输入选择** 00 = 从 ACMP1_P0输入. 01 = 从 ACMP1_P1输入. 10 = 从 ACMP1_P2输入. 11 = 从 ACMP1_P3输入.
[5:4]	**NEGSEL**	**比较器负端输入选择** 00 = ACMP1_N 管脚. 01 = 内部比较器参考电压 (CRV). 10 = Band-gap 电压. 11 = 保留.
[3]	**ACMPOINV**	**比较器输出反相** 0 = 比较器 1输出反相禁止. 1 = 比较器 1 输出反相使能
[2]	**保留**	保留.
[1]	**ACMPIE**	**比较器中断使能位** 0 = 比较器1中断禁用. 1 = 比较器1中断使能。如果WKEN (ACMP_CTL1[16]) 置1，唤醒中断功能也将使能。
[0]	**ACMPEN**	**比较器使能位** 0 = 比较器1禁用. 1 = 比较器1使能 .

模拟比较器状态寄存器 (ACMP_STATUS)

寄存器	偏移	R/W	描述	复位值
ACMP_STATUS	ACMP01_BA+0x08	R/W	模拟比较器状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							**ACMPWO**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留		**ACMPS1**	**ACMPS0**	保留		**WKIF1**	**WKIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留		**ACMPO1**	**ACMPO0**	保留		**ACMPIF1**	**ACMPIF0**

位	描述	
[31:17]	**保留**	保留.
[16]	**ACMPWO**	**比较器窗口输出** 这一位展示窗口比较模式的输出状态 0 = 正端输入电压超出窗口范围 1 = 正端输入电压在窗口范围.
[15:14]	**保留**	保留.
[13]	**ACMPS1**	**比较器1状态** 允许程序同步于PCLK 读取.，当比较器 1 禁止（ACMPEN (ACMP_CTL1[0])=0）时清0
[12]	**ACMPS0**	**比较器0状态** 允许程序同步于PCLK 读取.，当比较器 1 禁止（ACMPEN (ACMP_CTL0[0])=0）时清0.
[11:10]	**保留**	保留.
[9]	**WKIF1**	**比较器1 掉电唤醒中断标志** 当ACMP1唤醒中断事件发生该位将置1 0 = 没有掉电唤醒发生. 1 = 掉电唤醒发生. **注:** 该位写1清0 .
[8]	**WKIF0**	**比较器0 掉电唤醒中断标志** 当ACMP0唤醒中断事件发生该位将置1 0 = 没有掉电唤醒发生. 1 = 掉电唤醒发生. **注:** 该位写1清0 .
[7:6]	**保留**	保留.
[5]	**ACMPO1**	**比较器1 输出**

位	描述	
		允许程序同步于PCLK 读取.，当比较器 1 禁止（ACMPEN (ACMP_CTL1[0])=0）时清0
[4]	**ACMPO0**	**比较器0 输出** 允许程序同步于PCLK 读取.，当比较器 1 禁止（ACMPEN (ACMP_CTL0[0])=0）时清0
[3:2]	**保留**	保留.
[1]	**ACMPIF1**	**比较器1 中断标志** 当比较器1的输出检测到INTPOL (ACMP_CTL1[9:8])定义的条件成立，此位由硬件置1，当ACMPIE (ACMP_CTL1[1])为1会产生中断. **注:** 该位写1清0.
[0]	**ACMPIF0**	**比较器1 中断标志** 当比较器1的输出检测到INTPOL (ACMP_CTL0[9:8])定义的条件成立，此位由硬件置1，当ACMPIE (ACMP_CTL0[1])为1会产生中断. **注:** 该位写1清0.

ACMP参考电压控制寄存器 (ACMP_VREF)

寄存器	偏移	R/W	描述	复位值
ACMP_VREF	ACMP01_BA+0x0C	R/W	模拟比较器参考电压控制寄存器	0x0000_0000

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

位	描述	
[31:7]	保留	保留
[6]	**CRVSSEL**	**CRV 电压源选择** 0 = AV_{DD} 被选择作为 CRV 电压源. 1 = V_{REF}被选择作为CRV 电压源.
[5:4]	保留	保留.
[3:0]	**CRVCTL**	**比较器参考电压设定** CRV = CRV 电压源* (1/6+CRVCTL/24).

模拟比较器校准控制寄存器(ACMP_CALCTL)

寄存器	偏移	R/W	描述	复位值
ACMP_CALCTL	ACMP01_BA+0x10	R/W	模拟比较器校准控制寄存器	0x0000_00f0

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留						**CALRVS1**	**CALRVS0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留						**CALTRG1**	**CALTRG0**

位	描述	
[31:18]	保留	保留.
[17]	**CALRVS1**	**OPA1 校准参考电压选择** 0 = V_{REF} is $\frac{1}{2}$ AV_{DD}. 1 = V_{REF} 从高vcm到低vcm **注:** CALRVS0 和 CALRVS1必须要相同设定
[16]	**CALRVS0**	**OPA0校准参考电压选择** 0 = V_{REF} is $\frac{1}{2}$ AV_{DD}. 1 = V_{REF}从高vcm到低vcm **注:** CALRVS0 和 CALRVS1必须要相同设定
[15:2]	保留	保留.
[1]	**CALTRG1**	**OPA1校准触发位** 0 = 校准停止 1 = 校准触发 **注1:** 使能此位前提前使能HIRC及ACMPEN(ACMP_CTL1) **注 2:**软件下次触发校准时，硬件会自动清除此位 **注 3:** 如果用户连续触发校准，必须等待校准完成300us后才可以触发下一次校准
[0]	**CALTRG0**	**OPA0校准触发位** 0 = 校准停止 1 = 校准触发 **注1:** 使能此位前提前使能HIRC及ACMPEN(ACMP_CTL0) **注 2:**软件下次触发校准时，硬件会自动清除此位 **注 3:** 如果用户连续触发校准，必须等待校准完成300us后才可以触发下一次校准

模拟比较器校准状态寄存器(ACMP_CALSR)

寄存器	偏移	R/W	描述	复位值
ACMP_CALSR	ACMP01_BA+0x14	R	模拟比较器校准状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
保留							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
保留							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
保留	**CALPS1**	**CALNS1**	**DONE1**	保留	**CALPS0**	**CALNS0**	**DONE0**

位	描述	
[31:7]	保留	保留.
[6]	**CALPS1**	**模拟比较器1校准结果状态PMOS** 0 = 成功 1 = 失败
[5]	**CALNS1**	**模拟比较器1校准结果状态NMOS** 0 = 成功 1 = 失败
[4]	**DONE1**	**模拟比较器1校准完成状态** 0 = 正在校准 1 = 校准完成
[3]	保留	保留.
[2]	**CALPS0**	**模拟比较器0准结果状态PMOS** 0 = 成功 1 = 失败
[1]	**CALNS0**	**模拟比较器0准结果状态NMOS** 0 = 成功 1 = 失败

位	描述	
[0]	**DONE0**	**模拟比较器0准完成状态** 0 = 正在校准 1 = 校准完成

6.27 外设互联

6.27.1 概述

有些外设之间有内部互连，允许自主通讯或同步动作而不需要CPU的介入。不需要CPU与外设之间交互可以节约CPU资源，减小电源的消耗，而且操作不但没有软件延迟而是更快的响应。

6.27.2 外设互连矩阵表

源	目标					
	ADC	HIRC TRIM	BPWM	PWM	Timer	UART/USCI
ACMP	-	-	-	3	6	-
BOD	-	-	-	3	-	-
Clock Fail	-	-	-	3	-	-
CPU Lockup	-	-	-	3	-	-
LIRC	-	-	-	-	6	-
HXT	-	-	-	-	-	
LXT	-	2		-	-	
BPWM	1		4	-	-	-
PWM	1	-	4	4	-	8
Timer	1	-	5	5	7	-
USBD	-	2	-	-	-	-

表 6.27-1 外设互联矩阵表

6.27.3 功能描述

6.27.3.1 从 BPWM, PWM, TIMER 到 ADC

BPWM 触发 ADC 转换

BPWM0/1 可以作为ADC的触发源之一

设置BPWM触发ADC在章节6.25.5.12.描述

具体BPWM触发条件在章节6.11.5.16.描述.

PWM 触发 ADC 转换

PWM0/1 可以作为一种ADC转换的触发源.

设置ADC 外部硬件触发输入源来自于PWM触发的描述在章节6.25.5.11

PWM触发条件的详细描述在章节6.12.5.23

定时器触发 ADC 转换

定时器0~定时器3都可以作为一种ADC转换的触发源。当定时器计数值与定时器比较值匹配时或者

TMx_EXT管脚边沿变化与设定相符，定时器将触发ADC转换。

设置ADC外部硬件触发输入源来自于定时器触发的描述在章节6.25.5.10。

定时器触发条件的详细描述在章节6.7.5.10。

6.27.3.2 从 LXT 及 USBD 到 HIRC 校准

用 LXT 或者 USB 同步模式 去自动校准系统 HIRC 电路

这个芯片支持自动校准功能：校准HIRC（48 MHz RC 震荡）是根据精确的外部32.768kHz晶振振荡器或者内部USB同步模式，自动的获取精确的输出频率，全温度范围内误差0.25 %。

HIRC校准设定的详细描述在章节6.3.9

6.27.3.3 从 ACMP, BOD, 时钟故障及CPU锁死到PWM

ACMP_O 输出 和系统故障信号作为PWM 刹车源

PWM 刹车源可以是ACMP0/1_O输出信号或者某些不同的系统故障条件包括时钟故障，BOD检测和内核锁定。当系统错误，PWM刹车信号将产生，PWM输出将被设定为保护PWM控制的电压开关。

PWM刹车功能的详细描述在章节6.12.5.19.

6.27.3.4 从 BPWM/PWM 到 BPWM/PWM

BPWM 同步开始功能

从BPWM0, BPWM1, PWM0 或 PWM1中选择同步源，选择BPWM通道，使能同步功能并设置CNTSEN(BPWM_SSTRG[0])，所选择的BPWM通道将与同步源同时开始计数。

BPWM同步开始功能的详细描述请参考章节6.11.5.11.

PWM 同步开始功能

从PWM0或者PWM1选择同步源，然后选择PWM通道。一旦同步开始功能被使能和CNTSEN(PWM_SSTRG[0])被置位，被选择的PWM通道将开始计数。.

PWM同步开始功能的详细描述在章节6.12.5.21

6.27.3.5 从TIMER到BPWM/PWM

定时器产生触发脉冲作为BPWM 外部时钟源

定时器0 ~ 定时器3 可以产生触发脉冲作为BPWM0/BPWM1 外部时钟源

当定时器计数器的值和定时器比较值相匹配或者TMx_EXT管脚边沿变化与设定相符，定时器就会产生触发脉冲，相关的设定描述在章节6.7.5.10

BPWM时钟源设定的详细描述在章节6.11.3

定时器产生触发脉冲作为PWM 外部时钟源

定时器0 ~ 定时器3 可以产生触发脉冲作为PWM0/PWM1 外部时钟源

当定时器计数器的值和定时器比较值相匹配或者TMx_EXT管脚边沿变化与设定相符，定时器就会产生触发脉冲，相关的设定描述在章节6.7.5.10。

PWM时钟源设定的详细描述在章节6.12.3。

6.27.3.6 从 ACMP 和 LIRC 到 定时器捕获功能

测量 ACMP0/1 输出信号的时间间隔或者 LIRC 时钟速度

设定定时器捕获源来自于ACMP0/1 输出信号或者LIRC 时钟然后通过定时器捕获功能测量信号的间隔时间。用户可以用这个间隔时间的结果通过软件校准LIRC或者获取ACMP0/1输出的脉冲宽度。

详细的定时器捕获功能的描述在章节6.7.5.8 和 6.7.5.9。

6.27.3.7 从定时器0/2到定时器 1/3

内部定时器触发捕获模式

定时器0/2在事件计数模式下将强制计数外部事件，然后产生一个内部信号（INTR_TMR_TRG）去触发定时器1/3开始或者停止计数。定时器1/3将被强制定为捕获模式并且通过定时器0/2计数器状态来决定开始/停止触发计数。

详细的内部定时器触发捕获模式的介绍在章节6.7.5.11。

6.27.3.8 从PWM 到UART0/USCI0

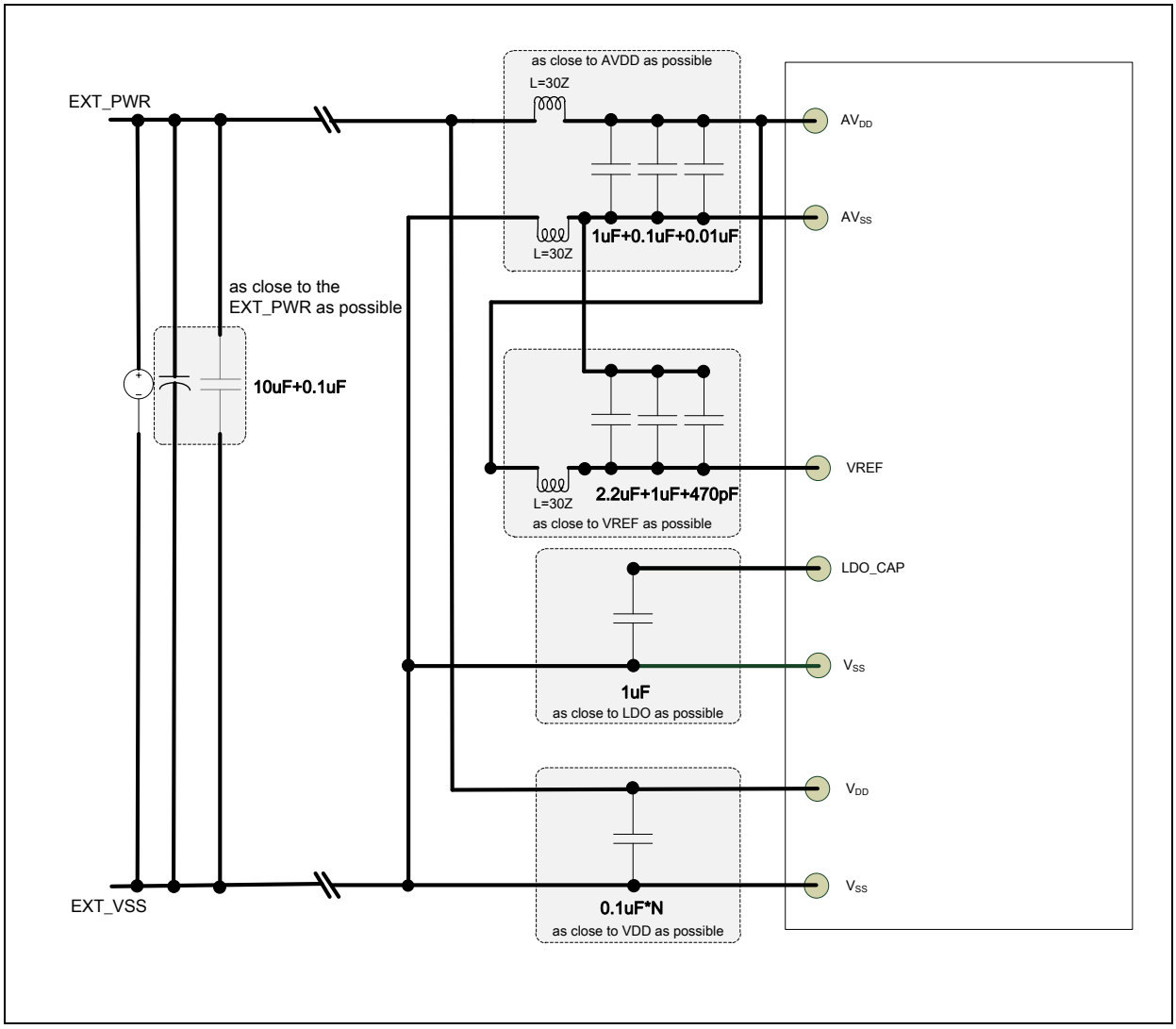
用 PWM 调制 UART0_TXD/USCI0_DAT1

该芯片支持UART0_TXD/USCI_DAT0以PWM通道进行调制。经过调制UART信号以PWM输出信号。这样就很容易开发IrDA信号协议。

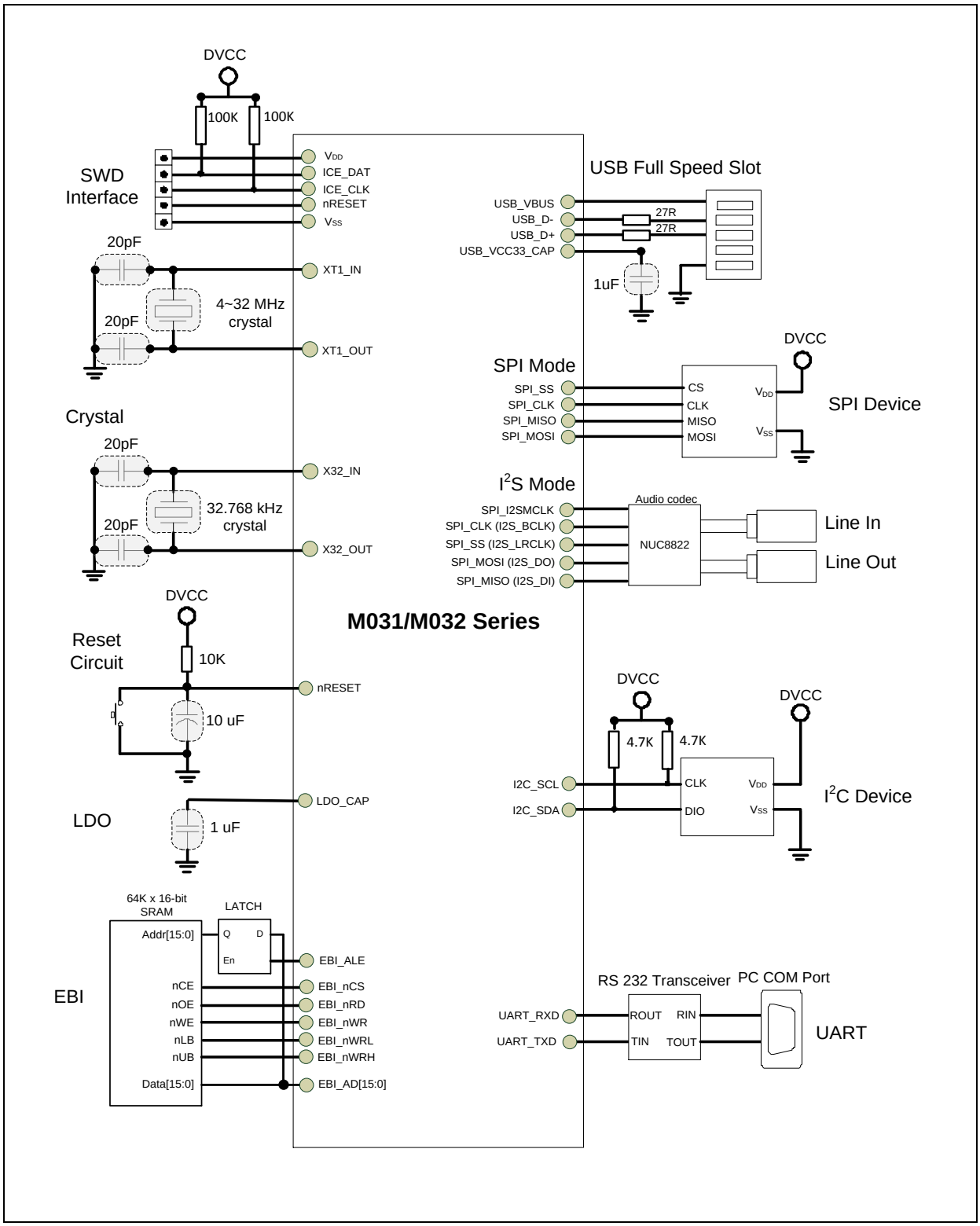
详细的调制设定描述在章节6.3.11

7 应用电路

7.1 供电电源电路



7.2 外设应用电路



注1： 建议在ICE_DAT和ICE_CLK引脚上都使用100kΩ上拉电阻。

注2： 建议在nRESET引脚上使用10kΩ上拉电阻和10 uF电容器。

8 电气特性

M031/M032 的详细电气特性请参考对应的数据手册

9 缩写

9.1 缩写

首字母缩写词	描述
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
BOD	Brown-out Detection
CAN	Controller Area Network
DAP	Debug Access Port
DES	Data Encryption Standard
EADC	Enhanced Analog-to-Digital Converter
EBI	External Bus Interface
EMAC	Ethernet MAC Controller
EPWM	Enhanced Pulse Width Modulation
FIFO	First In, First Out
FMC	Flash Memory Controller
FPU	Floating-point Unit
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	12 MHz Internal High Speed RC Oscillator
HXT	4~32 MHz External High Speed Crystal Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LIRC	10 kHz internal low speed RC oscillator (LIRC)
MPU	Memory Protection Unit
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PDMA	Peripheral Direct Memory Access
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation

QEI	Quadrature Encoder Interface
SD	Secure Digital
SPI	Serial Peripheral Interface
SPS	Samples per Second
TDES	Triple Data Encryption Standard
TK	Touch Key
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

表 9.1-1 缩写列表

10 修订历史

日期	版本	描述
2018.12.24	1.00	初始版本。
2019.02.25	1.01	1. 在 3.2 节中修改了 ISP ROM 的大小。 2. 修改第 6.7.3 节中的图 6.7-1。 3. 在 6.3.9 节、6.27.2 节和 6.27.3 节中修改 HIRC 调整参考时钟。 4. 在 6.6.7 节中修改对 PDMA_TACTSTS 寄存器的描述。
2019.05.15	1.02	1. 在第 6.22.7 节中修改 USBD_EPSTS0 寄存器的描述。 2. 修改第 6.25.5.9 节“内部参考电压”的内容。 3. 在表 6.4-4 ISP 命令列表中添加了内置 V_{BG} 的 A/D 转换结果。
2019.07.15	1.03	1. 在图 6.3-6 中添加一个 USB 模块，并移除温度传感器模块。 2. 在 4.1 节中增加多功能引脚表。
2019.11.04	2.00	添加 M031xI / M032xI / M031xG / M032xG / M032xC / M032xD 的新料号，并更新新料号的描述。
2020.04.29	2.01	1. 在第 4.1.4.1 节和第 4.1.4.2 节中修改多功能引脚图名称和多功能引脚表。 2. 在第 4.2 节和第 4.3 节中，将“引脚描述”表更改为“引脚映射”表和“引脚功能描述”表。 3. 在第 6.4.6 节中修改闪存访问时间控制寄存器的描述。 4. 在第 6.15.9 节中，将 QSPI 时钟分频器寄存器的频率名称从 SPI 修改为 QSPI。 5. 修改表 6.3-1 中的上电复位（POR）后的 SYS_RSTSTS 值。 6. 在第 4.3 节和第 7 章中添加有关 ICE_DAT，ICE_CLK 和 nRESET 引脚的硬件参考设计的说明。
2020.09.29	2.02	1. 添加新的料号 M031TE3AE 并更新第 3 章和第 4 章中对新料号的描述。 2. 在第 6.11.7 节中修改 BPWM_WGCTL0 寄存器表的格式。 3. 在 6.18.4 和 6.20.4 节中修改 USCI-UART 和 USCI-I2C 的基本配置。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.