

ARM[®] Cortex[®]-M0**32-位控制器**

NuMicro[®]家族
NUC029xAN_xAE 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

图集	9
表集	14
1 概述	15
2 特性	16
3 缩写	20
4 编号信息列表与管脚定义	21
4.1 NuMicro® NUC029xAN_xAE 系列选型指南	21
4.2 管脚配置	23
4.2.1 NuMicro® NUC029xAN_xAE 管脚图	23
4.3 管脚描述	27
4.3.1 NuMicro® NUC029xAN_xAE 管脚描述	27
5 功能描述	35
5.1 ARM® Cortex®-M0 内核	35
5.2 系统管理器	37
5.2.1 概述	37
5.2.2 系统复位	37
5.2.3 系统电源分布	38
5.2.4 系统存储器映射	39
5.2.5 系统存储器映射表	42
5.2.6 NUC029xAN系统管理器控制寄存器映射	44
5.2.7 NUC029FAE系统管理器控制寄存器映射	73
5.2.8 系统定时器(SysTick)	99
5.2.9 NUC029xAN系统定时器控制寄存器映射	99
5.2.10 NUC029FAE系统定时器控制寄存器映射	103
5.2.11 嵌套向量中断控制器(NVIC)	107
5.2.12 NUC029xAN NVIC 控制寄存器映射	112
5.2.13 NUC029FAE NVIC 控制寄存器映射	150
5.2.14 系统控制块(SCB)	184
5.2.15 NUC029xAN系统控制块寄存器映射	184
5.2.16 NUC029FAE系统控制块寄存器映射	192
5.3 NuMicro® NUC029xAN时钟控制器	200

5.3.1 概述	200
5.3.2 系统时钟 和 SysTick 时钟	203
5.3.3 掉电模式时钟	204
5.3.4 分频器输出	205
5.3.5 寄存器映射	206
5.3.6 寄存器描述	207
5.4 NuMicro® NUC029FAE时钟控制器	227
5.4.1 概述	227
5.4.2 系统时钟 和 SysTick 时钟	228
5.4.3 ISP时钟源选择	229
5.4.4 模块时钟源选择	229
5.4.5 掉电模式时钟	231
5.4.6 寄存器映射	232
5.4.7 寄存器描述	233
5.5 内存控制器 (FMC)	246
5.5.1 概述	246
5.5.2 特性	246
5.5.3 NUC029xAN FMC 框图	247
5.5.4 NUC029FAE FMC 框图	248
5.5.5 FMC组织结构	249
5.5.6 数据Flash	251
5.5.7 用户配置区	253
5.5.8 启动选择	258
5.5.9 应用内编程	260
5.5.10 系统内编程 (ISP)	261
5.5.11 寄存器控制过程	261
5.5.12 通过向量重新映射多启动	264
5.5.13 寄存器映射 (NUC029xAN)	266
5.5.14 寄存器描述 (NUC029xAN)	267
5.5.15 寄存器映射 (NUC029FAE)	276
5.5.16 寄存器描述 (NUC029FAE)	277
5.6 外部总线接口(EBI) (NUC029LAN/NUC029NAN)	285
5.6.1 概述	285

5.6.2 特性	285
5.6.3 EBI 框图	286
5.6.4 基本配置	287
5.6.5 功能描述	287
5.6.6 寄存器映射	292
5.6.7 寄存器描述	293
5.7 通用 I/O (GPIO)	298
5.7.1 概述	298
5.7.2 特性	298
5.7.3 基本配置	299
5.7.4 功能描述	299
5.7.5 GPIO 中断和唤醒功能	300
5.7.6 寄存器映射 (NUC029xAN)	301
5.7.7 寄存器描述 (NUC029xAN)	303
5.7.8 寄存器描述 (NUC029FAE)	316
5.7.9 寄存器描述 (NUC029FAE)	319
5.8 定时器控制器 (TIMER)	334
5.8.1 概述	334
5.8.2 特性	334
5.8.3 框图	335
5.8.4 基本配置	336
5.8.5 功能描述	337
5.8.6 寄存器映射 (NUC029xAN)	342
5.8.7 寄存器描述 (NUC029xAN)	344
5.8.8 寄存器映射 (NUC029FAE)	354
5.8.9 寄存器描述 (NUC029FAE)	355
5.9 PWM 发生器和捕捉定时器 (PWM) (NUC029xAN)	365
5.9.1 概述	365
5.9.2 特性	365
5.9.3 框图	367
5.9.4 基本配置	371
5.9.5 功能描述	371
5.9.6 寄存器映射	384
5.9.7 寄存器描述	386

5.10增强型PWM发生器(NUC029FAE)	414
5.10.1概述	414
5.10.2特性	414
5.10.3方框图	416
5.10.4基本配置	417
5.10.5功能描述	418
5.10.6PWM电机控制中断架构	428
5.10.7PWM计数器配置过程	429
5.10.8PWM计数器停止过程	430
5.10.9寄存器映射	431
5.10.10寄存器描述	433
5.11看门狗定时器(WDT)	469
5.11.1概述	469
5.11.2特性	469
5.11.3框图	470
5.11.4基本配置	470
5.11.5功能描述	471
5.11.6寄存器表	473
5.11.7寄存器表	474
5.12窗口式看门狗定时器 (WWDT) (NUC029xAN)	477
5.12.1概述	477
5.12.2特性	477
5.12.3框图	478
5.12.4基本配置	478
5.12.5功能描述	478
5.12.6寄存器表	481
5.12.7寄存器描述	482
5.13UART接口控制器 (UART)	487
5.13.1概述	487
5.13.2特性	487
5.13.3框图	488
5.13.4基本配置	490

5.13.5 功能描述	491
5.13.6 NUC029xAN 寄存器映射	502
5.13.7 NUC029xAN 寄存器描述	503
5.13.8 NUC029FAE 寄存器映射	525
5.13.9 NUC029FAE 寄存器描述	526
5.14 I ² C 串行接口控制器 (I ² C)	546
5.14.1 概述	546
5.14.2 特性	546
5.14.3 基本配置	547
5.14.4 功能描述	547
5.14.5 I ² C 协议	548
5.14.6 I ² C 协议寄存器	551
5.14.7 操作模式	555
5.14.8 NUC029xAN 寄存器映射	563
5.14.9 NUC029xAN 寄存器描述	564
5.14.10 NUC029FAE 寄存器映射	574
5.14.11 NUC029FAE 寄存器描述	575
5.15 串行外设接口 (SPI)	586
5.15.1 概述	586
5.15.2 特性	586
5.15.3 方框图	587
5.15.4 基本配置	587
5.15.5 功能描述	587
5.15.6 时序图	595
5.15.7 编程示例	596
5.15.8 NUC029xAN 寄存器映射	599
5.15.9 NUC029xAN 寄存器描述	600
5.15.10 NUC029FAE 寄存器映射	614
5.15.11 NUC029FAE 寄存器描述	615
5.16 模数转换器(ADC)	629
5.16.1 概述	629
5.16.2 特性	629

5.16.3 NUC029xAN方框图	630
5.16.4 NUC029FAE方框图	631
5.16.5 基本配置.....	631
5.16.6 功能描述.....	632
5.16.7 NUC029xAN寄存器映射	640
5.16.8 NUC029xAN寄存器描述	641
5.16.9 NUC029FAE寄存器映射	651
5.16.10 NUC029FAE寄存器描述	652
5.17 模拟比较器 (ACMP)	663
5.17.1 概述	663
5.17.2 特性	663
5.17.3 NUC029xAN方框图	664
5.17.4 NUC029FAE方框图	666
5.17.5 基本配置.....	667
5.17.6 功能描述.....	667
5.17.7 比较器参考电压(CRV) (NUC029FAE)	668
5.17.8 NUC029xAN寄存器映射	670
5.17.9 NUC029xAN寄存器描述	671
5.17.10 NUC029FAE寄存器映射	677
5.17.11 NUC029FAE寄存器描述	678
5.18 硬件除法器 (HDIV) (NUC029xAN)	684
5.18.1 概述	684
5.18.2 特性	684
5.18.3 基本配置.....	685
5.18.4 功能描述.....	685
5.18.5 寄存器 描述	686
5.18.6 寄存器描述	687
6 电气特性	692

7 封装尺寸	693
7.1 48-pin LQFP (7x7x1.4 mm)	693
7.2 48-pin QFN (7x7x0.8 mm)	694
7.3 33-pin QFN (5x5x0.8 mm)	695
7.4 33-pin QFN (4x4x0.8 mm)	696
7.5 20-pin TSSOP (6.5x4.4x1.2 mm)	697
8 修正历史	698

图集

图 4-1 NuMicro® NUC029 命名规则	22
图 4-2 NuMicro® NUC029LAN LQFP 48-pin 图	23
图 4-3 NuMicro® NUC029NAN QFN 48-pin 图	24
图 4-4 NuMicro® NUC029ZAN/NUC029TAN QFN 33-pin 图	25
图 4-5 NuMicro® NUC029FAE TSSOP 20-pin 图	26
图 5-1 功能控制框图	35
图 5-2 NuMicro® NUC029xAN 系列电源分布图	38
图 5-3 NuMicro® NUC029FAE 系列电源分布图	39
图 5-4 NuMicro® NUC029xAN 系统存储器映射	42
图 5-5 NuMicro® NUC029FAE 系统存储器映射	43
图 5-6 NuMicro® NUC029xAN 时钟发生器框图	200
图 5-7 NuMicro® NUC029xAN 时钟源控制器概图(1/2)	201
图 5-8 NuMicro® NUC029xAN 时钟源控制器概图(2/2)	202
图 5-9 NuMicro® NUC029xAN 系统时钟框图	203
图 5-10 NuMicro® NUC029xAN SysTick 时钟控制框图	203
图 5-11 NuMicro® NUC029xAN 分频器的时钟源	205
图 5-12 NuMicro® NUC029xAN 分频器框图	205
图 5-13 NuMicro® NUC029FAE 时钟发生器框图	227
图 5-14 NuMicro® NUC029FAE 系统时钟框图	228
图 5-15 NuMicro® NUC029FAE SysTick 时钟控制框图	228
图 5-16 NuMicro® NUC029FAE AHB HCLK 时钟源选择	229
图 5-17 NuMicro® NUC029FAE PCLK 外设时钟源选择	230
图 5-18 NUC029xAN Flash 存储器控制器框图	247
图 5-19 NUC029FAE Flash 存储器控制器框图	248
图 5-20 Flash 存储器组织结构 (NUC029xAN)	250
图 5-21 Flash 存储器组织结构 (NUC029FAE)	251
图 5-22 Flash 存储器结构 (NUC029xAN)	252
图 5-23 Flash 存储器结构 (NUC029FAE)	253
图 5-24 上电后启动选择 (BS)	259
图 5-25 从 APROM 和 LDROM 启动时程序的执行范围	259
图 5-26 IAP 功能使能时可执行地址范围	261
图 5-27 CBS[0] = 1 时通过 BS 位改变启动选项范例流程	262
图 5-28 ISP 流程范例	263

图5-29 通过向量重新映射多引导	265
图 5-30 EBI 框图	286
图 5-31 16位EBI数据宽度与16位器件连接	287
图 5-32 8位EBI数据宽度与8位设备连接	288
图 5-33 16位数据宽度的EBI时序控制波形	289
图 5-34 8位数据宽度EBI时序控制波形	290
图 5-35 插入空闲周期的EBI时序控制波形	291
图 5-36 推挽输出	299
图 5-37 开漏输出	300
图 5-38 准双向 I/O模式	300
图 5-39 定时器控制器框图 (NUC029xAN)	335
图 5-40 定时器控制器时钟源 (NUC029xAN)	335
图 5-41 定时器控制器框图 (NUC029FAE)	336
图 5-42 定时器控制器时钟源 (NUC029FAE)	336
图 5-43 连续计数模式	338
图 5-44 Inter-Timer 触发捕获时序图	341
图 5-45 PWM 发生器0 时钟源控制	367
图 5-46 PWM 发生器0 结构框图	367
图 5-47 PWM 发生器2 时钟源控制	368
图 5-48 PWM 发生器2 结构框图	368
图 5-49 PWM 发生器4 时钟源控制	369
图 5-50 PWM 发生器4 结构框图	369
图 5-51 PWM 发生器6 时钟源控制	370
图 5-52 PWM 发生器6 结构框图	370
图 5-53 定时器内部比较器输出	372
图 5-54 PWM 定时器操作时序	372
图 5-55 PWM 周期中断发生时序图	372
图 5-56 中心对齐输出波形	373
图 5-57 PWM 中心对齐中断发生时序图	374
图 5-58 PWM 双缓存图解	375
图 5-59 PWM 控制器输出占空比	375
图 5-60 死区发生器操作的PWM对输出	376
图 5-61 中心对齐模式下PWM 触发 ADC 标志 (PWMxTF) 时序图	377
图 5-62 边沿对齐时PWM 触发ADC标志 (PWMxTF) 时序图	378

图 5-63 中心对齐模式下 PWM 触发ADC 开始转换	379
图 5-64 捕捉操作时序	380
图 5-65 PWM A组 PWM定时器中断结构图	381
图 5-66 PWM B组 PWM定时器中断结构图	381
图 5-67 PWM方框图	416
图 5-68 PWM发生器 0 架构框图	416
图 5-69 PWM发生器2架构框图	417
图 5-70 PWM 发生器 4架构框图	417
图 5-71边沿对齐模式PWM	419
图 5-72 PWM边沿对齐时序图	419
图 5-73边沿对齐流程图	420
图 5-74 PWM-Timer内部比较器工作原理	421
图 5-75 PWM-Timer工作时序	421
图 5-76中心对齐模式	422
图 5-77中心对齐模式时序图	423
图 5-78 PWM中心对齐模式波形图	423
图 5-79中心对齐模式流程图(INT_TYPE = 0)	424
图 5-80 PWM 双缓存图解	425
图 5-81 PWM 控制器输出占空比	426
图 5-82 死区时间示意图	427
图 5-83上升沿死区PWM极性控制初始化状态	428
图 5-84电机控制PWM中断架构	429
图 5-85看门狗定时器时钟框图	470
图 5-86看门狗定时器模块图	470
图 5-87看门狗定时器定时溢出间隔和复位周期时序图	472
图 5-88 窗口看门狗定时器时钟控制	478
图 5-89 窗口看门狗定时器框图	478
图 5-90窗口看门狗定时器复位和重载	480
图 5-91 NUC029xAN UART 时钟控制框图	488
图 5-92 NUC029FAE UART 时钟控制框图	488
图 5-93 UART 控制器框图	489
图 5-94 自动流控模块框图	490
图 5-95 UART CTS 自动流控功能使能	495
图 5-96 UART RTS自动流控功能使能	495

图 5-97 UART RTS 软件控制流程.....	496
图 5-98 IrDA 控制模块框图.....	496
图 5-99 IrDA TX/RX 时序框图	497
图 5-100 RS-485 自动方向模式下的 RTS 驱动电平.....	499
图 5-101 软件控制下的RS-485 RTS 驱动电平	499
图 5-102 RS-485 帧结构	500
图 5-103 LIN的帧结构图.....	500
图 5-104 I ² C 总线时序	547
图 5-105 I ² C 协议	548
图 5-106 起始和停止条件	549
图 5-107 I ² C 总线上的位传输	549
图 5-108 I ² C 总线上的应答信号	550
图 5-109 主机传送数据到从机	550
图 5-110 主机读取从机数据.....	550
图 5-111 主机写时 二级缓存传输时序	551
图 5-112 从机读时 二级缓存传输时序	551
图 5-113 I ² C 数据移动方向	552
图 5-114 I ² C 超时计数器框图	554
图 5-115 根据当前 I ² C 状态码来控制 I ² C 总线	555
图 5-116 主机发送模式控制流程	556
图 5-117 主机接收模式控制流程	557
图 5-118 从机模式控制流程.....	558
图 5-119 GC 模式	560
图 5-120 随机读取 EEPROM	561
图 5-121 随机读取 EEPROM 的协议	562
图 5-122 SPI 方框图	587
图 5-123 SPI 主机模式应用方框图	588
图 5-124 SPI 从机模式应用方框图	588
图 5-125 一次传输32位	589
图 5-126 字节重排序	591
图 5-127 字节休眠时序波形.....	591
图 5-128 FIFO 模式框图	593
图 5-129 SPI主机模式时序	595
图 5-130 SPI主机模式时序 (SPICLK相位相反)	595

图 5-131 SPI从机模式时序	596
图 5-132 SPI 从机模式时序(SPICLK相位相反)	596
图 5-133 A/D 转换器方框图 (NUC029xAN)	630
图 5-134 A/D 转换器方框图 (NUC029FAE)	631
图 5-135 ADC外设时钟控制 (NUC029xAN)	632
图 5-136 ADC外设时钟控制 (NUC029FAE)	633
图 5-137 单次模式转换时序图 (NUC029xAN)	634
图 5-138 单次模式转换时序图 (NUC029FAE)	634
图 5-139 使能通道单次周期扫描模式时序图	636
图 5-140 使能通道连续扫描模式时序图	637
图 5-141 A/D转换结果监控逻辑框图 (NUC029xAN)	638
图 5-142 A/D转换结果监控逻辑框图 (NUC029FAE)	639
图 5-143 A/D 控制器中断	639
图 5-144 ADC单端输入采样结果映射图	642
图 5-145 ADC差分输入换换结果映射图	642
图 5-146 ADC单端输入转换结果映射图	653
图 5-147 模拟比较器0/1方框图 (NUC029xAN)	664
图 5-148 模拟比较器2/3方框图 (NUC029xAN)	665
图 5-149 模拟比较器方框图 (NUC029FAE)	666
图 5-150 模拟比较器控制器中断源 (NUC029xAN)	667
图 5-151 模拟比较器控制器中断源 (NUC029FAE)	668
图 5-152 比较器迟滞功能	668
图 5-153 比较器参考电压方框图	669
图 5-154 硬件除法器操作流程	685

表集

表 1-1 NuMicro® NUC029 系列差异列表.....	15
表 3-1 缩写表.....	20
表 4-1 NuMicro® NUC029xAN_xAE 系列选型指南	21
表 5-1 NuMicro® NUC029xAN片上控制器地址空间分配.....	41
表 5-2 NuMicro® NUC029FAE 片上控制器地址空间分配.....	41
表 5-3 异常模型	108
表 5-4 NuMicro® NUC029xAN 系统中断映射向量表	109
表 5-5 NuMicro® NUC029FAE 系统中断映射向量表	110
表 5-6 向量表格式	111
表 5-7 掉电模式控制表	209
表 5-8 NuMicro® NUC029FAE 外设时钟源选择表	231
表 5-9 掉电模式控制	234
表 5-10 Flash存储器地址映射 NUC029xAN.....	249
表 5-11 Flash存储器地址映射 NUC029FAE	249
表 5-12 数据Flash表（NUC029FAE）	252
表 5-13 启动选项	260
表 5-14 ISP Command List ISP 命令列表	264
表 5-15 EBI 时序表	289
表 5-16 输入捕捉模式操作流程	340
表 5-17 看门狗定时器定时溢出间隔周期选择.....	472
表 5-18 窗口看门狗定时器预分频值选择	479
表 5-19 WINCMP 设置限制	480
表 5-20 UART 控制器波特率公式表	491
表 5-21 UART 波特率参数和寄存器设置表	492
表 5-22 UART 控制器中断源和标志列表	493
表 5-23 UART 线控制的字长度和停止位长度的设置.....	494
表 5-24 UART 线控制校验位的设置	494
表 5-25 I ² C 状态码描述.....	553

1 概述

NuMicro® NUC029xAN_xAE 系列是以ARM® Cortex®-M0为内核的32位微控制器，适用于工业控制以及需要多种通信接口、高集成度、高性能、低成本的应用。Cortex®-M0是最新的ARM®嵌入式处理器，拥有32位机的性能以及与传统8位机相当的价格。NuMicro® NUC029xAN_xAE 系列包括5个型号：NUC029LAN, NUC029NAN, NUC029ZAN, NUC029TAN 及 NUC029FAE

NUC029LAN/NUC029NAN/NUC029ZAN/NUC029TAN最高可运行到50MHz, 工作电压2.5V ~ 5.5V, 温度-40°C ~ 85°C；NUC029FAE最高可运行到24MHz, 工作电压2.5V ~ 5.5V, 温度-40°C ~ 105°C；因此，NUC029系列可以支持工业控制及需要高效能CPU的应用；

NUC029LAN/NUC029NAN/NUC029ZAN/NUC029TAN 提供 64K/64K/32Kbyte flash, 4Kbyte data flash, 4Kbyte ISP flash以及4Kbyte SRAM。NUC029FAE 提供 16 Kbytes flash, 可配置大小的 Data Flash (与程序flash共享), 2 Kbytes ISP flash, 及 2Kbytes SRAM.

NUC029集成了一些系统级的外设功能，例如：I/O 口，EBI（外部总线接口），定时器，UART, SPI, I²C, PWM, ADC, WDT(看门狗定时器), WWDT(窗口型看门狗定时器)，模拟比较器，Brown-out Detector（掉电检测器），这可以显著降低外部器件数量，电路板空间以及系统成本。这些外设使得NUC029适用于广泛的应用。

此外，NuMicro® NUC029系列带有ISP（在系统编程）和ICP（在电路编程）功能，以及IAP(在应用中编程) 允许用户无需取下芯片，直接在电路板上对程序存储器进行升级。

项目	NUC029LANNUC029NAN/ NUC029ZAN/NUC029TAN	NUC029FAE
内核	最高 50 MHz	最高 24 MHz
工作温度	-40°C ~ +85°C	-40°C ~ +105°C
硬件除法器	√	-
	支持 PLL 作为时钟源	-
时钟控制	-	支持外部 32.768 kHz 晶振作为时钟源
	√	-
PWM	PWM发生及捕获定时器	增强型 PWM 发生器
ADC	12 位 SAR ADC, 最高 760k SPS (支持单次, 突发, 单周期及连续扫描模式)	10 位 SAR ADC, 最高 300k SPS (支持单次模式)
EBI	√	-
内置温度传感器	√	-

表 1-1 NuMicro® NUC029 系列差异列表

2 特性

- ARM® Cortex®-M0 内核
 - 运行频率可达50MHz
 - 一个 24位系统定时器
 - 支持低功耗睡眠模式
 - 单指令周期32位硬件乘法器
 - 嵌套向量中断控制器NVIC支持32个中断输入，每个中断有4个优先级
 - 支持串行调试（SWD）接口，2个观察点/4个断点
 - 提供硬件除法器(只有NUC029xAN 支持)，支持有符号32-bit 被除数和16-bit 除数
- 操作电压2.5 V 到 5.5 V
- 存储器
 - 16/32/64 KB Flash用于存储用户程序 (APROM)
 - 最大4 KB 用于存储ISP引导代码 (LDROM)
 - 最大 4 KB SRAM 用作内部高速暂存存储器(SRAM)
 - 4 KB Flash用于存储数据(DataFlash) (NUC029xAN)
 - 可配置大小的DataFlash (NUC029FAE)
- 时钟控制
 - 可编程的系统时钟源
 - 22.1184 MHz 内部高速振荡器
 - ◆ 通过外部32.768k Hz晶振(LXT)可以动态校准内部HIRC到22.1184MHz，精度±3%，支持温度范围-40°C 到 105°C (NUC029FAE)
 - 4~24 MHz 外部晶振输入
 - 10 kHz 低功耗振荡器，用于看门狗定时器及睡眠模式的唤醒
 - PLL支持CPU最高运行在50MHz (NUC029xAN)
 - 32.768 kHz外部晶振输入 (LXT)，用于掉电唤醒及系统操作时钟 (NUC029FAE)
- GPIO
 - LQFP/QFN 48管脚封装下最多40个通用 I/O (GPIO)
 - 四种I/O模式:
 - ◆ 准双向模式
 - ◆ 推挽输出模式
 - ◆ 开漏输出模式
 - ◆ 高阻抗输入模式
 - 可选择TTL输入或施密特触发输入
 - I/O管脚可被配置为边沿/电平触发模式的中断源
 - 较强的拉电流驱动能力和灌电流承受能力
 - 可以配置上电复位后的IO模式
- 定时器
 - 最多支持4组32位定时器，每组定时器均带有24位上数计数器和8位预分频器
 - 每个定时器有独立的时钟源
 - 提供四种计数模式：单次，周期，反转及连续计数模式
 - 24位定时器当前值可由定时器数据寄存器 (TDR) 读出
 - 支持事件计数功能，用于计数外部计数管脚输入的事件
 - 24位捕获值可以通过TCAP寄存器读出

- 支持外部捕获管脚，用于计算事件间隔
 - ◆ 外部捕获管脚可以复位24位计数器
 - ◆ 定时器中断信号可以将CPU从空闲/掉电状态唤醒
- 内部模拟比较器输出信号反转可以触发捕获(NUC029xAN)
- 支持定时器间触发模式(NUC029xAN)
- 支持比较器内部信号（CPO0，CPO1）间隔测量(NUC029FAE)
- WDT (看门狗定时器)
 - 多路时钟源选择
 - 支持在掉电模式和休眠模式下唤醒CPU的功能
 - 看门狗定时器溢出时可选择产生中断/系统复位
 - 超时复位延迟周期可以选择 $3/18/130/1026 * \text{WDT_CLK}$ (NUC029xAN)
- WWDT (窗口式看门狗) (NUC029xAN)
 - 11位预分频器，6位向下计数器提供宽范围的窗口选择
- PWM发生器及捕获定时器(NUC029xAN)
 - 最多支持4个16位PWM发生器，提供8路独立PWM输出或4对互补的PWM输出
 - 每个PWM发生器可以单独选择时钟源，时钟分频器，8位时钟预分频器，和死区发生器
 - PWM中断与PWM周期同步
 - 16位捕捉定时器(与PWM定时器共享)支持捕获输入信号的上升沿/下降沿
 - 支持捕获中断
 - 可选内部10K Hz时钟作为PWM时钟源
 - 支持极性反转
 - 支持中心对齐模式
 - 支持定时器duty中断使能
 - 两种PWM周期中断类型选择
 - 两种PWM占空比中断类型选择
 - 周期/占空比触发ADC功能
 - PWM定时器同步触发功能
- 增强型 PWM 发生器 (NUC029FAE)
 - 独立16位PWM占空比控制单元，最多3路输出
 - 支持组/同步/独立/互补模式
 - 支持单次或自动重载模式
 - 支持边沿对齐及中心对齐模式
 - 互补通道支持插入可编程死区
 - 每个输出可独立设置极性
 - 硬件错误刹车保护
 - 支持占空比，周期及刹车中断
 - 支持占空比/周期触发ADC转换
 - 定时器比较事件触发PWM做相位切换
 - 支持比较器事件触发PWM
 - 提供中断累加功能
- UART
 - 最多2路UART
 - 可编程波特率产生器
 - 收发器配有16字节的FIFO
 - 流量控制功能可选(CTS 和 RTS)
 - 支持IrDA(SIR) 功能

- 支持 RS-485 功能
- 支持LIN总线功能 (NUC029xAN)
- SPI
 - 最多支持两个SPI
 - 支持主/从模式
 - 全双工同步数据收发
 - 支持3线SPI功能
 - 数据长度8到32位可变
 - MSB 或 LSB 数据在前
 - Rx接收可以在时钟上升沿或下降沿锁存数据
 - Tx 发送可以在时钟上升沿或下降沿锁存数据
 - 32位传输支持字节暂停模式
 - 4级深度FIFO缓存
 - 可选PLL时钟源(NUC029xAN)
- I²C
 - 最多2路I²C设备
 - 支持主/从模式
 - 主从机间双向数据传送
 - 多主机总线（无中心主机）
 - 总线仲裁，可避免多主机同时传输数据时的冲突
 - 串行时钟同步机制允许设备通过一条总线实现设备间不同位速率的通信
 - 串行时钟同步可用作为握手机制，控制串行数据传输暂停或恢复
 - 可编程时钟允许多种波特率控制
 - 支持7位地址模式
 - 支持多地址识别(4个带屏蔽功能的从机地址)
 - 支持掉电唤醒功能
 - 支持FIFO功能 (NUC029FAE)
- ADC
 - NUC029xAN支持12位 SAR ADC，最高转换率 760k SPS; NUC029FAE支持10位SAR ADC，最高转换率300k SPS
 - 最多支持8路单端模拟输入通道
 - ◆ 或者4路差分模拟输入(NUC029xAN)
 - 4种操作模式 (NUC029FAE只支持单次模式)
 - ◆ 单次模式：A/D转换只在特定的通道上转换一次
 - ◆ 突发模式：A/D转换在特定的通道上连续转换，并将结果保存在FIFO中
 - ◆ 单周期扫描模式：A/D 转换在所有指定通道完成一次转换（从低序号通道到高序号通道）.
 - ◆ 连续扫描模式：A/D 转换连续执行单周期扫描模式直到软件停止A/D转换
 - A/D转换触发事件:
 - ◆ 软件写1到ADST位
 - ◆ 外部管脚信号 (STADC)
 - ◆ PWM触发，可以配置延时启动
 - 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
 - 每个通道都有单独的数据寄存器 (NUC029xAN)
 - 转换结果可以和指定的值相比较，当转换结果和比较寄存器的设定值相匹配时，用户可设定是否产生中断请求
 - 内部温度传感器输出转换(NUC029xAN)
- 模拟比较器

- 最多4路模拟比较器模块
- 外部输入或者内部band-gap电压可以作为模拟比较器的负输入端
- 比较结果改变时产生中断
- 支持掉电唤醒
- EBI (外部总线接口)用于访问外部设备(NUC029LAN/NUC029NAN)
 - 访问空间: 8位模式时64 KB, 16位模式时128KB
 - 支持8位或16位数据宽度
 - 16位数据宽度模式下支持字节写
- ISP (在系统编程) 及 ICP (在电路编程)
- IAP (在应用编程)
- 内置温度传感器, 1°C 分辨率 (NUC029xAN)
- BOD (掉电检测)
 - 可设置4个检测电平: 4.4V/3.7V/2.7V/2.2V
 - 支持掉电中断及复位选项
- 96位唯一ID (UID)
- LVR (低压复位)
 - 门槛电压: 2.0V
- 操作温度:
 - NUC029LAN/NUC029NAN/NUC029ZAN/NUC029TAN: -40°C~85°C
 - NUC029FAE:-40°C~105°C
- 稳定性: EFT > ± 4 KV, ESD HBM pass 4 KV
- 封装:
 - All Green package (RoHS)
 - 48-pin LQFP, 48-pin QFN, 33-pin QFN, 20-pin TSSOP

3 缩写

缩写	描述
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
BOD	Brown-out Detection
EBI	External Bus Interface
FIFO	First In, First Out
FMC	Flash Memory Controller
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	22.1184 MHz Internal High Speed RC Oscillator
HXT	4~24 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)
LXT	32.768 kHz External Low Speed Crystal Oscillator
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
SPS	Samples per Second
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

表 3-1 缩写表

4 编号信息列表与管脚定义

4.1 NuMicro® NUC029xAN_xAE 系列选型指南

Part Number	APROM (KB)	RAM (KB)	Data Flash (KB)	ISP ROM (KB)	I/O	Timer (32-Bit)	Connectivity			Comparator	WDT	WWDT	EBI	PLL	32.768 KHz Crystal Oscillator	ISP/ICP/IAP	Package	Operating Temperature Range(°C)			
							UART	SPI	I ² C												
NUC029LAN	64	4	4	4	40	4	2	2	2	8	8	-	4	√	√	√	-	√	-40 to +85		
NUC029NAN	64	4	4	4	40	4	2	2	2	8	8	-	4	√	√	√	-	√	-40 to +85		
NUC029ZAN	64	4	4	4	24	4	2	1	2	5	5	-	3*	√	√	-	√	-	√	-40 to +85	
NUC029TAN	32	4	4	4	24	4	2	1	2	5	5	-	3*	√	√	-	√	-	√	-40 to +85	
NUC029FAE	16	2	Config.	2	17	2	1	1	1	3	-	4	2**	√	-	-	-	√	√	TSSOP20	-40 to +105

表 4-1 NuMicro® NUC029xAN_xAE 系列选型指南

注:

*: ACMP3 只有正负输入端.

**: ACMP0 只有正负输入端, ACMP1只有正输入端

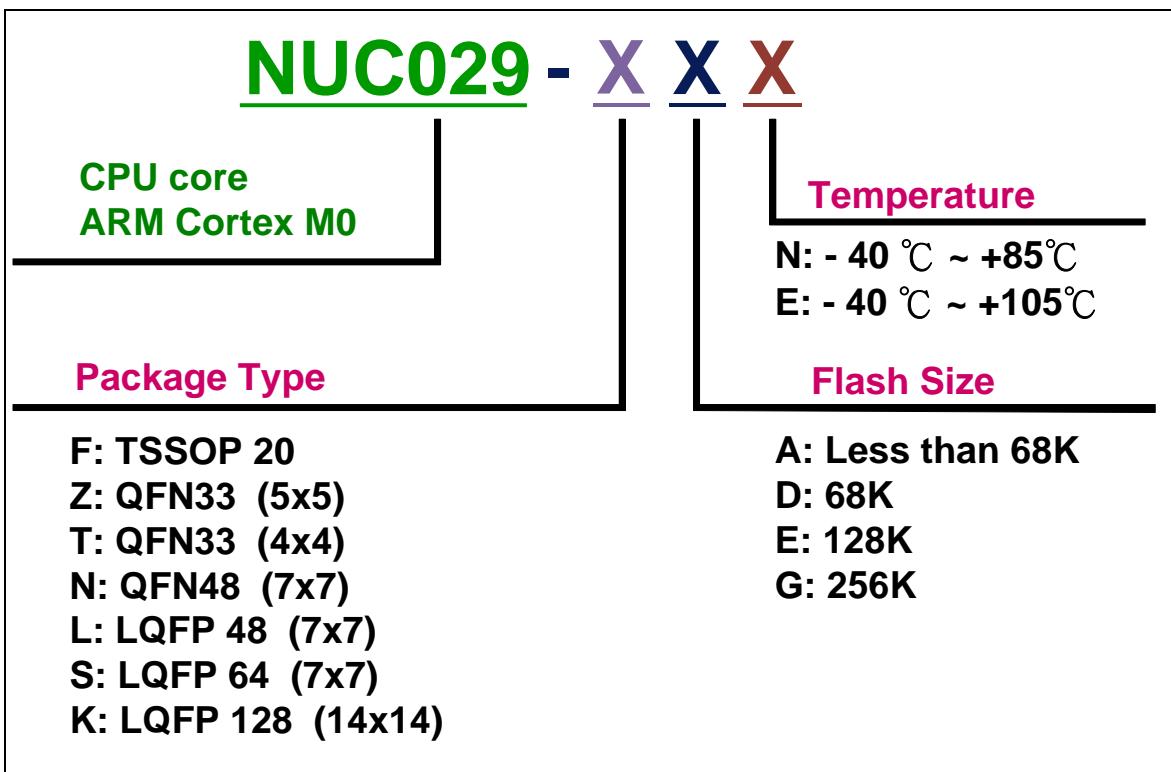


图 4-1NuMicro® NUC029 命名规则

4.2 管脚配置

4.2.1 NuMicro® NUC029xAN_xAE 管脚图

4.2.1.1 NuMicro® NUC029LAN LQFP 48 pin

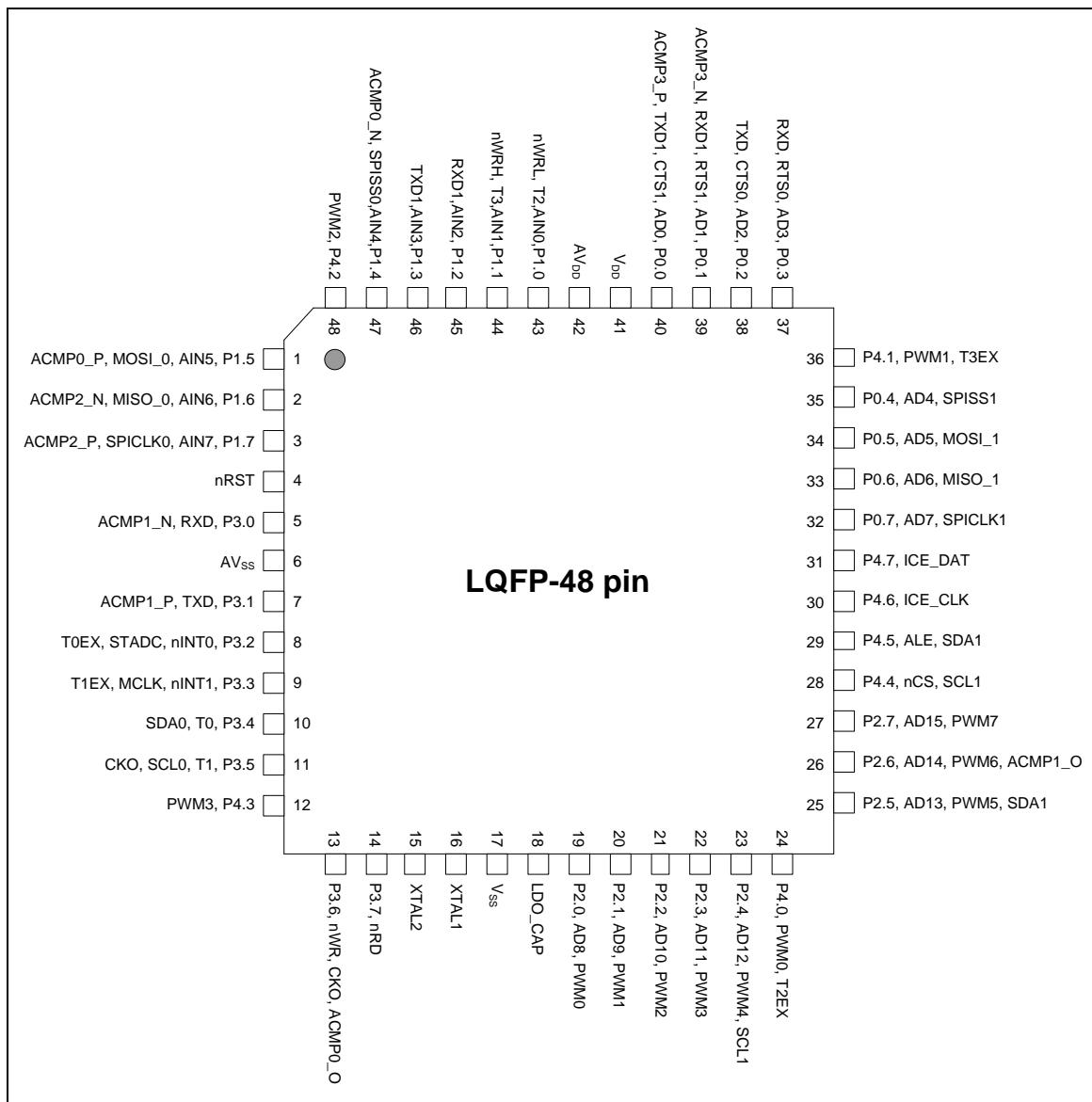


图 4-2 NuMicro® NUC029LAN LQFP 48-pin 图

4.2.1.2 NuMicro® NUC029NAN QFN 48 pin

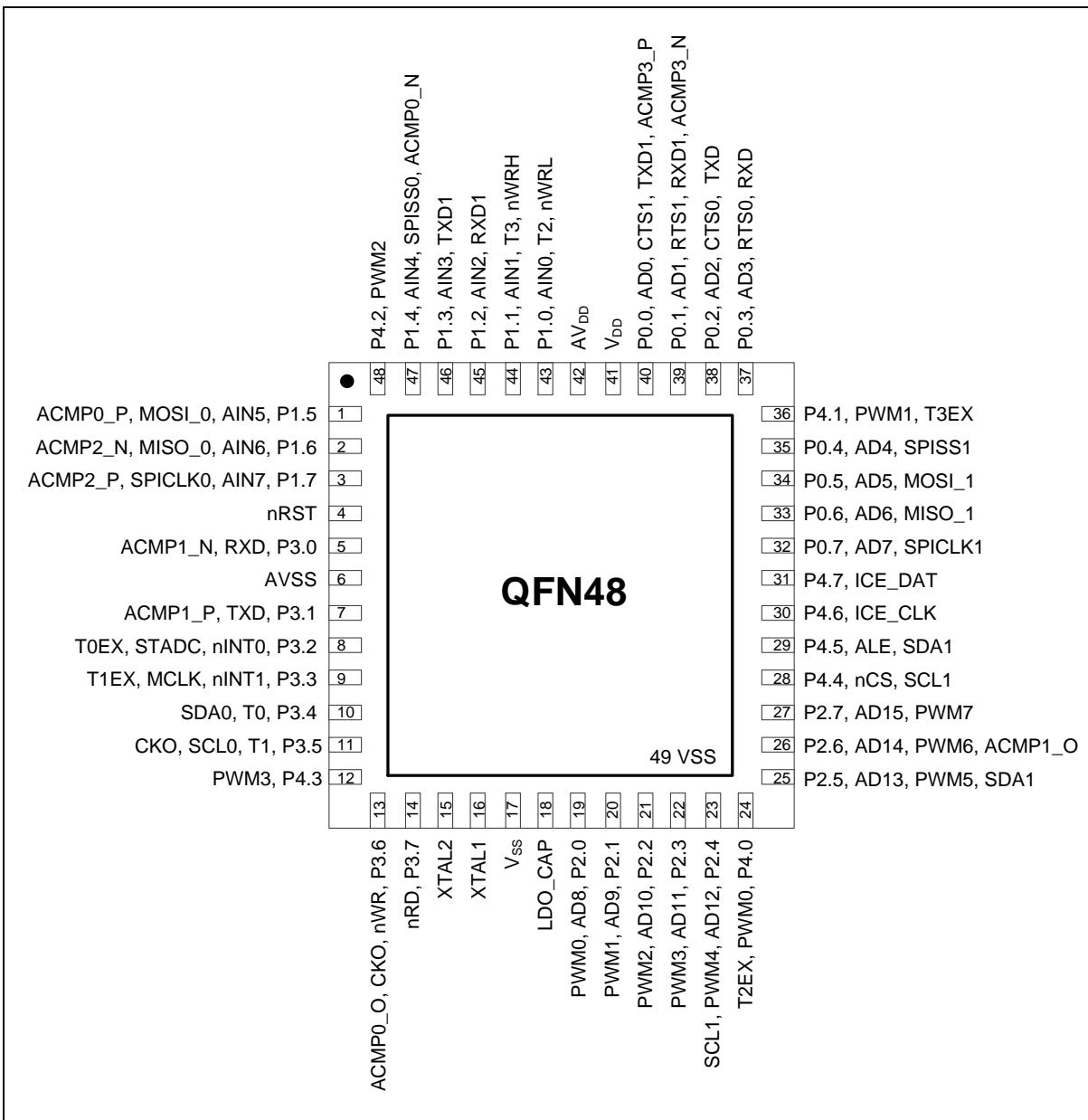


图 4-3 NuMicro® NUC029NAN QFN 48-pin 图

4.2.1.3 NuMicro® NUC029ZAN/NUC029TAN QFN 33 pin

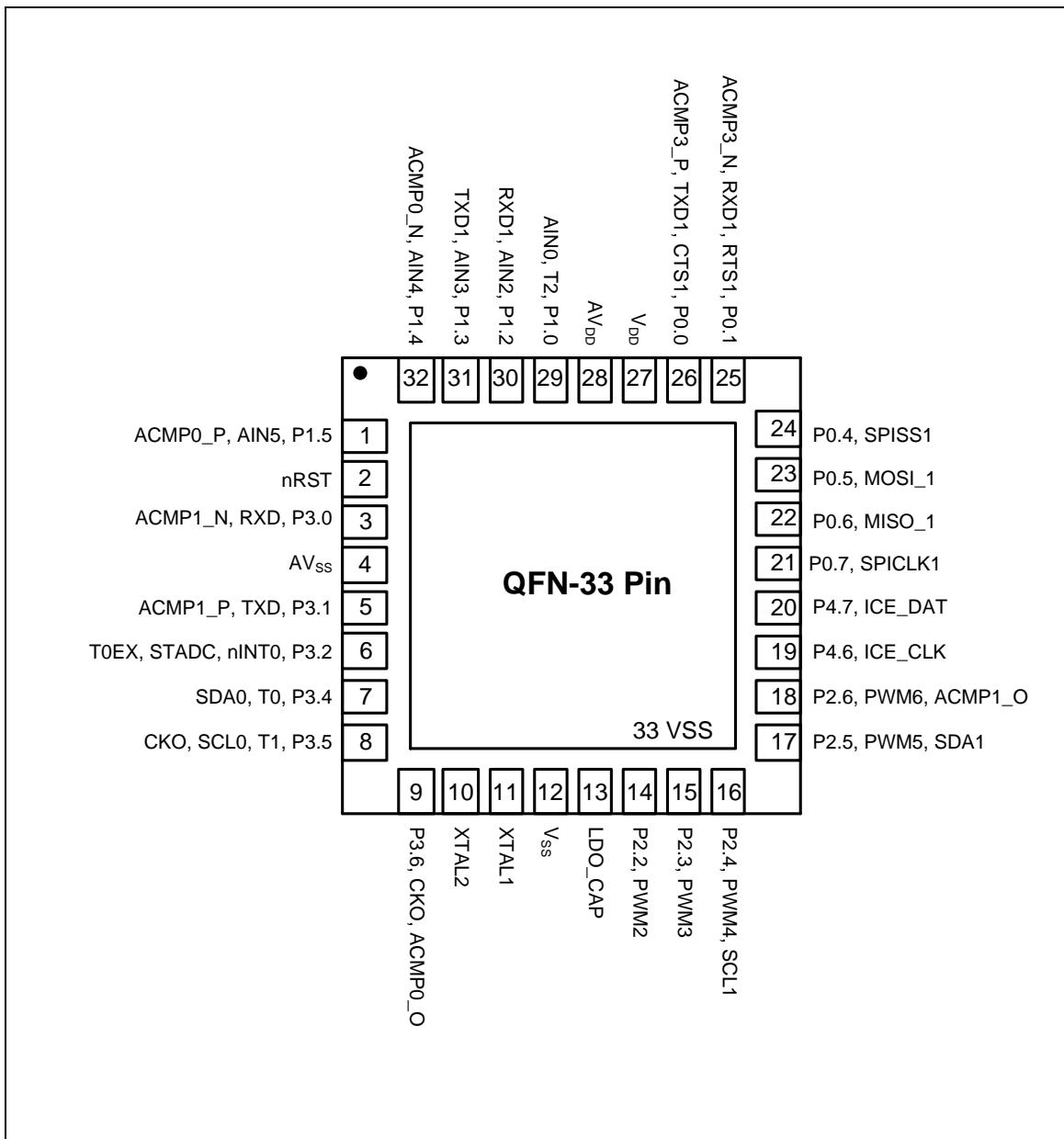


图 4-4 NuMicro® NUC029ZAN/NUC029TAN QFN 33-pin 图

4.2.1.4 NuMicro® NUC029FAE TSSOP 20 pin

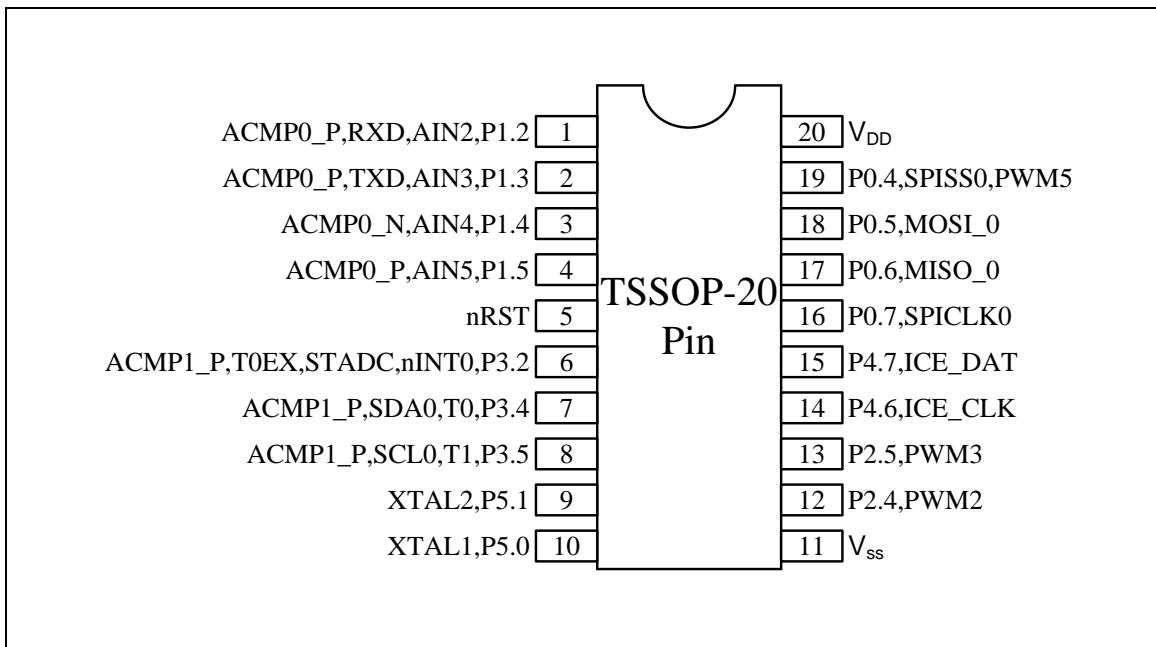


图 4-5 NuMicro® NUC029FAE TSSOP 20-pin 图

4.3 管脚描述

4.3.1 NuMicro® NUC029xAN_xAE 管脚描述

管脚号.		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
1	1	P1.5	I/O	通用数字I/O脚.
		AIN5	AI	ADC5模拟输入
		ACMP0_P	AI	比较器0正输入端.
		MOSI_0	I/O	SPI0 MOSI(主机输出, 从机输入).
2	—	P1.6	I/O	通用数字I/O脚.
		AIN6	AI	ADC6模拟输入.
		MISO_0	I/O	SPI0 MISO (主机输入, 从机输出) .
		ACMP2_N	AI	比较器2负输入端.
3	—	P1.7	I/O	通用数字I/O脚.
		AIN7	AI	ADC7模拟输入.
		SPICLK0	I/O	SPI0 串行时钟.
		ACMP2_P	AI	比较器2正输入端.
4	2	nRST	I (ST)	外部复位输入, 低有效, 内部有上拉; 拉低此管脚复位芯片到初始状态. 注: 建议于 nRST 管脚上加入 10 kΩ 的上拉电阻与 10 uF 的对地电容
5	3	P3.0	I/O	通用数字I/O脚.
		RXD ^[2]	I	UART0数据接收.
		ACMP1_N	AI	比较器1负输入端
6	4	AV _{SS}	AP	模拟地.
7	5	P3.1	I/O	通用数字I/O脚.
		TXD ^[2]	O	UART0数据发送.
		ACMP1_P	AI	比较器1正输入端
8	6	P3.2	I/O	通用数字I/O脚.
		nINT0	I	外部中断0输入脚
		STADC	I	ADC外部触发输入.
		T0EX	I	定时器0外部捕获/复位触发输入脚.
9	—	P3.3	I/O	通用数字I/O脚.
		nINT1	I	外部中断1输入脚.

管脚号.		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
		MCLK	O	EBI 外部时钟输出
		TIEX	I	定时器1外部捕获/复位触发输入脚.
10	7	P3.4	I/O	通用数字I/O脚.
		T0	I/O	定时器0外部事件计数输入
		SDA0	I/O	I2C0 数据输入/输出脚.
11	8	P3.5	I/O	通用数字I/O脚.
		T1	I/O	定时器1外部事件计数输入脚.
		SCL0	I/O	I2C0 时钟管脚.
		CKO ^[2]	O	除频输出管脚
12	—	P4.3	I/O	通用数字I/O脚.
		PWM3 ^[2]	I/O	PWM3 输出/捕获输入.
13	9 —	P3.6	I/O	通用数字I/O脚.
		CKO ^[2]	O	除频输出管脚
		ACMP0_O	O	模拟比较器0输出脚.
		nWR	O	EBI 写使能脚
14	—	P3.7	I/O	通用数字I/O脚.
		nRD	O	EBI 读使能脚
15	10	XTAL2	O	外部4~24 MHz高速晶振输出.
16	11	XTAL1	I (ST)	外部4~24 MHz高速晶振输入.
17	12	V _{ss}	P	数字电路地
	33			
18	13	LDO_CAP	P	LDO 输出脚.
19	—	P2.0	I/O	通用数字I/O脚.
		AD8	I/O	EBI 地址/数据总线第8位
		PWM0 ^[2]	I/O	PWM0输出/捕获输入
20	—	P2.1	I/O	通用数字I/O脚.
		AD9	I/O	地址/数据总线第9位
		PWM1 ^[2]	I/O	PWM1输出/捕获输入.
21	14	P2.2	I/O	通用数字I/O脚.

管脚号.		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
		PWM2 ^[2]	I/O	PWM2输出/捕获输入.
		— AD10	I/O	EBI 地址/数据总线第10位
22	15	P2.3	I/O	通用数字I/O脚.
		PWM3 ^[2]	I/O	PWM3 输出/捕获输入.
		— AD11	I/O	EBI 地址/数据总线第11位
23	16	P2.4	I/O	通用数字I/O脚.
		PWM4	I/O	PWM4 输出/捕获输入
		SCL1 ^[2]	I/O	I2C1 时钟.
		— AD12	I/O	EBI 地址/数据总线第12位
24	—	P4.0	I/O	通用数字I/O脚.
		PWM0 ^[2]	I/O	PWM0输出/捕获输入
		T2EX	I	定时器2外部捕获/复位触发输入
25	17	P2.5	I/O	通用数字I/O脚.
		PWM5	I/O	PWM5输出/捕获输入
		SDA1 ^[2]	I/O	I2C1 数据管脚.
		— AD13	I/O	EBI 地址/数据总线第13位
26	18	P2.6	I/O	通用数字I/O脚.
		PWM6	I/O	PWM6 输出/捕获输入
		ACMP1_O	O	模拟比较器输出脚
		— AD14	I/O	EBI 地址/数据总线第14位
27	—	P2.7	I/O	通用数字I/O脚.
		AD15	I/O	EBI 地址/数据总线第15位
		PWM7	I/O	PWM7输出/捕获输入
28	—	P4.4	I/O	通用数字I/O脚.
		nCS	O	EBI片选使能
		SCL1 ^[2]	I/O	I2C1 时钟.
29	—	P4.5	I/O	通用数字I/O脚.
		ALE	O	EBI地址锁存使能
		SDA1 ^[2]	I/O	I2C1 数据管脚.
30	19	P4.6	I/O	通用数字I/O脚.

管脚号.		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
		ICE_CLK	I	串行调试时钟 注: 建议于 ICE_CLK 管脚上加入 100 kΩ 的上拉电阻
31	20	P4.7	I/O	通用数字I/O脚.
		ICE_DAT	I/O	串行调试数据 注: 建议于 ICE_DAT 管脚上加入 100 kΩ 的上拉电阻
32	21	P0.7	I/O	通用数字I/O脚.
		SPICLK1	I/O	SPI1 时钟.
	—	AD7	I/O	EBI 地址/数据总线第7位
33	22	P0.6	I/O	通用数字I/O脚.
		MISO_1	I/O	SPI1 MISO (主机输入, 从机输出)
	—	AD6	I/O	EBI 地址/数据总线第6位
34	23	P0.5	I/O	通用数字I/O脚.
		MOSI_1	I/O	SPI1 MOSI (主机输出, 从机输入).
	—	AD5	I/O	EBI 地址/数据总线第5位
35	24	P0.4	I/O	通用数字I/O脚.
		SPISS1	I/O	SPI1 从机选择.
	—	AD4	I/O	EBI 地址/数据总线第4位
36	—	P4.1	I/O	通用数字I/O脚.
		PWM1 ^[2]	I/O	PWM1输出/捕获输入.
		T3EX	I	定时器3外部捕获/复位触发
37	—	P0.3	I/O	通用数字I/O脚.
		AD3	I/O	EBI 地址/数据总线第3位
		RTS0	O	UART0请求发送.
		RXD ^[2]	I	UART0数据接收.
38	—	P0.2	I/O	通用数字I/O脚.
		AD2	I/O	EBI 地址/数据总线第2位
		CTS0	I	UART0清除发送脚.
		TXD ^[2]	O	UART0数据发送.
39	25	P0.1	I/O	通用数字I/O脚.
		RTS1	O	UART1请求发送.

管脚号.		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
40	26	RXD1 ^[2]	I	UART1数据接收.
		ACMP3_N	AI	比较器3负端输入
		AD1	I/O	EBI 地址/数据总线第1位
41	27	P0.0	I/O	通用数字I/O脚.
		CTS1	I	UART1清除发送.
		TXD1 ^[2]	O	UART1数据发送.
		ACMP3_P	AI	比较器3正端输入
		AD0	I/O	EBI 地址/数据总线第0位
42	28	V _{DD}	P	数字电源, 用于I/O口, 内部LDO, PLL及数字电路
43	29	AV _{DD}	AP	模拟电源用于内部模拟电路
		P1.0	I/O	通用数字I/O脚.
		AIN0	AI	ADC0 模拟输入.
		T2	I/O	定时器2 外部事件计数输入.
44	30	nWRl	O	EBI 低字节写使能
		P1.1	I/O	通用数字I/O脚.
		AIN1	AI	ADC1 模拟输入.
		T3	I/O	定时器3 外部事件计数输入.
45	31	nWRH	O	EBI 高字节写使能
		P1.2	I/O	通用数字I/O脚.
		AIN2	AI	ADC2 模拟输入.
46	32	RXD1 ^[2]	I	UART1数据接收.
		P1.3	I/O	通用数字I/O脚.
		AIN3	AI	ADC3 模拟输入.
47	33	TXD1 ^[2]	O	UART1数据发送.
		P1.4	I/O	通用数字I/O脚.
		AIN4	AI	ADC4 模拟输入.
		ACMP0_N	AI	模拟比较器0负端输入
48	—	SPISS0	I/O	SPI0 从机选择.
		P4.2	I/O	通用数字I/O脚.

管脚号:		管脚名称	管脚类型	描述
LQFP/QFN 48-pin	QFN 33-pin			
		PWM2 ^[2]	I/O	PWM2输出/捕获输入.

注1: 引脚类型 I = 数字输入, O = 数字输出; AI = 模拟输入; P = 电源; AP = 模拟电源; ST = 施密特触发

注2: PWM0 ~ PWM3, RXD, TXD, RXD1, TXD1, SCL1, SDA1 及 CKO可以定义到不同的管脚, 但是一个功能只能赋给一个管脚, 例如: 软件不能将RXD同时赋给P0.3, P3.0

管脚号.	管脚名称	管脚类型	描述
TSSOP 20-pin			
1	P1.2	I/O	通用数字I/O脚.
	AIN2	AI	ADC2 模拟输入.
	RXD	I	UART0数据接收.
	ACMP0_P	AI	比较器0正输入端.
2	P1.3	I/O	通用数字I/O脚.
	AIN3	AI	ADC3 模拟输入.
	TXD	O	UART0数据发送.
	ACMP0_P	AI	比较器0正输入端.
3	P1.4	I/O	通用数字I/O脚.
	AIN4	AI	ADC4 模拟输入.
	ACMP0_N	AI	比较器0负输入端
4	P1.5	I/O	通用数字I/O脚.
	AIN5	AI	ADC5模拟输入
	ACMP0_P	AI	比较器0正输入端.
	AV _{ss}	AP	模拟地.
5	nRST	I (ST)	外部复位输入，低有效，内部有上拉;拉低此管脚复位芯片到初始状态. 注: 建议于 nRST 管脚上加入 10 kΩ 的上拉电阻与 10 uF 的对地电容
6	P3.2	I/O	通用数字I/O脚.
	nINT0	I	外部中断0输入脚
	STADC	I	ADC外部触发输入.
	T0EX	I	定时器0外部捕获/复位触发输入脚.
	ACMP1_P	AI	比较器1正输入端
7	P3.4	I/O	通用数字I/O脚.
	T0	I/O	定时器0外部事件计数输入
	SDA0	i/O	I2C0 数据输入/输出脚.
	ACMP1_P	AI	比较器1正输入端
8	P3.5	I/O	通用数字I/O脚.
	T1	I/O	定时器1外部事件计数输入脚.
	SCL0	I/O	I2C0 时钟管脚.
	ACMP1_P	AI	比较器1正输入端

管脚号.	管脚名称	管脚类型	描述
TSSOP 20-pin			
9	P5.1	I/O	通用数字I/O脚.
	XTAL2	O	外部4~24 MHz高速晶振输出.
10	P5.0	I/O	通用数字I/O脚.
	XTAL1	I (ST)	外部4~24 MHz高速晶振输入.
11	V _{ss}	P	数字电路地
12	P2.4	I/O	通用数字I/O脚.
	PWM2	I/O	PWM2 输出.
13	P2.5	I/O	通用数字I/O脚.
	PWM3	I/O	PWM3 输出.
14	P4.6	I/O	通用数字I/O脚.
	ICE_CLK	I	串行调试时钟 注: 建议于 ICE_CLK 管脚上加入100 kΩ 的上拉电阻
15	P4.7	I/O	通用数字I/O脚.
	ICE_DAT	I/O	串行调试数据 注: 建议于 ICE_DAT 管脚上加入100 kΩ 的上拉电阻
16	P0.7	I/O	通用数字I/O脚.
	SPICLK0	I/O	SPI0 串行时钟.
17	P0.6	I/O	通用数字I/O脚.
	MISO_0	I/O	SPI0 MISO (主机输入, 从机输出).
18	P0.5	I/O	通用数字I/O脚.
	MOSI_0	I/O	SPI0 MOSI(主机输出, 从机输入).
19	P0.4	I/O	通用数字I/O脚.
	SPISS0	I/O	SPI1 从机选择.
	PWM5	I/O	PWM5 输出.
20	V _{DD}	P	数字电源, 用于I/O口, 内部LDO,PLL及数字电路.

注1: 管脚类型 I = 数字输入, O = 数字输出; AI = 模拟输入; P = 电源管脚; AP = 模拟电源; ST = 施密特触发

5 功能描述

5.1 ARM® Cortex®-M0 内核

Cortex®-M0 处理器是一个可配置，具有多级流水线的32位RISC处理器。它拥有一个 AMBA AHB-Lite 接口并包含NVIC组件，同时有可选的硬件调试功能。该处理器可以执行Thumb 指令，并与其他Cortex®-M 系列处理器兼容。该处理器支持两种模式——Thread模式和Handler模式。异常时系统进入Handler模式，异常返回只能在Handler模式下执行。系统复位及异常返回后均可进入 Thread模式。图 5-1展示处理器的功能控制器

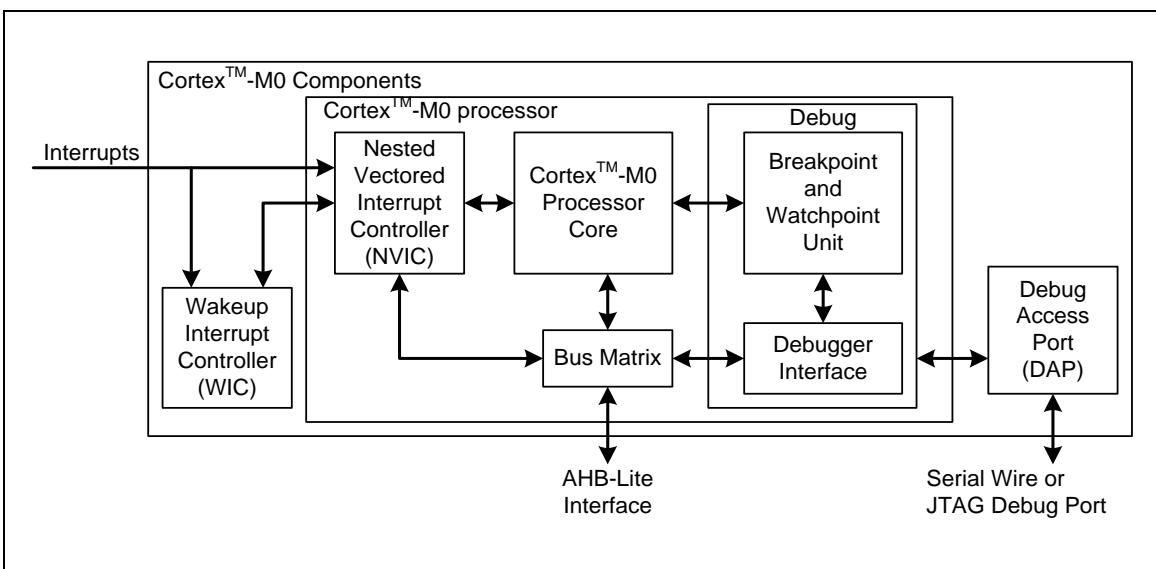


图 5-1 功能控制框图

内核提供下面一些组件及特性：

- 低门数处理器：
 - ARMv6-M Thumb® 指令集
 - Thumb-2 技术
 - ARMv6-M 兼容24位系统定时器
 - 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异常兼容模式：
ARMv6-M的C应用程序二进制接口(C-ABI)异常兼容模式允许用户在中断处理中使用纯C函数。

- 使用**WFI**指令进入低功耗空闲模式，使用**WFE**指令或从中断退出低功耗空闲模式
- NVIC
 - 32个外部中断，每个中断有4级优先级
 - 专用的不可屏蔽中断 (**NMI**)
 - 同时支持电平和脉冲触发中断
 - 支持中断唤醒控制器(**WIC**)，提供极低功耗空闲模式
- 调试支持
 - 四个硬件断点
 - 两个观察点
 - 用于非侵入代码分析的程序计数采样寄存器(**PCSR**)
 - 单步和向量捕捉能力
- 总线接口
 - 为所有的系统接口及存储器提供简单集成的单一的32位AMBA-3 AHB-Lite 系统接口
 - 支持**DAP**（调试使用端口）单一的32位从机端口

5.2 系统管理器

5.2.1 概述

系统管理包括如下章节

- 系统复位
- 系统电源架构
- 系统存储器映射
- 用于产品ID, 芯片复位及片上控制器复位, 多功能管脚控制的系统管理寄存器
- 系统定时器 (SysTick)
- 嵌套向量中断控制器(NVIC)
- 系统控制寄存器

5.2.2 系统复位

系统复位可以由如下事件触发, 这些复位事件标志可以通过寄存器**RSTRC**读出.

- 硬件复位
 - 上电复位(POR)
 - 复位脚 (nRST) 上有低电平
 - 看门狗定时溢出复位(WDT)
 - 低电压复位(LVR)
 - 欠压检测复位(BOD)
- 软件复位
 - MCU复位 - SYSRESETREQ(AIRCR[2])
 - CPU单次复位 - CPU_RST(IPRSTC1[1])
 - 芯片单次复位 - CHIP_RST(IPRSTC1[0])

注: MCU 复位和CPU复位之后, ISPCON.BS 的值保持不变。.

5.2.3 系统电源分布

该芯片的电源架构分为3个部分：

- 来自AV_{DD}和AV_{SS}的模拟电源，为模拟元件提供工作电压。AV_{DD}必须等于V_{DD}以避免漏电
- 来自V_{DD}和V_{SS}的数字电源，为内部稳压器和I/O引脚提供电压，内部稳压器向数字操作提供稳定的1.8V电压。
- 内部稳压器内置输出电容（NUC029FAE only）

内部稳压器（LDO_CAP）的输出，需要在相应管脚附近接一颗电容。模拟电源（AV_{DD}）应该和数字电源（V_{DD}）电压相同图 5-2 显示了 NuMicro® NUC029xAN 系列的电源分布，图 5-3 显示了 NuMicro® NUC029FAE 系列的电源分布：

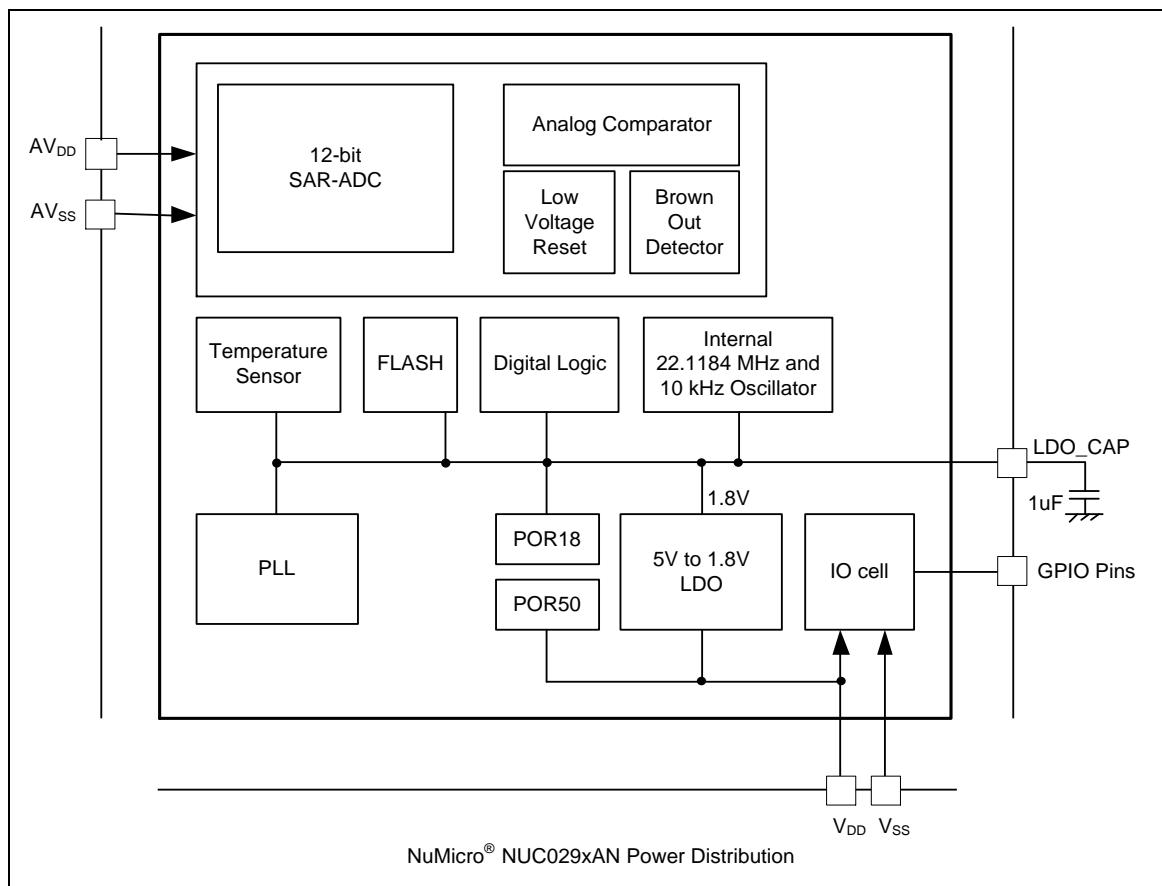


图 5-2 NuMicro® NUC029xAN 系列电源分布图

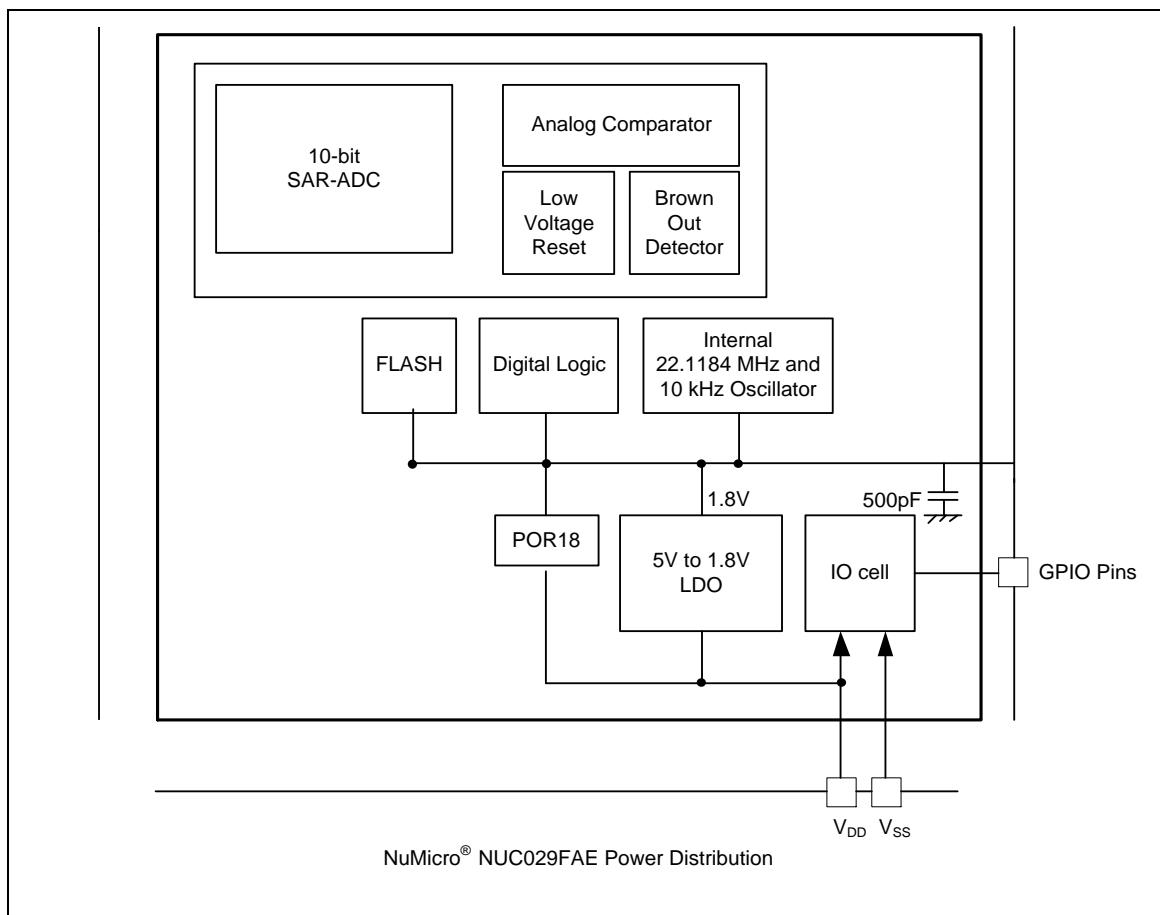


图 5-3 NuMicro® NUC029FAE 系列电源分布图

5.2.4 系统存储器映射

NuMicro® NUC029系列提供4G字节的寻址空间。每个片上模块存储器地址分配情况如下表所示。详细的寄存器定义和寻址空间以及编程细节,将在后续的各个片上外设描述章节里描述。NuMicro® NUC029系列仅支持小端数据格式。.

地址空间	标志	控制器
Flash 和 SRAM 内存空间		
0x0000_0000 – 0x0000_FFFF	FLASH_BA	FLASH内存空间(64KB)
0x2000_0000 – 0x2000_0FFF	SRAM_BA	SRAM内存空间(4KB)
EBI 空间 (0x6000_0000 ~ 0x6001_FFFF) (NUC029LAN/NUC029NAN Only)		
0x6000_0000 – 0x6001_FFFF	EBI_BA	外部存储器空间(128 KB)
AHB控制器空间(0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_BA	系统全局控制寄存器
0x5000_0200 – 0x5000_02FF	CLK_BA	时钟控制寄存器
0x5000_0300 – 0x5000_03FF	INT_BA	多路中断控制寄存器
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO (P0~P4) 控制寄存器
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash 存储器控制寄存器
0x5001_0000 – 0x5001_03FF	EBI_CTL_BA	EBI 控制寄存器(NUC029LAN/NUC029NAN Only)
0x5001_4000 – 0x5001_7FFF	HDIV_BA	硬件除法器(NUC029xAN Only)
APB控制器空间(0x4000_0000 ~ 0x400F_FFFF)		
0x4000_4000 – 0x4000_40FF	WDT_BA	看门狗控制寄存器
0x4000_4100 – 0x4000_7FFF	WWDT_BA	窗口看门狗控制寄存器(NUC029xAN only)
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0/Timer1 控制寄存器
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I2C0接口控制寄存器
0x4003_0000 – 0x4003_3FFF	SPI0_BA	带主/从功能的SPI0控制寄存器
0x4003_4000 – 0x4003_7FFF	SPI1_BA	带主/从功能的SPI1 控制寄存器
0x4004_0000 – 0x4004_3FFF	PWMA_BA	PWM0/1/2/3 控制寄存器
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0控制寄存器
0x400D_0000 – 0x400D_3FFF	ACMP01_BA	模拟比较器0/1控制寄存器
0x400E_0000 – 0x400E_FFFF	ADC_BA	模数转换器(ADC)控制寄存器
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2/Timer3控制寄存器
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I2C1接口控制寄存器(Nuc029xAN only))
0x4014_0000 – 0x4014_3FFF	PWMB_BA	PWM4/5/6/7控制寄存器
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 控制寄存器
0x401D_0000 – 0x401D_3FFF	ACMP23_BA	模拟比较器2/3控制寄存器
系统控制器空间(0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SYST_BA	系统 定时器控制寄存器

0xE000_E100 – 0xE000_ECFF	NVIC_BA	外部中断控制器控制寄存器
0xE000_ED00 – 0xE000_ED8F	SCB_BA	系统控制寄存器

表 5-1 NuMicro® NUC029xAN 片上控制器地址空间分配

地址空间	标志	控制器
Flash和 SRAM 内存空间		
0x0000_0000 – 0x0000_3FFF	FLASH_BA	FLASH内存空间(16 KB)
0x2000_0000 – 0x2000_0FFF	SRAM_BA	SRAM内存空间(2 KB)
AHB控制器空间(0x5000_0000 ~ 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_BA	系统全局控制寄存器
0x5000_0200 – 0x5000_02FF	CLK_BA	时钟控制寄存器
0x5000_0300 – 0x5000_03FF	INT_BA	多路中断控制寄存器
0x5000_4000 – 0x5000_7FFF	GP_BA	GPIO (P0~P5) 控制寄存器
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash 存储器控制寄存器
APB控制器空间 (0x4000_0000 ~ 0x401F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	看门狗控制寄存器
0x4001_0000 – 0x4001_3FFF	TMR_BA	Timer0/Timer1 控制寄存器
0x4002_0000 – 0x4002_3FFF	I2C_BA	I2C接口控制寄存器
0x4003_0000 – 0x4003_3FFF	SPI_BA	带主/从功能的SPI控制寄存器
0x4004_0000 – 0x4004_3FFF	PWMA_BA	PWM 控制寄存器
0x4005_0000 – 0x4005_3FFF	UART_BA	UART控制寄存器
0x400D_0000 – 0x400D_3FFF	ACMP_BA	模拟比较器控制寄存器
0x400E_0000 – 0x400E_3FFF	ADC_BA	模数转换器(ADC)控制寄存器
系统控制器空间 (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	外部中断控制器控制寄存器
0xE000_ED00 – 0xE000_ED8F	SCB_BA	系统控制块寄存器

表 5-2 NuMicro® NUC029FAE 片上控制器地址空间分配

5.2.5 系统存储器映射表

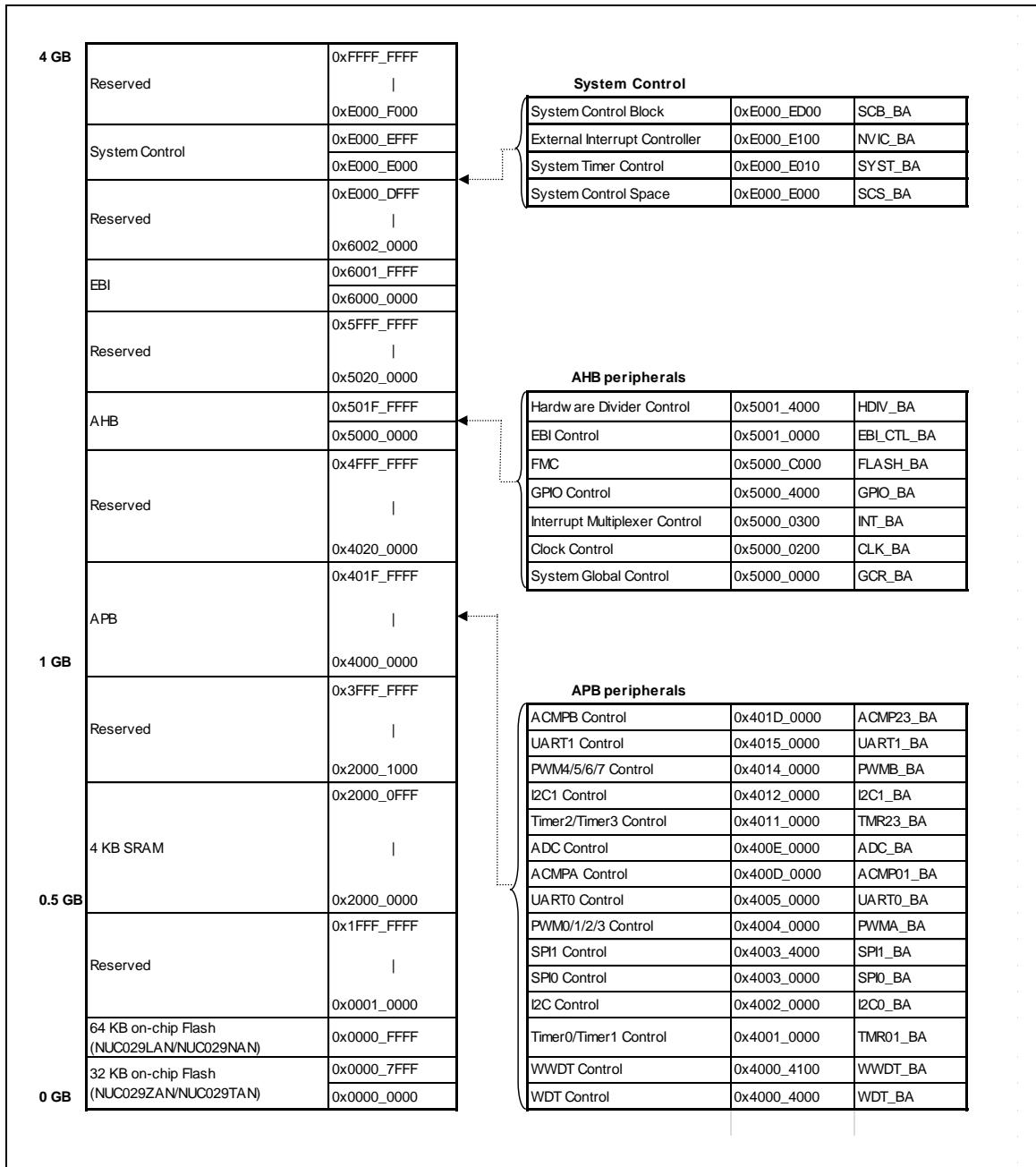


图 5-4 NuMicro® NUC029xAN 系统存储器映射

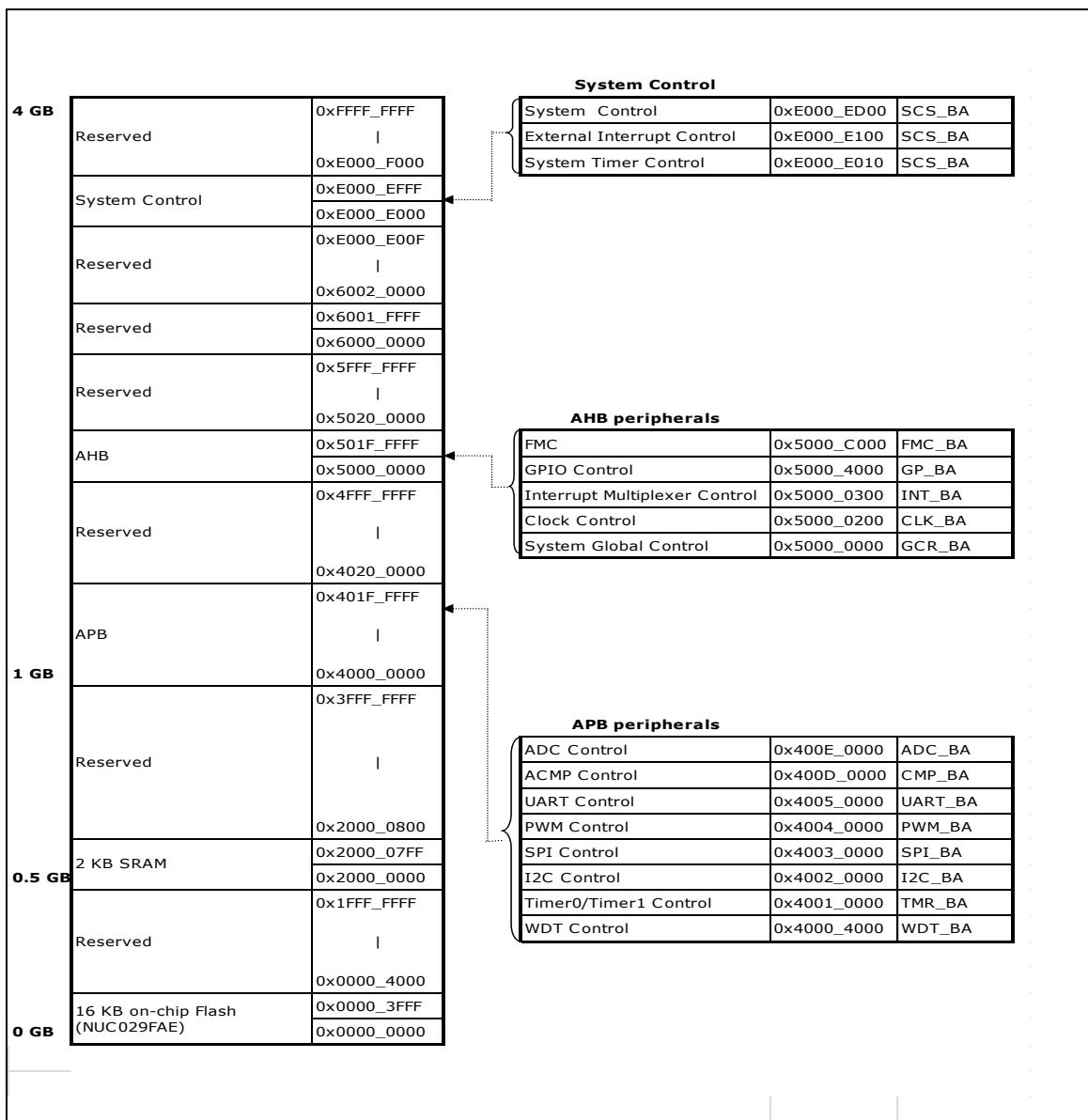


图 5-5 NuMicro® NUC029FAE 系统存储器映射

5.2.6 NUC029xAN系统管理器控制寄存器映射

R: 只读, W: 只写, R/W: 读和写

寄存器	偏移量	R/W	描述	复位后的值
GCR 基地址:				
GCR_BA = 0x5000_0000				
PDID	GCR_BA+0x00	R	设备ID寄存器	0xFFFF_FFFF
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	0x0000_0000
IPRSTC2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000
BODCR	GCR_BA+0x18	R/W	欠压检测控制寄存器	0x0000_008X
TEMPCR	GCR_BA+0x1C	R/W	温度传感器控制	0x0000_0000
PORCR	GCR_BA+0x24	R/W	上电复位控制寄存器	0x0000_00XX
P0_MFP	GCR_BA+0x30	R/W	P0 复用功能和输入类型控制寄存器	0x0000_0000
P1_MFP	GCR_BA+0x34	R/W	P1 复用功能和输入类型控制寄存器	0x0000_0000
P2_MFP	GCR_BA+0x38	R/W	P2 复用功能和输入类型控制寄存器	0x0000_0000
P3_MFP	GCR_BA+0x3C	R/W	P3 复用功能和输入类型控制寄存器	0x0000_0000
P4_MFP	GCR_BA+0x40	R/W	P4 输入类型控制寄存器	0x0000_00C0
REGWRPROT	GCR_BA+0x100	R/W	寄存器写保护控制寄存器	0x0000_0000

位设备ID寄存器(PDID)

寄存器	偏移量	R/W	描述	复位后的值
PDID	GCR_BA+0x00	R	设备ID寄存器	0xFFFF_FFFF ^[1]

[1] 每个型号的设备复位后都有一个唯一的默认ID.

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	设备ID码 该寄存器为设备的器件号码。软件可以通过读取该寄存器来识别所使用的器件。

NuMicro® NUC029xAN	产品设备识别码.
NUC029LAN	0x0029_5A00
NUC029NAN	0x0029_5A06
NUC029ZAN	0x0029_5A03
NUC029TAN	0x0029_5804

系统复位源寄存器(RSTSRC)

该寄存器提供一些信息用于软件识别上次引起芯片复位的复位源。.

寄存器	偏移量	R/W	描述	复位后的值
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX

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
RSTS_CPU	保留	RSTS MCU	RSTS_BOD	RSTS_LVR	RSTS_WDT	RSTS_RESET	RSTS_POR

位	描述	
[31:8]	保留	保留
[7]	RSTS_CPU	CPU复位标志 如果软件将 CPU_RST (IPRSTCR1[1])置为“1”导致Cortex-M0 CPU 内核和FLASH控制器(FMC) 复位, RSTS_CPU 标志将由硬件置位。. 1= 软件置CPU_RST为1时, 导致Cortex-M0 CPU 内核与FMC复位。 0= CPU无复位 注: 向该位写1清除为零。
[6]	保留	保留
[5]	RSTS MCU	MCU复位标志 RSTS MCU 由来自MCU Cortex_M0 核的“复位信号”置位, 以表示当前的复位源。 1= MCU Cortex_M0 在软件向SYSRESTREQ(AIRCR[2], AIRCR是应用中断和复位控制寄存器, 地址 = 0xE000ED0C)写1时, 发出复位信号以复位系统 0= MCU无复位 注: 向该位写1清除为零。
[4]	RSTS_BOD	欠压检测复位标志 RSTS_BOD标志位由欠压检测模块的“复位信号”置1, 用于表示当前复位源。 1: 欠压检测模块发出复位信号使系统复位。 0: BOD无复位 注: 向该位写1清除为零。
[3]	RSTS_LVR	低压检测复位标志 RSTS_LVR标志位由低压复位模块的“复位信号”置1, 用于表示当前复位源。

		1: 低压 LVR 模块发出复位信号使系统复位。 0: LVR无复位 注: 向该位写1清除为零.
[2]	RSTS_WDT	看门狗复位标志 RSTS_WDT 标志位由看门狗模块的"复位信号"置1, 用于说明当前复位源。 1: 看门狗模块发出复位信号使系统复位. 0: 没有看门狗复位信号 注: 向该位写1清除为零
[1]	RSTS_RESET	复位引脚复位标志 RSTS_RESET 标志位由nRST脚的"复位信号"置1, 用于说明当前复位源。 1: nRST脚上发出复位信号使系统复位. 0: nRST脚没有复位信号 注: 向该位写1清除为零.
[0]	RSTS_POR	上电复位标志 RSTS_POR 标志位由POR模块的"复位信号"或者写CHIP_RST (IPRSTC1[0])置1, 用于说明当前的复位源。 1: 上电复位POR发出复位信号或者CHIP_RST置1使系统复位. 0: 没有POR复位信号或者CHIP_RST没有被写1 注: 向该位写1清除为零.

外设复位控制寄存器1 (IPRSTC1)

寄存器	偏移量	R/W	描述	复位后的值
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	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			HDIV_RST	EBI_RST	保留	CPU_RST	CHIP_RST

位	描述	
[31:5]	保留	保留
[4]	HDIV_RST	HDIV 控制器复位 (写保护) 将该位置为1将产生一个复位信号送到硬件除法器。用户需要再将该位置0才能退出复位状态 0 = 硬件除法器正常工作 1 = 复位硬件除法器。 注: 该位是写保护的, 程序需要依次向地址0x5000_0100写入下列数据“59h”, “16h”, 和“88h”来关闭保护功能. 参考寄存器REGWRPROT, 地址 GCR_BA+0x100
[3]	EBI_RST	EBI 控制器复位 (写保护) 设置该位为“1”将产生复位信号到EBI。用户需要再将该位置“0”才能退出复位状态 0= EBI 控制器正常工作 1= EBI 控制器复位。 注: 该位是写保护的, 修改该位时, 需要依次向0x5000_0100写入"59h", "16h", "88h"解除寄存器保护。参考寄存器 REGWRPROT, 地址GCR_BA + 0x100.
[2]	保留	保留
[1]	CPU_RST	处理器内核一次性复位(写保护) 该位置1, CPU内核和Flash存储控制器复位。两个时钟周期后, 该位自动清零。 0 = 正常 1 = 复位CPU 注: 该位是写保护的, 修改该位时, 需要依次向0x5000_0100写入"59h", "16h", "88h"解除寄存器保护。参考寄存器 REGWRPROT, 地址GCR_BA + 0x100
[0]	CHIP_RST	芯片一次性复位 (写保护) 该位置1, 整个芯片复位, 包括CPU内核和所有外设均复位。, 两个时钟周期后, 该位自动清零。 CHIP_RST与上电复位一样, 所有的芯片模块都复位, 芯片缺省设置从CONFIG0/CONFIG1重

		<p>新加载 0：正常 1：复位芯片 注：该位是写保护的，修改该位时，需要依次向0x5000_0100写入“59h”，“16h”，“88h”解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100</p>
--	--	---

外设复位控制寄存器2 (IPRSTC2)

置”1”这些位将会产生异步复位信号给相应的IP。用户需要清零相应位来使IP离开复位状态。

寄存器	偏移量	R/W	描述	复位后的值
IPRSTC2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
保留			ADC_RST	保留			
23	22	21	20	19	18	17	16
ACMP23_RST	ACMP01_RST	PWM47_RST	PWM03_RST	保留		UART1_RST	UART0_RST
15	14	13	12	11	10	9	8
保留		SPI1_RST	SPI0_RST	保留		I2C1_RST	I2C0_RST
7	6	5	4	3	2	1	0
保留		TMR3_RST	TMR2_RST	TMR1_RST	TMR0_RST	GPIO_RST	保留

位	描述	
[31:29]	保留	保留
[28]	ADC_RST	ADC控制器复位 “0”: ADC 模块正常工作 “1”: ADC 模块复位
[27:24]	保留	保留
[23]	ACMP23_RST	模拟比较器B控制器复位 0 = 模拟比较器B控制器正常工作 1 = 模拟比较器B控制器复位
[22]	ACMP01_RST	模拟比较器A控制器复位 0 = 模拟比较器A控制器正常工作 1 = 模拟比较器A控制器复位
[21]	PWM47_RST	PWM4~7 控制器复位 0= PWM4~7 控制器正常工作 1= PWM4~7 控制器复位
[20]	PWM03_RST	PWM0~3 控制器复位 0= PWM0~3 控制器正常工作 1= PWM0~3 控制器复位
[19:18]	保留	保留
[17]	UART1_RST	UART1控制器复位

		0= UART1 正常工作 1= UART1 控制器复位
[16]	UART0_RST	UART0控制器复位 0= UART0 正常工作 1= UART0 控制器复位
[15:14]	保留	保留
[13]	SPI1_RST	SPI1控制器复位 0= SPI1 正常工作 1= SPI1 控制器复位
[12]	SPI0_RST	SPI0控制器复位 0= SPI0 正常工作 1= SPI0 控制器复位
[11:10]	保留	保留
[9]	I2C1_RST	I2C1控制器复位 0= I2C正常工作 1= I2C 控制器复位
[8]	I2C0_RST	I2C0控制器复位 0= I2C0正常工作 1= I2C0控制器复位
[7:6]	保留	保留
[5]	TMR3_RST	Timer3控制器复位 0= Timer3 正常工作 1= Timer3 控制器复位
[4]	TMR2_RST	Timer2控制器复位 0= Timer2 正常工作 1= Timer2 控制器复位
[3]	TMR1_RST	Timer1控制器复位 0= Timer1 正常工作 1= Timer1 控制器复位
[2]	TMR0_RST	Timer0控制器复位 0= Timer0 正常工作 1= Timer0 控制器复位
[1]	GPIO_RST	GPIO (P0~P4) 控制器复位 0= GPIO 正常工作 1= GPIO 控制器复位
[0]	保留	保留

欠压检测控制寄存器(BODCR)

BODCR 控制寄存器的部分值由flash配置区域（CONFIG0/CONFIG1）初始化并且是写保护的，编程这些被保护的位需要依次向地址0x5000_0100写入“59h”，“16h”，“88h”，以禁用寄存器保护。参考寄存器REGWRPROT，其地址为GCR_BA+0x100

寄存器	偏移量	R/W	描述				复位后的值
BODCR	GCR_BA+0x18	R/W	欠压检测控制寄存器				0x0000_008X

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
LVR_EN	BOD_OUT	BOD_LPM	BOD_INTF	BOD_RSTEN	BOD_VL		BOD_EN

位	描述	
[31:8]	保留	保留
[7]	LVR_EN	低压复位使能（写保护位） 输入电源电压低于LVR电路设置时，LVR复位整个芯片. 默认配置下LVR复位是使能的，LVR复位电压请参考后面LVR特性 1= 使能低电压复位功能，使能该位100US后，LVR功能生效(默认) 0= 禁用低电压复位功能 注： 该位是写保护的，修改该位时，需要依次向0x5000_0100写入"59h", "16h", "88h"解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100
[6]	BOD_OUT	欠压检测器输出状态 1 = 欠压检测输出状态为 1，表示检测到的电压低于 BOD_VL 设置。若 BOD_EN 是“0”，BOD 功能被关闭，该位始终保持为 “0” 0 = 欠压检测输出状态为0，表示检测到的电压高于BOD_VL设置
[5]	BOD_LPM	欠压检测器低功耗模式 (写保护) 1 = 使能 BOD 低功耗模式 0 = BOD 工作于正常模式(默认) 注1： BOD 在正常模式下消耗电流约为100uA, 低功耗模式下减少到当前的1/10, 但是BOD响应速度变慢。 注2： 该位是写保护的，修改该位时，需要依次向0x5000_0100写入"59h", "16h", "88h"解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100
[4]	BOD_INTF	欠压检测器中断标志位 1= 欠压检测到V _{DD} 下降到低于BOD_VL 的设定电压或V _{DD} 上升高于BOD_VL 的设定电压，该位

		被置为1，如果欠压中断被使能，则发生欠压中断。 0= 没有检测到任何电压由V _{DD} 下降或上升至BOD_VL 设定值。 注： 该位写1清除为0															
[3]	BOD_RSTEN	<p>欠压复位使能控制位 (写保护)</p> <p>1 = 使能欠压复位功能，当欠压检测功能使能后，检测到电压低于门槛电压，芯片发生复位 0 = 使能欠压中断功能</p> <p>当BOD_EN使能(BOD_EN = 1)，且中断被使能时(BOD_RSTEN=0)，当BOD_OUT = 1时，BOD中断将发生。该中断会持续到将BOD_EN设置为“0”。通过禁用CPU中的NVIC以禁用BOD中断或者通过将BOD_EN=0可禁止CPU响应中断，如果需要BOD功能时，可重新使能BOD_EN功能</p> <p>注1：当BOD功能使能(BOD_EN =1)并且BOD复位功能使能(BOD_RSTEN=1)时，当BOD检测到电压低于门槛电压时(BOD_OUT=1)，将发出信号复位芯片。</p> <p>注2：缺省值由flash中用户配置区寄存器CONFIG0 bit[2:0]决定。</p> <p>注3：该位是写保护的，修改该位时，需要依次向0x5000_0100写入”59h”，“16h”，“88h”解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100</p>															
[2:1]	BOD_VL	<p>欠压检测器阈电压选择 (写保护)</p> <p>默认值 由FLASH中用户配置寄存器CONFIG0 bit[22:21]决定</p> <table border="1"> <thead> <tr> <th>BOV_VL[1]</th> <th>BOV_VL[0]</th> <th>欠压值</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>4.4V</td> </tr> <tr> <td>1</td> <td>0</td> <td>3.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.7V</td> </tr> <tr> <td>0</td> <td>0</td> <td>2.2V</td> </tr> </tbody> </table> <p>注：该位是写保护的，修改该位时，需要依次向0x5000_0100写入”59h”，“16h”，“88h”解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100</p>	BOV_VL[1]	BOV_VL[0]	欠压值	1	1	4.4V	1	0	3.7V	0	1	2.7V	0	0	2.2V
BOV_VL[1]	BOV_VL[0]	欠压值															
1	1	4.4V															
1	0	3.7V															
0	1	2.7V															
0	0	2.2V															
[0]	BOD_EN	<p>欠压检测使能 (写保护位)</p> <p>默认值 由FLASH中用户配置寄存器CONFIG0 bit[23]决定</p> <p>1 = 使能欠压检测功能 0 = 禁用欠压检测功能</p> <p>注：该位是写保护的，修改该位时，需要依次向0x5000_0100写入”59h”，“16h”，“88h”解除寄存器保护。参考寄存器 REGWRPROT，地址GCR_BA + 0x100</p>															

温度传感器控制寄存器(TEMPER)

寄存器	偏移量	R/W	描述	复位后的值
TEMPER	GCR_BA+0x1C	R/W	温度传感器控制寄存器	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
VTEMP_EN							

位	描述	
[31:1]	保留	保留
[0]	VTEMP_EN	<p>温度传感器使能控制 这个bit 用来使能/禁止温度传感器功能. 1 = 使能温度传感器功能 0 = 禁止温度传感器功能(缺省) 注: 这个比特设为1之后, ADC通道7可以选择温度传感器输入, 温度值可以在ADC的转换结果中读到. ADC转换功能的细节请参考ADC章节</p>

上电复位控制寄存器(PORCR)

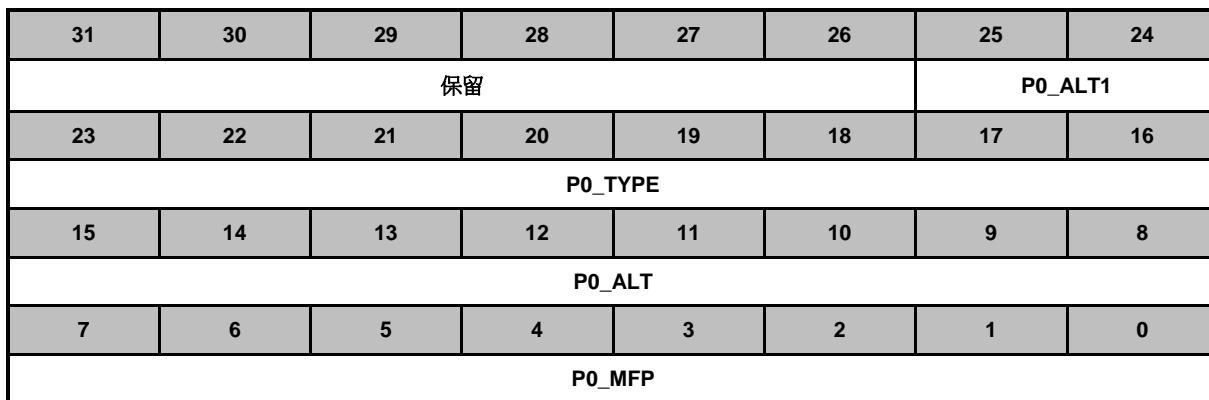
寄存器	偏移量	R/W	描述	复位后的值
PORCR	GCR_BA+0x24	R/W	上电复位控制寄存器	0x0000_00XX

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
POR_DIS_CODE							
7	6	5	4	3	2	1	0
POR_DIS_CODE							

位	描述	
[31:16]	保留	保留
[15:0]	POR_DIS_CODE	<p>上电复位使能控制（写保护）</p> <p>上电时，POR电路产生复位信号使整个芯片复位，但是电源部分的干扰可能引起POR再次发出复位信号。如果将POR_DIS_CODE 设置为0x5AA5，POR复位功能将被禁用，直到电源电压很低，导致POR_DIS_CODE 设置为其他值，或者由芯片的其他复位功能引起复位时POR功能重新有效，这些复位功能包括：</p> <p>nRST引脚复位，看门狗，LVR复位，BOD复位，ICE复位命令和软件复位。</p> <p>注：该寄存器是写保护的寄存器，写该位需要先向地址0x5000_0100依次写入“59h”，“16h”，“88h”解除寄存器保护。参考寄存器REGWRPROT的设置，其地址为GCR_BA + 0x100。</p>

Port0复用功能控制寄存器(P0_MFP)

寄存器	偏移量	R/W	描述	复位后的值
P0_MFP	GCR_BA+0x30	R/W	P0复用功能和输入类型控制寄存器	0x0000_0000



位	描述																
[31:26]	保留	保留															
[25]	P0_ALT1[1]	P0.1 功能选择1 P0.1 引脚的功能由 P0_MFP[1], P0_ALT[1], 和 P0_ALT1[1]决定。 细节描述请参考 P0_ALT[1].															
[24]	P0_ALT1[0]	P0.0功能选择1 P0.0 引脚的功能由P0_MFP[0], P0_ALT[0], 和 P0_ALT1[0].决定 细节描述请参考P0_ALT[0].															
[23:16]	P0_TYPE[n]	P0[7:0]史密特触发输入功能使能位 1= 使能P0[7:0] I/O史密特触发输入功能 0= 禁用P0[7:0] I/O史密特触发功能输入															
[15]	P0_ALT[7]	P0.7 功能选择位 P0.7 的功能取决于P0_MFP[7] 和 P0_ALT[7]. <table border="1"> <tr> <th>P0_ALT[7]</th> <th>P0_MFP[7]</th> <th>P0.7 的功能</th> </tr> <tr> <td>0</td> <td>0</td> <td>P0.7</td> </tr> <tr> <td>0</td> <td>1</td> <td>AD7(EBI)</td> </tr> <tr> <td>1</td> <td>0</td> <td>SPICLK1(SPI1)</td> </tr> <tr> <td>1</td> <td>1</td> <td>保留</td> </tr> </table>	P0_ALT[7]	P0_MFP[7]	P0.7 的功能	0	0	P0.7	0	1	AD7(EBI)	1	0	SPICLK1(SPI1)	1	1	保留
P0_ALT[7]	P0_MFP[7]	P0.7 的功能															
0	0	P0.7															
0	1	AD7(EBI)															
1	0	SPICLK1(SPI1)															
1	1	保留															
[14]	P0_ALT[6]	P0.6功能选择位 P0.6的功能取决于P0_MFP[6] 和 P0_ALT[6].															

		P0_ALT[6]	P0_MFP[6]	P0.6 的功能
		0	0	P0.6
		0	1	AD6(EBI)
		1	0	MISO_1(SPI1)
		1	1	保留
[13]	P0_ALT[5]	P0.5功能选择位 P0.5的功能取决于 P0_MFP[5] 和 P0_ALT[5].		
		P _ALT[5]	P0_MFP[5]	P0.5 的功能
		0	0	P0.5
		0	1	AD5(EBI)
		1		MOSI_1(SPI1)
		1	1	保留
[12]	P0_ALT[4]	P0.4功能选择位 of P0.4 的功能取决于 P0_MFP[4] 和 P0_ALT[4].		
		P0_ALT[4]	P0_MFP[4]	P0.4 的功能
		0	0	P0.4
		0	1	AD4(EBI)
		1	0	SPISS1(SPI1)
		1	1	保留
[11]	P0_ALT[3]	P0.3功能选择位 P0.3的功能取决于 P0_MFP[3] 和 P0_ALT[3].		
		P0_ALT[3]	P0_MFP[3]	P0.3的功能
		0	0	P0.3
		0	1	AD3(EBI)
		1	0	RTS0(UART0)
		1	1	RXD
[10]	P0_ALT[2]	P0.2功能选择位 P0.2的功能取决于 P0_MFP[2] 和 P0_ALT[2].		
		P0_ALT[2]	P0_MFP[2]	P0.2的功能
		0	0	P0.2

		0	1	AD2(EBI)	
		1	0	CTS0(UART0)	
		1	1	TXD	
[9]	P0_ALT[1]	P0.1功能选择位 P0.1 的功能取决于 P0_MFP[1]、P0_ALT[1]和P0_ALT1[1].			
		P0_ALT1[1]	P0_ALT[1]	P0_MFP[1]	P0.1的功能
		0	0	0	P0.1
		0	0	1	AD1(EBI)
		0	1	0	RTS1(UART1)
		0	1	1	RXD1
		1	0	0	ACMP3_N
[8]	P0_ALT[0]	P0.0功能选择位 P0.0的功能取决于 P0_MFP[0] 和、P0_ALT[0]和P0_ALT1[0].			
		P0_ALT1[0]	P0_ALT[0]	P0_MFP[0]	P0.0的功能
		0	0	0	P0.0
		0	0	1	A_0(EBI)
		0	1	0	CTS1(UART1)
		0	1	1	TXD1
[7:0]	P0_MFP[7:0]	P0功能选择位 P0 引脚的功能取决于 P0_MFP 和 P0_ALT. 参考P0_ALT的详细描述			

Port1复用功能控制寄存器(P1_MFP)

寄存器	偏移量	R/W	描述	复位后的值
P1_MFP	GCR_BA+0x34	R/W	P1复用功能与输入类型控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
P1_TYPE							
15	14	13	12	11	10	9	8
P1_ALT							
7	6	5	4	3	2	1	0
P1_MFP							

位	描述																	
[31:24]	保留	保留																
[23:16]	P1_TYPE[n]	P1[7:0] 输入史密特触发功能使能位 1= 使能P1[7:0] I/O输入史密特触发功能 0= 禁用P1[7:0] I/O输入史密特触发功能																
[15]	P1_ALT[7]	P1.7功能选择位 P1.7的功能取决于P1_MFP[7] 和 P1_ALT[7]. <table border="1" style="margin-left: 20px;"> <tr><th>P1_ALT[7]</th><th>P1_MFP[7]</th><th>P1.7 的功能</th></tr> <tr><td>0</td><td></td><td>P1.</td></tr> <tr><td>0</td><td>1</td><td>AIN7(ADC)</td></tr> <tr><td>1</td><td>0</td><td>SPICLK0(SPI0)</td></tr> <tr><td>1</td><td>1</td><td>ACMP2_P</td></tr> </table>		P1_ALT[7]	P1_MFP[7]	P1.7 的功能	0		P1.	0	1	AIN7(ADC)	1	0	SPICLK0(SPI0)	1	1	ACMP2_P
P1_ALT[7]	P1_MFP[7]	P1.7 的功能																
0		P1.																
0	1	AIN7(ADC)																
1	0	SPICLK0(SPI0)																
1	1	ACMP2_P																
[14]	P1_ALT[6]	P1.6功能选择位 P1.6 的功能取决于 P1_MFP[6] 和 P1_ALT[6]. <table border="1" style="margin-left: 20px;"> <tr><th>P1_ALT[6]</th><th>P1_MFP[6]</th><th>P1.6 的功能</th></tr> <tr><td>0</td><td></td><td>P1.</td></tr> <tr><td>0</td><td>1</td><td>AIN6(ADC)</td></tr> <tr><td>1</td><td>0</td><td>MISO_0(SPI0)</td></tr> <tr><td>1</td><td>1</td><td>ACMP2_N</td></tr> </table>		P1_ALT[6]	P1_MFP[6]	P1.6 的功能	0		P1.	0	1	AIN6(ADC)	1	0	MISO_0(SPI0)	1	1	ACMP2_N
P1_ALT[6]	P1_MFP[6]	P1.6 的功能																
0		P1.																
0	1	AIN6(ADC)																
1	0	MISO_0(SPI0)																
1	1	ACMP2_N																
[13]	P1_ALT[5]	P1.5功能选择位 P1.5的功能取决于P1_MFP[5] 和 P1_ALT[5].																

		P1_ALT[5]	P1_MFP[5]	P1.5 的功能
		0	0	P1.5
		0	1	AIN5(ADC)
		1	0	MOSI_0(SPI0)
		1	1	ACMP0_P
[12]	P1_ALT[4]	P1.4 功能选择位 P1.4的功能取决于P1_MFP[4] 和 P1_ALT[4].		
		P1_ALT[4]	P1_MFP[4]	P1.4的功能
		0	0	P1.4
			1	AIN4(ADC)
		1	0	SPISS0(SPI0)
		1	1	ACMP0_N
[11]	P1_ALT[3]	P1.3 功能选择位 P1.3的功能取决于P1_MFP[3] 和 P1_ALT[3].		
		P1_ALT[3]	P1_MFP[3]	P1.3的功能
		0	0	P1.3
		0	1	AIN3(ADC)
		1	0	TXD1(UART1)
		1	1	保留
[10]	P1_ALT[2]	P1.2 功能选择位 P1.2的功能取决于P1_MFP[2] 和 P1_ALT[2].		
		P1_ALT[2]	P1_MFP[2]	P1.2的功能
		0	0	P1.2
		0	1	AIN2(ADC)
		1	0	RXD1(UART1)
		1	1	保留
[9]	P1_ALT[1]	P1.1 功能选择位 P1.1的功能取决于P1_MFP[1] and P1_ALT[1].		
		P1_ALT[1]	P1_MFP[1]	P1.1的功能
		0	0	P1.1
			1	AIN1 (ADC)
		1	0	T3(Time 3)
		1	1	nWRH
[8]	P1_ALT[0]	P1.0 功能选择位 P1.0的功能取决于P1_MFP[0] and P1_ALT[0].		

		P1_ALT[0]	P1_MFP[0]	P1.0的功能
		0	0	P1.0
		0	1	AIN0(ADC)
		1	0	T2 (Timer2)
		1	1	nWRL
[7:0]	P1_MFP[7:0]	P1 功能选择位 P1的功能取决于P1_MFP 和 P1_ALT. 参考P1_ALT的详细描述		

Port2多功能控制寄存器(P2_MFP)

寄存器	偏移量	R/W	描述	复位后的值
P2_MFP	GCR_BA+0x38	R/W	P2复用功能与输入类型控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
P2_TYPE							
15	14	13	12	11	10	9	8
P2_ALT							
7	6	5	4	3	2	1	0
P2_MFP							

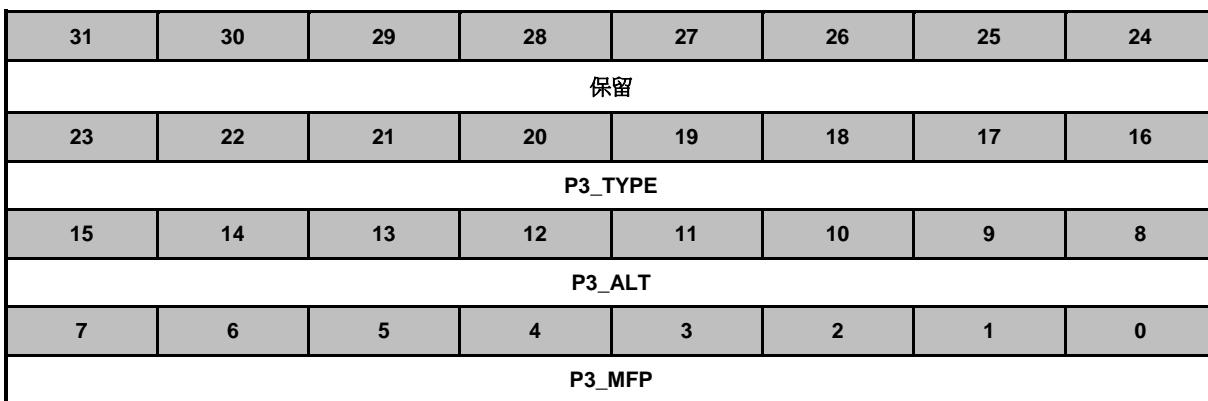
位	描述																	
[31:24]	保留	保留																
[23:16]	P2_TYPE[n]	P2[7:0] 输入史密特触发功能使能位 1= 使能P2[7:0] I/O输入史密特触发功能 0= 禁用P2[7:0] I/O输入史密特触发功能																
[15]	P2_ALT[7]	P2.7 功能选择位 P2.7的功能取决于P2_MFP[7] and P2_ALT[7]. <table border="1" style="margin-left: 20px;"> <tr><th>P2_ALT[7]</th><th>P2_MFP[7]</th><th>P2.7 的功能</th></tr> <tr><td></td><td>0</td><td>P2.7</td></tr> <tr><td>0</td><td>1</td><td>AD15 EBI)</td></tr> <tr><td>1</td><td>0</td><td>PWM7(PWMB 通道3)</td></tr> <tr><td>1</td><td>1</td><td>保留</td></tr> </table>		P2_ALT[7]	P2_MFP[7]	P2.7 的功能		0	P2.7	0	1	AD15 EBI)	1	0	PWM7(PWMB 通道3)	1	1	保留
P2_ALT[7]	P2_MFP[7]	P2.7 的功能																
	0	P2.7																
0	1	AD15 EBI)																
1	0	PWM7(PWMB 通道3)																
1	1	保留																
[14]	P2.6 功能选择位 P2.6的功能取决于P2_MFP[6] and P2_ALT[6]. <table border="1" style="margin-left: 20px;"> <tr><th>P2_ALT[]</th><th>P2_MFP[6]</th><th>P2.6 的功能</th></tr> <tr><td>0</td><td>0</td><td>P2.6</td></tr> <tr><td>0</td><td>1</td><td>AD14(EBI)</td></tr> <tr><td>1</td><td>0</td><td>PWM6(PWMB通道2)</td></tr> <tr><td>1</td><td>1</td><td>ACMP1_O</td></tr> </table>		P2_ALT[]	P2_MFP[6]	P2.6 的功能	0	0	P2.6	0	1	AD14(EBI)	1	0	PWM6(PWMB通道2)	1	1	ACMP1_O	
P2_ALT[]	P2_MFP[6]	P2.6 的功能																
0	0	P2.6																
0	1	AD14(EBI)																
1	0	PWM6(PWMB通道2)																
1	1	ACMP1_O																
[13]	P2_ALT[5]	P2.5 功能选择位 P2.5的功能取决于P2_MFP[5] and P2_ALT[5].																

		P2_ALT[5]	P2_MFP[5]	P2.5 的功能
		0	0	P2.5
		0	1	AD13(EBI)
		1	0	PWM5(PWMB 通道 1)
		1	1	SDA1(I2C1)
[12]	P2_ALT[4]	P2.4 功能选择位 P2.4的功能取决于P2_MFP[4] and P2_ALT[4].		
		P2_ALT[4]	P2_MFP[4]	P2.4的功能
		0	0	P2.4
		0	1	AD12(EBI)
		1	0	PWM4(PWMB通道0)
		1	1	SCL1(I2C1)
[11]	P2_ALT[3]	P2.3 功能选择位 P2.3的功能取决于P2_MFP[3] and P2_ALT[3].		
		P2_ALT[3]	P2_MFP[3]	P2.3的功能
		0	0	P2.
		0	1	AD11(EBI)
		1		PWM3(PWMA通道3)
		1	1	保留
[10]	P2_ALT[2]	P2.2 功能选择位 P2.2的功能取决于P2_MFP[2] 和 P2_ALT[2].		
		P2_ALT[2]	P2_MFP[2]	P2.2的功能
		0	0	P2.2
			1	AD10(EBI)
		1	0	PWM2(PWMA通道 2)
		1	1	保留
[9]	P2_ALT[1]	P2.1 功能选择位 P2.1的功能取决于P2_MFP[1] and P2_ALT[1].		
		P2_ALT[1]	P2_MFP[1]	P2.1的功能
		0	0	P2.1
		0	1	AD9(EBI)
		1	0	PWM1(PWMA通道1)

		1	1	保留	
[8]	P2_ALT[0]	P2.0 功能选择位 P2.0的功能取决于P2_MFP[0] and P2_ALT[0].			
		P2_ALT[0]	P2_MFP[0]	P2.0的功能	
		0		P2.0	
		0	1	AD8(EBI)	
		1	0	PWM0(PWMA通道0)	
[7:0]	P2_MFP[7:0]	P2 功能选择位 P2的功能取决于P2_MFP 和 P2_ALT. 参考P2_ALT的详细描述.			

Port3多功能控制寄存器(P3_MFP)

寄存器	偏移量	R/W	描述	复位后的值
P3_MFP	GCR_BA+0x3C	R/W	P3复用功能与输入类型控制寄存器	0x0000_0000



位	描述																
[31:24]	保留	保留															
[23:16]	P3_TYPE[n]	使能P3[7:0] 输入史密特触发功能 1= 使能P3[7:0] I/O输入史密特触发功能 0= 禁用P3[7:0] I/O输入史密特触发功能															
[15]	P3_ALT[7]	P3.7 功能选择位 P3.7的功能取决于P3_MFP[7] 和 P3_ALT[7]. <table border="1"> <thead> <tr> <th>P3_ALT[7]</th> <th>P3_MFP[7]</th> <th>P3.7 的功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>P3.7</td> </tr> <tr> <td>0</td> <td>1</td> <td>nRD(EBI)</td> </tr> <tr> <td>1</td> <td>x</td> <td>保留</td> </tr> </tbody> </table>	P3_ALT[7]	P3_MFP[7]	P3.7 的功能	0	0	P3.7	0	1	nRD(EBI)	1	x	保留			
P3_ALT[7]	P3_MFP[7]	P3.7 的功能															
0	0	P3.7															
0	1	nRD(EBI)															
1	x	保留															
[14]	P3_ALT[6]	P3.6 功能选择位 P3.6的功能取决于P3_MFP[6] 和 P3_ALT[6]. <table border="1"> <thead> <tr> <th>P3_ALT[6]</th> <th>P3_MFP[6]</th> <th>P3.6 的功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>P3.6</td> </tr> <tr> <td>0</td> <td>1</td> <td>nWR(EB)</td> </tr> <tr> <td>1</td> <td>0</td> <td>CKO(时钟除频输出)</td> </tr> <tr> <td>1</td> <td>1</td> <td>ACMP0_o</td> </tr> </tbody> </table>	P3_ALT[6]	P3_MFP[6]	P3.6 的功能	0	0	P3.6	0	1	nWR(EB)	1	0	CKO(时钟除频输出)	1	1	ACMP0_o
P3_ALT[6]	P3_MFP[6]	P3.6 的功能															
0	0	P3.6															
0	1	nWR(EB)															
1	0	CKO(时钟除频输出)															
1	1	ACMP0_o															
[13]	P3_ALT[5]	P3.5 功能选择位 P3.5的功能取决于P3_MFP[5] 和 P3_ALT[5]. <table border="1"> <thead> <tr> <th>P3_ALT[5]</th> <th>P3_MFP[5]</th> <th>P3.5 的功能</th> </tr> </thead> </table>	P3_ALT[5]	P3_MFP[5]	P3.5 的功能												
P3_ALT[5]	P3_MFP[5]	P3.5 的功能															

		0	0	P3.5
		0	1	T1(Timer1)外部事件计数器
		1	0	SCL(I ² C)
		1	1	ACMP1_P (ACMP)
[12]	P3_ALT[4]	P3.4 功能选择位 P3.4的功能取决于P3_MFP[4] 和 P3_ALT[4].		
		P3_ALT[4]	P3_MFP[4]	P3.4的功能
		0	0	P3.4
		0	1	T0(Timer0)
		1		SDA0(I ² C0)
		1	1	保留
[11]	P3_ALT[3]	P3.3 功能选择位 P3.3的功能取决于P3_MFP[3] 和 P3_ALT[3].		
		P3_ALT[3]	P3_MFP[3]	P3.3的功能
		0	0	P3.3
		0	1	nI_T
		1	0	MCLK(EBI)
		1	1	T1EX
[10]	P3_ALT[2]	P3.2 功能选择位 P3.2的功能取决于P3_MFP[2] and P3_ALT[2].		
		P3_ALT[2]	P3_MFP[2]	P3.2的功能
		0	0	P3.2
		0	1	nINT0
		1	0	T0EX
		1	1	保留
[9]	P3_ALT[1]	P3.1 功能选择位 P3.1的功能取决于P3_MFP[1] and P3_ALT[1].		
		P3_ALT[1]	P3_MFP[1]	P3.1的功能
		0	0	P3_1
		0	1	TXD(UART0)
		1	0	ACMP1_P
		1	1	保留
[8]	P3_ALT[0]	P3.0 功能选择位 P3.0的功能取决于P3_MFP[0] and P3_ALT[0].		
		P3_ALT[0]	P3_MFP[0]	P3.0的功能

		0	0	P3.0
		0	1	RXD(UART0)
		1	0	ACMP1_N
		1	1	保留
[7:0]	P3_MFP[7:0]	P3 功能选择位 P3的功能取决于P3_MFP 和 P3_ALT. 参考 P3_ALT的详细描述.		

Port4多功能控制寄存器(P4_MFP)

寄存器	偏移量	R/W	描述				复位后的值
P4_MFP	GCR_BA+0x40	R/W	P4复用功能与输入类型控制寄存器				0x0000_00C0

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
P4_TYPE							
15	14	13	12	11	10	9	8
P4_ALT							
7	6	5	4	3	2	1	0
P4_MFP							

位	描述														
[31:24]	保留	保留													
[23:16]	P4_TYPE[n]	P4[7:0] 输入史密特触发功能位 1= 使能P4[7:0] I/O输入史密特触发功能 0= 禁用P4[7:0] I/O输入史密特触发功能													
[15]	P4_ALT[7]	P4.7 功能选择位 P4.7的功能取决于P4_MFP[7] and P4_ALT[7]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>P4_ALT[7]</th> <th>P4_MFP[7]</th> <th>P4.7 的功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>P4.7</td> </tr> <tr> <td>0</td> <td>1</td> <td>ICE_DAT(ICE)</td> </tr> <tr> <td>1</td> <td>x</td> <td>保留</td> </tr> </tbody> </table>		P4_ALT[7]	P4_MFP[7]	P4.7 的功能	0	0	P4.7	0	1	ICE_DAT(ICE)	1	x	保留
P4_ALT[7]	P4_MFP[7]	P4.7 的功能													
0	0	P4.7													
0	1	ICE_DAT(ICE)													
1	x	保留													
[14]	P4_ALT[6]	P4.6 功能选择位 P4.6的功能取决于P4_MFP[6] and P4_ALT[6]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>P4_ALT[6]</th> <th>P4_MFP[6]</th> <th>P4.6 的功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>P4.6</td> </tr> <tr> <td>0</td> <td>1</td> <td>ICE_CLK(ICE)</td> </tr> <tr> <td>1</td> <td>x</td> <td>保留</td> </tr> </tbody> </table>		P4_ALT[6]	P4_MFP[6]	P4.6 的功能	0	0	P4.6	0	1	ICE_CLK(ICE)	1	x	保留
P4_ALT[6]	P4_MFP[6]	P4.6 的功能													
0	0	P4.6													
0	1	ICE_CLK(ICE)													
1	x	保留													
[13]	P4_ALT[5]	P4.5 功能选择位 P4.5的功能取决于P4_MFP[5] and P4_ALT[5]. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>P4_ALT[5]</th> <th>P4_MFP[5]</th> <th>P4.5 的功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>P4.5</td> </tr> </tbody> </table>		P4_ALT[5]	P4_MFP[5]	P4.5 的功能	0	0	P4.5						
P4_ALT[5]	P4_MFP[5]	P4.5 的功能													
0	0	P4.5													

		0	1	ALE(EBI)	
		1	0	SDA1(I ² C1)	
		1	1	保留	
[12]	P4_ALT[4]	P4.4 功能选择位 P4.4的功能取决于P4_MFP[4] and P4_ALT[4].			
		P4_ALT[4]	P4_MFP[4]	P4.4的功能	
		0	0	P4.4	
		0	1	nCS (EBI)	
		1	0	SCL1(I ² C1)	
		1	1	保留	
[11]	P4_ALT[3]	P4.3 功能选择位 P4.3的功能取决于P4_MFP[3] and P4_ALT[3].			
		P4_ALT[3]	P4_MFP[3]	P4.3的功能	
		0	0	P4.3	
		0	1	PWM3(PWMA通道3)	
		1	x	保留	
[10]	P4_ALT[2]	P4.2 功能选择位 P4.2的功能取决于P4_MFP[2] and P4_ALT[2].			
		P4_ALT[2]	P4_MFP[2]	P4.2的功能	
		0	0	P4.2	
		0	1	PWM2(PWMA通道2)	
		1	x	保留	
[9]	P4_ALT[1]	P4.1 功能选择位 P4.1的功能取决于P4_MFP[1] and P4_ALT[1].			
		P4_ALT[1]	P4_MFP[1]	P4.1的功能	
		0	0	P4.1	
			1	PWM1(PWMA 通道1)	
		1	0	T3EX	
		1	1	保留	
[8]	P4_ALT[0]	P4.0 功能选择位 P4.0的功能取决于P4_MFP[0] and P4_ALT[0].			
		P4_ALT[0]	P4_MFP 0]	P4.0的功能	
		0	0	P4.0	
		0	1	PWM0(PWMA通道0)	
		1	0	T2EX	

		1	1	保留
[7:0]	P4_MFP[7:0]	P4 功能选择位 P4的功能取决于P4_MFP 和 P4_ALT. 参考P4_ALT 的详细描述.		

写保护寄存器控制寄存器(REGWRPROT)

有些系统控制寄存器需要被保护起来，以防止误操作而影响芯片运行，这些系统控制寄存器在上电复位后是被保护的，直到用户关闭寄存器保护。用户如果要编程这些被保护的寄存器，需要写一个解锁序列到REGWRPROT寄存器。解锁序列就是连续写“59h”，“16h”，“88h”到REGWRPROT寄存器，地址是0x5000_010。在写这三个数据过程中，任何不同的数据值，不同的顺序或者任何对其他地址的写，都会中止解锁序列。

保护被关闭之后，用户可以检查保护禁止位（地址0x5000_0100的比特0），"1"表示保护关闭，"0"表示保护使能。然后用户可以更新被保护的目标寄存器的值，并随时可以通过写任何其它数据到地址"0x5000_0100"来使能寄存器保护。

写该寄存器来关闭/使能寄存器保护，读该寄存器来获取REGPROTDIS状态。

寄存器	偏移量	R/W	描述					复位后的值
REGWRPROT	GCR_BA+100	R/W	寄存器写保护控制寄存器					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	
REGWRPROT[7:1]							REGWRPROT [0] REGPROTDIS	

位	描述										
[31:16]	保留	保留									
[7:0]	REGWRPROT	写保护代码寄存器(只写) 一些寄存器有写保护功能，编程这些寄存器之前必须通过一个解锁序列来关闭写保护功能，解锁序列就是依次写“59h”，“16h”，“88h”到这个域，这个序列完成之后，REGPROTDIS 位将被置1，写保护寄存器可以正常写入数据。									
[0]	REGPROTDIS	写保护寄存器解锁标志 (只读) 1 = 解除写保护，写保护寄存器可以正常编程 0 = 使能寄存器的写保护，任何写受保护寄存器的操作将被忽略。 受保护的寄存器有： <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>寄存器</th> <th>地址</th> <th>注意事项</th> </tr> </thead> <tbody> <tr> <td>IPRSTC1</td> <td>0x500_0008</td> <td>外设复位控制寄存器1</td> </tr> <tr> <td>BODCR</td> <td>0x5_00_0018</td> <td>欠压检测控制寄存器</td> </tr> </tbody> </table>	寄存器	地址	注意事项	IPRSTC1	0x500_0008	外设复位控制寄存器1	BODCR	0x5_00_0018	欠压检测控制寄存器
寄存器	地址	注意事项									
IPRSTC1	0x500_0008	外设复位控制寄存器1									
BODCR	0x5_00_0018	欠压检测控制寄存器									

	PORCR	0x5000_0024	上电复位控制寄存器
	PWRCON	0x5000_0200	位[6]没有受保护，电源唤醒中断将清除该位
	APBCLK bit[0]	0x5000_0208	位[0]是看门狗时钟使能位
	CLKSEL0	0x5000_0210	HCLK和SysTick时钟源选择
	CLKSEL1 bit[1:0]	0x5000_0214	看门狗时钟源选择
	NMI_SEL bit[8]	0x5000_0380	NMI 中断使能
	ISPCON	0x5000_C000	Flash ISP控制
	ISPTRG	0x5000_C010	ISP触发控制
	WTCR	0x4000_4000	看门狗时钟控制
	FATCON	0x5000_C018	Flash访问时间控制
注：在相应寄存器的描述中，写保护的bit有标注”(写保护)“			

5.2.7 NUC029FAE系统管理器控制寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移地址	R/W	描述	复位值
GCR 基址: GCR_BA = 0x5000_0000				
PDID	GCR_BA+0x00	R	设备ID寄存器	0x0029_5415
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	0x0000_0000
IPRSTC2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000
BODCR	GCR_BA+0x18	R/W	欠压检测控制寄存器	0x0000_000X
P0_MFP	GCR_BA+0x30	R/W	P0 复用功能和输入类型控制寄存器	0x0000_0000
P1_MFP	GCR_BA+0x34	R/W	P1复用功能和输入类型控制寄存器	0x0000_0000
P2_MFP	GCR_BA+0x38	R/W	P2 复用功能和输入类型控制寄存器	0x0000_0000
P3_MFP	GCR_BA+0x3C	R/W	P3 复用功能和输入类型控制寄存器	0x0000_0000
P4_MFP	GCR_BA+0x40	R/W	P4复用功能和输入类型控制寄存器	0x0000_00C0
P5_MFP	GCR_BA+0x44	R/W	P5复用功能和输入类型控制寄存器	0x0000_0000
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC校准控制寄存器	0x0000_0000
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC校准中断使能控制寄存器	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC校准中断状态寄存器	0x0000_0000
REGWRPROT	GCR_BA+0x100	R/W	寄存器写保护控制寄存器	0x0000_0000

设备ID寄存器(PDID)

寄存器	偏移地址	R/W	描述	复位值
PDID	GCR_BA+0x00	R	设备ID寄存器	0x0029_5415 ^[1]

[1] 每个器件具有一个独一无二的默认复位值

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	设备ID码 该寄存器为设备的器件号码。软件可以通过读取该寄存器来识别所使用的器件。

NuMicro® NUC029FAE	设备ID识别码
NUC029FAE	0x0029_5415

系统复位源寄存器(RSTSRC)

该寄存器提供一些信息用于软件识别上次引起芯片复位的复位源。

寄存器	偏移地址	R/W	描述	复位值
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX

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
RSTS_CPU	保留	RSTS MCU	RSTS_BOD	保留	RSTS_WDT	RSTS_RESET	RSTS_POR

位	描述	
[31:8]	保留	保留
[7]	RSTS_CPU	CPU复位标志 如果软件写1到CPU_RST (IPRSTC1[1]) ,来复位Cortex®-M0内核和Flash内存控制器(FMC) , 硬件会自动置位该CPU复位标志位。 0 = CPU无复位 1 = Cortex®-M0 内核和FMC因为软件置CPU_RST为1而复位。 注: 软件可向该位写1来清0。
[6]	保留	保留
[5]	RSTS MCU	MCU复位标志 该复位标志由来自Cortex®-M0内核的“复位信号”置位，用于表示之前出现的复位源。 0 = Cortex®-M0无复位 1 = Cortex®-M0因为软件向SYSRESETREQ (AIRCR[2]位写1而发出复位信号，从而导致系统复位。AIRCR是应用中断和复位控制寄存器，其地址为0xE000ED0C，位于Cortex®-M0 内核系统控制寄存器中。 注: 软件可向该位写1来清0。
[4]	RSTS_BOD	欠压检测器复位标志 RSTS_BOD复位标志由欠压检测器的“复位信号”置位，用于表示导致之前复位的信号源。 0 = BOD无复位。 1 = BOD发出复位信号使系统复位。 注: 软件可向该位写1来清0。
[3]	保留	保留

[2]	RSTS_WDT	WDT复位标志 RSTS_WDT标志由看门狗定时器或窗口看门狗定时器的“复位信号”置位，用于表示导致之前复位的复位源。 0 = 看门狗定时器或窗口看门狗定时器无复位。 1 = 看门狗定时器或窗口看门狗定时器发出复位信号使系统复位。 注: 软件可向该位写1来清0。
[1]	RSTS_RESET	nRESET管脚复位标志 RSTS_RESET管脚复位标志由nRESET管脚的“复位信号”置位，用于表示导致之前复位的复位源。 0 = nRESET管脚无复位 1 = nRESET管脚发出复位信号使系统复位 注: 软件可向该位写1来清0。
[0]	RSTS_POR	POR复位标志 RSTS_POR复位标志由上电复位(POR)控制器或CHIP_RST(IPRSTC1[0])位发出“复位信号”置位，用于表示导致之前复位的复位源。 0 = POR或CHIP_RST无复位 1 = 上电复位(POR)或CHIP_RST发出复位信号使系统复位 注: 软件可向该位写1来清0。

外设复位控制寄存器1 (IPRSTC1)

寄存器	偏移地址	R/W	描述	复位值
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	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						CPU_RST	CHIP_RST

位	描述	
[31:2]	保留	保留
[1]	CPU_RST	<p>处理器内核一次性复位(写保护) 置该位为1将只复位处理器内核和Flash内存控制器(FMC)，该位将在2个时钟周期后自动清零。 0 = 处理器内核正常运行 1 = 处理器内核一次性复位 注: 该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。</p>
[0]	CHIP_RST	<p>芯片一次性复位(写保护) 置该位为1将复位整个芯片，包括处理器内核和所有外设，该位将在2个时钟周期后自动清零。 CHIPRST与POR复位一样，所有芯片控制器都复位，芯片设置从flash重新加载。 0 = 芯片正常运行 1 = 芯片一次性复位 注: 该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。</p>

外设复位控制寄存器2 (IPRSTC2)

设置这些位为1将产生异步复位信号给相应的模块控制器，用户需要设置这些位为0，才能将相应模块控制器从复位状态中释放。

寄存器	偏移地址	R/W	描述	复位值
IPRSTC2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
保留			ADC_RST	保留			
23	22	21	20	19	18	17	16
保留	ACMP_RST	保留	PWM_RST	保留		UART_RST	
15	14	13	12	11	10	9	8
保留			SPI_RST	保留			I2C_RST
7	6	5	4	3	2	1	0
保留				TMR1_RST	TMR0_RST	GPIO_RST	保留

位	描述	
[31:29]	保留	保留
[28]	ADC_RST	ADC控制器复位 0 = ADC控制器正常运行 1 = ADC控制器复位
[27:23]	保留	保留
[22]	ACMP_RST	ACMP控制器复位 0 = ACMP控制器正常运行 1 = ACMP控制器复位
[21]	保留	保留
[20]	PWM_RST	PWM控制器复位 0 = PWM控制器正常运行 1 = PWM控制器复位
[19:17]	保留	保留
[16]	UART_RST	UART控制器复位 0 = UART控制器正常运行 1 = UART控制器复位
[15:13]	保留	保留
[12]	SPI_RST	SPI控制器复位

		0 = SPI控制器正常运行 1 = SPI控制器复位
[11:9]	保留	保留
[8]	I ² C_RST	I ² C控制器复位 0 = I ² C控制器正常运行 1 = I ² C控制器复位
[7:4]	保留	保留
[3]	TMR1_RST	Timer1控制器复位 0 = Timer1控制器正常运行 1 = Timer1控制器复位
[2]	TMR0_RST	Timer0控制器复位 0 = Timer0控制器正常运行 1 = Timer0控制器复位
[1]	GPIO_RST	GPIO (P0~P5)控制器复位 0 = GPIO控制器正常运行 1 = GPIO控制器复位
[0]	保留	保留

欠压检测器控制寄存器(BODCR)

BODCR控制寄存器部分位的值由flash配置初始化，且为写保护位。假如用户需要编程写保护位的内容，需要一个解锁时序。解锁时序是依次连续的往地址为0x5000_0100的控制器写入0x59、0x16、0x88。在写以上三个数据的过程中，写入别的数据或其他写操作都将终止解锁时序。

解锁时序后，用户可以查看地址0x5000_0100的位0，1表示没锁定，0表示锁定。没锁定的情况下用户可以更新写保护寄存器的值。写任何其他数据到地址0x5000_0100将再次锁定写保护寄存器。

寄存器	偏移地址	R/W	描述	复位值
BODCR	GCR_BA+0x18	R/W	欠压检测器控制寄存器	0x0000_000X

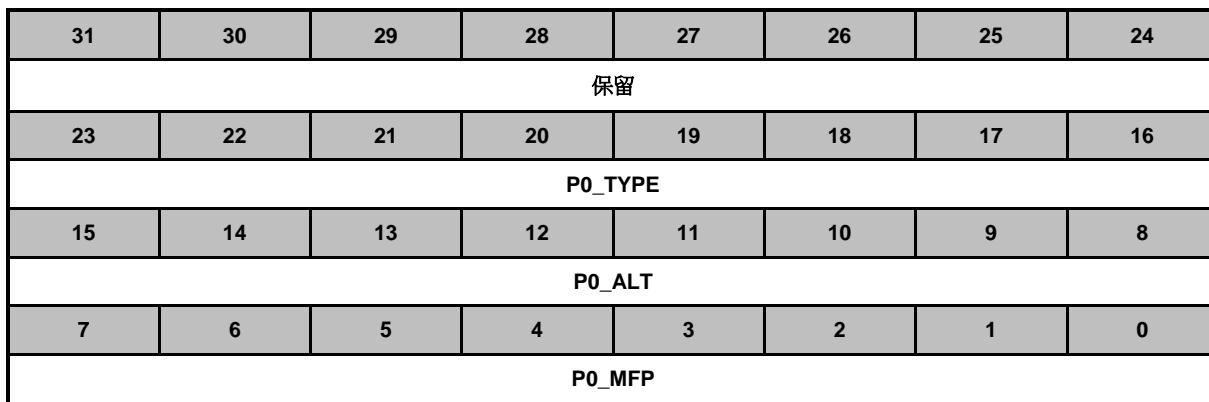
31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
保留							
7	6	5	4	3	2	1	0
保留	BOD_OUT	BOD_LPM	BOD_INTF	BOD_RSTEN	BOD_VL		BOD_VL_EXT

位	描述	
[31:7]	保留	保留
[6]	BOD_OUT	欠压检测器输出状态 0 = 欠压检测器输出0，表示检测到的电压比BOD_VL设置的高 1 = 欠压检测器输出1，表示检测到的电压比BOD_VL设置的低
[5]	BOD_LPM	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： BOD在正常模式消耗差不多100uA电流，低功耗模式将大约减少电流到1/10，但将减慢BOD的响应速度。 注2： 该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。
[4]	BOD_INTF	欠压检测器中断标志位 0 = 欠压检测器没有检测到V _{DD} 曾经下降或上升到BOD_VL设定的值 1 = 当检测到V _{DD} 下降或者上升到BOD_VL 设定的值，该位将置1，如果欠压中断使能，则欠压中断将发生 注： 该位写1清0
[3]	BOD_RSTEN	欠压复位使能控制位 (写保护) 0 = 欠压“中断”功能使能

		<p>当BOD_OUT为高时，如果BOD功能和BOD中断功能都使能，BOD将产生一次中断。BOD功能禁用后，中断也将退出。也可以通过禁用NVIC BOD中断或BOD功能来禁用BOD中断。</p> <p>1 = 欠压“复位”功能使能</p> <p>注1：当检测到电压低于设定的值时(BOD_OUT 高)，如果欠压检测器功能和欠压复位功能同时使能时，将产生一个信号来复位芯片。</p> <p>注2：默认值由flash控制器用户配置寄存器config0[20]位决定</p> <p>注3：该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。</p>															
[2:1]	BOD_VL	<p>欠压检测器阈电压选择(写保护)</p> <p>默认值由用户配置的flash控制寄存器的CBOV (CONFIG0[22:21])位设置。</p> <table border="1"> <thead> <tr> <th>BOV_VL[1]</th><th>BOV_VL[0]</th><th>掉电检测电压</th></tr> </thead> <tbody> <tr> <td>1</td><td>1</td><td>禁止 2.7V 和 3.7V</td></tr> <tr> <td>1</td><td>0</td><td>3.7V</td></tr> <tr> <td>0</td><td>1</td><td>2.7V</td></tr> <tr> <td>0</td><td>0</td><td>保留</td></tr> </tbody> </table> <p>注：该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。</p>	BOV_VL[1]	BOV_VL[0]	掉电检测电压	1	1	禁止 2.7V 和 3.7V	1	0	3.7V	0	1	2.7V	0	0	保留
BOV_VL[1]	BOV_VL[0]	掉电检测电压															
1	1	禁止 2.7V 和 3.7V															
1	0	3.7V															
0	1	2.7V															
0	0	保留															
[0]	BOD_VL_EXT	<p>欠压检测器使能控制(写保护位)</p> <p>默认值由用户配置的flash控制器的寄存器config0位[23]设置。</p> <p>如果config0位[23]设置为1，BOD_VL_EXT默认值为0。</p> <p>如果config0位[23]设置为0，BOD_VL_EXT默认值为1。</p> <p>0 = 欠压检测器功能禁用</p> <p>1 = 欠压检测器功能使能。</p> <table border="1"> <thead> <tr> <th>BOD_VL[1]</th><th>BOD_VL[0]</th><th>掉电检测电压</th></tr> </thead> <tbody> <tr> <td>1</td><td>1</td><td>4.4V</td></tr> <tr> <td>1</td><td>0</td><td>3.7V</td></tr> <tr> <td>0</td><td>1</td><td>2.7V</td></tr> <tr> <td>0</td><td>0</td><td>2.2V</td></tr> </tbody> </table> <p>注：该位为写保护位，编程该位需先解锁寄存器保护状态，向地址0x5000_0100依次写入“59h”，“16h”，和“88h”，具体内容请参考SYS_REGLCTL寄存器，其地址为GCR_BA+0x100。</p>	BOD_VL[1]	BOD_VL[0]	掉电检测电压	1	1	4.4V	1	0	3.7V	0	1	2.7V	0	0	2.2V
BOD_VL[1]	BOD_VL[0]	掉电检测电压															
1	1	4.4V															
1	0	3.7V															
0	1	2.7V															
0	0	2.2V															

Port0复用功能控制寄存器(P0_MFP)

寄存器	偏移地址	R/W	描述	复位值
P0_MFP	GCR_BA+0x30	R/W	P0复用功能和输入类型控制寄存器	0x0000_0000

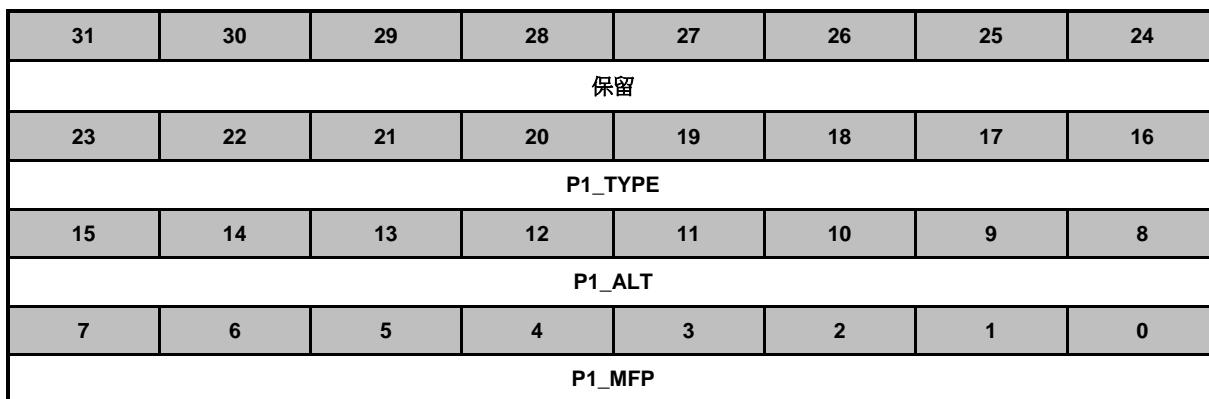


位	描述																
[31:24]	保留	保留															
[23:16]	P0_TYPE[n]	P0[7:0]TTL或施密特触发功能使能位 0 = P0[7:0] I/O输入为TTL功能 1 = P0[7:0] I/O输入为施密特触发功能															
[15]	P0_ALT[7]	P0.7功能选择位 位P0_MFP[7] 和 P0_ALT[7]. 决定P0.7 功能 <table border="1"> <tr><th>P0_ALT[7]</th><th>P0_MFP[7]</th><th>P0.7功能</th></tr> <tr><td>0</td><td>0</td><td>P0.7</td></tr> <tr><td>0</td><td>1</td><td>保留</td></tr> <tr><td>1</td><td>0</td><td>SPICLK0(SPI)</td></tr> <tr><td></td><td>1</td><td>保留</td></tr> </table>	P0_ALT[7]	P0_MFP[7]	P0.7功能	0	0	P0.7	0	1	保留	1	0	SPICLK0(SPI)		1	保留
P0_ALT[7]	P0_MFP[7]	P0.7功能															
0	0	P0.7															
0	1	保留															
1	0	SPICLK0(SPI)															
	1	保留															
[14]	P0_ALT[6]	P0.6 功能选择位 位P0_MFP[6] 和P0_ALT[6].决定 P0.6功能 <table border="1"> <tr><th>P0_ALT[6]</th><th>P0_MFP[6]</th><th>P0.6 功能</th></tr> <tr><td>0</td><td>0</td><td>P0.6</td></tr> <tr><td>0</td><td>1</td><td>保留</td></tr> <tr><td>1</td><td>0</td><td>MISO_0(SPI)</td></tr> <tr><td>1</td><td>1</td><td>保留</td></tr> </table>	P0_ALT[6]	P0_MFP[6]	P0.6 功能	0	0	P0.6	0	1	保留	1	0	MISO_0(SPI)	1	1	保留
P0_ALT[6]	P0_MFP[6]	P0.6 功能															
0	0	P0.6															
0	1	保留															
1	0	MISO_0(SPI)															
1	1	保留															

[13]	P0_ALT[5]	P0.5功能选择位 位P0_MFP[5]和 P0_ALT[5].决定P0.5功能
		P0_ALT[5] P0_MFP[5] P0.5 功能
		0 0 P0.5
		0 1 保留
		0 MOSI_0(SPI)
[12]	P0_ALT[4]	P0.4功能选择位 位P0_MFP[4] 和 P0_ALT[4].决定P0.4功能
		P0_ALT[4] P0_MFP[4] P0.4功能
		0 0 P0.4
		0 1 保留
		1 0 SPISS0(SPI)
[11:8]	保留	1 1 PWM5 (PWM)
		保留
[7:0]	P0_MFP[7:0]	P0复用功能选择位 P0的引脚功能取决于P0_MFP和P0_ALT. 详细信息参考P0_ALT描述

Port1复用功能控制寄存器(P1_MFP)

寄存器	偏移地址	R/W	描述	复位值
P1_MFP	GCR_BA+0x34	R/W	P1复用功能和输入类型控制寄存器	0x0000_0000

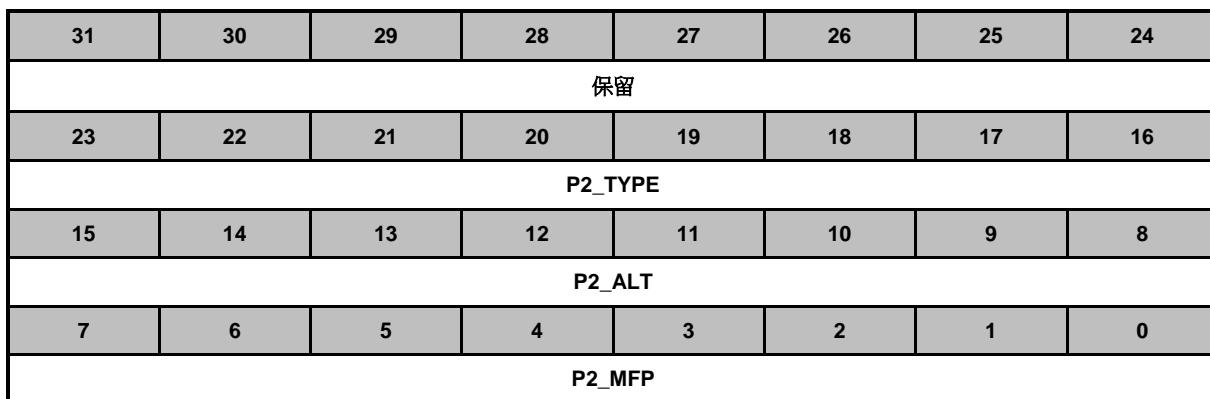


位	描述																
[31:24]	保留	保留															
[23:16]	P1_TYPE[n]	<p>P1[7:0]TTL或施密特触发功能使能位 0 = P1[7:0] I/O输入为TTL功能 1 = P1[7:0] I/O输入为施密特触发功能</p>															
[15:14]	保留	保留															
[13]	P1_ALT[5]	<p>P1.5功能选择位 位P1_MFP[5] 和 P1_ALT[5].决定 P1.5 功能</p> <table border="1"> <tr><th>P1_ALT[5]</th><th>P1_MFP[5]</th><th>P1.5功能</th></tr> <tr><td>0</td><td>0</td><td>P1.5</td></tr> <tr><td>0</td><td>1</td><td>AIN5(ADC)</td></tr> <tr><td>1</td><td>0</td><td>保留</td></tr> <tr><td>1</td><td>1</td><td>ACMP0_P(ACMP)</td></tr> </table>	P1_ALT[5]	P1_MFP[5]	P1.5功能	0	0	P1.5	0	1	AIN5(ADC)	1	0	保留	1	1	ACMP0_P(ACMP)
P1_ALT[5]	P1_MFP[5]	P1.5功能															
0	0	P1.5															
0	1	AIN5(ADC)															
1	0	保留															
1	1	ACMP0_P(ACMP)															
[12]	P1_ALT[4]	<p>P1.4功能选择位 位P1_MFP[4] 和 P1_ALT[4]决定P1.4功能</p> <table border="1"> <tr><th>P1_ALT[4]</th><th>P1_MFP[4]</th><th>P1.4功能</th></tr> <tr><td>0</td><td>0</td><td>P1.4</td></tr> <tr><td>0</td><td>1</td><td>AIN4(ADC)</td></tr> <tr><td>1</td><td>0</td><td>保留</td></tr> <tr><td>1</td><td>1</td><td>ACMP0_N(ACMP)</td></tr> </table>	P1_ALT[4]	P1_MFP[4]	P1.4功能	0	0	P1.4	0	1	AIN4(ADC)	1	0	保留	1	1	ACMP0_N(ACMP)
P1_ALT[4]	P1_MFP[4]	P1.4功能															
0	0	P1.4															
0	1	AIN4(ADC)															
1	0	保留															
1	1	ACMP0_N(ACMP)															
[11]	P1_ALT[3]	P1.3功能选择位															

		位P1_MFP[3] 和 P1_ALT[3]. 决定 P1.3 功能															
		<table border="1"> <thead> <tr> <th>P1_ALT[3]</th><th>P1_MFP[3]</th><th>P1.3功能</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>P1.3</td></tr> <tr> <td>0</td><td>1</td><td>AIN3(ADC)</td></tr> <tr> <td>1</td><td>0</td><td>TXD(UART)</td></tr> <tr> <td>1</td><td>1</td><td>ACMP0_P(ACMP)</td></tr> </tbody> </table>	P1_ALT[3]	P1_MFP[3]	P1.3功能	0	0	P1.3	0	1	AIN3(ADC)	1	0	TXD(UART)	1	1	ACMP0_P(ACMP)
P1_ALT[3]	P1_MFP[3]	P1.3功能															
0	0	P1.3															
0	1	AIN3(ADC)															
1	0	TXD(UART)															
1	1	ACMP0_P(ACMP)															
[10]	P1_ALT[2]	P1.2功能选择位 位P1_MFP[2] 和 P1_ALT[2]. 决定P1.2功能															
		<table border="1"> <thead> <tr> <th>P1_ALT[2]</th><th>P1_MFP[2]</th><th>P1.2功能</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>P1.</td></tr> <tr> <td>0</td><td>1</td><td>AIN2(ADC)</td></tr> <tr> <td>1</td><td>0</td><td>RXD(UART)</td></tr> <tr> <td>1</td><td>1</td><td>ACMP0_P(ACMP)</td></tr> </tbody> </table>	P1_ALT[2]	P1_MFP[2]	P1.2功能	0	0	P1.	0	1	AIN2(ADC)	1	0	RXD(UART)	1	1	ACMP0_P(ACMP)
P1_ALT[2]	P1_MFP[2]	P1.2功能															
0	0	P1.															
0	1	AIN2(ADC)															
1	0	RXD(UART)															
1	1	ACMP0_P(ACMP)															
[9:8]	保留	保留															
[7:0]	P1_MFP[7:0]	P1复用功能选择位 P1引脚功能取决于P1_MFP and P1_ALT 细节请参考P1_ALT描述															

Port2复用功能控制寄存器(P2_MFP)

寄存器	偏移地址	R/W	描述	复位值
P2_MFP	GCR_BA+0x38	R/W	P2复用功能和输入类型控制寄存器	0x0000_0000



位	描述																
[31:24]	保留	保留															
[23:16]	P2_TYPE[n]	<p>P2[7:0]施密特触发输入功能使能位 0 = P2[7:0] I/O输入为TTL功能 1 = P2[7:0] I/O输入为施密特触发功能</p>															
[15:14]	保留	保留															
[13]	P2_ALT[5]	<p>P2.5功能选择位 位P2_MFP[5] 和 _ALT[5]决定 P2.5功能</p> <table border="1"> <tr> <th>P2_ALT[5]</th> <th>P2_MFP[5]</th> <th>P2.5 功能</th> </tr> <tr> <td>0</td> <td>0</td> <td>P2.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>保留</td> </tr> <tr> <td>1</td> <td>0</td> <td>PWM3(PWM)</td> </tr> <tr> <td></td> <td>1</td> <td>保留</td> </tr> </table>	P2_ALT[5]	P2_MFP[5]	P2.5 功能	0	0	P2.5	0	1	保留	1	0	PWM3(PWM)		1	保留
P2_ALT[5]	P2_MFP[5]	P2.5 功能															
0	0	P2.5															
0	1	保留															
1	0	PWM3(PWM)															
	1	保留															

[12]	P2_ALT[4]	P2.4功能选择位 位P2_MFP[4] 和 P2_ALT[4].决定 P2.4功能													
		<table border="1"><thead><tr><th>P2_ALT[4]</th><th>P2_MFP[4]</th><th>P2.4 功能</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>P2.4</td></tr><tr><td>0</td><td>1</td><td>保留</td></tr><tr><td>1</td><td>0</td><td>PWM2(PWM)</td></tr><tr><td>1</td><td>1</td><td>保留</td></tr></tbody></table>	P2_ALT[4]	P2_MFP[4]	P2.4 功能	0	0	P2.4	0	1	保留	1	0	PWM2(PWM)	1
P2_ALT[4]	P2_MFP[4]	P2.4 功能													
0	0	P2.4													
0	1	保留													
1	0	PWM2(PWM)													
1	1	保留													
[11:8]	保留	保留													
[7:0]	P2_MFP[7:0]	P2复用功能选择位 P2 引脚功能取决于P2_MFP 和 P2_ALT 细节请参考P2_ALT描述													

Port3复用功能控制寄存器(P3_MFP)

寄存器	偏移地址	R/W	描述	复位值
P3_MFP	GCR_BA+0x3C	R/W	P3复用功能和输入类型控制寄存器	0x0000_0000

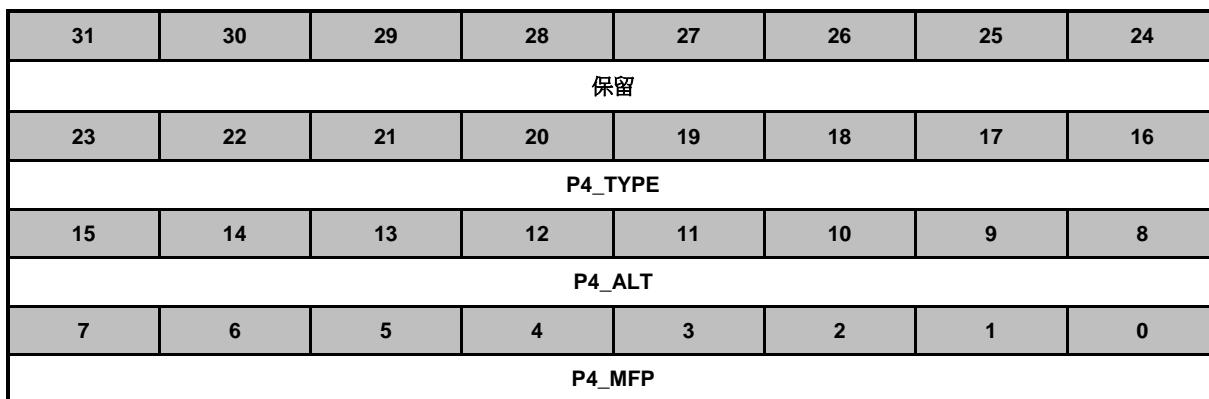
31	30	29	28	27	26	25	24
保留							P32CPP1
23	22	21	20	19	18	17	16
P3_TYPE							
15	14	13	12	11	10	9	8
P3_ALT							
7	6	5	4	3	2	1	0
P3_MFP							

位	描述																
[31:25]	保留	保留															
[24]	P32CPP1	P3.2功能选择扩展 0 = P3.2 功能由P3_ALT[2]和 P3_MFP[2]决定. 1 = P3.2设置为ACMP1的CPP1功能															
[23:16]	P3_TYPE[n]	P3[7:0] TTL或施密特触发输入功能使能位 0 = P3[7:0] I/O输入为TTL功能 1 = P3[7:0] I/O输入为施密特触发功能															
[15:14]	保留	保留															
[13]	P3_ALT[5]	P3.5功能选择位 位P3_MFP[5] 和 P3_ALT[5]决定 P3.5功能 <table border="1"> <thead> <tr> <th>P3_ALT[5]</th> <th>P3_MFP[5]</th> <th>P3.5 功能</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>P3.5</td></tr> <tr><td>0</td><td>1</td><td>T1(Timer)</td></tr> <tr><td>1</td><td>0</td><td>SCL0(I²C0)</td></tr> <tr><td>1</td><td>1</td><td>ACMP1_P (ACMP)</td></tr> </tbody> </table>	P3_ALT[5]	P3_MFP[5]	P3.5 功能	0	0	P3.5	0	1	T1(Timer)	1	0	SCL0(I ² C0)	1	1	ACMP1_P (ACMP)
P3_ALT[5]	P3_MFP[5]	P3.5 功能															
0	0	P3.5															
0	1	T1(Timer)															
1	0	SCL0(I ² C0)															
1	1	ACMP1_P (ACMP)															
[12]	P3_ALT[4]	P3.4功能选择位 位P3_MFP[4]和P3_ALT[4].决定P3.4功能 <table border="1"> <thead> <tr> <th>P3_ALT[4]</th> <th>P3_MFP[4]</th> <th>P3.4 功能</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>P3.4</td></tr> <tr><td>0</td><td>1</td><td>T0(Timer0 外部事件计数器)</td></tr> <tr><td>1</td><td></td><td>SDA0(I²C)</td></tr> </tbody> </table>	P3_ALT[4]	P3_MFP[4]	P3.4 功能	0	0	P3.4	0	1	T0(Timer0 外部事件计数器)	1		SDA0(I ² C)			
P3_ALT[4]	P3_MFP[4]	P3.4 功能															
0	0	P3.4															
0	1	T0(Timer0 外部事件计数器)															
1		SDA0(I ² C)															

		1	1	ACMP1_P (ACMP)	
[11]	保留	保留			
[10]	P3_ALT[2]	P3.2功能选择位 位P3_MFP[2] 和 P3_ALT[2]决定P3.2功能 注意：如果P32CPP1置1，P3_ALT[2] 和 P3_MFP[2]的设置将被忽略掉，且P3.2将被设置成ACMP1的CPP1功能	P3_ALT[2]	P3_MFP[2]	P3.2 功能
		0	0	P3.2	
		0	1	/INT0	
		1	0	T0EX	
		1	1	STADC(ADC)	
[9:8]	保留	保留.			
[7:0]	P3_MFP[7:0]	P3复用功能选择位 P3引脚功能取决于P3_MFP 和 P3_ALT. 细节请参考P3_ALT描述			

Port4复用功能控制寄存器(P4_MFP)

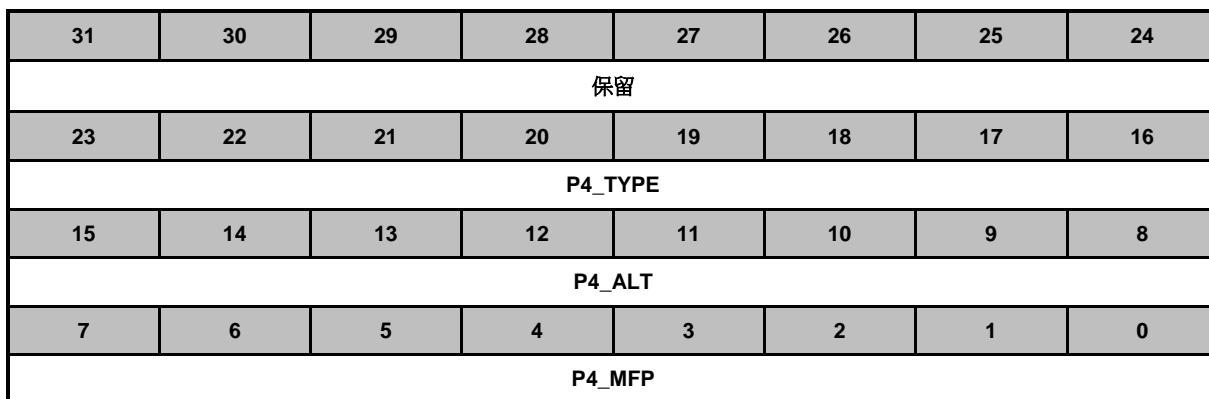
寄存器	偏移地址	R/W	描述				复位值
P4_MFP	GCR_BA+0x40	R/W	P4复用功能和输入类型控制寄存器				0x0000_00C0



位	描述		
[31:24]	保留	保留	
[23:16]	P4_TYPE[n]	P4[7:0] TTL或施密特触发输入功能使能位 0 = P4[7:0] I/O输入为TTL功能 1 = P4[7:0] I/O输入为施密特触发功能	
[15]	P4_ALT[7]	P4.7功能选择位 位P4_MFP[7] 和 P4_ALT[7].决定P4.7功能	
		P4_ALT[7]	P4_MFP[7]
		0	0
		0	1
		1	x
	P4_ALT[6]	P4.6功能选择位 位P4_MFP[6] 和 P4_ALT[6]决定P4.6功能	
[14]		P4_ALT[6]	P4_MFP[6]
		0	0
		0	1
		1	x
[13:8]	保留	保留	
[7:0]	P4_MFP[7:0]	P4复用功能选择位 P4引脚功能取决于P4_MFP 和 P4_ALT 细节参考P4_ALT描述	

Port5复用功能控制寄存器(P5_MFP)

寄存器	偏移地址	R/W	描述				复位值
P5_MFP	GCR_BA+0x44	R/W	P5复用功能和输入类型控制寄存器				0x0000_0000



位	描述													
[31:24]	保留	保留												
[23:16]	P5_TYPE[n]	<p>P5[7:0] TTL或施密特触发输入功能使能位 0 = P5[7:0] I/O施密特触发输入功能失能 1 = P5[7:0] I/O施密特触发输入功能使能</p>												
[15:10]	保留	保留												
[9]	P5_ALT[1]	<p>P5.1功能选择位 位P5_MFP[1] 和 P5_ALT[1]决定 P5.1 功能</p> <table border="1"> <tr><th>P5_ALT[1]</th><th>P5_MFP[1]</th><th>P5.1 功能</th></tr> <tr><td>0</td><td>0</td><td>P5.1</td></tr> <tr><td>0</td><td>1</td><td>XTAL2 (输出管脚)</td></tr> <tr><td>1</td><td>x</td><td>保留</td></tr> </table> <p>注: 为了使能外部XTAL功能，XTLCLK_EN (PWRCON[1:0])位及外部HXT或LXT晶振控制寄存器也必须被设置。</p>	P5_ALT[1]	P5_MFP[1]	P5.1 功能	0	0	P5.1	0	1	XTAL2 (输出管脚)	1	x	保留
P5_ALT[1]	P5_MFP[1]	P5.1 功能												
0	0	P5.1												
0	1	XTAL2 (输出管脚)												
1	x	保留												
[8]	P5_ALT[0]	<p>P5.0功能选择位 位P5_MFP[0] 和 P5_ALT[0]决定 P5.0功能</p> <table border="1"> <tr><th>P5_ALT[0]</th><th>P5_MFP[0]</th><th>P5.0 功能</th></tr> <tr><td>0</td><td>0</td><td>P5.0</td></tr> <tr><td>0</td><td>1</td><td>XTAL1 (输入管脚)</td></tr> <tr><td>1</td><td>x</td><td>保留</td></tr> </table> <p>注: 为了使能外部XTAL功能，XTLCLK_EN (PWRCON[1:0])位及外部HXT或LXT晶振控制寄存器也必须被设置。</p>	P5_ALT[0]	P5_MFP[0]	P5.0 功能	0	0	P5.0	0	1	XTAL1 (输入管脚)	1	x	保留
P5_ALT[0]	P5_MFP[0]	P5.0 功能												
0	0	P5.0												
0	1	XTAL1 (输入管脚)												
1	x	保留												

[7:0]	P5_MFP[7:0]	P5复用功能选择位 P5引脚功能取决于P5_MFP 和P5_ALT 细节请参考P5_ALT描述
-------	--------------------	---

HIRC校准控制寄存器(IRCTRIMCTL)

寄存器	偏移地址	R/W	描述	复位值
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC校准控制寄存器	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
TRIM_RETRY_CNT		TRIM_LOOP		保留			TRIM_SEL

位	描述											
[31:8]	保留	保留										
[7:6]	TRIM_RETRY_CNT[1:0]	<p>校准数值最大限制 该部分也定义了自动校准电路会在HIRC频率锁定前多少时间内试图更新HIRC校准值。 一旦HIRC被锁定，内部校准值更新计数器将会被复位。 假如校准值更新计数器到达限制值，但HIRC还没被锁定，那么自动校准操作将被禁用，TRIM_SEL将被清零。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">TRIM_RETRY_CNT</td> <td style="width: 50%;">Trim Retry数值最大限制</td> </tr> <tr> <td>00</td> <td>数值最大限制值是 64</td> </tr> <tr> <td>01</td> <td>数值最大限制值是128</td> </tr> <tr> <td>10</td> <td>数值最大限制值是256</td> </tr> <tr> <td>11</td> <td>数值最大限制值是512</td> </tr> </table>	TRIM_RETRY_CNT	Trim Retry数值最大限制	00	数值最大限制值是 64	01	数值最大限制值是128	10	数值最大限制值是256	11	数值最大限制值是512
TRIM_RETRY_CNT	Trim Retry数值最大限制											
00	数值最大限制值是 64											
01	数值最大限制值是128											
10	数值最大限制值是256											
11	数值最大限制值是512											
[5:4]	TRIM_LOOP[1:0]	<p>校准计算循环 该部分定义校准值是基于多少个LXT时钟计算出来的。 例如：假如TRIM_LOOP设置为“00”，自动校准电路将会基于4个LXT时钟偏差的平均值计算校准值。 00 = 校准值基于4个LXT时钟偏差的平均值计算出来的 01 = 校准值基于8个LXT时钟偏差的平均值计算出来的 10 = 校准值基于16个LXT时钟偏差的平均值计算出来的 11 = 校准值基于32个LXT时钟偏差的平均值计算出来的</p>										
[3:1]	保留	保留										

位	描述
[0]	TRIM_SEL 校准频率选择位 该位用来使能HIRC自动校准 当该位设置为1时，HIRC自动校准功能会基于LXT参考时钟自动校准HIRC为22.1184 MHz。 在自动校准过程中，如果检测到LXT时钟错误或者重试校准计数器到达限制值，该位会自动清零。 0 = HIRC自动校准功能禁止 1 = HIRC自动校准功能使能且HIRC被校准到22.1184 MHz

HIRC校准中断使能寄存器(IRCTRIMIEN)

寄存器	偏移地址	R/W	描述	复位值
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC校准中断使能寄存器	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
保留							
7	6	5	4	3	2	1	0
保留					32K_ERR_IEN	TRIM_FAIL_IE_N	保留

位	描述	
[31:3]	保留	保留
[2]	32K_ERR_IEN	<p>LXT时钟错误中断使能位 该位控制着在自动校准运算过程中，LXT时钟错误是否会使CPU产生中断。 如果该位为1，并且在自动校准运算期间，32K_ERR_INT被置位，将会触发中断，提示LXT时钟不准确。 0 = 禁用32K_ERR_INT触发CPU中断 1 = 使能32K_ERR_INT触发CPU中断 </p>
[1]	TRIM_FAIL_IEN	<p>校准失败中断使能位 该位控制当HIRC校准次数达到更新极限值，但HIRC频率仍然没有锁定在TRIM_SEL设定的目标频率时，是否产生一个中断。 如果该位为1，并且在自动校准运算期间，TRIM_FAIL_INT被置位，将会触发中断，提示HIRC校准次数已经达到了自动校准更新限制次数。 0 = 禁用TRIM_FAIL_INT 触发CPU中断 1 = 使能TRIM_FAIL_INT触发CPU中断 </p>
[0]	保留	保留

HIRC校准中断状态寄存器(IRCTRIMINT)

寄存器	偏移地址	R/W	描述	复位值
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC校准中断状态寄存器	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					32K_ERR_INT	TRIM_FAIL_INT	FREQ_LOCK

位	描述	
[31:3]	保留	保留
[2]	32K_ERR_INT	<p>LXT时钟错误中断状态 该位表明LXT时钟频率不精确。一旦该位置1，自动校准运算将会停止，且TRIM_SEL会被硬件自动清零。</p> <p>如果该位被置1且32K_ERR_IEN为高，将触发中断来提示LXT时钟频率不精确。软件可以通过向该位写1来清零。</p> <p>0 = LXT时钟频率是精确的 1 = LXT时钟频率是不精确的</p>
[1]	TRIM_FAIL_INT	<p>校准失败中断状态 该位表明HIRC校准次数已经达到极限值时，HIRC频率仍然没有锁定。一旦该位置1，自动校准运算将会停止，且TRIM_SEL会被硬件自动清零。</p> <p>如果该位被置1且TRIM_FAIL_IEN为高，将触发中断表明HIRC校准次数已经达到限制值。软件可以通过向该位写1来清零。</p> <p>0 = 校准值更新限制次数没到达 1 = 校准值更新限制次数到达且HIRC频率仍没锁定</p>
[0]	FREQ_LOCK	<p>HIRC频率锁定状态 该位表明HIRC频率锁定在22.1184 MHz. 该位为只读状态位，不会触发任何中断</p>

写保护寄存器控制寄存器(REGWRPROT)

有些寄存器需要被保护起来，以防止误写而影响芯片运行。这些系统寄存器在上电复位后就受保护，直到用户关闭寄存器保护。用户需要编程这些受保护寄存器的话，需要一个解锁时序。解锁时序是依次连续的往地址为0x5000_0100的控制器写入0x59、0x16、0x88。在写以上三个数据的过程中，写入其他任何数据、不同顺序写、或往别的地址写入数据都将终止解锁时序。

写保护功能禁用后，用户可以查看保护禁用位（地址0x5000_0100位0），1表示保护禁用，0表示保护使能。写保护禁用后，用户可以更新目标保护寄存器的值，往地址0x5000_0100写入任何值将使能寄存器保护。

写该寄存器来禁用/使能寄存器保护，读该寄存器获取REGPROTDIS状态。

寄存器	偏移地址	R/W	描述	复位值
REGWRPROT	GCR_BA+0x100	R/W	寄存器写保护控制寄存器	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
REGWRPROT[7:1]							REGWRPROT [0] REGPROTDIS

位	描述	
[31:16]	保留	保留
[7:0]	REGWRPROT	写保护代码寄存器(只写) 一些寄存器具有写保护功能。写这些寄存器前必须按照顺序依次往寄存器写入0x59、0x16、0x88来解锁写保护功能。在这个时序结束后，REGPROTDIS位会被置1，写保护寄存器可以正常写入数据。
[0]	REGPROTDIS	写保护寄存器解锁标志 (只读) 0 = 写保护寄存器的写保护功能使能，任何写入写保护寄存器的数据将被忽略 1 = 写保护寄存器的写保护功能被禁用 保护寄存器如下表所示
	寄存器	地址
	IPRSTC1	0x5000_0008
	BODCR	0x5000_0018
		注释
		外设复位控制寄存器1
		欠压检测器控制寄存器

	PWRCON	0x500_0200	位[6]没有受保护，电源唤醒中断将清除该位
	APBCLK bit[0]	0x5000_0208	位[0]是看门狗时钟使能位
	CLKSEL0	0x5000_0210	HCLK和SysTick时钟源选择
	CLKSEL1 bit[1:0]	0x5000_0214	看门狗时钟源选择
	NMI_SEL bit[8]	0x5000_0380	NMI中断使能
	ISPCON	0x5000_C000	Flash ISP控制
	ISPTRG	0x5000_C010	ISP触发控制
	WTCR	0x4000_4000	看门狗定时器控制
注: 写保护位在其描述旁边会标注“(写保护)”。			

5.2.8 系统定时器(SysTick)

Cortex-M0 包含一个集成的系统定时器, SysTick. SysTick 提供一种简单的, 24位写清零, 下数计数, 计数至0后自装载的计数器, 有一个灵活的控制机制。计数器可作为实时操作系统的节拍定时器或者作为一个简单的计数器。

使能后, 系统定时器从SysTick 当前值寄存器(SYST_CVR)的值向下计数到0, 并在下一个时钟边沿, 重新加载SysTick重装载值寄存器(SYST_RVR)的值到SysTick当前值寄存器(SYST_CVR), 然后随接下来的时钟递减。当计数器减到0时, 标志位COUNTFLAG置位, 标志位COUNTFLAG是读清0的。

复位后, SYST_CVR 的值未知。使能前, 软件应该写该寄存器使其清0。这样确保定时器在使能后以 SYST_RVR中的值计数, 而非任意值。

若SYST_RVR 是0, 在重新加载后, 定时器将保持当前值0, 这种机制可以用来在不使用系统定时器的使能位的情形下禁用系统定时器。

详情请参考“ARM® Cortex®-M0 Technical Reference Manual”与“ARM® v6-M Architecture Reference Manual”。

5.2.9 NUC029xAN系统定时器控制寄存器映射

R: 只读, **W:** 只写, **R/W:** 可读写

寄存器	偏移量	R/W	描述	复位后的值
SYST 基地址:				
SYST_BA = 0xE000_E010				
SYST_CSR	SYST_BA+0x00	R/W	SysTick控制与状态寄存器	0x0000_0000
SYST_RVR	SYST_BA+0x04	R/W	SysTick重装载值寄存器	0xXXXX_XXXX
SYST_CVR	SYST_BA+0x08	R/W	SysTick 当前值寄存器	0xXXXX_XXXX

SysTick 控制和状态寄存器 (SYST_CSR)

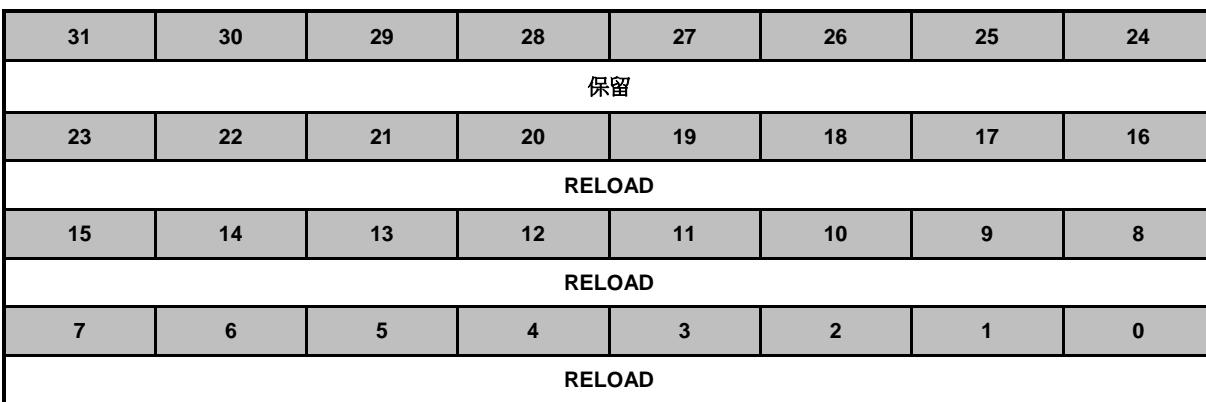
寄存器	偏移量	R/W	描述	复位后的值
SYST_CSR	SYST_BA+0x00	R/W	SysTick控制和状态寄存器	0x0000_0000

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					CLKSRC	TICKINT	ENABLE

位	描述	
[31:17]	保留	保留
[16]	COUNTFLAG	系统滴答计数器标志位 从上次该寄存器被读，如果定时器计数到0，则返回1。 计数由1到0时，COUNTFLAG 置位。 读该位或向当前值寄存器 (SYST_CVR) 写时，COUNTFLAG 被清零。
[15:3]	保留	保留
[2]	CLKSRC	系统滴答时钟源选择位 1= 内核时钟用于SysTick. 0= 时钟源可选，参考STCLK_S(CLKSEL0[5:3]).
[1]	TICKINT	系统滴答中断使能位 1 : 向下计数到0将导致SysTick 中断发生。通过软件寄存器写操作清SysTick 当前值寄存器的值将不会导致SysTick 发生中断。 0 : 向下计数到0不会引起SysTick发生中断。软件可以使用COUNTFLAG 来确定是否已经发生计数到0
[0]	ENABLE	系统滴答计数器使能位 1 : 计数器开始计数，并运行于周期模式。 0 : 禁用计数器

SysTick重加载值寄存器 (SYST_RVR)

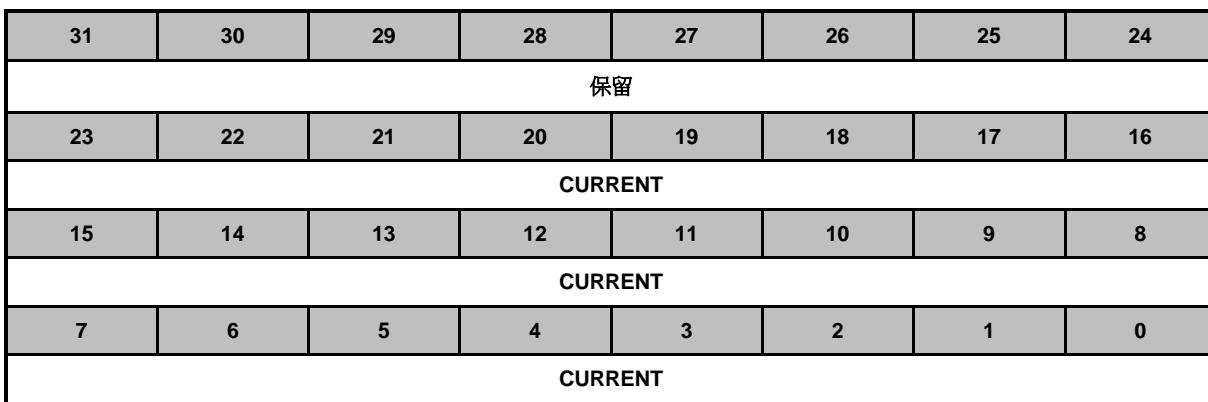
寄存器	偏移量	R/W	描述	复位后的值
SYST_RVR	SYST_BA+0x04	R/W	SysTick重加载值寄存器	0xFFFF_FFFF



位	描述	
[31:24]	保留	保留
[23:0]	RELOAD	系统滴答重加载值 当计数器达到0时，该值加载到当前值寄存器。

SysTick当前值寄存器 (SYST_CVR)

寄存器	偏移量	R/W	描述	复位后的值
SYST_CVR	SYST_BA+0x08	R/W	SysTick当前值寄存器	0xFFFF_FFFF



位	描述		
[31:24]	保留	保留	
[23:0]	CURRENT	系统滴答当前值 当前计数值，为采样时刻的计数器的值，计数器不提供读-修改-写保护功能，该寄存器为写清零的，软件写入任何值将清寄存器为0。不支持位RAZ (参看 SysTick重新加载值寄存器)。	

5.2.10 NUC029FAE系统定时器控制寄存器映射

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位后的值
SCS 基地址:				
SCS_BA = 0xE000_E000				
SYST_CSR	SCS_BA+0x10	R/W	SysTick控制与状态寄存器	0x0000_0000
SYST_RVR	SCS_BA+0x14	R/W	SysTick重装载值寄存器	0xFFFF_FFFF
SYST_CVR	SCS_BA+0x18	R/W	SysTick 当前值寄存器	0xFFFF_FFFF

SysTick控制和状态寄存器 (SYST_CSR)

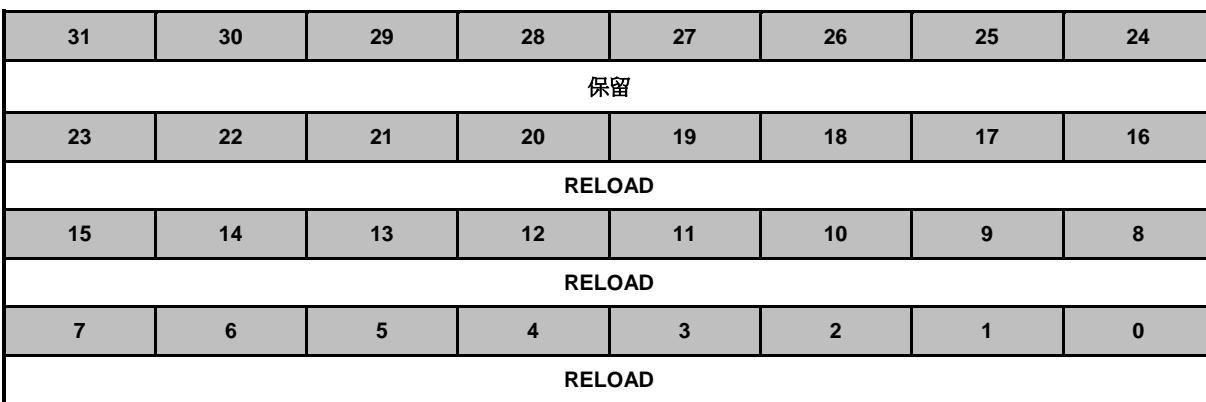
寄存器	偏移地址	R/W	描述	复位值
SYST_CSR	SCS_BA+0x10	R/W	SysTick控制和状态寄存器	0x0000_0000

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					CLKSRC	TICKINT	ENABLE

位	描述	
[31:17]	保留	保留
[16]	COUNTFLAG	系统滴答计数器标志位 从上次读取该寄存器后，如果定时器计数到0，则返回1 计数从1到0转变时，COUNTFLAG置位 读该位或写当前值寄存器将使COUNTFLAG清零
[15:3]	保留	保留
[2]	CLKSRC	系统滴答时钟源选择位 0 = 时钟源是可选的，参考STCLK_S(CLKSEL0[5:3]). 1 = 内核时钟用于SysTick 定时器
[1]	TICKINT	系统滴答中断使能位 0 = 向下计数到0不会引起SysTick 异常而挂起。软件可以根据COUNTFLAG标志位来确定是否已经发生计数到0的情况。 1 = 向下计数到0将引起SysTick异常而挂起。软件清除SysTick当前值寄存器值将不会导致SysTick挂起。
[0]	ENABLE	系统滴答计数器使能位 0 = 计数器禁用 1 = 计数器使能并运行于multi-shot方式

SysTick 重加载值寄存器 ((SYST_RVR)

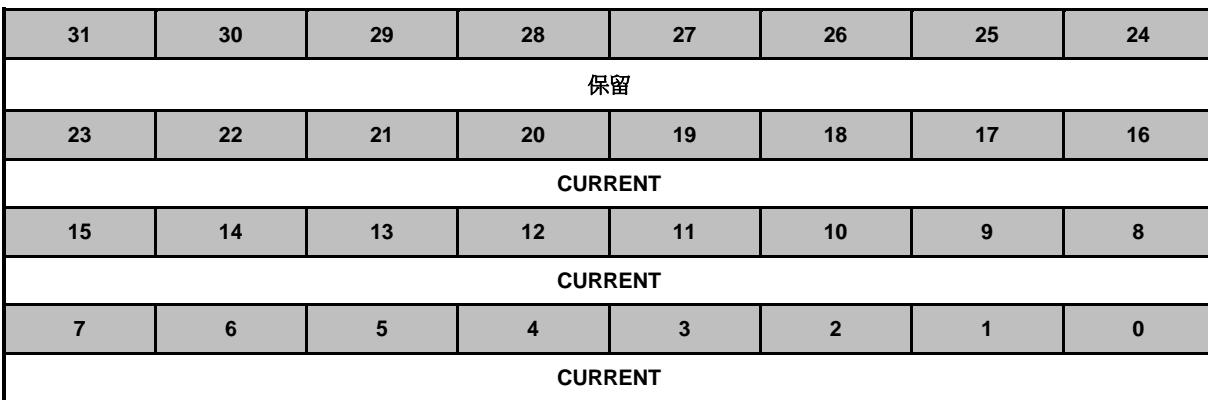
寄存器	偏移地址	R/W	描述	复位值
SYST_RVR	SCS_BA+0x14	R/W	SysTick重加载值寄存器	0XXXX_XXXX



位	描述	
[31:24]	保留	保留
[23:0]	RELOAD	系统滴答重加载值 当计数器计数到0时，该值将加载到当前值寄存器

SysTick当前值寄存器 (SYST_CVR)

寄存器	偏移地址	R/W	描述	复位值
SYST_CVR	SCS_BA+0x18	R/W	SysTick当前值寄存器	0xFFFF_FFFF



位	描述	
[31:24]	保留	保留
[23:0]	CURRENT	系统滴答当前值 当前计数值。该值为采样时刻的计数器的值。计数器不提供读修改写保护功能。该寄存器有写清除功能，软件写入任何值到该寄存器将清除该寄存器值为0。不支持位RAZ (参看 SysTick重新加载值寄存器)。

5.2.11 嵌套向量中断控制器(NVIC)

Cortex-M0提供中断控制器，作为异常模式的组成部分，称之为“嵌套向量中断控制器(NVIC)”。它与处理器内核紧密联系，并具有以下特性：

- 支持嵌套和向量中断
- 自动保存和恢复上下文
- 简化的精确的中断延迟

NVIC对所有支持的异常按优先级排序并处理，所有异常在“处理器模式”处理。NVIC结构支持具有四级优先级的32个(IRQ[31:0])离散中断。所有的中断和大多数系统异常可以配置为不同的优先级。当中断发生时，NVIC将比较新中断与当前中断的优先级，如果新中断优先级高于当前中断，则新中断将代替当前中断被处理。

当任何中断被响应时，中断服务程序（ISR）的起始地址从内存的向量表中取得。不需要由软件确定响应哪个中断，也不要软件跳转到相应ISP的起始地址。当取得起始地址时，NVIC将自动保存处理器状态，包括以下寄存器“PC, PSR, LR, R0~R3, R12”的值到栈中。在ISR结束时，NVIC 将从栈中恢复相关寄存器的值，恢复正常操作，因此处理器将花费更少的并且确定的时间去处理中断请求。

NVIC支持末尾连锁“Tail Chaining”，有效处理尾对尾中断“back-to-back interrupts”，即无需重复保存和恢复当前状态，从而减少从当前ISR结束切换到等待处理的ISR的延迟时间。NVIC还支持晚到“Late Arrival”，可以提升同时发生的ISR的效率。在当前ISR开始执行（保存处理器状态并获取起始地址阶段）之前，如果较高优先级中断请求发生，NVIC将立即选择处理更高优先级的中断，从而提高了实时性。

详情请参考“ARM® Cortex®-M0 Technical Reference Manual”与“ARM® v6-M Architecture Reference Manual”。

5.2.11.1 异常模式和系统中断映射

下表列出了NuMicro® NUC029系列支持的异常模型。软件可以对其中一些异常以及所有中断设置4级优先级。最高用户可配置优先级记为“0”，最低优先级记为“3”，所有用户可配置的优先级的默认值为“0”。注意：优先级“0”在整个系统中为第4优先级，排在“Reset”，“NMI”与“Hard Fault”之后。

异常名称	向量号	优先级
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
保留	4 ~ 10	保留
SVCall	11	可配置
保留	12 ~ 13	保留
PendSV	14	可配置
SysTick	15	可配置

Interrupt (IRQ0 ~ IRQ31)	16 ~ 47	可配置
--------------------------	---------	-----

表 5-3 异常模型

异常号	中断号 (Bit In Interrupt Registers)	中断名	源IP	中断描述	掉电唤醒
1 ~ 15	-	-	-	系统异常	-
16	0	BOD_INT	Brown-out	欠压检测中断	Yes
17	1	WDT_INT	WDT	看门狗定时器中断	Yes
18	2	EINT0	GPIO	P3.2 脚上的外部信号中断	Yes
19	3	EINT1	GPIO	P3.3 脚上的外部信号中断	Yes
20	4	P0/1_INT	GPIO	P0[7:0] / P1[7:0] 外部信号中断	Yes
21	5	P2/3/4_INT	GPIO	P2[7:0]/P3[7:0]/P4[7:0] 外部信号中断, 除 P3.2 和 P3.3	Yes
22	6	PWMA_INT	PWM0~3	PWM0, PWM1, PWM2 和 PWM3 中断	No
23	7	PWMB_INT	PWM4~7	PWM4, PWM5, PWM6 和 PWM7 中断	No
24	8	TMR0_INT	TMR0	Timer 0 中断	Yes
25	9	TMR1_INT	TMR1	Timer 1 中断	Yes
26	10	TMR2_INT	TMR2	Timer 2 中断	Yes
27	11	TMR3_INT	TMR3	Timer 3 中断	Yes
28	12	UART0_INT	UART0	UART0 中断	Yes
29	13	UART1_INT	UART1	UART1 中断	Yes
30	14	SPI0_INT	SPI0	SPI0 中断	No
31	15	SPI1_INT	SPI1	SPI1 中断	No
32 ~ 33	16 ~ 17	-	-	-保留	-
34	18	I2C0_INT	I ² C0	I ² C0 中断	Yes
35	19	I2C1_INT	I ² C1	I ² C1 中断	Yes
36 ~ 40	20 ~ 24	-	-	-保留	-
41	25	ACMP01_INT	ACMP0/1	模拟比较器0和模拟比较器1的中断	Yes
42	26	ACMP23_INT	ACMP2/3	模拟比较器2和模拟比较器3的中断	Yes
43	27	-	-	保留	-
44	28	PWRWU_INT	CLKC	从掉电状态唤醒的时钟控制器中断	Yes
45	29	ADC_INT	ADC	ADC 中断	No
46 ~ 47	30 ~ 31	-	-	-保留	-

表 5-4 NuMicro[®] NUC029xAN 系统中断映射向量表

异常号	中断号 (Bit In Interrupt Registers)	中断名	源IP	中断描述	掉电唤醒
1 ~ 15	-	-	-	系统异常	-
16	0	BOD_INT	Brown-out	欠压检测中断	Yes
17	1	WDT_INT	WDT	看门狗定时器中断	Yes
18	2	EINT0	GPIO	P3.2 脚上的外部信号中断	Yes
19	3	-	-	保留	-
20	4	P0/1_INT	GPIO	P0[7:0] / P1[7:0] 外部信号中断	Yes
21	5	P2/3/4_INT	GPIO	P2[7:0]/P3[7:0]/P4[7:0] 外部信号中断, 除 P3.2	Yes
22	6	PWM_INT	PWM	PWM中断	No
23	7	BRAKE_INT	PWM	PWM刹车中断	No
24	8	TMR0_INT	TMR0	Timer 0 中断	Yes
25	9	TMR1_INT	TMR1	Timer 1 中断	Yes
26 ~ 27	10 ~ 11	-	-	保留	-
28	12	UART_INT	UART	UART中断	Yes
29	13	-	-	保留	-
30	14	SPI_INT	SPI	SPI中断	No
31	15	-	-	保留	-
32	16	GP5_INT	GPIO	P5外部中断	Yes
33	17	HIRC_TRIM_INT	HIRC	-HIRC校准中断	NO
34	18	I2C_INT	I ² C	I ² C中断	Yes
35 ~ 40	19 ~ 24	-	-	保留	-
41	25	ACMP_INT	ACMP	-ACMP中断	Yes
42 ~ 43	26 ~ 27	-	-	保留	-
44	28	PWRWU_INT	CLKC	从掉电状态唤醒的时钟控制器中断	Yes
45	29	ADC_INT	ADC	ADC中断	No
46 ~ 47	30 ~ 31	-	-	保留	-

表 5-5 NuMicro® NUC029FAE 系统中断映射向量表

5.2.11.2 向量表

当任何中断被响应时，处理器会自动从内存的向量表中获取中断服务程序（ISR）的起始地址。对于ARMv6-M，向量表的基地址固定在0x00000000。向量表包括复位后栈指针的初始值，和所有异常处理函数的入口地址。上一节的向量号(异常号)决定了异常处理函数在向量表中的入口顺序。

向量表字偏移量	描述
0	SP_main - 主堆栈指针
Vector Number	异常入口指针，用向量号表示

表 5-6 向量表格式

5.2.11.3 操作描述

通过写相应中断使能设置寄存器或清使能寄存器位域，可以使能NVIC中断或禁用NVIC中断，这些寄存器通过写1使能和写1清除为零，读取这两个寄存器均返回当前相应中断的使能状态。当某一个中断被禁用时，中断声明将使该中断处于等待处理状态，然而，该中断不会被激活。如果某一个中断在被禁用时处于激活状态，该中断就保持在激活状态，直到通过复位或异常返回来清除。清使能位可以阻止相关中断被再次激活。

NVIC中断可以使用互补的寄存器对来挂起/解除挂起以使能/禁用这些中断，这些寄存器分别为**Set-Pending**寄存器与**Clear-Pending**寄存器，这些寄存器使用写1使能和写1清除的方式，读取这两种寄存器都返回相应中断的当前挂起状态。**Clear-Pending**寄存器不会对处于激活状态的中断的执行状态产生任何影响。

NVIC中断通过更新32位寄存器中的各个8位字段（每个寄存器支持4个中断）来分配中断的优先级。

与NVIC相关的通用寄存器都可以通过系统控制空间的内存区域访问，下一节将作出描述。

5.2.12 NUC029xAN NVIC 控制寄存器映射

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位后的值
NVIC 基地址:				
NVIC_BA = 0xE000_E100				
NVIC_ISER	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31 设置使能控制寄存器	0x0000_0000
NVIC_ICER	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31 清使能控制寄存器	0x0000_0000
NVIC_ISPR	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	0x0000_0000
NVIC_ICPR	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31 清挂起控制寄存器	0x0000_0000
NVIC_IPR0	NVIC_BA+0x300	R/W	IRQ0 ~ IRQ3 优先级 控制寄存器	0x0000_0000
NVIC_IPR1	NVIC_BA+0x304	R/W	IRQ4 ~ IRQ7优先级控制寄存器	0x0000_0000
NVIC_IPR2	NVIC_BA+0x308	R/W	IRQ8 ~ IRQ11优先级控制寄存器	0x0000_0000
NVIC_IPR3	NVIC_BA+0x30C	R/W	IRQ12 ~ IRQ15优先级控制寄存器	0x0000_0000
NVIC_IPR4	NVIC_BA+0x310	R/W	IRQ16 ~ IRQ19优先级控制寄存器	0x0000_0000
NVIC_IPR5	NVIC_BA+0x314	R/W	IRQ20 ~ IRQ23优先级控制寄存器	0x0000_0000
NVIC_IPR6	NVIC_BA+0x318	R/W	IRQ24 ~ IRQ27优先级控制寄存器	0x0000_0000
NVIC_IPR7	NVIC_BA+0x31C	R/W	IRQ28 ~ IRQ31优先级控制寄存器	0x0000_0000

IRQ0 ~ IRQ31设置使能控制寄存器 (NVIC_ISER)

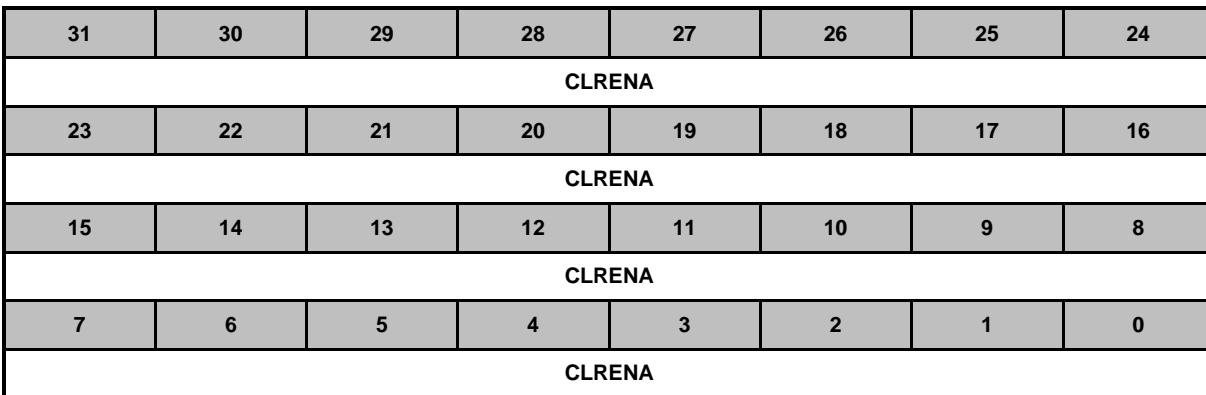
寄存器	偏移量	R/W	描述	复位的值
NVIC_ISER	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31设置使能控制寄存器	0x0000_0000

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	<p>中断使能寄存器 使能1个或多个中断，每位代表从IRQ0 ~ IRQ31的中断号(向量号：16 ~ 47). 写： 1 = 写1使能相关中断 0 = 无效 读： 1 = 相应中断是使能的 0 = 相应中断是禁止的 寄存器读取返回当前使能状态.</p>

IRQ0 ~ IRQ31清使能控制寄存器(NVIC_ICER)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ICER	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31清使能控制寄存器	0x0000_0000



位	描述	
[31:0]	CLRENA	<p>中断禁止寄存器</p> <p>禁用1个或多个中断，每位代表从IRQ0 ~ IRQ31的中断号(向量号：16 ~ 47)。</p> <p>写：</p> <p>1 = 写1 禁用相应中断 0 = 无效</p> <p>读：</p> <p>1 = 相应中断时使能的 0 = 相应中断时禁止的</p> <p>寄存器读取返回当前使能状态。</p>

IRQ0 ~ IRQ31设置挂起控制寄存器 (NVIC_ISPR)

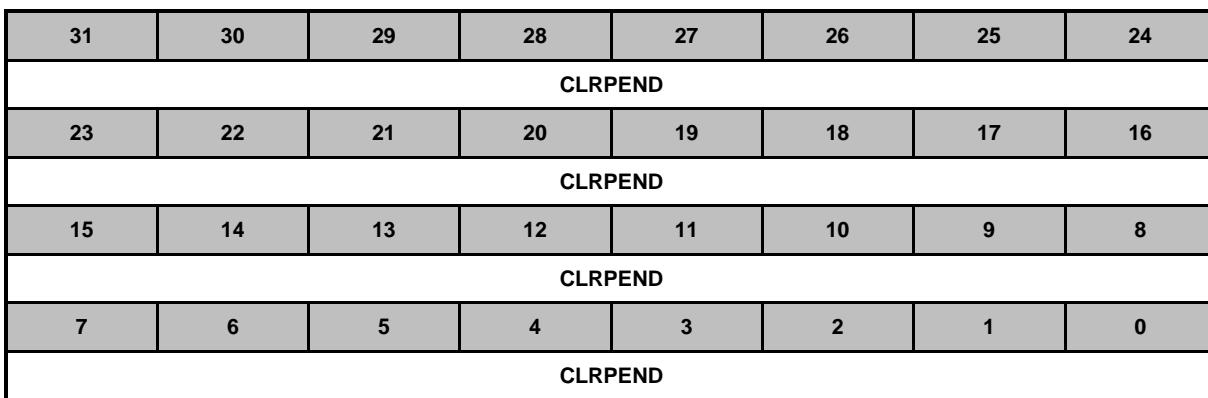
寄存器	偏移量	R/W	描述	复位后的值
NVIC_ISPR	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31设置挂起控制寄存器	0x0000_0000

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	<p>设置中断挂起寄存器</p> <p>写:</p> <p>1 = 写1, 挂起相应中断。每位代表从IRQ0 ~ IRQ31 的中断号(向量号: 16 ~ 47).</p> <p>0 = 无效</p> <p>读:</p> <p>1 = 相应中断处于挂起状态</p> <p>0 = 相应中断处于非挂起状态</p> <p>寄存器读取返回当前挂起状态</p>

IRQ0 ~ IRQ31清挂起控制寄存器 (NVIC_ICPR)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ICPR	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31清挂起控制寄存器	0x0000_0000



位	描述
[31:0]	CLRPEND 清中断挂起寄存器 写： 1 = 写1清除挂起的相应中断，每位代表从IRQ0 ~ IRQ31的中断号 (向量号： 16 ~ 47) 0 = 无效 读： 1 = 相应中断处于挂起状态 0 = 相应中断处于非挂起状态 寄存器读取返回当前挂起状态。

IRQ0 ~ IRQ3中断优先级寄存器 (NVIC_IPR0)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR0	NVIC_BA+0x300	R/W	IRQ0 ~ IRQ3中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		保留					
23	22	21	20	19	18	17	16
PRI_2		保留					
15	14	13	12	11	10	9	8
PRI_1		保留					
7	6	5	4	3	2	1	0
PRI_0		保留					

位	描述	
[31:30]	PRI_3	IRQ3优先级 “0” 表示最高优先级& “3” 表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_2	IRQ2优先级 “0” 表示最高优先级& “3” 表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_1	IRQ1优先级 “0” 表示最高优先级& “3” 表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_0	IRQ0优先级 “0” 表示最高优先级& “3” 表示最低优先级
[5:0]	保留	保留.

IRQ4 ~ IRQ7中断优先级寄存器 (NVIC_IPR1)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR1	NVIC_BA+0x304	R/W	IRQ4 ~ IRQ7中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		保留					
23	22	21	20	19	18	17	16
PRI_6		保留					
15	14	13	12	11	10	9	8
PRI_5		保留					
7	6	5	4	3	2	1	0
PRI_4		保留					

位	描述	
[31:30]	PRI_7	IRQ7优先级 “0” 表示最高优先级& “3” 表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_6	IRQ6优先级 “0” 表示最高优先级& “3” 表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_5	IRQ5优先级 “0” 表示最高优先级& “3” 表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_4	IRQ4优先级 “0” 表示最高优先级& “3” 表示最低优先级
[5:0]	保留	保留.

IRQ8 ~ IRQ11中断优先级寄存器 (NVIC_IPR2)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR2	NVIC_BA+0x308	R/W	IRQ8 ~ IRQ11中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		保留					
23	22	21	20	19	18	17	16
PRI_10		保留					
15	14	13	12	11	10	9	8
PRI_9		保留					
7	6	5	4	3	2	1	0
PRI_8		保留					

位	描述	
[31:30]	PRI_11	IRQ11优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_10	IRQ10优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_9	IRQ9优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_8	IRQ8优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ12 ~ IRQ15中断优先级寄存器 (NVIC_IPR3)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR3	NVIC_BA+0x30C	R/W	IRQ12 ~ IRQ15中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		保留					
23	22	21	20	19	18	17	16
PRI_14		保留					
15	14	13	12	11	10	9	8
PRI_13		保留					
7	6	5	4	3	2	1	0
PRI_12		保留					

位	描述	
[31:30]	PRI_15	IRQ15优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_14	IRQ14优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_13	IRQ13优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_12	IRQ12优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ16 ~ IRQ19中断优先级寄存器 (NVIC_IPR4)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR4	NVIC_BA+0x310	R/W	IRQ16 ~ IRQ19中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		保留					
23	22	21	20	19	18	17	16
PRI_18		保留					
15	14	13	12	11	10	9	8
PRI_17		保留					
7	6	5	4	3	2	1	0
PRI_16		保留					

位	描述	
[31:30]	PRI_19	IRQ19优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_18	IRQ18优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_17	IRQ17优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_16	IRQ16优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ20 ~ IRQ23中断优先级寄存器 (NVIC_IPR5)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR5	NVIC_BA+0x314	R/W	IRQ20 ~ IRQ23中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23	保留						
23	22	21	20	19	18	17	16
PRI_22	保留						
15	14	13	12	11	10	9	8
PRI_21	保留						
7	6	5	4	3	2	1	0
PRI_20	保留						

位	描述	
[31:30]	PRI_23	IRQ23优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_22	IRQ22优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_21	IRQ21优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_20	IRQ20优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ24 ~ IRQ27中断优先级寄存器 (NVIC_IPR6)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR6	NVIC_BA+0x318	R/W	IRQ24 ~ IRQ27中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		保留					
23	22	21	20	19	18	17	16
PRI_26		保留					
15	14	13	12	11	10	9	8
PRI_25		保留					
7	6	5	4	3	2	1	0
PRI_24		保留					

位	描述	
[31:30]	PRI_27	IRQ27优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_26	IRQ26优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_25	IRQ25优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_24	IRQ24优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ28 ~ IRQ31中断优先级寄存器 (NVIC_IPR7)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR7	NVIC_BA+0x31C	R/W	IRQ28 ~ IRQ31中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		保留					
23	22	21	20	19	18	17	16
PRI_30		保留					
15	14	13	12	11	10	9	8
PRI_29		保留					
7	6	5	4	3	2	1	0
PRI_28		保留					

位	描述	
[31:30]	PRI_31	IRQ31优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_30	IRQ30优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_29	IRQ29优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_28	IRQ28优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

中断源控制寄存器

除了与NVIC相关的中断控制寄存器外，NuMicro® NUC029xAN系列还有一些特殊控制寄存器以利于中断功能，包括“中断源识别”，“NMI 源选择”与“中断测试模式”。描述如下：

R: 只读, **W:** 只写, **R/W:** 可读写

寄存器	偏移量	R/W	描述	复位后的值
INT 基地址:				
INT_BA = 0x5000_0300				
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) 中断源识别	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) 中断源识别	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	IRQ2 ((EINT0) 中断源识别	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1) 中断源识别	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (P0/1) 中断源识别	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (P2/3/4) 中断源识别	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWMA) 中断源识别	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (PWMB) 中断源识别	0xXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) 中断源识别	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) 中断源识别	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (TMR2) 中断源识别	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (TMR3) 中断源识别	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0) 中断源识别	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (URT1) 中断源识别	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI0) 中断源识别	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	IRQ15 (SPI1) 中断源识别	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	-	保留	0xXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	-	保留	0xXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I ² C0) 中断源识别	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I ² C1) 中断源识别	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	-	保留	0xXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	-	保留	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	-	保留	0xXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	-	保留	0xXXXX_XXXX

IRQ24_SRC	INT_BA+0x60	-	保留	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP01) 中断源识别	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	R	IRQ26 (ACMP23) 中断源识别	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	-	保留	0xXXXX_XXXX
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) 中断源识别	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) 中断源识别	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	-	保留	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	-	保留	0xXXXX_XXXX
NMI_SEL	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	0x0000_0000

IRQ0 (BOD) 中断源识别寄存器(IRQ0_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) 中断源识别	0xFFFF_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
保留							
7	6	5	4	3	2	1	0
保留					INT_SRC		

位	描述
[2:0]	INT_SRC IRQ0 中断源识别 INT_SRC[2] = 保留. INT_SRC[1] = 保留. INT_SRC[0]: 0 = IRQ0 中断源不是来自 BOD 中断 (BOD_INT). 1 = IRQ0 中断源来自 BOD 中断 (BOD_INT). 注: 当中断标志被清除时，该位将被自动清除.

IRQ1 (WDT) 中断源识别寄存器(IRQ1_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) 中断源识别	0XXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ1中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ1 中断源不是来自看门狗中断 (WDT_INT). 1 = IRQ1中断源来自看门狗中断(WDT_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ2 (EINT0) 中断源识别寄存器(IRQ2_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ2中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ2中断源不是来自外部中断0 – P3.2 (EINT0). 1 = IRQ2中断源是来自外部中断0 – P3.2 (EINT0). 注: 当中断标志被清除时，该位将被自动清除。

IRQ3 (EINT1) 中断源识别寄存器(IRQ3_SRC)

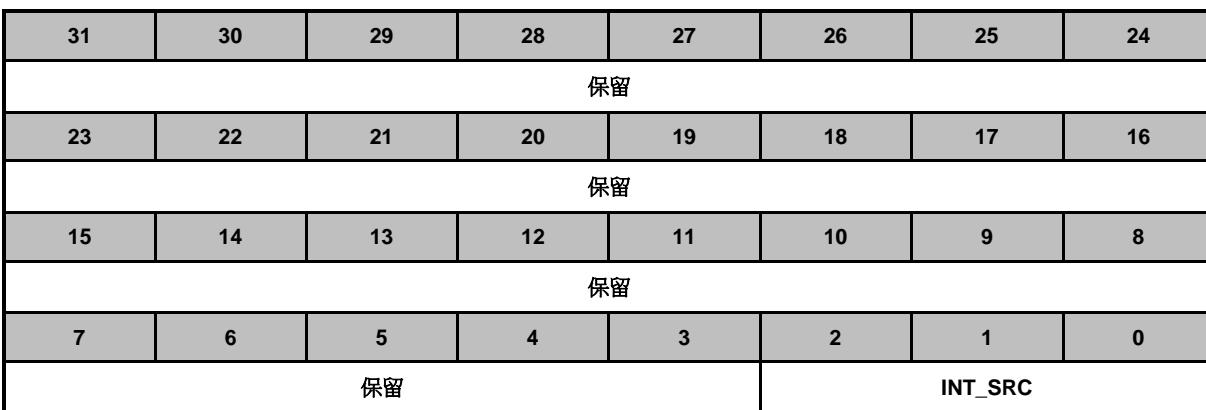
寄存器	偏移量	R/W	描述	复位后的值
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ3中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ3中断源不是来自外部中断1 – P3.3 (EINT1). 1 = IRQ3中断源是来自外部中断1 – P3.3 (EINT1). 注: 当中断标志被清除时，该位将被自动清除。

IRQ4 (P0/1) 中断源识别寄存器(IRQ4_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (P0/1) 中断源识别	0xXXXX_XXXX



位	描述	
[2:0]	INT_SRC	<p>IRQ4中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1]:</p> <p>0 = IRQ4中断源不是来自P1中断 (P1_INT).</p> <p>1 = IRQ4中断源来自P1中断 (P1_INT).</p> <p>INT_SRC[0]:</p> <p>0 = IRQ4中断源不是来自P0中断 (P0_INT).</p> <p>1 = IRQ4中断源来自P0中断(P0_INT).</p> <p>注1: 同一时刻，IRQ4 中断可以来自多个中断源.</p> <p>注2: 当中断标志被清除时，该位将被自动清除.</p>

IRQ5 (P2/3/4) 中断源识别寄存器(IRQ5_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (P2/3/4) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述
[2:0]	IRQ5中断源识别 INT_SRC[2]: 0 = IRQ5中断源不是来自P4中断(P4_INT). 1 = IRQ5中断源来自P4中断(P4_INT). INT_SRC[1]: 0 = IRQ5中断源不是来自P3中断 (P3_INT). 1 = IRQ5中断源来自P3中断(P3_INT). INT_SRC[0]: 0 = IRQ5中断源不是来自P2中断(P2_INT). 1 = IRQ5中断源来自P2中断(P2_INT). 注1: 同一时刻，IRQ5 中断可以来自多个中断源. 注2: 当中断标志被清除时，该位将被自动清除

IRQ6 (PWMA) 中断源识别寄存器(IRQ6_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWMA) 中断源识别	0xXXXX_XXXX

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				INT_SRC			

位	描述
[3:0]	INT_SRC IRQ6中断源识别 INT_SRC[3]: 0 = IRQ6中断源不是来自PWM3(PWMA 通道 3)中断(PWM3_INT). 1 = IRQ6中断源来自PWM3(PWMA 通道 3)中断(PWM3_INT). INT_SRC[2]: 0 = IRQ6中断源不是来自PWM2(PWMA 通道 2)中断(PWM2_INT). 1 = IRQ6中断源来自PWM2(PWMA 通道 2)中断(PWM2_INT). INT_SRC[1]: 0 = IRQ6中断源不是来自PWM1(PWMA 通道 1)中断(PWM1_INT). 1 = IRQ6中断源来自PWM1(PWMA 通道 1)中断(PWM1_INT). INT_SRC[0]: 0 = IRQ6中断源不是来自PWM0(PWMA 通道 0)中断(PWM0_INT). 1 = IRQ6中断源来自PWM0(PWMA 通道 0)中断(PWM0_INT). 注1: 同一时刻，IRQ6 中断可以来自多个中断源. 注2: 当中断标志被清除时，该位将被自动清除

IRQ7 (PWMB) 中断源识别寄存器(IRQ7_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (PWMB) 中断源识别	0xXXXX_XXXX

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				INT_SRC			

位	描述
[3:0]	IRQ7中断源识别 INT_SRC[3]: 0 = IRQ7中断源不是来自PWM7(PWMB 通道 3)中断 (PWM7_INT). 1 = IRQ7中断源来自PWM7(PWMB 通道 3)中断 (PWM7_INT). INT_SRC[2]: 0 = IRQ7中断源不是来自PWM6(PWMB 通道 2)中断 (PWM6_INT). 1 = IRQ7中断源来自PWM6(PWMB 通道 2)中断 (PWM6_INT). INT_SRC[1]: 0 = IRQ7中断源不是来自PWM5(PWMB 通道 1)中断 (PWM5_INT). 1 = IRQ7中断源来自PWM5(PWMB 通道 1)中断 (PWM5_INT). INT_SRC[0]: 0 = IRQ7中断源不是来自PWM4(PWMB 通道 0)中断 (PWM4_INT). 1 = IRQ7中断源来自PWM4(PWMB 通道 0)中断 (PWM4_INT). 注1: 同一时刻，IRQ7 中断可以来自多个中断源。 注2: 当中断标志被清除时，该位将被自动清除

IRQ8 (TMR0) 中断源识别寄存器(IRQ8_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ8中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ8中断源不是来自定时器0中断(TMR0_INT). 1 = IRQ8中断源来自定时器0中断(TMR0_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ9 (TMR1) 中断源识别寄存器(IRQ9_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ9中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ9中断源不是来自定时器1中断(TMR1_INT). 1 = IRQ9中断源来自定时器1中断(TMR1_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ10 (TMR2) 中断源识别寄存器(IRQ10_SRC)

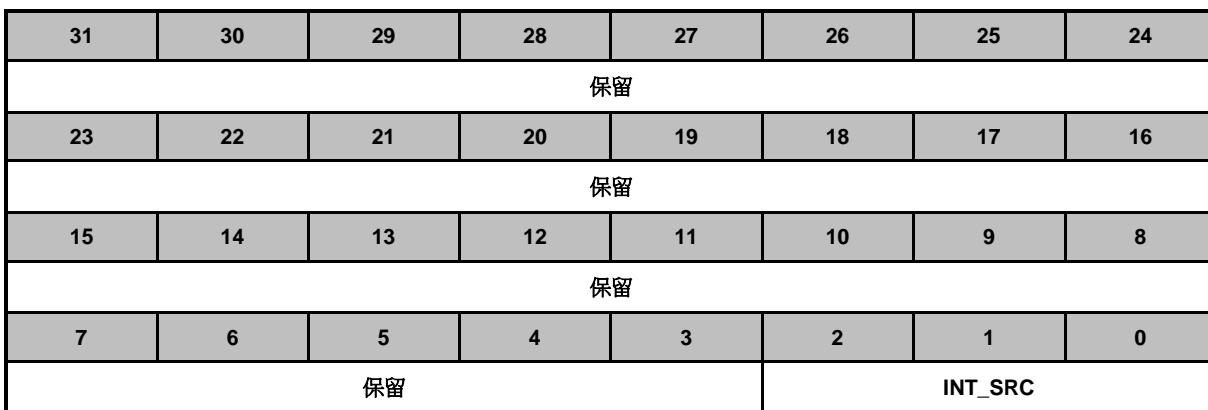
寄存器	偏移量	R/W	描述	复位后的值
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (TMR2) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ10中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ10中断源不是来自定时器2中断(TMR2_INT). 1 = IRQ10中断源来自定时器2中断(TMR2_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ11 (TMR3) 中断源识别寄存器(IRQ11_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (TMR3) 中断源识别	0xXXXX_XXXX



位	描述	
[2:0]	INT_SRC	IRQ11中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ11中断源不是来自定时器3中断(TMR3_INT). 1 = IRQ11中断源来自定时器3中断(TMR3_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ12 (UART0) 中断源识别寄存器(IRQ12_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ12中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ12中断源不是来自UART0中断(UART0_INT).</p> <p>1 = IRQ12中断源来自UART0中断(UART0_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除</p>

IRQ13 (UART1) 中断源识别寄存器(IRQ13_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (UART1) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ13中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ13中断源不是来自UART1中断(UART1_INT).</p> <p>1 = IRQ13中断源来自UART1中断(UART1_INT).</p> <p>注: 当中断标志被清除时, 该位将被自动清除.</p>

IRQ14 (SPI0) 中断源识别寄存器(IRQ14_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ14中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ14中断源不是来自SPI0中断(SPI0_INT). 1 = IRQ14中断源来自SPI0中断(SPI0_INT). 注: 当中断标志被清除时，该位将被自动清除.

IRQ15 (SPI1) 中断源识别寄存器(IRQ15_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ15_SRC	INT_BA+0x3C	R	IRQ15 (SPI1) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ15中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ15中断源不是来自SPI1中断(SPI1_INT). 1 = IRQ15中断源来自SPI1中断(SPI1_INT). 注: 当中断标志被清除时，该位将被自动清除.

IRQ18 (I²C0) 中断源识别寄存器(IRQ18_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I ² C0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ18中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ18中断源不是来自I ² C0中断(I2C0_INT). 1 = IRQ18中断源来自I ² C0中断(I2C0_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ19 (I²C1) 中断源识别寄存器(IRQ19_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I ² C1) 中断源识别	0xFFFF_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
保留							
7	6	5	4	3	2	1	0
保留					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ19中断源识别(</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ19中断源不是来自I²C1中断(I2C1_INT).</p> <p>1 = IRQ19中断源来自I²C1中断(I2C1_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除.</p>

IRQ25 (ACMP01) 中断源识别寄存器(IRQ25_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP01) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ25中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ25中断源不是来自ACMP01中断 (ACMP01_INT). 1 = IRQ25中断源来自ACMP01中断(ACMP01_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ26 (ACMP23) 中断源识别寄存器(IRQ26_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ26_SRC	INT_BA+0x68	R	IRQ26 (ACMP23) 中断源识别	0XXXX_XXXX

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					INT_SRC		

位	描述
[2:0]	INT_SRC IRQ26中断源识别(INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ26中断源不是来自ACMP23中断 (ACMP23_INT). 1 = IRQ26中断源来自ACMP23中断(ACMP23_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ28 (PWRWU) 中断源识别寄存器(IRQ28_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ28中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ28中断源不是来自睡眠唤醒中断(PWRWU_INT). 1 = IRQ28中断源来自睡眠唤醒中断(PWRWU_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ29 (ADC) 中断源识别寄存器(IRQ29_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ29中断源识别 INT_SRC[2] = 保留.. INT_SRC[1] = 保留. INT_SRC[0]: 0 = IRQ29中断源不是来自ADC中断 (ADC_INT). 1 = IRQ29中断源来自ADC中断(ADC_INT). 注: 当中断标志被清除时，该位将被自动清除

NMI中断源选择控制寄存器(NMI_SEL)

寄存器	偏移量	R/W	描述	复位后的值
NMI_SEL	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	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			NMI_SEL				

位	描述	
[31:9]	保留	保留
[8]	NMI_EN	NMI 中断使能 (写保护) 1 = 使能 NMI 中断 0 = 禁止 NMI 中断 注：这个比特是写保护的，意味着在写这个比特之前需要连续写“59h”，“16h”，“88h”到地址 0x5000_0100 来解锁。参考寄存器 REGWRPROT，地址 GCR_BA+0x100
[7:5]	保留	保留
[4:0]	NMI_SEL	NMI中断源选择 Cortex™-M0的NMI 中断源可以从32个外设中断中选择一个

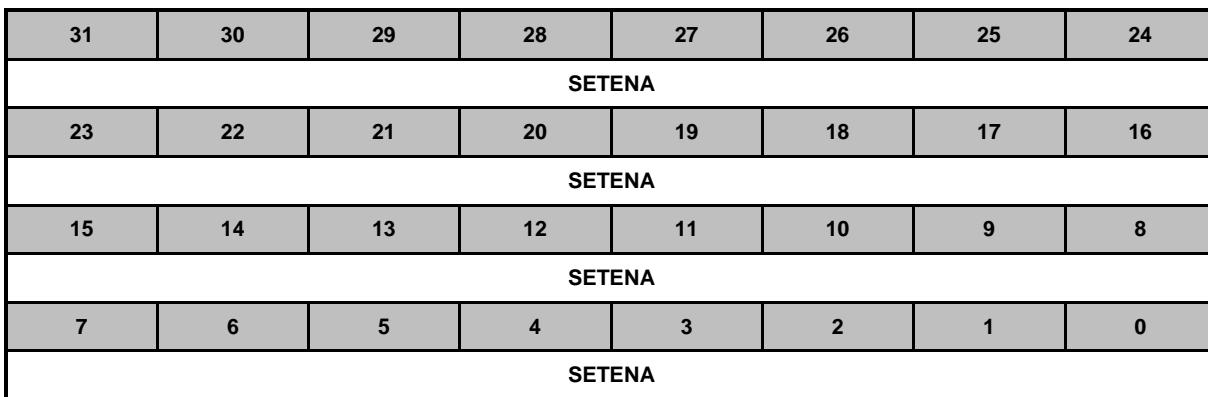
5.2.13 NUC029FAE NVIC 控制寄存器映射

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位后的值
SCS 基址:				
SCS_BA = 0xE000_E000				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 设置使能控制寄存器	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 清使能控制寄存器	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 清挂起控制寄存器	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 优先级 控制寄存器	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7优先级控制寄存器	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11优先级控制寄存器	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15优先级控制寄存器	0x0000_0000
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19优先级控制寄存器	0x0000_0000
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23优先级控制寄存器	0x0000_0000
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27优先级控制寄存器	0x0000_0000
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31优先级控制寄存器	0x0000_0000

IRQ0 ~ IRQ31设置使能控制寄存器 (NVIC_ISER)

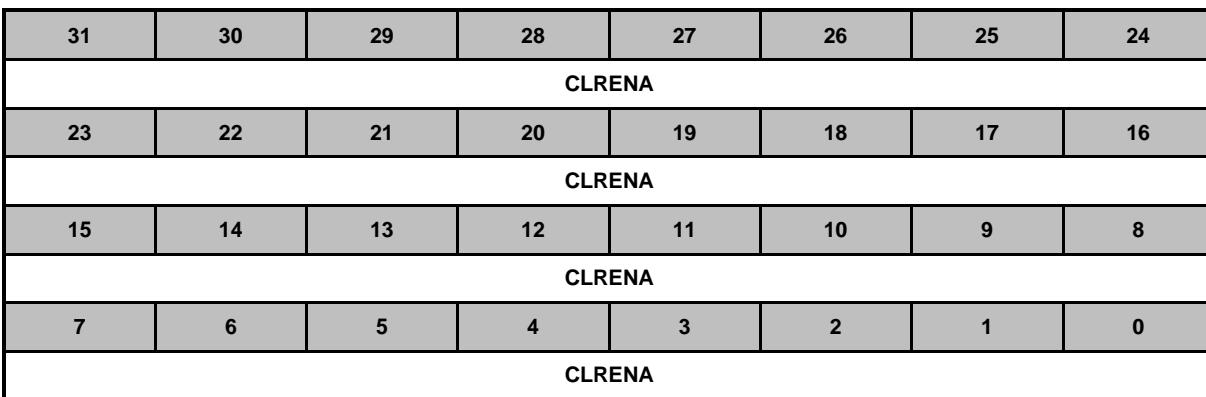
寄存器	偏移量	R/W	描述	复位的值
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31设置使能控制寄存器	0x0000_0000



位	描述	
[31:0]	SETENA	<p>中断使能寄存器 使能1个或多个中断，每位代表从IRQ0 ~ IRQ31的中断号(向量号：16 ~ 47). 写： 1 = 写1使能相关中断 0 = 无效 读： 1 = 相应中断是使能的 0 = 写相应中断是禁止的 寄存器读取返回当前使能状态.</p>

IRQ0 ~ IRQ31清使能控制寄存器(NVIC_ICER)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31清使能控制寄存器	0x0000_0000



位	描述	
[31:0]	CLRENA	<p>中断禁止寄存器 禁用1个或多个中断，每位代表从IRQ0 ~ IRQ31的中断号(向量号：16 ~ 47).</p> <p>写： 1 = 写1禁用相应中断 0 = 无效</p> <p>读： 1 = 相应中断时使能的 0 = 相应中断时禁止的 寄存器读取返回当前使能状态.</p>

IRQ0 ~ IRQ31设置挂起控制寄存器 (NVIC_ISPR)

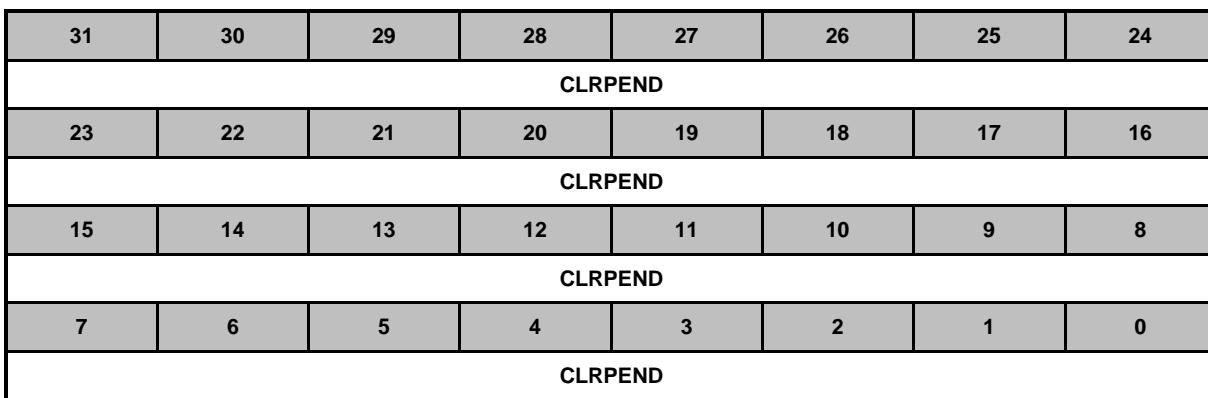
寄存器	偏移量	R/W	描述	复位后的值
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31设置挂起控制寄存器	0x0000_0000

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 设置中断挂起寄存器 写: 1 = 写1, 挂起相应中断。每位代表从IRQ0 ~ IRQ31 的中断号(向量号: 16 ~ 47). 0 = 无效 读: 1 = 相应中断处于挂起状态 0 = 相应中断处于非挂起状态 寄存器读取返回当前挂起状态

IRQ0 ~ IRQ31清挂起控制寄存器 (NVIC_ICPR)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31清挂起控制寄存器	0x0000_0000



位	描述
[31:0]	CLRPEND 清中断挂起寄存器 写： 1 = 写1清除挂起的相应中断，每位代表从IRQ0 ~ IRQ31的中断号 (向量号： 16 ~ 47) 0 = 无效 读： 1 = 相应中断处于挂起状态 0 = 相应中断处于非挂起状态 寄存器读取返回当前挂起状态。

IRQ0 ~ IRQ3中断优先级寄存器 (NVIC_IPR0)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		保留					
23	22	21	20	19	18	17	16
PRI_2		保留					
15	14	13	12	11	10	9	8
PRI_1		保留					
7	6	5	4	3	2	1	0
PRI_0		保留					

位	描述	
[31:30]	PRI_3	IRQ3优先级 “0” 表示最高优先级& “3” 表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_2	IRQ2优先级 “0” 表示最高优先级& “3” 表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_1	IRQ1优先级 “0” 表示最高优先级& “3” 表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_0	IRQ0优先级 “0” 表示最高优先级& “3” 表示最低优先级.
[5:0]	保留	保留.

IRQ4 ~ IRQ7中断优先级寄存器 (NVIC_IPR1)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		保留					
23	22	21	20	19	18	17	16
PRI_6		保留					
15	14	13	12	11	10	9	8
PRI_5		保留					
7	6	5	4	3	2	1	0
PRI_4		保留					

位	描述	
[31:30]	PRI_7	IRQ7优先级 “0” 表示最高优先级& “3” 表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_6	IRQ6优先级 “0” 表示最高优先级& “3” 表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_5	IRQ5优先级 “0” 表示最高优先级& “3” 表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_4	IRQ4优先级 “0” 表示最高优先级& “3” 表示最低优先级
[5:0]	保留	保留.

IRQ8 ~ IRQ11中断优先级寄存器 (NVIC_IPR2)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		保留					
23	22	21	20	19	18	17	16
PRI_10		保留					
15	14	13	12	11	10	9	8
PRI_9		保留					
7	6	5	4	3	2	1	0
PRI_8		保留					

位	描述	
[31:30]	PRI_11	IRQ11优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_10	IRQ10优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_9	IRQ9优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_8	IRQ8优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ12 ~ IRQ15中断优先级寄存器 (NVIC_IPR3)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		保留					
23	22	21	20	19	18	17	16
PRI_14		保留					
15	14	13	12	11	10	9	8
PRI_13		保留					
7	6	5	4	3	2	1	0
PRI_12		保留					

位	描述	
[31:30]	PRI_15	IRQ15优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_14	IRQ14优先级 “0”表示最高优先级&“3”表示最低优先级.
[21:16]	保留	保留.
[15:14]	PRI_13	IRQ13优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_12	IRQ12优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ16 ~ IRQ19中断优先级寄存器 (NVIC_IPR4)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		保留					
23	22	21	20	19	18	17	16
PRI_18		保留					
15	14	13	12	11	10	9	8
PRI_17		保留					
7	6	5	4	3	2	1	0
PRI_16		保留					

位	描述	
[31:30]	PRI_19	IRQ19优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_18	IRQ18优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_17	IRQ17优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_16	IRQ16优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ20 ~ IRQ23中断优先级寄存器 (NVIC_IPR5)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23	保留						
23	22	21	20	19	18	17	16
PRI_22	保留						
15	14	13	12	11	10	9	8
PRI_21	保留						
7	6	5	4	3	2	1	0
PRI_20	保留						

位	描述	
[31:30]	PRI_23	IRQ23优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_22	IRQ22优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_21	IRQ21优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_20	IRQ20优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

IRQ24 ~ IRQ27中断优先级寄存器 (NVIC_IPR6)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27中断优先级寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		保留					
23	22	21	20	19	18	17	16
PRI_26		保留					
15	14	13	12	11	10	9	8
PRI_25		保留					
7	6	5	4	3	2	1	0
PRI_24		保留					

位	描述	
[31:30]	PRI_27	IRQ27优先级 “0” 表示最高优先级 & “3” 表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_26	IRQ26优先级 “0” 表示最高优先级 & “3” 表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_25	IRQ25优先级 “0” 表示最高优先级 & “3” 表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_24	IRQ24优先级 “0” 表示最高优先级 & “3” 表示最低优先级
[5:0]	保留	保留.

IRQ28 ~ IRQ31中断优先级寄存器 (NVIC_IPR7)

寄存器	偏移量	R/W	描述				复位后的值
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31中断优先级寄存器				0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		保留					
23	22	21	20	19	18	17	16
PRI_30		保留					
15	14	13	12	11	10	9	8
PRI_29		保留					
7	6	5	4	3	2	1	0
PRI_28		保留					

位	描述	
[31:30]	PRI_31	IRQ31优先级 “0”表示最高优先级&“3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_30	IRQ30优先级 “0”表示最高优先级&“3”表示最低优先级
[21:16]	保留	保留.
[15:14]	PRI_29	IRQ29优先级 “0”表示最高优先级&“3”表示最低优先级
[13:8]	保留	保留.
[7:6]	PRI_28	IRQ28优先级 “0”表示最高优先级&“3”表示最低优先级
[5:0]	保留	保留.

中断源控制寄存器

除了与NVIC相关的中断控制寄存器外，NuMicro® NUC029FAE系列还有一些特殊控制寄存器以利于中断功能，包括“中断源识别”，“NMI 源选择”与“中断测试模式”。描述如下：

R: 只读, **W:** 只写, **R/W:** 可读写

寄存器	偏移量	R/W	描述	复位后的值
INT 基地址:				
INT_BA = 0x5000_0300				
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) 中断源识别	0xFFFF_FFFF
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) 中断源识别	0xFFFF_FFFF
IRQ2_SRC	INT_BA+0x08	R	IRQ2 ((EINT0) 中断源识别	0xFFFF_FFFF
IRQ3_SRC	INT_BA+0x0C	-	保留	0xFFFF_FFFF
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (P0/1) 中断源识别	0xFFFF_FFFF
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (P2/3/4) 中断源识别	0xFFFF_FFFF
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM) 中断源识别	0xFFFF_FFFF
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (BRAKE) 中断源识别	0xFFFF_FFFF
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) 中断源识别	0xFFFF_FFFF
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) 中断源识别	0xFFFF_FFFF
IRQ10_SRC	INT_BA+0x28	-	保留	0xFFFF_FFFF
IRQ11_SRC	INT_BA+0x2C	-	保留	0xFFFF_FFFF
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART) 中断源识别	0xFFFF_FFFF
IRQ13_SRC	INT_BA+0x34	-	保留	0xFFFF_FFFF
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI) 中断源识别	0xFFFF_FFFF
IRQ15_SRC	INT_BA+0x3C	-	保留	0xFFFF_FFFF
IRQ16_SRC	INT_BA+0x40	R	IRQ16 (P5) 中断源识别	0xFFFF_FFFF
IRQ17_SRC	INT_BA+0x44	R	IRQ17 (HIRC校准) 中断源识别	0xFFFF_FFFF
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I ² C) 中断源识别	0xFFFF_FFFF
IRQ19_SRC	INT_BA+0x4C	-	保留	0xFFFF_FFFF
IRQ20_SRC	INT_BA+0x50	-	保留	0xFFFF_FFFF
IRQ21_SRC	INT_BA+0x54	-	保留	0xFFFF_FFFF
IRQ22_SRC	INT_BA+0x58	-	保留	0xFFFF_FFFF

IRQ23_SRC	INT_BA+0x5C	-	保留	0xXXXX_XXXX
IRQ24_SRC	INT_BA+0x60	-	保留	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP) 中断源识别	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	-	保留	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	-	保留	0xXXXX_XXXX
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) 中断源识别	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) 中断源识别	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	-	保留	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	-	保留	0xXXXX_XXXX
NMI_CON	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	0x0000_0000
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ数字中断识别寄存器	0x0000_0000

IRQ0 (BOD) 中断源识别寄存器(IRQ0_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) 中断源识别	0XXXX_XXXX

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					INT_SRC		

位	描述	
[31:3]	保留	保留.
[2:0]	INT_SRC	<p>IRQ0 中断源识别</p> <p>INT_SRC[2] = 保留.</p> <p>INT_SRC[1] = 保留.</p> <p>INT_SRC[0]:</p> <p>0 = IRQ0 中断源不是来自 BOD 中断 (BOD_INT).</p> <p>1 = IRQ0 中断源来自 BOD 中断 (BOD_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除.</p>

IRQ1 (WDT) 中断源识别寄存器(IRQ1_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) 中断源识别	0XXXX_XXXX

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					INT_SRC		

位	描述	
[31:3]	保留	保留.
[2:0]	INT_SRC	<p>IRQ1中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ1 中断源不是来自看门狗中断 (WDT_INT).</p> <p>1 = IRQ1中断源来自看门狗中断(WDT_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除</p>

IRQ2 (EINT0) 中断源识别寄存器(IRQ2_SRC)

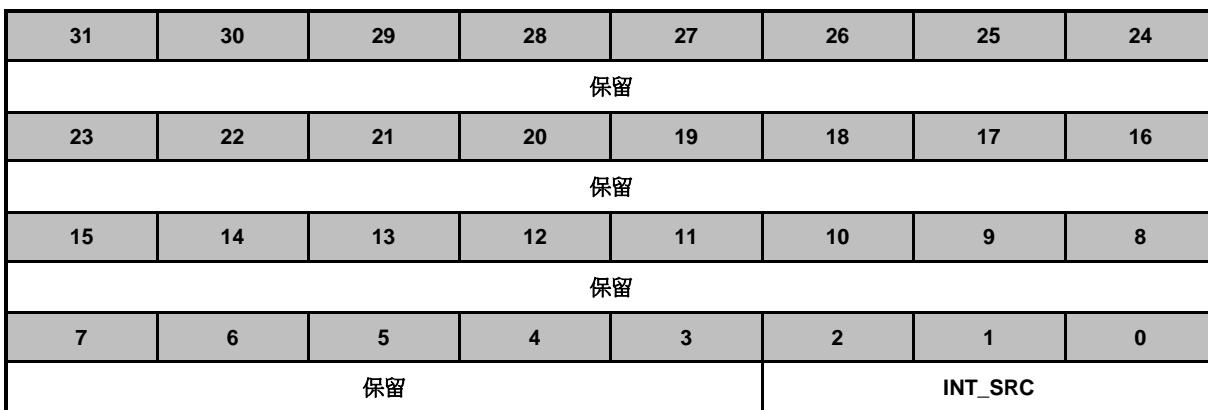
寄存器	偏移量	R/W	描述	复位后的值
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ2中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ2中断源不是来自外部中断0 – P3.2 (EINT0). 1 = IRQ2中断源是来自外部中断0 – P3.2 (EINT0). 注: 当中断标志被清除时，该位将被自动清除。

IRQ4 (P0/1) 中断源识别寄存器(IRQ4_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (P0/1) 中断源识别	0xXXXX_XXXX



位	描述	
[2:0]	INT_SRC	<p>IRQ4中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1]:</p> <p>0 = IRQ4中断源不是来自P1中断 (P1_INT).</p> <p>1 = IRQ4中断源来自P1中断 (P1_INT).</p> <p>INT_SRC[0]:</p> <p>0 = IRQ4中断源不是来自P0中断 (P0_INT).</p> <p>1 = IRQ4中断源来自P0中断(P0_INT).</p> <p>注1: 同一时刻，IRQ4 中断可以来自多个中断源.</p> <p>注2: 当中断标志被清除时，该位将被自动清除.</p>

IRQ5 (P2/3/4) 中断源识别寄存器(IRQ5_SRC)

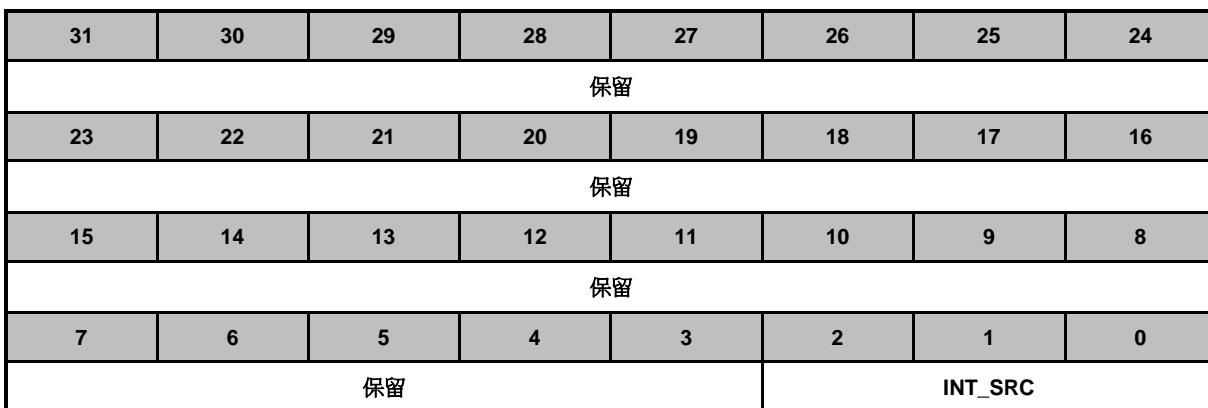
寄存器	偏移量	R/W	描述	复位后的值
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (P2/3/4) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述
[2:0]	INT_SRC IRQ5中断源识别 INT_SRC[2]: 0 = IRQ5中断源不是来自P4中断(P4_INT). 1 = IRQ5中断源来自P4中断(P4_INT). INT_SRC[1]: 0 = IRQ5中断源不是来自P3中断 (P3_INT). 1 = IRQ5中断源来自P3中断(P3_INT). INT_SRC[0]: 0 = IRQ5中断源不是来自P2中断(P2_INT). 1 = IRQ5中断源来自P2中断(P2_INT). 注1: 同一时刻，IRQ5 中断可以来自多个中断源. 注2: 当中断标志被清除时，该位将被自动清除

IRQ6 (PWM) 中断源识别寄存器(IRQ6_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM) 中断源识别	0xXXXX_XXXX



位	描述	
[3:0]	INT_SRC	IRQ6中断源识别 INT_SRC[2] = 保留； INT_SRC[1] = 保留； INT_SRC[0]: 0 = IRQ6中断源不是来自PWM中断(PWM_INT). 1 = IRQ6中断源来自PWM中断(PWM_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ7 (BRAKE) 中断源识别寄存器(IRQ7_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (BRAKE) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[3:0]	INT_SRC	IRQ7中断源识别 INT_SRC[2] = 保留； INT_SRC[1] = 保留； INT_SRC[0]: 0 = IRQ7中断源不是来自BRAKE中断(BRAKE_INT). 1 = IRQ7中断源来自BRAKE中断(BRAKE_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ8 (TMR0) 中断源识别寄存器(IRQ8_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ8中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ8中断源不是来自定时器0中断(TMR0_INT). 1 = IRQ8中断源来自定时器0中断(TMR0_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ9 (TMR1) 中断源识别寄存器(IRQ9_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ9中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ9中断源不是来自定时器1中断(TMR1_INT). 1 = IRQ9中断源来自定时器1中断(TMR1_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ12 (UART) 中断源识别寄存器(IRQ12_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ12中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ12中断源不是来自UART中断(UART_INT). 1 = IRQ12中断源来自UART中断(UART_INT). 注: 当中断标志被清除时，该位将被自动清除

IRQ14 (SPI) 中断源识别寄存器(IRQ14_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ14中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ14中断源不是来自SPI中断(SPI_INT). 1 = IRQ14中断源来自SPI中断(SPI_INT). 注: 当中断标志被清除时，该位将被自动清除.

IRQ16 (P5) 中断源识别寄存器(IRQ16_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ16_SRC	INT_BA+0x40	R	IRQ16 (P5) 中断源识别	0XXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ16中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ16中断源不是来自P5中断(P5_INT).</p> <p>1 = IRQ15中断源来自P5中断(P5_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除.</p>

IRQ17 (HIRC 校准) 中断源识别寄存器(IRQ17_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ17_SRC	INT_BA+0x44	R	IRQ17 (HIRC 校准) 中断源识别	0xFFFF_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
保留							
7	6	5	4	3	2	1	0
保留					INT_SRC		

位	描述
[2:0]	INT_SRC IRQ17中断源识别 INT_SRC[2] = 保留. INT_SRC[1] = 保留. INT_SRC[0]: 0 = IRQ17中断源不是来自HIRC校准中断(HIRC_TRIM_INT). 1 = IRQ17中断源是来自HIRC校准中断(HIRC_TRIM_INT).. 注: 当中断标志被清除时，该位将被自动清除.

IRQ18 (I²C) 中断源识别寄存器(IRQ18_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I ² C) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ18中断源识别 INT_SRC[2] =保留.. INT_SRC[1] =保留.. INT_SRC[0]: 0 = IRQ18中断源不是来自I ² C中断(I2C_INT). 1 = IRQ18中断源来自I ² C中断(I2C_INT). 注: 当中断标志被清除时，该位将被自动清除。

IRQ25 (ACMP) 中断源识别寄存器(IRQ25_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP) 中断源识别	0xFFFF_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
保留							
7	6	5	4	3	2	1	0
保留					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ25中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ25中断源不是来自ACMP中断 (ACMP_INT).</p> <p>1 = IRQ25中断源来自ACMP中断(ACMP_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除.</p>

IRQ28 (PWRWU) 中断源识别寄存器(IRQ28_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) 中断源识别	0xFFFF_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
保留							
7	6	5	4	3	2	1	0
保留					INT_SRC		

位	描述	
[2:0]	INT_SRC	<p>IRQ28中断源识别</p> <p>INT_SRC[2] =保留..</p> <p>INT_SRC[1] =保留..</p> <p>INT_SRC[0]:</p> <p>0 = IRQ28中断源不是来自睡眠唤醒中断(PWRWU_INT).</p> <p>1 = IRQ28中断源来自睡眠唤醒中断(PWRWU_INT).</p> <p>注: 当中断标志被清除时，该位将被自动清除</p>

IRQ29 (ADC) 中断源识别寄存器(IRQ29_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) 中断源识别	0xXXXX_XXXX

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					INT_SRC		

位	描述	
[2:0]	INT_SRC	IRQ29中断源识别 INT_SRC[2] = 保留.. INT_SRC[1] = 保留. INT_SRC[0]: 0 = IRQ29中断源不是来自ADC中断 (ADC_INT). 1 = IRQ29中断源来自ADC中断(ADC_INT). 注: 当中断标志被清除时，该位将被自动清除

NMI中断源选择控制寄存器(NMI_CON)

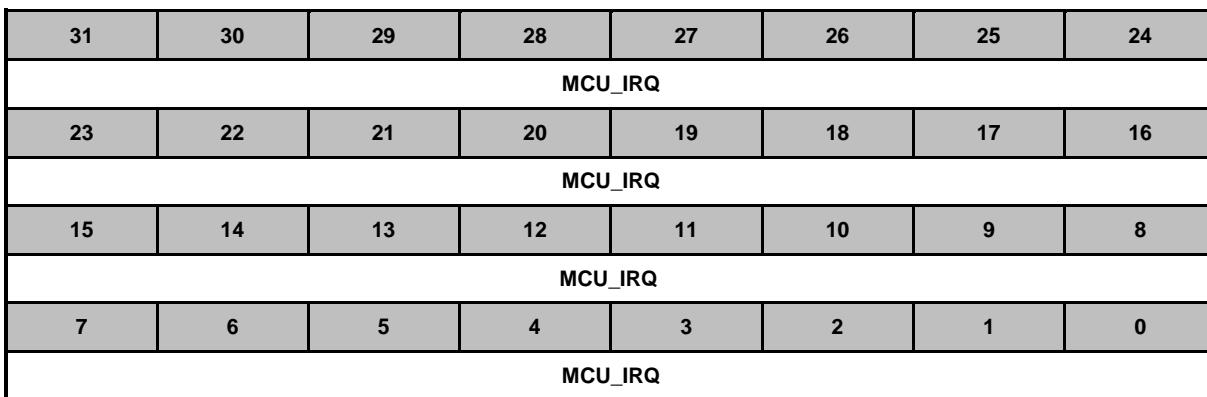
寄存器	偏移量	R/W	描述	复位后的值
NMI_CON	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	0x0000_0000

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							NMI_SEL_EN
7	6	5	4	3	2	1	0
保留			NMI_SEL				

位	描述	
[31:9]	保留	保留
[8]	NMI_SEL_EN	NMI 中断使能 (写保护) 1 = 使能 NMI 中断 0 = 禁止 NMI 中断 注：这个比特是写保护的，意味着在写这个比特之前需要连续写“59h”，“16h”，“88h”到地址 0x5000_0100 来解锁。参考寄存器 REGWRPROT，地址 GCR_BA+0x100
[7:5]	保留	保留
[4:0]	NMI_SEL	NMI中断源选择 通过设置 NMI_SEL ，Cortex™-M0的NMI 中断源可以从32个外设中断中选择一个

MCU 中断源控制寄存器(MCU_IRQ)

寄存器	偏移量	R/W	描述	复位后的值
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ 数字中断识别寄存器	0x0000_0000



位	描述
[31:0]	<p>MCU_IRQ[31:0]</p> <p>MCU IRQ 中断源</p> <p>MCU_IRQ收集来自外设的所有中断，然后形成到Cortex®-M0内核的同步中断。有正常和测试两种模式形成到Cortex®-M0的中断。</p> <p>MCU_IRQ收集来自每个外设的所有中断、并使他们同步，然后中断Cortex®-M0。</p> <p>当MCU_IRQ[n]为0时，设置MCU_IRQ[n]为1将生成一个到Cortex®-M0 NVIC[n]的中断。</p> <p>当MCU_IRQ[n]为1（意味着中断被声明）时，设置MCU_bit[n]为1将清除中断，设置MCU_IRQ[n] 为0无影响。</p>

5.2.14 系统控制块(SCB)

Cortex[®]-M0的状态和操作模式由系统控制寄存器控制，包括CPUID，Cortex[®]-M0中断优先级和Cortex[®]-M0电源管理都可以通过这些系统控制寄存器控制。

更多详情请参考“ARM[®] Cortex[®]-M0 Technical Reference Manual”与“ARM[®] v6-M Architecture Reference Manual”。

5.2.15 NUC029xAN系统控制块寄存器映射

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位后的值
SCB 基地址:				
SCB_BA = 0xE000_ED00				
CPUID	SCB_BA+0x000	R	CPUID 寄存器	0x410CC200
ICSR	SCB_BA+0x004	R/W	中断控制状态寄存器	0x0000_0000
AIRCR	SCB_BA+0x00C	R/W	应用中断和复位控制寄存器	0xFA05_0000
SCR	SCB_BA+0x010	R/W	系统控制寄存器	0x0000_0000
SHPR2	SCB_BA+0x01C	R/W	系统处理器优先级寄存器2	0x0000_0000
SHPR3	SCB_BA+0x020	R/W	系统处理器优先级寄存器3	0x0000_0000

CPUID 寄存器 (CPUID)

寄存器	偏移量	R/W	描述	复位后的值
CPUID	SCB_BA+0x000	R	CPUID 寄存器	0x410CC200

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

位	描述	
[31:24]	IMPLEMENTER	实施代码 由ARM分配执行码. (ARM = 0x41)
[23:20]	保留	保留
[19:16]	PART	处理器的架构 ARMv6-M 读取值为0xC
[15:4]	PARTNO	处理器产品编号 读取值为 0xC20.
[3:0]	REVISION	修订号 读取值为 0x0

中断控制状态寄存器(ICSR)

寄存器	偏移量	R/W	描述				复位后的值
ICSR	SCB_BA+0x004	R/W	中断控制状态寄存器				0x0000_0000

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

位	描述	
[31]	NMIPENDSET	<p>NMI设置挂起位</p> <p>写:</p> <p>0 = 无作用 1 = 改变NMI异常状态为未处理.</p> <p>读:</p> <p>0 = 没有NMI 异常等待处理 1 = 有NMI异常等待处理</p> <p>注: 设置该位将激活NMI, 由于NMI是最高优先级的异常, 正常情况下, 只要处理器检测到写'1'到这个位, 将马上进入中断处理函数, 然后这个位就会被自动清成'0'. 这就意味着如果在中断处理函数中读这个位返回'1', 就是NMI中断再次被置。</p>
[30:29]	保留	保留
[28]	PENDSVSET	<p>PendSV设置挂起位</p> <p>写:</p> <p>0 = 无作用 1 = 改变PendSV异常状态为未处理.</p> <p>读:</p> <p>0 = 没有PendSV异常等待处理 1 = 有PendSV异常等待处理</p> <p>注: 触发PendSV 中断。写"1"到这个位是唯一触发PendSV的方式。</p>
[27]	PENDSVCLR	<p>PendSV 清挂起位</p> <p>写:</p> <p>0 = 无作用 1 = 清除PendSV 中断.</p>

		注: 这个位是只写的, 当要清除PendSV中断的时候, 必须同时对PENDSVSET写"0", PENDSVCLR写"1"
[26]	PENDSTSET	SysTick异常设置挂起位 写: 1 = 触发SysTick中断. 0 = 无作用 读: 1 = 有SysTick中断等待处理 0 = 没有SysTick中断等待处理。
[25]	PENDSTCLR	SysTick异常清挂起位 写: 1 = 清除等待处理的SysTick.中断 0 = 无作用 注: 该位是只读位, 如果想要清除PENDST位, 则需同时将PENDSTSET置0并将PENDSTCLR置1.
[24]	保留	保留.
[23]	ISRPREEMPT	中断抢先占有位 如果置位, 当从调试状态退出时, 未处理的异常将被处理. 该位为只读位
[22]	ISRPENDING	中断挂起标志(不包括 NMI和Faults) 0 = 没有中断等待处理 1 = 有中断等待处理
[21:18]	保留	保留.
[17:12]	VECTPENDING	挂起异常中优先级最高的异常号 0 = 没有中断等待处理 非0 = 等待处理的最高优先级异常号
[11:6]	保留	保留.
[5:0]	VECTACTIVE	包含当前执行异常号 0 = 线程模式 非0: 当前正在处理的异常号

应用中断和复位控制寄存器(AIRCR)

寄存器	偏移量	R/W	描述	复位后的值
AIRCR	SCB_BA+0x00C	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
保留							
7	6	5	4	3	2	1	0
保留					SYSRESETREQ	VECTCLRACTIVE	保留

位	描述		
[31:16]	VECTORKEY	寄存器访问密钥 写该寄存器时，该域应该是0x05FA，否则写动作将被忽略。VECTORKEY域用来防止复位或者异常状态清除时意外写该寄存器 读出值为0xFA05.	
[15:3]	保留	保留	
[2]	SYSRESETREQ	系统复位请求 该位写1，产生复位信号给芯片表示有复位请求。 该位是只写的，在复位时自动清零。	
[1]	VECTCLRACTIVE	异常有效状态清除位 预留用于调试。写该寄存器时，用户必须将该位写0，否则行为无法预期	
[0]	保留	保留	

系统控制寄存器(SCR)

寄存器	偏移量	R/W	描述	复位后的值
SCR	SCB_BA+0x010	R/W	系统控制寄存器	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			SEVONPEN D	保留	SLEEPDEEP	SLEEPONEX IT	保留

位	描述	
[31:5]	保留	保留.
[4]	SEVONPEND	<p>挂起时发送事件</p> <p>0 = 只有使能的中断或者事件可以唤醒处理器，禁止的中断不能唤醒 1 = 使能的事件和所有中断，包括禁止的中断，可以唤醒处理器</p> <p>当事件或者中断进入未处理状态时，将把CPU从WFE指令唤醒。如果处理器没有正在等待一个事件，这个事件将被注册，在下一个WFE指令时起作用。 执行SEV指令或者外部事件也可以唤醒处理器</p>
[3]	保留	预留
[2]	SLEEPDEEP	<p>处理器深度睡眠和休眠模式选择</p> <p>0 = 睡眠 1 = 深度睡眠。</p>
[1]	SLEEPONEXIT	<p>Sleep-On-Exit使能</p> <p>当从Handler模式返回Thread模式时，是否进入睡眠模式</p> <p>0: 返回Thread模式时，不要进入睡眠 1: 当从中断处理函数返回Thread模式时进入睡眠或者深度睡眠模式</p> <p>设置这个比特为“1”，可以使能一个中断驱动的应用，避免从中断返回时进入一个空的主函数</p>
[0]	保留	预留

系统处理器优先级寄存器 2 (SHPR2)

寄存器	偏移量	R/W	描述				复位后的值
SHPR2	SCB_BA+0x01C	R/W	系统处理器优先级寄存器2				0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		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							

位	描述	
[31:30]	PRI_11	系统处理器11的优先级 – SVCall “0” 表示最高优先级& “3” 表示最低优先级
[29:0]	保留	保留.

系统处理器优先级寄存器3 (SHPR3)

寄存器	偏移量	R/W	描述	复位后的值
SHPR3	SCB_BA+0x020	R/W	系统处理器优先级寄存器3	0x0000_0000

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

位	描述	
[31:30]	PRI_15	系统处理器15的优先级– SysTick “0”表示最高优先级 & “3”表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_14	系统处理器14的优先级 – PendSV “0”表示最高优先级 & “3”表示最低优先级
[21:0]	保留	保留.

5.2.16 NUC029FAE系统控制块寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移量	R/W	描述	复位后的值
SCS 基地址:				
SCS_BA = 0xE000_E000				
CPUID	SCS_BA+0xD00	R	CPUID寄存器	0x410CC200
ICSR	SCS_BA+0xD04	R/W	中断控制状态寄存器	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
SHPR2	SCS_BA+0xD1C	R/W	系统处理器优先级寄存器2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	系统处理器优先级寄存器3	0x0000_0000

CPUID寄存器(CPUID)

寄存器	偏移量	R/W	描述	复位后的值
CPUID	SCS_BA+0xD00	R	CPUID 寄存器	0x410CC200

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

位	描述	
[31:24]	IMPLEMENTER	实施代码 实施代码由 ARM® 分配(ARM® = 0x41)
[23:20]	保留	保留
[19:16]	PART	处理器的架构 ARMv6-M读出来的值为0xC
[15:4]	PARTNO	处理器产品编号 读出来的值为0xC20
[3:0]	REVISION	修订号 读出来的值为0x0

中断控制状态寄存器(ICSР)

寄存器	偏移量	R/W	描述				复位后的值
ICSР	SCS_BA+0xD04	R/W	中断控制状态寄存器				0x0000_0000

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

位	描述	
[31]	NMIPENDSET	<p>NMI设置挂起位</p> <p>写操作:</p> <p>0 = 无效 1 = 将NMI异常挂起</p> <p>读操作:</p> <p>0 = NMI异常没有挂起 1 = NMI异常挂起</p> <p>注: 由于NMI是最高优先级异常, 所以通常处理器一检测到该位写1就会进入NMI异常处理。进入异常处理后, 处理器会将该位清零。这意味着只有当处理器正在执行NMI异常处理程序时再次产生NMI信号, NMI异常处理程序读取这一位才返回1。</p>
[30:29]	保留	保留
[28]	PENDSVSET	<p>PendSV设置挂起位</p> <p>写操作:</p> <p>0 = 无效 1 = 将PendSV异常挂起</p> <p>读操作:</p> <p>0 = PendSV异常没有挂起 1 = PendSV异常挂起</p> <p>注: 设置该位为1是设置PendSV 异常挂起的唯一方法。</p>
[27]	PENDSVCLR	<p>PendSV 清挂起位</p> <p>写操作:</p> <p>0 = 无效 1 = 清除PendSV异常挂起状态</p> <p>注: 该位为只写位。为了清除PENDSV位, 你必须同时往PENDSVSET写0, 往</p>

		PENDSVCLR写1。
[26]	PENDSTSET	<p>SysTick异常设置挂起位</p> <p>写操作: 0 = 无效 1 = 将SysTick异常挂起</p> <p>读操作: 0 = SysTick异常没有挂起 1 = SysTick异常挂起</p>
[25]	PENDSTCLR	<p>SysTick异常清挂起位</p> <p>写操作: 0 = 无效 1 = 清除SysTick异常挂起状态</p> <p>注: 该位为只读位。当你要清除PENDST位时, 必须同时往PENDSTSET写0, 往PENDSTCLR写1。</p>
[24]	保留	保留
[23]	ISRPREEMPT	<p>中断抢先占有位</p> <p>如果该位设置, 一个挂起的异常将会从调试停止状态退出并进入服务。</p> <p>该位只读</p>
[22]	ISR PENDING	<p>中断挂起标志(不包括 NMI和Faults)</p> <p>0 = 中断没有挂起 1 = 中断挂起 该位只读</p>
[21]	保留	保留
[20:12]	VECTPENDING	<p>挂起异常中优先级最高的异常号</p> <p>0 = 没有异常挂起 非0 = 挂起异常中优先级最高的异常号 这些位只读</p>
[11:6]	保留	保留
[5:0]	VECTACTIVE	<p>包含当前执行异常号</p> <p>0 = Thread模式 非0 = 当前执行异常的异常号 这些位为只读位</p>

应用中断和复位控制寄存器((AIRCR))

寄存器	偏移量	R/W	描述	复位后的值
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
保留							
7	6	5	4	3	2	1	0
保留					SYSRESETREQ	VECTCLRACTIVE	保留

位	描述	
[31:16]	VECTORKEY	<p>寄存器访问密钥</p> <p>写操作:</p> <p>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。</p> <p>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。</p> <p>读操作:</p> <p>读出来的值为0xFA05.</p>
[15:3]	保留	保留
[2]	SYSRESETREQ	<p>系统复位请求</p> <p>往该位写1将引起一个复位信号给芯片，表明有复位请求。</p> <p>该位为只写位，复位后自动清零。</p>
[1]	VECTCLRACTIVE	<p>异常有效状态清除位</p> <p>保留给调试使用。当写这个寄存器时，用户必须往该位写0，否则将出现不可预知的情况。</p>
[0]	保留	保留

系统控制寄存器(SCR)

寄存器	偏移量	R/W	描述	复位后的值
SCR	SCS_BA+0xD10	R/W	系统控制寄存器	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			SEVONPEN D	保留	SLEEPDEEP	SLEEPONE XIT	保留

位	描述		
[31:5]	保留	保留.	
[4]	SEVONPEND	挂起状态时发送事件位 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]	保留	保留	
[2]	SLEEPDEEP	处理器深度睡眠和休眠模式选择 控制处理器在低电模式时使用休眠还是深度休眠模式。 0 = 休眠模式 1 = 深度休眠模式	
[1]	SLEEPONEXIT	Sleep-On-Exit使能 该位表明当从Handler模式返回到Thread模式时，是否退出休眠 0 = 当从Thread模式返回时，不休眠 1 = 从ISR返回到Thread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]	保留	保留	

系统处理器优先级寄存器 2 (SHPR2)

寄存器	偏移量	R/W	描述				复位后的值
SHPR2	SCS_BA+0xD1C	R/W	系统处理器优先级寄存器2				0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		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							

位	描述	
[31:30]	PRI_11	系统处理器11优先级 – SVCall 0表示最高优先级，3表示最低优先级
[29:0]	保留	保留.

系统处理器优先级寄存器3 (SHPR3)

寄存器	偏移量	R/W	描述	复位后的值
SHPR3	SCS_BA+0xD20	R/W	系统处理器优先级寄存器3	0x0000_0000

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

位	描述	
[31:30]	PRI_15	系统处理器15优先级 – SysTick 0表示最高优先级，3表示最低优先级
[29:24]	保留	保留.
[23:22]	PRI_14	系统处理器14优先级 – PendSV 0表示最高优先级，3表示最低优先级
[21:0]	保留	保留.

5.3 NuMicro® NUC029xAN时钟控制器

5.3.1 概述

时钟控制器为整个芯片提供时钟，包括系统时钟和所有外设时钟。时钟控制器还通过独立的时钟ON/OFF控制、时钟源选择和时钟分频器，实现电源控制功能。在将PWR_DOWN_EN(PWRCON[7])和PD_WAIT_CPU(PWRCON[8])置位且Cortex®-M0执行WFI指令之后，芯片会进入掉电模式，在那之后，芯片等待被唤醒中断源触发以离开掉电模式。在掉电模式下，时钟控制器关闭外部4~24 MHz高速晶振(HXT)和内部22.1184MHz高速振荡器(HIRC)，以降低整个系统的功耗。下图为时钟发生器和时钟源控制概图。

时钟发生器由以下4个时钟源组成：

- 一个外部 4~24 MHz 高速晶振(HXT)
- 一个可编程的 PLL FOUT(PLL时钟源可以选择外部4~24MHz高速晶振(HXT)或者内部22.1184MHz(HIRC)高速振荡器)
- 一个内部 22.1184 MHz RC高速振荡器(HIRC)
- 一个内部 10KHz 低速振荡器(LIRC)

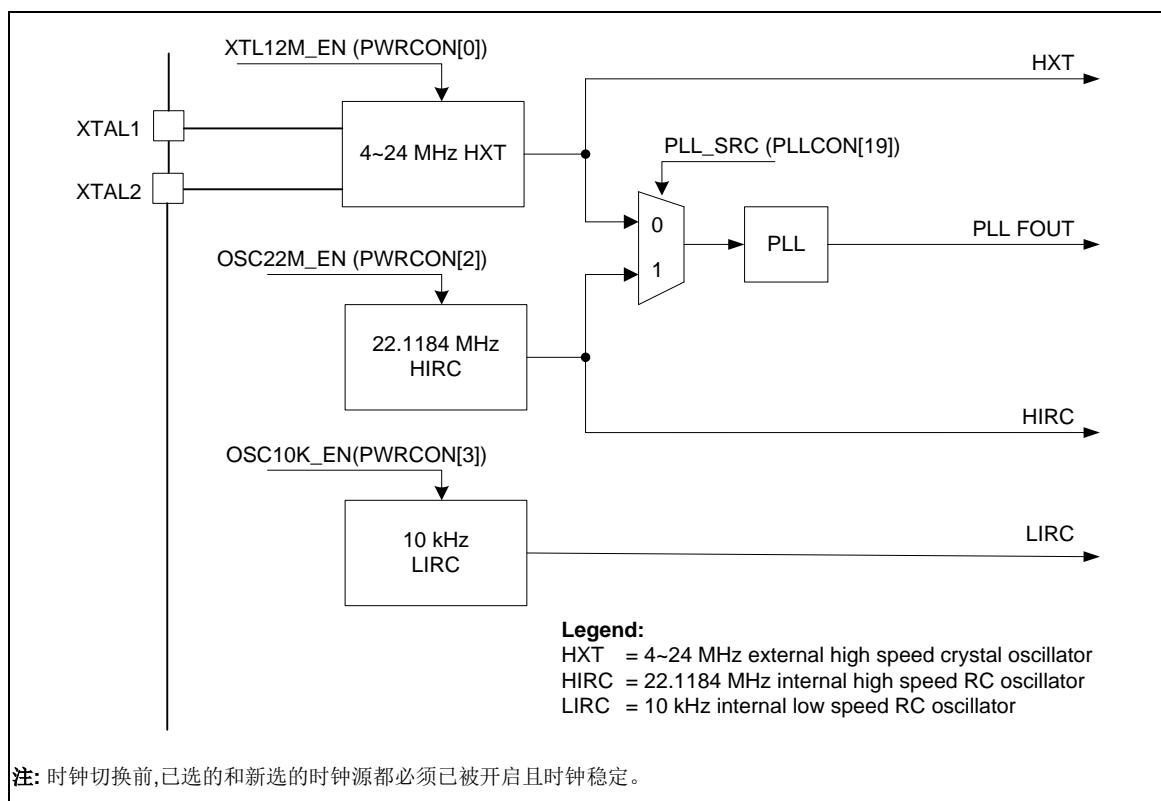


图 5-6 NuMicro® NUC029xAN 时钟发生器框图

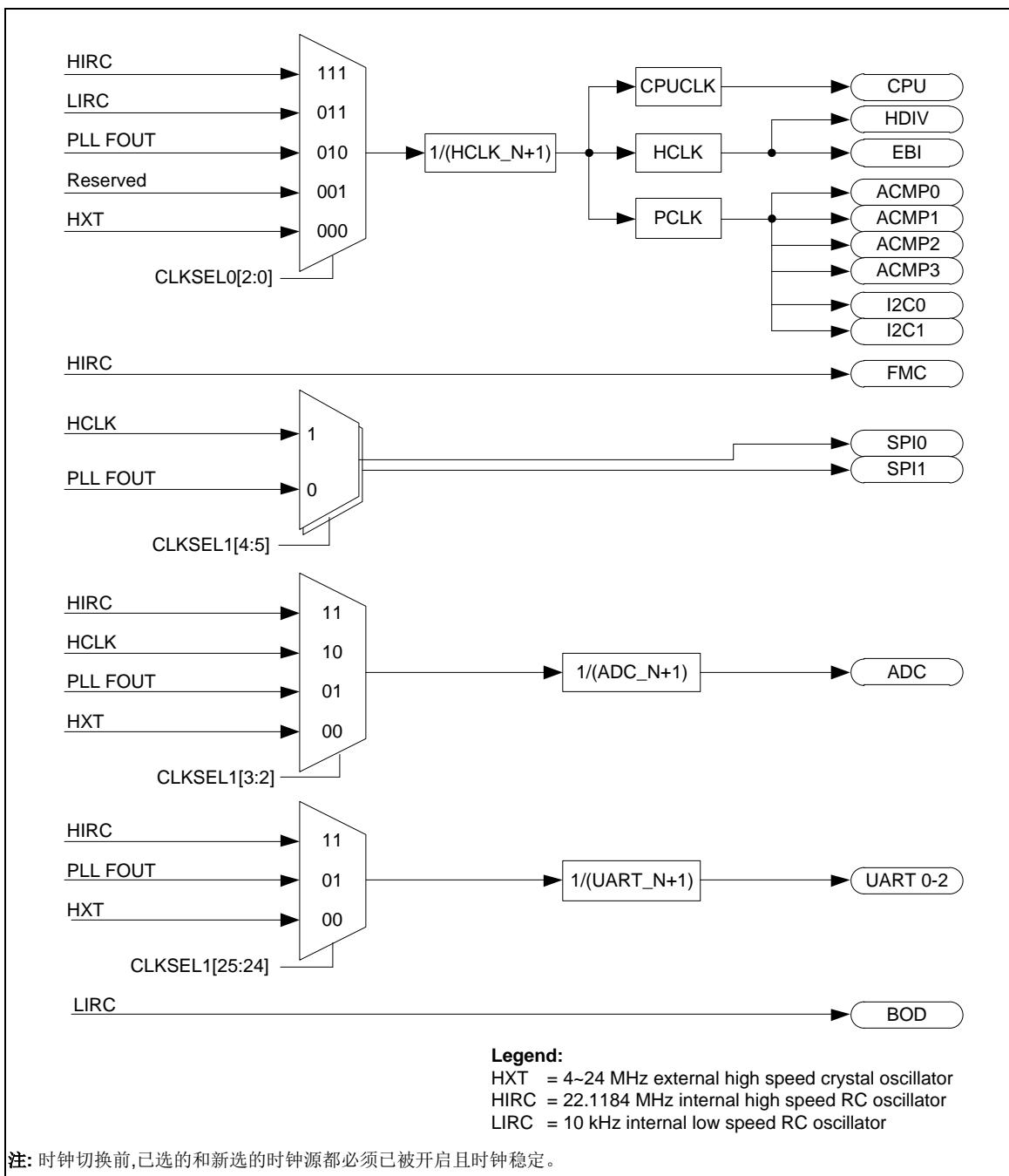


图 5-7 NuMicro® NUC029xAN 时钟源控制器概图(1/2)

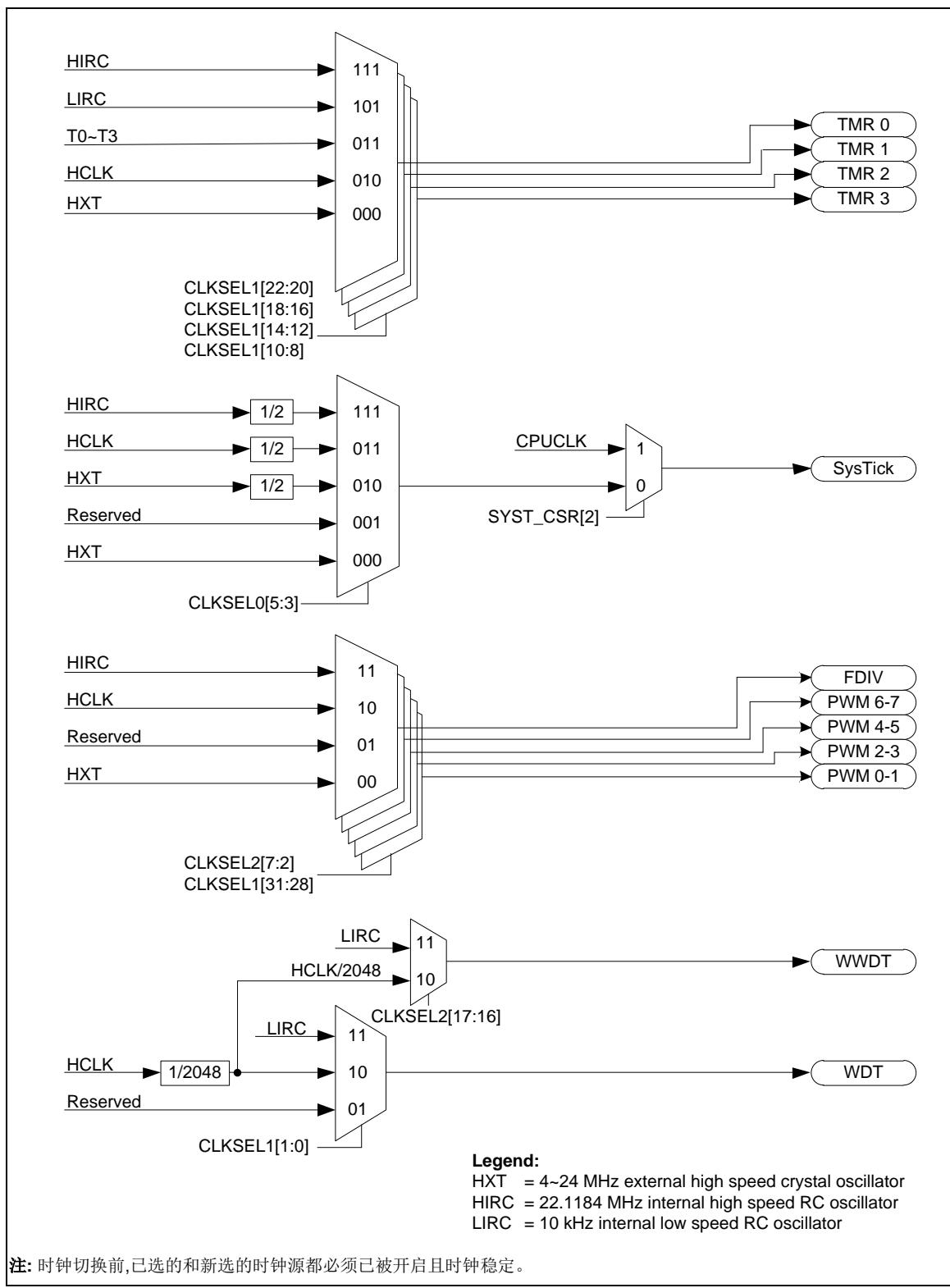


图 5-8 NuMicro® NUC029xAN 时钟源控制器概图(2/2)

5.3.2 系统时钟 和 SysTick 时钟

系统时钟有4个时钟源，由时钟发生器模块产生。使用寄存器HCLK_S(CLKSEL0[2:0])可以切换不同的时钟，系统时钟框图如图 5-9所示

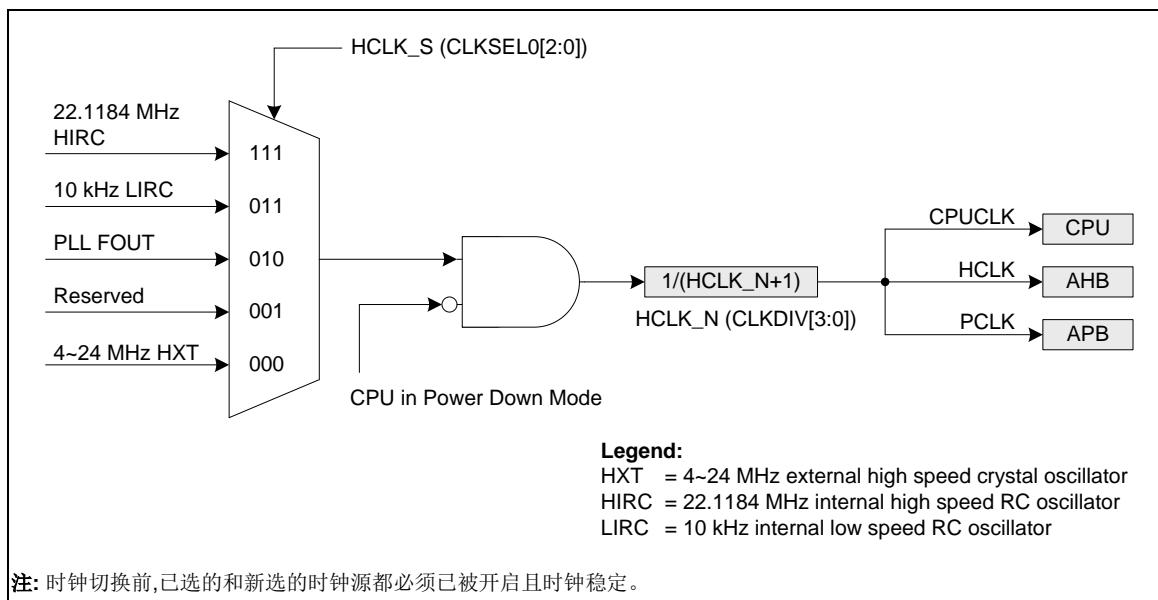


图 5-9 NuMicro® NUC029xAN 系统时钟框图

在Cortex®-M0核中的SysTick的时钟源可以使用CPU时钟或者外部时钟(SYST_CSR[2])。如果使用外部时钟，SysTick时钟(STCLK)有4个时钟源。时钟源切换使用寄存器STCLK_S(CLKSEL0[5:3])的设置。SysTick时钟框图如图 5-10所示。

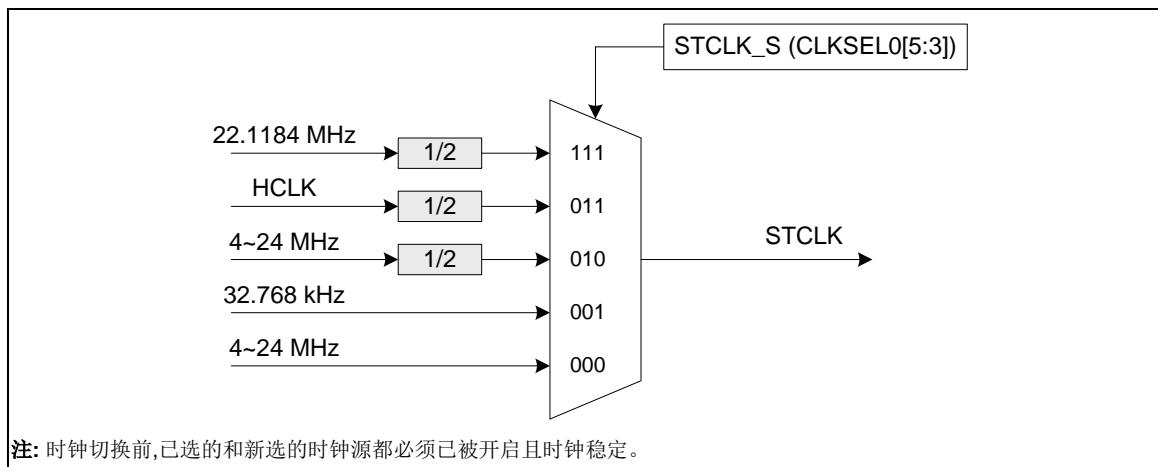


图 5-10 NuMicro® NUC029xAN SysTick 时钟控制框图

5.3.3 掉电模式时钟

当芯片进入掉电模式后，系统时钟、部分时钟源和部分外设时钟将会被禁用。还有部分时钟源和外设时钟在入掉电模式中仍然可以保持激活状态。

下列时钟仍然可以保持激活状态，

- 时钟生成器
 - 10 kHz 内部低速RC震荡时钟(LIRC)
- 外设时钟(采用10 kHz内部低速RC震荡时钟为时钟源时)

5.3.4 分频器输出

该设备配备有一个由16级2分频移位寄存器组成的分频器。可通过一个16选1的多路转换器选择16级移位寄存器中的一组，将其映射到CKO引脚并输出。所以有16种以2为幂的时钟分频选择，频率从 $F_{in}/2^1$ 到 $F_{in}/2^{16}$ ，其中 F_{in} 为输入到时钟分频器的时钟频率。

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

当写1到DIVIDER_EN(FRQDIV[4])时，链计数器开始计数，当写0到DIVIDER_EN(FRQDIV[4])时，链计数器持续计数直到分频时钟达到低状态并停留在低状态。

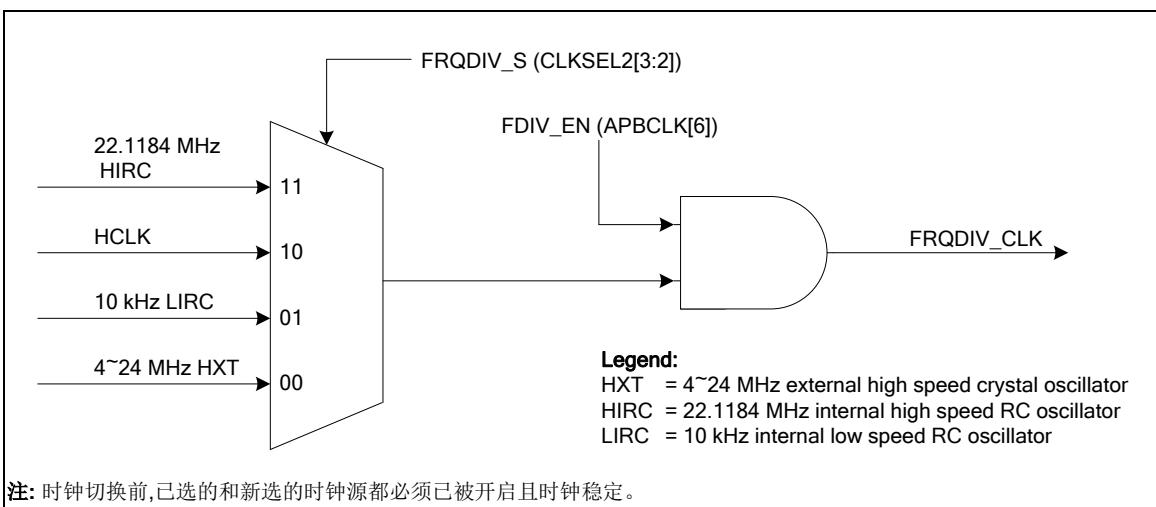


图 5-11 NuMicro® NUC029xAN 分频器的时钟源

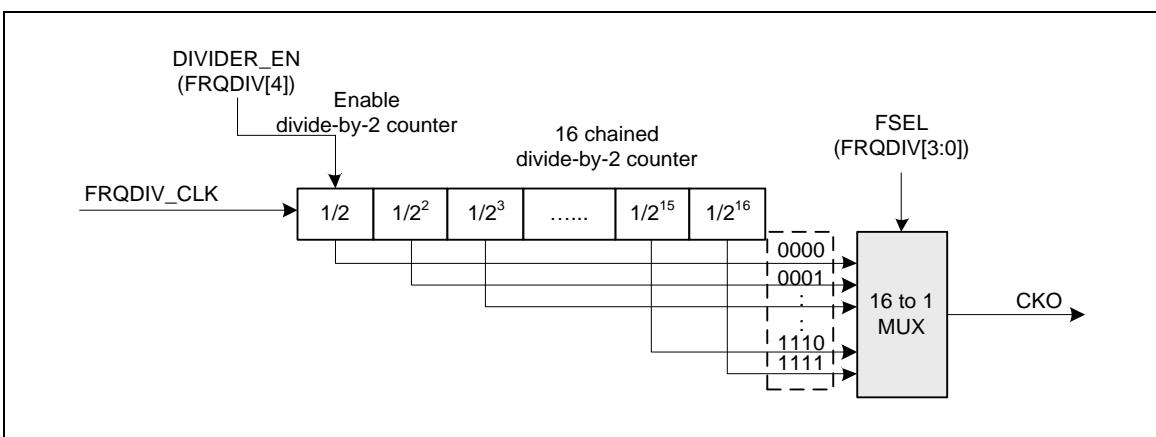


图 5-12 NuMicro® NUC029xAN 分频器框图

5.3.5 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
CLK基址:				
CLK_BA = 0x5000_0200				
PWRCON	CLK_BA + 00	R/W	系统掉电控制寄存器	0x0000_001X
AHBCLK	CLK_BA + 04	R/W	AHB 设备时钟使能控制寄存器	0x0000_000D
APBCLK	CLK_BA + 08	R/W	APB 设备时钟使能控制寄存器	0x0000_000X
CLKSTATUS	CLK_BA + 0C	R/W	时钟状态监控寄存器	0x0000_00XX
CLKSEL0	CLK_BA + 10	R/W	时钟源选择控制寄存器0	0x0000_003X
CLKSEL1	CLK_BA + 14	R/W	时钟源选择控制寄存器1	0xFFFF_FFFF
CLKDIV	CLK_BA + 18	R/W	时钟分频数寄存器	0x0000_0000
CLKSEL2	CLK_BA + 1C	R/W	时钟源选择控制寄存器2	0x0000_00FF
PLLCON	CLK_BA + 20	R/W	PLL控制寄存器	0x0005_C22E
FRQDIV	CLK_BA + 24	R/W	分频器控制寄存器	0x0000_0000

5.3.6 寄存器描述

掉电控制寄存器 (PWRCON)

除BIT[6]外，PWRCON的其他位都受保护。要编程这些被保护的位需要向地址0x5000_0100依次写入"59h", "16h", "88h"以关闭寄存器保护。参看寄存器REGWRPROT，地址GCR_BA + 0x100.

寄存器	偏移量	R/W	描述	复位值
PWRCON	CLK_BA +0x 00	R/W	系统掉电控制寄存器	0x0000_001X

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							PD_WAIT_CPU
7	6	5	4	3	2	1	0
PWR_DOWN_EN	PD_WU_STS	PD_WU_INT_EN	PD_WU_DLY	OSC10K_EN	OSC22M_EN	保留	XTL12M_EN

位	描述
[31:9]	保留
[8]	PD_WAIT_CPU 控制进入掉电模式的条件(写保护) 0 = PWR_DOWN_EN 置1时，芯片进入掉电模式 1 = PWR_DOWN_EN置1并且CPU执行WFI指令时，芯片进入掉电模式。
[7]	PWR_DOWN_EN 使能系统掉电模式(写保护) 该位置“1”，使能芯片的掉电模式，掉电行为取决于PD_WAIT_CPU 位 (a) PD_WAIT_CPU 为“0”，置位PWR_DOWN_EN 后，芯片马上进入掉电模式。 (b) PD_WAIT_CPU 为“1”，置位PWR_DOWN_EN 后，芯片不马上进入掉电模式，要等CPU执行WFI指令之后，芯片才进入掉电模式 芯片从掉电模式唤醒时，该位由硬件自动清零，在下次掉电时，用户需要重新置位该位。 掉电模式下，4~24MHz外部高速晶振（HXT）与22.1184 MHz内部高速振荡器(HIRC)被关闭，内部10 KHz低速振荡器(LIRC)不受该位控制。 掉电时，PLL 与系统时钟也被关闭，时钟源选择被忽略。如果外设以10KHz低速振荡器作为时钟，则其时钟不受该位控制 0 = 芯片工作于正常模式或CPU执行WFI指令进入空闲模式 1 = 芯片立即进入掉电模式 或等待CPU休眠命令WFI
[6]	PD_WU_STS 掉电唤醒状态标志 芯片从掉电模式下被唤醒时，该位置位 如果GPIO, UART , WDT, ACMP或者BOD唤醒CPU，该标志置位 注: 该位只有在 PD_WU_INT_EN (PWRCON[5]) 被置为 1的时候才会有效。写1清除为零。

[5]	PD_WU_INT_EN	掉电模式唤醒中断使能(写保护) 0 = 禁用 1 = 使能。从掉电唤醒时，产生中断。 注： 当 PD_WU_STS 和 PD_WU_INT_EN 都为高时中断将发生
[4]	PD_WU_DLY	唤醒延迟计数器使能(写保护) 当芯片从掉电模式唤醒时，该域可以控制延迟一定时钟周期以等待系统时钟稳定。 当芯片工作于外部4~24MHz高速晶振（HXT）时，延迟时间为4096个时钟周期，工作于内部22.1184MHz(HIRC)时，延迟256个时钟周期。 0 = 禁用时钟周期延迟 1 = 使能时钟周期延迟
[3]	OSC10K_EN	内部 10KHz 低速振荡器使能控制 (写保护) 0 = 禁用10KHz 低速振荡器 1 = 使能10KHz 低速振荡器
[2]	OSC22M_EN	内部 22.1184MHz 高速振荡器使能控制 (写保护) 0 = 禁用22.1184MHz高速振荡器 1 = 使能22.1184MHz高速振荡器
[1]	保留	保留
[0]	XTL12M_EN	外部4~24 MHz晶振使能控制 (写保护) 该位的默认值由flash用户配置寄存器config0 [26:24] 设置。当默认时钟源为外部高速晶振（4~24MHz时）。该位自动置1 0 = 禁用外部4 ~ 24 MHz高速晶振（HXT） 1 = 使能外部4 ~ 24 MHz高速晶振（HXT）

寄存器/指令模式	SLEEPDEEP (SCR[2])	PWR_DOWN_EN (PWRCON[8])	PD_WAIT_CPU (PWRCON[7])	CPU运行WFI 指令	时钟关闭
正常运行模式	0	0	0	NO	所有时钟可通过控制寄存器关闭
空闲模式 (CPU 进入休眠模式)	0	X	0	YES	仅CPU内部时钟关闭
掉电模式 (CPU 进入深度休眠模式)	1	1	1	YES	大部分时钟关闭，除了10kHz以及以10kHz 为时钟源的WDT模块仍然处于激活状态

表 5-7 掉电模式控制表

当芯片进入掉电模式时，用户可以通过一些中断唤醒CPU。在设定PWR_DOWN_EN位(PWRCON[7])之前用户应该使能相关中断源和NVIC IRQ使能位(NVIC_ISER)，以确保进入掉电模式以后可以成功唤醒。

AHB设备时钟使能控制寄存器 (AHBCLK)

该寄存器中各位用于使能/禁用系统和EBI时钟.

寄存器	偏移量	R/W	描述	复位值
AHBCLK	CLK_BA + 04	R/W	AHB设备时钟使能控制寄存器	0x0000_000D

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			HDIV_EN	EBI_EN	ISP_EN	保留	保留

位	描述	
[31:5]	保留	保留
[4]	HDIV_EN	除法器时钟使能控制 0 = 除法器时钟关闭 1 = 除法器时钟使能.
[3]	EBI_EN	EBI 控制器时钟使能控制 0 = 禁用 EBI 控制器时钟. 1 = 使能 EBI 控制器时钟.
[2]	ISP_EN	Flash ISP 控制器时钟使能控制 0 = 禁用 Flash ISP 控制器时钟. 1 = 使能 Flash ISP 控制器时钟.
[1:0]	保留	保留

APB 设备时钟使能控制寄存器(APBCLK)

该寄存器中的各位用于使能/禁用外设时钟.

寄存器	偏移量	R/W	描述	复位值
APBCLK	CLK_BA +0x 08	R/W	APB设备时钟使能控制寄存器	0x0000_000X

31	30	29	28	27	26	25	24
ACMP23_EN	ACMP01_EN	保留	ADC_EN	保留			
23	22	21	20	19	18	17	16
PWM67_EN	PWM45_EN	PWM23_EN	PWM01_EN	保留		UART1_EN	UART0_EN
15	14	13	12	11	10	9	8
保留		SPI1_EN	SPI0_EN	保留		I2C1_EN	I2C0_EN
7	6	5	4	3	2	1	0
保留	FDIV_EN	TMR3_EN	TMR2_EN	TMR1_EN	TMR0_EN	保留	WDT_EN

位	描述	
[31]	ACMP23_EN	模拟比较器 2/3 时钟使能控制 0 = 禁用 ACMP 2/3 时钟. 1 = 使能 ACMP 2/3 时钟.
[30]	ACMP_EN	模拟比较器0/1时钟使能控制 0 = 禁用 ACMP 0/1 时钟 1 = 使能 ACMP 0/1 时钟
[29]	保留	保留
[28]	ADC_EN	ADC时钟使能控制 0 = 禁用 ADC 时钟 1 = 使能 ADC 时钟
[27:24]	保留	保留
[23]	PWM67_EN	PWM_67 时钟使能控制 0 = 禁用 PWM67 时钟 1 = 使能 PWM67 时钟
[22]	PWM45_EN	PWM_45时钟使能控制 0 = 禁用 PWM45 时钟 1 = 使能 PWM45 时钟
[21]	PWM23_EN	PWM_23时钟使能控制 0 = 禁用 PWM23 时钟 1 = 使能 PWM23 时钟

[20]	PWM01_EN	PWM_01时钟使能控制 0 = 禁用 PWM01 时钟 1 = 使能 PWM01 时钟
[19:18]	保留	保留
[17]	UART1_EN	UART1 时钟使能控制 0 = 禁用 UART1 时钟 1 = 使能 UART1 时钟
[16]	UART0_EN	UART0 时钟使能控制 0 = 禁用 UART0 时钟 1 = 使能 UART0 时钟
[15:14]	保留	保留
[13]	SPI1_EN	SPI1 时钟使能控制 0 = 禁用 SPI1 时钟 1 = 使能 SPI1 时钟
[12]	SPI0_EN	SPI0 时钟使能控制 0 = 禁用 SPI0 时钟 1 = 使能 SPI0 时钟
[11:10]	保留	保留
[9]	I2C1_EN	I2C1 时钟使能控制 0 = 禁用 I2C1 时钟 1 = 使能 I2C1 时钟
[8]	I2C0_EN	I2C0 时钟使能控制 0 = 禁用 I2C0 时钟 1 = 使能 I2C0 时钟
[7]	保留	保留
[6]	FDIV_EN	分频器输出时钟使能控制 1 = 使能FDIV时钟 0 = 禁用FDIV时钟
[5]	TMR3_EN	Timer3 时钟使能控制 0 = 禁用定时器3时钟 1 = 使能定时器3时钟
[4]	TMR2_EN	Timer2 时钟使能控制 0 = 禁用定时器2时钟 1 = 使能定时器2时钟
[3]	TMR1_EN	Timer1 时钟使能控制 0 = 禁用定时器1时钟 1 = 使能定时器1时钟

[2]	TMR0_EN	Timer0 时钟使能控制 0 = 禁用定时器0时钟 1 = 使能定时器0时钟
[1]	保留	保留
[0]	WDT_EN	看门狗 时钟使能控制 (写保护) 0 = 禁用看门狗时钟 1 = 使能看门狗时钟 注： 该位是写保护的，对该位编程时，需要向0x5000_0100 依次写入“59h”，“16h”，“88h”来解除寄存器写保护，参考寄存器REGWRPROT，地址GCR_BA+0x100

时钟状态寄存器 (CLKSTATUS)

该寄存器各位用于监控芯片时钟是否稳定，时钟切换是否失败。

寄存器	偏移量	R/W	描述	复位值
CLKSTATUS	CLK_BA + 0C	R/W	时钟状态监控寄存器	0x0000_00XX

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
CLK_SW_FAIL	保留		OSC22M_STB	OSC10K_STB	PLL_STB	保留	XTL12M_STB

位	描述	
[31:8]	保留	保留-
[7]	CLK_SW_FAIL	<p>时钟切换失败标志 0 = 时钟切换成功 1 = 时钟切换失败</p> <p>注1：当软件切换系统时钟源时，该位被更新。如果切换目标时钟源稳定，该位清0；如果切换目标时钟源不稳定，该位置位</p> <p>注2：该位是只读的。选择的时钟源稳定后，硬件将自动切换系统时钟到选择的时钟源，CLK_SE_FAIL将被硬件自动清0</p>
[6:5]	保留	保留
[4]	OSC22M_STB	<p>内部 22.1184 MHz 高速振荡器(HIRC)稳定标志（只读） 0 = 内部 22.1184 MHz 高速振荡器(HIRC)不稳定或没有使能 1 = 内部 22.1184 MHz 高速振荡器(HIRC)稳定</p>
[3]	OSC10K_STB	<p>内部 10KHz 低速震荡器(LIRC)稳定标志（只读） 0 = 内部 10KHz 低速震荡器(LIRC)不稳定或没有使能 1 = 内部 10KHz 低速震荡器(LIRC)稳定</p>
[2]	PLL_STB	<p>PLL 时钟稳定标志（只读） 1 = PLL 时钟稳定 0 = PLL 时钟不稳定或没有使能</p>
[1]	保留	保留
[0]	XTL12M_STB	<p>外部 4-24MHz 高速晶振稳定标志（只读） 0 = 外部 4-24MHz 高速晶振不稳定或没有使能</p>

		1 = 外部4-24MHz高速晶振稳定
--	--	---------------------

时钟源选择控制寄存器 0 (CLKSEL0)

寄存器	偏移量	R/W	描述	复位值
CLKSEL0	CLK_BA + 0x10	R/W	时钟源选择控制寄存器0	0x0000_003X

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		STCLK_S			HCLK_S		

位	描述	
[31:6]	保留	保留
[5:3]	STCLK_S	<p>Cortex™-M0 SysTick 时钟源选择 (写保护)</p> <p>如果SYST_CSR[2] = 1, SysTick 时钟源来自HCLK.</p> <p>如果SYST_CSR[2] = 0, SysTick 时钟源由如下设定决定</p> <p>000 = 外部高速晶振HXT</p> <p>001 = 保留</p> <p>010 = 外部高速晶振HXT/2分频</p> <p>011 = HCLK/2</p> <p>111 = 内部高速振荡器HIRC/2分频.</p> <p>其它 = 保留</p> <p>注1: 该位是受保护位, 对该位编程时, 需要向0x5000_0100 依次写入“59h”, “16h”, “88h”来解除寄存器写保护, 参考寄存器REGWRPROT, 地址为GCR_BA + 0x100</p> <p>注2: 如果SysTick 时钟源不是来自HCLK(也就是SYST_CSR[2]=0), SysTick时钟源必须小于或者等于HCLK/2</p>

[2:0]	HCLK_S	<p>HCLK 时钟源选择 (写保护)</p> <p>000 = 外部高速晶振HXT 001 = 保留 010 = PLL 时钟 011 = LIRC 111 = HIRC Others = 保留.</p> <p>注1: 在时钟切换之前，相关时钟源(当前和新选)都必须打开并且稳定</p> <p>注2: 任何复位后，默认值将从用户配置寄存器CFOSC(CONFIG0[26:24]) 加载，所以默认值要么为000b 要么为 111b.</p> <p>注3: 该位是写保护位，对该位编程时，需要向0x5000_0100 依次写入“59h”，“16h”，“88h”来解除寄存器写保护，参考寄存器REGWRPROT，地址为GCR_BA + 0x100</p>
-------	--------	--

时钟源选择控制寄存器 1 (CLKSEL1)

在时钟切换前，相关的时钟源（当前和新选），必须都打开。

寄存器	偏移量	R/W	描述				复位值
CLKSEL1	CLK_BA + 0x14	R/W	时钟源选择控制寄存器1				0xFFFF_FFFF

31	30	29	28	27	26	25	24
PWM23_S	PWM01_S			保留		UART_S	
23	22	21	20	19	18	17	16
保留	TMR3_S			保留	TMR2_S		
15	14	13	12	11	10	9	8
保留	TMR1_S			保留	TMR0_S		
7	6	5	4	3	2	1	0
保留		SPI1_S	SPI0_S	ADC_S		WDT_S	

位	描述	
[31:30]	PWM23_S	PWM2 与 PWM3 的时钟源选择 PWM2 与 PWM3 使用相同的时钟源和相同的预分频 00 = HXT 01 = LIRC 10 = HCLK 11 = HIRC
[29:28]	PWM01_S	PWM0 与 PWM1 的时钟源选择 PWM0 与 PWM1 使用相同的时钟源和相同的预分频 00 = HXT 01 = LIRC 10 = HCLK 11 = HIRC
[27:26]	保留	保留
[25:24]	UART_S	UART时钟源选择 00 = HXT 01 = PLL 10 = 保留 11 = HIRC
[23]	保留	保留

[22:20]	TMR3_S	TIMER3 时钟源选择 000 = HXT 010 = HCLK 011 = 时钟来自外部T3引脚 101 = LIRC 111 = HIRC Others = 保留
[19]	保留	保留
[18:16]	TMR2_S	TIMER2 时钟源选择 000 = HXT 010 = HCLK 011 = 时钟来自外部T2引脚 101 = LIRC 111 = HIRC Others = 保留
[15]	保留	保留
[14:12]	TMR1_S	TIMER1 时钟源选择 000 = HXT 010 = HCLK 011 = 时钟来自外部T1引脚 101 = LIRC 111 = HIRC Others = 保留
[11]	保留	保留
[10:8]	TMR0_S	TIMER0 时钟源选择 000 = HXT 010 = HCLK 011 = 时钟来自外部T0引脚 101 = LIRC 111 = HIRC Others = 保留
[7:6]	保留	保留
[5]	SPI1_S	SPI1 时钟源选择 0 = PLL. 1 = HCLK.
[4]	SPI0_S	SPI0 时钟源选择 0 = PLL. 1 = HCLK

[3:2]	ADC_S	ADC 时钟源选择 00 = HXT 01 = PLL 10 = HCLK 11 = HIRC
[1:0]	WDT_S	看门狗 时钟源选择(写保护) 00 =保留 01 =保留 10 = HCLK/2048 11 = LIRC 注: 受保护位, 对该位编程时, 需要向0x5000_0100 依次写入“59h”, “16h”, “88h” 来解除寄存器写保护, 参考寄存器REGWRPROT, 地址为GCR_BA + 0x100

时钟源选择控制寄存器 2 (CLKSEL2)

在时钟切换前，相关的时钟源(当前和新选的)，必须都打开.

寄存器	偏移量	R/W	描述	复位值
CLKSEL2	CLK_BA + 0x1C	R/W	时钟源选择控制寄存器 2	0x0000_00FF

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留						WWDT_S	
15	14	13	12	11	10	9	8
保留							
7	6	5	4	3	2	1	0
PWM67_S	PWM45_S			FRQDIV_S		保留	

位	描述	
[31:18]	保留	保留
[17:16]	WWDT_S	窗口看门狗时钟源选择 10 = 时钟源来自HCLK/2048 11 = 时钟源来自LIRC
[15:8]	保留	保留
[7:6]	PWM67_S	PWM6 与 PWM7的时钟源选择 PWM6 与 PWM7使用相同的时钟源和相同的预分频器 00 = HXT 01 = LIRC 10 = HCLK 11 = HIRC
[5:4]	PWM45_S	PWM4 与 PWM5的时钟源选择 PWM4 与 PWM5使用相同的时钟源和相同的预分频器 00 = HXT 01 = LIRC 10 = HCLK 11 = HIRC
[3:2]	FRQDIV_S	时钟分频器时钟源选择 00 = HXT 01 = LIRC 10 = HCLK 11 = HIRC

[1:0]	保留	保留	
-------	----	----	--

时钟分频寄存器 (CLKDIV)

寄存器	偏移量	R/W	描述	复位值
CLKDIV	CLK_BA_+ 0x18	R/W	时钟分频寄存器	0x0000_0000

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

位	描述	
[31:24]	保留	保留
[23:16]	ADC_N	ADC时钟频率=ADC时钟源频率/ (ADC_N + 1)
[15:12]	保留	保留
[11:8]	UART_N	UART时钟频率 = (UART时钟源频率) / (UART_N + 1)
[7:4]	保留	保留
[3:0]	HCLK_N	HCLK 时钟频率 = (HCLK时钟源频率) / (HCLK_N + 1)

PLL控制寄存器 (PLLCON)

PLL的参考时钟输入来自外部高速晶振时钟HXT (4~24MHz) 或内部22.1184MHz高速振荡器HIRC，该寄存器用于控制PLL的输出频率和PLL的操作模式

寄存器	偏移量	R/W	描述	复位值
PLLCON	CLK_BA + 0x20	R/W	PLL控制寄存器	0x0005_C22E

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留				PLL_SRC	OE	BP	PD
15	14	13	12	11	10	9	8
OUT_DV		IN_DV				FB_DV	
7	6	5	4	3	2	1	0
FB_DV							

位	描述	
[19]	PLL_SRC	PLL时钟源选择 0 = PLL时钟源为外部高速晶振HXT (4~24MHz) 1 = PLL时钟源为22.1184 MHz 振荡器HIRC
[18]	OE	PLL OE (FOUT Enable)引脚控制 0 = 使能 PLL FOUT 1 = PLL FOUT 固定为低
[17]	BP	PLL 直通控制 0 = PLL 正常模式 (默认) 1 = PLL 时钟输出与时钟输入相同
[16]	PD	掉电模式 如果设置PWRCON寄存器的PWR_DOWN_EN位被设为"1"， PLL也会进入掉电模式 0 = PLL正常模式 1 = PLL 掉电模式(默认)
[15:14]	OUT_DV	PLL 输出分频控制 参考下表的公式
[13:9]	IN_DV	PLL输入分频控制 参考下表的公式
[8:0]	FB_DV	PLL反馈分频控制 参考下表的公式

PLL输出时钟频率设置:

$$F_{OUT} = F_{IN} \times \frac{NF}{NR} \times \frac{1}{NO}$$

约束条件:

1. $4\text{MHz} < F_{IN} < 24\text{MHz}$
2. $800\text{KHz} < \frac{F_{IN}}{2 * NR} < 7.5\text{MHz}$
3. $100\text{MHz} < F_{CO} = F_{IN} \times \frac{NF}{NR} < 200\text{MHz}$
 $120\text{MHz} < F_{CO}$ is preferred

符号	说明
FOUT	输出时钟频率
FIN	输入(参考)时钟频率
NR	输入分频 (IN_DV + 2)
NF	反馈分频 (FB_DV + 2)
NO	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 2 OUT_DV = "11" : NO = 4

默认PLL频率设置

PLLCON 默认值: 0xC22E
 FIN = 12 MHz
 NR = (1+2) = 3
 NF = (46+2) = 48
 NO = 4
 $F_{OUT} = 12/4 \times 48 \times 1/3 = 48\text{MHz}$

频率分频器控制寄存器 (FRQDIV)

寄存器	偏移量	R/W	描述	复位值
FRQDIV	CLK_BA+ 0x24	R/W	频率分频器控制寄存器	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			DIVIDER_EN	FSEL			

位	描述	
[31:5]	保留	保留
[4]	DIVIDER_EN	<p>频率分频器使能控制 0 = 禁用频率分频 1 = 使能频率分频</p>
[3:0]	FSEL	<p>分频器输出频率选择位 输出频率的公式是 $F_{out} = F_{in}/2^{(N+1)}$, F_{in} 为输入时钟频率 F_{out} 为分频器输出时钟频率 N 为FSEL[3:0]的值。</p>

5.4 NuMicro® NUC029FAE时钟控制器

5.4.1 概述

时钟控制器为整个芯片提供时钟，包括系统时钟和所有外设时钟。时钟控制器还通过独立的时钟ON/OFF控制、时钟源选择和时钟分频器，实现电源控制功能。在将PWR_DOWN_EN(PWRCON[7])和PD_WAIT_CPU(PWRCON[8])置位且Cortex®-M0执行WFI指令之后，芯片会进入掉电模式，在那之后，芯片等待被唤醒中断源触发以离开掉电模式。在掉电模式下，时钟控制器关闭外部4~24 MHz高速晶振(HXT)和内部22.1184MHz高速振荡器(HIRC)，以降低整个系统的功耗。下图显示时钟发生器和时钟源控制。

时钟发生器由以下3个时钟源组成：

- 外部 4~24 MHz 高速晶振(HXT) 或 32.768 kHz低速晶振(LXT)
- 内部 22.1184 MHz RC高速振荡器(HIRC)
- 内部 10KHz 低速振荡器(LIRC)

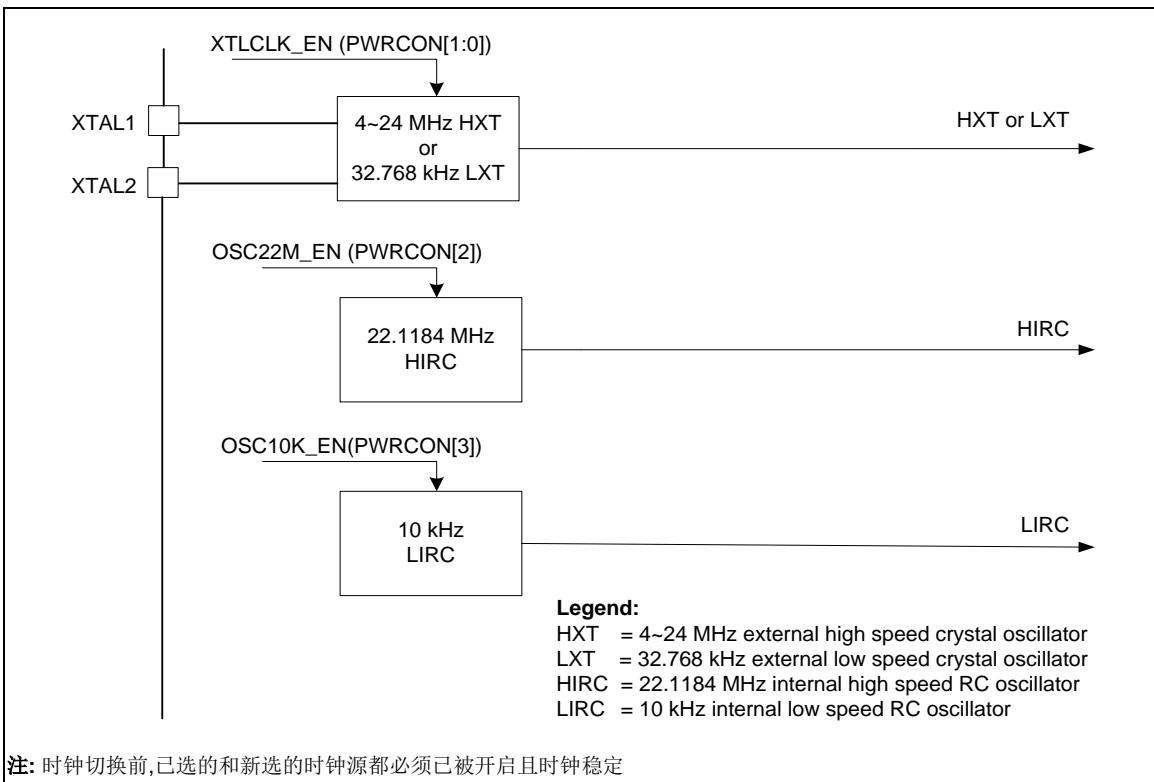


图 5-13 NuMicro® NUC029FAE 时钟发生器框图

5.4.2 系统时钟 和 SysTick 时钟

系统时钟有3个时钟源，由时钟发生器模块产生。使用寄存器HCLK_S(CLKSEL0[2:0])可以切换不同的时钟，系统时钟框图如图 5-14所示

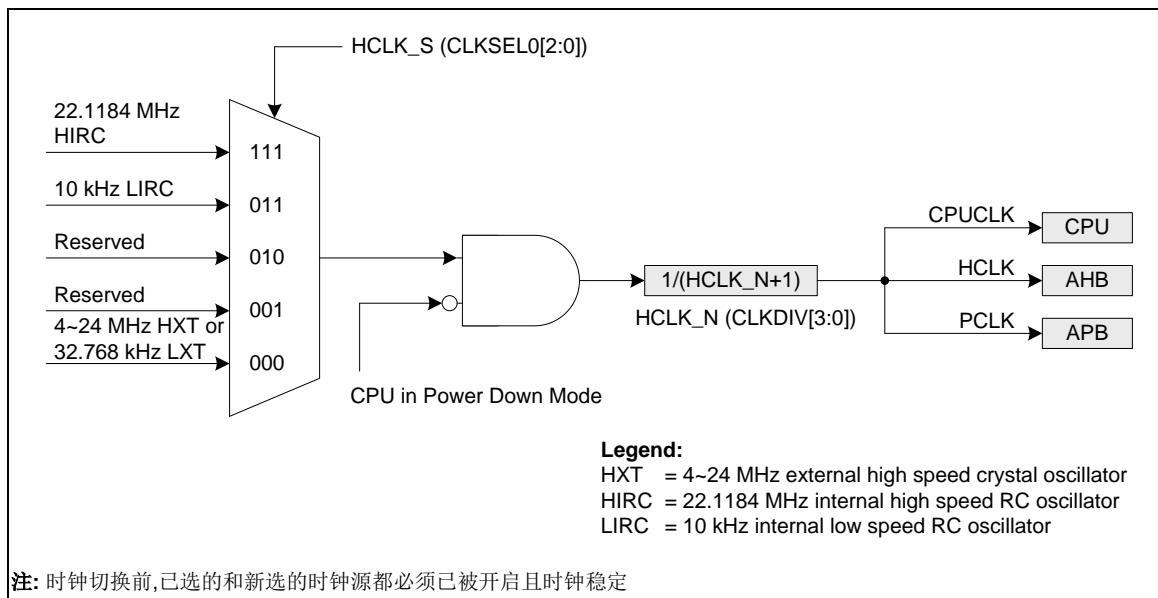


图 5-14 NuMicro® NUC029FAE 系统时钟框图

在Cortex®-M0核中的SysTick的时钟源可以使用CPU时钟或者外部时钟(SYST_CSR[2])。如果使用外部时钟，SysTick时钟(STCLK)有4个时钟源。时钟源切换使用寄存器STCLK_S(CLKSEL0[5:3])的设置。SysTick时钟框图如图 5-15所示。

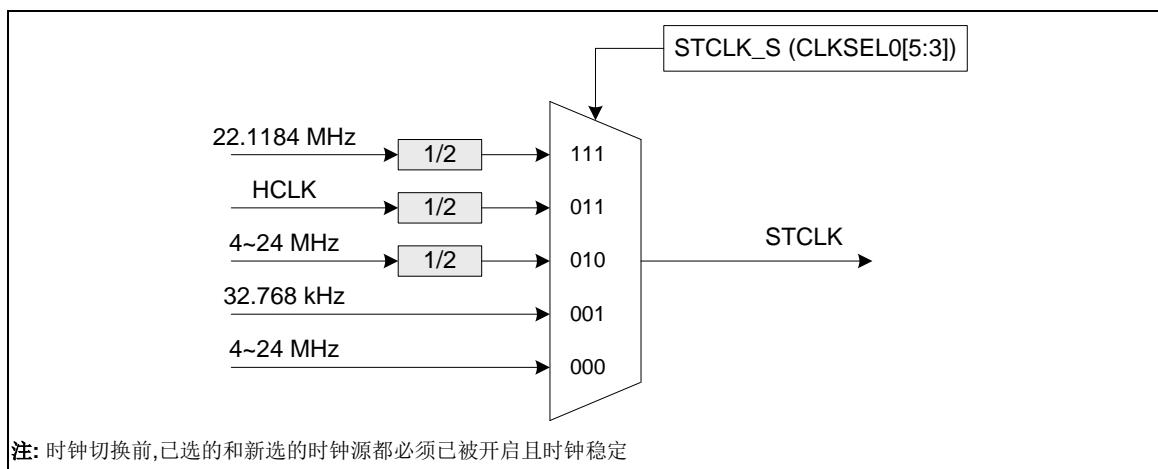


图 5-15 NuMicro® NUC029FAE SysTick 时钟控制框图

5.4.3 ISP时钟源选择

ISP时钟源来自AHB 时钟 (HCLK)。请参看寄存器AHBCLK。

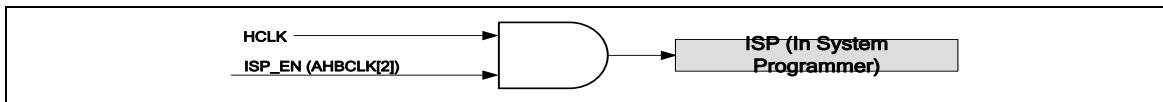


图 5-16 NuMicro® NUC029FAE AHB HCLK时钟源选择

5.4.4 模块时钟源选择

外设时钟有多个时钟源选择，不同的外设可切换不同的时钟源设置。

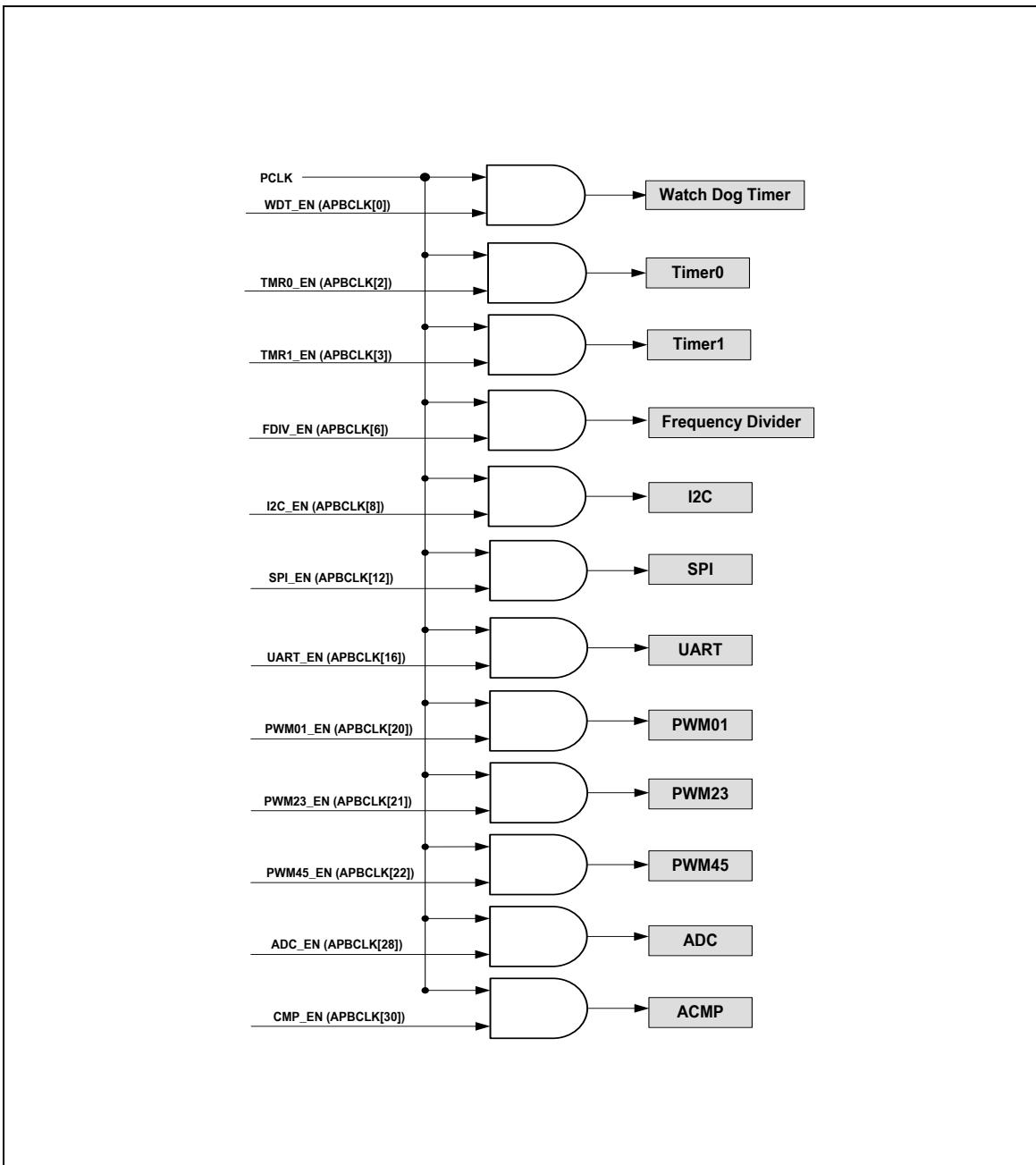


图 5-17 NuMicro® NUC029FAE PCLK 外设时钟源选择

	Ext. CLK (HXT Or LXT)	HIRC	LIRC	PCLK
WDT	Yes	No	Yes	Yes
Timer0	Yes	Yes	Yes	Yes
Timer1	Yes	Yes	Yes	Yes
I ² C	No	No	No	Yes
SPI	No	No	No	Yes
UART	Yes	Yes	No	No
PWM	No	No	No	Yes
ADC	Yes	Yes	No	Yes
ACMP	No	No	No	Yes

表 5-8 NuMicro® NUC029FAE 外设时钟源选择表

5.4.5 掉电模式时钟

当芯片进入掉电模式后，系统时钟、部分时钟源和部分外设时钟将会被禁用。还有部分时钟源和外设时钟在入掉电模式中仍然可以保持激活状态。

下列时钟仍然可以保持激活状态：

- 时钟生成器
 - 10 kHz 内部低速RC震荡时钟 (LIRC)
- 32.768 kHz外部低速晶振时钟(LXT)(如果PD_32K = 1 以及XTLCLK_EN[1:0] = 10)
- 外设时钟(采用10 kHz内部低速RC震荡时钟为时钟源)
 - Watchdog Clock看门狗时钟
 - Timer 0/1 Clock时钟

5.4.6 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
CLK基地址: CLK_BA = 0x5000_0200				
PWRCON	CLK_BA+0x00	R/W	系统掉电控制寄存器	0x0000_001C
AHBCLK	CLK_BA+0x04	R/W	AHB 设备时钟使能控制寄存器	0x0000_0005
APBCLK	CLK_BA+0x08	R/W	APB 设备时钟使能控制寄存器	0x0000_0001
CLKSTATUS	CLK_BA+0x0C	R/W	时钟状态监控寄存器	0x0000_0018
CLKSEL0	CLK_BA+0x10	R/W	时钟源选择控制寄存器0	0x0000_003F
CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0xAFFF_FFFF
CLKDIV	CLK_BA+0x18	R/W	时钟分频数寄存器	0x0000_0000
CLKSEL2	CLK_BA+0x1C	R/W	时钟源选择控制寄存器2	0x0000_00EF
FRQDIV	CLK_BA+0x24	R/W	分频器控制寄存器	0x0000_0000

5.4.7 寄存器描述

掉电控制寄存器Power-down Control Register (PWRCON)

除BIT[6]外，PWRCON的其他位都受保护。要编程这些被保护的位需要向地址0x5000_0100依次写入"59h", "16h", "88h"以禁用寄存器保护。参考寄存器REGWRPROT，地址GCR_BA + 0x100.

寄存器	偏移量	R/W	描述				复位值
PWRCON	CLK_BA+0x00	R/W	系统掉电控制寄存器				0x0000_001C

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						PD_32K	保留
7	6	5	4	3	2	1	0
PWR_DOWN_EN	PD_WU_STS	PD_WU_INT_EN	PD_WU_DLY	OSC10K_EN	OSC22M_EN	XTLCLK_EN	

位	描述	
[31:10]	保留	保留
[9]	PD_32K	掉电模式下使能 LXT 该位用于控制晶振时钟在掉电模式下是否有效。 0 = 掉电模式下无效 1 = 当XTLCLK_EN[1:0] = 10时，LXT在掉电模式下依旧有效。.
[8]	保留	保留
[7]	PWR_DOWN_EN	使能系统掉电模式(写保护) 芯片由掉电模式唤醒时，该位由硬件自动清零，在下次掉电时，用户需要重新置位该位。 掉电模式下，4~24MHz外部高速晶振（HXT），32.768 kHz外部低速晶振(LXT)与22.1184 MHz内部高速振荡器(HIRC)被关闭，内部10 KHz低速振荡器(LIRC)不受该位控制 掉电时，PLL与系统时钟也被关闭，时钟源选择被忽略。如果外设以10KHz低速振荡器作为时钟，则其时钟不受该位控制 0 = 芯片工作于正常模式或CPU执行WFI指令进入空闲模式 1 = 芯片立即进入掉电模式 或等待CPU休眠命令WFI
[6]	PD_WU_STS	掉电唤醒状态标志 芯片从掉电模式下被唤醒时，该位置位。 如果GPIO, UART, WDT, I ² C, ACMP, Timer 或BOD唤醒CPU，该标志置位 注:该位只有在 PD_WU_INT_EN (PWRCON[5]) 被置为 1 的时候才会有效。写1清除为零。

位	描述	
[5]	PD_WU_INT_EN	<p>掉电模式唤醒中断使能(写保护) 0 = 禁用 1 = 使能。从掉电唤醒时，产生中断。 注：当 PD_WU_STS 和 PD_WU_INT_EN 都为高时中断将发生</p>
[4]	PD_WU_DLY	<p>唤醒延迟计数器使能(写保护) 当芯片从掉电模式唤醒时，该域可以控制延迟一定时钟周期以等待系统时钟稳定。 当芯片工作于外部4~24MHz高速晶振（HXT）时，延迟时间为4096个时钟周期；工作于外部32.768 kHz低速晶振（LXT）时，延迟时间为4096个时钟周期；工作于内部22.1184MHz(HIRC)时，延迟16个时钟周期。 0 = 禁用时钟周期延迟 1 = 使能时钟周期延迟</p>
[3]	OSC10K_EN	<p>内部 10KHz 低速振荡器(LIRC)使能控制 (写保护) 0 = 禁用10KHz 低速振荡器(LIRC) 1 = 使能10KHz 低速振荡器(LIRC)</p>
[2]	OSC22M_EN	<p>内部 22.1184MHz高速振荡器(HIRC)使能控制 (写保护) 0 = 禁用22.1184MHz高速振荡器(HIRC) 1 = 使能22.1184MHz高速振荡器(HIRC) 注：OSC22M_EN默认值为1</p>
[1:0]	XTLCLK_EN[1:0]	<p>外部晶振HXT或LXT使能位 (写保护) 默认的时钟源为HIRC。这两位的默认值为"00"，且XTAL1、XTAL2管脚默认为GPIO功能。 00 = XTAL1和XTAL2为GPIO, 禁止LXT和HXT(默认值)。 01 = HXT使能。 10 = LXT使能。 11 = XTAL1为外部时钟输入管脚、XTAL2为GPIO管脚。 注意：为使能外部XTAL功能，必须在P5_MFP中设定P5_ALT[1:0] 和P5_MFP[1:0]。</p>

寄存器模式/指令模式	SLEEPDEEP (SCR[2])	PWR_DOWN_EN (PWRCON[7])	CPU运行WFI指令	时钟禁用
正常运行模式	0	0	NO	所有时钟皆可被禁用
空闲模式 (CPU 进入休眠模式)	0	0	YES	仅禁用 CPU 时钟
掉电模式 (CPU 进入深度休眠模)	1	1	YES	大部分时钟关闭，除了10kHz以及以10kHz 为时钟源的WDT模块仍然处于激活状态

表 5-9 掉电模式控制

当芯片进入掉电模式时，用户可以通过一些中断唤醒CPU。在设定PWR_DOWN_EN位(PWRCON[7])之前用户应该使能相关中断和NVIC IRQ使能位(NVIC_ISER)，以确保进入掉电模式以后可以成功唤醒。

AHB设备时钟使能控制寄存器 (AHBCLK)

该寄存器中各位用于使能/禁用系统时钟

寄存器	偏移量	R/W	描述	复位值
AHBCLK	CLK_BA+0x04	R/W	AHB设备时钟使能控制寄存器	0x0000_0005

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					ISP_EN	保留	保留

位	描述	
[31:3]	保留	保留
[2]	ISP_EN	Flash ISP 控制器时钟使能控制 0 = 禁用 Flash ISP 控制器时钟。 1 = 使能 Flash ISP 控制器时钟。
[1]	保留	保留
[0]	保留	保留

APB 设备时钟使能控制寄存器 (APBCLK)

该寄存器中的各位用于使能/禁用外设时钟.

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

31	30	29	28	27	26	25	24
保留	ACMP_EN	保留	ADC_EN	保留			
23	22	21	20	19	18	17	16
保留	PWM45_EN	PWM23_EN	PWM01_EN	保留			UART_EN
15	14	13	12	11	10	9	8
保留			SPI_EN	保留			I2C_EN
7	6	5	4	3	2	1	0
保留	FDIV_EN	保留		TMR1_EN	TMR0_EN	保留	WDT_EN

位	描述	
[31]	保留	保留
[30]	ACMP_EN	模拟比较器时钟使能控制 0 = 禁用 ACMP 时钟 1 = 使能 ACMP 时钟
[29]	保留	保留
[28]	ADC_EN	ADC时钟使能控制 0 = 禁用 ADC 时钟 1 = 使能 ADC 时钟
[27:23]	保留	保留
[22]	PWM45_EN	PWM_45时钟使能控制 0 = 禁用 PWM45 时钟 1 = 使能 PWM45 时钟
[21]	PWM23_EN	PWM_23时钟使能控制 0 = 禁用 PWM23 时钟 1 = 使能 PWM23 时钟
[20]	PWM01_EN	PWM_01时钟使能控制 0 = 禁用 PWM01 时钟 1 = 使能 PWM01时钟
[19:17]	保留	保留

位	描述	
[16]	UART_EN	UART 时钟使能控制 0 = 禁用 UART 时钟 1 = 使能 UART 时钟
[15:13]	保留	保留
[12]	SPI_EN	SPI 时钟使能控制 0 = 禁用 SPI 时钟 1 = 使能 SPI 时钟
[11:9]	保留	保留
[8]	I2C_EN	I2C 时钟使能控制 0 = 禁用 I2C 时钟 1 = 使能 I2C 时钟
[7]	保留	保留
[6]	FDIV_EN	分频器输出时钟使能控制 0 = 禁用FDIV时钟 1 = 使能FDIV时钟
[5:4]	保留	保留
[3]	TMR1_EN	Timer1 时钟使能控制 0 = 禁用定时器1时钟 1 = 使能定时器1时钟
[2]	TMR0_EN	Timer0 时钟使能控制 0 = 禁用定时器0时钟 1 = 使能定时器0时钟
[1]	保留	保留
[0]	WDT_EN	看门狗 时钟使能控制 (写保护) 0 = 禁用看门狗时钟 1 = 使能看门狗时钟 注: 该位是写保护的, 对该位编程时, 需要向0x5000_0100 依次写入“59h”, “16h”, “88h” 来解除寄存器写保护, 参考寄存器REGWRPROT, 地址GCR_BA+0x100

时钟状态寄存器(CLKSTATUS)

该寄存器各位用于监控芯片时钟是否稳定，时钟切换是否失败。

寄存器	偏移量	R/W	描述	复位值
CLKSTATUS	CLK_BA+0x0C	R/W	时钟状态监控寄存器	0x0000_0018

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
CLK_SW_FAIL	保留		OSC22M_STB	OSC10K_STB	保留		XTL_STB

位	描述	
[31:8]	保留	保留
[7]	CLK_SW_FAIL	时钟切换失败标志 0 = 时钟切换成功 1 = 时钟切换失败 注1： 当软件切换系统时钟源时，该位被更新。如果切换目标时钟源稳定，该位清0；如果切换目标时钟源不稳定，该位置位 注2： 该位是只读的。选择的时钟源稳定后，硬件将自动切换系统时钟到选择的时钟源，CLK_SE_FAIL将被硬件自动清0
[6:5]	保留	保留
[4]	OSC22M_STB	内部 22.1184 MHz 高速振荡器(HIRC)稳定标志 (只读) 0 = 内部 22.1184 MHz 高速振荡器(HIRC)不稳定或没有使能 1 = 内部 22.1184 MHz 高速振荡器(HIRC)稳定
[3]	OSC10K_STB	内部 10KHz 低速震荡器(LIRC)稳定标志 (只读) 0 = 内部 10KHz 低速震荡器(LIRC)不稳定或没有使能 1 = 内部 10KHz 低速震荡器(LIRC)稳定
[2:1]	保留	保留
[0]	XTL_STB	HXT 或 LXT 时钟稳定标志 0 = HXT 或 LXT 时钟不稳定或已禁止 1 = HXT 或 LXT 时钟稳定。

时钟源选择控制寄存器 0 (CLKSEL0)

寄存器	偏移量	R/W	描述	复位值
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
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		STCLK_S			HCLK_S		

位	描述	
[31:6]	保留	保留.
[5:3]	STCLK_S[2:0]	<p>Cortex™-M0 SysTick 时钟源选择 (写保护)</p> <p>如果SYST_CSR[2] = 1, SysTick 时钟源来自HCLK.</p> <p>如果SYST_CSR[2] = 0, SysTick 时钟源由如下设定决定</p> <p>000 = 外部晶振HXT 或 LXT</p> <p>001 = 保留</p> <p>010 = 外部晶振HXT 或LXT /2分频</p> <p>011 = 时钟源来自HCLK/2</p> <p>111 = 内部高速振荡器HIRC/2分频.</p> <p>其它 = 保留</p> <p>注1: 该位是受保护位, 对该位编程时, 需要向0x5000_0100 依次写入“59h”, “16h”, “88h”来解除寄存器写保护, 参考寄存器REGWRPROT, 地址为GCR_BA + 0x100</p> <p>注2: 如果SysTick 时钟源不是来自HCLK(也就是SYST_CSR[2]=0), SysTick时钟源必须小于或者等于HCLK/2</p> <p>注3: 设置 PWRCON[1:0],用于选择HXT 或LXT晶振时钟</p>

位	描述
[2:0]	<p>HCLK 时钟源选择 (写保护)</p> <p>000 = 外部晶振HXT 或LXT 001 = 保留 010 = 保留 011 = 内部低速振荡器时钟LIRC 111 = 内部高速振荡器时钟HIRC Others = 保留.</p> <p>注1: 在时钟切换之前，相关时钟源(当前和新选)都必须打开并且稳定</p> <p>注2: 该位是受保护位，对该位编程时，需要向0x5000_0100 依次写入“59h”，“16h”，“88h”来解除寄存器写保护，参考寄存器REGWRPROT，地址为GCR_BA + 0x100</p> <p>注3: 设置 PWRCON[1:0],用于选择HXT 或LXT晶振时钟</p>

时钟源选择控制寄存器1 (CLKSEL1)

在时钟切换前，相关的时钟源（当前和新选），必须都打开。

寄存器	偏移量	R/W	描述	复位值
CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0xAFFF_FFFF

31	30	29	28	27	26	25	24
保留						UART_S	
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
保留	TMR1_S			保留	TMR0_S		
7	6	5	4	3	2	1	0
保留			SPI_S	ADC_S		WDT_S	

位	描述	
[31:26]	保留	保留
[25:24]	UART_S[1:0]	UART时钟源选择 00 = 外部晶振HXT或LXT 01 = 保留 10 = HIRC 11 = HIRC <small>注:设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</small>
[23:15]	保留	保留
[14:12]	TMR1_S[2:0]	TIMER1 时钟源选择 000 = 外部晶振HXT或LXT 001 = LIRC 010 = HCLK 011 = 时钟来自外部触发 111 = HIRC Others = 保留 <small>注:设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</small>
[11]	保留	保留

位	描述	
[10:8]	TMR0_S[2:0]	<p>TIMER0 时钟源选择</p> <p>000 = 外部晶振HXT或LXT 001 = LIRC 010 = HCLK 011 = 时钟来自外部触发 111 = HIRC Others = 保留</p> <p>注:设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</p>
[7:5]	保留	保留
[4]	SPI_S	<p>SPI1 时钟源选择</p> <p>0 = 外部晶振HXT或LXT 1 = HCLK.</p> <p>注:设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</p>
[3:2]	ADC_S[1:0]	<p>ADC 时钟源选择</p> <p>00 = 外部晶振HXT或LXT 01 = 保留 10 = HCLK 11 = HIRC</p> <p>注:设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</p>
[1:0]	WDT_S[1:0]	<p>看门狗 时钟源选择(写保护)</p> <p>00 = 外部晶振HXT或LXT 01 = 保留 10 = HCLK/2048 11 = LIRC</p> <p>注1: 受保护位, 对该位编程时, 需要向0x5000_0100 依次写入“59h”, “16h”, “88h” 来解除寄存器写保护, 参考寄存器REGWRPROT, 地址为GCR_BA + 0x100</p> <p>注2: 设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟</p>

时钟源选择控制寄存器 2 (CLKSEL2)

在时钟切换前，相关的时钟源(当前和新选的)，必须都打开.

寄存器	偏移量	R/W	描述	复位值
CLKSEL2	CLK_BA+0x1C	R/W	时钟源选择控制寄存器 2	0x0000_00EF

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				FRQDIV_S		保留	

位	描述	
[31:4]	保留	保留
[3:2]	FRQDIV_S[1:0]	时钟分频器时钟源选择 00 = 外部晶振HXT或LXT 01 = 保留 10 = HCLK 11 = HIRC 注: 设置PWRCON[1:0], 用于选择HXT 或LXT晶振时钟
[1:0]	保留	保留

时钟分频寄存器 (CLKDIV)

寄存器	偏移量	R/W	描述	复位值
CLKDIV	CLK_BA+0x18	R/W	时钟分频寄存器	0x0000_0000

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

位	描述	
[31:24]	保留	保留
[23:16]	ADC_N[7:0]	ADC外设时钟分频因子 ADC时钟频率 = (ADC时钟源频率) / (ADC_N + 1)
[15:12]	保留	保留
[11:8]	UART_N[3:0]	UART 时钟分频因子 UART时钟频率 = (UART时钟源频率) / (UART_N + 1)
[7:4]	保留	保留
[3:0]	HCLK_N[3:0]	HCLK 时钟分频因子 HCLK 时钟频率 = (HCLK时钟源频率) / (HCLK_N + 1)

频率分频器控制寄存器 (FRQDIV)

寄存器	偏移量	R/W	描述	复位值
FRQDIV	CLK_BA+0x24	R/W	频率分频器控制寄存器	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		DIVIDER1	DIVIDER_EN	FSEL			

位	描述	
[31:6]	保留	保留
[5]	DIVIDER1	<p>频率分频器1使能控制 0 = 分频器输出频率由FSEL的值决定 1 = 分频器输出频率与输入频率相同</p>
[4]	DIVIDER_EN	<p>频率分频器使能控制 0 = 禁用频率分频 1 = 使能频率分频</p>
[3:0]	FSEL[3:0]	<p>分频器输出频率选择位 输出频率的公式是 $F_{out} = F_{in}/2^{(N+1)}$, F_{in} 为输入时钟频率 F_{out} 为分频器输出时钟频率 N 为FSEL[3:0]的值。</p>

5.5 内存控制器 (FMC)

5.5.1 概述

NuMicro® NUC029系列具有64K/32K/16K字节的片上FLASH (APROM)，用于存储应用程序，可以通过ISP处理更新。系统内编程 (ISP) 允许用户更新焊接在PCB板上的芯片中的程序。上电后，通过设置CONFIG0的启动选择位 (CBS) 决定Cortex®-M0 CPU从APROM还是LDROM读取代码。

此外，NuMicro® NUC029为用户提供额外的4k字节的数据FLASH，以供用户在系统掉电之前存储数据。NUC029xAN 提供额外的4K字节的数据FLASH，而NUC029FAE提供的DATAFLASH是与APROM共享空间，它的起始地址可由用户在CONFIG1中配置。

5.5.2 特性

- 高达50MHz的零等待连续地址读访问
- 64/32/16KB 应用程序存储空间(APROM)
- 4KB 系统内编程 (ISP) 空间(LDROM)
- NUC029xAN，固定的4kB数据FLASH
- NUC029FAE，数据FLASH大小及其起始地址可配置
- 所有内部Flash页擦除单位为512字
- 支持应用内编程(IAP)，可在APROM和LDROM间切换程序且无须重启
- 支持系统内编程(ISP)，可更新片上FLASH

5.5.3 NUC029xAN FMC 框图

FLASH存储器控制器由AHB从接口, ISP控制逻辑, 烧写器接口和FLASH宏接口时序控制逻辑组成。FLASH存储器控制器框图如下所示:

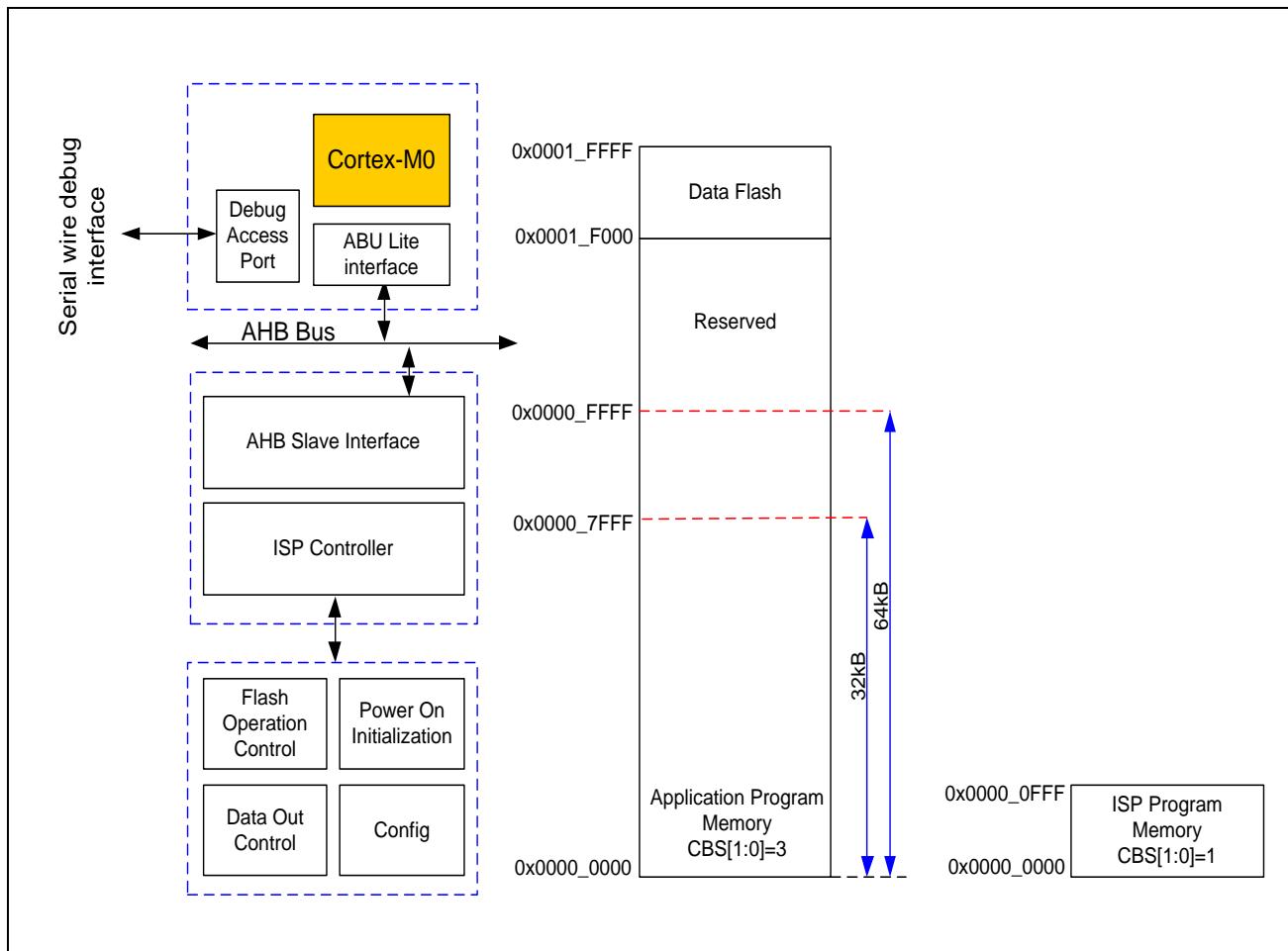


图 5-18 NUC029xAN Flash存储器控制器框图

5.5.4

NUC029FAE FMC 框图

FLASH存储器控制器由AHB从接口, ISP控制逻辑, 烧写器接口和FLASH宏接口时序控制逻辑组成。FLASH存储器控制器框图如下所示:

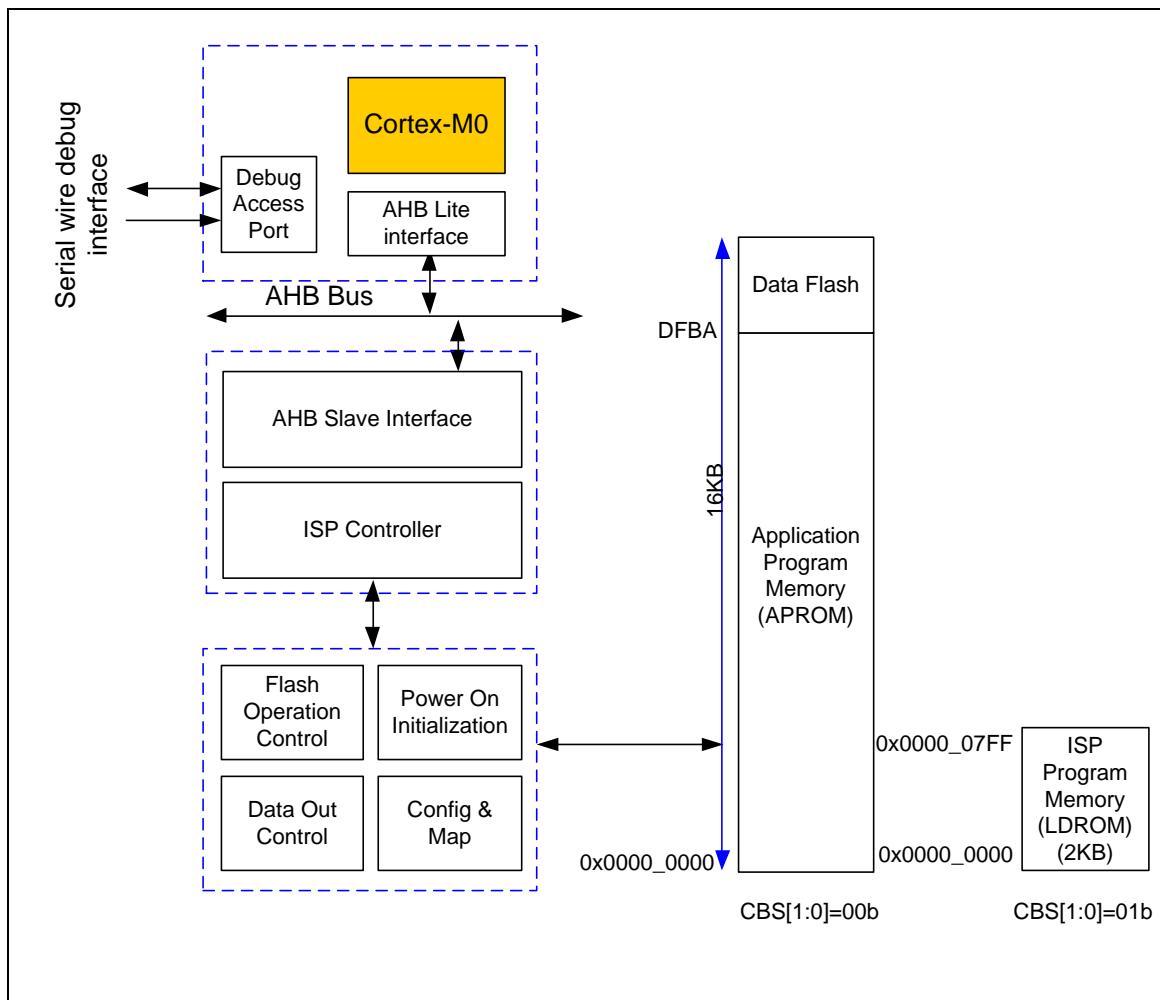


图 5-19 NUC029FAE Flash存储器控制器框图

5.5.5 FMC 组织结构

NUC029 的flash存储器由程序存储器(APROM)，数据FLASH，ISP加载程序存储器(LDROM)，和用户配置区共4块组成。

用户程序主要放在APROM中，用户将程序烧入APROM之后，设定系统从APROM启动就可以执行APROM中的程序了。

ISP启动程序空间LDROM用来实现系统内编程。LDROM空间和APROM空间是独立的，系统可以选择从LDROM启动，这样用户可避免当APROM中程序被破坏时，系统启动失败。

数据Flash用来存储数据。可以通过ISP读或者直接内存读取，亦可通过ISP寄存器进行编程。数据Flash的擦除单元为512字节。NUC029xAN中数据FLASH的大小为4KB，其起始地址固定为0x0001_F000；NUC029FAE中数据FLASH的空间是与APROM共享的，其大小和起始地址在CONFIG1中配置。

用户配置区提供几个字节来控制系统逻辑，如flash安全加密，启动选择，欠压电平，数据FLASH起始地址等。用户配置的作用类似保险丝，在上电期间，做为缺省配置，从FLASH存储器中加载到相应的控制寄存器中。

NuMicro®家族中，Flash内存映射和系统内存映射是不同的。Flash内存映射是当用户使用ISP命令读、写和擦除的时候使用。系统内存映射是CPU访问flash的时候使用，就是说系统地址是给CPU看的，Flash地址是给FMC看的。举个例子，NUC029xAN中，CPU读代码或者数据的时候。如果CBS[1:0] = 1，系统选择从LDROM启动，CPU可以从系统地址0x0~0xFFFF(LDROM)读取代码，但是如果用户想使用ISP命令从LDROM读取数据的时候，地址应该为0x0010_0000 ~ 0x0010_0FFF。

下表和图显示了APROM、LDROM、数据Flash和用户配置区Flash内存映射地址

区块名称	大小	开始地址	结束地址
AP-ROM	32/64 Kbytes	0x0000_0000	0x0000_7FFF (32 Kbytes) 0x0000_FFFF (64 Kbytes)
Data Flash	4 Kbytes	0x0001_F000	0x0001_FFFF
LD-ROM	4 Kbytes	0x0010_0000	0x0010_0FFF
User Configuration	1 Words	0x0030_0000	0x0030_0000

表 5-10 Flash存储器地址映射 NUC029xAN

区块名称	DFEN	大小	开始地址	结束地址
AP-ROM	0	(16-0.5*N) Kbytes	0x0000_0000	DFBA-1
AP-ROM	1	16 Kbytes	0x0000_0000	0x0000_3FFF
Data Flash	0	0.5*N Kbytes	DFBA	0x0000_3FFF
Data Flash	1	0 Kbytes	N/A	N/A
LD-ROM	x	2 Kbytes	0x0010_0000	0x0010_07FF
User Configuration	x	2 Words	0x0030_0000	0x0030_0007

表 5-11 Flash存储器地址映射 NUC029FAE

存储器组织结构如下所示:

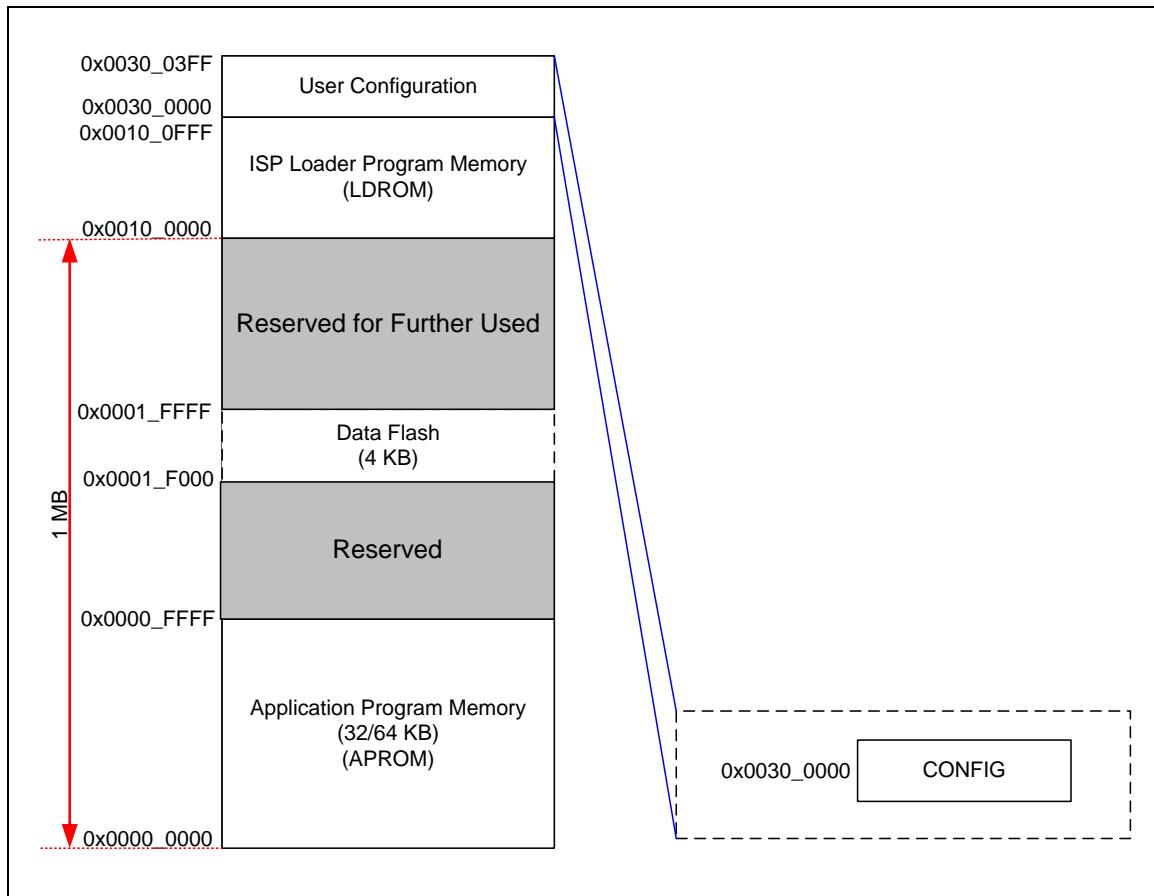


图 5-20 Flash 存储器组织结构 (NUC029xAN)

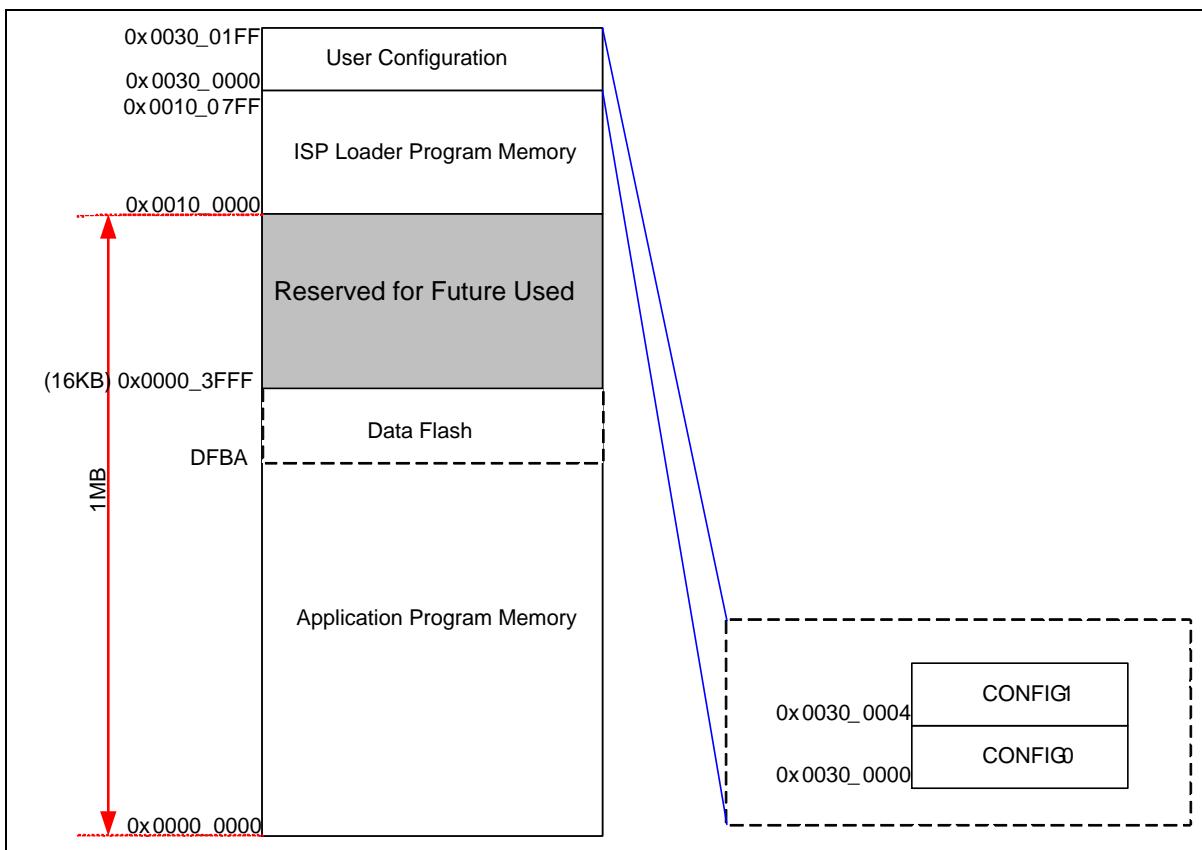


图 5-21 Flash存储器组织结构 (NUC029FAE)

5.5.6 数据 Flash

NUC029xAN:

NUC029xAN为用户提供数据FLASH用于存储数据。它可通过ISP寄存器进行读/写。擦除单位为512字节。若要改变一个字，需要先把所有128个字拷贝到另外页或SRAM中。对于32/64KB的flash设备，数据FLASH的大小为4KB，开始地址固定在0x0001_F000。

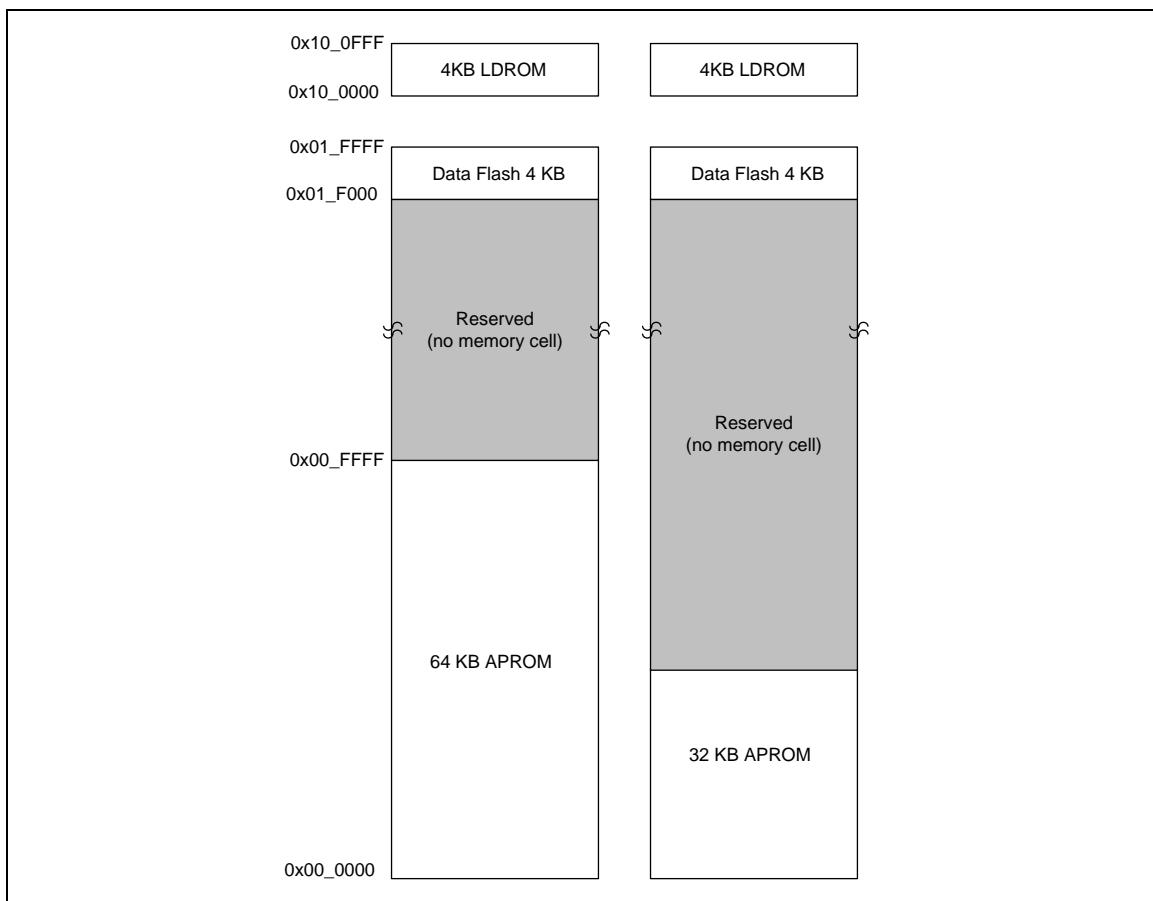


图 5-22 Flash 存储器结构 (NUC029xAN)

NUC029FAE:

NUC029FAE 为用户提供数据 FLASH 用于存储数据。它可通过 ISP 寄存器进行读/写。如果 DFEN(CONFIG0[0]) 为使能的，数据 FLASH 的起始地址则由 DFBA 设定，例如，对于 4K/2K/1K/0KB 数据 Flash，DFBA 设定值如下表中所列。擦除单位为 512 字节。若要改变一个字，需要先把所有 128 个字拷贝到另外页或 SRAM 中。

Data Flash	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
16K Flash	DFBA=0x0000_3000	DFBA=0x0000_3800	DFBA=0x0000_3C00	DFEN=1

表 5-12 数据 Flash 表 (NUC029FAE)

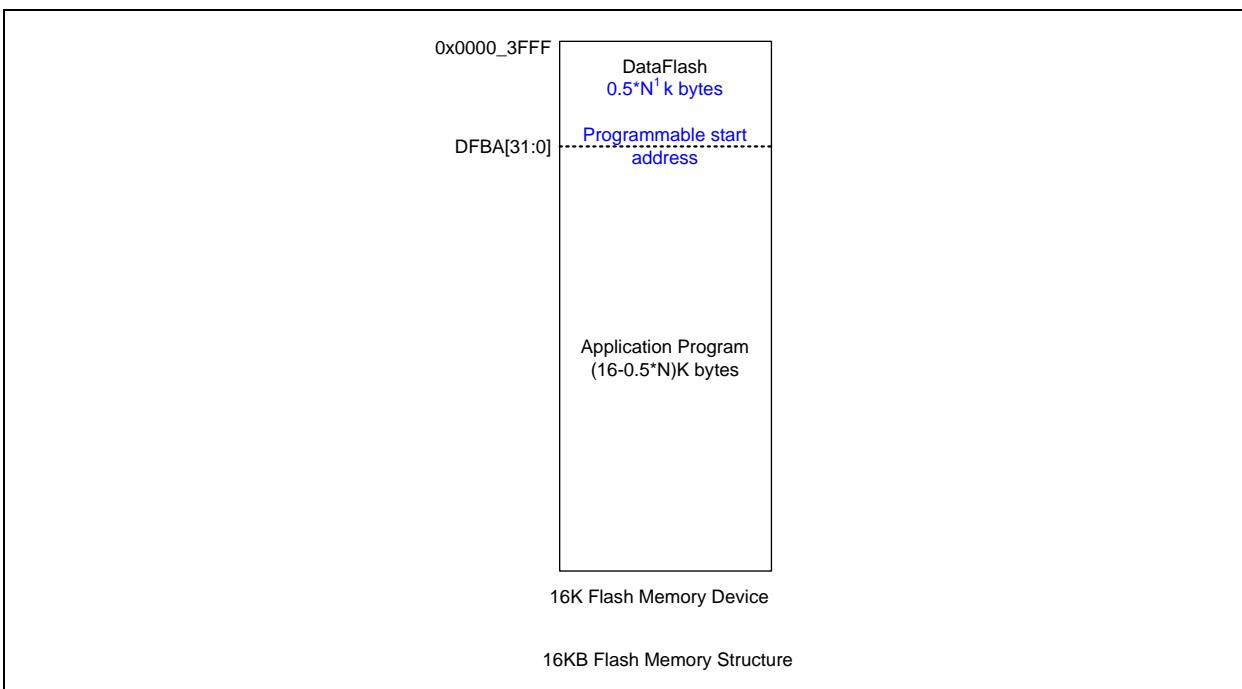


图 5-23 Flash 存储器结构 (NUC029FAE)

5.5.7 用户配置区

5.5.7.1 启动选择

用户配置区是内部可编程的配置区用于启动选择。用户配置区在Flash内存的地址0x300000，总共32个位。用户配置区的任何更改将在系统重启后生效。

NUC029xAN:

CONFIG0 (地址= 0x0030_0000)

31	30	29	28	27	26	25	24
CWDTEN	CWDTPDEN	保留			CFOSC		
23	22	21	20	19	18	17	16
CBODEN	CBOV		CBORST	保留			
15	14	13	12	11	10	9	8
保留				CIOINI	保留		
7	6	5	4	3	2	1	0
CBS		保留				LOCK	保留

BIT	描述											
[31]	CWDTEN	看门狗使能控制 0 = 芯片上电后使能看门狗定时器，并且强制其时钟源为内部10K振荡器。 1 = 上电时看门狗默认关闭.										
[30]	CWDTPDEN	看门狗时钟睡眠使能控制 0 = 看门狗的10K时钟源总是强制使能的，软件无法关闭. 1 = 当芯片进入睡眠模式，看门狗的10K时钟源由OSC10K_EN (PWRCON[3])控制 注：该位只有在CWDTEN = 0时才工作										
[31:27]	保留	保留										
[26:24]	CFOSC	复位后CPU 时钟源选择 <table border="1"> <tr> <td>CFOSC[2:0]</td> <td>时钟源</td> </tr> <tr> <td>000</td> <td>外部晶振时钟 (4 ~ 24MHz)</td> </tr> <tr> <td>111</td> <td>内部 RC 22.1184 MHz 振荡器时钟</td> </tr> <tr> <td>其他</td> <td>保留</td> </tr> </table> 复位发生后， CFOSC 的值将被加载到CLKSEL0.HCLK_S[2:0].	CFOSC[2:0]	时钟源	000	外部晶振时钟 (4 ~ 24MHz)	111	内部 RC 22.1184 MHz 振荡器时钟	其他	保留		
CFOSC[2:0]	时钟源											
000	外部晶振时钟 (4 ~ 24MHz)											
111	内部 RC 22.1184 MHz 振荡器时钟											
其他	保留											
[23]	CBODEN	欠压检测使能 0 = 上电后使能欠压检测 1 = 上电后禁用欠压检测										
[22:21]	CBOV	欠压电压选择 <table border="1"> <tr> <td>CBOV</td> <td>Brown-out Voltage</td> </tr> <tr> <td>11</td> <td>4.4V</td> </tr> <tr> <td>10</td> <td>3.7V</td> </tr> <tr> <td>01</td> <td>2.7V</td> </tr> <tr> <td>00</td> <td>2.2V</td> </tr> </table>	CBOV	Brown-out Voltage	11	4.4V	10	3.7V	01	2.7V	00	2.2V
CBOV	Brown-out Voltage											
11	4.4V											
10	3.7V											
01	2.7V											
00	2.2V											
[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
[19:11]	保留	保留
[10]	CIOINI	I/O 口初始状态选择 0 = 上电后所有 GPIO 默认为输入三态模式. 1 = 上电后所有GPIO 默认为准双向. 注: 建议在管脚ICE_DAT和ICE_CLK使用100 kΩ上拉电阻。
[9:8]	保留	保留
[7:6]	CBS	芯片启动选择 00 = 芯片从LDROM启动, IAP功能使能 01 = 芯片从LDROM启动, 没有IAP功能 10 = 芯片从APROM启动, IAP功能使能 11 = 芯片从APROM启动, 没有IAP功能 NUC029xAN , 用户可以设定 CBS[0] = 0 以支持 IAP 功能。当 CBS[0] = 0 时, LDROM 映射到系统地址 0x100000 , APROM 映射到系统地址 0x0 。不需要重新启动, CPU 可以访问 LDROM 和 APROM 的所有空间, 换句话说, 如果 IAP 功能使能, LDROM 和 APROM 中的函数可以互相调用。 注1: ISPCON 的 BS 位只有在 CBS[0] = 1 时才能用来控制启动切换 注2: 当CBS[0] = 0是, VECMAP 命令只可以将某个地址映射到0x0~0x1ff
[5:2]	保留	保留
[1]	LOCK	安全锁 0 = Flash 数据锁定 1 = Flash 数据不锁定. 当锁定了flash数据时, 仅有设备ID、唯一ID和用户配置区可以通过烧录器和ICP通过串行调试接口读出。其他数据锁定在0xFFFFFFFF。ISP 可以不管LOCK是否锁定都能读出数据。 为了解锁系统, 用户可以使用 ISP 命令关闭 LOCK 位或者通过 ICP 工具擦除整个芯片 (芯片擦除)
[0]	保留	保留

注: 用户区保留位应该维持为'1'

NUC029FAE:

CONFIG0 (地址= 0x0030_0000)

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
CBOVEXT	CBOV		CBORST	保留			
15	14	13	12	11	10	9	8
保留				CIOINI	保留		
7	6	5	4	3	2	1	0
CBS		保留				LOCK	DFEN

Config0	地址= 0x0030_0000																							
位	描述																							
[31:24]	保留	保留																						
[23]	CBOVEXT	欠压选择扩展 0 = 欠压选择包括2.2V, 2.7V, 3.7V 和 4.4V 1 = 欠压选择包括BOD禁止模式。如果要禁止BOD功能, CBOV[1:0]必须设置为11b 参看CBOV中的表																						
[22:21]	CBOV[1:0]	欠压电压选择 <table border="1" style="margin-left: 20px;"> <tr> <th>CBOVEXT</th> <th>CBOV[1:0]</th> <th>掉电检测电压</th> </tr> <tr><td>0</td><td>00</td><td>2.2V</td></tr> <tr><td>0</td><td>01</td><td>2.7V</td></tr> <tr><td>0</td><td>10</td><td>3.7V</td></tr> <tr><td>0</td><td>11</td><td>4.4V</td></tr> <tr><td>1</td><td>11</td><td>关闭 BOD 功能</td></tr> </table>					CBOVEXT	CBOV[1:0]	掉电检测电压	0	00	2.2V	0	01	2.7V	0	10	3.7V	0	11	4.4V	1	11	关闭 BOD 功能
CBOVEXT	CBOV[1:0]	掉电检测电压																						
0	00	2.2V																						
0	01	2.7V																						
0	10	3.7V																						
0	11	4.4V																						
1	11	关闭 BOD 功能																						
[20]	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																							
[19:11]	保留																							
[10]	I/O 口初始状态选择 0 = 上电后所有GPIO 默认为准双向. 1 = 上电后所有 GPIO 默认为输入三态模式.																							
[9:8]	保留																							
[7:6]	芯片启动选择 00 = 芯片从LDROM启动, IAP功能使能 01 = 芯片从LDROM启动, 没有IAP功能																							

Config0	地址= 0x0030_0000	
位	描述	
		<p>10 = 芯片从APROM启动, IAP功能使能 11 = 芯片从APROM启动, 没有IAP功能 NUC029FAE, 用户可以设定 CBS[0] = 0 以支持 IAP 功能。当 CBS[0] = 0时, LDROM 映射到系统地址 0x100000 , APROM 映射到系统地址0x0。不需要重新启动, CPU 可以访问 LDROM 和 APROM 的所有空间, 换句话说, 如果 IAP 功能使能, LDROM 和 APROM 中的函数可以互相调用。 注1: ISP CON 的 BS 位只有在 CBS[0] = 1时才能用来控制启动切换, CBS[0]为该寄存器的第6位。 注2: 当CBS[0] = 0是, VECMAP 只可以将APROM 或 LDROM的page 0映射到0x0~0x1ff</p>
[5:2]	保留	保留
[1]	LOCK	<p>安全锁 0 = Flash 数据锁定 1 = Flash 数据不锁定. 当锁定了flash数据时, 仅有设备ID、唯一ID和用户配置区 可以通过烧录器和ICP通过串行调试接口读出。其他数据锁定在0xFFFFFFFF。ISP 可以不管LOCK是否锁定都能读出数据. 为了解锁系统, 用户可以使用ISP命令关闭LOCK位或者通过ICP工具擦除整个芯片 (芯片擦除)</p>
[0]	DFEN	<p>数据FLASH使能控制 0 = 使能数据FLASH. 1 = 禁止数据Flash</p>

注: 用户区保留位应该维持为'1'

CONFIG1 (地址 = 0x0030_0004)

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		DFBA					
7	6	5	4	3	2	1	0
DFBA							

Config1	地址= 0x0030_0004	
位	描述	
[31:14]	保留	保留
[13:0]	DFBA[13:0]	数据FLASH起始地址 数据FLASH起始地址可由用户设定。因为芯片擦除单元为512字节，因此该设定中8-0位必须强制为0

5.5.7.2 欠压检测

NUC029系列都有欠压检测功能用于监控V_{DD}引脚的电压。如果V_{DD}电压低于 CBOV (CBOVEXT 仅 NUC029FAE 适用) 的设定值，如果 BOD 功能使能，BOD 事件将被触发。用户可以通过使能 CBORST 决定欠压时 BOD 复位系统还是发生 BOD 中断。因为不论何时只要 V_{DD} 电压低于 CBOV 设定，BOD 复位将马上发生，所以用户必须保证 CBOV 设定正确，以避免 BOD 一直产生复位。例如：如果 V_{DD} 电压是 3.3V, CBOV(CBOVEXT 仅 NUC029FAE 适用) 只能是 00'b or 01'b，否则，如果 BOD 复位使能，CBOV 为 10'b 或 11'b，系统将被停留在 BOD 复位状态。

5.5.8 启动选择

NUC029 系列提供系统内编程 (ISP) 特性，允许用户直接更新 PCB 板上芯片中的程序。提供 2K/4K 字节 LDROM 程序存储区专门用于存储 ISP 固件。用户可通过设置 CBS[1](CONFIG0[7]) 以选择从 APROM 启动还是 LDROM 启动。

除了设定从 APROM 还是 LDROM 启动，CBS(CONFIG0[7:6]) 还用来控制启动后系统内存映射。CBS[1](CONFIG0[7]) 中的值还会在启动之后载入到 BS(ISPCON[1]) 中。



图5-24 上电后启动选择 (BS)

当 CBS[0] = 1，并且 CBS[1] = 1 时，系统从APROM启动，APROM中的应用程序不能通过CPU读直接访问LDROM。同样，当CBS[0] = 1 而CBS[1] = 0时，系统从LDROM启动，LDROM中的程序也不能通过CPU读访问APROM中的程序。下图显示了APROM启动和LDROM启动时的系统内存映射。

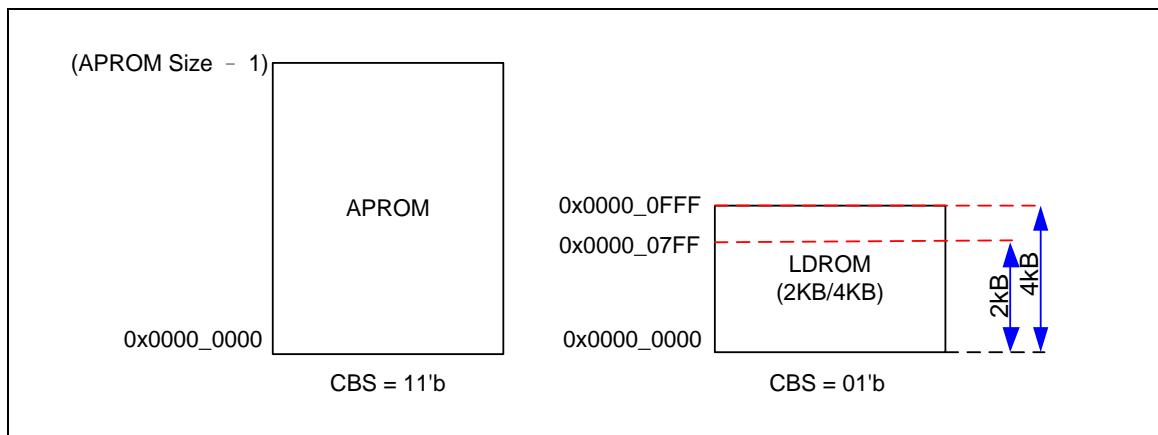


图5-25 从APROM和LDROM启动时程序的执行范围

如果应用程序在APROM中执行需要调用LDROM中的函数或者程序在LDROM中执行需要调用APROM中的函数，CBS[0] 需要设为 0，这被称为应用内编程(IAP)。

CBS[1:0]	启动选择
00	LDROM 启动支持IAP 功能 (NUC029xAN) 芯片从LDROM启动，程序执行范围包括LDROM 和大部分 APROM (除了第一页512个字节被LDROM映射)。

	<p>LDROM 地址范围0x0010_0000 ~ 0x0010_0FFF，同时第一页512字节被映射到0x0000_0000 ~ 0x0000_01FF. 通过ISP命令，地址0x0000_0000 ~ 0x0000_01FF 可以被重新映射到可执行范围的任何地方.</p>
	<p>LDROM 启动支持IAP 功能 (NUC029FAE) 芯片从LDROM启动，程序执行范围包括LDROM 和大部分 APROM (除了第一页512个 字节被LDROM映射). LDROM 地址范围0x0010_0000 ~ 0x0010_07FF，同时第一页512字节被映射到0x0000_0000 ~ 0x0000_01FF. 通过ISP命令，地址0x0000_0000 ~ 0x0000_01FF 可以被重新映射到可执行范围的任何地方. 在这种模式下，不管程序运行在LDROM或APROM中，APROM和LDROM都是可编程的，而数据FLASH则没有任何意义，因为 APROM和LDROM的任何一个区域都可以当做数据FLASH使用，DFBA在这种模式下没有用处。</p>
01	<p>LDROM 启动不支持 IAP 功能 芯片从LDROM启动，程序执行范围为 LDROM，APROM 中的程序不能直接访问必须通过ISP命令. 这种模式下，LDROM 是写保护的.</p>
10	<p>APROM 启动支持IAP功能(NUC029xAN) 芯片从APROM启动，程序执行范围包括LDROM 和 APROM LDROM地址范围0x0010_0000~0x0010_0FFF 通过ISP命令，地址0x0000_0000 ~ 0x0000_01FF可以被重新映射到可执行范围的任何地方.</p> <p>APROM 启动支持IAP功能 (NUC029FAE) 芯片从APROM启动，程序执行范围包括LDROM 和 APROM LDROM地址范围0x0010_0000~0x0010_07FF 通过ISP命令，地址0x0000_0000 ~ 0x0000_01FF可以被重新映射到可执行范围的任何地方. 在这种模式下，不管程序运行在LDROM或APROM中，APROM和LDROM都是可编程的，而数据FLASH则没有任何意义，因为 APROM和LDROM的任何一个区域都可以当做数据FLASH使用，DFBA在这种模式下没有用处。</p>
11	<p>APROM 启动不支持IAP功能 芯片从APROM 启动，程序执行范围为 APROM， LDROM中的程序不能直接访问必须通过ISP命令. 这种模式下，APROM是写保护的.</p>

表 5-13 启动选项

5.5.9 应用内编程

NUC029系列提供应用内编程(IAP)功能，运行代码可以在APROM和LDROM之间切换而不用复位系统。用户通过设定芯片启动选项CBS[1:0](CONFIG0[7:6]) 为10'b 或者 00'b，然后重新复位芯片可以使能IAP功能。

芯片从APROM 启动，支持 IAP 功能时(CBS[1:0] = 10'b)，执行范围包含所有的APROM 和 LDROM. APROM 系统地址空间保持原有大小，NUC029xAN中的 4 KB LDROM址空间则映射 0x0010_0000~ 0x0010_0FFF，而NUC029FAE的2 KB LDROM则映射到0x0010_0000~ 0x0010_07FF。

芯片从 LDROM启动，支持IAP功能时(CBS[1:0] = 00'b)， 执行范围包含所有的LDROM和大部分 APROM(除了第一页)。用户不能访问APROM的第一页，因为默认情况下，可执行范围的第一页映射为LDROM的第一页了。同时NUC029xAN 中LDROM的4KB地址空间仍映射在 0x0010_0000~0x0010_0FFF，NUC029FAE中LDROM的2KB地址空间仍映射在 0x0010_0000~0x0010_07FF。LDROM的第一页映射到0x0000_0000~0x0000_01FF。

IAP地址映射请参考下图

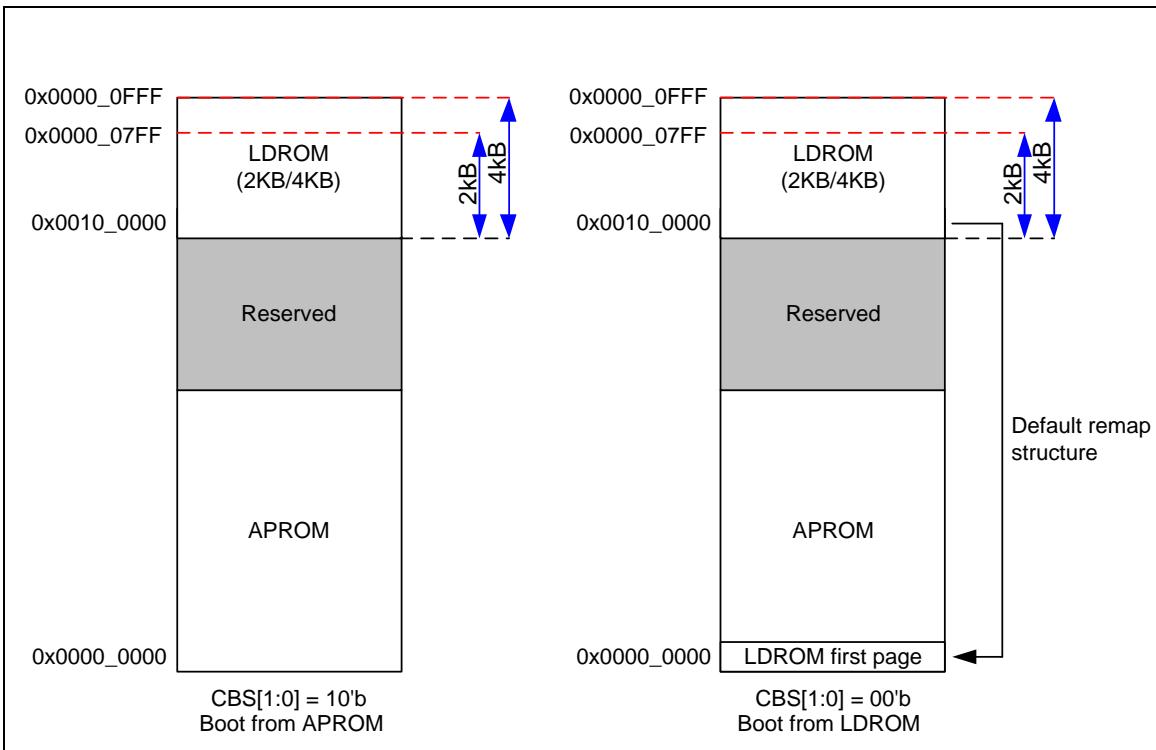


图 5-26 IAP功能使能时可执行地址范围

当芯片带 IAP 功能启动时，可执行范围中的任何页在任何时间都可以映射到第一页(0x0000_0000~0x0000_01FF). 用户可以通过ISP向量页重新映射命令并将要映射的地址填到 ISPADR 寄存器就可以改变第一页的映射地址。通过读 ISPSTA 寄存器的 VECMAP (ISPSTA[20:9]) 域，用户可以检查映射是否成功。

5.5.10 系统内编程 (ISP)

NUC029系列支持系统内编程，允许设备在软件的控制下重新升级程序，以避免在下载或编程失败时导致系统失败。并且，这种升级应用软件能力使得应用范围扩大成为可能。

为了支持系统内编程，NUC029系列包含了 LDROM 和 ISP控制器。用户可以将他们自己的ISP加载固件放在LDROM中，该加载程序通过ISP命令可以编程用户程序(APROM)。换句话说，就是固件提供更新用户程序的功能。ISP固件使用各种硬件外设接口可以很容易的接收到新的程序代码。ISP最常用的方式是通过UART。一般来说，PC通过串口传输新的APROM程序，ISP固件收到程序以后将其更新到APROM中

5.5.11 寄存器控制过程

NUC029系列芯片支持由用户配置区(CBS)定义从APROM还是LDROM启动。用户配置区改变之后

需要重新复位系统才能生效。CBS[0] = 1时，如果用户不想改变用户配置区就想要在APROM和LDROM之间切换，这就需要修改BS(ISPCON[1])位，然后复位 CPU_RST(IPRSTC1[1])（不会复位I/O和外设）或 SYSRESETREQ(AIRCR[2])（复位I/O和外设）。修改BS位切换启动选项如下图所示。通过BS位切换只有在CBS[0] = 1时才有效。

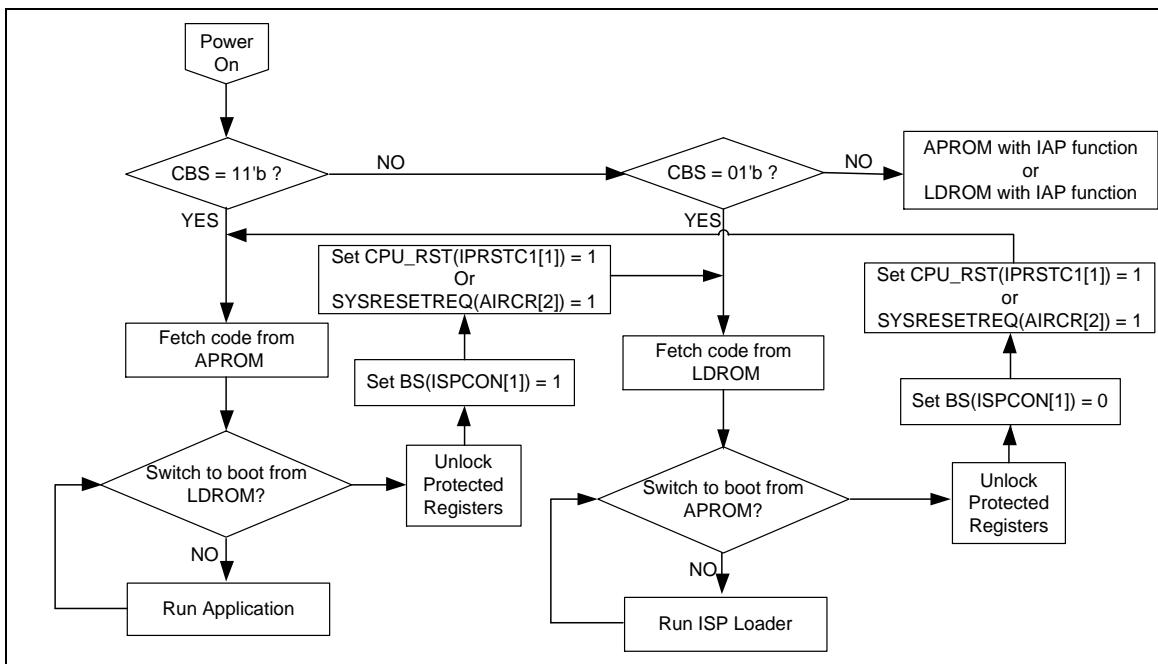


图 5-27 CBS[0] = 1 时通过BS位改变启动选项范例流程

软件运行在LDROM中时更新APROM或者运行在APROM中时更新LDROM可以避免更新失败后导致系统失败。

ISP控制器支持读、写、擦除内嵌的Flash。ISP控制器中的几个控制位是写保护的，因而在修改它们之前，软件需要向REGWRPROT寄存器依次写入0x59, 0x16 和 0x88。如果解锁成功，REGWRPROT的值将为1，解锁序列一定不能被打断，否则解锁将失败。

解锁保护寄存器后，用户需要设置ISPCON控制寄存器来决定更新LDROM，用户配置区，APROM和使能ISP控制器。

一旦ISPCON寄存器设置正确，用户可以设置ISPCMD寄存器决定要擦除、读还是写。设置ISPADR为目标FLASH内存地址(基于Flash内存组织结构)，ISPDAT用来设定要写或者返回读到(根据ISPCMD)的数据。

最后设定ISPGO(ISPTRG[0])位启动ISP操作。ISP操作完成之后，ISPGO(ISPTRG[0])将自动清0。为了确保ISP功能正确完成，在CPU继续执行指令之前，ISPGO之后需要添加ISB指令。

启动ISP_{GO}之后，需要检查几个错误条件。如果错误条件产生，ISP操作失败，其失败标志将会置位。ISPFF(ISPSTA[6])标志只能由软件清零。即使ISPFF保持为“1”，下一次ISP寄存器操作仍然可以启动。因此，建议在每次ISP操作后，检查ISPFF(ISPSTA[6])位，如果该位被置为1，就将其清零。

当ISPGO置位时，CPU将等待ISP操作结束，在此期间，外设仍然正常工作，如果有中断请求时，CPU仍然会先执行完ISP后再响应中断。当ISP操作完成之后，ISPGO由硬件自动清0。用户可以通过ISPGO位来检查是否ISP操作已经完成。用户应该在ISPGO置1后添加ISB指令，以确保指令正

确执行。

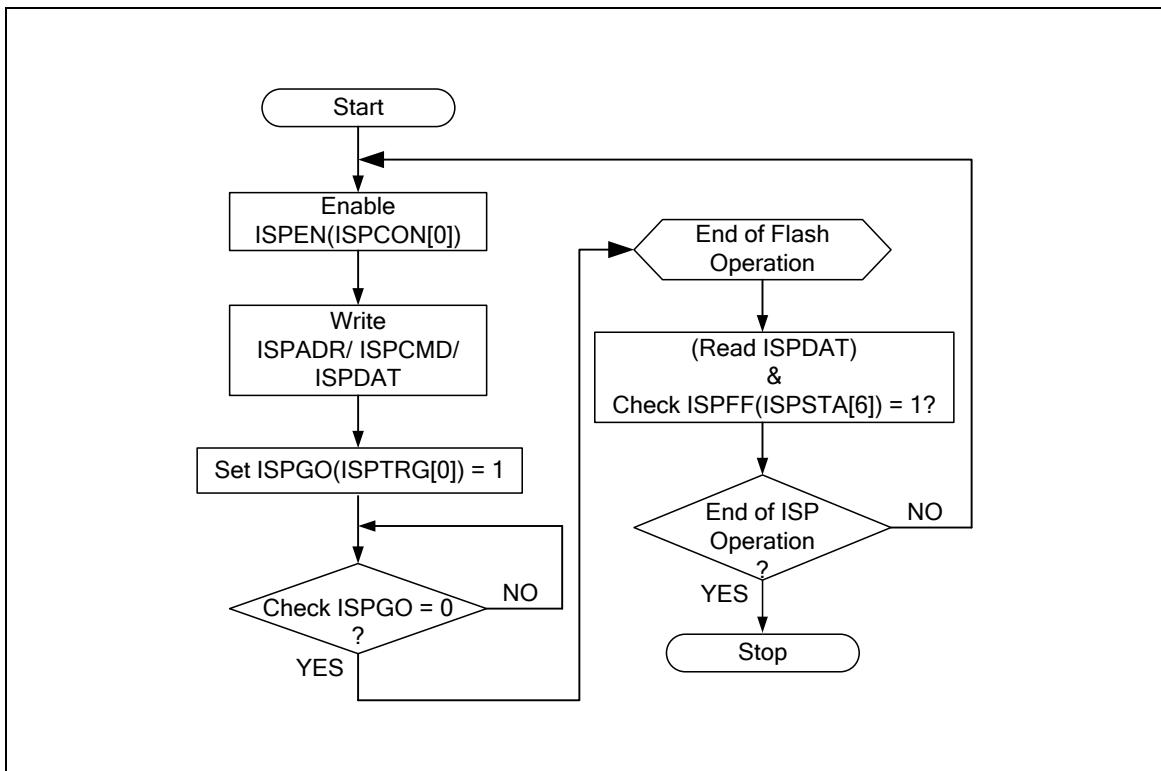


图 5-28 ISP 流程范例

下表列出了NUC029支持的ISP命令

ISP 命令	ISPCMD	ISPADR	ISPDAT
FLASH Page Erase	0x22	Flash内存有效地址 必须是 512 字节对齐	无关
FLASH Program	0x21	Flash内存有效地址	编程数据
FLASH Read	0x00	Flash内存有效地址	返回数据
Read Unique ID	0x04	0x0000_0000	Unique ID Word 0
		0x0000_0004	Unique ID Word 1
		0x0000_0008	Unique ID Word 2
		0x0000_0014	Unique CID Word 1 (仅NUC029FAE)
		0x0000_0018	Unique CID Word 2 (仅NUC029FAE)
		0x0000_001C	Unique CID Word 3 (仅NUC029FAE)
Read Company ID	0x0B	无关	Company ID (0xDA)
Vector Page Re-Map	0x2E	APROM 或者 LDROM中的某个页 必须是 512 字节对齐	无关

表 5-14 ISP Command List ISP 命令列表

5.5.12 通过向量重新映射多启动

NUC029系列通过向量重新映射支持从不同的地址启动。当 **CBS[0] = 0**, LDROM中所有页和 APROM中所有页都可以映射到地址 0x0。重新映射的地址可以从ISPSTA寄存器的 **VECMAP(ISPSTA[20:9])** 中读到。当**CBS[1:0] = 10'b**时，上电时默认映射地址为0。这就意味着上电时从APROM启动，向量页映射到APROM的第一页。当**CBS[1:0] = 00'b**时，映射地址为 0x10_0000。这就意味着向量页映射到LDROM的第一页，上电从LDROM启动。

重新映射地址可以通过向量页重新映射命令改变。通过向量页重新映射命令用户可以重新映射指定页为向量页，然后使用**CPU_RST(IPRSTC1[1])**（不复位I/O和外设）或者 **SYSRESETREQ(AIRCR[2])**（复位I/O和外设）来复位系统。**CPU**将从新的向量表取得堆栈指针和复位处理函数指针。

例如，如果用户有2个独立的应用程序都在APROM里面，分别命名为App0和App1。App0放在地址0，App1放在地址0x2000。**CBS[1:0] = 00'b**从LDROM启动。上电时系统将执行LDROM中的代码。LDROM中的code将决定从App0启动还是从App1启动。如果从App0启动，LDROM中的代码将负责使能ISP并重新映射向量页到地址0，然后通过**CPU_RST(IPRSTC1[1])**（不复位I/O 和外设）或者**SYSRESETREQ(AIRCR[2])**（复位 I/O 和外设）来复位CPU 从App0启动；如果从App1启动，中的代码将负责使能ISP并重新映射向量页到地址0x2000，然后通过**CPU_RST(IPRSTC1[1])**（不

复位 I/O 和外设) 或者SYSRESETREQ(AIRCR[2]) (复位 I/O 和外设) 来复位CPU 从App1启动。下表显示怎样通过向量表映射从不同的应用程序启动。

为了重新映射向量表，用户需要设置新的页地址到ISPADR, 填重新映射命令0x2E 到ISPCMD寄存器，然后通过将ISPGO=1触发ISP。用户通过读回VECMAP(ISPSTA[20:9])来确认新的向量表地址。

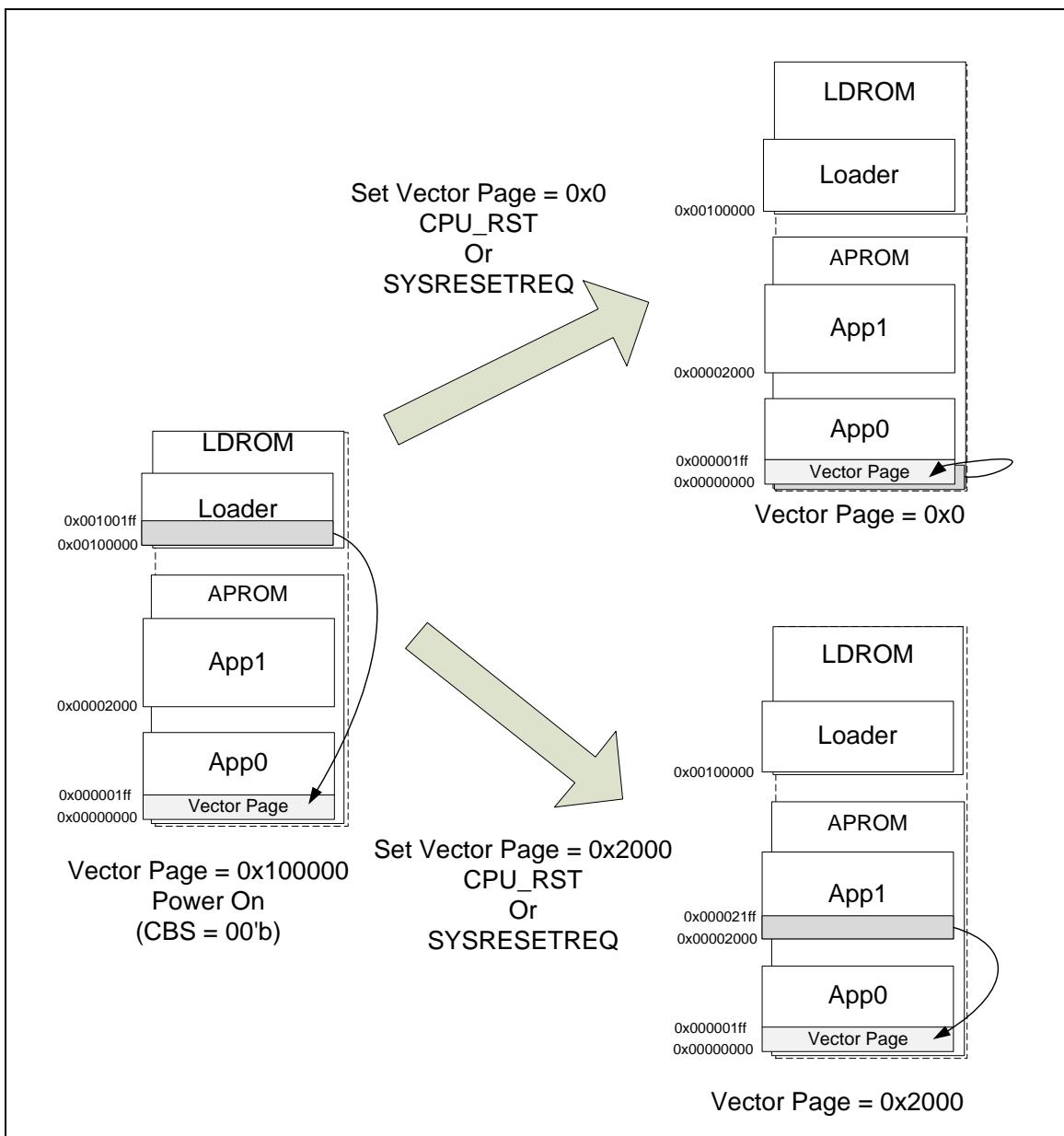


图5-29 通过向量重新映射多引导

5.5.13 寄存器映射 (NUC029xAN)

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 基地址:				
FMC_BA = 0x5000_C000				
ISPCON	FMC_BA+0x00	R/W	ISP控制寄存器	0x0000_0000
ISPADR	FMC_BA+0x04	R/W	ISP地址寄存器	0x0000_0000
ISPDAT	FMC_BA+0x08	R/W	ISP数据寄存器	0x0000_0000
ISPCMD	FMC_BA+0x0C	R/W	ISP命令寄存器	0x0000_0000
ISPTRG	FMC_BA+0x10	R/W	ISP触发寄存器	0x0000_0000
DFBADR	FMC_BA+0x14	R	数据Flash 起始地址	0x0001_F000
FATCON	FMC_BA+0x18	R/W	FLASH访问时间控制寄存器	0x0000_0000
ISPSTA	FMC_BA+0x40	R/W	ISP状态寄存器	0x0000_0000

5.5.14 寄存器描述 (NUC029xAN)

ISP控制寄存器 (ISPCON)

寄存器	偏移量	R/W	描述	复位值
ISPCON	FMC_BA+0x00	R/W	ISP 控制寄存器	0x0000_0000

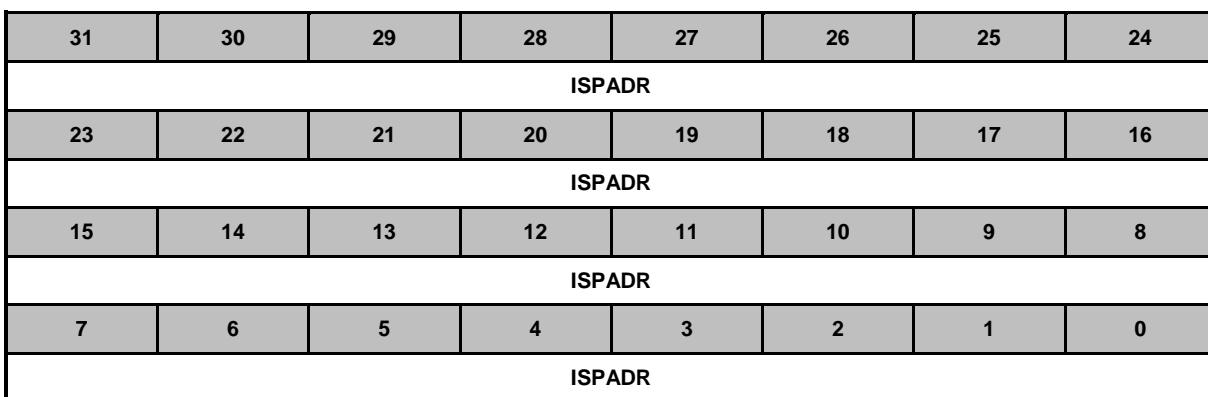
31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
保留							
7	6	5	4	3	2	1	0
保留	ISPFF	LDUEN	CFGUEN	APUEN	保留	BS	ISPEN

位	描述	
[31:7]	保留	保留
[6]	ISPFF	ISP失败标志 (写保护) 当ISP满足下列条件时，该位由硬件置位： (1) APUEN =0时，APROM对自身写入。 (2) LDROM对自身写入。 (3) 目标地址无效，如超过正常范围。 注：该位写1清零。
[5]	LDUEN	LDROM更新使能控制 (写保护) 0 = LDROM不能被更新 1 = MCU在APROM中运行时，LDROM可以被更新。
[4]	CFGUEN	配置区更新使能控制 (写保护) 不管此时程序是运行在APROM还是LDROM，写1使能软件通过ISP更新配置区。 0 = 禁止配置区更新 1 = 使能配置区更新
[3]	APUEN	APROM更新使能控制 (写保护) 0 =当芯片在APROM中运行时，APROM不能被更新 1 =当芯片在APROM中运行时，APROM可以被更新。
[2]	保留	保留
[1]	BS	启动选择 (写保护) 置位/清零该位可以选择下次复位时是由LDROM启动还是由APROM启动，该位可作为MCU启动状态标志，用于检查MCU是由LDROM还是APROM启动的。除了CPU复位(RSTS_CPU=1)

		和系统复位(RSTS_SYS), 任何复位后, 该位初始值为CONFIG0的CBS的值取反。 0 = 由 APROM启动 1 = 由LDROM启动
[0]	ISPEN	ISP 使能 (写保护) 该位是ISP 使能位, 设置该位可以使能ISP功能. 0 = 禁用 ISP 功能 1 = 使能 ISP 功能

ISP地址寄存器(ISPADR)

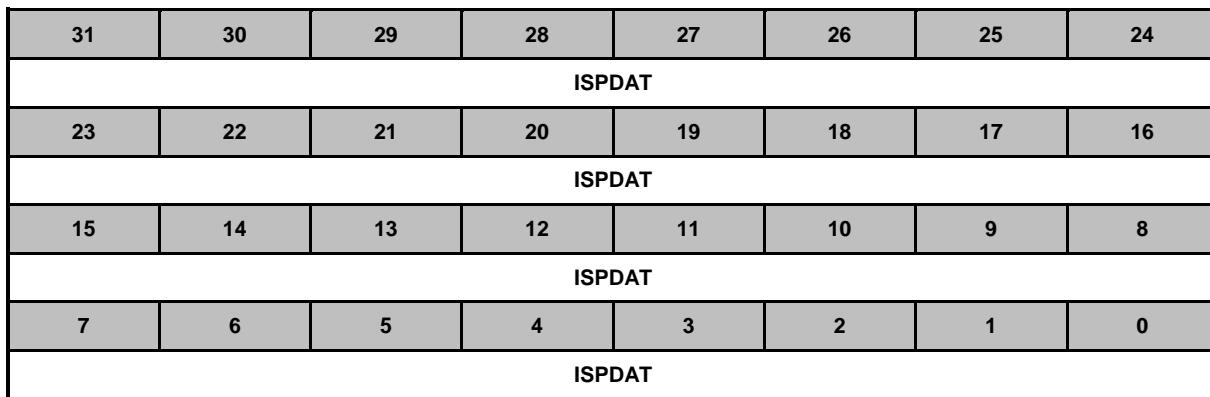
寄存器	偏移量	R/W	描述	复位值
ISPADR	FMC_BA+0x04	R/W	ISP地址寄存器	0x0000_0000



位	描述
[31:0]	ISPADR ISP 地址 NUC029xAN最大内置16k x 32-位 的flash (64 KB), 仅支持字编程. 执行ISP功能时, ISPARD[1:0] 必须为00b.

ISP数据寄存器 (ISPDAT)

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



位	描述	
[31:0]	ISPDAT	ISP 数据 ISP写操作之前，写数据到该寄存器 ISP读操作后，可从该寄存器读数据

ISP 命令 (ISPCMD)

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

位	描述	
[31:6]	保留	保留
[5:0]	ISPCMD	<p>ISP 命令</p> <p>ISP命令表如下:</p> <ul style="list-style-type: none"> 0x00 = 读 0x04 = 读 唯一 ID 0x0B = 读 公司 ID (0xDA) 0x21 = 写 0x22 = 擦除 0x2E = 向量页重映射

ISP 触发控制寄存器(ISPTRG)

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

位	描述	
[31:1]	保留	保留
[0]	ISPGO	ISP开始触发 (写保护) 写 1 开始ISP操作，当ISP操作结束后，该位由硬件自动清零。 0 = ISP 操作结束 1 = ISP 正在执行

数据FLASH基地址寄存器(DFBADR)

寄存器	地址	R/W/C	描述	复位值
DFBADR	FMC_BA+ 0x14	R	数据FLASH基地址	0x0001_F000

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

位	描述	
[31:0]	DFBADR	数据FLASH基地址 该寄存器为数据FLASH基地址寄存器,该寄存器只读. 对于 32/64KB flash 器件, 数据Flash的大小为4KB , 基地址固定为0x0001_F000.

Flash 访问时间控制寄存器 (FATCON)

寄存器	偏移量	R/W	描述	复位值
FATCON	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
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			LFOM	保留			

位	描述	
[31:8]	保留	保留
[7]	保留	保留
[6:5]	保留	保留
[4]	LFOM	低频优化模式(写保护位) 当芯片操作频率低于25MHz时，通过设置该位为1，系统可以工作得更有效率 0 = 禁用flash低频优化模式 1 = 使能flash低频优化模式
[0]	保留	保留

ISP 状态寄存器 (ISPSTA)

寄存器	偏移量	R/W	描述	复位值
ISPSTA	FMC_BA+0x40	R/W	ISP 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			VECMAP					
15	14	13	12	11	10	9	8	
VECMAP								
7	6	5	4	3	2	1	0	
保留	ISPFF	保留			CBS		ISPGO	

位	描述	
[31:21]	保留	保留
[20:9]	VECMAP	向量页映射地址 (只读) Flash地址空间{VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}被映射到 0x0000_0000~0x0000_01FF
[8:7]	保留	保留
[6]	ISPFF	ISP 失败标志(写保护) 触发ISP之后如果遇到如下情形，该位由硬件置位： (1) APROM对自身写入. (2) LDROM对自身写入. (3) CFGUEN=0时，CONFIG被擦除或修改 (4) 目标地址无效，譬如：地址超过正常范围. 该位写1清0. 注： 该位和ISPCON的bit[6]功能一样
[5:3]	保留	保留
[2:1]	CBS	芯片启动选择 (只读) 该位为CONFIG0的CBS位的镜像，
[0]	ISPGO	ISP 触发 (只读) 写1开始 ISP 操作，当ISP操作完成以后，该位由硬件自动清0 1 = ISP 操作正在进行中. 0 = ISP 操作已经完成. 注： 该位和ISPTRG的 bit0功能相同

5.5.15 寄存器映射 (NUC029FAE)

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 基地址: FMC_BA = 0x5000_C000				
ISPCON	FMC_BA+0x00	R/W	ISP控制寄存器	0x0000_0000
ISPADR	FMC_BA+0x04	R/W	ISP地址寄存器	0x0000_0000
ISPDAT	FMC_BA+0x08	R/W	ISP数据寄存器	0x0000_0000
ISPCMD	FMC_BA+0x0C	R/W	ISP命令寄存器	0x0000_0000
ISPTRG	FMC_BA+0x10	R/W	ISP触发寄存器	0x0000_0000
DFBA	FMC_BA+0x14	R	数据Flash 起始地址	0x0000_3800
ISPSTA	FMC_BA+0x40	R/W	ISP状态寄存器	0x0000_0000

5.5.16 寄存器描述 (NUC029FAE)

ISP 控制寄存器 (ISPCON)

寄存器	偏移量	R/W	描述	复位值
ISPCON	FMC_BA+0x00	R/W	ISP 控制寄存器	0x0000_0000

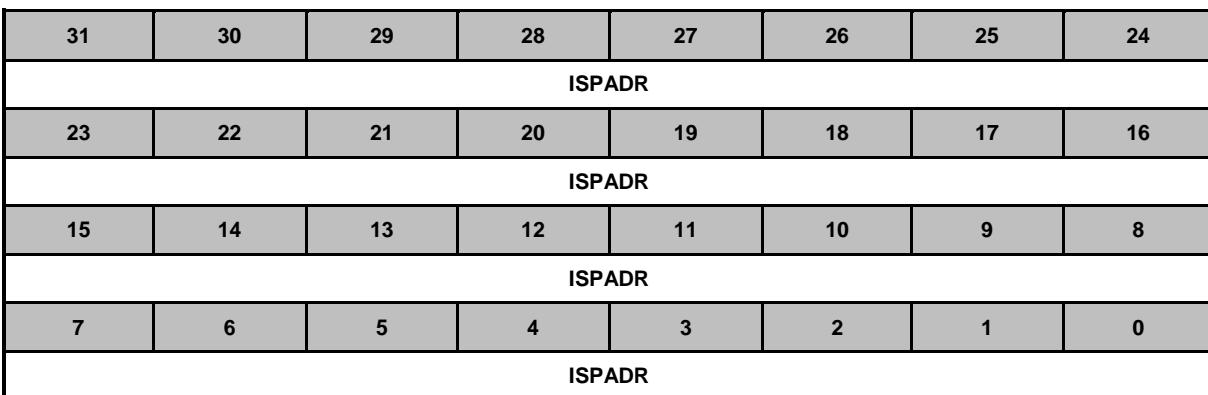
31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
保留							
7	6	5	4	3	2	1	0
保留	ISPFF	LDUEN	CFGUEN	APUEN	保留	BS	ISPEN

位	描述	
[31:7]	保留	保留
[6]	ISPFF	<p>ISP失败标志 (写保护) 当ISP满足下列条件时，该位由硬件置位： (1) APUEN =0或CBS[0]=1时，APROM对自身写入. (2) LDUEN = 0 或CBS[0]=1时,LDROM对自身写入. (3) CFGUEN=0时,用户配置被修改 (4) 目标地址无效，譬如：地址超过正常范围. 注：该位写 1 清零.</p>
[5]	LDUEN	<p>LDROM更新使能控制 (写保护) 0 = LDROM不能被更新 1 = MCU在APROM中运行时，LDROM可以被更新.</p>
[4]	CFGUEN	<p>配置区更新使能控制 (写保护) 不管此时程序是运行在APROM还是LDROM，写1使能软件通过ISP更新配置区。 0 = 禁止配置区更新 1 = 使能配置区更新</p>
[3]	APUEN	<p>APROM更新使能控制 (写保护) 0 =当芯片在APROM中运行时，APROM不能被更新 1 =当芯片在APROM中运行时，APROM可以被更新.</p>
[2]	保留	保留

位	描述
[1]	BS 启动选择（写保护） 置位/清零该位可以选择下次复位时是由LDROM启动还是由APROM启动，该位可作为MCU启动状态标志，用于检查MCU是由LDROM还是APROM启动的。除了CPU复位(RSTS_CPU=1)和系统复位(RSTS_SYS)，任何复位后，该位初始值为CONFIG0的CBS的值取反。 0 = 由 APROM启动 1 = 由LDROM启动
[0]	ISPEN ISP 使能（写保护） 该位是ISP 使能位，设置该位可以使能ISP功能。 0 = 禁用 ISP 功能 1 = 使能 ISP 功能

ISP 地址寄存器 (ISPADR)

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



位	描述								
[31:0]	ISPADR	ISP 地址 NUC029FAE 仅支持字编程. 执行ISP功能时, ISPARD[1:0] 必须为00b.							

ISP 数据寄存器 (ISPDAT)

寄存器	偏移量	R/W	描述	复位值
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 ISP写操作之前，写数据到该寄存器 ISP读操作后，可从该寄存器读数据

ISP命令 (ISPCMD)

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

位	描述	
[31:6]	保留	保留
[5:0]	ISPCMD	<p>ISP 命令</p> <p>ISP命令表如下：</p> <ul style="list-style-type: none"> 0x00 = 读 0x04 = 读 唯一 ID 0x0B = 读 公司 ID (0xDA) 0x21 = 写 0x22 = 擦除 0x2E = 向量页重映射

ISP 触发控制寄存器 (ISPTRG)

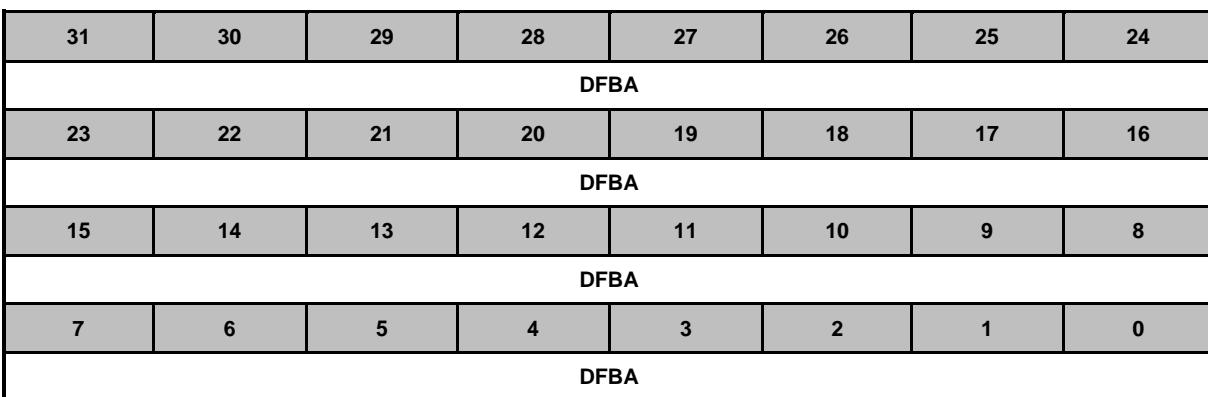
寄存器	偏移量	R/W	描述					复位值
ISPTRG	FMC_BA+0x10	R/W	ISP 触发控制寄存器					0x0000_0000

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							ISPGO

位	描述	
[31:1]	保留	保留
[0]	ISPGO	ISP开始触发 (写保护) 写 1 开始ISP操作，当ISP操作结束后，该位由硬件自动清零。 0 = ISP 操作结束 1 = ISP 正在执行

数据FLASH基地址寄存器 (DFBA)

寄存器	地址	R/W/C	描述	复位值
DFBA	FMC_BA+0x14	R	数据FLASH基地址	0x0000_3800



位	描述
[31:0]	DFBA 数据FLASH基地址 该寄存器为数据FLASH基地址寄存器,该寄存器只读. 数据 FLASH 基地址由用户设定。因为芯片擦除单元为512字节，所以其8-0位必须强制为0。

例如:

Data Flash	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
16K Flash	DFBA=0x0000_3000	DFBA=0x0000_3800	DFBA=0x0000_3C00	DFEN=1

ISP 状态寄存器 (ISPSTA)

寄存器	偏移量	R/W	描述	复位值
ISPSTA	FMC_BA+0x40	R/W	ISP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			VECMAP				
15	14	13	12	11	10	9	8
VECMAP							保留
7	6	5	4	3	2	1	0
保留	ISPFF	保留			CBS		ISPGO

位	描述	
[31:21]	保留	保留
[20:9]	VECMAP	向量页映射地址 (只读) Flash地址空间{VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}目前被映射到 0x0000_0000~0x0000_01FF
[8:7]	保留	保留
[6]	ISPFF	ISP 失败标志(写保护) 触发ISP之后如果遇到如下情形，该位由硬件置位： (1) APUEN =0或CBS[0]=1时，APROM对自身写入. (2) LDUEN = 0或CBS[0]=1时，LDROM对自身写入. (3) CFGUEN =0时,用户配置被修改 (4) 目标地址无效，譬如:地址超过正常范围. 该位写1清0. 注: 这个位和ISPCON的bit[6]功能一样
[5:3]	保留	保留
[2:1]	CBS	芯片启动选择 (只读) 该位为CONFIG0的CBS位的镜像,
[0]	ISPGO	ISP 触发 (只读) 写1开始 ISP 操作，当ISP操作完成以后，该位由硬件自动清0 0 = ISP 操作已经完成. 1 = ISP 操作正在进行中. 注: 该位和ISPTRG的 bit0功能相同

5.6 外部总线接口(EBI) (NUC029LAN/NUC029NAN)

5.6.1 概述

NuMicro® NUC029LAN/NUC029NAN配备一个外部总线接口 (EBI) , 用来访问外部设备。为节省外部设备与芯片的连接引脚数, EBI支持地址总线与数据总线复用模式, 地址锁存使能 (ALE)信号用于区分地址与数据周期.

5.6.2 特性

- 支持外部设备最大64K字节 (8位数据宽度)/128K字节(16位数据宽度)
- 基于HCLK的外部总线基时钟(MCLK)频率可调
- 支持8位或 16 位数据宽度
- 数据访问时间 (tACC), 地址锁存使能时间(tALE) 和地址保持时间(tAHD) 可调
- 支持地址总线和数据总线复用以节省地址管脚
- 空闲周期可配置用于不同的访问条件: 写命令结束(W2X), 连续读(R2R)
- 读/写操作支持0访问保持时间, 写操作有写缓冲以增强读/写效率

5.6.3 EBI 框图

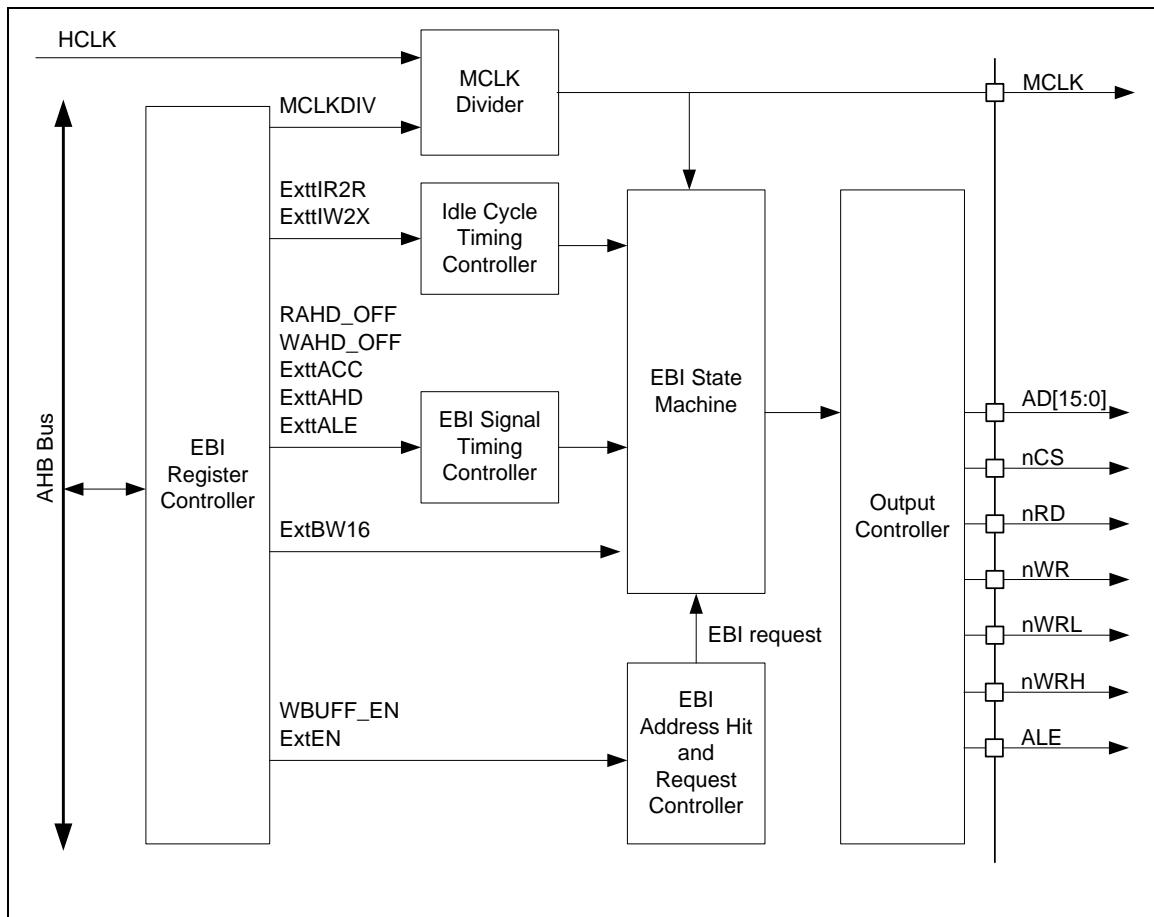


图 5-30 EBI 框图

5.6.4 基本配置

EBI基本配置如下：

- EBI设备时钟在AHBCLK[3]中使能
- AD[15:0]引脚功能在P0_MFP和P2_MFP寄存器中配置；nRD、nWR和MCLK引脚功能在P3_MFP寄存器中配置；ALE and nCS 引脚在P4_MFP 寄存器中配置；nWRL和nWRH引脚在P1_MFP寄存器中配置。

5.6.5 功能描述

5.6.5.1 EBI 区域和命中地址

EBI 地址映射在0x6000_0000 ~ 0x6001_FFFF，最大存储器空间为128K字节。当系统请求的地址命中EBI的存储空间时，相应的EBI片选信号(nCS)有效，EBI状态机开始工作。

对于8位设备(64Kbyte), EBI把该64K字节的设备同时映射到地址0x6000_0000 ~ 0x6000_FFFF 和 0x6001_0000 ~ 0x6001_FFFF.

对于16位设备(128Kbyte), EBI把该128K字节的设备映射到地址0x6000_0000 ~ 0x6001_FFFF

5.6.5.2 EBI 数据宽度连接

EBI控制器支持地址总线和数据总线复用。对于地址总线与数据总线分开的外部设备，与设备的连接需要额外的逻辑器件（锁存设备）来锁存地址。这种情况下，ALE需要连接到锁存设备以锁存地址。16位数据宽度引脚AD0 ~ AD15，8位数据宽度引脚AD0 ~ AD7作为锁存设备的输入引脚，锁存设备的输出引脚连到外部设备的Addr[15:0]。

对于16位设备，AD[15:0] 由地址线与16位数据线共享。对于 8位设备，仅AD[7:0] 由地址线与8位数据线共享，AD [15:8]则只作为地址线直接与8位设备连接

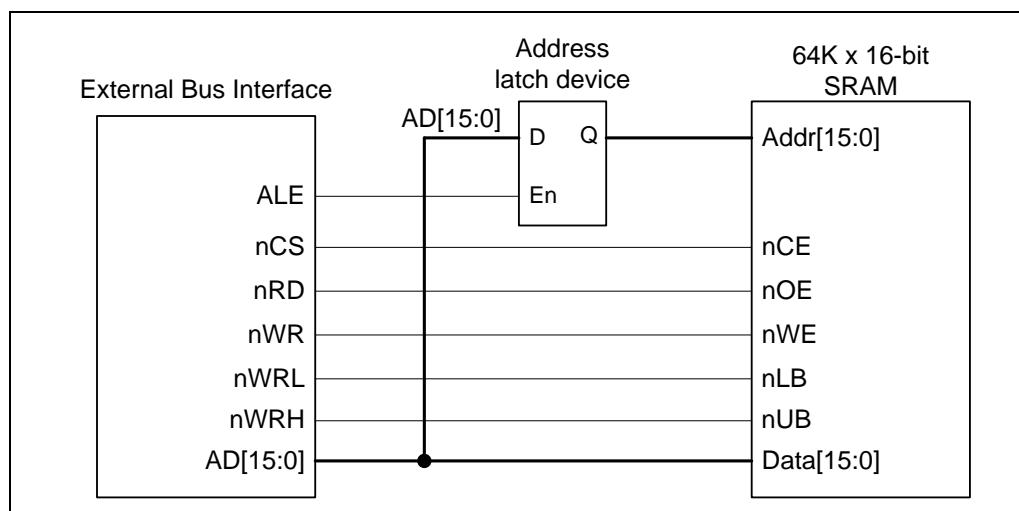


图 5-31 16位EBI数据宽度与16位器件连接

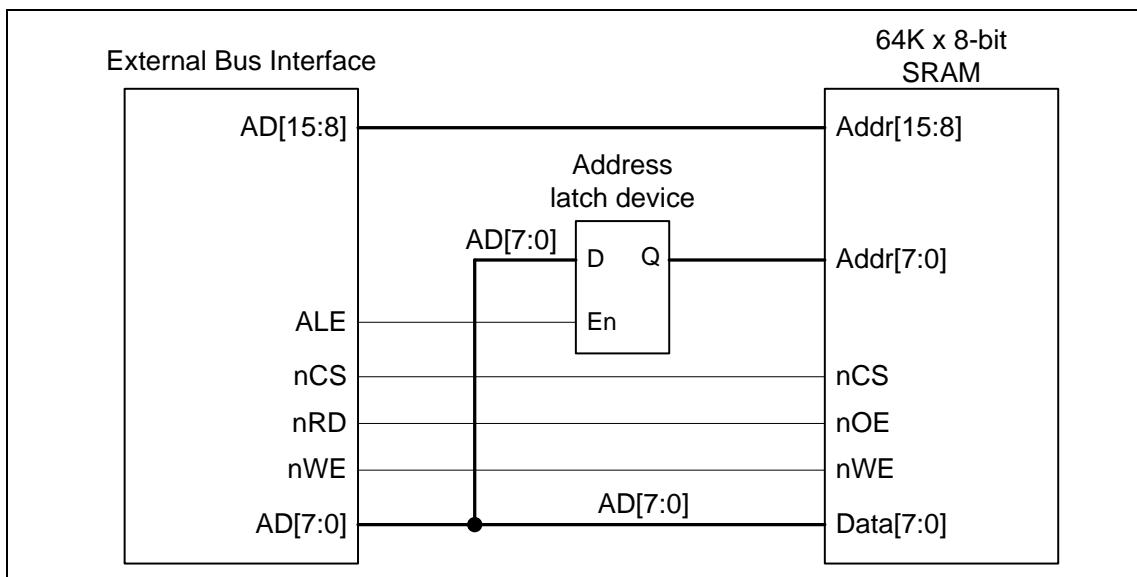


图 5-32 8位EBI数据宽度与8位设备连接

当系统访问数据宽度大于EBI的数据宽度时(8-bit / 16 bit 数据宽度), EBI控制器通过多次执行EBI访问来完成操作. 例如, 如果系统通过EBI设备请求32位数据, 如果EBI为8位数据宽度, EBI控制器将访问4次来完成操作.

5.6.5.3 EBI 操作控制

MCLK 控制

芯片中, EBI工作时, 通过MCLK同步所有EBI信号. 当芯片连接到工作频率较低的外部设备时, MCLK 可以通过设置寄存器EBICON 中的MCLKDIV[2:0]分频, 最小可达HCLK/32. 因此, EBI控制器可以适用于宽频率范围的EBI 设备。如果 MCLK的频率等于HCLK, EBI 信号由MCLK的上升沿同步, 其他情况下, EBI信号由MCLK的下降沿同步.

操作与访问时序控制

EBI开始访问时, 片选(nCS)置低并等待一个MCLK时钟周期, 等待地址建立时间(tASU)以便地址稳定。然后地址稳定后地址锁存使能信号ALE置高并保持一段时间 (tALE) 以用于地址锁存。地址锁存后, ALE 置低并等待一个MCLK 时钟周期, 等待地址锁存保持时间(tLHD) 和另一个插在tLHD后面的MCLK周期 (tA2D)用于总线转换 (地址到数据)。然后当读操作时nRD 拉低或写操作时nWR拉低。在访问保持时间tACC(用于读取输出稳定或者写完成)之后nRD和nWR被拉高。之后, EBI 信号保持数据访问保持时间(tAHD), 然后片选信号置高, 地址由当前访问控制释放。

EBI控制器提供如下表所示的灵活的EBI 时序控制以用于不同的外部设备. 在EBI 的时序控制中, tASU, tLHD 和 tA2D固定为1个MCLK周期, tAHD 可以通过设置寄存器EXTIME的ExttAHD[2:0]在1~8 个MCLK周期内调节, 通过设定RAHD_OFF/WAH_D_OFF位, 读/写时tAHD可以为0个MCLK周期; tACC可以通过设置寄存器ExttACC(EXTIME[7:3])在1~32 个MCLK周期内调节; tALE可以通过寄存器ExttALE(EBICON[18:16])在1~8 个MCLK 周期内调节.

参数	值	单位	描述
tASU	1	MCLK	地址锁存建立时间.
tALE	1 ~ 8	MCLK	ALE 高电平时间. 由ExttALE(EBICON[18:16])控制.
tLHD	1	MCLK	地址锁存保持时间.
tA2D	1	MCLK	地址到数据的延迟 (总线转换时间).
tACC	1 ~ 32	MCLK	数据访问时间. 由ExttACC(EXTIME[7:3]) 控制.
tAHD	1 ~ 8	MCLK	数据访问保持时间. 由ExttAHD(EXTIME[10:8])控制, 选择1 ~8 tAHD.
tAHD	0	MCLK	设置 RAHD_OFF/WAHD_OFF 位选择0个 tAHD.
IDLE	0 ~ 15	MCLK	空闲周期.由EXTIME 的ExtIR2R[3:0](EXTIME[27:24]) and ExtIW2X[3:0](EXTIME[15:12]) 控制

表 5-15 EBI 时序表

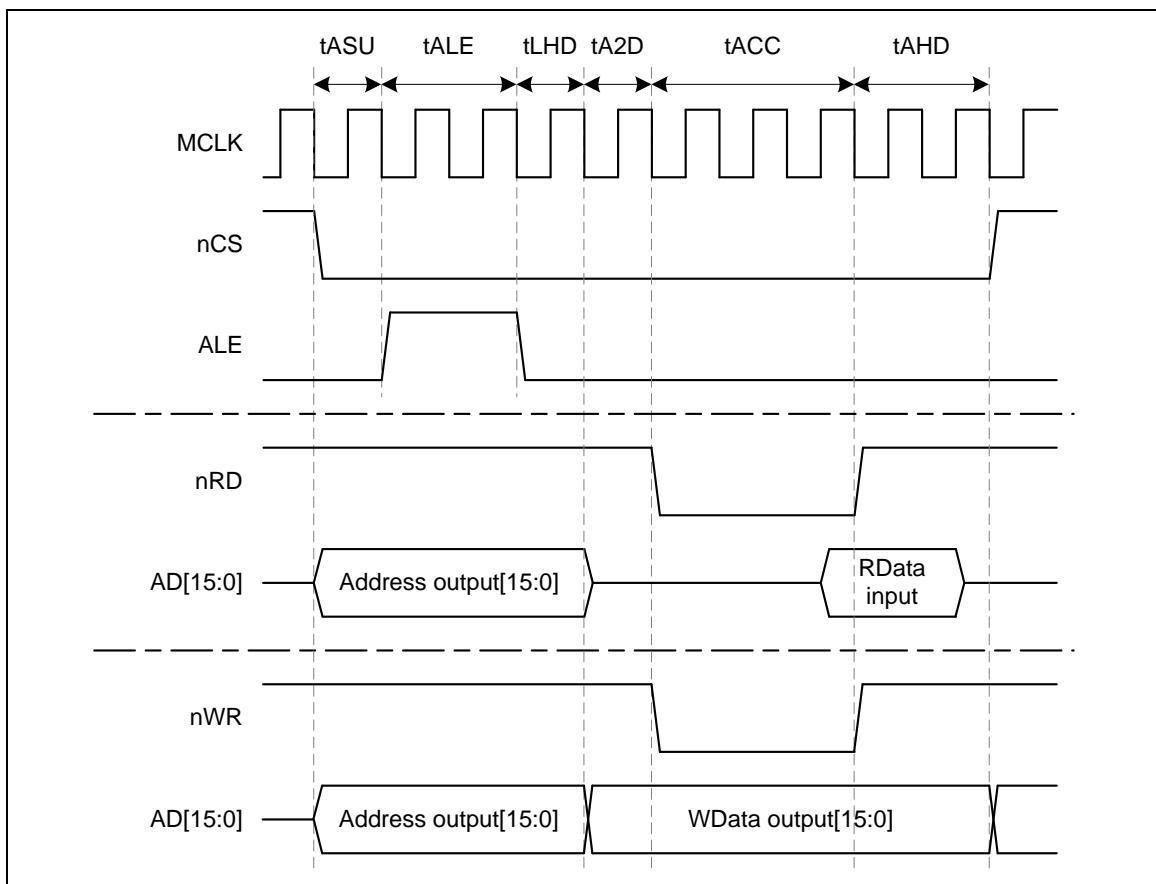


图 5-33 16位数据宽度的EBI时序控制波形

上图的时序波形是以16位数据宽度为例。此例中，AD0 ~ AD15用作地址[15:0]和数据[15:0]。当ALE拉高时，AD0 ~ AD15为地址输出。在地址锁存后(tLHD)，ALE拉低，AD0 ~ AD15总线转换成高阻以等待设备输出数据（在读取访问操作时），或用于写数据输出。

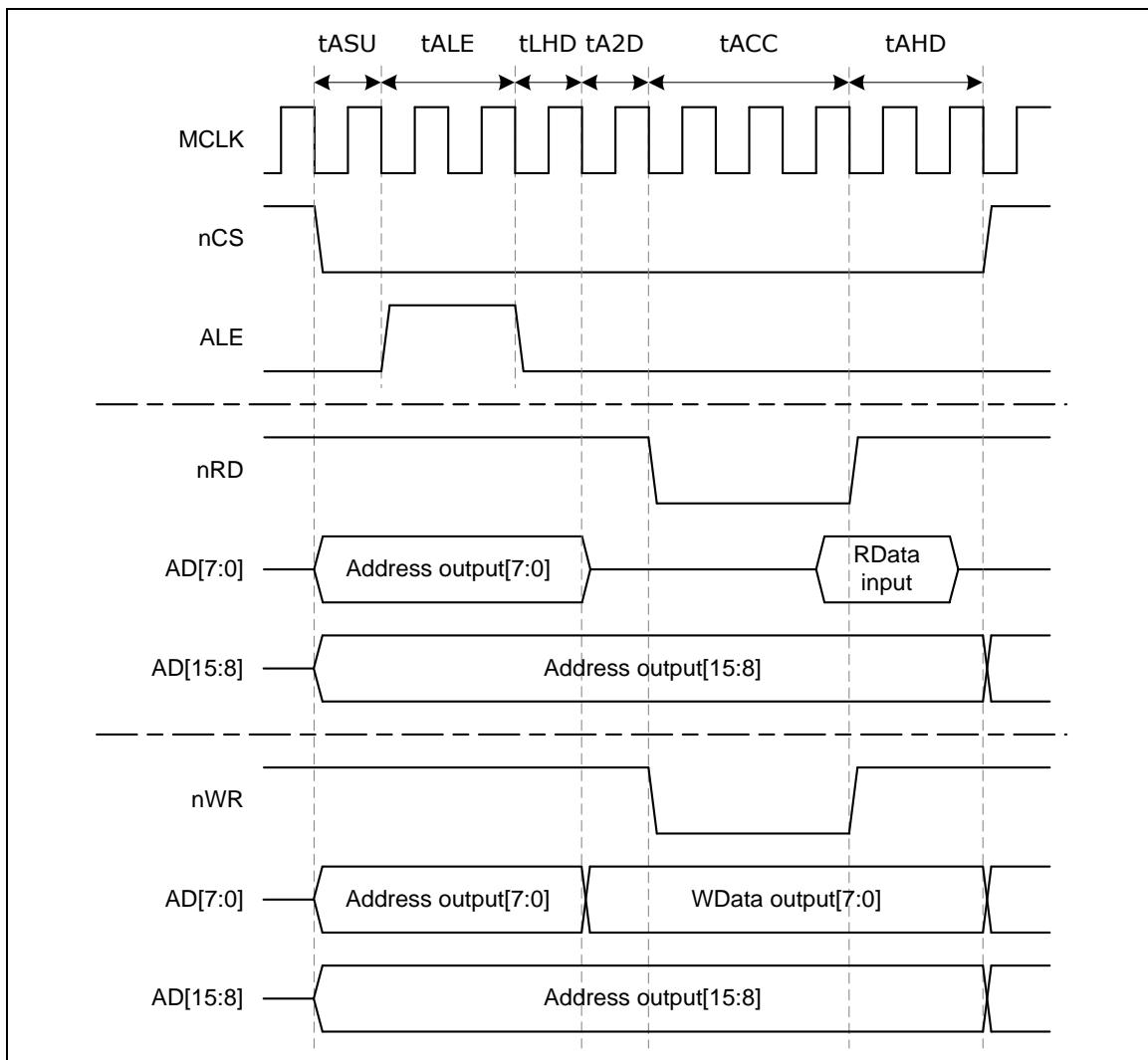


图 5-34 8位数据宽度EBI时序控制波形

上图的时序波形是以8位数据宽度为例。与16位数据宽度不同的是AD[15:8]的使用。在8位数据宽度时，AD[15:8]固定为地址位[15:8]的输出，所以外部锁存仅需要8位宽度。

插入空闲周期

当EBI连续访问时，如果设备访问速度远低于系统工作速度，可能会发生总线冲突。EBI控制器 支持额外的空闲周期以解决该问题。在空闲周期，EBI的所有控制信号无效。下图为空闲周期波形图：

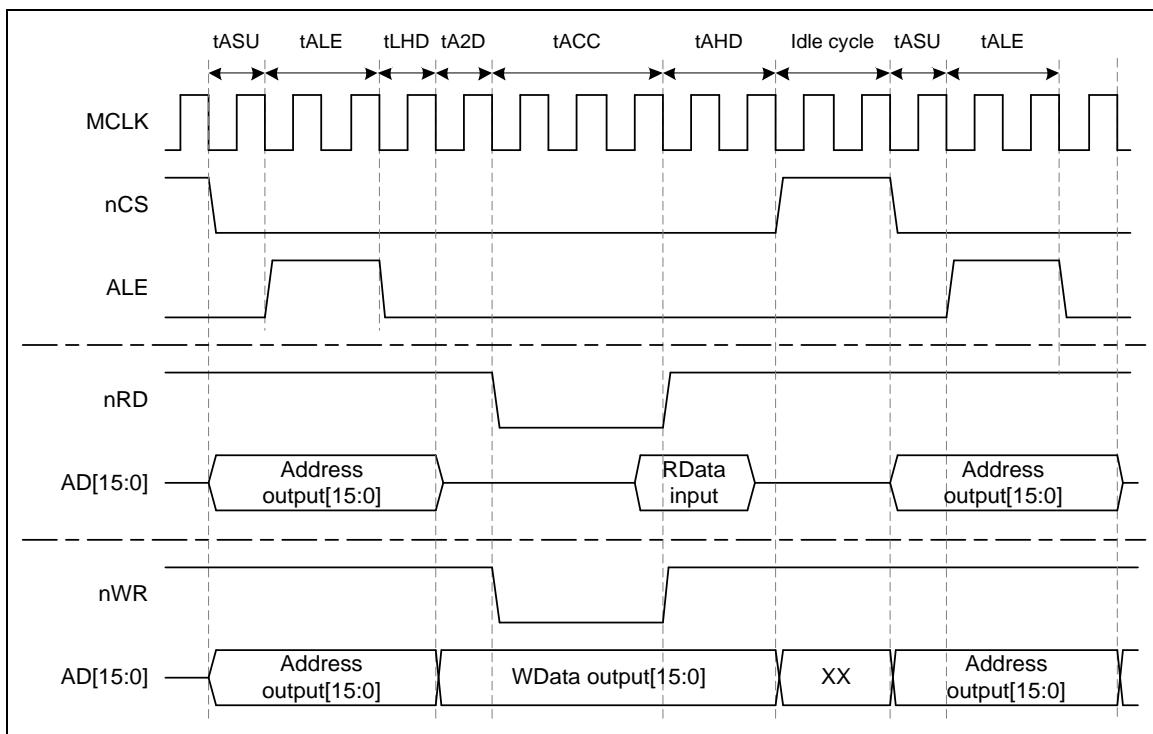


图 5-35 插入空闲周期的EBI时序控制波形

插入空闲周期的EBI时序控制波形：

1. 写访问之后
2. 读访问之后与下一个读访问之前

通过设置寄存器ExtIW2X(EXTIME [15:12]) 和 ExtIR2R(EXTIME[27:24]), 空闲周期可设定为0~15 MCLK.

写缓冲

当软件通过EBI总线写数据到外部设备的时候，EBI控制器将立即开始写动作，CPU一直等待EBI写操作完成。用户可以使能写缓冲功能提升CPU和EBI访问的性能。当EBI写缓冲功能使能时，在EBI控制器处理写操作时，CPU可以继续执行其他指令。这种情况下，有一个异常条件，如果EBI正在执行写操作，此时CPU通过EBI又执行其他数据访问，CPU则将等待。

5.6.6 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
EBI 基地址:				
EBI_BA = 0x5001_0000				
EBICON	EBI_BA+0x00	R/W	外部总线接口通用控制寄存器	0x0000_0000
EXTIME	EBI_BA+0x04	R/W	外部总线接口时序控制寄存器	0x0000_0000
EBICON2	EBI_BA+0x08	R/W	外部总线接口通用控制寄存器2	0x0000_0000

5.6.7 寄存器描述

外部总线接口控制寄存器 (EBICON)

寄存器	偏移量	R/W	描述	复位值
EBICON	EBI_BA+0x00	R/W	外部总线接口控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							
23	22	21	20	19	18	17	16
保留					ExttALE		
15	14	13	12	11	10	9	8
保留					MCLKDIV		
7	6	5	4	3	2	1	0
保留					ExtBW16	ExtEN	

位	描述																	
[31:19]	保留	保留																
[18:16]	ExttALE	<p>ALE的延长时间 该域用来控制地址锁存ALE 脉冲宽度 (tALE) $tALE = (\text{ExttALE} + 1) * \text{MCLK}$</p>																
[15:11]	保留	保留																
[10:8]	MCLKDIV	<p>外部输出时钟分频器 由 MCLKDIV 控制 EBI 输出时钟的频率，见下表:</p> <table border="1"> <thead> <tr> <th>MCLKDIV</th> <th>输出频率 (MCLK)</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>HCLK/1</td> </tr> <tr> <td>01</td> <td>HCLK/2</td> </tr> <tr> <td>010</td> <td>HCLK/4</td> </tr> <tr> <td>011</td> <td>HCLK/8</td> </tr> <tr> <td>100</td> <td>HCLK/16</td> </tr> <tr> <td>101</td> <td>HCLK/32</td> </tr> <tr> <td>11</td> <td>保留</td> </tr> </tbody> </table> <p>注: 默认输出时钟为HCLK/1</p>	MCLKDIV	输出频率 (MCLK)	000	HCLK/1	01	HCLK/2	010	HCLK/4	011	HCLK/8	100	HCLK/16	101	HCLK/32	11	保留
MCLKDIV	输出频率 (MCLK)																	
000	HCLK/1																	
01	HCLK/2																	
010	HCLK/4																	
011	HCLK/8																	
100	HCLK/16																	
101	HCLK/32																	
11	保留																	
[7:2]	保留	保留																
[1]	ExtBW16	EBI 数据宽度为 16位/8 位																

		该位配置数据总线是8位还是16位宽度。 0 = EBI 数据宽度为8位 1 = EBI 数据宽度为16位
[0]	ExtEN	EBI 使能控制 该位使能EBI功能。 0 = 禁用EBI功能 1 = 使能EBI功能

外部总线接口时序控制寄存器 (EXTIME)

寄存器	偏移量	R/W	描述	复位值
EXTIME	EBI_BA+0x04	R/W	外部总线接口时序控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留				ExtIR2R			
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
ExtIW2X				保留	ExttAHD		
7	6	5	4	3	2	1	0
ExttACC				保留			

位	描述	
[31:28]	保留	保留
[27:24]	ExtIR2R	读与读之间的空闲状态周期 ExtIR2R不等于0的时候, 如果读操作完成且下一个动作也是读时, 插入空闲状态周期且nCS信号拉高. 空闲状态周期 = (ExtIR2R * MCLK)
[23:16]	保留	保留
[15:12]	ExtIW2X	写之后的空闲状态周期 ExtIW2X不等于0的时候, 写完成之后, 将插入空闲状态且nCS拉高. 空闲状态周期 = (ExtIW2X * MCLK)
[11]	保留	保留
[10:8]	ExttAHD	EBI 数据访问保持时间 ExttAHD 定义数据访问保持时间(tAHD). $tAHD = (ExttAHD + 1) * MCLK$
[7:3]	ExttACC	EBI 数据访问时间 ExttACC 定义数据访问时间 (tACC). $tACC = (ExttACC + 1) * MCLK$
[2:0]	保留	保留

外部总线接口控制寄存器 2 (EBICON2)

寄存器	偏移量	R/W	描述	复位值
EBICON2	EBI_BA+0x08	R/W	外部总线接口控制寄存器2	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					WAHD_OFF	RAHD_OFF	WBUFF_EN

位	描述	
[31:3]	保留	保留
[2]	WAHD_OFF	写操作时访问保持时间关闭控制 0 = 通过EBI的写操作, tAHD由ExttAHD[2:0]控制 1 = 通过EBI的写操作, tAHD时间为0
[1]	RAHD_OFF	读操作时访问保持时间关闭控制 0 = 通过EBI的读操作, tAHD由ExttAHD[2:0]控制 1 = 通过EBI的读操作, tAHD时间为0
[0]	WBUFF_EN	EBI 写缓冲使能控制 使能该功能提升CPU和EBI的访问效率 0 = 禁用EBI写缓冲功能 1 = 使能EBI写缓冲功能

ISP 状态寄存器 (ISPSTA)

寄存器	偏移量	R/W	描述	复位值
ISPSTA	FMC_BA+0x40	R/W	ISP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			VECMAP					
15	14	13	12	11	10	9	8	
VECMAP								
7	6	5	4	3	2	1	0	
保留	ISPFF	保留			CBS		ISPGO	

位	描述	
[31:21]	保留	保留
[20:9]	VECMAP	向量页映射地址 (只读) Flash地址空间{VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}目前被映射到 0x0000_0000~0x0000_01FF
[8:7]	保留	保留
[6]	ISPFF	ISP 失败标志(写保护) 触发ISP之后如果遇到如下情形，该位由硬件置位： (1) APUEN =0或CBS[0]=1时，APROM对自身写入. (2) LDUEN = 0或CBS[0]=1时，LDROM对自身写入. (3) CFGUEN =0时,用户配置被修改 (4) 目标地址无效，譬如:地址超过正常范围. 该位写1清0. 注: 这个位和ISPCON的bit[6]功能一样
[5:3]	保留	保留
[2:1]	CBS	芯片启动选择 (只读) 该位为CONFIG0的CBS位的镜像,
[0]	ISPGO	ISP 触发 (只读) 写1开始 ISP 操作，当ISP操作完成以后，该位由硬件自动清0 0 = ISP 操作已经完成. 1 = ISP 操作正在进行中. 注: 该位和ISPTRG的 bit0功能相同

5.7 通用 I/O (GPIO)

5.7.1 概述

NuMicro® NUC029 最多有40个通用I/O引脚，这些引脚和其它功能共享。40个引脚分为6个端口，分别命名为P0, P1, P2, P3, P4和P5，每个端口最多有8个引脚。每个引脚都是独立的，都有相应的寄存器来控制引脚工作模式与数据。

每个引脚的I/O类型可由软件独立地配置为输入，输出，开漏或准双向模式。每个I/O引脚配有一个非常弱的独立的上拉电阻，VDD从5.0V 到 2.5V时，内部弱上拉电阻阻值大约为 $110\text{K}\Omega\sim300\text{K}\Omega$ 。

5.7.2 特性

- 4 种 I/O 模式:
 - 准双向
 - 推挽输出
 - 开漏输出
 - 输入模式带高阻
- 触发输入模式由Px_MFP[23:16]中的Px_TYPE[15:0]选择
- 每个I/O 引脚都可以作为中断源，支持边沿/电平触发
- 所有I/O引脚复位后的默认模式由CIOINI(CONFIG[10])决定
 - NUC029xAN:
 - 如果 CIOINI 为 0，所有GPIO引脚在芯片复位后为输入三态模式
 - 如果 CIOINI 为 1，所有GPIO引脚在芯片复位后为准双向模式(默认)
 - 在复位后，所有引脚的I/O模式都保持为准双向模式，且每个端口数据寄存器Px_DOUT[7:0]的值复位为0x000_00FF。
 - NUC029FAE:
 - 如果 CIOINI 为 0，所有GPIO引脚在芯片复位后为准双向模式
 - 如果 CIOINI 为 1，所有GPIO引脚在芯片复位后为输入三态模式(默认)
- 仅在准双向模式下，I/O引脚内部上拉电阻被使能
- 引脚中断功能使能后，引脚的唤醒功能也将被使能

5.7.3 基本配置

GPIO 引脚功能由 P0_MFP, P1_MFP, P2_MFP, P3_MFP, P4_MFP 和 P5_MFP(只有 NUC029FAE) 寄存器配置

5.7.4 功能描述

5.7.4.1 输入模式

设置 Px_PMD(PMDn[1:0]) 为 00'b , Px.n 为输入模式, I/O 引脚为三态 (高阻态), 没有输出驱动能力。Px_PIN 的值反映相应端口引脚的状态。

5.7.4.2 推挽输出模式

设置 Px_PMD(PMDn[1:0]) 为 01'b, Px.n 为推挽输出模式, I/O 引脚支持数字输出功能, 有拉电流/灌电流能力。Px_DOUT[n] 相应位的值被送到相应引脚上。

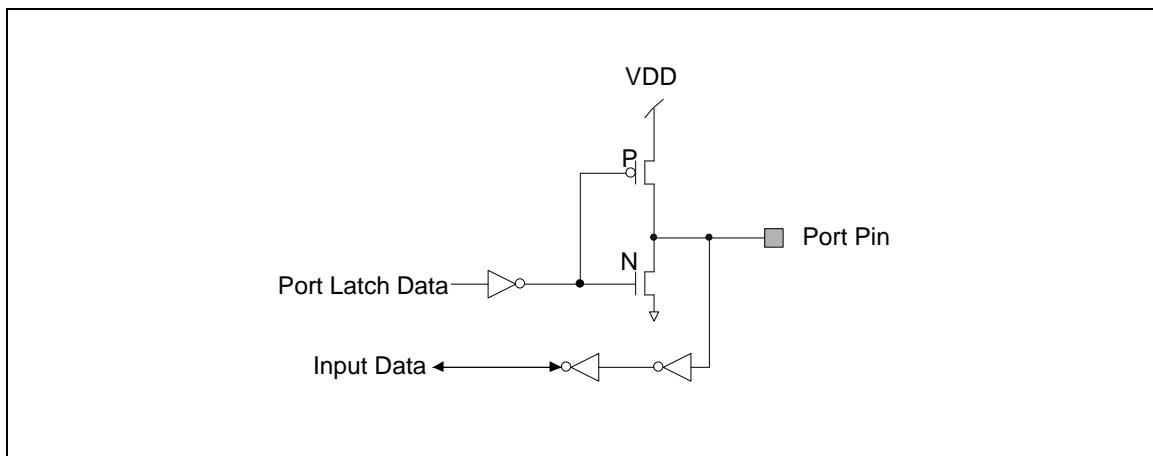


图 5-36 推挽输出

5.7.4.3 开漏输出模式

设置 Px_PMD(PMDn[1:0]) 为 10'b, Px.n 为开漏模式, I/O 支持数字输出功能, 但仅有灌电流能力, 为了把 I/O 引脚拉到高电平状态, 需要外接一颗上拉电阻。如果 Px_DOUT[n] 相应位的值为“0”, 引脚上输出低电平. 如果 Px_DOUT[n] 相应位的值为“1”, 该引脚由外部上拉电阻控制输出高电平。

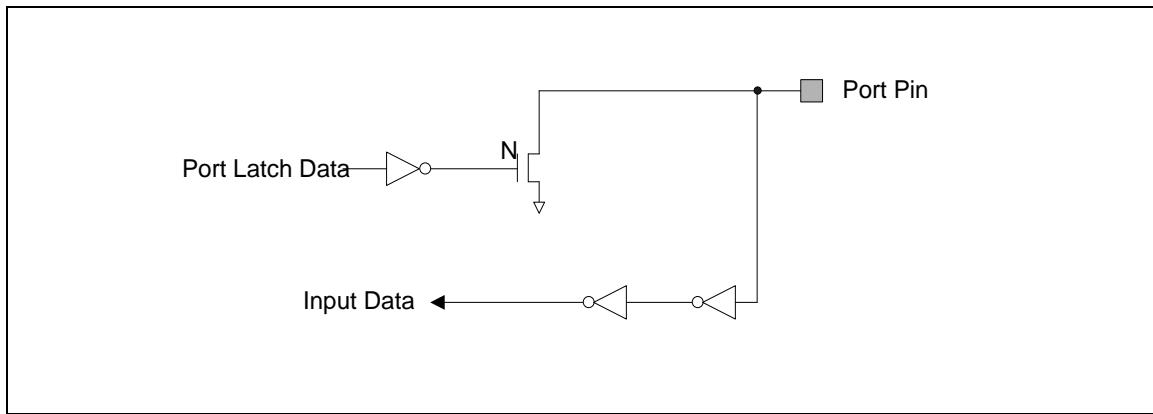


图 5-37 开漏输出

5.7.4.4 准双向模式

设置Px_PMD(PMDn[1:0]) 为 11'b, Px.n 引脚为准双向模式, I/O 同时支持数字输出和输入功能, 但拉电流仅达数百 uA。要实现数字输入功能, 需要先将Px_DOUT[n] 相应位置1。若Px_DOUT[n]相应位为"0", 引脚上输出为"低电平"。若Px_DOUT[n]相应位为"1", 该引脚将检查引脚值, 若引脚值为高, 没有任何动作, 若引脚值为低, 该引脚将驱动2个时钟周期的强拉电流, 然后关闭强输出驱动, 然后引脚状态由内部上拉电阻控制。需注意的是, 准双向模式拉电流的能力只有200uA到30uA, 其中V_{DD}由5.0 V 到 2.5 V。

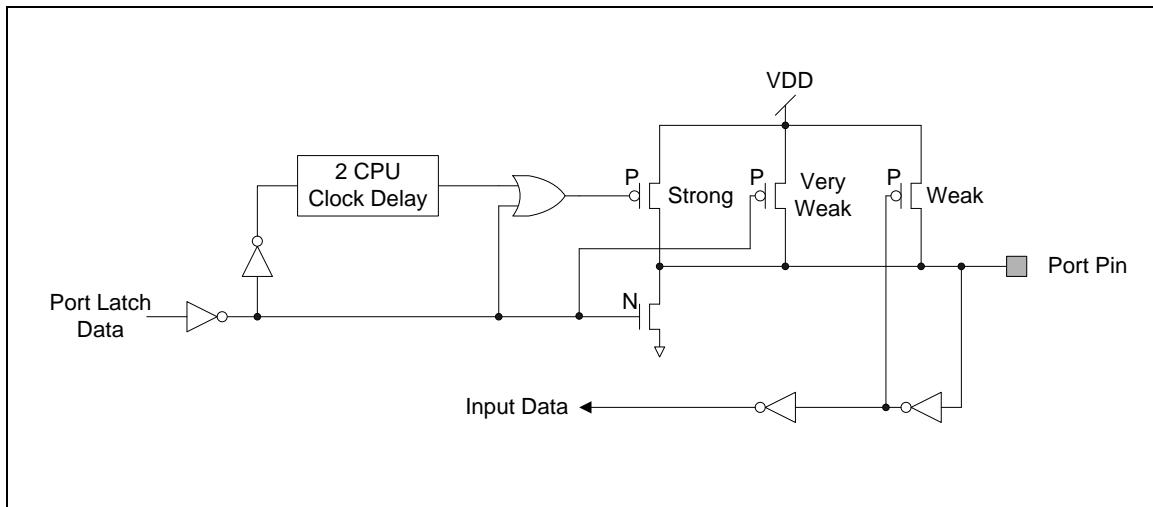


图 5-38 准双向 I/O 模式

5.7.5 GPIO 中断和唤醒功能

通过设置Px_IEN 和 Px_IMD 寄存器, 每个GPIO引脚都可以设置为中断源。有5种类型的中断条件可以选择：低电平触发、高电平触发、下降沿触发、上升沿触发、上升和下降沿都触发。对于边沿触发, 用户可以使能输入信号去抖动功能来避免由干扰导致的不期望的中断发生。de-bounce 时钟源和采样周期可以由DBNCECON 寄存器设定。

5.7.6 寄存器映射 (NUC029xAN)

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:				
GP_BA = 0x5000_4000				
P0_PMD	GP_BA+0x000	R/W	P0 I/O 模式控制	0x0000_XXXX
P0_OFFD	GP_BA+0x004	R/W	P0 关闭数字输入通路控制	0x0000_0000
P0_DOUT	GP_BA+0x008	R/W	P0 数据输出值	0x0000_00FF
P0_DMASK	GP_BA+0x00C	R/W	P0 数据输出写屏蔽	0x0000_0000
P0_PIN	GP_BA+0x010	R	P0 引脚数值	0x0000_00XX
P0_DBEN	GP_BA+0x014	R/W	P0 防抖动使能控制	0x0000_0000
P0_IMD	GP_BA+0x018	R/W	P0 中断模式控制	0x0000_0000
P0_IEN	GP_BA+0x01C	R/W	P0 中断使能控制	0x0000_0000
P0_ISRC	GP_BA+0x020	R/W	P0 中断源标志	0x0000_0000
P1_PMD	GP_BA+0x040	R/W	P1 I/O 模式控制	0x0000_XXXX
P1_OFFD	GP_BA+0x044	R/W	P1 关闭数字输入通路控制	0x0000_0000
P1_DOUT	GP_BA+0x048	R/W	P1 数据输出值	0x0000_00FF
P1_DMASK	GP_BA+0x04C	R/W	P1 数据输出写屏蔽	0x0000_0000
P1_PIN	GP_BA+0x050	R	P1 引脚数值	0x0000_00XX
P1_DBEN	GP_BA+0x054	R/W	P1 防抖动使能控制	0x0000_0000
P1_IMD	GP_BA+0x058	R/W	P1 中断模式控制	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 中断使能控制	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 中断源标志	0x0000_0000
P2_PMD	GP_BA+0x080	R/W	P2 I/O 模式控制	0x0000_XXXX
P2_OFFD	GP_BA+0x084	R/W	P2 关闭数字输入通路控制	0x0000_0000
P2_DOUT	GP_BA+0x088	R/W	P2 数据输出值	0x0000_00FF
P2_DMASK	GP_BA+0x08C	R/W	P2 数据输出写屏蔽	0x0000_0000
P2_PIN	GP_BA+0x090	R	P2 引脚数值	0x0000_00XX
P2_DBEN	GP_BA+0x094	R/W	P2 防抖动使能控制	0x0000_0000
P2_IMD	GP_BA+0x098	R/W	P2 中断模式控制	0x0000_0000

P2_IEN	GP_BA+0x09C	R/W	P2 中断使能控制	0x0000_0000
P2_ISRC	GP_BA+0x0A0	R/W	P2 中断源标志	0x0000_0000
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O 模式控制	0x0000_XXXX
P3_OFFD	GP_BA+0x0C4	R/W	P3 关闭数字输入通路控制	0x0000_0000
P3_DOUT	GP_BA+0x0C8	R/W	P3 数据输出值	0x0000_00FF
P3_DMASK	GP_BA+0x0CC	R/W	P3 数据输出写屏蔽	0x0000_0000
P3_PIN	GP_BA+0x0D0	R	P3 引脚数值	0x0000_00XX
P3_DBEN	GP_BA+0x0D4	R/W	P3 防抖动使能控制	0x0000_0000
P3_IMD	GP_BA+0x0D8	R/W	P3 中断模式控制	0x0000_0000
P3_IEN	GP_BA+0x0DC	R/W	P3 中断使能控制	0x0000_0000
P3_ISRC	GP_BA+0x0E0	R/W	P3 中断源标志	0x0000_0000
P4_PMD	GP_BA+0x100	R/W	P4 I/O 模式控制	0x0000_XXXX
P4_OFFD	GP_BA+0x104	R/W	P4 关闭数字输入通路控制	0x0000_0000
P4_DOUT	GP_BA+0x108	R/W	P4 数据输出值	0x0000_00FF
P4_DMASK	GP_BA+0x10C	R/W	P4 数据输出写屏蔽	0x0000_0000
P4_PIN	GP_BA+0x110	R	P4 引脚数值	0x0000_00XX
P4_DBEN	GP_BA+0x114	R/W	P4 防抖动使能控制	0x0000_0000
P4_IMD	GP_BA+0x118	R/W	P4 中断模式控制	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 中断使能控制	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 中断源标志	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	中断去抖动控制	0x0000_0020
P0n_PDIO n = 0, 1...7	GP_BA+0x200 + 0x04 * n	R/W	GPIO P0.n 引脚数据输入/输出	0x0000_000X
P1n_PDIO n = 0, 1...7	GP_BA+0x220 + 0x04 * n	R/W	GPIO P1.n 引脚数据输入/输出	0x0000_000X
P2n_PDIO n = 0, 1...7	GP_BA+0x240 + 0x04 * n	R/W	GPIO P2.n 引脚数据输入/输出	0x0000_000X
P3n_PDIO n = 0, 1...7	GP_BA+0x260 + 0x04 * n	R/W	GPIO P3.n 引脚数据输入/输出	0x0000_000X
P4n_PDIO n = 0, 1...7	GP_BA+0x280 + 0x04 * n	R/W	GPIO P4.n 引脚数据输入/输出	0x0000_000X

5.7.7 寄存器描述 (NUC029xAN)

Port 0-4 I/O 模式控制 (Px PMD)

寄存器	偏移量	R/W	描述	复位值
P0_PMD	GP_BA+0x000	R/W	P0 I/O 模式控制	0x0000_XXXX
P1_PMD	GP_BA+0x040	R/W	P1 I/O 模式控制	0x0000_XXXX
P2_PMD	GP_BA+0x080	R/W	P2 I/O 模式控制	0x0000_XXXX
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O 模式控制	0x0000_XXXX
P4_PMD	GP_BA+0x100	R/W	P4 I/O 模式控制	0x0000_XXXX

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

位	描述	
[31:16]	悶隱	保留.
[15 :0]	PMDn	<p>Port 0-4 I/O Pin[N] 模式控制</p> <p>决定Px.n每个引脚的I/O模式</p> <p>00 = Px.n为输入模式.</p> <p>01 = Px.n为推挽输出模式.</p> <p>10 = Px.n为开漏模式.</p> <p>11 = Px.n为准双向模式</p> <p>注1: x=0~4, n = 0~7</p> <p>注2: 该域的初始值由CIOINI (CONFIG[10])决定</p> <p>如果CIOINI等于1, 默认值为0x0000_FFFF, 就是说所有引脚在芯片上电后为准双向模式</p> <p>如果CIOINI等于0, 默认值为0x0000_0000, 就是说所有引脚在芯片上电后为输入三态模式</p>

Port 0-4 数字输入通路关闭控制 (Px_OFFD)

寄存器	偏移量	R/W	描述	复位值
P0_OFFD	GP_BA+0x004	R/W	P0 数字输入通道关闭控制	0x0000_0000
P1_OFFD	GP_BA+0x044	R/W	P1 数字输入通道关闭控制	0x0000_0000
P2_OFFD	GP_BA+0x084	R/W	P2 数字输入通道关闭控制	0x0000_0000
P3_OFFD	GP_BA+0x0C4	R/W	P3 数字输入通道关闭控制	0x0000_0000
P4_OFFD	GP_BA+0x104	R/W	P4 数字输入通道关闭控制	0x0000_0000

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

位	描述	
[31:24]	帳隱	保留
[23:16]	OFFD	<p>Port 0-4 Pin[N] 数字输入通道关闭控制</p> <p>该域每个位用来控制是否关闭相应Px.n引脚数字输入通路。如果某个引脚输入的是模拟信号，用户应该关闭相应引脚的数字通路避免漏电</p> <p>0 = 使能Px.n 数字输入通道 1 = 关闭Px.n 数字输入通道（数字输入连接到低电平）</p> <p>注: x=0~4, n = 0~7</p>
[15:0]	帳隱	保留

Port 0-4 数据输出值 (Px_DOUT)

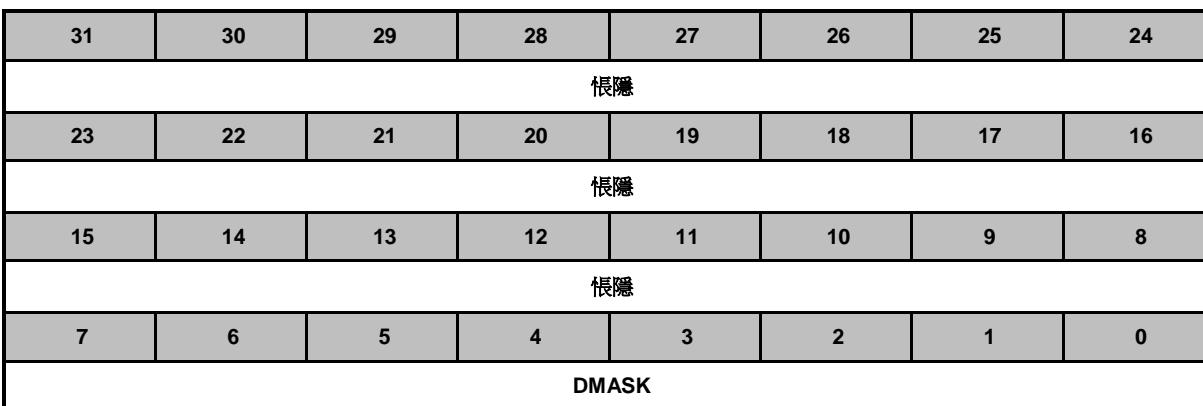
寄存器	偏移量	R/W	描述	复位值
P0_DOUT	GP_BA+0x008	R/W	P0 数据输出值	0x0000_00FF
P1_DOUT	GP_BA+0x048	R/W	P1 数据输出值	0x0000_00FF
P2_DOUT	GP_BA+0x088	R/W	P2 数据输出值	0x0000_00FF
P3_DOUT	GP_BA+0x0C8	R/W	P3 数据输出值	0x0000_00FF
P4_DOUT	GP_BA+0x108	R/W	P4 数据输出值	0x0000_00FF



位	描述	
[31:8]	保留	
[15:0]	DOUT[n]	<p>Port 0-4 Pin[N] 输出值 当Px.n配置成输出, 开漏, 或准双向模式时, 这些位控制Px.n引脚的状态 0 = 当Px.n配置成输出, 开漏, 或准双向模式时, Px.n 引脚状态驱动为低电平 1 = 当Px.n配置成输出, 开漏, 或准双向模式时, Px.n 引脚状态驱动为高电平 注: x=0~4, n = 0~7</p>

Port0-4 数据输出写屏蔽(Px_DMASK)

寄存器	偏移量	R/W	描述	复位值
P0_DMASK	GP_BA+0x00C	R/W	P0 数据输出写屏蔽	0x0000_0000
P1_DMASK	GP_BA+0x04C	R/W	P1 数据输出写屏蔽	0x0000_0000
P2_DMASK	GP_BA+0x08C	R/W	P2 数据输出写屏蔽	0x0000_0000
P3_DMASK	GP_BA+0x0CC	R/W	P3 数据输出写屏蔽	0x0000_0000
P4_DMASK	GP_BA+0x10C	R/W	P4 数据输出写屏蔽	0x0000_0000



位	描述	
[31:8]	保留	
[7:0]	DMASK[n]	<p>Port 0-4 Pin [N] 数据输出写屏蔽</p> <p>这些位用来保护寄存器Px_DOUT[n]的相应位。当DMASK[n]位设置为“1”时，Px_DOUT[n]的相应位被保护。如果写信号被屏蔽，写数据到保护位将被忽略</p> <p>0 = 相应的Px_DOUT [n] 位可以被更新 1 = 相应的Px_DOUT [n] 位受保护</p> <p>注1: x=0~4, n = 0~7 注2: 该功能只是保护相应的Px_DOUT[n]位，但是不保护Pxn_PDIO寄存器中的位</p>

Port 0-4 引脚状态 (Px_PIN)

寄存器	偏移量	R/W	描述	复位值
P0_PIN	GP_BA+0x010	R	P0 引脚状态	0x0000_00XX
P1_PIN	GP_BA+0x050	R	P1 引脚状态	0x0000_00XX
P2_PIN	GP_BA+0x090	R	P2 引脚状态	0x0000_00XX
P3_PIN	GP_BA+0xD0	R	P3 引脚状态	0x0000_00XX
P4_PIN	GP_BA+0x110	R	P4 引脚状态	0x0000_00XX



位	描述	
[31:8]	帳隱	保留
[15:0]	PIN[n]	Port 0-4 Pin [N] 引脚状态 这些位的值反映各个Px.n 引脚的实际状态为高还是低，如果某个位的值为1，则相应引脚的状态为高电平，否则就是低电平 注：x=0~4, n = 0~7

Port 0-4 去抖动使能 (Px_DBEN)

寄存器	偏移量	R/W	描述	复位值
P0_DBEN	GP_BA+0x014	R/W	P0 去抖动使能控制	0x0000_0000
P1_DBEN	GP_BA+0x054	R/W	P1 去抖动使能控制	0x0000_0000
P2_DBEN	GP_BA+0x094	R/W	P2 去抖动使能控制	0x0000_0000
P3_DBEN	GP_BA+0x0D4	R/W	P3 去抖动使能控制	0x0000_0000
P4_DBEN	GP_BA+0x114	R/W	P4 去抖动使能控制	0x0000_0000



位	描述	
[31:8]	悵隱	保留
[15:0]	DBEN[n]	<p>Port 0-4 Pin [N] 输入信号去抖动使能控制</p> <p>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，则输入信号被视为信号抖动，从而不触发中断。去抖动时钟源由DBNCECON[4]控制，去抖动采样周期由DBNCECON[3:0]控制</p> <p>0 = 禁用 Px.n去抖动功能 1 = 使能 Px.n去抖动功能</p> <p>去抖动功能仅限于边沿触发中断，不能用于电平触发中断。如果中断模式为电平触发，去抖动使能位将被忽略。</p> <p>注1: x=0~4, n = 0~7</p> <p>注2: Px.n引脚如果作为睡眠唤醒源，需在进入睡眠模式之前，关闭Px.n引脚的去抖动功能。以避免在系统由Px.n去抖动功能唤醒之后，发生第二次中断事件。</p>

Port 0-4 中断模式控制 (Px_IMD)

寄存器	偏移量	R/W	描述	复位值
P0_IMD	GP_BA+0x018	R/W	P0 中断模式控制	0x0000_0000
P1_IMD	GP_BA+0x058	R/W	P1 中断模式控制	0x0000_0000
P2_IMD	GP_BA+0x098	R/W	P2 中断模式控制	0x0000_0000
P3_IMD	GP_BA+0x0D8	R/W	P3 中断模式控制	0x0000_0000
P4_IMD	GP_BA+0x118	R/W	P4 中断模式控制	0x0000_0000



位	描述	
[31:8]	帳隱	保留
[15:0]	IMD[n]	<p>Port 0-4 Pin [N] 边沿或电平检测中断模式控制</p> <p>IMD[n] 用于控制中断是电平触发还是边沿触发。若中断是边沿触发，触发源可以由去抖动控制，如果中断是电平触发，输入信号源由一个HCLK时钟采样并产生中断</p> <p>0 = 边沿触发中断 1 = 电平触发中断</p> <p>如果引脚设置为电平触发中断，寄存器Px_IEN中只有一种电平(高电平/低电平)可以被设定。若设置两种电平都触发中断，设置将被忽略，不会产生中断</p> <p>去抖动功能只对于边沿触发中断有效。如果中断模式为电平触发，去抖动使能位将被忽略。</p> <p>注：x=0~4, n = 0~7</p>

Port 0-4 中断使能控制 (Px_IEN)

寄存器	偏移量	R/W	描述	复位值
P0_IEN	GP_BA+0x01C	R/W	P0 中断使能控制	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 中断使能控制	0x0000_0000
P2_IEN	GP_BA+0x09C	R/W	P2 中断使能控制	0x0000_0000
P3_IEN	GP_BA+0x0DC	R/W	P3 中断使能控制	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 中断使能控制	0x0000_0000

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

位	描述	
[31:24]	悶隱	保留
[23:16]	IR_EN[n]	<p>Port 0-4 Pin [N] 输入上升沿或输入高电平中断使能</p> <p>IR_EN[n] 位用于使能相应Px.n引脚的中断。置“1”也同时使能引脚唤醒功能</p> <p>当设置 IR_EN[n] 位为“1”时：</p> <p>如果中断是电平触发模式(IMD[n] = 1)，输入Px.n引脚的状态为高电平时，产生中断。</p> <p>如果中断是边沿触发模式(IMD[n] = 0)，输入Px.n引脚的状态由低电平到高电平变化时，产生中断</p> <p>0 = 禁用Px.n 高电平或由低电平到高电平变化的中断。</p> <p>1 = 使能Px.n 高电平或由低电平到高电平变化的中断</p> <p>注：x=0~4, n = 0~7</p>
[15:8]	悶隱	保留
[7:0]	IF_EN[n]	<p>Port 0-4 Pin [N] 输入下降沿或输入低电平中断使能</p> <p>IR_EN[n] 位用于使能相应Px.n引脚的中断。置“1”也同时使能引脚唤醒功能</p> <p>当设置IF_EN[n]位为“1”时：</p> <p>如果中断是电平触发模式(IMD[n] = 1)，输入Px.n引脚的状态为低电平时，产生中断。</p> <p>如果中断是边沿触发模式(IMD[n] = 0)，输入Px.n引脚的状态由高电平到低电平变化时，产生中断</p>

		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: x=0~4, n = 0~7
--	--	--

Port 0-4 中断源标志 (Px_ISRC)

寄存器	偏移量	R/W	描述	复位值
P0_ISRC	GP_BA+0x020	R/W	P0 中断源标志	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 中断源标志	0x0000_0000
P2_ISRC	GP_BA+0x0A0	R/W	P2 中断源标志	0x0000_0000
P3_ISRC	GP_BA+0x0E0	R/W	P3 中断源标志	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 中断源标志	0x0000_0000



位	描述	
[31:8]	帳隱	保留
[7:0]	ISRC[n]	<p>Port 0-4 Pin [N] 中断源标志</p> <p>写：</p> <p>0 = 无动作 1 = 清相应的中断标志</p> <p>读：</p> <p>0 = Px.n没有中断 1 = Px.n产生中断</p> <p>注： x=0~4, n = 0~7</p>

中断去抖动周期控制 (DBNCECON)

寄存器	偏移量	R/W	描述	复位值
DBNCECON	GP_BA+0x180	R/W	中断去抖动控制	0x0000_00020

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

位	描述	
[31:6]	帳隱	保留
[5]	ICLK_ON	<p>中断时钟打开模式 0 = 只有当I/O引脚的相应位Px_IEN 等于1时，边沿检测电路才被激活 1 = 所有I/O引脚的边沿检测电路在芯片复位后总是激活的 注：如果没有特别考量，建议关闭该位以节省系统功耗</p>
[4]	DBCLKSRC	<p>去抖动计数器时钟源选择 0 = 去抖动计数器时钟源为 HCLK 1 = 去抖动计数器时钟源为内部10KHz 内部低速时钟</p>
[3:0]	DBCLKSEL	<p>去抖动采样周期选择 0000 = 每 1 个时钟周期采样中断输入一次. 0001 = 每 2 个时钟周期采样中断输入一次. 0010 = 每 4 个时钟周期采样中断输入一次. 0011 = 每 8 个时钟周期采样中断输入一次. 0100 = 每16 个时钟周期采样中断输入一次. 0101 = 每32 个时钟周期采样中断输入一次. 0110 = 每64 个时钟周期采样中断输入一次. 0111 = 每128 个时钟周期采样中断输入一次. 1000 = 每256 个时钟周期采样中断输入一次. 1001 = 每2^*256 个时钟周期采样中断输入一次. 1010 = 每4^*256 个时钟周期采样中断输入一次. 1011 = 每8^*256 个时钟周期采样中断输入一次. 1100 = 每16^*256 个时钟周期采样中断输入一次. 1101 = 每32^*256 个时钟周期采样中断输入一次.</p>

		1110 = 每64*256 个时钟周期采样中断输入一次. 1111 = 每128*256 个时钟周期采样中断输入一次.
--	--	---

GPIO Px.n 引脚数据输入/输出 (PxN_PDIO)

寄存器	偏移量	R/W	描述	复位值
P0n_PDIO n = 0,1...7	GP_BA+0x200 + 0x04 * n	R/W	GPIO P0.n 引脚数据输入/输出	0x0000_000X
P1n_PDIO n = 0,1...7	GP_BA+0x220 + 0x04 * n	R/W	GPIO P1.n 引脚数据输入/输出	0x0000_000X
P2n_PDIO n = 0,1...7	GP_BA+0x240 + 0x04 * n	R/W	GPIO P2.n 引脚数据输入/输出	0x0000_000X
P3n_PDIO n = 0,1...7	GP_BA+0x260 + 0x04 * n	R/W	GPIO P3.n 引脚数据输入/输出	0x0000_000X
P4n_PDIO n = 0,1...7	GP_BA+0x280 + 0x04 * n	R/W	GPIO P4.n 引脚数据输入/输出	0x0000_000X

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

位	描述	
[31:1]	悶隱	保留
[0]	Pxn_PDIO	<p>GPIO Px.N 引脚数据输入/输出 写该位可以控制一个GPIO引脚的输出值 0 = 设置相应的GPIO引脚为低 1 = 设置相应的GPIO引脚为高 读这个寄存器可以得到IO引脚的状态 0 = 相应的GPIO引脚状态为低 1 = 相应的GPIO引脚状态为高 例如：写到P00_DOUT的值，会反应到寄存器 P0_DOUT[0]中，读P00_DOUT将得到 P0_PIN[0]的值 注1: x = 0~4, n = 0~7. 注2: 该寄存器的写操作不受寄存器 Px_DMASK[n]的影响.</p>

5.7.8 寄存器描述 (NUC029FAE)

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:				
GP_BA = 0x5000_4000				
P0_PMD	GP_BA+0x000	R/W	P0 I/O 模式控制	0x0000_XXXX
P0_OFFD	GP_BA+0x004	R/W	P0 关闭数字输入通路控制	0x0000_0000
P0_DOUT	GP_BA+0x008	R/W	P0 数据输出值	0x0000_00F3
P0_DMASK	GP_BA+0x00C	R/W	P0 数据输出写屏蔽	0x0000_0000
P0_PIN	GP_BA+0x010	R	P0 引脚数值	0x0000_00XX
P0_DBEN	GP_BA+0x014	R/W	P0 防抖动使能控制	0x0000_0000
P0_IMD	GP_BA+0x018	R/W	P0 中断模式控制	0x0000_0000
P0_IEN	GP_BA+0x01C	R/W	P0 中断使能控制	0x0000_0000
P0_ISRC	GP_BA+0x020	R/W	P0 中断源标志	0x0000_0000
P1_PMD	GP_BA+0x040	R/W	P1 I/O 模式控制	0x0000_XXXX
P1_OFFD	GP_BA+0x044	R/W	P1 关闭数字输入通路控制	0x0000_0000
P1_DOUT	GP_BA+0x048	R/W	P1 数据输出值	0x0000_003D
P1_DMASK	GP_BA+0x04C	R/W	P1 数据输出写屏蔽	0x0000_0000
P1_PIN	GP_BA+0x050	R	P1 引脚数值	0x0000_00XX
P1_DBEN	GP_BA+0x054	R/W	P1 防抖动使能控制	0x0000_0000
P1_IMD	GP_BA+0x058	R/W	P1 中断模式控制	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 中断使能控制	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 中断源标志	0x0000_0000
P2_PMD	GP_BA+0x080	R/W	P2 I/O 模式控制	0x0000_XXXX
P2_OFFD	GP_BA+0x084	R/W	P2 关闭数字输入通路控制	0x0000_0000
P2_DOUT	GP_BA+0x088	R/W	P2 数据输出值	0x0000_007C
P2_DMASK	GP_BA+0x08C	R/W	P2 数据输出写屏蔽	0x0000_0000
P2_PIN	GP_BA+0x090	R	P2 引脚数值	0x0000_00XX
P2_DBEN	GP_BA+0x094	R/W	P2 防抖动使能控制	0x0000_0000
P2_IMD	GP_BA+0x098	R/W	P2 中断模式控制	0x0000_0000

P2_IEN	GP_BA+0x09C	R/W	P2 中断使能控制	0x0000_0000
P2_ISRC	GP_BA+0x0A0	R/W	P2 中断源标志	0x0000_0000
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O 模式控制	0x0000_XXXX
P3_OFFD	GP_BA+0x0C4	R/W	P3 关闭数字输入通路控制	0x0000_0000
P3_DOUT	GP_BA+0x0C8	R/W	P3 数据输出值	0x0000_0077
P3_DMASK	GP_BA+0x0CC	R/W	P3 数据输出写屏蔽	0x0000_0000
P3_PIN	GP_BA+0xD0	R	P3 引脚数值	0x0000_00XX
P3_DBEN	GP_BA+0xD4	R/W	P3 防抖动使能控制	0x0000_0000
P3_IMD	GP_BA+0xD8	R/W	P3 中断模式控制	0x0000_0000
P3_IEN	GP_BA+0xDC	R/W	P3 中断使能控制	0x0000_0000
P3_ISRC	GP_BA+0xE0	R/W	P3 中断源标志	0x0000_0000
P4_PMD	GP_BA+0x100	R/W	P4 I/O 模式控制	0x0000_XXXX
P4_OFFD	GP_BA+0x104	R/W	P4 关闭数字输入通路控制	0x0000_0000
P4_DOUT	GP_BA+0x108	R/W	P4 数据输出值	0x0000_00C0
P4_DMASK	GP_BA+0x10C	R/W	P4 数据输出写屏蔽	0x0000_0000
P4_PIN	GP_BA+0x110	R	P4 引脚数值	0x0000_00XX
P4_DBEN	GP_BA+0x114	R/W	P4 防抖动使能控制	0x0000_0000
P4_IMD	GP_BA+0x118	R/W	P4 中断模式控制	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 中断使能控制	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 中断源标志	0x0000_0000
P5_PMD	GP_BA+0x140	R/W	P5 I/O 模式控制	0x0000_XXXX
P5_OFFD	GP_BA+0x144	R/W	P5 关闭数字输入通路控制	0x0000_0000
P5_DOUT	GP_BA+0x148	R/W	P5 数据输出值	0x0000_003F
P5_DMASK	GP_BA+0x14C	R/W	P5 数据输出写屏蔽	0x0000_0000
P5_PIN	GP_BA+0x150	R	P5 引脚数值	0x0000_00XX
P5_DBEN	GP_BA+0x154	R/W	P5 防抖动使能控制	0x0000_0000
P5_IMD	GP_BA+0x158	R/W	P5 中断模式控制	0x0000_0000
P5_IEN	GP_BA+0x15C	R/W	P5 中断使能控制	0x0000_0000
P5_ISRC	GP_BA+0x160	R/W	P5 中断源标志	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	中断去抖动控制	0x0000_0020

P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 引脚数据输入/输出	0x0000_0001
P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 引脚数据输入/输出	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 引脚数据输入/输出	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 引脚数据输入/输出	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 引脚数据输入/输出	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 引脚数据输入/输出	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 引脚数据输入/输出	0x0000_0001
P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 引脚数据输入/输出	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 引脚数据输入/输出	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 引脚数据输入/输出	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 引脚数据输入/输出	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 引脚数据输入/输出	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 引脚数据输入/输出	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 引脚数据输入/输出	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 引脚数据输入/输出	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 引脚数据输入/输出	0x0000_0001
P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 引脚数据输入/输出	0x0000_0001

5.7.9 寄存器描述 (NUC029FAE)

Port 0-5 I/O 模式控制 (Px_PMD)

寄存器	偏移量	R/W	描述	复位值
P0_PMD	GP_BA+0x000	R/W	P0 I/O 模式控制	0x0000_XXXX
P1_PMD	GP_BA+0x040	R/W	P1 I/O 模式控制	0x0000_XXXX
P2_PMD	GP_BA+0x080	R/W	P2 I/O 模式控制	0x0000_XXXX
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O 模式控制	0x0000_XXXX
P4_PMD	GP_BA+0x100	R/W	P4 I/O 模式控制	0x0000_XXXX
P5_PMD	GP_BA+0x140	R/W	P5 I/O 模式控制	0x0000_XXXX

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

位	描述	
[31:16]	悶隱	保留
[15 :0]	PMDn	<p>Port 0-5 I/O Pin[N] 模式控制</p> <p>决定Px.n每个引脚的I/O模式， 默认模式由CIOINI (CONFIG0[10])控制</p> <p>00 = Px.n为 输入模式.</p> <p>01 = Px.n 为推挽输出模式.</p> <p>10 = Px.n 为开漏模式.</p> <p>11 = Px.n 为准双向模式</p> <p>注1: x=0~5, n = 0~7</p> <p>注2:</p> <p>P0_PMD[7:0] 保留</p> <p>P1_PMD[15:12], [3:0] 保留</p> <p>P2_PMD[15:12], [7:0] 保留</p> <p>P3_PMD[15:12], [7:6], [3:0] 保留</p> <p>P4_PMD[11:0] 保留</p> <p>P5_PMD[15:4] 保留</p>

Port 0-5 数字输入通路关闭控制 (Px_OFFD)

寄存器	偏移量	R/W	描述	复位值
P0_OFFD	GP_BA+0x004	R/W	P0 数字输入通道关闭控制	0x0000_0000
P1_OFFD	GP_BA+0x044	R/W	P1 数字输入通道关闭控制	0x0000_0000
P2_OFFD	GP_BA+0x084	R/W	P2 数字输入通道关闭控制	0x0000_0000
P3_OFFD	GP_BA+0x0C4	R/W	P3 数字输入通道关闭控制	0x0000_0000
P4_OFFD	GP_BA+0x104	R/W	P4 数字输入通道关闭控制	0x0000_0000
P5_OFFD	GP_BA+0x144	R/W	P5 数字输入通道关闭控制	0x0000_0000

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

位	描述	
[31:24]	帳隱	保留
[23:16]	OFFD	<p>Port 0-5 Pin[N] 数字输入通道关闭控制</p> <p>该域每个位用来控制是否关闭相应Px.n引脚数字输入通路。如果某个引脚输入的是模拟信号，用户应该关闭相应引脚的数字通路避免漏电</p> <p>0 = 使能Px.n 数字输入通道 1 = 关闭Px.n 数字输入通道（数字输入连接到低电平）</p> <p>注1: x=0~5, n = 0~7 注2: P0_OFFD[19:16] 保留 P1_OFFD [23:22], [17:16] 保留 P2_OFFD [23:22], [19:16] 保留 P3_OFFD [23:22], [19], [17:16] 保留 P4_OFFD [21:16] 保留 P5_OFFD [23:18] 保留</p>
[15:0]	帳隱	保留

Port 0-5 数据输出值 (Px_DOUT)

寄存器	偏移量	R/W	描述	复位值
P0_DOUT	GP_BA+0x008	R/W	P0 数据输出值	0x0000_00F3
P1_DOUT	GP_BA+0x048	R/W	P1 数据输出值	0x0000_003D
P2_DOUT	GP_BA+0x088	R/W	P2 数据输出值	0x0000_007C
P3_DOUT	GP_BA+0x0C8	R/W	P3 数据输出值	0x0000_0077
P4_DOUT	GP_BA+0x108	R/W	P4 数据输出值	0x0000_00C0
P5_DOUT	GP_BA+0x148	R/W	P5 数据输出值	0x0000_003F



位	描述	
[31:8]	帳隱	保留
[7:0]	DOUT[n]	<p>Port 0-5 Pin[N] 输出值</p> <p>当Px.n配置成输出, 开漏, 或准双向模式时, 这些位控制Px.n引脚的状态</p> <p>0 = 当Px.n配置成输出, 开漏, 或准双向模式时, Px.n 引脚状态驱动为低电平 1 = 当Px.n配置成输出, 开漏, 或准双向模式时, Px.n 引脚状态驱动为高电平</p> <p>注1: x=0~5, n = 0~7</p> <p>注2:</p> <ul style="list-style-type: none"> P0_DOUT[3:0] 保留. P1_DOUT[7:6], [1:0] 保留 P2_DOUT[7:6], [3:0] 保留 P3_DOUT[7:6], [3], [1:0] 保留 P4_DOUT[5:0] 保留 P5_DOUT[7:2] 保留

Port0-5 数据输出写屏蔽 (Px_DMASK)

寄存器	偏移量	R/W	描述	复位值
P0_DMASK	GP_BA+0x00C	R/W	P0 数据输出写屏蔽	0x0000_0000
P1_DMASK	GP_BA+0x04C	R/W	P1 数据输出写屏蔽	0x0000_0000
P2_DMASK	GP_BA+0x08C	R/W	P2 数据输出写屏蔽	0x0000_0000
P3_DMASK	GP_BA+0x0CC	R/W	P3 数据输出写屏蔽	0x0000_0000
P4_DMASK	GP_BA+0x10C	R/W	P4 数据输出写屏蔽	0x0000_0000
P5_DMASK	GP_BA+0x14C	R/W	P5 数据输出写屏蔽	0x0000_0000



位	描述	
[31:8]	保留	保留
[7:0]	DMASK[n]	<p>Port 0-5 Pin [N] 数据输出写屏蔽</p> <p>这些位用来保护寄存器Px_DOUT[n]的相应位。当DMASK[n]位设置为“1”时，Px_DOUT[n]的相应位被保护。如果写信号被屏蔽，写数据到保护位将被忽略。</p> <p>0 = 相应的Px_DOUT [n] 位可以被更新 1 = 相应的Px_DOUT [n] 位受保护</p> <p>注1: x=0~5, n = 0~7 注2: 该功能只是保护相应的Px_DOUT[n]位，但是不保护Pxn_PDIO寄存器中的位 注3: P0_DMASK[3:0] 保留 P1_DMASK[7:6], [1:0] 保留 P2_DMASK[7:6], [3:0] 保留 P3_DMASK[7:6], [3], [1:0] 保留. P4_DMASK[5:0] 保留 P5_DMASK[7:2] 保留</p>

Port 0-5 引脚状态 (Px_PIN)

寄存器	偏移量	R/W	描述	复位值
P0_PIN	GP_BA+0x010	R	P0 引脚状态	0x0000_00XX
P1_PIN	GP_BA+0x050	R	P1 引脚状态	0x0000_00XX
P2_PIN	GP_BA+0x090	R	P2 引脚状态	0x0000_00XX
P3_PIN	GP_BA+0x0D0	R	P3 引脚状态	0x0000_00XX
P4_PIN	GP_BA+0x110	R	P4 引脚状态	0x0000_00XX
P5_PIN	GP_BA+0x150	R	P5 引脚状态	0x0000_00XX



位	描述	
[31:8]	帳隱	保留
[15:0]	PIN[n]	<p>Port 0-5 Pin [N] 引脚状态</p> <p>这些位的值反映各个Px.n 引脚的实际状态为高还是低，如果某个位的值为1，则相应引脚的状态为高电平，否则就是低电平</p> <p>注1: x=0~5, n = 0~7</p> <p>注 2:</p> <ul style="list-style-type: none"> P0_PIN[3:0] 保留 P1_PIN[7:6], [1:0] 保留 P2_PIN[7:6], [3:0] 保留 P3_PIN[7:6], [3], [1:0] 保留. P4_PIN[5:0] 保留. P5_PIN[7:2] 保留

Port 0-5 去抖动使能 (Px_DBEN)

寄存器	偏移量	R/W	描述	复位值
P0_DBEN	GP_BA+0x014	R/W	P0 去抖动使能控制	0x0000_0000
P1_DBEN	GP_BA+0x054	R/W	P1 去抖动使能控制	0x0000_0000
P2_DBEN	GP_BA+0x094	R/W	P2 去抖动使能控制	0x0000_0000
P3_DBEN	GP_BA+0x0D4	R/W	P3 去抖动使能控制	0x0000_0000
P4_DBEN	GP_BA+0x114	R/W	P4 去抖动使能控制	0x0000_0000
P5_DBEN	GP_BA+0x154	R/W	P5 去抖动使能控制	0x0000_0000



位	描述	
[31:8]	悵隱	保留
[15:0]	DBEN[n]	<p>Port 0-5 Pin [N] 输入信号去抖动使能控制</p> <p>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，则输入信号被视为信号抖动，从而不触发中断。去抖动时钟源由DBNCECON[4]控制，去抖动采样周期由DBNCECON[3:0]控制</p> <p>0 = 禁用 Px.n去抖动功能 1 = 使能 Px.n去抖动功能</p> <p>去抖动功能仅限于边沿触发中断，不能用于电平触发中断。如果中断模式为电平触发，去抖动使能位将被忽略。</p> <p>注1: x=0~5, n = 0~7</p> <p>注2: Px.n引脚如果作为睡眠唤醒源，需在进入睡眠模式之前，关闭Px.n引脚的去抖动功能。以避免在系统由Px.n去抖动功能唤醒之后，发生第二次中断事件。</p> <p>注3:</p> <p>P0_DBEN[3:0] 保留 P1_DBEN[7:6], [1:0] 保留 P2_DBEN[7:6], [3:0] 保留 P3_DBEN[7:6], [3], [1:0] 保留 P4_DBEN[5:0] 保留</p>

		P5_DBEN[7:2] 保留	
--	--	-----------------	--

Port 0-5 中断模式控制 (Px_IMD)

寄存器	偏移量	R/W	描述	复位值
P0_IMD	GP_BA+0x018	R/W	P0 中断模式控制	0x0000_0000
P1_IMD	GP_BA+0x058	R/W	P1 中断模式控制	0x0000_0000
P2_IMD	GP_BA+0x098	R/W	P2 中断模式控制	0x0000_0000
P3_IMD	GP_BA+0x0D8	R/W	P3 中断模式控制	0x0000_0000
P4_IMD	GP_BA+0x118	R/W	P4 中断模式控制	0x0000_0000
P5_IMD	GP_BA+0x158	R/W	P5 中断模式控制	0x0000_0000



位	描述	
[31:8]	帳隱	保留
[15:0]	IMD[n]	<p>Port 0-5 Pin [N] 边沿或电平检测中断模式控制</p> <p>IMD[n] 用于控制中断是电平触发还是边沿触发。若中断是边沿触发，触发源可以由去抖动控制，如果中断是电平触发，输入信号源由一个HCLK时钟采样并产生中断</p> <p>0 = 边沿触发中断 1 = 电平触发中断</p> <p>如果引脚设置为电平触发中断，寄存器Px_IEN中只有一种电平(高电平/低电平)可以被设定。若设置两种电平都触发中断，设置将被忽略，不会产生中断</p> <p>去抖动功能只对于边沿触发中断有效。如果中断模式为电平触发，去抖动使能位将被忽略。</p> <p>注1: x=0~5, n = 0~7 注 2:</p> <p>P0_IMD[3:0] 保留 P1_IMD[7:6], [1:0] 保留 P2_IMD[7:6], [3:0] 保留 P3_IMD[7:6], [3], [1:0] 保留。 P4_IMD[5:0] 保留 P5_IMD[7:2] 保留</p>

Port 0-5 中断使能控制 (Px_IEN)

寄存器	偏移量	R/W	描述	复位值
P0_IEN	GP_BA+0x01C	R/W	P0 中断使能控制	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 中断使能控制	0x0000_0000
P2_IEN	GP_BA+0x09C	R/W	P2 中断使能控制	0x0000_0000
P3_IEN	GP_BA+0x0DC	R/W	P3 中断使能控制	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 中断使能控制	0x0000_0000
P5_IEN	GP_BA+0x15C	R/W	P5 中断使能控制	0x0000_0000

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

位	描述	
[31:24]	悶隱	保留
[23:16]	IR_EN[n]	<p>Port 0-5 Pin [N] 输入上升沿或输入高电平中断使能</p> <p>IR_EN[n] 用于使能相应Px.n引脚的中断。置“1”也同时使能引脚唤醒功能</p> <p>当设置 IR_EN[n] 位为“1”时：</p> <p>如果中断是电平触发模式(IMD[n] = 1)，输入Px.n引脚的状态为高电平时，产生中断。</p> <p>如果中断是边沿触发模式(IMD[n] = 0)，输入Px.n引脚的状态由低电平到高电平变化时，产生中断</p> <p>0 = 禁用Px.n 高电平或由低电平到高电平变化的中断。</p> <p>1 = 使能Px.n 高电平或由低电平到高电平变化的中断</p> <p>注1： n=0~5, n = 0~7</p> <p>注 2：</p> <p>P0_IEN[19:16] 保留</p> <p>P1_IEN[23:22], [17:16] 保留</p> <p>P2_IEN[23:22], [19:16] 保留</p> <p>P3_IEN[23:22], [19], [17:16] 保留</p> <p>P4_IEN[21:16] 保留</p>

		P5_IEN[23:18] 保留
[15:8]	保留	
[7:0]	IF_EN[n]	<p>Port 0-5 Pin [N] 输入下降沿或输入低电平中断使能 IR_EN[n] 位用于使能相应Px.n引脚的中断。置“1”也同时使能引脚唤醒功能 当设置IF_EN[n]位为“1”时： 如果中断是电平触发模式(IMD[n] = 1)，输入Px.n引脚的状态为低电平时，产生中断。 如果中断是边沿触发模式(IMD[n] = 0)，输入Px.n引脚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1： x=0~5, n = 0~7 注 2： P0_IEN[3:0] 保留 P1_IEN[7:6], [1:0] 保留. P2_IEN[7:6], [3:0] 保留 P3_IEN[7:6], [3], [1:0] 保留 P4_IEN[5:0] 保留 P5_IEN[7:2] 保留 </p>

Port 0-5 中断源标志 (Px_ISRC)

寄存器	偏移量	R/W	描述	复位值
P0_ISRC	GP_BA+0x020	R/W	P0 中断源标志	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 中断源标志	0x0000_0000
P2_ISRC	GP_BA+0x0A0	R/W	P2 中断源标志	0x0000_0000
P3_ISRC	GP_BA+0x0E0	R/W	P3 中断源标志	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 中断源标志	0x0000_0000
P5_ISRC	GP_BA+0x160	R/W	P5 中断源标志	0x0000_0000

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

位	描述	
[31:8]	帳隱	保留
[7:0]	ISRC[n]	<p>Port 0-5 Pin [N] 中断源标志</p> <p>写：</p> <p>0 = 无动作 1 = 清相应的中断标志</p> <p>读：</p> <p>0 = Px.n没有中断 1 = Px.n产生中断</p> <p>注1： x=0~4, n = 0~7</p> <p>注2：</p> <p>P0_ISRC[3:0] 保留 P1_ISRC[7:6], [1:0] 保留. P2_ISRC[7:6], [3:0] 保留 P3_ISRC[7:6], [3], [1:0] 保留 P4_ISRC[5:0] 保留 P5_ISRC[7:2] 保留</p>

中断去抖动周期控制 (DBNCECON)

寄存器	偏移量	R/W	描述	复位值
DBNCECON	GP_BA+0x180	R/W	中断去抖动控制	0x0000_00020

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

位	描述	
[31:6]	帳隱	保留
[5]	ICLK_ON	<p>中断时钟打开模式 0 = 只有当I/O引脚的相应位Px_IEN 等于1时，边沿检测电路才被激活 1 = 所有I/O引脚的边沿检测电路在芯片复位后总是激活的 注：如果没有特别考量，建议关闭该位以节省系统功耗</p>
[4]	DBCLKSRC	<p>去抖动计数器时钟源选择 0 = 去抖动计数器时钟源为 HCLK 1 = 去抖动计数器时钟源为内部10KHz 内部低速时钟</p>
[3:0]	DBCLKSEL	<p>去抖动采样周期选择 0000 = 每 1 个时钟周期采样中断输入一次. 0001 = 每 2 个时钟周期采样中断输入一次. 0010 = 每 4 个时钟周期采样中断输入一次. 0011 = 每 8 个时钟周期采样中断输入一次. 0100 = 每16 个时钟周期采样中断输入一次. 0101 = 每32 个时钟周期采样中断输入一次. 0110 = 每64 个时钟周期采样中断输入一次. 0111 = 每128 个时钟周期采样中断输入一次. 1000 = 每256 个时钟周期采样中断输入一次. 1001 = 每2^*256 个时钟周期采样中断输入一次. 1010 = 每4^*256 个时钟周期采样中断输入一次. 1011 = 每8^*256 个时钟周期采样中断输入一次. 1100 = 每16^*256 个时钟周期采样中断输入一次. 1101 = 每32^*256 个时钟周期采样中断输入一次.</p>

		1110 = 每64*256 个时钟周期采样中断输入一次. 1111 = 每128*256 个时钟周期采样中断输入一次.
--	--	---

GPIO Px.n 引脚数据输入/输出 (Px[n]_PDIO)

P[x][n]_PDIO: x = 0~5, n = 0~7

寄存器	偏移量	R/W	描述	复位值
P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 引脚数据输入/输出	0x0000_0001
P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 引脚数据输入/输出	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 引脚数据输入/输出	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 引脚数据输入/输出	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 引脚数据输入/输出	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 引脚数据输入/输出	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 引脚数据输入/输出	0x0000_0001
P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 引脚数据输入/输出	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 引脚数据输入/输出	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 引脚数据输入/输出	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 引脚数据输入/输出	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 引脚数据输入/输出	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 引脚数据输入/输出	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 引脚数据输入/输出	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 引脚数据输入/输出	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 引脚数据输入/输出	0x0000_0001
P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 引脚数据输入/输出	0x0000_0001

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

位	描述
---	----

[31:1]	保留	
[0]	Pxn_PDIO	<p>GPIO Px.N 引脚数据输入/输出</p> <p>写该位可以控制一个GPIO引脚的输出值 0 = 设置相应的GPIO引脚为低 1 = 设置相应的GPIO引脚为高 读这个寄存器可以得到GPIO引脚的状态 0 = 相应的GPIO引脚状态为低 1 = 相应的GPIO引脚状态为高 例如：写到P04_DOUT的值，会反应到寄存器 P0_DOUT[4]中，读P04_PDIO将得到P0_PIN[4]的值</p> <p>注1：x = 0~5, n = 0~7. 注2：该寄存器的写操作不受寄存器 Px_DMASK[n]的影响.</p>

5.8 定时器控制器 (TIMER)

5.8.1 概述

定时器控制器包括4组32位的定时器，TIMER0~TIMER3，方便用户实现定时控制应用。定时器模块可支持例如频率测量，时间延迟，时钟产生，外部输入引脚的事件计数和外部捕捉引脚的间隔测量等功能。

5.8.2 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
- 每个定时器都有独立的时钟源
- 4种计数模式：单脉冲模式(one-shot)，周期模式(periodic)，反转输出模式(toggle)和连续计数(continuous counting)模式
- 超时周期= (定时器输入时钟源的周期) * (8-bit 预分频 + 1) * (24-bit TCMP)
- 最大计数周期 = $(1 / T \text{ MHz}) * (2^8) * (2^{24})$, T 是定时器时钟源的周期
- 24位上数计数器的值，可通过TDR (定时器数据寄存器) 读取
- 支持事件计数功能，可以对外部输入信号的事件计数(T0~T3)
- 24位捕获值可以通过TCAP(定时器捕获数据寄存器)读取
- 支持外部捕获引脚(T0EX~T3EX)用于间隔测量
- 支持外部捕获引脚(T0EX~T3EX)用于复位24位上数计数器
- 定时器中断支持将芯片从空闲/睡眠模式唤醒
- 当内部ACMP输出信号发生改变时可以触发定时器内部捕获(仅NUC029xAN支持)
- 支持Inter-Timer触发模式(仅NUC029xAN支持)
- 支持内部信号(CPO0, CPO1) 用于间隔测量(仅NUC029FAE支持)

5.8.3 框图

定时器控制器框图和时钟源控制如下所示

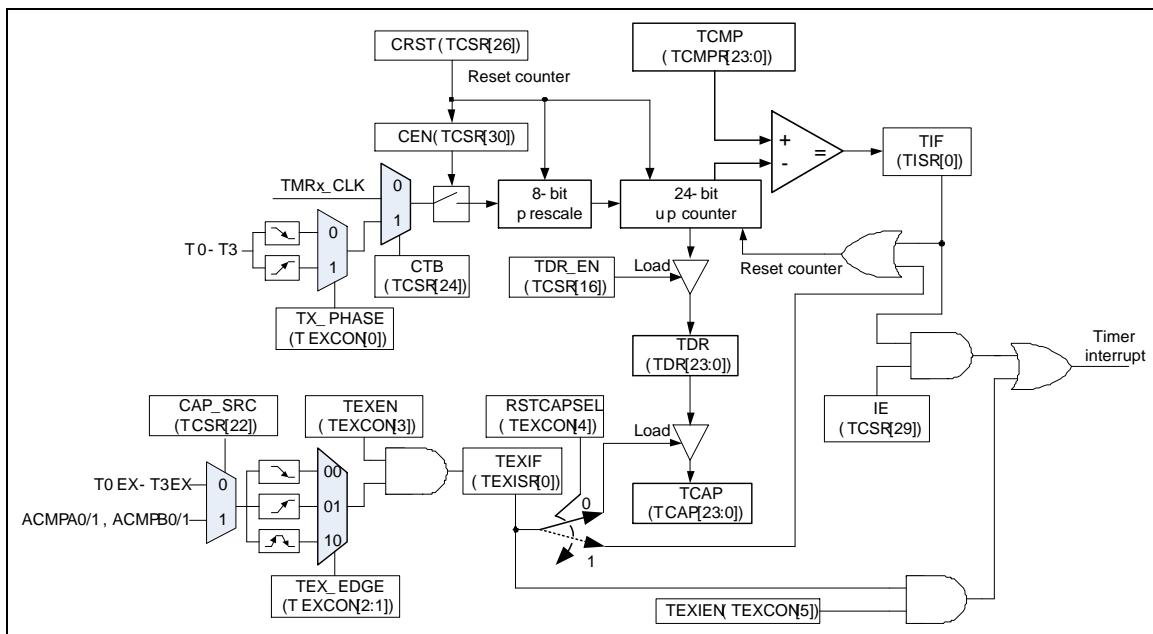


图 5-39 定时器控制器框图 (NUC029xAN)

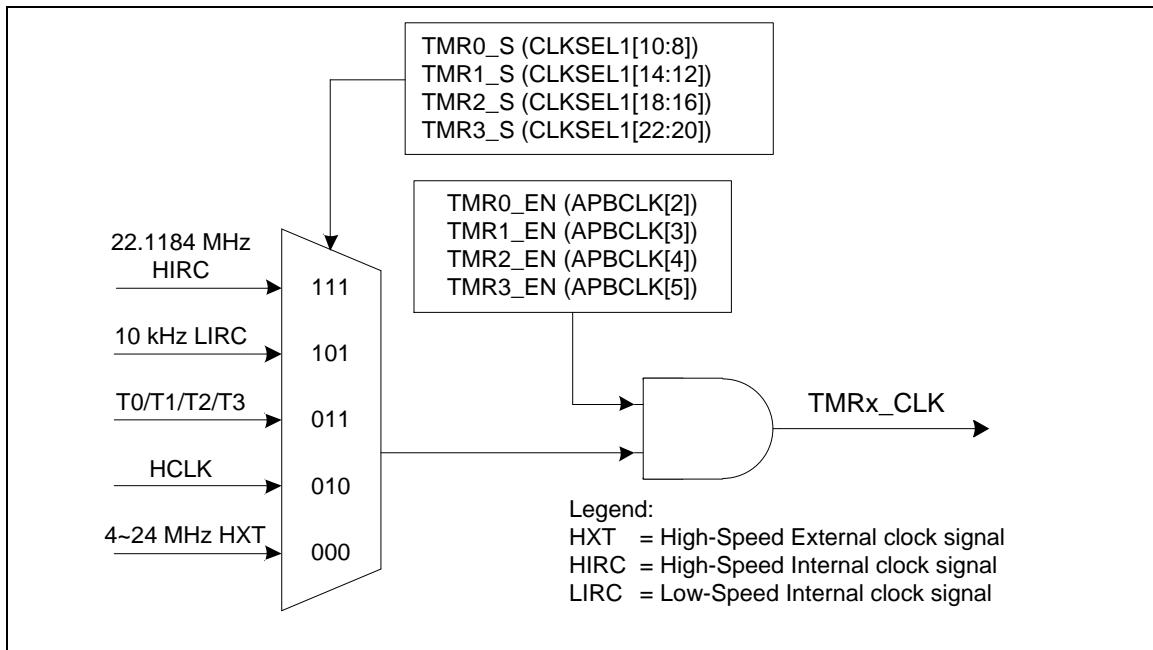


图 5-40 定时器控制器时钟源 (NUC029xAN)

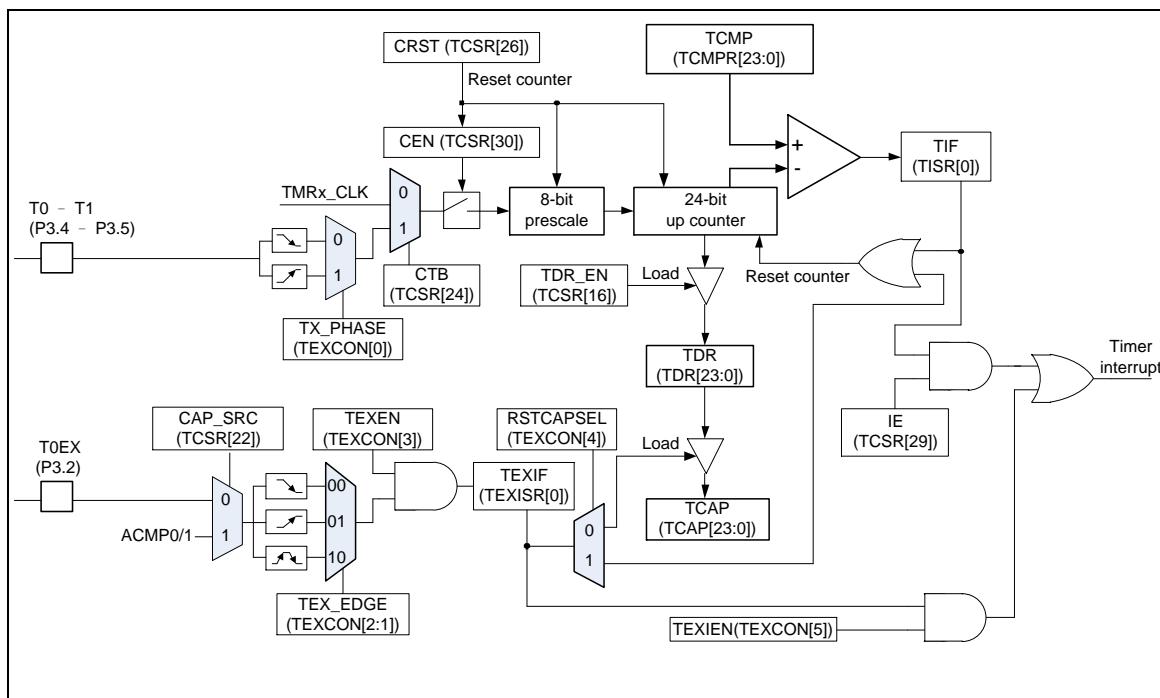


图 5-41 定时器控制器框图 (NUC029FAE)

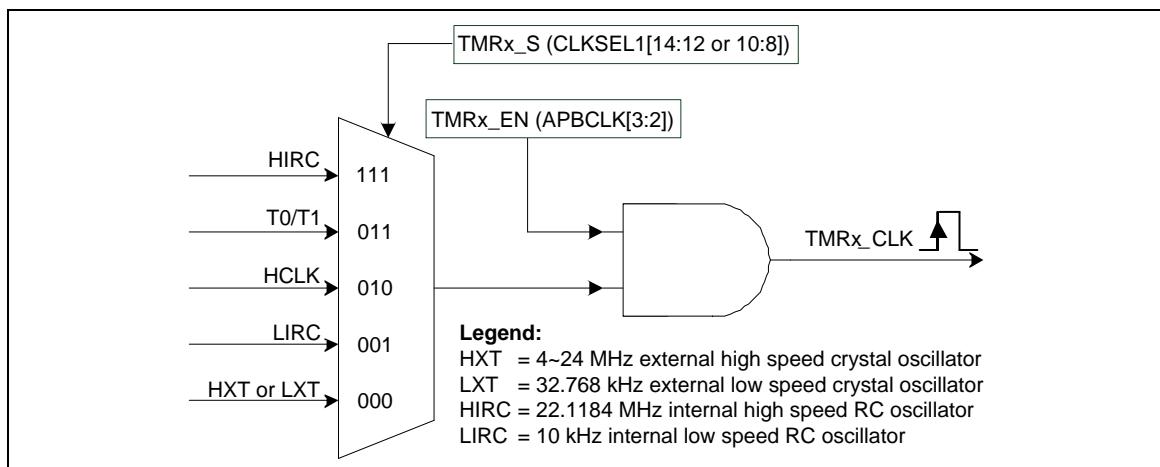


图 5-42 定时器控制器时钟源 (NUC029FAE)

5.8.4 基本配置

NUC029xAN:

Timer0 ~ Timer3 的时钟在 TMRx_EN(APBCLK[5:2]) 中使能，Timer0的时钟源在 TMR0_S(CLKSEL1[10:8])中选择，Timer1的时钟源在TMR1_S (CLKSEL1[14:12])中选择，Timer2的时钟源在TMR2_S(CLKSEL1[18:16])中选择，Timer3的时钟源在TMR3_S (CLKSEL1[22:20])中选择。

NUC029FAE:

定时器Timer0 ~ Timer1的时钟可以在寄存器TMRx_EN(APBCLK[3:2])使能，Timer0的时钟源在寄存器TMR0_S(CLKSEL1[10:8])中选择，Timer1的时钟源在寄存器TMR1_S(CLKSEL1[14:12])中选择。

5.8.5 功能描述

5.8.5.1 定时器中断标志

定时器控制器支持2种中断标志：一个是TIF(I2CTOC[0])标志，当计数器的值(TDR)等于定时器比较寄存器的值(TCMP(TCMR[23:0]))时被置位；另一个是TEXIF(TEXISR[0])标志，当TxEX引脚上的信号发生符合TEX_EDGE(TEXCONx[2:1])的设定的转变时被置位。

5.8.5.2 定时器操作模式

定时器控制器提供4种工作模式，单脉冲(one-shot)模式、周期(periodic)模式、反转输出(toggle)模式和连续计数(continuous counting)模式。

5.8.5.2.1 单脉冲模式

如果定时器工作在单脉冲模式(TCSR_x[28:27] = 00)且CEN(TCSR[30])位置1，定时器的计数器开始向上计数。一旦TDR的值达到定时器比较寄存器(TCMP)的值时，定时器中断标志TIF置位，TDR的值和CEN位被定时器控制器清0，然后定时器停止计数。此时，如果IE(TCSR[29]) (中断使能位)有置1，则将产生定时器中断信号，中断发生。

5.8.5.2.2 周期模式

如果定时器工作在周期模式(TCSR_x[28:27] = 01)且CEN(TCSR[30])位置1，定时器计数器开始向上计数。一旦TDR的值达到定时器比较寄存器(TCMP)的值时，定时器中断标志TIF置位，TDR的值将被定时器控制器清0，计数器再次开始计数。此时，如果IE有被使能，则将产生定时器中断信号，中断发生。该模式下定时器一直计数，并不断与TCMP比较，直到CEN位被清0。

5.8.5.2.3 反转输出模式

如果定时器工作在反转输出模式(TCSR_x[28:27] = 10)且CEN(TCSR[30])位置1，定时器计数器开始向上计数。反转输出模式和周期模式几乎相同，除了当TIF被置为1时反转输出将从T0 ~ T3输出信号，而且输出引脚上的信号将周期性发生反转，反转输出模式输出的信号占空比是50%。

如果是NUC029FAE，输出引脚可能是Tx或TxEX，具体使用哪个，取决于寄存器TCSR的TOUT_PIN(TCSR[18])位设置。

5.8.5.2.4 连续计数模式

如果定时器工作在连续计数模式(TCSR_x[28:27] = 11)且CEN(TCSR[30])位置1，定时器开始向上计

数。一旦TDR的值等于TCMP的值，TIF被置位并且TDR继续保持向上计数。同时，如果IE使能，则将产生定时器中断信号，中断发生。用户可以立即改变TCMP的值而无需关闭计数器并重新计数。

例如，TCMP的值先被设置为80，当TDR的值等于80时，TIF(定时器中断标志)将被置位，定时器计数器继续向上计数，但是TDR的值不会返回到零，而是继续计数81, 82, 83, ··· to $2^{24}-1$, 0, 1, 2, 3, ··· to $2^{24}-1$ ，不断的重复。接下来，如果用户设置TCMP为200，且清除TIF标志，当TDR的值达到200，TIF将再次被置位。最后，用户设置TCMP为500，并再一次清零TIF，当TDR的值达到500，TIF将再次被置位。

在该模式下，定时器计数是连续的，所以这种操作模式被称为连续计数模式。

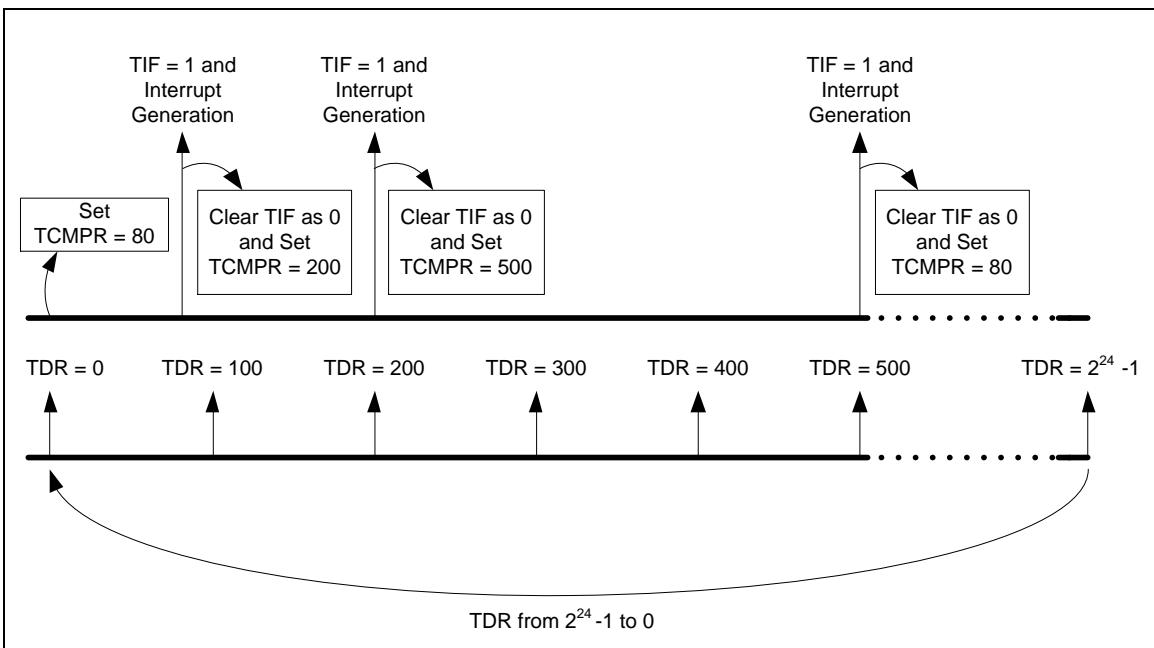


图 5-43 连续计数模式

5.8.5.3 事件计数模式

定时器提供一个功能可以用来数外部Tx引脚($x=0\sim3$)的事件个数，事件的个数反映在TDR寄存器中。也被称为事件计数功能。该功能中，CTB(TCSR[24])位必须被设为1，而且外设时钟源也必须选HCLK。

软件可以通过TCDB(TEXCON[7])位使能/关闭Tx引脚的防抖动功能。如果禁止防抖动功能，输入事件的频率应该小于HCLK频率的 $1/3$ ；如果使能防抖动功能，输入事件的频率应该小于HCLK频率的 $1/8$ 。否则，返回的TDR的值将不正确。软件通过TX_PHASE(TEXCON[0])位可以选择检测的Tx引脚的边沿相位。

事件计数模式下，定时器计数模式可以选择单脉冲模式、周期模式和连续计数模式来对Tx引脚上的输入事件计数，TDR为计数得到的值。

5.8.5.4 事件功能 (仅NUC029xAN)

事件捕获模式

计数器提供输入捕获功能，当TxEx引脚(x= 0~3)上的信号发生改变时，可以将捕获TDR的值存到TCAP寄存器中。在该模式，为了将TxEX引脚用作事件捕获功能，RSTCAPSEL(TEXCON[4])应该设为0并且定时器时钟源应该选择HCLK。

软件可以通过TEXDB(TEXCON[6])位使能/关闭TxEx引脚的防抖动功能。如果禁止防抖动功能，输入事件的频率应该小于HCLK 频率的1/3；如果使能防抖动功能，输入事件的频率应该小于HCLK 频率的1/8 。否则，捕获功能可能工作的不正常。软件通过EX_EDGE (TEXCON[2:1])位可以选择检测的TxEX引脚的边沿相位。

事件捕获模式下，软件不必考虑计数器工作在哪种模式下，只要TxEX引脚上的信号发生变化(符合用户设定)，捕获事件就会发生。

事件复位计数器模式

定时器提供复位计数器值功能，当TxEX引脚(x= 0~3)的信号发生边沿转变时，可以复位TDR的值。该模式下大多设定和事件捕获功能一样，除了RSTCAPSEL(TEXCON[4]) 应该设为1。

5.8.5.5 捕捉输入功能 (仅NUC029FAE)

输入捕捉或复位功能是用来捕捉或复位计数器的值。捕捉功能通过CAP_MODE(TEXCON[8])寄存器，可配置为自由计数和触发计数捕捉模式。自由计数捕捉模式,复位模式, 触发计数捕捉模式详细描述如下：输入源由CAP_SRC(TCSR[19])位设置，设定为是TxEX 还是比较器输出(CPOx)。

自由计数捕捉模式

如果CAP_MODE设置为0，TEXEN(TEXCON[3])设置为1，且RSTCAPSEL(TEXCON[4])设置为0，当TEX引脚(定时器外部引脚)触发条件发生时，TDR将被捕捉到TCAP寄存器。TEX的触发边沿由TEX_EDGE来设置。

复位模式

如果CAP_MODE设置为0，TEXEN(TEXCON[3])设置为1，且RSTCAPSEL设置为1，当TEX引脚(定时器外部引脚)触发条件发生时，TDR将被复位为0。TEX 的触发边沿由TEX_EDGE(TEXCON[2:1])来设置。详细操作方法如表表 5-16所描述。

触发计数捕捉模式

如果CAP_MODE设置为1，TEXEN(TEXCON[3])设置为1，且RSTCAPSEL设置为0，当TEX引脚(定时器外部引脚)触发条件发生时，TDR将被复位为0，同时TDR被捕捉到TCAP寄存器。TEX的触发边沿由TEX_EDGE来设置。详细操作方法如表表 5-16所描述

当TEX触发发生时，TEXIF(定时器外部中断标志)位被设置为1，同时如果TEXIEN(定时器外部中断使能位)设置为1，则产生中断信号，CPU中断发生。在TEX的消抖电路关闭时，TEX源操作频率必须少于1/3 HCLK频率，在TEX的消抖电路打开时，TEX源操作频率必须少于1/8 HCLK频率，以保证捕捉功能能够正常工作。可以通过设置TEXDB (TEXCON[6])使能或禁止TEX引脚的捕捉消抖功能。

功能	CAP_MODE (TEXCON[8])	RSTCAPN (TEXCON[4])	TEX_EDGE (TEXCON[2:1])	操作描述
自由计数捕捉模式	0	0	00	检测到定时器外部输入引脚由1变0时，TDR被捕捉到TCAP
	0	0	01	检测到定时器外部输入引脚由0变1时，TDR被捕捉到TCAP
	0	0	10	检测到定时器外部输入引脚由1变0或由0变1时，TDR被捕捉到TCAP
	0	0	11	保留
复位计数模式	0	1	00	检测到定时器外部输入引脚由1变0时。TDR复位为0。
	0	1	01	检测到定时器外部输入引脚由0变1时。TDR复位为0。
	0	1	10	检测到定时器外部输入引脚由1变0或由0变1时。TDR复位为0。
	0	1	11	保留
触发计数捕捉模式	1	0	00	下降沿触发： 检测到定时器外部输入引脚第一个由1变0跳变时，将复位TDR为0，然后开始计数，当第2个由1变0产生时，计数停止。
	1	0	01	上升沿触发： 检测到定时器外部输入引脚第一个由0变1，将复位TDR为0，然后开始计数，当第2个由0变1产生时，计数停止。
	1	0	10	电平切换触发： 检测到定时器外部输入引脚由1变0，将复位TDR为0，然后开始计数，当由0变1时，停止计数。
	1	0	11	电平切换触发： 检测到定时器外部输入引脚由0变1，将复位TDR为0，然后开始计数，当由1变0时，停止计数。

表 5-16 输入捕捉模式操作流程

5.8.5.6 Inter-Timer 触发捕获模式(仅NUC029xAN)

该模式下，Timer0/2 被迫使工作在计数模式，用来计数外部事件，并且会产生一个内部信号 (INTR_TMR_TRG) 来触发 Timer1/3 开始或者停止计数。Timer1/3 也被迫工作在捕获模式，由 Timer0/2 计数器状态触发开始/停止计数。

使能 Timer0 Inter-timer 触发捕获功能，Timer1 也被迫工作在触发-计数捕获功能；使能 Timer2 Inter-timer 触发捕获功能，Timer3 也被迫工作在触发-计数捕获功能。

启动触发

当Timer0/2 中的INTR_TRG_EN 被置位时，当Timer0/2 24-位计数器的值(TDR)从0x0变为0x1时，INTR_TMR_TRG信号将变为高电平，此时Timer1/3计数器将立即自动开始计数。

停止触发

当 Timer0/2 TDR 的值等于 Timer0/2 TCMPR 的值时，Timer0/2 的INTR_TMR_TRG 信号将变为低电平。然后 Timer0/2 计数器功能将被关闭并且INTR_TRG_EN 将被硬件清为0。之后Timer1/3 也将停止计数。同时，Timer1/3 TDR 的值将被存到 Timer1/3 TCAP 寄存器中。

用户可以使用 inter-timer 触发功能来更精确的测量外部事件(Tx)的周期。下图 5-44显示了Timer0 Inter-Timer触发捕获模式的参考流程，此模式下，Timer0工作在事件计数模式，Timer1当作触发计数捕获模式。

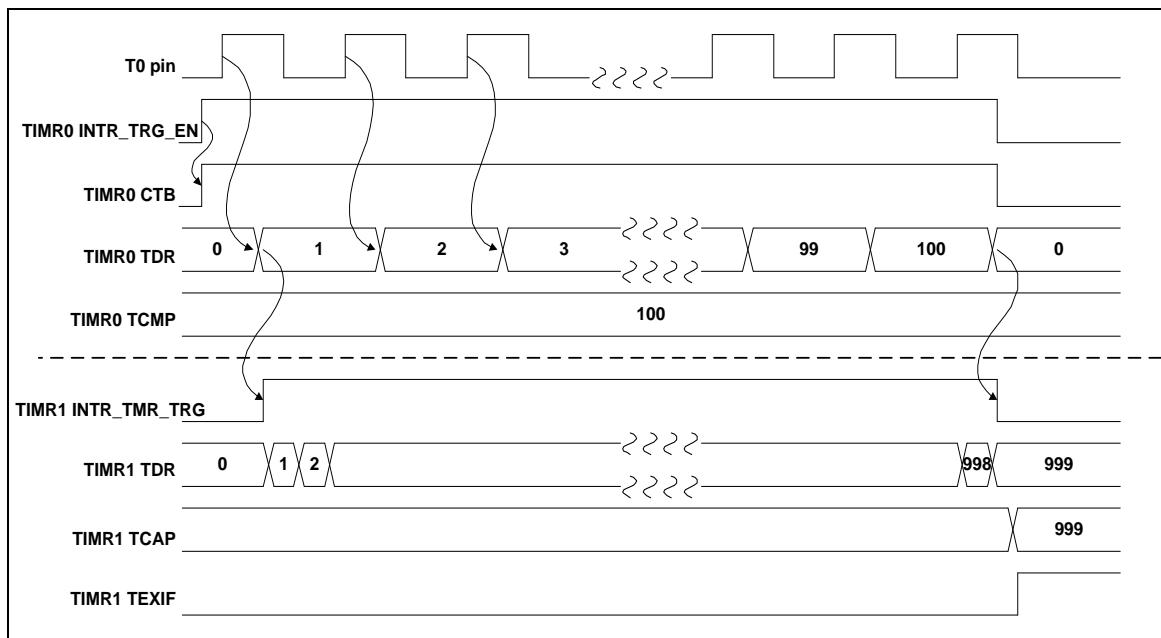


图 5-44 Inter-Timer 触发捕获时序图

5.8.5.7 来自ACMP的内部捕获触发(仅NUC029xAN)

事件捕获功能也可以由内部输出信号ACMPA0(Timer0), ACMPA1(Timer1), ACMPB0(Timer2) 或者 ACMPB1(Timer3)的反转触发。捕获功能的详细设定和事件捕获模式一样。

5.8.6 寄存器映射 (NUC029xAN)

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 = 0x4001_0000				
TMR23_BA = 0x4011_0000				
TCSR0	TMR01_BA+0x00	R/W	Timer0 控制和状态寄存器	0x0000_0005
TCMPR0	TMR01_BA+0x04	R/W	Timer0 比较寄存器	0x0000_0000
TISR0	TMR01_BA+0x08	R/W	Timer0 中断状态寄存器	0x0000_0000
TDR0	TMR01_BA+0x0C	R	Timer0 数据寄存器	0x0000_0000
TCAP0	TMR01_BA+0x10	R	Timer0 捕捉数据寄存器	0x0000_0000
TEXCON0	TMR01_BA+0x14	R/W	Timer0 外部控制寄存器	0x0000_0000
TEXISR0	TMR01_BA+0x18	R/W	Timer0 外部中断状态寄存器	0x0000_0000
TCSR1	TMR01_BA+0x20	R/W	Timer1 控制和状态寄存器	0x0000_0005
TCMPR1	TMR01_BA+0x24	R/W	Timer1 比较寄存器	0x0000_0000
TISR1	TMR01_BA+0x28	R/W	Timer1 中断状态寄存器	0x0000_0000
TDR1	TMR01_BA+0x2C	R	Timer1 数据寄存器	0x0000_0000
TCAP1	TMR01_BA+0x30	R	Timer1 捕捉数据寄存器	0x0000_0000
TEXCON1	TMR01_BA+0x34	R/W	Timer1 外部控制寄存器	0x0000_0000
TEXISR1	TMR01_BA+0x38	R/W	Timer1 外部中断状态寄存器	0x0000_0000
TCSR2	TMR23_BA+0x00	R/W	Timer2 控制和状态寄存器	0x0000_0005
TCMPR2	TMR23_BA+0x04	R/W	Timer2 比较寄存器	0x0000_0000
TISR2	TMR23_BA+0x08	R/W	Timer2 中断状态寄存器	0x0000_0000
TDR2	TMR23_BA+0x0C	R	Timer2 数据寄存器	0x0000_0000
TCAP2	TMR23_BA+0x10	R	Timer2 捕捉数据寄存器	0x0000_0000
TEXCON2	TMR23_BA+0x14	R/W	Timer2 外部控制寄存器	0x0000_0000
TEXISR2	TMR23_BA+0x18	R/W	Timer2 外部中断状态寄存器	0x0000_0000
TCSR3	TMR23_BA+0x20	R/W	Timer3 控制和状态寄存器	0x0000_0005
TCMPR3	TMR23_BA+0x24	R/W	Timer3 比较寄存器	0x0000_0000
TISR3	TMR23_BA+0x28	R/W	Timer3 中断状态寄存器	0x0000_0000
TDR3	TMR23_BA+0x2C	R	Timer3 数据寄存器	0x0000_0000

TCAP3	TMR23_BA+0x30	R	Timer3 捕捉数据寄存器	0x0000_0000
TEXCON3	TMR23_BA+0x34	R/W	Timer3 外部控制寄存器	0x0000_0000
TEXISR3	TMR23_BA+0x38	R/W	Timer3 外部中断状态寄存器	0x0000_0000

5.8.7 寄存器描述 (NUC029xAN)

定时器控制寄存器 (TCSR)

寄存器	偏移量	R/W	描述				复位值
TCSR0	TMR01_BA+0x00	R/W	Timer0 控制和状态寄存器				0x0000_0005
TCSR1	TMR01_BA+0x20	R/W	Timer1 控制和状态寄存器				0x0000_0005
TCSR2	TMR23_BA+0x00	R/W	Timer2 控制和状态寄存器				0x0000_0005
TCSR3	TMR23_BA+0x20	R/W	Timer3 控制和状态寄存器				0x0000_0005

31	30	29	28	27	26	25	24
DBGACK_TMR	CEN	IE	MODE		CRST	CACT	CTB
23	22	21	20	19	18	17	16
WAKE_EN	CAP_SRC	TOUT_SEL	PERIODIC_SEL	INTR_TRG_EN	帳隱		TDR_EN
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
PRESCALE							

位	描述	
[31]	DBGACK_TMR	仿真器 (ICE) 调试模式响应禁止控制位（写保护位） 0 = 仿真器调试模式响应影响定时器计数。 当CPU被执行调试模式时，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CPU是否执行调试模式时，定时器计数器将持续计数下去。
[30]	CEN	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CEN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(TCSR[28:27] = 00)，当相应的定时中断标志TIF产生时，该位由硬件自动清零。
[29]	IE	中断使能位 0 = 禁用定时器中断 1 = 使能定时器中断 注意：如果该位使能，当定时器中断标志TIF置1时，定时器中断信号产生并通知CPU。
[28:27]	MODE	定时器计数模式选择

		00 = 定时器工作在单脉冲模式 (one-shot) 01 = 定时器工作在周期模式 (Periodic) 10 = 定时器工作在 Toggle 模式 11 = 定时器工作在连续计数模式(Continuous Counting)
[26]	CRST	定时器复位位 0 = 该位写 0 无效 1 = 如果CACT = 1, 复位8位预分频计数器, 24位向上计数器的值和CEN位
[25]	CACT	定时器激活状态位 (只读) 该位表示24位向上计数器的状态。 0 = 定时器计数器未激活 1 = 定时器计数器激活
[24]	CTB	外部事件计数模式使能位 该位用来使能外部计数管脚功能。当定时器用作事件计数器, 该位需要设置成1, 并选择HCLK作为定时器时钟源, 详细请参看5.8.5.3章节 0 = 禁用外部事件计数器模式 1 = 使能外部事件计数器模式
[23]	WAKE_EN	唤醒功能使能位 0 = 唤醒触发事件禁止 1 = 唤醒触发事件使能
[22]	CAP_SRC	捕获引脚源选择 0 = 捕获功能来自TxEX引脚. 1 = 捕获功能来自内部 ACMPx 输出信号.
[21]	TOUT_SEL	反转输出引脚选择 0 = 反转输出到Tx 引脚. 1 = 反转输出到TxEx 引脚
[20]	PERIODIC_SEL	周期模式行为选择使能 0 = 周期模式行为选择被关闭. 周期模式下当定时器正在工作, 此时用户更新 TCMP, TDR将被复位成默认值。 1 = 使能周期模式行为选择. 周期模式下当定时器正在工作, 此时用户更新 TCMP, 定时器行为如下： 如果更新的TCMP 的值 > TDR , TCMP 将被更新并且TDR继续计数。 如果更新的TCMP 的值 = TDR , 定时器超时中断将立即发生。 如果更新的TCMP 的值< TDR , TDR将被复位成默认值。
[19]	INTR_TRG_EN	Inter-Timer 触发模式使能控制 设置该位将使能 inter-timer 触发捕获功能. Timer0/2 将工作在计数模式, 对外部时钟源或事件计数, Timer1/3 将工作在捕获功能的触发计数模式. 0 = 禁止Inter-Timer 触发捕获模式. 1 = 使能Inter-Timer 触发捕获模式. 注: 对于Timer1/3, 该位被忽略并且读到的值总是0.
[18:17]	帳隱	保留

[16]	TDR_EN	数据加载使能控制 当置位TDR_EN, TDR (定时器数据寄存器) 将不断更新为24位向上计数器的值 0 = 禁止定时器数据寄存器 更新 1 = 当定时器计数器激活时, 使能定时器数据寄存器 更新
[15:8]	保留	
[7:0]	PRESCALE	预分频计数器 定时器的输入时钟源被(PRESCALE+1)预分频, 然后再输入到定时器。如果该域为0(PRESCALE =0), 则不进行预分频.

定时器比较寄存器 (TCMPR)

寄存器	偏移量	R/W	描述	复位值
TCMPR0	TMR01_BA+0x04	R/W	Timer0 比较寄存器	0x0000_0000
TCMPR1	TMR01_BA+0x24	R/W	Timer1 比较寄存器	0x0000_0000
TCMPR2	TMR23_BA+0x04	R/W	Timer2 比较寄存器	0x0000_0000
TCMPR3	TMR23_BA+0x24	R/W	Timer3 比较寄存器	0x0000_0000

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

位	描述	
[31:24]	悵隱	保留
[23:0]	TCMP	<p>定时器比较值</p> <p>TCMP是24位比较寄存器。当内部24位向上计数器的值等于TCMP的值时, TIF将被置为1。</p> <p>定时溢出周期= (定时器输入时钟周期) * (8-位PRESCALE + 1) * (24-位TCMP)</p> <p>注1: 不能在TCMP里写0x0或0x1, 否则内核将运行到未知状态。</p> <p>注2: 如果定时器工作在连续计数模式, 即使用户写一个新的值到TCMP, 24位上数计数器还将保持继续向上计数; 如果计数器工作在其它模式除了周期模式, 如果软件向该寄存器写入新的值, 24-bit上数计数器将使用新的TCMP值, 并重新开始计数。</p>

定时器中断状态寄存器 (TISR)

寄存器	偏移量	R/W	描述	复位值
TISR0	TMR01_BA+0x08	R/W	Timer0 中断状态寄存器	0x0000_0000
TISR1	TMR01_BA+0x28	R/W	Timer1 中断状态寄存器	0x0000_0000
TISR2	TMR23_BA+0x08	R/W	Timer2 中断状态寄存器	0x0000_0000
TISR3	TMR23_BA+0x28	R/W	Timer3 中断状态寄存器	0x0000_0000

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

位	描述	
[31:2]	悵隱	保留
[1]	TWF 定时器唤醒标志 该位指示定时器状态的中断唤醒标志 0 = 定时器没有导致芯片唤醒。 1 = 如果定时器产生超时中断信号，芯片从空闲/睡眠模式唤醒。 注：该位写1清0	
[0]	TIF 定时器中断标志 该位指示当内部24位计数器的值达到TCMP的值时定时器的中断状态标志 0 = 无影响 1 = TDR的值等于TCMP的值 注：该位写1清0	

定时器数据寄存器 (TDR)

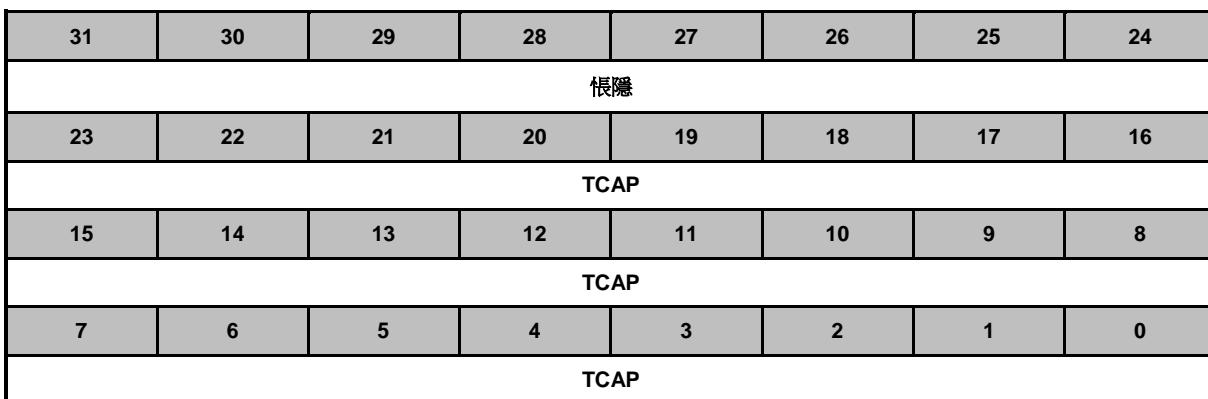
寄存器	偏移量	R/W	描述	复位值
TDR0	TMR01_BA+0x0C	R	Timer0 数据寄存器	0x0000_0000
TDR1	TMR01_BA+0x2C	R	Timer1 数据寄存器	0x0000_0000
TDR2	TMR23_BA+0x0C	R	Timer2 数据寄存器	0x0000_0000
TDR3	TMR23_BA+0x2C	R	Timer3 数据寄存器	0x0000_0000

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

位	描述	
[31:24]	保留	保留
[23:0]	TDR	定时器数据寄存器 如果TDR_EN 置1， 用户可以读TDR寄存器取得当前24位上数计数器的值

定时器捕捉数据寄存器 (TCAP)

寄存器	偏移地址	R/W	描述	复位值
TCAP0	TMR01_BA+0x10	R	Timer0 捕捉数据寄存器	0x0000_0000
TCAP1	TMR01_BA+0x30	R	Timer1 捕捉数据寄存器	0x0000_0000
TCAP2	TMR23_BA+0x10	R	Timer2 捕捉数据寄存器	0x0000_0000
TCAP3	TMR23_BA+0x30	R	Timer3 捕捉数据寄存器	0x0000_0000



位	描述	
[31:24]	保留	保留
[23:0]	TCAP	定时器捕捉数据寄存器 当TEXIF标志被置1时，当前TDR的值将自动立即被加载到 TCAP域。

定时器外部控制寄存器 (TEXCON)

寄存器	偏移量	R/W	描述	复位值
TEXCON0	TMR01_BA+0x14	R/W	Timer0 外部控制寄存器	0x0000_0000
TEXCON1	TMR01_BA+0x34	R/W	Timer1 外部控制寄存器	0x0000_0000
TEXCON2	TMR23_BA+0x14	R/W	Timer2 外部控制寄存器	0x0000_0000
TEXCON3	TMR23_BA+0x34	R/W	Timer3 外部控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
悶隱							
7	6	5	4	3	2	1	0
TCDB	TEXDB	TEXIEN	RSTCAPSEL	TEXEN	TEX_EDGE		TX_PHASE

位	描述	
[31:8]	悶隱	保留
[7]	TCDB	<p>定时器外部计数输入引脚防抖动使能控制 0 = 关闭Tx引脚防抖动功能 1 = 使能Tx引脚防抖动功能 如果这个位 被使能, Tx引脚的输入信号边沿将被防抖动电路检测.</p>
[6]	TEXDB	<p>定时器外部捕获输入引脚防抖动使能控制 0 = 关闭TxEX引脚防抖动功能 1 = 使能TxEX引脚防抖动功能 如果这个位 被使能, TxEX 引脚的输入信号边沿将被防抖动电路检测</p>
[5]	TEXIEN	<p>定时器外部捕获中断使能控制 0 = 禁止TxEX引脚检测中断 1 = 使能TxEX引脚检测中断 如果TEXIEN被使能, 当TEXIF标志被置位时, 将产生外部捕获中断信号, CPU中断发生</p>
[4]	RSTCAPSEL	<p>定时器外部复位计数器/外部捕获模式选择 0 = TxEX 引脚上的信号转变被用作定时器捕获功能, 如果.TEXIF标志被设置为1, TDR值将存到TCAP中。 1 = TxEX 引脚上的信号转变被用来复位24位向上计数器.</p>
[3]	TEXEN	<p>定时器外部引脚功能使能 该位用来使能TxEX引脚上的RSTCAPSEL功能.</p>

		0 = TxEX 引脚上的RSTCAPSEL功能被忽略. 1 = TxEX引脚上的RSTCAPSEL功能有效
[2:1]	TEX_EDGE	定时器外部捕获引脚边沿检测选择 00 = TxEX引脚上1到 0 的转变将被检测. 01 = TxEX引脚上0到 1 的转变将被检测. 10 = TxEX引脚上1到 0或0到1 的转变都将被检测. 11 = 保留.
[0]	TX_PHASE	定时器外部计数引脚相位 检测选择 这个位用来指示检测的Tx引脚的信号相位. 0 = Tx引脚上一个下降沿转变将被计数. 1 = Tx引脚上一个上升沿转变将被计数.

定时器外部中断状态寄存器 (TEXISR)

寄存器	偏移地址	R/W	描述	复位值
TEXISR0	TMR01_BA+0x18	R/W	Timer0 外部中断状态寄存器	0x0000_0000
TEXISR1	TMR01_BA+0x38	R/W	Timer1 外部中断状态寄存器	0x0000_0000
TEXISR2	TMR23_BA+0x18	R/W	Timer2 外部中断状态寄存器	0x0000_0000
TEXISR3	TMR23_BA+0x38	R/W	Timer3 外部中断状态寄存器	0x0000_0000

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

位	描述	
[31:1]	悵隱	保留
[0]	TEXIF	<p>定时器外部捕获中断标志</p> <p>这个位指示定时器外部捕获中断标志状态。</p> <p>当TEXEN使能时，TxEX 引脚选择为外部捕获功能。此时TxEX 引脚上发生的转变与TEX_EDGE的设定信号一致时，该位将由硬件置位。.</p> <p>例如，TEXEN = 1, TEX_EDGE = 00, TEX引脚上一个 1 到 0的转变将导致TEXIF 被置。</p> <p>0 = TxEX引脚中断没有发生 1 = TxEX引脚中断发生</p> <p>注：该位写1清0.</p>

5.8.8 寄存器映射 (NUC029FAE)

R: 只读, W:只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位值
TMR 基地址				
TMR_BA = 0x4001_0000				
TCSR0	TMR_BA+0x00	R/W	Timer0 控制和状态寄存器	0x0000_0005
TCMPR0	TMR_BA+0x04	R/W	Timer0 比较寄存器	0x0000_0000
TISR0	TMR_BA+0x08	R/W	Timer0 中断状态寄存器	0x0000_0000
TDR0	TMR_BA+0x0C	R	Timer0 数据寄存器	0x0000_0000
TCAP0	TMR_BA+0x10	R	Timer0 捕捉数据寄存器	0x0000_0000
TEXCON0	TMR_BA+0x14	R/W	Timer0 外部控制寄存器	0x0000_0000
TEXISR0	TMR_BA+0x18	R/W	Timer0 外部中断状态寄存器	0x0000_0000
TCSR1	TMR_BA+0x20	R/W	Timer1 控制和状态寄存器	0x0000_0005
TCMPR1	TMR_BA+0x24	R/W	Timer1 比较寄存器	0x0000_0000
TISR1	TMR_BA+0x28	R/W	Timer1 中断状态寄存器	0x0000_0000
TDR1	TMR_BA+0x2C	R	Timer1 数据寄存器	0x0000_0000
TCAP1	TMR_BA+0x30	R	Timer1 捕捉数据寄存器	0x0000_0000
TEXCON1	TMR_BA+0x34	R/W	Timer1 外部控制寄存器	0x0000_0000
TEXISR1	TMR_BA+0x38	R/W	Timer1 外部中断状态寄存器	0x0000_0000

5.8.9 寄存器描述 (NUC029FAE)

定时器控制寄存器 (TCSR)

寄存器	偏移量	R/W	描述			复位值
TCSR0	TMR_BA+0x00	R/W	Timer0 控制和状态寄存器			0x0000_0005
TCSR1	TMR_BA+0x20	R/W	Timer1 控制和状态寄存器			0x0000_0005

31	30	29	28	27	26	25	24
DBGACK_TMR	CEN	IE	MODE		CRST	CACT	CTB
23	22	21	20	19	18	17	16
WAKE_EN	帳隱			CAP_SRC	TOUT_PIN	PERIODIC_SEL	TDR_EN
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
PRESCALE							

位	描述	
[31]	DBGACK_TMR	仿真器 (ICE) 调试模式响应禁止控制位（写保护位） 0 = 仿真器调试模式响应影响定时器计数。 当CPU被执行调试模式时，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CPU是否执行调试模式时，定时器计数器将持续计数下去。
[30]	CEN	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CEN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(TCSR[28:27] = 00)，当相应的定时中断标志TIF产生时，该位由硬件自动清零。
[29]	IE	中断使能位 0 = 禁用定时器中断 1 = 使能定时器中断 注意：如果该位使能，当定时器中断标志TIF置1时，定时器中断信号产生并通知CPU。
[28:27]	MODE[1:0]	定时器操作模式选择 模式 定时器操作模式 00 定时器操作在One-shot模式。相应的中断发生一次(如果IE使能)，CEN由硬件自动清0.

位	描述		
		01	定时器操作在周期模式。相应的中断周期性发生(如果IE使能)
		10	定时器操作在反转输出模式。中断信号周期性发生(如果IE使能)。相应的输出信号(tout)来回切换，占空比50%。
		11	定时器操作在连续计数模式。当TDR = TCMR时(如果IE使能)中断将发生。然而，24位向上计数器继续计数。详细请参考5.8.5.2.4节连续计数模式的描述
[26]	CRST	定时器复位位 0 = 无作用 1 = 如果CACT = 1, 复位8位预分频计数器, 24位向上计数器的值和CEN位	
[25]	CACT	定时器激活状态位 (只读) 该位表示24位向上计数器的状态。 0 = 定时器计数器未激活 1 = 定时器计数器激活	
[24]	CTB	计数模式使能位 该位用来使能外部计数管脚功能。当定时器用作事件计数器, 该位需要设置成1, 并选择HCLK作为定时器时钟源, 详细请参看5.8.5.3章节 0 = 禁用外部事件计数器模式 1 = 使能外部事件计数器模式	
[23]	WAKE_EN	唤醒功能使能位 如果WAKE_EN (UA_IER[6])置1, 且TIF 或 TEXIF (TEXISR[0])置位时, 定时器将产生一个唤醒触发事件给CPU 0 = 唤醒触发事件禁止 1 = 唤醒触发事件使能	
[22:20]	保留		
[19]	CAP_SRC	捕捉管脚源选择 TCSR0: 0 = 捕捉功能的源是来自TOEX管脚 1 = 捕捉功能的源是来自内部ACMP0输出信号 TCSR1: 0 = 保留 1 = 捕捉功能的源是来自内部ACMP1输出信号	
[18]	TOUT_PIN	反转输出引脚选择 TCSR0: 0 = 反转输出引脚为T0 引脚 1 = 反转输出引脚为TOEX 引脚 TCSR1: 0 = 反转输出引脚为T1 引脚 1 = 保留	
[17]	PERIODIC_SEL	周期模式行为选择使能	

位	描述
	<p>0 = 在One-shot或周期模式下，当用户更新 TCMP，定时器计数器将复位。 1 = 在One-shot或周期模式下，当用户更新 TCMP，定时器行为如下： 如果更新的TCMP 的值 > TDR，定时器计数器将保持继续计数而且不复位。 如果更新的TCMP 的值 <= TDR，定时器计数器将复位。</p>
[16]	<p>TDR_EN</p> <p>数据加载使能控制</p> <p>当置位TDR_EN， TDR (定时器数据寄存器) 将不断更新为24位向上计数器的值 0 = 禁止定时器数据寄存器更新 1 = 当定时器计数器激活时，使能定时器数据寄存器 更新</p>
[15:8]	<p>保留</p>
[7:0]	<p>PRESCALE [7:0]</p> <p>预分频计数器</p> <p>定时器的输入时钟源被(PRESCALE+1)预分频，然后再输入到定时器。如果该域为0(PRESCALE =0)，则不进行预分频.</p>

定时器比较寄存器 (TCMPR)

寄存器	偏移量	R/W	描述	复位值
TCMPR0	TMR_BA+0x04	R/W	Timer0 比较寄存器	0x0000_0000
TCMPR1	TMR_BA+0x24	R/W	Timer1比较寄存器	0x0000_0000

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

位	描述	
[31:24]	保留	保留
[23:0]	TCMP[23:0]	<p>定时器比较值</p> <p>TCMP是24位比较寄存器。当内部24位向上计数器的值等于TCMP的值时, TIF将被置为1。</p> <p>定时溢出周期= (定时器输入时钟周期) * (8-位PRESCALE + 1) * (24-位TCMP)</p> <p>注: 不能在TCMP里写0x0或0x1, 否则内核将运行到未知状态.</p>

定时器中断状态寄存器 (TISR)

寄存器	偏移量	R/W	描述	复位值
TISR0	TMR_BA+0x08	R/W	Timer0 中断状态寄存器	0x0000_0000
TISR1	TMR_BA+0x28	R/W	Timer1 中断状态寄存器	0x0000_0000

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

位	描述	
[31:2]	帳隱	保留
[1]	TWF	<p>定时器唤醒标志 该位指示定时器状态的中断唤醒标志 0 = 定时器没有导致芯片唤醒。 1 = 如果定时器产生超时中断信号，芯片从空闲/睡眠模式唤醒。 注：该位写1清0</p>
[0]	TIF	<p>定时器中断标志 该位指示当TDR值达到TCMP的值时定时器的中断标志状态 0 = 无影响 1 = TDR的值等于TCMP的值 注：该位写1清0</p>

定时器数据寄存器 (TDR)

寄存器	偏移量	R/W	描述	复位值
TDR0	TMR_BA+0x0C	R	Timer0 数据寄存器	0x0000_0000
TDR1	TMR_BA+0x2C	R	Timer1 数据寄存器	0x0000_0000

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

位	描述	
[31:24]	保留	保留
[23:0]	TDR[23:0]	如果TDR_EN (TCSR[16])置1， 用户可以读TDR寄存器取得当前24位上数计数器的值

定时器捕捉数据寄存器 (TCAP)

寄存器	偏移量	R/W	描述	复位值
TCAP0	TMR_BA+0x10	R	Timer0 捕捉数据寄存器	0x0000_0000
TCAP1	TMR_BA+0x30	R	Timer1 捕捉数据寄存器	0x0000_0000

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

位	描述	
[31:24]	悵隱	保留
[23:0]	TCAP[23:0]	定时器捕捉数据寄存器 当TEXIF标志被置1时，当前TDR的值将自动立即被加载到 TCAP域

定时器外部控制寄存器 (TEXCON)

寄存器	偏移量	R/W	描述	复位值
TEXCON0	TMR_BA+0x14	R/W	Timer0 外部控制寄存器	0x0000_0000
TEXCON1	TMR_BA+0x34	R/W	Timer1 外部控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
悶隱							
7	6	5	4	3	2	1	0
TCDB	TEXDB	TEXIEN	RSTCAPSEL	TEXEN	TEX_EDGE	TX_PHASE	

位	描述	
[31:9]	悶隱	保留
[8]	CAP_MODE	捕获模式选择 0 = 定时器计数复位功能或定时器捕捉功能的自由计数模式 1 = 定时器捕捉功能的触发计数模式
[7]	TCDB	定时器外部计数输入引脚防抖动使能控制 0 = 关闭Tx (x = 0~1)引脚防抖动功能 1 = 使能Tx (x = 0~1)引脚防抖动功能 如果这个位被使能, Tx (x = 0~1)引脚的输入信号边沿将被防抖动电路检测.
[6]	TEXDB	定时器外部捕获输入引脚防抖动使能控制 TEXCON0: 0 = 关闭TOEX引脚防抖动功能 1 = 使能TOEX引脚防抖动功能 如果这个位被使能, TOEX引脚的输入信号边沿将被防抖动电路检测 TEXCON1: 保留
[5]	TEXIEN	定时器外部捕获中断使能控制 TEXCON0: 0 = 禁止TOEX引脚检测中断 1 = 使能TOEX引脚检测中断 如果TEXIEN被使能, 当TEXIF标志被置位时, 将产生外部捕获中断信号, CPU中断发生 TEXCON1:

位	描述
	保留
[4]	<p>RSTCAPSEL</p> <p>定时器外部复位计数器/外部捕获模式选择</p> <p>TEXCON0: 0 = TOEX引脚上的信号转变被用作定时器捕获功能, 如果.TEXIF标志被设置为1, TDR值将存到TCAP中。 1 = TOEX引脚上的信号转变被用来复位24位向上计数器</p> <p>TEXCON1: 保留</p>
[3]	<p>TEXEN</p> <p>定时器外部引脚功能使能控制</p> <p>TEXCON0: 该位用来使能TOEX引脚上的RSTCAPSEL功能 0 = TOEX引脚上的RSTCAPSEL功能被忽略. 1 = TOEX引脚上的RSTCAPSEL功能有效</p> <p>TEXCON1: 保留</p>
[2:1]	<p>TEX_EDGE</p> <p>[1:0]</p> <p>定时器外部捕获引脚边沿检测选择</p> <p>TEXCON0: 00 = TOEX引脚上1到0的转变将被检测. 01 = TOEX引脚上0到1的转变将被检测. 10 = TOEX引脚上1到0或0到1的转变都将被检测. 11 = 保留.</p> <p>TEXCON1: 保留</p>
[0]	<p>TX_PHASE</p> <p>定时器外部计数引脚相位 检测选择</p> <p>这个位用来指示检测的Tx (x = 0~1)引脚的信号相位. 0 = Tx (x = 0~1)引脚上一个下降沿转变将被计数. 1 = Tx (x = 0~1)引脚上一个上升沿转变将被计数.</p>

定时器外部中断状态寄存器 (TEXISR)

寄存器	偏移量	R/W	描述	复位值
TEXISR0	TMR_BA+0x18	R/W	Timer0 外部中断状态寄存器	0x0000_0000
TEXISR1	TMR_BA+0x38	R/W	Timer1 外部中断状态寄存器	0x0000_0000

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

位	描述	
[31:1]	悵隱	保留
[0]	TEXIF	<p>定时器外部捕获中断标志</p> <p>TEXISR0:</p> <p>这个位指示定时器外部捕获中断标志状态。</p> <p>当TEXEN使能时，TOEX引脚选择为外部捕获功能。此时TOEX引脚上发生一个转变与TEX_EDGE的设定信号一致时，该位将由硬件置位。.</p> <p>例如，TEXEN = 1, TEX_EDGE = 00, TEX引脚上一个1到0的转变将导致TEXIF被置.</p> <p>0 = TOEX引脚中断没有发生 1 = TOEX引脚中断发生</p> <p>注：该位写1清0.</p> <p>TEXISR0:</p> <p>保留</p>

5.9 PWM 发生器和捕捉定时器 (PWM) (NUC029xAN)

5.9.1 概述

NuMicro® NUC029xAN系列有2个PWM单元，共有4组PWM发生器，可配置成8个独立的PWM输出，PWM0~PWM7，或者4个互补的PWM对，(PWM0, PWM1), (PWM2, PWM3), (PWM4, PWM5) 和 (PWM6, PWM7)，带4个可编程的死区发生器

每组PWM发生器带有8位预分频器，一个时钟分频器提供5种分频(1, 1/2, 1/4, 1/8, 1/16)，两个PWM定时器包括2个时钟选择器，两个16位PWM计数器用于PWM周期控制，两个16位比较器用于PWM占空比控制以及一个死区发生器。4组PWM发生器提供8个独立的PWM中断标志，当相应的PWM周期向下计数器达到零时这些中断标志由硬件置位。每个PWM中断源和它相应的中断使能位可以导致PWM发生中断。PWM发生器可以配置为单触发模式产生(仅一个PWM周期)或自动重载模式(连续输出PWM波形)。

当DZEN01(PC[4])置位，PWM0与PWM1实现互补的PWM对功能，这一对PWM的周期，占空比和死区时间由PWM0定时器和死区发生器0决定。同样，PWM互补对(PWM2, PWM3), (PWM4, PWM5)与(PWM6, PWM7)分别由PWM2, PWM4与PWM6的定时器和死区发生器2, 4, 6控制，PWM定时器架构请参考下图。

为防止PWM输出不稳定波形，16位向下计数器和16位比较器采用双缓存。当用户向计数器/比较器寄存器写入值的时候，只有当向下计数器的值达到0时，被更新的值才会被装载到16位计数器/比较器。该双缓冲特性避免PWM输出波形上产生毛刺。

当16位向下计数器达到0时，中断请求产生。如果PWM定时器被配置为自动重装载模式，当向下计数器达到0时，会自动重新装载PWM计数器寄存器(CNRx)的值，并开始递减计数，如此连续重复。如果定时器设为单触发模式，当向下计数器达到0时，向下计数器停止计数，并产生一个中断请求。

PWM计数器比较器的值(CMRx)用于高电平脉冲宽度调制，当向下计数器的值与比较寄存器的值相同时，计数器控制逻辑反转输出为高电平。

PWM定时器可复用为数字输入捕捉功能。如果捕捉功能使能，PWM的输出引脚将被切换至捕捉输入模式。捕捉器0和PWM0使用同一个定时器，捕捉器1和PWM1使用另一组定时器，以此类推。因此在使用捕捉功能之前，用户必须预先配置PMW定时器。捕捉功能使能后，捕捉器在输入通道有上升沿跳变时，将PWM计数器的值锁存至捕捉上升沿锁存寄存器(CRLR)，在输入通道有下降沿跳变时将PWM计数器值锁存至捕捉下降沿锁存寄存器(CFLR)。捕捉通道0的中断是可编程的，通过设定CRL_IE0(CCR0[1])(上升沿锁存中断使能)和CFL_IE0(CCR0[2])(下降沿锁存中断使能)来决定中断发生的条件。通过设置CRL_IE1(CCR0[17])和CFL_IE1(CCR0[18])，捕捉通道1有同样的特性。通过设置CCR2中的相应的控制位，每组的通道2到通道3有同样的特性。对于每一组，不管捕捉何时产生中断0/1/2/3，PWM计数器0/1/2/3都将在该时刻重载。

最大的捕捉频率受捕捉中断延迟限制。捕捉中断发生时，软件至少要执行三个步骤：读PIIR以得到中断源，读CRLRx/CFLRx(x=0~3)以得到捕捉值，最后写1清PIIR为0。如果中断延迟要花时间T0完成，在这段时间内(T0)，捕捉信号一定不能翻转。在这种情况下，最大的捕捉频率将是1/T0。

5.9.2 特性

5.9.2.1 PWM 功能：

- 2个PWM组 (PWMA/PWMB)，支持8个PWM通道或者4个互补的PWM通道
- 每组PWM有2个PWM发生器，每个PWM发生器支持一个8位的预分频器，一个时钟分频器，两个PWM定时器，一个死区发生器和两路PWM输出

- 最高16位分辨率
- 单触发模式或自动重载模式
- 支持边沿对齐或者中心对齐
- PWM触发ADC启动转换

5.9.2.2 捕捉功能:

- 与PWM发生器共享时序控制逻辑
- 8路捕捉输入通道与8个PWM输出通道复用
- 每个通道支持一个上升沿锁存寄存器(CRLRx)，一个下降沿锁存寄存器(CFLRx)和捕捉中断标志(CAPIFx)

5.9.3 框图

下图按对说明PWM架构(例如： PWM定时器0/1为一对， 定时器2/3为另外一对， 等等)。

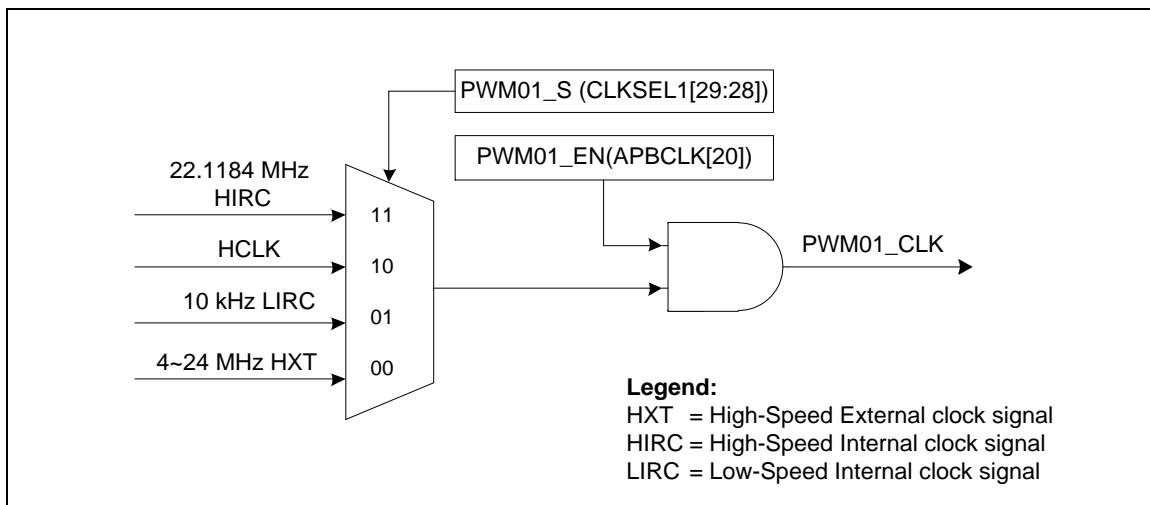


图 5-45 PWM 发生器0 时钟源控制

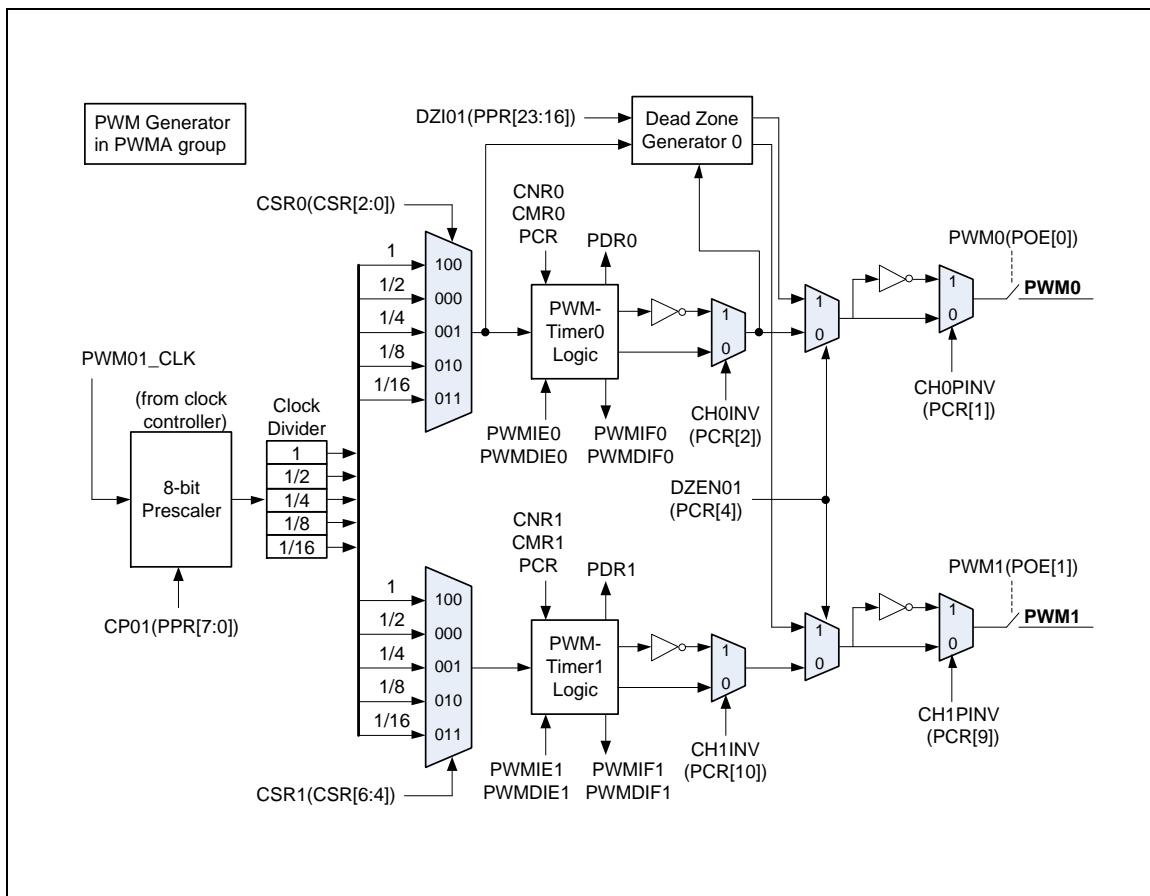


图 5-46 PWM 发生器0 结构框图

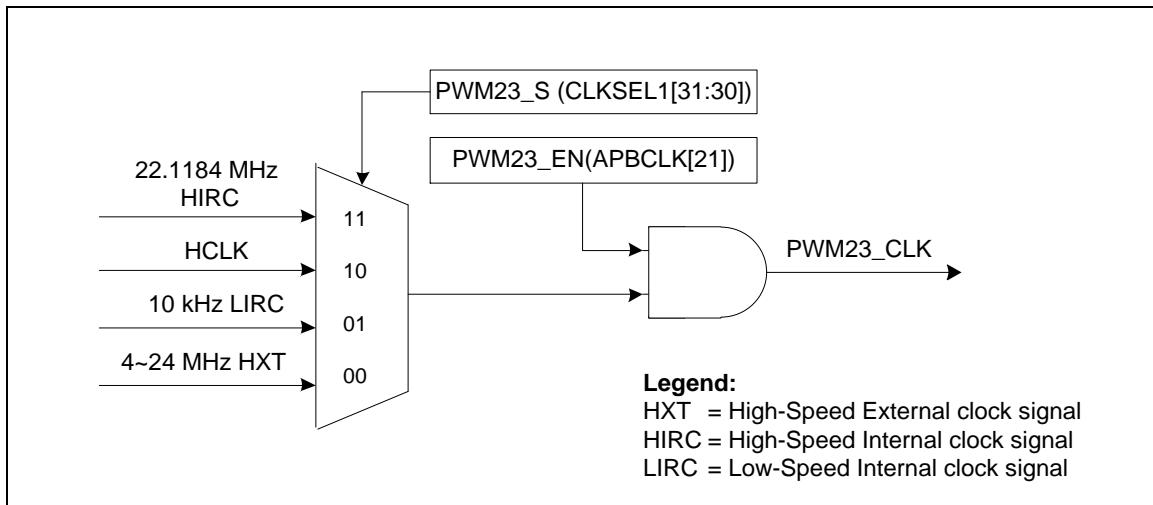


图 5-47 PWM 发生器2 时钟源控制

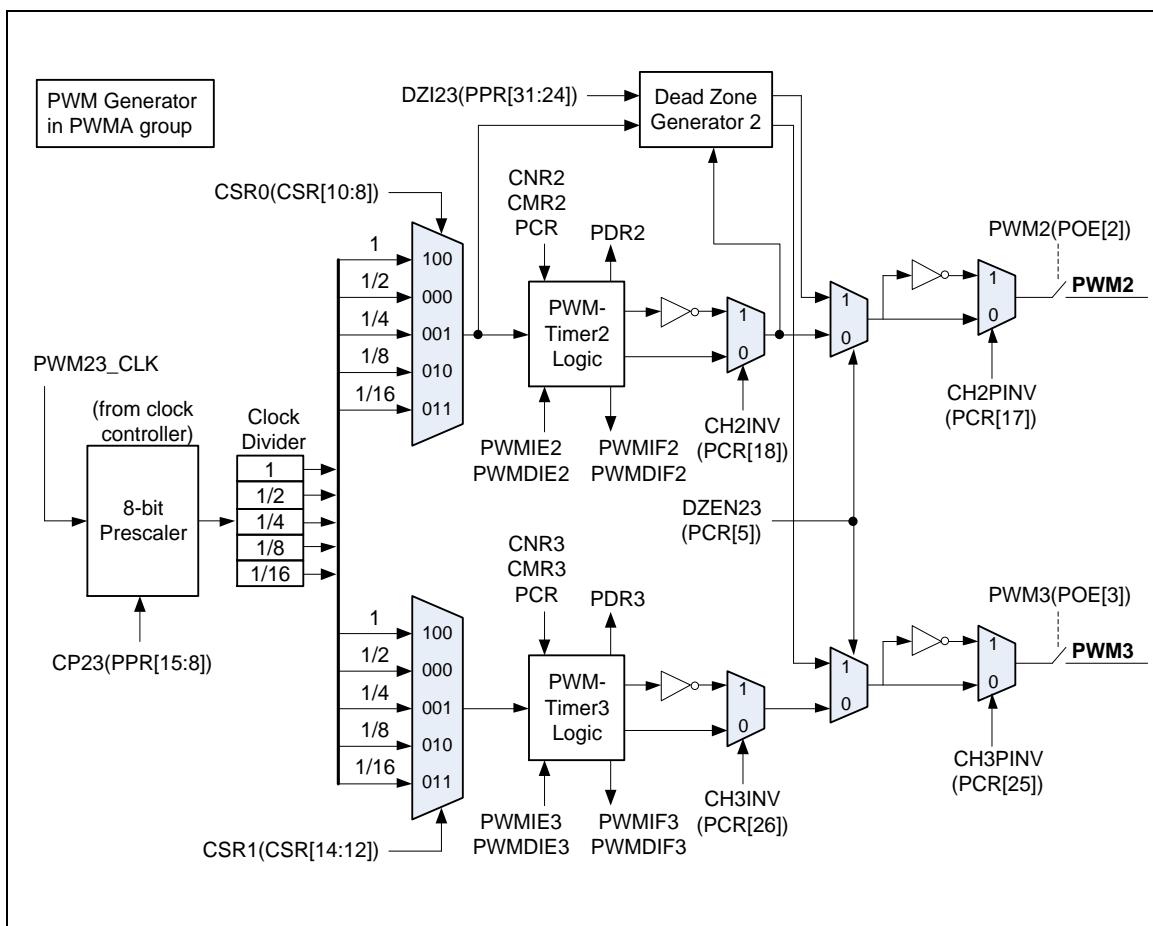


图 5-48 PWM 发生器2 结构框图

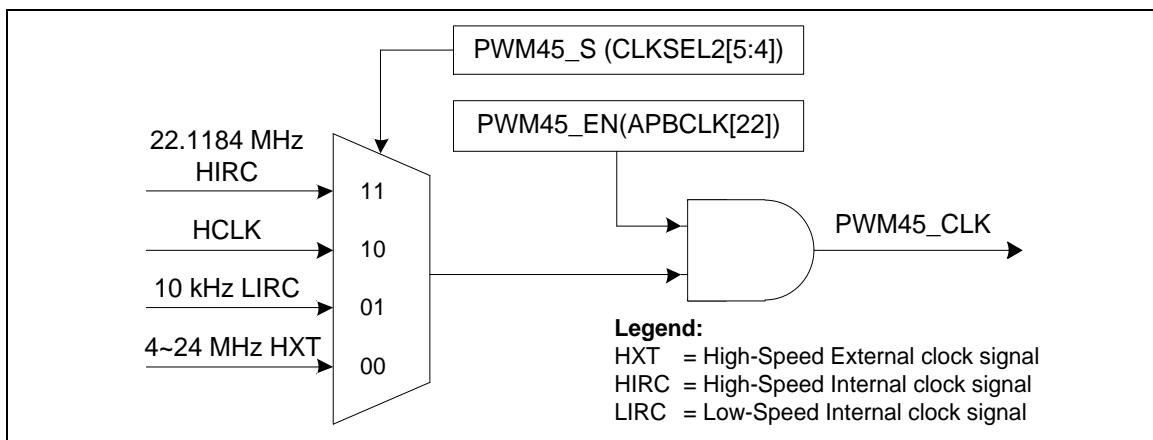


图 5-49 PWM 发生器4 时钟源控制

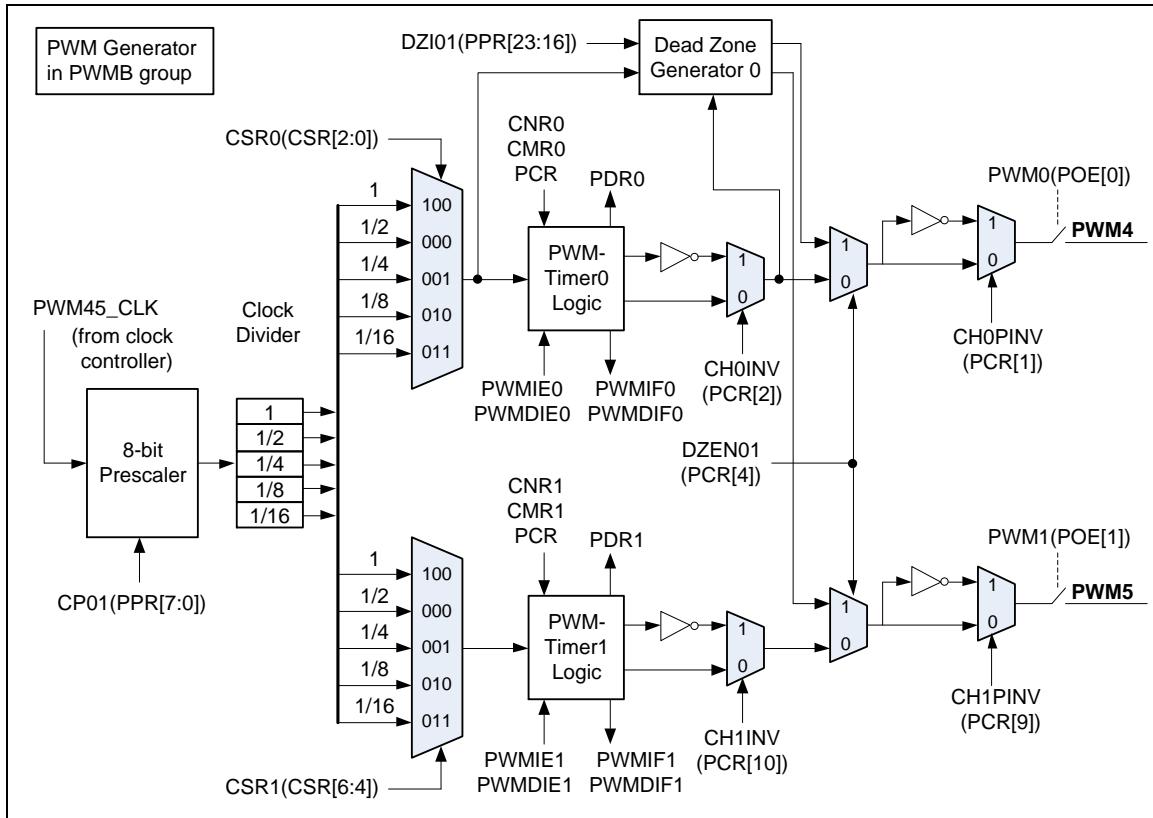


图 5-50 PWM 发生器4 结构框图

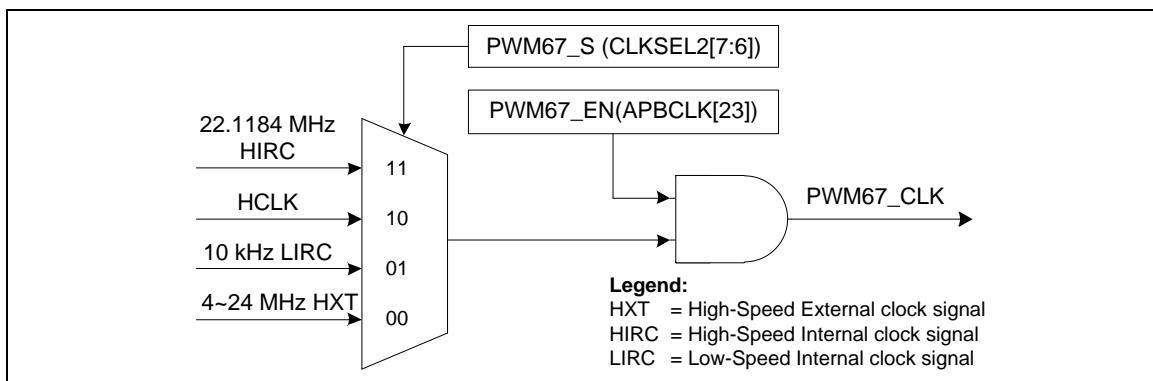


图 5-51 PWM 发生器6 时钟源控制

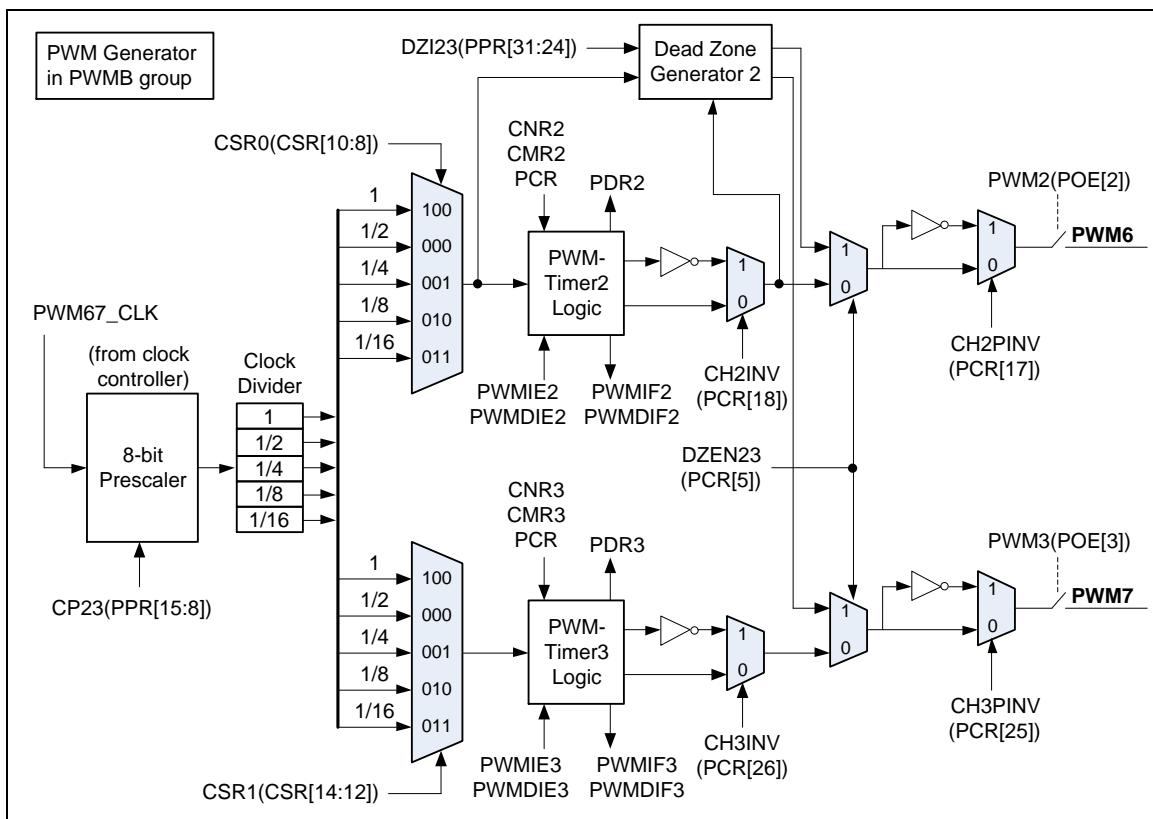


图 5-52 PWM 发生器6 结构框图

5.9.4 基本配置

PWM 引脚在P2_MFP and P4_MFP 寄存器中配置。

PWM 时钟在APBCLK[23:20]中使能。PWM 时钟源在CLKSEL1[31:28] 和 CLKSEL2[7:4]中选择。

5.9.5 功能描述

5.9.5.1 PWM-定时器操作

PWM控制器支持两种操作类别：边沿对齐和中心对齐

边沿对齐PWM (下数计数器)

边沿对齐PWM输出模式下，16位PWM计数器从CNRx开始向下数，直到计数器的值等于CMRx，然后PWMx发生器反转输出为低电平。计数器继续向下数直到0，此时PWMx发生器反转输出为高电平，如果CHxMODE=1，CMRx(新的)和CNRx(新的)被更新，此时如果PWM中断使能(PIERx=1)，PWM中断将发生。

PWM 周期和占空比控制由PWM向下计数器寄存器(CNR)以及PWM比较寄存器(CMR)配置。PWM 定时器工作时序如下图所示。. 脉宽调制的公式如下，PWM定时器比较器的说明如图所示。注意：相关PWM通道相应的GPIO引脚必须配置成PWM功能(使能POE和禁用CAPENR)。

- PWM 频率 = $\text{PWM}_{xy_CLK}/[(\text{prescale}+1)*(\text{clock divider})*(\text{CNR}+1)]$; xy代表01, 23, 45 或 67, 取决于所选择的PWM通道.
- 占空比 = $(\text{CMR}+1)/(\text{CNR}+1)$
- CMR >= CNR: PWM输出总为高
- CMR < CNR: PWM低脉冲宽度= $(\text{CNR}-\text{CMR})$ 单位^[1]; PWM高脉冲宽度 = $(\text{CMR}+1)$ 单位
- CMR = 0: PWM 低脉冲宽度 = (CNR) 单位; PWM高脉冲宽度= 1 单位

注: [1]. 单位 = 一个PWM时钟周期

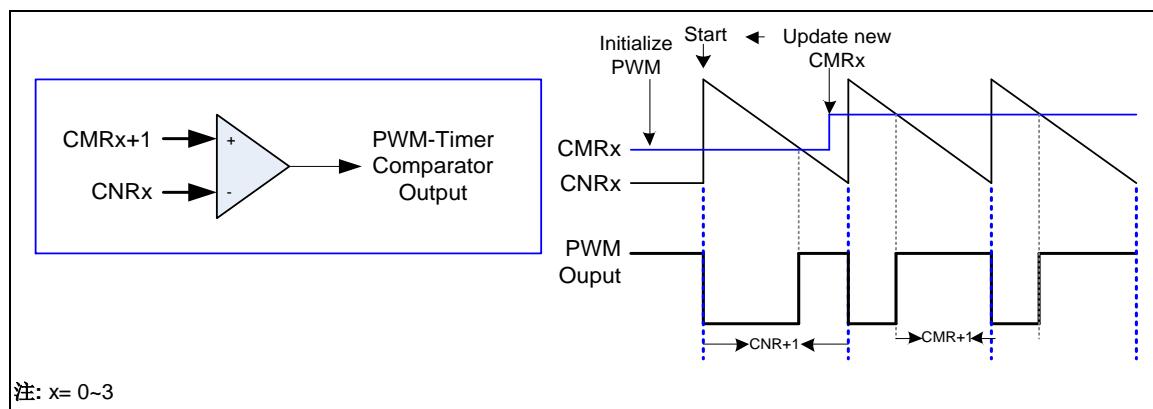


图 5-53 定时器内部比较器输出

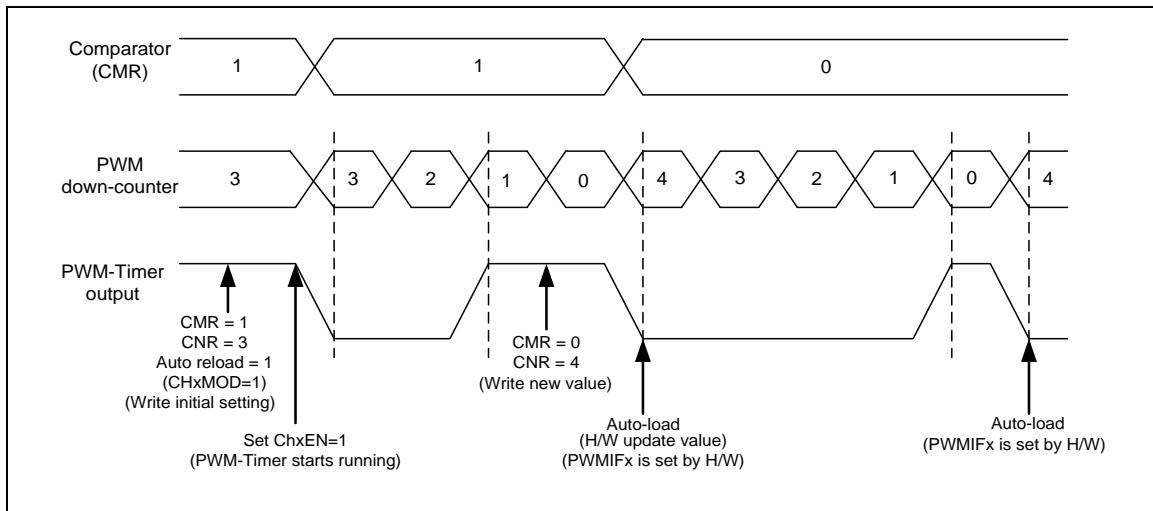


图 5-54 PWM 定时器操作时序

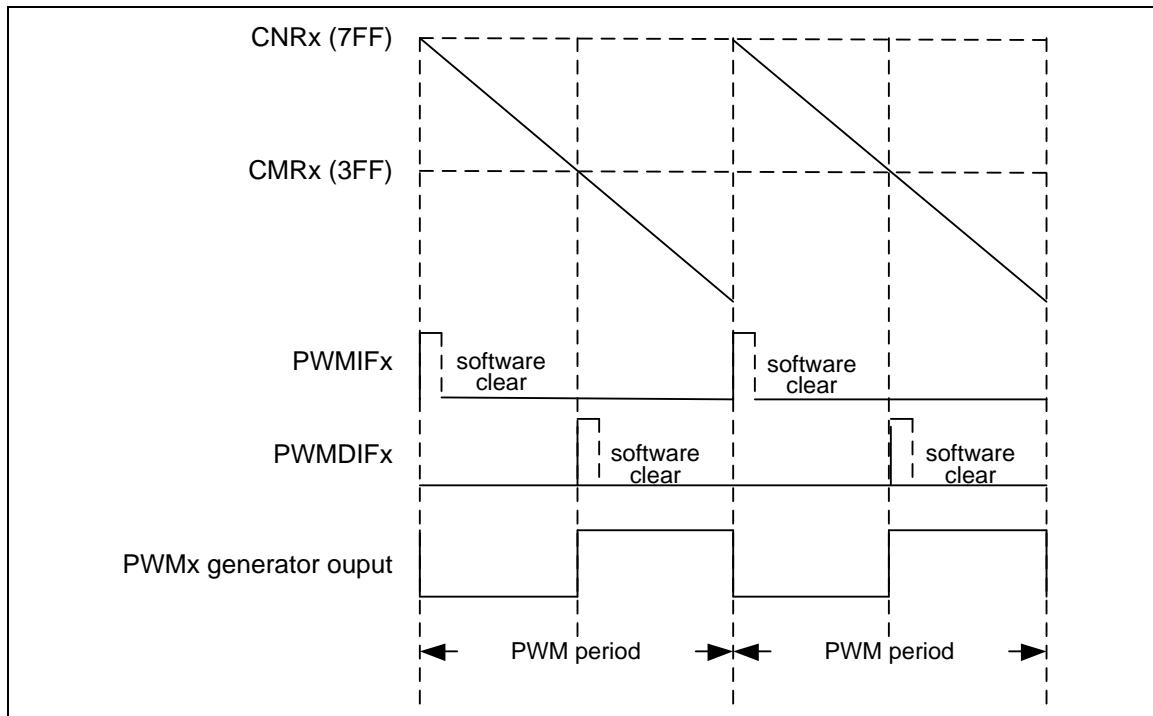


图 5-55 PWM 周期中断发生时序图

中心对齐 PWM (上数/下数计数器)

当PWM配置为中心对齐模式时，其计数器为上数/下数模式。PWM计数器从0开始上数直到等于CMRx的值，然后PWMx发生器反转输出为低电平。计数器继续上数直到等于CNRx的值，然后计数器自动开始下数直到再次等于CMRx的值，然后PWM发生器反转输出为高电平。计数器继续下数一直到0，如果CHxMODE = 1，CNRx和CMRx的值将被重新加载，再次开始上述的循环。

- PWM 频率 = $\text{PWM}_{xy_CLK}/[(\text{prescale}+1) * (\text{clock divider}) * (\text{CNR}+1)]$; 根据选择的PWM通道，xy 应该是 01, 23, 45 or 67。
- 占空比 = $[(2 \times \text{CMR}) + 1]/[2 \times (\text{CNR}+1)]$
- CMR > CNR: PWM 输出总是为高
- CMR <= CNR: PWM 低电平宽度= $2 \times (\text{CNR}-\text{CMR}) + 1$ 单位^[1]; PWM 高电平宽度 = $(2 \times \text{CMR}) + 1$ 单位
- CMR = 0: PWM低电平宽度= $2 \times \text{CNR} + 1$ 单位; PWM 高电平宽度= 1 单位

注 [1]: 单位 = 一个 PWM 时钟周期

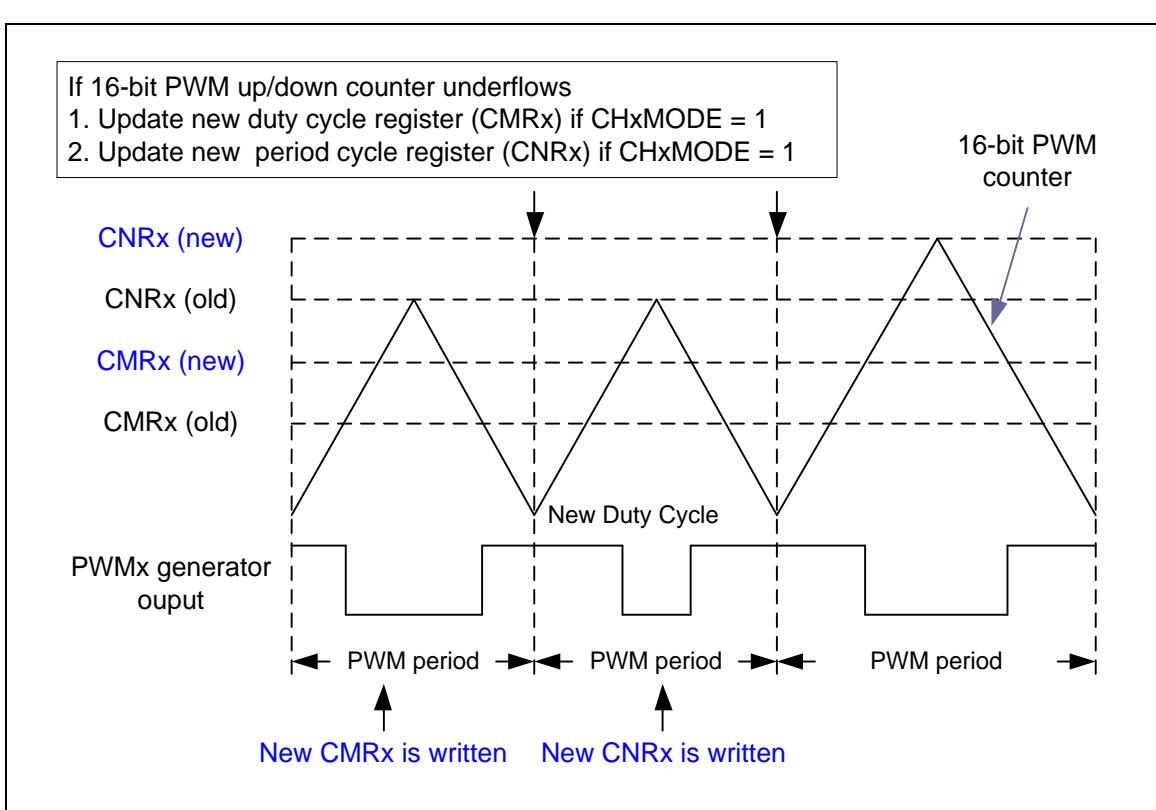


图 5-56 中心对齐输出波形

中心对齐模式下，系统可以产生2种中断：周期中断和工作中断，有4种时序：如果INTxxTYPE (PIER[17:16]) = 0，计数器下数到0时发生PWM周期中断；INTxxTYPE (PIER[17:16]) = 1，计数器上数到等于CNRx的值时发生PWM周期中断，就是说在PWM周期的中点发生中断；如果INTxxDTYPE (PIER[25:24]) = 0，计数器下数等于CMRx的值发生PWM工作中断；如果INTxxDTYPE (PIER[25:24]) = 1，计数器上数等于CMRx的值发生PWM工作中断。

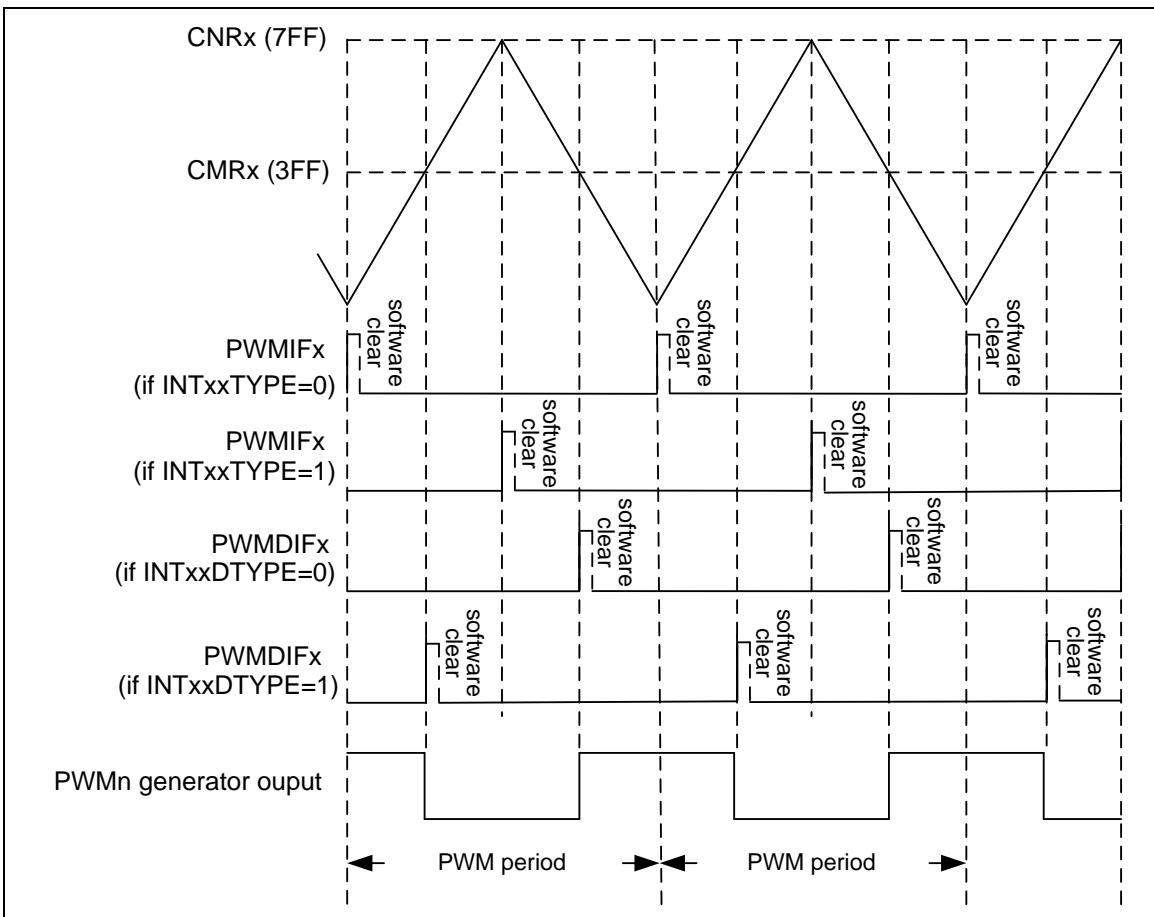


图 5-57 PWM 中心对齐中断发生时序图

5.9.5.2 PWM 双缓存，自动重载以及单触发模式

PWM定时器具有双缓存功能。新的值将在下一个周期开始时重新加载，不会影响当前定时器的工作。新的PWM计数器的值可写入CNRx，当前PWM计数器的值可以从PDRx读取。

PWM0 控制寄存器(PCR) 的CH0MOD 位定义 PWM0 是自动重载模式还是单次触发模式。如果 CH0MOD 被设置为 1，PWM0 为自动重载模式；如果 CH0MOD 被设置为 0，PWM0 为单次触发模式。推荐设置 CH0EN 为 1 使能 PWM0 计数器开始计数之前，就应该先设好 CH0MOD，因为 PWM0 操作模式的改变会把 CNR0 和 CMR0 的内容清为 0，同时会复位 PWM0 的周期和占空比。如果 PWM0 工作在单次触发模式，填好 CMR0 和 CNR0 之后，设置 CH0EN 为 1 使能 PWM0 开始计数。在 PWM0 计数器从 CNR0 下数到 0 时，CNR0 和 CMR0 寄存器的值将被硬件清为 0，PWM0 计数器将停止工作。软件需要填新的 CMR0 和 CNR0 的值启动下一次单次触发模式。当重新开始下一次单次触发模式

时，CMR0应该先被写，因为CNR0一旦被写，PWM0计数器将自动重新开始计数。如果PWM0工作在自动重载模式，填好CNR0和CMR0之后，设置CH0EN等于1启动PWM0开始计数。当计数器下数到0时，CNR0的值将被自动加载到PWM0计数器中。如果CNR0设为0，PWM0计数器将停止工作。PWM1 ~ PWM7工作方式和PWM0相同。

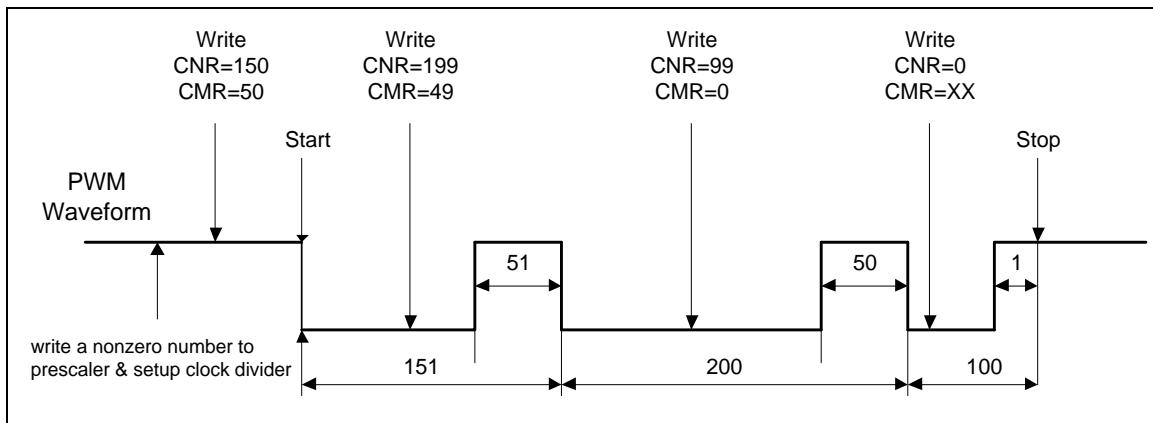


图 5-58 PWM 双缓存图解

5.9.5.3 调制占空比

双缓存功能允许CMRx在当前周期的任意时刻被写入。写入的值将在下个周期生效。

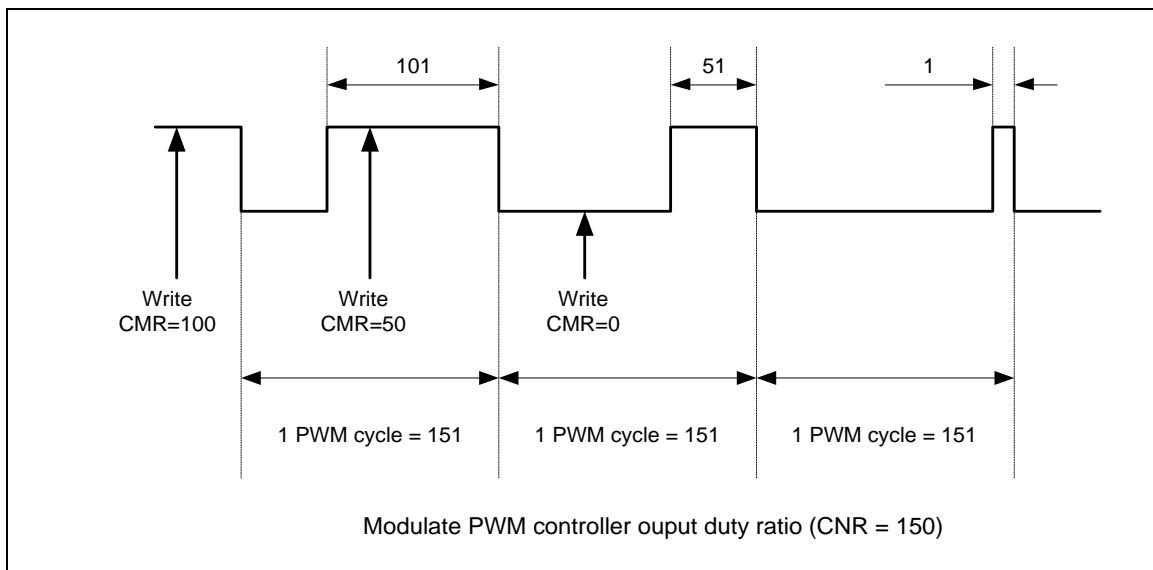


图 5-59 PWM 控制器输出占空比

5.9.5.4 死区发生器

PWM控制器提供死区发生器，用于保护功率器件。该功能产生可编程的时隙来延迟PWM上升沿输出时间点，用户可通过编程PPRx.DZI确定死区间隔。

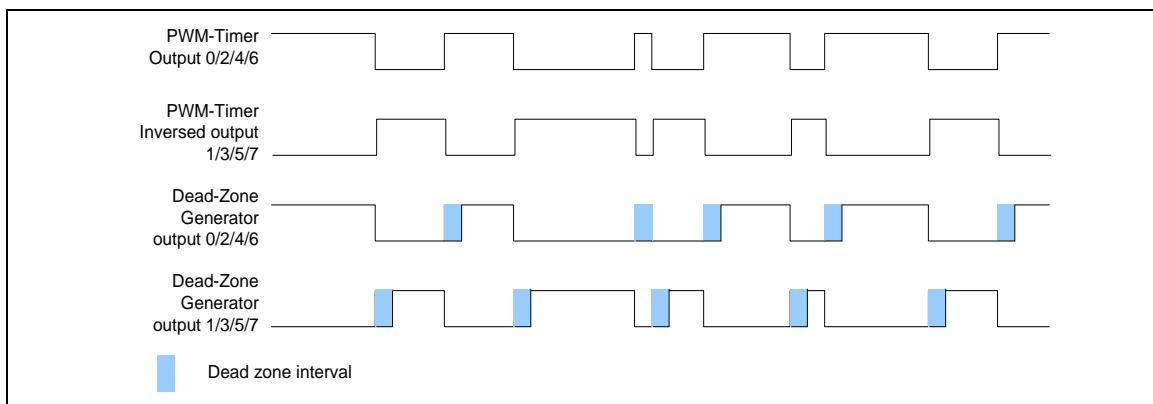


图 5-60 死区发生器操作的 PWM 对输出

5.9.5.5 PWM定时器触发ADC开始转换

PWM 可以触发ADC开始转换。当 PWM 工作在中心对齐模式时，有4种PWM触发ADC开始转换的时序；当 PWM 工作在边沿对齐模式时，有2种PWM触发ADC开始转换的时序。

对于中心对齐模式，通过设定INTxxDTYPE/ INTxxTYPE 和 PWMxDTEN/ PWMxTEN，PWM可以在不同的时序触发ADC开始转换。触发ADC时序和设定如下所示，共4种。

INTxTEN	INTxxTYPE	描述
1	0	下数时PWM 计数器等于0时触发
1	1	上数时PWM 计数器等于 CNRx + 1 时触发

INTxDTEN	INTxxDTYPE	描述
1	0	下数时PWM 计数器等于CMRx时触发
1	1	上数时PWM 计数器等于CMRx 时触发

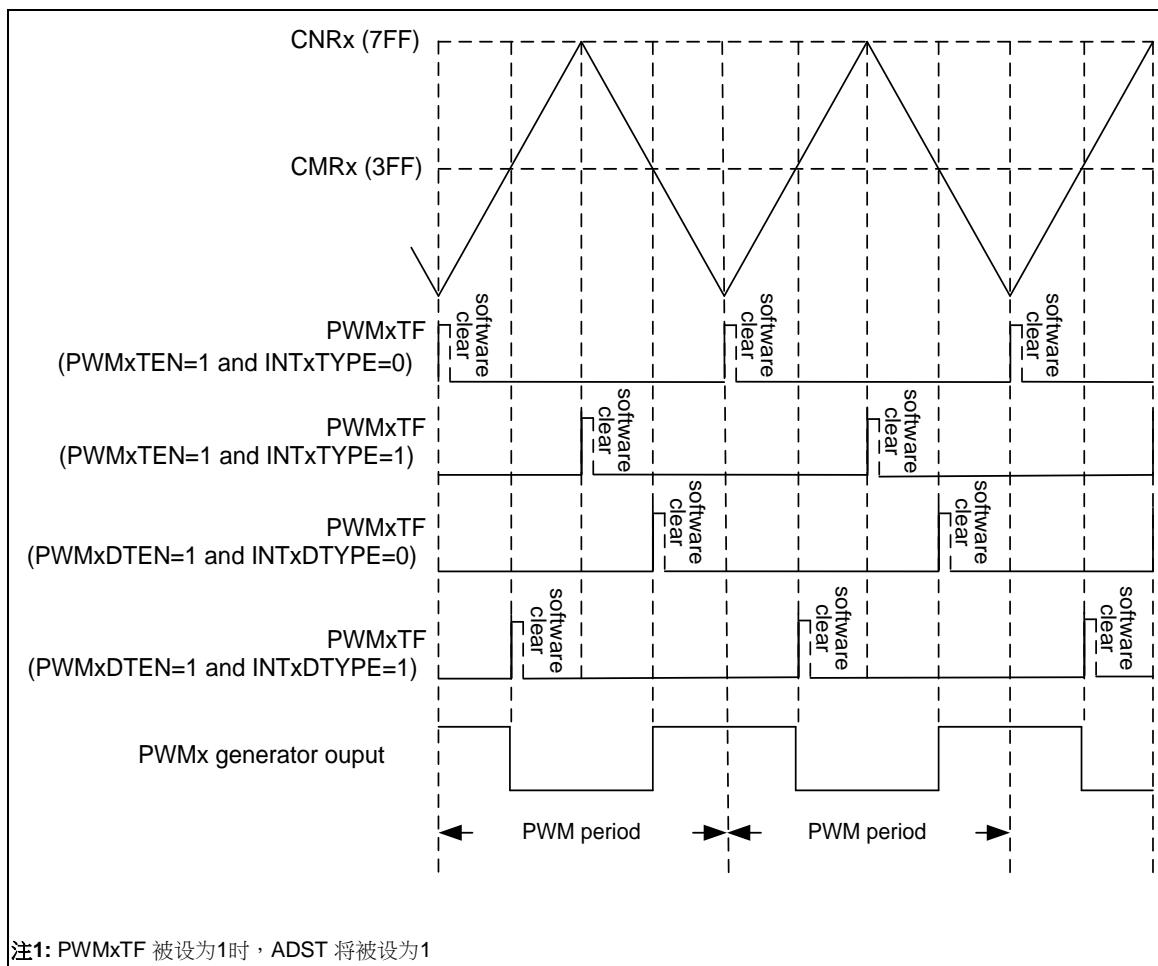


图 5-61 中心对齐模式下 PWM 触发 ADC 标志 (PWMxTF) 时序图

对于边沿对齐模式，通过设定 PWMxDTEN/ PWMxTEN，PWM可以在不同的时序触发 ADC 开始转换。触发ADC时序和设定如下所示，共2种。

INTxDTEN	描述
1	下数时PWM 计数器等于CMRx 触发
INTxTEN	描述
1	下数时PWM 计数器等于0触发

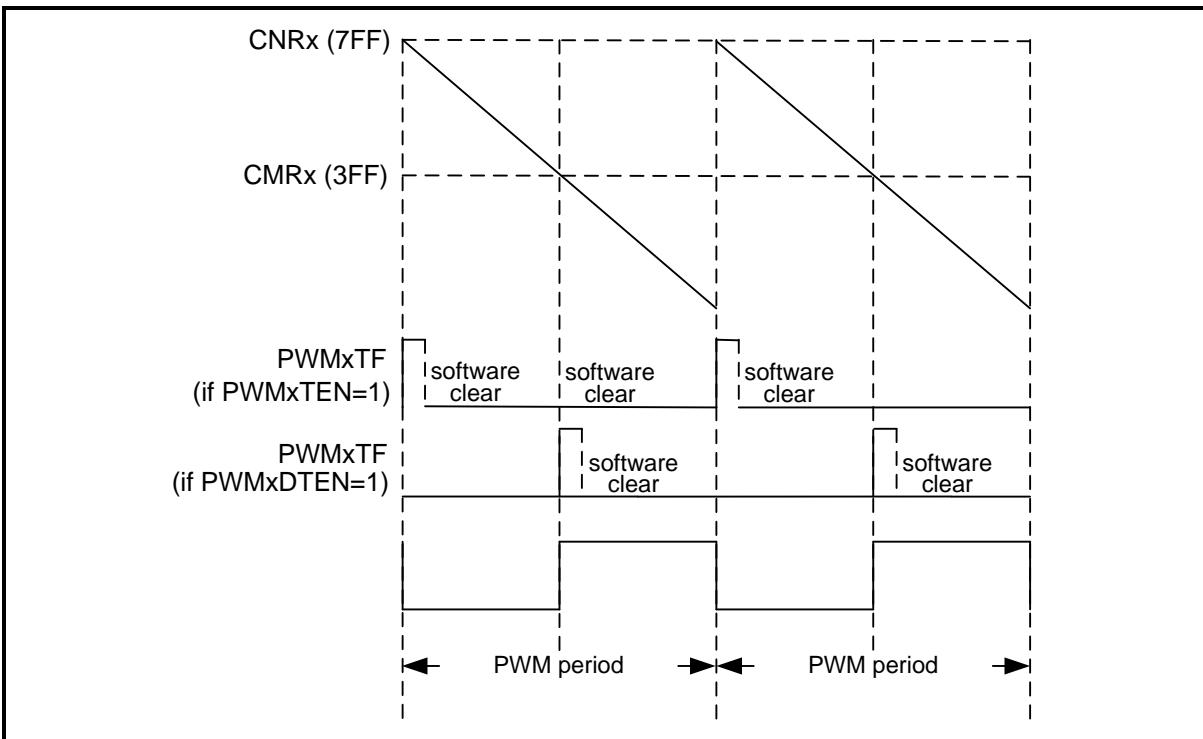


图 5-62 边沿对齐时 PWM 触发ADC标志 (PWMxTF) 时序图

5.9.5.6 PWM 触发 ADC 开始转换步骤

1. 配置预分频寄存器 (PPRx) 设置预分频值 (CPxx).
2. 配置时钟选择寄存器 (CSR_x) 选择时钟源除频值(CSR_x).
3. 配置 PWM 控制寄存器 (PCR_x) 设置自动重载模式 (CH_xMOD = 1), PWM 对齐类型 (PWM_{xx}TYPE) 并关闭 PWM 计数器 (CH_xEN = 0).
4. 配置 PWM 控制寄存器 (PCR_x) 设置反转使能/关闭 (CH_xINV), 极性反转使能/关闭 (CH_xPINV) 和死区发生器使能/关闭 (DZEN_{xx}). (可选)
5. 配置 PWM 中断使能寄存器 (PIER) 设置 PWM 周期中断类型 (INT_{xx}TYPE) 和 PWM 工作中断类型(INT_{xx}DTYPE).
6. 配置 PWM 触发控制寄存器(TCON) 使能/禁止 PWM 周期触发 ADC (PWM_xTEN) 和 PWM 工作触发 ADC (PWM_xDTEN).
7. 配置比较寄存器(CMR_x) 设置 PWM 高电平时间 (CMRx).
8. 配置 PWM 计数寄存器(CNR_x) 设置 PWM 的周期(CNRx).
9. 配置 PWM 输出使能 寄存器(POE) 使能 PWM 输出通道 (PWM_x = 1)
10. 配置ADC 触发延迟控制寄存器(ADTDCR) (在ADC 控制器中)设置 PWM 触发延迟时间 (PTDT) (可选)
11. 配置ADC 通道使能寄存器(ADCHER) (在 ADC 控制器中)使能模拟输入通道 (CHEN_x =1)
12. 配置ADC 控制 寄存器(ADCR) (在 ADC 控制器中) 设置硬件触发源来自PWM (TRGS = 3), 使

能外部触发($TRGEN = 1$)，A/D 转换器操作模式($ADMD = 0,2,3$) 和 A/D 转换器使能($ADEN = 1$)

13. 配置PWM 控制 寄存器(PCR) 使能 PWM 开始工作 ($CHxEN = 1$)
14. 当PWM触发标志(PWMxTF)被硬件设为1时，ADST(ADCR[11]) 位将被硬件设为 1
15. 软件可以轮询 A/D 转换结束标志 – ADF (ADSR[0]) 来检查转换是否结束.
16. 单次模式和单周期扫描模式下ADST (ADCR[11]) 将被自动清为0；循环扫描模式下，ADST 维持为1直到软件将其清0.

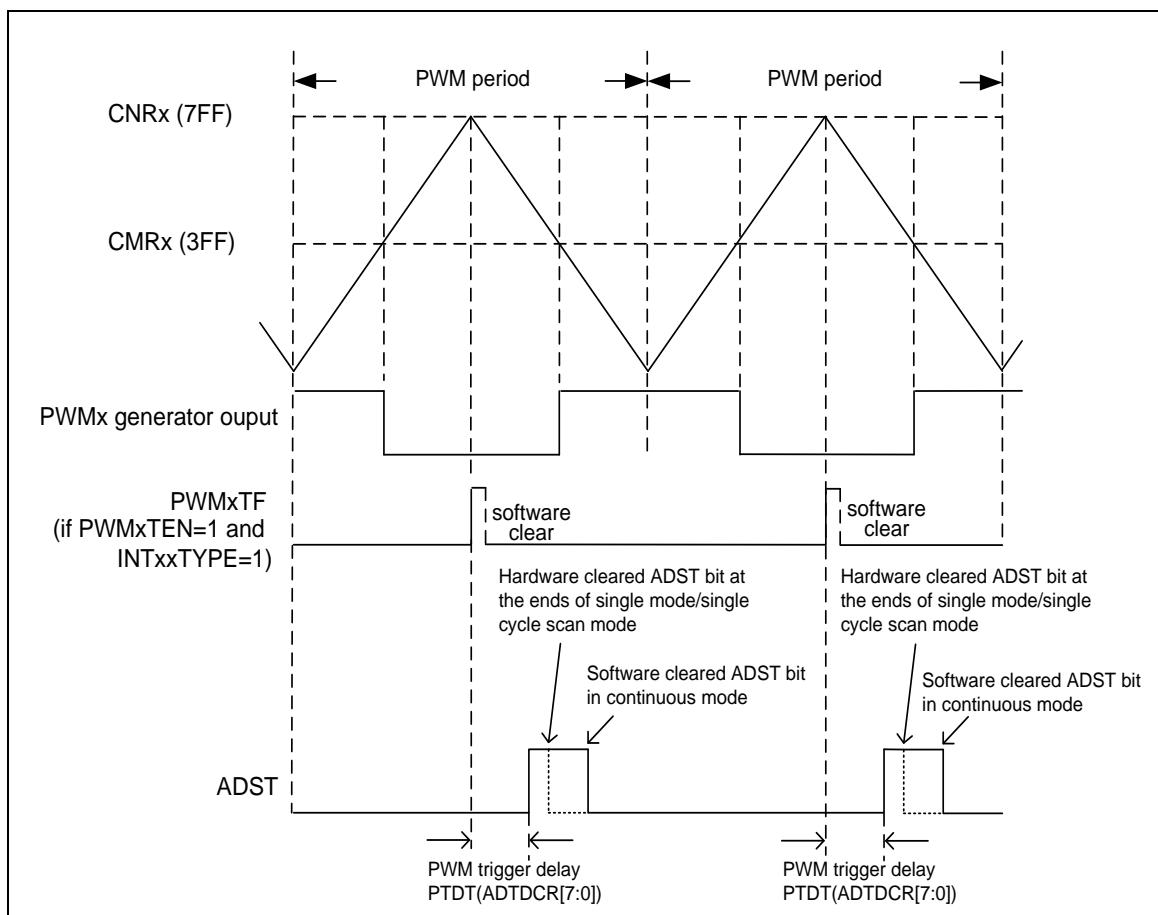


图 5-63 中心对齐模式下 PWM 触发ADC 开始转换

5.9.5.7 捕获操作

捕获器0和PWMO共用同一个定时器，捕获器1和PWM1共用另一个定时器，以此类推。在输入通道有上升沿跳变时捕获器将PWM计数器的值锁存至CRLRx寄存器，在输入通道有下降沿跳变时将PWM计数器的值锁存至CFLRx寄存器。捕获通道0的中断是可编程的，通过设定CCR0[1] (上升沿锁存中断使能)和CCR0[2](下降沿锁存中断使能) 来决定中断发生的条件。通过设置CCR0[17]和CCR0[18]捕获通道1有同样的特性。无论捕获模块何时触发一个捕获中断，相应的PWM计数器都将在此刻重载CNRx的值。注：捕获通道相应的GPIO管脚必须配置成捕获功能(禁用POE和使能CAPENR)。

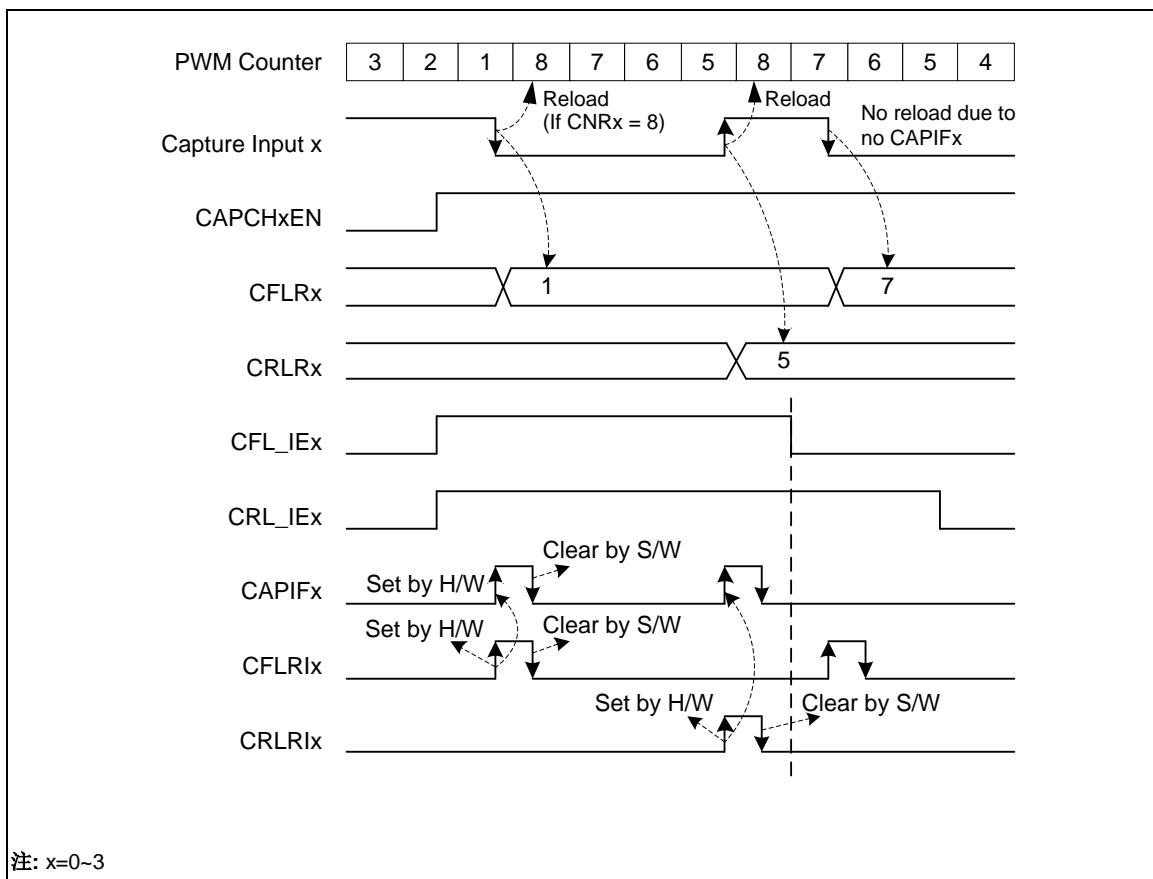


图 5-64 捕捉操作时序

在上述范例中，CNR 为8：

1. 捕捉中断标志(CAPIFx)置位时，PWM计数器将重装载CNRx的值.
2. 通道低脉冲宽度为(CNR + 1 - CRLR).
3. 通道高脉冲宽度为(CNR + 1 - CFLR).

5.9.5.8 PWM定时器中断架构

有8个PWM中断，`PWM0_INT~PWM7_INT`，被分到2个中断向量`PWMA_INT`和`PWMB_INT`。`PWM 0`与捕获器0共用同一个中断。`PWM1`与捕获器1共用同一个中断，以此类推。因此，同一通道的PWM功能和捕获功能不能同时使用。下图说明了PWM定时器中断结构。

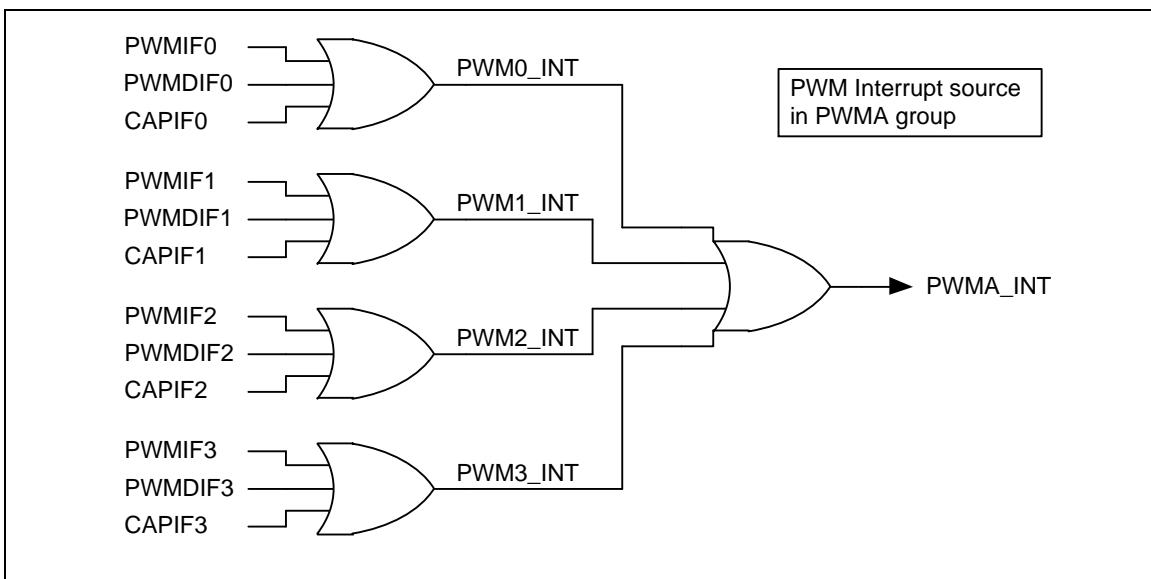


图 5-65 PWM A组 PWM定时器中断结构图

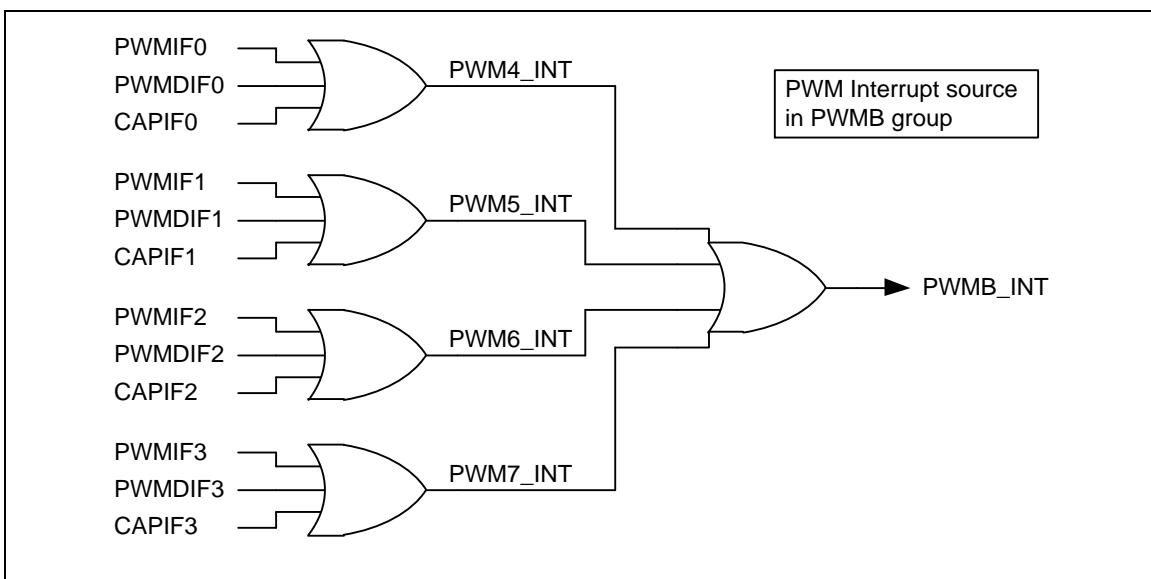


图 5-66 PWM B组 PWM定时器中断结构图

5.9.5.9 PWM定时器启动步骤

推荐使用如下步骤启动PWM驱动器。

1. 配置预分频寄存器(PPR), 设置预分频(CPx_x)
2. 配置时钟除数选择寄存器 (CSR), 选择时钟除数(CSR_x)
3. 配置PWM控制寄存器(PCR), 设置自动重载模式(CHxMOD = 1)并关闭PWM计数器(CHxEN = 0)
4. 配置PWM控制寄存器(PCR), 设置反向打开/关闭(CHxINV), 死区发生器打开/关闭

(DZENxx) (可选)

5. 配置比较器寄存器(CMRx)，设定PWM占空比(CMRx)
6. 配置PWM计数器寄存器 (CNRx) 设定PWM周期(CNRx).
7. 配置PWM中断使能寄存器(PIER)，使能PWM周期中断(PWMIE). (可选)
8. 配置PWM输出使能寄存器(POE)，使能PWM输出通道(PWMx = 1)
9. 配置PWM控制寄存器(PCR)，使能PWM定时器开始运行(CHxEN = 1)

5.9.5.10 单次触发模式下PWM定时器重新启动步骤

单次触发模式下，PWM 波形产生一次以后，PWM定时器将自动停止计数，CNRx 和 CMRx 将被硬件自动清0。软件必须重填 CMRx 和 CNRx 的值来再次启动PWM。单次触发模式下推荐下列步骤来重新启动 PWM。

1. 配置比较寄存器 (CMRx)，设置PWM 占空比(CMRx).
2. 配置PWM 计数器寄存器 (CNRx) 来设置PWM的周期 (CNRx)。一旦写CNRx寄存器，PWM将马上启动，波形将产生

5.9.5.11 PWM-定时器停止步骤

方式1:

设定16位计数器(CNRx)为0，并监视数据寄存器(PDRx，16位向下计数器的当前值)。当PDRx达到0，关闭PWM定时器(PCR的CHxEN位)。(推荐)

方式2:

设定16位计数器(CNR)为0，当中断请求发生，关闭PWM定时器(PCR的CHxEN位)。(推荐)

方式3:

直接关闭PWM定时器(PCR的CHxEN位)。(不推荐)

不推荐方式3的原因是：禁用CHxEN将立即停止PWM输出信号，引起PWM输出占空比的改变，这可能导致电机控制电路损坏。

5.9.5.12 PWM-定时器同步启动步骤

PWMA组定时器1~3 和PWMB组定时器0~3 可以与PWM A组定时器0同步启动。PWM定时器同步启动步骤如下：

1. 配置预分频寄存器(PPR)，设置预分频值 (CPxx).
2. 配置时钟选择寄存器(CSR) ，选择时钟源除频值 (CSRx).
3. 配置PWM控制寄存器(PCR)，设置反转自动加载模式(CHxMOD = 1), PWM 对齐类型 (PWMxxTYPE) 并关闭PWM定时器 (CHxEN = 0).
4. 配置PWM控制寄存器(PCR) ，设置反转打开/关闭 (CHxINV), 极性反转打开/关闭(CHxPINV) 和死区发生器 打开/关闭(DZENn). (可选)

5. 配置比较寄存器(CMRx)，设置 PWM 高电平 (CMRx).
6. 配置PWM 计数寄存器 (CNRx) ，设置PWM周期 (CNRx).
7. 配置PWM 中断使能寄存器(PIER) ，设置 PWM 周期中断类型 (INTxxTYPE), PWM 工作中断类型 (INTxxDTYPE), PWM 周期中断使能位 (PWMIEx) 和 PWM 工作中断使能位 (PWMDIEx). (可选)
8. 配置PWM 输出使能寄存器(POE) ，使能PWM输出通道 (PWMx)
9. 配置PWM 同步控制寄存器(PSCR)使能 PWM 同步使能位 (PSSENx = 1)
10. 配置PWMA组 控制寄存器(PCR) 使能 A组 PWM定时器0 开始计数 (A组PCR中CH0EN = 1). 所有相关的PWM定时器将和A组的定时器0同步启动.

5.9.5.13 捕获开始步骤

1. 配置预分频寄存器(PPR), 设置时钟预分频(CPx_x).
2. 配置时钟选择寄存器(CSR), 选择时钟除频值(CSRx).
3. 配置PWM控制寄存器(PCR), 选择自动加载模式(CHxMOD = 1) 并关闭PWM定时器 (CHxEN = 0).
4. 配置 PWM捕获控制寄存器(CCRx) , 设置上升沿中断使能(CRL_IEx)和下降沿中断使能(CFL_IEx)以及输入信号反向打开/关闭(INVx).
5. 配置PWM计数器寄存器(CNRx), 设置计数周期(CNRx).
6. 配置PWM控制寄存器(PCR), 使能PWM计数器开始计数(CHxEN = 1).
7. 使能捕获输入通道使能定时器(CAPENR) , 使能相应的GPIO引脚作为捕获功能.

5.9.6 寄存器映射

R: 只读, W: 只写, R/W: 可读写

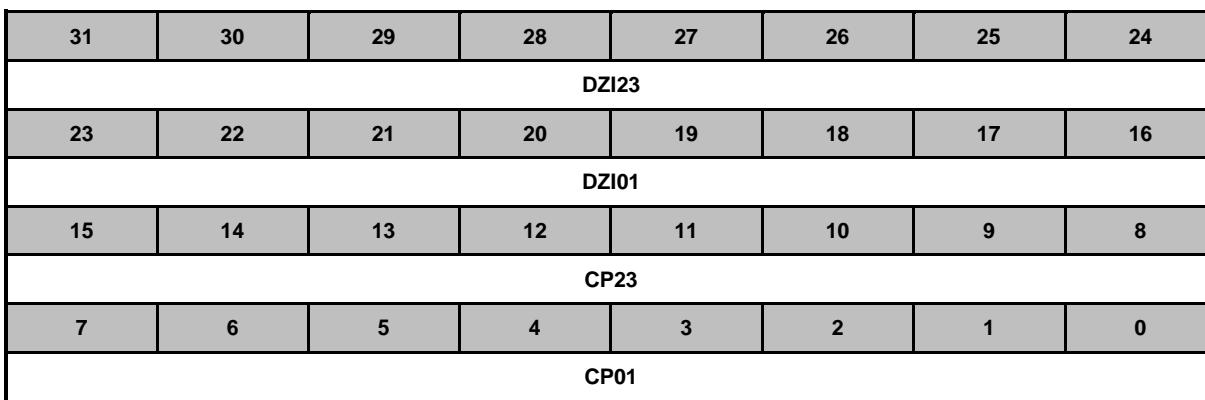
寄存器	偏移量	R/W	描述	复位值
PWM 基地址:				
PWM A组				
PWMA_BA = 0x4004_0000				
PWM B组				
PWMB_BA = 0x4014_0000				
PPR	PWMA_BA+0x00 PWMB_BA+0x00	R/W	PWM 预分频寄存器	0x0000_0000
CSR	PWMA_BA+0x04 PWMB_BA+0x04	R/W	PWM 时钟源除频选择寄存器	0x0000_0000
PCR	PWMA_BA+0x08 PWMB_BA+0x08	R/W	PWM 控制寄存器	0x0000_0000
CNR0	PWMA_BA+0x0C PWMB_BA+0x0C	R/W	PWM 计数寄存器0	0x0000_0000
CMR0	PWMA_BA+0x10 PWMB_BA+0x10	R/W	PWM 比较寄存器 0	0x0000_0000
PDR0	PWMA_BA+0x14 PWMB_BA+0x14	R	PWM 数据寄存器 0	0x0000_0000
CNR1	PWMA_BA+0x18 PWMB_BA+0x18	R/W	PWM 计数寄存器 1	0x0000_0000
CMR1	PWMA_BA+0x1C PWMB_BA+0x1C	R/W	PWM 比较寄存器1	0x0000_0000
PDR1	PWMA_BA+0x20 PWMB_BA+0x20	R	PWM 数据寄存器1	0x0000_0000
CNR2	PWMA_BA+0x24 PWMB_BA+0x24	R/W	PWM 计数寄存器 2	0x0000_0000
CMR2	PWMA_BA+0x28 PWMB_BA+0x28	R/W	PWM 比较寄存器2	0x0000_0000
PDR2	PWMA_BA+0x2C PWMB_BA+0x2C	R	PWM 数据寄存器2	0x0000_0000
CNR3	PWMA_BA+0x30 PWMB_BA+0x30	R/W	PWM 计数寄存器3	0x0000_0000
CMR3	PWMA_BA+0x34 PWMB_BA+0x34	R/W	PWM 比较寄存器3	0x0000_0000
PDR3	PWMA_BA+0x38 PWMB_BA+0x38	R	PWM 数据寄存器 3	0x0000_0000
PIER	PWMA_BA+0x40	R/W	PWM 中断使能寄存器	0x0000_0000

	PWMB_BA+0x40			
PIIR	PWMA_BA+0x44 PWMB_BA+0x44	R/W	PWM 中断标志寄存器	0x0000_0000
CCR0	PWMA_BA+0x50 PWMB_BA+0x50	R/W	PWM 捕捉控制寄存器0	0x0000_0000
CCR2	PWMA_BA+0x54 PWMB_BA+0x54	R/W	PWM 捕捉控制寄存器2	0x0000_0000
CRLR0	PWMA_BA+0x58 PWMB_BA+0x58	R	PWM 捕捉上升沿锁存寄存器 (通道 0)	0x0000_0000
CFLR0	PWMA_BA+0x5C PWMB_BA+0x5C	R	PWM 捕捉下降沿锁存寄存器(通道0)	0x0000_0000
CRLR1	PWMA_BA+0x60 PWMB_BA+0x60	R	PWM 捕捉上升沿锁存寄存器通道1)	0x0000_0000
CFLR1	PWMA_BA+0x64 PWMB_BA+0x64	R	PWM 捕捉下降沿锁存寄存器 (通道1)	0x0000_0000
CRLR2	PWMA_BA+0x68 PWMB_BA+0x68	R	PWM 捕捉上升沿锁存寄存器 (通道2)	0x0000_0000
CFLR2	PWMA_BA+0x6C PWMB_BA+0x6C	R	PWM 捕捉下降沿锁存寄存器r (通道2)	0x0000_0000
CRLR3	PWMA_BA+0x70 PWMB_BA+0x70	R	PWM 捕捉上升沿锁存寄存器 (通道3)	0x0000_0000
CFLR3	PWMA_BA+0x74 PWMB_BA+0x74	R	PWM 捕捉下降沿锁存寄存器(通道3)	0x0000_0000
CAPENR	PWMA_BA+0x78 PWMB_BA+0x78	R/W	PWM 捕捉输入 0~3 使能寄存器	0x0000_0000
POE	PWMA_BA+0x7C PWMB_BA+0x7C	R/W	PWM 通道0~3输出使能寄存器	0x0000_0000
TCON	PWMA_BA+0x80 PWMB_BA+0x80	R/W	PWM 通道0~3触发控制寄存器	0x0000_0000
TSTATUS	PWMA_BA+0x84 PWMB_BA+0x84	R/W	PWM 触发状态寄存器	0x0000_0000
PSCR	PWMA_BA+0x98 PWMB_BA+0x98	R/W	PWM 同步控制寄存器	0x0000_0000

5.9.7 寄存器描述

PWM 预分频寄存器(PPR)

寄存器	偏移量	R/W	描述	复位值
PPR	PWMA_BA+0x00 PWMB_BA+0x00	R/W	PWM 预分频寄存器	0x0000_0000



位	描述
[31:24]	DZI23 通道2和通道3死区间隔设定寄存器(PWMA组的PWM2与PWM3, PWMB组的PWM6与PWM7) 该8位寄存器决定死区长度。 死区时间单位长度 = [(prescale+1)*(clock source divider)]/ PWMxy_CLK (xy 可能是 23 或者 67, 取决于PWM通道选择).
[23:16]	DZI01 通道0和通道1死区间隔设定寄存器(PWMA组的PWM0与PWM1, PWMB组的PWM4与PWM5) 该8位寄存器决定死区长度。 死区时间单位长度 = [(prescale+1)*(clock source divider)]/ PWMxy_CLK (xy 可能是 01 或者 45, 取决于PWM通道选择).
[15:8]	CP23 时钟预分频器 2 (PWMA组的PWM 定时器 2 & 3, PWMB组的PWM 定时器 6 & 7) 在输入相应的PWM计数器之前, 时钟输入根据(CP23 + 1)分频 如果CP23=0, 预分频器2输出时钟停止, 相应的PWM 计数器2和3也停止.
[7:0]	CP01 时钟预分频器0(PWMA组的PWM 定时器 0 & 1, PWMB组的PWM 定时器 4 & 5) 在输入相应的PWM计数器之前, 时钟输入根据(CP01 + 1)分频 如果CP01=0, 预分频器0输出时钟停止。相应的PWM计数器0和1也停止.

PWM 时钟选择寄存器(CSR)

寄存器	偏移量	R/W	描述	复位值
CSR	PWMA_BA+0x04 PWMB_BA+0x04	R/W	PWM 时钟源除频选择寄存器	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱	CSR3			帳隱	CSR2		
7	6	5	4	3	2	1	0
帳隱	CSR1			帳隱	CSR0		

位	描述	
[31:15]	帳隱	保留
[14:12]	CSR3	PWM定时器 3 时钟源除频选择(PWM定时器3对应于A组， PWM定时器7对应于B组) 为PWM定时器3选择时钟源除频
		CSR3 [14:12]
		输入时钟分频
		000
		2
		001
[11]	CSR2	4
		010
[10:8]	CSR2	8
		011
		16
		100
		1
[7]	帳隱	保留
[6:4]	CSR1	PWM定时器 2 时钟源除频选择(PWM定时器2 对应于A组， PWM定时器6 对应于B组) 为PWM定时器2选择时钟源除频。 (表格同CSR3)
		为PWM定时器1选择时钟源除频。 (表格同CSR3)
[3]	帳隱	保留
[2:0]	CSR0	PWM定时器 0 时钟源除频选择(PWM定时器0 对应于A组， PWM定时器4 对应于B组)

		为PWM定时器0选择时钟源除频。 (表格同CSR3)
--	--	-------------------------------

PWM 控制寄存器 (PCR)

寄存器	偏移量	R/W	描述	复位值
PCR	PWMA_BA+0x08 PWMB_BA+0x08	R/W	PWM 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PWM23TYPE	PWM01TYPE	帳隱		CH3MOD	CH3INV	CH3PINV	CH3EN
23	22	21	20	19	18	17	16
帳隱				CH2MOD	CH2INV	CH2PINV	CH2EN
15	14	13	12	11	10	9	8
帳隱				CH1MOD	CH1INV	CH1PINV	CH1EN
7	6	5	4	3	2	1	0
帳隱		DZEN23	DZEN01	CH0MOD	CH0INV	CH0PINV	CH0EN

位	描述
[31]	PWM23TYPE PWM23 对齐类型选择位 (A组的PWM2 和 PWM3 为一对，B组的PWM6 和 PWM7 为一对) 0 = 边沿对齐类型. 1 = 中心对齐类型
[30]	PWM01TYPE PWM01 对齐类型选择位(A组的PWM0 和 PWM1 为一对，B组的PWM4 和 PWM5 为一对) 0 = 边沿对齐类型. 1 = 中心对齐类型.
[29:28]	帳隱 保留
[27]	CH3MOD PWM定时器3 自动重载/单触发模式选择(A组的PWM定时器3 和B组的PWM定时器7) 0 = 单触发模式 1 = 自动重载模式 注: 如果该位的值发生改变, 会导致CNR3 和CMR3 被清0
[26]	CH3INV PWM定时器3 输出反转使能控制(A组的PWM定时器3 和B组的PWM定时器7) 0 = 反转关闭 1 = 反转打开
[25]	CH3PINV PWM定时器3 输出极性反转使能控制 (A组的PWM定时器3 和B组的PWM定时器7) 0 = 极性反转禁止. 1 = 极性反转使能.
[24]	CH3EN PWM定时器3 使能控制(A组的PWM定时器3 和B组的PWM定时器7) 0 = 相应PWM定时器停止运行 1 = 相应PWM定时器开始运行

[23:20]	帳隱	保留
[19]	CH2MOD	PWM定时器2 自动重载/单触发模式选择 (A组的PWM定时器2 和B组的PWM定时器6) 0 = 单触发模式 1 = 自动重载模式 注: 如果该位的值发生改变, 会导致CNR2 和CMR2 被清0.
[18]	CH2INV	PWM定时器2 输出反转使能控制 (A组的PWM定时器2 和B组的PWM定时器6) 0 = 反转关闭 1 = 反转打开
[17]	CH2PINV	PWM定时器2 输出极性反转使能控制 (A组的PWM 定时器 2和B组的 PWM 定时器 6) 0 = 极性反转禁止. 1 = 极性反转使能.
[16]	CH2EN	PWM定时器2 使能 (A组的PWM定时器2 和B组的PWM定时器6) 0 = 相应PWM定时器停止运行 1 = 相应PWM定时器开始运行
[15:12]	帳隱	保留
[11]	CH1MOD	PWM定时器1 自动重载/单触发模式选择(A组的PWM定时器1 和B组的PWM定时器5) 0 = 单触发模式 1 = 自动重载模式 注: 如果该位的值发生改变, 会导致CNR1 和CMR1 被清0.
[10]	CH1INV	PWM定时器1 输出反转使能 (A组的PWM定时器1 和B组的PWM定时器5) 0 = 反转关闭 1 = 反转打开
[9]	CH1PINV	PWM定时器1 输出极性反转使能控制 (A组的PWM 定时器 1和B组的 PWM 定时器 5) 0 = 极性反转禁止. 1 = 极性反转使能
[8]	CH1EN	PWM定时器1 使能控制 (A组的PWM 定时器 1和B组的 PWM 定时器 5) 0 = 相应PWM定时器停止运行 1 = 相应PWM定时器开始运行
[7:6]	帳隱	保留
[5]	DZEN23	死区发生器2使能控制(A组的PWM2 和 PWM3 为一对, B组的PWM6 和 PWM7 为一对) 0 = 禁用 1 = 使能 注: 当死区发生器使能时, PWM A组的PWM2与PWM3将成为互补对, PWM B组的PWM6与PWM7将成为互补对.
[4]	DZEN01	死区发生器0使能控制(A组的PWM0 和 PWM1 为一对, B组的PWM4 和 PWM5 为一对) 0 = 禁用 1 = 使能 注: 当死区发生器使能时, PWM A组的PWM0与PWM1将成为互补对, PWM B组的PWM4与PWM5将成为互补对.

[3]	CH0MOD	PWM定时器0 自动加载/单触发模式选择 (A组的PWM 定时器0和B组的 PWM 定时器4) 0 = 单触发模式 1 = 自动重载模式 注: 如果该位的值发生改变, 会导致CNRO 和CMRO 被清0.
[2]	CH0INV	PWM定时器0 输出反转使能控制 (A组的PWM 定时器0和B组的 PWM 定时器4) 0 = 反转关闭 1 = 反转打开
[1]	CH0PINV	PWM定时器0 输出极性反转使能控制 (A组的PWM 定时器0和B组的 PWM 定时器4) 0 = 极性反正禁止. 1 = 极性反正使能
[0]	CH0EN	PWM定时器0 使能控制 (A组的PWM 定时器0和B组的 PWM 定时器4) 0 = 相应PWM定时器停止运行 1 = 相应PWM定时器开始运行

PWM 计数器寄存器 3-0 (CNR3-0)

寄存器	偏移量	R/W	描述	复位值
CNR0	PWMA_BA+0x0C PWMB_BA+0x0C	R/W	PWM 计数器寄存器0	0x0000_0000
CNR1	PWMA_BA+0x18 PWMB_BA+0x18	R/W	PWM 计数器寄存器1	0x0000_0000
CNR2	PWMA_BA+0x24 PWMB_BA+0x24	R/W	PWM 计数器寄存器2	0x0000_0000
CNR3	PWMA_BA+0x30 PWMB_BA+0x30	R/W	PWM 计数器寄存器3	0x0000_0000

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

位	描述	
[31:16]	帳隱	保留
[15:0]	CNRx	<p>PWM 计数器载入值 CNR 决定 PWM 的周期. $\text{PWM 频率} = \text{PWMxy_CLK} / ((\text{prescale}+1) * (\text{clock divider}) * (\text{CNR}+1))$; xy代表01, 23, 45或 67, 取决于所选择的PWM通道 对于边沿对齐类型:</p> <ul style="list-style-type: none"> 占空比 = $(\text{CMR}+1)/(\text{CNR}+1)$. $\text{CMR} \geq \text{CNR}$: PWM输出总是高. $\text{CMR} < \text{CNR}$: PWM低脉冲宽度= $(\text{CNR}-\text{CMR})$ 单位; PWM高脉冲宽度= $(\text{CMR}+1)$ 单位 $\text{CMR} = 0$: PWM低脉冲宽度= (CNR) 单位; PWM高脉冲宽度= 1 单位 <p>对于中心对齐类型:</p> <ul style="list-style-type: none"> 占空比= $[(2 \times \text{CMR}) + 1]/[2 \times (\text{CNR}+1)]$ $\text{CMR} > \text{CNR}$: PWM 输出总是高

- | | |
|--|--|
| | <ul style="list-style-type: none">● CMR <= CNR: PWM 低脉冲宽度 = $2 \times (\text{CNR}-\text{CMR}) + 1$单位; PWM 高脉冲宽度 = $(2 \times \text{CMR}) + 1$单位● CMR = 0: PWM 低脉冲宽度 = $2 \times \text{CNR} + 1$单位; PWM 高脉冲宽度 = 1单位
(单位 = 一个 PWM 时钟周期). <p>注1:写数据到CNR寄存器后将在下一个PWM周期生效</p> <p>注2:CNR的值被设为0时，PWM输出总是为高</p> <p>注3:PWM工作在中心对齐模式时，CNR的值应该在0x0001 to 0xFFFF之间。如果CNR等于0x0000或者0xFFFF，PWM工作将不正常</p> |
|--|--|

PWM 比较器寄存器 3-0 (CMR3-0)

寄存器	偏移量	R/W	描述	复位值
CMR0	PWMA_BA+0x10 PWMB_BA+0x10	R/W	PWM 比较器寄存器0	0x0000_0000
CMR1	PWMA_BA+0x1C PWMB_BA+0x1C	R/W	PWM 比较器寄存器1	0x0000_0000
CMR2	PWMA_BA+0x28 PWMB_BA+0x28	R/W	PWM 比较器寄存器2	0x0000_0000
CMR3	PWMA_BA+0x34 PWMB_BA+0x34	R/W	PWM 比较器寄存器3	0x0000_0000

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

位	描述	
[31:16]	帳隱	保留
[15:0]	CMRx	<p>PWM 比较器寄存器</p> <p>CMR 决定 PWM 的占空比</p> <p>PWM 频率 = PWMxy_CLK/[(prescale+1)*(clock divider)*(CNR+1)]; xy代表 01, 23, 45 or 67, 取决于所选择的PWM通道</p> <p>对于边沿对齐类型:</p> <ul style="list-style-type: none"> 占空比 = (CMR+1)/(CNR+1) CMR >= CNR: PWM 输出总是高 CMR < CNR: PWM低脉冲宽度= (CNR-CMR) 单位; PWM高脉冲宽度 = (CMR+1) 单位. CMR = 0: PWM低脉冲宽度= (CNR) 单位; PWM 高脉冲宽度 = 1 单位. <p>对于中心对齐类型:</p> <ul style="list-style-type: none"> 占空比= [(2 x CMR) + 1]/[2 x (CNR+1)]

- | | |
|--|---|
| | <ul style="list-style-type: none">● CMR > CNR: PWM 输出总是高● CMR <= CNR: PWM 低脉冲宽度 = $2 \times (\text{CNR}-\text{CMR}) + 1$ 单位; PWM 高脉冲宽度 = $(2 \times \text{CMR}) + 1$ 单位● CMR = 0: PWM 低脉冲宽度 = $2 \times \text{CNR} + 1$ 单位; PWM 高脉冲宽度 = 1 单位
(单位 = 一个PWM时钟周期) <p>注: CMR写入数据后将在下一个PWM周期生效</p> |
|--|---|

PWM 数据寄存器 3-0 (PDR 3-0)

寄存器	偏移量	R/W	描述	复位值
PDR0	PWMA_BA+0x14 PWMB_BA+0x14	R	PWM 数据寄存器0	0x0000_0000
PDR1	PWMA_BA+0x20 PWMB_BA+0x20	R	PWM 数据寄存器1	0x0000_0000
PDR2	PWMA_BA+0x2C PWMB_BA+0x2C	R	PWM 数据寄存器2	0x0000_0000
PDR3	PWMA_BA+0x38 PWMB_BA+0x38	R	PWM 数据寄存器3	0x0000_0000

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

位	描述	
[31:16]	帳隱	保留
[15:0]	PDRx	PWM 数据寄存器 用户可以查询PDR得到16位计数器的当前值.

PWM 中断使能寄存器 (PIER)

寄存器	偏移量	R/W	描述	复位值
PIER	PWMA_BA+0x40 PWMB_BA+0x40	R/W	PWM 中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱						INT23DTYPE	INT01DTYPE
23	22	21	20	19	18	17	16
悵隱						INT23TYPE	INT01TYPE
15	14	13	12	11	10	9	8
悵隱				PWMDIE3	PWMDIE2	PWMDIE1	PWMDIE0
7	6	5	4	3	2	1	0
悵隱				PWMIE3	PWMIE2	PWMIE1	PWMIE0

位	描述	
[31:26]	悵隱	保留
[25]	INT23DTYPE	<p>PWM23 工作中断类型选择 (A组的 PWM2 和 PWM3 对和B组的 PWM6 和 PWM7 对) 0 = PWM计数器下数时如果其值等于CMRx寄存器的值，PWMDIFx 将被设。如果此时 PWM触发ADC功能使能(PWMxDTEN=1)，将触发ADC开始转换 1 = PWM计数器上数时如果其值等于CMRx寄存器的值，PWMDIFx 将被设。如果此时 PWM触发ADC功能使能(PWMxDTEN=1)，将触发ADC开始转换 注:只有在中心对齐模式下才可以设置 INT23DTYPE = 1</p>
[24]	INT01DTYPE	<p>PWM01 工作中断类型选择 (A组的 PWM0 和 PWM1 对和B组的 PWM4 和 PWM5 对) 0 = PWM计数器下数时如果其值等于CMRx寄存器的值，PWMDIFx 将被设。如果此时 PWM触发ADC功能使能(PWMxDTEN=1)，将触发ADC开始转换 1 = PWM计数器上数时如果其值等于CMRx寄存器的值，PWMDIFx 将被设。如果此时 PWM触发ADC功能使能(PWMxDTEN=1)，将触发ADC开始转换 注:只有在中心对齐模式下才可以设置 INT01DTYPE = 1</p>
[23:18]	悵隱	保留
[17]	INT23TYPE	<p>PWM23 周期中断类型选择 (A组的 PWM2 和 PWM3 对和B组的 PWM6 和 PWM7 对) 0 = PWM计数器的下溢值等于0时，PWMIFx 将被置。如果此时PWM触发ADC功能使能(PWMxTEN=1)，ADC开始转换 1 = PWM计数器的值等于CNRx时，PWMIFx 将被置。如果此时PWM触发ADC功能使能(PWMxTEN=1)，ADC开始转换 注: 只有PWM工作在中心对齐模式时，才可以设置INT23TYPE = 1</p>
[16]	INT01TYPE	<p>PWM01 周期中断类型选择 (A组的 PWM0 和 PWM1 对和B组的 PWM4 和 PWM5 对) 0 = PWM计数器的下溢值等于0时，PWMIFx 将被置。如果此时PWM触发ADC功能使能(PWMxTEN=1)，ADC开始转换</p>

		1 = PWM计数器的值等于CNRx时，PWMIFx 将被置。如果此时PWM触发ADC功能使能(PWMxTEN=1)，ADC开始转换 注：只有PWM工作在中心对齐模式时，才可以设置INT01TYPE = 1.
[15:12]	悵隱	保留
[11]	PWMDIE3	PWM 通道 3 工作中断使能控制 0 = 关闭 PWM 通道 3 工作中断。 1 = 使能 PWM 通道 3 工作中断.
[10]	PWMDIE2	PWM 通道 2 工作中断使能控制 0 = 关闭 PWM 通道 2 工作中断 . 1 = 使能 PWM 通道 2 工作中断.
[9]	PWMDIE1	PWM 通道 1 工作中断使能控制 0 = 关闭 PWM 通道 1 工作中断 . 1 = 使能 PWM 通道 1 工作中断.
[8]	PWMDIE0	PWM 通道 0 工作中断使能控制 0 = 关闭 PWM 通道 0 工作中断 . 1 = 使能 PWM 通道 0 工作中断.
[7:4]	悵隱	保留
[3]	PWMIE3	PWM 通道 3 周期中断使能控制 0 = 关闭 PWM 通道 3 周期中断 . 1 = 使能 PWM 通道 3 周期中断.
[2]	PWMIE2	PWM 通道 2 周期中断使能控制 0 = 关闭 PWM 通道 2 周期中断 . 1 = 使能 PWM 通道 2 周期中断.
[1]	PWMIE1	PWM 通道 1 周期中断使能控制 0 = 关闭 PWM 通道 1 周期中断 . 1 = 使能 PWM 通道 1 周期中断.
[0]	PWMIE0	PWM 通道 0 周期中断使能控制 0 = 关闭 PWM 通道 0 周期中断 . 1 = 使能 PWM 通道 0 周期中断.

PWM 中断标志寄存器 (PIIR)

寄存器	偏移量	R/W	描述	复位值
PIIR	PWMA_BA+0x44 PWMB_BA+0x44	R/W	PWM 中断标志寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱				PWMDIF3	PWMDIF2	PWMDIF1	PWMDIF0
7	6	5	4	3	2	1	0
悵隱				PWMIF3	PWMIF2	PWMIF1	PWMIF0

位	描述	
[31:12]	悵隱	保留
[11]	PWMDIF3	<p>PWM 通道 3 工作中断状态</p> <p>当 INTDTYPE23 = 0时，PWM通道3下数计数器的值等于CMR3时，该位被置1.</p> <p>当 INTDTYPE23 = 1时，PWM通道3上数计数器的值等于CMR3时，该位被置1.</p> <p>注: 写 1 清除该位.</p>
[10]	PWMDIF2	<p>PWM 通道 2 工作中断状态</p> <p>当 INTDTYPE23 = 0时，PWM通道2下数计数器的值等于CMR2时，该位被置1.</p> <p>当 INTDTYPE23 = 1时，PWM通道2上数计数器的值等于CMR2时，该位被置1.</p> <p>注: 写 1 清除该位.</p>
[9]	PWMDIF1	<p>PWM 通道 1 工作中断状态</p> <p>当INTDTYPE01 = 0时，PWM通道1下数计数器的值等于CMR1时，该位被置1.</p> <p>当INTDTYPE01 = 1时，PWM通道1上数计数器的值等于CMR1时，该位被置1.</p> <p>注: 写 1 清除该位</p>
[8]	PWMDIFO	<p>PWM 通道 0 工作中断状态</p> <p>当INTDTYPE01 = 0时，PWM通道0 下数计数器的值等于CMR0时，该位被置1.</p> <p>当INTDTYPE01 = 1时，PWM通道0 上数计数器的值等于CMR0时，该位被置1.</p> <p>注: 写 1 清除该位</p>
[7:4]	悵隱	保留
[3]	PWMIF3	<p>PWM 通道3 周期中断状态</p> <p>INTTYPE23 = 0 · 当 PWM通道3向下计数至0时，硬件将该位置1。</p>

		INTTYPE23 = 1，当PWM通道3向上计数至CNR3时，硬件将该位置1。 注：软件写1清该位。
[2]	PWMIF2	PWM 通道2 周期中断状态 INTTYPE23 = 0，当PWM通道2向下计数至0时，硬件将该位置1。 INTTYPE23 = 1，当PWM通道2向上计数至CNR2时，硬件将该位置1。 注：软件写1清该位。
[1]	PWMIF1	PWM 通道1 周期中断状态 INTTYPE01 = 0，当PWM通道1向下计数至0时，硬件将该位置1。 INTTYPE01 = 1，当PWM通道1向上计数至CNR1时，硬件将该位置1。 注：软件写1清该位。
[0]	PWMIF0	PWM 通道0 周期中断状态 INTTYPE01 = 0，当PWM通道0向下计数至0时，硬件将该位置1。 INTTYPE01 = 1，当PWM通道0向上计数至CNR0 时，硬件将该位置1。 注：软件写1清该位。

捕获控制寄存器 (CCR0)

寄存器	偏移量	R/W	描述	复位值
CCR0	PWMA_BA+0x50 PWMB_BA+0x50	R/W	PWM 捕获控制寄存器 0	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
CFLRI1	CRLRI1	帳隱	CAPIF1	CAPCH1EN	CFL_IE1	CRL_IE1	INV1
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
CFLRI0	CRLRI0	帳隱	CAPIF0	CAPCH0EN	CFL_IE0	CRL_IE0	INV0

位	描述	
[31:24]	帳隱	保留
[23]	CFLRI1	CFLR1锁存指示位控制 PWM组输入通道1有下降沿转变时， PWM向下计数器的值锁存到CFLR1寄存器， 同时该位由硬件置位。 注： 该位写1清0.
[22]	CRLRI1	CRLR1锁存指示位控制 PWM组输入通道1有上升沿转变时， PWM向下计数器的值锁存到CRLR1寄存器， 同时该位由硬件置位。 注： 该位写1清0.
[21]	帳隱	保留
[20]	CAPIF1	捕捉通道1中断标志 如果PWM组通道1上升沿锁存中断使能(CRL_IE1=1), PWM组通道1有上升沿转变将使CAPIF1为高； 同样, 如果下降沿锁存中断使能(CFL_IE1=1), 下降沿转变将使CAPIF1 为高。 注： 该位写1清0.
[19]	CAPCH1EN	捕捉器通道1使能控制 0 = 禁用PWM组通道1的捕捉功能 1 = 使能PWM组通道1的捕捉功能。 注1： 使能时，捕捉锁存PWM计数器并保存到CRLR(上升沿锁定)和CFLR(下降沿锁定)。 注2： 禁用时，捕捉器不更新CRLR和CFLR，并禁用PWM组通道1捕获中断。
[18]	CFL_IE1	通道1下降沿锁存中断使能控制 0 = 禁用下降沿锁存中断

		<p>1 = 使能下降沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道1有下降沿转变, 捕捉器产生中断.</p>
[17]	CRL_IE1	<p>通道1上升沿锁存中断使能控制 0 = 禁用上升沿锁存中断 1 = 使能上升沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道1有上升沿转变, 捕捉器产生中断</p>
[16]	INV1	<p>捕获通道1反向使能控制 0 = 反向关闭 1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入。</p>
[15:8]	保留	保留
[7]	CFLRIO	<p>CFLR0锁存指示位控制 在PWM组输入通道0有下降沿转变时, PWM向下计数器的值锁存到CFLR0寄存器, 同时该位由硬件置位。 注: 该位写1清0.</p>
[6]	CRLRIO	<p>CRLR0锁存指示位控制 在PWM组输入通道0有上升沿转变时, PWM向下计数器的值锁存到CRLR0寄存器, 同时该位由硬件置位。 注: 该位写1清0.</p>
[5]	保留	保留
[4]	CAPIFO	<p>捕捉通道0中断标志 如果PWM组通道0上升沿锁存中断使能(CRL_IE0=1), PWM组通道0有上升沿转变将使CAPIFO为高; 同样, 如果下降沿锁存中断使能(CFL_IE0=1), 下降沿转变将使CAPIFO为高. 注: 该位写1清0.</p>
[3]	CAPCH0EN	<p>捕捉器通道0 使能控制 0 = 禁用PWM组通道0 的捕捉功能 1 = 使能PWM组通道0 的捕捉功能. 注1: 使能时, 捕捉锁存PWM计数器并保存到CRLR(上升沿锁定)和CFLR(下降沿锁定). 注2: 禁用时, 捕捉器不更新CRLR和CFLR, 并禁用PWM组通道0捕获中断.</p>
[2]	CFL_IE0	<p>通道0 下降沿锁存中断使能控制 0 = 禁用下降沿锁存中断 1 = 使能下降沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道0有下降沿转变, 捕捉器产生中断.</p>
[1]	CRL_IE0	<p>通道0 上升沿锁存中断使能控制 0 = 禁用上升沿锁存中断 1 = 使能上升沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道0有上升沿转变, 捕捉器产生中断</p>
[0]	INV0	<p>捕获通道0反向使能控制 0 = 反向关闭 1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入。</p>

捕获控制寄存器 (CCR2)

寄存器	偏移量	R/W	描述	复位值
CCR2	PWMA_BA+0x54 PWMB_BA+0x54	R/W	PWM 捕获控制寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
CFLRI3	CRLRI3	帳隱	CAPIF3	CAPCH3EN	CFL_IE3	CRLIE3	INV3
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
CFLRI2	CRLRI2	帳隱	CAPIF2	CAPCH2EN	CFLIE2	CRLIE2	INV2

位	描述	
[31:24]	帳隱	保留
[23]	CFLRI3	CFLR3 锁存指示位控制 PWM组输入通道3有下降沿转变时， PWM向下计数器的值锁存到CFLR3寄存器， 同时该位由硬件置位。 注： 该位写1清0.
[22]	CRLRI3	CRLR3锁存指示位控制 PWM组输入通道3有上升沿转变时， PWM向下计数器的值锁存到CRLR3寄存器， 同时该位由硬件置位。 注： 该位写1清0.
[21]	帳隱	保留
[20]	CAPIF3	捕捉通道3中断标志 如果PWM组通道3 上升沿锁存中断使能(CRLIE3=1), PWM组通道3有上升沿转变将使CAPIF3为高； 同样, 如果下降沿锁存中断使能(CFLIE3=1), 下降沿转变将使CAPIF3为高. 注： 该位写1清0.
[19]	CAPCH3EN	捕捉器通道3 使能控制 0 = 禁用PWM组通道3的捕捉功能 1 = 使能PWM组通道3的捕捉功能. 注1： 使能时, 捕捉锁存PWM计数器并保存到CRLR(上升沿锁定)和CFLR(下降沿锁定). 注2： 禁用时, 捕捉器不更新CRLR和CFLR, 并禁用PWM组通道3捕获中断.
[18]	CFLIE3	通道3 下降沿锁存中断使能控制 0 = 禁用下降沿锁存中断

		<p>1 = 使能下降沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道3有下降沿转变, 捕捉器产生中断.</p>
[17]	CRL_IE3	<p>通道3 上升沿锁存中断使能控制 0 = 禁用上升沿锁存中断 1 = 使能上升沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道3有上升沿转变, 捕捉器产生中断</p>
[16]	INV3	<p>捕获通道3 反向使能控制 0 = 反向关闭 1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入</p>
[15:8]	保留	保留
[7]	CFLRI2	<p>CFLR2 锁存指示位控制 在PWM组输入通道2有下降沿转变时, PWM向下计数器的值锁存到CFLR2寄存器, 同时该位由硬件置位。 注: 该位写1清0.</p>
[6]	CRLRI2	<p>CRLR2 锁存指示位控制 在PWM组输入通道2有上升沿转变时, PWM向下计数器的值锁存到CRLR2寄存器, 同时该位由硬件置位。 注: 该位写1清0.</p>
[5]	保留	保留
[4]	CAPIF2	<p>捕捉通道2 中断标志 如果PWM组通道2上升沿锁存中断使能(CRL_IE2=1), PWM组通道2有上升沿转变将使CAPIF2为高; 同样, 如果下降沿锁存中断使能(CFL_IE2=1), 下降沿转变将使CAPIF2为高. 注: 该位写1清0.</p>
[3]	CAPCH2EN	<p>捕捉器通道2 使能控制 0 = 禁用PWM组通道2的捕捉功能 1 = 使能PWM组通道2的捕捉功能. 注1: 使能时, 捕捉锁存PWM计数器并保存到CRLR(上升沿锁定)和CFLR(下降沿锁定). 注2: 禁用时, 捕捉器不更新CRLR和CFLR, 并禁用PWM组通道2捕获中断.</p>
[2]	CFL_IE2	<p>通道2下降沿锁存中断使能控制 0 = 禁用下降沿锁存中断 1 = 使能下降沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道2有下降沿转变, 捕捉器产生中断</p>
[1]	CRL_IE2	<p>通道2上升沿锁存中断使能控制 0 = 禁用上升沿锁存中断 1 = 使能上升沿锁存中断 注: 使能时, 如果捕捉器检测到PWM组通道2有上升沿转变, 捕捉器产生中断</p>
[0]	INV2	<p>捕获通道2 反向使能控制 0 = 反向关闭 1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入</p>

捕捉器上升沿锁存寄存器 3-0 (CRLR3-0)

寄存器	偏移量	R/W	描述	复位值
CRLR0	PWMA_BA+0x58 PWMB_BA+0x58	R	PWM 捕捉上升沿锁存寄存器(通道 0)	0x0000_0000
CRLR1	PWMA_BA+0x60 PWMB_BA+0x60	R	PWM 捕捉上升沿锁存寄存器(通道 1)	0x0000_0000
CRLR2	PWMA_BA+0x68 PWMB_BA+0x68	R	PWM 捕捉上升沿锁存寄存器(通道 2)	0x0000_0000
CRLR3	PWMA_BA+0x70 PWMB_BA+0x70	R	PWM 捕捉上升沿锁存寄存器(通道 3)	0x0000_0000



位	描述	
[31:16]	帳隱	保留
[15:0]	CRLRx	捕捉上升沿锁存寄存器 当通道0/1/2/3 有上升沿时转变时，锁存PWM计数器的值到这个寄存器

捕捉下降沿锁存寄存器 3-0 (CFLR3-0)

寄存器	偏移量	R/W	描述	复位值
CFLR0	PWMA_BA+0x5C PWMB_BA+0x5C	R	PWM 捕捉下降沿锁存寄存器(通道 0)	0x0000_0000
CFLR1	PWMA_BA+0x64 PWMB_BA+0x64	R	PWM 捕捉下降沿锁存寄存器(通道1)	0x0000_0000
CFLR2	PWMA_BA+0x6C PWMB_BA+0x6C	R	PWM 捕捉下降沿锁存寄存器(通道2)	0x0000_0000
CFLR3	PWMA_BA+0x74 PWMB_BA+0x74	R	PWM 捕捉下降沿锁存寄存器(通道3)	0x0000_0000



位	描述	
[31:16]	帳隱	保留
[15:0]	CFLRx	捕捉下降沿锁存寄存器 当通道0/1/2/3 有下降沿转变时，锁存PWM计数器的值到这个寄存器.

捕获输入使能寄存器 (CAPENR)

寄存器	偏移量	R/W	描述	复位值
CAPENR	PWMA_BA+0x78 PWMB_BA+0x78	R/W	PWM 捕捉输入0~3 使能寄存器	0x0000_0000

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

位	描述	
[31:4]	悶隱	保留
[3:0]	CAPENR	<p>捕捉输入使能寄存器 0 = 关闭捕获输入(PWMx 复用脚输入对捕捉器不产生影响) 1 = 打开捕获输入 (PWMx 复用脚将影响捕捉器功能.)</p> <p>CAPENR</p> <p>位 3210 用于 PWM A组</p> <p>位 xxx1 → 使能捕捉通道0。捕捉输入通道0可以选择从 P2.0或者P4.0 输入。通过设置多功能引脚寄存器用户只能选择1个引脚</p> <p>位 xx1x → 使能捕捉通道1。捕捉输入通道1可以选择从 P2.1或者P4.1 输入。通过设置多功能引脚寄存器用户只能选择1个引脚</p> <p>位 x1xx → 使能捕捉通道2。捕捉输入通道2可以选择从 P2.2或者P4.2 输入。通过设置多功能引脚寄存器用户只能选择1个引脚</p> <p>位 1xxx → 使能捕捉通道3。捕捉输入通道0可以选择从 P2.3或者P4.3 输入。通过设置多功能引脚寄存器用户只能选择1个引脚</p> <p>位 3210用于 PWM B组</p> <p>位 xxx1 → 使能捕捉通道0。捕捉输入通道0可以通过设置多功能引脚寄存器选择从 P2.4 输入</p> <p>位 xx1x → 使能捕捉通道1。捕捉输入通道0可以通过设置多功能引脚寄存器选择从 P2.5 输入</p> <p>位 x1xx → 使能捕捉通道2。捕捉输入通道0可以通过设置多功能引脚寄存器选择从 P2.6 输入</p> <p>位 1xxx → 使能捕捉通道3。捕捉输入通道3可以通过设置多功能引脚寄存器选择从 P2.7 输入</p>

PWM 输出使能寄存器 (POE)

寄存器	偏移量	R/W	描述	复位值
POE	PWMA_BA+0x7C PWMB_BA+0x7C	R/W	PWM 输出使能寄存器(通道0~3)	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
帳隱				PWM3	PWM2	PWM1	PWM0

位	描述	
[31:4]	帳隱	保留
[3]	PWM3	PWM 通道3输出使能寄存器 0 = 禁止PWM通道3输出到IO引脚 1 = 使能PWM通道3输出到IO引脚 注: GPIO相应管脚必须切换到PWM功能
[2]	PWM2	PWM 通道2输出使能寄存器 0 = 禁止PWM通道2输出到IO引脚 1 = 使能PWM通道2输出到IO引脚 注: GPIO相应管脚必须切换到PWM功能
[1]	PWM1	PWM 通道1输出使能寄存器 0 = 禁止PWM通道1输出到IO引脚 1 = 使能PWM通道1输出到IO引脚 注: GPIO相应管脚必须切换到PWM功能
[0]	PWM0	PWM 通道0输出使能寄存器 0 = 禁止PWM通道0输出到IO引脚 1 = 使能PWM通道0输出到IO引脚 注: GPIO相应管脚必须切换到PWM功能

PWM 触发控制寄存器 (TCON)

寄存器	偏移量	R/W	描述	复位值
TCON	PWMA_BA+0x80 PWMB_BA+0x80	R/W	PWM 通道0~3 触发控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱				PWM3DTEN	PWM2DTEN	PWM1DTEN	PWM0DTEN
7	6	5	4	3	2	1	0
帳隱				PWM3TEN	PWM2TEN	PWM1TEN	PWM0TEN

位	描述	
[31:12]	帳隱	保留
[11]	PWM3DTEN	PWM 通道3 工作触发 ADC 使能控制 0 = 关闭PWM 通道3 工作触发ADC 功能. 1 = 使能PWM通道 3 工作触发ADC功能. PWM 工作在边沿对齐类型时，当PWM 通道3 计数器下数等于CMR3 时，使能该位可以使PWM 触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM 通道3计数器上数/下数(根据INT23DTYPE的设定)等于CMR3时，使能该位可以使PWM 触发ADC开始转换
[10]	PWM2DTEN	PWM 通道2 工作触发 ADC 使能控制 0 = 关闭PWM 通道2 工作触发ADC 功能. 1 = 使能PWM通道 2 工作触发ADC功能. PWM 工作在边沿对齐类型时，当PWM 通道2 计数器下数等于CMR2 时，使能该位可以使PWM 触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM 通道2计数器上数/下数(根据INT23DTYPE的设定)等于CMR2时，使能该位可以使PWM 触发ADC开始转换
[9]	PWM1DTEN	PWM 通道1 工作触发 ADC 使能控制 0 = 关闭PWM 通道1 工作触发ADC 功能. 1 = 使能PWM通道 1 工作触发ADC功能. PWM 工作在边沿对齐类型时，当PWM 通道1 计数器下数等于CMR1 时，使能该位可以使PWM 触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM 通道1计数器上数/下数(根据INT01DTYPE的设定)等于CMR1时，使能该位可以使PWM 触发ADC开始转换

[8]	PWM0DTEN	PWM 通道0 工作触发 ADC 使能控制 0 = 关闭PWM 通道0 工作触发ADC 功能. 1 = 使能PWM通道 0 工作触发ADC功能. PWM 工作在边沿对齐类型时，当PWM 通道0 计数器下数等于CMR0 时，使能该位可以使PWM 触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM 通道0计数器上数/下数(根据INT01DTYPE的设定)等于CMR1时，使能该位可以使PWM 触发ADC开始转换
[7:4]	保留	保留
[3]	PWM3TEN	PWM 通道3 周期触发ADC使能控制 0 = 关闭PWM 通道3 周期触发ADC 功能. 1 = 使能PWM通道 3 周期触发ADC 功能. PWM 工作在边沿对齐类型时，当PWM通道3 计数器下数等于0 时，使能该位可以使PWM触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM通道3 计数器上数等于(CNR3+1)或者下数等于0时(根据INT23TYPE的设定)，使能该位可以使PWM 触发ADC开始转换.
[2]	PWM2TEN	PWM 通道2 周期触发ADC使能控制 0 = 关闭PWM 通道2 周期触发ADC 功能. 1 = 使能PWM通道 2 周期触发ADC 功能. PWM 工作在边沿对齐类型时，当PWM通道2 计数器下数等于0 时，使能该位可以使PWM触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM通道2 计数器上数等于(CNR2+1)或者下数等于0时(根据INT23TYPE的设定)，使能该位可以使PWM 触发ADC开始转换.
[1]	PWM1TEN	PWM 通道1 周期触发ADC使能控制 0 = 关闭PWM 通道1 周期触发ADC 功能. 1 = 使能PWM通道 1 周期触发ADC 功能. PWM 工作在边沿对齐类型时，当PWM通道1 计数器下数等于0 时，使能该位可以使PWM触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM通道1 计数器上数等于(CNR1+1)或者下数等于0时(根据INT01TYPE的设定)，使能该位可以使PWM 触发ADC开始转换.
[0]	PWM0TEN	PWM 通道0 周期触发ADC使能控制 0 = 关闭PWM 通道0 周期触发ADC 功能. 1 = 使能PWM通道 0 周期触发ADC 功能. PWM 工作在边沿对齐类型时，当PWM通道0 计数器下数等于0 时，使能该位可以使PWM触发ADC开始转换 PWM 工作在中心对齐类型时，当PWM通道0 计数器上数等于(CNR0+1)或者下数等于0时(根据INT01TYPE的设定)，使能该位可以使PWM 触发ADC开始转换.

PWM 触发状态寄存器 (TSTATUS)

寄存器	偏移量	R/W	描述	复位值
TSTATUS	PWMA_BA+0x84 PWMB_BA+0x84	R/W	PWM 触发状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
悶隱							
7	6	5	4	3	2	1	0
悶隱				PWM3TF	PWM2TF	PWM1TF	PWM0TF

位	描述	
[31:4]	悶隱	保留
[3]	PWM3TF	PWM 通道3 触发 ADC 标志 当PWM通道3 触发ADC条件匹配时，该位由硬件置1。此时如果ADC选择由PWM触发转换，ADC转换将开始。 注：该位写1清0。
[2]	PWM2TF	PWM 通道2 触发 ADC 标志 当PWM通道2 触发ADC条件匹配时，该位由硬件置1。此时如果ADC选择由PWM触发转换，ADC转换将开始。 注：该位写1清0。
[1]	PWM1TF	PWM 通道1 触发 ADC 标志 当PWM通道1 触发ADC条件匹配时，该位由硬件置1。此时如果ADC选择由PWM触发转换，ADC转换将开始。 注：该位写1清0。
[0]	PWM0TF	PWM 通道0 触发 ADC 标志 当PWM通道0 触发ADC条件匹配时，该位由硬件置1。此时如果ADC选择由PWM触发转换，ADC转换将开始。 注：该位写1清0。

PWM 同步控制寄存器 (PSCR)

寄存器	偏移量	R/W	描述	复位值
PSCR	PWMA_BA+0x98 PWMB_BA+0x98	R/W	PWM 同步控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留							PSSEN3
23	22	21	20	19	18	17	16
保留							PSSEN2
15	14	13	12	11	10	9	8
保留							PSSEN1
7	6	5	4	3	2	1	0
保留							PSSEN0

位	描述	
[31:25]	保留	保留
[24]	PSSEN3	通道3 PWM定时器同步开始使能控制 0 = 关闭通道3 PWM 定时器 同步开始. 1 = 使能通道3 PWM 定时器 同步开始. 如果该位设为1，当软件写1到PWMA组的CH0EN(PCR[0])启动定时器时，PWM组通道3的定时器将和PWMA组通道0的定时器同步启动
[23:17]	保留	保留
[16]	PSSEN2	通道2 PWM定时器同步开始使能控制 0 = 关闭通道2 PWM 定时器 同步开始. 1 = 使能通道2 PWM 定时器 同步开始. 如果该位设为1，当软件写1到PWMA组的CH0EN(PCR[0])启动定时器时，PWM组通道2的定时器将和PWMA组通道0的定时器同步启动
[15:9]	保留	保留
[8]	PSSEN1	通道1 PWM定时器同步开始使能控制 0 = 关闭通道1 PWM 定时器 同步开始. 1 = 使能通道1 PWM 定时器 同步开始. 如果该位设为1，当软件写1到PWMA组的CH0EN(PCR[0])启动定时器时，PWM组通道1的定时器将和PWMA组通道0的定时器同步启动
[7:1]	保留	保留
[0]	PSSEN0	通道0 PWM定时器同步开始使能控制 0 = 关闭通道0 PWM 定时器 同步开始.

		<p>1 = 使能通道0 PWM 定时器 同步开始。 如果该位设为1，当软件写1到PWMA组的CH0EN(PCR[0])启动定时器时，PWM组通道0的定时器将和PWMA组通道0的定时器同步启动</p>
--	--	--

5.10 增强型PWM发生器(NUC029FAE)

5.10.1 概述

NuMicro® NUC029FAE微控制器PWM支持6路PWM发生器，可以配置为相互独立的3路PWM输出PWM2, PWM3和PWM5，或配置成3对分别带有可编程死区发生器的互补PWM，通道(PWM0, PWM1), (PWM2, PWM3) 和 (PWM4, PWM5)。

每对PWM共用一个8位预分频器。每个PWM通道又自带一个时钟分频器可以提供5种分频系数(1, 1/2, 1/4, 1/8, 1/16)。每路PWM输出有一个独立的16位计数器用以PWM周期控制，和一个16位的比较器用以调节占空比。6组发生器提供12路独立PWM中断标志，相关PWM通道的周期和占空比如果与计数器相符，PWM中断将会被硬件置1。每一路PWM中断各有中断使能位。每个PWM发生器可以配置成单次模式（产生一个PWM信号周期）或者循环模式（连续输出PWM波形）

为了防止PWM输出引脚输出不稳定波形，16位周期向下计数器和比较器带有双缓冲区。当用户向计数器和比较器寄存器写数据时，所写数据会在下一个周期加载。这种双缓存的特质保证了PWM稳定平滑的输出。

除了PWM，电机控制还需要Timer、ACMP和ADC共同工作。为了更精确的控制电机，我们提供一些寄存器，不仅可以配置PWM同时也可以配置Timer、ADC和ACMP。这样做，可以节省更多的CPU时间且便于控制电机，尤其是在直流无刷电机中。

5.10.2 特性

PWM有以下特性：

- 6个独立的16位PWM占空比控制单元控制最多6个PWM输出：
 - 3路独立的输出—PWM2, PWM3 和 PWM5
 - 三组互补 PWM 对,每组 PWM 对的一根引脚与另外一根引脚互补，且可以插入可编程的死区时间—(PWM0, PWM1), (PWM2, PWM3) 和 (PWM4, PWM5)
- 支持组控制位— PWM2 和 PWM4 与 PWM0, PWM3输出同步， PWM5与PWM1输出同步
- PWM有单次模式（只支持边沿对齐类型）或者自动装载模式
- 16位分辨率
- 支持边沿对齐，中心对齐三种模式
- 一对PWM中，支持可编程死区插入
- PWM0到PWM5每路PWM有独立的极性控制
- 硬件故障刹车保护
 - 两种中断类型:
 - 向下计数器比较匹配（边沿对齐和中心对齐）或者下溢（边沿对齐模式）的

时候，提出同步中断请求。

■ 发生外部故障刹车请求条件

◆ BKP0: EINT0 或 CPO1

- PWM信号缺省状态下为正逻辑，之后的PWM口的有效状态由极性控制寄存器来控制
- 支持每个通道独立触发ADC中断，包括：上升CMR比较（中心对齐模式），CNR比较（中心对齐模式），下降CMR比较，周期值比较，。
- 定时器比较事件可以触发PWM，实现直流无刷电机相切换
- 支持ACMP输出事件触发PWM强制输出最长一个周期的低电平，该特性可用于步进电机驱动
- 支持中断累计功能

5.10.3 方框图

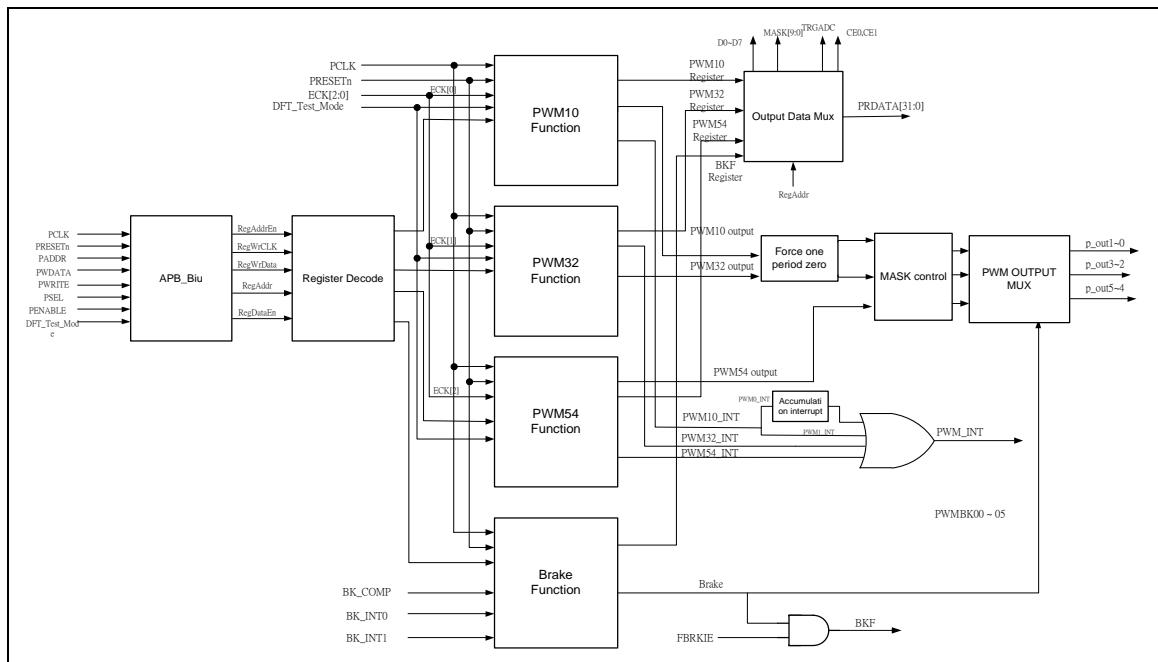


图 5-67 PWM方框图

下图列举出了PWM对的架构 (例如. PWM-Timer 0/1 组成一对, PWM-Timer 2/3 组成另一对, 等等).

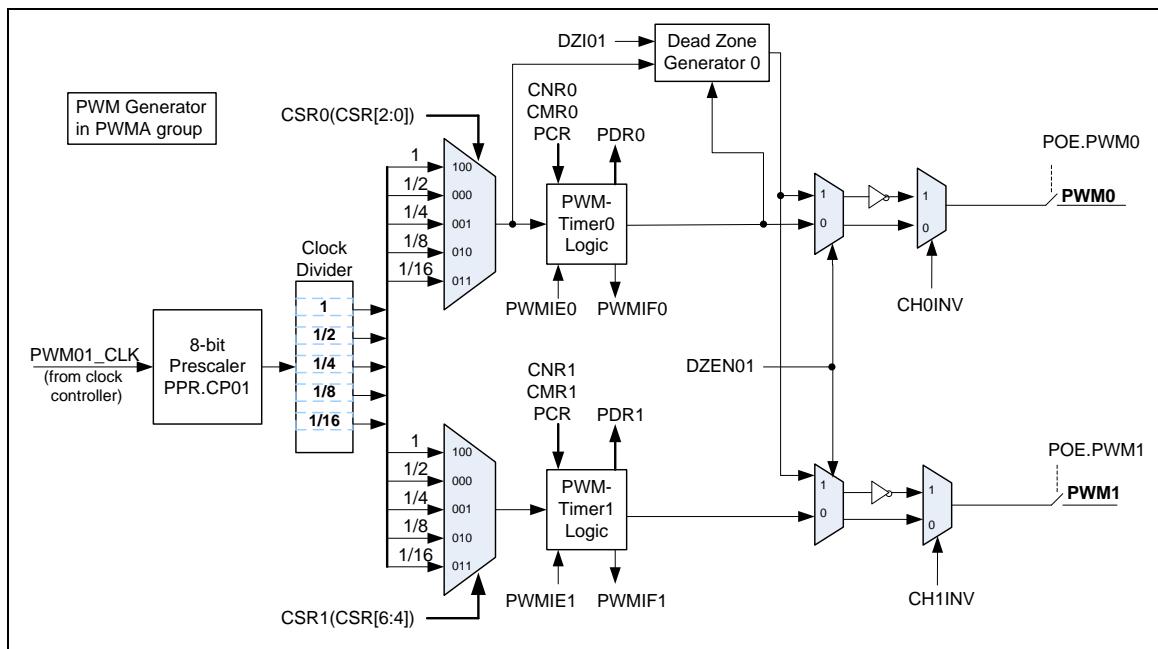


图 5-68 PWM 发生器 0 架构框图

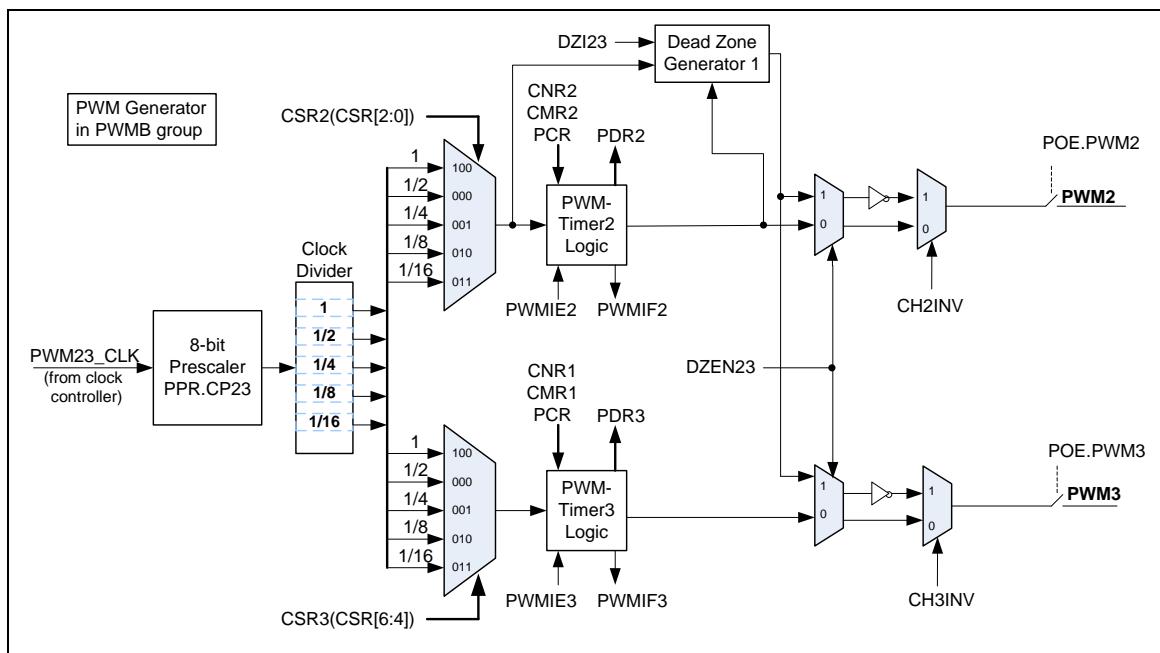


图 5-69 PWM发生器2架构框图

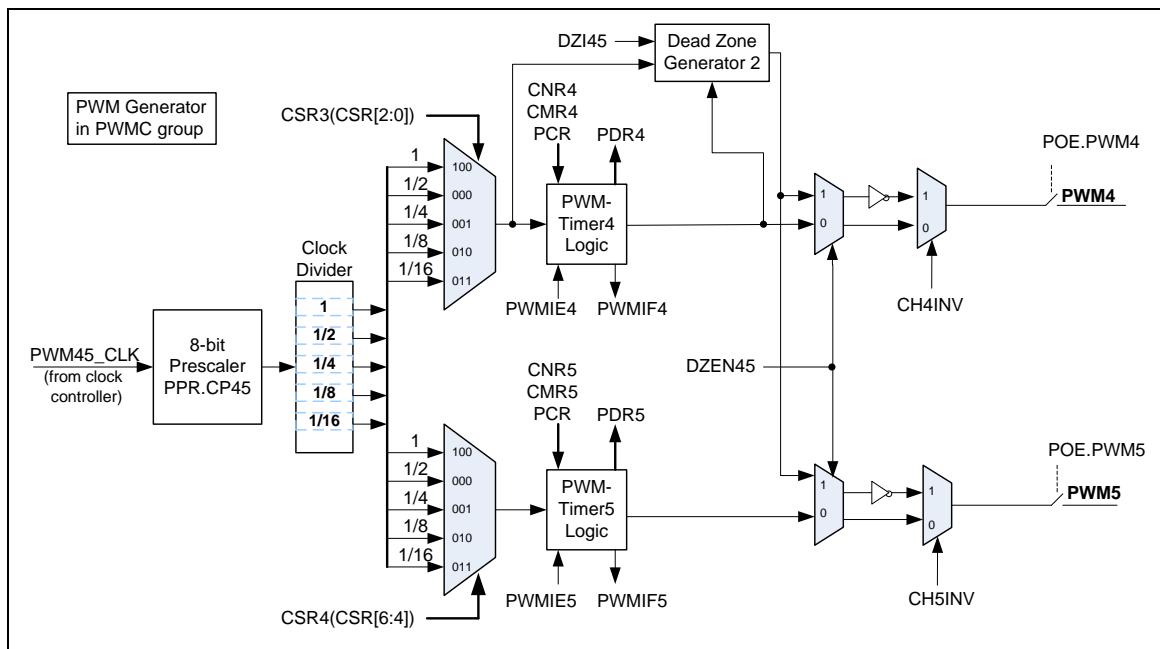


图 5-70 PWM 发生器 4 架构框图

5.10.4 基本配置

PWM的管脚功能选择在P0_MFP 和 P2_MFP寄存器中配置。

PWM的时钟可以在APBCLK[22:20] 位被使能。 PWM 时钟源必须为HCLK。

5.10.5 功能描述

5.10.5.1 PWM-Timer 的工作模式

支持2种公式模式：边沿对齐，中心对齐模式

以下是公式周期和占空比的计算公式：

边沿对齐（向下计数）：

$$\text{占空比} = (\text{CMR}+1) / (\text{CNR}+1)$$

$$\text{高电平时间} = (\text{CMP} + 1) * (\text{时钟周期})$$

$$\text{周期} = (\text{CNR}+1) * (\text{时钟周期})$$

中心对齐（上下计数）：

$$\text{占空比} = (\text{CNR} - \text{CMR}) / (\text{CNR}+1)$$

$$\text{高电平时间} = (\text{CNR} - \text{CMR}) * 2 * (\text{时钟周期})$$

$$\text{周期} = (\text{CNR}+1) * 2 * (\text{时钟周期})$$

边沿对齐PWM（向下计数）

在边沿对齐PWM输出模式下，16位PWM计数器会在每个周期从CNRn开始向下计数，与上个周期CMRn值进行比较，当比较值相符就会使对应通道PWMn输出为高电平。计数器会继续向下计数至0；这样会触发PWMn输出为低电平，当前CMRn和CNRn值在CHnMODE =1情况下会更新，如果PWM中断使能(PIER.n=1)，将请求PWM中断产生。

下图描述了边沿对齐模式下PWM的时序流程。

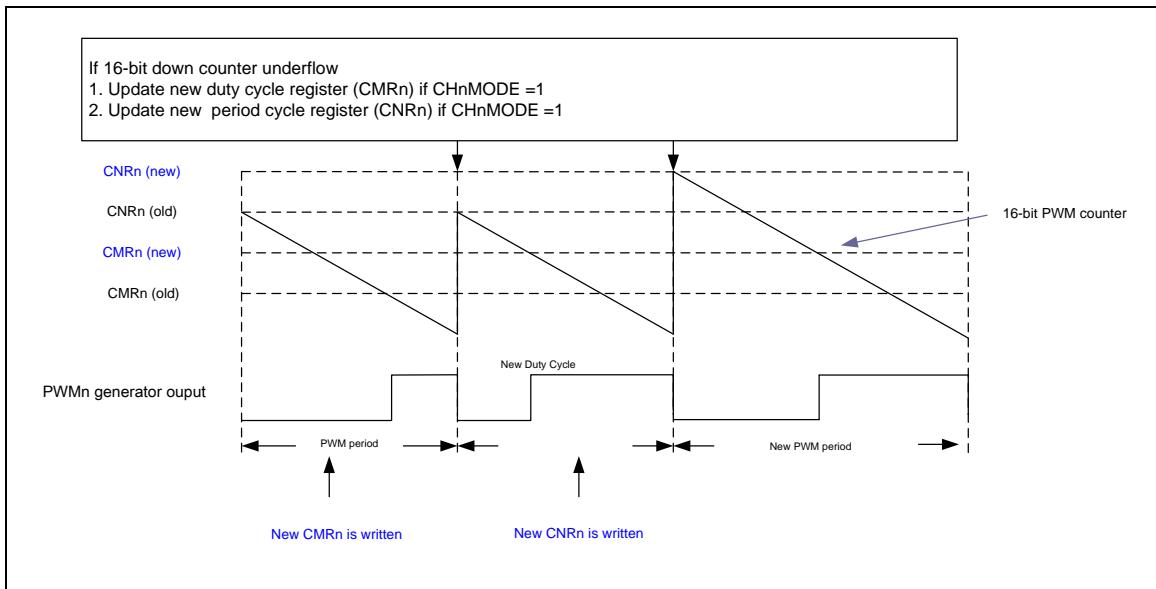


图 5-71 边沿对齐模式 PWM

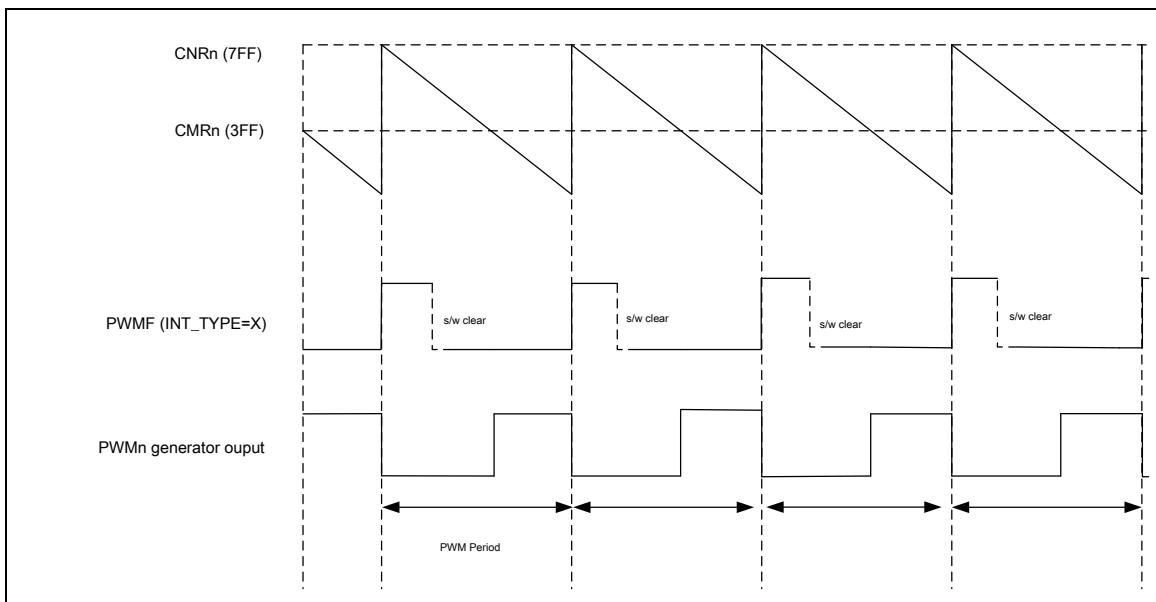


图 5-72 PWM边沿对齐时序图

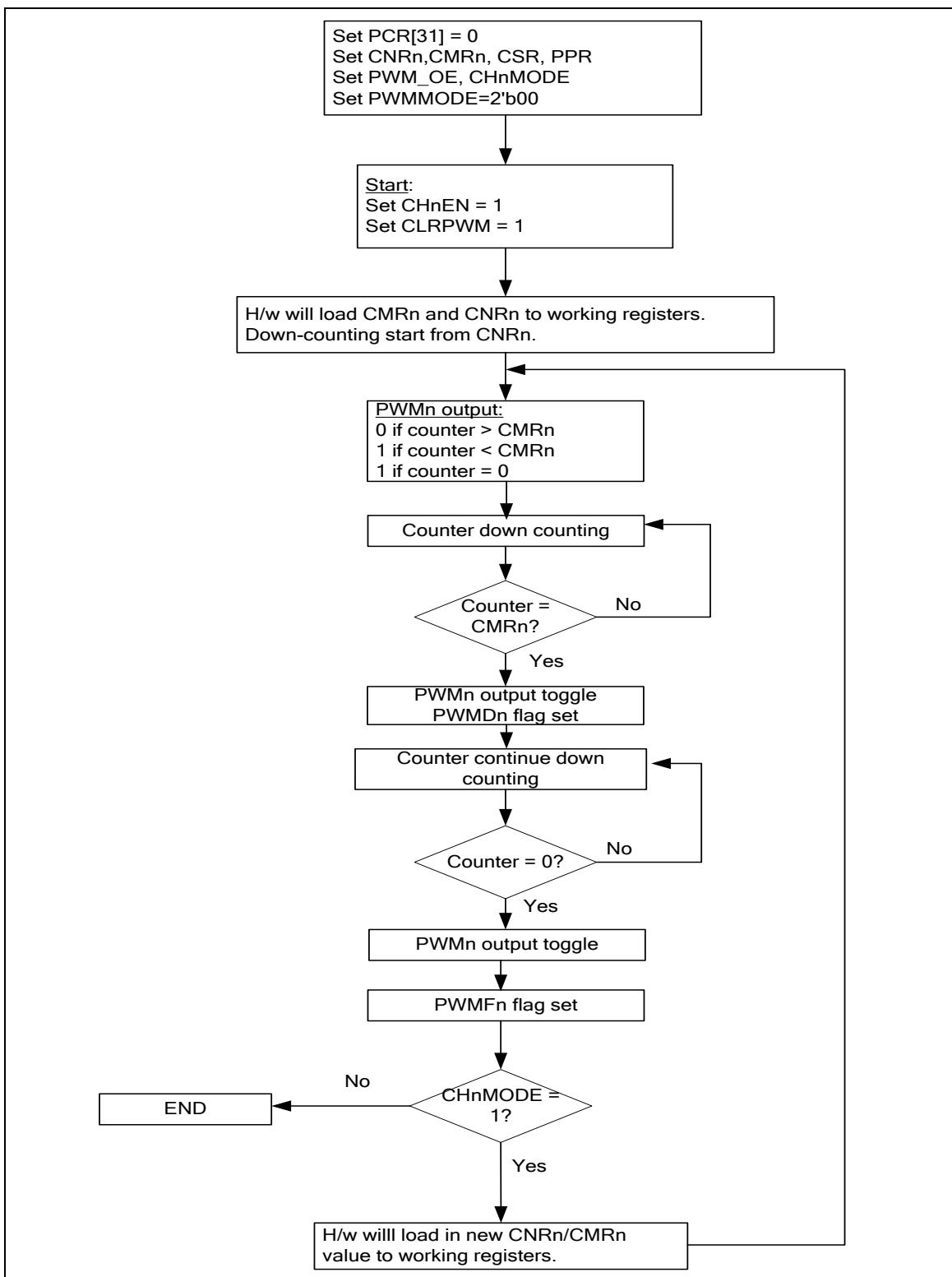


图 5-73 边沿对齐流程图

PWM的周期和占空比由向下计数器寄存器(CNRn) 和比较寄存器(CMRn)决定。下图展示了脉冲宽度。

度调制遵循公式， PWM-Timer 比较器原理和PWM-Timer 时序流程。注意PWM通道相应的GPIO管脚必须配置成PWM功能(使能PWMPOE)。

PWM频率= $HCLK/((\text{prescale}+1)*(\text{clock divider})/(\text{CNR}+1))$; 这里xy, 可以是01, 23 或 45 。这取决于 PWM的通道选择

占空比= $(\text{CMR}+1)/(\text{CNR}+1)$

$\text{CMR} \geq \text{CNR}$: PWM输出总是high

$\text{CMR} < \text{CNR}$: PWM low宽度= $(\text{CNR}-\text{CMR})$ unit[1]; PWM high宽度= $(\text{CMR}+1)$ unit

$\text{CMR} = 0$: PWM low宽度= (CNR) unit; PWM high宽度= 1 unit

注意: 1. Unit =一个PWM时钟周期.

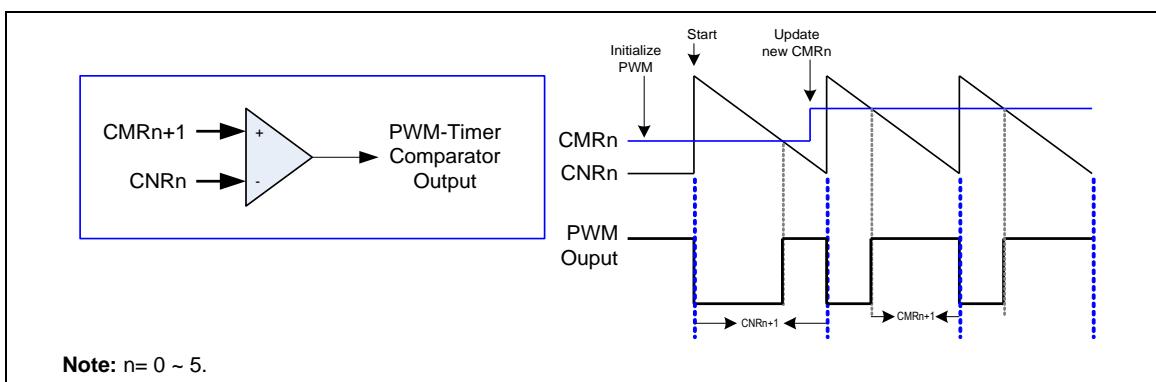


图 5-74 PWM-Timer 内部比较器工作原理

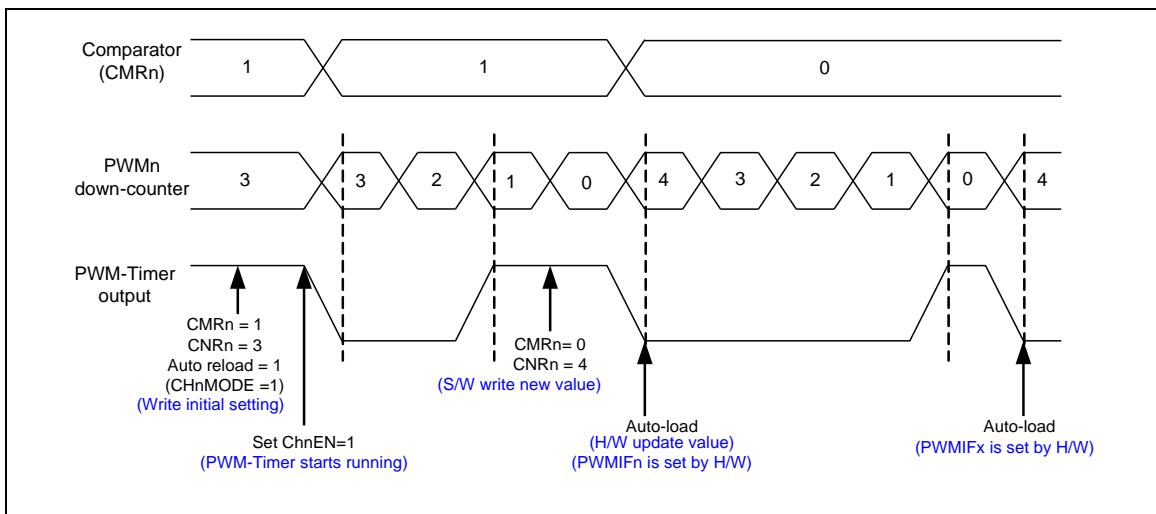


图 5-75 PWM-Timer 工作时序

中心对齐模式（向上/向下计数）

中心对齐模式下，PWM的上/下计数模式时间参数配置好以后将会产生相应的PWM波形。当PWM计数器的值从0计到CMR_n（旧值），将会从0重新开始计数。这将使PWM_n通道发生器就输出高电平。计数器将继续向上计数直到CNR_n（旧值）。然后，计数器会自动继续向下计数，当PWM计数器再次达到CMR_n（旧值），PWM_n产生输出翻转信号到低电平。一旦PWM计数器计数到0，当前CNR_n和CMR_n值在CHnMODE = 1情况下会被更新。

在中心对齐模式中下，如果INT_TYPE (PIER[17]) = 0的话，PWM周期中断可以在向下计数溢出时请求中断，也就是在PWM周期的start (end)位置。或者如果INT_TYPE (PIER[17]) = 1，向上计数器的值达到CNR_n，也就是PWM周期的中心点时请求中断。

PWM频率= HCLK/((prescale+1)*(时钟分频))/(CNR+1); 这里xy, 可以是01, 23或45 , 它取决于所选的PWM通道。

占空比= (CNR - CMR) / (CNR+1)

CMR >= CNR: PWM总是输出low

CMR < CNR: PWM low宽度= (CNR - CMR) * 2 units[1]; PWM high宽度= (CMR+1) * 2 units

CMR = 0: PWM low宽度= CNR * 2 units; PWM high宽度= 2 units

注意: 1. Unit =一个 PWM时钟周期.

下图描述了中心对齐模式下，PWM时序流程。

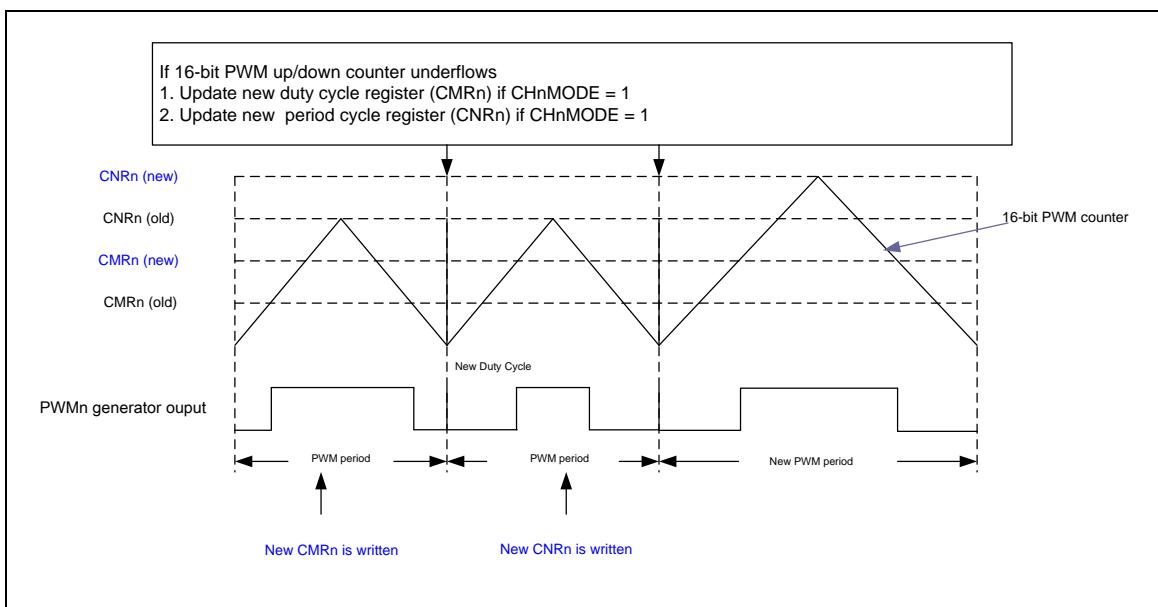


图 5-76 中心对齐模式

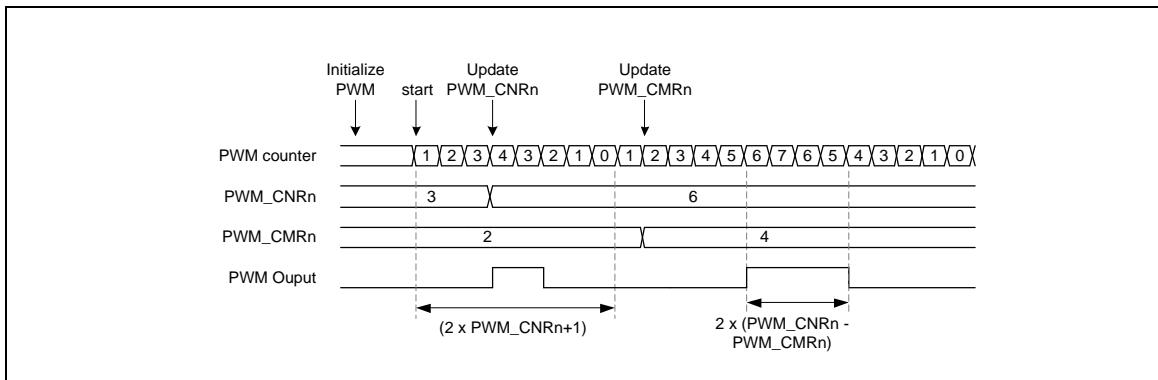


图 5-77 中心对齐模式时序图

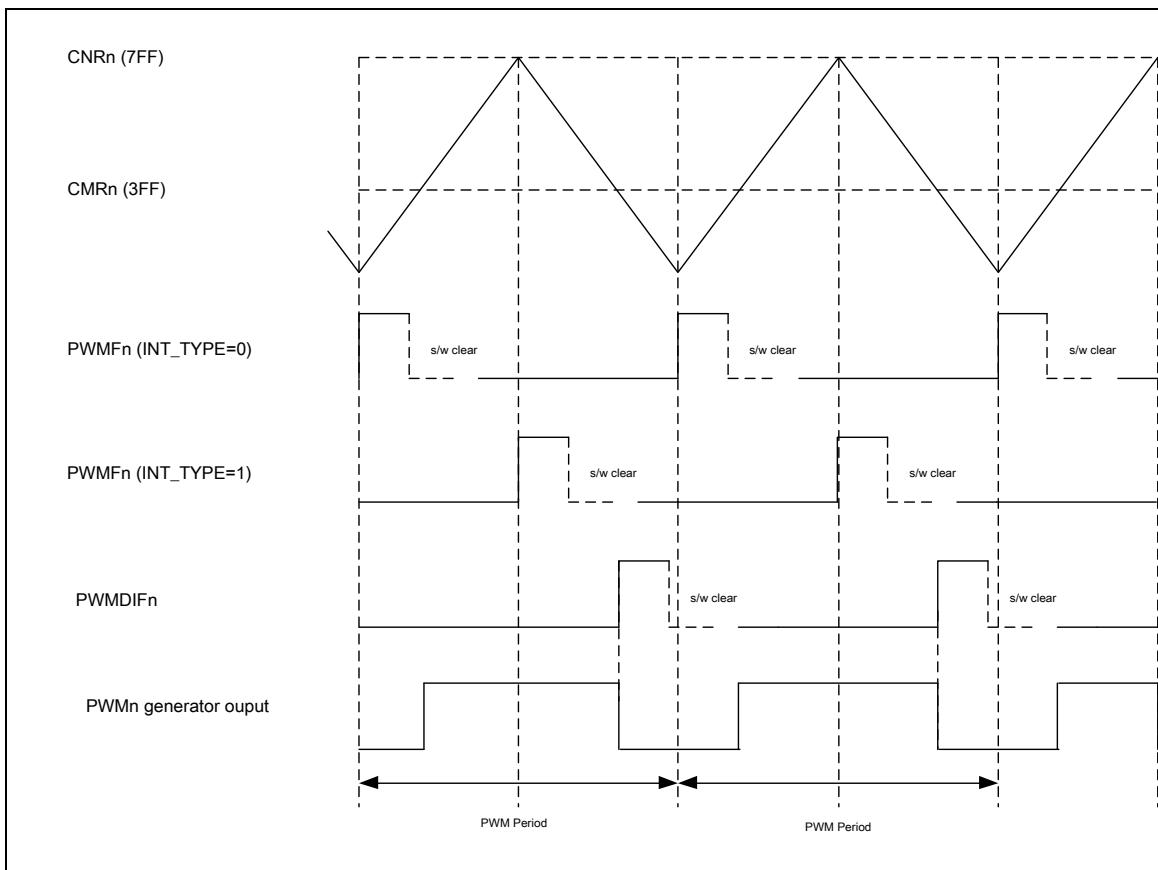


图 5-78 PWM 中心对齐模式波形图

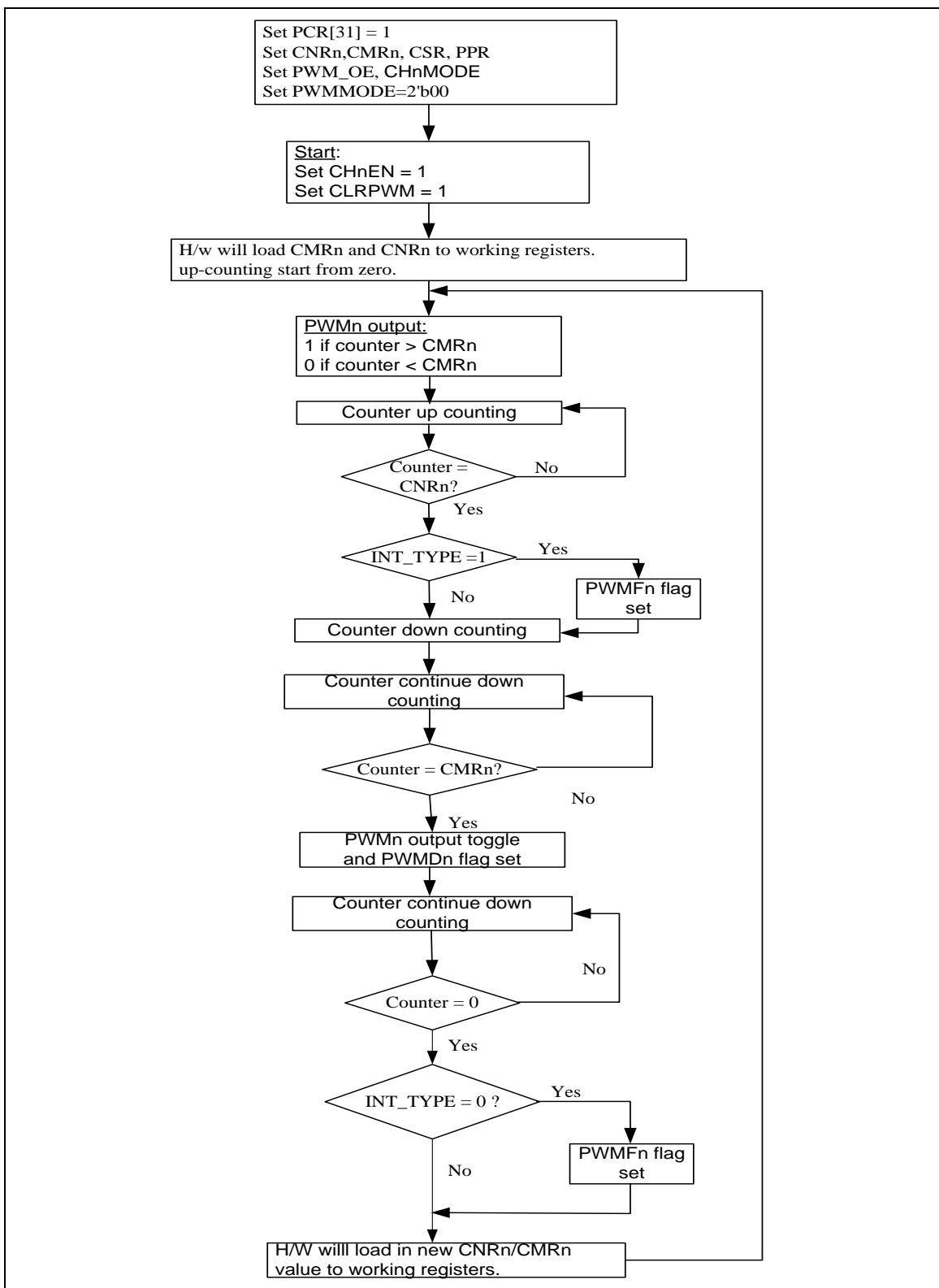


图 5-79 中心对齐模式流程图(INT_TYPE = 0)

5.10.5.2 PWM 双缓冲, 自动装载和单次工作模式

NuMicro® NUC029FAE微控制器，PWM带有双缓冲功能,重新加载配置值，在下一周期开始更新，不影响当前的计数器操作。这样PWM的设定周期值就被写到CNRn.

如果CH0MOD位设置为0，PWM0将配置成单次模式, 如果CH0MOD位设置为1，将配置为自动重载模式。建议PWM0切换模式之前，先将CH0EN位设置为1,使PWM0计数器运行起来。因为PWM0的模式一旦改变，CNR0和CMR0的内容将被清除为0，会重置PWM0周期和占空比设置。如果是PWM0运行在单次模式,应该先写入CMR0和CNR0的值,然后把CH0EN置为1,使PWM0计数器开始运行。当PWM0计数器从CNR0值计数到0时, CNR0和CMR0将被硬件置0。新的CMR0和CNR0值需要由软件写入。重新启动下一个单次模式周期时,应该先写入CMR0值，因为一旦CNR0写一个非零值，PWM0计数器就开始自动重启计数了。

如果PWM0运行在自动重载模式,应该先写入CMR0和CNR0,然后把CH0EN置为1,使PWM0计数器开始运行。CNR0的值当PWM0计数器向下计数达到0后，将重新加载。如果CNR0被置为0, PWM0计数器将挂起。PWM1 ~ PWM5 与PWM0功能类似。

注意:单次模式只支持边沿对齐模式.

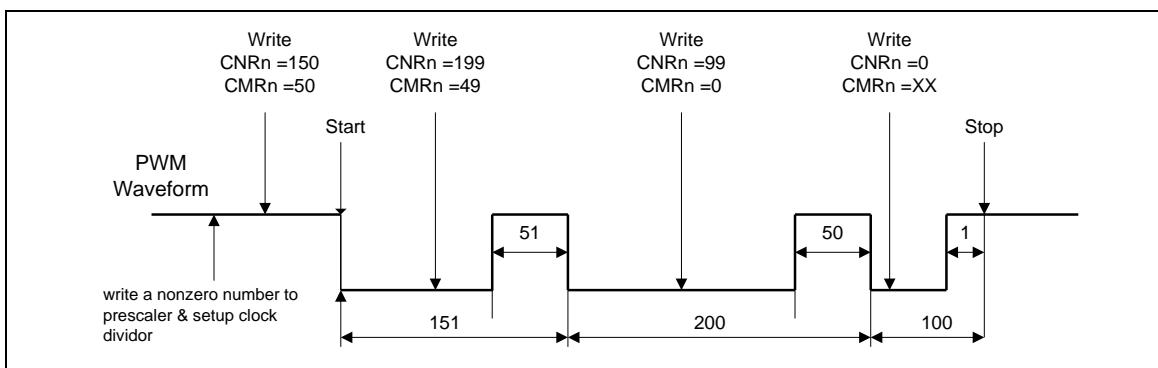


图 5-80 PWM 双缓存图解

5.10.5.3 调节占空比

双缓存功能支持CMRn在当前周期的任何时间写入。当前写入值将在下个周期生效。

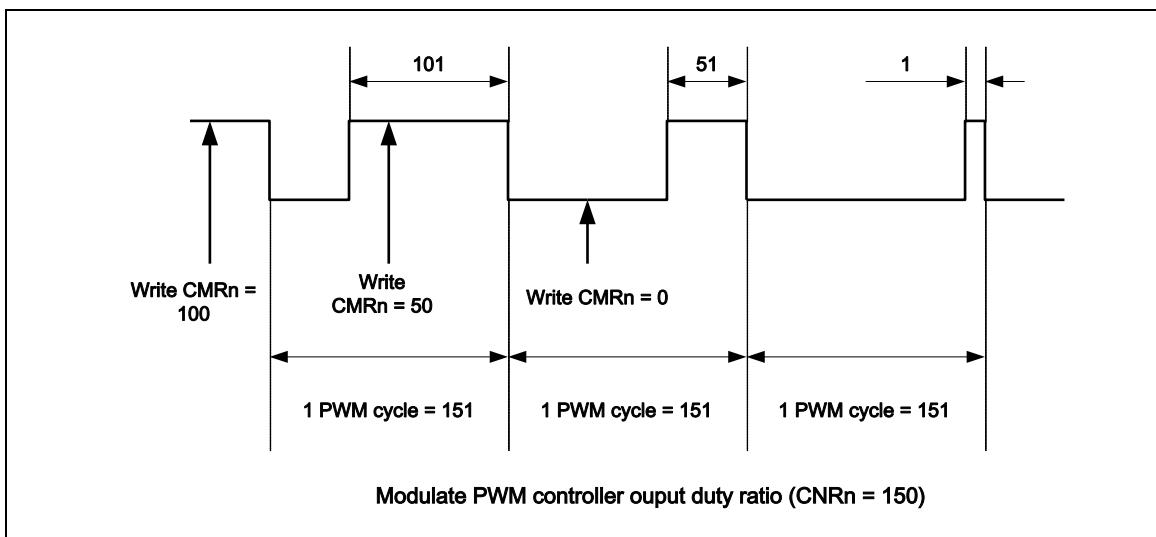


图 5-81 PWM 控制器输出占空比

5.10.5.4 PWM 工作模式

这种强大的PWM单元支持独立模式，可应用于直流或无刷直流电机系统,辅以死区功能可用于交流感应电机和同步电机的应用，同步模式使每组的两个引脚相位同相。此外，组模式可以强制使PWM2和PWM4与PWM0发生器同步，或许可以在直流和无刷直流电机应用中简化更新占空比操作。

5.10.5.5 独立模式

当PWMMOD[1:0] = 00时，独立模式使能。默认状况下，PWM工作在独立模式，6个PWM通道输出各自的周期和占空比。

5.10.5.6 互补模式

当PWMMOD[1:0] = 01时启用互补模式。

在这个模块下有三个专为互补模式提供的占空比-周期发生器，共有三对PWM输出。6个输出通道根据奇偶分组。互补模式下，奇偶PWM通道PG_n与相应的偶数PWM通道互补。如PG1与PG0互补，PG3与PG2互补，PG5与PG4互补。PWM的周期参数由其自身的16位计数器提供，它也可作为可选的预分频设置值。

5.10.5.7 死区插入时间

死区发生器在一组互补的PWM中的关闭管脚和打开管脚波形之间插入的偏移时间叫做“死区时间”。这样能对外接的电源切换设备起到保护作用。一对PWM输出通道有一个8位向下计数器产生死去时间。死区输出维持到计数器向下计数到0。

死区时间按照以下函数计算：

死区时间 = PWM_CLK * (DZIxy[7:0]+1).这里xy, 可以是01, 23, 45

下图示在一对通道中插入死区时间的时序图。

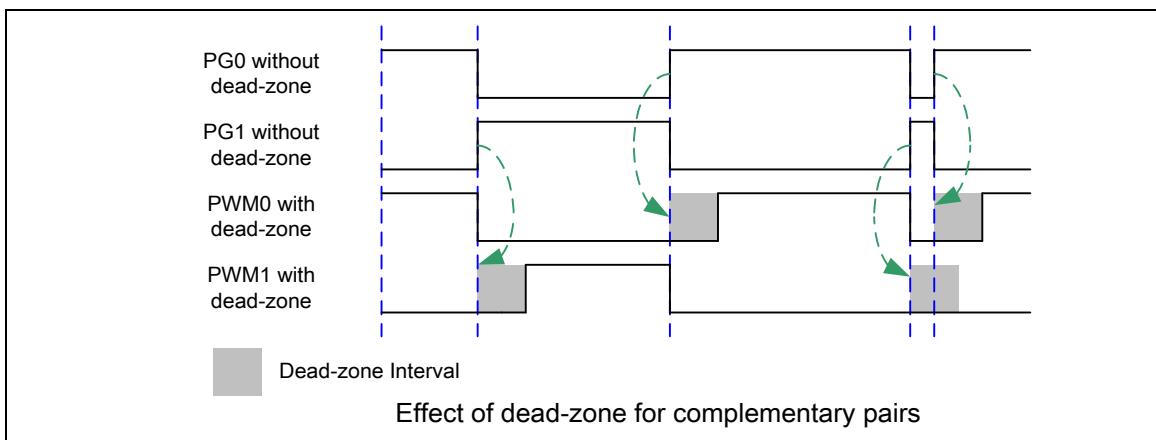


图 5-82 死区时间示意图

在电源切换的应用中，死区时间插入避免上下半桥开关同时打开的情况。所以插入死区时间控制对系统正常工作是至关重要的。通道打开和关闭不能实现立即切换，所以在切换过程中需要留有一些时间，死区功能就恰好实现了这个延时。

5.10.5.8 同步模式

当 PWMMOD[1:0] = 10 时，同步模式打开。

同步模式中，PWM 组中两个输出信号完全相同。

PG1=PG0, PG3=PG2 and PG5=PG4.

5.10.5.9 组模式

当 GRP (PCR[30]) = 1 时，使能组模式

PWM 外设支持组模式控制功能，此模式中所有 PWM 通道输出占空比都由 PWM0 占空比寄存器控制。

如果 GRP = 1，那么 (PG2, PG3) 和 (PG4, PG5) 对会跟随 (PG0, PG1)，意味着：

PG4 = PG2 = PG0;

当 PWMMOD[1:0] = 01，如果互补模式被使能的话，PG5 = PG3 = PG1 = (PG0 的反向波形)。

注意：对应用来说，不要同时用组模式和同步模式，这样同步模式实现不了。

5.10.5.10 极性控制

PWM0 ~ PWM5 每个 PWM 通道都有独立的极性控制，默认极性是输出高电平。

下图展示了不同极性的 PWM 初始化状态

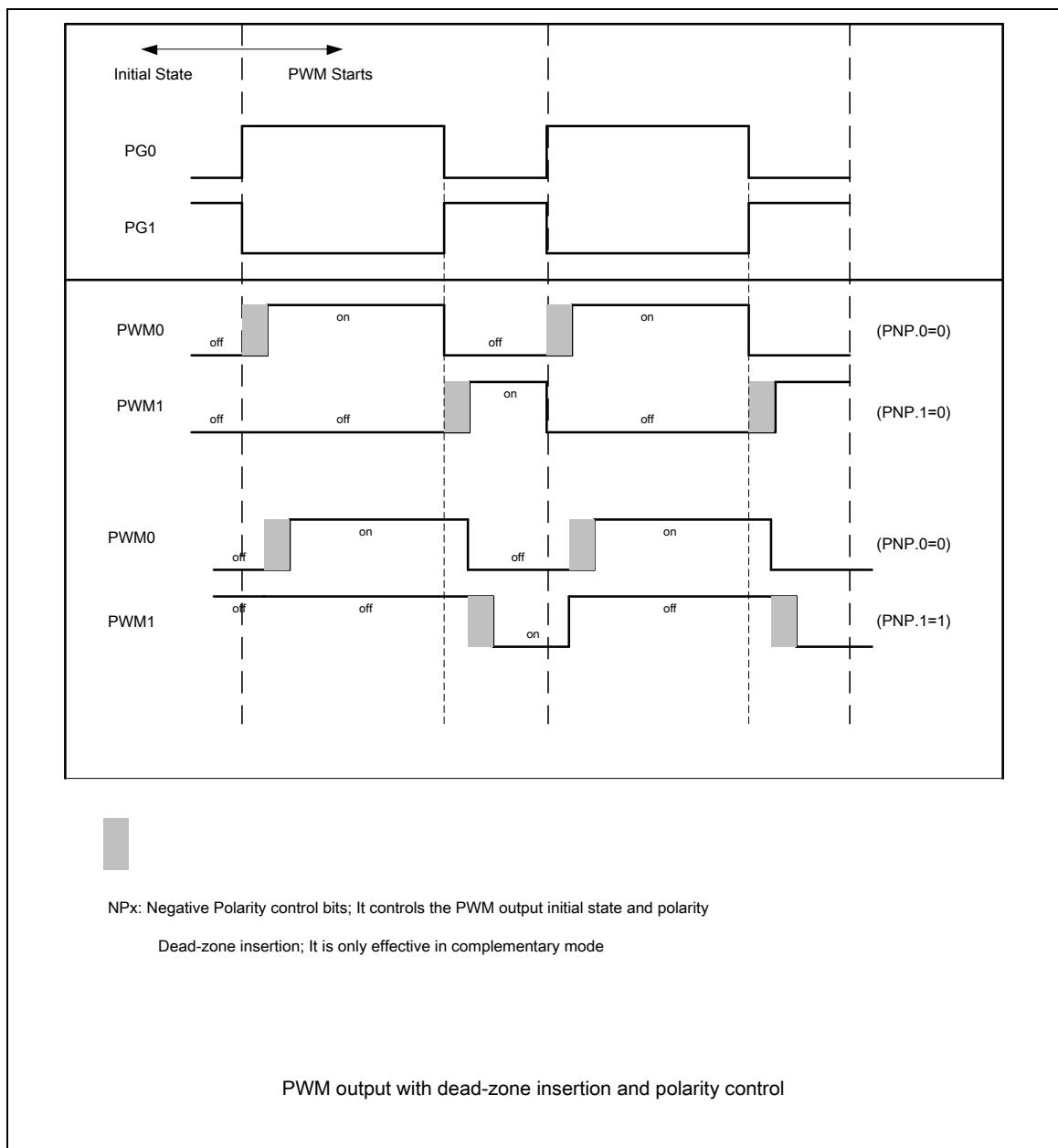


图 5-83 上升沿死区 PWM 极性控制初始化状态

5.10.6 PWM 电机控制中断架构

PWM 单元有四个中断源，分别是 PWM 周期中断标志 (PWMPIF)，PWM 占空比中断标志 (PWMDIF)，Brake0 中断标志 (BKF0) 和 Brake1 中断标志 (BKF1)。BRKIE (PIER[16]) 控制刹车中断使能；PWMPIEn (PIER[0] ~ PIER[5]) 控制 PWM 周期中断使能；PWMDIEn (PIER[8] ~ PIER[13]) 控制 PWM 占空比中断使能。请注意，所有的中断标志都由硬件置位，且必须通过软件使能。

下图展示了电机控制中 PWM 中断的架构

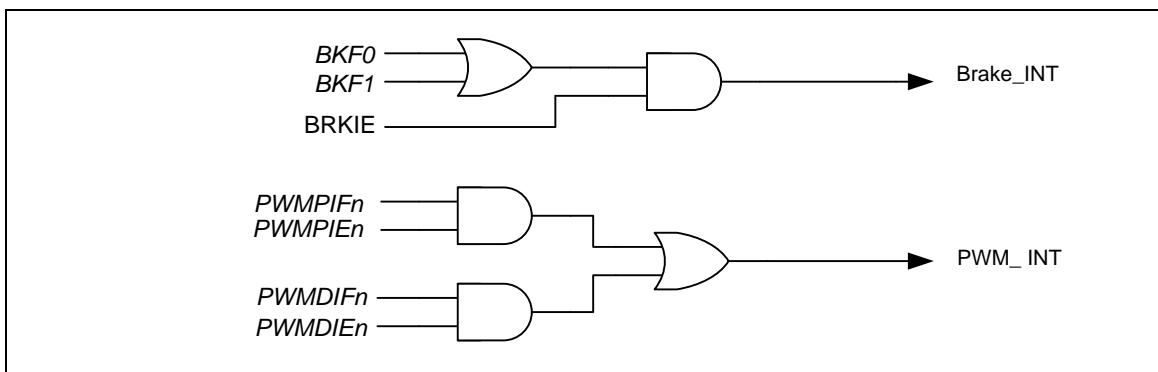


图 5-84 电机控制 PWM 中断架构

5.10.6.1 PWM 刹车功能

PWM 支持外部两个刹车管脚 BKF0 和 BKF1 管脚；故障刹车功能由 PFBCON 寄存器控制。

因为刹车状况重置也会自动产生 BKF 标志，用户可以查询该标志或使能相关刹车中断位来判断什么情况下产生了刹车中断。

注意：当刹车事件发生时，PWM0 ~ PWM5 使能位会被硬件清除。用户要先清掉刹车事件标志位 BKF，重新使能相关通道。

5.10.6.2 PWM 换相功能

PWM 可以通过配置 PHCHG 和 PHCHG_NXT 寄存器可以实现换相功能，当计数器溢出事件发生时，PHCHG 的值将会被 PHCHG_NXT 的值自动更新。PHCHG 寄存器位的值将与 PHCHG_NXT 保持一样。每次 PHCHG 的内容更新后，相关功能将会改变。

5.10.7 PWM 计数器配置过程

以下是 PWM 计数器配置过程

1. 配置预分频寄存器(PPR)以设置预分频(CPx_x)。
2. 配置时钟选择寄存器(CSR)选择时钟源(CSR_x)。
3. 配置 PWM 控制寄存器(PCR)来使能自动加载模式(CHxMOD = 1)，PWM 计数器对齐类型选择(PWMTYPE)和关闭 PWM 计数器(CHxEN = 0)。
4. 配置 PWM 控制寄存器(PCR)选择翻转是否打开(CHxINV)，死区时间发生器打开/关闭(DZENn) (可选)
5. 配置 PDZIR 寄存器来选择死区时间 (可选)
6. 配置比较寄存器(CMR_x) 来设置占空比(CMR_x)。
7. 配置 PWM 计数器寄存器(CNR_x)，用以设置计数器加载值(CNR_x)。
8. 配置 PWM 中断使能寄存器(PIER)来设置 PWM 周期中断类型(INTTYPE)，PWM 周期中断使能位(PWMPIEx)，和 PWM 占空比中断使能位(PWMDIEx) (可选)
9. 配置 PWM 输出使能寄存器(POE)来使能 PWM 输出通道(PWM_x)

10. 配置PWM控制寄存器(PCR)使能PWM计数器(CHxEN = 1)

5.10.8 PWM计数器停止过程

模式1:

设置16bit计数器寄存器(CNRx)到0。当中断请求发生，关闭PWM计数器(PCR的CHxEN).(推荐做法)

模式2:

直接关闭PWM计数器(PCR的CHxEN) (不推荐)

不推荐的原因是关闭CHxEN会使PWM输出信号立即停止，输出的波形将是不完整的，所以PWM的输出波形的占空比也不正确的，这在电机控制中将会带来灾难性后果。

5.10.9 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
PWM基址:				
PWM_BA = 0x4004_0000				
PPR	PWM_BA+0x00	R/W	PWM 时钟预分频寄存器	0x0000_0000
CSR	PWM_BA+0x04	R/W	PWM时钟选择寄存器	0x0000_0000
PCR	PWM_BA+0x08	R/W	PWM控制寄存器	0x0000_0000
CNR0	PWM_BA+0x0C	R/W	PWM计数器周期寄存器0	0x0000_0000
CNR1	PWM_BA+0x10	R/W	PWM 计数器周期寄存器1	0x0000_0000
CNR2	PWM_BA+0x14	R/W	PWM 计数器周期寄存器2	0x0000_0000
CNR3	PWM_BA+0x18	R/W	PWM 计数器周期寄存器3	0x0000_0000
CNR4	PWM_BA+0x1C	R/W	PWM 计数器周期寄存器4	0x0000_0000
CNR5	PWM_BA+0x20	R/W	PWM计数器周期寄存器5	0x0000_0000
CMR0	PWM_BA+0x24	R/W	PWM 比较寄存器0	0x0000_0000
CMR1	PWM_BA+0x28	R/W	PWM比较寄存器1	0x0000_0000
CMR2	PWM_BA+0x2C	R/W	PWM 比较寄存器2	0x0000_0000
CMR3	PWM_BA+0x30	R/W	PWM 比较寄存器 3	0x0000_0000
CMR4	PWM_BA+0x34	R/W	PWM 比较寄存器4	0x0000_0000
CMR5	PWM_BA+0x38	R/W	PWM 比较寄存器5	0x0000_0000
PIER	PWM_BA+0x54	R/W	PWM中断使能控制寄存器	0x0000_0000
PIIR	PWM_BA+0x58	R/W	PWM中断指示寄存器	0x0000_0000
PWMPOE	PWM_BA+0x5C	R/W	PWM通道 0~5输出使能寄存器	0x0000_0000
PFBCON	PWM_BA+0x60	R/W	PWM 故障刹车控制寄存器	0x0000_0000
PDZIR	PWM_BA+0x64	R/W	PWM 死区时间间隔寄存器	0x0000_0000
TRGCON0	PWM_BA+0x68	R/W	PWM触发控制寄存器0	0x0000_0000
TRGCON1	PWM_BA+0x6C	R/W	PWM触发控制寄存器1	0x0000_0000
TRGSTS0	PWM_BA+0x70	R/W	PWM触发状态寄存器0	0x0000_0000
TRGSTS1	PWM_BA+0x74	R/W	PWM触发状态寄存器1	0x0000_0000
PHCHG	PWM_BA+0x78	R/W	Phase相位修改寄存器	0x0000_3F00

PHCHGNXT	PWM_BA+0x7C	R/W	下一相位修改寄存器	0x0000_3F00
PHCHGMASK	PWM_BA+0x80	R/W	相位修改屏蔽寄存器	0x0000_0000
INTACCUCTL	PWM_BA+0x84	R/W	周期中断累加控制寄存器	0x0000_00F0

5.10.10 寄存器描述

PWM时钟预分频寄存器(PPR)

寄存器	偏移地址	R/W	描述	复位值
PPR	PWM_BA+0x00	R/W	PWM时钟预分频寄存器	0x0000_0000

31	30	29	28	27	26	25	24
恒定							
23	22	21	20	19	18	17	16
CP45							
15	14	13	12	11	10	9	8
CP23							
7	6	5	4	3	2	1	0
CP01							

位	描述	
[31:24]	恒定	保留.
[23:16]	CP45[7:0]	PWM 计数器 4和5的时钟预分频4 输入时钟源输入到相应的PWM计数器前被(CP45 + 1)分频。 如果CP45 = 0,时钟预分频 4 输出时钟将被停止。所以相应的PWM计数器将被停止。
[15:8]	CP23[7:0]	PWM 计数器 2和3的时钟预分频2 输入时钟源输入到相应的PWM计数器前被(CP23 + 1)分频。 如果CP23 = 0,时钟预分频 2 输出时钟将被停止。所以相应的PWM计数器将被停止。
[7:0]	CP01[7:0]	PWM 计数器 0和1的时钟预分频0 输入时钟源输入到相应的PWM计数器前被(CP01 + 1)分频。 如果CP01 = 0,时钟预分频 0 输出时钟将被停止。所以相应的PWM计数器将被停止。

PWM时钟选择寄存器(CSR)

寄存器	偏移地址	R/W	描述	复位值
CSR	PWM_BA+0x04	R/W	PWM 时钟选择寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱	CSR5			悶隱	CSR4		
15	14	13	12	11	10	9	8
悶隱	CSR3			悶隱	CSR2		
7	6	5	4	3	2	1	0
悶隱	CSR1			悶隱	CSR0		

位	描述	
[31:23]	悶隱	保留.
[22:20]	CSR5[2:0]	<p>计数器 5时钟分频选择 选择PWM 计数器的时钟源. 000 =输入时钟频率除以 2. 001 =输入时钟频率除以4. 010 =输入时钟频率除以8. 011 =输入时钟频率除以16. 100 =输入时钟频率除以1.</p>
[19]	悶隱	保留.
[18:16]	CSR4[2:0]	<p>计数器 4时钟分频选择 选择PWM 计数器的时钟源. (表内容与CSR5相同)</p>
[15]	悶隱	保留.
[14:12]	CSR3[2:0]	<p>计数器 3时钟分频选择 选择PWM 计数器的时钟源. (表内容与CSR5相同)</p>
[11]	悶隱	保留.
[10:8]	CSR2[2:0]	<p>计数器 2时钟分频选择 选择PWM 计数器的时钟源. (表内容与CSR5相同)</p>
[7]	悶隱	保留.

位	描述	
[6:4]	CSR1[2:0]	计数器 1时钟分频选择 选择PWM 计数器的时钟源.. (表内容与CSR5相同)
[3]	保留	
[2:0]	CSR0[2:0]	计数器 0时钟分频选择 选择PWM 计数器的时钟源. (表内容与CSR5相同)

PWM控制寄存器(PCR)

寄存器	偏移地址	R/W	描述	复位值
PCR	PWM_BA+0x08	R/W	PWM控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PWMTYPE	GRP	PWMMOD		CLRPWM	DZEN45	DZEN23	DZEN01
23	22	21	20	19	18	17	16
CH5MOD	CH5INV	帳隱	CH5EN	CH4MOD	CH4INV	帳隱	CH4EN
15	14	13	12	11	10	9	8
CH3MOD	CH3INV	帳隱	CH3EN	CH2MOD	CH2INV	帳隱	CH2EN
7	6	5	4	3	2	1	0
CH1MOD	CH1INV	帳隱	CH1EN	CH0MOD	CH0INV	DB_MODE	CH0EN

位	描述	
[31]	PWMTYPE	PWM计数器对齐方式选择位 0 =边沿对齐方式 1 =中心对齐方式
[30]	GRP	成组功能使能位 0 = 所有 PWM 通道信号时序相互对立 1 = 统一由PWM0控制的PWM0, PWM2 和 PWM4通道信号时序。统一由PWM1控制的PWM1, PWM3 和 PWM5通道信号时序。
[29:28]	PWMMOD[1:0]	PWM 工作模式选择位 00 = 独立模式. 01 = 互补模式. 10 = 同步模式. 11 = 保留.
[27]	CLRPWM	清除PWM计数器控制位 0 = 不清PWM计数器 1 = 所有 16-bit PWM 计数器清 0x0000. 注意: 该位有硬件自动清除
[26]	DZEN45	死区4 计数器使能位 (PWM4 和 PWM5为 PWM 组的一对) 0 = 死区4 发生器禁止. 1 = 死区4 发生器使能. 注意: 当死区发生器被使能, PWM4 和 PWM5对变成PWM组的互补对.

位	描述
[25]	DZEN23 死区2计数器使能位 (PWM2 和 PWM3为PWM组的一对) 0 = 死区2发生器禁止. 1 = 死区2发生器使能. 注意: 当死区发生器被使能, PWM2 和 PWM3对变成PWM组的互补对.
[24]	DZEN01 死区0计数器使能位 (PWM0 和 PWM1为PWM组的一对) 0 = 死区0发生器禁止. 1 = 死区0发生器使能. 注意: 当死区发生器被使能, PWM0 和 PWM1对变成PWM组的互补对.
[23]	CH5MOD PWM计数器5自动装载/One-Shot模式 0 = One-shot模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR5和CMR5被清零.
[22]	CH5INV PWM5输出反向使能位 0 = PWM5输出反向禁止 1 = PWM5输出反向使能
[21]	保留
[20]	CH5EN PWM计数器5使能开始运行 0 = 相应的PWM计数器停止. 1 = 相应的PWM计数器开始运行.
[19]	CH4MOD PWM计数器4自动装载/One-Shot模式 0 = One-shot模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR4和CMR4被清零.
[18]	CH4INV PWM4输出反向使能位 0 = PWM4输出反向禁止 1 = PWM4输出反向使能
[17]	保留
[16]	CH4EN PWM计数器4使能开始运行 0 = 相应的PWM计数器停止. 1 = 相应的PWM计数器开始运行.
[15]	CH3MOD PWM计数器3自动装载/One-Shot模式 0 = One-shot模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR3和CMR3被清零.
[14]	CH3INV PWM3输出反向使能位 0 = PWM4输出反向禁止 1 = PWM4输出反向使能
[13]	保留

位	描述
[12]	CH3EN PWM 计数器3使能开始运行 0 = 相应的 PWM计数器停止. 1 = 相应的 PWM计数器开始运行.
[11]	CH2MOD PWM 计数器2自动装载/One-Shot 模式 0 = One-shot 模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR2和CMR2被清零.
[10]	CH2INV PWM2输出反向使能位 0 = PWM4输出反向禁止 1 = PWM4输出反向使能
[9]	保留
[8]	CH2EN PWM 计数器2使能开始运行 0 = 相应的 PWM计数器停止. 1 = 相应的 PWM计数器开始运行.
[7]	CH1MOD PWM 计数器1自动装载/One-Shot 模式 0 = One-shot 模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR1和CMR1被清零.
[6]	CH1INV PWM1输出反向使能位 0 = PWM1输出反向禁止 1 = PWM1输出反向使能
[5]	保留
[4]	CH1EN PWM 计数器1使能开始运行 0 = 相应的 PWM计数器停止. 1 = 相应的 PWM计数器开始运行.
[3]	CH0MOD PWM 计数器0自动装载/One-Shot 模式 0 = One-shot 模式. 1 = 自动装载模式 注意:如果该位由0变1, 将使CNR0和CMR0被清零.
[2]	CH0INV PWM0输出反向使能位 0 = PWM0输出反向禁止 1 = PWM0输出反向使能
[1]	DB_MODE PWM 在Debug 模式配置位(仅在DEBUG模式下) 0 = 安全模式: 计数器被停止, PWM输出被关闭, 反相器处于安全状态。计数器可以从上次计数停止处继续计数。 1 = 正常模式:在某些特殊情况下, 计数器继续正常运行可能非常危险, 因为有一个固定占比的波形输出到反向器 (没有了中断服务程序)

位	描述	
[0]	CH0EN	PWM 计数器 0 开始运行 0 = 相应的 PWM 计数器停止运行. 1 = 相应的 PWM 计数器开始运行

PWM计数器寄存器0-5 (CNR0-5)

寄存器	偏移地址	R/W	描述	复位值
CNR0	PWM_BA+0x0C	R/W	PWM计数器寄存器0	0x0000_0000
CNR1	PWM_BA+0x10	R/W	PWM计数器寄存器1	0x0000_0000
CNR2	PWM_BA+0x14	R/W	PWM计数器寄存器2	0x0000_0000
CNR3	PWM_BA+0x18	R/W	PWM计数器寄存器3	0x0000_0000
CNR4	PWM_BA+0x1C	R/W	PWM计数器寄存器4	0x0000_0000
CNR5	PWM_BA+0x20	R/W	PWM计数器寄存器5	0x0000_0000

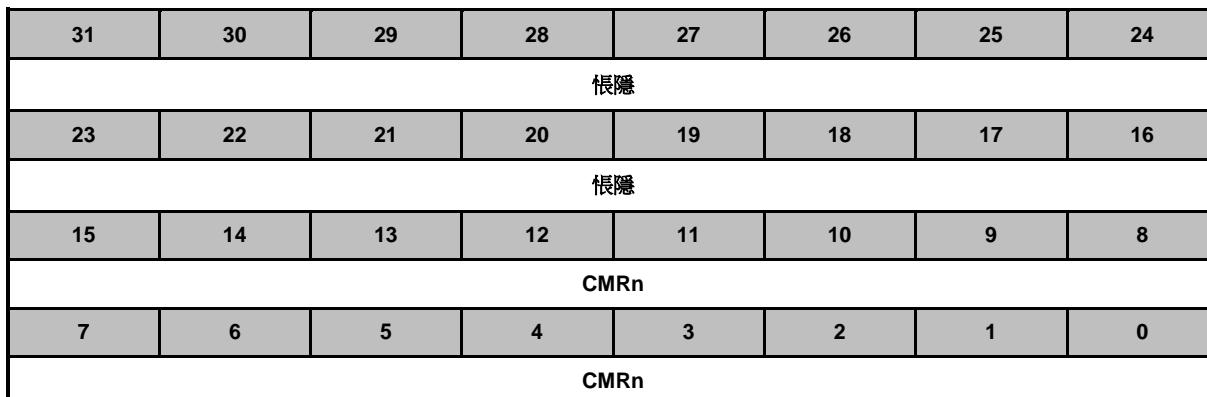
31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
CNRn							
7	6	5	4	3	2	1	0
CNRn							

位	描述	
[31:16]	帳隱	保留.
[15:0]	CNRn	<p>PWM计数器周期装载值</p> <p>CNRn表示 PWM 计数器周期值. n = 0, 1..5.</p> <p>边沿对齐方式:</p> <p>PWM 频率 = HCLK/((prescale+1)*(clock divider))/(CNRn+1); 这里 xy, 根据所选的PWM通道可以是01, 23, 45</p> <p>占空比= (CMRn+1)/(CNRn+1).</p> <p>CMRn >= CNRn: PWM 输出总是high.</p> <p>CMRn < CNRn: PWM low宽度= (CNRn-CMRn) unit; PWM high宽度= (CMRn+1) unit.</p> <p>CMRn = 0: PWM low宽度= (CNRn) unit; PWM high宽度= 1 unit.</p> <p>中心对齐方式:</p> <p>PWM频率= HCLK/((prescale+1)*(clock divider))/ (2*CNRn+1); 这里xy, 根据所选的PWM通道可以是01, 23, 45</p> <p>占空比= (CNRn - CMRn)/(CNRn+1).</p> <p>CMRn >= CNRn: PWM输出总是low.</p> <p>CMRn < CNRn: PWM low宽度= (CMRn + 1) * 2 unit; PWM high宽度= (CNRn - CMRn) * 2 unit.</p> <p>CMRn = 0: PWM low宽度= 2 unit; PWM high宽度= (CNRn) * 2 unit.</p>

位	描述
	(Unit = 一个 PWM时钟周期). 注意: 任何对CNRn写操作将会影响到下一个PWM周期.

PWM比较器寄存器0-5 (CMR0-5)

寄存器	偏移地址	R/W	描述	复位值
CMR0	PWM_BA+0x24	R/W	PWM比较器寄存器0	0x0000_0000
CMR1	PWM_BA+0x28	R/W	PWM比较器寄存器1	0x0000_0000
CMR2	PWM_BA+0x2C	R/W	PWM比较器寄存器2	0x0000_0000
CMR3	PWM_BA+0x30	R/W	PWM比较器寄存器3	0x0000_0000
CMR4	PWM_BA+0x34	R/W	PWM比较器寄存器4	0x0000_0000
CMR5	PWM_BA+0x38	R/W	PWM比较器寄存器5	0x0000_0000



位	描述	
[31:16]	保留	保留.
[15:0]	CMRn	<p>PWM比较器位</p> <p>CMR决定了PWM的占空比, $n = 0, 1..5$.</p> <p>边沿对齐模式:</p> <p>$\text{PWM频率} = \text{HCLK}/((\text{prescale}+1)*(\text{clock divider})) / (\text{CNRn}+1)$; 这里xy, 根据所选的PWM通道可以是01, 23, 45.</p> <p>占空比= $(\text{CMRn}+1)/(\text{CNRn}+1)$.</p> <p>$\text{CMRn} \geq \text{CNRn}$: PWM 输出总是high.</p> <p>$\text{CMRn} < \text{CNRn}$: PWM low宽度= $(\text{CNRn}-\text{CMRn})$ unit; PWM high宽度= $(\text{CMRn}+1)$ unit.</p> <p>$\text{CMRn} = 0$: PWM low宽度= (CNRn) unit; PWM high宽度= 1 unit.</p> <p>中心对齐方式:</p> <p>$\text{PWM频率} = \text{HCLK}/((\text{prescale}+1)*(\text{clock divider})) / (2*\text{CNRn}+1)$; 这里xy, 根据所选的PWM通道可以是01, 23, 45.</p> <p>占空比= $(\text{CNRn} - \text{CMRn})/(\text{CNRn}+1)$.</p> <p>$\text{CMRn} \geq \text{CNRn}$: PWM 输出总是low.</p> <p>$\text{CMRn} < \text{CNRn}$: PWM low宽度= $(\text{CMRn} + 1) * 2$ unit; PWM high宽度= $(\text{CNRn} - \text{CMRn}) * 2$ unit.</p>

位	描述
	<p>CMRn = 0: PWM low宽度= 2 unit; PWM high宽度= (CNRn) * 2 unit.</p> <p>(Unit =一个 PWM时钟周期).</p> <p>注意: 任何对CMRn写操作将会影响到下一个PWM周期</p>

PWM中斷使能寄存器(PIER)

寄存器	偏移地址	R/W	描述	复位值
PIER	PWM_BA+0x54	R/W	PWM 中斷使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱						INT_TYPE	BRKIE
15	14	13	12	11	10	9	8
悶隱		PWMDIE5	PWMDIE4	PWMDIE3	PWMDIE2	PWMDIE1	PWMDIE0
7	6	5	4	3	2	1	0
悶隱		PWMPIE5	PWMPIE4	PWMPIE3	PWMPIE2	PWMPIE1	PWMPIE0

位	描述	
[31:18]	悶隱	保留.
[17]	INT_TYPE	PWM 中断类型选择 0 = 如果 PWM 计数器下溢, PWMPIfn将被置1. 1 = 如果 PWM 计数器的值与CNRn寄存器匹配, PWMPIfn将被置1. 注意: 当PWM为中心对齐方式时, 该位有效.
[16]	BRKIE	Fault Brake0 和 Fault Brake1中断使能位 0 = BKFO和BKF1触发PWM中断禁止. 1 = BKFO和BKF1触发PWM中断使能.
[15:14]	悶隱	保留.
[13]	PWMDIE5	PWM 通道 5 占空比中断使能控制位 0 = 禁止. 1 = 使能.
[12]	PWMDIE4	PWM 通道 4 占空比中断使能控制位 0 = 禁止. 1 = 使能.
[11]	PWMDIE3	PWM 通道 3 占空比中断使能控制位 0 = 禁止. 1 = 使能.
[10]	PWMDIE2	PWM 通道 2 占空比中断使能控制位 0 = 禁止. 1 = 使能.

位	描述	
[9]	PWMDIE1	PWM 通道 1 占空比中断使能控制位 0 = 禁止. 1 = 使能.
[8]	PWMDIE0	PWM 通道 0占空比中断使能控制位 0 = 禁止. 1 = 使能.
[7:6]	保留	保留.
[5]	PWMPIE5	PWM通道5周期中断使能位 0 =中断禁止. 1 =中断使能.
[4]	PWMPIE4	PWM通道4周期中断使能位 0 =中断禁止. 1 =中断使能.
[3]	PWMPIE3	PWM通道3周期中断使能位 0 =中断禁止. 1 =中断使能.
[2]	PWMPIE2	PWM通道2周期中断使能位 0 =中断禁止. 1 =中断使能.
[1]	PWMPIE1	PWM通道1周期中断使能位 0 =中断禁止. 1 =中断使能.
[0]	PWMPIE0	PWM通道0周期中断使能位 0 =中断禁止. 1 =中断使能.

PWM 中断指示寄存器 (PIIR)

寄存器	偏移地址	R/W	描述	复位值
PIIR	PWM_BA+0x58	R/W	PWM 中断指示寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱						BKF1	BKF0
15	14	13	12	11	10	9	8
悵隱		PWMDIF5	PWMDIF4	PWMDIF3	PWMDIF2	PWMDIF1	PWMDIF0
7	6	5	4	3	2	1	0
悵隱		PWMPIF5	PWMPIF4	PWMPIF3	PWMPIF2	PWMPIF1	PWMPIF0

位	描述	
[31:18]	悵隱	保留.
[17]	BKF1	<p>PWM Brake1 标志 0 = BKP1管脚产生一个下降沿信号时, PWM 刹车不会产生刹车标志. 1 = 当 PWM 刹车在 BKP1管脚检测到一个下降沿信号, 该标志将会被置1. 注意: 该位可以被软件写1清0.</p>
[16]	BKF0	<p>PWM Brake0 标志 0 = BKPO管脚产生一个下降沿信号时, PWM 刹车不会产生刹车标志. 1 = 当 PWM 刹车在BKPO管脚检测到一个下降沿信号, 该标志将会被置1. 注意: 该位可以被软件写1清0.</p>
[15:14]	悵隱	保留.
[13]	PWMDIF5	<p>PWM 通道 5占空比中断标志 当通道5的PWM计数器向下计数到CMR5的值时, 该标志被硬件置1. 注意: 该位可以被软件写1清0.</p>
[12]	PWMDIF4	<p>PWM 通道 4占空比中断标志 当通道5的PWM计数器向下计数到CMR4的值时, 该标志被硬件置1. 注意: 该位可以被软件写1清0.</p>
[11]	PWMDIF3	<p>PWM 通道 3占空比中断标志 当通道5的PWM计数器向下计数到CMR3的值时, 该标志被硬件置1. 注意: 该位可以被软件写1清0.</p>
[10]	PWMDIF2	<p>PWM 通道 2占空比中断标志 当通道5的PWM计数器向下计数到CMR2的值时, 该标志被硬件置1. 注意: 该位可以被软件写1清0.</p>

位	描述
[9]	PWMIF1 PWM 通道 1占空比中断标志 当通道5的PWM计数器向下计数到CMR1的值时，该标志被硬件置1. 注意: 该位可以被软件写1清0.
[8]	PWMIF0 PWM 通道0占空比中断标志 当通道5的PWM计数器向下计数到CMR0的值时，该标志被硬件置1. 注意: 该位可以被软件写1清0.
[7:6]	保留
[5]	PWMIF5 PWM 通道 5周期中断标志 当PMM5的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.
[4]	PWMIF4 PWM 通道 4周期中断标志 当PMM4的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.
[3]	PWMIF3 PWM 通道3周期中断标志 当PMM3的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.
[2]	PWMIF2 PWM 通道 2周期中断标志 当PMM2的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.
[1]	PWMIF1 PWM 通道 1周期中断标志 当PMM1的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.
[0]	PWMIF0 PWM 通道 0周期中断标志 当PMM0的计数器向下计数到0时，该位被置1. 注意: 该位可以被软件写1清0.

注意: 用户可以通过对PIIR相应位写1清中断标志

PWM输出控制寄存器(PWMPOE)

寄存器	偏移地址	R/W	描述	复位值
PWMPOE	PWM_BA+0x5C	R/W	PWM 通道 0~5输出使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱							
7	6	5	4	3	2	1	0
悵隱		PWM5	悵隱	PWM3	PWM2	悵隱	

位	描述	
[31:6]	悵隱	保留.
[5]	PWM5	PWM 通道 5输出使能位 0 = PWM 通道 5 输出禁止. 1 = PWM 通道 5 输出使能. 注意: 相应的GPIO 管脚必须切换到PWM功能
[4]	悵隱	悵隱
[3]	PWM3	PWM 通道 3输出使能位 0 = PWM 通道3 输出禁止. 1 = PWM 通道 3 输出使能. 注意: 相应的GPIO 管脚必须切换到PWM功能
[2]	PWM2	PWM 通道 2输出使能位 0 = PWM 通道 2 输出禁止. 1 = PWM 通道 2输出使能. 注意: 相应的GPIO 管脚必须切换到PWM功能
[1:0]	悵隱	保留.

PWM故障刹车控制寄存器(PFBCON)

寄存器	偏移地址	R/W	描述	复位值
PFBCON	PWM_BA+0x60	R/W	PWM故障刹车可控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
D7BK07	D6BK06	PWMBK05	悵隱	PWMBK03	PWMBK02	悵隱	
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱							
7	6	5	4	3	2	1	0
BKF	悵隱				CPO0BKEN	悵隱	BKEN0

位	描述	
[31]	D7BK07	通道 7 刹车输出电平选择位 0 = 当故障刹车产生后, 通道 7 输出low 1 = 当故障刹车产生后, 通道 7 输出high.
[30]	D6BK06	通道 6 刹车输出电平选择位 0 = 当故障刹车产生后, 通道 6 输出low 1 = 当故障刹车产生后, 通道 6 输出high.
[29]	PWMBK05	PWM 通道5刹车输出电平选择位 0 = 当故障刹车产生后, PWM 通道输出low 1 = 当故障刹车产生后, PWM 通道输出high.
[28]	悵隱	保留.
[27]	PWMBK03	PWM 通道3刹车输出电平选择位 0 = 当故障刹车产生后, PWM 通道输出low 1 = 当故障刹车产生后, PWM 通道输出high.
[26]	PWMBK02	PWM 通道2刹车输出电平选择位 0 = 当故障刹车产生后, PWM 通道输出low 1 = 当故障刹车产生后, PWM 通道输出high.
[25:8]	悵隱	保留.
[7]	BKF	PWM 故障刹车事件状态标志(写1清零) 0 = 当故障刹车产生后, PWM输出初始状态。 1 = 当故障刹车产生后, PWM输出故障刹车状态. 注意: 该位可以被软件写1清0, 但必须在PWM计数器启动前清零.

位	描述	
[6:3]	悵隱	保留.
[2]	CPO0BKEN	BKP0 故障刹车功能源选择位 0 = EINT1作为BKP1的一个刹车源. 1 = CP CPO0 O1 作为BKP1的一个刹车源.
[1]	悵隱	保留.
[0]	BKEN0	使能BKP0管脚触发故障刹车功能0 0 = 禁止BKP0管脚触发刹车功能 0(EINT0 或 CPO1) 1 = 使能BKP0管脚的下降沿可以触发刹车功能0.

PWM死区长度寄存器(PDZIR)

寄存器	偏移地址	R/W	描述	复位值
PDZIR	PWM_BA+0x64	R/W	PWM死区长度控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
DZI45							
15	14	13	12	11	10	9	8
DZI23							
7	6	5	4	3	2	1	0
DZI01							

位	描述	
[31:24]	悶隱	保留.
[23:16]	DZI45[7:0]	通道4 和 通道5 对(PWM4 和 PWM5 对)的死区长度寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由CSR位所决定.
[15:8]	DZI23[7:0]	通道2 和 通道3 对(PWM2 和 PWM3 对)的死区长度寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由CSR位所决定.
[7:0]	DZI01[7:0]	通道0 和 通道1 对(PWM0 和 PWM1对)的死区长度寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由CSR位所决定.

PWM触发ADC控制寄存器(TRGCON0)

寄存器	偏移地址	R/W	描述	复位值			
TRGCON0	PWM_BA+0x68	R/W	PWM 触发控制寄存器0	0x0000_0000			

31	30	29	28	27	26	25	24
保留				P3TRGEN	CM3TRGFEN	CNT3TRGEN	CM3TRGREN
23	22	21	20	19	18	17	16
保留				P2TRGEN	CM2TRGFEN	CNT2TRGEN	CM2TRGREN
15	14	13	12	11	10	9	8
保留				P1TRGEN	CM1TRGFEN	CNT1TRGEN	CM1TRGREN
7	6	5	4	3	2	1	0
保留				P0TRGEN	CM0TRGFEN	CNT0TRGEN	CM0TRGREN

位	描述	
[31:28]	保留	保留.
[27]	P3TRGEN	<p>通道3 在0点触发ADC使能位 当通道3的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[26]	CM3TRGFEN	<p>通道3 向下比较触发 ADC使能位 当在向下计数方向时，通道3的计数值与CMR3相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[25]	CNT3TRGEN	<p>通道3 中心点触发 ADC使能位 当通道3的计数值与CNR3相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM在边沿对齐模式下，设置该位无意义也无影响。</p>
[24]	CM3TRGREN	<p>通道3 向上比较触发 ADC使能位 当在向上计数方向时，通道3的计数值与CMR3相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM在边沿对齐模式下，设置该位无意义也无影响..</p>

位	描述	
[23:20]	保留	保留.
[19]	P2TRGEN	<p>通道2 在0点触发ADC使能位 当通道2的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[18]	CM2TRGFEN	<p>通道2 向下比较触发 ADC使能位 当在向下计数方向时，通道2的计数值与CMR2相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[17]	CNT2TRGEN	<p>通道2 中心点触发 ADC使能位 当通道2的计数值与CNR2相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响.</p>
[16]	CM2TRGREN	<p>通道2 向上比较触发 ADC使能位 当在向上计数方向时，通道2的计数值与CMR2相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响.</p>
[15:12]	保留	保留.
[11]	P1TRGEN	<p>通道1 在0点触发ADC使能位 当通道1的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[10]	CM1TRGFEN	<p>通道1 向下比较触发 ADC使能位 当在向下计数方向时，通道1的计数值与CMR1相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[9]	CNT1TRGEN	<p>通道1 中心点触发 ADC使能位 当通道1的计数值与CNR1相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响.</p>

位	描述
[8]	CM1TRGREN 通道1 向上比较触发 ADC使能位 当在向上计数方向时，通道1的计数值与 CMR1 相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意： 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响。
[7:4]	保留
[3]	P0TRGEN 通道0 在0点触发ADC使能位 当通道0的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意： 该位在中心对齐模式和边沿对齐模式下均有效
[2]	CM0TRGFEN 通道0 向下比较触发 ADC使能位 当在向下计数方向时，通道0的计数值与 CMR0 相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意： 该位在中心对齐模式和边沿对齐模式下均有效
[1]	CNT0TRGEN 通道0中心点触发 ADC使能位 当通道0的计数值与 CNR0 相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意： 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响。
[0]	CM0TRGREN 通道0 向上比较触发 ADC使能位 当在向上计数方向时，通道0的计数值与 CMR0 相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意： 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响。

PWM触发ADC控制寄存器(TRGCON1)

寄存器	偏移地址	R/W	描述	复位值
TRGCON1	PWM_BA+0x6C	R/W	PWM 触发控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
悶隱				P5TRGEN	CM5TRGFEN	CNT5TRGEN	CM5TRGREN
7	6	5	4	3	2	1	0
悶隱				P4TRGEN	CM4TRGFEN	CNT4TRGEN	CM4TRGREN

位	描述
[31:12]	悶隱 保留.
[11]	P5TRGEN 通道5在0点触发ADC使能位 当通道5的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效
[10]	CM5TRGFEN 通道5向下比较触发 ADC使能位 当在向下计数方向时，通道5的计数值与 CMR5 相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效
[9]	CNT5TRGEN 通道5中心点触发 ADC使能位 当通道5的计数值与 CNR5 相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响。
[8]	CM5TRGREN 通道5 向上比较触发 ADC使能位 当在向上计数方向时，通道5的计数值与 CMR5 相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响。

位	描述	
[7:4]	保留	保留.
[3]	P4TRGEN	<p>通道4在0点触发ADC使能位 当通道4的计数值等于0时使能PWM触发ADC功能。 0 = PWM条件触发ADC功能禁止. 1 = PWM条件触发ADC功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[2]	CM4TRGFEN	<p>通道4向下比较触发 ADC使能位 当在向下计数方向时，通道4的计数值与CMR4相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位在中心对齐模式和边沿对齐模式下均有效</p>
[1]	CNT4TRGEN	<p>通道4中心点触发 ADC使能位 当通道4的计数值与CNR4相等时使能 PWM 触发ADC功能 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响</p>
[0]	CM4TRGREN	<p>通道4 向上比较触发 ADC使能位 当在向上计数方向时，通道4的计数值与CMR4相等时使能PWM触发ADC功能。 0 = PWM 触发 ADC 功能禁止. 1 = PWM触发 ADC 功能使能. 注意: 该位仅在PWM中心对齐模式下有效。当PWM 在边沿对齐模式下，设置该位无意义也无影响.</p>

PWM触发状态寄存器(TRGSTS0)

寄存器	偏移地址	R/W	描述				复位值
TRGSTS0	PWM_BA+0x70	R/W	PWM触发状态寄存器0				0x0000_0000

31	30	29	28	27	26	25	24
保留			PERID3FLAG		CMR3FLAG_F	CNT3FLAG	CMR3FLAG_R
23	22	21	20	19	18	17	16
保留			PERID2FLAG		CMR2FLAG_F	CNT2FLAG	CMR2FLAG_R
15	14	13	12	11	10	9	8
保留			PERID1FLAG		CMR1FLAG_F	CNT1FLAG	CMR1FLAG_R
7	6	5	4	3	2	1	0
保留			PERID0FLAG		CMR0FLAG_F	CNT0FLAG	CMR0FLAG_R

位	描述
[31:28]	保留.
[27]	PERID3FLAG 通道3 在0点触发ADC标志 当通道3的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[26]	CMR3FLAG_F 通道3 向下比较触发ADC标志 当通道3的计数值与CMR相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[25]	CNT3FLAG 通道3中心点触发ADC 标志 当通道3的计数值与CNR相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[24]	CMR3FLAG_R 通道3向上比较触发ADC标志 当通道3的计数值向上比较到CMR时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.
[23:20]	保留.
[19]	PERID2FLAG 通道2 在0点触发ADC标志 当通道3的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[18]	CMR2FLAG_F 通道2 向下比较触发ADC标志 当通道2的计数器计数到CMR, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.

位	描述
[17]	CNT2FLAG 通道2中心点触发ADC 标志 当通道2的计数值与 CNR 相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[16]	CMR2FLAG_R 通道2向上比较触发ADC标志 当通道2的计数值向上比较到 CMR 时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.
[15:12]	保留.
[11]	PERID1FLAG 通道1 在0点触发ADC标志 当通道1的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[10]	CMR1FLAG_F 通道1向下比较触发ADC标志 当通道1的计数器计数到 CMR , 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[9]	CNT1FLAG 通道1中心点触发ADC 标志 当通道1的计数值与 CNR 相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[8]	CMR1FLAG_R 通道1向上比较触发ADC标志 当通道1的计数值向上比较到 CMR 时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.
[7:4]	保留.
[3]	PERID0FLAG 通道0 在0点触发ADC标志 当通道0的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[2]	CMR0FLAG_F 通道0向下比较触发ADC标志 当通道0的计数器计数到 CMR , 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[1]	CNT0FLAG 通道0中心点触发ADC 标志 当通道0的计数值与 CNR 相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[0]	CMR0FLAG_R 通道0向上比较触发ADC标志 当通道0的计数值向上比较到 CMR 时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.

PWM触发状态寄存器(TRGSTS1)

寄存器	偏移地址	R/W	描述	复位值
TRGSTS1	PWM_BA+0x74	R/W	PWM触发状态寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱				PERID5FLAG	CMR5FLAG_F	CNT5FLAG	CMR5FLAG_R
7	6	5	4	3	2	1	0
悵隱				PERID4FLAG	CMR4FLAG_F	CNT4FLAG	CMR4FLAG_R

位	描述	
[31:12]	悵隱	保留.
[11]	PERID5FLAG	通道5 在0点触发ADC标志 当通道5的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[10]	CMR5FLAG_F	通道5向下比较触发ADC标志 当通道5的计数器计数到 CMR , 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[9]	CNT5FLAG	通道5中心点触发ADC 标志 当通道5的计数值与 CNR 相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[8]	CMR5FLAG_R	通道5向上比较触发ADC标志 当通道5的计数值向上比较到 CMR 时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.
[7:4]	悵隱	保留.
[3]	PERID4FLAG	通道4 在0点触发ADC标志 当通道4的计数器计数到0点, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[2]	CMR4FLAG_F	通道4向下比较触发ADC标志 当通道4的计数器计数到 CMR , 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.

位	描述	
[1]	CNT4FLAG	通道4中心点触发ADC 标志 当通道4的计数值与 CNR 相等时, 该位将被置位来触发ADC. 注意: 该位可以被软件写1清0.
[0]	CMR4FLAG_R	通道4向上比较触发ADC标志 当通道4的计数值向上比较到 CMR 时, 该位将被置位来触发ADC.. 注意: 该位可以被软件写1清0.

相位修改寄存器(PHCHG)

寄存器	偏移地址	R/W	描述	复位值			
PHCHG	PWM_BA+0x78	R/W	相位修改寄存器	0x0000_3F00			

31	30	29	28	27	26	25	24
CE0	T0	CMP0SEL		CH31TOFF0	CH21TOFF0	帳隱	
23	22	21	20	19	18	17	16
CE1	T1	CMP1SEL		CH31TOFF1	CH21TOFF1	帳隱	
15	14	13	12	11	10	9	8
ACCNT1	ACCNT0	PWM5	帳隱	PWM3	PWM2	帳隱	
7	6	5	4	3	2	1	0
D7	D6	D5	帳隱	D3	D2	帳隱	

位	描述
[31]	CE0 ACMPO 触发PWM使能位 0 = ACMPO触发PWM禁止. 1 = ACMPO触发PWM使能. 注意: 如果ACCNT0被置位, 当ACMPO触发 PWM时该位被自动清零.
[30]	T0 TIMER0 触发PWM 使能位 0 = TIMER0触发PWM禁止. 1 = TIMER0 触发PWM使能. 当该位被置1, 当TIMER0溢出时, PHCHG_NXT寄存器的内容将更新到PHCHG寄存器
[29:28]	CMP0SEL[1:0] ACMPO 正极输入源选择位 00 = 选择P1.5 作为ACMPO的正极输入. 01 = 选择P1.0作为ACMPO的正极输入. 10 = 选择P1.2作为ACMPO的正极输入. 11 = 选择P1.3作为ACMPO的正极输入.
[27]	CH31TOFF0 ACMPO 触发通道3输出一个周期使能位 如果设置该位, ACMPO 触发该通道将强迫PWM 通道3 输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用 0 = PWM0_CH3 禁止输出一个周期的低电平. 1 = PWM0_CH3 使能输出一个周期的低电平. 注意: 该功能仅适用于PWM2, PWM3.

位	描述
[26]	<p>ACMP0 触发通道2输出一个周期使能位 如果设置该位, ACMP0 触发该通道将强迫PWM 通道2 输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用 0 = PWM0_CH2 禁止输出一个周期的低电平. 1 = PWM0_CH2使能输出一个周期的低电平. 注意: 该功能仅适用于PWM2, PWM3.</p>
[25:24]	保留.
[23]	<p>ACMP1 触发PWM使能位 0 = ACMP1触发PWM禁止. 1 = ACMP1触发PWM使能. 注意: 如果ACCNT1被置位, 当 ACMP1 触发 PWM时该位被自动清零.</p>
[22]	<p>T1 触发PWM 使能位 0 = TIMER1触发PWM禁止. 1 = TIMER1触发PWM使能. 当该位被置1, 当TIMER1溢出时, PHCHG_NXT寄存器的内容将更新到PHCHG</p>
[21:20]	<p>ACMP1 正极输入源选择位 00 = 选择P3.1作为ACMP1的正极输入. 01 = 选择P3.2作为ACMP1的正极输入. 10 = 选择P3.3作为ACMP1的正极输入. 11 = 选择P3.4作为ACMP1的正极输入.</p>
[19]	<p>ACMP1触发通道3输出一个周期使能位 如果设置该位, ACMP1触发该通道将强迫PWM 通道3 输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用 0 = 禁止输出一个周期的低电平. 1 = 使能输出一个周期的低电平. 注意: 该功能仅适用于PWM2, PWM3</p>
[18]	<p>ACMP1触发通道2输出一个周期使能位 如果设置该位, ACMP1触发该通道将强迫PWM 通道2输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用 0 = 禁止输出一个周期的低电平. 1 = 使能输出一个周期的低电平. 注意: 该功能仅适用于PWM2, PWM3.</p>
[17:16]	保留.
[15]	<p>ACMP1触发时硬件自动清CE1位 0 = 当 ACMP1触发PWM时硬件将自动清除该位. 1 = 当 ACMP1触发PWM时硬件不自动清除该位</p>
[14]	<p>当ACMP0触发时硬件自动清CE0位 0 = 当 ACMP0触发PWM时硬件将自动清除该位. 1 = 当 ACMP0触发PWM时硬件不自动清除该位</p>

位	描述	
[13]	PWM5	PWM通道5 输出屏蔽使能位 0 = 输出PHCHG寄存器中D5设定的电平 1 = 输出通道5中的原始波形
[12]	保留	保留.
[11]	PWM3	PWM通道3 输出屏蔽使能位 0 = 输出PHCHG寄存器中D3设定的电平 1 = 输出通道3中的原始波形
[10]	PWM2	PWM通道2输出屏蔽使能位 0 = 输出PHCHG寄存器中D2设定的电平 1 = 输出通道2中的原始波形
[9:8]	保留	保留.
[7]	D7	当MASK7为1, 通道7的输出电平为D7所设电平 0 = 输出low. 1 =输出high.
[6]	D6	当MASK6为1, 通道6的输出电平为D6所设电平 0 = 输出low. 1 =输出high.
[5]	D5	当PWM5为0, 通道6的输出电平为D5所设电平 0 = 输出low. 1 =输出high.
[4]	保留	保留.
[3]	D3	当PWM3为0, 通道6的输出电平为D3所设电平 0 = 输出low. 1 =输出high.
[2]	D2	当PWM2为0, 通道2的输出电平为D2所设电平 0 = 输出low. 1 =输出high.
[1:0]	保留	保留.

下一相位修改寄存器(PHCHGNXT)

寄存器	偏移地址	R/W	描述				复位值
PHCHGNXT	PWM_BA+0x7C	R/W	下一相位修改寄存器				0x0000_3F00

31	30	29	28	27	26	25	24
CE0	T0	CMP0SEL		CH31TOFF0	CH21TOFF0	帳隱	
23	22	21	20	19	18	17	16
CE1	T1	CMP1SEL		CH31TOFF1	CH21TOFF1	帳隱	
15	14	13	12	11	10	9	8
ACCNT1	ACCNT0	PWM5	帳隱	PWM3	PWM2	帳隱	
7	6	5	4	3	2	1	0
D7	D6	D5	帳隱	D3	D2	帳隱	

位	描述
[31]	CE0 ACMPO 触发PWM使能位 0 = ACMPO触发PWM禁止. 1 = ACMPO 触发PWM使能. 注意: 如果ACCNT0为1,当ACMPO触发PWM, 该位将自动被清零 .
[30]	T0 TMRO 触发PWM使能位 0 = TMRO触发PWM禁止. 1 = TMRO触发PWM使能 当该位被置1, 当TIMER0溢出时, PHCHG_NXT寄存器的内容将更新到PHCHG
[29:28]	CMP0SEL[1:0] ACMPO 正极输入源选择位 00 = 选择 P1.5做ACMPO的正极输入. 01 = 选择P1.0做ACMPO的正极输入. 10 = 选择P1.2做ACMPO的正极输入. 11 = 选择P1.3做ACMPO的正极输入.
[27]	CH31TOFF0 ACMPO 触发通道3输出一个周期使能位 如果设置该位, ACMPO 触发该通道将强迫PWM 通道3 输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用 0 = 禁止输出一个周期的低电平. 1 = 使能输出一个周期的低电平. 注意: 该功能仅适用于PWM2, PWM3.

位	描述
[26]	<p>CH21TOFF0</p> <p>ACMP0 触发通道2输出一个周期使能位</p> <p>如果设置该位, ACMP0 触发该通道将强迫PWM 通道2输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用</p> <p>0 = 禁止输出一个周期的低电平.</p> <p>1 = 使能输出一个周期的低电平.</p> <p>注意: 该功能仅适用于PWM2, PWM3.</p>
[25:24]	保留.
[23]	<p>CE1</p> <p>ACMP1 触发PWM使能位</p> <p>0 = ACMP1触发PWM禁止.</p> <p>1 = ACMP1触发PWM使能.</p> <p>注意: 如果ACCNT1被置位, 当 ACMP1 触发 PWM时该位被自动清零.</p>
[22]	<p>T1</p> <p>TMR1触发PWM 使能位</p> <p>0 = TIMER1触发PWM禁止.</p> <p>1 = TIMER1触发PWM使能.</p> <p>当该位被置1, 当TMR1溢出时, PHCHG_NXT寄存器的内容将更新到PHCHG</p>
[21:20]	<p>CMP1SEL[1:0]</p> <p>ACMP1 正极输入源选择位</p> <p>00 = 选择P3.1作为ACMP1的正极输入.</p> <p>01 = 选择P3.2作为ACMP1的正极输入.</p> <p>10 = 选择P3.3作为ACMP1的正极输入.</p> <p>11 = 选择P3.4作为ACMP1的正极输入.</p>
[19]	<p>CH31TOFF1</p> <p>ACMP1 触发通道3输出一个周期使能位</p> <p>如果设置该位, ACMP1 触发该通道将强迫PWM 通道3输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用</p> <p>0 = 禁止输出一个周期的低电平.</p> <p>1 = 使能输出一个周期的低电平.</p> <p>注意: 该功能仅适用于PWM2, PWM3.</p>
[18]	<p>CH21TOFF1</p> <p>ACMP1 触发通道3输出一个周期使能位</p> <p>如果设置该位, ACMP1 触发该通道将强迫PWM 通道2输出最长达一个周期的低电平, 该特性主要用于步进马达方面的应用</p> <p>0 = 禁止输出一个周期的低电平.</p> <p>1 = 使能输出一个周期的低电平.</p> <p>注意: 该功能仅适用于PWM2, PWM3.</p>
[17:16]	保留.
[15]	<p>ACCNT1</p> <p>当ACMP1触发时硬件自动清CE1位</p> <p>0 = 当ACMP1触发PWM时硬件将自动清除该位.</p> <p>1 = 当ACMP1触发PWM时硬件不自动清除该位</p>
[14]	<p>ACCNT0</p> <p>当ACMP0触发时硬件自动清CE0位</p> <p>0 = 当ACMP0触发PWM时硬件将自动清除该位.</p> <p>1 = 当ACMP0触发PWM时硬件不自动清除该位</p>

位	描述	
[13]	PWM5	PWMn通道5 输出屏蔽使能位 0 = 输出PHCHG寄存器中bit5位设定的电平 1 = 输出通道5的原始波形
[12]	保留	
[11]	PWM3	PWMn通道3 输出屏蔽使能位 0 = 输出PHCHG寄存器中bit3位设定的电平 1 = 输出通道3的原始波形
[10]	PWM2	PWMn通道2 输出屏蔽使能位 0 = 输出PHCHG寄存器中bit2位设定的电平 1 = 输出通道2的原始波形
[9:8]	保留	
[7]	D7	当MASK7为 1, 通道 7的输出电平为D7所设电平 0 =输出low. 1 =输出high.
[6]	D6	当MASK6为 1, 通道 6的输出电平为D6所设电平 0 =输出low. 1 =输出high.
[5]	D5	当PWM5为 0, 通道 5的输出电平为D5所设电平 0 =输出low. 1 =输出high.
[4]	保留	
[3]	D3	当PWM3为 0, 通道 3的输出电平为D3所设电平 0 =输出low. 1 =输出high.
[2]	D2	当PWM2为 0, 通道 2的输出电平为D2所设电平 0 =输出low. 1 =输出high.
[1:0]	保留	

相位修改屏蔽寄存器(PHCHGMASK)

寄存器	偏移地址	R/W	描述	复位值
PHCHGMASK	PWM_BA+0x80	R/W	相位修改屏蔽寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱						CMPMASK	
7	6	5	4	3	2	1	0
MASK7	MASK6	悵隱					

位	描述	
[31:10]	悵隱	保留.
[9]	CMPMASK[1]	ACMP1的正极输入控制 0 = ACMP的输入由CMP_CR1控制. 1 = ACMP 的输入由PHCHG寄存器的CMP1SEL位来控制.
[8]	CMPMASK[0]	ACMP0的正极输入控制 0 = ACMP的输入由CMP_CR0控制. 1 = ACMP 的输入由PHCHG寄存器的CMP0SEL位来控制.
[7]	MASK7	D7屏蔽使能位 0 = 输出 GPIO P0.0的原始信号. 1 = 输出 寄存器PHCHG D7位设定的电平
[6]	MASK6	D6屏蔽使能位 0 = 输出 GPIO P0.1的原始信号. 1 = 输出 寄存器PHCHG D6位设定的电平
[5:0]	悵隱	保留.

周期中断累加控制寄存器(INTACCUCTL)

寄存器	偏移地址	R/W	描述	复位值
INTACCUCTL	PWM_BA+0x84	R/W	周期中断累加控制寄存器	0x0000_00F0

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

位	描述	
[31:8]	保留	保留.
[7:4]	PERIODCNT[3:0]	中断累加计数器 当INTACCUENO被置1, 当PWMPIF0标志置1时PERIODCNT递减1, 当PERIODCNT递减到0, PWM0中断将产生, PERIODCNT的值将重载.
[3:1]	保留	保留.
[0]	INTACCUENO	中断累加使能位 0 = 中断累加禁止 1 = 中断累加使能.

5.11 看门狗定时器(WDT)

5.11.1 概述

设计看门狗定时器的目的是，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，看门狗定时器还支持系统从空闲/掉电模式唤醒功能。

5.11.2 特性

- 18位的自由向上计数的看门狗定时器，可用来满足用户溢出时间间隔要求
- 溢出时间间隔($2^4 \sim 2^{18}$)个WDT_CLK时钟周期可选，如果WDT_CLK = 10 kHz，那么溢出时间间隔是104ms ~ 26.3168
- 系统复位保持时间($1 / \text{WDT_CLK} * 63$)
- 支持看门狗定时器复位延时时间可选(仅NUC029xAN)
 - 支持可选择的WDT复位延时周期，包括(1026、130、18 或3) * WDT_CLK个WDT_CLK周期
- 配置CWDTE0 (CONFIG0[31] 看门狗使能)位为0(仅NUC029xAN)，可以使芯片上电或复位条件下看门狗强制打开。
- 如果WDT的时钟源选择为内部10 kHz时钟，支持看门狗定时器溢出唤醒功能

5.11.3 框图

看门狗定时器时钟框图如下：

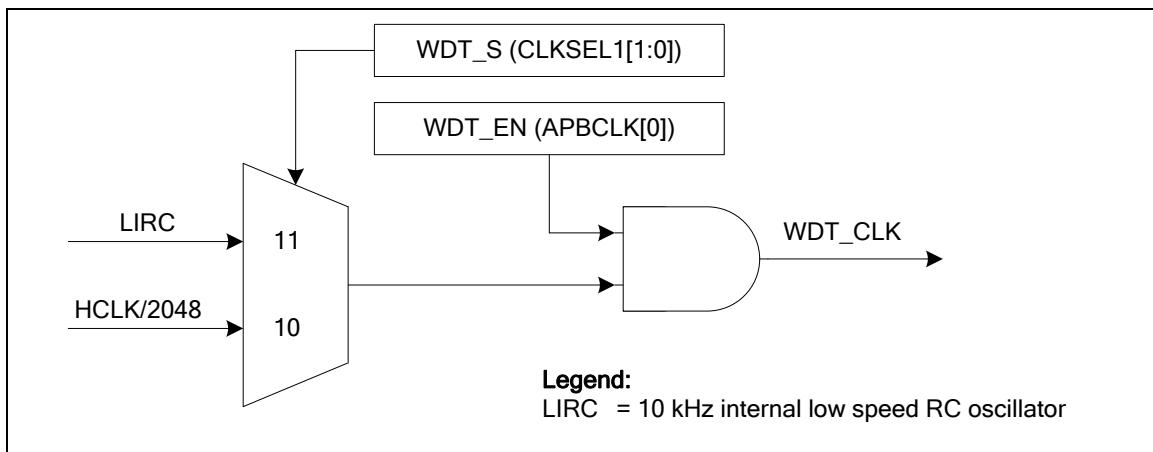


图 5-85 看门狗定时器时钟框图

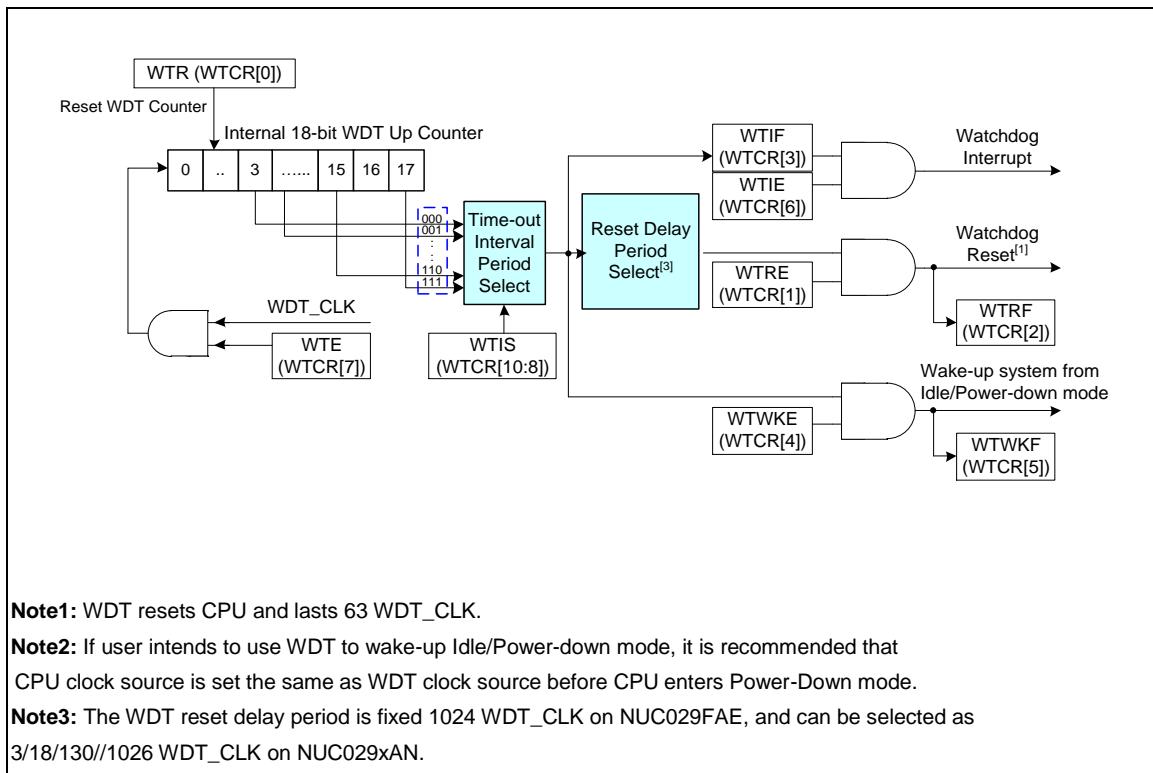


图 5-86 看门狗定时器模块图

5.11.4 基本配置

看门狗定时器的时钟通过WDT_EN(APBCLK[0])位使能，时钟源由WDT_S(CLKSEL1[1:0])位选

择。

当CONFIG[31]设置为0，NUC029xAN可以在在芯片上电或复位后，系统将强制使能看门狗定时器，并会激活10k时钟。

5.11.5 功能描述

看门狗定时器(WDT)包含一个18位的可设置溢出时间间隔的自由向上计数器。表 5-17 显示了WDT 溢出时间间隔选择，表 5-17 显示了WDT溢出时间间隔和复位周期时序。

5.11.5.1.1 WDT 定时溢出中断

对WTE(WTCR[7])置位可以使能WDT功能，然后WDT计数器开始向上计数。通过设置WTIS(WTCR[10:8])，可以选择8个不同的溢出时间间隔。当WDT计数器计到WTIS设定值时，WDT将产生溢出中断，然后WDT的溢出中断标志WTIF(WTCR[3])立即被置为1。

5.11.5.1.2 WDT 复位延时周期和复位系统

看门狗中断标志WTIF(WTCR[3])被置位后，会产生一个固定的复位延时周期 T_{RSTD} 。用户应在 T_{RSTD} 复位延时计完之前，对WTR(WTCR[0])置1，来复位18位的WDT向上计数器的值，以防止产生WDT时间溢出复位信号。如 T_{RSTD} 时间计满以后，WDT向上计数器的值仍没有被清除，WDT控制器将对WTRF(WTCR[2])置1，如果WTRE(WTCR[1])位有被使能，芯片立即进入复位状态。请参考图 5-87 看门狗定时器时间溢出间隔和复位周期时序， T_{RST} 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，WTRF(WTCR[2])标志将会保持为1. 用户可以通过检查WTRF(WTCR[2])标志来确认系统是否是因WDT定时溢出而产生的复位。

5.11.5.1.3 WDT 唤醒

如果看门狗时钟源选择为内部10k或者外部低速时钟，而且WTWKE(WTCR[4])位被使能，当看门狗定时溢出中断产生后，系统能从Power-down模式下唤醒。同时，WTWKF(WTCR[5])标志会被自动置1。用户可以通过查询WTWKF(WTCR[5])标志知道系统是否是被看门狗定时溢出中断唤醒。

WTIS	定时溢出间隔周期 T_{TIS}	复位延时周期(NUC029xAN)	
		T_{RSTD}	T_{RSTD}
000	$2^4 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
001	$2^6 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
010	$2^8 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
011	$2^{10} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
100	$2^{12} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
101	$2^{14} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
110	$2^{16} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$
111	$2^{18} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$	$1024 * T_{WDT}$

表 5-17 看门狗定时器定时溢出间隔周期选择

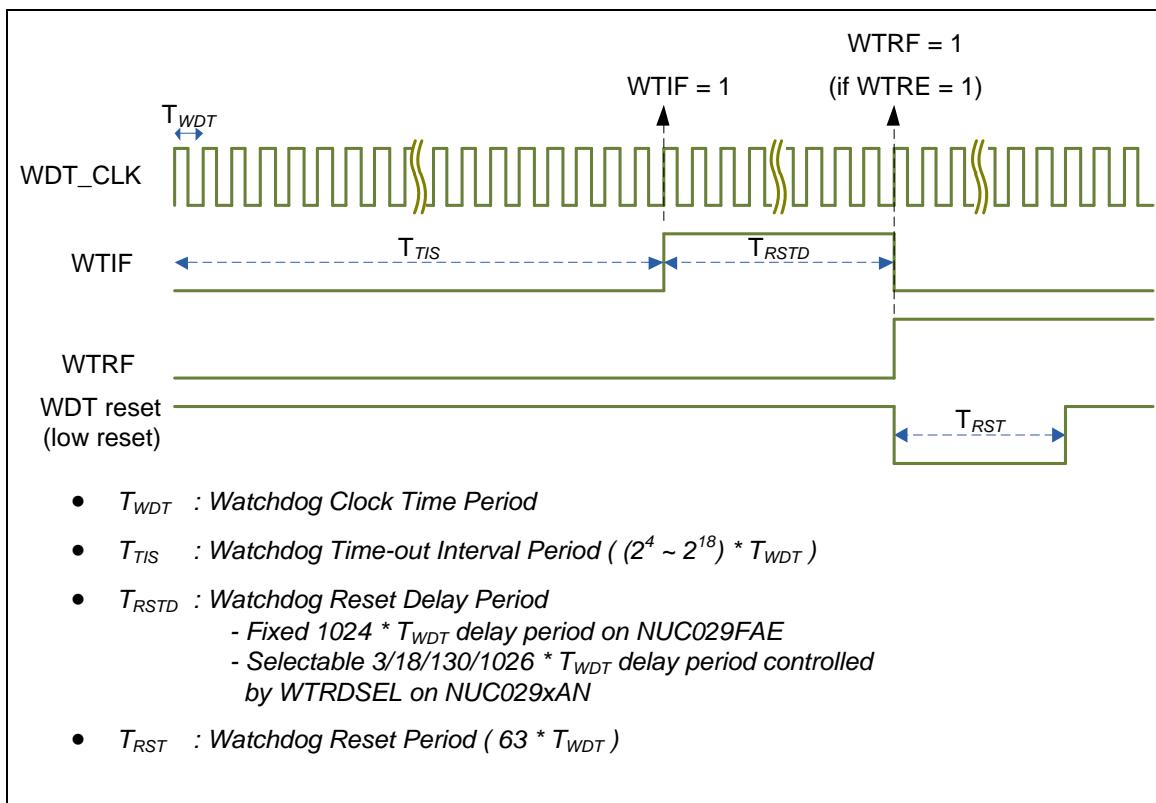


图 5-87 看门狗定时器定时溢出间隔和复位周期时序图

5.11.6 寄存器表

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移地址	R/W	描述	复位值
WDT基址址				
WDT_BA = 0x4000_4000				
WTCR	WDT_BA+0x00	R/W	看门狗定时器控制寄存器	0x0000_0700
WTCRALT	WDT_BA+0x04	R/W	看门狗定时器选择控制寄存器 (仅NUC029xAN)	0x0000_0000

5.11.7 寄存器表

看门狗定时器控制寄存器(WTCR)

寄存器	偏移地址	R/W	描述	复位值
WTCR	WDT_BA+0x00	R/W	看门狗定时器控制寄存器	0x0000_0700

注意:在编程这些位之前需将0x59, 0x16, 0x88写入到地址GCR_BA + 0x100的REGWRPROT寄存器，以解锁寄存器状态。

31	30	29	28	27	26	25	24
DBGACK_WDT	悵隱						
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱					WTIS		
7	6	5	4	3	2	1	0
WTE	WTIE	WTWKF	WTWKE	WTIF	WTRF	WTRE	WTR

位	Description	
[31]	DBGACK_WDT	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导致CPU运行停止时，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由ICE停止还是运行，看门狗继续计数。
[30:11]	Reserved	Reserved.
[10:8]	WTIS	看门狗定时器溢出时间间隔选择 (写保护) 以下三个位用于选择看门狗定时溢出周期 000 = $2^4 * T_{WDT}$. 001 = $2^6 * T_{WDT}$. 010 = $2^8 * T_{WDT}$. 011 = $2^{10} * T_{WDT}$. 100 = $2^{12} * T_{WDT}$. 101 = $2^{14} * T_{WDT}$. 110 = $2^{16} * T_{WDT}$. 111 = $2^{18} * T_{WDT}$.
[7]	WTE	看门狗定时器使能位 (写保护) 0 = 看门狗定时器禁止. (该操作会复位看门狗计数器的值.) 1 = 看门狗定时器使能.
[6]	WTIE	看门狗定时器定时溢出中断使能位(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.
[5]	WTWKF	看门狗定时器溢出唤醒标志 该位表示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 模式由看门狗定时溢出中断唤醒 注1: 该位为写1清零.
[4]	WTWKE	看门狗定时器定时溢出唤醒功能控制位 (写保护) 如果此位被置1, 当定时溢出中断标志WTIF被置1且中断使能位WTIE被使能时, 看门狗定时溢出中断信号将会触发事件唤醒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 注意: 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10k时钟。
[3]	WTIF	看门狗定时器定时溢出标志 当看门狗定时器计数器的值达到溢出时间间隔时, 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. 注意: 该位写1清零
[2]	WTRF	看门狗定时器复位标志 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. 注意: 该位写1清零.
[1]	WTRE	看门狗定时器复位使能位 (写保护) 该位是看门狗定时溢出复位功能使能位。如果看门狗计数器的值达到定时溢出的值, 且固定的WDT复位延时时间已经计完, 如果此时还没有对计数器清零, 如果该位被设置, 会发生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 注意: NUC029FAE的延时时间为 $1024 * T_{WDT}$ NUC029xAN 的延时时间时可选的, 3/18/130/1026 * TWDT
[0]	WTR	清看门狗计数器 (写保护) 0 = 不影响. 1 = 18位看门狗定时器计数值复位清零. 注意: 该位硬件自动清零

看门狗定时器选择控制寄存器(WTCRALT)

寄存器	偏移地址	R/W	描述	复位值
WTCRALT	WDT_BA+0x04	R/W	看门狗定时器选择控制寄存器 (仅NUC029xAN)	0x0000_0000

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

位	描述	
[31:2]	帳隱	保留.
[1:0]	WTRDSEL	<p>看门狗定时器复位延时选择 (写保护)</p> <p>当看门狗定时器发生溢出后，用户可以在看门狗复位延时的时段内，通过可以对看门狗计数器清零，以防止看门狗溢出复位。对不同的看门狗溢出时间，用户也可以通过设置RSTDSEL来选择适当的看门狗复位延时时间。</p> <p>00 = 看门狗定时器复位延时时间是$1026 * WDT_CLK$. 01 = 看门狗定时器复位延时时间是$130 * WDT_CLK$. 10 = 看门狗定时器复位延时时间是$18 * WDT_CLK$. 11 = 看门狗定时器复位延时时间是$3 * WDT_CLK$.</p> <p>注意：如果看门狗定时器定时溢出复位发生，该寄存器的值将被清零</p>

5.12 窗口式看门狗定时器 (WWDT) (NUC029xAN)

5.12.1 概述

NuMicro® NUC029xAN 支持窗口看门狗定时器(WWDT)用于在一个特定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

5.12.2 特性

- 6位向下计数值WWDTCVAL (WWDTVAL[5:0]) 和6位比较窗口值WINCMP (WWDTCR[21:16])，使得窗口周期更加灵活
- 支持4-位值(PSCSEL)选择窗口看门狗预分频值，预分频计数器（WWDT）最大可达11位

5.12.3 框图

窗口看门狗定时器的时钟控制及功能框图如下图。

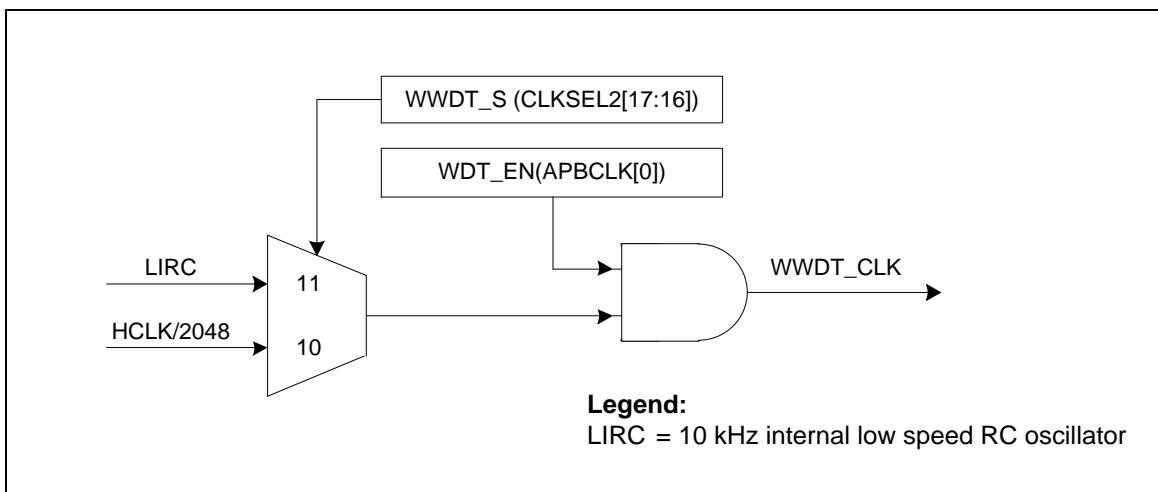


图 5-88 窗口看门狗定时器时钟控制

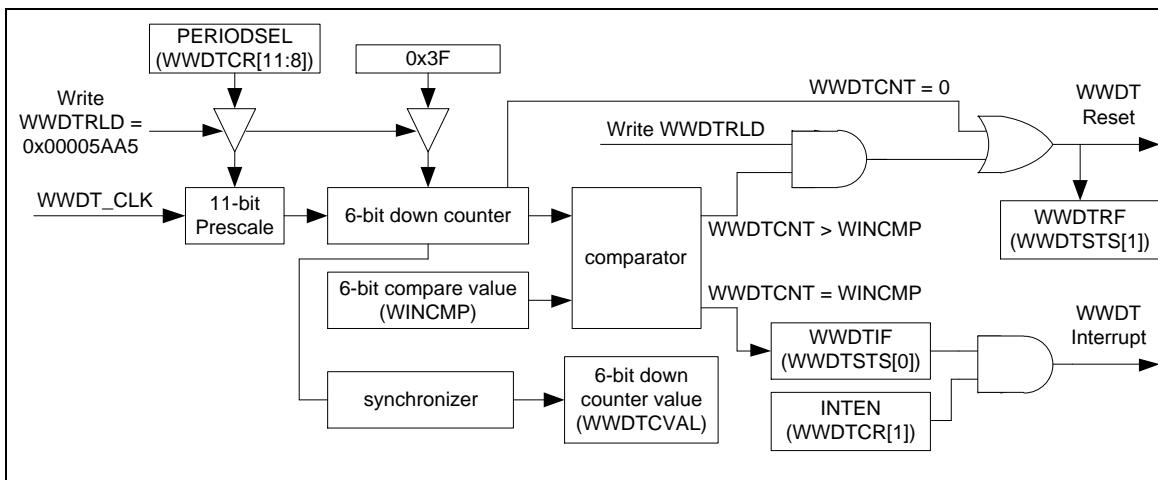


图 5-89 窗口看门狗定时器框图

5.12.4 基本配置

窗口看门狗定时器的外设时钟通过 CLK_APBCLK[0] 来使能，时钟源通过 CLK_CLKSEL2[17:16] 来选择

5.12.5 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10k低速RC振荡时钟源(LIRC)，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，对应预分频值如下

PERIODSEL	Prescale Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK=10 KHz)
0000	1	$1 * 64 * T_{WWDT}$	6.4 ms
0001	2	$2 * 64 * T_{WWDT}$	12.8 ms
0010	4	$4 * 64 * T_{WWDT}$	25.6 ms
0011	8	$8 * 64 * T_{WWDT}$	51.2 ms
0100	16	$16 * 64 * T_{WWDT}$	102.4 ms
0101	32	$32 * 64 * T_{WWDT}$	204.8 ms
0110	64	$64 * 64 * T_{WWDT}$	409.6 ms
0111	128	$128 * 64 * T_{WWDT}$	819.2 ms
1000	192	$192 * 64 * T_{WWDT}$	1.2288 s
1001	256	$256 * 64 * T_{WWDT}$	1.6384 s
1010	384	$384 * 64 * T_{WWDT}$	2.4576 s
1011	512	$512 * 64 * T_{WWDT}$	3.2768 s
1100	768	$768 * 64 * T_{WWDT}$	4.9152 s
1101	1024	$1024 * 64 * T_{WWDT}$	6.5536 s
1110	1536	$1536 * 64 * T_{WWDT}$	9.8304 s
1111	2048	$2048 * 64 * T_{WWDT}$	13.1072 s

表 5-18 窗口看门狗定时器预分频值选择

5.12.5.1.1 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。为了防止程序误操作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，用户不能禁止窗口看门狗定时器(WWDTEN)，不能修改计数器预分频周期(PERIODSEL)，也不能修改窗口比较值(WINCMPE)，除非芯片复位

5.12.5.1.2 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(WWDTCVAL)等于比较值(WINCMPE)时，WWDTIF会被置1，并且WWDTIF可以被软件清零。如果WWDTIE位被使能，当WWDTIF位被硬件置1时，就会产生窗口看门狗比较匹配中断

5.12.5.1.3 WWDT 复位系统

当WWDTIF产生后，用户必须通过对WWDT_RLDCNT寄存器写0x00005AA5进行重载使窗口看门狗计数器的值回到初始值0x3F，这样可以防止窗口看门狗定时器的值继续向下计数到0，从而产生窗口看门狗定时器复位系统信号通知系统复位。如果WWDTCVAL的当前值大于WINCMP的值，此时用户对WWDTRLD寄存器写0x00005AA5，窗口看门狗定时器复位系统信号将立刻产生，并导致芯片复位。

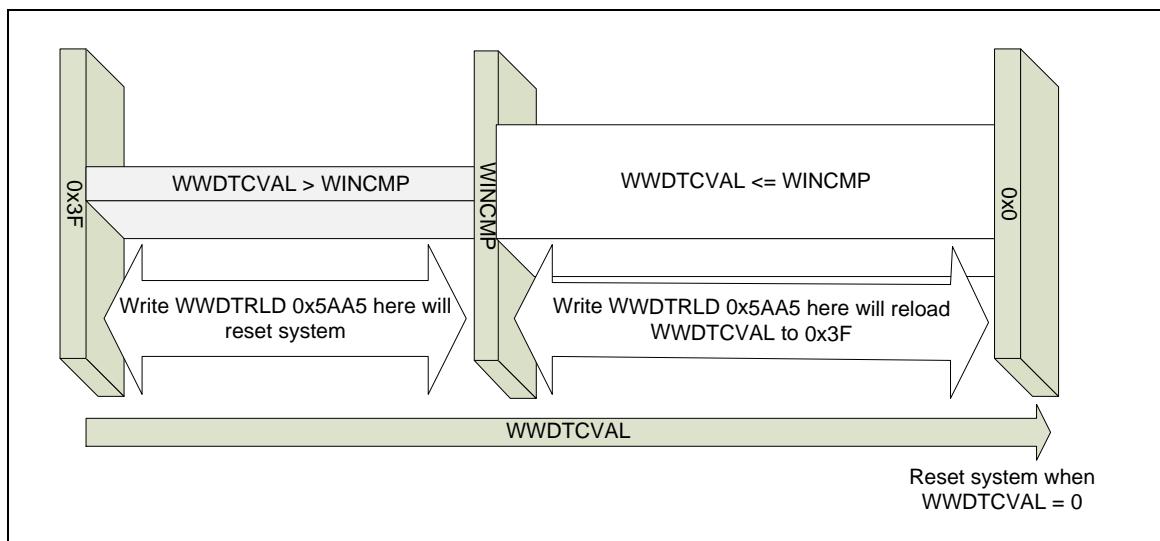


图 5-90 窗口看门狗定时器复位和重载

5.12.5.1.4 WWDT 窗口设置限制

当用户对WWDTRLD寄存器写0x00005AA5来重载窗口看门狗计数器的值到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PERIODSEL的值为0000，即计数器预分频值设置为1，WINCMP的值必须大于2；否则，当WWDTIF标志产生后，写WWDTRLD寄存器来重载窗口看门狗定时器的值为0x3F操作无效，将会一直出现窗口看门狗定时器引起的系统复位。

PERIODSEL	Prescale Value	Valid WINCMP Value
0000	1	0x3 ~ 0x3F
0001	2	0x2 ~ 0x3F
Others	Others	0x0 ~ 0x3F

表 5-19 WINCMP 设置限制

5.12.6 寄存器表

R: 只读, W: 只写, R/W: 可读写

寄存器	偏移量	R/W	描述	复位值
WWDT 基地址:				
WWDT_BA = 0x4000_4100				
WWDTRLD	WWDT_BA+0x00	W	窗口看门狗定时器重载计数器寄存器	0x0000_0000
WWDTCSR	WWDT_BA+0x04	R/W	窗口看门狗定时器控制寄存器	0x003F_0800
WWDTSR	WWDT_BA+0x08	R/W	窗口看门狗定时器状态寄存器	0x0000_0000
WWDTCSR	WWDT_BA+0x0C	R	窗口看门狗定时器计数器值寄存器	0x0000_003F

5.12.7 寄存器描述

窗口看门狗定时器重载计数器寄存器(WWDTRLD)

寄存器	偏移量	R/W	描述	复位值
WWDTRLD	WWDT_BA+0x00	W	窗口看门狗定时器重载计数器寄存器	0x0000_0000

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

位	描述
[31:0]	WWDTRLD 窗口看门狗定时器重载计数器寄存器 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:用户只能在当窗口看门狗计数器的值在0到WINCMP之间才可以写WWDTRLD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WINCMP时写WWDTRLD寄存器，窗口看门狗定时器复位信号会立即产生。

窗口看门狗定时器控制寄存器(WWDTCR)

寄存器	偏移量	R/W	描述	复位值
WWDTCR	WWDT_BA+0x04	R/W	窗口看门狗定时器控制寄存器	0x003F_0800

注意:此寄存器在芯片上电或复位后只能写一次

31	30	29	28	27	26	25	24
DBGACK_W WDT	帳隱						
23	22	21	20	19	18	17	16
帳隱		WINCMP					
15	14	13	12	11	10	9	8
帳隱				PERIODSEL			
7	6	5	4	3	2	1	0
帳隱						WWDTIE	WWDTEN

位	描述
[31]	DBGACK_WWDT 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，窗口看门狗定时器向下计数器将会保持继续计数，不管CPU是否被ICE暂停。
[30:22]	帳隱
[21:16]	WINCMP 窗口看门狗定时器窗口比较值 设置此寄存器来调整有效重载窗口 注: 当窗口看门狗定时器计数器的值在0到WINCMP之间时，用户才可以写寄存器WWDTRLD来重载窗口看门狗定时器计数器的值。如果在当前窗口看门狗定时器计数器的值大于WINCMP时用户去写WWDTRLD寄存器，窗口看门狗定时器复位信号会立即产生。
[15:12]	帳隱
[11:8]	PERIODSEL 窗口看门狗定时器计数器预分频周期选择位 0000 = 预分频为1; 最大定时溢出周期是 $1 * 64 * \text{WWDT_CLK}$. 0001 = 预分频为2; 最大定时溢出周期是 $2 * 64 * \text{WWDT_CLK}$. 0010 = 预分频为4; 最大定时溢出周期是 $4 * 64 * \text{WWDT_CLK}$. 0011 = 预分频为8; 最大定时溢出周期是 $8 * 64 * \text{WWDT_CLK}$. 0100 = 预分频为16; 最大定时溢出周期是 $16 * 64 * \text{WWDT_CLK}$. 0101 = 预分频为32; 最大定时溢出周期是 $32 * 64 * \text{WWDT_CLK}$. 0110 = 预分频为64; 最大定时溢出周期是 $64 * 64 * \text{WWDT_CLK}$. 0111 = 预分频为128; 最大定时溢出周期是 $128 * 64 * \text{WWDT_CLK}$. 1000 = 预分频为192; 最大定时溢出周期是 $192 * 64 * \text{WWDT_CLK}$.

		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]	悵隱	悵隱
[1]	WWDTIE	窗口看门狗定时器中断使能位 如果该位被使能，窗口看门狗定时器计数器比较匹配中断信号会产生，并通知CPU 0 = 窗口看门狗定时器计数器比较匹配中断禁止 1 = 窗口看门狗定时器计数器比较匹配中断使能
[0]	WWDTEN	窗口看门狗定时器使能位 设置该位来使能窗口看门狗定时器计数器计数 0 = 窗口看门狗定时器计数器停止。 1 = 窗口看门狗定时器计数器计数开始

窗口看门狗定时器状态寄存器(WWDTSR)

寄存器	偏移量	R/W	描述	复位值
WWDTSR	WWDT_BA+0x08	R/W	窗口看门狗定时器状态寄存器	0x0000_0000

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

位	描述	
[31:2]	帳隱	帳隱
[1]	WWDTRF	<p>窗口看门狗定时器溢出复位标志 该位指示系统是否已被窗口看门狗定时器溢出复位 0 = 窗口看门狗定时器没有发生复位 1 = 窗口看门狗定时器发生了复位 注: 该位是写1清零</p>
[0]	WWDTIF	<p>窗口看门狗定时器比较匹配中断标志 该位指示当窗口看门狗定时器计数器的值与WINCMP的值相匹配时，窗口看门狗定时器的比较匹配中断状态标志 0 = 窗口看门狗定时器计数器的值与WINCMP的值不匹配 1 = 窗口看门狗定时器计数器的值与WINCMP的值匹配 注: 该位是写1清零</p>

窗口看门狗定时器计数器值寄存器(WWDTCVR)

寄存器	偏移量	R/W	描述	复位值
WWDTCVR	WWDT_BA+0x0C	R	窗口看门狗定时器计数器值寄存器	0x0000_003F

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

位	描述	
[31:6]	帳隱	帳隱
[5:0]	WWDTCVAL	窗口看门狗定时器计数值 WWDTCVAL将会被持续更新，以指示6位向下计数器的值

5.13 UART 接口控制器 (UART)

5.13.1 概述

NuMicro® NUC029 提供2个通用异步收/发器（UART）通道，UART支持普通速度UART，并支持流控制。UART控制器对从外设收到的数据执行串到并的转换，对来自CPU的数据执行并到串的转换。UART控制器同时支持 IrDA SIR 功能和 RS-485 功能。NUC029xAN 还支持LIN主/从功能。每个UART控制器支持 6 种类型的中断。NUC029xAN 还拥有第七种中断，LIN 接收 **break** 域检测中断 (LIN_RX_BREAK_INT)。

5.13.2 特性

- 全双工，异步通信
- 独立的各 16 字节的接收/发送 FIFO 供用户装载数据
- 支持硬件自动流控/流控制功能(CTS, RTS)和可编程的RTS流控制触发电平
- 可编程的接收缓冲触发级别
- 每个通道都支持独立的可编程的波特率发生器
- 支持CTS 唤醒功能
- 支持8位接收缓冲超时侦测功能
- 通过设置DLY (UA_TOR [15:8]) 寄存器可以编程在上一个停止位与下一个开始位之间传输数据的延迟时间
- 支持**break**错误，帧错误，奇偶校验错误和接收/发送缓冲溢出检测功能
- 完全可编程的串行接口特性
 - 拥有可将数据长度编程为5位，6位，7位，8位的特性
 - 可编程的奇偶校验位，偶校验、奇校验和无校验或 Stick 校验位的产生和检测
 - 可编程的停止位长度，1、1.5 或 2 停止位的产生
- 支持 IrDA SIR 功能模式
 - 普通模式下支持 3/16位持续时间
- 支持 RS-485 功能模式
 - 支持 RS-485 9-位模式
 - 支持硬件或软件编程 RTS 引脚控制收发器的传输方向
- 支持 LIN 功能模式 (仅限 NUC029xA)
 - 支持LIN主/从模式
 - 支持发送端可编程的**break**产生功能
 - 支持接收端**break**检测功能

5.13.3 框图

UART 时钟控制和框图如下所示。

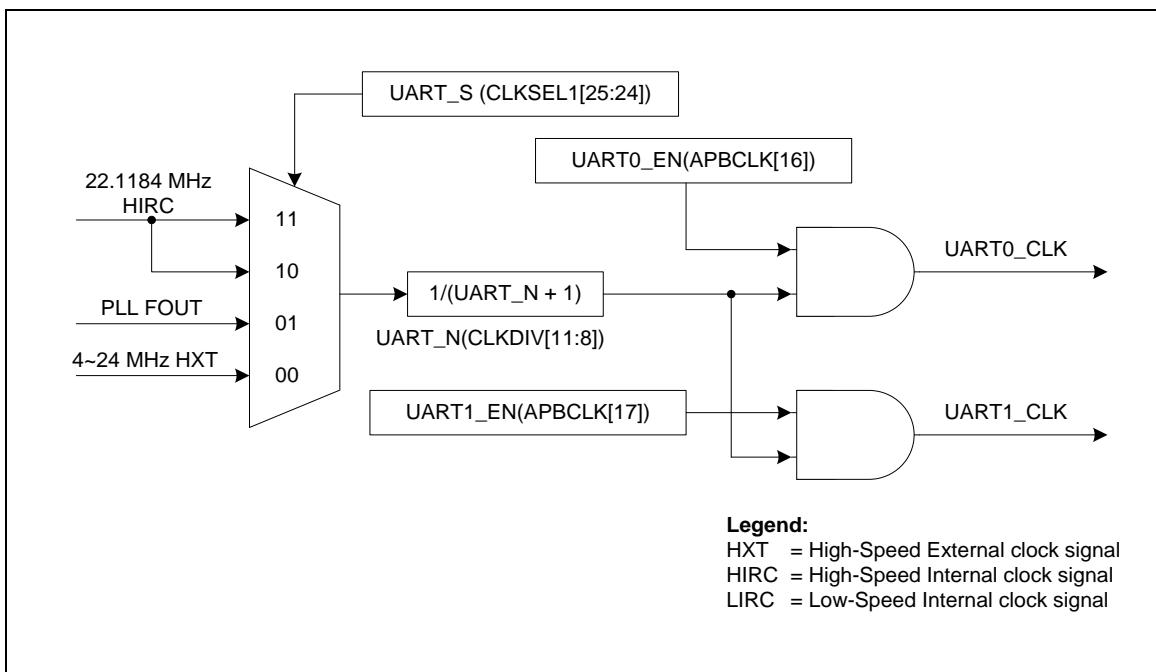


图 5-91 NUC029xAN UART 时钟控制框图

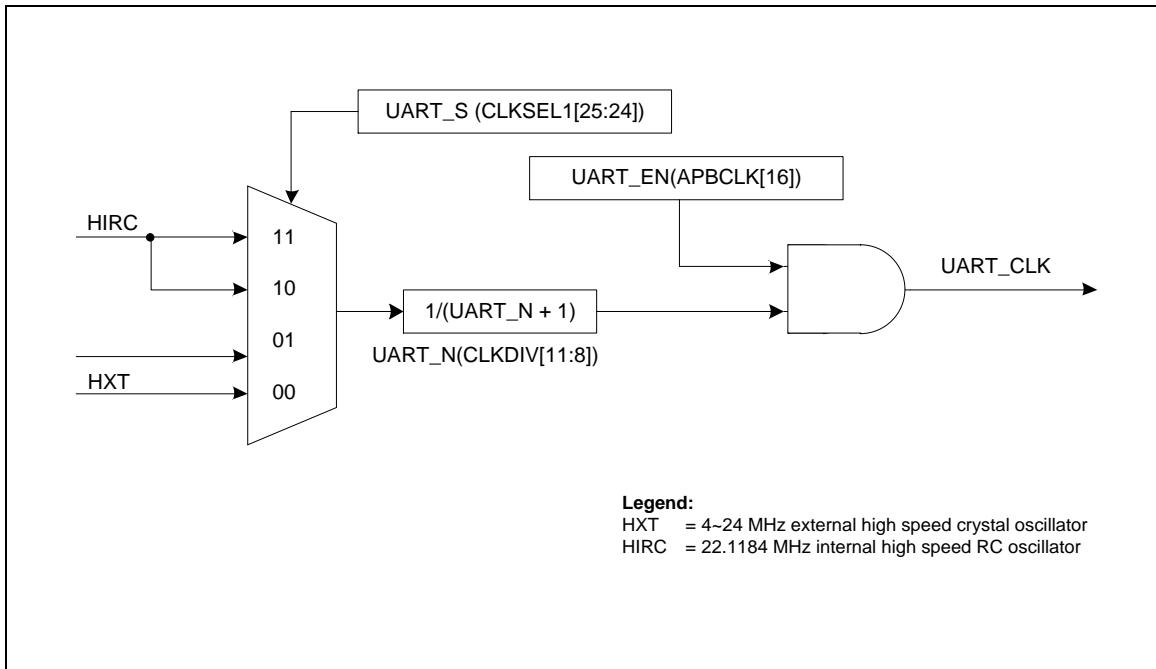


图 5-92 NUC029FAE UART 时钟控制框图

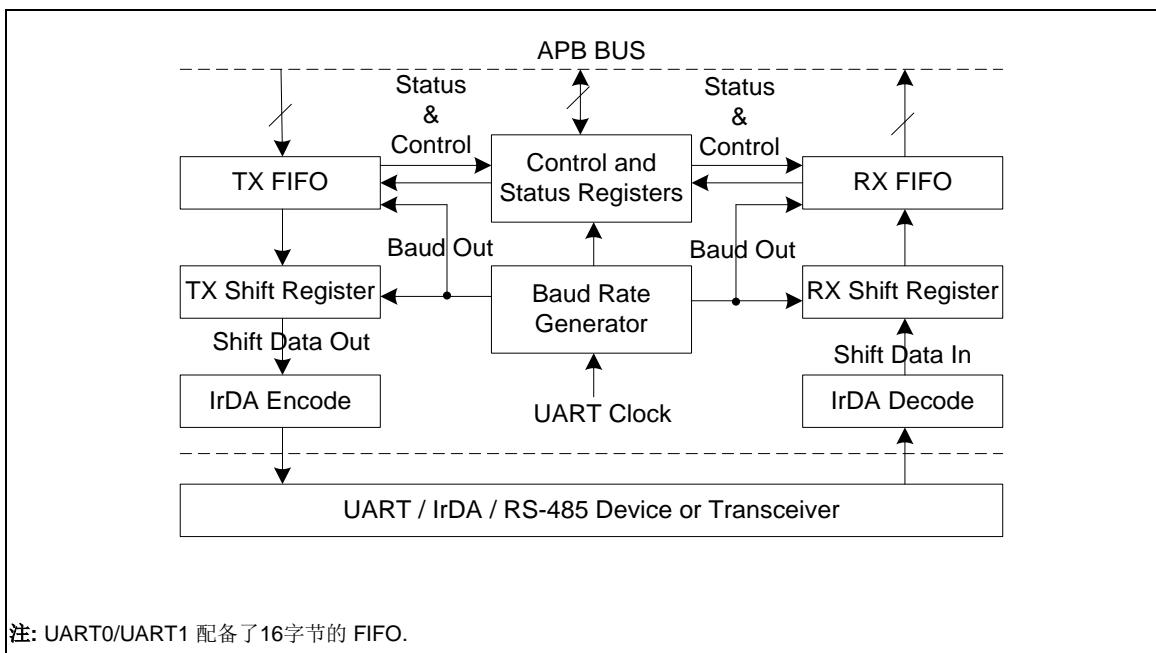


图 5-93 UART 控制器框图

每个方框的细节如下所述:

TX_FIFO (TX 缓存)

发送器用一个16字节的 FIFO 做缓存来降低向 CPU 提出中断请求的次数。

RX_FIFO (RX 缓存)

接收器用一个16字节的FIFO做缓存(每个字节加三个错误位)来降低向 CPU 提出中断请求的次数。

TX Shift Register (TX 移位寄存器)

该模块将要发送的数据改变为串行输出。

RX Shift Register (RX 移位寄存器)

该模块将接收到的数据改变为串行输入。

Baud Rate Generator (波特率发生器)

将外部时钟除以一个除数来获得需要的波特率时钟，参考波特率方程。

IrDA Encode (IrDA 编码)

IrDA 编码控制模块。

IrDA Decode (IrDA 解码)

IrDA 解码控制模块

Control and Status Register (控制和状态寄存器)

该组寄存器包含FIFO控制寄存器 (UA_FCR)、FIFO状态寄存器 (UA_FSR) 和用于收发器的线控制寄存器 (UA_LCR)。超时控制寄存器 (UA_TOR) 标识了超时中断的情况。该组寄存器还包含中断使能寄存器 (UA_IER) 和中断状态寄存器 (UA_ISR) 来使能或禁止中断响应并且标识发生

的中断响应。共有6种类型的中断：发送FIFO空中断（THRE_INT）、达到接收阈值中断（RDA_INT）、线状态中断（奇偶错误，帧错误和break中断）（RLS_INT）、超时中断（TOUT_INT）、缓存错误中断（BUF_ERR_INT）。NUC029xAN还有MODEM/唤醒状态中断（MODEM_INT），LIN接收break域侦测中断（LIN_RX_BREAK_INT）。

另外，自动流控框图如下所示：

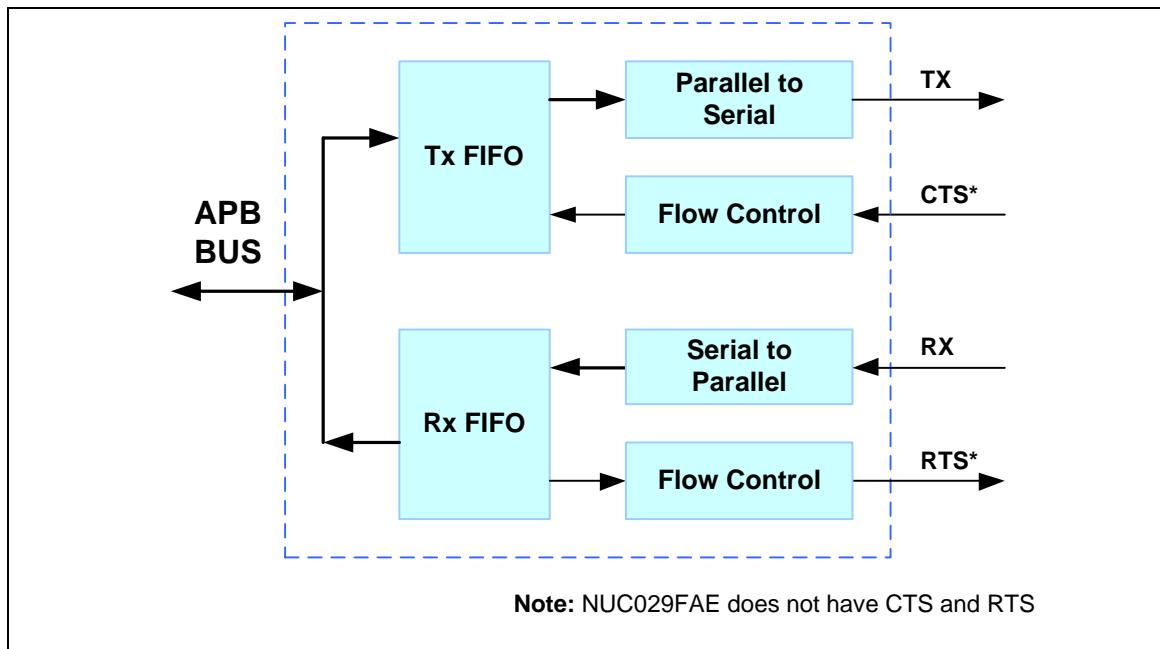


图 5-94 自动流控模块框图

5.13.4 基本配置

NUC029xAN:

UART0 功能引脚在 P0_MFP 寄存器中配置，UART1功能引脚在 P3_MFP 寄存器中配置。

UART0 的时钟在UART0_EN(APBCLK[16])中使能，UART1在UART1_EN(APBCLK[17])中使能。

UART 控制器时钟源在UART_S(CLKSEL[25:24])中选择。

UART 控制器时钟预分频值由UART_N(CLKDIV[11:8])决定。

NUC029FAE:

UART 功能引脚在 P1_MFP 寄存器中配置。

UART 在UART_EN(APBCLK[16])中使能。

UART 控制器时钟源在UART_S(CLKSEL[25:24])中选择。

UART 控制器时钟预分频值由UART_N(CLKDIV[11:8])决定

5.13.5 功能描述

UART控制器支持四种功能模式：UART模式、IrDA模式和RS-485模式。NUC029xAN还支持LIN模式。用户可以通过UA_FUN_SEL寄存器来选择相应的功能模式。

5.13.5.1 UART 控制器波特率发生器

UART控制器包括一个可编程的波特率发生器，它可以将输入时钟分频来得到收/发器需要的时钟。波特率公式是：波特率 = $UART_CLK / M * [BRD + 2]$ ，其中M和BRD在波特率分频寄存器UA_BAUD中定义。下表列出了不同条件下的波特率方程和UART波特率参数设置。通过设置相应的参数和寄存器可以得到零误差的波特率。IrDA模式下，波特率发生器必须工作在模式0。

模式	DIV_X_EN	DIV_X_ONE	Divider X	BRD	M	波特率公式
模式 0	0	0	B	A	16	$UART_CLK / [16 * (A+2)]$
模式 1	1	0	B	A	B+1	$UART_CLK / [(B+1) * (A+2)]$, B 必须 ≥ 8
模式 2	1	1	无关	A	1	$UART_CLK / (A+2)$, A 必须 ≥ 8

表 5-20 UART 控制器波特率公式表

UART 外设时钟 = 22.1184 MHz					
波特率	模式 0	模式 1		模式 2	
921600	不支持	A=0, B=11	0x2B00_0000	A=22	0x3000_0016
460800	A=1	A=1, B=15 A=2, B=11	0x2F00_0001 0x2B00_0002	A=46	0x3000_002E
230400	A=4	A=4, B=15 A=6, B=11	0x2F00_0004 0x2B00_0006	A=94	0x3000_005E
115200	A=10	A=10, B=15 A=14, B=11	0x2F00_000A 0x2B00_000E	A=190	0x3000_00BE
57600	A=22	A=22, B=15 A=30, B=11	0x2F00_0016 0x2B00_001E	A=382	0x3000_017E
38400	A=34	A=62, B=8 A=46, B=11 A=34, B=15	0x2800_003E 0x2B00_002E 0x2F00_0022	A=574	0x3000_023E
19200	A=70	A=126, B=8 A=94, B=11	0x2800_007E 0x2B00_005E	A=1150	0x3000_047E

		A=70, B=15	0x2F00_0046		
9600	A=142	A=254, B=8 A=190, B=11 A=142, B=15	0x2800_00FE 0x2B00_00BE 0x2F00_008E	A=2302	0x3000_08FE
4800	A=286	A=510, B=8 A=382, B=11 A=286, B=15	0x2800_01FE 0x2B00_017E 0x2F00_011E	A=4606	0x3000_11FE

表 5-21 UART 波特率参数和寄存器设置表

5.13.5.2 UART 控制器FIFO控制与状态

UART控制器内嵌一个16字节的发送FIFO (TX_FIFO) 和一个16字节的接收 FIFO (RX_FIFO)，用来降低向 CPU 提出中断请求的次数。CPU可以随时读UART的状态。回报的状态信息包括中断、正在被UART执行的传输操作的条件，以及接收数据时可能发生的3种错误情形（奇偶错误、帧错误和break错误）。FIFO控制和状态也支持所有的UART、IrDA、RS-485和LIN（仅限NUC029xAN）功能模式。

5.13.5.3 UART 控制器唤醒功能

当芯片进入睡眠模式时，外部的 CTS 状态改变将唤醒芯片。任何模式下都可以使用该唤醒功能。用户必须使能MODEN_INT 中断才能使用该唤醒功能。

5.13.5.4 UART 控制器中断和状态

每个UART控制器支持5种中断类型：

- 接收阙级别到达中断 (RDA_INT)
- 发送FIFO空中断 (THRE_INT)
- 线状态中断 (奇偶错误、帧错误或break错误) (RLS_INT)
- 接收缓存超时中断 (TOUT_INT)
- 缓存错误中断 (BUF_ERR_INT)

NUC029xAN 还支持另外两种类型的中断：

- MODEM/唤醒状态中断 (MODEM_INT)
- LIN接收break域侦测中断 (LIN_RX_BREAK_INT)

下表描述了中断源和中断标志。当中断使能且有中断标志时就会产生中断。用户必须在中断生产后清除中断标志。

中断源	中断指示	中断使能位	中断标志	清中断标志方式
接收数据有效中断	RDA_INT	RDA_IEN	RDA_IF	读 UA_RBR

发送保持寄存器空中断	THRE_INT	THRE_IEN	THRE_IF	写 UA_THR
接收 Line 状态中断 接收 Line 状态中断 Modem 状态中断 (仅限 NUC029xAN)	RLS_INT	RLS_IEN	RLS_IF = BIF 或 FEF 或 PEF	写 '1' 到 BIF/FEF/PEF
			RLS_IF = NUC029xAN: RS485_ADD_DETF NUC029FAE BIF 或 FEF 或 PEF 或 RS485_ADD_DETF	写 '1' 到 NUC029xAN: RS485_ADD_DETF NUC029FAE BIF/FEF/PEF/ RS485_ADD_DETF
Modem 状态中断 (仅限 NUC029xAN)	MODEM_INT	MODEM_IEN	MODEM_IF = DCTSF	写 '1' 到 DCTSF
RX 定时器溢出中断	TOUT_INT	RTO_IEN	TOUT_IF	读 UA_RBR
缓存错误中断	BUF_ERR_INT	BUF_ERR_IEN	BUF_ERR_IF = (TX_OVER_IF or RX_OVER_IF)	写 '1' 到 TX_OVER_IF / RX_OVER_IF
LIN RX Break 域侦测 中断 (仅限 NUC029xA)	LIN_RX_BREAK_INT	LIN_RX_BRK_IEN	LIN_RX_BREAK_IF	写 '1' 到 LIN_RX_BREAK_IF

表 5-22 UART 控制器中断源和标志列表

5.13.5.5 *UART 功能模式*

UART 控制器提供UART 功能 (用户需要设置 UA_FUN_SEL [1:0] 为 '00' 使能 UART 功能模式)。UART 波特率最高达 1 Mbps。

UART 提供全双工异步通讯。发送器和接收器都包含 16 个字节的 FIFO 用于存放数据。接收数据时，用户可以编程接收缓存触发级别和接收缓存超时中断。发送数据时，上一个停止位和下一个开始位之间的延迟时间，可以通过 DLY (UA_TOR [15:8]) 寄存器设定。UART 支持硬件自动流控和流控制功能(CTS, RTS)，RTS 流控触发电平可编程，并且串行接口的所有特性都可以编程的。

UART 线控制功能

通过设置UA_LCR寄存器，UART 控制器全部的串行接口特性都支持可编程。软件上可以通过 UA_LCR 寄存器来编程字长度、停止位和校验位。下表列出了UART 字长度和停止位长度的设定和校验位的设定。

NSB (UA_LCR[2])	WLS (UA_LCR[1:0])	字长度 (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 5-23 UART 线控制的字长度和停止位长度的设置

奇偶校验 类型	SPE (UA_LCR[5])	EPE (UA_LCR[4])	PBE (UA_LCR[3])	描述
无校验	x	x	0	无奇偶校验位输出
奇校验	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
偶校验	0	1	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。
奇偶校验位强制置1	1	0	1	奇偶校验位总是逻辑1
奇偶校验位强制置0	1	1	1	不管数据位中1的个数是多少（奇偶计数），奇偶校验位永远都是逻辑1

表 5-24 UART 线控制校验位的设置

注：当TE_FLAG(UA_FSR[28])为非空时，用户不能改变线控制设置

UART 自动流控功能（仅限NUC029xAN）

UART支持自动流控功能，该功能用到两根信号线CTS (清零发送) 和RTS (请求发送)来控制UART与外部设备（如Modem）间的的数据传输。当自动流控使能后，只有等到UART对外部设备发出有效的RTS信号后才允许接收数据，否则，不接收。当RX FIFO接收到数据的数量达等于RTS_TRI_LEVEL(UA_FCR [19:16])位的值后，RTS信号会被取消。当UART检测到外部设备给CTS信号脚有效信号后，UART开始发送数据，否则，UART不会发送数据。

下面的框图展示了UART CTS自动流控功能模式。用户必须设置AUTO_CTS_EN (UA_IER [13])来使能CTS自动流控功能。LEV_CTS (UA_MCR [8])可以设置CTS管脚为输入的有效状态。CTS管脚上任何输入信号的变化将导致DCTS(UA_MSR [0])被置1，然后TX数据将从TX FIFO中自动发出。

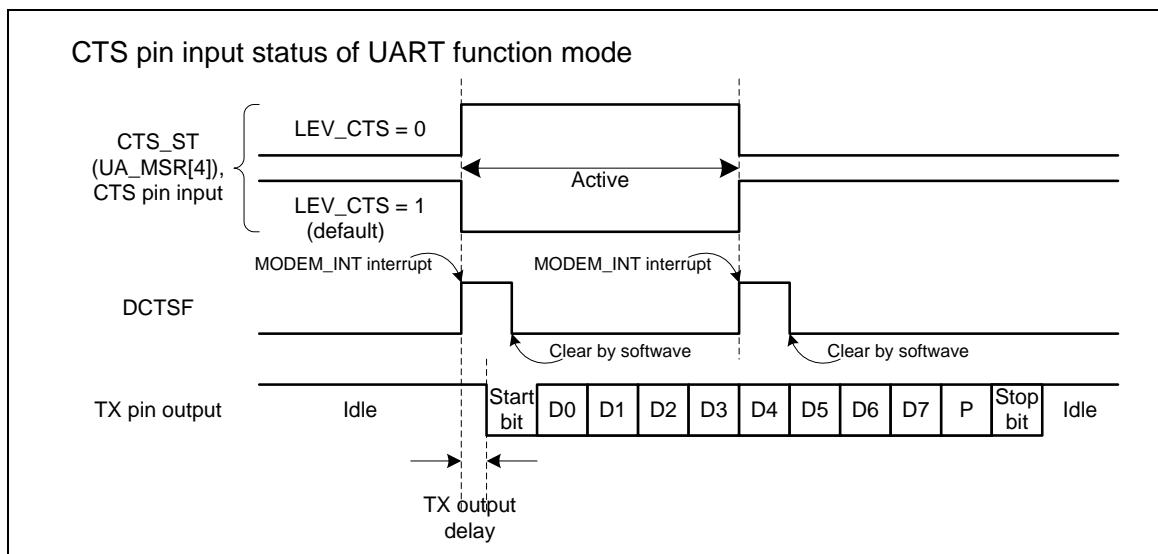


图 5-95 UART CTS 自动流控功能使能

如下图所示，UART RTS 自动流控模式(AUTO_RTS_EN(UA_IER[12])=1)下，内部nRTS信号由UART FIFO控制器控制，根据RTS_TRI_LEV(UA_FCR[19:16])的设定电平触发。

设定 LEV_RTS(UA_MCR[9]) 可以控制 RTS 引脚将nRTS信号是反转输出还是不反转输出。用户可以读RTS_ST(UA_MCR[13]) 位来得到真实的 RTS 引脚输出电平逻辑状态。

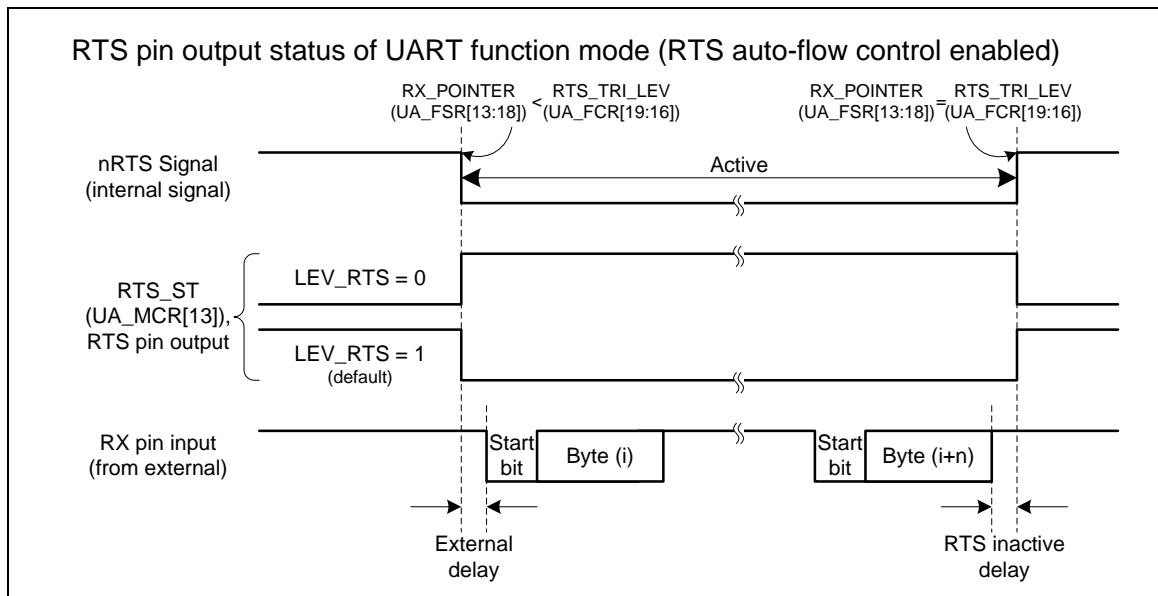


图 5-96 UART RTS自动流控功能使能

如下图所示，软件模式下 (AUTO_RTS_EN(UA_IER[12])=0)，软件通过编程RTS(UA_MCR[1])控制位，来实现对RTS 流控的直接控制。

设置 LEV_RTS(UA_MCR[9]) 可以控制RTS 引脚将RTS(UA_MCR[1])位反转输出，还是不反转输出。用户可以读RTS_ST(UA_MCR[13]) 位来得到实际的 RTS 引脚输出电平逻辑状态。

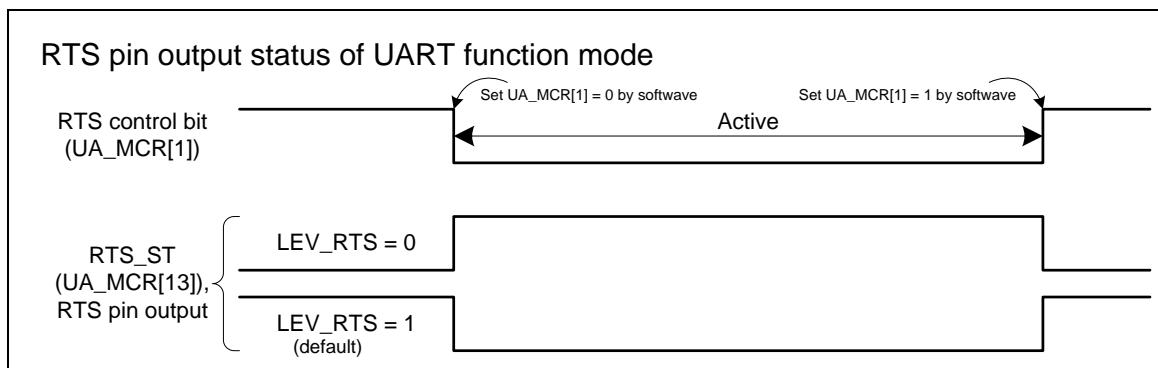


图 5-97 UART RTS 软件控制流程

5.13.5.6 IrDA 功能模式

UART控制器也提供Serial IrDA (SIR, 串行红外)功能(用户必须设置IrDA_EN(UA_FUN_SEL[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，包括一个起始位，8个数据位，一个停止位。最大数据速率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有10ms的时间间隔，该延迟特性需通过软件来完成。

在IrDA模式下，寄存器DIV_X_EN (UA_BAUD [29])必须被禁止。

波特率 = Clock / (16 * BRD)，这里BRD是UART_BAUD寄存器中定义的波特率分频器。

下图展示了IrDA控制模块框图：

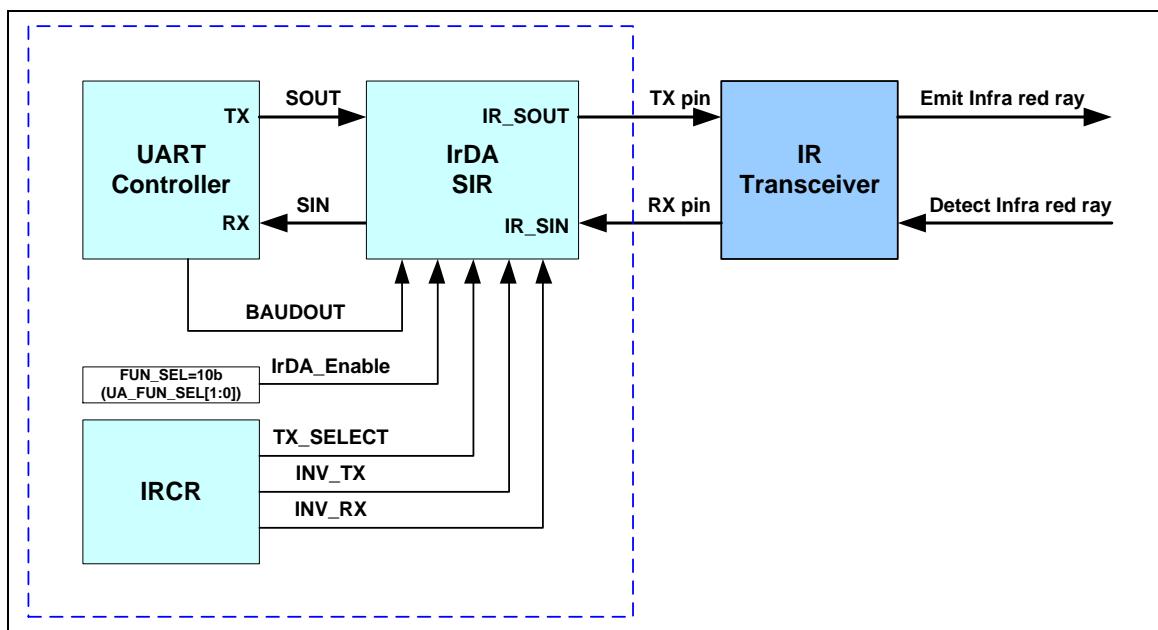


图 5-98 IrDA 控制模块框图

IrDA SIR 发送编码器

IrDA SIR 传送编码调制采用 Non-Return-to Zero (NRZ)不归零编码，将数据流从 UART输出。IrDA

SIR 物理层指定使用归零反向调制编码 (Return-to-Zero, Inverted (RZI)), 用一个红外光脉冲代表逻辑 0, 被调制的输出脉冲流被发送到外部输出驱动器和红外光发射二极管。

在正常模式下, 传输脉冲的宽度为 3/16 波特率周期。

IrDA SIR 接收解码器

IrDA SIR 接收解码器以归零(Return-to-Zero)方式解调由输入探测器输入的比特流, 并输出 NRZ比特流到 UART 作为数据输入。解码器在空闲模式输入通常为高。(因为IRCR (INV_RX [6]) 默认设定为1)

当解码器输入低时, 表示一个起始位。

IrDA SIR 操作

IrDA SIR 编码/解码器提供UART数据流和半双工串行SIR接口间互相转换的功能。. 下图是IrDA编码/解码器波形图。

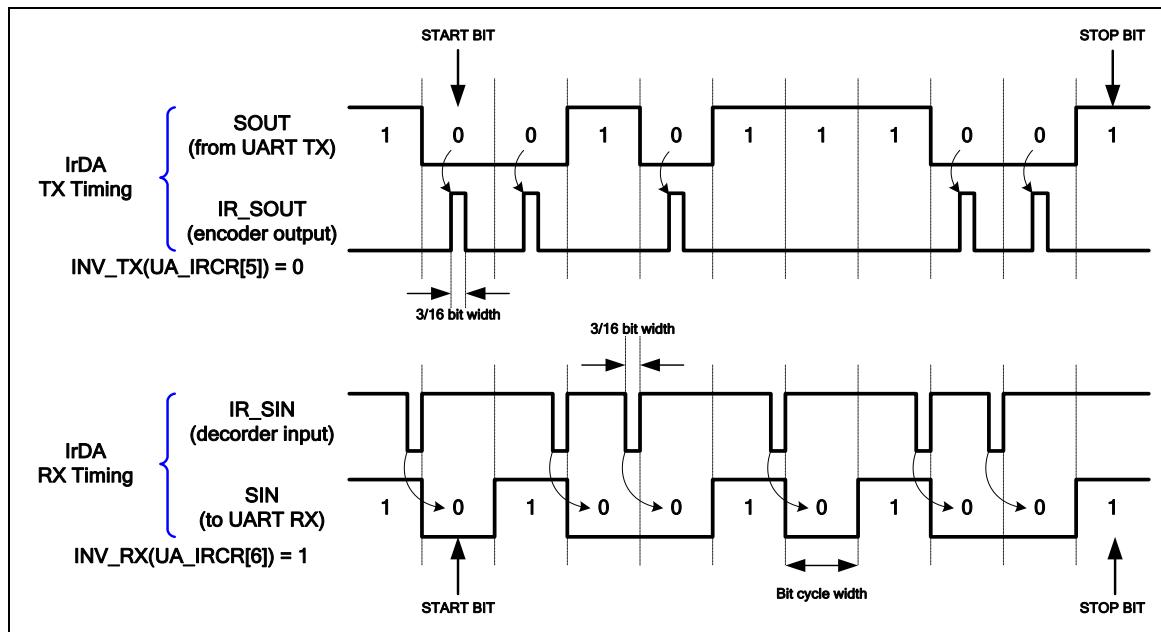


图 5-99 IrDA TX/RX 时序框图

5.13.5.7 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UA_FUN_SEL [1:0]为“11”来使能RS-485功能）。RS-485收发器的控制是通过使用异步串行接口的RTS控制信号, 来使能RS-485驱动。RS-485模式下的RX和TX大多数特性与UART相同。

对于NUC029xAN而言, 方向控制可以由RTS引脚或者对GPIO编程 (RTS0使用P0.3 和 RTS1使用P0.1) 来由软件实现。对于NUC029FAE而言, 用户必须通过软件控制RTS (UA_MCR[1]) 来直接控制RTS信号。

控制器可以被配置成可寻址的RS-485从机, RS-485主机发送器可以通过设置奇偶校验位 (第九位) 为1来确认一个地址字符。对于数据字符而言, 奇偶校验位应设置为0。软件可以通过设置寄存

器UA_LCR来控制第九位（当PBE、EPE和SPE置位时，第九位发送0，当PBE和EPE置位时，且SPE清零时，第九位发送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 自动地址识别操作模式(AAD)和RS-485自动方向控制操作模式(AUD)。软件可通过编程UA_ALT_CSR寄存器选择任何工作模式，通过设置DL(UA_TOR [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)

在RS-485普通多点操作模式下，首先软件必须决定在侦测到地址字节之前的数据是否存储到RX FIFO中。如果软件想忽略任何检测到地址字节之前的数据，则流程应设置RX_DIS (UA_FCR [8])，然后使能RS485_NMM (UA_ALT_CSR [8])，这样，接收器忽略所有的数据直至检测到地址字节 (bit9=1) 并将地址字节数据存储于RX FIFO中。如果软件想接收检测到地址位之前的所有数据，则流程为禁止RX_DIS (UA_FCR [8])，然后使能RS485_NMM (UA_ALT_CSR [8])，这样，接收器就会接收所有数据。

如果检测到地址字节(bit 9 = 1)，RS-485控制器会向CPU产生一个中断，软件可通过设定RX_DIS (UA_FCR [8])来决定是否接收后面的数据字节并保存到RX FIFO中。若软件设置RX_DIS (UA_FCR [8])关闭接收功能，当检测到下一个地址字节时，控制器会自动清RX_DIS (UA_FCR [8])位，并将地址字节数据存储到RX-FIFO。

RS-485 自动地址识别操作模式 (AAD)

在RS-485自动地址识别操作模式下，接收器在检测到地址字节 (bit9=1) 并且地址字节数据与ADDR_MATCH (UA_ALT_CSR[31:24])的值相匹配之前，忽略所有数据。然后地址字节数据将被存储在RX FIFO中。所有收到的字节数据将被接受，并存储在RX FIFO中，直到收到的地址字节不匹配ADDR_MATCH (UA_ALT_CSR[31:24]) 的值为止。

RS-485 自动方向模式 (AUD)

RS-485控制器的另一个可选的功能是**RS-485自动方向控制功能**。用户必须设置RS485_AUD (UA_ALT_CSR[10])为1来使能RS-485自动方向模式。使用来自异步串行口的RTS控制信号来控制RS-485收发器。RTS线被连接到RS-485收发器使能引脚上，以便设置RTS线为高（逻辑1）使能RS-485收发器；设置RTS为低（逻辑0），使收发器进入三态状态。用户通过设置寄存器UA_MCR 中的LEV_RTS来改变RTS驱动电平。

下图说明了在AUD模式下，RS-485 RTS驱动电平。当TX数据传输时，RTS引脚将被自动驱动。

设置 LEV_RTS(UA_MCR[9])可以控制RTS引脚输出驱动电平。用户可以读RTS_ST(UA_MCR[13])位来得到RTS引脚上实际输出的电平逻辑状态。

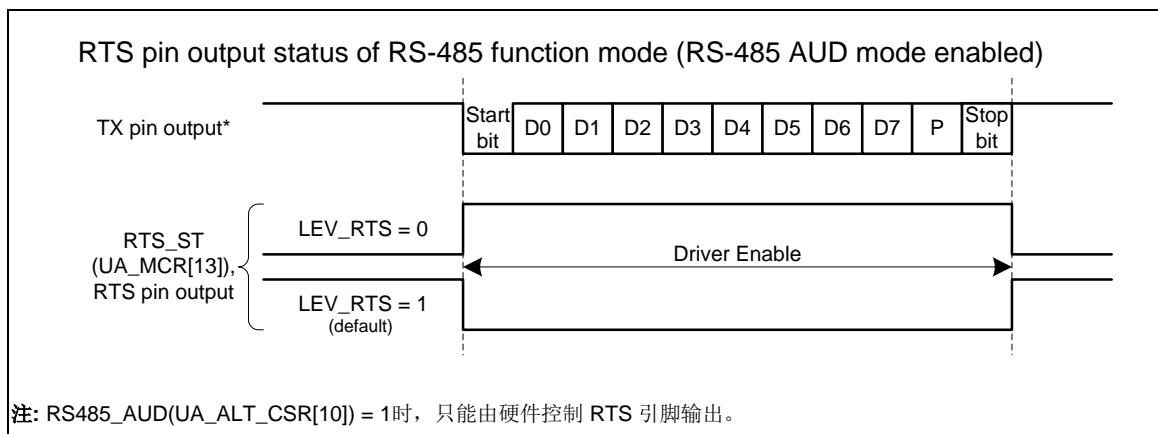


图 5-100 RS-485 自动方向模式下的 RTS 驱动电平

下面的框图展示了通过软件(RS485_AUD(UA_ALT_CSR[10])=0)控制RS-485 RTS驱动电平。RTS驱动电平通过设置RTS(UA_MCR[1])位来控制。

设置RTS(UA_MCR[1])位可以控制RTS管脚信号是否有效，通过设置LEV_RTS(UA_MCR[9])可以控制输出电平是否反向。用户可以读RTS_ST(UA_MCR[13])位来获取RTS管脚上实际的输出电压逻辑状态。

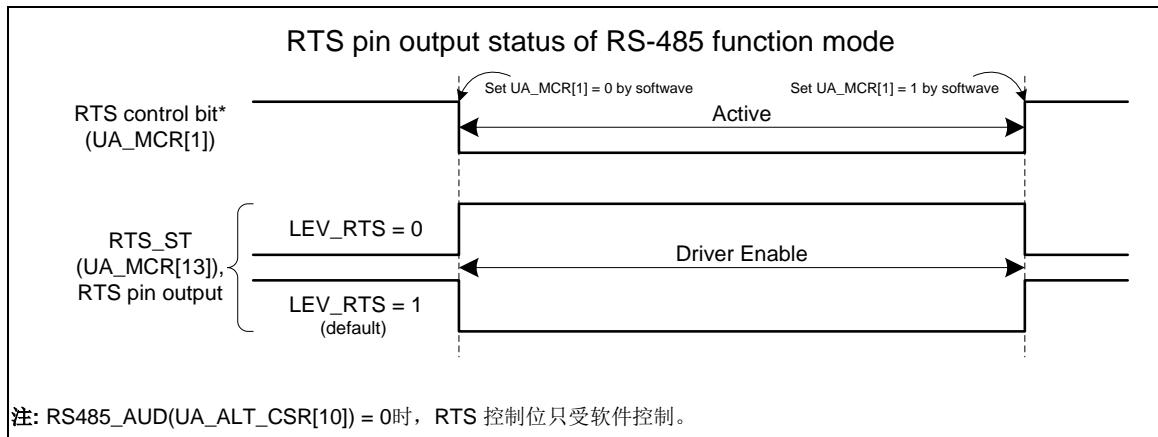


图 5-101 软件控制下的RS-485 RTS 驱动电平

编程流程示例:

1. 设置 FUN_SEL(UA_FUN_SEL[1:0])来选择 RS-485 功能。
2. 设置 RX_DIS(UA_FCR[8])位来决定使能或禁用RS-485接收器。
3. 设置 RS485_NMM 模式或者 RS485_AAD 模式。
4. 如果选择RS485_AAD(UA_ALT_CSR[9])模式， ADDR_MATCH(UA_ALT_CSR[31:24])需要填入自动地址匹配值。
5. 设置 RS485_AUD(UA_ALT_CSR[10]) 来决定是否为自动方向控制。

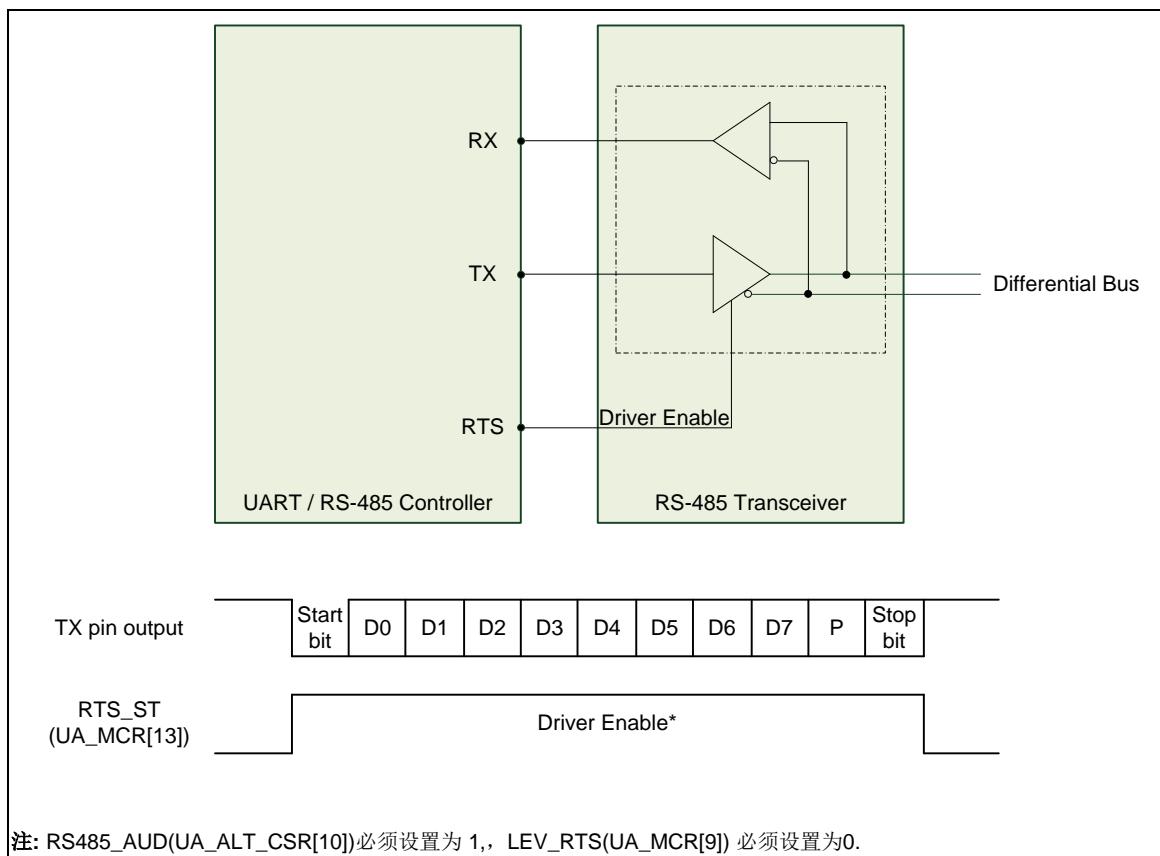


图 5-102 RS-485 帧结构

5.13.5.8 LIN (Local Interconnection Network) 功能模式 (仅限 NUC029xAN)

UART 控制器支持 LIN 功能（用户必须设定 UA_FUN_SEL[1:0] 为 01 来使能 LIN 功能）。在 LIN 模式下，根据 LIN 的标准，每个字节域的开始都是由一个值为 0 的起始位（显性的），后面跟随 8 个位的数据（LSB 优先），最后以一个值为 1 的停止位（隐性的）结束。下图是 LIN 功能模式的帧结构：

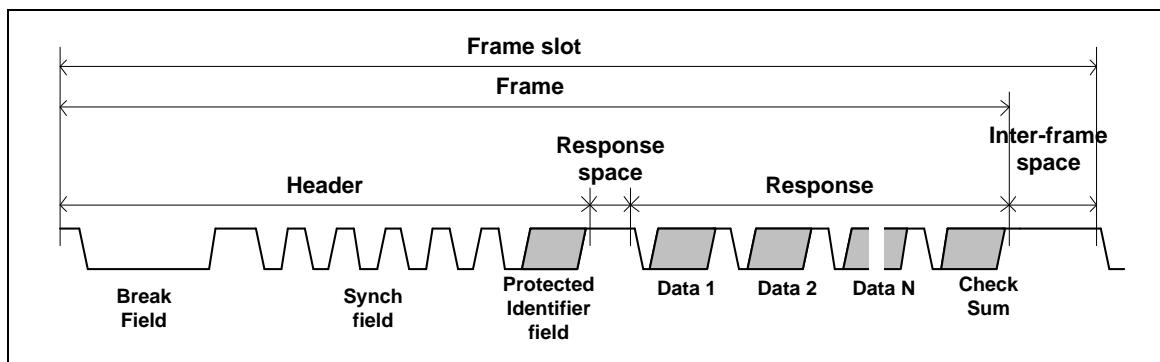


图 5-103 LIN 的帧结构图

LIN总线发送 (TX) 编程流程如下：

1. 设置 UA_FUN_SEL[1:0] 为 01 来使能 LIN 总线模式。

2. 设置 UA_LIN_BKFL(UA_ALT_CSR[3:0])寄存器来选择 break 域长度。（break 域的长度为UA_LIN_BKFL+2）
3. 将THR(UA_THR[7:0])设置为0x55，来请求发送同步域。
4. 通过写受保护的标识值到UA_THR中，来请求标识域传输。
5. 设置 LIN_TX_EN(UA_ALT_CSR[7]) 位来开始传输（当 break 操作完成后，LIN_TX_EN(UA_ALT_CSR[7])将会被自动清除）。
6. 当THR寄存器中最后一个字节中的STOP位发送到总线上后，硬件将会设置UA_FSR寄存器中的TE_FLAG位为1。
7. 填N个字节的数据和校验码到THR(UA_THR[7:0])，然后重复步骤5和步骤6来发送数据。

LIN 总结接收 (RX) 编程流程如下：

1. 设置 UA_FUN_SEL[1:0] 为 01 来使能 LIN 总线模式。
2. 设置 LIN_RX_EN(UA_ALT_CSR[6]) 来使能 LIN RX模式。
3. 等待标志 LIN_RX_BREAK_IF(UA_ISR[7]) 来检测RX是否接收到 break 域。
4. 等待标志RDA_IF(UA_ISR[0])并且从RBR(UA_RBR[7:0])寄存器读回数据。

5.13.6 NUC029xAN 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:				
UART0_BA = 0x4005_0000				
UART1_BA = 0x4015_0000				
UA_RBR x = 0, 1	UARTx_BA+0x00	R	UART接收缓存寄存器	未定义
UA_THR x = 0, 1	UARTx_BA+0x00	W	UART发送保持寄存器	未定义
UA_IER x = 0, 1	UARTx_BA+0x04	R/W	UART 中断使能寄存器	0x0000_0000
UA_FCR x = 0, 1	UARTx_BA+0x08	R/W	UART FIFO 控制寄存器	0x0000_0101
UA_LCR x = 0, 1	UARTx_BA+0x0C	R/W	UART Line 控制寄存器	0x0000_0000
UA_MCR x = 0, 1	UARTx_BA+0x10	R/W	UART Modem 控制寄存器	0x0000_0200
UA_MSR x = 0, 1	UARTx_BA+0x14	R/W	UART Modem 状态寄存器	0x0000_0110
UA_FSR x = 0, 1	UARTx_BA+0x18	R/W	UART FIFO 状态寄存器	0x1040_4000
UA_ISR x = 0, 1	UARTx_BA+0x1C	R/W	UART 中断状态寄存器	0x0000_0002
UA_TOR x = 0, 1	UARTx_BA+0x20	R/W	UART 超时寄存器	0x0000_0000
UA_BAUD x = 0, 1	UARTx_BA+0x24	R/W	UART 波特率分频寄存器	0x0F00_0000
UA_IRCR x = 0, 1	UARTx_BA+0x28	R/W	UART IrDA 控制寄存器	0x0000_0040
UA_ALT_CSR x = 0, 1	UARTx_BA+0x2C	R/W	UART 选择控制/状态寄存器	0x0000_0000
UA_FUN_SEL x = 0, 1	UARTx_BA+0x30	R/W	UART 功能选择寄存器	0x0000_0000

5.13.7 NUC029xAN 寄存器描述

UART 接收缓存寄存器 (UA_RBR)

寄存器	偏移	R/W	描述	复位值
UA_RBR x = 0, 1	UARTx_BA+0x00	R	UART 接收缓存寄存器	未定义

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
RBR							

位	描述	
[31:8]	保留	保留
[7:0]	RBR	接收缓存寄存器 (只读) 通过读此寄存器, UART 将返回一个从UART_RX引脚接收到的 8-位数据 (LSB优先)。

UART 发送保持寄存器 (UA_THR)

寄存器	偏移	R/W	描述	复位值
UA_THR x=0,1	UARTx_BA+0x00	W	UART 发送保持寄存器	未定义

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
THR							

位	描述	
[31:8]	保留	保留
[7:0]	THR	发送保持寄存器 通过写该寄存器， UART 将通过UART_TX引脚送出 8-位数据(LSB优先)。

UART 中断使能寄存器 (UA_IER)

寄存器	偏移	R/W	描述	复位值
UA_IER x=0,1	UARTx_BA+0x04	R/W	UART 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
保留		AUTO_CTS_EN	AUTO_RTS_EN	TIME_OUT_EN	保留		LIN_RX_BRK_IEN
7	6	5	4	3	2	1	0
保留	WAKE_EN	BUF_ERR_IEN	RTO_IEN	MODEM_IEN	RLS_IEN	THRE_IEN	RDA_IEN

位	描述	
[31:14]	保留	保留
[13]	AUTO_CTS_EN	<p>CTS 自动流控使能控制 0 = 禁用 CTS 自动流控 1 = 使能 CTS 自动流控 注: 当使能CTS自动流控时, UART将会在CTS输入有效时(直到CTS有效, UART才会发送数据)发送数据到外部设备。</p>
[12]	AUTO_RTS_EN	<p>RTS 自动流控使能控制 0 = 禁用 RTS 自动流控 1 = 使能 RTS 自动流控 注: 当 RTS 自动流控使能时, 如果RX FIFO中接收的字节数和RTS_TRI_LEVEL (UA_FCR [19:16])相等, UART 将再次使RTS信号失效。</p>
[11]	TIME_OUT_EN	<p>超时计数器使能控制 0 = 禁用超时计数器 1 = 使能超时计数器</p>
[10:9]	保留	保留
[8]	LIN_RX_BRK_IEN	<p>LIN RX Break 域侦测中断使能控制 0 = 禁用 LIN 总线 RX break 域中断 1 = 使能 LIN 总线 RX break 域中断 注: 该位供LIN功能模式使用</p>
[7]	保留	保留
[6]	WAKE_EN	UART 唤醒功能使能控制

		0 = 禁用 UART 唤醒功能 1 = 使能 UART 唤醒功能 注: 当芯片进入休眠模式时, 外部CTS信号的改变将会唤醒芯片。
[5]	BUF_ERR_IEN	缓存错误中断使能控制 0 = 禁用 BUF_ERR_INT 1 = 使能 BUF_ERR_INT
[4]	RTO_IEN	RX 超时中断使能控制 0 = 禁用 TOUT_INT 1 = 使能 TOUT_INT
[3]	MODEM_IEN	Modem 状态中断使能控制 0 = 禁用 MODEM_INT 1 = 使能 MODEM_INT
[2]	RLS_IEN	接收 Line 状态中断使能控制 0 = 禁用 RLS_INT 1 = 使能 RLS_INT
[1]	THRE_IEN	发送保持寄存器空中断使能控制 0 = 禁用 THRE_INT 1 = 使能 THRE_INT.
[0]	RDA_IEN	接收数据有效中断使能控制 0 = 禁用 RDA_INT 1 = 使能 RDA_INT

UART FIFO 控制寄存器 (UA_FCR)

寄存器	偏移	R/W	描述	复位值
UA_FCR x=0,1	UARTx_BA+0x08	R/W	UART FIFO 控制寄存器	0x0000_0101

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

位	描述	
[31:20]	保留	保留
[19:16]	RTS_TRI_LEV	RTS 触发级别 (用于自动流控) 0000 = RTS 触发级别为 1 byte. 0001 = RTS 触发级别为 4 bytes. 0010 = RTS 触发级别为 8 bytes. 0011 = RTS 触发级别为 14 bytes. 其它 = 保留 注: 该域用于RTS自动流控.
[15:9]	保留	保留
[8]	RX_DIS	禁止接收寄存器 接收器是否禁用 (设置1禁止接收) 1 = 禁止接收 0 = 使能接收 注: 该位用于RS-485 普通多点模式。 必须在设置RS-485_NMM (UA_ALT_CSR [8])之前被设置好。
[7:4]	RFITL	RX FIFO 中断 (RDA_INT) 触发级别 当接收FIFO中的字节数与RFITL匹配时, RDA_IF 将被置位 (如果UA_IER寄存器中的RDA_IEN被使能, 将发生中断)。 0000 = RX FIFO 中断触发级别为 1 byte. 0001 = RX FIFO 中断触发级别为 4 bytes. 0010 = RX FIFO 中断触发级别为 8 bytes. 0011 = RX FIFO 中断触发级别为 14 bytes.

		其它 = 保留
[3]	保留	保留
[2]	TFR	TX 域软件复位 当 TX_RST 置位, 发送 FIFO 中的所有字节和 TX 内部状态机都会被清零。 0 = 无效 1 = 复位 TX 内部状态机和指针 注: 该位自动清零需要至少3个UART时钟周期
[1]	RFR	RX 域软件复位 当 RX_RST 置位, 接收 FIFO 中所有字节和 RX 内部状态机都将被清零。 0 = 无效 1 = 复位 RX 内部状态机和指针 注: 该位自动清零需要至少3个UART时钟周期
[0]	保留	保留

UART Line 控制寄存器 (UA_LCR)

寄存器	偏移	R/W	描述	复位值
UA_LCR x=0,1	UARTx_BA+0x0C	R/W	UART Line 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	BCB	SPE	EPE	PBE	NSB	WLS	

位	描述	
[31:7]	保留	保留
[6]	BCB	<p>Break 控制位 当该位设置为逻辑1时，串行数据输出（TX）将被强制处于Spacing状态（逻辑0）。该位仅作用于 TX，对传输逻辑不起作用。 0 = 禁用 Break 控制 1 = 使能 Break 控制 </p>
[5]	SPE	<p>Stick 校验使能控制 0 = 禁止 Stick 校验 1 = 当 PBE(UA_LCR[3]) 和 EBE (UA_LCR[4]) 为逻辑1时，校验位发送且检验值为逻辑0；当 PBE(UA_LCR[3]) 为1， EBE(UA_LCR[4]) 为0时，校验位发送且检验值为逻辑1。 </p>
[4]	EPE	<p>偶校验使能控制 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节中被发送和检验 该位仅当 PBE (UA_LCR[3]) 置位时才有效 </p>
[3]	PBE	<p>校验位使能控制 0 = 无校验位 1 = 每一个发送字符中都产生奇偶校验位，对每一个传进来的数据进行奇偶校验位检测。 </p>
[2]	NSB	<p>“STOP 位”的个数 0 = 发送数据时，产生一个“STOP 位” 1 = 当发送数据，选择 5-位字长度时，产生 1.5 个“STOP 位”。当选择 6-, 7- 和 8-位字长度时，产生 2 个“STOP 位”。 </p>
[1:0]	WLS	字长度选择

		00 = 字长度为5位 01 = 字长度为6位 10 = 字长度为7位 11 = 字长度为8位
--	--	--

UART Modem 控制寄存器 (UA_MCR)

寄存器	偏移	R/W	描述	复位值
UA_MCR x=0,1	UARTx_BA+0x10	R/W	UART Modem 控制寄存器	0x0000_0200

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		RTS_ST	保留			LEV_RTS	保留
7	6	5	4	3	2	1	0
保留						RTS	保留

位	描述	
[31:14]	保留	保留
[13]	RTS_ST	<p>RTS 引脚状态 (只读) 该位反映出 RTS 引脚输出电压的逻辑状态。 0 = RTS 引脚输出低电平逻辑状态。 1 = RTS 引脚输出高电平逻辑状态。</p>
[12:10]	保留	保留
[9]	LEV_RTS	<p>RTS 引脚有效电平 该位定义了 RTS 引脚输出有效电平状态 0 = RTS 引脚输出高电平有效 1 = RTS 引脚输出低电平有效 注1: UART功能模式请参考 图 5-96 和 图 5-97。 注2: RS-485功能模式表参考 图 5-100 和 图 5-101。</p>
[8:2]	保留	保留
[1]	RTS	<p>RTS (Request-To-Send) 信号控制 该位直接控制内部RTS信号是否有效，然后驱动RTS引脚输出LEV_RTS定义的电平。 0 = RTS 信号有效 1 = RTS 信号无效 注1: UART模式下，如果RTS自动流控功能使能，该位无效。 注2: RS-485模式下，如果自动方向模式(AUD)使能，该位无效。</p>
[0]	保留	保留

UART Modem 状态寄存器 (UA_MSR)

寄存器	偏移	R/W	描述	复位值
UA_MSR x=0,1	UARTx_BA+0x14	R/W	UART Modem 状态寄存器	0x0000_0110

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							LEV_CTS
7	6	5	4	3	2	1	0
保留			CTS_ST	保留			DCTSF

位	描述	
[31:9]	保留	保留
[8]	LEV_CTS	<p>CTS 引脚有效电平 该位定义了CTS引脚输入的有效电平 0 = CTS 引脚输入高电平有效 1 = CTS 引脚输入高电平有效 注: 详情参考 图 5-95</p>
[7:5]	保留	保留
[4]	CTS_ST	<p>CTS 引脚状态 (只读) 该位反映了CTS引脚输入电平的逻辑状态。 0 = CTS 引脚输入低电平逻辑状态 1 = CTS 引脚输入高电平逻辑状态 注:当UART时钟使能，并且CTS多功能引脚被选择时，该位表示 CTS 引脚的状态。</p>
[3:1]	保留	保留
[0]	DCTSF	<p>检测 CTS 状态改变标志位 无论何时CTS输入管脚上有电平变化，该位将被置位，当MODEM_IEN (UA_IER [3])位被置位，将会产生Modem中断。 0 = CTS 输入管脚没有电平变化 1 = CTS 输入管脚有电平变化 注:该位写1清零。</p>

UART FIFO 状态寄存器 (UA_FSR)

寄存器	偏移	R/W	描述	复位值
UA_FSR x=0,1	UARTx_BA+0x18	R/W	UART FIFO 状态寄存器	0x1040_4000

31	30	29	28	27	26	25	24
保留			TE_FLAG	保留			TX_OVER_IF
23	22	21	20	19	18	17	16
TX_FULL	TX_EMPTY	TX_POINTER					
15	14	13	12	11	10	9	8
RX_FULL	RX_EMPTY	RX_POINTER					
7	6	5	4	3	2	1	0
保留	BIF	FEF	PEF	RS485_ADD_DETF	保留	RX_OVER_IF	

位	描述		
[31:29]	保留	保留	
[28]	TE_FLAG	发送器空标志 (只读) 当TX FIFO (UA_THR)为空，并且最后一个字节的STOP位也已经被发送完毕，那么该位将由硬件置1。 0 = TX FIFO 非空 1 = TX FIFO 为空 注：当TX FIFO 不为空或最后一个字节没有全部被发送完，此位自动清零。	
[27:25]	保留	保留	
[24]	TX_OVER_IF	TX 溢出错误中断标志 如果TX FIFO (UA_THR)满了，此时如果再向UA_THR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清零。	
[23]	TX_FULL	发送 FIFO 满 (只读) 该位表示TX FIFO是否已满。 0 = TX FIFO 未满 1 = TX FIFO 已满 注：当TX FIFO 中数据个数等于16时，该位将被置为1，否则由硬件清零。	
[22]	TX_EMPTY	发送 FIFO 空 (只读) 该位表示TX FIFO是否已空。 0 = TX FIFO 未空	

		<p>1 = TX FIFO 已空 注:当 TX FIFO 的最后一个字节传输到发送移位寄存器时, 硬件置位该位。当写数据到 THR (TX FIFO 非空) 时, 此位被清零。</p>
[21:16]	TX_POINTER	<p>TX FIFO 指针 (只读) 该位指示 TX FIFO 缓存区指针位置。当 CPU 写一个字节到 UA_THR 时, TX_POINTER 将加1。当 TX FIFO 发送一个字节到发送移位寄存器中, TX_POINTER 将减1。 TX_POINTER 的最大值是 15。当 TX FIFO 缓冲区所填充数据数量达到 16 时, TX_FULL 位将被置 1, 而且 TX_POINTER 将为零。此时, 如果 TX FIFO 中发送一个字节到发送移位寄存器, TX_FULL 位将被清零, TX_POINTER 将显示为 15。</p>
[15]	RX_FULL	<p>接收 FIFO 满 (只读) 该位表示 RX FIFO 是否已满。 0 = RX FIFO 未满 1 = RX FIFO 已满 注:当 RX FIFO 中数据个数等于 16 时, 该位将被置为 1, 否则由硬件清零。</p>
[14]	RX_EMPTY	<p>接收 FIFO 空 (只读) 该位表示 RX FIFO 是否已空。 0 = RX FIFO 非空。 1 = RX FIFO 已空。 注:当 RX FIFO 中最后一个字节被 CPU 读取后, 硬件将对此位置 1, 当 UART 接收到新数据后, 此位将被清零。</p>
[13:8]	RX_POINTER	<p>RX FIFO 指针 (只读) 该位指示 RX FIFO 缓存区指针位置。当 UART 从外部设备接收到一个字节, RX_POINTER 将累加 1。当 RX FIFO 的数据被 CPU 读取一个字节, RX_POINTER 将递减 1。 RX_POINTER 显示的最大值是 15。当 RX FIFO 的数据量达到 16 时, RX_FULL 位将被置 1, RX_POINTER 显示 0。此时, 如果 RX FIFO 当中的数据被 CPU 读取一个后, RX_FULL 将被清零, RX_POINTER 显示 15。</p>
[7]	保留	保留
[6]	BIF	<p>Break 中断标志 (只读) 每当接收到数据输入 (RX) 维持在 “spacing state” (logic 0) 的时间长于一个全字的传输时间 (即“start bit” + data 位 + parity + stop 位 的总时间), 该位置 1。 0 = 未产生 Break 中断 1 = 已产生 Break 中断 注:此位只读, 但是软件可以给其或 RFR (UA_FCR [1]) 写 1 清零。</p>
[5]	FEF	<p>帧错误标志 (只读) 每当接收到的字符没有有效的 “停止位” (即跟着最后的数据位或奇偶校验位的停止位检测为 0) 该位置 1。 0 = 未产生帧错误 1 = 已产生帧错误 注:此位只读, 但是软件可以给其或 RFR (UA_FCR [1]) 写 1 清零。</p>
[4]	PEF	<p>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时, 该位置 1。 0 = 未产生奇偶校验错误 1 = 已产生奇偶校验错误 注:此位只读, 但是软件可以给其或 RFR (UA_FCR [1]) 写 1 清零。</p>

[3]	RS485_ADD_DETF	RS-485 地址字节检测标志 当设置RS485_ADD_EN (UA_ALT_CSR[15])为1使能地址检测模式，且接收到带地址位(bit 9 =1)的数据时，该位将被置1。 注1: 此域适用于RS-485功能模式。 注2: 该位写1清零。
[2:1]	保留	保留
[0]	RX_OVER_IF	RX 溢出错误中断标志 位当RX FIFO溢出时被置1。 如果接收到的数据字节数量大于RX_FIFO (UA_RBR)的大小（16个字节），该位将被置位。 0 = RX FIFO 未溢出 1 = RX FIFO 已溢出 注: 该位写1清零。

UART 中断状态控制寄存器 (UA_ISR)

寄存器	偏移	R/W	描述	复位值
UA_ISR x=0,1	UARTx_BA+0x1C	R/W	UART 中断状态控制寄存器	0x0000_0002

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
LIN_RX_BRE AK_INT	保留	BUF_ERR_IN T	TOUT_INT	MODEM_INT	RLS_INT	THRE_INT	RDA_INT
7	6	5	4	3	2	1	0
LIN_RX_BRE AK_IF	保留	BUF_ERR_IF	TOUT_IF	MODEM_IF	RLS_IF	THRE_IF	RDA_IF

位	描述	
[31:14]	保留	保留
[15]	LIN_RX_BREAK_I NT	LIN 总线RX Break域侦测中断标志 (只读) 如果 LIN_RX_BRK_IEN 和 LIN_RX_BREAK_IF 都被设为1，则该位为1。 0 = 未产生LIN RX Break 中断 1 = 已产生LIN RX Break 中断
[14]	保留	保留
[13]	BUF_ERR_INT	缓存错误中断标志 (只读) 如果BUF_ERR_IEN 和 BUF_ERR_IF都被设为1，则该位为1。 0 = 未产生缓存错误中断 1 = 已产生缓存错误中断
[12]	TOUT_INT	定时器溢出中断标志 (只读) 如果RTO_IEN 和 TOUT_IF都被设为1，则该位为1。 0 = 未产生定时器溢出中断 1 = 已产生定时器溢出中断
[11]	MODEM_INT	MODEM 状态中断标志 (只读) 如果MODEM_IEN 和 MODEM_IF都被设为1，则该位为1。 0 = 未产生 Modem 中断 1 = 已产生 Modem 中断
[10]	RLS_INT	接收 Line 状态中断标志 (只读) 如果 RLS_IEN 和 RLS_IF都被设为1，则该位为1。 0 = 未产生 RLS 中断

		1 = 已产生 RLS 中断
[9]	THRE_INT	<p>发送保持寄存器空中断标志（只读） 如果 THRE_IEN 和 THRE_IF都被设为1，则该位为1。 0 = 未产生 THRE 中断 1 = 已产生 THRE 中断</p>
[8]	RDA_INT	<p>接收数据有效中断标志（只读） 如果 RDA_IEN 和 RDA_IF都被设为1，则该位为1。 0 = 未产生 RDA 中断 1 = 已产生 RDA 中断</p>
[7]	LIN_RX_BREAK_IF	<p>LIN 总线 RX Break 域侦测标志 当 RX 接收到 LIN Break 域时，该位将被置位。 如果LIN_RX_BRK_IEN (UA_IER [8]) 被使能， LIN RX Break 域中断将发生。 0 = 未收到 LIN RX Break 1 = 已收到 LIN RX Break 注: 该位写1清零。</p>
[6]	保留	保留
[5]	BUF_ERR_IF	<p>缓存错误中断标志（只读） 当 TX/RX FIFO溢出标志(TX_OVER_IF 或者 RX_OVER_IF)被置位时，该位被置位。 当 BUF_ERR_IF 被置位时，代表传输不正确。 如果BUF_ERR_IEN (UA_IER [5])被使能， 缓存错误中断将发生。 0 = 未产生缓存错误中断标志 1 = 已产生缓存错误中断标志 注: 该位为只读，当TX_OVER_IF 和 RX_OVER_IF都清零时，该位恢复为0。</p>
[4]	TOUT_IF	<p>定时器溢出中断标志（只读） 当 RX FIFO非空，RX FIFO处于非活动状态，且超时计数器计数到TOIC 寄存器的值时，该位置位。 若RTO_IEN (UA_IER [4])使能， 将产生 超时中断。 0 = 未产生定时器溢出中断标志 1 = 已产生定时器溢出中断标志 注: 该位为只读位，用户可以通过读UA_RBR (RX处于活动状态) 来清零。</p>
[3]	MODEM_IF	<p>MODEM 中断标志（只读） 当 CTS引脚状态(DCTSF=1)改变，该位置位。若UA_IER [MODEM_IEN]使能， 产生 Modem中断。 0 = 未产生 Modem 中断标志 1 = 已产生 Modem 中断标志 注:该位为只读位，当DCTSF清零时（写1到DCTSF清零），该位恢复为0。</p>
[2]	RLS_IF	<p>接收 Line中断标志（只读） 当 RX 接收数据有奇偶校验错误、帧错误、Break错误时（至少3个bit中的一个， BIF, FEF 和 PEF， 置位），该位置位。若RLS_IEN (UA_IER [2]) 使能， 将产生RLS 中断。 0 = 未产生 RLS 中断标志 1 = 已产生 RLS 中断标志 注1: 在RS-485功能模式下，该位被置位的情形包含“接收器检测和接收到地址字节字符（bit 9 = 1）位”。同时RS485_ADD_DETF (UA_FSR[3])也会被置位。</p>

		<p>注2: 该位是只读的, 当BIF、FEF、PEF 和 RS485_ADD_DETF 都被清零时, 该位将被清零。</p>
[1]	THRE_IF	<p>发送保持寄存器空中断标志 (只读) 当TX FIFO 的最后一个数据发送到发送器移位寄存器上时, 该位置位。如果THRE_IEN (UA_IER [1])使能, 将发生THRE中断。 0 = 未产生 THRE 中断标志 1 = 已产生 THRE 中断标志 注: 该位只读, 写数据到THR 清零该位 (TX FIFO 非空)。</p>
[0]	RDA_IF	<p>接收数据有效中断标志 (只读) 当RX FIFO 中的字节数等于RFITL 时, 那么 RDA_IF 被置位。如果使能RDA_IEN (UA_IER [0]), 将产生RDA 中断。 0 = 未产生 RDA 中断标志 1 = 已产生 RDA 中断标志 注: 该位只读, 当RX FIFO 中未读取字节数少于阀值 (RFITL) 时, 该位清零。</p>

UART 定时器溢出寄存器 (UA_TOR)

寄存器	偏移	R/W	描述	复位值
UA_TOR x=0,1	UARTx_BA+0x20	R/W	UART 定时器溢出寄存器	0x0000_0000



位	描述	
[31:16]	保留	保留
[15:8]	DLY	<p>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。</p> 
[7:0]	TOIC	<p>定时器溢出中断比较器 每当 RX FIFO 接收到新数据后，超时计数器复位并开始计数(计数时钟频率=波特率)。一旦超时计数器 (TOUT_CNT) 的值和超时比较器 (TOIC) 相等，且 RTO_IEN (UA_IER [4]) 使能，接收超时中断将产生 (TOUT_INT)。一个新的输入数据或 RX FIFO 为空将清除 TOUT_INT。为了避免收到一个字节马上发生超时中断，TOIC 的值应该设为 40 到 255 之间。例如：如果 TOIC 设为 40，当 UART 传输设置为 1 停止位和无奇偶校验位时，定时溢出中断将在 4 个字符没有收到之后产生。</p>

UART 波特率分频寄存器(UA_BAUD)

寄存器	偏移	R/W	描述	复位值
UA_BAUD x=0,1	UARTx_BA+0x24	R/W	UART波特率分频寄存器	0x0F00_0000



位	描述	
[31:30]	保留	保留
[29]	DIV_X_EN	分频 X 使能控制 BRD = 波特率分频值， 波特率方程如下： 波特率 = Clock / [M * (BRD + 2)]; M的默认值为16。 0 = 禁用分频器 X (M=16) 1 = 使能分频器 X (M = X+1, 但是 DIVIDER_X[27:24] 必须 >= 8) 注1: 详情参考 5.13.5.1 章节 注2: 在IrDA 模式下 该位须禁用
[28]	DIV_X_ONE	分频系数X等于1 0 = 分频系数 M = X (M = X+1, 但是 DIVIDER_X[27:24] 必须 >= 8). 1 = 分频系数 M = 1 (M = 1, 但是 BRD [15:0] 必须>= 8). Note1: 详情参考 5.13.5.1 章节。
[27:24]	DIVIDER_X	分频器 X 波特率分频 M = X+1
[23:16]	保留	保留
[15:0]	BRD	波特率分频器 这些位表示波特率分频值。

UART IrDA 控制寄存器 (UART_IRCR)

寄存器	偏移	R/W	描述	复位值
UA_IRCR x=0,1	UARTx_BA+0x28	R/W	UART IrDA控制寄存器	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							
7	6	5	4	3	2	1	0
保留	INV_RX	INV_TX	保留			TX_SELECT	保留

位	描述	
[31:7]	保留	保留
[6]	INV_RX	反转 RX 输入控制 0 = 不反转 1 = 反转RX输入信号
[5]	INV_TX	反转 TX 输出控制 0 = 不反转 1 = 反转TX输出信号
[4:2]	保留	保留
[1]	TX_SELECT	IrDA 接收器使能控制 0 = 使能 IrDA 接收器 1 = 使能 IrDA 发送器
[0]	保留	保留

UART选择控制/状态寄存器 (UA_ALT_CSR)

寄存器	偏移	R/W	描述	复位值
UA_ALT_CSR x=0,1	UARTx_BA+0x2C	R/W	UART选择控制/状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_MATCH							
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
RS485_ADD_EN	保留				RS485_AUD	RS485_AAD	RS485_NMM
7	6	5	4	3	2	1	0
LIN_TX_EN	LIN_RX_EN	保留		UA_LIN_BKFL			

位	描述		
[31:24]	ADDR_MATCH	地址匹配值寄存器 该域包含RS-485 的地址匹配值。 注: 该域用于RS-485自动地址识别模式	
[23:16]	保留	保留	
[15]	RS485_ADD_EN	RS485 地址识别使能控制位 该位用于使能RS485地址识别模式。 0 = 禁用RS485地址识别模式 1 = 使能RS485地址识别模式 注: 该域用于RS-485的所有模式	
[14:11]	保留	保留	
[10]	RS485_AUD	RS485 自动方向模式 (AUD) 控制位 0 = 禁用 RS-485 自动方向操作模式(AUO) 1 = 使能 RS-485 自动方向操作模式(AUO) 注: RS485_AAD 或 RS485_NMM 操作模式下有效.	
[9]	RS485_AAD	RS485 自动地址识别操作模式 (AAD) 控制位 0 = 禁用 RS-485 自动地址识别操作模式(AAD) 1 = 使能 RS-485 自动地址识别操作模式(AAD) 注: RS-485_NMM 操作模式下该位无效	
[8]	RS485_NMM	RS485 普通多点操作模式 (NMM) 控制位 0 = 禁用 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RS485_AAD 操作模式下无效.
[7]	LIN_TX_EN	LIN TX Break 模式使能控制位 0 = 禁止 LIN TX Break 模式 1 = 使能 LIN TX Break 模式 注: 当TX break 域传输操作完成时, 该位将被自动清0。
[6]	LIN_RX_EN	LIN RX 使能控制位 0 = 禁止 LIN RX 模式 1 = 使能 LIN RX 模式
[5:4]	保留	保留
[3:0]	UA_LIN_BKFL	UART LIN Break 域长度 该区域的数值表示了一个4位LIN TX break域的长度。 注: break 域的长度为 UA_LIN_BKFL + 2。

UART 功能选择寄存器 (UA_FUN_SEL)

寄存器	偏移	R/W	描述	复位值
UA_FUN_SEL x=0,1	UARTx_BA+0x30	R/W	UART功能选择寄存器	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						FUN_SEL	

位	描述	
[31:2]	保留	保留
[1:0]	FUN_SEL	<p>功能选择</p> <p>UART控制器功能模式选择:</p> <p>00 = UART功能模式</p> <p>01 = LIN功能模式</p> <p>10 = IrDA功能模式</p> <p>11 = RS-485功能模式</p>

5.13.8 NUC029FAE 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:				
UART_BA = 0x4005_0000				
UA_RBR	UART_BA+0x00	R	UART接收缓存寄存器	未定义
UA_THR	UART_BA+0x00	W	UART发送保持寄存器	未定义
UA_IER	UART_BA+0x04	R/W	UART 中断使能控制寄存器	0x0000_0000
UA_FCR	UART_BA+0x08	R/W	UART FIFO 控制寄存器	0x0000_0101
UA_LCR	UART_BA+0x0C	R/W	UART Line 控制寄存器	0x0000_0000
UA_MCR	UART_BA+0x10	R/W	UART Modem 控制寄存器	0x0000_0200
UA_FSR	UART_BA+0x18	R/W	UART FIFO 状态寄存器	0x1040_4000
UA_ISR	UART_BA+0x1C	R/W	UART 中断状态寄存器	0x0000_0002
UA_TOR	UART_BA+0x20	R/W	UART 超时寄存器	0x0000_0000
UA_BAUD	UART_BA+0x24	R/W	UART 波特率分频寄存器	0x0F00_0000
UA_IRCR	UART_BA+0x28	R/W	UART IrDA 控制寄存器	0x0000_0040
UA_ALT_CSR	UART_BA+0x2C	R/W	UART 选择控制/状态寄存器	0x0000_0000
UA_FUN_SEL	UART_BA+0x30	R/W	UART 功能选择寄存器	0x0000_0000

5.13.9 NUC029FAE 寄存器描述

接收缓存寄存器(UA_RBR)

寄存器	偏移	R/W	描述	复位值
UA_RBR	UART_BA+0x00	R	UART接收缓存寄存器	未定义

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
RBR							

位	描述	
[31:8]	保留	保留
[7:0]	RBR[7:0]	接收缓存寄存器（只读） 通过读此寄存器，UART 将返回一个从RX引脚接收到的 8-位数据 (LSB优先)。

发送保持寄存器T (UA_THR)

寄存器	偏移	R/W	描述	复位值
UA_THR	UART_BA+0x00	W	UART 发送保持寄存器	未定义

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
THR							

位	描述	
[31:8]	保留	保留
[7:0]	THR	发送保持寄存器 通过写该寄存器, UART 将通过 TX 引脚送出 8-位数据(LSB优先)。

中断使能控制寄存器(UA_IER)

寄存器	偏移	R/W	描述	复位值
UA_IER	UART_BA+0x04	R/W	UART中断使能控制寄存器	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				TIME_OUT_EN	保留		
7	6	5	4	3	2	1	0
保留		BUF_ERR_IEN	RTO_IEN	MODEM_IEN	RLS_IEN	THRE_IEN	RDA_IEN

位	描述	
[31:12]	保留	保留
[11]	TIME_OUT_EN	超时计数器使能控制 0 = 禁用超时计数器 1 = 使能超时计数器
[10:6]	保留	保留
[5]	BUF_ERR_IEN	缓存错误中断使能控制 0 = 禁用 INT_BUF_ERR 1 = 使能 INT_BUF_ERR
[4]	RTO_IEN	RX 超时中断使能控制 0 = 屏蔽TOUT_INT 1 = 使能 TOUT_INT
[3]	MODEM_IEN	Modem 状态中断使能控制 0 = 屏蔽 MODEM_INT 1 = 使能 MODEM_INT
[2]	RLS_IEN	接收 Line 状态中断使能控制 0 = 屏蔽 RLS_INT 1 = 使能 RLS_INT
[1]	THRE_IEN	发送保持寄存器空中断使能控制 0 = 屏蔽 THRE_INT 1 = 使能 THRE_INT

位	描述	
[0]	RDA_IEN	接收数据有效中断使能控制 0 =屏蔽 RDA_INT 1 =使能 RDA_INT

FIFO控制寄存器(UA_FCR)

寄存器	偏移	R/W	描述	复位值
UA_FCR	UART_BA+0x08	R/W	UART FIFO 控制寄存器	0x0000_0101

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
RFITL				保留	TFR	RFR	保留

位	描述	
[31:16]	保留	保留
[15:9]	保留	保留
[8]	RX_DIS	<p>禁止接收控制 接收器是否禁用 (设置1禁止接收) 0 = 使能接收 1 = 禁止接收</p> <p>注1: 该位仅用于RS-485 普通多点模式。 必须在设置RS-485_NMM (UA_ALT_CSR [8])之前被设置好，以避免接收到未知数据。</p> <p>注2: 在RS-485 普通多点模式下，RS-485 接收到一个地址字节后，该位 (RX_DIS) 由硬件清零。</p>
[7:4]	RFITL[3:0]	<p>RX FIFO 中断 (RDA_INT) 触发级别 当接收FIFO中的字节数与RFITL匹配时，RDA_IF 将被置位 (如果UA_IER寄存器中的RDA_IEN被使能，将发生中断)。</p> <p>0000 = RX FIFO 中断触发级别为 1 byte. 0001 = RX FIFO 中断触发级别为 4 bytes. 0010 = RX FIFO 中断触发级别为 8 bytes. 0011 = RX FIFO 中断触发级别为 14 bytes. 其它 = 保留</p>
[3]	保留	保留
[2]	TFR	<p>TX 域软件复位 当 TX_RST 置位，发送 FIFO 中的所有字节和TX内部状态机都会被清零。 0 = 无效 1 = 复位 TX 内部状态机和指针</p>

位	描述
	注: 该位自动清零需要至少3个UART时钟周期
[1]	RFR RX 域软件复位 当 RX_RST 置位, 接收 FIFO 中所有字节和RX内部状态机都将被清零。 0 = 无效 1 =复位 RX 内部状态机和指针 注: 该位自动清零需要至少3个UART时钟周期
[0]	保留

Line 控制寄存器 (UA_LCR)

寄存器	偏移	R/W	描述	复位值
UA_LCR	UART_BA+0x0C	R/W	UART Line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	BCB	SPE	EPE	PBE	NSB	WLS	

位	描述	
[31:7]	保留	保留
[6]	BCB	Break 控制位 当该位设置为逻辑1时，串行数据输出（TX）将被强制处于Spacing状态（逻辑0）。该位仅作用于TX，对传输逻辑不起作用。 0 = 禁用 Break 控制 1 = 使能 Break 控制
[5]	SPE	Stick 校验使能控制 0 = 禁止 Stick 校验 1 = 当 PBE(UA_LCR[3]) 和 EBE (UA_LCR[4]) 为逻辑1时，校验位发送且检验值为逻辑0；当 PBE(UA_LCR[3]) 为1， EBE(UA_LCR[4]) 为0时，校验位发送且检验值为逻辑1。
[4]	EPE	偶校验使能控制 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节中被发送和检验 该位仅当PBE (UA_LCR[3])置位时才有效。
[3]	PBE	校验位使能控制 0 = 无校验位 1 = 每一个发送字符中都产生奇偶校验位，对每一个传进来的数据进行奇偶校验位检测。
[2]	NSB	“STOP 位”的个数 0 = 发送数据时，产生一个“STOP 位” 1 = 当发送数据，选择5位字长度时，产生1.5个“STOP 位”。当选择6-, 7- 和 8-位字长度时，产生2个“STOP 位”。
[1:0]	WLS[1:0]	字长度选择 00 = 字长度为5位

位	描述
	01 = 字长度为6位
	10 = 字长度为7位
	11 = 字长度为8位

MODEM 控制寄存器(UA_MCR)

寄存器	偏移	R/W	描述	复位值
UA_MCR	UART_BA+0x10	R/W	UART Modem 控制寄存器	0x0000_0200

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		RTS_ST	保留			LEV_RTS	保留
7	6	5	4	3	2	1	0
保留						RTS	保留

位	描述	
[31:14]	保留	保留
[13]	RTS_ST	RTS 引脚状态 (只读) 该位反映出 RTS 引脚输出电压的逻辑状态。 0 = RTS 引脚输出低电平逻辑状态。 1 = RTS 引脚输出高电平逻辑状态。
[12:10]	保留	保留
[9]	LEV_RTS	RTS 引脚有效电平 该位定义了RTS引脚输出有效电平状态 0 = RTS 引脚输出高电平有效 1 = RTS 引脚输出低电平有效 注1: 参考 UART 功能模式。 注2: 参考 RS-485 功能模式。
[8:2]	保留	保留
[1]	RTS	RTS (Request-To-Send) 信号控制 该位直接控制内部RTS信号是否有效，然后驱动RTS引脚输出LEV_RTS定义的电平。 0 = RTS 信号有效 1 = RTS 信号无效
[0]	保留	保留

FIFO 状态寄存器(UA_FSR)

寄存器	偏移	R/W	描述				复位值
UA_FSR	UART_BA+0x18	R/W	UART FIFO 状态寄存器				0x1040_4000

31	30	29	28	27	26	25	24
保留			TE_FLAG	保留			TX_OVER_IF
23	22	21	20	19	18	17	16
TX_FULL	TX_EMPTY	TX_POINTER					
15	14	13	12	11	10	9	8
RX_FULL	RX_EMPTY	RX_POINTER					
7	6	5	4	3	2	1	0
保留	BIF	FEF	PEF	RS-485_ADD_DETF	保留		RX_OVER_IF

位	描述	
[31:29]	保留	保留
[28]	TE_FLAG	<p>发送器空标志（只读） 当TX FIFO (UA_THR)为空，并且最后一个字节的STOP位也已经被发送完毕，那么该位将由硬件置1。 0 = TX FIFO 非空 1 = TX FIFO 为空 注：当TX FIFO 不为空或最后一个字节没有全部被发送完，此位自动清零。</p>
[27:25]	保留	保留
[24]	TX_OVER_IF	<p>TX 溢出错误中断标志 如果TX FIFO (UA_THR)满了，此时如果再向UA_THR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清零。</p>
[23]	TX_FULL	<p>发送 FIFO 满（只读） 该位表示TX FIFO是否已满。 0 = TX FIFO 未满 1 = TX FIFO 已满 注：当TX FIFO 中数据个数等于16时，该位将被置为1，否则由硬件清零。</p>
[22]	TX_EMPTY	<p>发送 FIFO 空（只读） 该位表示TX FIFO是否已空。 0 = TX FIFO 未空 1 = TX FIFO 已空</p>

位	描述
	<p>注:当 TX FIFO 的最后一个字节传输到发送移位寄存器时, 硬件置位该位。当写数据到 THR (TX FIFO 非空) 时, 此位被清零。</p>
[21:16]	<p>TX_POINTER [5:0]</p> <p>TX FIFO 指针 (只读)</p> <p>该位指示 TX FIFO 缓存区指针位置。当 CPU 写一个字节到 UA_THR 时, TX_POINTER 将加1。当 TX FIFO 发送一个字节到发送移位寄存器中, TX_POINTER 将减1。</p> <p>TX_POINTER 的最大值是15。当 TX FIFO 缓冲区所填充数据数量达到16时, TX_FULL 位将被置1, 而且 TX_POINTER 将为零。此时, 如果 TX FIFO 中发送一个字节到发送移位寄存器, TX_FULL 位将被清零, TX_POINTER 将显示为15。</p>
[15]	<p>RX_FULL</p> <p>接收 FIFO 满 (只读)</p> <p>该位表示 RX FIFO 是否已满。</p> <p>0 = RX FIFO 未满 1 = RX FIFO 已满</p> <p>注:当 RX FIFO 中数据个数等于16时, 该位将被置为1, 否则由硬件清零。</p>
[14]	<p>RX_EMPTY</p> <p>接收 FIFO 空 (只读)</p> <p>该位表示 RX FIFO 是否已空。</p> <p>0 = RX FIFO 非空。 1 = RX FIFO 已空。</p> <p>注:当 RX FIFO 中最后一个字节被 CPU 读取后, 硬件将对此位置1, 当 UART 接收到新数据后, 此位将被清零。</p>
[13:8]	<p>RX_POINTER [5:0]</p> <p>RX FIFO 指针 (只读)</p> <p>该位指示 RX FIFO 缓存区指针位置。当 UART 从外部设备接收到一个字节, RX_POINTER 将累加1。当 RX FIFO 的数据被 CPU 读取一个字节, RX_POINTER 将递减1。</p> <p>RX_POINTER 显示的最大值是15。当 RX FIFO 的数据量达到16时, RX_FULL 位将被置1, RX_POINTER 显示0。此时, 如果 RX FIFO 当中的数据被 CPU 读取一个后, RX_FULL 将被清零, RX_POINTER 显示15。</p>
[7]	保留
[6]	<p>BIF</p> <p>Break 中断标志 (只读)</p> <p>每当接收到数据输入 (RX) 维持在 “spacing state” (logic 0) 的时间长于一个全字的传输时间 (即“start bit” + data 位 + parity + stop 位 的总时间), 该位置 1。</p> <p>0 = 未产生 Break 中断 1 = 已产生 Break 中断</p> <p>注:此位只读, 但是软件可以给其写1清零。</p>
[5]	<p>FEF</p> <p>帧错误标志 (只读)</p> <p>每当接收到的字符没有有效的 “停止位” (即跟着最后的数据位或奇偶校验位的停止位检测为0) 该位置 1。</p> <p>0 = 未产生帧错误 1 = 已产生帧错误</p> <p>注:此位只读, 但是软件可以给其写1清零。</p>
[4]	<p>PEF</p> <p>奇偶校验错误标志 (只读)</p> <p>每当接收到的字符没有有效的奇偶校验位时, 该位置 1。</p> <p>0 = 未产生奇偶校验错误 1 = 已产生奇偶校验错误</p>

位	描述
	注: 此位只读, 但是软件可以给其写1清零。
[3]	RS-485 地址字节检测标志 当设置RS485_ADD_EN (UA_ALT_CSR[15])为1使能地址检测模式, 且接收到带地址位(bit 9 =1)的数据时, 该位将被置1。 注1: 此域适用于RS-485功能模式。 注2: 该位写1清零。
[2:1]	保留
[0]	RX 溢出错误中断标志 位当RX FIFO溢出时被置1。 如果接收到的数据字节数量大于RX_FIFO (UA_RBR)的大小 (16个字节), 该位将被置位。 0 = RX FIFO 未溢出 1 = RX FIFO 已溢出 注: 该位写1清零。

中断状态寄存器 (UA_ISR)

寄存器	偏移	R/W	描述	复位值
UA_ISR	UART_BA+0x1C	R/W	UART 中断状态寄存器	0x0000_0002

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		BUF_ERR_IN_T	TOUT_INT	保留	RLS_INT	THRE_INT	RDA_INT
7	6	5	4	3	2	1	0
保留		BUF_ERR_IF	TOUT_IF	保留	RLS_IF	THRE_IF	RDA_IF

位	描述	
[31:14]	保留	保留
[13]	BUF_ERR_INT	缓存错误中断标志 (只读) 如果BUF_ERR_IEN 和 BUF_ERR_IF都被设为1，则该位为1。 0 = 未产生缓存错误中断 1 = 已产生缓存错误中断
[12]	TOUT_INT	定时器溢出中断标志 (只读) 如果RTO_IEN 和 TOUT_IF都被设为1，则该位为1。 0 = 未产生定时器溢出中断 1 = 已产生定时器溢出中断
[11]	保留	保留
[10]	RLS_INT	接收 Line 状态中断标志 (只读) 如果 RLS_IEN 和 RLS_IF都被设为1，则该位为1。 0 = 未产生 RLS 中断 1 = 已产生 RLS 中断
[9]	THRE_INT	发送保持寄存器空中断标志 (只读) 如果 THRE_IEN 和 THRE_IF都被设为1，则该位为1。 0 = 未产生 THRE 中断 1 = 已产生 THRE 中断
[8]	RDA_INT	接收数据有效中断标志 (只读) 如果 RDA_IEN 和 RDA_IF都被设为1，则该位为1。 0 = 未产生 RDA 中断

位	描述	
	1 = 已产生 RDA 中断	
[7:6]	保留	保留
[5]	BUF_ERR_IF	<p>缓存错误中断标志（只读） 当 TX/RX FIFO溢出标志(TX_OVER_IF 或者 RX_OVER_IF)被置位时，该位被置位。当 BUF_ERR_IF 被置位时，代表传输不正确。如果BUF_ERR_IEN (UA_IER [5])被使能，缓存错误中断将发生。</p> <p>0 = 未产生缓存错误中断标志 1 = 已产生缓存错误中断标志</p> <p>注: 该位为只读，当TX_OVER_IF 和 RX_OVER_IF都清零时，该位恢复为0。</p>
[4]	TOUT_IF	<p>定时器溢出中断标志（只读） 当 RX FIFO非空，RX FIFO处于非活动状态，且超时计数器计数到TOIC 寄存器的值时，该位置位。若RTO_IEN (UA_IER [4])使能，将产生超时中断。</p> <p>0 = 未产生定时器溢出中断标志 1 = 已产生定时器溢出中断标志</p> <p>注: 该位为只读位，用户可以通过读UA_RBR (RX处于活动状态) 来清零。</p>
[3]	保留	保留
[2]	RLS_IF	<p>接收 Line中断标志（只读） 当 RX 接收数据有奇偶校验错误、帧错误、Break错误时（至少3个bit中的一个，BIF， FEF 和 PEF， 置位），该位置位。若RLS_IEN (UA_IER [2]) 使能，将产生RLS 中断。</p> <p>0 = 未产生 RLS 中断标志 1 = 已产生 RLS 中断标志</p> <p>注1: 在RS-485功能模式下，该位被置位的情形包含“接收器检测和接收到地址字节字符 (bit 9 = 1) 位”。同时RS485_ADD_DETF (UA_FSR[3])也会被置位。</p> <p>注2: 该位是只读的，当BIF、FEF 、 PEF 和 RS485_ADD_DETF 都被清零时，该位将被清零。</p>
[1]	THRE_IF	<p>发送保持寄存器空中断标志（只读） 当TX FIFO 的最后一个数据发送到发送器移位寄存器上时，该位置位。如果THRE_IEN (UA_IER [1])使能，将发生THRE中断。</p> <p>0 = 未产生 THRE 中断标志 1 = 已产生 THRE 中断标志</p> <p>注: 该位只读，写数据到THR 清零该位 (TX FIFO 非空)。</p>
[0]	RDA_IF	<p>接收数据有效中断标志（只读） 当RX FIFO 中的字节数等于RFITL 时，那么 RDA_IF 被置位。如果使能RDA_IEN (UA_IER [0])，将产生RDA 中断。</p> <p>0 = 未产生 RDA 中断标志 1 = 已产生 RDA 中断标志</p> <p>注: 该位只读，当RX FIFO 中未读取字节数少于阀值 (RFITL) 时，该位清零。</p>

超时寄存器(UA_TOR)

寄存器	偏移	R/W	描述	复位值
UA_TOR	UART_BA+0x20	R/W	UART 超时寄存器	0x0000_0000



位	描述	
[31:16]	保留	保留
[15:8]	DLY[7:0]	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[7:0]	定时器溢出中断比较器 每当 RX FIFO 接收到新数据后，超时计数器复位并开始计数(计数时钟频率=波特率)。一旦超时计数器 (TOUT_CNT) 的值和超时比较器 (TOIC) 相等，且 RTO_IEN (UA_IER [4]) 使能，接收超时中断将产生 (TOUT_INT)。一个新的输入数据或 RX FIFO 为空将清除 TOUT_INT。为了避免收到一个字节马上发生超时中断，TOIC 的值应该设为 40 到 255 之间。例如：如果 TOIC 设为 40，当 UART 传输设置为 1 停止位和无奇偶校验位时，定时溢出中断将在 4 个字符没有收到之后产生。

波特率分频寄存器(UA_BAUD)

寄存器	偏移	R/W	描述	复位值
UA_BAUD	UART_BA+0x24	R/W	UART波特率分频寄存器	0x0F00_0000



位	描述	
[31:30]	保留	保留
[29]	DIV_X_EN	<p>分频 X 使能控制 BRD = 波特率分频值, 波特率方程如下: 波特率 = Clock / [M * (BRD + 2)]; M的默认值为16。 0 = 禁用分频器 X (M=16) 1 = 使能分频器 X (M = X+1, 但是 DIVIDER_X[27:24] 必须 >= 8) Note: 在IrDA 模式下 该位须禁用 </p>
[28]	DIV_X_ONE	<p>分频系数X等于1 0 = 分频系数 M = X (M = X+1, 但是 DIVIDER_X[27:24] 必须 >= 8). 1 = 分频系数 M = 1 (M = 1, 但是 BRD [15:0] 必须>= 8). Note: 详情参考 5.13.5.1 章节。 </p>
[27:24]	DIVIDER_X[3:0]	分频器 X 波特率分频 M = X+1.
[23:16]	保留	保留
[15:0]	BRD[15:0]	波特率分频器 这些位表示波特率分频值。

IrDA 控制寄存器(IRCR)

寄存器	偏移	R/W	描述	复位值
UA_IRCR	UART_BA+0x28	R/W	UART IrDA 控制寄存器	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							
7	6	5	4	3	2	1	0
保留	INV_RX	INV_TX	保留			TX_SELECT	保留

位	描述	
[31:7]	保留	保留
[6]	INV_RX	反转 RX 输入控制 0 = 不反转 1 = 反转RX输入信号
[5]	INV_TX	反转 TX 输出控制 0 = 不反转 1 = 反转TX输出信号
[4:2]	保留	保留
[1]	TX_SELECT	IrDA 接收器使能控制 0 = 使能 IrDA 接收器 1 = 使能 IrDA 发送器
[0]	保留	保留

UART 选择控制/状态寄存器(UA_ALT_CSR)

寄存器	偏移	R/W	描述	复位值
UA_ALT_CSR	UART_BA+0x2C	R/W	UART 选择控制/状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_MATCH							
23	22	21	20	19	18	17	16
保留							
15	14	13	12	11	10	9	8
RS485_ADD_EN	保留				RS485_AUD	RS485_AAD	RS485_NMM
7	6	5	4	3	2	1	0
保留							

位	描述	
[31:24]	ADDR_MATCH [7:0]	地址匹配值寄存器 该域包含RS-485 的地址匹配值。 注: 该域用于RS-485自动地址识别模式
[23:16]	保留	保留
[15]	RS485_ADD_EN	RS485 地址识别使能控制位 该位用于使能RS485地址识别模式。 0 = 禁用RS-485地址识别模式 1 = 使能RS-485地址识别模式 注: 该域用于RS-485的所有模式.
[14:11]	保留	保留
[10]	RS485_AUD	RS485 自动方向模式 (AUD) 控制位 0 = 禁用 RS-485 自动方向操作模式(AUO) 1 = 使能 RS-485 自动方向操作模式(AUO) 注: RS485_AAD 或 RS485_NMM 操作模式下有效.
[9]	RS485_AAD	RS485 自动地址识别操作模式 (AAD) 控制位 0 = 禁用 RS-485 自动地址识别操作模式(AAD) 1 = 使能 RS-485 自动地址识别操作模式(AAD) 注: RS-485_NMM 操作模式下该位无效
[8]	RS485_NMM	RS-485 普通多点操作模式 (NMM) 控制位 0 = 禁用 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RS485_AAD 操作模式下无效.

位	描述	
[7:0]	保留	保留

UART 功能选择寄存器 (UA_FUN_SEL)

寄存器	偏移	R/W	描述	复位值
UA_FUN_SEL	UART_BA+0x30	R/W	UART 功能选择寄存器	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						FUN_SEL	

位	描述	
[31:2]	保留	保留
[1:0]	FUN_SEL	<p>功能选择</p> <p>00 = UART功能模式</p> <p>01 = 保留</p> <p>10 = IrDA功能模式</p> <p>11 = RS-485功能模式</p>

5.14 I²C 串行接口控制器 (I²C)

5.14.1 概述

I²C是一种双线双向串行总线，为设备之间的数据交换提供了一种简单有效的方法。标准 I²C 是多主机总线，包含冲突检测和仲裁机制，以防止在两个或多个主机试图同时控制总线时发生数据冲突。

5.14.2 特性

I²C使用两根线（SDA和SCL）让连接在总线上的不同设备实现传输数据，总线的主要特征包括：

- 支持两个 I²C 接口
- 支持主机/从机模式
- 主从机之间双向数据传送
- 多主机主线（无中心主机）
- 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
- 串行同步时钟可以被用来作为握手机制实现挂起和恢复串行传输
- 内建14位溢出定时器，当I²C总线中止且定时器溢出，产生I²C中断
- 可编程的时钟可以用于多种速率控制
- 支持7位从地址模式
- 支持多地址识别（4组从机地址带mask选项）
- 支持唤醒模式
- 支持 FIFO 功能（仅限 NUC029FAE）

5.14.3 基本配置

NUC029xAN:

I^2C_0 的基本配置如下：

- I^2C_0 管脚由寄存器 P3_MFP [13:12] 配置。
- 使能 I^2C_0 时钟 (I2C0_EN) 由寄存器 APBCLK [8] 配置。
- 复位 I^2C_0 控制器 (I2C0_RST) 由寄存器 IPRSTC2 [8] 配置。

I^2C_1 的基本配置如下：

- I^2C_1 管脚由寄存器 P2_MFP [5:4] 或 P4_MFP[5:4] 配置。
- 使能 I^2C_1 时钟(I2C1_EN) 由寄存器 APBCLK [9] 配置。
- 复位 I^2C_1 控制器(I2C1_RST) 寄存器IPRSTC2 [9] 配置。

NUC029FAAE:

I^2C 的基本配置如下：

- I^2C 管脚由寄存器 P3_MFP [7:6] 配置。
- 使能 I^2C 时钟(I2C_EN) 由寄存器 APBCLK [8] 配置。
- 复位 I^2C 控制器(I2C_RST) 寄存器 IPRSTC2 [8] 配置。

5.14.4 功能描述

在 I^2C 总线中，数据是在主机和从机之间传送。数据在SCL和SDA上逐一字节同步传送。每一个数据的字节长度是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，每个传输字节后跟随一个应答位，每个位在SCL为高时采样；因此，SDA 线只有在SCL为低时才可以改变，在SCL为高时SDA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 I^2C 总线时序的细节请参考下图。

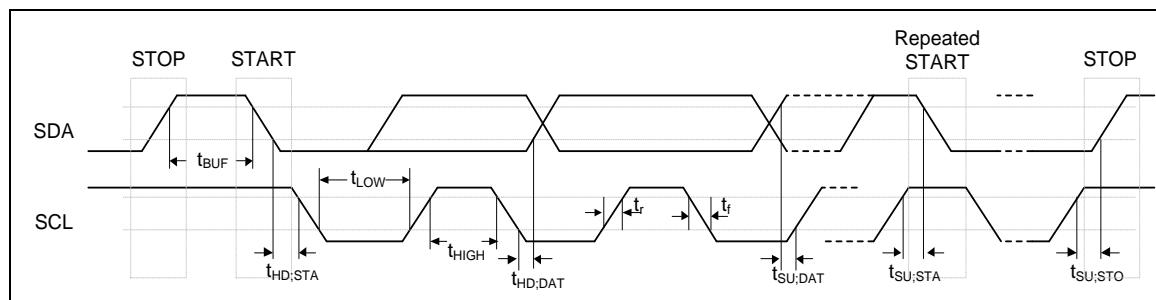


图 5-104 I^2C 总线时序

该设备片上I²C提供了一个符合I²C总线规范的串行总线接口。I²C端口自动处理字节传输。使能该端口需要将寄存器I2CON的ENS1位置为'1'。I²C硬件接口通过管脚SDA和SCL连接到I²C总线上。当I/O管脚作为I²C端口使用时，用户必须事先设定该管脚为I²C功能。

注：使用I²C时，需要给SDA和SCL引脚增加一组上拉电阻，因为这两个引脚为开漏脚。

5.14.4.1.1 二级缓存模式（仅限NUC029FAE）

对于NUC029FAE而言，有一个二级缓存来提高I²C总线性能。在二级缓存模式里，即使在当前数据已被传送的情况下或在上一个接收到的数据还未被读回的情况下，下一个传送的数据或上一个接收的数据仍处于活动状态。

当发生SI事件，I²C SCL总线会被拉低。NSTRETCH位用来控制在发生SI事件时，I²C SCL总线上的电平是否被拉低。

当二级缓存使能且中断事件使能被设置时，就会产生under run或over run中断。

5.14.5 I²C 协议

下图展示了典型的I²C协议。通常，一个标准的通讯包含四个部分：

- 1) 产生起始或复位起始信号
- 2) 从机地址和R/W位传输
- 3) 数据传输
- 4) 停止信号的产生

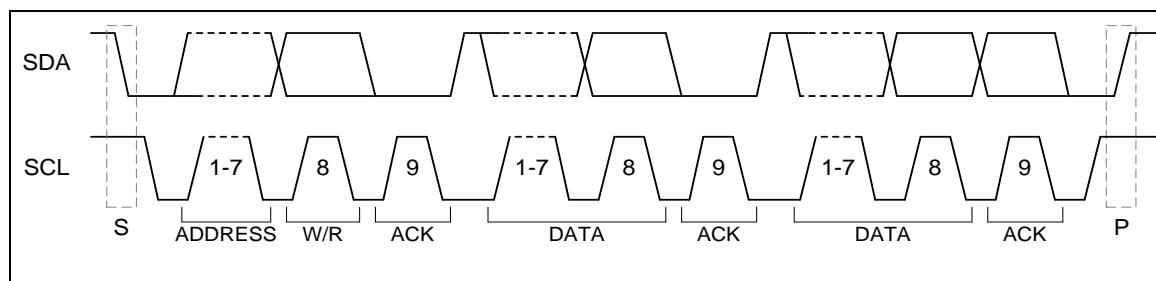


图 5-105 I²C 协议

5.14.5.1 起始或复位起始信号

当总线处于空闲状态下，意味着没有主机设备占用总线（SCL和SDA都为高），主机可以通过发送起始信号来发起传输。起始信号，通常表示为S-bit，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数据传输的开始。

重复起始信号并不是两个起始信号之间的停止信号，通常表示为Sr-bit。主机用这种方式来与另一个从机进行通信，或者与相同的从机进行相反的传输方向上的通信（从写设备到读设备），而不必释放总线。

5.14.5.2 停止信号

主机可以通过产生一个停止信号来终止数据传送。停止信号，通常表示为P-bit，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。

下图展示了起始信号、重复起始信号和停止信号的波形：

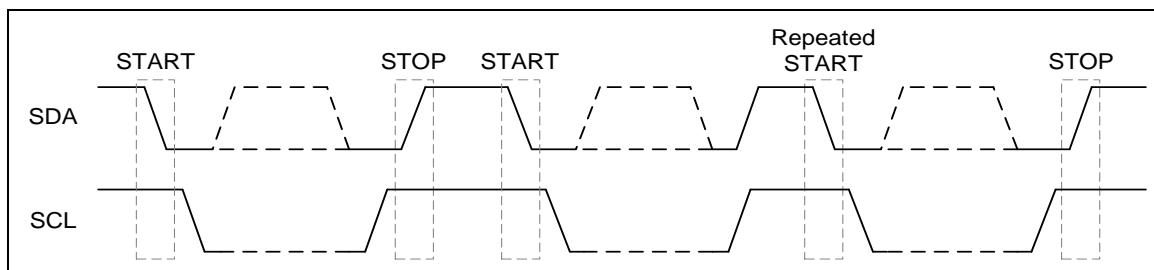


图 5-106 起始和停止条件

5.14.5.3 从机地址传输

在起始信号后，主机会立即传送的第一个字节数据就是从机地址（SLA）。该字节是由称为从机地址的7位跟随1位读/写（R/W）位构成。R/W位，对于从机信号而言，表示着数据的传输方向。系统中不可能让两个从机有相同的地址，只有被主机寻址的从机设备才会通过在第9个SCL时钟周期内，将SDA拉低作为应答信号。

5.14.5.4 数据传输

当从机接收到了带有读/写位的正确地址时，数据将会依照读/写位按规定传输。每一个传输的字节都会在SCL的第9个时钟周期跟随一个应答信号。如果从机产生一个非应答信号（NACK），主机可以产生一个停止信号来中止数据传输或者产生一个重复起始信号来开始新一轮的传输。

当主机作为一个接收设备时，给从机产生无应答信号（NACK），从机就会释放数据线来让主机产生停止或重复起始信号。

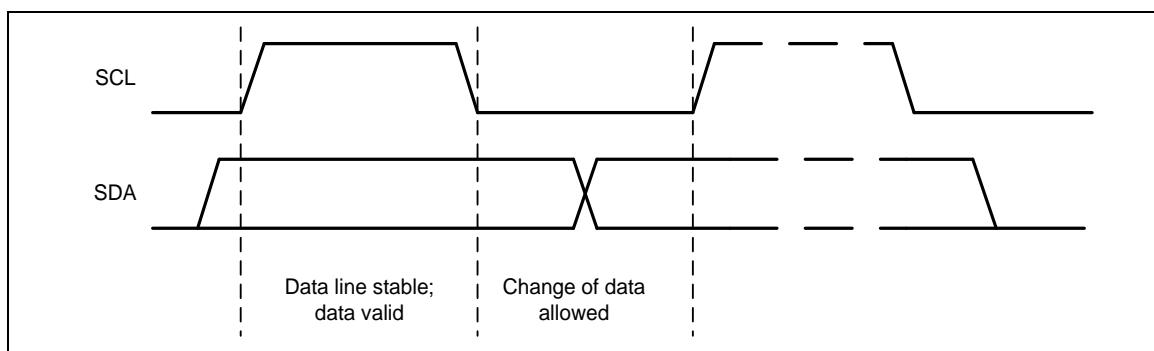
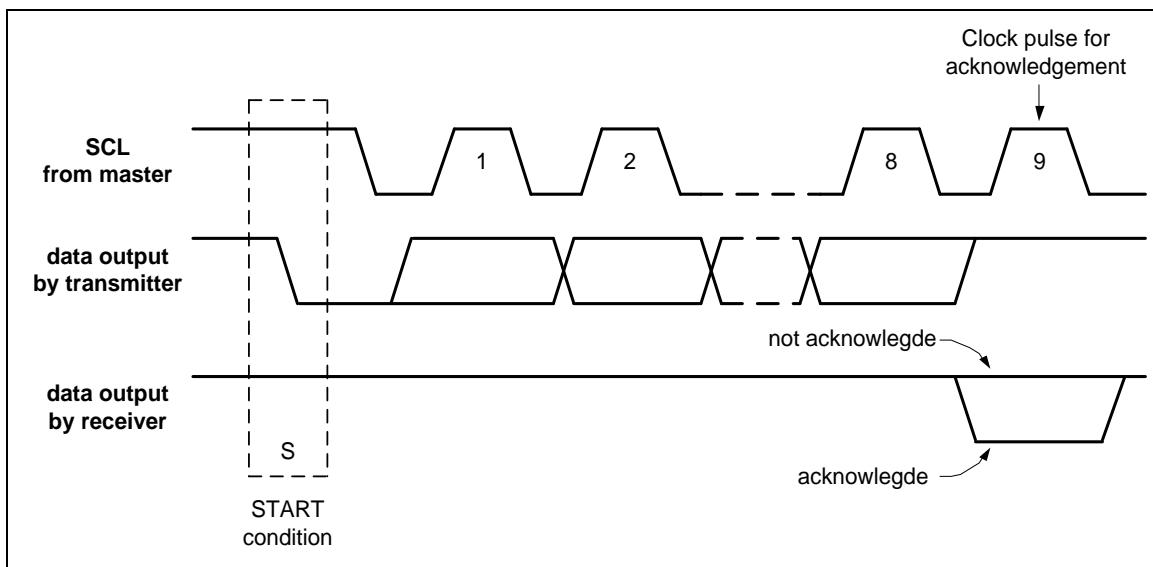


图 5-107 I²C 总线上的位传输

图 5-108 I²C 总线上的应答信号

5.14.5.5 I²C 总线上的数据传输

下图展示了主机发送数据到从机的过程。主机发送7位地址和1位写标志到从机，表示主机想要发送数据到从机。在从机回应应答信号后，主机继续传送数据。

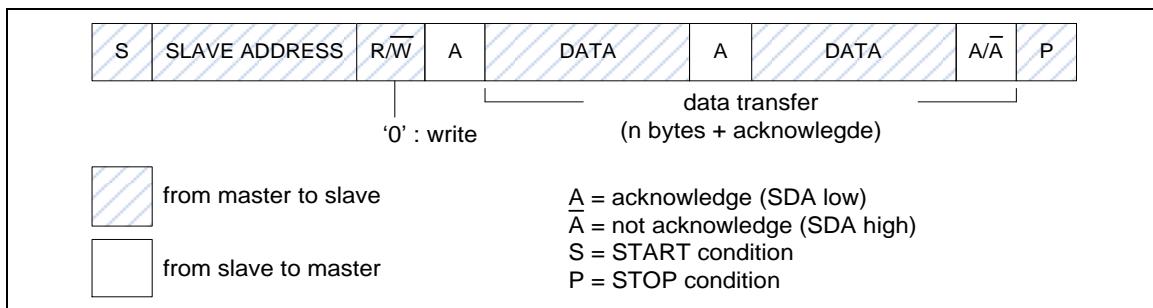


图 5-109 主机传送数据到从机

下图展示了主机读取从机数据的过程。主机发送7位地址和1位读标志到从机，表示主机想要读取从机数据。在从机回应应答信号后，从机开始传送数据给主机。

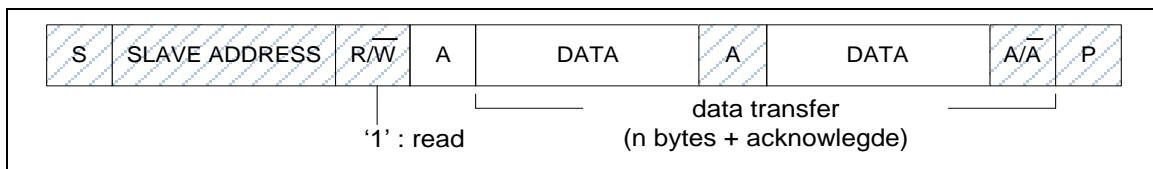


图 5-110 主机读取从机数据

5.14.5.6 I²C 总线的二级缓存模式 (仅限NUC029FAE)

使能二级缓存是用来临时存储I²C发送或接收的数据。这样，可以提高I²C总线的性能。如果TWOFF_EN设置为1，在当前SI清除之后，用于重复起始的STA位或者STO位应当被置位。

例如：4个数据被传输后再停止。STO位应该在第3个数据的SI事件清除后，进行置位。这时，第4个数据才可以被传输，在第4个数据传输完成后，I²C停止。

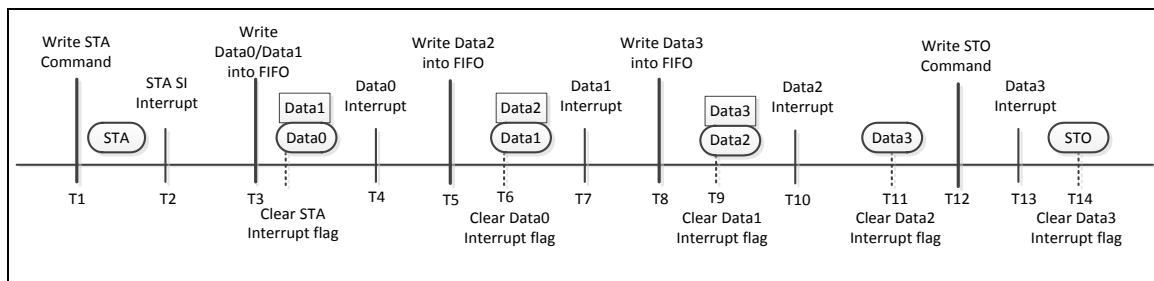


图 5-111 主机写时 二级缓存传输时序

例如：从机模式下，要收到4个数据。在第1个数据加载进接收缓存后，控制器才能从I²C总线上接收第2个数据，用户可以在第1个中断状态清除后，读到第1个数据。

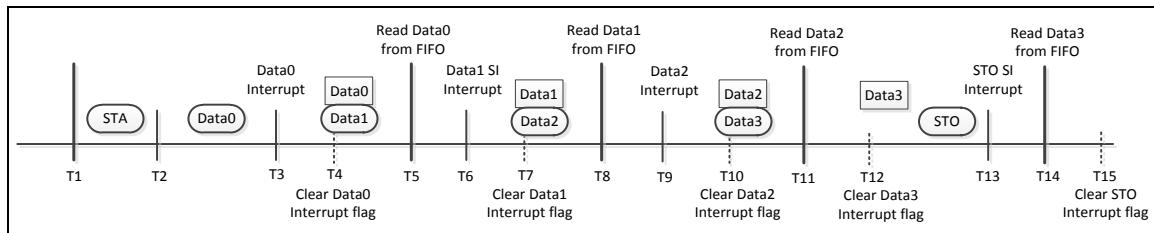


图 5-112 从机读时 二级缓存传输时序

5.14.6 I²C 协议寄存器

可通过以下15个特殊功能寄存器来控制I²C接口：I2CON（控制寄存器）、I2CSTATUS（状态寄存器）、I2CDAT（数据寄存器）、I2CADDRn（地址寄存器，n=0~3）、I2CADMn（地址屏蔽寄存器，n=0~3）、I2CLK（时钟速率寄存器）、I2CTOC（超时计数器寄存器）、I2CWKUPCON（唤醒控制寄存器）和I2CWKUPSTS（唤醒状态寄存器）。

地址寄存器 (I2CADDR)

I²C端口带有4个从机地址寄存器I2CADDRn (n=0~3)。当I²C处于主机模式时，这四个寄存器的值是无关的。在从机模式下，位域I2CADDRn [7:1]必须装入芯片自身的从机地址，当I2CADDRn地址

与接收的从机地址符合时，I²C硬件将应答。

I²C端口支持广播呼叫功能。当GC位(I2CADDRn [0])被置位，I²C端口硬件会响应广播呼叫地址(00H)。清GC位可禁止广播呼叫功能。

GC位被置位，且I²C处于从机模式时，在主机发送广播呼叫地址到I²C总线上之后，I²C可以接收广播呼叫地址00H，然后它将跟随GC模式的状态。

从机地址掩码寄存器 (I2CADM)

I²C总线控制器支持多地址识别，带有4组地址屏蔽寄存器I2CADMn (n=0~3)。当地址屏蔽寄存器某一位置1，表示接收到的地址的相应位将被忽略。如果该位置0，表示接收到的地址的相应位应当与地址寄存器中相应位的值完全一致。

数据寄存器 (I2CDAT)

该寄存器存储的内容是准备发送的或刚接收的一个字节的数据。只要不在移位处理的过程中，CPU可以直接读写这8位I2CDAT [7:0]。当 I²C 处于定义的状态下，且串行中断标志(SI)被置位，I2CDAT[7:0]中的数据保持稳定。在数据被移出的过程中，总线上的数据同时被移入，I2CDAT[7:0]总是包含出现在总线上的最后一个字节数据。

应答位由I²C的硬件控制，CPU不能访问。串行数据在SCL线上串行时钟脉冲的上升沿被移入I2CDAT[7:0]。当一个字节被移入到I2CDAT [7:0]后，I2CDAT [7:0]中的串行数据是可以使用的，应答位(ACK或NACK)由控制逻辑在第9个时钟返回。为了在发送数据的时候监控总线状态，当发送I2CDATA[7:0]中数据到总线的时候，总线上的数据将同时被移入I2CDATA[7:0]中。在这种发送数据的情况下，串行数据在SCL时钟脉冲的下降沿从I2CDAT[7:0]被移出，在SCL时钟脉冲的上升沿被移入I2CDAT[7:0]。

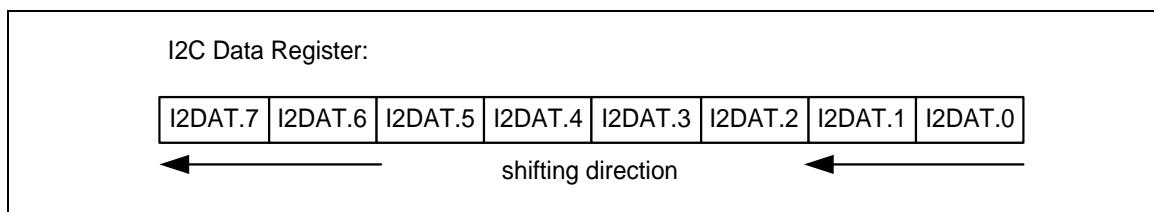


图 5-113 I²C 数据移动方向

控制寄存器 (I2CON)

CPU可以直接读写寄存器I2CON[7:0]。当I2C端口通过ENS1 (I2CON [6])等于1使能后，内部状态由I2CON 和 I2C逻辑硬件控制。

I2CON有2位会受到硬件的影响：SI位在I2C硬件请求一个串行中断时被置位，STO位在总线上出现停止条件或者ENS1=0时被清零。

当有新的状态码产生并存储在I2CSTATUS寄存器中时，I2C中断标志位SI (I2CON [3])由硬件置位，如同时使能中断位EI (I2CON [7]) 置位，则产生I2C中断请求。I2CSTATUS[7:0]中保存内部状态码，在SI被软件清除前，其内容保持不变

状态寄存器 (I2CSTATUS)

I2CSTATUS [7:0] 是一个8位只读寄存器。位域I2CSTATUS[7:0]包含状态码，有26个可能的状态码。当I2CSTATUS [7:0]的内容是F8H时，没有串行中断请求。所有其它的I2CSTATUS [7:0]的值对应于定义的I2C状态。当进入这些状态中的任一个，就会产生状态中断请求(SI = 1)。在SI被硬件置位1个机器周期后，有效状态码出现在I2CSTATUS [7:0]中，并保持稳定直至SI被软件复位。

另外，00H状态表示总线错误。总线错误在起始或停止信号出现在帧结构非法的位置时发生。总线错误可能在串行传输地址字节，数据字节或应答位期间发生。为了将I2C从总线错误中恢复，需要置位STO，清除SI从而进入没有寻址的从机模式，然后清除STO释放总线并等待新的通信。I2C总线在总线错误时不能识别停止信号。

主机模式		从机模式	
状态	描述	状态	描述
0x08	开始	0xA0	从机发送重新开始或停止
0x10	主机重复开始	0xA8	从机发送地址ACK
0x18	主机发送地址ACK	0xB0	从机发送仲裁丢失
0x20	主机发送地址NACK	0xB8	从机发送数据ACK
0x28	主机发送数据ACK	0xC0	从机发送数据NACK
0x30	主机发送数据NACK	0xC8	从机发送最后数据ACK
0x38	主机仲裁丢失	0x60	从机接收地址 ACK
0x40	主机地址接收ACK	0x68	从机接收仲裁丢失
0x48	主机地址接收NACK	0x80	从机接收数据 ACK
0x50	主机数据接收 ACK	0x88	从机接收数据NACK
0x58	主机数据接收 NACK	0x70	广播模式地址ACK
0x00	总线错误	0x78	广播模式仲裁 Lost
		0x90	广播模式数据 ACK
		0x98	广播模式数据 NACK
0xF8	总线释放 注：主从模式都存在“0xF8”状态，它不会引起中断。		

表 5-25 I²C 状态码描述

时钟波特率位 (I2CLK)

当I²C在主机模式下，I²C数据的波特率由I2CLK[7:0]寄存器决定。在从机模式下时I2CLK[7:0]的值不重要。在从机模式下，I²C将自动与主机I²C设备时钟频率同步。

I²C数据波特率设定公式：I²C的数据波特率 = (system clock) / (4x(I2CLK[7:0]+1))，如果system clock = 16MHz, I2CLK[7:0]= 40(28H), I²C的数据波特率 = 16MHz /(4X (40 +1)) = 97.5K比特/秒。

超时计数寄存器 (I2CTOC)

有一个14位的超时计数器可以用于处理I²C总线挂起。当超时计数功能使能后，计数器开始向上计数直至溢出（TIF=1），并且产生I²C中断给CPU，或者将ENTI清零来停止计数。当超时计数器使能后，写1到SI标志将会复位计数器，在SI清零之后计数器会重新开始计数。如果I²C总线挂起，会导致I2STATUS及SI标志在一段时间内不再更新。该14位超时计数器可能溢出并向CPU产生I²C中断请求。关于14位超时计数器参考下图，用户可通过对TIF位写1清0该标志。

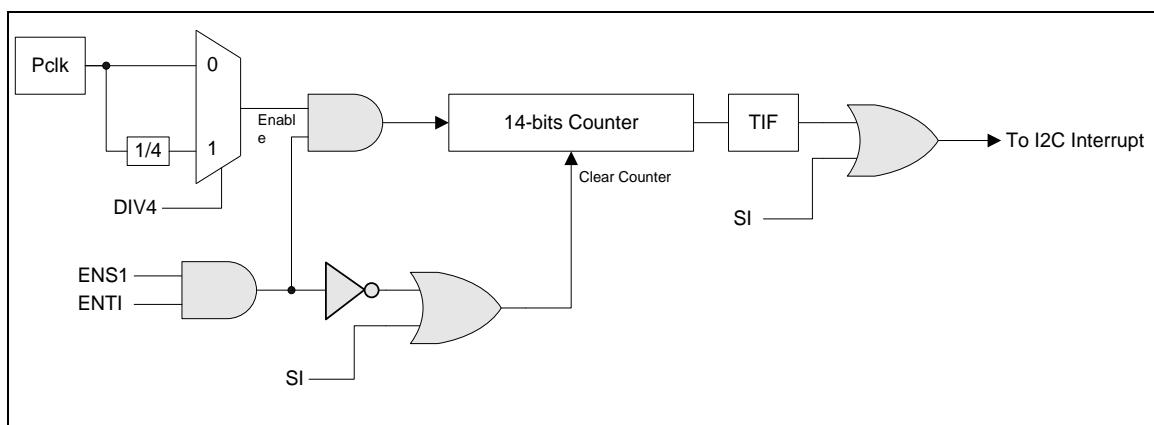


图 5-114 I²C 超时计数器框图

唤醒控制寄存器 (I2CWKUPCON) (NUC029xAN) / I²C 控制寄存器2 (I2CON2) (NUC029FAE)

当芯片进入低功耗模式时，另一个I²C主机可以通过寻址当前I²C设置来唤醒该芯片。在进入低功耗模式前，用户必须配置相关的设置。当地址与四个地址寄存器中任意一个匹配时，芯片被唤醒，此次通讯中，随后的数据将被忽略。

唤醒状态寄存器 (I2CWKUPSTS) (NUC029xAN) / I²C 状态寄存器2 (I2CSTATUS2) (NUC029FAE)

当系统被其它I²C主机唤醒时，WKUPIF会被置位来表示该事件。用户需要对其写1清零。

当TWOFF_EN置位时，另一个状态位用来表示当前FIFO状态（仅限NUC029FAE）

5.14.7 操作模式

片上I²C端口支持3种操作模式：主机，从机和广播呼叫模式。

在实际应用中，I²C端口可以作为主机或从机。在从机模式，I²C端口寻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，并且从机打算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个时钟被发出，此时，如果中断被使能，则在主机和从机设备上都会发生一次中断请求。当微控制器希望成为总线主机时，在进入主机模式之前，硬件等待总线空闲以使可能的从机动作不会被打断。在主机模式下，如果总线仲裁失败，I²C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。

为了控制每个模式下I²C总线传输，用户需要根据I2CSTATUS寄存器中的当前状态码设置I2CON和I2CDAT寄存器。换句话说，对于每个I²C总线动作，用户都需要通过I2CSTATUS寄存器来检测当前状态，然后设置I2CON和I2CDAT寄存器，去进行下一次总线动作。最终，通过I2CSTATUS来检测响应状态。

I2CON [3]寄存器中的SI标志被清除以后，I2CON寄存器中的STA、STO和AA用来控制I2C硬件的下一个状态。在新动作完成时，一个新的状态码将被更新到I2CSTATUS寄存器中，SI将被置。如果I2C中断控制位EI(I2CON [7])被置位，新状态码的合适的动作或者软件流程可以放到中断服务程序中执行。

下图描述当前I²C状态码是0x08，然后设置I2CDATA=SLA+W并且(STA,STO,SI,AA) = (0,0,1,x)发送地址到I²C总线上。如果某个从机地址匹配，将响应ACK，然后I2CSTATUS寄存器的值将变为0x18。

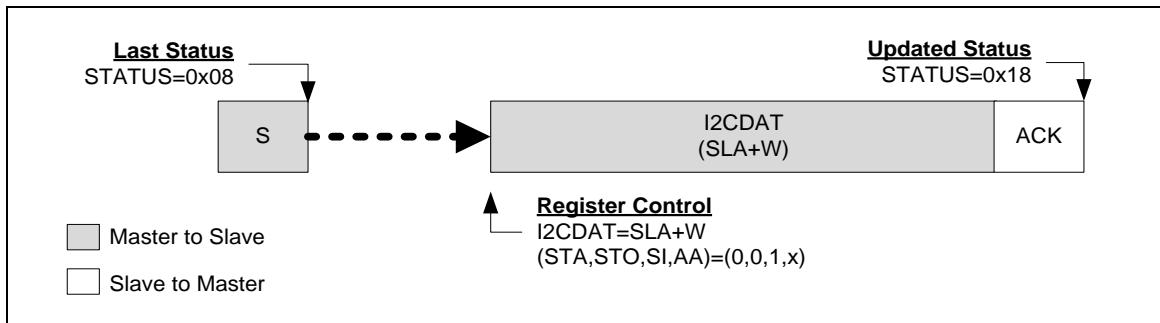


图 5-115 根据当前 I²C 状态码来控制 I²C 总线

5.14.7.1 主机模式

下图中所有可能的I²C主机协议都显示出来了，用户需要遵守合适的路径来实现I²C协议。

换句话说，用户可以发送一个起始信号到总线，在起始信号发成功发出后，I²C将会被设置为主机传送模式或主机接收模式。新的状态码将变为0x80。跟随着起始信号，用户可以发送从机地址，读写位，数据，重复起始信号和停止信号来执行I²C协议。

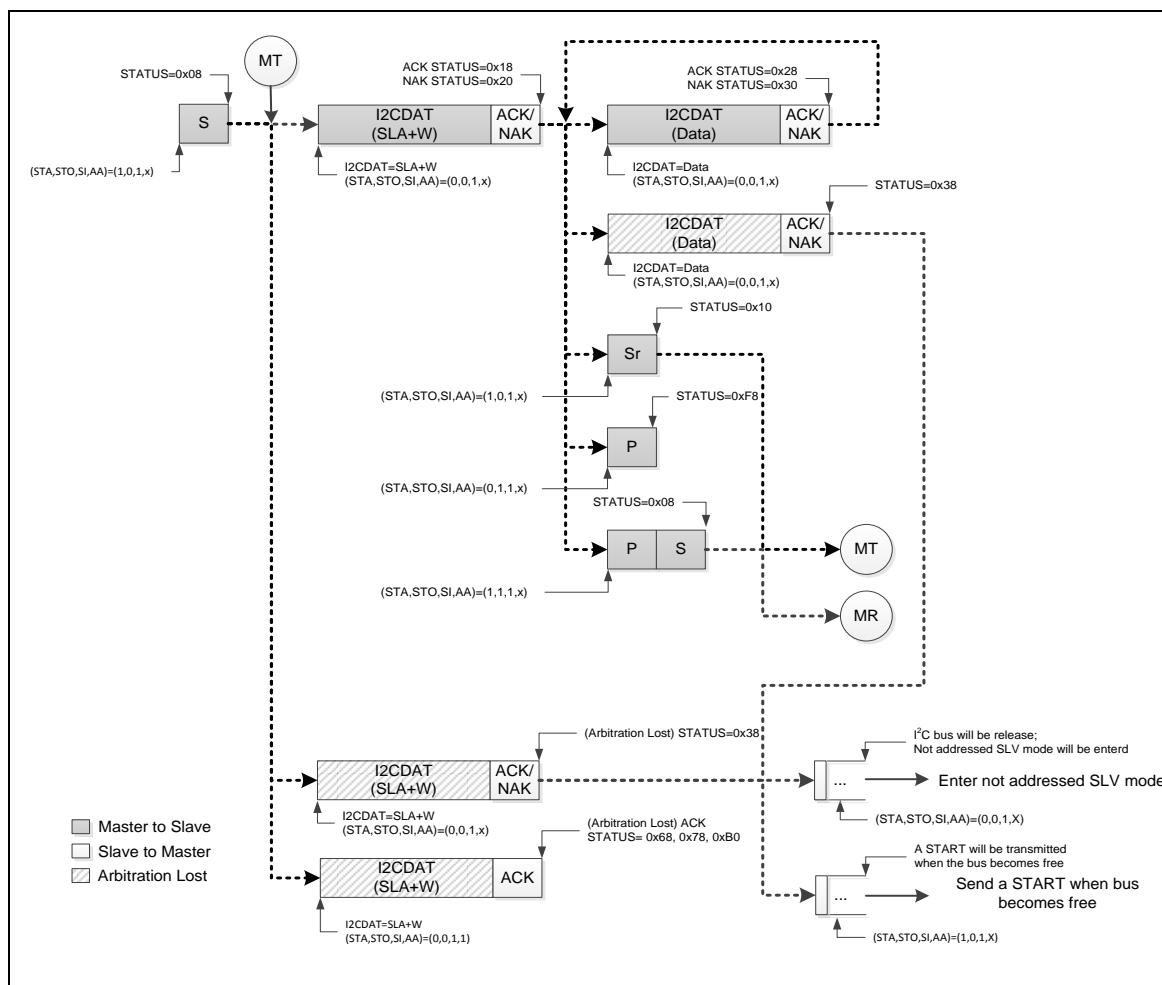


图 5-116 主机发送模式控制流程

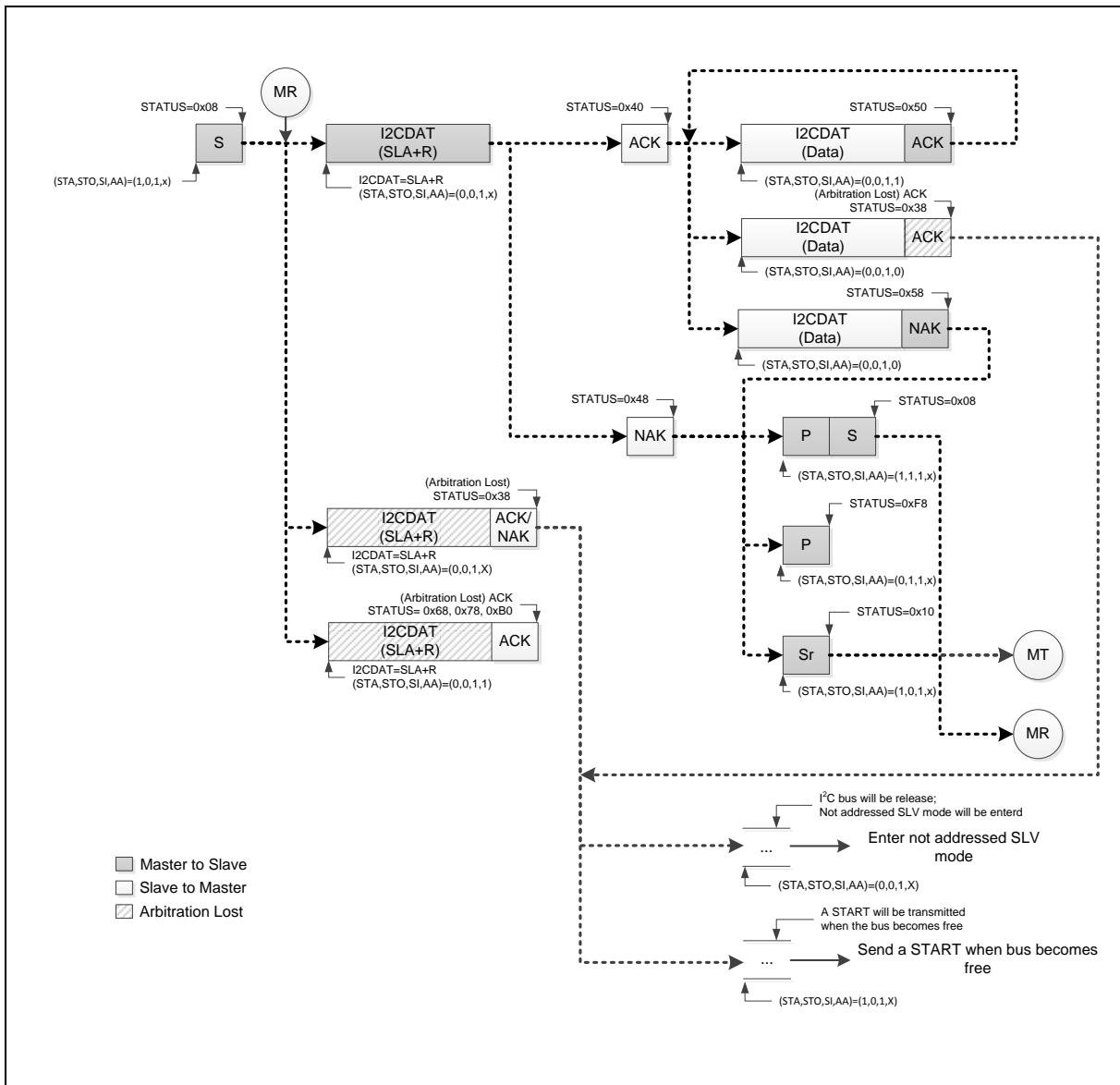


图 5-117 主机接收模式控制流程

如果I2C在主机模式下并且仲裁丢失，状态码将变为0x38，在状态0x38下，用户可以设置(STA, STO, SI, AA) = (1, 0, 1, X)在总线变成空闲时发送起始信号来重新开始主机操作。另外，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。

5.14.7.2 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I²CADDRx设置从机地址，设置(STA, STO, SI, AA) = (0, 0, 1, 1)来让I²C识别主机发送的地址。图 5-118示意了从机模式的所有流程。用户需要遵循合适的流程（如图 5-118 所示）来实现自己的I²C协议。

如果在主机模式仲裁丢失，I2C端口立即切换到从机模式并且在同一串口传输中识别自己的从机地址。

址。如果在仲裁丢失后识别到地址是SLA+W（主机写数据到从机），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: 在I²C通讯时，在从机模式下当写“1”清除SI标志时，SCL时钟会被释放。

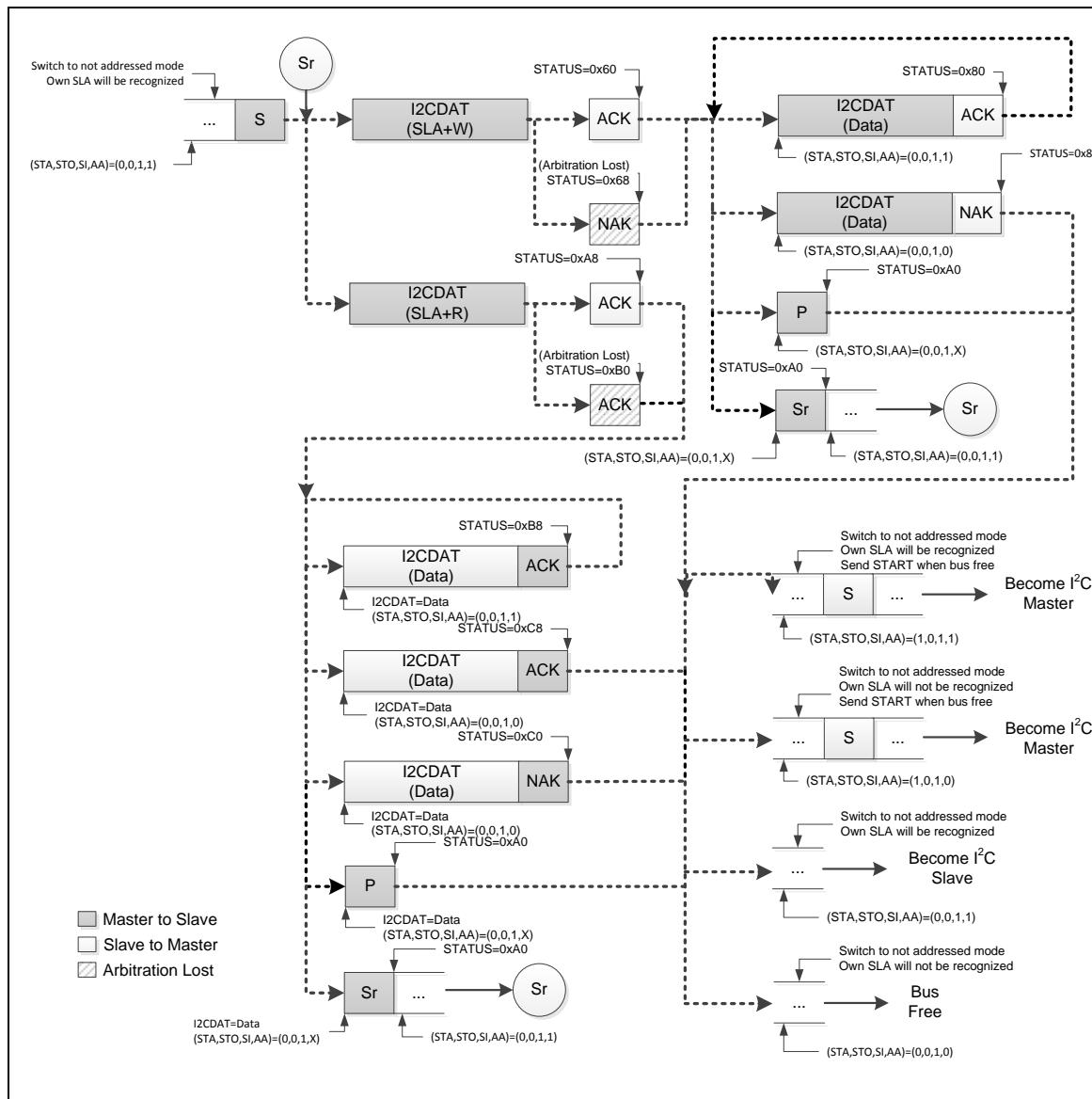


图 5-118 从机模式控制流程

如果I²C在已寻址从机模式下还在接收数据，但是收到了STOP或者重复START信号，状态码将变成0xA0，用户应该遵守上图0x88状态码的行为来处理。

如果I²C在已寻址从机模式下还在发送数据，但是收到了STOP或者重复START信号，状态码将变

成0xA0，用户应该遵守上图0xC8状态码的行为来处理。

注：从机获得0x88, 0xC8, 0xC0 和0xA0状态后,从机会切换到无地址模式，自身SLA不会被辨识。如果进入这种状态，从机不再从主机接收任何信号或地址。需要复位才能离开这种状态。

5.14.7.3 广播呼叫模式 (GC)

如果GC位 (I²C_ADDRx[0]) 被置位, I²C端口硬件将响应广播呼叫地址 (00H)。用户可以清除GC位来禁止广播呼叫功能。当GC位被置位，并且I²C在从模式时，在主机发送广播呼叫地址到I²C总线上时，它可以接收广播呼叫地址0x00，然后遵循广播呼叫模式的状态。

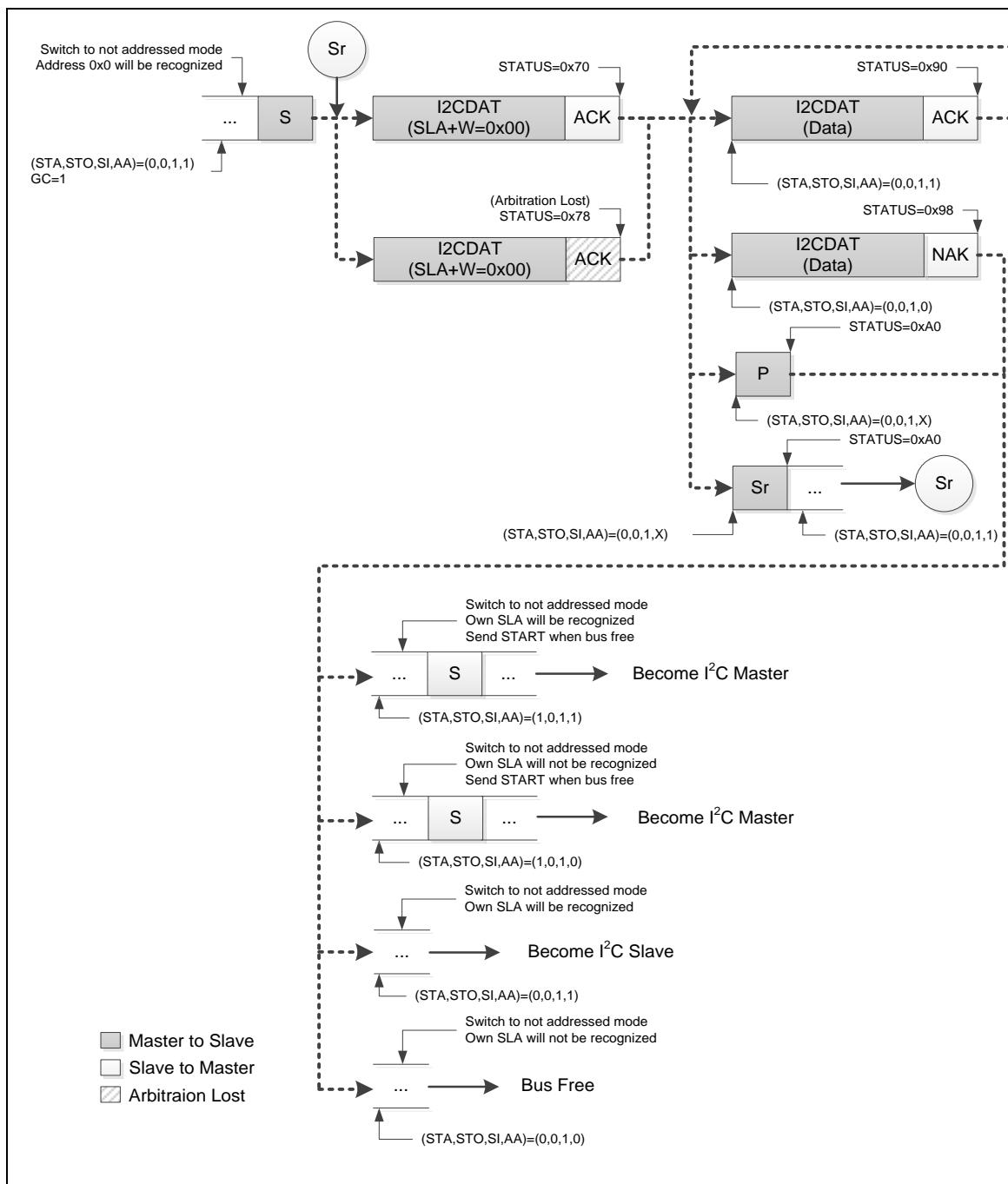


图 5-119 GC 模式

当I²C还在接收数据时，但是收到了STOP或者重复START信号，状态码将变成0xA0。当得到0xA0状态时，用户应该遵守上图中状态码为0x98的情形来进行处理。

注:在从机得到0x98和0xA0状态码之后，从机会切到无地址模式，并且自身的SLA也不会被识别。如果进入这个状态，从机将不会接收来自主机的任何I²C信号和地址。此时，I²C控制器应该被复位以离开此状态。

5.14.7.4 多主机

在某些应用中，I²C总线上同时有2个或者多个主机来访问从机，主机可能同时发送数据。该I²C具备冲突检测和裁决的功能，可以避免数据冲突，支持多主机模式。

- 当 I2C_STATUS = 0x38时，会发生“仲裁失败”。仲裁失败事件可能发生在发送START信号、数据信号或者STOP信号时，在总线空闲的时候，用户可以设置 (STA, STO, SI, AA) = (1, 0, 1, X) 来再次发送START 信号，或者设置(STA, STO, SI, AA) = (0, 0, 1, X) 来发送STOP信号返回无地址从机模式。
- 当I2C_STATUS = 0x00时，会发生“总线错误”，为了将I2C从总线错误中恢复，可以设置STO并清除SI，之后STO会被清除总线也得以释放。
 - 设置 (STA, STO, SI, AA) = (0, 1, 1, X) 来停止当前传输
 - 设置 (STA, STO, SI, AA) = (0, 0, 1, X) 来释放总线

5.14.7.5 随机读 EEPROM 的例子

使用I²C读EEPROM时，使用下列步骤配置I²C相关寄存器

1. 设置多功能引脚“P3_MFP”和“ALT_MFP”作为SCL和SDA引脚
2. 使能 I²C APB 时钟：“APBCLK”寄存器中I2C_EN=1
3. 设置 I²C_RST=1 复位 I²C 控制器，然后设置“IPRSTC2”寄存器中的I2C_RST=0使I²C控制器到正常操作模式
4. “I2CON”寄存器中设置ENS1=1，使能 I²C 控制器
5. 在“I2CLK”寄存器中，给I²C时钟分频寄存器一个数值来设置合适的I²C时钟频率
6. “NVIC_ISER”寄存器中，设置SETENA=0x00040000使能I²C中断
7. “I2CON”寄存器中，设置EI=1，使能 I²C 中断
8. 设置 I²C 地址寄存器：“I2CADDR0~I2CADDR3”

随机读操作是访问EEPROM中的一种访问方法。该方法允许主机访问EEPROM中的任何地址。下图显示了EEPROM 随机读操作的流程。

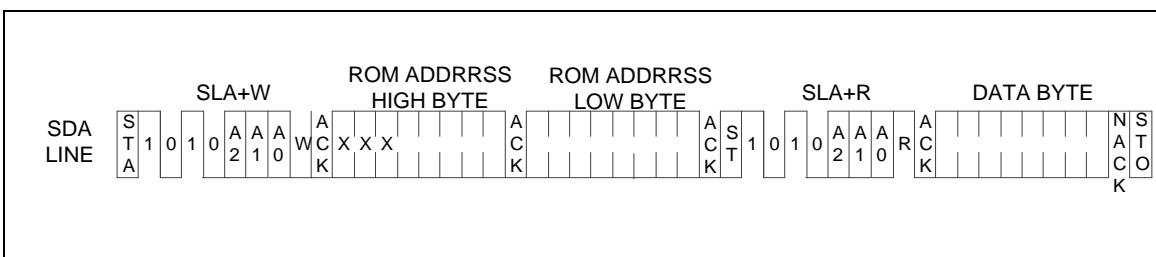


图 5-120 随机读取 EEPROM

下图显示了怎样使用I²C控制器来实现 EEPROM 随机读协议。

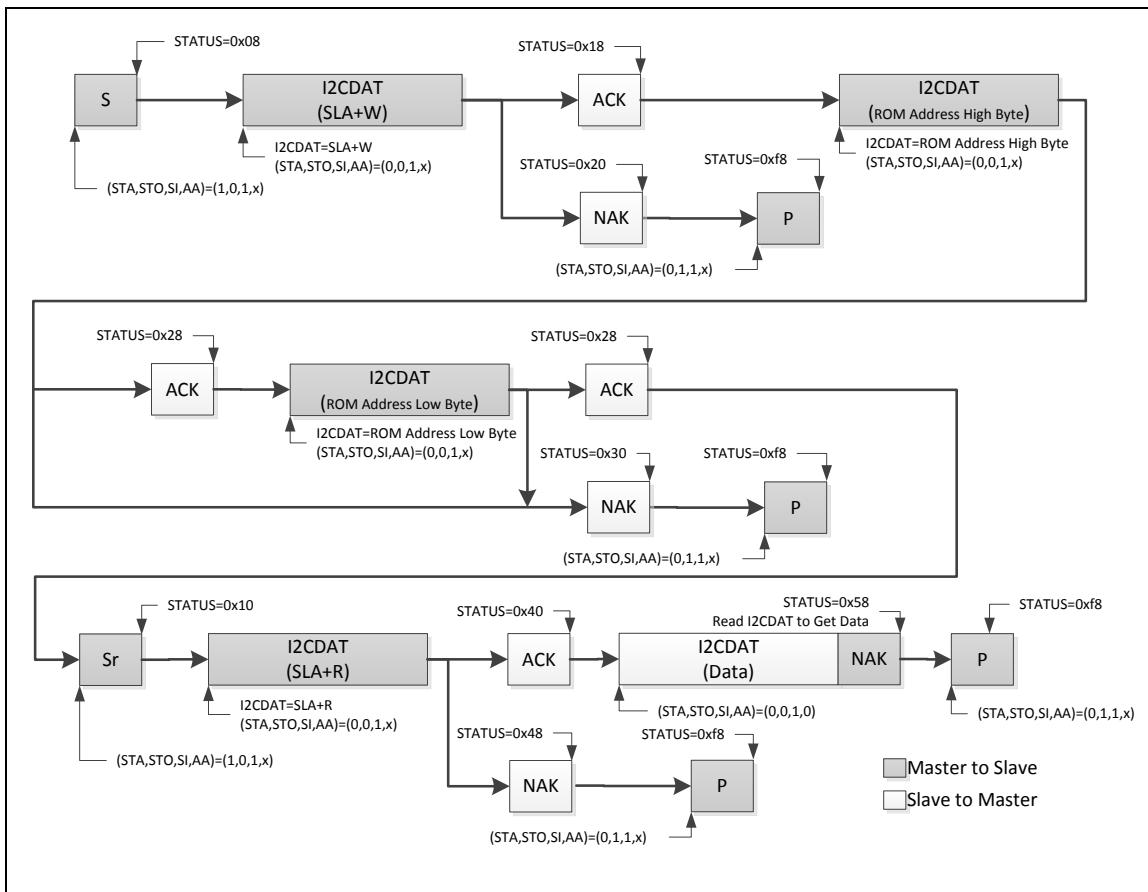


图 5-121 随机读取 EEPROM 的协议

作为主机，I²C 控制器发送 START信号到总线上，然后发送SLA+W(从机地址+写)到EEPROM，之后是2个字节的要读的EEPROM数据的地址。最后，重复START信号并发送SLA+R，之后就能从EEPROM传出要读取的数据了。

5.14.8 NUC029xAN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:				
I2C0_BA = 0x4002_0000				
I2C1_BA = 0x4012_0000				
I2CON x=0,1	I2Cx_BA+0x00	R/W	I ² C控制寄存器	0x0000_0000
I2CADDR0 x=0,1	I2Cx_BA+0x04	R/W	I ² C从机地址寄存器0	0x0000_0000
I2CDAT x=0,1	I2Cx_BA+0x08	R/W	I ² C数据寄存器	0x0000_0000
I2CSTATUS x=0,1	I2Cx_BA+0x0C	R	I ² C状态寄存器	0x0000_00F8
I2CLK x=0,1	I2Cx_BA+0x10	R/W	I ² C时钟分频寄存器	0x0000_0000
I2CTOC x=0,1	I2Cx_BA+0x14	R/W	I ² C超时控制寄存器	0x0000_0000
I2CADDR1 x=0,1	I2Cx_BA+0x18	R/W	I ² C从机地址寄存器1	0x0000_0000
I2CADDR2 x=0,1	I2Cx_BA+0x1C	R/W	I ² C从机地址寄存器2	0x0000_0000
I2CADDR3 x=0,1	I2Cx_BA+0x20	R/W	I ² C从机地址寄存器3	0x0000_0000
I2CADM0 x=0,1	I2Cx_BA+0x24	R/W	I ² C从机地址掩码寄存器0	0x0000_0000
I2CADM1 x=0,1	I2Cx_BA+0x28	R/W	I ² C从机地址掩码寄存器1	0x0000_0000
I2CADM2 x=0,1	I2Cx_BA+0x2C	R/W	I ² C从机地址掩码寄存器2	0x0000_0000
I2CADM3 x=0,1	I2Cx_BA+0x30	R/W	I ² C从机地址掩码寄存器3	0x0000_0000
I2CWKUPCON x=0,1	I2Cx_BA+0x3C	R/W	I ² C唤醒控制寄存器	0x0000_0000
I2CWKUPSTS x=0,1	I2Cx_BA+0x40	R/W	I ² C唤醒状态寄存器	0x0000_0000

5.14.9 NUC029xAN 寄存器描述

I²C 控制寄存器 (I2CON)

寄存器	偏移	R/W	描述	复位值
I2CON	I2Cx_BA+0x00	R/W	I ² C 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
EI	ENS1	STA	STO	SI	AA	保留	

位	描述	
[31:8]	保留	保留
[7]	EI	I²C中断使能控制 0 = 禁用 I ² C 中断 1 = 使能 I ² C 中断
[6]	ENS1	I²C控制器使能控制 0 = 禁用 I ² C 控制器 1 = 使能 I ² C 控制器 置位该位以使能I ² C串行功能控制器。当ENS1=1时，I ² C串行功能使能。多功能引脚的功能必须先设置为I ² C功能。
[5]	STA	I²C 起始控制 设置STA为逻辑1来进入主机模式，当总线空闲时，I ² C硬件会发出起始或者重复起始信号到总线上。
[4]	STO	I²C 停止控制 在主机模式下，置位STO将向总线传输停止条件，然后I ² C硬件会检查总线状态。一旦检测到停止条件，该位将被硬件自动清零。在从机模式下，置位STO会将I ² C硬件复位至“无地址”的从机模式。这意味着该设备不再处于从机接收模式，不能从主机发送设备接收数据。
[3]	SI	I²C中断标志 当一个新的I ² C状态出现在 I2CSTATUS 寄存器时，硬件置位SI标志。如果EI (I2CON [7])已经置位，将发生I ² C中断。SI 必须由软件清零。向该位写1清零。
[2]	AA	接收应答控制位 若在地址或数据接收之前，AA=1，在SCL线上的应答时钟脉冲期间返回应答信号（SDA为低）的条件是：1、从机应答主机发送的地址信息；2、接收设备应答发送设备发送的数据。 若在地址或数据接收之前，AA=0，在SCL线上的应答时钟脉冲期间会返回非应答信号（SDA

		为高)。
[1:0]	保留	保留

I²C 数据寄存器 (I2CDAT)

寄存器	偏移	R/W	描述	复位值
I2CDAT	I2Cx_BA+0x08	R/W	I ² C 数据寄存器	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
I2CDAT							

位	描述	
[31:8]	保留	保留
[7:0]	I2CDAT	I²C 数据寄存器 Bit [7:0] 为8位I ² C串行端口的传输数据。

I²C 状态寄存器 (I2CSTATUS)

寄存器	偏移	R/W	描述	复位值
I2CSTATUS	I2Cx_BA+0x0C	R	I ² C 状态寄存器	0x0000_00F8

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
I2CSTATUS							

位	描述	
[31:8]	保留	保留
[7:0]	I2CSTATUS	I²C 状态 低三位始终是0；高5位包含状态码。总共有26种状态码；当I2CSTATUS的值是F8H，不会产生中断请求；其它的所有的I2CSTATUS值都对应定义的I ² C的状态。当进入这些状态时会产生一个状态中断请求(SI=1)。一个有效的状态码在SI被硬件设为'1'后的一个周期内出现在I2CSTATUS中，并保持稳定直至SI被软件复位后的下一个周期。另外，状态码是00H时表示总线错。当'起始'或'结束'信号出现在不正确的帧位置时会产生总线错误。比如在串行传输地址字节、数据字节或者应答位时都可能出现总线错误。

I²C 时钟分频寄存器 (I2CLK)

寄存器	偏移	R/W	描述	复位值
I2CLK	I2Cx_BA+0x10	R/W	I ² C 时钟分频寄存器	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
I2CLK							

位	描述	
[31:8]	保留	保留
[7:0]	I2CLK	<p>I²C 时钟分频寄存器</p> <p>I²C 时钟比特率: I²C 数据波特率 = PCLK / (4x (I2CLK+1)).</p> <p>注: I2CLK 的最小值为 4。</p>

I²C 超时控制寄存器 (I2CTOC)

寄存器	偏移	R/W	描述	复位值
I2CTOC	I2Cx_BA+0x14	R/W	I ² C 超时控制寄存器	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					ENTI	DIV4	TIF

位	描述	
[31:3]	保留	保留
[2]	ENTI	超时计数使能控制 0 = 禁用超时计数器 1 = 使能超时计数器 注: 当14位超时计数器被使能时, SI被清0后, 14位超时计数器开始计数。写1到SI中会使计数器复位, SI被清零后计数器又重新开始计数。
[1]	DIV4	超时计数输入时钟除以4 0 = 禁用超时计数输入时钟除以4 1 = 使能超时计数输入时钟除以4 注: 使能后, 超时时间延长4倍。
[0]	TIF	超时标志 当I ² C超时发生时, 这个位由硬件置位。如果I ² C中断同时使能位(EI)设置为1, 将中断CPU。 注: 软件写"1"清零。

I²C 从机地址寄存器 (I2CADDRx)

寄存器	偏移	R/W	描述	复位值
I2CADDR0	I2Cx_BA+0x04	R/W	I ² C 从机地址寄存器0	0x0000_0000
I2CADDR1	I2Cx_BA+0x18	R/W	I ² C 从机地址寄存器1	0x0000_0000
I2CADDR2	I2Cx_BA+0x1C	R/W	I ² C 从机地址寄存器2	0x0000_0000
I2CADDR3	I2Cx_BA+0x20	R/W	I ² C 从机地址寄存器3	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
I2CADDR							GC

位	描述	
[31:8]	保留	保留
[7:1]	I2CADDR	I²C 地址寄存器 主机模式下，该寄存器内容没有意义。从机模式下，高7位作为芯片本身的地址。如果任一地址符合，I ² C 硬件将会自动应答。
[0]	GC	广播呼叫功能使能控制 0 = 禁用广播呼叫功能 1 = 使能广播呼叫功能

I²C 从机地址掩码寄存器 (I2CADMx)

寄存器	偏移	R/W	描述	复位值
I2CADM0	I2Cx_BA+0x24	R/W	I ² C 从机地址掩码寄存器0	0x0000_0000
I2CADM1	I2Cx_BA+0x28	R/W	I ² C 从机地址掩码寄存器1	0x0000_0000
I2CADM2	I2Cx_BA+0x2C	R/W	I ² C 从机地址掩码寄存器2	0x0000_0000
I2CADM3	I2Cx_BA+0x30	R/W	I ² C 从机地址掩码寄存器3	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
I2CADM							保留

位	描述	
[31:8]	保留	保留
[7:1]	I2CADM	I²C 地址掩码 0 = 禁止 I ² C地址掩码 (接收到的相应地址必须完全符合地址寄存器) 1 = 使能 I ² C地址掩码 (不关心收到的相应地址位)
[0]	保留	保留

I²C 唤醒控制寄存器 (I2CWKUPCON)

寄存器	偏移	R/W	描述	复位值
I2CWKUPCON	I2Cx_BA+0x3C	R/W	I ² C 唤醒控制寄存器	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
							WKUPEN

位	描述	
[31:1]	保留	保留
[0]	WKUPEN	I²C 唤醒功能使能控制 1 =使能 I ² C 唤醒功能 0 =禁用 I ² C 唤醒功能

I²C 唤醒状态寄存器 (I2CWKUPSTS)

寄存器	偏移	R/W	描述	复位值
I2CWKUPSTS	I2Cx_BA+0x40	R/W	I ² C 唤醒状态寄存器	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
							WKUPIF

位	描述	
[31:1]	保留	保留
[0]	WKUPIF	I²C 唤醒中断标志 当芯片从睡眠模式由 I ² C 唤醒时，该位被置为1。软件可以写1清零。

5.14.10 NUC029FAE 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:				
I2C_BA = 0x4002_0000				
I2CON	I2C_BA+0x00	R/W	I ² C 控制寄存器	0x0000_0000
I2CADRR0	I2C_BA+0x04	R/W	I ² C 从机地址寄存器 0	0x0000_0000
I2CDAT	I2C_BA+0x08	R/W	I ² C 数据寄存器	0x0000_0000
I2CSTATUS	I2C_BA+0x0C	R	I ² C 状态寄存器	0x0000_00F8
I2CLK	I2C_BA+0x10	R/W	I ² C 时钟分频寄存器	0x0000_0000
I2CTOC	I2C_BA+0x14	R/W	I ² C 超时计数器寄存器	0x0000_0000
I2CADDR0	I2C_BA+0x04	R/W	I ² C 从机地址寄存器 0	0x0000_0000
I2CADDR1	I2C_BA+0x18	R/W	I ² C 从机地址寄存器 1	0x0000_0000
I2CADDR2	I2C_BA+0x1C	R/W	I ² C 从机地址寄存器 2	0x0000_0000
I2CADDR3	I2C_BA+0x20	R/W	I ² C 从机地址寄存器 3	0x0000_0000
I2CADM0	I2C_BA+0x24	R/W	I ² C 从机地址掩码寄存器 0	0x0000_0000
I2CADM1	I2C_BA+0x28	R/W	I ² C 从机地址掩码寄存器 1	0x0000_0000
I2CADM2	I2C_BA+0x2C	R/W	I ² C 从机地址掩码寄存器 2	0x0000_0000
I2CADM3	I2C_BA+0x30	R/W	I ² C 从机地址掩码寄存器 3	0x0000_0000
I2CCON2	I2C_BA+0x3C	R/W	I ² C 控制寄存器 2	0x0000_0000
I2CSTATUS2	I2C_BA+0x40	R/W	I ² C 状态寄存器 2	0x0000_0000

5.14.11 NUC029FAE寄存器描述

I²C控制寄存器 (I2CON)

寄存器	偏移	R/W	描述	复位值
I2CON	I2C_BA+0x00	R/W	I ² C 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
EI	ENS1	STA	STO	SI	AA	保留	

位	描述	
[31:8]	保留	保留
[7]	EI	I²C中断使能控制 0 = 禁用 I ² C 中断 1 = 使能 I ² C 中断
[6]	ENS1	I²C控制器使能控制 0 = 禁用 I ² C 控制器 1 = 使能 I ² C 控制器 置位该位以使能I ² C串行功能控制器。当ENS1=1时，I ² C串行功能使能。多功能引脚的功能必须先设置为I ² C功能。
[5]	STA	I²C 起始控制位 设置STA为逻辑1来进入主机模式，当总线空闲时，I ² C硬件会发出起始或者重复起始信号到总线上。
[4]	STO	I²C 停止控制位 在主机模式下，置位STO将向总线传输停止条件，然后I ² C硬件会检查总线状态。一旦检测到停止条件，该位将被硬件自动清零。在从机模式下，置位STO会将I ² C硬件复位至“无地址”的从机模式。这意味着该设备不再处于从机接收模式，不能从主机发送设备接收数据。
[3]	SI	I²C中断标志 当一个新的I ² C状态出现在 I2CSTATUS 寄存器时，硬件置位SI标志。如果EI (I2CON [7])已经置位，将发生I ² C中断。SI 必须由软件清零。向该位写1清零。

位	描述	
[2]	AA	接收应答控制位 若在地址或数据接收之前, AA=1, 在SCL线上的应答时钟脉冲期间返回应答信号 (SDA为低) 的条件是: 1、从机应答主机发送的地址信息; 2、接收设备应答发送设备发送的数据。若在地址或数据接收之前, AA=0, 在SCL线上的应答时钟脉冲期间会返回非应答信号 (SDA为高)。
[1:0]	保留	保留

I²C 数据寄存器(I2CDAT)

寄存器	偏移	R/W	描述	复位值
I2CDAT	I2C_BA+0x08	R/W	I ² C 数据寄存器	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
I2CDAT							

位	描述	
[31:8]	保留	保留
[7:0]	I2CDAT[7:0]	I²C 数据位 Bit [7:0] 为8位I ² C串行端口的传输数据。

I²C 状态寄存器(I2CSTATUS)

寄存器	偏移	R/W	描述	复位值
I2CSTATUS	I2C_BA+0x0C	R	I ² C 状态寄存器	0x0000_00F8

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
I2CSTATUS							

位	描述	
[31:8]	保留	保留
[7:0]	I2CSTATUS[7:0]	I²C 状态位 低三位始终是0；高5位包含状态码。总共有26种状态码：当I2CSTATUS的值是F8H，不会产生中断请求；其它的所有的I2CSTATUS值都对应定义的I ² C的状态。当进入这些状态时会产生一个状态中断请求(SI=1)。一个有效的状态码在SI被硬件设为'1'后的一个周期内出现在I2CSTATUS中，并保持稳定直至SI被软件复位后的下一个周期。另外，状态码是00H时表示总线错。当‘起始’或‘结束’信号出现在不正确的帧位置时会产生总线错误。比如在串行传输地址字节、数据字节或者应答位时都可能出现总线错误。

I²C 时钟分频寄存器 (I2CLK)

寄存器	偏移	R/W	描述	复位值
I2CLK	I2C_BA+0x10	R/W	I ² C 时钟分频寄存器	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
I2CLK							

位	描述	
[31:8]	保留	保留
[7:0]	I2CLK[7:0]	I²C 时钟分频位 I ² C 时钟比特率: I ² C 数据波特率 = PCLK / (4x (I2CLK+1)). 注: I2CLK 的最小值为 4。

I²C 超时计数器寄存器 (I2CTOC)

寄存器	偏移	R/W	描述	复位值
I2CTOC	I2C_BA+0x14	R/W	I ² C 超时计数器寄存器	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					ENTI	DIV4	TIF

位	描述	
[31:3]	保留	保留
[2]	ENTI	超时计数使能控制 0 = 禁用超时计数器 1 = 使能超时计数器 注: 当14位超时计数器被使能时，SI被清0后，14位超时计数器开始计数。写1到SI中会使计数器复位，SI被清零后计数器又重新开始计数。
[1]	DIV4	超时计数输入时钟除以4 0 = 禁用超时计数输入时钟除以4 1 = 使能超时计数输入时钟除以4 注: 使能后，超时时间延长4倍。
[0]	TIF	超时标志 当I ² C超时发生时，这个位由硬件置位。如果I ² C中断同时使能位(EI)设置为1，将中断CPU。 注: 软件写"1"清零。

I²C 从机地址寄存器 (I2CADDRx)

寄存器	偏移	R/W	描述	复位值
I2CADDR0	I2C_BA+0x04	R/W	I ² C 从机地址寄存器 0	0x0000_0000
I2CADDR1	I2C_BA+0x18	R/W	I ² C 从机地址寄存器 1	0x0000_0000
I2CADDR2	I2C_BA+0x1C	R/W	I ² C 从机地址寄存器 2	0x0000_0000
I2CADDR3	I2C_BA+0x20	R/W	I ² C 从机地址寄存器 3	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
I2CADDR							GC

位	描述	
[31:8]	保留	保留
[7:1]	I2CADDR	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7位作为芯片本身的地址。如果任一地址符合，I ² C 硬件将会自动应答。
[0]	GC	广播呼叫功能使能控制 0 = 禁用广播呼叫功能 1 = 使能广播呼叫功能

I²C 从机地址掩码寄存器 (I2CADMx)

寄存器	偏移	R/W	描述	复位值
I2CADM0	I2C_BA+0x24	R/W	I ² C 从机地址掩码寄存器 0	0x0000_0000
I2CADM1	I2C_BA+0x28	R/W	I ² C 从机地址掩码寄存器 1	0x0000_0000
I2CADM2	I2C_BA+0x2C	R/W	I ² C 从机地址掩码寄存器 2	0x0000_0000
I2CADM3	I2C_BA+0x30	R/W	I ² C 从机地址掩码寄存器 3	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
I2CADM							保留

位	描述	
[31:8]	保留	保留
[7:1]	I2CADM	I²C 地址掩码位 0 = 禁止 I ² C地址掩码 (接收到的相应地址必须完全符合地址寄存器) 1 = 使能 I ² C地址掩码 (不关心收到的相应地址位)
[0]	保留	保留

I²C 控制寄存器 2 (I2CCON2)

寄存器	偏移	R/W	描述	复位值
I2CCON2	I2C_BA+0x3C	R/W	I ² C 控制寄存器 2	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			UNDER_INTEN	OVER_INTEN	NOSTRETCH	TWOFF_EN	WAKEUPEN

位	描述	
[31:5]	保留	保留
[4]	UNDER_INTEN	I²C UNDER RUN 中断控制位 当TWOFF 使能，且在发送FIFO发生under run事件时，置位 UNDER_INTEN将会发送中断到系统。 0 = 禁用 1 = 使能
[3]	OVER_INTEN	I²C OVER RUN 中断控制位 当TWOFF 使能，且在发送FIFO发生over run事件时，置位 OVER_INTEN 将会发送中断到系统。 0 = 禁用 1 = 使能
[2]	NOSTRETCH	I²C 总线不受控 0=如果主机模式下，SI没有被清除，I ² C SCL总线由硬件控制。 1=如果主机模式下，SI没有被清除，I ² C SCL总线不由硬件控制。
[1]	TWOFF_EN	二级缓存使能控制 0 = 禁用 1 = 使能 使能I ² C发送或接收缓存，用来提升I ² C总线的性能。如果该位为1，用来重复起始的STA或STO应该在当前SI被清除后置位。假如要传送4个数据然后停止，STO位应该在第3个数据中断被清除后置位，此时第4个数据可以被传送，I ² C将会在第4个数据传送完成后停止。

位	描述	
[0]	WAKEUPEN	<p>唤醒使能控制 0 = 禁用 I²C 唤醒功能 1 = 使能 I²C 唤醒功能 当系统进入低功耗模式下，且接收到的数据与地址寄存器中的一个地址相匹配，就可以通过 I²C 总线将系统唤醒。</p>

I²C 状态寄存器2 (I2CSTATUS 2)

寄存器	偏移	R/W	描述	复位值
I2CSTATUS2	I2C_BA+0x40	R/W	I ² C 状态寄存器 2	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			UNDERUN	OVERUN	EMPTY	FULL	WAKEUPIF

位	描述	
[31:5]	保留	保留
[4]	UNDERUN	I²C UNDER RUN 状态位 当TWOFF_EN = 1时， 该位表示发送缓存处于under run状态。
[3]	OVERUN	I²C OVER RUN 状态位 当TWOFF_EN = 1时， 该位表示接收缓存处于over run状态。
[2]	EMPTY	I²C 二级缓存空 当TWOFF_EN = 1时， 该位表示接收缓存是否为空。
[1]	FULL	I²C 二级缓存满 当TWOFF_EN = 1时， 该位表示接收缓存是否为满。
[0]	WAKEUPIF	I²C 唤醒中断标志 当芯片从低功耗模式下被I ² C唤醒时， 该位设置为1。软件写1清零。

5.15 串行外设接口 (SPI)

5.15.1 概述

串行外设接口(SPI)是一个工作于全双工模式下的同步串行数据通讯协议。设备通过4线双向接口工作于主机/从机模式进行通讯。NuMicro® NUC029系列包括最多2组SPI控制器，将从外设接收到的数据进行串并转换，或将要发送到外设的数据进行并串转换。每组SPI控制器都可被设置成主机；也可设置为被片外主机设备控制的从机。

5.15.2 特性

- 多达两组SPI控制器
- 支持主机/从机模式
- 传输字长度可配置为8到32位
- 提供独立的4阶深度的发送和接收FIFO缓冲
- 支持MSB优先或LSB优先的传输顺序
- 支持字节重排序功能
- 支持字节或字休眠模式
- 支持从机3线模式
- 支持PLL时钟源(只有NUC029xAN)

5.15.3 方框图

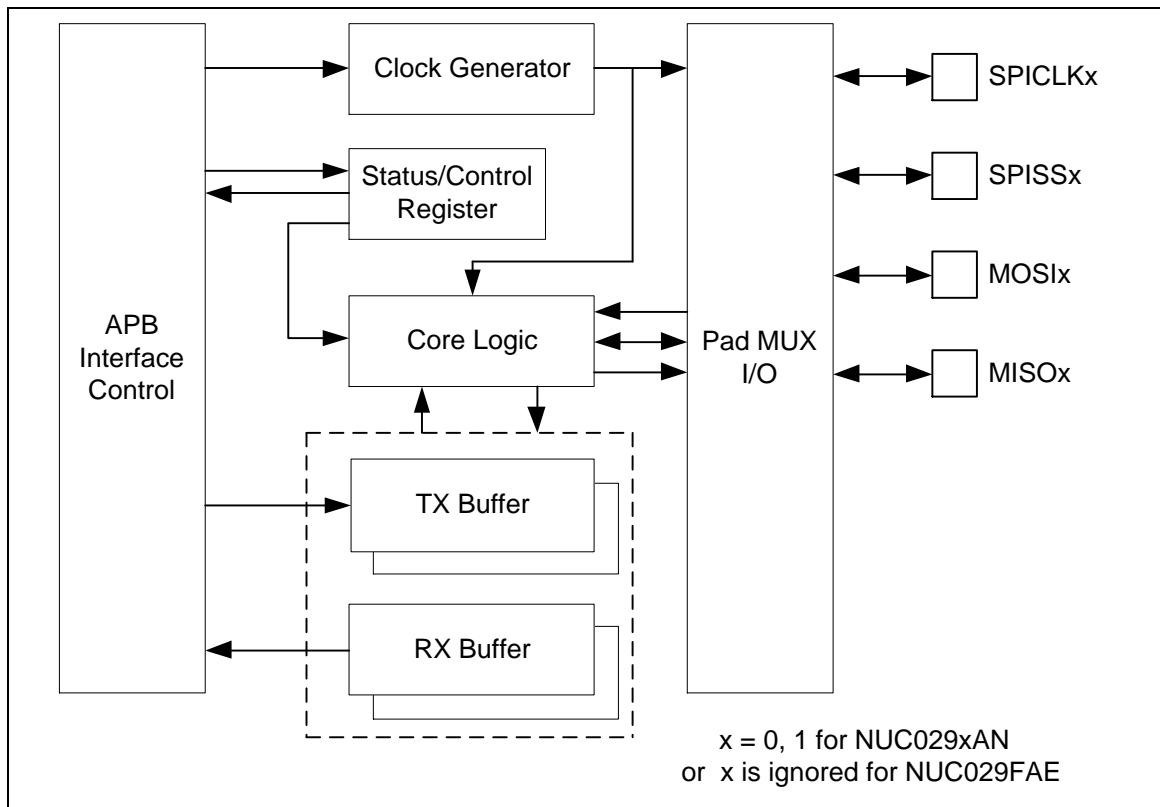


图 5-122 SPI 方框图

5.15.4 基本配置

NUC029xAN:

SPI引脚功能在寄存器P0_MFP和P1_MFP配置。SPI0和SPI1外设时钟可以在APBCLK[12]和APBCLK[13]使能。SPI外设时钟源在CLKSEL1[4]和CLKSEL1[5]选择。

NUC029FAE:

SPI引脚功能在寄存器P0_MFP配置。SPI外设时钟可以在SPI_EN(APBCLK[12])位使能，其时钟源可以在SPI_S(CLKSEL1[4])位选择。

5.15.5 功能描述

5.15.5.1 术语

SPI外设时钟和SPI总线时钟

SPI控制器需要SPI外设时钟驱动SPI逻辑单元实现数据传输。SPI总线时钟是SPICLKx引脚上的时钟。SPI外设时钟频率由设置的时钟源，BCn(SPI_CNTRL2[31])选项和时钟分频器

(DIVIDER(SPI_DIVIDER[7:0]))决定。CLKSEL1寄存器的SPIx_S位决定了SPI外设时钟的时钟源。BCn位设置为0，SPI时钟频率的计算方法和以前的产品兼容。SPI_DIVIDER寄存器设置的DIVIDER值决定了时钟频率的分频值。

在SPI主机模式，SPI外设时钟等于SPI总线时钟。

在SPI从机模式，SPI总线时钟由外围主机设备提供。从机设备的SPI外设时钟频率必须比连接在一起的主机设备的总线时钟快。不管主机模式还是从机模式，SPI外设时钟不能比APB时钟频率快。

主机/从机模式

SPI控制器可以通过SLAVE位(SPI_CNTRL[18])设置为主机或从机模式来与外部的SPI从机或主机设备通讯。主机或从机模式的应用方框图分别如下：

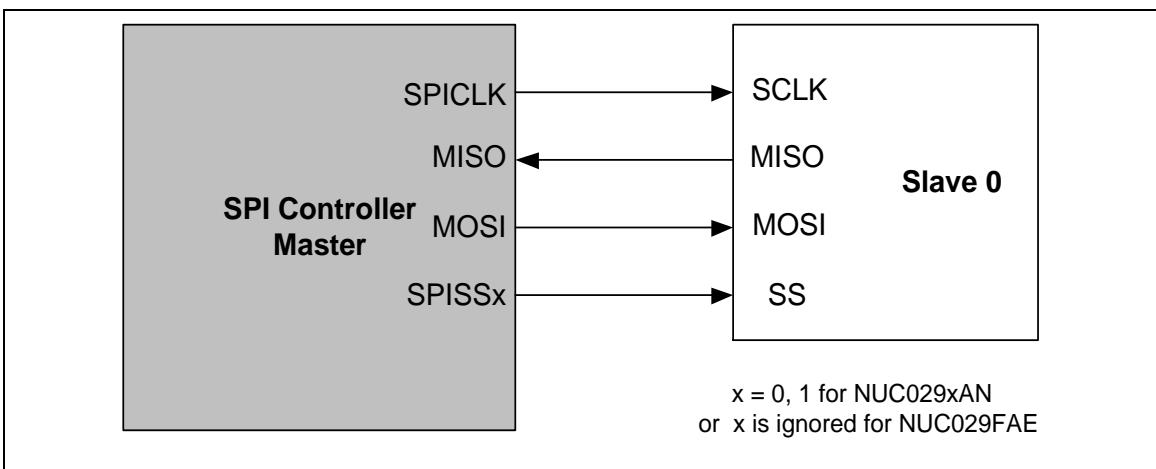


图 5-123 SPI 主机模式应用方框图

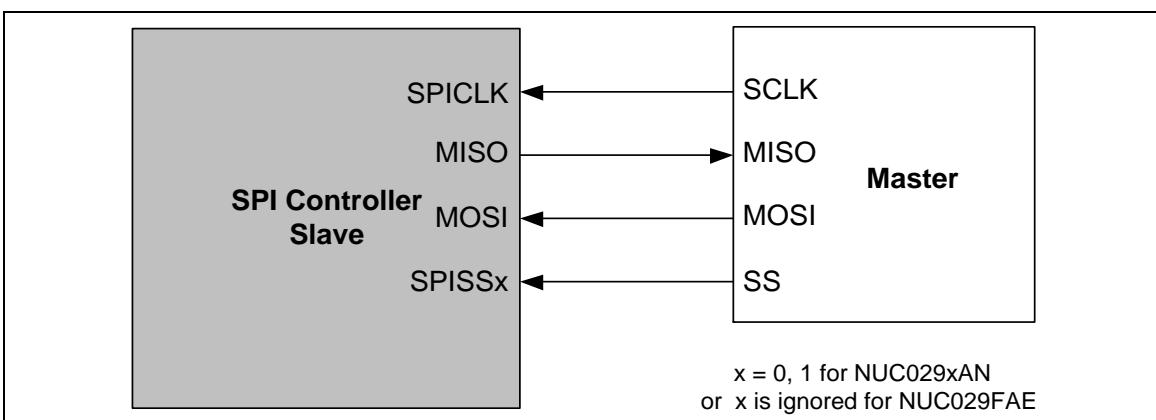


图 5-124 SPI 从机模式应用方框图

时钟极性

CLKP(SPI_CNTRL[11])位决定总线时钟空闲状态。如果CLKP = 1，SPICLK输出空闲状态为高；如果CLKP = 0，SPICLK输出空闲状态为低。

发送/接收位长度

传输字的位长度在TX_BIT_LEN(SPI_CNTRL[7:3])位配置。在发送和接收中，传输字的位长度均可配置高达32位。

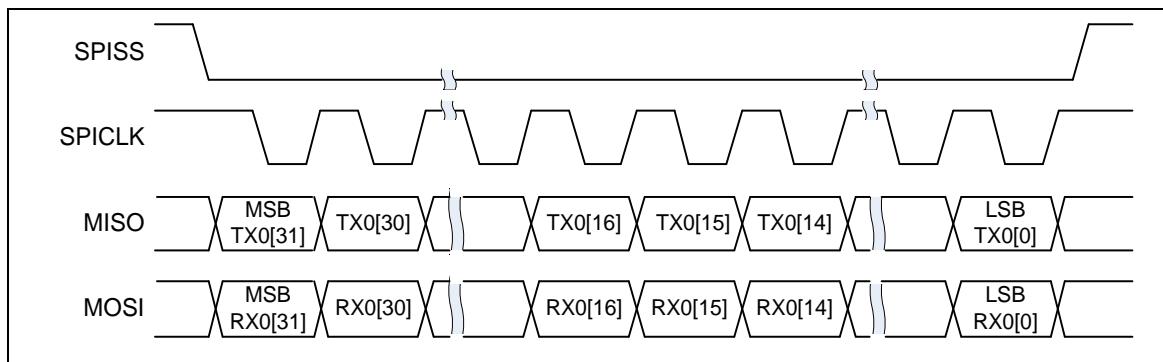


图 5-125 一次传输32位

LSB 优先

LSB位(SPI_CNTRL[10])决定传输/发送数据时，数据传输机制是LSB优先还是MSB优先。

发送边沿

TX_NEG(SPI_CNTRL[2])位决定数据在总线时钟SPICLK的下降沿还是上升沿发送。

接收边沿

RX_NEG(SPI_CNTRL[1])位决定数据在总线时钟SPICLK的下降沿还是上升沿接收。

注: TX_NEG和RX_NEG的设置是互斥的，也就是说，不能发送和接收数据在同一个时钟沿。

字休眠

在主机模式下，SP_CYCLE(SPI_CNTRL[15:12])在两个连续的传输字之间提供一个可配置的休眠间隔。休眠间隔为上一次传输字的最后一个时钟沿到接下来这次传输字的第一个时钟沿。SP_CYCLE 默认值为 0x3 (3.5 个总线时钟周期)，如果软件禁止 FIFO 模式，SP_CYCLE(SPI_CNTRL[15:12]) 设置对字休眠间隔将不起作用。

从机选择

在主机模式，每一个SPI控制器可以通过从机选择引脚SPISS_x驱动一个片外的从机设备。在从机模式，片外主机设备通过SPISS_x输入引脚驱动从机选择信号到SPI控制器。在主机/从机模式，从机选择信号的有效电平可以在SS_LVL(SPI_SSR[2])位设置为低有效或高有效，SS_LTRIG(SPI_SSR[4])位决定从机选择信号SPISS_x是电平触发还是边沿触发。触发条件的选择取决于连接的从机/主机设备的类型。

在从机模式下，如果SS_LTRIG(SPI_SSR[4])位被配置成电平触发，则寄存器LTRIG_FLAG(SPI_SSR[5])用来表示在一个传输完成后接收到的数据数量和位数量是否满足寄存器TX_BIT_LEN(SPI_CNTRL[7:3])的设定值（传输完成是指当从机选择信号未激活或SPI控制器完成了一次数据传输时，单位传输中断标志置1）。

电平触发 / 边沿触发

在从机模式下，从机选择信号可以被配置为电平触发或边沿触发。边沿触发模式，数据从一个有效的从机选择信号边沿开始，到一个无效的从机选择信号边沿结束。当检测到一个无效边沿，寄存器(SPI_CNTRL[16])的单元传输中断标志将设置为1。如果主机没有发送一个无效边沿给从机，传输过程不会完成，从机的单元传输中断标志不会被置位。电平触发模式，当有下面的两个条件中的一个发生，从机的单元传输中断标志将会被置位。第一个条件是传输的数据位与TX_BIT_LEN(SPI_CNTRL[7:3])的设定值相匹配时，从机的单元传输中断标志将会被置位。第二个条件，如果主机正在传输期间的时候设置从机选择管脚为无效电平，将会强制从机设备终结当前传输，而不管已经传输了多少位数据，从机的单元传输中断标志将会被置位。用户可以读取LTRIG_FLAG(SPI_SSR[5])位来检查数据是否已经全部传完。

5.15.5.2 自动从机选择

在主机模式下，如果AUTOSS(SPI_SSR[3])设置为1，从机选择信号将会自动产生，并根据SSR(SPI_SSR[0])是否使能，将从机选择信号输出到SPISSx管脚上。这意味着当通过设置GO_BUSY(SPI_CNTRL[0])位来开始数据传输时，从机选择信号将由SPI控制器自动设置为有效状态，在数据传输结束后自动被设置为无效状态。如果AUTOSS(SPI_SSR[3])位被清零，从机选择输出信号需要通过手工置位/清除SSR(SPI_SSR[0])，从而使从机进入激活或非激活状态。从机选择输出信号的激活电平状态由SS_LVL(SPI_SSR[2])位来定义。

5.15.5.3 字节重排序功能

当传输设置为MSB优先(LSB = 0)且使能字节重排序功能时，存储在Tx和Rx缓存中的数据将会按照[Byte0, Byte1, Byte2, Byte3]重新排序，前提是位长度被配置为32位(TX_BIT_LEN(SPI_CNTRL[7:3]) = 0)。发送/接收数据的顺序将会变为Byte0, Byte1, Byte2, Byte3。如果TX_BIT_LEN(SPI_CNTRL[7:3])设置为24位，Tx和Rx缓存中的数据将会按照[未知Byte, Byte0, Byte1, Byte2]重新排序，SPI控制器将会按照Byte0, Byte1, Byte2的顺序发送/接收数据。每一个字节将会按照MSB优先的规则来发送/接收。16位模式的规则与上面相同。字节重排序功能只在TX_BIT_LEN(SPI_CNTRL[7:3])为16, 24和32位时适用。

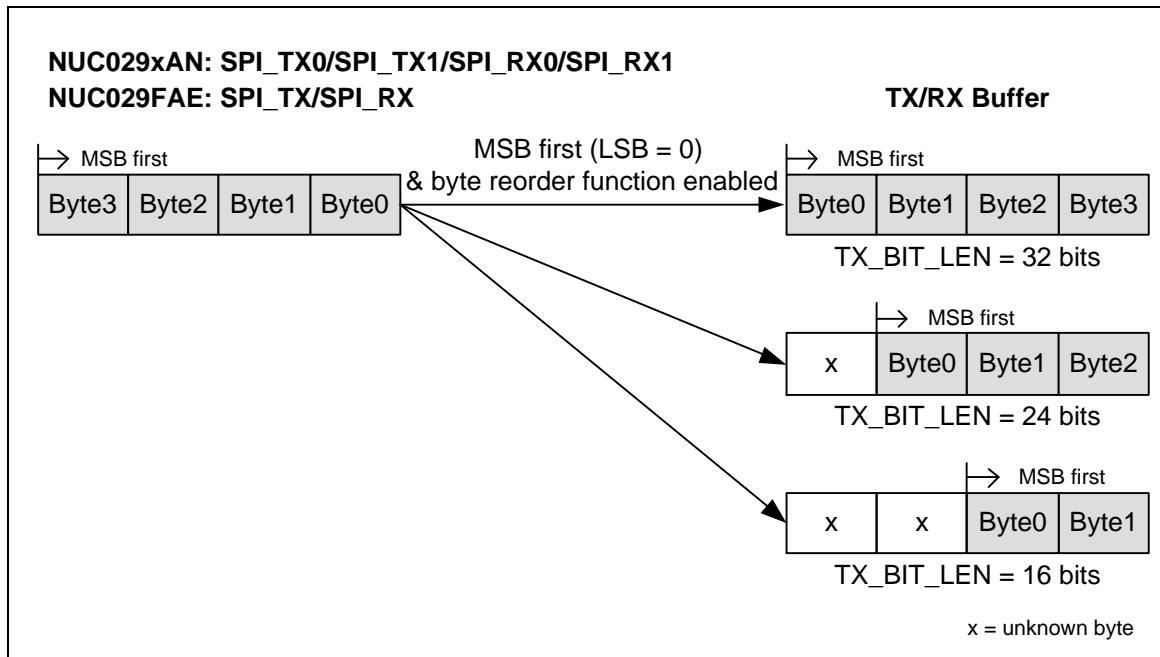


图 5-126 字节重排序

5.15.5.4 字节休眠功能

字节休眠功能与字休眠功能都在SP_CYCLE中进行设置。在主机模式下，如果通过设置SPI_CTRL[19]为1来使能字节重排序功能，硬件将会在两个连续传输字节之间插入一个0.5 ~ 15.5个串行时钟周期的休眠间隔。TX_BIT_LEN(SPI_CTRL[7:3])可以设置为16, 24和32位。

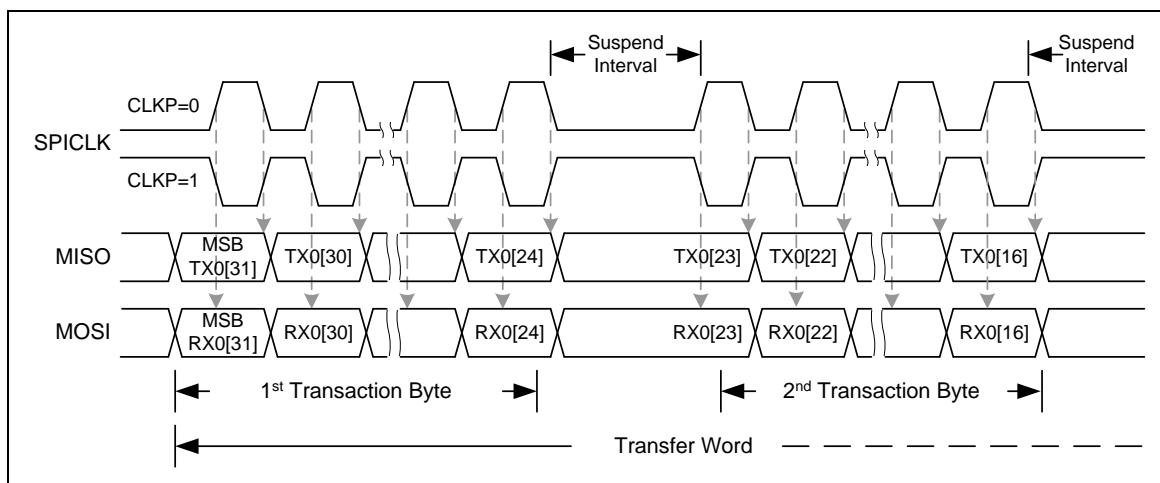


图 5-127 字节休眠时序波形

5.15.5.5 从机3线模式

当NOSLVSEL(SPI_CTRL2[8])位被软件置1使能从机3线模式时，在从机模式下SPI控制器可以在没有从机选择信号下工作。NOSLVSEL(SPI_CTRL2[8])位仅在从机模式下有效。在与一个主机通讯时，仅需要三个管脚，SPICLK, MISO和MOSI。SPISS脚可以配置成一个GPIO。当

NOSLVSEL(SPI_CNTRL2[8])位被设为1时，在GO_BUSY(SPI_CNTRL[0])位被置为1后，SPI从机将开始准备发送和接收数据。当接收数据位的个数满足TX_BIT_LEN(SPI_CNTRL[7:3])长度定义时，单元传输中断标志IF(SPI_CNTRL[16])将会置位。

注：在从机3线模式下，SS_LTRIG(SPI_SSR[4])位应该被设置为1。

5.15.5.6 FIFO 模式

当FIFO(SPI_CNTRL[21])被设为1时，SPI控制器支持FIFO模式。SPI控制器配备了4个32位发送和接收FIFO缓存。

发送FIFO缓存是一个4层深度，32位宽，先进先出的寄存器缓存。数据可以通过软件写SPI_TX0(NUC029xAN)或SPI_TX(NUC029FAE)寄存器写入发送FIFO缓存。存储在发送FIFO缓存的数据会被传输控制逻辑读取并发送出去。如果4层发送FIFO缓存满了，TX_FULL(SPI_STATUS[27])位会被置1。当SPI传输逻辑单元发送FIFO缓存中的最后一个数据时，那么4层发送FIFO缓存为空，TX_EMPTY(SPI_STATUS[26])位被置1。注意最后一笔传输还在进行时TX_EMPTY(SPI_STATUS[26])标志已被置1。

接收FIFO缓存也是一个4层深度，32位宽，先进先出的寄存器缓存。接收控制逻辑存储接收到的数据到该缓存。FIFO缓存数据可以通过软件从SPI_TX0(NUC029xAN)或SPI_TX(NUC029FAE)寄存器读取。FIFO还带有一些相关状态位，如RX_EMPTY(SPI_STATUS[24])和RX_FULL(SPI_STATUS[25])，用来表明当前FIFO缓存的状态。

在FIFO模式下，发送和接收阀值可以通过软件设置TX_THRESHOLD(SPI_FIFO_CTL[29:28])和RX_THRESHOLD(SPI_FIFO_CTL[25:24])来设定。当存储在发送FIFO缓存的有效数据计数小于或等于TX_THRESHOLD(SPI_FIFO_CTL[29:28])设定，TX_INTSTS(SPI_STATUS[4])位会被置1。当存储在接收FIFO缓存的有效数据计数大于RX_THRESHOLD(SPI_FIFO_CTL[25:24])设定，RX_INTSTS(SPI_STATUS[0])位会被置1。

在FIFO模式下，4个数据可以事先通过软件写入SPI发送FIFO缓存。当SPI控制器工作在FIFO模式下时，SPI_CTL寄存器中的GO_BUSY(SPI_CNTRL[0])位由硬件控制，SPI_CNTRL寄存器的内容不会被软件修改，除非FIFOEN位被清零使FIFO模式禁止。

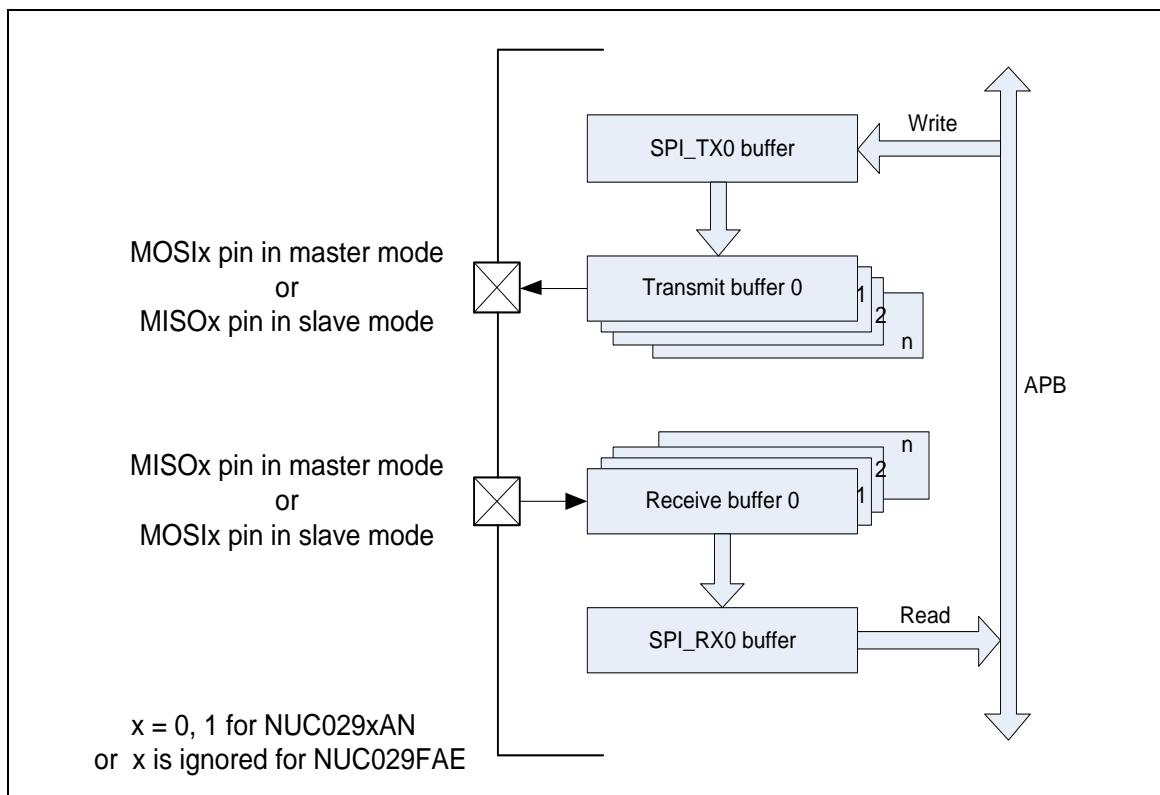


图 5-128 FIFO 模式框图

SPI 主机模式和 FIFO 模式使能

在主机模式下，进行发送操作中，当 FIFO 位设置为 1，软件写第一个数据到 SPI_TX0(NUC029xAN) 或 SPI_TX(NUC029FAE) 寄存器时，TX_EMPTY(SPI_CTRL[26]) 标志将会被清零。只要发送 FIFO 缓存非空，发送操作立即开始。用户可以立即写入下一个要发送的数据到 SPI_TX0(NUC029xAN) 或 SPI_TX(NUC029FAE) 寄存器中。在 FIFO 模式下，SPI 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 SP_CYCLE(SPI_CTRL[15:12]) 的设定值决定。只要 TX_FULL(SPI_CTRL[27]) 标志为 0，用户就可以写数据到 SPI_TX0(NUC029xAN) 或 SPI_TX(NUC029FAE) 寄存器。

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，SPI_TX0(NUC029xAN) 或 SPI_TX(NUC029FAE) 寄存器没有被更新，则传输停止。

在主机接收操作中，串行数据从 MISOx 管脚接收并被存储在接收 FIFO 缓存。当接收 FIFO 缓存中包含未读的数据时，RX_EMPTY(SPI_CTRL[24]) 将会被清除为 0。只要 RXEMPTY(SPI_CTRL[24]) 标志为 0，软件就可以从 SPI_RX0(NUC029xAN) 或 SPI_RX(NUC029FAE) 寄存器读取接收到的数据。如果接收 FIFO 缓存包含 4 个未读数据，RX_FULL(SPI_CTRL[25]) 标志将会被设置为 1。此时，SPI 控制器将会停止接收数据，直到软件读取 SPI_RX0(NUC029xAN) 或 SPI_RX(NUC029FAE) 寄存器。

SPI 从机模式和 FIFO 模式使能

从机模式下，当 FIFO 位被设置为 1，GO_BUSY(SPI_CTRL[0]) 位将会被硬件自动设置为 1。如果

用户想停止从机模式SPI数据传输，FIFO位和GO_BUSY(SPI_CNTRL[0])位必须用软件都清零。

在从机发送操作中，当软件写数据到SPI_TX0(NUC029xAN)或SPI_TX(NUC029FAE)寄存器，数据将会被加载到发送FIFO缓存，且TX_EMPTY(SPI_CNTRL[26])标志将会被清0。当从设备从主机接收到时钟信号，发送操作将会开始。只要TX_FULL(SPI_CNTRL[27])标志为0，软件就可以写数据到SPI_TX0(NUC029xAN)或SPI_TX(NUC029FAE)寄存器。在所有数据都被 SPI 发送逻辑单元发送出去，且软件没有更新 SPI_TX0(NUC029xAN) 或 SPI_TX(NUC029FAE) 寄存器，TX_EMPTY(SPI_CNTRL[26])标志将会被设置为1。

在从机接收操作中，串行数据从MOSIx管脚被接收，并被存储到SPI_RX0(NUC029xAN)或SPI_RX(NUC029FAE)寄存器。接收机制与主机模式接收操作类似。

5.15.5.7 中断

SPI单位传输中断

当SPI控制器完成一个单位传输，单位传输中断标志IF(SPI_CNTRL[16])将会被置位。如果中断使能位IE(SPI_CNTRL[17])被置位，则单位传输中断事件将会给CPU产生中断。单位传输中断标志位只能写1清零。

SPI从机3线模式起始中断

在从机3线模式下，当从机检测到SPI时钟信号时，3线模式会产生起始中断标志，SLV_START_INTSTS(SPI_CNTRL2[11])将会被置为1。如果SSTA_INTEN(SPI_CNTRL2[10])被设置为1，SPI控制器将会触发一个中断。如果接收到的数据位小于TX_BIT_LEN(SPI_CNTRL[7:3])的设定要求，在由用户定义的期望的时间内再没有串行时钟输入，用户可以设置SLV_ABORT(SPI_CNTRL2[9])位来中止当前传输。如果软件设置SLV_ABORT(SPI_CNTRL2[9])位为1，单位传输中断标志IF将会被置位。

接收 FIFO 超时中断

在FIFO模式下，有超时功能用于通知用户。如果超时中断使能位SPI_FIFO_CTL[21]置1，在FIFO里有一个接收到的数据，并且在主机模式下超过64个SPI外设时钟周期，或在从机模式下超过576个SPI外设时钟周期没有被软件读取，会发出一个超时中断。

传送 FIFO 中断

在 FIFO 模式下，如果发送 FIFO 缓存的有效数据计数小于或等于 TX_THRESHOLD(SPI_FIFO_CTL[29:28]) 的设定值，发送 FIFO 中断标志会被置1。如果 SPI_FIFO_CTL[3]置1，发送FIFO中断使能，SPI控制器会产生一个发送FIFO中断到系统。

接收 FIFO 中断

在FIFO模式下，如果接收FIFO缓存的有效数据计数大于RX_THRESHOLD(SPI_FIFO_CTL[25:24])的设定值，接收FIFO中断标志会被置1。如果SPI_FIFO_CTL[2]置1，接收FIFO中断使能，SPI控制器会产生一个接收FIFO中断到系统。

5.15.6 时序图

从机选择信号的有效状态可以由SS_LVL(SPI_SSR[2])位和SS_LTRIG(SPI_SSR[4])位来定义。串行时钟(SPICLK)的空闲状态可以通过CLKP(SPI_CNTRL[11])配置为高电平或低电平。传输字段长度在TX_BIT_LEN(SPI_CNTRL[7:3])中定义，发送/接收数据是以MSB或LSB优先由LSB(SPI_CNTRL[10])位定义。用户可以通过设置TX_NEG/RX_NEG(SPI_CNTRL[2]/SPI_CNTRL[1])寄存器来选择发送/接收数据时串行时钟的边沿。四个SPI发送/接收及相关设置如下。

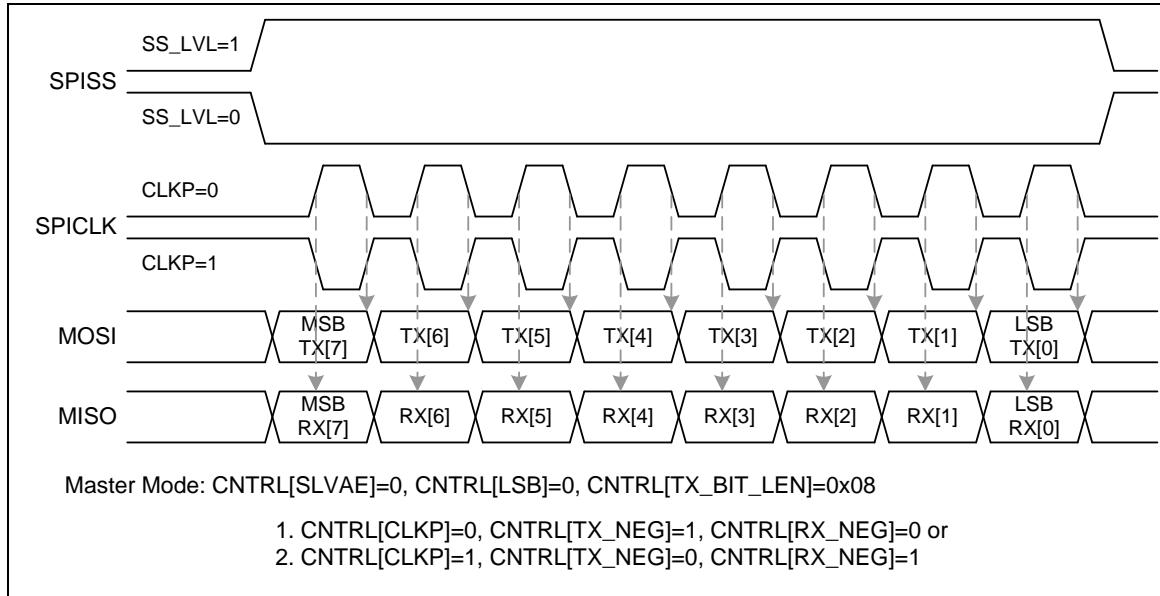


图 5-129 SPI 主机模式时序

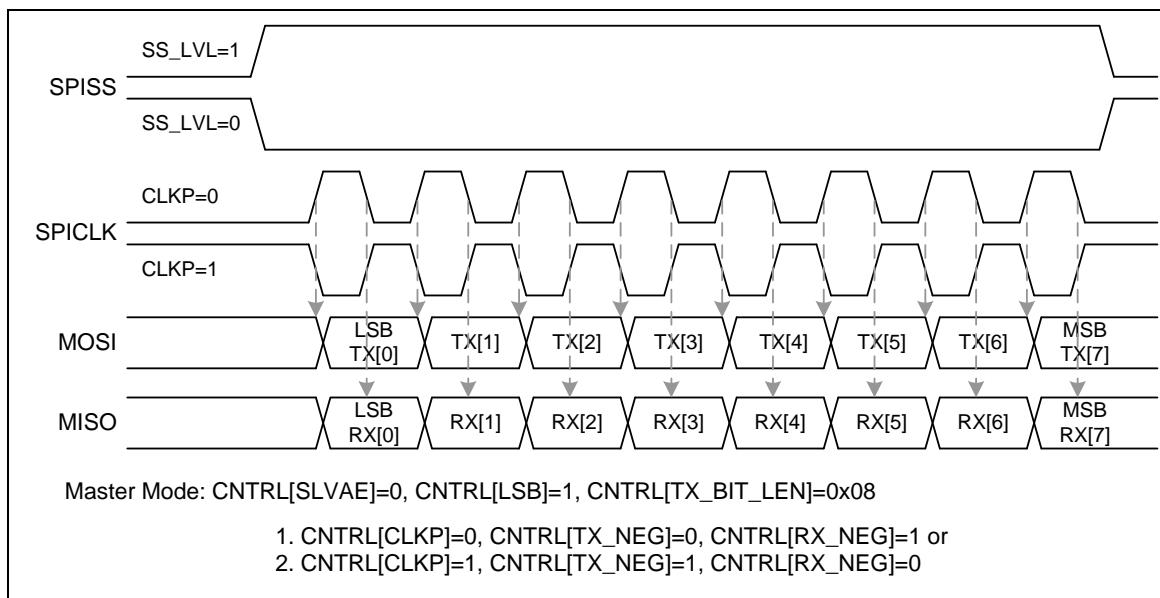


图 5-130 SPI 主机模式时序 (SPICLK 相位相反)

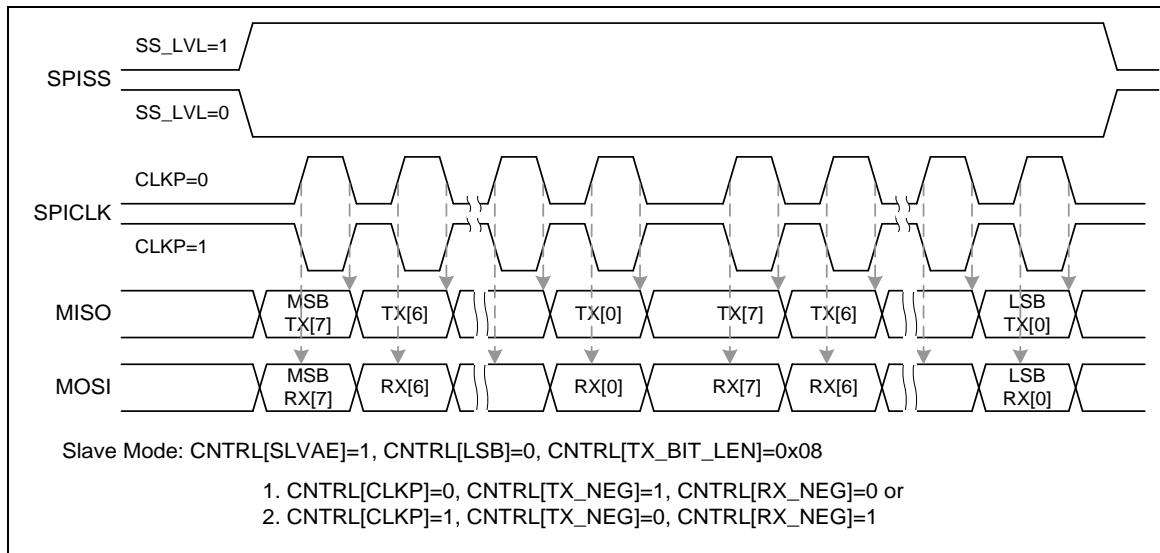


图 5-131 SPI 从机模式时序

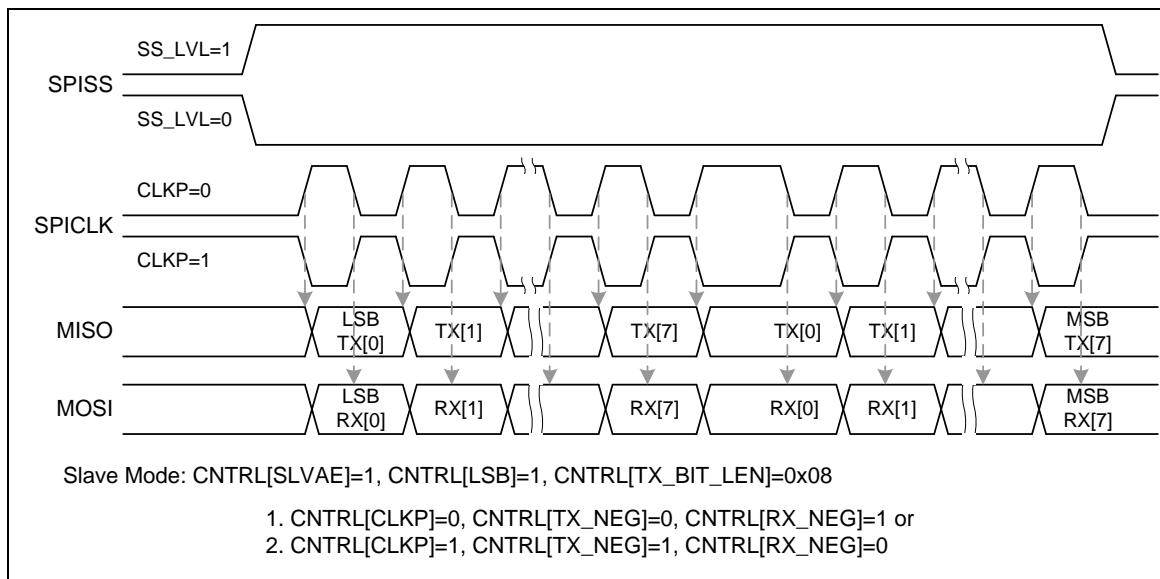


图 5-132 SPI 从机模式时序(SPICLK相位相反)

5.15.7 编程示例

例 1: NUC029xAN SPI 控制器作为主机去访问一个片外从机设备，过程如下：

- 数据在串行时钟上升沿锁存
- 数据在串行时钟下降沿传输
- MSB 先传输
- 数据宽度8位

- SPI_CLK空闲模式为低电平状态
- 每次只发送/接收一个字节
- 与一个片外从机相连，从机选择信号为低电平有效

操作流程如下：

- 1) 设置DIVIDER (SPI_DIVIDER[7:0])寄存器来决定串行时钟输出频率
- 2) 写一个合适的值到SPI_CTRL寄存器来控制SPI主机操作
 1. 通过SLAVE位 (SPI_CTRL[18] = 0) 设置SPI控制器作为主机
 2. 通过CLKP位(SPI_CTRL[11] = 0)设置串行时钟空闲状态为低电平
 3. 通过TX_NEG位(SPI_CTRL[2] = 1)设置数据在串行时钟下降沿传输
 4. 通过RX_NEG位(SPI_CTRL[1] = 0)设置数据在串行时钟上升沿锁存
 5. 通过TX_BIT_LEN(SPI_CTRL[7:3] = 0x08)设置一次字传输的长度为8位
 6. 通过 MSB 位 (SPI_CTRL[10] = 0) 设置 MSB 优先， 不需要在意 SP_CYCLE 位 (SPI_CTRL[15:12])，因为这种情况下它不是FIFO模式
- 3) 给SPI_SSR寄存器写入合适的值来实现主机的相关设置
 1. 清除自动从机选择位AUTOSS(SPI_SSR[3] = 0)
在从机选择激活电平控制位SS_LVL (SPI_SSR[2] = 0)设置低电平来触发从机选择信号。
 2. 通过设置从机选择控制位SSR (SPI_SSR[0])来设置从机选择信号，来激活片外从机设备。
- 4) 若SPI主机尝试发送(写)一个字节的数据到片外从机设备，则将所要发送的数据写入SPI_TX0寄存器。
- 5) 如果 SPI 主机只是要从片外从机设备接收 (读) 一个字节的数据，不必管被传输出去的数据是什么，寄存器SPI_TX0不需要通过软件更新。
- 6) 使能GO_BUSY 位 (SPI_CTRL[0] = 1)来开始SPI数据传输。
- 7) 等待 SPI 中断发生 (如果中断使能位IE位被使能) 或轮询检测GO_BUSY位直到被硬件自动清零。
- 8) 从SPI_RX0[7:0]中读出接收到的一个字节数据。
- 9) 重复步骤 4) 继续其他数据传输或设置SSR[0]位来停止片外从机设备。

例 2: NUC029FAE SPI 控制器作为从机设备和一片外主机设备相连。外设主机通过SPI 接口与从机通信。过程如下：

- 数据在串行时钟上升沿锁存
- 数据在串行时钟下降沿传输
- LSB优先传输
- 数据宽度8位

- SPICLK空闲状态为高电平
- 每次发送/接收一个字节
- 从机选择信号为高电平触发

操作流程如下：

- 1) 设置DIVIDER (SPI_DIVIDER[7:0])来决定从机外设时钟频率。该频率必须大于SPI总线时钟频率。
- 2) 给寄存器SPI_SS写入合适的值来实现从机模式的相关设置。
设置从机选择有效电平位SS_LVL (SPI_SS[2] = 1)和从机选择触发电平位SS_LTRIG (SPI_SS[4] = 1)来选择高电平触发作为从机选择信号。
- 3) 通过设置SPI_CCTRL寄存器来控制 SPI 从机的行为
 1. 通过SLAVE位(SPI_CCTRL[18] = 1)设置SPI控制器为从机设备
 2. 通过CLKP位(SPI_CCTRL[11] = 1)设置串行时钟的空闲状态为高电平
 3. 通过TX_NEG 位 (SPI_CCTRL[2] = 1)设置数据在串行时钟的下降沿传输
 4. 通过RX_NEG 位 (SPI_CCTRL[1] = 0)设置数据在串行时钟的上升沿锁存
 5. 通过TX_BIT_LEN位(SPI_CCTRL[7:3] = 0x08)设置一次字传输的长度为8位
 6. 通过LSB 位 (SPI_CCTRL[10] = 1)设置LSB优先传输
- 4) 如果 SPI 从机要发送(被读)一个字节的数据到片外主机，则将所要发送的数据写入到SPI_TX寄存器
- 5) 如果 SPI 从机只是要从片外主机接收(被写)一字节数据，用户不必关心什么数据将被传输，不需要软件更新SPI_TX寄存器
- 6) 设置GO_BUSY 位 (SPI_CCTRL[0] = 1)来等待片外主机设备的从机选择触发输入和串行时钟输入，以便开始在 SPI 接口传输数据
- 7) 等待 SPI 中断发生（如果中断使能位IE被使能）或轮询检测GO_BUSY位直到被硬件自动清零
- 8) 从寄存器SPI_RX[7:0]中读取接收到的一个字节数据
- 9) 重复步骤4) 来继续其它数据传输或清除GO_BUSY来停止数据传输

5.15.8 NUC029xAN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:				
SPI0_BA = 0x4003_0000				
SPI1_BA = 0x4003_4000				
SPI_CTRL x=0,1	SPIx_BA+0x00	R/W	控制和状态寄存器	0x0500_0004
SPI_DIVIDER x=0,1	SPIx_BA+0x04	R/W	时钟分频器寄存器	0x0000_0000
SPI_SSR x=0,1	SPIx_BA+0x08	R/W	从机选择寄存器	0x0000_0000
SPI_RX x=0,1	SPIx_BA+0x10	R	数据接收寄存器	0x0000_0000
SPI_TX x=0,1	SPIx_BA+0x20	W	数据发送寄存器	0x0000_0000
SPI_CTRL2 x=0,1	SPIx_BA+0x3C	R/W	控制和状态寄存器2	0x0000_0000
SPI_FIFO_CTL x=0,1	SPIx_BA+0x40	R/W	SPI FIFO控制寄存器	0x2200_0000
SPI_STATUS x=0,1	SPIx_BA+0x44	R/W	SPI状态寄存器	0x0500_0000

5.15.9 NUC029xAN寄存器描述

SPI控制和状态寄存器 (SPI_CNTRL)

寄存器	偏移	R/W	描述				复位值
SPI_CNTRL	SPIx_BA+0x00	R/W	控制和状态寄存器				0x0500_0004

31	30	29	28	27	26	25	24
悶隱				TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY
23	22	21	20	19	18	17	16
悶隱		FIFO	悶隱	REORDER	SLAVE	IE	IF
15	14	13	12	11	10	9	8
SP_CYCLE				CLKP	LSB	悶隱	
7	6	5	4	3	2	1	0
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

位	描述	
[31:28]	悶隱	保留.
[27]	TX_FULL	发送 FIFO缓冲满标志(只读) 该位为SPI_STATUS[27]的镜像 0 = 发送FIFO缓冲未满 1 = 发送FIFO缓冲满
[26]	TX_EMPTY	发送 FIFO缓冲空标志(只读) 该位为SPI_STATUS[26]的镜像 0 =发送 FIFO缓冲非空 1 =发送FIFO缓冲空
[25]	RX_FULL	接收缓冲满标志(只读) 该位为SPI_STATUS[25]的镜像 0 = 接收 FIO缓冲未满 1 = 接收FIFO缓冲满
[24]	RX_EMPTY	接收FIFO缓冲空标志(只读) 该位为 SPI_CNTRL[24].的镜像 0 =接收FIFO缓冲非空 1 =接收FIFO缓冲空
[23:22]	悶隱	保留
[21]	FIFO	FIFO模式使能控制 在主机模式下, 如果 FIFO 模式被使能, 在数据被写入8层深度的发送 FIFO 中之后, GO_BUSY将会自动被硬件设置为 1。当所有存储在发送FIFO缓存的数据全部发送出去, GO_BUSY会清零。

		0 = FIFO模式禁用 1 = FIFO模式使能 注: 在使能 FIFO 模式前, 其他相关的设置必须事先设定。
[20:19]	REORDER	字节重排序功能和字节休眠功能选择 00 = 字节重排序功能和字节休眠功能禁用 01 = 字节重排序功能使能。字节休眠间隙由SP_CYCLE设置, SP_CYCLE设为0禁用字节休眠功能。 10 = 保留。 11 = 保留。 注:字节重排序功能只有在TX_BIT_LEN定义为16、24或32位时有效。
[18]	SLAVE	从机模式控制 0 = 主机模式. 1 = 从机模式.
[17]	IE	单元发送中断使能控制 0 = SPI单位发送中断禁用 1 = SPI 单位发送中断使能
[16]	IF	单位发送中断标志 0 = 从该位被清0后, 无发送事务完成 1 = SPI控制器完成一次单元发送 注: 该位写1清0
[15:12]	SP_CYCLE	休眠间隔 (只有主机) 该四位提供一个在传输过程中连续两个发送/接收事务之间可配置的休眠间隔。休眠间隔是指前一个事务的最后一个时钟边沿和后一个事务的第一个时钟边沿之间的间隔。 默认值是 0x3. 休眠间隔长度根据如下公式获得: $(SP_CYCLE[3:0] + 0.5) * \text{SPICLK时钟周期}$ 例: SP_CYCLE = 0x0 ... 0.5 SPICLK时钟周期. SP_CYCLE = 0x1 ... 1.5 SPICLK时钟周期. SP_CYCLE = 0xE ... 14.5 SPICLK时钟周期. SP_CYCLE = 0xF ... 15.5 SPICLK时钟周期.
[11]	CLKP	时钟极性 0 = SPICLK空闲状态为低. 1 = SPICLK空闲状态为高.
[10]	LSB	LSB优先 0 = MSB 优先发送/接收 1 = LSB 优先发送/接收
[9:8]	悵隱	保留.
[7:3]	TX_BIT_LEN	发送位长度 该位域指定一次发送/接收多少位。最小位长为8, 最多可以达到32位。 TX_BIT_LEN = 0x08 ... 8 位.

		TX_BIT_LEN = 0x09 ... 9 位。 TX_BIT_LEN = 0x1F ... 31 位. TX_BIT_LEN = 0x00 ... 32 位.
[2]	TX_NEG	在下降沿发送 0 = 在SPICLK 上升沿发送数据信号 1 = 在SPICLK下降沿发送数据信号
[1]	RX_NEG	在下降沿接收 0 = 在SPICLK上升沿接收数据信号 1 = 在SPICLK下降沿接收数据信号
[0]	GO_BUSY	SPI发送触发和忙状态 在FIFO模式，该位由硬件控制，软件不能更改它。 如果FIFO模式禁用，在数据发送过程中，该位将保持为1；当数据发送完成后，该位自动清0。 0 =如果SPI正在传输时，向该位写0来停止传输。 1 =在主机模式下，写 1 到该位开始 SPI 数据传输。在从机模式下，写 1 到该位表示从机已经准备好与主机进行通信。 注 1: 当FIFO模式被禁用时，在写1到GO_BUSY位前，所有的配置必须事先配置好。 注 2: 在SPI从机模式下，如果FIFO模式被禁止，在数据传输时SPI总线时钟会保持在空闲状态，当从机选择信号处于非活动状态时，GO_BUSY位不会被清零

SPI分频器寄存器(SPI_DIVIDER)

寄存器	偏移	R/W	描述	复位值
SPI_DIVIDER	SPIx_BA+0x04	R/W	时钟分频器寄存器	0x0000_0000

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

位	描述	
[31:8]	悶隱	保留.
[7:0]	DIVIDER	<p>时钟分频器</p> <p>该区域的值是一个频率除数，来决定SPI外设时钟频率f_{spi}，SPI主机的时钟频率由SPICLK管脚输出。该频率可由下面的公式获得：</p> <p>当BCn位（SPI_CCTRL2[31]）为0时：</p> $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER + 1) * 2}$ <p>当BCn为1时，</p> $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER + 1)}$ <p>$f_{SPI_clock_src}$ 为SPI外设时钟源，定义在CLKSEL1寄存器中。</p>

SPI 从机选择寄存器(SPI_SS_R)

寄存器	偏移	R/W	描述	复位值
SPI_SS_R	SPIx_BA+0x08	R/W	从机选择寄存器	0x0000_0000

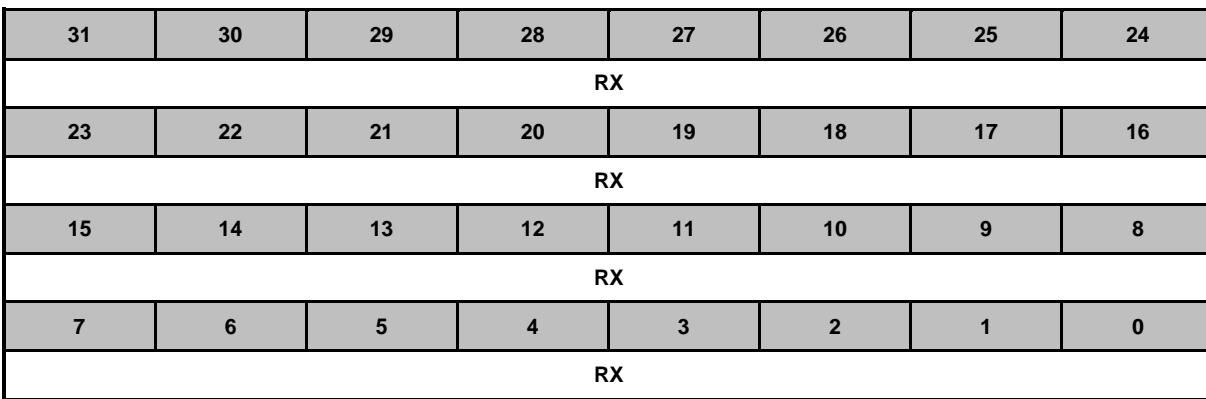
31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱							
7	6	5	4	3	2	1	0
悵隱		LTRIG_FLAG	SS_LTRIG	AUTOSS	SS_LVL	悵隱	SSR

位	描述	
[31:6]	悵隱	保留.
[5]	LTRIG_FLAG	<p>电平触发标志 当SS_LTRIG位在从机模式下被置位，读该位的值可用来表示一次传输完成后接收位的数量是否达到要求 0 = 一次传输的个数或位长度不满足指定的要求 1 = 传输的个数及位长度满足TX_BIT_LEN定义的要求 注: 该位只读，且只在从机模式下有效</p>
[4]	SS_LTRIG	<p>从机选择电平触发使能(只有从机) 0 = 从机选择信号为边沿触发（默认）。SS_LVL决定该信号的有效状态为下降沿还是上升沿。 1 = 从机选择信号为电平触发。SS_LVL决定该信号的有效状态为低电平还是高电平。</p>
[3]	AUTOSS	<p>自动从机选择功能使能 (只有主机) 0 = 如果该位为0，从机选择信号将由设置/清除SSR位激活/不激活。 1 = 如果该位为1，SPISSx将自动产生。当通过设置GO_BUSY开始发送/接收，从机选择信号将会由SPI控制器自动产生，当发送/接收完成后，从机选择信号自动消失。</p>
[2]	SS_LVL	<p>从机选择有效电平 该位定义从机选择信号SPISSx的有效状态。 0 = 从机选择信号SPISSx在低电平/下降沿有效 1 = 从机选择信号SPISSx在高电平/上升沿有效</p>
[1]	悵隱	保留.
[0]	SSR	<p>从机选择控制 (只有主机) 如果AUTOSS位为0 0 = 设置SPISSx为无效状态</p>

		<p>1 = 设置SPISSx为有效状态 如果AUTOSS位为1 0 = 保持SPISSx为无效状态 1 = SPISSx在发送/接收的时间内被自动驱动到有效状态，而其他时间为无效状态。SPI_SS的有效状态类型由SS_LVL位指定</p>
--	--	--

SPI数据接收寄存器 (SPI_RX)

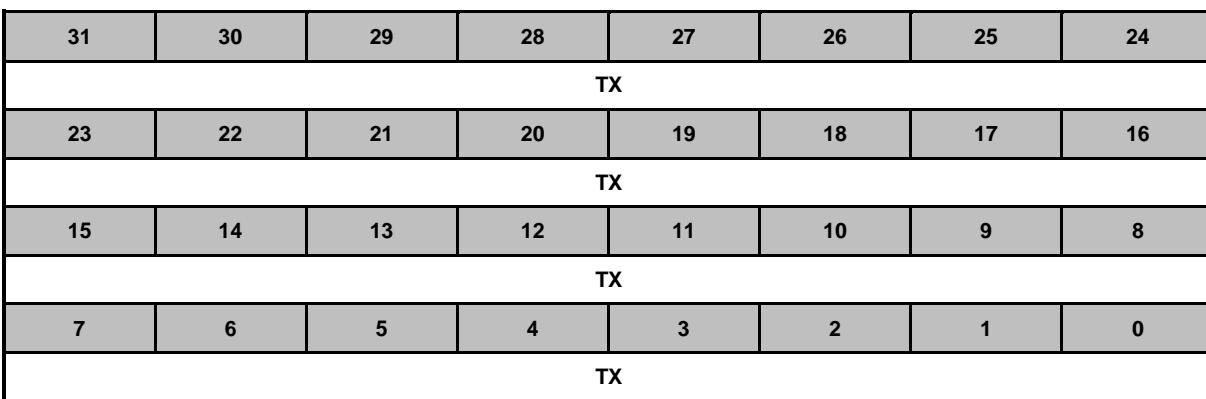
寄存器	偏移	R/W	描述	复位值
SPI_RX	SPIx_BA+0x10	R	数据接收寄存器	0x0000_0000



位	描述	
[31:0]	RX	<p>数据接收寄存器</p> <p>数据接收寄存器内保存最后一次传输所接收到的数据。有效的位取决于SPI_CNTRL寄存器中的发送位长度域。</p> <p>例如，如果X_BIT_LEN配置为0x08且TX_NUM设置为0x0，RX0[7:0]内保存了收到的数据，其它位的数据未知，数据接收寄存器是只读寄存器。</p>

SPI 数据发送寄存器 (SPI_TX)

寄存器	偏移	R/W	描述	复位值
SPI_TX	SPIx_BA+0x20	W	数据发送寄存器	0x0000_0000



位	描述
[31:0]	TX 数据发送寄存器 数据发送寄存器保存下一次将要发送的数据。有效位取决于CNTRL寄存器中的发送位长度域。 例如，如果TX_BIT_LEN为0x08且TX_NUM为0x0，TX[7:0]内的值将会在下一次传输中被发送出去。如果TX_BIT_LEN为0x00且TX_NUM为0x1，SPI控制器将使用同样的配置连续执行两个32位发送/接收。TX0[31:0]先传输，然后再传输TX1[31:0]。

SPI控制和状态寄存器2 (SPI_CNTRL2)

寄存器	偏移	R/W	描述	复位值
SPI_CNTRL2	SPIx_BA+0x3C	R/W	控制和状态寄存器2	0x0000_0000

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

位	描述
[31:17]	悶隱 保留.
[16]	SS_INT_OPT 从机选择无效中断选项 该设置只在SPI控制器配置为从机电平触发有效。 0 = 当从机选择信号变为无效电平时，IF位不会置1。 1 = 当从机选择信号变为无效电平时，IF位会置1。
[15:12]	悶隱 保留.
[11]	SLV_START_INTSTS 从机3线模式开始中断状态 该位表明是否开始3线模式传输。该位为SPI_STATUS[11]的镜像。 0 = 从SSTA_INTEN位设置为1后，从机没有检测到任何SPI总线时钟 1 = 在3线模式下，已经开始传输。当传输完成或该位写1，该位将被自动清0。
[10]	SSTA_INTEN 从机3线模式开始中断使能 在3线模式下传输开始时，该位用于使能中断。如果在传输开始后由用户定义的时间内没有传输完成的中断，则用户可以置位SLV_ABORT位强制完成传输。 0 = 禁用传输开始中断 1 = 使能传输开始中断，当当前传输完成或SLV_START_INTSTS位被清0，该位将被清0。
[9]	SLV_ABORT 从机3线模式异常控制 在正常操作中，当接收到数据符合TX_BIT_LEN的要求位的值时，会有中断事件。 如果接收到的位数少于要求的值而且在3线模式的从机模式下，在一次传输的时间后没有更多的总线时钟输入时，用户可以设定该位来强制完成当前的传输，然后用户就可以收到一个传输完成的中断事件。 注：软件设置该位为1后，该位会被硬件自动清零。
[8]	NOSLVSEL 从机3线模式使能

		该位用于在从机模式下忽略从机选择信号。当该位设为1时，SPI控制器可以工作于3线接口，包括SPICLK, SPI_MISO 和 SPI_MOSI。 0 = 控制器为4线双向接口。 1 = 在从机模式，控制器为3线双向接口。当该位置1时，在GO_BUSY 置1后，控制器将准备发送/接收数据。 注: 在从机3线模式，SS_LTRIG 位(SPI_SSR[4]) 需设置为1.
[7:0]	悵隱	保留.

SPI FIFO控制寄存器(SPI_FIFO_CTL)

寄存器	偏移	R/W	描述	复位值
SPI_FIFO_CTL	SPIx_BA+0x40	R/W	SPI FIFO控制寄存器	0x2200_0000

31	30	29	28	27	26	25	24
悶隱		TX_THRESHOLD			悶隱		RX_THRESHOLD
23	22	21	20	19	18	17	16
悶隱		TIMEOUT_IN_TEN	悶隱				
15	14	13	12	11	10	9	8
悶隱							
7	6	5	4	3	2	1	0
悶隱	RXOV_INTEN	悶隱		TX_INTEN	RX_INTEN	TX_CLR	RX_CLR

位	描述	
[31:30]	悶隱	保留.
[29:28]	TX_THRESHOLD	发送 FIFO 阈值 如果发送FIFO缓冲的有效数据数量小于或等于TX_THRESHOLD 设置的值，TX_INTSTS 位会被置1，否则将会被清0.
[27:26]	悶隱	保留.
[25:24]	RX_THRESHOLD	接收 FIFO 阈值 如果接收FIFO缓冲的有效数据数量大于RX_THRESHOLD 设置的值，RX_INTSTS 位将会被设置为1，否则RX_INTSTS 的值将会被清0
[23:22]	悶隱	保留.
[21]	TIMEOUT_INTEN	接收FIFO超时中断使能控制 0 = 超时中断禁用 1 = 超时中断使能
[20:7]	悶隱	保留.
[6]	RXOV_INTEN	接收 FIFO溢出中断使能控制 0 = 接收FIFO溢出中断禁用 1 = 接收FIFO溢出中断使能
[5:4]	悶隱	保留.
[3]	TX_INTEN	发送阈值中断使能控制 0 = 发送阈值中断禁用 1 = 发送阈值中断使能

[2]	RX_INTEN	接收阈值中断使能控制 0 = 接收阈值中断禁用 1 = 接收阈值中断使能
[1]	TX_CLR	清除发送FIFO缓冲 0 = 无影响. 1 = 清除发送FIFO缓冲， TX_FULL标志将会被清除为0且TX_EMPTY标志将会被设置为1。当软件往该位写1后，该位将被清0，且发送FIFO被清除。
[0]	RX_CLR	清除接收FIFO缓冲 0 = 无影响. 1 = 清除接收FIFO缓冲， RX_FULL标志将会被清除为0且RX_EMPTY标志将会被设置为1。当软件往该位写1后，该位将被清0，且接收FIFO被清除。

SPI状态寄存器 (SPI_STATUS)

寄存器	偏移	R/W	描述	复位值
SPI_STATUS	SPIx_BA+0x44	R/W	SPI 状态寄存器	0x0500_0000

31	30	29	28	27	26	25	24
TX_FIFO_COUNT				TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY
23	22	21	20	19	18	17	16
帳隱			TIMEOUT	帳隱			IF
15	14	13	12	11	10	9	8
RX_FIFO_COUNT				SLV_START_INTSTS	帳隱		
7	6	5	4	3	2	1	0
帳隱			TX_INTSTS	帳隱	RX_OVERRUN	帳隱	RX_INTSTS

位	描述
[31:28]	TX_FIFO_COUNT 发送 FIFO 数据数量(只读) 表明发送FIFO缓冲的有效数据的数量
[27]	TX_FULL 发送 FIFO 缓冲满标志(只读) 该位为SPI_CNTRL[27]的镜像 0 = 发送FIFO缓冲非满 1 = 发送FIFO缓冲满
[26]	TX_EMPTY 发送 FIFO 缓冲空标志(只读) 该位为SPI_CNTRL[26]的镜像 0 = 发送FIFO缓冲非空 1 = 发送FIFO缓冲空
[25]	RX_FULL 接收FIFO缓冲满标志(只读) 该位为SPI_CNTRL[25]的镜像 0 = 接收 FIFO 缓冲非满 1 = 接收 FIFO 缓冲满
[24]	RX_EMPTY 接收FIFO缓冲空标志(只读) 该位为SPI_CNTRL[24]的镜像 0 = 接收 FIFO 缓冲非空 1 = 接收 FIFO 缓冲空
[23:21]	帳隱 保留.
[20]	TIMEOUT 超时中断标志 0 = 无接收FIFO超时事件 1 = 接收FIFO缓冲非空，且在主机模式下超过64个SPI时钟周期没有被读出来或在从机模式下超过576个SPI外设时钟周期没有被读出来。当接收FIFO缓冲由软件读

		出来，超时状态将被自动清0。 注:该位写1清0
[19:17]	悵隱	保留.
[16]	IF	SPI单位发送中断标志 该位为SPI_CNTRL[16]的镜像. 0 = 发送还没完成 1 = SPI控制器完成一次单位传输 注: 该位写1清0
[15:12]	RX_FIFO_COUNT	接收FIFO数据数量(只读) 表明接收FIFO缓冲的有效数据数量
[11]	SLV_START_INTSTS	从机开始中断状态 该位表明3线模式传输开始，该位为SPI_CNTRL2[11]的镜像。 0 = 传输还没开始 1 = 3线模式传输开始。传输结束或往该位写1都会将该位清0.
[10:5]	悵隱	保留.
[4]	TX_INTSTS	发送FIFO阈值中断状态(只读) 0 = 发送FIFO缓冲的有效数据数量大于TX_THRESHOLD 设置的值 1 = 发送FIFO缓冲的有效数据数量小于或等于TX_THRESHOLD 设置的值 注: 如果TX_INTEN = 1 和TX_INTSTS = 1, SPI控制器将产生一个SPI中断请求
[3]	悵隱	保留.
[2]	RX_OVERRUN	接收FIFO溢出状态 当接收FIFO缓冲满了，接下来发送过来的数据将会被丢弃，且该位将置1. 注: 该位写1清0
[1]	悵隱	保留.
[0]	RX_INTSTS	接收FIFO阈值中断状态 (只读) 0 = 接收FIFO 缓冲里的有效数据数量少于或等于RX_THRESHOLD 设置的值。 1 = 接收FIFO缓冲里的有效数据数量大于RX_THRESHOLD 设置的值。 注: 如果RX_INTEN = 1且RX_INTSTS = 1，SPI控制器将产生一个SPI中断请求。

5.15.10 NUC029FAE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:				
SPI_BA = 0x4003_0000				
SPI_CNSTRL	SPI_BA+0x00	R/W	SPI控制和状态寄存器	0x0500_3004
SPI_DIVIDER	SPI_BA+0x04	R/W	SPI时钟分频器寄存器	0x0000_0000
SPI_SSR	SPI_BA+0x08	R/W	SPI从机选择寄存器	0x0000_0000
SPI_RX	SPI_BA+0x10	R	SPI数据接收寄存器	0x0000_0000
SPI_TX	SPI_BA+0x20	W	SPI数据发送寄存器	0x0000_0000
SPI_CNSTRL2	SPI_BA+0x3C	R/W	SPI控制和状态寄存器2	0x0000_0000
SPI_FIFO_CTL	SPI_BA+0x40	R/W	SPI FIFO控制寄存器	0x2200_0000
SPI_STATUS	SPI_BA+0x44	R/W	SPI 状态寄存器	0x0500_0000

5.15.11 NUC029FAE寄存器描述

SPI控制和状态寄存器(SPI_CNTRL)

寄存器	偏移	R/W	描述	复位值
SPI_CNTRL	SPI_BA+0x00	R/W	SPI控制和状态寄存器	0x0500_3004

31	30	29	28	27	26	25	24
悵隱				TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY
23	22	21	20	19	18	17	16
悵隱		FIFO	悵隱	REORDER	SLAVE	IE	IF
15	14	13	12	11	10	9	8
SP_CYCLE				CLKP	LSB	悵隱	
7	6	5	4	3	2	1	0
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

位	Description	
[31:28]	Reserved	保留.
[27]	TX_FULL	<p>发送 FIFO缓冲满标志(只读) 0 =发送FIFO缓冲非满 1 =发送 FIFO缓冲满 注: 该位为SPI_STATUS[27]的镜像</p>
[26]	TX_EMPTY	<p>发送FIFO缓冲空标志(只读) 0 = 发送FIFO缓冲非空 1 = 发送FIFO缓冲空. 注:该位为SPI_STAUTS[26]的镜像</p>
[25]	RX_FULL	<p>接收FIFO缓冲满标志(只读) 0 = 接收FIFO缓冲非满 1 = 接收FIFO缓冲满 注: 该位为SPI_STATUS[25]的镜像</p>
[24]	RX_EMPTY	<p>接收FIFO缓冲空标志(只读) 0 = 接收FIFO缓冲非空 1 = 接收FIFO缓冲空 注: 该位为SPI_CNTRL[24]的镜像</p>
[23:22]	Reserved	保留.
[21]	FIFO	<p>FIFO模式使能控制 0 = FIFO模式禁用 1 = FIFO模式使能</p>

		<p>注 1: 在使能FIFO模式前，其他相关设置必须提前设置</p> <p>注 2: 在主机模式,如果FIFO模式使能，往4级深度发送FIFO缓冲写数据将使GO_BUSY位自动置1. 当储存在FIFO缓冲的所有数据发送后，GO_BUSY 位将返回0.</p>
[20]	Reserved	保留.
[19]	REORDER	<p>字节重排序功能</p> <p>0 = 字节重排序功能禁用 1 = 字节重排序功能使能</p> <p>注: 该设置只有在TX_BIT_LEN 设为16、24或32位时才有效。</p>
[18]	SLAVE	<p>从机模式控制</p> <p>0 = 主机模式. 1 = 从机模式.</p>
[17]	IE	<p>单位传输中断使能控制</p> <p>0 = SPI 单位传输中断禁止 1 = SPI 单位传输中断使能</p>
[16]	IF	<p>单位传输中断标志</p> <p>0 = 传输还没完成 1 = SPI控制器完成一个单位传输</p> <p>注 1: 该位写1清0 注 2: 该位为SPI_STATUS[16]的镜像</p>
[15:12]	SP_CYCLE[3:0]	<p>休眠间隙 (只有主机)</p> <p>该四位提供一个在传输过程中连续两个发送/接收事务之间可配置的休眠间隔。休眠间隔是指前一个事务的最后一个时钟边沿和后一个事务的第一个时钟边沿之间的间隔。默认值是0x3. 休眠间隔长度根据如下公式获得:</p> $(SP_CYCLE[3:0] + 0.5) * \text{SPICLK时钟周期}$ <p>例:</p> <p>SP_CYCLE = 0x0 ... 0.5个SPICLK时钟周期 SP_CYCLE = 0x1 ... 1.5个SPICLK时钟周期 SP_CYCLE = 0xE ... 14.5个SPICLK时钟周期 SP_CYCLE = 0xF ... 15.5个SPICLK时钟周期</p>
[11]	CLKP	<p>时钟极性</p> <p>0 = SPICLK空闲状态为低电平 1 = SPICLK空闲状态为高电平</p>
[10]	LSB	<p>LSB优先</p> <p>0 = MSB优先发送/接收。 1 = LSB优先发送/接收。</p>
[9:8]	Reserved	保留.
[7:3]	TX_BIT_LEN[4:0]	<p>发送位长度</p> <p>该位域指定在一个发送/接收事务中，多少个数据位将会被传输。最小位长是 8，最多可以达到 32 位。</p>

		TX_BIT_LEN = 0x08 ... 8 位。 TX_BIT_LEN = 0x09 ... 9 位。 TX_BIT_LEN = 0x1F ... 31 位。 TX_BIT_LEN = 0x00 ... 32 位。
[2]	TX_NEG	下降沿发送 0 = SPICLK 上升沿发送数据 1 = SPICLK 下降沿发送数据
[1]	RX_NEG	下降沿接收 0 = SPICLK 上升沿接收数据 1 = SPICLK 下降沿接收数据
[0]	GO_BUSY	SPI发送控制位和忙状态 如果FIFO模式使能，该位将由硬件控制且只读。 如果FIFO模式禁用，在数据传输过程中，该位保持为1。传输结束后，该位自动清0。 0 = 如果SPI在传输，往该位写0停止数据传输 1 = 在主机模式，往该位写1开始SPI数据传输。在从机模式，往该位写1表明从机准备好与主机通讯。 注 1: 如果FIFO模式禁用，在写1到GO_BUSY位之前，所有配置将要配置好。 注 2: 在SPI从机模式，如果FIFO模式禁用 且SPI总线时钟在数据传输过程中保持为空闲状态，当从机选择信号变为无效状态时，GO_BUSY 位将会被清除为0.

SPI分频寄存器 (SPI_DIVIDER)

寄存器	偏移	R/W	描述	复位值
SPI_DIVIDER	SPI_BA+0x04	R/W	SPI时钟分频寄存器	0x0000_0000



位	描述	
[31:8]	保留	保留.
[7:0]	DIVIDER	<p>时钟分频位 (只有主机)</p> <p>该区域的值是一个频率除数，来决定SPI外设时钟频率f_{spi}，SPI主机的时钟频率由SPICLK管脚输出。该频率可由下面的公式获得：</p> <p>如果BCn位(SPI_CNTRL2[31])设置为0</p> $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER + 1) * 2}$ <p>如果BCn位设置为1</p> $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER + 1)}$ <p>$f_{SPI_clock_src}$ 为CLKSEL1 寄存器定义的SPI外设时钟源</p>

SPI从机选择寄存器 (SPI_SS_R)

寄存器	偏移	R/W	描述	复位值
SPI_SS_R	SPI_BA+0x08	R/W	SPI从机选择寄存器	0x0000_0000

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

位	描述	
[31:6]	帳隱	保留.
[5]	LTRIG_FLAG	<p>电平触发标志(只读, 只有从机) 当SS_LTRIG位在从机模式下被置位, 读该位的值可用来表示一次传输完成后接收位的数量是否达到要求。 0 = 一次传输的个数或位长度不满足指定的要求 1 = 传输的个数及位长度满足TX_BIT_LEN定义的要求</p>
[4]	SS_LTRIG	<p>从机选择电平触发使能位 (只有从机) 0 = 输入从机选择信号为边沿触发 1 = 输入从机选择信号为电平触发</p>
[3]	AUTOSS	<p>自动从机选择功能使能位 (只有主机) 0 = 置位/清除SSR位, SPISS引脚信号将激活/失效 1 = SPISS管脚信号由硬件自动产生, 这意味着当通过设置GO_BUSY来开始发送/接收时, 从机选择信号将会由SPI控制器激活, 当发送/接收完成时, 该信号又会处于无效状态。</p>
[2]	SS_LVL	<p>从机选择有效电平 (只有从机) 该位定义从机选择信号(SPISS)的有效状态 如果SS_LTRIG 位为1: 0 = 从机选择信号SPISS为低电平有效 1 = 从机选择信号SPISS为高电平有效 如果SS_LTRIG 位为0: 0 = 从机选择信号SPISS下降沿有效 1 = 从机选择信号SPISS上升沿有效</p>
[1]	帳隱	保留.
[0]	SSR	从机选择控制位 (只有主机)

		<p>如果AUTOSS位为0 0 = 设置SPISS为无效状态 1 = 设置SPISS为有效状态 如果AUTOSS位为1 0 = 保持SPISS为无效状态 1 = SPISS在发送/接收的时间内被自动驱动到激活状态，而其他时间为非激活状态。</p>
--	--	--

SPI数据接收寄存器 (SPI_RX)

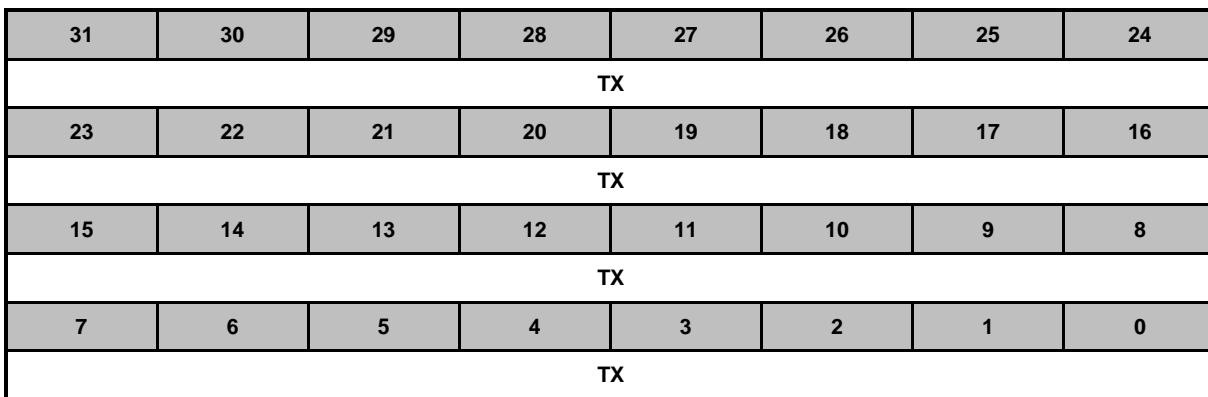
寄存器	偏移	R/W	描述	复位值
SPI_RX	SPI_BA+0x10	R	SPI 数据接收寄存器	0x0000_0000

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	<p>数据接收位 (只读)</p> <p>数据接收寄存器内保存最后一次传输所接收到的数据。有效的位取决于SPI_CNTRL寄存器中的传输位长度域。</p> <p>例如，如果TX_BIT_LEN配置为0x08，RX[7:0]内保存了收到的数据，其它位的数据未知，数据接收寄存器是只读寄存器。</p>

SPI数据发送寄存器(SPI_TX)

寄存器	偏移	R/W	描述	复位值
SPI_TX	SPI_BA+0x20	W	SPI数据发送寄存器	0x0000_0000



位	描述								
[31:0]	TX	数据发送位(只写) 数据发送寄存器保存下一次将要发送的数据。有效的位取决于CNTRL寄存器中的传输位长度域。 例如，如果TX_BIT_LEN为0x08，TX[7:0]内的值将会在下一次传输中被发送出去。							

SPI控制和状态寄存器2 (SPI_CTRL2)

寄存器	偏移	R/W	描述	复位值
SPI_CTRL2	SPI_BA+0x3C	R/W	控制和状态寄存器2	0x0000_0000

31	30	29	28	27	26	25	24
BCn	悶隱						
23	22	21	20	19	18	17	16
悶隱							SS_INT_OPT
15	14	13	12	11	10	9	8
悶隱				SLV_START_INTSTS	SSTA_INTE_N	SLV_ABORT	NOSLVSEL
7	6	5	4	3	2	1	0
悶隱							

位	描述
[31]	BCn 时钟配置向后兼容选项 0 = 时钟配置向后兼容 1 = 时钟配置不向后兼容 注: 详情参考SPI_DIVIDER寄存器
[30:17]	悶隱 保留.
[16]	SS_INT_OPT 从机选择无效中断选项(只有从机) 0 = 当从机选择信号变为无效电平时，IF位不会被置1。 1 = 当从机选择信号变为无效电平时，IF位会被置1。 注: 该设置仅在从机设备SPI控制器配置为电平触发有效
[15:12]	悶隱 保留.
[11]	SLV_START_INTSTS 从机3线模式开始中断状态(只有从机) 该位表明是否开始3线模式传输。 0 = 自从SSTA_INTE_N置1后，从机没有检测到SPI总线时钟 1 = 3线模式传输已经开始 注 1: 当传输完成或软件往该位写1，该位将自动清0. 注 2: 该位为SPI_STATUS[11]的镜像
[10]	SSTA_INTE_N 从机3线模式开始中断使能控制(只有从机) 在3线模式下传输开始时，该位用于使能中断。如果在传输开始后由用户定义的时间内没有传输完成的中断，则用户可以置位SLV_ABORT位强制完成传输。 0 = 禁用传输开始中断 1 = 使能传输开始中断 注: 该位在传输完成后清零或SLV_START_INTSTS位被清除后清零。

[9]	SLV_ABORT	<p>从机3线模式异常控制位(只有从机)</p> <p>在正常操作中，当接收到数据符合TX_BIT_LEN的要求位的值时，会有中断事件。</p> <p>如果接收到的位数少于要求的值而且在3线模式的从机模式下，在一次传输的时间后没有更多的总线时钟输入时，用户可以设定该位来强制完成当前的传输，然后用户就可以收到一个传输完成的中断事件。</p> <p>0 = 当NOSLVSEL设置为1时，不强制完成传输。 1 = 当NOSLVSEL设置为1时，强制完成传输。</p> <p>注:软件设置该位为1后，该位会被硬件自动清零。</p>
[8]	NOSLVSEL	<p>从机3线模式使能控制(只有从机)</p> <p>SPI控制器工作在3线接口，包含SPICLK、SPI_MISO 和SPI_MOSI</p> <p>0 = 控制器为4线双向接口 1 = 在从机模式下，控制器为3线双向接口。在GO_BUSY 置1后，控制器将准备发送/接收数据。</p> <p>注: 在3线模式，SS_LTRIG 位(SPI_SSR[4]) 需设为1.</p>
[7:0]	悵隱	保留.

SPI FIFO控制寄存器 (SPI_FIFO_CTL)

寄存器	偏移	R/W	描述	复位值
SPI_FIFO_CTL	SPI_BA+0x40	R/W	SPI FIFO控制寄存器	0x2200_0000

31	30	29	28	27	26	25	24
		TX_THRESHOLD			RX_THRESHOLD		
23	22	21	20	19	18	17	16
		TIMEOUT_IN_TEN					
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
	RXOV_INTEN			TX_INTEN	RX_INTEN	TX_CLR	RX_CLR

位	描述	
[31:30]		保留.
[29:28]	TX_THRESHOLD [1:0]	发送FIFO阈值 如果发送FIFO缓存的有效数据数量小于或者等于TX_THRESHOLD设置的值，TX_INTSTS将会被设置为1，否则TX_INTSTS将会被设置为0。
[27:26]		保留.
[25:24]	RX_THRESHOLD [1:0]	接收FIFO 阈值 如果接收FIFO缓存有效数据数量大于RX_THRESHOLD设置的值，RX_INTSTS将会被设置为1，否则RX_INTSTS将会被设置为0。
[23:22]		保留.
[21]	TIMEOUT_INTEN	接收 FIFO 超时中断使能控制 0 = 超时中断禁用 1 = 超时中断使能
[20:7]		保留.
[6]	RXOV_INTEN	接收 FIFO溢出中断使能控制 0 = 接收FIFO溢出中断禁用 1 = 接收FIFO溢出中断使能
[5:4]		保留.
[3]	TX_INTEN	发送阈值中断使能控制 0 = 发送阈值中断禁用 1 = 发送阈值中断使能

[2]	RX_INTEN	接收阈值中断使能控制 0 = 接收阈值中断禁用. 1 = 接收阈值中断使能
[1]	TX_CLR	清除发送FIFO缓冲 0 = 无影响 1 = 清除发送FIFO缓冲. 注: 该位将在软件往该位写1和发送FIFO清除的情况下由硬件清0。
[0]	RX_CLR	清除接收FIFO缓冲 0 = 无影响. 1 = 清除接收 FIFO缓冲 注: 该位将在软件往该位写1和接收FIFO清除的情况下由硬件清0。

SPI 状态寄存器 (SPI_STATUS)

寄存器	偏移	R/W	描述	复位值
SPI_STATUS	SPI_BA+0x44	R/W	SPI状态寄存器	0x0500_0000

31	30	29	28	27	26	25	24
TX_FIFO_COUNT				TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY
23	22	21	20	19	18	17	16
悶隱			TIMEOUT	悶隱			IF
15	14	13	12	11	10	9	8
RX_FIFO_COUNT				SLV_STAR_I NTSTS	悶隱		
7	6	5	4	3	2	1	0
悶隱			TX_INTSTS	悶隱	RX_OVERRUN	悶隱	RX_INTSTS

位	描述	
[31:28]	TX_FIFO_COUNT[3:0]	发送FIFO数据数量(只读) 表明发送FIFO缓冲的有效数据数量
[27]	TX_FULL	发送FIFO缓冲满标志(只读) 0 = 发送FIFO缓冲未满 1 = 发送FIFO缓冲满 注: 该位为SPI_CNTRL[27]的镜像
[26]	TX_EMPTY	发送FIFO缓冲空标志(只读) 0 = 发送FIFO缓冲非空 1 = 发送FIFO缓冲空 注: 该位为SPI_CNTRL[26]的镜像
[25]	RX_FULL	接收FIFO缓冲满标志(只读) 0 = 接收FIFO 缓冲未满 1 = 接收FIFO缓冲满 注: 该位为SPI_CNTRL[25]的镜像
[24]	RX_EMPTY	接收FIFO缓冲空标志(只读) 0 = 接收FIFO缓冲非空 1 = 接收FIFO缓冲空 注: 该位为SPI_CNTRL[24]的镜像
[23:21]	悶隱	保留.
[20]	TIMEOUT	超时中断标志 0 = 无接收FIFO超时事件 1 = 接收FIFO缓冲非空，且在主机模式下超过64个SPI总线时钟没有被读或在从机模式下超

		过576个SPI外设时钟周期没有被读。当接收FIFO缓冲由软件读出来后，超时状态将被自动清除。 注: 该位写1清0
[19:17]	保留	保留.
[16]	IF	SPI单位发送中断标志 0 = 发送未完成 1 = SPI控制器已完成一次单位传输 注 1: 该位写1清0 注 2: 该位为 SPI_CNTRL[16]的镜像
[15:12]	RX_FIFO_COUNT[3:0]	接收FIFO数据数量(只读) 表明接收FIFO缓冲的有效数据数量
[11]	SLV_START_INT_STS	从机开始中断状态(只有从机) 该位用于表明从机3线模式传输已经开始 0 = 从SSTA_INTEN 位置1后，从机没有检测到任何SPI总线时钟。 1 = 从机3线模式传输已经开始 注 1: 传输完成或往该位写1将清除该位。 注 2: 该位为SPI_CNTRL2[11]的镜像
[10:5]	保留	保留.
[4]	TX_INTSTS	发送FIFO阈值中断状态 (只读) 0 = 发送FIFO缓冲的有效数据数量大于TX_THRESHOLD 设置的值 1 = 发送FIFO缓冲的有效数据数量小于或等于TX_THRESHOLD 设置的值 注: 如果TX_INTEN = 1 且 TX_INTSTS = 1, SPI控制器将产生一个SPI中断请求
[3]	保留	保留.
[2]	RX_OVERRUN	接收 FIFO溢出状态 当接收缓冲满了后，接下来的数据将被丢弃，且该位将置1 0 = 接收FIFO无溢出 1 = 接收FIFO溢出 注: 该位写1清0
[1]	保留	保留.
[0]	RX_INTSTS	接收 FIFO阈值中断状态 (只读) 0 = 接收FIFO缓冲的有效数据数量小于或等于RX_THRESHOLD 设置的值 1 = 接收FIFO缓冲的有效数据数量大于RX_THRESHOLD 设置的值 注: 如果RX_INTEN = 1 且 RX_INTSTS = 1, SPI控制器将产生一个SPI中断请求

5.16 模数转换器(ADC)

5.16.1 概述

NuMicro® NUC029xAN 包含一个12位逐次逼近型模数转换器(SAR A/D转换器)，包含8个输入通道；NuMicro® NUC029FAE 包含一个10位逐次逼近型模数转换器(SAR A/D转换器)，包含8个通道。

NUC029xAN 的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；NUC029FAE 只支持单次模式。

A/D转换器可以通过软件、PWM或外部STADC引脚触发转换。

5.16.2 特性

- 模拟输入电压范围: 0~AV_{DD}
- 12位分辨率和10位精度保证(NUC029xAN)
- 10位分辨率和8位精度保证(NUC029FAE)
- 多达8路单端模拟输入通道
 - 或4组互补模拟输入通道 (NUC029xAN)
- 高达760 kSPS采样率 (NUC029xAN)
- 300 KSPS (V_{DD} 4.5V - 5.5V) 、200 KSPS (V_{DD} 2.5V - 5.5V)转换率 (NUC029FAE)
- 四种操作模式 (NUC029FAE只支持单次模式)
 - 单次模式: A/D在指定通道完成一次转换
 - **Burst**模式: A/D转换器采样和转换指定单个通道，并将结果循序的储存在FIFO中
 - 单周期扫描模式: A/D按顺序从编号最小的通道到编号最大的通道转换一次所有选定的通道。
 - 连续扫描模式: A/D转换器连续执行单周期扫描模式直到软件停止A/D转换。
- A/D转换可以开始于:
 - 通过软件写1到ADST位(ADCR[11])
 - PWM触发，可选择延时多少周期开始转换
 - 外部STADC引脚
- 每个通道的转换结果存储在相应数据寄存器内，并带有有效和溢出标志
- 每个通道拥有独立的数据寄存器(NUC029xAN)
- 转换结果可以和指定的值相比较，当转换结果和比较寄存器的设定值相匹配时，用户可设定是否产生中断请求.
- 通道 7 支持 3 个输入源:
 - 外部模拟电压
 - 内部 Band-gap 电压
 - 内部温度传感器输出(NUC029xAN)

5.16.3 NUC029xAN 方框图

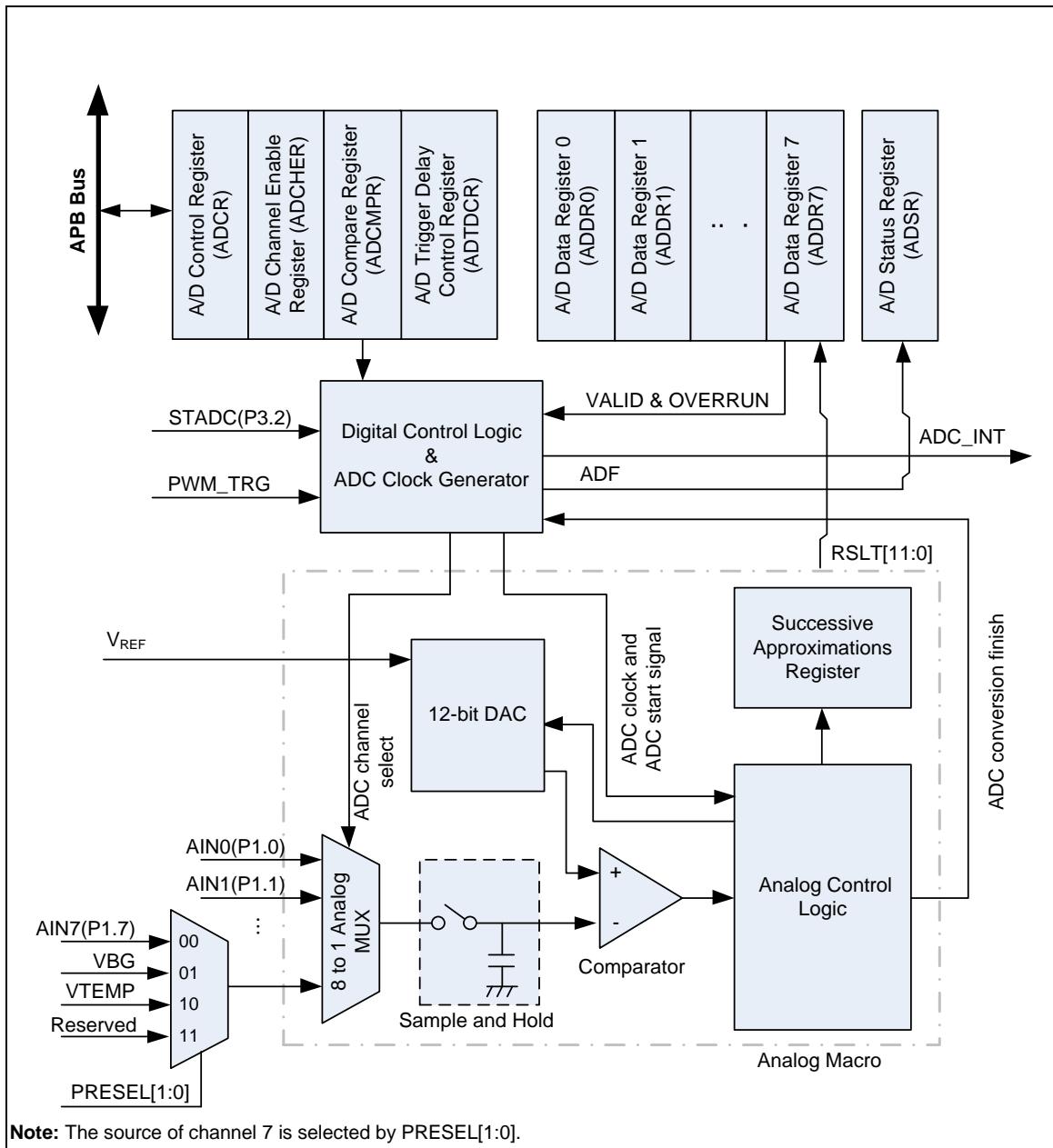


图 5-133 A/D 转换器方框图 (NUC029xAN)

5.16.4 NUC029FAE 方框图

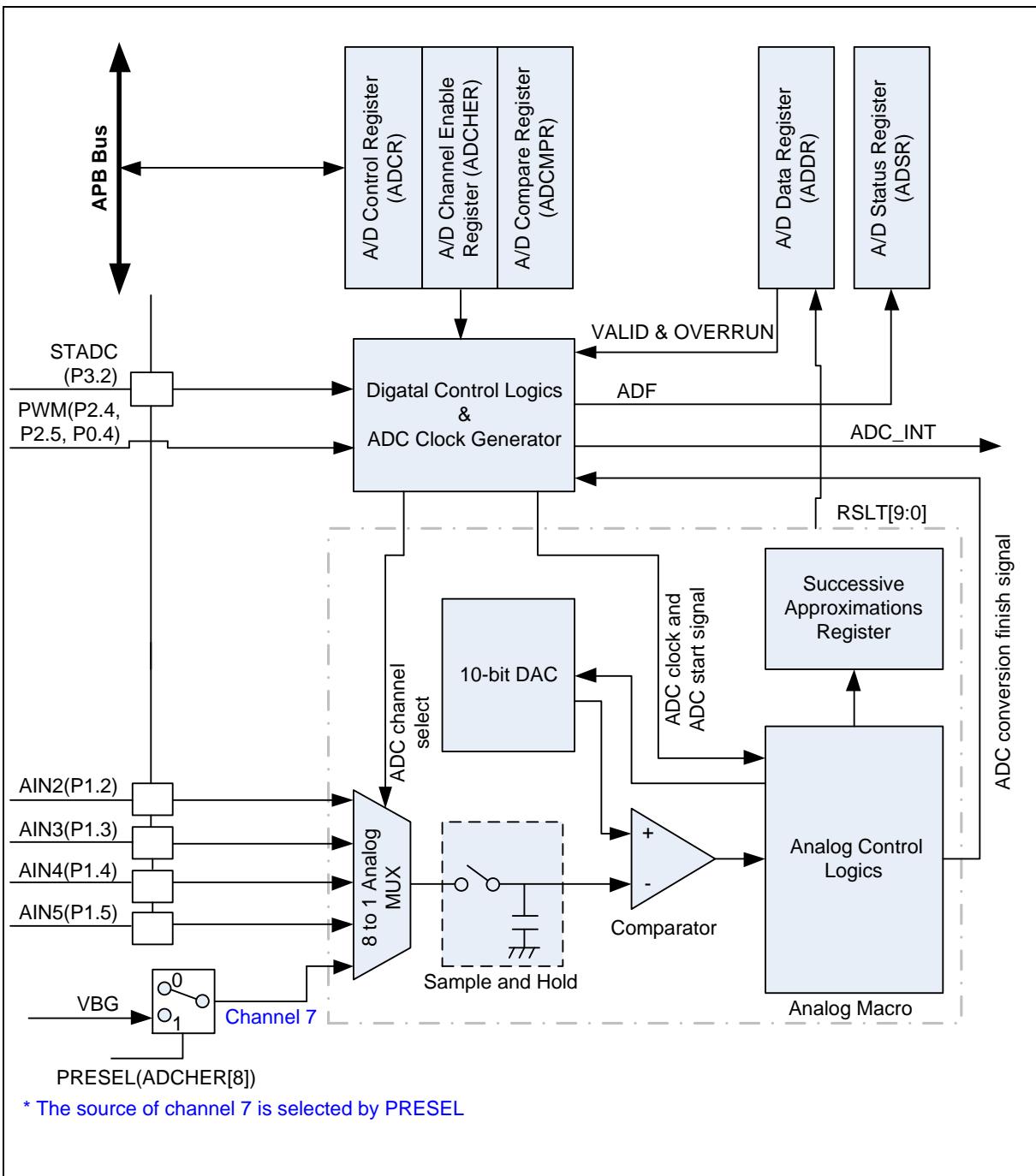


图 5-134 A/D 转换器方框图 (NUC029FAE)

5.16.5 基本配置

ADC引脚功能在P1_MFP寄存器配置。建议关闭模拟输入引脚的数字输入通道，以避免漏电。用户

可以通过配置P1_OFFD 寄存器关闭数字输入通道。

ADC 外设时钟可以在ADC_EN(APBCLK[28])使能。ADC外设时钟源在ADC_S(CLKSEL1[3:2])选择。时钟预分频在ADC_N(CLKDIV[23:16])设置。

5.16.6 功能描述

NUC029xAN:

该A/D转换器通过逐次逼近的方式运行，分辨率为12位。该ADC有四种操作模式：单次模式、Burst模式、单次周期扫描模式和连续周期扫描模式。当用户想要改变操作模式或者模拟输入通道，为了避免错误的操作，软件必须提前清除ADST(ADCR[11])位为0。

NUC029FAE:

A/D转换器通过逐次逼近的方式运行，分辨率为10位。当改变模拟输入通道使能时，为了避免错误的操作，软件必须清除ADCR 寄存器的ADST(ADCR[11])位为0。在ADST位清除时，A/D转换器立即放弃现在的转换并进入空闲状态。

5.16.6.1 ADC外设时钟产生器

NUC029xAN:

最大采样率高达760 kSPS. 该ADC拥有四个时钟源，可通过ADC_S(CLKSEL1[3:2])选择，ADC外设时钟频率按如下公式进行8位预分频：

ADC 外 设 时 钟 频 率 = (ADC 外 设 时 钟 源 频 率) / (ADC_N+1);
8位 ADC_N 位于寄存器CLKDIV[23:16].

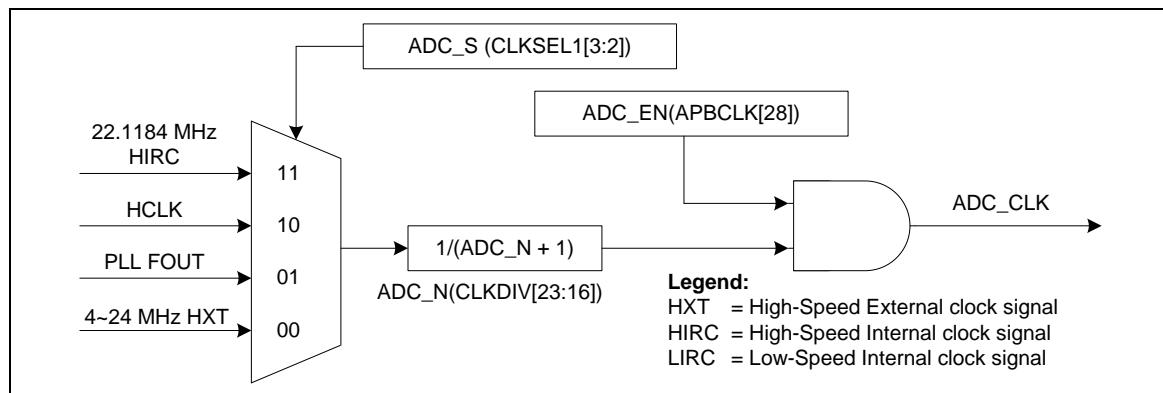


图 5-135 ADC外设时钟控制 (NUC029xAN)

NUC029FAE:

最大采样率高达760 kSPS. 该ADC拥有四个时钟源，可通过ADC_S(CLKSEL1[3:2])选择，通过XTLCLK_EN(PWRCON[1:0])选择HXT 和 LXT。ADC外设时钟频率按如下公式进行8位预分频：

ADC 外 设 时 钟 频 率 = (ADC 外 设 时 钟 源 频 率) / (ADC_N+1);
8位ADC_N位于寄存器CLKDIV[23:16].

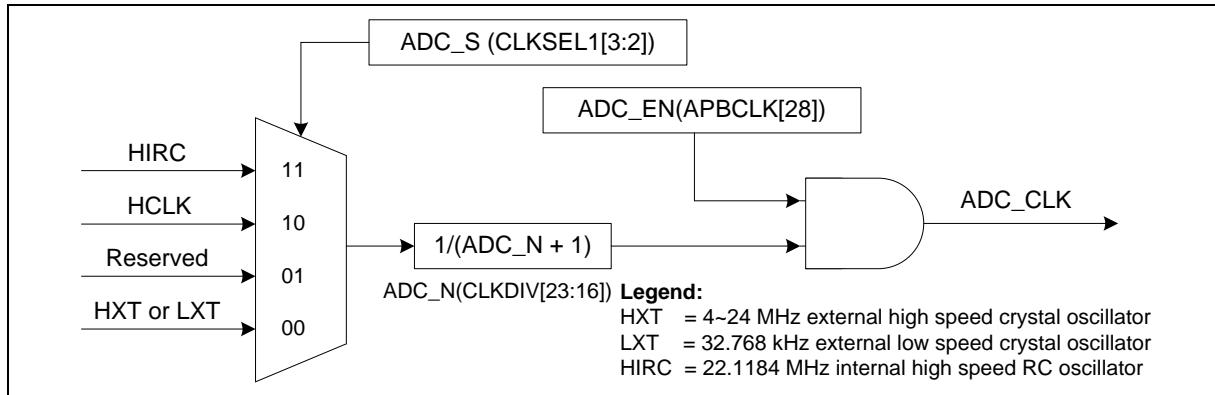


图 5-136 ADC 外设时钟控制 (NUC029FAE)

5.16.6.2 操作模式

NUC029xAN 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，NUC029FAE 的 A/D 转换器只支持单次模式。

5.16.6.2.1 单次模式

单次模式，A/D 转换器只对指定的一个通道进行一次转换。操作流程如下：

- 当软件设置 ADST(ADCR[11]) 位为 1 或外部触发输入，A/D 转换器开始转换。
- 当 A/D 转换器转换结束，NUC029xAN 的转换结果储存在对应通道的 A/D 数据寄存器中，而 NUC029FAE 的转换结果存储在 A/D 数据寄存器中。
- ADF(ADSR[0]) 位置 1。如果 ADIE(ADCR[1]) 位置 1，ADC 中断将产生。
- 在 A/D 转换过程中，ADST(ADCR[11]) 位保持为 1。在 A/D 转换结束后，ADST 位自动清 0 且 A/D 转换器进入空闲状态。

注 1：如果在单次模式，软件使能不止一个通道，只有编号最小的通道被选中，其他使能的通道将被忽略。

注 2：在 ADC 转换时，如果 ADST 位被清 0，BUSY 位将被立即清 0，ADC 不能完成当前转换且 A/D 转换器直接进入空闲状态。

单次模式的一个示例时序图如下所示：

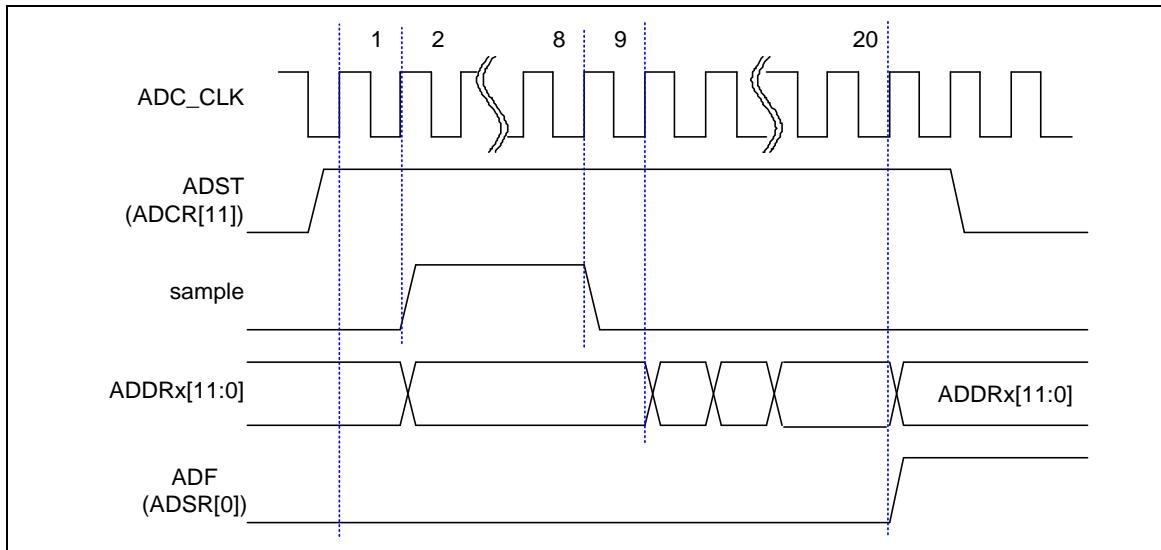


图 5-137 单次模式转换时序图 (NUC029xAN)

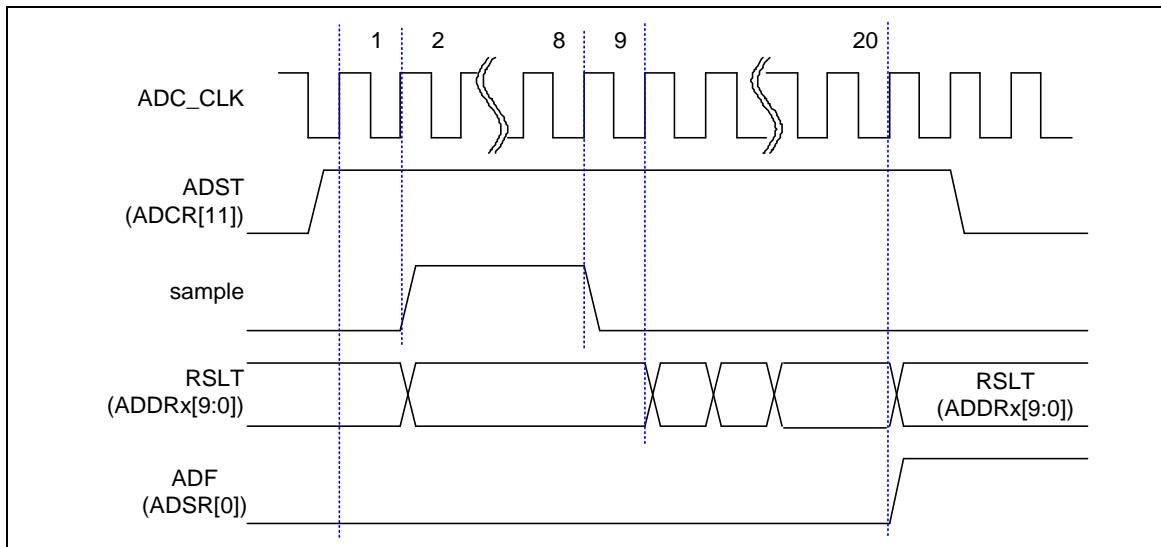


图 5-138 单次模式转换时序图 (NUC029FAE)

5.16.6.2.2 Burst模式

在Burst模式，A/D转换器采样和转换指定单个通道并将转换结果按序存放在FIFO中（最多8次采样）。操作步骤如下：

1. 当ADCR 的ADST(ADCR[11])位通过软件或外部触发输入设置为1，A/D开始转换编号做小的通道
2. 当指定通道的A/D转换完成后，转换结果按顺序传输到FIFO中，只能通过A/D数据寄存器0访

问。

3. 当在FIFO中超过4次采样结果，ADF(ADSR[0])位将置1。这个时候，如果ADIE(ADCR[1])位置1，在A/D转换结束后，将产生一个ADC中断。
4. 如果ADST(ADCR[11])位保持为1，将一直重复步骤2和3。当ADST(ADCR[11])位清0后，ADC将不能完成当前转换且A/D转换器直接进入空闲状态。

注: 在Burst模式，如果软件使能不止一个通道，只有编号最小的通道被转换，其他使能的通道将被忽略。

5.16.6.2.3 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ADST(ADCR[11])位通过软件或外部触发输入设置为1，编号最小的通道将开始A/D转换。
2. 当每一个使能的通道A/D转换完成，转换结果按顺序传输到每个通道相应的A/D数据寄存器。
3. 当所有使能的通道转换完成，ADF(ADSR[0])位将置1。如果ADC中断功能使能，ADC中断产生。
4. 在ADC完成一次周期转换，ADST(ADCR[11])位将自动清除为0且A/D转换进入空闲状态。如果在所有使能的ADC通道转换完成之前，ADST(ADCR[11])位清为0，ADC将不能完成当前转换且A/D转换器直接进入空闲状态。

使能通道(0, 2, 3 和 7)的单次周期扫描模式的一个示例时序图如下：

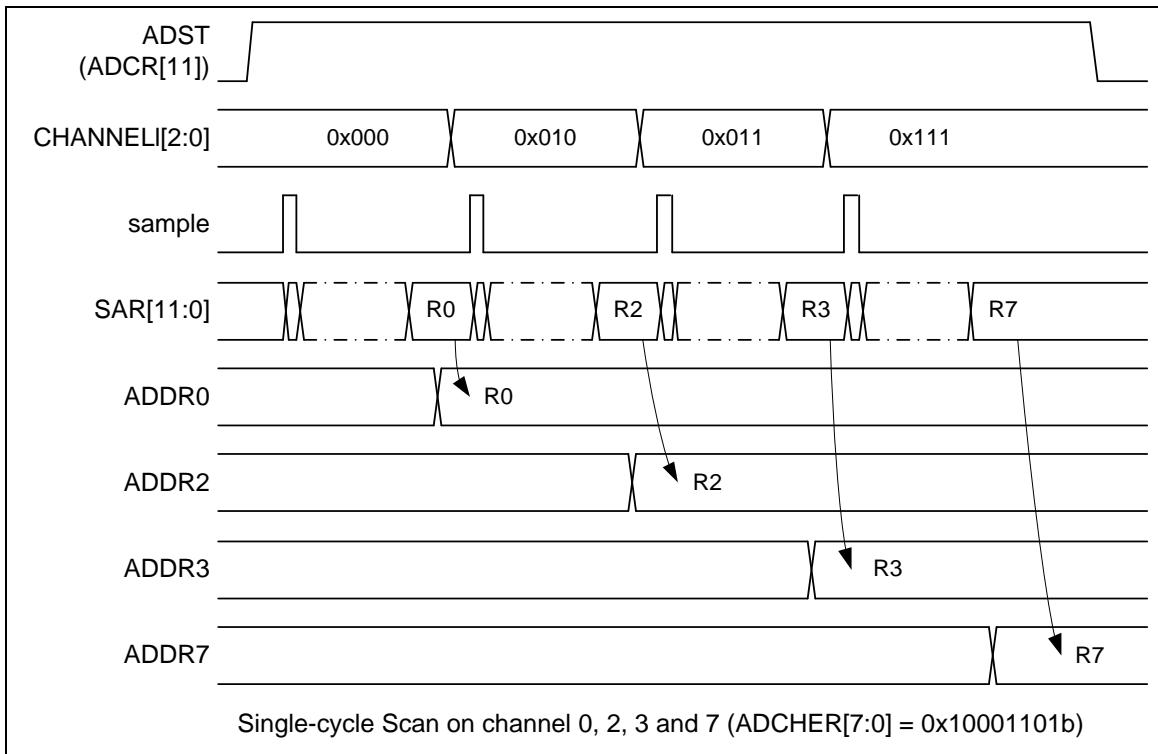


图 5-139 使能通道单次周期扫描模式时序图

5.16.6.2.4 连续扫描模式

在连续扫描模式下，A/D转换器循环转换CHEN(ADCR[7 :0])使能的通道（最多8个通道），操作流程如下：

1. 当ADST(ADCR[11])位通过软件或外部触发输入置1，A/D开始转换编号最小的通道。
2. 当每个使能的通道A/D转换完成，每个使能通道的转换结果存储在每个使能通道对应的A/D寄存器中。
3. 当A/D转换器按顺序完成所有使能通道的转换，ADF(ADSR[0])将置1。如果ADC中断功能使能，ADC中断发生。如果软件没有清除ADST(ADCR[11])位，使能的通道将再一次从编号最小的通道开始转换。
4. 如果ADST(ADCR[11])位保持为1，将重复执行步骤2~3。当ADST(ADCR[11])清0后，ADC不能完成当前转换且A/D转换器直接进入空闲状态。

使能通道(0, 2, 3 和 7)连续扫描模式的一个示例时序图如下所示：

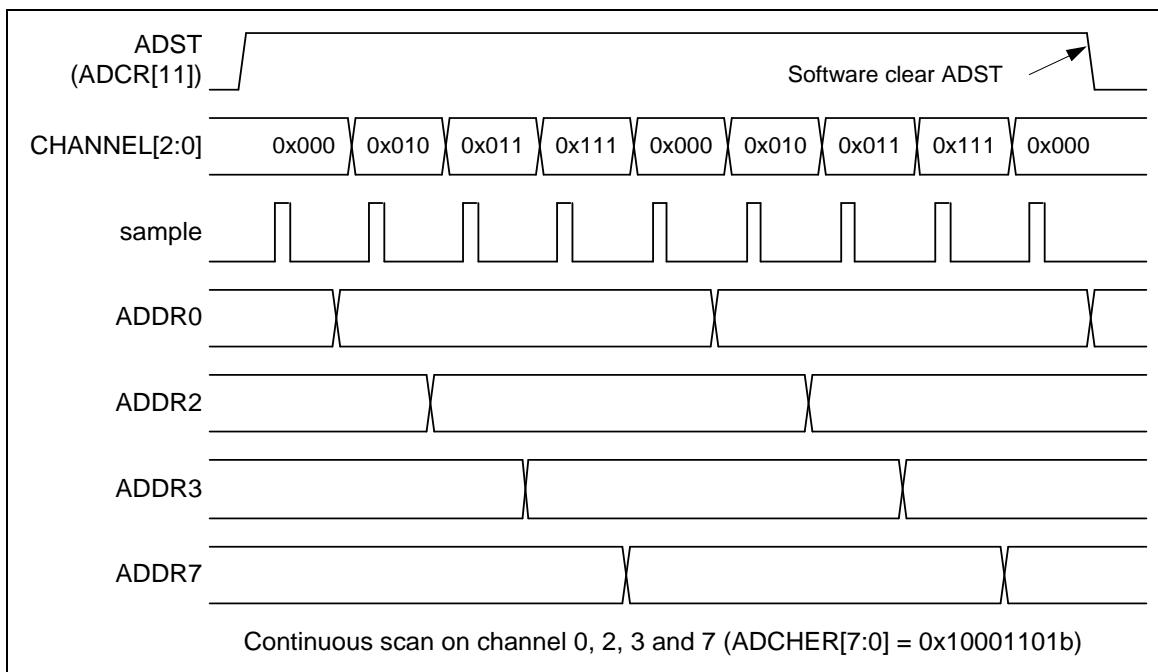


图 5-140 使能通道连续扫描模式时序图

5.16.6.3 外部触发输入采样和A/D转换时间

NUC029xAN:

在单次扫描模式，可以通过外部引脚触发A/D转换。TRGEN(ADCR[8])位设置为0使能ADC外部触发功能，设置TRGS(ADCR[5:4])为00b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(ADCR[11])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。

NUC029FAE:

A/D可以通过外部引脚条件触发转换。TRGEN(ADCR[8])位设置为1使能ADC外部触发功能，设置TRGS(ADCT[5:4])为00b选择外部触发输入为STADC引脚。软件可以通过设置TRGCOND(ADCR[6])选择下降沿或上升沿触发。一个8位的采样计数器用于抗尖峰脉冲。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。

5.16.6.4 PWM 触发

NUC029xAN:

在单周期扫描模式，A/D可以通过PWM触发转换。设置TRGEN(ADCR[8])位为1使能ADC外部触发功能，设置TRGS(ADCT[5:4])为11b选择外部硬件触发输入源为PWM。当PWM触发使能时，设置PTDT(ADTDCR[7:0])位可以在PWM触发和ADC开始转换之间插入一个延时时间。

NUC029FAE:

A/D也可以通过PWM触发转换。TRGEN(ADCR[8]) 设置为高使能ADC外部硬件触发功能，设置TRGS(ADCT[5:4] 位为 11b 选择外部硬件触发输入源为 PWM。当 PWM 触发使能时，设置PTDT(ADTDCR[7:0])位可以在 PWM 触发和 ADC 开始转换之间插入一个延时时间。

5.16.6.5 通过比较模式功能监控转换结果

NuMicro® NUC029 系列 ADC 控制器提供两个比较寄存器 (ADCMPO 和 ADCMPR1) 来监控最多两个通道。软件可以通过设置 CMPCH(ADCMPRx[5:3]) 选择监控哪个通道。CMPCOND(ADCMPRx[2]) 位指定比较条件。如果 CMPCOND(ADCMPRx[2]) 位为 0，当转换结果小于 CMPD[11:0] (NUC029xAN) / CMPD[9:0] (NUC029xAN) 的值时，内部匹配计数器将加 1；如果 CMPCOND(ADCMPRx[2]) 位为 1，当转换结果大于或等于 CMPD[11:0] (NUC029xAN) / CMPD[9:0] (NUC029xAN) 的值时，内部匹配计数器将减 1。当 CMPCH(ADCMPRx[5:3]) 指定的通道转换完成时，将自动触发一次比较。当比较结果和设定值相匹配，比较匹配寄存器加 1，否则比较匹配计数器将清 0。当匹配计数器到达(CMPMATCNT+1)设置的值时，CMF 位将置 1，如果 CMPIE(ADCMPRx[1]) 位设置为 1，将产生一个 ADC_INT 中断。在扫描模式，软件可以通过该功能监控外部输入引脚电压变化而不会增加软件负载。详细逻辑框图如下：

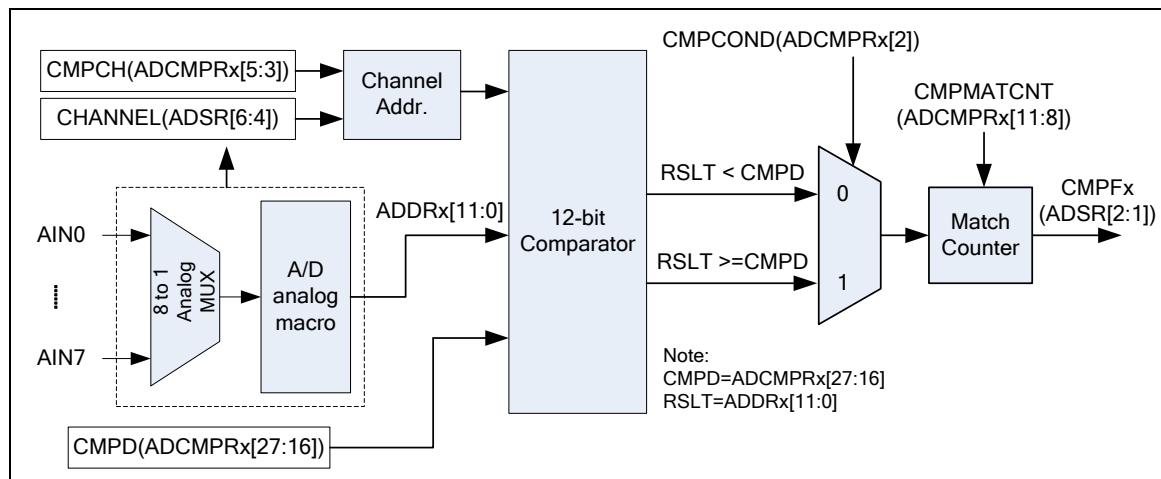
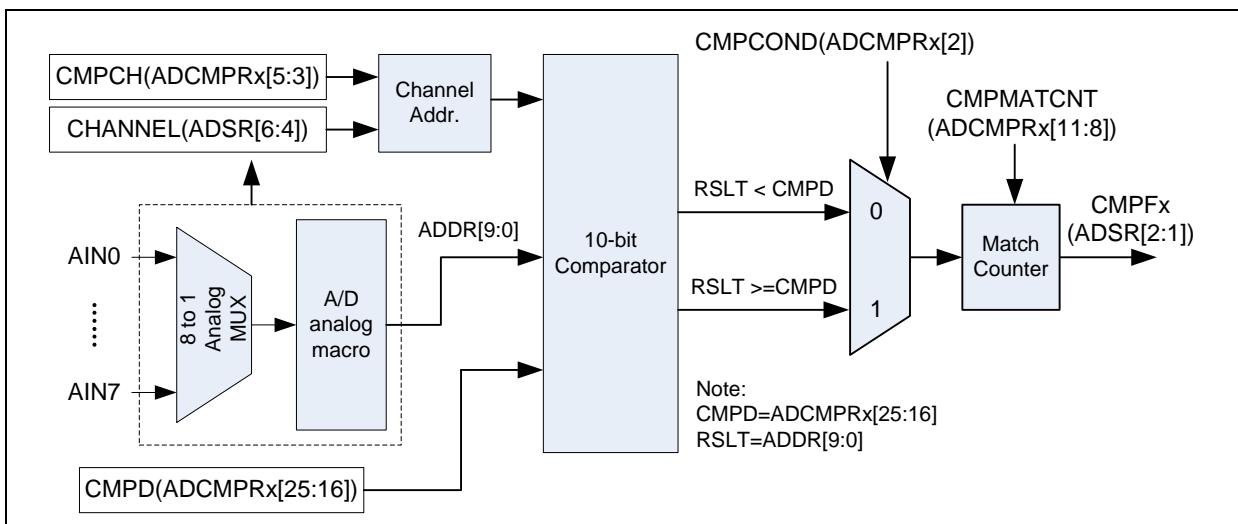


图 5-141 A/D 转换结果监控逻辑框图 (NUC029xAN)



5.16.6.6 中断源

ADC 中断有 3 个中断源。当 ADC 完成转换，A/D 转换完成标志 ADF(ADSR[0]) 将被置 1。CMPF0(ADSR[1]) 和 CMPF1(ADSR[2]) 为比较功能的比较标志位。当转换结果与 ADCMPR0/1 设置的条件匹配时，相应的位将置 1。当 ADF(ADSR[0]), CMPF0(ADSR[1]) 和 CMPF1(ADSR[2]) 标志有一个置 1 且相应中断使能位 ADIE(ADCR[1]) 和 CMPIE(ADCMPRx[1]) 置 1，ADC 中断将产生。软件可以清除这些标志位来取消中断请求。

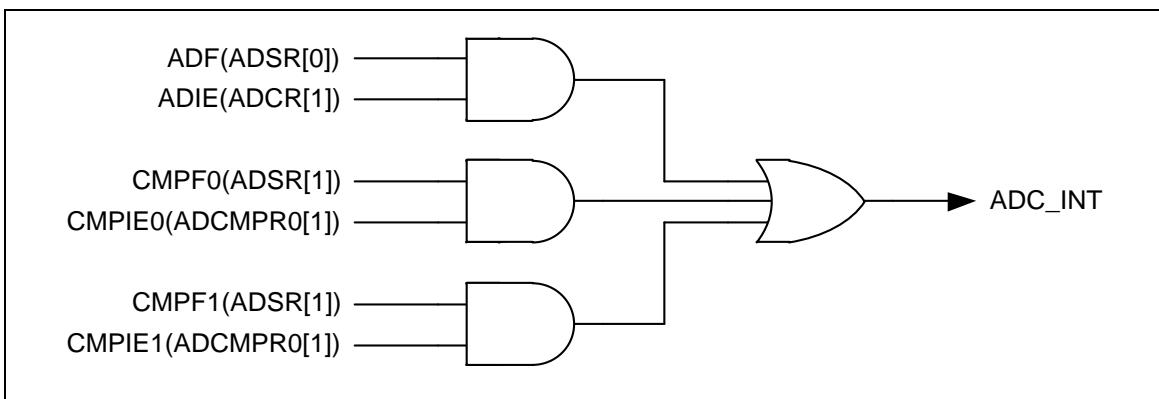


图 5-143 A/D 控制器中断

5.16.7 NUC029xAN 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 = 0x400E_0000				
ADDR0	ADC_BA+0x00	R	ADC 数据寄存器0	0x0000_0000
ADDR1	ADC_BA+0x04	R	ADC数据寄存器 1	0x0000_0000
ADDR2	ADC_BA+0x08	R	ADC数据寄存器2	0x0000_0000
ADDR3	ADC_BA+0x0C	R	ADC数据寄存器3	0x0000_0000
ADDR4	ADC_BA+0x10	R	ADC数据寄存器4	0x0000_0000
ADDR5	ADC_BA+0x14	R	ADC数据寄存器5	0x0000_0000
ADDR6	ADC_BA+0x18	R	ADC数据寄存器6	0x0000_0000
ADDR7	ADC_BA+0x1C	R	ADC数据寄存器7	0x0000_0000
ADCR	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000
ADCHER	ADC_BA+0x24	R/W	ADC通道使能寄存器	0x0000_0000
ADCMR0	ADC_BA+0x28	R/W	ADC比较寄存器0	0x0000_0000
ADCMR1	ADC_BA+0x2C	R/W	ADC比较寄存器1	0x0000_0000
ADSR	ADC_BA+0x30	R/W	ADC状态寄存器	0x0000_0000
ADTDCR	ADC_BA+0x44	R/W	ADC 触发延时控制寄存器	0x0000_0000

5.16.8 NUC029xAN 寄存器描述

ADC数据寄存器 (ADDR0 ~ ADDR7)

寄存器	偏移	R/W	描述	复位值
ADDR0	ADC_BA+0x00	R	ADC 数据寄存器0	0x0000_0000
ADDR1	ADC_BA+0x04	R	ADC数据寄存器1	0x0000_0000
ADDR2	ADC_BA+0x08	R	ADC数据寄存器2	0x0000_0000
ADDR3	ADC_BA+0x0C	R	ADC数据寄存器3	0x0000_0000
ADDR4	ADC_BA+0x10	R	ADC数据寄存器4	0x0000_0000
ADDR5	ADC_BA+0x14	R	ADC数据寄存器5	0x0000_0000
ADDR6	ADC_BA+0x18	R	ADC数据寄存器6	0x0000_0000
ADDR7	ADC_BA+0x1C	R	ADC数据寄存器7	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱						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]	帳隱	保留.
[17]	VALID	有效标志 0 = RSLT里的数据无效 1 = RSLT里的数据有效 当相应通道完成转换后，该位将置1。在ADDR 寄存器被读之后，该位将被硬件清0。 该位只读。
[16]	OVERRUN	溢出标志(只读) 0 = RSLT里的数据没有发生覆盖 1 = RSLT里的数据发生覆盖 如果RSLT的转换值在新的转换结果加载到该寄存器之前还没读走，OVERRUN 置1。 在ADDR寄存器被读之后，该位由硬件清0。
[15:0]	RSLT	A/D转换结果

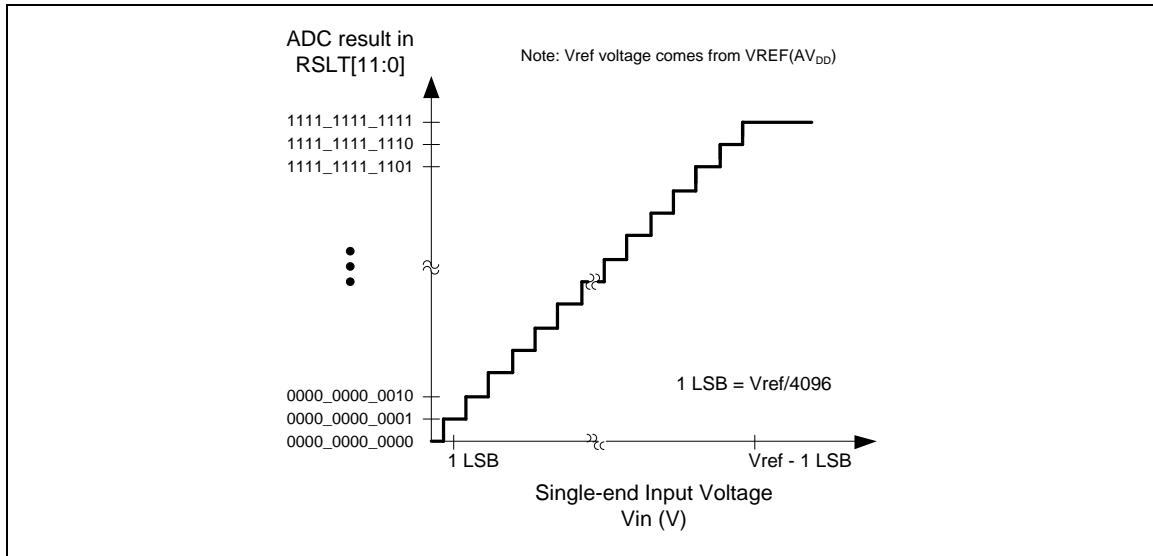
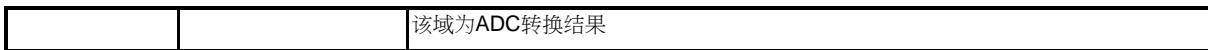


图 5-144 ADC单端输入采样结果映射图

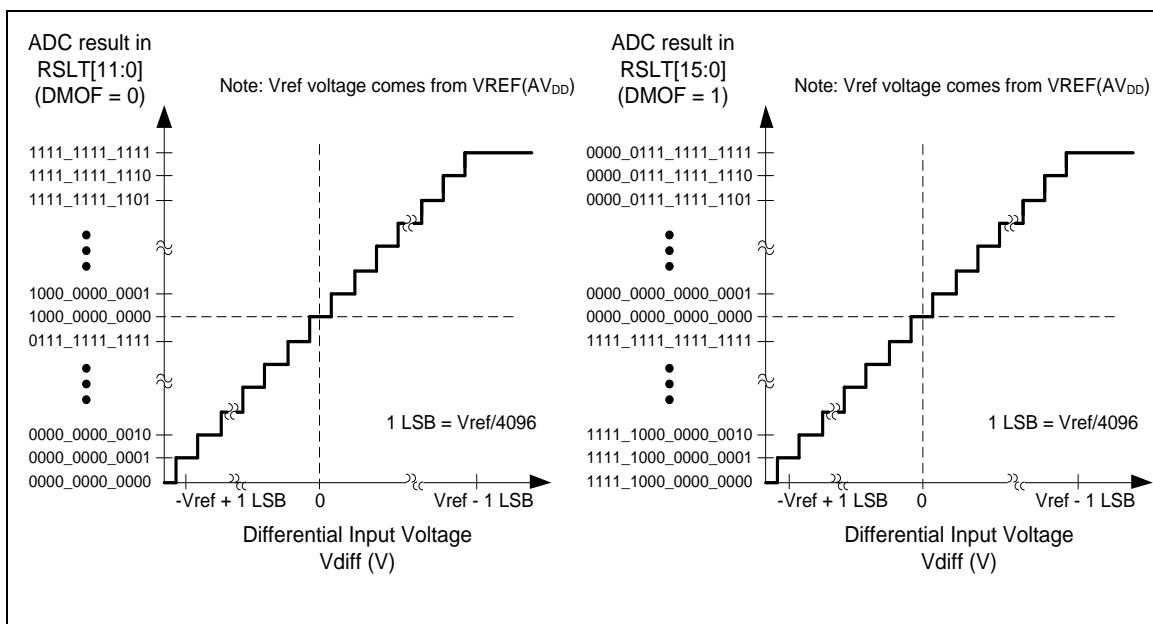


图 5-145 ADC差分输入换换结果映射图

ADC 控制寄存器 (ADCR)

寄存器	偏移	R/W	描述	复位值
ADCR	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
DMOF	帳隱						
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱				ADST	DIFFEN	帳隱	TRGEN
7	6	5	4	3	2	1	0
TRGCOND		TRGS		ADMD		ADIE	ADEN

位	描述		
[31]	DMOF	互补输入模式输出格式 0 = A/D转换结果被存储在ADDRx寄存器的RSLT域，格式为无符号格式（直接二进制格式） 1 = A/D转换结果被存储在ADDRx寄存器的RSLT域，格式为2的补码格式。	
[30:12]	帳隱	保留.	
[11]	ADST	A/D转换开始 ADST位可以通过软件或外部STADC引脚置1。在单次模式和单周期扫描模式结束后，ADST将被硬件自动清0。在连续扫描模式和Burst模式，A/D将持续转换直到软件往该位写0或芯片复位。 0 = 转换停止且A/D转换器进入空闲状态。 1 = 转换开始	
[10]	DIFFEN	互补输入模式控制 0 = 单端模拟输入模式. 1 = 差分模拟输入模式 差分输入配对通道 AIN0 V _{plus} V _{minus} AIN1 AIN2 AIN3 AIN4 AIN5 AIN6 AIN7 差分输入电压 (V_{diff}) = $V_{plus} - V_{minus}$, V_{plus} 为模拟输入; V_{minus} 为反向模拟输入 注: 在差分输入模式下,两个相关的通道中只有偶数通道需要在ADCHER寄存器中使能，转换结果将放置于使能通道相应的数据寄存器里。	

[9]	帳隱	保留.
[8]	TRGEN	<p>外部触发使能控制</p> <p>使能或禁用外部STADC引脚触发A/D转换。如果外部触发使能，ADST位可以通过选择的硬件触发源置1。</p> <p>0= 外部触发禁用 1= 外部触发使能</p> <p>注: ADC外部触发功能只在单周期扫描模式下支持。</p>
[7:6]	TRGCOND	<p>外部触发条件</p> <p>这两位决定外部STADC引脚是电平触发还是边沿触发。在电平触发模式下，触发信号必须保持在平稳状态至少8个PCLK；在边沿触发模式下，触发信号必须保持至少4个PCLK。</p> <p>00 = 低电平 01 = 高电平 10 = 下降沿 11 = 上升沿</p>
[5:4]	TRGS	<p>硬件触发源</p> <p>00 = 外部STADC引脚触发A/D转换 11 = PWM触发A/D转换</p> <p>在改变TRGS之前，软件需清除TRGEN位和ADST位</p>
[3:2]	ADMD	<p>A/D转换器运行模式控制</p> <p>00 = 单次转换 01 = Burst转换 10 = 单周期扫描 11 = 连续扫描</p> <p>注: 在改变操作模式之前，软件需首先清除ADST位。 注: 在Burst模式下，A/D转换结果始终存在数据寄存器0.</p>
[1]	ADIE	<p>A/D中断使能控制</p> <p>如果ADIE位被置1，A/D产生转换完成中断。</p> <p>0 = A/D中断功能禁用 1 = A/D中断功能使能</p>
[0]	ADEN	<p>A/D 转换器使能</p> <p>0 = A/D 转换器禁用 1 = A/D转换器使能</p> <p>注: 在开始A/D转换功能之前，该位需设置为1。该位清0，将禁用A/D转换器模拟电路，以此达到省电的目的。</p>

ADC通道使能寄存器 (ADCHER)

寄存器	偏移	R/W	描述	复位值
ADCHER	ADC_BA+0x24	R/W	ADC通道使能寄存器	0x0000_0000

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

位	描述	
[31:10]	帳隱	保留.
[9:8]	PRESEL[1:0]	<p>模拟通道7输入源选择 00 = 外部模拟输入. 01 = 内部band-gap电压 10 = 内部温度传感器. 11 = 保留. 注: 当ADC通道7的模拟输入源为band-gap 电压，ADC外设时钟频率必须小于300kHz</p>
[7:0]	CHEN	<p>模式输入通道使能控制 设置CHEN[7:0]使能相应模拟输入通道7 ~ 0. 如果DIFFEN位设置为1，只有偶数编号的通道需要使能。 0 = 通道禁用 1 = 通道使能</p>

A/D 比较寄存器0/1 (ADCMPR0/1)

寄存器	偏移	R/W	描述				复位值
ADCMPR0	ADC_BA+0x28	R/W	ADC比较寄存器 0				0x0000_0000
ADCMPR1	ADC_BA+0x2C	R/W	ADC比较寄存器1				0x0000_0000

31	30	29	28	27	26	25	24
悶隱				CMPD			
23	22	21	20	19	18	17	16
CMPD							
15	14	13	12	11	10	9	8
悶隱				CMPMATCNT			
7	6	5	4	3	2	1	0
悶隱		CMPCH			CMPCOND	CMPIE	CMPPEN

位	描述	
[31:28]	悶隱	保留.
[27:16]	CMPD	比较数据 该12位数据用于对比指定通道的转换结果
[15:12]	悶隱	保留.
[11:8]	CMPMATCNT	比较匹配计数 当指定A/D通道模拟转换结果与CMPCOND[2]定义的对比条件匹配，内部匹配计数器加1。当内部计数器达到(CMPMATCNT +1)设定值时，CMPPFx 位将置1.
[7:6]	悶隱	保留.
[5:3]	CMPCH	比较通道选择 000 =选择通道0转换结果进行比较 001 =选择通道1转换结果进行比较 010 =选择通道2转换结果进行比较 011 =选择通道3转换结果进行比较 100 =选择通道4转换结果进行比较 101 =选择通道5转换结果进行比较 110 =选择通道6转换结果进行比较 111 =选择通道7转换结果进行比较
[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 (ADCMPRx[27:16])时，内部匹配计数器加1。 1= 设置比较条件为当12位A/D转换结果大于或的等于12位CMPD (ADCMPRx[27:16])

		时，内部匹配计数器加1。 注: 当内部计数器达 (CMPMATCNT +1), CMPFx位置1
[1]	CMPIE	比较中断使能控制 如果比较功能使能，且比较条件与CMPCOND 和 CMPMATCNT设置的条件匹配， CMPFx位将置1。同时，如果CMPIE设置为1，将产生比较中断请求。 0 = 比较功能中断禁用 1 = 比较功能中断使能
[0]	CMPEN	比较使能控制 当转换数据加载到ADDR寄存器中时，设置该位为1使能ADC控制器比较CMPD[11:0]与指定通道的转换结果。 0 = 比较功能禁用 1 = 比较功能使能

A/D状态寄存器 (ADSR)

寄存器	偏移	R/W	描述	复位值
ADSR	ADC_BA+0x30	R/W	ADC状态寄存器	0x0000_0000

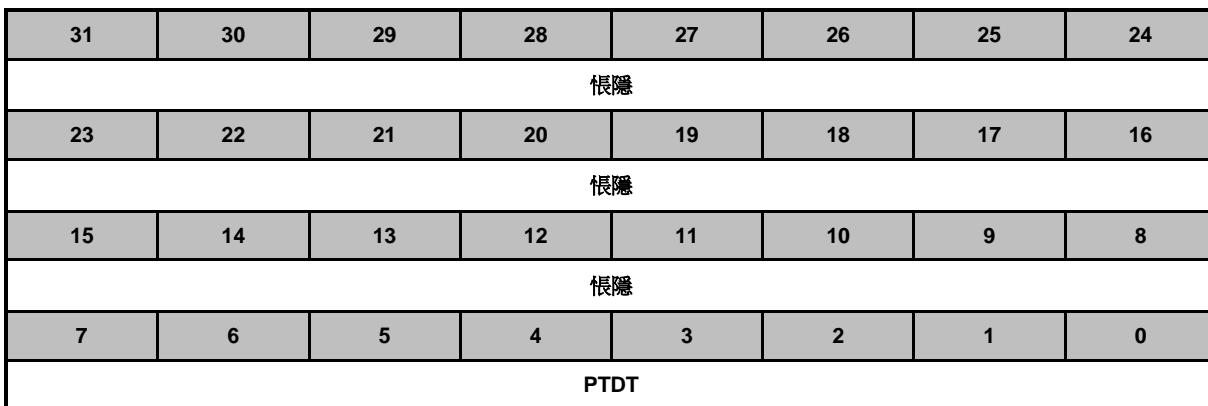
31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
OVERRUN							
15	14	13	12	11	10	9	8
VALID							
7	6	5	4	3	2	1	0
帳隱	CHANNEL			BUSY	CMPF1	CMPF0	ADF

位	描述	
[31:24]	帳隱	保留.
[23:16]	OVERRUN	溢出标志 (只读) 该位为ADDRx 寄存器的OVERRUN 位的镜像 当ADC处于Burst模式下且FIFO溢出，OVERRUN[7:0] 所有位将都被置1。
[15:8]	VALID	数据有效标志 (只读) ADDRx 寄存器的VALID 位的镜像 当ADC处于Burst模式且所有转换结果有效，VALID[7:0] 所有位都将被置1.
[7]	帳隱	保留.
[6:4]	CHANNEL	当前转换通道 如果BUSY = 1，该域反映当前转换通道；如果BUSY = 0，该域反映接下来转换的通道。 该域只读。
[3]	BUSY	忙/空闲 该位为ADCR寄存器的ADST位的镜像，该位只读。 0 = A/D 转换器处于空闲状态. 1 = A/D 转换器正在转换.
[2]	CMPF1	比较标志1 当选择通道的A/D转换结果与ADCMR1设置的条件匹配时，该位置1。该位写1清0。 0 = ADDR 里的转换结果与ADCMR1 设置的不匹配 1 = ADDR 里的转换结果与ADCMR1 设置的匹配
[1]	CMPF0	比较标志0 当选择通道的A/D转换结果与ADCMR0 设置的条件匹配时，该位置1。该位写1清0。

		0 = ADDR里的转换结果与ADCMR0 设置的不匹配 1 = ADDR里的转换结果与ADCMR0 设置的匹配
[0]	ADF	A/D转换完成标志 表明A/D转换结束的一个状态标志，软件可以往该位写1来清0。 ADF在下述三种条件置1： 1. 在单次模式，A/D转换完成 2. 在单周期扫描模式和连续扫描模式，所有指定通道A/D转换完成。 3. 在Burst模式，FIFO超过4个采样值。

ADC触发延时控制寄存器 (ADTDCR)

寄存器	偏移	R/W	描述	复位值
ADTDCR	ADC_BA+0x44	R/W	ADC触发延时控制寄存器	0x0000_0000



位	描述	
[31:8]	帳隱	保留.
[7:0]	PTDT	PWM 触发延时时间 设置该域，在PWM触发之后，将延时开始ADC转换。 PWM触发延时时间为(4 * PTDT) * 系统时钟

5.16.9 NUC029FAE 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基址: ADC_BA = 0x400E_0000				
ADDR	ADC_BA+0x00	R	ADC 数据寄存器0	0x0000_0000
ADCR	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000
ADCHER	ADC_BA+0x24	R/W	ADC通道使能寄存器	0x0000_0000
ADCMR0	ADC_BA+0x28	R/W	ADC 比较寄存器0	0x0000_0000
ADCMR1	ADC_BA+0x2C	R/W	ADC 比较寄存器1	0x0000_0000
ADSR	ADC_BA+0x30	R/W	ADC状态寄存器	0x0000_0000
ADTDCR	ADC_BA+0x44	R/W	ADC触发延时控制寄存器	0x0000_0000
ADSAMP	ADC_BA+0x48	R/W	ADC采样时间计数器寄存器	0x0000_0000

5.16.10 NUC029FAE 寄存器描述

ADC数据寄存器 (ADDR)

寄存器	偏移	R/W	描述	复位值
ADDR	ADC_BA+0x00	R	ADC数据寄存器 0	0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱						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]	帳隱	保留.
[17]	VALID	有效位 0 = RSLT里的数据无效 1 = RSLT里的数据有效 当相应通道转换完成，该位置1。在ADDR寄存器被读之后，该位将被硬件自动清0。 该位只读。
[16]	OVERRUN	溢出标志 (只读) 0 = RSLT 里的数据没有被覆盖 1 = RSLT 里的数据被覆盖 如果在新的转换结果加载到这个寄存器之前，RSLT的转换结果没有被读出来， OVERRUN 将置1。在ADDR寄存器被读之后，该位由硬件清0。
[15:0]	RSLT	A/D转换结果 该域包含ADC转换结果

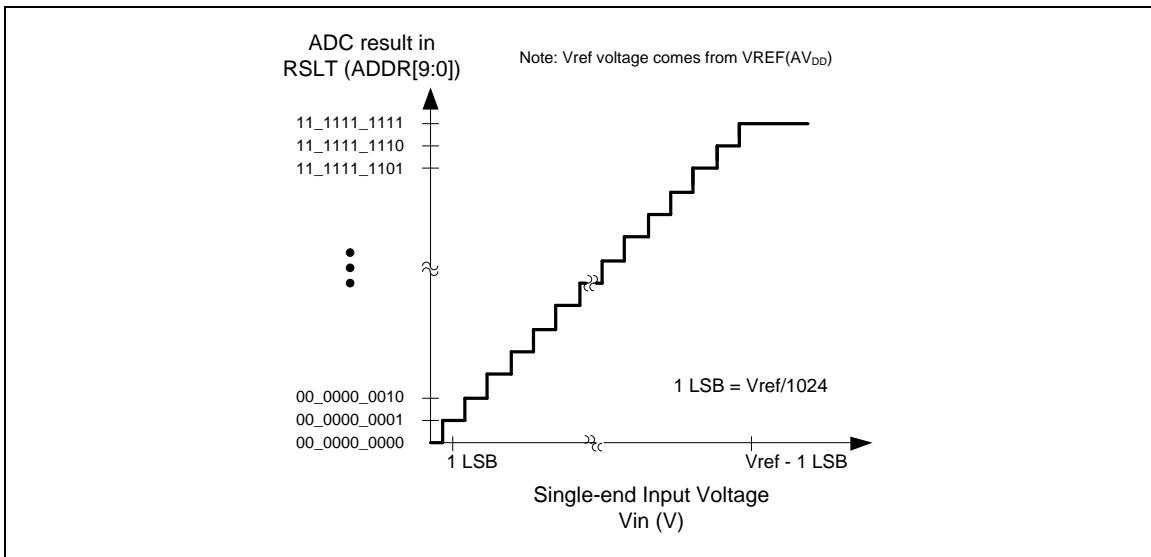


图 5-146 ADC单端输入转换结果映射图

ADC控制寄存器 (ADCR)

寄存器	偏移	R/W	描述	复位值
ADCR	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悶隱							
23	22	21	20	19	18	17	16
悶隱							
15	14	13	12	11	10	9	8
悶隱				ADST	悶隱		TRGEN
7	6	5	4	3	2	1	0
悶隱	TRGCOND	TRGS		悶隱		ADIE	ADEN

位	描述	
[31:12]	悶隱	保留.
[11]	ADST	A/D转换开始 ADST位可通过软件、PWM触发或外部STADC引脚置1。转换完成之后，ADST将由硬件自动清0。 0 = 转换停止且A/D转换器进入空闲状态 1 = 转换开始
[10:9]	悶隱	保留.
[8]	TRGEN	外部触发使能控制 使能或禁止外部STADC引脚触发A/D转换。如果外部触发使能，ADST位可以通过选择的硬件触发源置1。 0= 外部触发禁止 1=外部触发使能
[7]	悶隱	保留.
[6]	TRGCOND	外部触发条件 该位决定外部STADC引脚触发事件为下降沿或上升沿。边沿触发信号必须分别保持高和低状态在平稳状态至少4个PCLK。 0 = 下降沿. 1 = 上升沿.
[5:4]	TRGS	硬件触发源 00 = 外部STADC引脚触发A/D转换 11 = PWM触发A/D转换 其他 = 保留. 注: 在更改TRGS 之前，软件必须禁用TRGEN 和 ADST。

[3:2]	悵隱	保留.
[1]	ADIE	A/D中断使能控制 如果ADIE位置1，将产生A/D转换结束中断请求 0 = A/D中断功能禁用 1 = A/D中断功能使能
[0]	ADEN	A/D转换器使能 0 = A/D 转换器禁用 1 = A/D转换器使能 注: 在开始A/D转换功能之前，该位必须设为1。该位清0将禁用A/D转换器模拟电路，以此达到省电的目的。

ADC通道使能寄存器 (ADCHER)

寄存器	偏移	R/W	描述	复位值
ADCHER	ADC_BA+0x24	R/W	ADC通道使能寄存器	0x0000_0000

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

位	描述	
[31:9]	帳隱	保留.
[8]	PRESEL	模式通道7输入源选择 0 = 保留. 1 = 内部 band-gap 电压. 注: 当ADC通道7的模拟输入源选择band-gap 电压，ADC外设时钟频率必须小于300 kHz.
[7:6]	帳隱	保留.
[5]	CHEN5	模拟输入通道5使能控制 0 = 通道5禁用 1 = 通道5使能
[4]	CHEN4	模拟输入通道4使能控制 0 = 通道4禁用 1 = 通道4使能
[3]	CHEN3	模拟输入通道3使能控制 0 = 通道3禁用 1 = 通道3使能
[2]	CHEN2	模拟输入通道2使能控制 0 = 通道2禁用 1 = 通道2使能
[1:0]	帳隱	保留.

A/D比较寄存器 0/1 (ADCMR0/1)

寄存器	偏移	R/W	描述				复位值
ADCMR0	ADC_BA+0x28	R/W	ADC比较寄存器0				0x0000_0000
ADCMR1	ADC_BA+0x2C	R/W	ADC比较寄存器1				0x0000_0000

31	30	29	28	27	26	25	24
悶隱						CMPD	
23	22	21	20	19	18	17	16
CMPD							
15	14	13	12	11	10	9	8
悶隱				CMPMATCNT			
7	6	5	4	3	2	1	0
悶隱		CMPCH			CMPCOND	CMPIE	CMPEN

位	描述	
[31:26]	悶隱	保留.
[25:16]	CMPD	比较数据 该10位数据用于对比指定通道的转换结果
[15:12]	悶隱	保留.
[11:8]	CMPMATCNT	比较匹配计数 当指定A/D通道模拟转换结果与CMPCOND 设定的对比条件匹配，内部匹配计数器加1。在内部计数器达到 (CMPMATCNT +1)设定值， CMPFx位置1。
[7:6]	悶隱	保留.
[5:3]	CMPCH	比较通道选择 000 = 保留. 001 = 保留. 010 = 选择通道2转换结果进行对比 011 = 选择通道3转换结果进行对比 100 = 选择通道4转换结果进行对比 101 = 选择通道5转换结果进行对比 110 = 保留. 111 = 选择通道7转换结果进行对比
[2]	CMPCOND	比较条件 0 = 设置比较条件为当10位A/D转换结果小于10位CMPD (ADCMR _x [25:16])时，内部匹配计数器加1。 1 = 设置比较条件为当10位A/D转换结果大于或等于10位CMPD (ADCMR _x [25:16])

		时，内部匹配计数器加1。 注: 当内部计数器达到(CMPMATICNT +1)的值时， CMPFx位将置1.
[1]	CMPIE	比较中断使能控制 如果比较功能使能，且比较条件与CMPCOND 和 CMPMATICNT设置的匹配，CMPFx将置1。同时，如果CMPIE置1，将产生比较中断请求。 0 = 比较功能中断禁止 1 = 比较功能中断使能
[0]	CMPEN	比较使能控制 当转换数据加载到ADDR寄存器后，设置该位为1使能ADC控制器比较CMPD[9:0]与指定通道转换结果。 0 = 比较功能禁止 1 = 比较功能使能

A/D 状态寄存器 (ADSR)

寄存器	偏移	R/W	描述	复位值
ADSR	ADC_BA+0x30	R/W	ADC状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
悵隱							
23	22	21	20	19	18	17	16
悵隱							
15	14	13	12	11	10	9	8
悵隱							
7	6	5	4	3	2	1	0
悵隱	CHANNEL			BUSY	CMPF1	CMPF0	ADF

位	描述	
[31:17]	悵隱	保留.
[16]	OVERRUN	溢出标志(只读) ADDR寄存器的OVERRUN (ADSR[16])位的镜像
[15:9]	悵隱	保留.
[8]	VALID	数据有效标志(只读) ADDRx 寄存器的VALID位的镜像
[7]	悵隱	保留.
[6:4]	CHANNEL	当前转换通道(只读) 如果BUSY = 1，该域反映当前转换通道；如果BUSY = 0，该域反映下一个转换通道。
[3]	BUSY	忙/空闲(只读) ADCR 寄存器的ADST位的镜像 0 = A/D转换器处于空闲状态 1 = A/D转换器在转换
[2]	CMPF1	比较标志 1 当选择通道A/D转换结果与ADCMR1设置的条件匹配，该位置1。软件可以向该位写1来清0。 0 = ADDR里的转换结果与ADCMR1 设置的不匹配 1 = ADDR里的转换结果与ADCMR1 设置的匹配
[1]	CMPF0	比较标志 0 当选择通道A/D转换结果与ADCMR0设置的条件匹配，该位置1。软件可以向该位写1来清0。 0 = ADDR里的转换结果与ADCMR0 设置的不匹配

		1 = ADDR里的转换结果与ADCMPRO 设置的匹配
[0]	ADF	A/D转换完成标志 表明A/D转换结束的一个状态标志。当A/D转换结束，ADF置1。软件可以往该位写1来清0。

ADC触发延时控制寄存器 (ADTDCR)

寄存器	偏移	R/W	描述	复位值
ADTDCR	ADC_BA+0x44	R/W	ADC触发延时控制寄存器	0x0000_0000

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

位	描述	
[31:8]	悶隱	保留.
[7:0]	PTDT	PWM触发延时时间 设置该域，将在PWM触发之后延时开始ADC转换。 PWM触发延时时间为 $(4 * \text{PTDT}) * \text{系统时钟}$

ADC采样寄存器 (ADSAMP)

寄存器	偏移	R/W	描述	复位值
ADSAMP	ADC_BA+0x48	R/W	ADC采样时间计数器寄存器	0x0000_0000

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

位	描述	
[31:4]	帳隱	保留.
[3:0]	ADSAMPCNT	<p>ADC 采样计数器</p> <p>如果ADC输入不稳定，用户可以设置该寄存器增加采样时间以此获得一个稳定的ADC输入信号。默认采样时间为1个ADC时钟周期。额外时钟数将用来增加采样时钟周期。</p> <p>0000 = 0 个额外ADC采样时钟 0001 = 1个额外ADC采样时钟 0010 = 2个额外ADC采样时钟 0011 = 4个额外ADC采样时钟 0100 = 8个额外ADC采样时钟 0101 = 16个额外ADC采样时钟 0110 = 32个额外ADC采样时钟 0111 = 64个额外ADC采样时钟 1000 = 128个额外ADC采样时钟 1001 = 256个额外ADC采样时钟 1010 = 512个额外ADC采样时钟 1011 = 1024个额外ADC采样时钟 1100 = 1024个额外ADC采样时钟 1101 = 1024个额外ADC采样时钟 1110 = 1024个额外ADC采样时钟 1111 = 1024个额外ADC采样时钟</p>

5.17 模拟比较器 (ACMP)

5.17.1 概述

NuMicro[®] NUC029系列包含多达4个比较器，可以在不同配置下使用。当正输入电压大于负输入电压，比较器输出逻辑1，否则输出逻辑0。每个比较器可以配置为当比较器输出值改变时产生中断请求。

5.17.2 特性

- 多达四个比较器模拟模块
- 模拟输入电压范围: 0~ V_{DD}
- 支持迟滞功能
- 每个比较器的负输入可选择内部参考电压源
- 四个模拟比较器拥有两个中断向量
- 负节点可选择外部输入或内部band-gap 电压
- 比较结果改变中断
- 掉电唤醒

5.17.3 NUC029xAN方框图

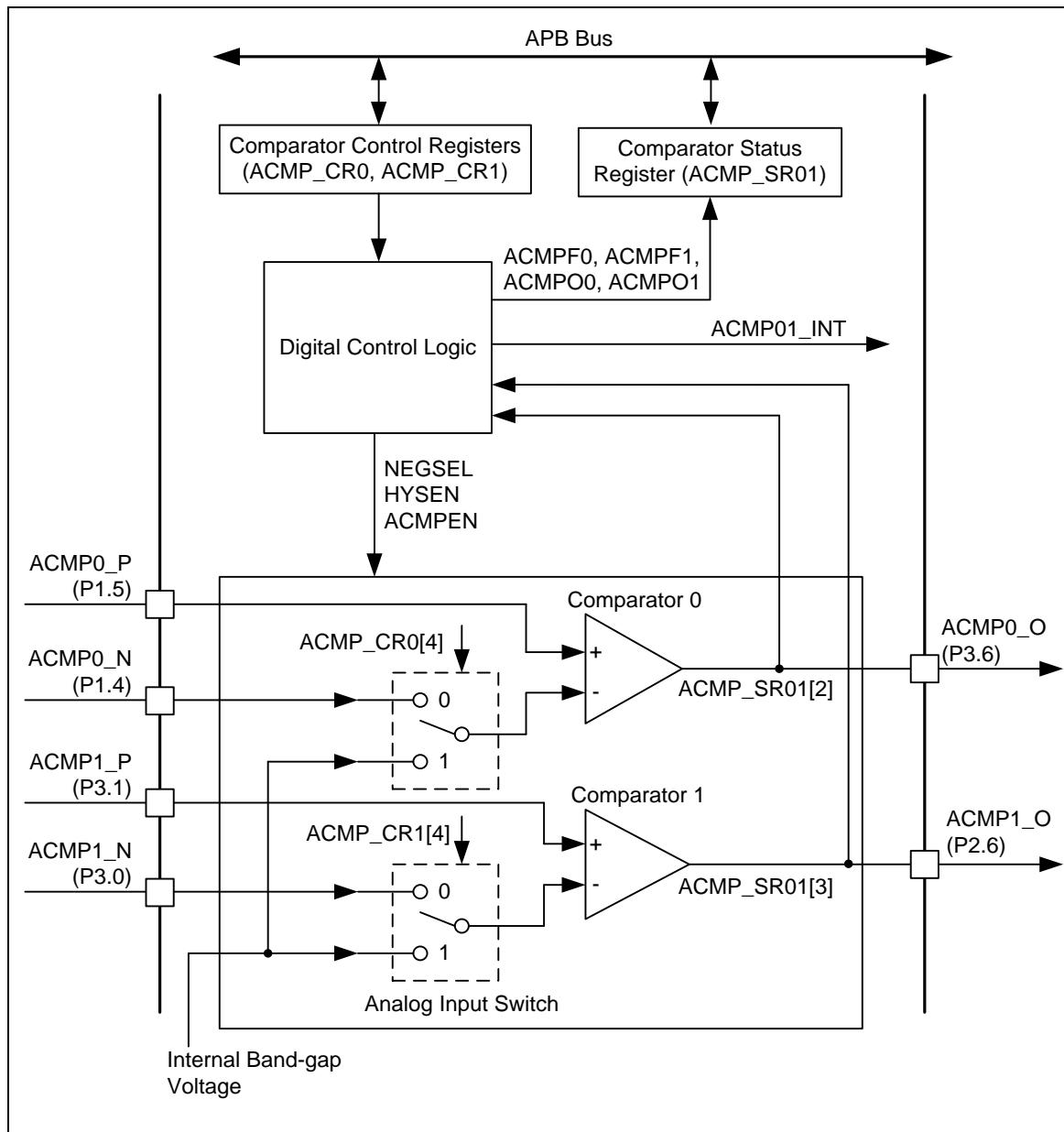


图 5-147 模拟比较器0/1方框图 (NUC029xAN)

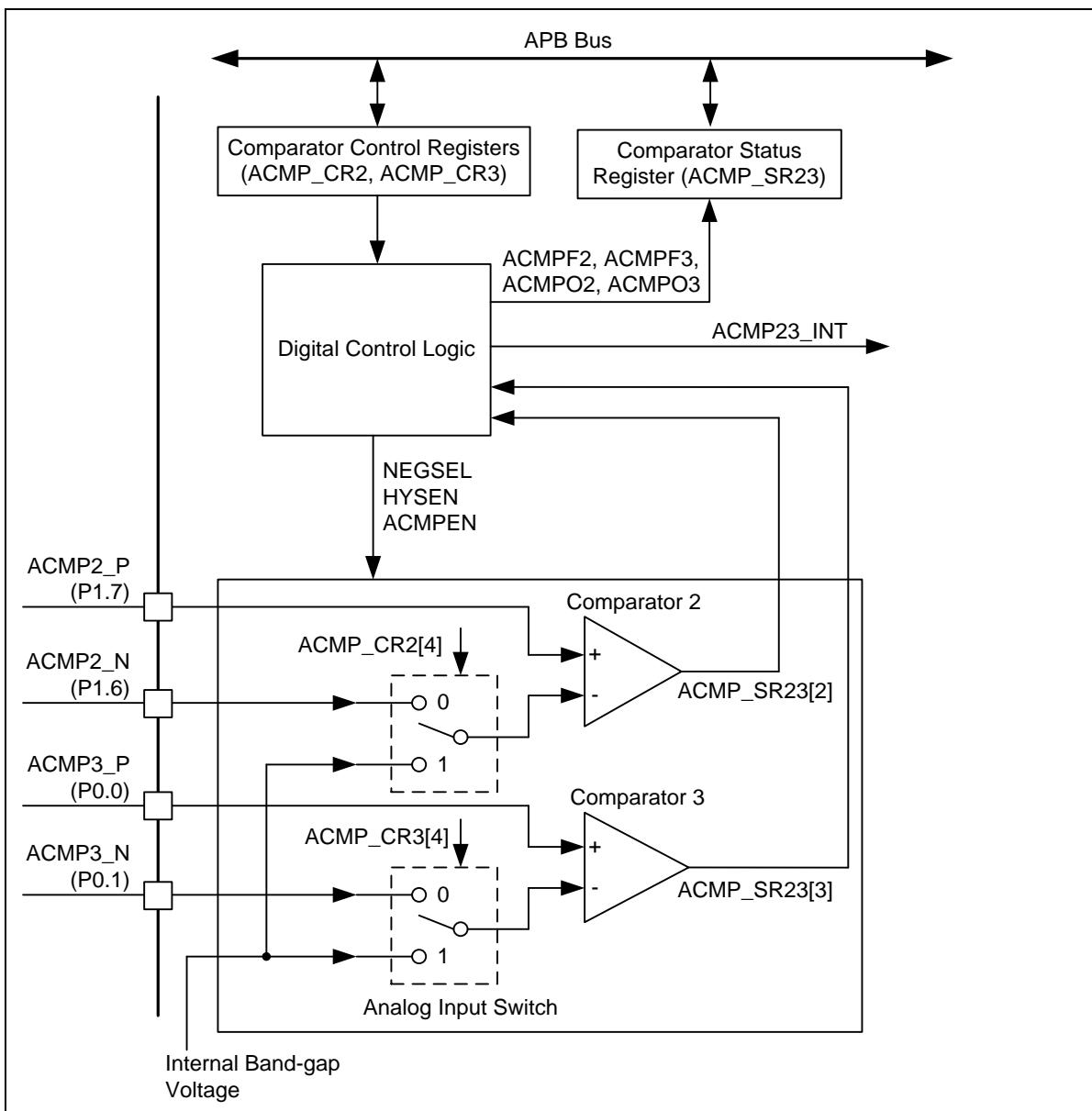


图 5-148 模拟比较器2/3方框图 (NUC029xAN)

5.17.4 NUC029FAE方框图

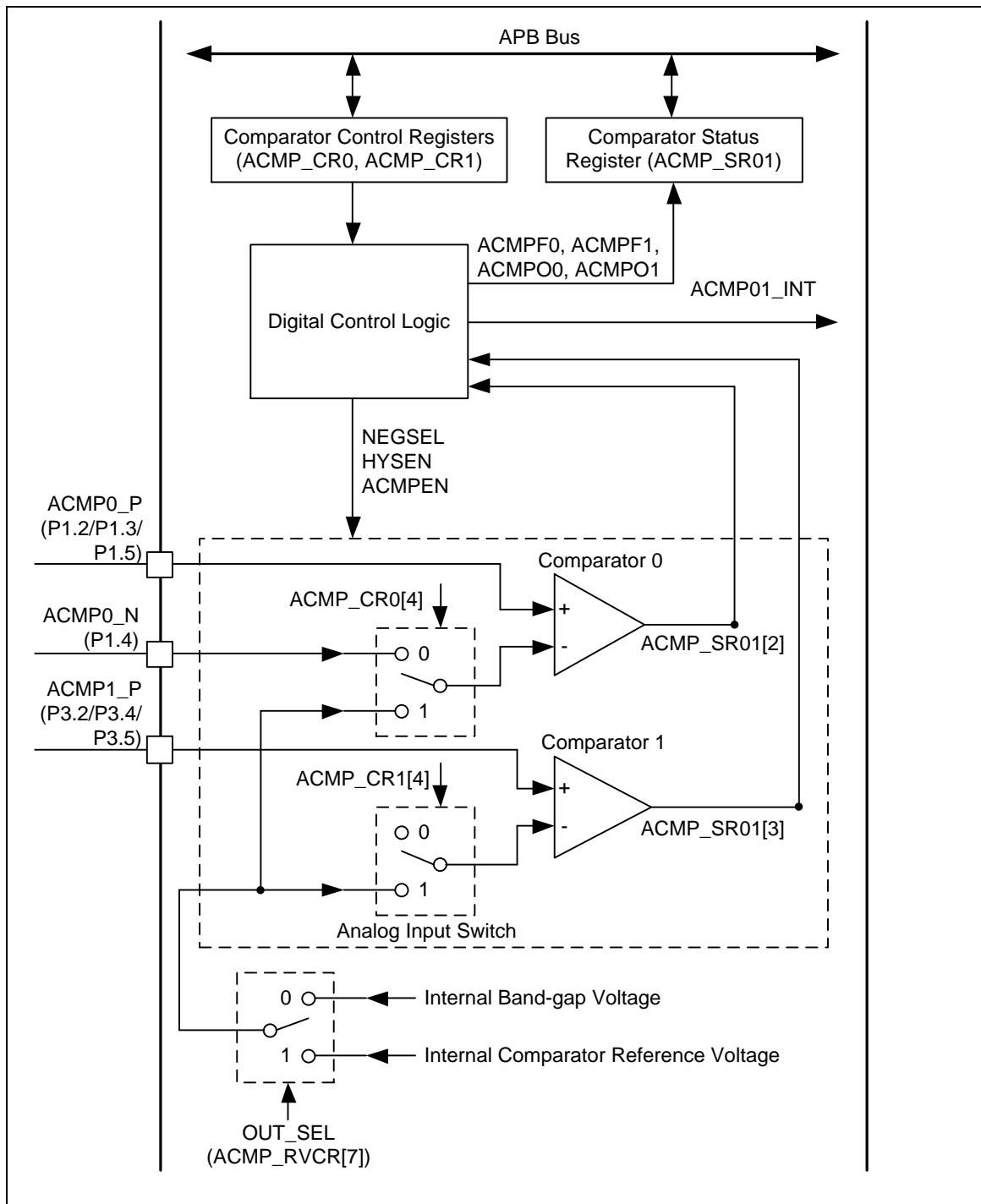


图 5-149 模拟比较器方框图 (NUC029FAE)

5.17.5 基本配置

NUC029xAN:

ACMP引脚功能在P0_MFP, P1_MFP, P2_MFP 和 P3_MFP寄存器中配置。建议禁用模拟输入引脚的数字输入通路，以避免漏电。数字输入通路可通过配置P0_OFFD, P1_OFFD 和 P3_OFFD 寄存器禁用。

ACMP01 和 ACMP23 外设时钟可以通过设置 ACMP01_EN(APBCLK[30]) 和 ACMP23_EN(APBCLK[31]) 为1使能。

NUC029FAE:

ACMP 引脚功能在P1_MFP, P2_MFP 和 P3_MFP寄存器中配置。建议禁用模拟输入引脚的数字输入通路，以避免漏电。数字输入通路可通过配置P1_OFFD 和 P3_OFFD寄存器禁用。

ACMP外设时钟可以通过设置ACMP_EN(APBCLK[30]) 为1使能

5.17.6 功能描述

5.17.6.1 中断源

NUC029xAN:

比较器的输出通过PCLK采样，且反映在ACMP_SR01/23寄存器的ACMPO_x。如果ACMP_CR_x寄存器的ACMPIE 设置为1，比较器中断使能。如果比较器输出状态改变，比较器中断将产生且相应标志ACMPFx将置1。软件可以往该位写1来清0。

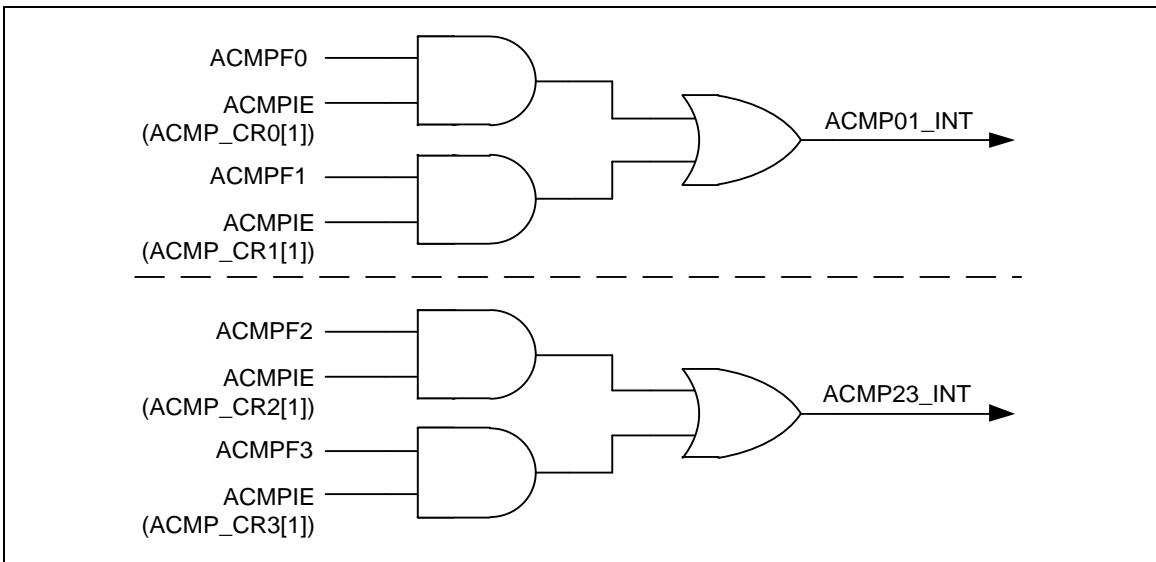


图 5-150 模拟比较器控制器中断源 (NUC029xAN)

NUC029FAE:

比较器的输出通过PCLK采样，且反映在ACMPO_x(ACMP_SR01[3] 和 ACMP_SR01[2])中。如果ACMPIE(ACMP_CR_x[1])置1，比较器中断使能。如果比较器输出状态改变，比较器中断产生且相应的标志ACMPFx(ACMP_SR01[1] 和 ACMP_SR01[0])将置1。软件可以往对应标志位写1来清0对应标志位。

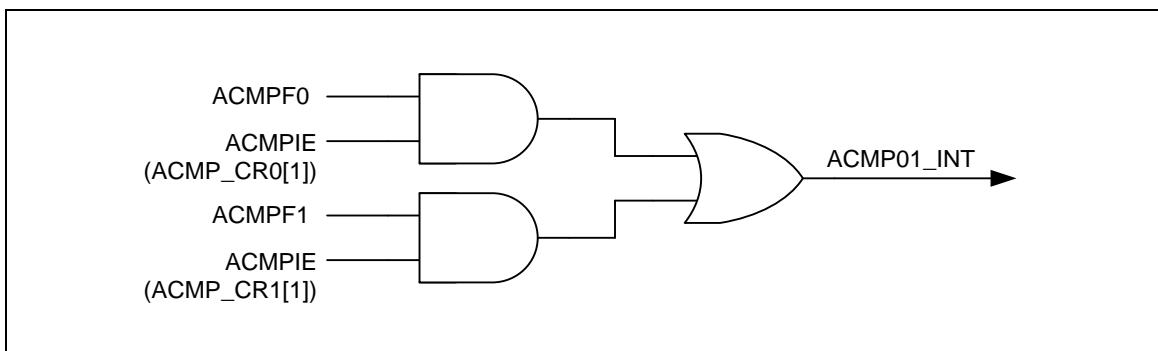


图 5-151 模拟比较器控制器中断源 (NUC029FAE)

5.17.6.2 迟滞功能

模拟比较器提供迟滞功能，使比较器输出转变更平稳。如果比较器输出0，它将不会变成1，直到正输入电压大于负输入电压一个正迟滞电压。同样，如果比较器输出1，它将不会变成0，直到正输入电压掉到负输入电压一个负迟滞电压之下。

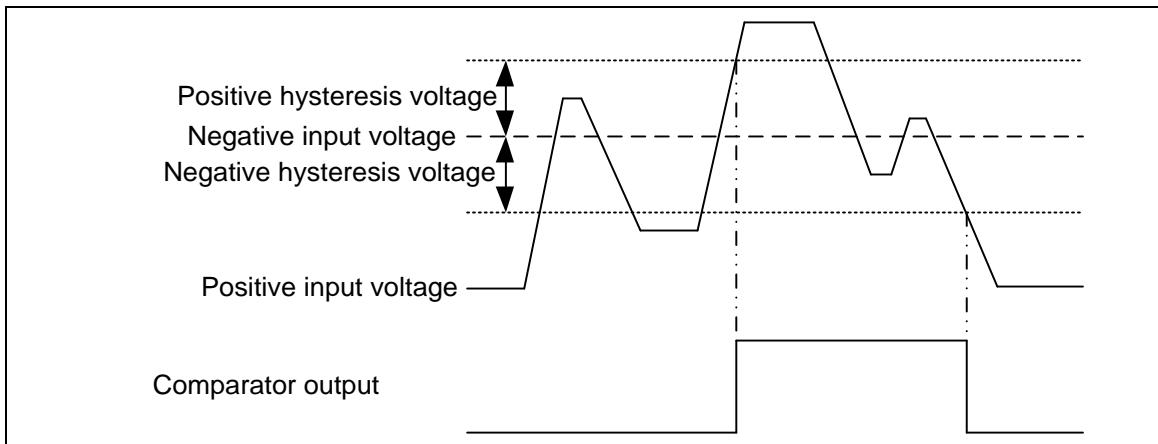


图 5-152 比较器迟滞功能

5.17.7 比较器参考电压(CRV) (NUC029FAE)

5.17.7.1 介绍

比较器参考电压 (CRV) 模块负责为比较器产生参考电压。CRV模块包含梯形电阻和模拟转换器，用户可以使用CRVS(ACMP_RVCR[3:0])设置CRV输出电压，通过设置OUT_SEL(ACMP_RVCR[7])选择ACMP的参考电压。

5.17.7.2 特性

- 用户通过设置CRVS(ACMP_RVCR[3:0])选择参考电压。
- 当设置OUT_SEL(ACMP_RVCR[7]) = 0(选择Band-gap电压)时，将自动禁用梯形电阻以减少功耗。

CRV模块的方框图如下所示：

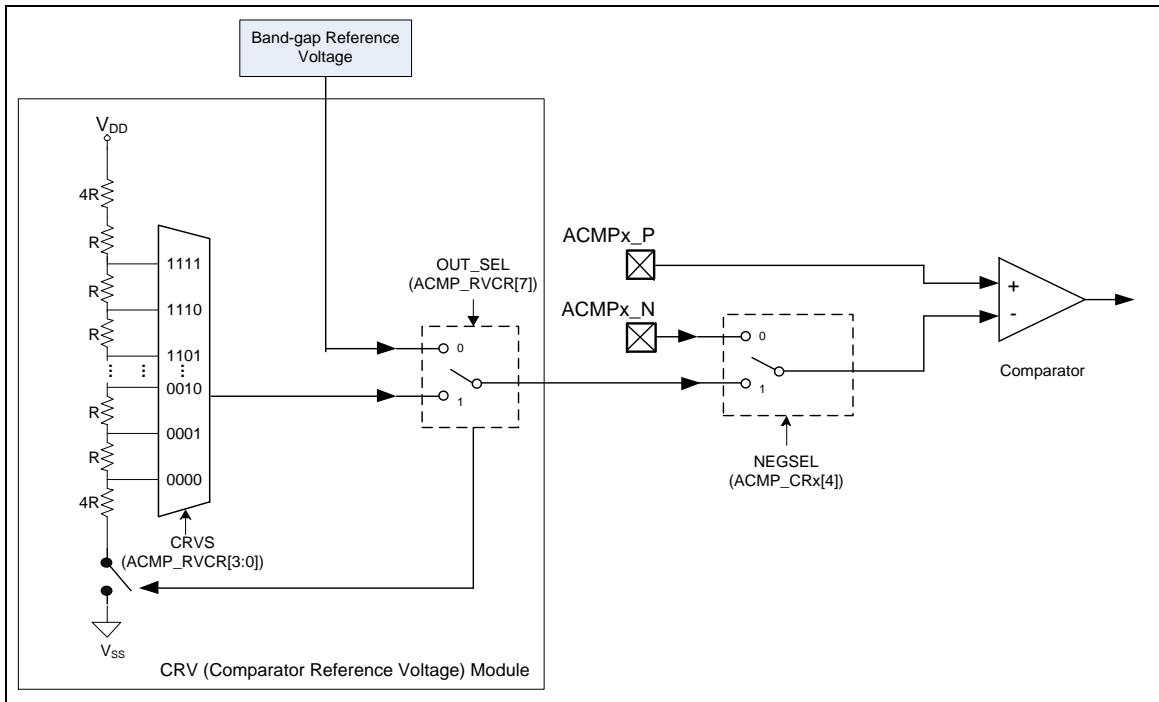


图 5-153 比较器参考电压方框图

5.17.8 NUC029xAN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
ACMP 基地址:				
ACMP01_BA = 0x400D_0000				
ACMP23_BA = 0x401D_0000				
ACMP_CR0	ACMP01_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000
ACMP_CR1	ACMP01_BA+0x04	R/W	模拟比较器1控制寄存器	0x0000_0000
ACMP_SR01	ACMP01_BA+0x08	R/W	模拟比较器0/1状态寄存器	0x0000_0000
ACMP_CR2	ACMP23_BA+0x00	R/W	模拟比较器 2控制寄存器	0x0000_0000
ACMP_CR3	ACMP23_BA+0x04	R/W	模拟比较器3控制寄存器	0x0000_0000
ACMP_SR23	ACMP23_BA+0x08	R/W	模拟比较器2/3状态寄存器	0x0000_0000

5.17.9 NUC029xAN寄存器描述

模拟比较器0 控制寄存器 (ACMP_CR0)

寄存器	偏移	R/W	描述	复位值
ACMP_CR0	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
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
帳隱			NEGSEL	帳隱	HYSN	ACMPIE	ACMPEN

位	描述	
[31:5]	帳隱	保留.
[4]	NEGSEL	模拟比较器0负输入选择 0 = 比较器负输入源来自ACMP0_N引脚 1 = 比较器负输入源来自内部T band-gap电压
[3]	帳隱	保留.
[2]	HYSN	模拟比较器0迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能
[1]	ACMPIE	模拟比较器0中断使能控制 0 = 中断功能禁用 1 = 中断功能使能
[0]	ACMPEN	模拟比较器0使能控制 0 = 模拟比较器0禁用 1 = 模拟比较器0使能 注: 在ACMPEN 位置1后, 模拟比较器输出需要等待2us稳定时间。

模拟比较器1 控制寄存器(ACMP_CR1)

寄存器	偏移	R/W	描述	复位值
ACMP_CR1	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							
15	14	13	12	11	10	9	8
保留							
7	6	5	4	3	2	1	0
保留			NEGSEL	保留	HYSEN	ACMPIE	ACMPEN

位	描述	
[31:5]	保留	保留.
[4]	NEGSEL	模拟比较器1 负输入选择 0 = 比较器负输入源来自ACMP1_N引脚 1 = 比较器负输入源来自内部band-gap电压
[3]	保留	保留.
[2]	HYSEN	模拟比较器1迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能
[1]	ACMPIE	模拟比较器1中断使能控制 0 = 中断功能禁用 1 = 中断功能使能
[0]	ACMPEN	模拟比较器1使能控制 0 = 模拟比较器1禁用 1 = 模拟比较器1使能 注: 在ACMPEN 置1后, 模拟比较器输出需要等待2us稳定时间。

模拟比较器0/1状态寄存器 (ACMP_SR01)

寄存器	偏移	R/W	描述	复位值
ACMP_SR01	ACMP01_BA+0x08	R/W	模拟比较器 0/1 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				ACMPO1	ACMPO0	ACMPF1	ACMPF0

位	描述	
[31:4]	保留	保留.
[3]	ACMPO1	模拟比较器 1 输出 与 APB 时钟同步，允许通过软件读。当比较器 1 禁用(ACMP_CR1[0] = 0)时，清 0。 0 = 模拟比较器 1 输出 0 1 = 模拟比较器 1 输出 1
[2]	ACMPO0	模拟比较器 0 输出 与 APB 时钟同步，允许通过软件读。当比较器 0 禁用(ACMP_CR0[0] = 0)时，清 0。 0 = 模拟比较器 0 输出 0 1 = 模拟比较器 0 输出 1
[1]	ACMPF1	模拟比较器 1 标志 每当比较器 1 输出状态改变，该位将由硬件设置。如果 ACMP_CR1[1] 设置为 1，将产生一个中断。 0 = 模拟比较器 1 输出没有改变 1 = 从该位清 0 后，模拟比较器 1 输出改变 注：该位写 1 清 0
[0]	ACMPF0	模拟比较器 0 标志 每当比较器 0 输出状态改变，该位将由硬件设置。如果 ACMP_CR0[1] 置 1，将产生一个中断。 0 = 模拟比较器 0 输出没有改变 1 = 从该位清 0 后，模拟比较器 0 输出改变 注：该位写 1 清 0

模拟比较器 2 控制寄存器(ACMP_CR2)

寄存器	偏移	R/W	描述	复位值
ACMP_CR2	ACMP23_BA+0x00	R/W	模拟比较器2控制寄存器	0x0000_0000

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

位	描述		
[31:5]	悶隱	保留.	
[4]	NEGSEL	模拟比较器2负输入选择 0 = 比较器负输入源来自ACMP2_N引脚 1 = 比较器负输入源来自内部band-gap电压	
[3]	悶隱	保留.	
[2]	HYSEN	模拟比较器2迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能	
[1]	ACMPIE	模拟比较器2中断使能控制 0 = 中断功能禁用 1 = 中断功能使能	
[0]	ACMPEN	模拟比较器2使能控制 0 = 模拟比较器2禁用 1 = 模拟比较器2使能 注: 在ACMPEN置1后, 模拟比较器输出需要等待2us稳定时间。	

模拟比较器3控制寄存器 (ACMP_CR3)

寄存器	偏移	R/W	描述	复位值
ACMP_CR3	ACMP23_BA+0x04	R/W	模拟比较器3控制寄存器	0x0000_0000

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

位	描述	
[31:5]	悶隱	保留.
[4]	NEGSEL	模拟比较器3负输入选择 0 = 比较器负输入源来自ACMP3_N 引脚 1 = 比较器负输入源来自内部band-gap电压
[3]	悶隱	保留.
[2]	HYSEN	模拟比较器3迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能
[1]	ACMPIE	模拟比较器3中断使能控制 0 = 中断功能禁用 1 = 中断功能使能
[0]	ACMPEN	模拟比较器3使能控制 0 = 模拟比较器3禁用 1 = 模拟比较器3使能 注: 在ACMPEN 位置1后, 模拟比较器输出需要等待2us稳定时间

模拟比较器2/3状态寄存器(ACMP_SR23)

寄存器	偏移	R/W	描述	复位值
ACMP_SR23	ACMP23_BA+0x08	R/W	模拟比较器2/3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				ACMPO3	ACMPO2	ACMPF3	ACMPF2

位	描述	
[31:4]	保留	保留.
[3]	ACMPO3	模拟比较器3输出 与APB时钟同步且允许通过软件读。当比较器3禁用(ACMP_CR3[0] = 0)时清0. 0 = 模拟比较器3输出0 1 = 模拟比较器3输出1
[2]	ACMPO2	模拟比较器2 输出 与APB时钟同步且允许通过软件读。当比较器2禁用(ACMP_CR2[0] = 0)时清0. 0 = 模拟比较器2输出0 1 = 模拟比较器2输出1
[1]	ACMPF3	模拟比较器3标志 每当比较器3输出状态改变，该位将由硬件设置。如果ACMP_CR3[1] 设为1，将产生中断 0 = 模拟比较器3输出没有改变 1 = 自该位清0后，模拟比较器3输出改变。 注: 改为写1清0
[0]	ACMPF2	模拟比较器2标志 每当比较器2输出状态改变，该位将由硬件设置。如果ACMP_CR2[1] 置1，将产生中断。 0 = 模拟比较器2输出没有改变 1 = 自该位清0后，模拟比较器2输出改变。 注: 该位写1清0

5.17.10 NUC029FAE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
ACMP 基地址:				
ACMP_BA = 0x400D_0000				
ACMP_CR0	ACMP_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000
ACMP_CR1	ACMP_BA+0x04	R/W	模拟比较器1控制寄存器	0x0000_0000
ACMP_SR01	ACMP_BA+0x08	R/W	模拟比较器0/1状态寄存器	0x0000_0000
ACMP_RVCR	ACMP_BA+0x0C	R/W	模拟比较器参考电压控制寄存器	0x0000_0000

5.17.11 NUC029FAE 寄存器描述

模拟比较器0控制寄存器 (ACMP_CR0)

寄存器	偏移	R/W	描述	复位值
ACMP_CRO	ACMP_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
帳隱	CPP0SEL		帳隱				
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱						FALLING	RISING
7	6	5	4	3	2	1	0
帳隱			NEGSEL	帳隱	HYSEN	ACMPIE	ACMPEN

位	描述	
[31]	保留	保留.
[30:29]	CPP0SEL[1:0]	<p>模拟比较器0正输入选择 00 = CPP0来自P1.5引脚 01 = CPP0来自P1.0引脚 10 = CPP0来自P1.2引脚 11 = CPP0来自P1.3引脚</p>
[28:10]	保留	保留.
[9]	FALLING	<p>模拟比较器0下降沿触发使能控制 0 = 模拟比较器0下降沿触发PWM或定时器使能。 1 = 模拟比较器0下降沿触发禁用 注: 该位只有在模拟比较器0触发PWM或定时器时有效。</p>
[8]	RISING	<p>模拟比较器0上升沿触发使能控制 0 = 模拟比较器0上升沿触发PWM或定时器使能 1 = 模拟比较器0上升沿触发禁用 注: 该位只有在模拟比较器0触发PWM或定时器时有效。</p>
[7:5]	保留	保留.
[4]	NEGSEL	<p>模拟比较器0负输入选择 0 = 比较器负输入源来自CPN0引脚。 1 = 比较器负输入源来自内部band-gap 电压或比较器参考电压。</p>
[3]	保留	保留.

[2]	HYSEN	模拟比较器0迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能
[1]	ACMPIE	模拟比较器0中断使能控制 0 = 中断功能禁用 1 = 中断功能使能
[0]	ACMPEN	模拟比较器0使能控制 0 = 模拟比较器0禁用 1 = 模拟比较器0使能 注: 在该位置1后，模拟比较器输出需要等待2us稳定时间。

模拟比较器1控制寄存器 (ACMP_CR1)

寄存器	偏移	R/W	描述	复位值
ACMP_CR1	ACMP_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
帳隱	CPP1SEL		帳隱				
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱						FALLING	RISING
7	6	5	4	3	2	1	0
帳隱			NEGSEL	帳隱	HYSEN	ACMPIE	ACMPEN

位	描述	
[31]	保留	保留.
[30:29]	CPP1SEL[1:0]	<p>模拟比较器1正输入选择</p> <p>00 = CPP1来自P3.1引脚</p> <p>01 = CPP1来自P3.2引脚</p> <p>10 = CPP1来自P3.4引脚</p> <p>11 = CPP1来自P3.5引脚</p>
[28:10]	保留	保留.
[9]	FALLING	<p>模拟比较器1下降沿触发使能控制</p> <p>0 = 模拟比较器1下降沿触发PWM或定时器使能</p> <p>1 = 模拟比较器1下降沿触发禁用</p> <p>注: 该位只有在模拟比较器1触发PWM或定时器时有效。</p>
[8]	RISING	<p>模拟比较器1上升沿触发使能控制</p> <p>0 = 模拟比较器1上升沿触发PWM或定时器使能</p> <p>1 = 模拟比较器1上升沿触发禁用</p> <p>注: 该位只有在模拟比较器1触发PWM或定时器时有效。</p>
[7:5]	保留	保留.
[4]	NEGSEL	<p>模拟比较器1负输入选择</p> <p>0 = 比较器的负输入源来自CPN1引脚</p> <p>1 = 比较器负输入源来自内部band-gap电压或比较器参考电压</p>
[3]	保留	保留.

[2]	HYSEN	模拟比较器1迟滞使能控制 0 = 迟滞功能禁用 1 = 迟滞功能使能
[1]	ACMPIE	模拟比较器1中断使能控制 0 = 中断功能禁用 1 = 中断功能使能
[0]	ACMPEN	模拟比较器1使能控制 0 = 模拟比较器1禁用 1 = 模拟比较器1使能 注: 在该位置1后，模拟比较器输出需要等待2us稳定时间

模拟比较器0/1状态寄存器 (ACMP_SR01)

寄存器	偏移	R/W	描述				复位值
ACMP_SR01	ACMP_BA+0x08	R/W	模拟比较器0/1状态寄存器				0x0000_0000

31	30	29	28	27	26	25	24
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
帳隱				ACMPO1	ACMPO0	ACMPF1	ACMPF0

位	描述	
[31:4]	帳隱	保留.
[3]	ACMPO1	模拟比较器1输出 与APB时钟同步且允许通过软件读。当比较器1禁用 (ACMPEN(ACMP_CR1[0]) = 0) 时清0。 0 = 模拟比较器1输出0 1 = 模拟比较器1输出1
[2]	ACMPO0	模拟比较器0输出 与APB时钟同步且允许通过软件读。当比较器0禁用 (ACMPEN(ACMP_CR0[0]) = 0) 时清0。 0 = 模拟比较器0输出0 1 = 模拟比较器0输出1
[1]	ACMPF1	模拟比较器1标志 每当比较器1输出状态改变，该位将由硬件设置。如果ACMPIE(ACMP_CR1[1]) = 1，将产生中断。 0 = 模拟比较器1输出没有改变 1 = 模拟比较器1输出改变 注: 该位可通过软件写1清0
[0]	ACMPF0	模拟比较器0标志 每当比较器0输出状态改变，该位将由硬件设置。如果ACMPIE(ACMP_CR0[1]) = 1，将产生中断。 0 = 模拟比较器0输出没有改变 1 = 模拟比较器0输出改变 注: 该位可通过软件写1清0

模拟比较器参考电压控制寄存器 (ACMP_RVCR)

寄存器	偏移	R/W	描述					复位值
ACMP_RVCR	ACMP_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	
OUT_SEL	保留			CRVS				

位	描述	
[31:8]	保留	保留.
[7]	OUT_SEL	CRV模块输出选择 0 = Band-gap电压 1 = 内部比较器参考电压
[6:4]	保留	保留.
[3:0]	CRVS[3:0]	比较器参考电压设置 比较器参考电压= $V_{DD} * (1 / 6 + CRVS[3:0] / 24)$.

5.18 硬件除法器 (HDIV) (NUC029xAN)

5.18.1 概述

NuMicro[®] NUC029xAN 有硬件除法器 (HDIV). HDIV在高性能应用场合很有用。该硬件除法器是一个有符号的整数除法器，能输出商和余数。

5.18.2 特性

- 有符号（2的补码）整数计算
- 32位被除数，16位除数计算能力
- 32位商和32位余数输出（16位余数加上符号扩展到32位）
- 除0警告标志
- 一次计算花费6个HCLK时钟
- 写除数触发计算
- 当读商和余数时将自动等待计算完成

5.18.3 基本配置

在使用硬件除法器之前，硬件除法器的时钟必须被使能。使能硬件除法器需要设置HDIV_EN (AHBCLK[4]) 为1。

5.18.4 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。如果CPU在硬件除法器计算完成之前读DIVQUO 或 DIVREM，CPU将等待直到硬件除法器计算完成。因此，在触发硬件除法器计算之后，CPU可以总是得到有效的结果而不需要软件延时。

如果除数为0，DIVSTS 的DIV0标志将被置1.

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。

下图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。

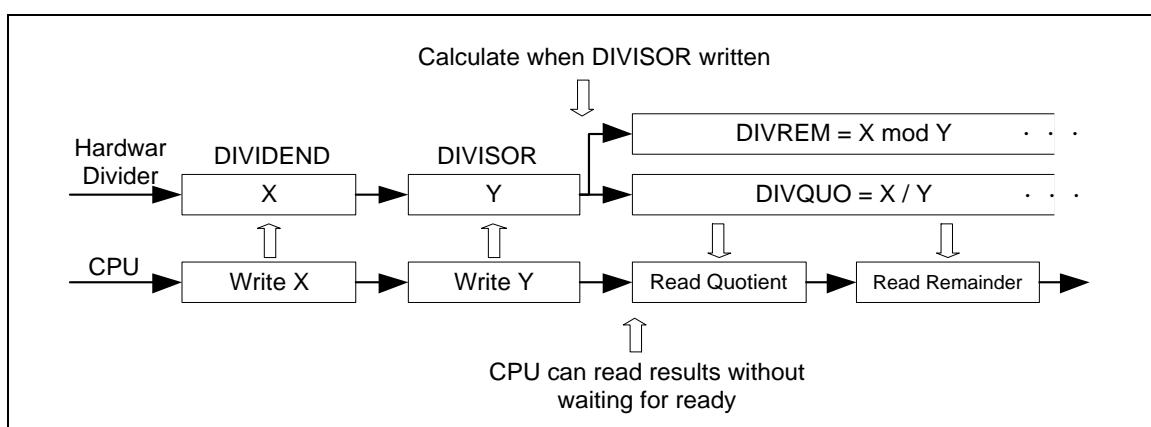


图 5-154 硬件除法器操作流程

5.18.5 寄存器描述

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移	R/W	描述	复位值
HDIV 基地址: HDIV_BA = 0x5001_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

5.18.6 寄存器描述

被除数源寄存器 (DIVIDEND)

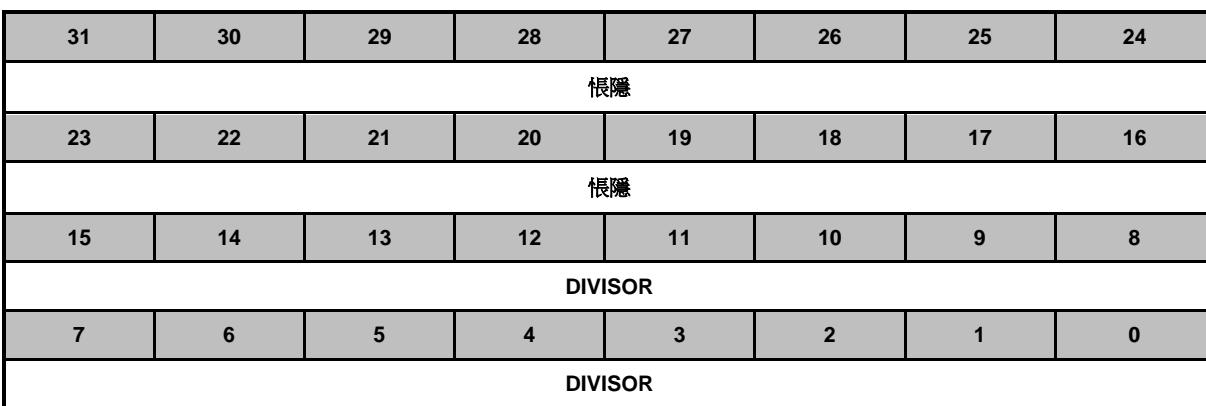
描述	偏移	R/W	描述	复位值
DIVIDEND	HDIV_BA+0x00	R/W	被除数源寄存器	0x0000_0000

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 该寄存器在除法器启动计算前给予被除数

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



位	描述	
[31:16]	保留	保留.
[15:0]	DIVISOR	<p>除数源 该寄存器在除法器启动计算前给予除数 注: 当该寄存器被写之后, 硬件除法器将开始计算。</p>

商结果寄存器 (DIVQUO)

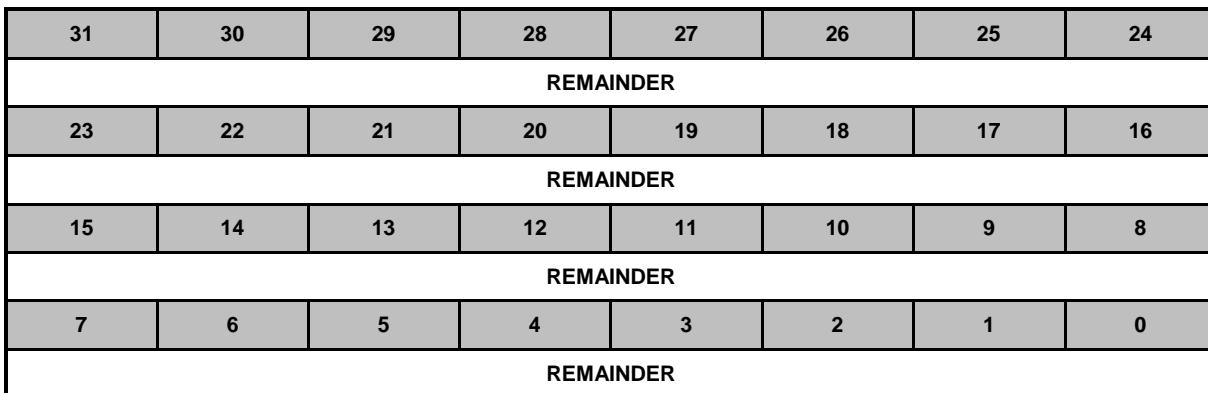
寄存器	偏移	R/W	描述	复位值
DIVQUO	HDIV_BA+0x08	R/W	商结果寄存器	0x0000_0000

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 在计算完成之后，该寄存器保存除法器的商。

余数结果寄存器 (DIVREM)

寄存器	偏移	R/W	描述	复位值
DIVREM	H DIV_BA+0x0C	R/W	余数结果寄存器	0x0000_0000



位	描述							
[31:16]	REMAINDER[31:16]	REMAINDER[15:0]符号扩展 硬件除法器的余数是一个16位有符号整数(REMAINDER[15:0])，通过符号扩展(REMAINDER[31:16])到32位整数。						
[15:0]	REMAINDER[15:0]	余数结果 在计算完成之后，该寄存器保存除法器的余数。						

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
帳隱							
23	22	21	20	19	18	17	16
帳隱							
15	14	13	12	11	10	9	8
帳隱							
7	6	5	4	3	2	1	0
帳隱						DIV0	帳隱

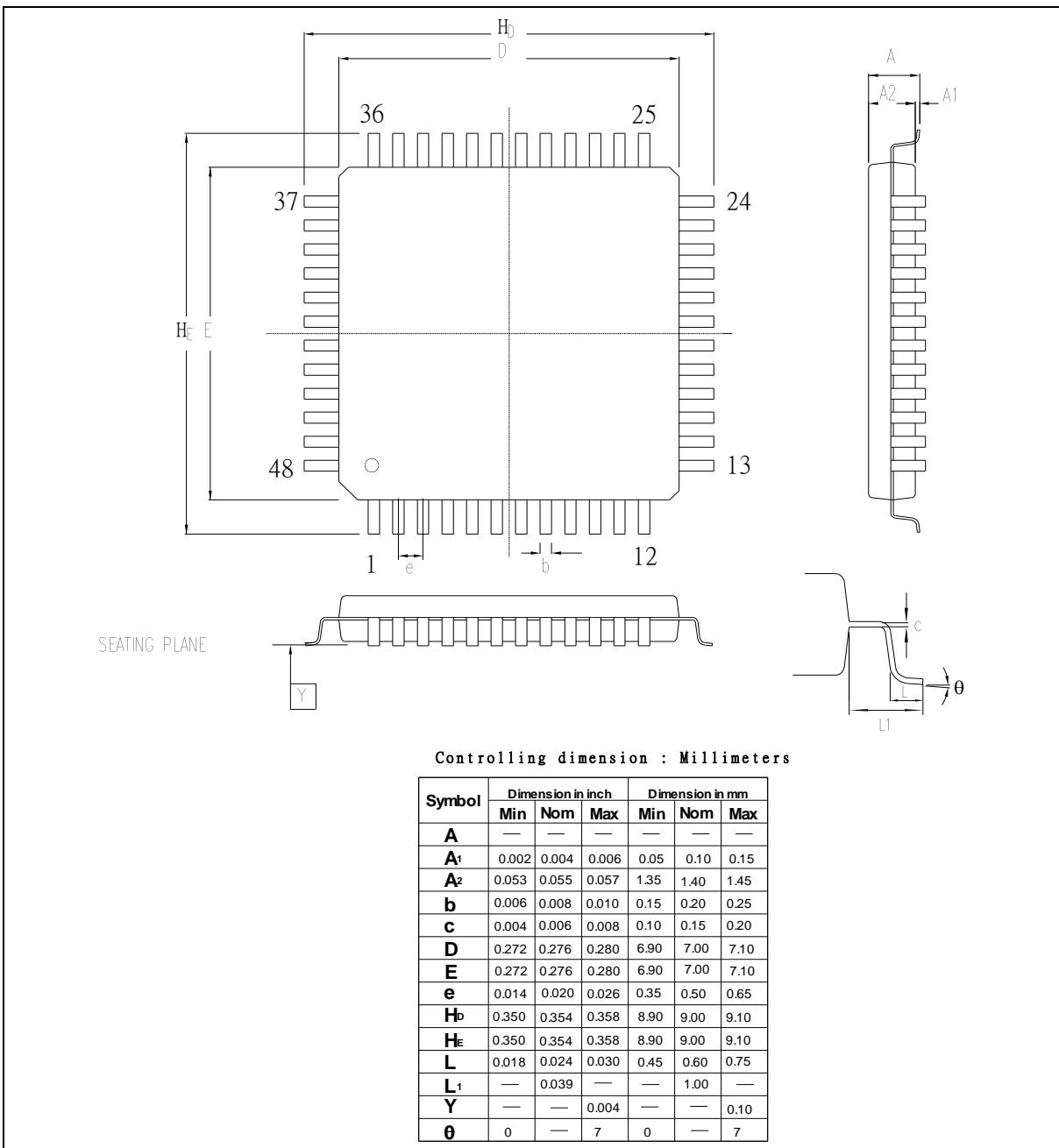
位	描述	
[31:2]	帳隱	保留.
[1]	DIV0	除数为0警告 0 = 除数不是0. 1 = 除数是0. 注: DIV0标志用于表明被0除的情况，在DIVISOR 被写后更新。该寄存器只读。
[0]	帳隱	保留.

6 电气特性

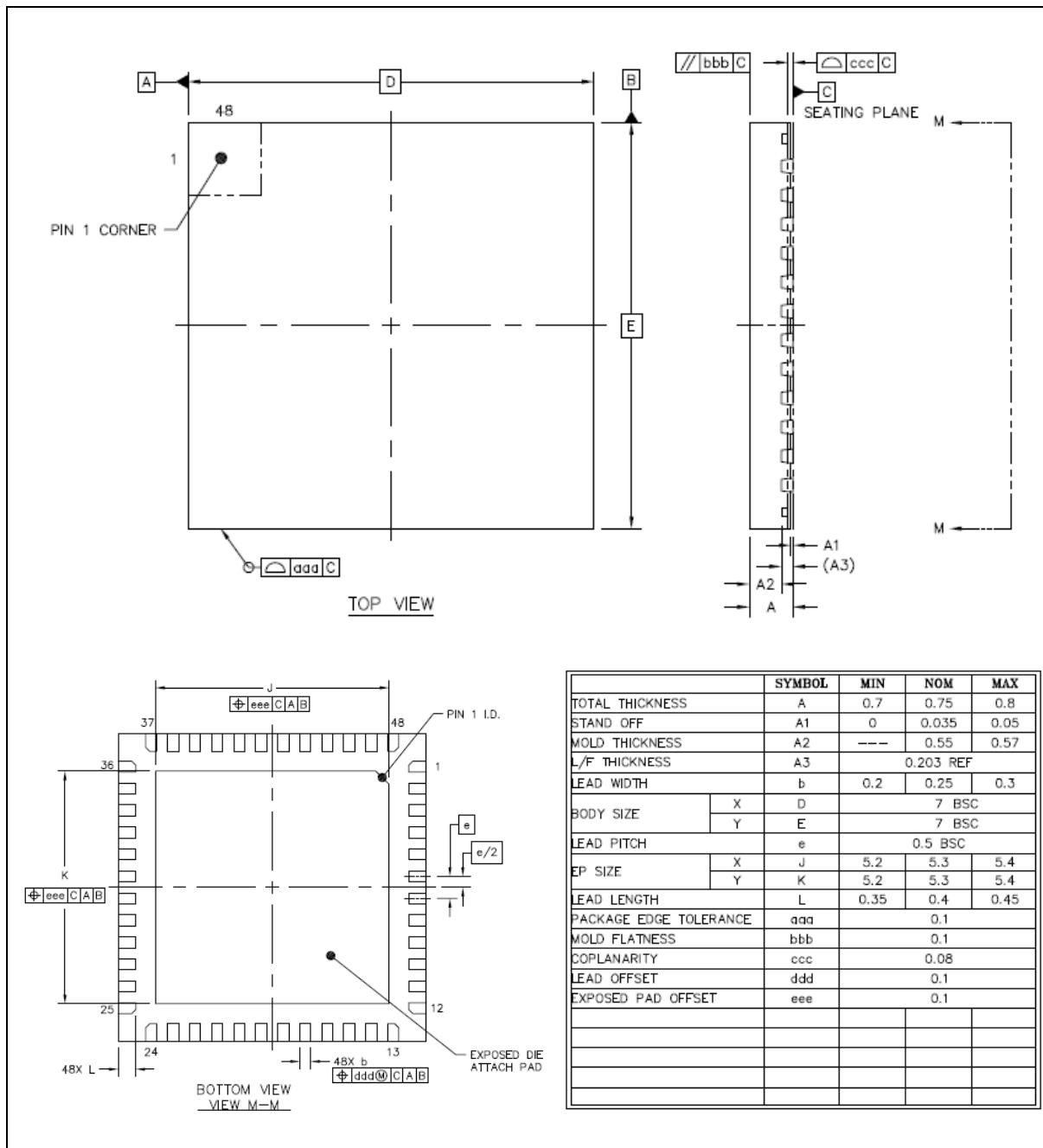
NuMicro[®] NUC029系列电气特性信息请参考NuMicro[®] NUC029系列数据手册。

7 封装尺寸

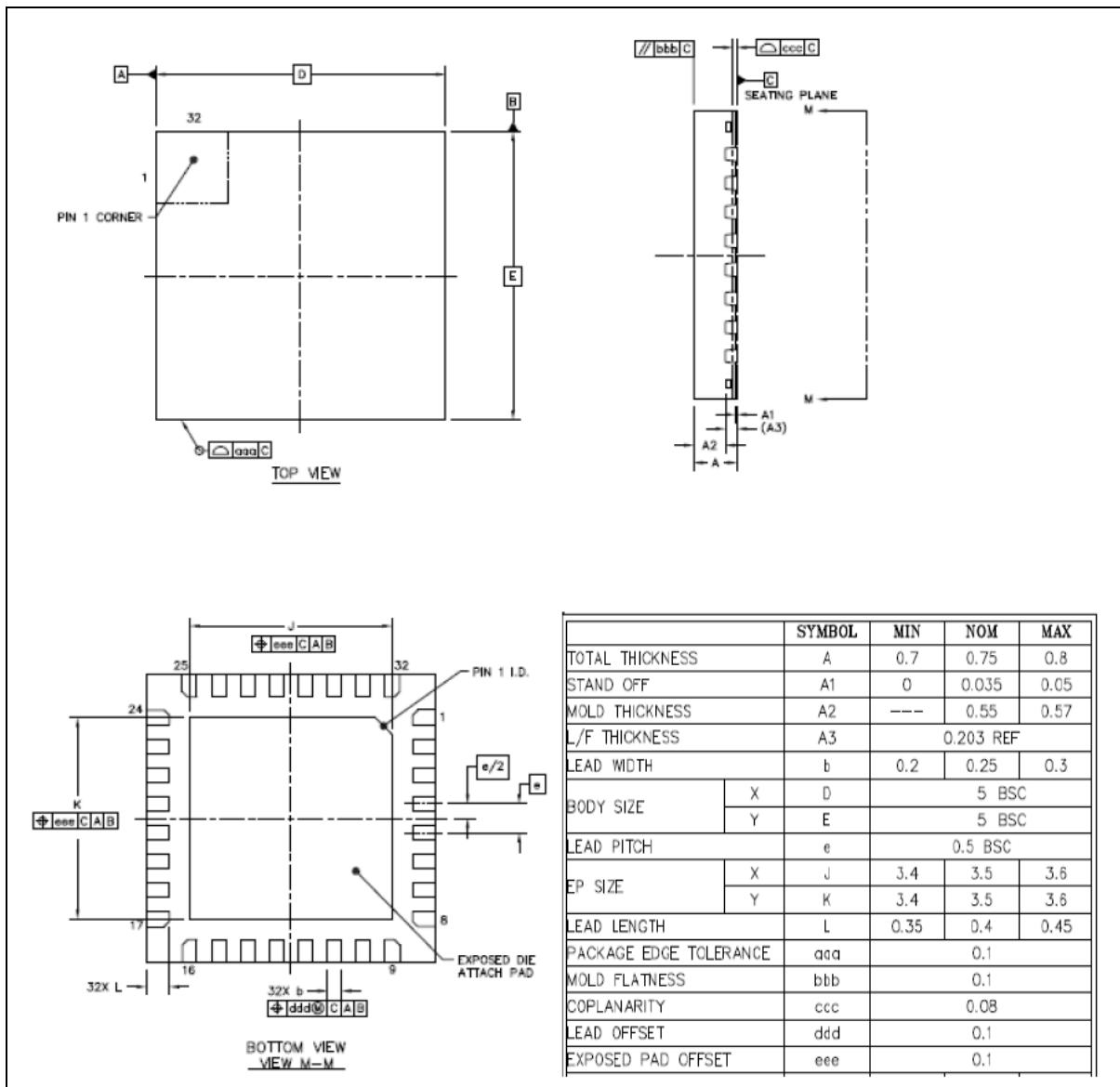
7.1 48-pin LQFP (7x7x1.4 mm)



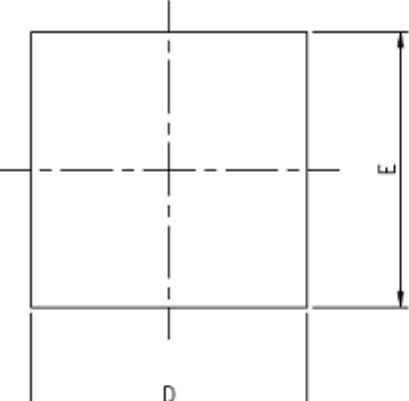
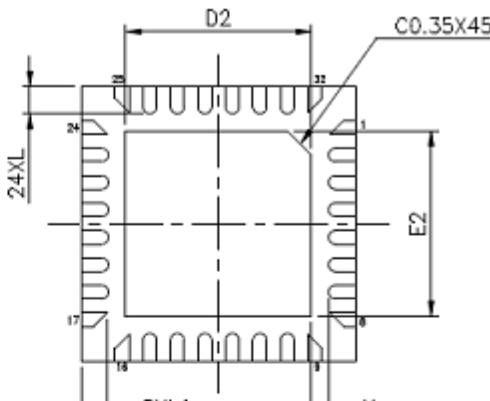
7.2 48-pin QFN (7x7x0.8 mm)



7.3 33-pin QFN (5x5x0.8 mm)

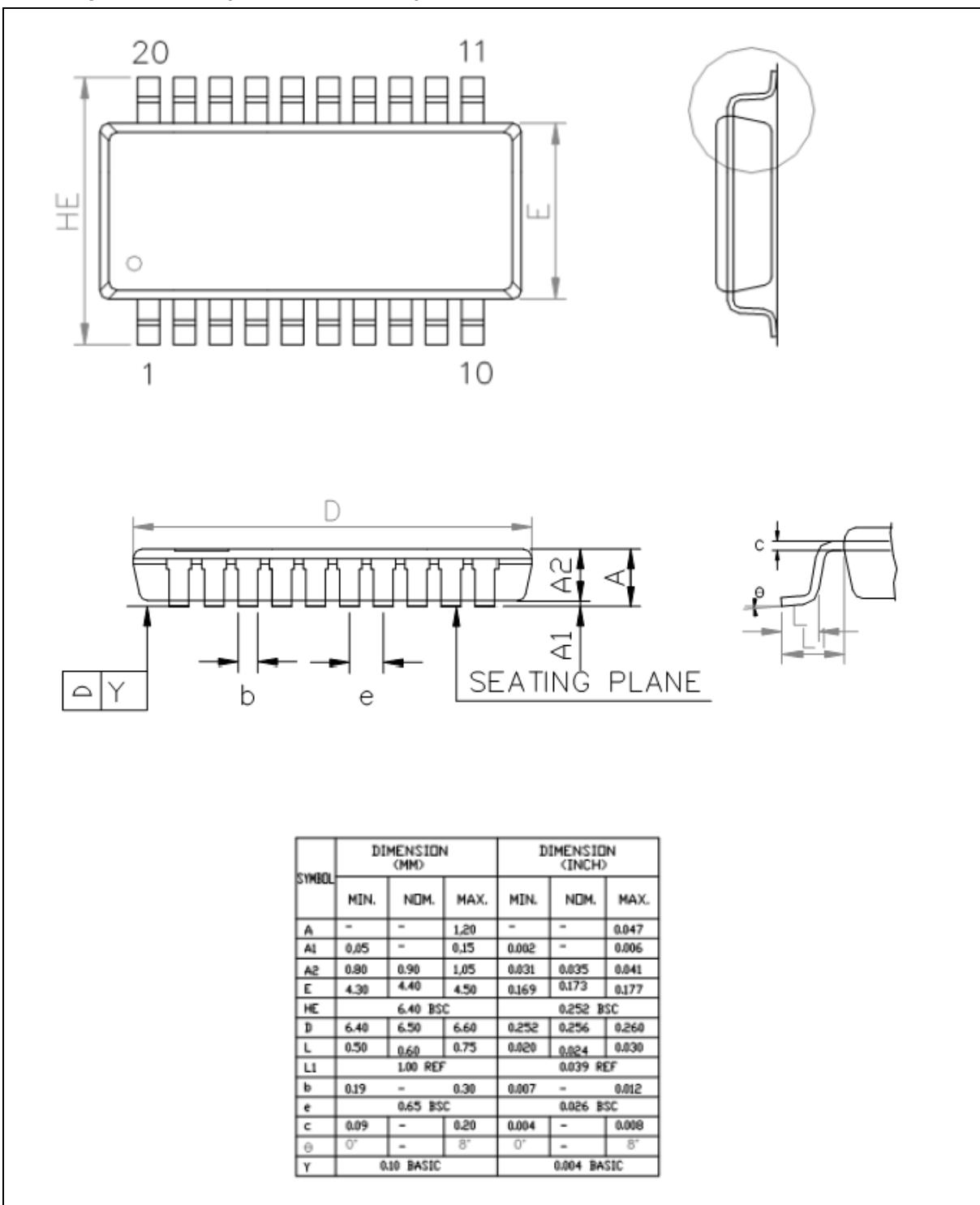


7.4 33-pin QFN (4x4x0.8 mm)

								
SYMBOLS	MIN.	NOM.	MAX.					
A	0.70	0.75	0.80					
A1	0.00	0.02	0.05					
A3	0.20 REF.							
b	0.15	0.20	0.25					
D	4.00 BSC							
E	4.00 BSC							
e	0.40 BSC							
L	0.35	0.40	0.45					
L1	0.35	0.40	0.45					
K	0.20	—	—					
Symbol		L/F SIZE		Dimension in mm				
D2		114X114 mm		MIN.	NOR.	MAX.		
E2				2.6	2.7	2.8		
				2.6	2.7	2.8		

注: QFN33 4*4 lead length 0.4mm is only for NUC029xAN, not for NUC029xAE

7.5 20-pin TSSOP (6.5x4.4x1.2 mm)



8 修正历史

日期	版本	描述
2014.05.27	1.00	1. 初版.
2014.08.26	1.01	1. 调整 NuMicro® NUC029 4.1 章節 NuMicro® NUC029xAN_xAE 系列选型指南. 2. 调整 5.2.6.章节中nRD, nWR, nWRH, nWRL, nINT0 和 nINT1的字体
2015.05.18	1.02	1. 修改第 5 章功能描述的章节的顺序. 2. 修改注释部分的字体. 3. 修改 5.2.7 章节 P3.5 功能选择位 的字体 4. 修改 NUC029xAN 系列 5.3.4. 章节分频器输出 字体 5. 增加 5.2 和 5.3 章节中 时钟切换注意事项 6. 在 5.9.5.5 章节中增加 表 5.16 输入捕捉模式操作流程 7. 在 NUC029xAN 系列中移去模拟比较器输出反向功能相关描述. 8. 更新 7.2 章节中 33-pin QFN (4x4)的封装尺寸.
2017.06.23	1.03	1. 增加新型号 NUC029ZAN.
2017.12.11	1.04	1. 增加新型号 NUC029NAN.
2019.01.09	1.05	1. 更新 7.2 章节中 48-pin QFN (7x7x0.8 mm)的封装尺寸
2019.05.06	1.06	1. 修改 NUC029 技术参考手册的文件名称, 由 TRM_NUC029 修改为 TRM_NUC029xAN_xAE 2. 修改 7.3 与 7.4 小节的标题名称, 厚度应为 0.8mm 3. 修改章节 1 概述中的错字, NUC029xAN_xAE 系列的数量应为 5 种
2019.08.28	1.07	1. 修正图 4-3 中, Pin 30 ~ 34 的错误叙述 2. 修正图 5-4 中的错误叙述, WDT 与 WWDT 位置应交换
2020.01.31	1.08	1. 修正错误, 5.16 模数转换器(ADC)章节中, NUC029FAE 只支持单次模式
2020.04.09	1.09	1. 於 4.3.1 小節與第章增加 ICE_DAT, ICE_CLK 與 nRst 管腳的外部建議電路說明

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.