

NuMicro™ Nano100系列

技术参考手册

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

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

All data and specifications are subject to change without notice.

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

www.nuvoton.com

目录

1	概述.....	14
2	特性.....	16
2.1	Nano100 特性 – Base Line	16
2.2	Nano110 特性 – LCD Line	22
2.3	Nano120 特性 – USB Connectivity Line.....	28
2.4	Nano130 特性 – Advanced Line.....	34
3	编号信息列表及管脚名称定义.....	40
3.1	NuMicro™ Nano100系列选型编码.....	40
3.2	NuMicro™ Nano100产品选型指南.....	41
3.2.1	NuMicro™ Nano100 Base Line选型指南.....	41
3.2.2	NuMicro™ Nano110 LCD Line选型指南.....	41
3.2.3	NuMicro™ Nano120 USB Connectivity Line选型指南.....	41
3.2.4	NuMicro™ Nano130 Advanced Line选型指南.....	42
3.3	管脚配置	43
3.3.1	NuMicro™ Nano100管脚图	43
3.3.2	NuMicro™ Nano110管脚图	46
3.3.3	NuMicro™ Nano120管脚图	48
3.3.4	NuMicro™ Nano130管脚图	51
3.4	管脚功能描述	53
3.4.1	NuMicro™ Nano100管脚描述.....	53
3.4.1	NuMicro™ Nano110管脚描述.....	65
3.4.2	NuMicro™ Nano120管脚描述.....	80
3.4.3	NuMicro™ Nano130管脚描述.....	92
4	框图.....	107
4.1	Nano100 框图.....	107
4.2	Nano110 框图.....	108
4.3	Nano120 框图.....	109
4.4	Nano130 框图.....	110
5	功能描述.....	111
5.1	内存结构	111
5.1.1	概述	111
5.1.2	存储器映射	111
5.2	嵌套向量中断控制器 (NVIC).....	113
5.2.1	概述	113
5.2.2	特征	113
5.2.3	异常模式和系统中断映射	113
5.2.4	向量表	115

5.2.5	操作说明	115
5.2.6	NVIC控制寄存器映射	116
5.2.7	NVIC 控制寄存器描述	117
5.2.8	中断源控制寄存器	129
5.3	系统管理器	137
5.3.1	概述	137
5.3.2	特征	137
5.3.3	功能描述	137
5.3.4	寄存器映射	140
5.3.5	寄存器描述	142
5.4	时钟控制器	219
5.4.1	概述	219
5.4.2	特征	219
5.4.3	框图	220
5.4.4	功能描述	220
5.4.5	寄存器和内存的映射	223
5.4.6	寄存器描述	224
5.5	模数转换器 (ADC)	248
5.5.1	概述	248
5.5.2	特征	248
5.5.3	框图	249
5.5.4	功能描述	250
5.5.5	寄存器映射	259
5.5.6	寄存器 描述	261
5.6	数模转换器	281
5.6.1	概述	281
5.6.2	特征	281
5.6.3	模块图	281
5.6.4	功能描述	282
5.6.5	寄存器映射	287
5.6.6	寄存器描述	288
5.7	DMA控制器	293
5.7.1	概述	293
5.7.2	特征	293
5.7.3	框图	295
5.7.4	功能描述	296
5.7.5	寄存器映射	299
5.7.6	寄存器 描述	301
5.8	外部总线接口	346
5.8.1	概述	346
5.8.2	特征	346
5.8.3	框图	346

5.8.4	功能描述	347
5.8.5	寄存器映射	351
5.8.6	寄存器 描述	351
5.9	FLASH内存控制器 (FMC)	355
5.9.1	概述	355
5.9.2	特征	355
5.9.3	框图	355
5.9.4	功能描述	356
5.9.5	寄存器映射	368
5.9.6	寄存器 描述	368
5.10	通用 I/O 控制器	376
5.10.1	概述	376
5.10.2	特征	376
5.10.3	框图	377
5.10.4	功能描述	377
5.10.5	寄存器映射	378
5.10.6	寄存器 描述	384
5.11	I ² C	399
5.11.1	概述	399
5.11.2	特征	400
5.11.3	功能描述	400
5.11.4	寄存器映射	410
5.11.5	寄存器 描述	411
5.12	I ² S	423
5.12.1	概述	423
5.12.2	特性	423
5.12.3	框图	423
5.12.4	功能描述	424
5.12.5	寄存器和存储器映射	426
5.12.6	寄存器描述	427
5.13	LCD 显示驱动器	439
5.13.1	概述	439
5.13.2	特性	439
5.13.3	框图	440
5.13.4	功能描述	441
5.13.5	寄存器和存储器映射	444
5.13.6	寄存器描述	445
5.13.7	应用电路	452
5.14	脉宽调制(PWM)	456
5.14.1	概述	456
5.14.2	特性	456
5.14.3	框图	458

5.14.4	功能描述	460
5.14.5	寄存器和存储器映射	468
5.14.6	寄存器描述	470
5.15	RTC	494
5.15.1	概述	494
5.15.2	特性	494
5.15.3	框图	495
5.15.4	功能描述	496
5.15.5	寄存器和存储器映射	499
5.15.6	寄存器描述	501
5.16	智能卡主机接口 (SC)	518
5.16.1	概述	518
5.16.2	特性	518
5.16.3	框图	519
5.16.4	功能描述	521
5.16.5	寄存器和存储器映射	527
5.16.6	寄存器描述	528
5.17	SPI	562
5.17.1	概述	562
5.17.2	特性	562
5.17.3	框图	563
5.17.4	功能描述	563
5.17.5	寄存器和存储器映射	577
5.17.6	寄存器描述	578
5.18	定时器控制器	594
5.18.1	概述	594
5.18.2	特性	594
5.18.3	框图	594
5.18.4	功能描述	595
5.18.5	寄存器和存储器映射	599
5.18.6	寄存器描述	600
5.19	UART 控制器	613
5.19.1	概述	613
5.19.2	特征	615
5.19.3	框图	616
5.19.4	功能描述	619
5.19.5	寄存器和存储器映射	627
5.19.6	寄存器描述	628
5.20	USB	651
5.20.1	概述	651
5.20.2	特性	651
5.20.3	框图	652

5.20.4	功能描述	652
5.20.5	寄存器和存储器映射	656
5.20.6	寄存器描述	658
5.21	看门狗定时器控制器	674
5.21.1	概述	674
5.21.2	特征	675
5.21.3	框图	675
5.21.4	功能描述	675
5.21.5	寄存器和存储器映射	677
5.21.6	寄存器描述	678
5.22	窗口看门狗定时控制器	682
5.22.1	概述	682
5.22.2	特征	682
5.22.3	框图	682
5.22.4	功能描述	682
5.22.5	寄存器存储器映射	684
5.22.6	寄存器描述	684
6	ARM® CORTEX™-M0 内核	689
6.1	概述	689
6.2	特性	689
6.3	系统定时器(SysTick)	690
6.3.1	系统定时器控制寄存器映射	690
6.3.2	系统定时器控制寄存器描述	692
6.4	系统控制寄存器	695
6.4.1	系统控制寄存器映射	695
6.4.2	系统控制寄存器描述	696
7	应用电路	701
7.1	LCD Charge Pump	701
7.1.1	C-type 1/3 Bias	701
7.1.2	C-type 1/2 Bias	701
7.1.3	内部R-type	701
7.1.4	外部R-type	702
7.2	ADC 应用电路	703
7.2.1	电压参考源	703
7.3	DAC 应用电路	705
7.3.1	电压参考源	705
7.4	整个芯片应用电路	707
8	功耗	708
9	电气特性	709
9.1	绝对最大额定值	709

9.2	Nano100/Nano110/Nano120/Nano130 DC电气特性	710
9.3	AC电气特性	716
9.3.1	外部输入时钟	716
9.3.2	外部 4~24MHz XTAL 振荡器	716
9.3.3	外部 32.768 kHz 晶振	717
9.3.4	内部 12 MHz 振荡器	717
9.3.5	内部 10 kHz 振荡器	717
9.4	模拟量特性	718
9.4.1	12-bit ADC	718
9.4.2	欠压检测	718
9.4.3	上电复位	719
9.4.4	温度传感器	719
9.4.5	12-bit DAC	719
9.4.6	LCD	720
9.4.7	内部参考电压	721
9.4.8	USB PHY 规格	721
10	封装尺寸	723
10.1	LQFP128 (14x14x1.4 mm footprint 2.0 mm)	723
10.2	LQFP64 (10x10x1.4 mm footprint 2.0 mm)	724
10.3	LQFP64 (7x7x1.4 mm footprint 2.0 mm)	725
10.4	LQFP48 (7x7x1.4 mm footprint 2.0 mm)	727
10.5	QFN48 (7x7x0.85 mm)	728
11	版本历史	729

图列表

图 3-1 NuMicro™ Nano100系列选型编码	40
图 3-2 NuMicro™ Nano100 LQFP 128-管脚图.....	43
图 3-3 NuMicro™ Nano100 LQFP 64-pin管脚图.....	44
图 3-4 NuMicro™ Nano100 LQFP 48-pin管脚图	45
图 3-5 NuMicro™ Nano110 LQFP 128-pin管脚图	46
图 3-6 NuMicro™ Nano110 LQFP 64-pin管脚图	47
图 3-7 NuMicro™ Nano120 LQFP 128-pin管脚图	48
图 3-8 NuMicro™ Nano120 LQFP 64-pin管脚图	49
图 3-9 NuMicro™ Nano120 LQFP 48-pin管脚图	50
图 3-10 NuMicro™ Nano130 LQFP 128-pin管脚图	51
图 3-11 NuMicro™ Nano130 LQFP 64-pin管脚图	52
图 4-1 NuMicro™ Nano100框图.....	107
图 4-2 NuMicro™ Nano110框图.....	108
图4-3 NuMicro™ Nano120 框图.....	109
图 4-4 NuMicro™ Nano130 框图.....	110
图 5-1 电源模式.....	138
图 5-2 时钟控制器框图	220
图 5-3 分频器的时钟源	222
图 5-4 分频器框图	222
图 5-5 ADC 和DAC 框图.....	249
图 5-6 ADC转换器框图	250
图 5-7 ADC 时钟控制.....	250
图 5-8 ADC单一模式转换时序图.....	251
图 5-9 ADC在使能通道上单周期扫描模式的时序图.....	252
图 5-10 ADC使能通道上的连续扫描模式时序图.....	253
图 5-11 ADC转换结果监控逻辑图	254
图 5-12 ADC控制器中断	254
图 5-13 ADC开始转换的条件.....	255
图 5-14 采样网络模型.....	255
图 5-15 递增采样时间波形.....	256
图 5-16 额外采样和保持时钟周期作为信号源输出阻抗 $R_i(k\Omega)$ 的功能。在该图中给出的结果是在额定条件下测定的 (typical process corner, $V_{DD} = AV_{DD} = 3.3\text{ V}$, LDO output = 1.8 V, $T_{\text{junction}} = 50^\circ\text{C}$, ADC_CLK=42 MHz, $V_{\text{REF}} = AV_{DD}$)	256

图 5-17 额外采样和保持时钟周期 作为信号源输出阻抗 $R_i(k\Omega)$ 的功能。在该图中给出的结果是在最坏情况的条件下测定的 (slow process corner, $V_{DD} = AV_{DD} = 1.8\text{ V}$, LDO output = 1.62 V, $T_{junction} = -40^\circ\text{C}$, ADC_CLK=42 MHz, VREF= AV_{DD})	257
图 5-18 DAC0 和DAC1模块图	282
图 5-19 DAC0 和DAC1未分组更新范例	284
图 5-20 DAC0 和DAC1 分组更新范例	285
图 5-21 DAC 中断	286
图 5-22 DMA 控制器框图	295
图 5-23 DMA时钟控制图	295
图 5-24 CRC 发生器模块图	296
图 5-25 VDMA块传输	297
图 5-26 EBI框图	347
图 5-27 16-位 EBI 数据宽度与 16-位设备的连接	347
图 5-28 8-位 EBI 数据宽度和 8-位设备的连接	348
图 5-29 16-位数据宽度的时序控制波形	349
图 5-30 8-位数据宽度时序控制波形	350
图 5-31 插入空闲周期的时序控制波形	351
图 5-32 Flash 内存控制器框图	355
图 5-33 内存组织结构	357
图 5-34 CONFIG0中CBS 的Flash内存映射	359
图 5-35 32/64/128KB内存结构	360
图 5-36 当ISP 操作时, CPU 暂停工作	364
图 5-37 ISP操作流程	365
图 5-38 GPIO 框图	377
图 5-39 推挽输出	377
图 5-40 开漏输出	378
图 5-41 I ² C 协议	400
图 5-42 I ² C主机向从机传输数据	401
图 5-43 I ² C 主机从从机读取数据	401
图 5-44 I ² C起始 (START) 和停止 (STOP) 条件	401
图 5-45 I ² C 总线上的位传输	402
图 5-46 I ² C总线上的应答信号	402
图 5-47 I ² C超时计数器框图	403
图 5-48 I ² C正常主机传输状态	404

图 5-49 I ² C主机传送模式	405
图 5-50 I ² C主机接收模式	406
图 5-51 I ² C从机接收模式	407
图 5-52 I ² C从机发送模式	408
图 5-53 I ² C广播呼叫模式	409
图 5-54 I ² C 总线错误时序	410
图 5-55 I ² S 控制器框图	423
图 5-56 I ² S 总线时序图 (Format = 0)	424
图 5-57 MSB 对齐时序图 (Format = 1)	424
图 5-58 不同I ² S 模式下FIFO 的内容	425
图 5-59 LCD 驱动器框图	440
图 5-60 LCD 存储映射	441
图 5-61 COM 信号波形	442
图 5-62 SEG 信号波形	442
图 5-63 COM-SEG 信号波形, 1/6 占空比, 1/3 偏压	443
图 5-64 1/3 偏压 (外部 梯形电阻)	452
图 5-65 1/2 偏压 (外部 梯形电阻)	453
图 5-66 1/3 偏压 (带电容的梯形电阻)	454
图 5-67 1/2 偏压 (带电容的梯形电阻)	455
图 5-68 1/3 偏压 (电荷泵)	455
图 5-69 1/2 偏压 (电荷泵)	455
图 5-70 PWM0 时钟	458
图 5-71 用于通道 0, 1 的PWM0 发生器	458
图 5-72 PWM1 时钟	458
图 5-73用于通道 2, 3 的PWM1 发生器	459
图 5-74 PWM-定时器内部比较器输出	460
图 5-75通道 0 PWM-定时器操作时序	461
图 5-76 PWM双缓存说明	461
图 5-77 PWM控制器输出占空比	462
图 5-78带死区发生器PWM-对输出操作	462
图 5-79 PWM捕捉操作时序	463
图 5-80 PWM-定时器中断	464
图 5-81 PWM-定时器停止方式1	466

图 5-82 RTC 框图.....	495
图 5-83 SC 时钟控制框图（时钟控制器中的4-bit 预分频计数器）	519
图 5-84 SC 控制器框图.....	520
图 5-85 SC 数据字符	521
图 5-86 SC 激活序列	522
图 5-87 SC 暖复位序列.....	523
图 5-88 SC 释放序列	524
图 5-89 初始化字符 TS	524
图 5-90 SC 错误信号	525
图 5-91 SPI 框图	563
图 5-92 SPI主机模式应用框图	564
图 5-93 SPI从机模式应用框图	564
图 5-94 SPI 可变时钟频率	566
图 5-95 SPI 一个事务中数据位长为 32 位.....	566
图 5-96 SPI 字节重排序.....	567
图 5-97 SPI 字节 休眠 模式	568
图 5-98 SPI两位传输模式	569
图 5-99 SPI两位传输模式时序图.....	569
图 5-100 SPI DUAL-I/O 输出序列	570
图 5-101 SPI FIFO 模式控制时序	572
图 5-102 主机模式下 SPI 时序.....	573
图 5-103 主机模式下 SPI 时序(Alternate Phase of SPICLK & LSB = 1).....	573
图 5-104 主机模式下 SPI 时序(Alternate Phase of SPICLK & LSB = 0).....	574
图 5-105 从机模式下SPI 时序.....	574
图 5-106 从机模式下SPI 时序(Alternate Phase of SPICLK).....	575
图 5-107 定时器控制器框图	595
图 5-108 定时器时钟控制器图	595
图 5-109 定时器时钟控制器	597
图 5-110 定时器间互触发模式	599
图 5-112 UART时钟控制图	616
图 5-113 UART框图	617
图 5-114 UART自动流控框图	619
图 5-115 UART自动波特率框图.....	619
图 5-116 UART CTSn唤醒情况1	620

图 5-117 UART CTSn唤醒情况2	620
图 5-118 UART DATA唤醒	620
图 5-119 IrDA框图	621
图 5-120 IrDA TX/RX 时序图	622
图5-121 RS-485帧结构.....	624
图 5-122 LIN帧结构	624
图 5-123 USB 框图.....	652
图 5-124 SRAM 中各端点的分配	653
图 5-125 USB 唤醒中断操作流程.....	654
图 5-126 USB 数据 IN 事务紧跟 Setup 事务.....	655
图 5-127 USB 数据输出事务.....	655
图 5-128看门狗定时器框图	675
图 5-129看门狗定时器时钟控制图	675
图 5-130看门狗中断和复位信号时序	676
图 5-131窗口看门狗定时器框图.....	682
图 5-132 看门狗控制器框图	682
图 6-1 M0功能模块	689
图 9-1典型晶振应用电路.....	716

表格列表

表 1-1 所支持的接口列表	15
表 3-1 Nano100 Base Line 选型表	41
表 3-2 Nano110 LCD Line 选型表	41
表 3-3 Nano120 USB Connectivity Line 选型表	41
表 3-4 Nano130 Advanced Line 选型表	42
表 5-1 异常模式	114
表 5-2 系统中断映射	115
表 5-3 向量表格式	115
表 5-4 再次进入掉电模式的条件	138
表 5-5 在电源模式的IP 时钟ON/OFF	139
表 5-6 外设时钟	221
表 5-7 电源模式和时钟	226
表 5-8 内存地址映射	356
表 5-9 Flash 大小	356
表 5-10 启动选择	358
表 5-11 启动选择和支持功能	358
表 5-12 ISP操作命令	367
表 5-13 UART波特率等式	614
表 5-14 UART波特率设置	614
表 5-15 UART中断源和标志	638
表 5-16 看门狗超时间隔选择	676
表 5-17 窗口看门狗预分频值选择	683

1 概述

Nano100 系列是超低功耗的 32-位内嵌 ARM® Cortex™-M0 核的微控制器。它能工作在1.8V 到 3.6V的宽电压范围，最高可运行到42MHz，内建32K/64K/128K-字节的flash 以及8K/16K-字节的SRAM。 Nano100 系列集成了 LCD 4x40或6x38 (COM/Segment), USB 2.0 全速功能, RTC, 12-位 SAR ADC, 12-位DAC, 提供高性能的外围接口连接, 如UART, SPI, I²C, I²S, GPIOs, 用于外部内存映射设备访问的EBI (External Bus Interface) 以及用于智能卡的 ISO-7816-3。支持欠压检测, 保留 RAM 的掉电模式以及通过多个外围接口快速唤醒的功能。

Nano100 系列提供低电压, 低功耗, 低待机电流, 高集成度外设, 高效率操作, 快速唤醒及最低成本的32-位的微控制器。Nano100 系列适用于广泛的电池设备应用, 诸如:

- 便携式数据采集器 (Portable Data Collector)
- 便携式医疗监护仪 (Portable Medical Monitor)
- 便携式 RFID 读卡器 (Portable RFID Reader)
- 便携式条形码扫描仪 (Portable Barcode Scanner)
- 安全报警系统 (Security Alarm System)
- 系统管理程序 (System Supervisors)
- 电表 (Power Metering)
- USB 配件 (USB Accessories)
- 智能卡读卡器 (Smart Card Reader)
- 无线游戏控制器 (Wireless Game Control Device)
- IPTV 智能遥控器 (IPTV Remote Smart Keyboard)
- 无线传感器节点设备 (Wireless Sensors Node Device (WSN))
- 无线RF4CE 遥控器 (Wireless RF4CE Remote Control)
- 无线音频 (Wireless Audio)
- 无线自动仪表读卡器 (Wireless Automatic Meter Reader(AMR))
- 电子不停车收费系统 (Electronic Toll Collection(ETC))

Nano100 Base line 是超低功耗的32-位内嵌 ARM® Cortex™-M0 核的微控制器。它能工作在1.8V 到 3.6V的宽电压范围，最高可运行到42MHz，内建32K/64K/128K-字节的flash 以及8K/16K-字节的SRAM。它集成了RTC, 12-通道12-位 SAR ADC, 2-通道12-位 DAC, 提供高性能的外围接口连接, 诸如2xUART, 3xSPI, 2xI²C, I²S, GPIOs, 用于外部内存映射设备访问的EBI (External Bus Interface), 3xISO-7816-3 用于智能卡。支持欠压检测, 保留 RAM 的掉电模式以及通过多个外围接口快速唤醒的功能。

Nano110 LCD line是超低功耗的32-位内嵌 ARM® Cortex™-M0 核的微控制器。它能工作在1.8V 到 3.6V的宽电压范围，最高可运行到42MHz，内建32K/64K/128K -字节的flash 以及8K/16K-字节的SRAM。它集成了LCD 4x40或6x38(COM/Segment), RTC, 12-通道12-位 SAR ADC, 2-通道12-位 DAC, 提供高性能的外围接口连接, 诸如2xUART, 2xSPI, 2xI²C, I²S, GPIOs, 用于外部内存映射设备访问的EBI (External Bus Interface), 3xISO-7816-3 用于智能卡。支持欠压检测, 保留 RAM 的掉电模式以及通过多个外围接口快速唤醒的功能。

Nano120 USB connectivity line 是超低功耗的32-位内嵌 ARM® Cortex™-M0 核的微控制器。它能工作在1.8V 到 3.6V的宽电压范围，最高可运行到42MHz，内建32K/64K/128K -字节的flash 以及8K/16K-字节的SRAM。它集成了 USB 2.0 全速设备功能，RTC，12-通道12-位 SAR ADC，2-通道 12-位 DAC，提供高性能的外围接口连接，诸如2xUART，3xSPI，2xI²C，I²S，GPIOs，用于外部内存映射设备访问的EBI (External Bus Interface)，3xISO-7816-3 用于智能卡。支持欠压检测，保留 RAM 的掉电模式以及通过多个外围接口快速唤醒的功能。

Nano130 Advance line是超低功耗的32-位内嵌 ARM® Cortex™-M0 核的微控制器。它能工作在1.8V 到 3.6V的宽电压范围，最高可运行到42MHz，内建32K/64K/128K-字节的flash 以及8K/16K-字节的SRAM。它集成了LCD 4x40或6x38 (COM/Segment)，USB 2.0 全速设备功能，RTC，12-通道12-位 SAR ADC，2-通道12-位 DAC，提供高性能的外围接口连接，诸如2xUART，2xSPI，2xI²C，I²S，GPIOs，用于外部内存映射设备访问的EBI (External Bus Interface)，3xISO-7816-3 用于智能卡。支持欠压检测，保留 RAM 的掉电模式以及通过多个外围接口快速唤醒的功能。

产品线	UART	SPI	I ² C	I ² S	USB	LCD	ADC	DAC	RTC	EBI	SC	Timer
Nano100	•	•	•	•			•	•	•	•	•	•
Nano110	•	•	•	•		•	•	•	•	•	•	•
Nano120	•	•	•	•	•		•	•	•	•	•	•
Nano130	•	•	•	•	•	•	•	•	•	•	•	•

表 1-1 所支持的接口列表

2 特性

该器件的功能依赖于产品线和他们的子系统产品。

2.1 Nano100 特性 – Base Line

- 内核
 - ◆ ARM® Cortex™-M0 内核最高运行到 42 MHz
 - ◆ 一个 24-位系统定时器
 - ◆ 支持低功耗睡眠模式
 - ◆ 单周期 32-位硬件乘法器
 - ◆ 嵌套向量中断控制器 (NVIC) 用于控制32个中断源，每个中断源可设置为4个优先级
 - ◆ 支持串行线调试 (SWD) 带2个观察点/4个断点
- 欠压检测
 - ◆ 内建 2.5V/2.0V/1.7V BOD 用于宽泛的工作电压范围操作
- Flash EPROM 存储器
 - ◆ 高达 42 MHz 时的不连续地址读访问零等待状态
 - ◆ 64K/32K/123K-字节应用程序存储器 (APROM)
 - ◆ 4KB 在系统编程 (ISP) 加载程序存储器 (LDROM)
 - ◆ 可编程数据 flash 起始地址和存储器大小以512字节为页擦除单元
 - ◆ 在系统编程 (ISP)/在应用编程 (IAP) 更新芯片的Flash EPROM
- SRAM 存储器
 - ◆ 16K/8K-字节内建 SRAM
 - ◆ 支持 DMA 模式
- DMA: 支持 8 通道: 1 VDMA 通道和 6 PDMA 通道和1 CRC 通道
 - ◆ VDMA
 - 内存-到-内存传输
 - 支持跨距的块传输
 - 支持 字/半字/字节边界地址
 - 支持地址方向: 增长和减少
 - ◆ PDMA
 - 外设-到-内存, 内存-到-外设, 内存-到-内存 传输
 - 支持字边界地址
 - 在内存-到-内存模式, 支持字对齐传输长度
 - 在外设-到-内存, 内存-到-外设传输模式, 支持 字/半字/字节 对齐传输长度
 - 外设作为源或目标时, 支持字/半字/字节 传输数据宽度

- 支持地址方向：增长，固定和回绕
- ◆ CRC
 - 支持4个通用的多项式 CRC-CCITT, CRC-8, CRC-16, 和 CRC-32
 - ◆ CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - ◆ CRC-8: $X^8 + X^2 + X + 1$
 - ◆ CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - ◆ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 时钟控制
 - ◆ 针对不同应用可灵活选择时钟
 - ◆ 当打开自动修正功能，在全温度范围，内置 12MHz OSC可以校正到0.25%的偏差 (系统必须有外部 32.768 kHz 晶振输入)，否则在全温度范围12MHz OSC 会有 2 % 偏差。
 - ◆ 低功率10 kHz OSC 用于 看门狗 和低功耗操作模式
 - ◆ 支持一组PLL, 高达120 MHz, 用于高性能的系统运行和 USB 应用(48 MHz).
 - ◆ 外部 4~24 MHz 晶振输入用于精准的定时操作
 - ◆ 外部 32.768 kHz晶振输入用于 RTC 及低功耗模式操作
- GPIO
 - ◆ 三种 I/O 模式：
 - 推挽输出
 - 开漏输出
 - 高阻输入
 - ◆ 所有输入为 Schmitt 触发
 - ◆ I/O 引脚可被配置为边沿/电平触发模式的中断源
 - ◆ 支持高驱动和高灌入 IO 模式
 - ◆ 支持 5V 输入，除了PA.0 ~ PA.7, PD.0 ~ PD.1 和 PC.6 ~ PC.7
- Timer
 - ◆ 支持4组32位定时器，每个定时器有一个24位向上计数定时器和一个8位预分频计数器
 - ◆ 每个定时器有独立的时钟源
 - ◆ 提供 单次、周期、输出翻转和连续计数操作模式
 - ◆ 内部触发事件到 ADC, DAC, 与 PDMA 模块
 - ◆ 支持 PDMA 模式
 - ◆ Timer 可以从掉电或空闲模式唤醒系统
- 看门狗定时器

- ◆ 时钟源来自 LIRC. (内部 10KHz 低速振荡器时钟)
- ◆ 从1.6ms ~ 26sec 有可选的定时溢出周期 (取决于所选的时钟源)
- ◆ 看门狗定时溢出的中断/复位选择
- ◆ WDT 可以从掉电模式唤醒系统
- 窗口看门狗定时器(WWDT)
 - ◆ 6-位 下数计数器 6-位 比较值使窗口周期灵活可变
 - ◆ 可选的 WWDT 时钟预分频计数器使 WWDT 溢出间隔可变.
- RTC
 - ◆ 通过频率补偿寄存器 (FCR) 支持软件频率补偿功能
 - ◆ 支持 RTC 计数(秒, 分, 小时) 及万年历功能 (日, 月, 年)
 - ◆ 支持闹铃寄存器 (秒, 分, 小时, 日, 月, 年)
 - ◆ 可选择为12小时制或24小时
 - ◆ 闰年自动识别
 - ◆ 支持周期时间滴答中断, 包括8个可选周期 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 和 1 秒
 - ◆ RTC 可以从掉电模式唤醒系统
 - ◆ 支持 80 字节备用寄存器以及一个探测引脚来清除这些备用寄存器的内容
- PWM/Capture
 - ◆ 内建 2 个 PWM 模块, 每个模块有2个 16-位 PWM 产生器
 - ◆ 提供 8个 PWM 输出或四组互补配对 PWM 输出
 - ◆ 每个 PWM 产生器配有一个时钟分频器, 一个8-位时钟预分频, 2个时钟选择器以及一个用于互补配对 PWM 的死区发生器
 - ◆ 最多8路16-位数字捕捉定时器 (共享 PWM 定时器), 提供8路捕捉输入 (上升, 下降或两者皆可)
 - ◆ 支持 one shot 和连续模式
 - ◆ 支持捕捉 (Capture) 中断
- UART
 - ◆ 最多 2 组16-字节 FIFO UART 控制器
 - ◆ UART 端口支持流控 (TX, RX, CTSn 和 RTSn)
 - ◆ 支持 IrDA (SIR) 功能
 - ◆ 支持 LIN 功能
 - ◆ 支持 RS-485 9 位模式和方向控制
 - ◆ 可编程波特率发生器
 - ◆ 支持 PDMA 模式
 - ◆ UART 可以从掉电模式唤醒系统

- SPI
 - ◆ 最高支持 3 组 SPI 控制器
 - ◆ 主机速率最高到 32MHz，从机最高至16MHz
 - ◆ 支持 SPI/MICROWIRE 主机/从机模式
 - ◆ 全双工同步串行数据传输
 - ◆ 可变传输数据长度（4 到 32 位）
 - ◆ 可配置 MSB 或 LSB 在前的数据传输
 - ◆ 在时钟上升沿或下降沿接收 (RX) 还是发送 (TX) 是独立配置的
 - ◆ 当 SPI 控制器作为主机时，2 条从机/设备选择线；作为从机时，1条从机/设备选择线
 - ◆ 支持 32-bit 传输模式下的字节睡眠模式
 - ◆ 支持 2 通道 PDMA 请求，1个用于发送，另一个用于接收
 - ◆ 支持三线模式，无从机选择信号，双向接口
 - ◆ SPI 可以从掉电模式唤醒系统
- I²C
 - ◆ 最多支持 2 组 I²C 设备
 - ◆ 主机/从机最高达 1Mbit/s
 - ◆ 主从机之间双向数据传输
 - ◆ 多主机总线支持（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
 - ◆ 串行时钟同步可作为握手方式控制总线上数据暂停及恢复传送
 - ◆ 内建的一个 14-位超时计数器在 I²C 总线中止以及超时计数器溢出时会请求 I²C 中断
 - ◆ 可编程的时钟适用于不同速率控制
 - ◆ 支持 7-位 地址模式
 - ◆ 支持多地址识别（4个从机地址带mask选项）
- I²S
 - ◆ 外部音频 CODEC 接口
 - ◆ 可做主机也可作从机模式
 - ◆ 能处理8, 16, 24 和 32 位字
 - ◆ 支持单声道和立体声的音频数据
 - ◆ 支持 I²S 和 最高有效位数据格式
 - ◆ 提供两组 8 字的FIFO数据缓存，一组用于发送，一组用于接收
 - ◆ 缓冲区超过可编程边界时，产生中断请求

- ◆ 支持两组 PDMA 请求，一组用于发送，另一组用于接收
- ADC
 - ◆ 12-位 SAR ADC 最高 2Msps 转换速率
 - ◆ 最多 12-通道单端模式外部输入(PA.0 ~ PA.7 和 PD.0 ~ PD.3)
 - ◆ 6个内部通道来自 DAC0, DAC1, 内部参考电压 (Int_VREF), 温度传感器, AVDD, 和 AVSS.
 - ◆ 支持三个参考电压源: VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 支持单扫描, 单周期扫描和连续扫描模式
 - ◆ 每个通道有独立的结果寄存器
 - ◆ 只能扫描使能的通道
 - ◆ 阈值电压侦测 (比较功能)
 - ◆ 软件编程或外部管脚触发开始转换
 - ◆ 支持 PDMA 模式
 - ◆ 支持最多四个定时器超时事件 (TMR0, TMR1, TMR2 和 TMR3) 使能 ADC
- DAC
 - ◆ 12-位单输出 400K 转换速率
 - ◆ 支持三个参考电压源 VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 两 DACs 的同步更新能力 (组功能)
 - ◆ 支持最多四个定时器超时事件(TMR0, TMR1, TMR2 和 TMR3) , 软件或 PDMA 触发 DAC 去进行转换
- SmartCard (SC)
 - ◆ 依据 ISO-7816-3 T=0, T=1
 - ◆ 支持最多 3 个 ISO-7816-3 端口
 - ◆ 单独的接收/发送 4 字节项目 FIFO用于数据负载
 - ◆ 可编程的传送时钟频率
 - ◆ 可编程接收器缓存触发水平
 - ◆ 可编程保卫时间选择 (11 ETU ~ 266 ETU)
 - ◆ 一个 24-位和两个 8 位超时计数器用于回复请求 (Answer to Reset (ATR))并等待时间运行
 - ◆ 支持自动逆转换功能
 - ◆ 支持停止时钟电平和时钟停止 (时钟保持) 功能
 - ◆ 支持发送器和接收器错误重试和错误限制功能
 - ◆ 支持硬件激活序列处理
 - ◆ 支持硬件热重置序列处理
 - ◆ 支持硬件释放序列处理

- ◆ 支持当检测到卡被移除时，硬件自动释放序列
- ◆ 支持 UART 模式 (半双工)
- EBI (External bus interface) (外部总线接口)
 - ◆ 可访问的空间：8位模式为64KB 或16位模式为 128KB
 - ◆ 支持 8bit/16bit 数据宽度
 - ◆ 在16-位 数据宽度模式下支持字节写入
- 一个内建的温度传感器，分辨率为 1℃
- 96-位独一无二的ID
- 128-位唯一客户ID
- 工作温度：-40℃~85℃
- 封装：
 - ◆ 绿色（无铅）封装 (RoHS)
 - ◆ LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7) / QFN 48-pin(7x7)

2.2 Nano110 特性 – LCD Line

- Core
 - ◆ ARM® Cortex™-M0 内核最高运行到42 MHz
 - ◆ 一个 24-位系统定时器
 - ◆ 支持低功耗睡眠模式
 - ◆ 单周期 32-位硬件乘法器
 - ◆ 嵌套向量中断控制器 (NVIC) 用于控制32个中断源，每个中断源可设置为4个优先级
 - ◆ 支持串行线调试 (SWD) 带2个观察点/4个断点
- 欠压检测
 - ◆ 内建2.5V/2.0V/1.7V BOD 用于宽泛的工作电压范围操作
- Flash EPROM 存储器
 - ◆ 高达 42MHz 时的不连续地址读访问零等待状态
 - ◆ 64K/32K/128K-字节应用程序存储器 (APROM)
 - ◆ 4KB 在系统编程 (ISP) 加载程序存储器 (LDROM)
 - ◆ 可编程数据 flash 起始地址和存储器大小以512字节为页擦除单元
 - ◆ 在系统编程 (ISP)/在应用编程 (IAP) 更新芯片的Flash EPROM
- SRAM 存储器
 - ◆ 16K/8K-字节内建 SRAM
 - ◆ 支持 DMA 模式
- DMA : 支持 8 通道: 1个 VDMA 通道, 6 PDMA 通道, 和 1个 CRC 通道
 - ◆ VDMA
 - 内存-到-内存传输
 - 支持跨距的块传输
 - 支持 字/半字/字节边界地址
 - 支持地址方向: 增长和减少
 - ◆ PDMA
 - 外设-到-内存, 内存-到-外设, 内存-到-内存 传输
 - 支持字边界地址
 - 在内存-到-内存模式, 支持字对齐传输长度
 - 在外设-到-内存, 内存-到-外设传输模式, 支持 字/半字/字节 对齐传输长度
 - 外设作为源或目标时, 支持字/半字/字节 传输数据宽度
 - 支持地址方向: 增长, 固定和回绕
 - ◆ CRC

- 支持4个通用多项式 CRC-CCITT, CRC-8, CRC-16, 和 CRC-32
 - ◆ CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - ◆ CRC-8: $X^8 + X^2 + X + 1$
 - ◆ CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - ◆ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 时钟控制
 - ◆ 针对不同应用可灵活选择时钟
 - ◆ 当打开自动修正功能, 在全温度范围, 内置 12MHz OSC可以校正到0.25%的偏差 (系统必须有外部 32.768 kHz 晶振输入), 否则在全温度范围12MHz OSC 会有 2 % 偏差。
 - ◆ 低功率10 kHz OSC 用于 看门狗 和低功耗操作模式
 - ◆ 支持一组PLL, 高达120 MHz, 用于高性能的系统运行和 USB 应用(48 MHz).
 - ◆ 外部 4~24 MHz 晶振输入用于精准的定时操作
 - ◆ 外部 32.768 kHz晶振输入用于 RTC 及低功耗模式操作
- GPIO
 - ◆ 三种 I/O 模式:
 - 推挽输出
 - 开漏输出
 - 高阻输入
 - ◆ 所有输入为 Schmitt 触发
 - ◆ I/O 引脚可被配置为边沿/电平触发模式的中断源
 - ◆ 支持高驱动和高灌入 IO 模式
 - ◆ 支持 5V 输入, 除了PA.0 ~ PA.7, PD.0 ~ PD.1 和 PC.6 ~ PC.7
- Timer
 - ◆ 支持4组32位定时器, 每个定时器有一个24位向上计数定时器和一个8位预分频计数器
 - ◆ 每个定时器有独立的时钟源
 - ◆ 提供 单次、周期、输出翻转和连续计数操作模式
 - ◆ 内部触发事件到 ADC, DAC, 与 PDMA 模块
 - ◆ 支持 PDMA 模式
 - ◆ Timer 可以从掉电或空闲模式唤醒系统
- 看门狗定时器
 - ◆ 时钟源来自 LIRC. (内部 10KHz 低速震荡器时钟)
 - ◆ 从1.6ms ~ 26sec 有可选的定时溢出周期 (取决于所选的时钟源)

- ◆ 看门狗定时溢出的中断/复位选择
- ◆ WDT 可以从掉电模式唤醒系统
- 窗口看门狗定时器(WWDT)
 - ◆ 6-位 下数计数器 6-位 比较值使窗口周期灵活可变
 - ◆ 可选的 WWDT 时钟预分频计数器使 WWDT 溢出间隔可变.
- RTC
 - ◆ 通过频率补偿寄存器 (FCR) 支持软件频率补偿功能
 - ◆ 支持 RTC 计数(秒, 分, 小时) 及万年历功能 (日, 月, 年)
 - ◆ 支持闹铃寄存器 (秒, 分, 小时, 日, 月, 年)
 - ◆ 可选择为12小时制或24小时
 - ◆ 闰年自动识别
 - ◆ 支持周期时间滴答中断, 包括8个可选周期 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 和 1 秒
 - ◆ RTC 可以从掉电模式唤醒系统
 - ◆ 支持 80 字节备用寄存器以及一个探测引脚来清除这些备用寄存器的内容
- PWM/Capture
 - ◆ 内建 2 个 PWM 模块, 每个模块有2个 16-位 PWM 产生器
 - ◆ 提供 8个 PWM 输出或四组互补配对 PWM 输出
 - ◆ 每个 PWM 产生器配有一个时钟分频器, 一个8-位时钟预分频, 2个时钟选择器以及一个用于互补配对 PWM 的死区发生器
 - ◆ 最多8路16-位数字捕捉定时器 (共享 PWM 定时器), 提供8路捕捉输入 (上升, 下降或两者皆可)
 - ◆ 支持 one shot 和连续模式
 - ◆ 支持捕捉 (Capture) 中断
- UART
 - ◆ 最多 2 组16-字节 FIFO UART 控制器
 - ◆ UART 端口支持流控 (TX, RX, CTSn 和 RTSn)
 - ◆ 支持 IrDA (SIR) 功能
 - ◆ 支持 LIN 功能
 - ◆ 支持 RS-485 9 位模式和方向控制
 - ◆ 可编程波特率发生器
 - ◆ 支持 PDMA 模式
 - ◆ UART 可以从掉电模式唤醒系统
- SPI
 - ◆ 最高支持 3 组 SPI 控制器

- ◆ 主机速率最高到 32 MHz，从机最高至 16 MHz
- ◆ 支持SPI/MICROWIRE 主机/从机模式
- ◆ 全双工同步串行数据传输
- ◆ 可变传输数据长度（4 到 32 位）
- ◆ 可配置 MSB 或 LSB 在前的数据传输
- ◆ 在时钟上升沿或下降沿接收 (RX) 还是发送 (TX) 是独立配置的
- ◆ 当 SPI 控制器作为主机时，2 条从机/设备选择线；作为从机时，1条从机/设备选择线
- ◆ 支持32-bit 传输模式下的字节睡眠模式
- ◆ 支持2 通道 PDMA 请求，1个用于发送，另一个用于接收
- ◆ 支持三线模式，无从机选择信号，双向接口
- ◆ SPI 可以从掉电或空闲模式唤醒系统
- I²C
 - ◆ 最多支持 2 组 I²C 设备
 - ◆ 主机/从机最高达 1Mbit/s
 - ◆ 主从机之间双向数据传输
 - ◆ 多主机总线支持（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
 - ◆ 串行时钟同步可作为握手方式控制总线上数据暂停及恢复传送
 - ◆ 内建的一个 14-位超时计数器在 I²C 总线中止以及超时计数器溢出时会请求 I²C 中断
 - ◆ 可编程的时钟适用于不同速率控制
 - ◆ 支持 7-位 地址模式
 - ◆ 支持多地址识别（4个从机地址带mask选项）
- I²S
 - ◆ 外部音频 CODEC 接口
 - ◆ 可做主机也可作从机模式
 - ◆ 能处理8, 16, 24 和 32 位字
 - ◆ 支持单声道和立体声的音频数据
 - ◆ 支持 I²S 和 最高有效位数据格式
 - ◆ 提供两组 8 字的FIFO数据缓存，一组用于发送，一组用于接收
 - ◆ 缓冲区超过可编程边界时，产生中断请求
 - ◆ 支持两组 PDMA 请求，一组用于发送，另一组用于接收

- ADC
 - ◆ 12-位 SAR ADC 最高 2Msps 转换速率
 - ◆ 最多 12-通道单端模式外部输入(PA.0 ~ PA.7 和 PD.0 ~ PD.3)
 - ◆ 6个内部通道来自 DAC0, DAC1, 内部参考电压 (Int_VREF), 温度传感器, AVDD, 和 AVSS.
 - ◆ 支持三个参考电压源: VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 支持单扫描, 单周期扫描和连续扫描模式
 - ◆ 每个通道有独立的结果寄存器
 - ◆ 只能扫描使能的通道
 - ◆ 阈电压侦测 (比较功能)
 - ◆ 软件编程或外部管脚触发开始转换
 - ◆ 支持 PDMA 模式
 - ◆ 支持最多四个定时器超时事件 (TRM0, TMR1, TMR2 和 TMR3) 使能 ADC
- DAC
 - ◆ 12-位单输出 400K 转换速率
 - ◆ 支持三个参考电压源 VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 两 DACs 的同步更新能力 (组功能)
 - ◆ 支持最多四个定时器超时事件(TMR0, TMR1, TMR2 和 TMR3) , 软件或 PDMA 触发 DAC 去进行转换
- SmartCard (SC)
 - ◆ 依据 ISO-7816-3 T=0, T=1
 - ◆ 支持最多 3 个 ISO-7816-3 端口
 - ◆ 单独的接收/发送 4 字节项目 FIFO用于数据负载
 - ◆ 可编程的传送时钟频率
 - ◆ 可编程接收器缓存触发水平
 - ◆ 可编程保卫时间选择 (11 ETU ~ 266 ETU)
 - ◆ 一个 24-位和两个 8 位超时计数器用于回复请求 (Answer to Reset (ATR))并等待时间运行
 - ◆ 支持自动逆转换功能
 - ◆ 支持停止时钟电平和时钟停止 (时钟保持) 功能
 - ◆ 支持发送器和接收器错误重试和错误限制功能
 - ◆ 支持硬件激活序列处理
 - ◆ 支持硬件热重置序列处理
 - ◆ 支持硬件释放序列处理
 - ◆ 支持当检测到卡被移除时, 硬件自动释放序列

- ◆ 支持 UART 模式 (半双工)
- LCD
 - ◆ LCD 驱动最多支持4 COM x 40 SEG 或 6 COM x 38 SEG
 - ◆ 支持静态, 1/2 偏置和 1/3 偏置电压
 - ◆ 六种显示模式: 静态, 1/2占空率, 1/3占空率、1/4 占空率, 1/5占空率和 1/6占空率.
 - ◆ 通过分频器可选 LCD 的频率
 - ◆ 可配置的帧频率
 - ◆ 内部电荷泵, 可调节的对比度调节装置
 - ◆ 可配置的电荷泵频率
 - ◆ 闪烁性能
 - ◆ 支持R-type/C-type方式
 - ◆ LCD 帧中断
- EBI (External bus interface) (外部总线接口)
 - ◆ 可访问的空间: 8位模式为64KB 或16位模式为 128KB
 - ◆ 支持 8bit/16bit 数据宽度
 - ◆ 在16-位 数据宽度模式下支持字节写入
- 一个内建的温度传感器, 分辨率为 1℃
- 96-位独一无二的 ID
- 128-位唯一客户ID
- 工作温度: -40℃~85℃
- 封装:
 - ◆ 绿色(无铅)封装 (RoHS)
 - ◆ LQFP 128-pin(14x14) / 64-pin(10x10) / 64-pin(7x7)

2.3 Nano120 特性 – USB Connectivity Line

- Core
 - ◆ ARM® Cortex™-M0 内核最高运行到42 MHz
 - ◆ 一个 24-位系统定时器
 - ◆ 支持低功耗睡眠模式
 - ◆ 单周期 32-位硬件乘法器
 - ◆ 嵌套向量中断控制器 (NVIC) 用于控制32个中断源，每个中断源可设置为4个优先级
 - ◆ 支持串行线调试 (SWD) 带2个观察点/4个断点
- 欠压检测
 - ◆ 内建2.5V/2.0V/1.7V BOD 用于宽泛的工作电压范围操作
- Flash EPROM 存储器
 - ◆ 高达 42MHz 时的不连续地址读访问零等待状态
 - ◆ 64K/32K/128K-字节应用程序存储器 (APROM)
 - ◆ 4KB 在系统编程 (ISP) 加载程序存储器 (LDROM)
 - ◆ 可编程数据 flash 起始地址和存储器大小以512字节为页擦除单元
 - ◆ 在系统编程 (ISP)/在应用编程 (IAP) 更新芯片的Flash EPROM
- SRAM 存储器
 - ◆ 16K/8K-字节内建 SRAM
 - ◆ 支持 DMA 模式
- DMA : 支持 8 通道: 1个 VDMA 通道, 6 PDMA 通道, 和 1个 CRC 通道
 - ◆ VDMA
 - 内存-到-内存传输
 - 支持跨距的块传输
 - 支持 字/半字/字节边界地址
 - 支持地址方向: 增长和减少
 - ◆ PDMA
 - 外设-到-内存, 内存-到-外设, 内存-到-内存 传输
 - 支持字边界地址
 - 在内存-到-内存模式, 支持字对齐传输长度
 - 在外设-到-内存, 内存-到-外设传输模式, 支持 字/半字/字节 对齐传输长度
 - 外设作为源或目标时, 支持字/半字/字节 传输数据宽度
 - 支持地址方向: 增长, 固定和回绕
 - ◆ CRC

- 支持4个通用多项式 CRC-CCITT, CRC-8, CRC-16, 和 CRC-32
 - ◆ CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - ◆ CRC-8: $X^8 + X^2 + X + 1$
 - ◆ CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - ◆ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 时钟控制
 - ◆ 针对不同应用可灵活选择时钟
 - ◆ 当打开自动修正功能, 在全温度范围, 内置 12MHz OSC可以校正到0.25%的偏差 (系统必须有外部 32.768 kHz 晶振输入), 否则在全温度范围12MHz OSC 会有 2 % 偏差。
 - ◆ 低功率10 kHz OSC 用于 看门狗 和低功耗操作模式
 - ◆ 支持一组PLL, 高达120 MHz, 用于高性能的系统运行和 USB 应用(48 MHz).
 - ◆ 外部 4~24 MHz 晶振输入用于精准的定时操作
 - ◆ 外部 32.768 kHz晶振输入用于 RTC 及低功耗模式操作
- GPIO
 - ◆ 三种 I/O 模式:
 - 推挽输出
 - 开漏输出
 - 高阻输入
 - ◆ 所有输入为 Schmitt 触发
 - ◆ I/O 引脚可被配置为边沿/电平触发模式的中断源
 - ◆ 支持高驱动和高灌入 IO 模式
 - ◆ 支持 5V 输入, 除了PA.0 ~ PA.7, PD.0 ~ PD.1 和 PC.6 ~ PC.7
- Timer
 - ◆ 支持4组32位定时器, 每个定时器有一个24位向上计数定时器和一个8位预分频计数器
 - ◆ 每个定时器有独立的时钟源
 - ◆ 提供 单次、周期、输出翻转和连续计数操作模式
 - ◆ 内部触发事件到 ADC, DAC, 与 PDMA 模块
 - ◆ 支持 PDMA 模式
 - ◆ Timer 可以从掉电或空闲模式唤醒系统
- 看门狗定时器
 - ◆ 时钟源来自 LIRC. (内部 10KHz 低速震荡器时钟)
 - ◆ 从1.6ms ~ 26sec 有可选的定时溢出周期 (取决于所选的时钟源)

- ◆ 看门狗定时溢出的中断/复位选择
- ◆ WDT 可以从掉电模式唤醒系统
- 窗口看门狗定时器(WWDT)
 - ◆ 6-位 下数计数器 6-位 比较值使窗口周期灵活可变
 - ◆ 可选的 WWDT 时钟预分频计数器使 WWDT 溢出间隔可变.
- RTC
 - ◆ 通过频率补偿寄存器 (FCR) 支持软件频率补偿功能
 - ◆ 支持 RTC 计数(秒, 分, 小时) 及万年历功能 (日, 月, 年)
 - ◆ 支持闹铃寄存器 (秒, 分, 小时, 日, 月, 年)
 - ◆ 可选择为12小时制或24小时
 - ◆ 闰年自动识别
 - ◆ 支持周期时间滴答中断, 包括8个可选周期 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 和 1 秒
 - ◆ RTC 可以从掉电模式唤醒系统
 - ◆ 支持 80 字节备用寄存器以及一个探测引脚来清除这些备用寄存器的内容
- PWM/Capture
 - ◆ 内建 2 个 PWM 模块, 每个模块有2个 16-位 PWM 产生器
 - ◆ 提供 8个 PWM 输出或四组互补配对 PWM 输出
 - ◆ 每个 PWM 产生器配有一个时钟分频器, 一个8-位时钟预分频, 2个时钟选择器以及一个用于互补配对 PWM 的死区发生器
 - ◆ 最多8路16-位数字捕捉定时器 (共享 PWM 定时器), 提供8路捕捉输入 (上升, 下降或两者皆可)
 - ◆ 支持 one shot 和连续模式
 - ◆ 支持捕捉 (Capture) 中断
- UART
 - ◆ 最多 2 组16-字节 FIFO UART 控制器
 - ◆ UART 端口支持流控 (TX, RX, CTSn 和 RTSn)
 - ◆ 支持 IrDA (SIR) 功能
 - ◆ 支持 LIN 功能
 - ◆ 支持 RS-485 9 位模式和方向控制
 - ◆ 可编程波特率发生器
 - ◆ 支持 PDMA 模式
 - ◆ UART 可以从掉电模式唤醒系统
- SPI
 - ◆ 最高支持 3 组 SPI 控制器

- ◆ 主机速率最高到 32 MHz，从机最高至 16 MHz
- ◆ 支持SPI/MICROWIRE 主机/从机模式
- ◆ 全双工同步串行数据传输
- ◆ 可变传输数据长度（4 到 32 位）
- ◆ 可配置 MSB 或 LSB 在前的数据传输
- ◆ 在时钟上升沿或下降沿接收 (RX) 还是发送 (TX) 是独立配置的
- ◆ 当 SPI 控制器作为主机时，2 条从机/设备选择线；作为从机时，1条从机/设备选择线
- ◆ 支持32-bit 传输模式下的字节睡眠模式
- ◆ 支持2 通道 PDMA 请求，1个用于发送，另一个用于接收
- ◆ 支持三线模式，无从机选择信号，双向接口
- ◆ SPI 可以从掉电或空闲模式唤醒系统
- I²C
 - ◆ 最多支持 2 组 I²C 设备
 - ◆ 主机/从机最高达 1Mbit/s
 - ◆ 主从机之间双向数据传输
 - ◆ 多主机总线支持（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
 - ◆ 串行时钟同步可作为握手方式控制总线上数据暂停及恢复传送
 - ◆ 内建的一个 14-位超时计数器在 I²C 总线中止以及超时计数器溢出时会请求 I²C 中断
 - ◆ 可编程的时钟适用于不同速率控制
 - ◆ 支持 7-位 地址模式
 - ◆ 支持多地址识别（4个从机地址带mask选项）
- I²S
 - ◆ 外部音频 CODEC 接口
 - ◆ 可做主机也可作从机模式
 - ◆ 能处理8, 16, 24 和 32 位字
 - ◆ 支持单声道和立体声的音频数据
 - ◆ 支持 I²S 和 最高有效位数据格式
 - ◆ 提供两组 8 字的FIFO数据缓存，一组用于发送，一组用于接收
 - ◆ 缓冲区超过可编程边界时，产生中断请求
 - ◆ 支持两组 PDMA 请求，一组用于发送，另一组用于接收

- ADC
 - ◆ 12-位 SAR ADC 最高 2Msps 转换速率
 - ◆ 最多 12-通道单端模式外部输入(PA.0 ~ PA.7 和 PD.0 ~ PD.3)
 - ◆ 6个内部通道来自 DAC0, DAC1, 内部参考电压 (Int_VREF), 温度传感器, AVDD, 和 AVSS.
 - ◆ 支持三个参考电压源: VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 支持单扫描, 单周期扫描和连续扫描模式
 - ◆ 每个通道有独立的结果寄存器
 - ◆ 只能扫描使能的通道
 - ◆ 阈电压侦测 (比较功能)
 - ◆ 软件编程或外部管脚触发开始转换
 - ◆ 支持 PDMA 模式
 - ◆ 支持最多四个定时器超时事件 (TMR0, TMR1, TMR2 和 TMR3) 使能 ADC
- DAC
 - ◆ 12-位单输出 400K 转换速率
 - ◆ 支持三个参考电压源 VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 两 DACs 的同步更新能力 (组功能)
 - ◆ 支持最多四个定时器超时事件(TMR0, TMR1, TMR2 和 TMR3) , 软件或 PDMA 触发 DAC 去进行转换
- SmartCard (SC)
 - ◆ 依据 ISO-7816-3 T=0, T=1
 - ◆ 支持最多 3 个 ISO-7816-3 端口
 - ◆ 单独的接收/发送 4 字节项目 FIFO用于数据负载
 - ◆ 可编程的传送时钟频率
 - ◆ 可编程接收器缓存触发水平
 - ◆ 可编程保卫时间选择 (11 ETU ~ 266 ETU)
 - ◆ 一个 24-位和两个 8 位超时计数器用于回复请求 (Answer to Reset (ATR))并等待时间运行
 - ◆ 支持自动逆转换功能
 - ◆ 支持停止时钟电平和时钟停止 (时钟保持) 功能
 - ◆ 支持发送器和接收器错误重试和错误限制功能
 - ◆ 支持硬件激活序列处理
 - ◆ 支持硬件热重置序列处理
 - ◆ 支持硬件释放序列处理
 - ◆ 支持当检测到卡被移除时, 硬件自动释放序列

- ◆ 支持 UART 模式 (半双工)
- USB 2.0 全速设备
 - ◆ 一组 USB 2.0 FS 设备 12Mbps
 - ◆ 片上 USB 收发器
 - ◆ 提供一个中断源用于4个中断事件
 - ◆ 支持 Control, Bulk In/Out, Interrupt 和 Isochronous 传输
 - ◆ 当总线没有信号持续3 ms, 自动挂起功能
 - ◆ 提供 8 个可编程的端点
 - ◆ 包含 512 字节的内部 SRAM 作为 USB 缓存
 - ◆ 提供远程唤醒功能
- EBI (External bus interface) (外部总线接口)
 - ◆ 可访问的空间: 8位模式为64KB 或16位模式为 128KB
 - ◆ 支持 8bit/16bit 数据宽度
 - ◆ 在16-位 数据宽度模式下支持字节写入
- 一个内建的温度传感器, 分辨率为 1℃
- 96-位独一无二 的ID
- 128-位唯一客户ID
- 工作温度: -40℃~85℃
- 封装:
 - ◆ 绿色 (无铅) 封装 (RoHS)
 - ◆ LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7)

2.4 Nano130 特性 – Advanced Line

- Core
 - ◆ ARM® Cortex™-M0 内核最高运行到42 MHz
 - ◆ 一个 24-位系统定时器
 - ◆ 支持低功耗睡眠模式
 - ◆ 单周期 32-位硬件乘法器
 - ◆ 嵌套向量中断控制器 (NVIC) 用于控制32个中断源，每个中断源可设置为4个优先级
 - ◆ 支持串行线调试 (SWD) 带2个观察点/4个断点
- 欠压检测
 - ◆ 内建2.5V/2.0V/1.7V BOD 用于宽泛的工作电压范围操作
- Flash EPROM 存储器
 - ◆ 高达 42MHz 时的不连续地址读访问零等待状态
 - ◆ 64K/32K/128K-字节应用程序存储器 (APROM)
 - ◆ 4KB 在系统编程 (ISP) 加载程序存储器 (LDROM)
 - ◆ 可编程数据 flash 起始地址和存储器大小以512字节为页擦除单元
 - ◆ 在系统编程 (ISP)/在应用编程 (IAP) 更新芯片的Flash EPROM
- SRAM 存储器
 - ◆ 16K/8K-字节内建 SRAM
 - ◆ 支持 DMA 模式
- DMA : 支持 8 通道: 1个 VDMA 通道, 6 PDMA 通道, 和 1个 CRC 通道
 - ◆ VDMA
 - 内存-到-内存传输
 - 支持跨距的块传输
 - 支持 字/半字/字节边界地址
 - 支持地址方向: 增长和减少
 - ◆ PDMA
 - 外设-到-内存, 内存-到-外设, 内存-到-内存 传输
 - 支持字边界地址
 - 在内存-到-内存模式, 支持字对齐传输长度
 - 在外设-到-内存, 内存-到-外设传输模式, 支持 字/半字/字节 对齐传输长度
 - 外设作为源或目标时, 支持字/半字/字节 传输数据宽度
 - 支持地址方向: 增长, 固定和回绕
 - ◆ CRC

- 支持4个通用多项式 CRC-CCITT, CRC-8, CRC-16, 和 CRC-32
 - ◆ CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - ◆ CRC-8: $X^8 + X^2 + X + 1$
 - ◆ CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - ◆ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 时钟控制
 - ◆ 针对不同应用可灵活选择时钟
 - ◆ 当打开自动修正功能，在全温度范围，内置 12MHz OSC可以校正到0.25%的偏差 (系统必须有外部 32.768 kHz 晶振输入)，否则在全温度范围12MHz OSC 会有 2 % 偏差。
 - ◆ 低功率10 kHz OSC 用于 看门狗 和低功耗操作模式
 - ◆ 支持一组PLL, 高达120 MHz, 用于高性能的系统运行和 USB 应用(48 MHz).
 - ◆ 外部 4~24 MHz 晶振输入用于精准的定时操作
 - ◆ 外部 32.768 kHz晶振输入用于 RTC 及低功耗模式操作
- GPIO
 - ◆ 三种 I/O 模式:
 - 推挽输出
 - 开漏输出
 - 高阻输入
 - ◆ 所有输入为 Schmitt 触发
 - ◆ I/O 引脚可被配置为边沿/电平触发模式的中断源
 - ◆ 支持高驱动和高灌入 IO 模式
 - ◆ 支持 5V 输入，除了PA.0 ~ PA.7, PD.0 ~ PD.1 和 PC.6 ~ PC.7
- Timer
 - ◆ 支持4组32位定时器，每个定时器有一个24位向上计数定时器和一个8位预分频计数器
 - ◆ 每个定时器有独立的时钟源
 - ◆ 提供 单次、周期、输出翻转和连续计数操作模式
 - ◆ 内部触发事件到 ADC, DAC, 与 PDMA 模块
 - ◆ 支持 PDMA 模式
 - ◆ Timer 可以从掉电或空闲模式唤醒系统
- 看门狗定时器
 - ◆ 时钟源来自 LIRC. (内部 10KHz 低速震荡器时钟)
 - ◆ 从1.6ms ~ 26sec 有可选的定时溢出周期 (取决于所选的时钟源)

- ◆ 看门狗定时溢出的中断/复位选择
- ◆ WDT 可以从掉电模式唤醒系统
- 窗口看门狗定时器(WWDT)
 - ◆ 6-位 下数计数器 6-位 比较值使窗口周期灵活可变
 - ◆ 可选的 WWDT 时钟预分频计数器使 WWDT 溢出间隔可变.
- RTC
 - ◆ 通过频率补偿寄存器 (FCR) 支持软件频率补偿功能
 - ◆ 支持 RTC 计数(秒, 分, 小时) 及万年历功能 (日, 月, 年)
 - ◆ 支持闹铃寄存器 (秒, 分, 小时, 日, 月, 年)
 - ◆ 可选择为12小时制或24小时
 - ◆ 闰年自动识别
 - ◆ 支持周期时间滴答中断, 包括8个可选周期 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 和 1 秒
 - ◆ RTC 可以从掉电模式唤醒系统
 - ◆ 支持 80 字节备用寄存器以及一个探测引脚来清除这些备用寄存器的内容
- PWM/Capture
 - ◆ 内建 2 个 PWM 模块, 每个模块有2个 16-位 PWM 产生器
 - ◆ 提供 8个 PWM 输出或四组互补配对 PWM 输出
 - ◆ 每个 PWM 产生器配有一个时钟分频器, 一个8-位时钟预分频, 2个时钟选择器以及一个用于互补配对 PWM 的死区发生器
 - ◆ 最多8路16-位数字捕捉定时器 (共享 PWM 定时器), 提供8路捕捉输入 (上升, 下降或两者皆可)
 - ◆ 支持 one shot 和连续模式
 - ◆ 支持捕捉 (Capture) 中断
- UART
 - ◆ 最多 2 组16-字节 FIFO UART 控制器
 - ◆ UART 端口支持流控 (TX, RX, CTSn 和 RTSn)
 - ◆ 支持 IrDA (SIR) 功能
 - ◆ 支持 LIN 功能
 - ◆ 支持 RS-485 9 位模式和方向控制
 - ◆ 可编程波特率发生器
 - ◆ 支持 PDMA 模式
 - ◆ UART 可以从掉电模式唤醒系统
- SPI
 - ◆ 最高支持 3 组 SPI 控制器

- ◆ 主机速率最高到 32 MHz，从机最高至 16 MHz
- ◆ 支持SPI/MICROWIRE 主机/从机模式
- ◆ 全双工同步串行数据传输
- ◆ 可变传输数据长度（4 到 32 位）
- ◆ 可配置 MSB 或 LSB 在前的数据传输
- ◆ 在时钟上升沿或下降沿接收 (RX) 还是发送 (TX) 是独立配置的
- ◆ 当 SPI 控制器作为主机时，2 条从机/设备选择线；作为从机时，1条从机/设备选择线
- ◆ 支持32-bit 传输模式下的字节睡眠模式
- ◆ 支持2 通道 PDMA 请求，1个用于发送，另一个用于接收
- ◆ 支持三线模式，无从机选择信号，双向接口
- ◆ SPI 可以从掉电或空闲模式唤醒系统
- I²C
 - ◆ 最多支持 2 组 I²C 设备
 - ◆ 主机/从机最高达 1Mbit/s
 - ◆ 主从机之间双向数据传输
 - ◆ 多主机总线支持（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
 - ◆ 串行时钟同步可作为握手方式控制总线上数据暂停及恢复传送
 - ◆ 内建的一个 14-位超时计数器在 I²C 总线中止以及超时计数器溢出时会请求 I²C 中断
 - ◆ 可编程的时钟适用于不同速率控制
 - ◆ 支持 7-位 地址模式
 - ◆ 支持多地址识别（4个从机地址带mask选项）
- I²S
 - ◆ 外部音频 CODEC 接口
 - ◆ 可做主机也可作从机模式
 - ◆ 能处理8, 16, 24 和 32 位字
 - ◆ 支持单声道和立体声的音频数据
 - ◆ 支持 I²S 和 最高有效位数据格式
 - ◆ 提供两组 8 字的FIFO数据缓存，一组用于发送，一组用于接收
 - ◆ 缓冲区超过可编程边界时，产生中断请求
 - ◆ 支持两组 PDMA 请求，一组用于发送，另一组用于接收

- ADC
 - ◆ 12-位 SAR ADC 最高 2Msps 转换速率
 - ◆ 最多 12-通道单端模式外部输入(PA.0 ~ PA.7 和 PD.0 ~ PD.3)
 - ◆ 6个内部通道来自 DAC0, DAC1, 内部参考电压 (Int_VREF), 温度传感器, AVDD, 和 AVSS.
 - ◆ 支持三个参考电压源: VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 支持单扫描, 单周期扫描和连续扫描模式
 - ◆ 每个通道有独立的结果寄存器
 - ◆ 只能扫描使能的通道
 - ◆ 阈电压侦测 (比较功能)
 - ◆ 软件编程或外部管脚触发开始转换
 - ◆ 支持 PDMA 模式
 - ◆ 支持最多四个定时器超时事件 (TRM0, TMR1, TMR2 和 TMR3) 使能 ADC
- DAC
 - ◆ 12-位单输出 400K 转换速率
 - ◆ 支持三个参考电压源 VREF 引脚, 内部参考电压 (Int_VREF), 和 AVDD.
 - ◆ 两 DACs 的同步更新能力 (组功能)
 - ◆ 支持最多四个定时器超时事件(TMR0, TMR1, TMR2 和 TMR3) , 软件或 PDMA 触发 DAC 去进行转换
- SmartCard (SC)
 - ◆ 依据 ISO-7816-3 T=0, T=1
 - ◆ 支持最多 3 个 ISO-7816-3 端口
 - ◆ 单独的接收/发送 4 字节项目 FIFO用于数据负载
 - ◆ 可编程的传送时钟频率
 - ◆ 可编程接收器缓存触发水平
 - ◆ 可编程保卫时间选择 (11 ETU ~ 266 ETU)
 - ◆ 一个 24-位和两个 8 位超时计数器用于回复请求 (Answer to Reset (ATR))并等待时间运行
 - ◆ 支持自动逆转换功能
 - ◆ 支持停止时钟电平和时钟停止 (时钟保持) 功能
 - ◆ 支持发送器和接收器错误重试和错误限制功能
 - ◆ 支持硬件激活序列处理
 - ◆ 支持硬件热重置序列处理
 - ◆ 支持硬件释放序列处理
 - ◆ 支持当检测到卡被移除时, 硬件自动释放序列

- ◆ 支持 UART 模式 (半双工)
- LCD
 - ◆ LCD 驱动最多支持4 COM x 40 SEG 或 6 COM x 38 SEG
 - ◆ 支持静态, 1/2 偏置和 1/3 偏置电压
 - ◆ 六种显示模式: 静态, 1/2占空率, 1/3占空率、1/4 占空率, 1/5占空率和 1/6占空率.
 - ◆ 通过分频器可选 LCD 的频率
 - ◆ 可配置的帧频率
 - ◆ 内部电荷泵, 可调节的对比度调节装置
 - ◆ 可配置的电荷泵频率
 - ◆ 闪烁性能
 - ◆ 支持R-type/C-type方式
 - ◆ LCD 帧中断
- USB 2.0 全速设备
 - ◆ 一组 USB 2.0 FS 设备 12Mbps
 - ◆ 片上 USB 收发器
 - ◆ 提供一个中断源用于4个中断事件
 - ◆ 支持 Control, Bulk In/Out, Interrupt 和 Isochronous 传输
 - ◆ 当总线没有信号持续3 ms, 自动挂起功能
 - ◆ 提供 8 个可编程的端点
 - ◆ 包含 512 字节的内部 SRAM 作为 USB 缓存
 - ◆ 提供远程唤醒功能
- EBI (External bus interface) (外部总线接口)
 - ◆ 可访问的空间: 8位模式为64KB 或16位模式为 128KB
 - ◆ 支持 8bit/16bit 数据宽度
 - ◆ 在16-位 数据宽度模式下支持字节写入
- 一个内建的温度传感器, 分辨率为 1°C
- 96-位独一无二 的ID
- 128-位唯一客户ID
- 工作温度: -40°C ~85°C
- 封装:
 - ◆ 绿色 (无铅) 封装 (RoHS)
 - ◆ LQFP 128-pin(14x14) / 64-pin(7x7)

3 编号信息列表及管脚名称定义

3.1 NuMicro™ Nano100 系列选型编码

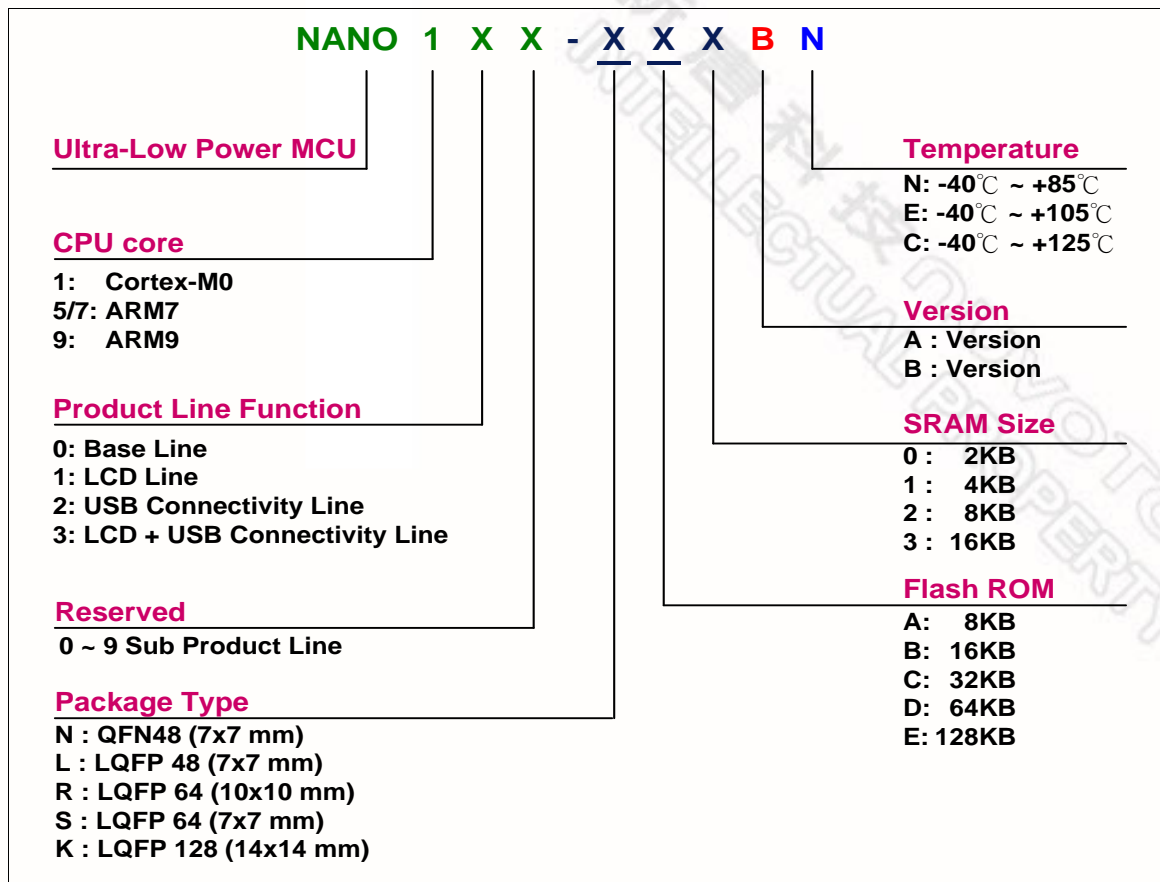


图 3-1 NuMicro™ Nano100 系列选型编码

3.2 NuMicro™ Nano100 产品选型指南

3.2.1 NuMicro™ Nano100 Base Line选型指南

Part No.	Flash (Kbytes)	SRAM (Kbytes)	Data Flash	ISP ROM (Kbytes)	I/O	Timer (32-bit)	Connectivity				I2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10KHz 12MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP ICP	Package
							UART	SPI	I2C	USB												
NANO100NC2BN	32K	8K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	0	7	V	-	V	8	-	2	2	V	QFN48*
NANO100ND2BN	64K	8K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100ND3BN	64K	16K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100NE3BN	128K	16K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100LC2BN	32K	8K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LD2BN	64K	8K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LD3BN	64K	16K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LE3BN	128K	16K	Configurable	4K	up to 38	4x32-bit	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100SC2BN	32K	8K	Configurable	4K	up to 52	4x32-bit	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SD2BN	64K	8K	Configurable	4K	up to 52	4x32-bit	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SD3BN	64K	16K	Configurable	4K	up to 52	4x32-bit	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SE3BN	128K	16K	Configurable	4K	up to 52	4x32-bit	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100KD3BN	64K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	-	2	3	V	LQFP128
NANO100KE3BN	128K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	-	2	3	V	LQFP128

QFN*48* : 7x7, pitch 0.5 mm ; LQFP48 : 7x7, pitch 0.5 mm ; LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

表 3-1 Nano100 Base Line 选型表

3.2.2 NuMicro™ Nano110 LCD Line选型指南

Part No.	Flash (Kbytes)	SRAM (Kbytes)	Data Flash	ISP ROM (Kbytes)	I/O	Timer (32-bit)	Connectivity				I2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10KHz 12MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP ICP	Package
							UART	SPI	I2C	USB												
NANO110SC2BN	32K	8K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SD2BN	64K	8K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SD3BN	64K	16K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SE3BN	128K	16K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110RC2BN	32K	8K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RD2BN	64K	8K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RD3BN	64K	16K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RE3BN	128K	16K	Configurable	4K	up to 51	4x32-bit	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110KC2BN	32K	8K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KD2BN	64K	8K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KD3BN	64K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KE3BN	128K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128

LQFP64 : 7x7, pitch 0.4 mm ; LQFP64* : 10x10, pitch 0.5 mm ; LQFP128 : 14x14, pitch 0.4 mm

表 3-2 Nano110 LCD Line 选型表

3.2.3 NuMicro™ Nano120 USB Connectivity Line选型指南

Part No.	Flash (Kbytes)	SRAM (Kbytes)	Data Flash	ISP ROM (Kbytes)	I/O	Timer (32-bit)	Connectivity				I2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10KHz 12MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP ICP	Package
							UART	SPI	I2C	USB												
NANO120LC2BN	32K	8K	Configurable	4K	up to 34	4x32-bit	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LD2BN	64K	8K	Configurable	4K	up to 34	4x32-bit	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LD3BN	64K	16K	Configurable	4K	up to 34	4x32-bit	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LE3BN	128K	16K	Configurable	4K	up to 34	4x32-bit	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120SC2BN	32K	8K	Configurable	4K	up to 48	4x32-bit	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SD2BN	64K	8K	Configurable	4K	up to 48	4x32-bit	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SD3BN	64K	16K	Configurable	4K	up to 48	4x32-bit	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SE3BN	128K	16K	Configurable	4K	up to 48	4x32-bit	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120KD3BN	64K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	-	2	3	V	LQFP128
NANO120KE3BN	128K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	-	2	3	V	LQFP128

LQFP48 : 7x7, pitch 0.5 mm ; LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

表 3-3 Nano120 USB Connectivity Line 选型表

3.2.4 NuMicro™ Nano130 Advanced Line选型指南

Part No.	Flash (Kbytes)	SRAM (Kbytes)	Data Flash	ISP ROM (Kbytes)	I/O	Timer (32-bit)	Connectivity				I2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10KHz 12MHz	PDMA	LCD	DAC (12-bit)	ISO- 7816-3	ISP ICP	Package
							UART	SP	I2C	USB												
NANO130SC2BN	32K	8K	Configurable	4K	up to 47	4x32-bit	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SD2BN	64K	8K	Configurable	4K	up to 47	4x32-bit	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SD3BN	64K	16K	Configurable	4K	up to 47	4x32-bit	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SE3BN	128K	16K	Configurable	4K	up to 47	4x32-bit	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130KC2BN	32K	8K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KD2BN	64K	8K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KD3BN	64K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KE3BN	128K	16K	Configurable	4K	up to 86	4x32-bit	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128

LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

表 3-4 Nano130 Advanced Line 选型表

3.3 管脚配置

3.3.1 NuMicro™ Nano100 管脚图

3.3.1.1 NuMicro™ Nano100 LQFP 128-pin

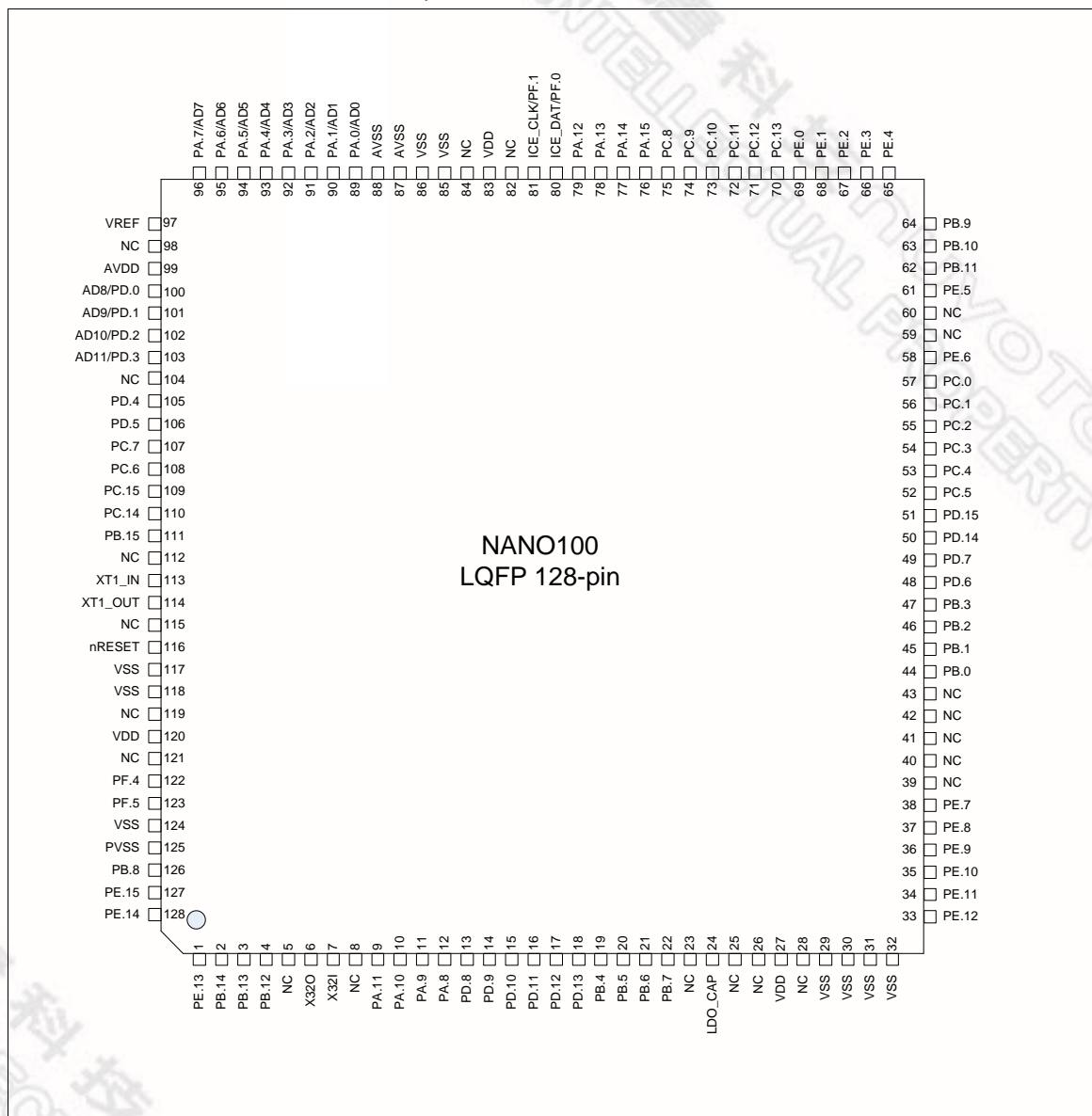


图 3-2 NuMicro™ Nano100 LQFP 128-管脚图

3.3.1.2 NuMicro™ Nano100 LQFP 64-pin

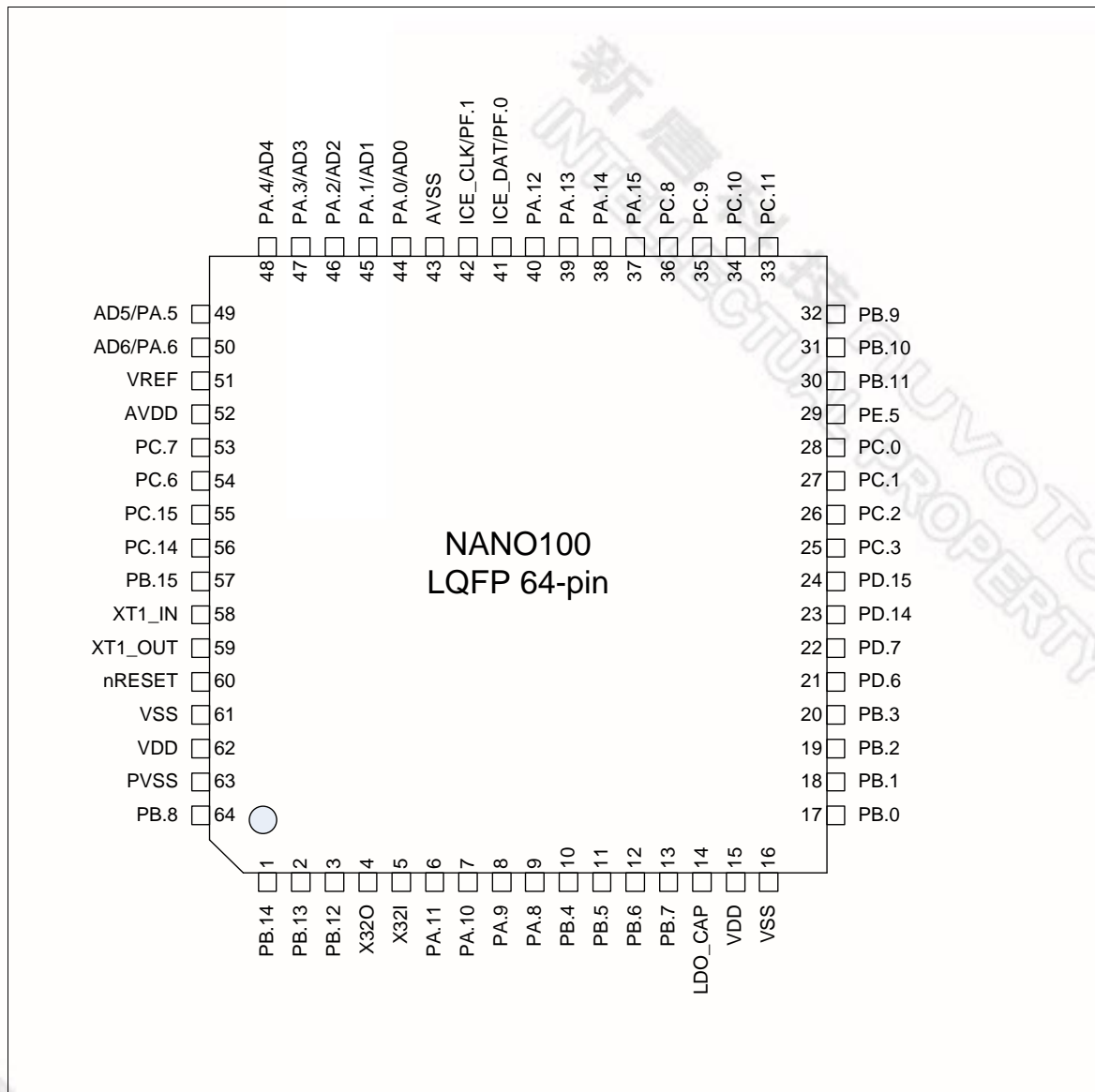


图 3-3 NuMicro™ Nano100 LQFP 64-pin 管脚图

3.3.1.3 NuMicro™ Nano100 LQFP/QFN 48-pin

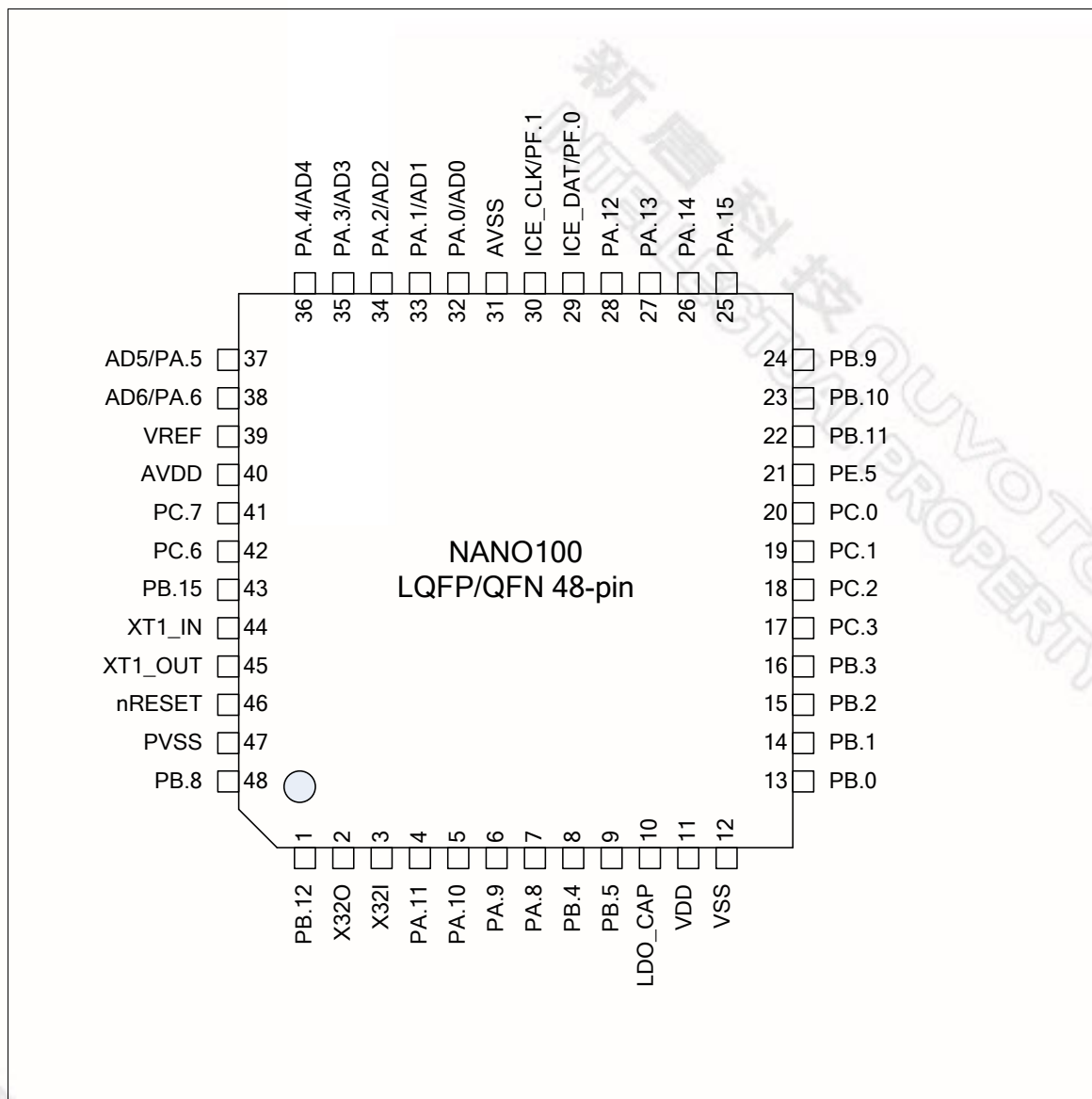


图 3-4 NuMicro™ Nano100 LQFP 48-pin 管脚图

3.3.2 NuMicro™ Nano110 管脚图

3.3.2.1 NuMicro™ Nano110 LQFP 128-pin

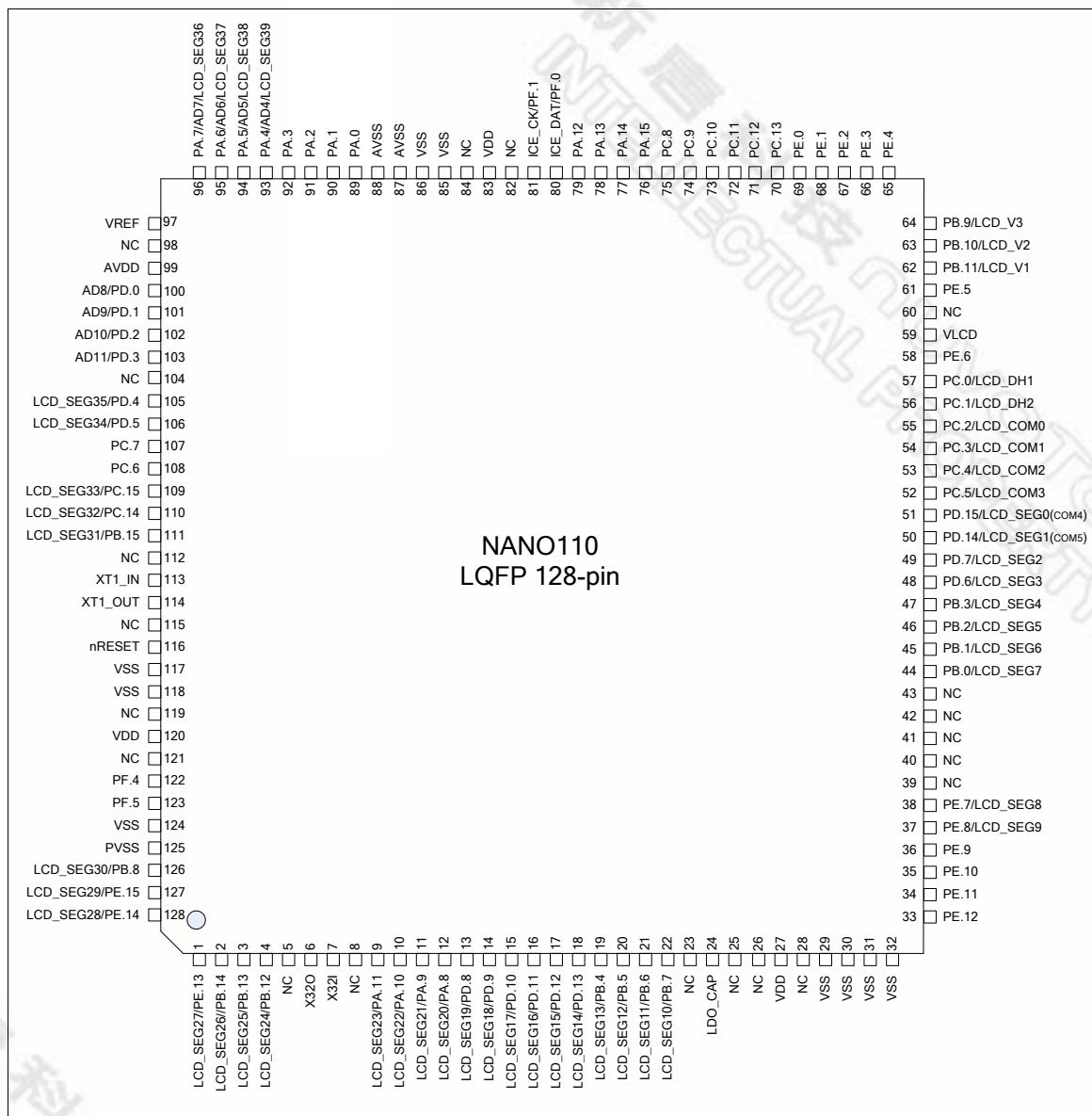


图 3-5 NuMicro™ Nano110 LQFP 128-pin 管脚图

3.3.2.2 NuMicro™ Nano110 LQFP 64-pin

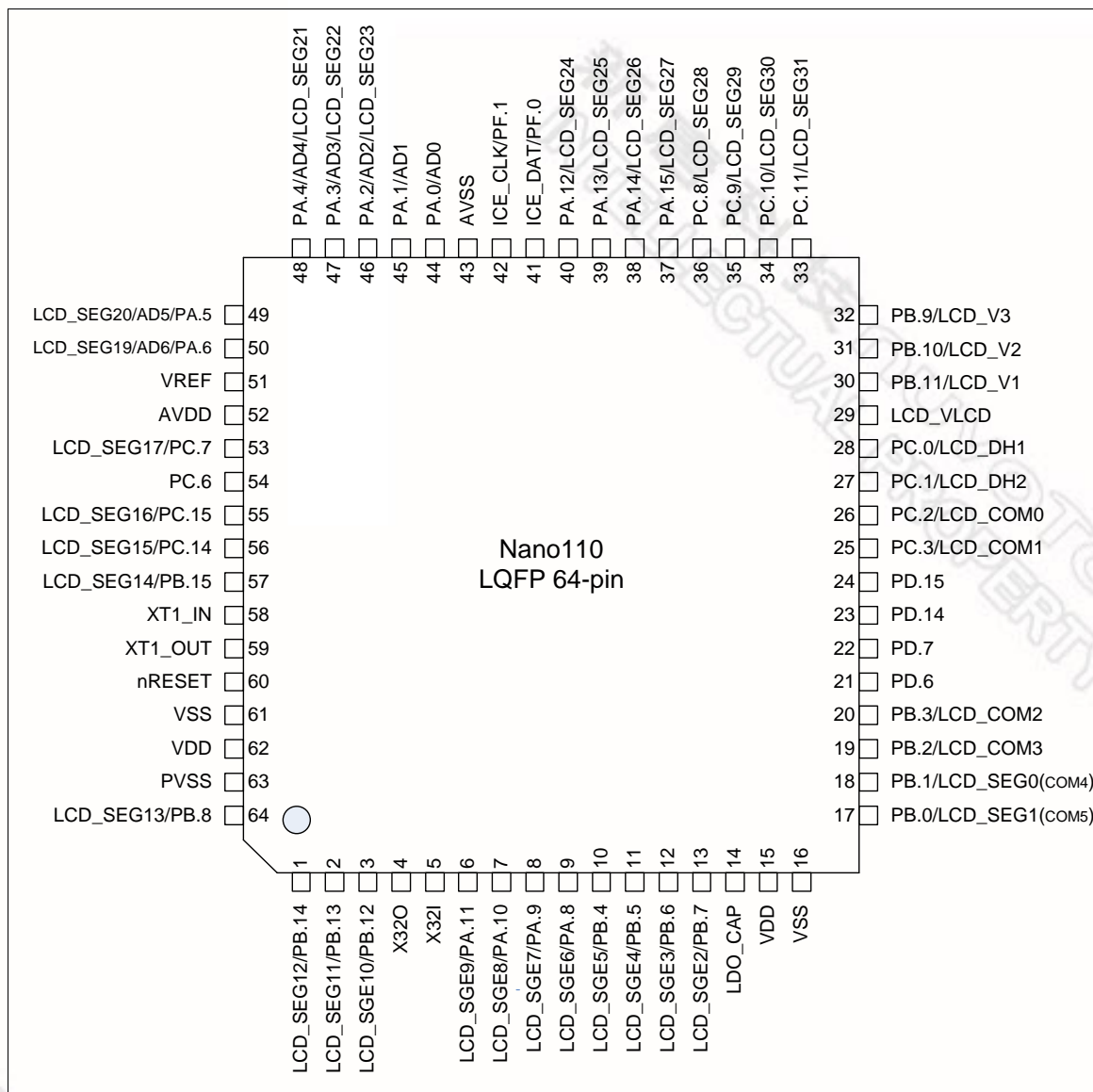
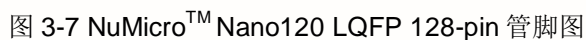


图 3-6 NuMicro™ Nano110 LQFP 64-pin 管脚图



3.3.3.2 NuMicro™ Nano120 LQFP 64-pin

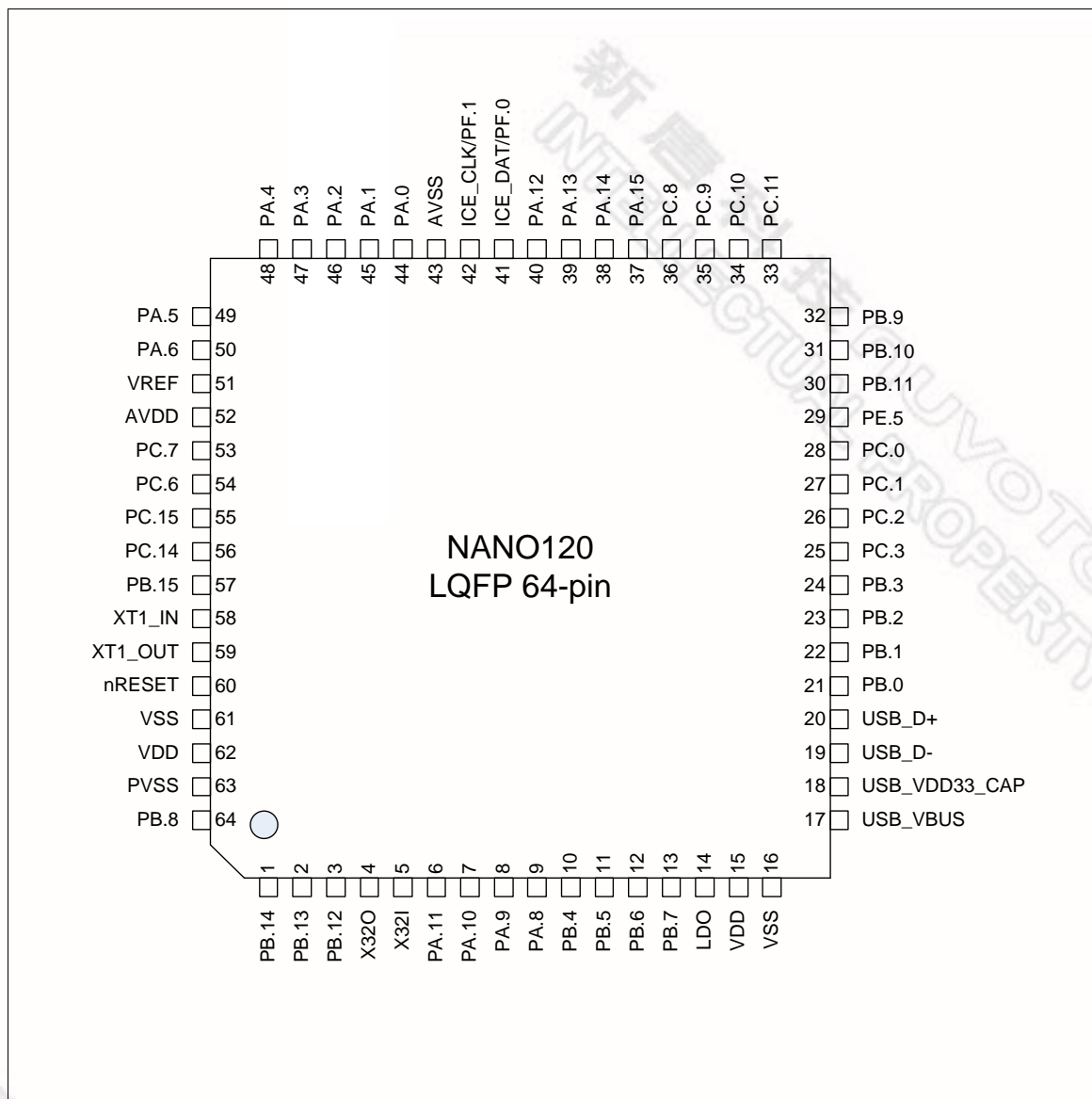


图 3-8 NuMicro™ Nano120 LQFP 64-pin 管脚图

3.3.3.3 NuMicro™ Nano120 LQFP 48-pin

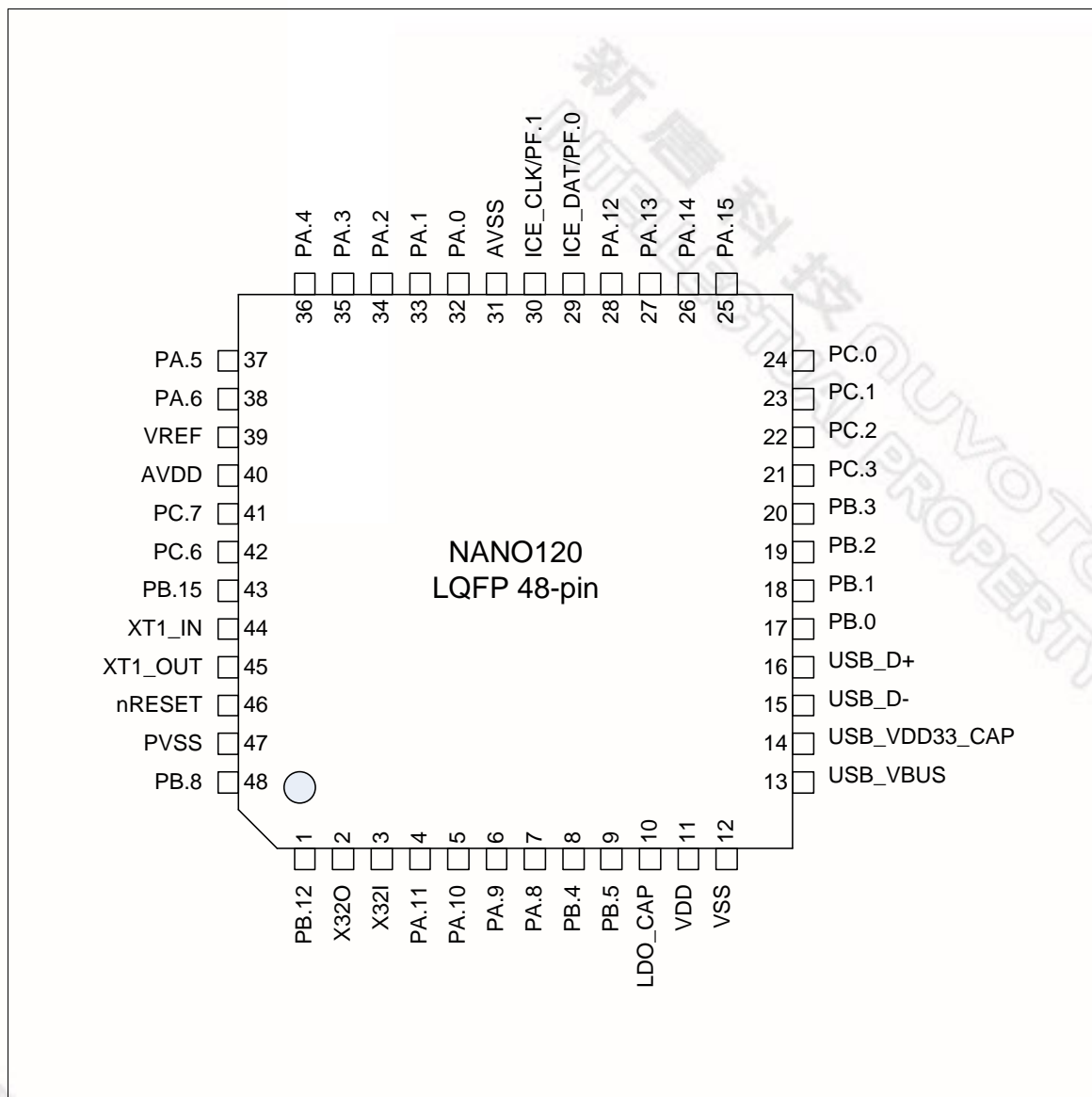


图 3-9 NuMicro™ Nano120 LQFP 48-pin 管脚图

3.3.4.1 NuMicro™ Nano130 LQFP 128-pin

图 3-10 NuMicro™ Nano130 LQFP 128-pin 管脚图

3.3.4.2 NuMicro™ Nano130 LQFP 64-pin

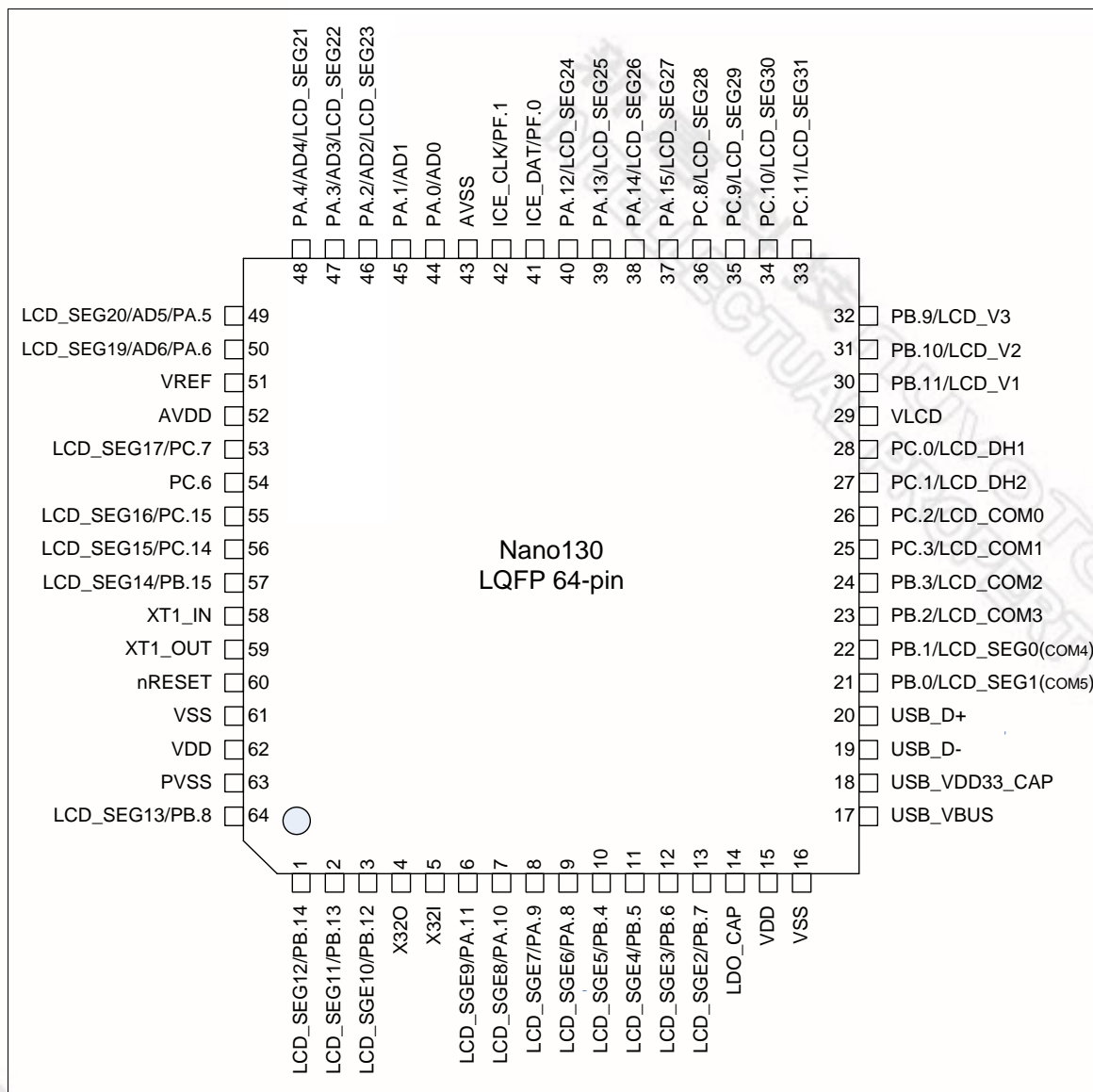


图 3-11 NuMicro™ Nano130 LQFP 64-pin 管脚图

3.4 管脚功能描述

3.4.1 NuMicro™ Nano100 管脚描述

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
1			PE.13	I/O	通用数字输入/输出管脚
2	1		PB.14	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
			SC2_CD	I	SmartCard2 卡侦测管脚
			SPI2_SS1	I/O	SPI2 2 nd 从机选择管脚
3	2		PB.13	I/O	通用数字输入/输出管脚
			EBI_AD1	I/O	EBI 地址/数据总线位1
4	3	1	PB.12	I/O	通用数字输入/输出管脚
			EBI_AD0	I/O	EBI 地址/数据总线位0
			CLKO	O	频率分频输出管脚
5					NC
6	4	2	X32O	O	外部 32.768 kHz 晶振输出管脚
7	5	3	X32I	I	外部32.768 kHz晶振输入管脚
8					NC
9	6	4	PA.11	I/O	通用数字输入/输出管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
			EBI_nRD	O	EBI 读使能输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_MOSI0	I/O	SPI2 1st MOSI (主机输出, 从机输入) 脚
10	7	5	PA.10	I/O	通用数字输入/输出管脚
			I2C1_SDA	I/O	I ² C1 数据输入/输出管脚
			EBI_nWR	O	EBI 写使能输出管脚
			SC0_PWR	O	SmartCard0 电源管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
11	8	6	PA.9	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			SC0_DAT	I/O	SmartCard0 DATA 管脚(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 串行时钟管脚
12	9	7	PA.8	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据输入/输出管脚
			SC0_CLK	O	SmartCard0 时钟管脚(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
13			PD.8	I/O	通用数字输入/输出管脚
14			PD.9	I/O	通用数字输入/输出管脚
15			PD.10	I/O	通用数字输入/输出管脚
16			PD.11	I/O	通用数字输入/输出管脚
17			PD.12	I/O	通用数字输入/输出管脚
18			PD.13	I/O	通用数字输入/输出管脚
19	10	8	PB.4	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SC0_CD	I	SmartCard0 卡检测管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
20	11	9	PB.5	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1 数据发送输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚
21	12		PB.6	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1 请求发送输出管脚
			EBI_ALE	O	EBI 地址锁存使能输出管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
22	13		PB.7	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1 清发送输入管脚
			EBI_nCS	O	EBI 片选使能输出管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
23					NC
24	14	10	LDO_CAP	P	LDO 输出管脚
25					NC
26					NC
27	15	11	VDD	P	电源供应管脚，为IO端口和LDO源提供电源
28					NC
29	16	12	VSS	P	接地
30			VSS	P	接地
31			VSS	P	接地
32			VSS	P	接地
33			PE.12	I/O	通用数字输入/输出管脚
34			PE.11	I/O	通用数字输入/输出管脚
35			PE.10	I/O	通用数字输入/输出管脚
36			PE.9	I/O	通用数字输入/输出管脚
37			PE.8	I/O	通用数字输入/输出管脚
38			PE.7	I/O	通用数字输入/输出管脚
39					NC
40					NC
41					NC
42					NC
43					NC
44	17	13	PB.0	I/O	通用数字输入/输出管脚
			UART0_RXD	I	UART0 数据接收输入管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出，从机输入) 脚
45	18	14	PB.1	I/O	通用数字输入/输出管脚
			UART0_TXD	O	UART0 数据发送输出引脚
			SPI1_MISO0	I/O	SPI1 1 st MISO(主机输入，从机输出) 脚
46	19	15	PB.2	I/O	通用数字输入/输出管脚
			UART0_RTSn	O	UART0 请求发送输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			EBI_nWRL	O	EBI 低字节写使能输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
47	20	16	PB.3	I/O	通用数字输入/输出管脚
			UART0_CTSn	I	UART0 清除发送输入管脚
			EBI_nWRH	O	EBI 高字节写使能输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚
48	21		PD.6	I/O	通用数字输入/输出管脚
49	22		PD.7	I/O	通用数字输入/输出管脚
50	23		PD.14	I/O	通用数字输入/输出管脚
51	24		PD.15	I/O	通用数字输入/输出管脚
52			PC.5	I/O	通用数字输入/输出管脚
			SPI0_MOSI1	I/O	SPI0 2 nd MOSI (主机输出, 从机输入) 脚
53			PC.4	I/O	通用数字输入/输出管脚
			SPI0_MISO1	I/O	SPI0 2 nd MISO (主机输入, 从机输出) 脚
54	25	17	PC.3	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			I2S_DO	O	I ² S 数据输出管脚
			SC1_RST	O	SmartCard1 RST 管脚
55	26	18	PC.2	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			I2S_DI	I	I ² S 数据输入管脚
			SC1_PWR	O	SmartCard1 电源管脚
56	27	19	PC.1	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚
			I2S_BCLK	I/O	I ² S 位时钟管脚
			SC1_DAT	I/O	SmartCard1 数据管脚(SC1_UART_RXD)
57	28	20	PC.0 / MCLKO	I/O	通用数字输入/输出管脚 / 模块时钟输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			I2S_LRCLK	I/O	I ² S 左右通道时钟
			SC1_CLK	O	SmartCard1 时钟管脚(SC1_UART_TXD)
58			PE.6	I/O	通用数字输入/输出管脚
59					NC
60					NC
61	29	21	PE.5	I/O	通用数字输入/输出管脚
			PWM1_CH1	I/O	PWM1 通道1输出管脚
62	30	22	PB.11	I/O	通用数字输入/输出管脚
			PWM1_CH0	I/O	PWM1 通道0输出管脚
			TM3	O	Timer3 外部计数器输入
			SC2_DAT	I/O	SmartCard2 数据管脚(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
63	31	23	PB.10	I/O	通用数字输入/输出管脚
			SPI0_SS1	I/O	SPI0 2 nd 从机选择管脚
			TM2	O	Timer2外部计数器输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
64	32	24	PB.9	I/O	通用数字输入/输出管脚
			SPI1_SS1	I/O	SPI1 2 nd 从机选择管脚
			TM1	O	Timer1 外部计数器输入
			SC2_RST	O	SmartCard2 RST 管脚
			INT0	I	外部中断0输入管脚
65			PE.4	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
66			PE.3	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
67			PE.2	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
68			PE.1	I/O	通用数字输入/输出管脚.
			PWM1_CH3	I/O	PWM1 通道3输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚
69			PE.0	I/O	通用数字输入/输出管脚
			PWM1_CH2	I/O	PWM1 通道2输出管脚
			I2S_MCLK	O	I ² S 主机时钟输出管脚
70			PC.13	I/O	通用数字输入/输出管脚
			SPI1_MOSI1	I/O	SPI1 2 nd MOSI (主机输出, 从机输入) 脚
			PWM1_CH1	O	PWM1 通道1输出管脚
			SNOOPER	I	探听管脚
			INT1	I	外部中断0
			I2C0_SCL	O	I ² C0 时钟管脚
71			PC.12	I/O	通用数字输入/输出管脚
			SPI1_MISO1	I/O	SPI1 2 nd MISO(主机输入, 从机输出) 脚
			PWM1_CH0	O	PWM1 通道0输出管脚
			INT0	I	外部中断0输入管脚
			I2C0_SDA	I/O	I ² C0 数据输入/输出管脚
72	33		PC.11	I/O	通用数字输入/输出管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出, 从机输入) 脚
			UART1_TXD	O	UART1 数据发送输出管脚
73	34		PC.10	I/O	通用数字输入/输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO (主机输入, 从机输出) 脚
			UART1_RXD	I	UART1 数据接收输入管脚
74	35		PC.9	I/O	通用数字输入/输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
75	36		PC.8	I/O	通用数字输入/输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			EBI_MCLK	O	EBI 外部时钟输出管脚
			I2C1_SDA	I/O	I ² C1 数据输入/输出管脚
76	37	25	PA.15	I/O	通用数字输入/输出管脚
			PWM0_CH3	I/O	PWM0 通道3输出管脚
			I2S_MCLK	O	I ² S 主机时钟输出管脚
			TC3	I	Timer3 捕捉输入管脚
			SC0_PWR	O	SmartCard0 电源管脚
			UART0_TXD	O	UART0 数据发送输出管脚
77	38	26	PA.14	I/O	通用数字输入/输出管脚
			PWM0_CH2	I/O	PWM0 通道2输出管脚
			EBI_AD15	I/O	EBI 地址/数据总线位15
			TC2	I	Timer2 捕捉输入管脚
			UART0_RXD	I	UART0 数据接收输入管脚
78	39	27	PA.13	I/O	通用数字输入/输出管脚
			PWM0_CH1	I/O	PWM0通道1输出管脚
			EBI_AD14	I/O	EBI 地址/数据总线位14
			TC1	I	Timer1 捕捉输入管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
79	40	28	PA.12	I/O	通用数字输入/输出管脚
			PWM0_CH0	I/O	PWM0 通道0输出管脚
			EBI_AD13	I/O	EBI 地址/数据总线位13
			TC0	I	Timer0 捕捉输入
			I2C0_SDA	I/O	I ² C0 数据输入/输出管脚
80	41	29	ICE_DAT	I/O	调试器的串行数据管脚
			PF.0	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0输入管脚
81	42	30	ICE_CLK	I	调试器的串行时钟管脚
			PF.1	I/O	通用数字输入/输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			CLKO	O	分频器输出管脚
			INT1	I	外部中断1输入管脚
82					NC
83			VDD	P	电源供应管脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
84					NC
85			VSS	P	地
86			VSS	P	地
87	43	31	AVSS	AP	模拟地
88			AVSS	AP	模拟地
89	44	32	PA.0	I/O	通用数字输入/输出管脚
			AD0	AI	ADC模拟输入0
			SC2_CD	I	SmartCard2 卡侦测
90	45	33	PA.1	I/O	通用数字输入/输出管脚
			AD1	AI	ADC 模拟输入1
			EBI_AD12	I/O	EBI 地址/数据 总线位12
91	46	34	PA.2	I/O	通用数字输入/输出管脚
			AD2	AI	ADC 模拟输入2
			EBI_AD11	I/O	EBI 地址数/据总线位11
			UART1_RXD	I	UART1 数据接收输入管脚
92	47	35	PA.3	I/O	通用数字输入/输出管脚
			AD3	AI	ADC 模拟输入3
			EBI_AD10	I/O	EBI 地址/数据总线位10
			UART1_TXD	O	UART1 数据发送输出管脚
93	48	36	PA.4	I/O	通用数字输入/输出管脚
			AD4	AI	ADC 模拟输入4
			EBI_AD9	I/O	EBI 地址/数据总线位9
			SC2_PWR	O	SmartCard2电源管脚
			I2C0_SDA	I/O	I ² C0数据输入/输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
94	49	37	PA.5	I/O	通用数字输入/输出管脚
			AD5	AI	ADC模拟输入5
			EBI_AD8	I/O	EBI 地址/数据总线位8
			SC2_RST	O	SmartCard2 RST 管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
95	50	38	PA.6	I/O	通用数字输入/输出管脚
			AD6	AI	ADC 模拟输入6
			EBI_AD7	I/O	EBI 地址/数据总线位7
			TC3	I	Timer3 捕捉输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			PWM0_CH3	O	PWM0 通道3输出
96			PA.7	I/O	通用数字输入/输出管脚
			AD7	AI	ADC 模拟输入7
			EBI_AD6	I/O	EBI 地址/数据总线位6
			TC2	I	Timer2 捕捉输入
			SC2_DAT	I/O	SmartCard2 数据管脚(SC2_UART_RXD)
			PWM0_CH2	O	PWM0 通道2输出
97	51	39	VREF	AP	ADC参考电压输入管脚
98					NC
99	52	40	AVDD	AP	P内部模拟电路电源
100			PD.0	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1数据接收输入管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			SC1_CLK	O	SmartCard1时钟管脚(SC1_UART_TXD)
			AD8	AI	ADC 模拟输入8
101			PD.1	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1 数据发送输出管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			SC1_DAT	I/O	SmartCard1 数据管脚.(SC1_UART_RXD)
			AD9	AI	ADC 模拟输入9
102			PD.2	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1 请求发送输出管脚
			I2S_LRCLK	I/O	I ² S 左右声道时钟
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			SC1_PWR	O	SmartCard1 电源管脚
			AD10	AI	ADC 模拟输入10
103			PD.3	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1 清发送输入管脚
			I2S_BCLK	I/O	I ² S 位时钟管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			SC1_RST	O	SmartCard1 RST 管脚
			AD11	AI	ADC 模拟输入11
104					NC
105			PD.4	I/O	通用数字输入/输出管脚
			I2S_DI	I	I ² S 数据输入
			SPI2_MISO1	I/O	SPI2 2 nd MISO (主机输入, 从机输出) 脚
			SC1_CD	I	SmartCard1 卡检测
106			PD.5	I/O	通用数字输入/输出管脚
			I2S_DO	O	I ² S 数据输出
			SPI2_MOSI1	I/O	SPI2 2 nd MOSI (主机输出, 从机输入) 脚
107	53	41	PC.7	I/O	通用数字输入/输出管脚
			DA1_OUT	AO	DAC1 输出
			EBI_AD5	I/O	EBI 地址/数据 总线 bit5
			TC1	I	Timer1 捕捉输入
			PWM0_CH1	O	PWM1 通道1 输出
108	54	42	PC.6	I/O	通用数字输入/输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
			DA0_OUT	I	DAC0 输出
			EBI_AD4	I/O	EBI 地址/数据 总线bit4
			TC0	I	Timer 0 捕捉输入
			SC1_CD	I	SmartCard1 卡检测管脚
			PWM0_CH0	O	PWM0 通道0 输出
109	55		PC.15	I/O	通用数字输入/输出管脚
			EBI_AD3	I/O	EBI 地址/数据 总线 bit3
			TC0	I	Timer0 捕捉输入
			PWM1_CH2	O	PWM1 通道1 输出
110	56		PC.14	I/O	通用数字输入/输出管脚
			EBI_AD2	I/O	EBI 地址/数据 总线 bit2
			PWM1_CH3	I/O	PWM1 通道3 输出
111	57	43	PB.15	I/O	通用数字输入/输出管脚
			INT1	I	外部中断1输入管脚
			SNOOPER	I	探听管脚
			SC1_CD	I	SmartCard1 卡侦测管脚
112					NC
113	58	44	XT1_IN	O	外部 4~24 MHz晶振输出管脚
			PF.3	I/O	通用数字输入/输出管脚
114	59	45	XT1_OUT	I	外部 4~24 MHz晶振输入管脚
			PF.2	I/O	通用数字输入/输出管脚
115					NC
116	60	46	nRESET	I	外部复位输入：低有效，置低复位MCU为初始状态，带内部上拉。
117	61		VSS	P	地
118			VSS	P	地
119					NC
120	62		VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP/QFN 48			
121					NC
122			PF.4	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出管脚
123			PF.5	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
124			VSS	P	地
125	63	47	PVSS	P	PLL 地
126	64	48	PB.8	I/O	通用数字输入/输出管脚
			STADC	I	ADC 外部触发输入
			TM0	I	Timer0 外部计数输入
			INT0	I	外部中断0输入管脚
			SC2_PWR	O	SmartCard2 电源输入
127			PE.15	I/O	通用数字输入/输出管脚
128			PE.14	I/O	通用数字输入/输出管脚

注:

1. 管脚类型 I = 数字输入 (Digital Input), O = 数字输出 (Digital Output); AI= 模拟输入 (Analog Input); AO= 模拟输出 (Analog Output); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)

3.4.1 NuMicro™ Nano110 管脚描述

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
1			PE.13	I/O	通用数字输入/输出管脚
			LCD_SEG27	O	LCD SEG27 at LQFP128
2	1		PB.14	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
			SC2_CD	I	SmartCard2卡侦测
			SPI2_SS1	I/O	SPI2 2 nd 从机选择管脚
			LCD_SEG12	O	LCD SEG12 at LQFP64
			LCD_SEG26	O	LCD SEG26 at LQFP128
3	2		PB.13	I/O	通用数字输入/输出管脚
			EBI_AD1	I/O	EBI地址/数据总线 bit1
			LCD_SEG11	O	LCD SEG11 at LQFP64
			LCD_SEG25	O	LCD SEG 25 at LQFP128
4	3		PB.12	I/O	通用数字输入/输出管脚
			EBI_AD0	I/O	EBI 地址数据总线bit0
			CLKO	O	分频器输出管脚
			LCD_SEG10	O	LCD SEG10 at LQFP64
			LCD_SEG24	O	LCD SEG24 at LQFP128
5					NC
6	4		X32O	O	外部 32.768 kHz 晶振输出管脚
7	5		X32I	I	外部 32.768 kHz 晶振输入管脚
8					NC
9	6		PA.11	I/O	通用数字输入/输出管脚
			I2C1_SCL	I/O	I ² C1时钟管脚
			EBI_nRD	O	EBI 读使能输出管脚
			SC0_RST	O	SmartCard0 RST管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			LCD_SEG9	O	LCD SEG 9 at LQFP64
			LCD_SEG23	O	LCD SEG 23 at LQFP128

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
10	7		PA.10	I/O	通用数字输入/输出管脚
			I2C1_SDA	I/O	I ² C1 数据 输入/输出管脚
			EBI_nWR	O	EBI 写使能输出管脚
			SC0_PWR	O	SmartCard0 电源管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			LCD_SEG8	O	LCD SEG8 at LQFP64
			LCD_SEG22	O	LCD SEG22 at LQFP128
11	8		PA.9	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
			SC0_DAT	I/O	SmartCard0 DATA 管脚(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 串行时钟管脚
			LCD_SEG7	O	LCD SEG7 at LQFP64
			LCD_SEG21	O	LCD SEG21 at LQFP128
12	9		PA.8	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据输入/输出管脚
			SC0_CLK	O	SmartCard0 时钟管脚(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			LCD_SEG6	O	LCD SEG6 at LQFP64
			LCD_SEG20	O	LCD SEG20 at LQFP128
13			PD.8	I/O	通用数字输入/输出管脚
			LCD_SEG19	O	LCD SEG 19 at LQFP128
14			PD.9	I/O	通用数字输入/输出管脚
			LCD_SEG18	O	LCD SEG 18 at LQFP128
15			PD.10	I/O	通用数字输入/输出管脚
			LCD_SEG17	O	LCD SEG 17 at LQFP128
16			PD.11	I/O	通用数字输入/输出管脚
			LCD_SEG16	O	LCD SEG 16 at LQFP128
17			PD.12	I/O	通用数字输入/输出管脚
			LCD_SEG15	O	LCD SEG 15 at LQFP128

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
18			PD.13	I/O	通用数字输入/输出管脚
			LCD_SEG14	O	LCD SEG 14 at LQFP128
19	10		PB.4	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SC0_CD	I	SmartCard0 卡检测管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			LCD_SEG5	O	LCD SEG 5 at LQFP64
			LCD_SEG13	O	LCD SEG 13 at LQFP128
20	11		PB.5	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1数据发送输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_CLK	I/O	SPI2串行时钟管脚
			LCD_SEG4	O	LCD SEG 4 at LQFP64
			LCD_SEG12	O	LCD SEG 12 at LQFP128
21	12		PB.6	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1 请求发送输出管脚
			EBI_ALE	O	EBI 地址锁存使能输出管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			LCD_SEG3	O	LCD SEG 3 at LQFP64
			LCD_SEG11	O	LCD SEG 11 at LQFP128
22	13		PB.7	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1 清发送输入管脚
			EBI_nCS	O	EBI 芯片选择使能输出管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			LCD_SEG2	O	LCD SEG 2 at LQFP64
			LCD_SEG10	O	LCD SEG 10 at LQFP128
23					NC
24	14		LDO_CAP	P	LDO 输出管脚
25					NC

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
26					NC
27	15		VDD	P	电源供应管脚，为IO端口和LDO源提供电源
28					NC
29	16		VSS	P	地
30			VSS	P	地
31			VSS	P	地
32			VSS	P	地
33			PE.12	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1清除发送输入管脚
34			PE.11	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1请求发送输出管脚
35			PE.10	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1数据发送输出管脚
36			PE.9	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
37			PE.8	I/O	通用数字输入/输出管脚
			LCD_SEG9	O	LCD SEG 9 at LQFP128
38			PE.7	I/O	通用数字输入/输出管脚
			LCD_SEG8	O	LCD SEG 8 at LQFP128
39					NC
40					NC
41					NC
42					NC
43					NC
44	17		PB.0	I/O	通用数字输入/输出管脚
			UART0_RXD	I	UART0 数据接收输入管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出，从机输入) 脚
			LCD_SEG1	O	LCD SEG1 at LQFP64
			LCD_SEG7	O	LCD SEG7 at LQFP128

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
45	18		PB.1	I/O	通用数字输入/输出管脚
			UART0_TXD	O	UART0 数据发送输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO(主机输入, 从机输出) 脚
			LCD_SEG0	O	LCD SEG 0 at LQFP64
			LCD_SEG6	O	LCD SEG 6 at LQFP128
46	19		PB.2	I/O	通用数字输入/输出管脚
			UART0_RTSn	O	UART0 请求发送输出管脚
			EBI_nWRL	O	EBI 低字节写使能输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			LCD_COM3	O	LCD COM3 at LQFP64
			LCD_SEG5	O	LCD SEG5 at LQFP128
47	20		PB.3	I/O	通用数字输入/输出管脚
			UART0_CTSn	I	UART0 清除发送输入管脚
			EBI_nWRH	O	EBI 高字节写使能输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚
			LCD_COM2	O	LCD COM2 at LQFP64
			LCD_SEG4	O	LCD SEG4 at LQFP128
48	21		PD.6	I/O	通用数字输入/输出管脚
			LCD_SEG3	O	LCD SEG 3 at LQFP128
49	22		PD.7	I/O	通用数字输入/输出管脚
			LCD_SEG2	O	LCD SEG 2 at LQFP128
50	23		PD.14	I/O	通用数字输入/输出管脚
			LCD_SEG1	O	LCD SEG 1 at LQFP128 (或作为LCD_COM5)
51	24		PD.15	I/O	通用数字输入/输出管脚
			LCD_SEG0	O	LCD SEG 0 at LQFP128 (或作为LCD_COM4)
52			PC.5	I/O	通用数字输入/输出管脚
			SPI0_MOSI1	I/O	SPI0 2 nd MOSI (主机输出, 从机输入) 脚
			LCD_COM3	O	LCD COM3 at LQFP128
53			PC.4	I/O	通用数字输入/输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SPI0_MISO1	I/O	SPI0 2 nd MISO (主机输入, 从机输出) 脚
			LCD_COM2	O	LCD COM2 at LQFP128
54	25		PC.3	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			I2S_DO	O	I ² S 数据输出
			SC1_RST	O	SmartCard1 RST 管脚
			LCD_COM1	O	LCD COM1 at LQFP64
			LCD_COM1	O	LCD COM1 at LQFP128
55	26		PC.2	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			I2S_DI	I	I ² S 数据输入
			SC1_PWR	O	SmartCard1 PWR pin
			LCD_COM0	O	LCD COM 0 at LQFP64
			LCD_COM0	O	LCD COM 0 at LQFP128
56	27		PC.1	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚
			I2S_BCLK	I/O	I ² S bit 时钟管脚
			SC1_DAT	I/O	SmartCard1 DATA 管脚(SC1_UART_RXD)
			LCD_DH2	O	LCD DH2 at LQFP64
			LCD_DH2	O	LCD DH2 at LQFP128
57	28		PC.0 / MCLKO	I/O	通用数字输入/输出管脚 / 模块时钟输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚
			I2S_LRCLK	I/O	I2S 左右声道时钟
			SC1_CLK	O	SmartCard1 时钟管脚(SC1_UART_TXD)
			LCD_DH1	O	LCD DH1 at LQFP64
			LCD_DH1	O	LCD DH1 at LQFP100
58			PE.6	I/O	通用数字输入/输出管脚
59	29		LCD_VLCD	AO	LCD VLCD 管脚
60					NC

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
61			PE.5	I/O	通用数字输入/输出管脚
62	30		PB.11	I/O	通用数字输入/输出管脚
			PWM1_CH0	I/O	PWM1通道0 输出
			TM3	O	Timer3外部计数输入
			SC2_DAT	I/O	SmartCard2 DATA 管脚(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			LCD_V1	O	LCD V1 at LQFP64
			LCD_V1	O	LCD V1 at LQFP128
63	31		PB.10	I/O	通用数字输入/输出管脚
			SPI0_SS1	I/O	SPI0 2 nd 从机选择管脚
			TM2	O	Timer2外部计数输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			LCD_V2	O	LCD V2 at LQFP64
			LCD_V2	O	LCD V2 at LQFP128
64	32		PB.9	I/O	通用数字输入/输出管脚
			SPI1_SS1	I/O	SPI1 2 nd 从机选择管脚
			TM1	O	Timer1外部计数输入
			SC2_RST	O	SmartCard2 RST 管脚
			INT0	I	外部中断0 输入管脚
			LCD_V3	O	LCD V3 at LQFP64
			LCD_V3	O	LCD V3 at LQFP128
65			PE.4	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
66			PE.3	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
67			PE.2	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
68			PE.1	I/O	通用数字输入/输出管脚
			PWM1_CH3	I/O	PWM1 通道3输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚
69			PE.0	I/O	通用数字输入/输出管脚
			PWM1_CH2	I/O	PWM1通道2 输出
			I2S_MCLK	O	I ² S主机时钟输出管脚
70			PC.13	I/O	通用数字输入/输出管脚
			SPI1_MOSI1	I/O	SPI1 2 nd MOSI (主机输出, 从机输入) 脚
			PWM1_CH1	O	PWM1 通道1 输出
			SNOOPER	I	监听管脚
			INT1	I	外部中断1
			I2C0_SCL	O	I ² C0 时钟管脚
71			PC.12	I/O	通用数字输入/输出管脚
			SPI1_MISO1	I/O	SPI1 2 nd MISO(主机输入, 从机输出) 脚
			PWM1_CH0	O	PWM1 通道0 输出
			INT0	I	外部中断0 输入管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
72	33		PC.11	I/O	通用数字输入/输出管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出, 从机输入) 脚
			UART1_TXD	O	UART1 数据发送输出管脚
			LCD_SEG31	O	LCD SEG31 at LQFP64
73	34		PC.10	I/O	通用数字输入/输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO(主机输入, 从机输出) 脚
			UART1_RXD	I	UART1 数据接收输入管脚
			LCD_SEG30	O	LCD SEG30 at LQFP64
74	35		PC.9	I/O	通用数字输入/输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			LCD_SEG29	O	LCD SEG29 at LQFP64
75	36		PC.8	I/O	通用数字输入/输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚
			EBI_MCLK	O	EBI 外部时钟输出管脚
			I2C1_SDA	I/O	I ² C1 数据 输入/输出管脚
			LCD_SEG28	O	LCD SEG28 at LQFP64
76	37		PA.15	I/O	通用数字输入/输出管脚
			PWM0_CH3	I/O	PWM0 通道3 输出
			I2S_MCLK	O	I ² S 主机时钟输出管脚
			TC3	I	Timer3 捕捉输入
			SC0_PWR	O	SmartCard0 电源管脚
			UART0_TXD	O	UART0数据发送输出管脚
			LCD_SEG27	O	LCD SEG27 at LQFP64
77	38		PA.14	I/O	通用数字输入/输出管脚
			PWM0_CH2	I/O	PWM0通道2 输出
			EBI_AD15	I/O	EBI 地址/数据 总线 bit15
			TC2	I	Timer 2 捕捉输入
			UART0_RXD	I	UART0 数据接收输入管脚
			LCD_SEG26	O	LCD SEG26 at LQFP64
78	39		PA.13	I/O	通用数字输入/输出管脚
			PWM0_CH1	I/O	PWM0 通道1 输出
			EBI_AD14	I/O	EBI 地址/数据 总线 bit14
			TC1	I	Timer1 捕捉输入
			I2C0_SCL	I/O	I ² C0 时钟管脚
			LCD_SEG25	O	LCD SEG25 at LQFP64
79	40		PA.12	I/O	通用数字输入/输出管脚
			PWM0_CH0	I/O	PWM0 通道0 输出
			EBI_AD13	I/O	EBI 地址/数据 总线 bit13

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			TC0	I	Timer 0 捕捉输入
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
			LCD_SEG24	O	LCD SEG24 at LQFP64
80	41		ICE_DAT	I/O	调试器的串行数据管脚
			PF.0	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
81	42		ICE_CLK	I	调试器的串行时钟管脚
			PF.1	I/O	通用数字输入/输出管脚
			CLKO	O	分频器输出管脚
			INT1	I	外部中断1 输入管脚
82					NC
83			VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
84					NC
85			VSS	P	地
86			VSS	P	地
87	43		AVSS	AP	模拟电路地
88			AVSS	AP	模拟电路地
89	44		PA.0	I/O	通用数字输入/输出管脚
			AD0	AI	ADC模拟输入0
			SC2_CD	I	SmartCard2 卡侦测
90	45		PA.1	I/O	通用数字输入/输出管脚
			AD1	AI	ADC 模拟输入1
			EBI_AD12	I/O	EBI 地址/数据 总线 bit12
91	46		PA.2	I/O	通用数字输入/输出管脚
			AD2	AI	ADC 模拟输入2
			EBI_AD11	I/O	EBI 地址/数据 总线 bit11
			UART1_RXD	I	UART1 数据接收输入管脚
			LCD_SEG23*	AO	LCD SEG23 at LQFP64

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
92	47		PA.3	I/O	通用数字输入/输出管脚
			AD3	AI	ADC 模拟输入3
			EBI_AD10	I/O	EBI 地址/数据 总线 bit10
			UART1_TXD	O	UART1 数据发送输出管脚
			LCD_SEG22*	AO	LCD SEG22 at LQFP64
93	48		PA.4	I/O	通用数字输入/输出管脚
			AD4	AI	ADC 模拟输入4
			EBI_AD9	I/O	EBI 地址/数据 总线 bit9
			SC2_PWR	O	SmartCard2 电源管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
			LCD_SEG21	AO	LCD SEG21 at LQFP64
			LCD_SEG39*	AO	LCD SEG39 at LQFP128
94	49		PA.5	I/O	通用数字输入/输出管脚
			AD5	AI	ADC 模拟输入5
			EBI_AD8	I/O	EBI地址/数据 总线 bit8
			SC2_RST	O	SmartCard2 RST 管脚
			I2C0_SCL	I/O	I ² C0时钟管脚
			LCD_SEG20*	AO	LCD SEG 19 at LQFP64
			LCD_SEG38*	AO	LCD SEG 37 at LQFP128
95	50		PA.6	I/O	通用数字输入/输出管脚
			AD6	AI	ADC模拟输入6
			EBI_AD7	I/O	EBI 地址/数据 总线 bit7
			TC3	I	Timer3 捕捉输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			PWM0_CH3	O	PWM0通道3 输出
			LCD_SEG19*	AO	LCD SEG 19 at LQFP64
			LCD_SEG37*	AO	LCD SEG 37 at LQFP128
96			PA.7	I/O	通用数字输入/输出管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			AD7	AI	ADC 模拟输入7
			EBI_AD6	I/O	EBI 地址/数据 总线 bit6
			TC2	I	Timer2 捕捉输入
			SC2_DAT	I/O	SmartCard2 DATA 管脚(SC2_UART_RXD)
			PWM0_CH2	O	PWM0 通道2 输出
			LCD_SEG36*	AO	LCD SEG36 at LQFP128
97	51		VREF	AP	ADC 参考电压输入
98					NC
99	52		AVDD	AP	内部模拟电路电源
100			PD.0	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			SC1_CLK	O	SmartCard1 时钟管脚(SC1_UART_TXD)
			AD8	AI	ADC 模拟输入8
101			PD.1	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1数据发送输出管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚
			SC1_DAT	I/O	SmartCard1 DATA 管脚(SC1_UART_RXD)
			AD9	AI	ADC 模拟输入9
102			PD.2	I/O	通用数字输入/输出管脚
			UART1_RTSn		UART1请求发送输出管脚
			I2S_LRCLK	I/O	I2S 左右声道时钟
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			SC1_PWR	O	SmartCard1 电源管脚
			AD10	AI	ADC 模拟输入10
103			PD.3	I/O	通用数字输入/输出管脚
			UART1_CTSn		UART1 清除发送输入管脚
			I2S_BCLK	I/O	I2S bit 时钟管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			SC1_RST	O	SmartCard1 RST 管脚
			AD11	AI	ADC模拟输入11
104					NC
105			PD.4	I/O	通用数字输入/输出管脚
			I2S_DI	I	I ² S 数据输入
			SPI2_MISO1	I/O	SPI2 2 nd MISO (主机输入, 从机输出) 脚
			SC1_CD	I	SmartCard1 卡检测
			LCD_SEG35	AO	LCD SEG35 at LQFP128
106			PD.5	I/O	通用数字输入/输出管脚
			I2S_DO	O	I ² S 数据输出
			SPI2_MOSI1	I/O	SPI2 2 nd MOSI (主机输出, 从机输入) 脚
			LCD_SEG34	AO	LCD SEG 34 at LQFP128
107	53		PC.7	I/O	通用数字输入/输出管脚
			DA1_OUT	AO	DAC 1 输出
			EBI_AD5	I/O	EBI地址/数据 总线bit5
			TC1	I	Timer1捕捉输入
			PWM0_CH1	O	PWM1通道1 输出
			LCD_SEG17*	AO	LCD SEG 17 at LQFP64
108	54		PC.6	I/O	通用数字输入/输出管脚
			DA0_OUT	I	DAC0输出
			EBI_AD4	I/O	EBI 地址/数据 总线 bit4
			TC0	I	Timer 0 捕捉输入
			SC1_CD	I	SmartCard1 卡检测管脚
			PWM0_CH0	O	PWM0 通道0 输出
109	55		PC.15	I/O	通用数字输入/输出管脚
			EBI_AD3	I/O	EBI 地址/数据 总线 bit3
			TC0	I	Timer0 捕捉输入

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			PWM1_CH2	O	PWM1 通道1 输出
			LCD_SEG16	AO	LCD SEG16 at LQFP64
			LCD_SEG33	AO	LCD SEG33 at LQFP128
110	56		PC.14	I/O	通用数字输入/输出管脚
			EBI_AD2	I/O	EBI 地址/数据 总线 bit2
			PWM1_CH3	I/O	PWM1 通道3 输出
			LCD_SEG15	AO	LCD SEG15 at LQFP64
			LCD_SEG32	AO	LCD SEG32 at LQFP128
111	57		PB.15	I/O	通用数字输入/输出管脚
			INT1	I	外部中断1 输入管脚
			SNOOPER	I	探听管脚
			LCD_SEG14	AO	LCD SEG14 at LQFP64
			LCD_SEG31	AO	LCD SEG31 at LQFP128
112					NC
113	58	44	XT1_IN	O	外部 4~24 MHz晶振输出管脚
			PF.3	I/O	通用数字输入/输出管脚
114	59	45	XT1_OUT	I	外部 4~24 MHz晶振输入管脚
			PF.2	I/O	通用数字输入/输出管脚
115					NC
116	60		nRESET	I	外部复位输入：低有效，置低复位MCU为初始状态，带内部上拉
117	61		VSS	P	地
118			VSS	P	地
119					NC
120	62		VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
121					NC
122			PF.4	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚

管脚号			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
123			PF.5	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
124			VSS	P	地
125	63		PVSS	P	PLL 地
126	64		PB.8	I/O	通用数字输入/输出管脚
			STADC	I	ADC 外部触发输入
			TM0	I	Timer0 外部计数输入
			INT0	I	外部中断0 输入管脚
			SC2_PWR	O	SmartCard2 电源管脚
			LCD_SEG13	AO	LCD SEG 13 at LQFP64
			LCD_SEG30	AO	LCD SEG 30 at LQFP128
127			PE.15	I/O	通用数字输入/输出管脚
			LCD_SEG29	O	LCD SEG 29 at LQFP128
128			PE.14	I/O	通用数字输入/输出管脚
			LCD_SEG28	O	LCD SEG 28 at LQFP128

注:

- 管脚类型 I = 数字输入 (Digital Input), O = 数字输出 (Digital Output); AI= 模拟输入 (Analog Input); AO= 模拟输出 (Analog Output); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)
- *: ADC与LCD共享的管脚其输出电压不可高于 VDD

3.4.2 NuMicro™ Nano120 管脚描述

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
1			PE.13	I/O	通用数字输入/输出管脚
2	1		PB.14	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0输入管脚
			SC2_CD	I	SmartCard2 卡侦测管脚
			SPI2_SS1	I/O	SPI2 2 nd 从机选择管脚
3	2		PB.13	I/O	通用数字输入/输出管脚
			EBI_AD1	I/O	EBI 地址/数据 总线 bit1
4	3	1	PB.12	I/O	通用数字输入/输出管脚
			EBI_AD0	I/O	EBI 地址/数据 总线 bit0
			CLKO	O	分频器输出管脚
5					NC
6	4	2	X32O	O	外部 32.768 kHz 晶振输出管脚
7	5	3	X32I	I	外部 32.768 kHz 晶振输入管脚
8					NC
9	6	4	PA.11	I/O	通用数字输入/输出管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
			EBI_nRD	O	EBI 读使能输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
10	7	5	PA.10	I/O	通用数字输入/输出管脚
			I2C1_SDA	I/O	I2C1 数据 输入/输出 管脚
			EBI_nWR	O	EBI 写使能输出管脚
			SC0_PWR	O	SmartCard0 电源管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
11	8	6	PA.9	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
			SC0_DAT	I/O	SmartCard0 DATA 管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SPI2_CLK	I/O	SPI2 串行时钟管脚(SC0_UART_RXD)
12	9	7	PA.8	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出管脚
			SC0_CLK	O	SmartCard0 时钟管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
13			PD.8	I/O	通用数字输入/输出管脚
14			PD.9	I/O	通用数字输入/输出管脚
15			PD.10	I/O	通用数字输入/输出管脚
16			PD.11	I/O	通用数字输入/输出管脚
17			PD.12	I/O	通用数字输入/输出管脚
18			PD.13	I/O	通用数字输入/输出管脚
19	10	8	PB.4	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SC0_CD	I	SmartCard0 卡检测管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
20	11	9	PB.5	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1数据发送输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_CLK	I/O	SPI2串行时钟管脚
21	12		PB.6	I/O	通用数字输入/输出管脚
			UART1_nRTS	O	UART1 请求发送输出管脚
			EBI_ALE	O	EBI 地址锁存使能输出管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
22	13		PB.7	I/O	通用数字输入/输出管脚
			UART1_nCTS	I	UART1 清除发送输入管脚
			EBI_nCS	O	EBI 芯片选择使能输出管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
23					NC

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
24	14	10	LDO_CAP	P	LDO输出管脚
25					NC
26					NC
27	15	11	VDD	P	电源供应管脚，为IO端口和LDO源提供电源
28					NC
29	16	12	VSS	P	地
30			VSS	P	地
31			VSS	P	地
32			VSS	P	地
33			PE.12	I/O	通用数字输入/输出管脚
34			PE.11	I/O	通用数字输入/输出管脚
35			PE.10	I/O	通用数字输入/输出管脚
36			PE.9	I/O	通用数字输入/输出管脚
37			PE.8	I/O	通用数字输入/输出管脚
38			PE.7	I/O	通用数字输入/输出管脚
39					NC
40	17	13	USB_VBUS	USB	电源：来自 USB Host 或 HUB.
41	18	14	USB_VDD33_CAP	USB	内部电源调节器 3.3V 去耦管脚
42	19	15	USB_D-	USB	USB 差分信号D-
43	20	16	USB_D+	USB	USB 差分信号D+
44	21	17	PB.0	I/O	通用数字输入/输出管脚
			UART0_RXD	I	UART0 数据接收输入管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出, 从机输入) 脚
45	22	18	PB.1	I/O	通用数字输入/输出管脚
			UART0_TXD	O	UART0 数据发送输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO (主机输入, 从机输出) 脚
46	23	19	PB.2	I/O	通用数字输入/输出管脚
			UART0_nRTS	O	UART0 请求发送输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			EBI_nWRL	O	EBI 低字节写使能输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
47	24	20	PB.3	I/O	通用数字输入/输出管脚
			UART0_nCTS	I	UART0 请求发送输入管脚
			EBI_nWRH	O	EBI 高字节写使能输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚
48			PD.6	I/O	通用数字输入/输出管脚
49			PD.7	I/O	通用数字输入/输出管脚
50			PD.14	I/O	通用数字输入/输出管脚
51			PD.15	I/O	通用数字输入/输出管脚
52			PC.5	I/O	通用数字输入/输出管脚
			SPI0_MOSI1	I/O	SPI0 2 nd MOSI (主机输出, 从机输入) 脚
53			PC.4	I/O	通用数字输入/输出管脚
			SPI0_MISO1	I/O	SPI0 2 nd MISO (主机输入, 从机输出) 脚
54	25	21	PC.3	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			I2S_DO	O	I ² S 数据输出
			SC1_RST	O	SmartCard1 RST 管脚
55	26	22	PC.2	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			I2S_DI	I	I ² S 数据输入
			SC1_PWR	O	SmartCard1 PWR 管脚
56	27	23	PC.1	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚
			I2S_BCLK	I/O	I2S bit 时钟管脚
			SC1_DAT	I/O	SmartCard1 DATA 管脚
57	28	24	PC.0 / MCLKO	I/O	通用数字输入/输出管脚 / 模块时钟输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			I2S_LRCLK	I/O	I ² S 左右声道时钟
			SC1_CLK	O	SmartCard1 时钟管脚
58			PE.6	I/O	通用数字输入/输出管脚
59					NC
60					NC
61	29		PE.5	I/O	通用数字输入/输出管脚
			PWM1_CH1	I/O	PWM1通道1 输出
62	30		PB.11	I/O	通用数字输入/输出管脚
			PWM1_CH0	I/O	PWM1通道0 输出
			TM3	O	Timer3外部计数输入
			SC2_DAT	I/O	SmartCard2 DATA 管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
63	31		PB.10	I/O	通用数字输入/输出管脚
			SPI0_SS1	I/O	SPI0 2 nd 从机选择管脚
			TM2	O	Timer2外部计数输入
			SC2_CLK	O	SmartCard2 时钟管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
64	32		PB.9	I/O	通用数字输入/输出管脚
			SPI1_SS1	I/O	SPI1 2 nd 从机选择管脚
			TM1	O	Timer1 外部计数输入
			SC2_RST	O	SmartCard2 RST 管脚
			INT0	I	外部中断0 输入管脚
65			PE.4	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
66			PE.3	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
67			PE.2	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
68			PE.1	I/O	通用数字输入/输出管脚
			PWM1_CH3	I/O	PWM1通道3 输出
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚
69			PE.0	I/O	通用数字输入/输出管脚
			PWM1_CH2	I/O	PWM1通道2 输出
			I2S_MCLK	O	I ² S主机时钟输出管脚
70			PC.13	I/O	通用数字输入/输出管脚
			SPI1_MOSI1	I/O	SPI1 2 nd MOSI (主机输出, 从机输入) 脚
			PWM1_CH1	O	PWM1通道1 输出
			SNOOPER	I	探听管脚
			INT1	I	外部中断0 输入管脚
			I2C0_SCL	O	I ² C0时钟管脚
71			PC.12	I/O	通用数字输入/输出管脚
			SPI1_MISO1	I/O	SPI1 2 nd MISO (主机输入, 从机输出) 脚
			PWM1_CH0	O	PWM1通道0 输出
			INT0	I	外部中断0 输入管脚
			I2C0_SDA	I/O	I ² C0数据 输入/输出 管脚
72	33		PC.11	I/O	通用数字输入/输出管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出, 从机输入) 脚
			UART1_TXD	O	UART1 数据发送输出管脚
73	34		PC.10	I/O	通用数字输入/输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO (主机输入, 从机输出) 脚
			UART1_RXD	I	UART1 数据接收输入管脚
74	35		PC.9	I/O	通用数字输入/输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
75	36		PC.8	I/O	通用数字输入/输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			EBI_MCLK	O	EBI 外部时钟输出管脚
			I2C1_SDA	I/O	I2C1 数据 输入/输出 管脚
76	37	25	PA.15	I/O	通用数字输入/输出管脚
			PWM0_CH3	I/O	PWM0 通道3输出
			I2S_MCLK	O	I ² S 主机时钟输出管脚
			TC3	I	Timer3 捕捉输入
			SC0_PWR	O	SmartCard0 电源管脚
			UART0_TXD	O	UART0数据发送输出管脚
77	38	26	PA.14	I/O	通用数字输入/输出管脚
			PWM0_CH2	I/O	PWM0 通道2 输出
			EBI_AD15	I/O	EBI 地址/数据 总线 bit15
			TC2	I	Timer2 捕捉输入
			UART0_RXD	I	UART0 数据接收输入管脚
78	39	27	PA.13	I/O	通用数字输入/输出管脚
			PWM0_CH1	I/O	PWM0通道1输出
			EBI_AD14	I/O	EBI 地址/数据 总线 bit14
			TC1	I	Timer1 捕捉输入
			I2C0_SCL	I/O	I ² C0 时钟管脚
79	40	28	PA.12	I/O	通用数字输入/输出管脚
			PWM0_CH0	I/O	PWM0 通道0 输出
			EBI_AD13	I/O	EBI 地址/数据 总线 bit13
			TC0	I	Timer0 捕捉输入
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
80	41	29	ICE_DAT	I/O	调试器的串行数据管脚
			PF.0	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
81	42	30	ICE_CLK	I	调试器的串行时钟管脚
			PF.1	I/O	通用数字输入/输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			CLKO	O	分频器输出管脚
			INT1	I	外部中断1 输入管脚
82					NC
83			VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
84					NC
85			VSS	P	地
86			VSS	P	地
87	43	31	AVSS	AP	模拟电路地
88			AVSS	AP	模拟电路地
89	44	32	PA.0	I/O	通用数字输入/输出管脚
			AD0	AI	ADC模拟输入0
			SC2_CD	I	SmartCard2 卡侦测管脚
90	45	33	PA.1	I/O	通用数字输入/输出管脚
			AD1	AI	ADC 模拟输入1
			EBI_AD12	I/O	EBI 地址/数据 总线 bit12
91	46	34	PA.2	I/O	通用数字输入/输出管脚
			AD2	AI	ADC 模拟输入2
			EBI_AD11	I/O	EBI 地址/数据 总线 bit11
			UART1_RXDRX1	I	UART1 数据接收输入管脚
92	47	35	PA.3	I/O	通用数字输入/输出管脚
			AD3	AI	ADC模拟输入3
			EBI_AD10	I/O	EBI 地址/数据 总线 bit10
			UART1_TXD	O	UART1 数据发送输出管脚
93	48	36	PA.4	I/O	通用数字输入/输出管脚
			AD4	AI	ADC 模拟输入4
			EBI_AD9	I/O	EBI 地址/数据 总线 bit9
			SC2_PWR	O	SmartCard2电源管脚
			I2C0_SDA	I/O	I ² C0数据 输入/输出 管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
94	49	37	PA.5	I/O	通用数字输入/输出管脚
			AD5	AI	ADC 模拟输入5
			EBI_AD8	I/O	EBI 地址/数据 总线 bit8
			SC2_RST	O	SmartCard2 RST 管脚
			I2C0_SCL	I/O	I ² C0时钟管脚
95	50	38	PA.6	I/O	通用数字输入/输出管脚
			AD6	AI	ADC 模拟输入6
			EBI_AD7	I/O	EBI 地址/数据 总线 bit7
			TC3	I	Timer3捕捉输入
			SC2_CLK	O	SmartCard2 时钟管脚
			PWM0_CH3	O	PWM0通道3 输出
96			PA.7	I/O	通用数字输入/输出管脚
			AD7	AI	ADC 模拟输入7
			EBI_AD6	I/O	EBI 地址/数据 总线 bit6
			TC2	I	Timer2 捕捉输入
			SC2_DAT	I/O	SmartCard2 DATA 管脚
			PWM0_CH2	O	PWM0 通道2 输出
97	51	39	VREF	AP	ADC 参考电压输入
98					NC
99	52	40	AVDD	AP	内部模拟电路电源
100			PD.0	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			SC1_CLK	O	SmartCard1 时钟管脚
			AD8	AI	ADC 模拟输入8
101			PD.1	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1数据发送输出管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SC1_DAT	I/O	SmartCard1 DATA 管脚.
			AD9	AI	ADC 模拟输入9
102			PD.2	I/O	通用数字输入/输出管脚
			UART1_nRTS	O	UART1 请求发送输出管脚
			I2S_LRCLK	I/O	I ² S 左右声道时钟
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			SC1_PWR	O	SmartCard1 电源管脚
			AD10	AI	ADC 模拟输入10
103			PD.3	I/O	通用数字输入/输出管脚
			UART1_nCTS	I	UART1 清除发送输入管脚
			I2S_BCLK	I/O	I ² S bit 时钟管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			SC1_RST	O	SmartCard1 RST 管脚
			AD11	AI	ADC 模拟输入11
104					NC
105			PD.4	I/O	通用数字输入/输出管脚
			I2S_DI	I	I ² S 数据输入
			SPI2_MISO1	I/O	SPI2 2 nd MISO (主机输入, 从机输出) 脚
			SC1_CD	I	SmartCard1 卡检测
106			PD.5	I/O	通用数字输入/输出管脚
			I2S_DO	O	I ² S 数据输出
			SPI2_MOSI1	I/O	SPI2 2 nd MOSI (主机输出, 从机输入) 脚
107	53	41	PC.7	I/O	通用数字输入/输出管脚
			DA1+OUT	AO	DAC 1 输出
			EBI_AD5	I/O	EBI 地址/数据 总线 bit5
			TC1	I	Timer1 捕捉输入
			PWM0_CH1	O	PWM0 通道1 输出
108	54	42	PC.6	I/O	通用数字输入/输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			DA0_OUT	I	DAC0 输出
			EBI_AD4	I/O	EBI 地址/数据 总线 bit4
			TC0	I	Timer 0 捕捉输入
			SC1_CD		SmartCard1 卡检测管脚
			PWM0_CH0	O	PWM0 通道0 输出
109	55		PC.15	I/O	通用数字输入/输出管脚
			EBI_AD3	I/O	EBI 地址/数据 总线 bit3
			TC0	I	Timer0 捕捉输入
			PWM1_CH2	O	PWM1 通道2 输出
110	56		PC.14	I/O	通用数字输入/输出管脚
			EBI_AD2	I/O	EBI 地址/数据 总线 bit2
			PWM1_CH3	I/O	PWM1 通道3 输出
111	57	43	PB.15	I/O	通用数字输入/输出管脚
			INT1	I	外部中断1 输入管脚
			SNOOPER	I	探听管脚
			SC1_CD	I	NC
112					
113	58	44	XT1_IN	O	外部 4~24 MHz 晶振输出管脚
			PF.3	I/O	通用数字输入/输出管脚
114	59	45	XT1_OUT	I	外部 4~24 MHz晶振输入管脚
			PF.2	I/O	通用数字输入/输出管脚
115					NC
116	60	46	nRESET	I	外部复位输入：低有效，置低复位MCU为初始状态，带内部上拉。
117	61		VSS	P	地
118			VSS	P	地
119					NC
120	62		VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
121					NC
122			PF.4	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0数据 输入/输出 管脚
123			PF.5	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
124			VSS	P	地
125	63	47	PVSS	P	PLL 地
126	64	48	PB.8	I/O	通用数字输入/输出管脚
			STADC	I	ADC 外部触发输入
			TM0	I	Timer0 外部计数输入
			INT0	I	外部中断0 输入管脚
			SC2_PWR	O	SmartCard2 电源管脚
127			PE.15	I/O	通用数字输入/输出管脚
128			PE.14	I/O	通用数字输入/输出管脚

注:

- 管脚类型 I = 数字输入 (Digital Input), O = 数字输出 (Digital Output); AI= 模拟输入 (Analog Input); AO= 模拟输出 (Analog Output); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)

3.4.3 NuMicro™ Nano130 管脚描述

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
1			PE.13	I/O	通用数字输入/输出管脚
			LCD_SEG27	O	LCD SEG 27 at LQFP128
2	1		PB.14	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
			SC2_CD	I	SmartCard2 卡检测
			SPI2_SS1	I/O	SPI2 2 nd 从机选择管脚
			LCD_SEG12	O	LCD SEG12 at LQFP64
			LCD_SEG26	O	LCD SEG26 at LQFP128
3	2		PB.13	I/O	通用数字输入/输出管脚
			EBI_AD1	I/O	EBI 地址/数据 总线 bit1
			LCD_SEG11	O	LCD SEG11 at LQFP64
			LCD_SEG25	O	LCD SEG25 at LQFP128
4	3		PB.12	I/O	通用数字输入/输出管脚
			EBI_AD0	I/O	EBI 地址/数据 总线 bit0
			CLKO	O	分频器输出管脚
			LCD_SEG10	O	LCD SEG10 at LQFP64
			LCD_SEG24	O	LCD SEG 24 at LQFP128
5					NC
6	4		X32O	O	外部 32.768 kHz 晶振输出管脚
7	5		X32I	I	外部 32.768 kHz 晶振输入管脚
8					NC
9	6		PA.11	I/O	通用数字输入/输出管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
			EBI_nRD	O	EBI 读使能输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			LCD_SEG9	O	LCD SEG 9 at LQFP64

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			LCD_SEG23	O	LCD SEG23 at LQFP128
10	7		PA.10	I/O	通用数字输入/输出管脚
			I2C1_SDA	I/O	I ² C1 数据 输入/输出 管脚
			EBI_nWR	O	EBI 写使能输出管脚
			SC0_PWR	O	SmartCard0 电源管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			LCD_SEG8	O	LCD SEG 8 at LQFP64
			LCD_SEG22	O	LCD SEG22 at LQFP128
11	8		PA.9	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
			SC0_DAT	I/O	SmartCard0 DATA 管脚(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 串行时钟管脚
			LCD_SEG7	O	LCD SEG7 at LQFP64
			LCD_SEG21	O	LCD SEG21 at LQFP128
12	9		PA.8	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
			SC0_CLK	O	SmartCard0 时钟管脚(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			LCD_SEG6	O	LCD SEG6 at LQFP64
			LCD_SEG20	O	LCD SEG20 at LQFP128
13			PD.8	I/O	通用数字输入/输出管脚
			LCD_SEG19	O	LCD SEG19 at LQFP128
14			PD.9	I/O	通用数字输入/输出管脚
			LCD_SEG18	O	LCD SEG18 at LQFP128
15			PD.10	I/O	通用数字输入/输出管脚
			LCD_SEG17	O	LCD SEG17 at LQFP128
16			PD.11	I/O	通用数字输入/输出管脚
			LCD_SEG16	O	LCD SEG 16 at LQFP128
17			PD.12	I/O	通用数字输入/输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			LCD_SEG15	O	LCD SEG15 at LQFP128
18			PD.13	I/O	通用数字输入/输出管脚
			LCD_SEG14	O	LCD SEG14 at LQFP128
19	10		PB.4	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SC0_CD	I	SmartCard0 卡检测管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			LCD_SEG5	O	LCD SEG 5 at LQFP64
			LCD_SEG13	O	LCD SEG 13 at LQFP128
20	11		PB.5	I/O	通用数字输入/输出管脚
			UART1_TXD	O	UART1 数据发送输出管脚
			SC0_RST	O	SmartCard0 RST 管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚
			LCD_SEG4	O	LCD SEG 4 at LQFP64
			LCD_SEG12	O	LCD SEG12 at LQFP128
21	12		PB.6	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1 请求发送输出管脚
			EBI_ALE	O	EBI 地址锁存使能输出管脚
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			LCD_SEG3	O	LCD SEG 3 at LQFP64
			LCD_SEG11	O	LCD SEG 11 at LQFP128
22	13		PB.7	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1 清除发送输入管脚
			EBI_nCS	O	EBI 芯片选择使能输出管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			LCD_SEG2	O	LCD SEG2 at LQFP64
			LCD_SEG10	O	LCD SEG10 at LQFP128
23					NC
24	14		LDO_CAP	P	LDO 输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
25					NC
26					NC
27	15		VDD	P	电源供应管脚，为IO端口和LDO源提供电源
28					NC
29	16		VSS	P	地
30			VSS	P	地
31			VSS	P	地
32			VSS	P	地
33			PE.12	I/O	通用数字输入/输出管脚
34			PE.11	I/O	通用数字输入/输出管脚
35			PE.10	I/O	通用数字输入/输出管脚
36			PE.9	I/O	通用数字输入/输出管脚
37			PE.8	I/O	通用数字输入/输出管脚
			LCD_SEG9	O	LCD SEG9 at LQFP128
38			PE.7	I/O	通用数字输入/输出管脚
			LCD_SEG8	O	LCD SEG8 at LQFP128
39					NC
40	17		USB_VBUS	USB	电源：来自 USB Host 或 HUB.
41	18		USB_VDD33_CAP	USB	内部电源调节器 3.3V 去耦管脚
42	19		USB_D-	USB	USB 差分信号D-
43	20		USB_D+	USB	USB 差分信号D+
44	21		PB.0	I/O	通用数字输入/输出管脚
			UART0_RXD	I	UART0 数据接收输入管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出，从机输入) 脚
			LCD_SEG1	O	LCD SEG1 at LQFP64(或作为LCD_COM5)
			LCD_SEG7	O	LCD SEG7 at LQFP128
45	22		PB.1	I/O	通用数字输入/输出管脚
			UART0_TXD	O	UART0 数据发送输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO(主机输入，从机输出) 脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			LCD_SEG0	O	LCD SEG 0 at LQFP64(或作为LCD_COM4)
			LCD_SEG6	O	LCD SEG6 at LQFP128
46	23		PB.2	I/O	通用数字输入/输出管脚
			UART0_RTSn	O	UART0 请求发送输出管脚
			EBI_nWRL	O	EBI 低字节写使能输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			LCD_COM3	O	LCD COM 3 at LQFP64
			LCD_SEG5	O	LCD SEG 5 at LQFP128
47	24		PB.3	I/O	通用数字输入/输出管脚
			UART0_CTSn	I	UART0 清除发送输入管脚
			EBI_nWRH	O	EBI 高字节写使能输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚
			LCD_COM2	O	LCD COM2 at LQFP64
			LCD_SEG4	O	LCD SEG4 at LQFP128
48			PD.6	I/O	通用数字输入/输出管脚
			LCD_SEG3	O	LCD SEG3 at LQFP128
49			PD.7	I/O	通用数字输入/输出管脚
			LCD_SEG2	O	LCD SEG 2 at LQFP128
50			PD.14	I/O	通用数字输入/输出管脚
			LCD_SEG1	O	LCD SEG1 at LQFP128
51			PD.15	I/O	通用数字输入/输出管脚
			LCD_SEG0	O	LCD SEG 0 at LQFP128
52			PC.5	I/O	通用数字输入/输出管脚
			SPI0_MOSI1	I/O	SPI0 2 nd MOSI (主机输出, 从机输入) 脚
			LCD_COM3	O	LCD COM3 at LQFP128
53			PC.4	I/O	通用数字输入/输出管脚
			SPI0_MISO1	I/O	SPI0 2 nd MISO (主机输入, 从机输出) 脚
			LCD_COM2	O	LCD COM 2 at LQFP128
54	25		PC.3	I/O	通用数字输入/输出管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			I2S_DO	O	I ² S 数据输出
			SC1_RST	O	SmartCard1 RST 管脚
			LCD_COM1	O	LCD COM1 at LQFP64
			LCD_COM1	O	LCD COM1 at LQFP128
55	26		PC.2	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			I2S_DI	I	I ² S 数据输入
			SC1_PWR	O	SmartCard1 PWR 管脚
			LCD_COM0	O	LCD COM0 at LQFP64
			LCD_COM0	O	LCD COM0 at LQFP128
56	27		PC.1	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚
			I2S_BCLK	I/O	I ² S bit 时钟管脚
			SC1_DAT	I/O	SmartCard1 DATA 管脚(SC1_UART_RXD)
			LCD_DH2	O	LCD DH2 at LQFP64
			LCD_DH2	O	LCD DH2 at LQFP128
57	28		PC.0 / MCLKO	I/O	通用数字输入/输出管脚 / 模块时钟输出管脚
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚
			I2S_LRCLK	I/O	I ² S 左右声道时钟
			SC1_CLK	O	SmartCard1 时钟管脚(SC0_UART_TXD)
			LCD_DH1	O	LCD DH1 at LQFP64
			LCD_DH1	O	LCD DH1 at LQFP100
58			PE.6	I/O	通用数字输入/输出管脚
59	29		LCD_VLCD	AO	LCD VLCD 管脚
60					NC
61			PE.5		通用数字输入/输出管脚
62	30		PB.11	I/O	通用数字输入/输出管脚
			PWM1_CH0	I/O	PWM1 通道0 输出

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			TM3	O	Timer3 外部计数输入
			SC2_DAT	I/O	SmartCard2 DATA管脚(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
			LCD_V1	O	LCD V1 at LQFP64
			LCD_V1	O	LCD V1 at LQFP128
63	31		PB.10	I/O	通用数字输入/输出管脚
			SPI0_SS1	I/O	SPI0 2 nd 从机选择管脚
			TM2	O	Timer2 外部计数输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
			LCD_V2	O	LCD V2 at LQFP64
			LCD_V2	O	LCD V2 at LQFP128
64	32		PB.9	I/O	通用数字输入/输出管脚
			SPI1_SS1	I/O	SPI1 2 nd 从机选择管脚
			TM1	O	Timer1 外部计数输入
			SC2_RST	O	SmartCard2 RST 管脚
			INT0	I	外部中断0 输入管脚
			LCD_V3	O	LCD V3 at LQFP64
			LCD_V3	O	LCD V3 at LQFP128
65			PE.4	I/O	通用数字输入/输出管脚
			SPI0_MOSI0	I/O	SPI0 1 st MOSI (主机输出, 从机输入) 脚
66			PE.3	I/O	通用数字输入/输出管脚
			SPI0_MISO0	I/O	SPI0 1 st MISO (主机输入, 从机输出) 脚
67			PE.2	I/O	通用数字输入/输出管脚
			SPI0_CLK	I/O	SPI0 串行时钟管脚
68			PE.1	I/O	通用数字输入/输出管脚
			PWM1_CH3	I/O	PWM1 通道3 输出
			SPI0_SS0	I/O	SPI0 1 st 从机选择管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
69			PE.0	I/O	通用数字输入/输出管脚
			PWM1_CH2	I/O	PWM1 通道2 输出
			I2S_MCLK	O	I ² S 主机时钟输出管脚
70			PC.13	I/O	通用数字输入/输出管脚
			SPI1_MOSI1	I/O	SPI1 2 nd MOSI (主机输出, 从机输入) 脚
			PWM1_CH1	O	PWM1 通道1 输出
			SNOOPER	I	探听管脚
			INT1	I	外部中断0 输入管脚
			I2C0_SCL	O	I2C0 时钟管脚
71			PC.12	I/O	通用数字输入/输出管脚
			SPI1_MISO1	I/O	SPI1 2 nd MISO(主机输入, 从机输出) 脚
			PWM1_CH0	O	PWM1 通道0 输出
			INT0	I	外部中断0 输入管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出管脚
72	33		PC.11	I/O	通用数字输入/输出管脚
			SPI1_MOSI0	I/O	SPI1 1 st MOSI (主机输出, 从机输入) 脚
			UART1_TXD	O	UART1 数据发送输出管脚
			LCD_SEG31	O	LCD SEG31 at LQFP64
73	34		PC.10	I/O	通用数字输入/输出管脚
			SPI1_MISO0	I/O	SPI1 1 st MISO(主机输入, 从机输出) 脚
			UART1_RXD	I	UART1 数据接收输入管脚
			LCD_SEG30	O	LCD SEG30 at LQFP64
74	35		PC.9	I/O	通用数字输入/输出管脚
			SPI1_CLK	I/O	SPI1 串行时钟管脚
			I2C1_SCL	I/O	I ² C1 时钟管脚
			LCD_SEG29	O	LCD SEG29 at LQFP64
75	36		PC.8	I/O	通用数字输入/输出管脚
			SPI1_SS0	I/O	SPI1 1 st 从机选择管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			EBI_MCLK	O	EBI 外部时钟输出管脚
			I2C1_SDA	I/O	I ² C1 数据 输入/输出 管脚
			LCD_SEG28	O	LCD SEG28 at LQFP64
76	37		PA.15	I/O	通用数字输入/输出管脚
			PWM0_CH3	I/O	PWM0 通道3 输出
			I2S_MCLK	O	I ² S 主机时钟输出管脚
			TC3	I	Timer3 捕捉输入
			SC0_PWR	O	SmartCard0 电源管脚
			UART0_TXD	O	UART0 数据发送输出管脚
			LCD_SEG27	O	LCD SEG27 at LQFP64
77	38		PA.14	I/O	通用数字输入/输出管脚
			PWM0_CH2	I/O	PWM0 通道2 输出
			EBI_AD15	I/O	EBI 地址/数据 总线 bit15
			TC2	I	Timer 2 捕捉输入
			UART0_RXD	I	UART0 数据接收输入管脚
			LCD_SEG26	O	LCD SEG26 at LQFP64
78	39		PA.13	I/O	通用数字输入/输出管脚
			PWM0_CH1	I/O	PWM0 通道1 输出
			EBI_AD14	I/O	EBI 地址/数据 总线 bit14
			TC1	I	Timer1 捕捉输入
			I2C0_SCL	I/O	I ² C0 时钟管脚
			LCD_SEG25	O	LCD SEG25 at LQFP64
79	40		PA.12	I/O	通用数字输入/输出管脚
			PWM0_CH0	I/O	PWM0 通道0 输出
			EBI_AD13	I/O	EBI 地址/数据 总线 bit13
			TC0	I	Timer 0 捕捉输入
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
			LCD_SEG24	O	LCD SEG24 at LQFP64

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
80	41		ICE_DAT	I/O	调试器的串行数据管脚
			PF.0	I/O	通用数字输入/输出管脚
			INT0	I	外部中断0 输入管脚
81	42		ICE_CLK	I	调试器的串行时钟管脚
			PF.1	I/O	通用数字输入/输出管脚
			CLKO	O	分频器输出管脚
			INT1	I	外部中断1 输入管脚
82					NC
83			VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
84					NC
85			VSS	P	地
86			VSS	P	地
87	43		AVSS	AP	模拟电路地
88			AVSS	AP	模拟电路地
89	44		PA.0	I/O	通用数字输入/输出管脚
			AD0	AI	ADC 模拟输入0
			SC2_CD	I	SmartCard2 卡检测
90	45		PA.1	I/O	通用数字输入/输出管脚
			AD1	AI	ADC 模拟输入1
			EBI_AD12	I/O	EBI 地址/数据 总线 bit12
91	46		PA.2	I/O	通用数字输入/输出管脚
			AD2	AI	ADC 模拟输入2
			EBI_AD11	I/O	EBI 地址/数据 总线 bit11
			UART1_RXD	I	UART1 数据接收输入管脚
			LCD_SEG23*	AO	LCD SEG23 at LQFP64
92	47		PA.3	I/O	通用数字输入/输出管脚
			AD3	AI	ADC 模拟输入3
			EBI_AD10	I/O	EBI 地址/数据 总线 bit10

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			UART1_TXD	O	UART1 数据发送输出管脚
			LCD_SEG22*	AO	LCD SEG22 at LQFP64
93	48		PA.4	I/O	通用数字输入/输出管脚
			AD4	AI	ADC 模拟输入4
			EBI_AD9	I/O	EBI 地址/数据 总线 bit9
			SC2_PWR	O	SmartCard2 电源管脚
			I2C0_SDA	I/O	I ² C0 数据 输入/输出 管脚
			LCD_SEG21*	AO	LCD SEG21 at LQFP64
			LCD_SEG39*	AO	LCD SEG39 at LQFP128
94	49		PA.5	I/O	通用数字输入/输出管脚
			AD5	AI	ADC 模拟输入5
			EBI_AD8	I/O	EBI 地址/数据 总线 bit8
			SC2_RST	O	SmartCard2 RST 管脚
			I2C0_SCL	I/O	I ² C0 时钟管脚
			LCD_SEG20*	AO	LCD SEG20 at LQFP64
			LCD_SEG38*	AO	LCD SEG38 at LQFP128
95	50		PA.6	I/O	通用数字输入/输出管脚
			AD6	AI	ADC 模拟输入6
			EBI_AD7	I/O	EBI 地址/数据 总线 bit7
			TC3	I	Timer3 捕捉输入
			SC2_CLK	O	SmartCard2 时钟管脚(SC2_UART_TXD)
			PWM0_CH3	O	PWM0 通道3 输出
			LCD_SEG19*	AO	LCD SEG19 at LQFP64
			LCD_SEG37*	AO	LCD SEG37 at LQFP128
96			PA.7	I/O	通用数字输入/输出管脚
			AD7	AI	ADC 模拟输入7
			EBI_AD6	I/O	EBI 地址/数据 总线 bit6
			TC2	I	Timer2 捕捉输入

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
			SC2_DAT	I/O	SmartCard2 DATA 管脚(SC2_UART_RXD)
			PWM0_CH2	O	PWM0 通道2 输出
			LCD_SEG36*	AO	LCD SEG36 at LQFP128
97	51		VREF	AP	ADC 参考电压输入
98					NC
99	52		AVDD	AP	内部模拟电路电源
100			PD.0	I/O	通用数字输入/输出管脚
			UART1_RXD	I	UART1 数据接收输入管脚
			SPI2_SS0	I/O	SPI2 1 st 从机选择管脚
			SC1_CLK	O	SmartCard1 时钟管脚(SC1_UART_TXD)
			AD8	AI	ADC 模拟输入8
101			PD.1	I/O	通用数字输入/输出管脚
			TX1	O	UART1 数据发送输出管脚
			SPI2_CLK	I/O	SPI2 串行时钟管脚
			SC1_DAT	I/O	SmartCard1 DATA 管脚(SC1_UART_RXD)
			AD9	AI	ADC 模拟输入9
102			PD.2	I/O	通用数字输入/输出管脚
			UART1_RTSn	O	UART1 请求发送输出管脚
			I2S_LRCLK	I/O	I ² S 左右声道时钟
			SPI2_MISO0	I/O	SPI2 1 st MISO (主机输入, 从机输出) 脚
			SC1_PWR	O	SmartCard1 电源管脚
			AD10	AI	ADC 模拟输入10
103			PD.3	I/O	通用数字输入/输出管脚
			UART1_CTSn	I	UART1 清除发送输入管脚
			I2S_BCLK	I/O	I ² S bit 时钟管脚
			SPI2_MOSI0	I/O	SPI2 1 st MOSI (主机输出, 从机输入) 脚
			SC1_RST	O	SmartCard1 RST 管脚
			AD11	AI	ADC模拟输入11

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
104					NC
105			PD.4	I/O	通用数字输入/输出管脚
			I2S_DI	I	I ² S 数据输入
			SPI2_MISO1	I/O	SPI2 2 nd MISO (主机输入, 从机输出) 脚
			SC1_CD	I	SmartCard1 卡检测
			LCD_SEG35	AO	LCD SEG35 at LQFP128
106			PD.5	I/O	通用数字输入/输出管脚
			I2S_DO	O	I ² S 数据输出
			SPI2_MOSI1	I/O	SPI2 2 nd MOSI (主机输出, 从机输入) 脚
			LCD_SEG34	AO	LCD SEG 34 at LQFP128
107	53		PC.7	I/O	通用数字输入/输出管脚
			DA1_OUT	AO	DAC 1 输出
			EBI_AD5	I/O	EBI 地址/数据 总线 bit5
			TC1	I	Timer1 捕捉输入
			PWM0_CH1	O	PWM1 通道1 输出
			LCD_SEG17*	AO	LCD SEG17 at LQFP64
108	54		PC.6	I/O	通用数字输入/输出管脚
			DA0_OUT	I	DAC0 输出
			EBI_AD4	I/O	EBI 地址/数据 总线 bit4
			TC0	I	Timer 0 捕捉输入
			SC1_CD		SmartCard1 卡检测管脚
			PWM0_CH0	O	PWM0 通道0 输出
109	55		PC.15	I/O	通用数字输入/输出管脚
			EBI_AD3	I/O	EBI 地址/数据 总线 bit3
			TC0	I	Timer0 捕捉输入
			PWM1_CH2	O	PWM1 通道1 输出
			LCD_SEG16	AO	LCD SEG16 at LQFP64
			LCD_SEG33	AO	LCD SEG33 at LQFP128

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
110	56		PC.14	I/O	通用数字输入/输出管脚
			EBI_AD2	I/O	EBI 地址/数据 总线 bit2
			PWM1_CH3	I/O	PWM1 通道3 输出
			LCD_SEG15	AO	LCD SEG15 at LQFP64
			LCD_SEG32	AO	LCD SEG32 at LQFP128
111	57		PB.15	I/O	通用数字输入/输出管脚
			INT1	I	外部中断1 输入管脚
			SNOOPER	I	探听管脚
			SC1_CD	I	SmartCard1 卡检测
			LCD_SEG14	AO	LCD SEG14 at LQFP64
			LCD_SEG31	AO	LCD SEG31 at LQFP128
112					NC
113	58		XT1_IN	O	外部 4~24 MHz 晶振输出管脚
			PF.3	I/O	通用数字输入/输出管脚
114	59		XT1_OUT	I	外部 4~24 MHz晶振输入管脚
			PF.2	I/O	通用数字输入/输出管脚
115					NC
116	60		nRESET	I	外部复位输入：低有效，置低复位MCU为初始状态，带内部上拉。
117	61		VSS	P	地
118			VSS	P	地
119					NC
120	62		VDD	P	电源供应引脚，为IO端口、内部PLL电路LDO源和数字电路提供电源
121					NC
122			PF.4	I/O	通用数字输入/输出管脚
			I2C0_SDA	I/O	I2C0 数据 输入/输出 管脚
123			PF.5	I/O	通用数字输入/输出管脚
			I2C0_SCL	I/O	I2C0 时钟管脚

管脚号.			管脚名称	管脚类型	描述
LQFP 128	LQFP 64	LQFP 48			
124			VSS	P	地
125	63		PVSS	I/O	PLL 地
126	64		PB.8	I/O	通用数字输入/输出管脚
			STADC	I	ADC 外部触发输入
			TM0	I	Timer0 外部计数输入
			INT0	I	外部中断0 输入管脚
			SC2_PWR	O	SmartCard2 电源管脚
			LCD_SEG13	AO	LCD SEG13 at LQFP64
			LCD_SEG30	AO	LCD SEG30 at LQFP128
127			PE.15	I/O	通用数字输入/输出管脚
			LCD_SEG29	O	LCD SEG 29 at LQFP128
128			PE.14	I/O	通用数字输入/输出管脚
			LCD_SEG28	O	LCD SEG 28 at LQFP128

注:

1. 管脚类型 I = 数字输入 (Digital Input), O = 数字输出 (Digital Output); AI= 模拟输入 (Analog Input); AO= 模拟输出 (Analog Output); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)
2. *: ADC与LCD共享的管脚其输出电压不可高于 VDD

4 框图

4.1 Nano100 框图

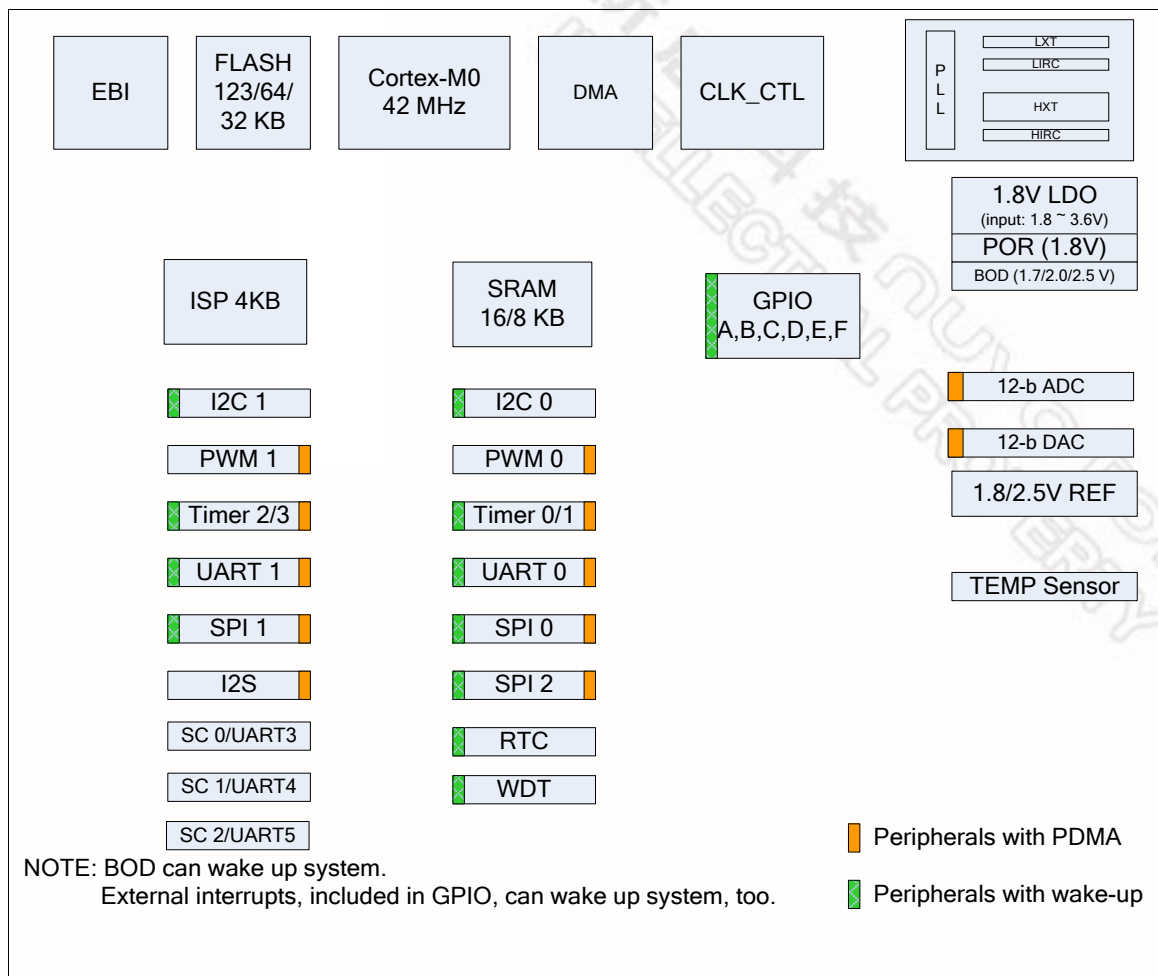


图 4-1 NuMicro™ Nano100 框图

4.2 Nano110 框图

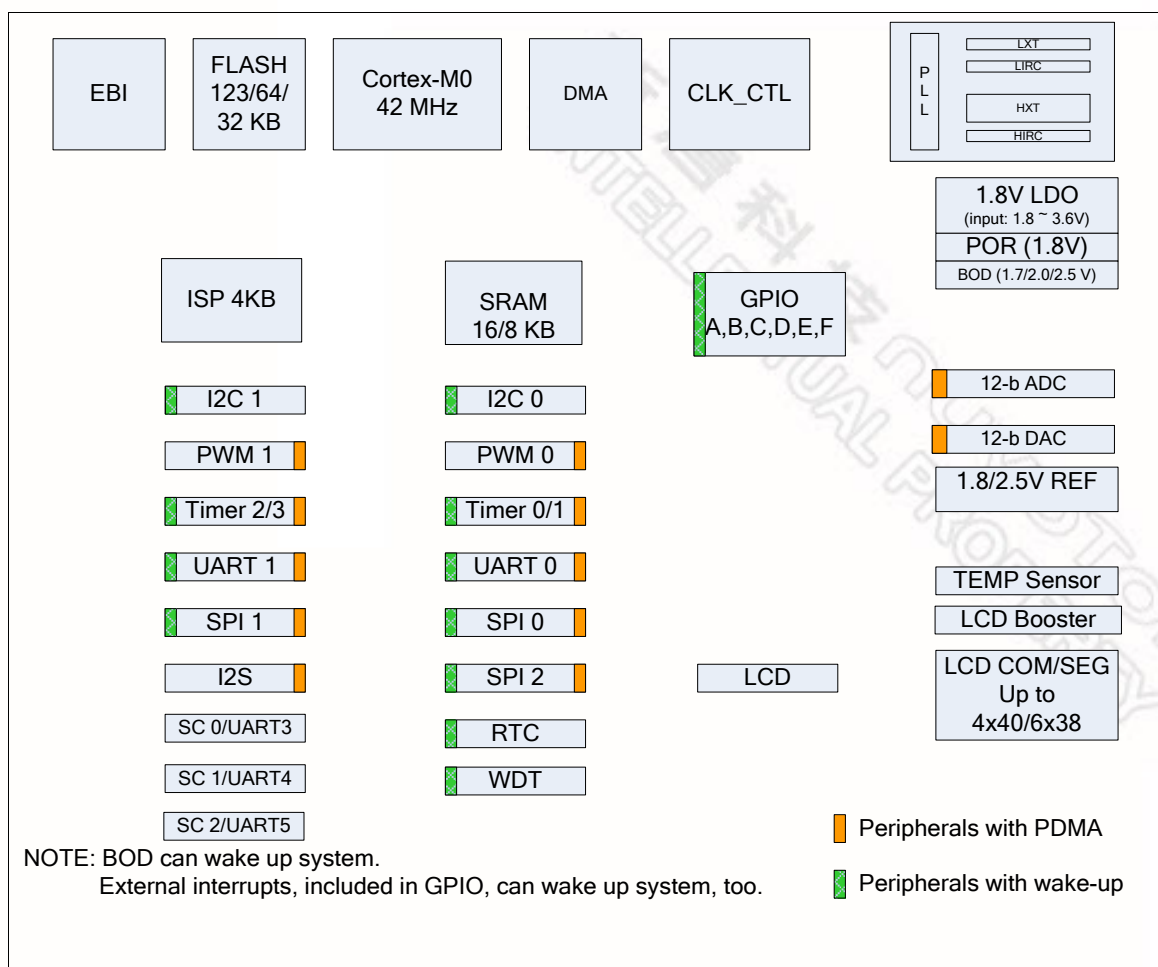


图 4-2 NuMicro™ Nano110 框图

4.3 Nano120 框图

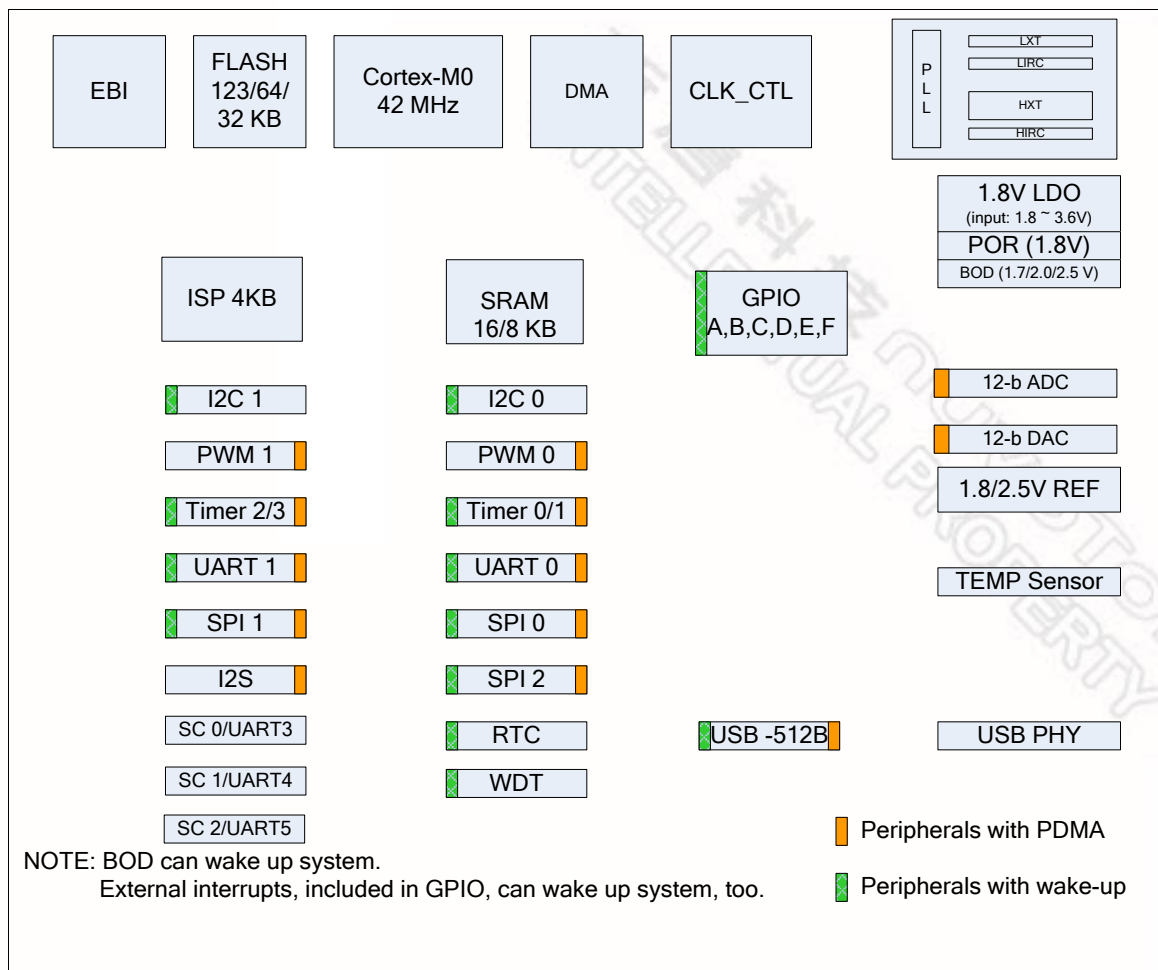


图 4-3 NuMicro™ Nano120 框图

4.4 Nano130 框图

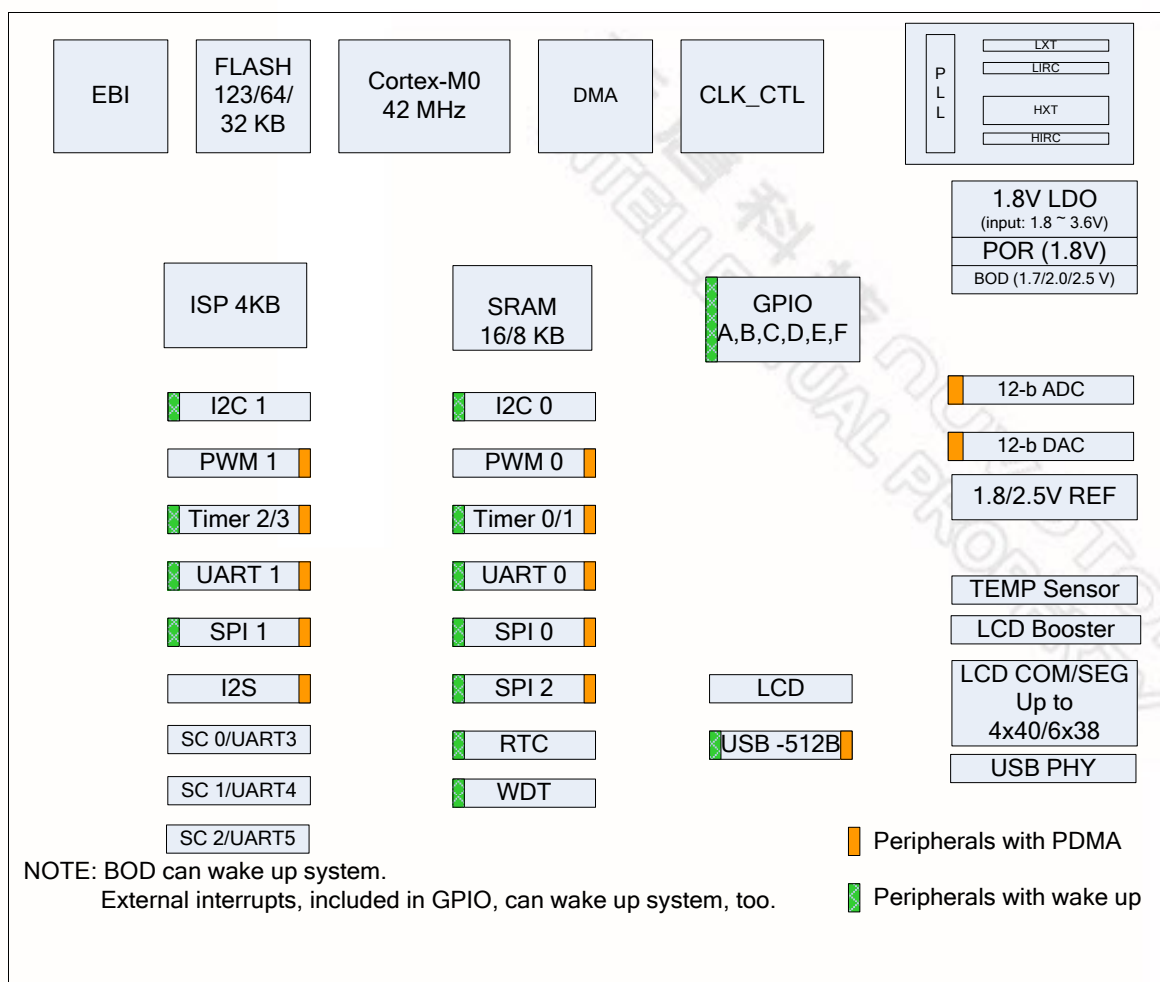


图 4-4 NuMicro™ Nano130 框图

5 功能描述

5.1 内存结构

5.1.1 概述

Nano100提供 4G-字节的寻址空间。内存分配情况见下表，对各片上外设的详细的寄存器描述，内存空间，和编程指南，稍后章节将有详细描述。Nano100系列仅支持小端数据格式。

5.1.2 存储器映射

各片上控制器的存储器分配见下表。

Address Space	Token	Modules
Flash & SRAM Memory Space		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH 内存空间(128KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM 内存空间(16KB)
0x6000_0000 --- 0x6001_FFFF	EXTMEM_BA	External 内存空间(128KB)
AHB Modules Space (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 控制寄存器
0x5000_8000 – 0x5000_BFFF	DMA_BA	DMA 控制寄存器
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash 内存控制寄存器
0x5001_0000 – 0x5001_03FF	EBI_BA	外部总线接口控制寄存器
APB1 Modules Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	看门狗控制寄存器
0x4000_8000 – 0x4000_BFFF	RTC_BA	RTC 控制寄存器
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0/Timer1 控制寄存器
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I ² C0 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
0x4004_0000 – 0x4004_3FFF	PWM0_BA	PWM 0控制寄存器
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART 0控制寄存器
0x4006_0000 – 0x4006_3FFF	USBD_BA	USB 2.0 FS 设备控制寄存器
0x400A_0000 – 0x400A_3FFF	DAC_BA	DAC 控制寄存器

0x400B_0000 – 0x400B_3FFF	LCD_BA	LCD 控制寄存器
0x400D_0000 – 0x400D_3FFF	SPI2_BA	带主/从功能的SPI2 控制寄存器
0x400E_0000 – 0x400E_3FFF	ADC12_BA	12-位 ADC控制寄存器
APB2 Modules Space (0x4010_0000 ~ 0x401F_FFFF)		
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2/Timer3控制寄存器
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I ² C1 接口控制寄存器
0x4013_0000 – 0x4013_3FFF	SPI1_BA	带主/从功能的SPI1 控制寄存器
0x4014_0000 – 0x4014_3FFF	PWM1_BA	PWM1 控制寄存器
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1控制寄存器
0x4019_0000 – 0x4019_3FFF	SC0_BA	Smart Card 0控制寄存器
0x401A_0000 – 0x401A_3FFF	I2S_BA	I ² S控制寄存器
0x401B_0000 – 0x401B_3FFF	SC1_BA	Smart Card 1控制寄存器
0x401C_0000 – 0x401C_3FFF	SC2_BA	Timer2/Timer3控制寄存器
System Control Space (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	SCS_BA	系统 控制寄存器

5.2 嵌套向量中断控制器 (NVIC)

5.2.1 概述

Cortex-M0 提供中断控制器，用于总体管理异常，称之为“嵌套向量中断控制器 (NVIC)”。NVIC 和处理器内核紧密相连，它提供以下特征。

5.2.2 特征

- 支持嵌套和向量中断
- 自动保存和恢复处理器状态
- 动态改变优先级
- 简化的和确定的中断时间

NVIC 依照优先级处理所有支持的异常，所有异常在“处理器模式”处理。NVIC 结构支持 32(IRQ[31:0]) 个离散中断，每个中断可以支持 4 级离散中断优先级。所有的中断和大多数系统异常可以配置为不同优先级。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，则立即处理新中断。

当接受任何中断时，ISR的开始地址可从内存的向量表中取得。不需要确定哪个中断被响应，也不要软件分配相关中断服务程序 (ISR) 的开始地址。当获取中断入口地址时，NVIC 将自动保存处理状态到栈中，包括以下寄存器“PC, PSR, LR, R0~R3, R12”的值。在ISR结束时，NVIC 将从栈中恢复相关寄存器的值，进行正常操作，因此花费少量且确定的时间处理中断请求。

NVIC 支持末尾连锁“Tail Chaining”，有效处理背对背中断“back-to-back interrupts”，即无需保存和恢复当前状态从而减少在切换当前ISR时的延迟时间。NVIC 还支持迟到“Late Arrival”，改善同时发生的ISR的效率。当较高优先级中断请求发生在当前ISR开始执行之前（保持处理器状态和获取起始地址阶段），NVIC 将立即处理更高优先级的中断，从而提高了实时性。

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

5.2.3 异常模式和系统中断映射

Nano100 系列支持下表所列的异常模式。与所有中断一样，软件可以对其中一些中断设置4级优先级。最高优先级为“0”，最低优先级为“3”，所有用户可配置的优先级的默认值为“0”。注意：优先级为“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 5-1 异常模式

向量号	中断号 (在中断寄存器的位)	中断名称	源 IP	中断描述
0 ~ 15	-	-	-	系统异常
16	0	BOD_INT	Brown-out	欠压检测中断
17	1	WDT_INT	WDT	看门狗定时器中断
18	2	EINT0	GPIO	PB.14 管脚上的外部信号中断
19	3	EINT1	GPIO	PB.15 管脚上的外部信号中断
20	4	GPABC_INT	GPIO	PA[15:0]/PB[13:0]/PC[15:0] 的外部信号中断
21	5	GPDEF_INT	GPIO	PD[15:0]/PE[15:0]/PF[5:0] 的外部信号中断
22	6	PWM0_INT	PWM0	PWM 0中断
23	7	PWM1_INT	PWM1	PWM中断
24	8	TMR0_INT	TMR0	Timer 0中断
25	9	TMR1_INT	TMR1	Timer 1中断
26	10	TMR2_INT	TMR2	Timer 2中断
27	11	TMR3_INT	TMR3	Timer 3中断
28	12	UART0_INT	UART0	UART0中断
29	13	UART1_INT	UART1	UART1中断
30	14	SPI0_INT	SPI0	SPI0中断
31	15	SPI1_INT	SPI1	SPI1中断
32	16	SPI2_INT	SPI2	SPI2中断
33	17	IRC_INT	IRC	IRC TRIM中断
34	18	I2C0_INT	I2C0	I ² C0中断
35	19	I2C1_INT	I2C1	I ² C1中断
36	20	SC2_INT	SC2	Smart Card 2中断
37	21	SC0_INT	SC0	Smart Card 0中断
38	22	SC1_INT	SC1	Smart Card 1中断

向量号	中断号 (在中断寄存器的位)	中断名称	源 IP	中断描述
39	23	USB_INT	USBD	USB FS 设备中断
41	25	LCD_INT	LCD	LCD中断
42	26	DMA_INT	DMA	DMA中断
43	27	I2S_INT	I2S	I ² S中断
44	28	PD_WU_INT	CLKC	从掉电状态唤醒的时钟控制器中断
45	29	ADC_INT	ADC	ADC 中断
46	30	DAC_INT	DAC	DAC 中断
47	31	RTC_INT	RTC	实时时钟中断

表 5-2 系统中断映射

5.2.4 向量表

响应中断时，处理器自动从内存的向量表中取出中断服务例程（ISR）的起始地址。对于 ARMv6-M，向量表的基地址为 0x00000000。向量表包括复位后堆栈的初始值以及所有异常处理的入口地址。向量号表示处理异常的先后次序。

向量表字偏移量	描述
0	SP_main – 主栈指针
向量号	异常入口指针，用向量号表示

表 5-3 向量表格式

5.2.5 操作说明

通过写相应中断使能置位寄存器或清使能寄存器，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 清零，寄存器读取返回当前相应中断的使能状态，当中断禁用时，中断声明将使中断挂起，因此中断不被激活，如果在禁用时中断被激活，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。清使能位可以阻止新的相应中断被激活。

NVIC 中断可以使用互补的寄存器对来挂起/取消挂起以使能/禁用这些中断，这些寄存器分别为 Set-Pending 寄存器与 Clear-Pending，可以写 1 使能和写 1 禁用，这些寄存器读取返回当前相应中断的状态。对寄存器 Clear-Pending 操作不会影响执行中的中断。

NVIC 中断依次更新32位寄存器中的各个8位字段（每个寄存器支持4个中断）。

与 NVIC 相关的通用寄存器都可以在内存系统控制空间寄存器（SCS_BA）其中的一块寄存器区域中设置，下一节将作出描述。

5.2.6 NVIC 控制寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
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 + 180	R/W	IRQ0 ~ IRQ31 清除使能控制寄存器	0x0000_0000
NVIC_ISPR	SCS_BA + 200	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	0x0000_0000
NVIC_ICPR	SCS_BA + 280	R/W	IRQ0~IRQ31清除挂起控制寄存器	0x0000_0000
NVIC_IPR0	SCS_BA + 400	R/W	IRQ0 ~ IRQ3 优先级控制寄存器	0x0000_0000
NVIC_IPR1	SCS_BA + 404	R/W	IRQ4 ~ IRQ7 优先级控制寄存器	0x0000_0000
NVIC_IPR2	SCS_BA + 408	R/W	IRQ8 ~ IRQ11 优先级控制寄存器	0x0000_0000
NVIC_IPR3	SCS_BA + 40C	R/W	IRQ12 ~ IRQ15 优先级控制寄存器	0x0000_0000
NVIC_IPR4	SCS_BA + 410	R/W	IRQ16 ~ IRQ19 优先级控制寄存器	0x0000_0000
NVIC_IPR5	SCS_BA + 414	R/W	IRQ20 ~ IRQ23 优先级控制寄存器	0x0000_0000
NVIC_IPR6	SCS_BA + 418	R/W	IRQ24 ~ IRQ27 优先级控制寄存器	0x0000_0000
NVIC_IPR7	SCS_BA + 41C	R/W	IRQ28 ~ IRQ31 优先级控制寄存器	0x0000_0000

5.2.7 NVIC 控制寄存器描述

IRQ0 ~ IRQ31 设置使能控制寄存器 (NVIC_ISER)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ISER	SCS_BA + 0x100	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							

Bits	描述	
[31:0]	SETENA[31:0]	<p>使能32个中的一个或者多个中断，每位代表 IRQ0 ~ IRQ31 的中断号（向量号：从16到47）。写 1 使能相关中断</p> <p>写 0 无效</p> <p>读取该寄存器返回当前使能状态。</p>

IRQ0 ~ IRQ31 清除使能控制寄存器 (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 清除使能控制寄存器	0x0000_0000

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

Bits	描述
[31:0]	<p>禁用32个中的一个或者多个中断，每位代表 IRQ0 ~ IRQ31 的中断号（向量号：从16到47）。</p> <p>写 1 禁用相应中断</p> <p>写 0 无效</p> <p>读取该寄存器返回当前使能状态。</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							

Bits	描述	
[31:0]	SETPEND[31:0]	<p>写 1，由软件控制挂起相应中断。每位代表 IRQ0 ~ IRQ31 的中断号（向量号：从16到47）。</p> <p>写 0 无效</p> <p>读取该寄存器返回当前等待处理的中断状态。</p>

IRQ0 ~ IRQ31 清除挂起控制寄存器 (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清除挂起控制寄存器	0x0000_0000

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

Bits	描述
[31:0]	<p>写 1 清除，由软件控制清除等待处理的中断，每位代表 IRQ0 ~ IRQ31 的中断号（向量号：从16到47）。</p> <p>写 0 无效</p> <p>读取该寄存器返回当前等待处理的中断状态。</p>

IRQ0 ~ IRQ3 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_3[1:0]	IRQ3 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_2[1:0]	IRQ2 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_1[1:0]	IRQ1 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_0[1:0]	IRQ0 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ4 ~ IRQ7 中断优先级控制寄存器 (NVIC_IPR1)

寄存器	偏移量	R/W	描述	复位后的值
NVIC_IPR1	SCS_BA + 404	R/W	IRQ4 ~ IRQ7 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_7[1:0]	IRQ7 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_6[1:0]	IRQ6 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_5[1:0]	IRQ5 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_4[1:0]	IRQ4 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ8 ~ IRQ11 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_11[1:0]	IRQ11优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_10[1:0]	IRQ10 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_9[1:0]	IRQ9 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_8[1:0]	IRQ8 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ12 ~ IRQ15 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_15[1:0]	IRQ15 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_14[1:0]	IRQ14 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_13[1:0]	IRQ13 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_12[1:0]	IRQ12 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ16 ~ IRQ19 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_19[1:0]	IRQ19 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_18[1:0]	IRQ18 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_17[1:0]	IRQ17 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_16[1:0]	IRQ16 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ20 ~ IRQ23 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_23[1:0]	IRQ23 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_22[1:0]	IRQ22 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_21[1:0]	IRQ21 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_20[1:0]	IRQ20 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ24 ~ IRQ27 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_27[1:0]	IRQ27 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_26[1:0]	IRQ26 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_25[1:0]	IRQ25 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_24[1:0]	IRQ24 优先级 “0”表示最高优先级 & “3”表示最低优先级

IRQ28 ~ IRQ31 中断优先级控制寄存器 (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 中断优先级控制寄存器	0x0000_0000

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

Bits	描述	
[31:30]	PRI_31[1:0]	IRQ31 优先级 “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_30[1:0]	IRQ30 优先级 “0”表示最高优先级 & “3”表示最低优先级
[15:14]	PRI_29[1:0]	IRQ29 优先级 “0”表示最高优先级 & “3”表示最低优先级
[7:6]	PRI_28[1:0]	P IRQ28 优先级 “0”表示最高优先级 & “3”表示最低优先级

5.2.8 中断源控制寄存器

除了 NVIC 相关的中断控制寄存器外, NuMicro™ Nano100 系列还有一些特殊寄存器便于中断处理, 包括“中断源识别”, “NMI 源选择”与“中断测试模式”。描述如下:

5.2.8.1 中断源控制寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
INT_BA = 0x5000_0300				
IRQ0_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD_INT) 中断源识别	0xFFFF_FFFF
IRQ1_SRC	INT_BA+0x04	R	MCU IRQ1 (WDT_INT) 中断源识别	0xFFFF_FFFF
IRQ2_SRC	INT_BA+0x08	R	MCU IRQ2 (EINT0) 中断源识别	0xFFFF_FFFF
IRQ3_SRC	INT_BA+0x0C	R	MCU IRQ3 (EINT1) 中断源识别	0xFFFF_FFFF
IRQ4_SRC	INT_BA+0x10	R	MCU IRQ4 (GPABC_INT) 中断源识别	0xFFFF_FFFF
IRQ5_SRC	INT_BA+0x14	R	MCU IRQ5 (GPDEF_INT) 中断源识别	0xFFFF_FFFF
IRQ6_SRC	INT_BA+0x18	R	MCU IRQ6 (PWM0_INT) 中断源识别	0xFFFF_FFFF
IRQ7_SRC	INT_BA+0x1C	R	MCU IRQ7 (PWM1_INT) 中断源识别	0xFFFF_FFFF
IRQ8_SRC	INT_BA+0x20	R	MCU IRQ8 (TMR0_INT) 中断源识别	0xFFFF_FFFF
IRQ9_SRC	INT_BA+0x24	R	MCU IRQ9 (TMR1_INT) 中断源识别	0xFFFF_FFFF
IRQ10_SRC	INT_BA+0x28	R	MCU IRQ10 (TMR2_INT) 中断源识别	0xFFFF_FFFF
IRQ11_SRC	INT_BA+0x2C	R	MCU IRQ11 (TMR3_INT) 中断源识别	0xFFFF_FFFF
IRQ12_SRC	INT_BA+0x30	R	MCU IRQ12 (UART0_INT) 中断源识别	0xFFFF_FFFF
IRQ13_SRC	INT_BA+0x34	R	MCU IRQ13 (UART1_INT) 中断源识别	0xFFFF_FFFF
IRQ14_SRC	INT_BA+0x38	R	MCU IRQ14 (SPI0_INT) 中断源识别	0xFFFF_FFFF
IRQ15_SRC	INT_BA+0x3C	R	MCU IRQ15 (SPI1_INT) 中断源识别	0xFFFF_FFFF
IRQ16_SRC	INT_BA+0x40	R	MCU IRQ16 (SPI2_INT) 中断源识别	0xFFFF_FFFF
IRQ17_SRC	INT_BA+0x44	R	MCU IRQ17 (IRC_INT) 中断源识别	0xFFFF_FFFF
IRQ18_SRC	INT_BA+0x48	R	MCU IRQ18 (I2C0_INT) 中断源识别	0xFFFF_FFFF
IRQ19_SRC	INT_BA+0x4C	R	MCU IRQ19 (I2C1_INT) 中断源识别	0xFFFF_FFFF
IRQ20_SRC	INT_BA+0x50	R	MCU IRQ20 (保留) 中断源识别	0xFFFF_FFFF
IRQ21_SRC	INT_BA+0x54	R	MCU IRQ21 (SC0_INT) 中断源识别	0xFFFF_FFFF
IRQ22_SRC	INT_BA+0x58	R	MCU IRQ22 (SC1_INT) 中断源识别	0xFFFF_FFFF

IRQ23_SRC	INT_BA+0x5C	R	MCU IRQ23 (USB_INT) 中断源识别	0xFFFF_XXXX
IRQ25_SRC	INT_BA+0x64	R	MCU IRQ25 (LCD_INT) 中断源识别	0xFFFF_XXXX
IRQ26_SRC	INT_BA+0x68	R	MCU IRQ26 (DMA_INT) 中断源识别	0xFFFF_XXXX
IRQ27_SRC	INT_BA+0x6C	R	MCU IRQ27 (I2S_INT) 中断源识别	0xFFFF_XXXX
IRQ28_SRC	INT_BA+0x70	R	MCU IRQ28 (PDWU_INT) 中断源识别	0xFFFF_XXXX
IRQ29_SRC	INT_BA+0x74	R	MCU IRQ29 (ADC_INT) 中断源识别	0xFFFF_XXXX
IRQ30_SRC	INT_BA+0x78	R	MCU IRQ30 (DAC_INT) 中断源识别	0xFFFF_XXXX
IRQ31_SRC	INT_BA+0x7C	R	MCU IRQ31 (RTC_INT) 中断源识别	0xFFFF_XXXX
NMI_SEL	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 号识别寄存器	0x0000_0000

5.2.8.2 中断源控制寄存器描述

中断源识别寄存器 (IRQn_SRC)

寄存器	偏移量	R/W	描述	复位后的值
IRQn_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD) 中断源识别	0xXXXX_XXXX
		:	
	INT_BA+0x7C		MCU IRQ31 (RTC) 中断源识别	

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

地址	INT 号	Bits	描述
INT_BA+0x00	0	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = BOD_INT
INT_BA+0x04	1	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = WDT_INT
INT_BA+0x08	2	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = EINT0 - PB.14 上的外部中断 0
INT_BA+0x0C	3	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = EINT1 - PB.15 上的外部中断 1
INT_BA+0x10	4	[2:0]	Bit2 = GPC_INT Bit1 = GPB_INT Bit0 = GPA_INT
INT_BA+0x14	5	[2:0]	Bit2 = GPF_INT Bit1 = GPE_INT Bit0 = GPD_INT

地址	INT 号	Bits	描述
INT_BA+0x18	6	[3:0]	Bit3 = PWM0_CH3_INT Bit2 = PWM0_CH2_INT Bit1 = PWM0_CH1_INT Bit0 = PWM0_CH0_INT
INT_BA+0x1C	7	[3:0]	Bit3 = PWM1_CH3_INT Bit2 = PWM1_CH2_INT Bit1 = PWM1_CH1_INT Bit0 = PWM1_CH0_INT
INT_BA+0x20	8	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR0_INT
INT_BA+0x24	9	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR1_INT
INT_BA+0x28	10	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR2_INT
INT_BA+0x2C	11	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR3_INT
INT_BA+0x30	12	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = UART0_INT
INT_BA+0x34	13	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = UART1_INT
INT_BA+0x38	14	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI0_INT
INT_BA+0x3C	15	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI1_INT
INT_BA+0x40	16	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI2_INT
INT_BA+0x44	17	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = IRC_INT

地址	INT 号	Bits	描述
INT_BA+0x48	18	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2C0_INT
INT_BA+0x4C	19	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2C1_INT
INT_BA+0x50	20	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC2_INT
INT_BA+0x54	21	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC0_INT
INT_BA+0x58	22	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC1_INT
INT_BA+0x5C	23	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = USB_INT
INT_BA+0x64	25	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = LCD_INT
INT_BA+0x68	26	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = DMA_INT
INT_BA+0x6C	27	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2S_INT
INT_BA+0x70	28	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = PD_WU_INT
INT_BA+0x74	29	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = ADC_INT
INT_BA+0x78	30	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = DAC_INT
INT_BA+0x7C	31	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0

地址	INT 号	Bits	描述
			Bit0 = RTC_INT

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
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-				NMI_SEL[4:0]			

Bits	描述	
[31:8]		保留
[7:5]	-	保留
[4:0]	NMI_SEL[4:0]	Cortex-M0 的 NMI 中断可以从中断[31:0]之中选择一个。 NMI_SEL bit[4:0] 用于选择NMI中断源

MCU 中断请求源寄存器 (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 中断请求源寄存器	0x0000_0000

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

Bits	描述	
[31:0]	MCU_IRQ[31:0]	<p>MCU IRQ 源寄存器</p> <p>MCU_IRQ 从外围设备收集所有中断，并向 Cortex-M0 内核产生同步中断，产生此中断的模式有两种，分别是正常模式和测试模式。</p> <p>从每个外围设备收集所有中断和同步这些中断，然后触发 Cortex-M0 中断。</p> <p>MCU_IRQ[n] 为 0 时：置 MCU_IRQ[n] 为 1，Cortex_M0 NVIC[n] 将产生一个中断。</p> <p>MCU_IRQ[n] 为 1 时：（意味着有中断请求），这时置 MCU_IRQ[n] 为 1，将清除中断；置 MCU_IRQ[n] 为 0 则无效。</p>

5.3 系统管理器

5.3.1 概述

系统管理器主要控制电源模式，唤醒源，系统复位和系统内存映射。 也提供关于 产品ID，芯片复位，IP 复位，和多功能管脚控制。 .

5.3.2 特征

- 电源模式和唤醒源
- 系统复位
- 系统内存映射
- 系统管理寄存器：
 - ◆ 产品ID
 - ◆ 芯片和IP 复位
 - ◆ 多功能管脚控制

5.3.3 功能描述

5.3.3.1 电源模式和唤醒源

有几种电源模式，电源模式取决于时钟状态 (ON 或OFF).

时钟:

- 外部32.768 KHz 低速晶振 (LXT)
- 外部4~ 24MHz 高速晶振 (HXT)
- 内部RC 12MHz 高速振荡器时钟(HIRC)
- 内部10KHz 低速振荡器时钟 (LIRC)

电源模式:

- 正常模式: CPU 正常工作且所有时钟ON。
- 空闲模式: CPU 进入睡眠模式，CPU 时钟停止且其他时钟 ON。
- 掉电模式: 除LXT、LIRC和SRAM 保留，所有时钟停止,。

芯片进入掉电模式后，下表的唤醒源可以唤醒芯片到正常模式，并列各外设再次进入掉电模式的条件

唤醒源	唤醒条件	系统可再次进入掉电模式之条件
External	-	在软件写 1 清除 GPIOx_ISRC 位之后
Interrupts		
UART	Data 唤醒	唤醒后可立即进入
	CTS _n 唤醒	唤醒后经 2 个 UART_CLK
GPIO	-	在软件写 1 清除 GPIOx_ISRC 位之后
RTC	-	唤醒后经 1 个 RTC_CLK (约 30us)

USB	-	唤醒后可立即进入
SPI	-	在 SPI 从时钟由高电平转为低电平后
Timer	TMRx_ISR[TCAP_IS]	在软件设定定时器 TMRx_ISR[SW_RST](软件复位)或软件写 1 清除 TMRx_ISR[TCAP_IS]之后
	TMRx_ISR[TMR_IS]	在软件设定定时器 TMRx_ISR[SW_RST](软件复位)或软件写 1 清除 TMRx_ISR[TMR_IS]之后
WDT	-	唤醒后可立即进入
BOD	-	在电位高于目标板后或 BOFx_INT_EN 设为低
I2C	-	唤醒后可立即进入

表 5-4 再次进入掉电模式的条件

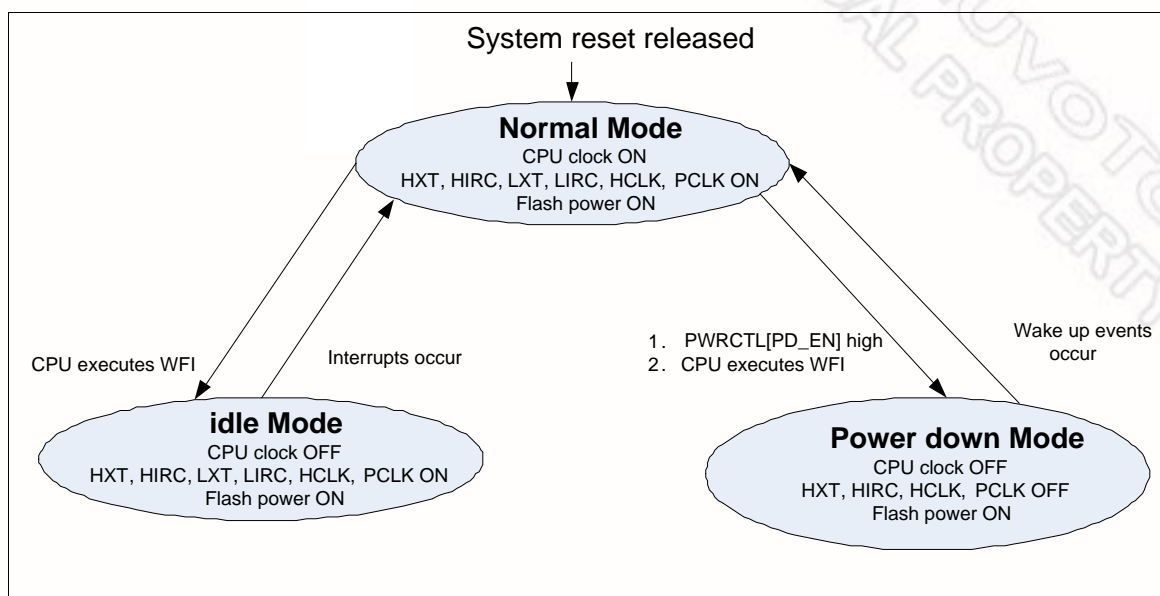


图 5-1 电源模式

寄存器	正常模式	空闲模式	掉电模式
唤醒时间	-	-	7us: 自唤醒事件发生至CPU核第一个有效时钟 10us: 自中断事件发生至中断服务程序第一指令
HXT (4~24 MHz XTL)	ON	ON	-
HIRC (12 MHz OSC)	ON	ON	-
LXT (32 kHz XTL)	ON	ON	ON/OFF ¹
LIRC (10 kHz OSC)	ON	ON	ON/OFF ²
PLL	ON	ON	-
LDO	ON	ON	ON

CPU	ON	-	-
HCLK/PCLK	ON	ON	-
SRAM retention	ON	ON	ON
FLASH ³	ON	ON	ON
GPIO	ON	ON	-
DMA	ON	ON	-
I ² C	ON	ON	-
PWM4	ON	ON	--
TIMER ⁵	ON	ON	--
UART	ON	ON	-
SPI	ON	ON	-
RTC	ON	ON	ON/OFF ⁴
WDT	ON	ON	ON/OFF ⁵
USB	ON	ON	-
LCD	ON	ON	ON/OFF ⁶
I ² S	ON	ON	-
ADC	ON	ON	-
DAC	ON	ON	-

表 5-5 在电源模式的 IP 时钟 ON/OFF

1. LXT (32KHz XTL) ON 或OFF取决于SW设置的运行模式。
2. LIRC (10KHz OSC) ON 或OFF取决于SW设置的运行模式。
3. 如果32KHz XTL 是ON, RTC 可以工作。
4. 如果10K HzOSC 是ON, WDT 可以工作。
5. 如果32KHz XTL 是ON, LCD可以工作。

注意: 如果CPU 时钟没有选择HIRC (12 MHz RC 振荡器), 在进入掉电模式之前用户一定要使能HIRC以避免唤醒失败。掉电电路在系统进入掉电模式时将自动关闭HIRC。在系统唤醒之后, 如果不需要HIRC, 用户可以再次关闭 HIRC以降低功耗。

5.3.3.2 系统复位

系统复位包括下面所列的事件, 可以从 RST_SRC 寄存器读这些复位事件。

上电复位 (POR)

欠压复位(BOD)

/RESET 引脚的低电平复位

看门狗定时器溢出复位

Cortex-M0 MCU 复位

5.3.3.3 自动修正

该芯片支持自动修正功能：HIRC修正(12 MHz RC 振荡器)根据精确的LXT (32.768 kHz 晶体振荡器)自动得到准确的HIRC输出频率，在全温度范围内0.25%的偏差。例如，在USB应用中，PLL需要一个准确的12 MHz输入时钟来输出48 MHz的时钟。在这个例子中，如果用户不想使用12 MHz HXT作为PLL输入时钟，那么需要焊接32.768 kHz晶振在系统中，并通过设置HIRCTRMCTL[TRIM_SEL]为“10”设置HIRC目标输出频率为12 MHz，并且自动修正功能打开。推荐设置TRIM_LOOP和TRIM_RETRY_CNT都为“11”来得到更好的结果。

5.3.4 寄存器映射

写保护寄存器的解锁顺序

一些系统控制寄存器需要被保护以避免不注意的写入错误干扰芯片的正常工作。这些系统控制寄存器在上电之后是被锁着的，除非用户把它打开。用户要写这些被保护的寄存器需要一个开锁的顺序。开锁的顺序是连续的写入数据“59h”，“16h”“88h”到解锁控制器地址0x5000_0100。任何不同的数据值或不同的顺序或者在写入这三个数据期间有任何其他的操作都会破坏整个的解锁过程。然而用户不需考虑写入这三笔数据之间的时间，仅需依序写入即可。

锁被打开之后，用户可以检查锁的状态位RegLockAddr[0]。“1”是解锁状态，“0”是锁状态。之后用户可以更新目标寄存器的值。如果RegUnLock是1，写任何数据到地址“0x5000_0100”可以重新锁住受保护的寄存器。

寄存器	偏移量	R/W	描述	复位后的值
GCR_BA = 0x5000_0000				
PDID	GCR_BA+0x00	R	器件 ID 寄存器	0x0014_0018[1]
RST_SRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX
IPRST_CTL1	GCR_BA+0x08	R/W	外设复位控制寄存器1	0x0000_0000
IPRST_CTL2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000
TEMPCTL	GCR_BA+0x20	R/W	温度传感器控制寄存器	0x0000_0000
PA_L_MFP	GCR_BA+0x30	R/W	端口A 低字节 多功能控制寄存器	0x0000_0000
PA_H_MFP	GCR_BA+0x34	R/W	端口A 高字节 多功能控制寄存器	0x0000_0000
PB_L_MFP	GCR_BA+0x38	R/W	端口B低字节 多功能控制寄存器	0x0000_0000
PB_H_MFP	GCR_BA+0x3C	R/W	端口B高字节 多功能控制寄存器	0x0000_0000
PC_L_MFP	GCR_BA+0x40	R/W	端口C低字节 多功能控制寄存器	0x0000_0000
PC_H_MFP	GCR_BA+0x44	R/W	端口C高字节 多功能控制寄存器	0x0000_0000
PD_L_MFP	GCR_BA+0x48	R/W	端口D低字节 多功能控制寄存器	0x0000_0000
PD_H_MFP	GCR_BA+0x4C	R/W	端口D高字节 多功能控制寄存器	0x0000_0000

PE_L_MFP	GCR_BA+0x50	R/W	端口E低字节 多功能控制寄存器	0x0000_0000
PE_H_MFP	GCR_BA+0x54	R/W	端口E高字节 多功能控制寄存器	0x0000_0000
PF_L_MFP	GCR_BA+0x58	R/W	端口F低字节 多功能控制寄存器	0x0077_7777
PORCTL	GCR_BA+0x60	R/W	上电复位控制寄存器	0x0000_0000
BODCTL	GCR_BA+0x64	R/W	欠压检测控制寄存器	0x00FF_F0xx
BODSTS	GCR_BA+0x68	R	欠压检测状态寄存器	0x0000_0000
Int_VREFCTL	GCR_BA+0x6C	R/W	参考电压控制寄存器	0x0000_0F00
IRCTRMCTL	GCR_BA+0x80	R/W	HIRC 修正控制寄存器	0x0000_0000
IRCTRMIEN	GCR_BA+0x84	R/W	HIRC 修正中断使能寄存器	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC修正中断状态寄存器	0x0000_0000
RegLockAddr	GCR_BA+0x100	R/W	寄存器解锁地址寄存器	0x0000_0000

5.3.5 寄存器描述

器件 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	器件 ID 寄存器	0x0014_0018[1]

[1] 每个器件独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							

Bits	描述	
[31:0]	PDID[31:0]	<p>产品器件识别码</p> <p>该寄存器反映器件的器件号码。软件可以读该寄存器来识别所使用的器件。</p>

系统复位源寄存器(RST_SRC)

该寄存器提供一些特殊信息用于软件识别引起芯片上次复位操作的复位源。

寄存器	偏移量	R/W	描述	复位后的值
RST_SRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00xx

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

Bits	描述	
[31:8]	-	保留
[7]	RSTS_CPU	RSTS_CPU 标志位通过硬件来设置 如果软件写“1”到 CPU_RST (IPRST_CTL1[1]) 会复位Cortex-M0 CPU的核和 Flash 控制器(FMC)。. 1 = Cortex-M0 CPU 内核与 FMC 因为软件置 CPU_RST 为 1 而复位。 0 = CPU 无复位 向该位写 1 清零
[6]	-	保留
[5]	RSTS_SYS	RSTS_SYS 标志位由来自 Cortex-M0 核的“复位信号”置位，用于表示导致之前复位的复位源。 1= 通过软件写1到SYSRESTREQ(AIRCR[2], Cortex_M0发生了复位信号使系统复位, 0 = Cortex-M0 无复位 向该位写 1 清零
[4]	RSTS_BOD	RSTS_BOD 标志位由欠压检测模块的“复位信号”置位，用于表示导致之前复位的复位源。 1 = 欠压检测模块发出复位信号使系统复位。 0 = BOD 无复位 向该位写 1 清零
[3]	-	保留
[2]	RSTS_WDT	RSTS_WDT 标志位由看门狗模块的“复位信号”置位，用于表示导致之前复位的复位源。 1 = 看门狗模块发出复位信号使系统复位。

Bits	描述	
		0 = 看门狗无复位 向该位写 1 清零
[1]	RSTS_PAD	RSTS_PAD 标志位由RESET脚的复位信号来置位，用于表示导致之前复位的复位源。 1= RESET脚发生复位信号，使系统复位。 0=RESET脚没有复位信号 向该位写 1 清零
[0]	RSTS_POR	RSTS_POR 标志位由上电复位模块(POR) 或 CHIP_RST (IPRSTC1[0])的复位信号来置位，用于表示导致之前复位的复位源。 1= 上电复位 (POR) 或CHIP_RST 发生复位信号使系统复位 0= POR 或CHIP_RST没有复位信号 向该位写 1 清零

IP 复位控制寄存器1 (IPRST_CTL1)

寄存器	偏移量	R/W	描述	复位后的值
IPRST_CTL1	GCR_BA+0x08	R/W	IP 复位控制寄存器 1	0x0000_0000

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

Bits	描述	
[31:4]	-	保留
[3]	EBI_RST	EBI 控制器复位 （写保护位） 向该位置1会产生EBI复位信号. 软件需要向该位置0来释放复位信号 0= 正常工作 1= EBI IP 复位
[2]	DMA_RST	DMA 控制器复位 （写保护位） 向该位置1会产生DMA复位信号. 软件需要向该位置0来释放复位信号 0= 正常工作 1= DMA IP 复位
[1]	CPU_RST	CPU 内核复位 （写保护位） 设置该位仅复位 CPU 内核和Flash 存储控制器 (FMC)，该位将在2个时钟周期后自动清零 0= 正常工作 1= CPU 复位
[0]	CHIP_RST	CHIP 复位. （写保护位） 设置该位复位整个芯片，包括CPU 内核和所有外设，该位将在2个时钟周期后自动清零。 CHIP_RST 与上电复位 (POR) 一样，所有芯片控制器都复位，芯片设置从 flash 重新加载。 0= 正常工作 1= CHIP 复位 注意: 在下列条件下，芯片设置从 flash 重新加载

Bits	描述	
		上电复位 欠压检测复位 复位脚低电平 置位 IPRST_CTL1[CHIP_RST]

IP 复位控制寄存器2 (IPRST_CTL2)

寄存器	偏移量	R/W	描述	复位后的值
IPRST_CTL2	GCR_BA+0x0C	R/W	IP复位控制寄存器 2	0x0000_0000

31	30	29	28	27	26	25	24
SC1_RST	SC0_RST	I2S_RST	ADC_RST	USBD_RST	LCD_RST	DAC_RST	-
23	22	21	20	19	18	17	16
-	-	PWM1_RST	PWM0_RST	-	-	UART1_RST	UART0_RST
15	14	13	12	11	10	9	8
-	SPI2_RST	SPI1_RST	SPI0_RST	-	-	I2C1_RST	I2C0_RST
7	6	5	4	3	2	1	0
SC2_RST	-	TMR3_RST	TMR2_RST	TMR1_RST	TMR0_RST	GPIO_RST	-

Bits	描述	
[31]	SC1_RST	SC 1 控制器复位 0= SC1 控制器正常工作 1= SC1 控制器复位
[30]	SC0_RST	SC 0 控制器复位 0= SC0 控制器正常工作 1= SC0 控制器复位
[29]	I2S_RST	I2S 控制器复位 0= I2S控制器正常工作 1= I2S控制器复位
[28]	ADC_RST	ADC控制器复位 0= ADC 控制器正常工作 1= ADC 控制器复位
[27]	USBD_RST	USB 设备控制器复位 0= USB控制器正常工作 1= USB控制器复位
[26]	LCD_RST	LCD 控制器复位 0= LCD USB控制器正常工作 1= LCD控制器复位
[25]	DAC_RST	DAC控制器复位 0= DAC控制器正常工作

Bits	描述	
		1= DAC控制器复位
[24]	-	保留
[23:22]	-	保留
[21]	PWM1_RST	PWM1 控制器复位 0= PWM1控制器正常工作 1= PWM1控制器复位
[20]	PWM0_RST	PWM0控制器复位 0= PWM0控制器正常工作 1= PWM0控制器复位
[19:18]	-	保留
[17]	UART1_RST	UART1控制器复位 0= UART1控制器正常工作 1= UART1控制器复位
[16]	UART0_RST	UART0控制器复位 0= UART0控制器正常工作 1= UART0控制器复位
[15]	-	保留
[14]	SPI2_RST	SPI2控制器复位 0= SPI2控制器正常工作 1= SPI2控制器复位
[13]	SPI1_RST	SPI1控制器复位 0= SPI1控制器正常工作 1= SPI1控制器复位
[12]	SPI0_RST	SPI0控制器复位 0= SPI0控制器正常工作 1= SPI0控制器复位
[11:10]	-	保留
[9]	I2C1_RST	I²C1控制器复位 0= I ² C1控制器正常工作 1= I ² C1控制器复位
[8]	I2C0_RST	I²C0控制器复位 0= I ² C0控制器正常工作 1= I ² C0控制器复位
[7]	SC2_RST	SmartCard 2 控制器复位 0= SmartCard 2控制器正常工作

Bits	描述	
		1= SmartCard 2控制器复位
[6]	-	保留
[5]	TMR3_RST	Timer3控制器复位 0= Timer3控制器正常工作 1= Timer3控制器复位
[4]	TMR2_RST	Timer2控制器复位 0= Timer2控制器正常工作 1= Timer2控制器复位
[3]	TMR1_RST	Timer1控制器复位 0= Timer1控制器正常工作 1= Timer1控制器复位
[2]	TMR0_RST	Timer0控制器复位 0= Timer0控制器正常工作 1= Timer0控制器复位
[1]	GPIO_RST	GPIO控制器复位 0= GPIO控制器正常工作 1= GPIO控制器复位
[0]	-	保留

温度传感器控制寄存器(TEMPCTL)

寄存器	偏移量	R/W	描述	复位后的值
TEMPCTL	GCR_BA+0x20	R/W	温度传感器控制寄存器	0x0000_0000

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

Bits	描述	
[31:1]		保留
[0]	VTEMP_EN	温度传感器使能 1= 使能温度传感器功能 0=禁止温度传感器功能 (默认)

多功能端口 A 低字节控制寄存器(PA_L_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PA_L_MFP	GCR_BA+0x30	R/W	多功能端口 A 低字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PA7_MFP			-	PA6_MFP		
23	22	21	20	19	18	17	16
-	PA5_MFP			-	PA4_MFP		
15	14	13	12	11	10	9	8
-	PA3_MFP			-	PA2_MFP		
7	6	5	4	3	2	1	0
-	PA1_MFP			-	PA0_MFP		

Bits	描述																	
[31]	-	保留																
[30:28]	PA7_MFP[2:0]	PA.7功能选择 LQFP-128封装:																
		<table><tr><th>PA7_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 36</td></tr><tr><td>101</td><td>PWM0通道2</td></tr><tr><td>100</td><td>SmartCard 2 数据管脚</td></tr><tr><td>011</td><td>Timer 2 捕捉事件</td></tr><tr><td>010</td><td>EBI XD[6]</td></tr><tr><td>001</td><td>ADC 输入通道7</td></tr><tr><td>其他</td><td>GPIOA[7]</td></tr></table>	PA7_MFP	功能	111	LCD SEG 36	101	PWM0通道2	100	SmartCard 2 数据管脚	011	Timer 2 捕捉事件	010	EBI XD[6]	001	ADC 输入通道7	其他	GPIOA[7]
		PA7_MFP	功能															
		111	LCD SEG 36															
		101	PWM0通道2															
		100	SmartCard 2 数据管脚															
		011	Timer 2 捕捉事件															
		010	EBI XD[6]															
		001	ADC 输入通道7															
		其他	GPIOA[7]															
LQFP-64封装: 无																		
LQFP-48封装: 无																		
[27]	-	保留																

Bits	描述																	
[26:24]	PA6_MFP[2:0]	PA.6功能选择 LQFP-128封装:																
		<table><tr><th>PA6_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 37</td></tr><tr><td>101</td><td>PWM0通道3</td></tr><tr><td>100</td><td>SmartCard 2 时钟</td></tr><tr><td>011</td><td>Timer 3 捕捉功能</td></tr><tr><td>010</td><td>EBI XD[7]</td></tr><tr><td>001</td><td>ADC 输入通道 6</td></tr><tr><td>其他</td><td>GPIOA[6]</td></tr></table>	PA6_MFP	功能	111	LCD SEG 37	101	PWM0通道3	100	SmartCard 2 时钟	011	Timer 3 捕捉功能	010	EBI XD[7]	001	ADC 输入通道 6	其他	GPIOA[6]
		PA6_MFP	功能															
		111	LCD SEG 37															
		101	PWM0通道3															
		100	SmartCard 2 时钟															
		011	Timer 3 捕捉功能															
		010	EBI XD[7]															
		001	ADC 输入通道 6															
		其他	GPIOA[6]															
		LQFP-64封装:																
		<table><tr><th>PA6_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 19</td></tr><tr><td>101</td><td>PWM0通道3</td></tr><tr><td>100</td><td>SmartCard 2 时钟</td></tr><tr><td>011</td><td>Timer 3 捕捉功能</td></tr><tr><td>010</td><td>EBI XD[7]</td></tr><tr><td>001</td><td>ADC 输入通道 6</td></tr><tr><td>其他</td><td>GPIOA[6]</td></tr></table>	PA6_MFP	功能	111	LCD SEG 19	101	PWM0通道3	100	SmartCard 2 时钟	011	Timer 3 捕捉功能	010	EBI XD[7]	001	ADC 输入通道 6	其他	GPIOA[6]
		PA6_MFP	功能															
		111	LCD SEG 19															
		101	PWM0通道3															
		100	SmartCard 2 时钟															
		011	Timer 3 捕捉功能															
		010	EBI XD[7]															
		001	ADC 输入通道 6															
		其他	GPIOA[6]															
		LQFP-48封装:																
		<table><tr><th>PA6_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0通道3</td></tr><tr><td>100</td><td>SmartCard 2 时钟</td></tr><tr><td>011</td><td>Timer 3 捕捉功能</td></tr><tr><td>010</td><td>EBI XD[7]</td></tr><tr><td>001</td><td>ADC 输入通道 6</td></tr><tr><td>其他</td><td>GPIOA[6]</td></tr></table>	PA6_MFP	功能	101	PWM0通道3	100	SmartCard 2 时钟	011	Timer 3 捕捉功能	010	EBI XD[7]	001	ADC 输入通道 6	其他	GPIOA[6]		
		PA6_MFP	功能															
		101	PWM0通道3															
		100	SmartCard 2 时钟															
011	Timer 3 捕捉功能																	
010	EBI XD[7]																	
001	ADC 输入通道 6																	
其他	GPIOA[6]																	
[23]	-	保留																
[22:20]	PA5_MFP[2:0]	PA.5功能选择 LQFP-128封装:																
		<table><tr><th>PA5_MFP</th><th>功能</th></tr></table>	PA5_MFP	功能														
PA5_MFP	功能																	

Bits	描述	
		111
		101
		100
		010
		001
		其他
		LCD SEG 38
		I ² C0 SCL
		SmartCard2 RST
		EBI AD[8]
		ADC 输入通道 5
		GPIOA[5]
		LQFP-64封装:
		PA5_MFP
		功能
		111
		101
		100
		010
		001
		其他
		LCD SEG 20
		I ² C0 SCL
		SmartCard2 RST
		EBI AD[8]
		ADC输入通道5
		GPIOA[5]
		LQFP-48封装:
		PA5_MFP
		功能
		101
		100
		010
		001
		其他
		I ² C0 SCL
		SmartCard2 RST
		EBI AD[8]
		ADC 输入通道5
		GPIOA[5]
[19]	-	保留
[18:16]	PA4_MFP[2:0]	PA.4 功能选择
		LQFP-128封装:
		PA4_MFP
		功能
		111
		101
		100
		010
		001
		其他
		LCD SEG 39
		I ² C0 SDA
		SmartCard 2 电源
		EBI AD[9]
		ADC输入通道 4
		GPIOA[4]

Bits	描述																											
		<div>LQFP-64 封装:</div> <table><tr><th>PA4_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 21</td></tr><tr><td>101</td><td>I²C0 SDA</td></tr><tr><td>100</td><td>SmartCard 2 电源</td></tr><tr><td>010</td><td>EBI AD[9]</td></tr><tr><td>001</td><td>ADC 输入通道 4</td></tr><tr><td>其他</td><td>GPIOA[4]</td></tr></table> <div>LQFP-48 封装:</div> <table><tr><th>PA4_MFP</th><th>功能</th></tr><tr><td>101</td><td>I²C0SDA</td></tr><tr><td>100</td><td>SmartCard 2 电源</td></tr><tr><td>010</td><td>EBI AD[9]</td></tr><tr><td>001</td><td>ADC输入通道 4</td></tr><tr><td>其他</td><td>GPIOA[4]</td></tr></table>	PA4_MFP	功能	111	LCD SEG 21	101	I ² C0 SDA	100	SmartCard 2 电源	010	EBI AD[9]	001	ADC 输入通道 4	其他	GPIOA[4]	PA4_MFP	功能	101	I ² C0SDA	100	SmartCard 2 电源	010	EBI AD[9]	001	ADC输入通道 4	其他	GPIOA[4]
PA4_MFP	功能																											
111	LCD SEG 21																											
101	I ² C0 SDA																											
100	SmartCard 2 电源																											
010	EBI AD[9]																											
001	ADC 输入通道 4																											
其他	GPIOA[4]																											
PA4_MFP	功能																											
101	I ² C0SDA																											
100	SmartCard 2 电源																											
010	EBI AD[9]																											
001	ADC输入通道 4																											
其他	GPIOA[4]																											
[15]	-	保留																										
		<div>PA.3 功能选择</div> <div>LQFP-128封装:</div> <table><tr><th>PA3_MFP</th><th>功能</th></tr><tr><td>101</td><td>UART1_TXD</td></tr><tr><td>010</td><td>EBI AD[10]</td></tr><tr><td>001</td><td>ADC 输入通道 3</td></tr><tr><td>Others</td><td>GPIOA[3]</td></tr></table> <div>LQFP-64封装:</div> <table><tr><th>PA3_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 22</td></tr><tr><td>101</td><td>UART1_TXD</td></tr><tr><td>010</td><td>EBI AD[10]</td></tr><tr><td>001</td><td>ADC输入通道3</td></tr><tr><td>Others</td><td>GPIOA[3]</td></tr></table> <div>LQFP-48封装:</div>	PA3_MFP	功能	101	UART1_TXD	010	EBI AD[10]	001	ADC 输入通道 3	Others	GPIOA[3]	PA3_MFP	功能	111	LCD SEG 22	101	UART1_TXD	010	EBI AD[10]	001	ADC输入通道3	Others	GPIOA[3]				
PA3_MFP	功能																											
101	UART1_TXD																											
010	EBI AD[10]																											
001	ADC 输入通道 3																											
Others	GPIOA[3]																											
PA3_MFP	功能																											
111	LCD SEG 22																											
101	UART1_TXD																											
010	EBI AD[10]																											
001	ADC输入通道3																											
Others	GPIOA[3]																											
[14:12]	PA3_MFP[2:0]																											

Bits	描述				
		PA3_MFP	功能		
		101	UART1_TXD		
		010	EBI AD[10]		
		001	ADC输入通道 3		
		其他	GPIOA[3]		
[11]	-	保留			
[10:8]	PA2_MFP[2:0]	PA.2功能选择			
		LQFP-128封装:			
		PA2_MFP	功能		
		101	UART1_RXD		
		010	EBI AD[11]		
		001	ADC输入通道2		
		其他	GPIOA[2]		
		LQFP-64封装:			
		PA2_MFP	功能		
		111	LCD SEG 23		
		101	UART1_RXD		
		010	EBI AD[11]		
		001	ADC输入通道2		
		其他	GPIOA[2]		
		LQFP-48封装:			
		PA2_MFP	功能		
		101	UART1_RXD		
		010	EBI AD[11]		
		001	ADC输入通道2		
		其他	GPIOA[2]		
		[7]	-	保留	
		[6:4]	PA1_MFP[2:0]	PA.1功能选择	
				LQFP-128封装:	
				PA1_MFP	功能
				010	EBI AD[12]

Bits	描述									
		<table><tr><td>001</td><td>ADC 输入通道 1</td></tr><tr><td>其他</td><td>GPIOA[1]</td></tr></table>	001	ADC 输入通道 1	其他	GPIOA[1]				
		001	ADC 输入通道 1							
		其他	GPIOA[1]							
		LQFP-64封装:								
		<table><tr><td>PA1_MFP</td><td>功能</td></tr><tr><td>010</td><td>EBI AD[12]</td></tr><tr><td>001</td><td>ADC 输入通道 1</td></tr><tr><td>其他</td><td>GPIOA[1]</td></tr></table>	PA1_MFP	功能	010	EBI AD[12]	001	ADC 输入通道 1	其他	GPIOA[1]
		PA1_MFP	功能							
		010	EBI AD[12]							
		001	ADC 输入通道 1							
		其他	GPIOA[1]							
		LQFP-48封装:								
		<table><tr><td>PA1_MFP</td><td>功能</td></tr><tr><td>010</td><td>EBI AD[12]</td></tr><tr><td>001</td><td>ADC 输入通道1</td></tr><tr><td>其他</td><td>GPIOA[1]</td></tr></table>	PA1_MFP	功能	010	EBI AD[12]	001	ADC 输入通道1	其他	GPIOA[1]
		PA1_MFP	功能							
		010	EBI AD[12]							
		001	ADC 输入通道1							
其他	GPIOA[1]									
[3]	-	保留								
[2:0]	PA0_MFP[2:0]	PA.0功能选择								
		LQFP-128封装:								
		<table><tr><td>PA0_MFP</td><td>功能</td></tr><tr><td>100</td><td>SmartCard 2 card detect</td></tr><tr><td>001</td><td>ADC 输入通道0</td></tr><tr><td>其他</td><td>GPIOA[0]</td></tr></table>	PA0_MFP	功能	100	SmartCard 2 card detect	001	ADC 输入通道0	其他	GPIOA[0]
		PA0_MFP	功能							
		100	SmartCard 2 card detect							
		001	ADC 输入通道0							
		其他	GPIOA[0]							
		LQFP-64封装:								
		<table><tr><td>PA0_MFP</td><td>功能</td></tr><tr><td>100</td><td>SmartCard 2 card detect</td></tr><tr><td>001</td><td>ADC输入通道0</td></tr><tr><td>其他</td><td>GPIOA[0]</td></tr></table>	PA0_MFP	功能	100	SmartCard 2 card detect	001	ADC输入通道0	其他	GPIOA[0]
		PA0_MFP	功能							
		100	SmartCard 2 card detect							
		001	ADC输入通道0							
		其他	GPIOA[0]							
		LQFP-48封装:								
		<table><tr><td>PA0_MFP</td><td>功能</td></tr><tr><td>100</td><td>SmartCard 2 card detect</td></tr><tr><td>001</td><td>ADC输入通道 0</td></tr><tr><td>其他</td><td>GPIOA[0]</td></tr></table>	PA0_MFP	功能	100	SmartCard 2 card detect	001	ADC输入通道 0	其他	GPIOA[0]
		PA0_MFP	功能							
100	SmartCard 2 card detect									
001	ADC输入通道 0									
其他	GPIOA[0]									

多功能端口A 高字节 控制寄存器(PA_H_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PA_H_MFP	GCR_BA+0x34	R/W	多功能端口 A 高字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
保留	PA15_MFP			保留	PA14_MFP		
23	22	21	20	19	18	17	16
保留	PA13_MFP			保留	PA12_MFP		
15	14	13	12	11	10	9	8
保留	PA11_MFP			保留	PA10_MFP		
7	6	5	4	3	2	1	0
保留	PA9_MFP			保留	PA8_MFP		

Bits	描述															
[31]	-	保留.														
[30:28]	PA15_MFP[2:0]	PA.15功能选择														
		LQFP-128封装:														
		<table><tr><th>PA15_MFP</th><th>功能</th></tr><tr><td>110</td><td>UART0 TX</td></tr><tr><td>100</td><td>SmartCard 0 电源</td></tr><tr><td>011</td><td>Timer3捕捉事件</td></tr><tr><td>010</td><td>I²S MCLK</td></tr><tr><td>001</td><td>PWM0通道3</td></tr><tr><td>其他</td><td>GPIOA[15]</td></tr></table>	PA15_MFP	功能	110	UART0 TX	100	SmartCard 0 电源	011	Timer3捕捉事件	010	I ² S MCLK	001	PWM0通道3	其他	GPIOA[15]
		PA15_MFP	功能													
		110	UART0 TX													
		100	SmartCard 0 电源													
		011	Timer3捕捉事件													
		010	I ² S MCLK													
		001	PWM0通道3													
		其他	GPIOA[15]													
		LQFP-64 封装:														
		<table><tr><th>PA15_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 27</td></tr><tr><td>110</td><td>UART0 TX</td></tr><tr><td>100</td><td>SmartCard 0 电源</td></tr><tr><td>011</td><td>Timer3捕捉事件</td></tr><tr><td>010</td><td>I²S MCLK</td></tr><tr><td>001</td><td>PWM0通道3</td></tr></table>	PA15_MFP	功能	111	LCD SEG 27	110	UART0 TX	100	SmartCard 0 电源	011	Timer3捕捉事件	010	I ² S MCLK	001	PWM0通道3
		PA15_MFP	功能													
		111	LCD SEG 27													
		110	UART0 TX													
		100	SmartCard 0 电源													
011	Timer3捕捉事件															
010	I ² S MCLK															
001	PWM0通道3															

Bits	描述		
		其他	GPIOA[15]
		LQFP-48 封装：	
		PA15_MFP	功能
		110	UART0 TX
		100	SmartCard 0 电源
		011	Timer3 捕捉事件
		010	I ² S MCLK
		001	PWM0 通道 3
		其他	GPIOA[15]
		[27]	-
[26:24]	PA14_MFP[2:0]	PA.14功能选择	
		LQFP-128 封装：	
		PA14_MFP	功能
		110	UART0 RX
		011	Timer2捕捉事件
		010	EBI AD[15]
		001	PWM0通道2
		其他	GPIOA[14]
		LQFP-64 封装：	
		PA14_MFP	功能
		111	LCD SEG 26
		110	UART0 RX
		011	Timer2捕捉事件
		010	EBI AD[15]
		001	PWM0通道2
		其他	GPIOA[14]
		LQFP-48 封装：	
		PA14_MFP	功能
		110	UART0 RX

Bits	描述		
		011	Timer2捕捉事件
		010	EBI AD[15]
		001	PWM0通道2
		其他	GPIOA[14]
[23]	-	保留	
[22:20]	PA13_MFP[2:0]	PA.13功能选择	
		LQFP-128 封装 :	
		PA13_MFP	功能
		101	I ² C0 SCL
		011	Timer1捕捉事件
		010	EBI AD[14]
		001	PWM0通道1
		其他	GPIOA[13]
		LQFP-64 封装 :	
		PA13_MFP	功能
		111	LCD SEG 25
		101	I ² C0 SCL
		011	Timer1捕捉事件
		010	EBI AD[14]
		001	PWM0通道1
		其他	GPIOA[13]
		LQFP-48 封装 :	
		PA13_MFP	功能
		101	I ² C0 SCL
		011	Timer1捕捉事件
		010	EBI AD[14]
		001	PWM0通道1
		其他	GPIOA[13]
[19]	-	保留	

Bits	描述															
[18:16]	PA12_MFP[2:0]	PA.12功能选择 LQFP-128封装：														
		<table><tr><th>PA12_MFP</th><th>功能</th></tr><tr><td>101</td><td>I²C0 SDA</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[13]</td></tr><tr><td>001</td><td>PWM0通道0</td></tr><tr><td>其他</td><td>GPIOA[12]</td></tr></table>	PA12_MFP	功能	101	I ² C0 SDA	011	Timer0捕捉事件	010	EBI AD[13]	001	PWM0通道0	其他	GPIOA[12]		
		PA12_MFP	功能													
		101	I ² C0 SDA													
		011	Timer0捕捉事件													
		010	EBI AD[13]													
		001	PWM0通道0													
		其他	GPIOA[12]													
		LQFP-64 封装：														
		<table><tr><th>PA12_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 24</td></tr><tr><td>101</td><td>I²C0 SDA</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[13]</td></tr><tr><td>001</td><td>PWM0通道0</td></tr><tr><td>其他</td><td>GPIOA[12]</td></tr></table>	PA12_MFP	功能	111	LCD SEG 24	101	I ² C0 SDA	011	Timer0捕捉事件	010	EBI AD[13]	001	PWM0通道0	其他	GPIOA[12]
		PA12_MFP	功能													
		111	LCD SEG 24													
		101	I ² C0 SDA													
		011	Timer0捕捉事件													
		010	EBI AD[13]													
		001	PWM0通道0													
		其他	GPIOA[12]													
		LQFP-48 封装：														
		<table><tr><th>PA12_MFP</th><th>功能</th></tr><tr><td>101</td><td>I²C0 SDA</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[13]</td></tr><tr><td>001</td><td>PWM0通道0</td></tr><tr><td>其他</td><td>GPIOA[12]</td></tr></table>	PA12_MFP	功能	101	I ² C0 SDA	011	Timer0捕捉事件	010	EBI AD[13]	001	PWM0通道0	其他	GPIOA[12]		
		PA12_MFP	功能													
		101	I ² C0 SDA													
011	Timer0捕捉事件															
010	EBI AD[13]															
001	PWM0通道0															
其他	GPIOA[12]															
[15]	-	保留														
[14:12]	PA11_MFP[2:0]	PA.11功能选择 LQFP-128 封装：														
		<table><tr><th>PA11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 23</td></tr><tr><td>100</td><td>SPI2 MOSIO</td></tr><tr><td>011</td><td>SmartCard0 RST</td></tr><tr><td>010</td><td>EBI nRE</td></tr></table>	PA11_MFP	功能	111	LCD SEG 23	100	SPI2 MOSIO	011	SmartCard0 RST	010	EBI nRE				
		PA11_MFP	功能													
		111	LCD SEG 23													
		100	SPI2 MOSIO													
		011	SmartCard0 RST													
		010	EBI nRE													

Bits	描述					
		<table><tr><td>001</td><td>I²C1 SCL</td></tr><tr><td>其他</td><td>GPIOA[11]</td></tr></table>	001	I ² C1 SCL	其他	GPIOA[11]
		001	I ² C1 SCL			
		其他	GPIOA[11]			
		LQFP-64 封装：				
		PA11_MFP	功能			
		111	LCD SEG 9			
		100	SPI2 MOSIO			
		011	SmartCard0 RST			
		010	EBI nRE			
		001	I ² C1 SCL			
		其他	GPIOA[11]			
		LQFP-48 封装：				
		PA11_MFP	功能			
		100	SPI2 MOSIO			
		011	SmartCard0 RST			
		010	EBI nRE			
		001	I ² C1 SCL			
		其他	GPIOA[11]			
		[11]	-	保留		
[10:8]	PA10_MFP[2:0]	PA.10功能选择				
		LQFP-128 封装：				
		PA10_MFP	功能			
		111	LCD SEG 22			
		100	SPI2 MISO0			
		011	SmartCard0 电源			
		010	EBI nWE			
		001	I ² C1 SDA			
		其他	GPIOA[10]			
		LQFP-64 封装：				
		PA10_MFP	功能			
		111	LCD SEG 8			
		100	SPI2 MISO0			

Bits	描述																														
	<table border="1"> <tr> <td>011</td><td>SmartCard0 电源</td></tr> <tr> <td>010</td><td>EBI nWE</td></tr> <tr> <td>001</td><td>I²C1 SDA</td></tr> <tr> <td>其他</td><td>GPIOA[10]</td></tr> </table> <p>LQFP-48 封装：</p> <table border="1"> <tr> <th>PA10_MFP</th><th>功能</th></tr> <tr> <td>100</td><td>SPI2 MISO0</td></tr> <tr> <td>011</td><td>SmartCard0 电源</td></tr> <tr> <td>010</td><td>EBI nWE</td></tr> <tr> <td>001</td><td>I²C1 SDA</td></tr> <tr> <td>其他</td><td>GPIOA[10]</td></tr> </table>	011	SmartCard0 电源	010	EBI nWE	001	I ² C1 SDA	其他	GPIOA[10]	PA10_MFP	功能	100	SPI2 MISO0	011	SmartCard0 电源	010	EBI nWE	001	I ² C1 SDA	其他	GPIOA[10]										
011	SmartCard0 电源																														
010	EBI nWE																														
001	I ² C1 SDA																														
其他	GPIOA[10]																														
PA10_MFP	功能																														
100	SPI2 MISO0																														
011	SmartCard0 电源																														
010	EBI nWE																														
001	I ² C1 SDA																														
其他	GPIOA[10]																														
[7]	- 保留																														
[6:4]	<p>PA.9功能选择</p> <p>LQFP-128 封装：</p> <table border="1"> <tr> <th>PA9_MFP</th><th>功能</th></tr> <tr> <td>111</td><td>LCD SEG 21</td></tr> <tr> <td>100</td><td>SPI2 SCLK</td></tr> <tr> <td>011</td><td>SmartCard0 DATA</td></tr> <tr> <td>001</td><td>I²C0 SCL</td></tr> <tr> <td>其他</td><td>GPIOA[9]</td></tr> </table> <p>LQFP-64 封装：</p> <table border="1"> <tr> <th>PA9_MFP</th><th>功能</th></tr> <tr> <td>111</td><td>LCD SEG 7</td></tr> <tr> <td>100</td><td>SPI2 SCLK</td></tr> <tr> <td>011</td><td>SmartCard0 DATA</td></tr> <tr> <td>001</td><td>I²C0 SCL</td></tr> <tr> <td>其他</td><td>GPIOA[9]</td></tr> </table> <p>LQFP-48 封装：</p> <table border="1"> <tr> <th>PA9_MFP</th><th>功能</th></tr> <tr> <td>100</td><td>SPI2 SCLK</td></tr> <tr> <td>011</td><td>SmartCard0 DATA</td></tr> </table>	PA9_MFP	功能	111	LCD SEG 21	100	SPI2 SCLK	011	SmartCard0 DATA	001	I ² C0 SCL	其他	GPIOA[9]	PA9_MFP	功能	111	LCD SEG 7	100	SPI2 SCLK	011	SmartCard0 DATA	001	I ² C0 SCL	其他	GPIOA[9]	PA9_MFP	功能	100	SPI2 SCLK	011	SmartCard0 DATA
PA9_MFP	功能																														
111	LCD SEG 21																														
100	SPI2 SCLK																														
011	SmartCard0 DATA																														
001	I ² C0 SCL																														
其他	GPIOA[9]																														
PA9_MFP	功能																														
111	LCD SEG 7																														
100	SPI2 SCLK																														
011	SmartCard0 DATA																														
001	I ² C0 SCL																														
其他	GPIOA[9]																														
PA9_MFP	功能																														
100	SPI2 SCLK																														
011	SmartCard0 DATA																														

Bits	描述		
		001	I ² C0 SCL
		其他	GPIOA[9]
[3]	-	保留	
[2:0]	PA8_MFP[2:0]	PA.8功能选择	
		LQFP-128 封装 :	
		PA8_MFP	功能
		111	LCD SEG 20
		100	SPI2 1 st 从机选择管脚
		011	SmartCard0 时钟
		001	I ² C0 SDA
		其他	GPIOA[8]
		LQFP-64 封装 :	
		PA8_MFP	功能
		111	LCD SEG 6
		100	SPI2 1 st 从机选择管脚
		011	SmartCard0时钟
		001	I ² C0 SDA
		其他	GPIOA[8]
		LQFP-48 封装 :	
		PA8_MFP	功能
		100	SPI2 1 st 从机选择管脚
		011	SmartCard0时钟
		001	I ² C0 SDA
		其他	GPIOA[8]

多功能端口 B 低字节控制寄存器(PB_L_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PB_L_MFP	GCR_BA+0x38	R/W	多功能端口 B 低字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PB7_MFP			-	PB6_MFP		
23	22	21	20	19	18	17	16
-	PB5_MFP			-	PB4_MFP		
15	14	13	12	11	10	9	8
-	PB3_MFP			-	PB2_MFP		
7	6	5	4	3	2	1	0
-	PB1_MFP			-	PB0_MFP		

Bits	描述		
[31]	-	保留	
[30:28]	PB7_MFP[2:0]	PB.7 功能选择	
		LQFP-128封装：	
		PB7_MFP	功能
		111	LCD SEG 10
		100	SPI2 MOSIO
		010	EBI nCS
		001	UART1 CTSn
		其他	GPIOB[7]
		LQFP-64 封装：	
		PB7_MFP	功能
		111	LCD SEG 2
		100	SPI2 MOSIO
		010	EBI nCS
		001	UART1 CTSn
		其他	GPIOB[7]
		LQFP-48 封装：	
		无	

Bits	描述													
[27]	-	保留												
[26:24]	PB6_MFP[2:0]	PB.6功能选择 LQFP-128 封装：												
		<table><tr><th>PB6_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 11</td></tr><tr><td>100</td><td>SPI2 MISO0</td></tr><tr><td>010</td><td>EBI ALE</td></tr><tr><td>001</td><td>UART1 RTSn</td></tr><tr><td>其他</td><td>GPIOB[6]</td></tr></table>	PB6_MFP	功能	111	LCD SEG 11	100	SPI2 MISO0	010	EBI ALE	001	UART1 RTSn	其他	GPIOB[6]
		PB6_MFP	功能											
		111	LCD SEG 11											
		100	SPI2 MISO0											
		010	EBI ALE											
		001	UART1 RTSn											
		其他	GPIOB[6]											
		LQFP-64 封装：												
		<table><tr><th>PB6_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 3</td></tr><tr><td>100</td><td>SPI2 MISO0</td></tr><tr><td>010</td><td>EBI ALE</td></tr><tr><td>001</td><td>UART1 RTSn</td></tr><tr><td>其他</td><td>GPIOB[6]</td></tr></table>	PB6_MFP	功能	111	LCD SEG 3	100	SPI2 MISO0	010	EBI ALE	001	UART1 RTSn	其他	GPIOB[6]
		PB6_MFP	功能											
		111	LCD SEG 3											
		100	SPI2 MISO0											
		010	EBI ALE											
		001	UART1 RTSn											
		其他	GPIOB[6]											
		LQFP-48 封装：												
		无.												
[23]	-	保留												
[22:20]	PB5_MFP[2:0]	PB.5功能选择 LQFP-128 封装：												
		<table><tr><th>PB5_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 12</td></tr><tr><td>100</td><td>SPI2 SCLK</td></tr><tr><td>011</td><td>SmartCard0 RST</td></tr><tr><td>001</td><td>UART1 TX</td></tr><tr><td>其他</td><td>GPIOB[5]</td></tr></table>	PB5_MFP	功能	111	LCD SEG 12	100	SPI2 SCLK	011	SmartCard0 RST	001	UART1 TX	其他	GPIOB[5]
		PB5_MFP	功能											
		111	LCD SEG 12											
		100	SPI2 SCLK											
		011	SmartCard0 RST											
		001	UART1 TX											
		其他	GPIOB[5]											
		LQFP-64 封装：												
		<table><tr><th>PB5_MFP</th><th>功能</th></tr></table>	PB5_MFP	功能										
		PB5_MFP	功能											

Bits	描述	
		111
		011
		100
		001
		其他
		LCD SEG 4
		SmartCard0 RST
		SPI2 SCLK
		UART1 TX
		GPIOB[5]
		LQFP-48 封装 :
		PB5_MFP
		功能
		011
		100
		001
		其他
		SmartCard0 RST
		SPI2 SCLK
		UART1 TX
		GPIOB[5]
		LQFP-48 封装 :
[19]	-	保留
[18:16]	PB4_MFP[2:0]	PB.4功能选择
		LQFP-128 封装 :
		PB4_MFP
		功能
		111
		100
		011
		001
		其他
		LCD SEG 13
		SPI2 1 st 从机选择管脚
		SmartCard0 卡侦测
		UART1 RX
		GPIOB[4]
		LQFP-64 封装 :
		PB4_MFP
		功能
		111
		100
		011
		001
		其他
		LCD SEG 5
		SPI2 1 st 从机选择管脚
		SmartCard0卡侦测
		UART1 RX
		GPIOB[4]
		LQFP-48 封装 :
		PB4_MFP
		功能
		100
		SPI2 1 st 从机选择管脚

Bits	描述													
		<table><tr><td>011</td><td>SmartCard0卡侦测</td></tr><tr><td>001</td><td>UART1 RX</td></tr><tr><td>其他</td><td>GPIOB[4]</td></tr></table>	011	SmartCard0卡侦测	001	UART1 RX	其他	GPIOB[4]						
011	SmartCard0卡侦测													
001	UART1 RX													
其他	GPIOB[4]													
[15]	-	保留												
[14:12]	PB3_MFP[2:0]	PB.3功能选择 LQFP-128封装：												
		<table><tr><th>PB3_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 4</td></tr><tr><td>011</td><td>SPI1 1st 从机选择管脚</td></tr><tr><td>010</td><td>EBI nWRH</td></tr><tr><td>001</td><td>UART0 CTSn</td></tr><tr><td>其他</td><td>GPIOB[3]</td></tr></table>	PB3_MFP	功能	111	LCD SEG 4	011	SPI1 1 st 从机选择管脚	010	EBI nWRH	001	UART0 CTSn	其他	GPIOB[3]
		PB3_MFP	功能											
		111	LCD SEG 4											
		011	SPI1 1 st 从机选择管脚											
		010	EBI nWRH											
		001	UART0 CTSn											
		其他	GPIOB[3]											
		LQFP-64 封装：												
		<table><tr><th>PB3_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD COM 2</td></tr><tr><td>011</td><td>SPI1 1st从机选择管脚</td></tr><tr><td>010</td><td>EBI nWRH</td></tr><tr><td>001</td><td>UART0 CTSn</td></tr><tr><td>其他</td><td>GPIOB[3]</td></tr></table>	PB3_MFP	功能	111	LCD COM 2	011	SPI1 1 st 从机选择管脚	010	EBI nWRH	001	UART0 CTSn	其他	GPIOB[3]
		PB3_MFP	功能											
		111	LCD COM 2											
		011	SPI1 1 st 从机选择管脚											
		010	EBI nWRH											
		001	UART0 CTSn											
		其他	GPIOB[3]											
		LQFP-48 封装：												
		<table><tr><th>PB3_MFP</th><th>功能</th></tr><tr><td>011</td><td>SPI1 1st 从机选择管脚</td></tr><tr><td>010</td><td>EBI nWRH</td></tr><tr><td>001</td><td>UART0 nCTS</td></tr><tr><td>其他</td><td>GPIOB[3]</td></tr></table>	PB3_MFP	功能	011	SPI1 1 st 从机选择管脚	010	EBI nWRH	001	UART0 nCTS	其他	GPIOB[3]		
		PB3_MFP	功能											
		011	SPI1 1 st 从机选择管脚											
		010	EBI nWRH											
		001	UART0 nCTS											
		其他	GPIOB[3]											

Bits	描述															
		<table><tr><th>PB2_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 5</td></tr><tr><td>011</td><td>SPI1 SCLK</td></tr><tr><td>010</td><td>EBI nWRL</td></tr><tr><td>001</td><td>UART0 RTSn</td></tr><tr><td>其他</td><td>GPIOB[2]</td></tr></table>	PB2_MFP	功能	111	LCD SEG 5	011	SPI1 SCLK	010	EBI nWRL	001	UART0 RTSn	其他	GPIOB[2]		
		PB2_MFP	功能													
		111	LCD SEG 5													
		011	SPI1 SCLK													
		010	EBI nWRL													
		001	UART0 RTSn													
		其他	GPIOB[2]													
		LQFP-64 封装：														
		<table><tr><th>PB2_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD COM 3</td></tr><tr><td>011</td><td>SPI1 SCLK</td></tr><tr><td>010</td><td>EBI nWRL</td></tr><tr><td>001</td><td>UART0 RTSn</td></tr><tr><td>其他</td><td>GPIOB[2]</td></tr></table>	PB2_MFP	功能	111	LCD COM 3	011	SPI1 SCLK	010	EBI nWRL	001	UART0 RTSn	其他	GPIOB[2]		
		PB2_MFP	功能													
		111	LCD COM 3													
		011	SPI1 SCLK													
		010	EBI nWRL													
		001	UART0 RTSn													
		其他	GPIOB[2]													
		LQFP-48 封装：														
		<table><tr><th>PB2_MFP</th><th>功能</th></tr><tr><td>011</td><td>SPI1 SCLK</td></tr><tr><td>010</td><td>EBI nWRL</td></tr><tr><td>001</td><td>UART0 RTSn</td></tr><tr><td>其他</td><td>GPIOB[2]</td></tr></table>	PB2_MFP	功能	011	SPI1 SCLK	010	EBI nWRL	001	UART0 RTSn	其他	GPIOB[2]				
		PB2_MFP	功能													
		011	SPI1 SCLK													
		010	EBI nWRL													
		001	UART0 RTSn													
其他	GPIOB[2]															
[7]	-	保留														
[6:4]	PB1_MFP[2:0]	<table><tr><td colspan="2">PB.1功能选择</td></tr><tr><td colspan="2">LQFP-128 封装：</td></tr><tr><th>PB1_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 6</td></tr><tr><td>011</td><td>SPI1 MISO0</td></tr><tr><td>001</td><td>UART0 TX</td></tr><tr><td>其他</td><td>GPIOB[1]</td></tr></table>	PB.1功能选择		LQFP-128 封装：		PB1_MFP	功能	111	LCD SEG 6	011	SPI1 MISO0	001	UART0 TX	其他	GPIOB[1]
		PB.1功能选择														
		LQFP-128 封装：														
		PB1_MFP	功能													
		111	LCD SEG 6													
		011	SPI1 MISO0													
		001	UART0 TX													
		其他	GPIOB[1]													
		LQFP-64 封装：														
		<table><tr><th>PB1_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 0</td></tr><tr><td>011</td><td>SPI1 MISO0</td></tr></table>	PB1_MFP	功能	111	LCD SEG 0	011	SPI1 MISO0								
		PB1_MFP	功能													
111	LCD SEG 0															
011	SPI1 MISO0															

Bits	描述											
		<table><tr><td>001</td><td>UART0 TX</td></tr><tr><td>其他</td><td>GPIOB[1]</td></tr></table>	001	UART0 TX	其他	GPIOB[1]						
		001	UART0 TX									
		其他	GPIOB[1]									
		LQFP-48 封装：										
		<table><tr><th>PB1_MFP</th><th>功能</th></tr><tr><td>011</td><td>SPI1 MISO0</td></tr><tr><td>001</td><td>UART0 TX</td></tr><tr><td>其他</td><td>GPIOB[1]</td></tr></table>	PB1_MFP	功能	011	SPI1 MISO0	001	UART0 TX	其他	GPIOB[1]		
		PB1_MFP	功能									
		011	SPI1 MISO0									
		001	UART0 TX									
其他	GPIOB[1]											
[3]												
-	保留											
[2:0]	PB0_MFP[2:0]	PB.0功能选择										
		LQFP-128 封装：										
		<table><tr><th>PB0_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 7</td></tr><tr><td>011</td><td>SPI1 MOSI0</td></tr><tr><td>001</td><td>UART0 RX</td></tr><tr><td>其他</td><td>GPIOB[0]</td></tr></table>	PB0_MFP	功能	111	LCD SEG 7	011	SPI1 MOSI0	001	UART0 RX	其他	GPIOB[0]
		PB0_MFP	功能									
		111	LCD SEG 7									
		011	SPI1 MOSI0									
		001	UART0 RX									
		其他	GPIOB[0]									
		LQFP-64 封装：										
		<table><tr><th>PB0_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 1</td></tr><tr><td>011</td><td>SPI1 MOSI0</td></tr><tr><td>001</td><td>UART0 RX</td></tr><tr><td>其他</td><td>GPIOB[0]</td></tr></table>	PB0_MFP	功能	111	LCD SEG 1	011	SPI1 MOSI0	001	UART0 RX	其他	GPIOB[0]
		PB0_MFP	功能									
		111	LCD SEG 1									
		011	SPI1 MOSI0									
		001	UART0 RX									
		其他	GPIOB[0]									
		LQFP-48 封装：										
		<table><tr><th>PB0_MFP</th><th>功能</th></tr><tr><td>011</td><td>SPI1 MOSI0</td></tr><tr><td>001</td><td>UART0 RX</td></tr><tr><td>其他</td><td>GPIOB[0]</td></tr></table>	PB0_MFP	功能	011	SPI1 MOSI0	001	UART0 RX	其他	GPIOB[0]		
		PB0_MFP	功能									
		011	SPI1 MOSI0									
		001	UART0 RX									
其他	GPIOB[0]											

多功能端口 B 高字节控制寄存器(PB_H_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PB_H_MFP	GCR_BA+0x3C	R/W	多功能端口 B 高字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PB15_MFP			-	PB14_MFP		
23	22	21	20	19	18	17	16
-	PB13_MFP			-	PB12_MFP		
15	14	13	12	11	10	9	8
-	PB11_MFP			-	PB10_MFP		
7	6	5	4	3	2	1	0
-	PB9_MFP			-	PB8_MFP		

Bits	描述		
[31]	-	保留	
[30:28]	PB15_MFP[2:0]	PB.15功能选择	
		LQFP-128 封装：	
		PB15_MFP	功能
		111	LCD SEG 31
		100	SmartCard1卡侦测
		011	探听管脚
		001	外部中断1
		其他	GPIOB[15]
		LQFP-64封装：	
		PB15_MFP	功能
		111	LCD SEG 14
		100	SmartCard1卡侦测
		011	探听管脚
		001	外部中断1
		其他	GPIOB[15]
		LQFP-48封装：	

Bits	描述																			
		<table><tr><th>PB15_MFP</th><th>功能</th></tr><tr><td>100</td><td>SmartCard1卡侦测</td></tr><tr><td>011</td><td>探听管脚</td></tr><tr><td>001</td><td>外部中断1</td></tr><tr><td>其他</td><td>GPIOB[15]</td></tr></table>	PB15_MFP	功能	100	SmartCard1卡侦测	011	探听管脚	001	外部中断1	其他	GPIOB[15]								
PB15_MFP	功能																			
100	SmartCard1卡侦测																			
011	探听管脚																			
001	外部中断1																			
其他	GPIOB[15]																			
[27]	-	保留																		
		<table><tr><td colspan="2">PB.14功能选择</td></tr><tr><td colspan="2">LQFP-128封装：</td></tr><tr><th>PB14_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 26</td></tr><tr><td>100</td><td>SPI2 2nd从机选择管脚</td></tr><tr><td>011</td><td>SmartCard 2卡侦测</td></tr><tr><td>001</td><td>外部中断0</td></tr><tr><td>其他</td><td>GPIOB[14]</td></tr></table>	PB.14功能选择		LQFP-128封装：		PB14_MFP	功能	111	LCD SEG 26	100	SPI2 2 nd 从机选择管脚	011	SmartCard 2卡侦测	001	外部中断0	其他	GPIOB[14]		
PB.14功能选择																				
LQFP-128封装：																				
PB14_MFP	功能																			
111	LCD SEG 26																			
100	SPI2 2 nd 从机选择管脚																			
011	SmartCard 2卡侦测																			
001	外部中断0																			
其他	GPIOB[14]																			
[26:24]	PB14_MFP[2:0]	<table><tr><td colspan="2">LQFP-64 封装：</td></tr><tr><th>PB14_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 12</td></tr><tr><td>100</td><td>SPI2 2nd 从机选择管脚</td></tr><tr><td>011</td><td>SmartCard 2卡侦测</td></tr><tr><td>001</td><td>外部中断0</td></tr><tr><td>其他</td><td>GPIOB[14]</td></tr></table> <table><tr><td colspan="2">LQFP-64 封装：</td></tr><tr><td colspan="2">无</td></tr></table>	LQFP-64 封装：		PB14_MFP	功能	111	LCD SEG 12	100	SPI2 2 nd 从机选择管脚	011	SmartCard 2卡侦测	001	外部中断0	其他	GPIOB[14]	LQFP-64 封装：		无	
LQFP-64 封装：																				
PB14_MFP	功能																			
111	LCD SEG 12																			
100	SPI2 2 nd 从机选择管脚																			
011	SmartCard 2卡侦测																			
001	外部中断0																			
其他	GPIOB[14]																			
LQFP-64 封装：																				
无																				
[23]	-	保留																		
		<table><tr><td colspan="2">PB.13功能选择</td></tr><tr><td colspan="2">LQFP-128 封装：</td></tr><tr><th>PB13_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 25</td></tr><tr><td>010</td><td>EBI AD[1]</td></tr><tr><td>其他</td><td>GPIOB[13]</td></tr></table>	PB.13功能选择		LQFP-128 封装：		PB13_MFP	功能	111	LCD SEG 25	010	EBI AD[1]	其他	GPIOB[13]						
PB.13功能选择																				
LQFP-128 封装：																				
PB13_MFP	功能																			
111	LCD SEG 25																			
010	EBI AD[1]																			
其他	GPIOB[13]																			
[22:20]	PB13_MFP[2:0]																			

Bits	描述																													
		<div>LQFP-64 封装：</div> <table><tr><th>PB13_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 11</td></tr><tr><td>010</td><td>EBI AD[1]</td></tr><tr><td>其他</td><td>GPIOB[13]</td></tr></table> <div>LQFP-48 封装：</div> <div>无</div>	PB13_MFP	功能	111	LCD SEG 11	010	EBI AD[1]	其他	GPIOB[13]																				
PB13_MFP	功能																													
111	LCD SEG 11																													
010	EBI AD[1]																													
其他	GPIOB[13]																													
[19]	-	保留																												
		<div>PB.12功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PB12_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 24</td></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>010</td><td>EBI AD[0]</td></tr><tr><td>其他</td><td>GPIOB[12]</td></tr></table> <div>LQFP-64 封装：</div> <table><tr><th>PB12_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 10</td></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>010</td><td>EBI AD[0]</td></tr><tr><td>其他</td><td>GPIOB[12]</td></tr></table> <div>LQFP-48 封装：</div> <table><tr><th>PB12_MFP</th><th>功能</th></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>010</td><td>EBI AD[0]</td></tr><tr><td>其他</td><td>GPIOB[12]</td></tr></table>	PB12_MFP	功能	111	LCD SEG 24	100	FRQDIV_CLK	010	EBI AD[0]	其他	GPIOB[12]	PB12_MFP	功能	111	LCD SEG 10	100	FRQDIV_CLK	010	EBI AD[0]	其他	GPIOB[12]	PB12_MFP	功能	100	FRQDIV_CLK	010	EBI AD[0]	其他	GPIOB[12]
PB12_MFP	功能																													
111	LCD SEG 24																													
100	FRQDIV_CLK																													
010	EBI AD[0]																													
其他	GPIOB[12]																													
PB12_MFP	功能																													
111	LCD SEG 10																													
100	FRQDIV_CLK																													
010	EBI AD[0]																													
其他	GPIOB[12]																													
PB12_MFP	功能																													
100	FRQDIV_CLK																													
010	EBI AD[0]																													
其他	GPIOB[12]																													
[18:16]	PB12_MFP[2:0]																													
		<div>PB.11功能选择</div> <div>LQFP-128 封装：</div>																												
[15]	-	保留																												
		<div>PB.11功能选择</div> <div>LQFP-128 封装：</div>																												
[14:12]	PB11_MFP[2:0]																													

Bits	描述															
		<table><tr><th>PB11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V3</td></tr><tr><td>101</td><td>SPI0 MISO0</td></tr><tr><td>100</td><td>SmartCard 2 DATA</td></tr><tr><td>010</td><td>Timer3 外部事件输入 或是Timer3高低电平输出</td></tr><tr><td>001</td><td>PWM1通道0</td></tr><tr><td>其他</td><td>GPIOB[11]</td></tr></table>	PB11_MFP	功能	111	LCD V3	101	SPI0 MISO0	100	SmartCard 2 DATA	010	Timer3 外部事件输入 或是Timer3高低电平输出	001	PWM1通道0	其他	GPIOB[11]
		PB11_MFP	功能													
		111	LCD V3													
		101	SPI0 MISO0													
		100	SmartCard 2 DATA													
		010	Timer3 外部事件输入 或是Timer3高低电平输出													
		001	PWM1通道0													
		其他	GPIOB[11]													
		LQFP-64 封装：														
		<table><tr><th>PB11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V3</td></tr><tr><td>101</td><td>SPI0 MISO0</td></tr><tr><td>100</td><td>SmartCard 2 DATA</td></tr><tr><td>010</td><td>Timer3 外部事件输入 或是Timer3高低电平输出</td></tr><tr><td>001</td><td>PWM1通道0</td></tr><tr><td>其他</td><td>GPIOB[11]</td></tr></table>	PB11_MFP	功能	111	LCD V3	101	SPI0 MISO0	100	SmartCard 2 DATA	010	Timer3 外部事件输入 或是Timer3高低电平输出	001	PWM1通道0	其他	GPIOB[11]
		PB11_MFP	功能													
		111	LCD V3													
		101	SPI0 MISO0													
		100	SmartCard 2 DATA													
		010	Timer3 外部事件输入 或是Timer3高低电平输出													
		001	PWM1通道0													
		其他	GPIOB[11]													
		LQFP-48 封装：														
		<table><tr><th>PB11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V3</td></tr><tr><td>101</td><td>SPI0 MISO0</td></tr><tr><td>100</td><td>SmartCard 2 DATA</td></tr><tr><td>010</td><td>Timer3 外部事件输入 或是Timer3高低电平输出</td></tr><tr><td>001</td><td>PWM1通道0</td></tr><tr><td>其他</td><td>GPIOB[11]</td></tr></table>	PB11_MFP	功能	111	LCD V3	101	SPI0 MISO0	100	SmartCard 2 DATA	010	Timer3 外部事件输入 或是Timer3高低电平输出	001	PWM1通道0	其他	GPIOB[11]
		PB11_MFP	功能													
		111	LCD V3													
		101	SPI0 MISO0													
		100	SmartCard 2 DATA													
		010	Timer3 外部事件输入 或是Timer3高低电平输出													
001	PWM1通道0															
其他	GPIOB[11]															
[11]	-	保留														
[10:8]	PB10_MFP[2:0]	PB.10功能选择														
		LQFP-128封装：														
		<table><tr><th>PB10_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V2</td></tr><tr><td>101</td><td>SPI0 MOSIO</td></tr></table>	PB10_MFP	功能	111	LCD V2	101	SPI0 MOSIO								
		PB10_MFP	功能													
111	LCD V2															
101	SPI0 MOSIO															

Bits	描述															
		<table><tr><td>100</td><td>SmartCard 2 时钟</td></tr><tr><td>010</td><td>Timer2 外部事件输入 或是Timer2高低电平输出</td></tr><tr><td>001</td><td>SPI0 2nd 从机选择管脚</td></tr><tr><td>其他</td><td>GPIOB[10]</td></tr></table>	100	SmartCard 2 时钟	010	Timer2 外部事件输入 或是Timer2高低电平输出	001	SPI0 2 nd 从机选择管脚	其他	GPIOB[10]						
		100	SmartCard 2 时钟													
		010	Timer2 外部事件输入 或是Timer2高低电平输出													
		001	SPI0 2 nd 从机选择管脚													
		其他	GPIOB[10]													
		LQFP-64 封装：														
		<table><tr><th>PB10_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V2</td></tr><tr><td>101</td><td>SPI0 MOSIO</td></tr><tr><td>100</td><td>SmartCard 2时钟</td></tr><tr><td>010</td><td>Timer2 外部事件输入 或是Timer2高低电平输出</td></tr><tr><td>001</td><td>SPI0 2nd从机选择管脚</td></tr><tr><td>其他</td><td>GPIOB[10]</td></tr></table>	PB10_MFP	功能	111	LCD V2	101	SPI0 MOSIO	100	SmartCard 2时钟	010	Timer2 外部事件输入 或是Timer2高低电平输出	001	SPI0 2 nd 从机选择管脚	其他	GPIOB[10]
		PB10_MFP	功能													
		111	LCD V2													
		101	SPI0 MOSIO													
		100	SmartCard 2时钟													
		010	Timer2 外部事件输入 或是Timer2高低电平输出													
		001	SPI0 2 nd 从机选择管脚													
		其他	GPIOB[10]													
LQFP-48 封装：																
无																
[7]	-	保留														
[6:4]	PB9_MFP[2:0]	PB.9功能选择 LQFP-128封装：														
		<table><tr><th>PB9_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V1</td></tr><tr><td>101</td><td>外部中断 0</td></tr><tr><td>100</td><td>SmartCard 2 RST</td></tr><tr><td>010</td><td>Timer1外部事件输入 或是Timer1高低电平输出</td></tr><tr><td>001</td><td>SPI1 2nd从机选择管脚</td></tr><tr><td>其他</td><td>GPIOB[9]</td></tr></table>	PB9_MFP	功能	111	LCD V1	101	外部中断 0	100	SmartCard 2 RST	010	Timer1外部事件输入 或是Timer1高低电平输出	001	SPI1 2 nd 从机选择管脚	其他	GPIOB[9]
		PB9_MFP	功能													
		111	LCD V1													
		101	外部中断 0													
		100	SmartCard 2 RST													
		010	Timer1外部事件输入 或是Timer1高低电平输出													
		001	SPI1 2 nd 从机选择管脚													
		其他	GPIOB[9]													
		LQFP-64 封装：														
		<table><tr><th>PB9_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD V1</td></tr><tr><td>101</td><td>外部中断 0</td></tr><tr><td>100</td><td>SmartCard 2 RST</td></tr></table>	PB9_MFP	功能	111	LCD V1	101	外部中断 0	100	SmartCard 2 RST						
		PB9_MFP	功能													
		111	LCD V1													
		101	外部中断 0													
100	SmartCard 2 RST															

Bits	描述		
		010 Timer1外部事件输入 或是Timer1高低电平输出	
		001 SPI1 2 nd 从机选择管脚	
		其他 GPIOB[9]	
	LQFP-48 封装： None		
[3]	-	保留	
[2:0]	PB8_MFP[2:0]	PB.8功能选择 LQFP-128 封装：	
		PB8_MFP	功能
		111	LCD SEG 30
		100	SmartCard 2 电源
		011	外部中断0
		010	Timer0外部事件输入 或是Timer0高低电平输出
		001	ADC 外部触发
		其他	GPIOB[8]
		LQFP-64 封装：	
		PB8_MFP	功能
		111	LCD SEG 13
		100	SmartCard 2电源
		011	外部中断0
		010	Timer0外部事件输入 或是Timer0高低电平输出
		001	ADC外部触发
		其他	GPIOB[8]
		LQFP-48 封装：	
		PB8_MFP	功能
		100	SmartCard 2 电源
		011	外部中断0
010	Timer0外部事件输入 或是Timer0高低电平输出		

Bits	描述		
		001	ADC外部触发
		其他	GPIOB[8]

多功能端口C 低字节控制寄存器(PC_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PC_L_MFP	GCR_BA+0x40	R/W	多功能端口C 低字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PC7_MFP			-	PC6_MFP		
23	22	21	20	19	18	17	16
-	PC5_MFP			-	PC4_MFP		
15	14	13	12	11	10	9	8
-	PC3_MFP			-	PC2_MFP		
7	6	5	4	3	2	1	0
-	PC1_MFP			-	PC0_MFP		

Bits	描述															
[31]	-	保留														
[30:28]	PC7_MFP[2:0]	PC.7功能选择														
		LQFP-128封装：														
		<table><tr><th>PC7_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0通道1</td></tr><tr><td>011</td><td>Timer1捕捉事件</td></tr><tr><td>010</td><td>EBI AD[5]</td></tr><tr><td>001</td><td>DA out1</td></tr><tr><td>其他</td><td>GPIOC[7]</td></tr></table>	PC7_MFP	功能	101	PWM0通道1	011	Timer1捕捉事件	010	EBI AD[5]	001	DA out1	其他	GPIOC[7]		
		PC7_MFP	功能													
		101	PWM0通道1													
		011	Timer1捕捉事件													
		010	EBI AD[5]													
		001	DA out1													
		其他	GPIOC[7]													
		LQFP-64 封装：														
		<table><tr><th>PC7_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 17</td></tr><tr><td>101</td><td>PWM0 通道 1</td></tr><tr><td>011</td><td>Timer1捕捉事件</td></tr><tr><td>010</td><td>EBI AD[5]</td></tr><tr><td>001</td><td>DA输出1</td></tr><tr><td>其他</td><td>GPIOC[7]</td></tr></table>	PC7_MFP	功能	111	LCD SEG 17	101	PWM0 通道 1	011	Timer1捕捉事件	010	EBI AD[5]	001	DA输出1	其他	GPIOC[7]
		PC7_MFP	功能													
		111	LCD SEG 17													
		101	PWM0 通道 1													
		011	Timer1捕捉事件													
010	EBI AD[5]															
001	DA输出1															
其他	GPIOC[7]															

Bits	描述																																							
		<div>LQFP-48 封装：</div> <table><tr><th>PC7_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0通道1</td></tr><tr><td>011</td><td>Timer1捕捉事件</td></tr><tr><td>010</td><td>EBI AD[5]</td></tr><tr><td>001</td><td>DA输出1</td></tr><tr><td>其他</td><td>GPIOC[7]</td></tr></table>	PC7_MFP	功能	101	PWM0通道1	011	Timer1捕捉事件	010	EBI AD[5]	001	DA输出1	其他	GPIOC[7]																										
PC7_MFP	功能																																							
101	PWM0通道1																																							
011	Timer1捕捉事件																																							
010	EBI AD[5]																																							
001	DA输出1																																							
其他	GPIOC[7]																																							
[27]	-	保留																																						
		<div>PC.6功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PC6_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0 通道 0</td></tr><tr><td>100</td><td>SmartCard1 卡侦测</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[4]</td></tr><tr><td>001</td><td>DA 输出0</td></tr><tr><td>其他</td><td>GPIOC[6]</td></tr></table> <div>LQFP-64 封装：</div> <table><tr><th>PC6_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0通道0</td></tr><tr><td>100</td><td>SmartCard1卡侦测</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[4]</td></tr><tr><td>001</td><td>DA 输出0</td></tr><tr><td>其他</td><td>GPIOC[6]</td></tr></table> <div>LQFP-48 封装：</div> <table><tr><th>PC6_MFP</th><th>功能</th></tr><tr><td>101</td><td>PWM0通道0</td></tr><tr><td>100</td><td>SmartCard1卡侦测</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[4]</td></tr></table>	PC6_MFP	功能	101	PWM0 通道 0	100	SmartCard1 卡侦测	011	Timer0捕捉事件	010	EBI AD[4]	001	DA 输出0	其他	GPIOC[6]	PC6_MFP	功能	101	PWM0通道0	100	SmartCard1卡侦测	011	Timer0捕捉事件	010	EBI AD[4]	001	DA 输出0	其他	GPIOC[6]	PC6_MFP	功能	101	PWM0通道0	100	SmartCard1卡侦测	011	Timer0捕捉事件	010	EBI AD[4]
PC6_MFP	功能																																							
101	PWM0 通道 0																																							
100	SmartCard1 卡侦测																																							
011	Timer0捕捉事件																																							
010	EBI AD[4]																																							
001	DA 输出0																																							
其他	GPIOC[6]																																							
PC6_MFP	功能																																							
101	PWM0通道0																																							
100	SmartCard1卡侦测																																							
011	Timer0捕捉事件																																							
010	EBI AD[4]																																							
001	DA 输出0																																							
其他	GPIOC[6]																																							
PC6_MFP	功能																																							
101	PWM0通道0																																							
100	SmartCard1卡侦测																																							
011	Timer0捕捉事件																																							
010	EBI AD[4]																																							
[26:24]	PC6_MFP[2:0]																																							

Bits	描述																					
		<table><tr><td>001</td><td>DA 输出0</td></tr><tr><td>其他</td><td>GPIOC[6]</td></tr></table>	001	DA 输出0	其他	GPIOC[6]																
001	DA 输出0																					
其他	GPIOC[6]																					
[23]	-	保留																				
[22:20]	PC5_MFP[2:0]	<table><tr><td colspan="2">PC.5功能选择</td></tr><tr><td colspan="2">LQFP-128 封装：</td></tr><tr><td>PC5_MFP</td><td>功能</td></tr><tr><td>111</td><td>LCD COM 3</td></tr><tr><td>001</td><td>SPI0 MOSI1</td></tr><tr><td>其他</td><td>GPIOC[5]</td></tr><tr><td colspan="2">LQFP-64 封装：</td></tr><tr><td colspan="2">无</td></tr><tr><td colspan="2">LQFP-48 封装：</td></tr><tr><td colspan="2">无</td></tr></table>	PC.5功能选择		LQFP-128 封装：		PC5_MFP	功能	111	LCD COM 3	001	SPI0 MOSI1	其他	GPIOC[5]	LQFP-64 封装：		无		LQFP-48 封装：		无	
PC.5功能选择																						
LQFP-128 封装：																						
PC5_MFP	功能																					
111	LCD COM 3																					
001	SPI0 MOSI1																					
其他	GPIOC[5]																					
LQFP-64 封装：																						
无																						
LQFP-48 封装：																						
无																						
[19]	-	保留																				
[18:16]	PC4_MFP[2:0]	<table><tr><td colspan="2">PC.4功能选择</td></tr><tr><td colspan="2">LQFP-128 封装：</td></tr><tr><td>PC4_MFP</td><td>功能</td></tr><tr><td>111</td><td>LCD COM 2</td></tr><tr><td>001</td><td>SPI0 MISO1</td></tr><tr><td>其他</td><td>GPIOC[4]</td></tr><tr><td colspan="2">LQFP-64 封装：</td></tr><tr><td colspan="2">无</td></tr><tr><td colspan="2">LQFP-48 封装：</td></tr><tr><td colspan="2">无</td></tr></table>	PC.4功能选择		LQFP-128 封装：		PC4_MFP	功能	111	LCD COM 2	001	SPI0 MISO1	其他	GPIOC[4]	LQFP-64 封装：		无		LQFP-48 封装：		无	
PC.4功能选择																						
LQFP-128 封装：																						
PC4_MFP	功能																					
111	LCD COM 2																					
001	SPI0 MISO1																					
其他	GPIOC[4]																					
LQFP-64 封装：																						
无																						
LQFP-48 封装：																						
无																						
[15]	-	保留																				
[14:12]	PC3_MFP[2:0]	<table><tr><td colspan="2">PC.3功能选择</td></tr><tr><td colspan="2">LQFP-128封装：</td></tr><tr><td>PC3_MFP</td><td>功能</td></tr><tr><td>111</td><td>LCD COM 1</td></tr><tr><td>100</td><td>SmartCard1 RST</td></tr></table>	PC.3功能选择		LQFP-128封装：		PC3_MFP	功能	111	LCD COM 1	100	SmartCard1 RST										
PC.3功能选择																						
LQFP-128封装：																						
PC3_MFP	功能																					
111	LCD COM 1																					
100	SmartCard1 RST																					

Bits	描述													
		<table><tr><td>010</td><td>I²S Dout</td></tr><tr><td>001</td><td>SPI0 MOSI1</td></tr><tr><td>其他</td><td>GPIOC[3]</td></tr></table>	010	I ² S Dout	001	SPI0 MOSI1	其他	GPIOC[3]						
		010	I ² S Dout											
		001	SPI0 MOSI1											
		其他	GPIOC[3]											
		LQFP-64 封装：												
		<table><tr><th>PC3_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD COM 1</td></tr><tr><td>100</td><td>SmartCard1 RST</td></tr><tr><td>010</td><td>I²S Dout</td></tr><tr><td>001</td><td>SPI0 MOSI0</td></tr><tr><td>其他</td><td>GPIOC[3]</td></tr></table>	PC3_MFP	功能	111	LCD COM 1	100	SmartCard1 RST	010	I ² S Dout	001	SPI0 MOSI0	其他	GPIOC[3]
		PC3_MFP	功能											
		111	LCD COM 1											
		100	SmartCard1 RST											
		010	I ² S Dout											
		001	SPI0 MOSI0											
		其他	GPIOC[3]											
		LQFP-48 封装：												
		<table><tr><th>PC3_MFP</th><th>功能</th></tr><tr><td>100</td><td>SmartCard1 RST</td></tr><tr><td>010</td><td>I²S Dout</td></tr><tr><td>001</td><td>SPI0 MOSI0</td></tr><tr><td>其他</td><td>GPIOC[3]</td></tr></table>	PC3_MFP	功能	100	SmartCard1 RST	010	I ² S Dout	001	SPI0 MOSI0	其他	GPIOC[3]		
		PC3_MFP	功能											
		100	SmartCard1 RST											
010	I ² S Dout													
001	SPI0 MOSI0													
其他	GPIOC[3]													
[11]	-	保留												
[10:8]	PC2_MFP[2:0]	PC.2功能选择												
		LQFP-128 封装：												
		<table><tr><th>PC2_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD COM 0</td></tr><tr><td>100</td><td>SmartCard1 电源</td></tr><tr><td>010</td><td>I²S Din</td></tr><tr><td>001</td><td>SPI0 MISO0</td></tr><tr><td>其他</td><td>GPIOC[2]</td></tr></table>	PC2_MFP	功能	111	LCD COM 0	100	SmartCard1 电源	010	I ² S Din	001	SPI0 MISO0	其他	GPIOC[2]
		PC2_MFP	功能											
		111	LCD COM 0											
		100	SmartCard1 电源											
		010	I ² S Din											
		001	SPI0 MISO0											
		其他	GPIOC[2]											
		LQFP-64 封装：												
		<table><tr><th>PC2_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD COM 0</td></tr><tr><td>100</td><td>SmartCard1 电源</td></tr><tr><td>010</td><td>I²S Din</td></tr><tr><td>001</td><td>SPI0 MISO0</td></tr></table>	PC2_MFP	功能	111	LCD COM 0	100	SmartCard1 电源	010	I ² S Din	001	SPI0 MISO0		
		PC2_MFP	功能											
		111	LCD COM 0											
		100	SmartCard1 电源											
		010	I ² S Din											
		001	SPI0 MISO0											

Bits	描述																																			
		<table><tr><td>其他</td><td>GPIOC[2]</td></tr></table> LQFP-48 封装： <table><tr><th>PC2_MFP</th><th>功能</th></tr><tr><td>100</td><td>SmartCard1 电源</td></tr><tr><td>010</td><td>I²S Din</td></tr><tr><td>001</td><td>SPI0 MISO0</td></tr><tr><td>其他</td><td>GPIOC[2]</td></tr></table>	其他	GPIOC[2]	PC2_MFP	功能	100	SmartCard1 电源	010	I ² S Din	001	SPI0 MISO0	其他	GPIOC[2]																						
其他	GPIOC[2]																																			
PC2_MFP	功能																																			
100	SmartCard1 电源																																			
010	I ² S Din																																			
001	SPI0 MISO0																																			
其他	GPIOC[2]																																			
[7]	-	保留																																		
		PC.1功能选择 LQFP-128 封装： <table><tr><th>PC1_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD DH2</td></tr><tr><td>100</td><td>SmartCard1 DATA</td></tr><tr><td>010</td><td>I²S BCLK</td></tr><tr><td>001</td><td>SPI0 SCLK</td></tr><tr><td>其他</td><td>GPIOC[1]</td></tr></table> LQFP-64 封装： <table><tr><th>PC1_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD DH2</td></tr><tr><td>100</td><td>SmartCard1 DATA</td></tr><tr><td>010</td><td>I²S BCLK</td></tr><tr><td>001</td><td>SPI0 SCLK</td></tr><tr><td>其他</td><td>GPIOC[1]</td></tr></table> LQFP-48 封装： <table><tr><th>PC1_MFP</th><th>功能</th></tr><tr><td>100</td><td>SmartCard1 DATA</td></tr><tr><td>010</td><td>I²S BCLK</td></tr><tr><td>001</td><td>SPI0 SCLK</td></tr><tr><td>其他</td><td>GPIOC[1]</td></tr></table>	PC1_MFP	功能	111	LCD DH2	100	SmartCard1 DATA	010	I ² S BCLK	001	SPI0 SCLK	其他	GPIOC[1]	PC1_MFP	功能	111	LCD DH2	100	SmartCard1 DATA	010	I ² S BCLK	001	SPI0 SCLK	其他	GPIOC[1]	PC1_MFP	功能	100	SmartCard1 DATA	010	I ² S BCLK	001	SPI0 SCLK	其他	GPIOC[1]
PC1_MFP	功能																																			
111	LCD DH2																																			
100	SmartCard1 DATA																																			
010	I ² S BCLK																																			
001	SPI0 SCLK																																			
其他	GPIOC[1]																																			
PC1_MFP	功能																																			
111	LCD DH2																																			
100	SmartCard1 DATA																																			
010	I ² S BCLK																																			
001	SPI0 SCLK																																			
其他	GPIOC[1]																																			
PC1_MFP	功能																																			
100	SmartCard1 DATA																																			
010	I ² S BCLK																																			
001	SPI0 SCLK																																			
其他	GPIOC[1]																																			
[6:4]	PC1_MFP[2:0]																																			
		PC.0功能选择																																		
[3]	-	保留																																		
[2:0]	PC0_MFP[2:0]																																			

Bits	描述	
	LQFP-128 封装 :	
	PC0_MFP	功能
	111	LCD DH1
	100	SmartCard1 时钟
	010	I ² S WS
	001	SPI0 1 st 从机选择管脚
	其他	GPIOC[0]
	LQFP-64 封装 :	
	PC0_MFP	功能
	111	LCD DH1
	100	SmartCard1 时钟
	010	I ² S WS
	001	SPI0 1 st 从机选择管脚
	其他	GPIOC[0]
	LQFP-48 封装 :	
	PC0_MFP	功能
	100	SmartCard1 时钟
	010	I ² S WS
	001	SPI0 1 st 从机选择管脚
	其他	GPIOC[0]

多功能端口C 高字节控制寄存器(PC_H_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PC_H_MFP	GCR_BA+0x44	R/W	多功能端口C高字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PC15_MFP			-	PC14_MFP		
23	22	21	20	19	18	17	16
-	PC13_MFP			-	PC12_MFP		
15	14	13	12	11	10	9	8
-	PC11_MFP			-	PC10_MFP		
7	6	5	4	3	2	1	0
-	PC9_MFP			-	PC8_MFP		

Bits	描述													
[31]	-	保留												
[30:28]	PC15_MFP[2:0]	PC.15 功能选择												
		LQFP-128 封装：												
		<table><tr><th>PC15_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 33</td></tr><tr><td>100</td><td>PWM1通道2</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[3]</td></tr><tr><td>其他</td><td>GPIOC[15]</td></tr></table>	PC15_MFP	功能	111	LCD SEG 33	100	PWM1通道2	011	Timer0捕捉事件	010	EBI AD[3]	其他	GPIOC[15]
		PC15_MFP	功能											
		111	LCD SEG 33											
		100	PWM1通道2											
		011	Timer0捕捉事件											
		010	EBI AD[3]											
		其他	GPIOC[15]											
		LQFP-64 封装：												
		<table><tr><th>PC15_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 16</td></tr><tr><td>100</td><td>PWM1通道2</td></tr><tr><td>011</td><td>Timer0捕捉事件</td></tr><tr><td>010</td><td>EBI AD[3]</td></tr><tr><td>其他</td><td>GPIOC[15]</td></tr></table>	PC15_MFP	功能	111	LCD SEG 16	100	PWM1通道2	011	Timer0捕捉事件	010	EBI AD[3]	其他	GPIOC[15]
		PC15_MFP	功能											
		111	LCD SEG 16											
		100	PWM1通道2											
	011	Timer0捕捉事件												
	010	EBI AD[3]												
	其他	GPIOC[15]												
	LQFP-48 封装：													

Bits	描述															
		无														
[27]	-	保留														
[26:24]	PC14_MFP[2:0]	PC.14功能选择 LQFP-128 封装：														
		<table><tr><th>PC14_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 32</td></tr><tr><td>100</td><td>PWM1通道3</td></tr><tr><td>010</td><td>EBI AD[2]</td></tr><tr><td>其他</td><td>GPIOC[14]</td></tr></table>	PC14_MFP	功能	111	LCD SEG 32	100	PWM1通道3	010	EBI AD[2]	其他	GPIOC[14]				
		PC14_MFP	功能													
		111	LCD SEG 32													
		100	PWM1通道3													
		010	EBI AD[2]													
		其他	GPIOC[14]													
		LQFP-64 封装：														
		<table><tr><th>PC14_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 15</td></tr><tr><td>100</td><td>PWM1通道3</td></tr><tr><td>010</td><td>EBI AD[2]</td></tr><tr><td>其他</td><td>GPIOC[14]</td></tr></table>	PC14_MFP	功能	111	LCD SEG 15	100	PWM1通道3	010	EBI AD[2]	其他	GPIOC[14]				
		PC14_MFP	功能													
		111	LCD SEG 15													
		100	PWM1通道3													
		010	EBI AD[2]													
		其他	GPIOC[14]													
		LQFP-48 封装：														
无																
[23]	-	保留														
[22:20]	PC13_MFP[2:0]	PC.13功能选择 LQFP-128 封装：														
		<table><tr><th>PC13_MFP</th><th>功能</th></tr><tr><td>110</td><td>I²C0 SCL</td></tr><tr><td>101</td><td>外部中断1</td></tr><tr><td>100</td><td>探听管脚</td></tr><tr><td>010</td><td>PWM1通道1</td></tr><tr><td>001</td><td>SPI1 MOSI1</td></tr><tr><td>其他</td><td>GPIOC[13]</td></tr></table>	PC13_MFP	功能	110	I ² C0 SCL	101	外部中断1	100	探听管脚	010	PWM1通道1	001	SPI1 MOSI1	其他	GPIOC[13]
		PC13_MFP	功能													
		110	I ² C0 SCL													
		101	外部中断1													
		100	探听管脚													
		010	PWM1通道1													
		001	SPI1 MOSI1													
		其他	GPIOC[13]													
		LQFP-64 封装：														
		无														

Bits	描述													
		LQFP-48 封装： 无												
[19]	-	保留												
[18:16]	PC12_MFP[2:0]	PC.12功能选择 LQFP-128 封装：												
		<table><tr><th>PC12_MFP</th><th>功能</th></tr><tr><td>110</td><td>I²C0 SDA</td></tr><tr><td>101</td><td>外部中断0</td></tr><tr><td>010</td><td>PWM1通道0</td></tr><tr><td>001</td><td>SPI1 MISO1</td></tr><tr><td>其他</td><td>GPIOC[12]</td></tr></table>	PC12_MFP	功能	110	I ² C0 SDA	101	外部中断0	010	PWM1通道0	001	SPI1 MISO1	其他	GPIOC[12]
		PC12_MFP	功能											
		110	I ² C0 SDA											
		101	外部中断0											
		010	PWM1通道0											
		001	SPI1 MISO1											
		其他	GPIOC[12]											
		LQFP-64 封装： 无												
		LQFP-48 封装： 无												
[15]	-	保留												
[14:12]	PC11_MFP[2:0]	PC.11功能选择 LQFP-128 封装：												
		<table><tr><th>PC11_MFP</th><th>功能</th></tr><tr><td>101</td><td>UART1 TX</td></tr><tr><td>001</td><td>SPI1 MOSI0</td></tr><tr><td>其他</td><td>GPIOC[11]</td></tr></table>	PC11_MFP	功能	101	UART1 TX	001	SPI1 MOSI0	其他	GPIOC[11]				
		PC11_MFP	功能											
		101	UART1 TX											
		001	SPI1 MOSI0											
		其他	GPIOC[11]											
		LQFP-64 封装：												
		<table><tr><th>PC11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 31</td></tr><tr><td>101</td><td>UART1 TX</td></tr><tr><td>001</td><td>SPI1 MOSI0</td></tr><tr><td>其他</td><td>GPIOC[11]</td></tr></table>	PC11_MFP	功能	111	LCD SEG 31	101	UART1 TX	001	SPI1 MOSI0	其他	GPIOC[11]		
		PC11_MFP	功能											
		111	LCD SEG 31											
101	UART1 TX													
001	SPI1 MOSI0													
其他	GPIOC[11]													
LQFP-48 封装： 无														

Bits	描述		
[11]	-	保留	
[10:8]	PC10_MFP[2:0]	PC.10功能选择	
		LQFP-128封装：	
		PC10_MFP	功能
		101	UART1 RX
		001	SPI1 MISO0
		其他	GPIOC[10]
		LQFP-64 封装：	
		PC10_MFP	功能
		111	LCD SEG 30
		101	UART1 RX
		001	SPI1 MISO0
		其他	GPIOC[10]
		LQFP-48 封装：	
		无	
		[7]	-
[6:4]	PC9_MFP[2:0]	PC.9功能选择	
		LQFP-128封装：	
		PC9_MFP	功能
		101	I ² C1 SCL
		001	SPI1 SCLK
		其他	GPIOC[9]
		LQFP-64 封装：	
		PC9_MFP	功能
		111	LCD SEG 29
		101	I ² C1 SCL
		001	SPI1 SCLK
		其他	GPIOC[9]
		LQFP-48 封装 :None	

Bits	描述													
[3]	-	保留												
[2:0]	PC8_MFP[2:0]	PC.8功能选择												
		LQFP-128封装：												
		<table><tr><th>PC8_MFP</th><th>功能</th></tr><tr><td>101</td><td>I²C1 SDA</td></tr><tr><td>010</td><td>EBI MCLK</td></tr><tr><td>001</td><td>SPI1 1st从机选择管脚</td></tr><tr><td>其他</td><td>GPIOC[8]</td></tr></table>	PC8_MFP	功能	101	I ² C1 SDA	010	EBI MCLK	001	SPI1 1 st 从机选择管脚	其他	GPIOC[8]		
		PC8_MFP	功能											
		101	I ² C1 SDA											
		010	EBI MCLK											
		001	SPI1 1 st 从机选择管脚											
		其他	GPIOC[8]											
		LQFP-64 封装：												
		<table><tr><th>PC8_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 28</td></tr><tr><td>101</td><td>I²C1 SDA</td></tr><tr><td>010</td><td>EBI XCLK</td></tr><tr><td>001</td><td>SPI1 1st 从机选择管脚</td></tr><tr><td>其他</td><td>GPIOC[8]</td></tr></table>	PC8_MFP	功能	111	LCD SEG 28	101	I ² C1 SDA	010	EBI XCLK	001	SPI1 1 st 从机选择管脚	其他	GPIOC[8]
		PC8_MFP	功能											
		111	LCD SEG 28											
		101	I ² C1 SDA											
		010	EBI XCLK											
		001	SPI1 1 st 从机选择管脚											
		其他	GPIOC[8]											
		LQFP-48 封装：												
无														

多功能端口D低字节控制寄存器(PD_L_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PD_L_MFP	GCR_BA+0x48	R/W	多功能端口D低字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PD7_MFP			-	PD6_MFP		
23	22	21	20	19	18	17	16
-	PD5_MFP			-	PD4_MFP		
15	14	13	12	11	10	9	8
-	PD3_MFP			-	PD2_MFP		
7	6	5	4	3	2	1	0
-	PD1_MFP			-	PD0_MFP		

Bits	描述							
[31]	-	保留						
[30:28]	PD7_MFP[2:0]	PD.7 功能选择 LQFP-128 封装：						
		<table><tr><th>PD7_MDP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 2</td></tr><tr><td>其他</td><td>GPIOD[7]</td></tr></table>	PD7_MDP	功能	111	LCD SEG 2	其他	GPIOD[7]
		PD7_MDP	功能					
		111	LCD SEG 2					
		其他	GPIOD[7]					
		LQFP-64 封装：						
无								
LQFP-48 封装：								
无								
[27]	-	保留						
[26:24]	PD6_MFP[2:0]	PD.6 功能选择 LQFP-128封装：						
		<table><tr><th>PD6_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 3</td></tr><tr><td>其他</td><td>GPIOD[6]</td></tr></table>	PD6_MFP	功能	111	LCD SEG 3	其他	GPIOD[6]
		PD6_MFP	功能					
		111	LCD SEG 3					
		其他	GPIOD[6]					
		LQFP-64 封装：						
无								

Bits	描述		
		LQFP-48 封装： 无	
[23]	-	保留	
[22:20]	PD5_MFP[2:0]	PD.5 功能选择 LQFP-128封装：	
		PD5_MFP	功能
		111	LCD SEG 34
		011	SPI2 MOSI1
		010	I ² S Dout
		其他	GPIOD[5]
		LQFP-64 封装： 无	
		LQFP-48 封装： 无	
[19]	-	保留	
[18:16]	PD4_MFP[2:0]	PD.4 功能选择 LQFP-128 封装：	
		PD4_MFP	功能
		111	LCD SEG 35
		100	SmartCard1 卡侦测
		011	SPI2 MISO1
		010	I ² S Din
		其他	GPIOD[4]
		LQFP-64 封装： 无	
		LQFP-48 封装： 无	
[15]	-	保留	
[14:12]	PD3_MFP[2:0]	PD.3 功能选择 LQFP-128封装：	
		PD3_MFP	功能

Bits	描述	
		101
		100
		011
		010
		001
		其他
		ADC输入通道11 SmartCard1 复位 SPI2 MOSI0 I ² S BCLK UART1 CTSn GPIOD[3]
		LQFP-64 封装 :
		无
		LQFP-48 封装 :
		无
[11]	-	保留
[10:8]	PD2_MFP[2:0]	PD.2 功能选择
		LQFP-128封装 :
		PD2_MFP
		功能
		101
		100
		011
		010
		001
		其他
		ADC输入通道10 SmartCard1 电源 SPI2 MISO0 I ² S WS UART1 RTSn GPIOD[2]
		LQFP-64 封装 :
		无
		LQFP-48 封装 :
		无
[7]	-	保留
[6:4]	PD1_MFP[2:0]	PD.1 功能选择
		LQFP-128 封装 :
		PD1_MFP
		功能
		101
		100
		ADC输入通道9 SmartCard1 DATA

Bits	描述	
		011
		SPI2 SCLK
		001
		UART1 TX
		其他
		GPIOD[1]
		LQFP-64 封装 :
		无
		LQFP-48 封装 :
		无
[3]	-	保留
[2:0]	PD0_MFP[2:0]	PD.0 功能选择
		LQFP-128 封装 :
		PD0_MFP
		功能
		101
		ADC 输入通道8
		100
		SmartCard1 时钟
		011
		SPI2 1 st 从机选择管脚
		001
		UART1 RX
		其他
		GPIOD[0]
		LQFP-64 封装 :
		无
		LQFP-48 封装 :
		无

多功能端口D高字节控制寄存器(PD_H_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PD_H_MFP	GCR_BA+0x4C	R/W	多功能端口D高字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PD15_MFP			-	PD14_MFP		
23	22	21	20	19	18	17	16
-	PD13_MFP			-	PD12_MFP		
15	14	13	12	11	10	9	8
-	PD11_MFP			-	PD10_MFP		
7	6	5	4	3	2	1	0
-	PD9_MFP			-	PD8_MFP		

Bits	描述		
[31]	-	保留	
[30:28]	PD15_MFP[2:0]	PD.15功能选择	
		LQFP-128 封装：	
		PD15_MFP	功能
		111	LCD SEG 0
		其他	GPIOD[15]
		LQFP-64 封装：	
		无	
		LQFP-48 封装：	
		无	
[27]	-	保留	
[26:24]	PD14_MFP[2:0]	PD.14功能选择	
		LQFP-128 封装：	
		PD14_MFP	功能
		111	LCD SEG 1
		其他	GPIOD[14]
		LQFP-64 封装：	
		无	

Bits	描述							
		LQFP-48 封装 : 无						
[23]	-	保留						
[22:20]	PD13_MFP[2:0]	PD.13功能选择 LQFP-128 封装 :						
		<table><tr><th>PD13_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 14</td></tr><tr><td>其他</td><td>GPIOD[13]</td></tr></table>	PD13_MFP	功能	111	LCD SEG 14	其他	GPIOD[13]
		PD13_MFP	功能					
		111	LCD SEG 14					
		其他	GPIOD[13]					
LQFP-64 封装 : 无								
LQFP-48 封装 : 无								
[19]	-	保留						
[18:16]	PD12_MFP[2:0]	PD.12功能选择 LQFP-128 封装 :						
		<table><tr><th>PD12_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 15</td></tr><tr><td>其他</td><td>GPIOD[12]</td></tr></table>	PD12_MFP	功能	111	LCD SEG 15	其他	GPIOD[12]
		PD12_MFP	功能					
		111	LCD SEG 15					
		其他	GPIOD[12]					
LQFP-64 封装 : 无								
LQFP-48 封装 : 无								
[15]	-	保留						
[14:12]	PD11_MFP[2:0]	PD.11功能选择 LQFP-128封装 :						
		<table><tr><th>PD11_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 16</td></tr><tr><td>其他</td><td>GPIOD[11]</td></tr></table>	PD11_MFP	功能	111	LCD SEG 16	其他	GPIOD[11]
		PD11_MFP	功能					
		111	LCD SEG 16					
		其他	GPIOD[11]					
LQFP-64 封装 :								

Bits	描述							
		无 LQFP-48 封装： 无						
[11]	-	保留						
[10:8]	PD10_MFP[2:0]	PD.10功能选择 LQFP-128封装：						
		<table><tr><th>PD10_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 17</td></tr><tr><td>其他</td><td>GPIOD[10]</td></tr></table>	PD10_MFP	功能	111	LCD SEG 17	其他	GPIOD[10]
		PD10_MFP	功能					
		111	LCD SEG 17					
		其他	GPIOD[10]					
LQFP-64 封装：								
无								
LQFP-48 封装： 无								
[7]	-	保留						
[6:4]	PD9_MFP[2:0]	PD.9功能选择 LQFP-128封装：						
		<table><tr><th>PD9_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 18</td></tr><tr><td>LCD SEG 18</td><td>GPIOD[9]</td></tr></table>	PD9_MFP	功能	111	LCD SEG 18	LCD SEG 18	GPIOD[9]
		PD9_MFP	功能					
		111	LCD SEG 18					
		LCD SEG 18	GPIOD[9]					
LQFP-64 封装：								
无								
LQFP-48 封装： 无								
[3]	-	保留						
[2:0]	PD8_MFP[2:0]	PD.8功能选择 LQFP-128 封装：						
		<table><tr><th>PD8_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 19</td></tr><tr><td>其他</td><td>GPIOD[8]</td></tr></table>	PD8_MFP	功能	111	LCD SEG 19	其他	GPIOD[8]
		PD8_MFP	功能					
		111	LCD SEG 19					
		其他	GPIOD[8]					
LQFP-64 封装：								

Bits	描述	
		无
		LQFP-48 封装 :
		无

多功能端口 E 低字节控制寄存器(PE_L_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PE_L_MFP	GCR_BA+0x50	R/W	多功能端口 E 低字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PE7_MFP			-	PE6_MFP		
23	22	21	20	19	18	17	16
-	PE5_MFP			-	PE4_MFP		
15	14	13	12	11	10	9	8
-	PE3_MFP			-	PE2_MFP		
7	6	5	4	3	2	1	0
-	PE1_MFP			-	PE0_MFP		

Bits	描述							
[31]	-	保留						
[30:28]	PE7_MFP[2:0]	PE.7功能选择 LQFP-128封装：						
		<table><tr><th>PE7_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 8</td></tr><tr><td>其他</td><td>GPIOE[7]</td></tr></table>	PE7_MFP	功能	111	LCD SEG 8	其他	GPIOE[7]
		PE7_MFP	功能					
		111	LCD SEG 8					
		其他	GPIOE[7]					
LQFP-64 封装：								
无								
LQFP-48 封装：								
无								
[27]	-	保留						
[26:24]	PE6_MFP[2:0]	PE.6功能选择 LQFP-128 封装：						
		GPIOE[6]						
		LQFP-64 封装：						
		GPIOE[6]						
		LQFP-48 封装：						
GPIOE[6]								
[23]	-	保留						
[22:20]	PE5_MFP[2:0]	PE.5功能选择						

Bits	描述																			
		<div>LQFP-128封装：</div> <table><tr><th>PE5_MFP</th><th>功能</th></tr><tr><td>001</td><td>PWM1 Channel 1</td></tr><tr><td>其他</td><td>GPIOE[5]</td></tr></table> <div>LQFP-64封装：</div> <div>NANO100/NANO120 系列:</div> <table><tr><th>PE5_MFP</th><th>功能</th></tr><tr><td>001</td><td>PWM1 Channel 1</td></tr><tr><td>其他</td><td>GPIOE[5]</td></tr></table> <div>NANO110/130系列:</div> <div>无</div> <div>LQFP-48封装：</div> <div>NANO100系列:</div> <table><tr><th>PE5_MFP</th><th>功能</th></tr><tr><td>001</td><td>PWM1 Channel 1</td></tr><tr><td>其他</td><td>GPIOE[5]</td></tr></table> <div>NANO110/NANO120/NANO130系列:</div> <div>无</div>	PE5_MFP	功能	001	PWM1 Channel 1	其他	GPIOE[5]	PE5_MFP	功能	001	PWM1 Channel 1	其他	GPIOE[5]	PE5_MFP	功能	001	PWM1 Channel 1	其他	GPIOE[5]
PE5_MFP	功能																			
001	PWM1 Channel 1																			
其他	GPIOE[5]																			
PE5_MFP	功能																			
001	PWM1 Channel 1																			
其他	GPIOE[5]																			
PE5_MFP	功能																			
001	PWM1 Channel 1																			
其他	GPIOE[5]																			
[19]	-	保留																		
[18:16]	PE4_MFP[2:0]	<div>PE.4功能选择</div> <div>LQFP-128封装：</div> <table><tr><th>PE4_MFP</th><th>功能</th></tr><tr><td>110</td><td>SPI0 MISO0</td></tr><tr><td>其他</td><td>GPIOE[4]</td></tr></table> <div>LQFP-64 封装：</div> <div>无</div> <div>LQFP-48 封装：</div> <div>无</div>	PE4_MFP	功能	110	SPI0 MISO0	其他	GPIOE[4]												
PE4_MFP	功能																			
110	SPI0 MISO0																			
其他	GPIOE[4]																			
[15]	-	保留																		
[14:12]	PE3_MFP[2:0]	PE.3功能选择																		

Bits	描述									
		<div>LQFP-128封装：</div> <table><tr><th>PE3_MFP</th><th>功能</th></tr><tr><td>110</td><td>SPI0 MISO0</td></tr><tr><td>其他</td><td>GPIOE[3]</td></tr></table> <div>LQFP-64 封装：</div> <div>无</div> <div>LQFP-48 封装：</div> <div>无</div>	PE3_MFP	功能	110	SPI0 MISO0	其他	GPIOE[3]		
PE3_MFP	功能									
110	SPI0 MISO0									
其他	GPIOE[3]									
[11]	-	保留								
[10:8]	PE2_MFP[2:0]	<div>PE.2功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PE2_MFP</th><th>功能</th></tr><tr><td>110</td><td>SPI0 SCLK</td></tr><tr><td>其他</td><td>GPIOE[2]</td></tr></table> <div>LQFP-64 封装：</div> <div>无</div> <div>LQFP-48 封装：</div> <div>无</div>	PE2_MFP	功能	110	SPI0 SCLK	其他	GPIOE[2]		
PE2_MFP	功能									
110	SPI0 SCLK									
其他	GPIOE[2]									
[7]	-	保留								
[6:4]	PE1_MFP[2:0]	<div>PE.1功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PE1_MFP</th><th>功能</th></tr><tr><td>110</td><td>SPI0 1st 从机选择管脚</td></tr><tr><td>001</td><td>PWM1 通道 3</td></tr><tr><td>其他</td><td>GPIOE[1]</td></tr></table> <div>LQFP-64 封装：</div> <div>无</div> <div>LQFP-48 封装：</div>	PE1_MFP	功能	110	SPI0 1 st 从机选择管脚	001	PWM1 通道 3	其他	GPIOE[1]
PE1_MFP	功能									
110	SPI0 1 st 从机选择管脚									
001	PWM1 通道 3									
其他	GPIOE[1]									

Bits	描述									
		无								
[3]	-	保留								
[2:0]	PE0_MFP[2:0]	PE.0功能选择								
		LQFP-128 封装：								
		<table><tr><th>PE0_MFP</th><th>功能</th></tr><tr><td>010</td><td>I²S MCLK</td></tr><tr><td>001</td><td>PWM1 通道 2</td></tr><tr><td>其他</td><td>GPIOE[0]</td></tr></table>	PE0_MFP	功能	010	I ² S MCLK	001	PWM1 通道 2	其他	GPIOE[0]
		PE0_MFP	功能							
		010	I ² S MCLK							
		001	PWM1 通道 2							
		其他	GPIOE[0]							
LQFP-64 封装：										
无										
LQFP-48 封装：										
无										

多功能端口 E 高字节控制寄存器(PE_H_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PE_H_MFP	GCR_BA+0x54	R/W	多功能端口E高字节控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	PE15_MFP			-	PE14_MFP		
23	22	21	20	19	18	17	16
-	PE13_MFP			-	PE12_MFP		
15	14	13	12	11	10	9	8
-	PE11_MFP			-	PE10_MFP		
7	6	5	4	3	2	1	0
-	PE9_MFP			-	PE8_MFP		

Bits	描述							
[31]	-	保留						
[30:28]	PE15_MFP[2:0]	PE.15 功能选择 LQFP-128 封装：						
		<table><tr><th>PE15_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 2</td></tr><tr><td>其他</td><td>GPIOE[15]</td></tr></table>	PE15_MFP	功能	111	LCD SEG 2	其他	GPIOE[15]
		PE15_MFP	功能					
		111	LCD SEG 2					
		其他	GPIOE[15]					
		LQFP-64 封装：						
无								
LQFP-48 封装：								
无								
[27]	-	保留						
[26:24]	PE14_MFP[2:0]	PE.14功能选择 LQFP-128 封装：						
		<table><tr><th>PE14_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 28</td></tr><tr><td>其他</td><td>GPIOE[14]</td></tr></table>	PE14_MFP	功能	111	LCD SEG 28	其他	GPIOE[14]
		PE14_MFP	功能					
		111	LCD SEG 28					
		其他	GPIOE[14]					
		LQFP-64 封装：						
无								

Bits	描述							
		LQFP-48 封装： 无						
[23]	-	保留						
[22:20]	PE13_MFP[2:0]	PE.13功能选择 LQFP-128封装：						
		<table><tr><th>PE13_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 27</td></tr><tr><td>其他</td><td>GPIOE[13]</td></tr></table>	PE13_MFP	功能	111	LCD SEG 27	其他	GPIOE[13]
		PE13_MFP	功能					
		111	LCD SEG 27					
		其他	GPIOE[13]					
LQFP-64 封装： 无								
LQFP-48 封装： 无								
[19]	-	保留						
[18:16]	PE12_MFP[2:0]	PE.12功能选择 LQFP-128封装：						
		<table><tr><th>PE12_MFP</th><th>功能</th></tr><tr><td>111</td><td>UART1 CTSn (仅NANO110有效)</td></tr><tr><td>其他</td><td>GPIOE[12]</td></tr></table>	PE12_MFP	功能	111	UART1 CTSn (仅NANO110有效)	其他	GPIOE[12]
		PE12_MFP	功能					
		111	UART1 CTSn (仅NANO110有效)					
		其他	GPIOE[12]					
LQFP-64 封装： 无								
LQFP-48 封装： 无								
[15]	-	保留						
[14:12]	PE11_MFP[2:0]	PE.11功能选择 LQFP-128封装：						
		<table><tr><th>PE11_MFP</th><th>功能</th></tr><tr><td>111</td><td>UART1 RTSn (仅NANO110有效)</td></tr><tr><td>其他</td><td>GPIOE[11]</td></tr></table>	PE11_MFP	功能	111	UART1 RTSn (仅NANO110有效)	其他	GPIOE[11]
		PE11_MFP	功能					
		111	UART1 RTSn (仅NANO110有效)					
		其他	GPIOE[11]					
LQFP-64 封装：								

Bits	描述							
		无 LQFP-48 封装： 无						
[11]	-	保留						
[10:8]	PE10_MFP[2:0]	PE.10功能选择 LQFP-128 封装：						
		<table><tr><th>PE10_MFP</th><th>功能</th></tr><tr><td>111</td><td>UART1 TX (仅NANO110有效)</td></tr><tr><td>其他</td><td>GPIOE[10]</td></tr></table>	PE10_MFP	功能	111	UART1 TX (仅NANO110有效)	其他	GPIOE[10]
		PE10_MFP	功能					
		111	UART1 TX (仅NANO110有效)					
		其他	GPIOE[10]					
LQFP-64 封装：								
无								
LQFP-48 封装：								
无								
[7]	-	保留						
[6:4]	PE9_MFP[2:0]	PE.9功能选择 LQFP-128 封装：						
		<table><tr><th>PE9_MFP</th><th>功能</th></tr><tr><td>111</td><td>UART1 RX (仅NANO110有效)</td></tr><tr><td>其他</td><td>GPIOE[9]</td></tr></table>	PE9_MFP	功能	111	UART1 RX (仅NANO110有效)	其他	GPIOE[9]
		PE9_MFP	功能					
		111	UART1 RX (仅NANO110有效)					
		其他	GPIOE[9]					
LQFP-64 封装：								
无								
LQFP-48 封装：								
无								
[3]	-	保留						
[2:0]	PE8_MFP[2:0]	PE.8功能选择 LQFP-128封装：						
		<table><tr><th>PE8_MFP</th><th>功能</th></tr><tr><td>111</td><td>LCD SEG 9</td></tr><tr><td>其他</td><td>GPIOE[8]</td></tr></table>	PE8_MFP	功能	111	LCD SEG 9	其他	GPIOE[8]
		PE8_MFP	功能					
		111	LCD SEG 9					
		其他	GPIOE[8]					
LQFP-64 封装：								
无								

Bits	描述	
		LQFP-48 封装： 无

多功能端口F低字节控制寄存器(PF_L_MFP)

寄存器	偏移量	R/W	描述	复位后的值
PF_L_MFP	GCR_BA+0x58	R/W	多功能端口F低字节控制寄存器	0x0077_7777

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-	PF5_MFP			-	PF4_MFP		
15	14	13	12	11	10	9	8
-	PF3_MFP			-	PF2_MFP		
7	6	5	4	3	2	1	0
-	PF1_MFP			-	PF0_MFP		

Bits	描述							
[31:23]	-	保留						
[22:20]	PF5_MFP[2:0]	PF.5 功能选择 LQFP-128 封装：						
		<table><tr><th>PF5_MFP</th><th>功能</th></tr><tr><td>001</td><td>I²C0 SCL</td></tr><tr><td>其他</td><td>GPIOF[5]</td></tr></table>	PF5_MFP	功能	001	I ² C0 SCL	其他	GPIOF[5]
		PF5_MFP	功能					
		001	I ² C0 SCL					
		其他	GPIOF[5]					
LQFP-64 封装：								
无								
LQFP-48 封装：								
无								
[19]	-	保留						
[18:16]	PF4_MFP[2:0]	PF.4功能选择 LQFP-128 封装：						
		<table><tr><th>PF4_MFP</th><th>功能</th></tr><tr><td>001</td><td>I²C0 SDA</td></tr><tr><td>其他</td><td>GPIOF[4]</td></tr></table>	PF4_MFP	功能	001	I ² C0 SDA	其他	GPIOF[4]
		PF4_MFP	功能					
		001	I ² C0 SDA					
		其他	GPIOF[4]					
LQFP-64 封装：								
无								

Bits	描述							
		LQFP-48 封装： 无						
[15]	-	保留						
[14:12]	PF3_MFP[2:0]	PF.3功能选择 LQFP-128 封装：						
		<table><tr><th>PF3_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT IN</td></tr><tr><td>其他</td><td>GPIOF[3]</td></tr></table>	PF3_MFP	功能	111	HXT IN	其他	GPIOF[3]
		PF3_MFP	功能					
		111	HXT IN					
		其他	GPIOF[3]					
		LQFP-64 封装：						
		<table><tr><th>PF3_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT IN</td></tr><tr><td>其他</td><td>GPIOF[3]</td></tr></table>	PF3_MFP	功能	111	HXT IN	其他	GPIOF[3]
		PF3_MFP	功能					
		111	HXT IN					
		其他	GPIOF[3]					
		LQFP-48 封装：						
		<table><tr><th>PF3_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT IN</td></tr><tr><td>其他</td><td>GPIOF[3]</td></tr></table>	PF3_MFP	功能	111	HXT IN	其他	GPIOF[3]
PF3_MFP	功能							
111	HXT IN							
其他	GPIOF[3]							
LQFP-48 封装：								
<table><tr><th>PF3_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT IN</td></tr><tr><td>其他</td><td>GPIOF[3]</td></tr></table>	PF3_MFP	功能	111	HXT IN	其他	GPIOF[3]		
PF3_MFP	功能							
111	HXT IN							
其他	GPIOF[3]							
LQFP-48 封装：								
<table><tr><th>PF3_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT IN</td></tr><tr><td>其他</td><td>GPIOF[3]</td></tr></table>	PF3_MFP	功能	111	HXT IN	其他	GPIOF[3]		
PF3_MFP	功能							
111	HXT IN							
其他	GPIOF[3]							
[11]	-	保留						
[10:8]	PF2_MFP[2:0]	PF.2功能选择 LQFP-128 封装：						
		<table><tr><th>PF2_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT OUT</td></tr><tr><td>其他</td><td>GPIOF[2]</td></tr></table>	PF2_MFP	功能	111	HXT OUT	其他	GPIOF[2]
		PF2_MFP	功能					
		111	HXT OUT					
		其他	GPIOF[2]					
		LQFP-64 封装：						
		<table><tr><th>PF2_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT OUT</td></tr><tr><td>其他</td><td>GPIOF[2]</td></tr></table>	PF2_MFP	功能	111	HXT OUT	其他	GPIOF[2]
		PF2_MFP	功能					
		111	HXT OUT					
		其他	GPIOF[2]					
		LQFP-48 封装：						
		<table><tr><th>PF2_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT OUT</td></tr></table>	PF2_MFP	功能	111	HXT OUT		
PF2_MFP	功能							
111	HXT OUT							
LQFP-48 封装：								
<table><tr><th>PF2_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT OUT</td></tr></table>	PF2_MFP	功能	111	HXT OUT				
PF2_MFP	功能							
111	HXT OUT							
LQFP-48 封装：								
<table><tr><th>PF2_MFP</th><th>功能</th></tr><tr><td>111</td><td>HXT OUT</td></tr></table>	PF2_MFP	功能	111	HXT OUT				
PF2_MFP	功能							
111	HXT OUT							

Bits	描述																															
		<table><tr><td>其他</td><td>GPIOF[2]</td></tr></table>	其他	GPIOF[2]																												
其他	GPIOF[2]																															
[7]	-	保留																														
[6:4]	PF1_MFP[2:0]	<div>PF.1功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PF1_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE CLOCK</td></tr><tr><td>101</td><td>外部中断1</td></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table> <div>LQFP-64 封装：</div> <table><tr><th>PF1_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE CLOCK</td></tr><tr><td>101</td><td>外部中断1</td></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table> <div>LQFP-48 封装：</div> <table><tr><th>PF1_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE clock</td></tr><tr><td>101</td><td>外部中断1</td></tr><tr><td>100</td><td>FRQDIV_CLK</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table>	PF1_MFP	功能	111	ICE CLOCK	101	外部中断1	100	FRQDIV_CLK	其他	GPIOF[1]	PF1_MFP	功能	111	ICE CLOCK	101	外部中断1	100	FRQDIV_CLK	其他	GPIOF[1]	PF1_MFP	功能	111	ICE clock	101	外部中断1	100	FRQDIV_CLK	其他	GPIOF[1]
PF1_MFP	功能																															
111	ICE CLOCK																															
101	外部中断1																															
100	FRQDIV_CLK																															
其他	GPIOF[1]																															
PF1_MFP	功能																															
111	ICE CLOCK																															
101	外部中断1																															
100	FRQDIV_CLK																															
其他	GPIOF[1]																															
PF1_MFP	功能																															
111	ICE clock																															
101	外部中断1																															
100	FRQDIV_CLK																															
其他	GPIOF[1]																															
[3]	-	保留																														
[2:0]	PF0_MFP[2:0]	<div>PF.0功能选择</div> <div>LQFP-128 封装：</div> <table><tr><th>PF0_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE DATA</td></tr><tr><td>101</td><td>外部中断0</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table>	PF0_MFP	功能	111	ICE DATA	101	外部中断0	其他	GPIOF[1]																						
PF0_MFP	功能																															
111	ICE DATA																															
101	外部中断0																															
其他	GPIOF[1]																															

Bits	描述									
		LQFP-64 封装 :								
		<table><tr><th>PF0_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE DATA</td></tr><tr><td>101</td><td>外部中断0</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table>	PF0_MFP	功能	111	ICE DATA	101	外部中断0	其他	GPIOF[1]
		PF0_MFP	功能							
		111	ICE DATA							
		101	外部中断0							
		其他	GPIOF[1]							
		LQFP-48 封装 :								
		<table><tr><th>PF0_MFP</th><th>功能</th></tr><tr><td>111</td><td>ICE DATA</td></tr><tr><td>101</td><td>外部中断0</td></tr><tr><td>其他</td><td>GPIOF[1]</td></tr></table>	PF0_MFP	功能	111	ICE DATA	101	外部中断0	其他	GPIOF[1]
		PF0_MFP	功能							
		111	ICE DATA							
		101	外部中断0							
		其他	GPIOF[1]							

上电复位控制寄存器(PORCTL)

寄存器	偏移量	R/W	描述	复位后的值
PORCTL	GCR_BA+0x60	R/W	上电复位控制寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[15:0]	POR_DIS_CODE[15:0]	这个寄存器被用作上电复位使能控制（写保护位） 当上电的时候,电源电路会产生一个复位信号复位芯片的所有功能模块,但是电源的噪音会引起不断的上电动作。如果将POR_DIS_CODE 设置成 0x5AA5, 上电复位功能将被禁止, 当POR_DIS_CODE被设置成其他值, 上电复位功能会重新开启。或者是其他的复位功能, 包括RESET脚, 看门狗复位, 欠压检测复位, ICE复位命令, 软件复位等使POR_DIS_CODE 值复位。上电复位功能也会重新开启

欠压侦测控制寄存器(BODCTL)

BODCTL 控制寄存器的部分位 通过 flash configuration来初始化

寄存器	偏移量	R/W	描述	复位后的值
BODCTL	GCR_BA+64	R/W	欠压侦测控制寄存器	0x00FF_F0xx

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-				-			
15	14	13	12	11	10	9	8
-				-	BOD25_INT_EN	BOD20_INT_EN	BOD17_INT_EN
7	6	5	4	3	2	1	0
-	BOD25_RST_EN	BOD20_RST_EN	BOD17_RST_EN	-	BOD25_EN	BOD20_EN	BOD17_EN

Bits	描述	
[31:11]	-	保留
[10]	BOD25_INT_EN	BOD 2.5 V 中断使能 （写保护位） 1 = BOD25 发生时会产生中断 0 = BOD25 发生时不会产生中断
[9]	BOD20_INT_EN	BOD 2.0 V 中断使能 （写保护位） 1 = BOD20 发生时会产生中断 0 = BOD20 发生时不会产生中断
[8]	BOD17_INT_EN	BOD 1.7 V 中断使能 （写保护位） 1 = BOD17 发生时会产生中断 0 = BOD17 发生时不会产生中断
[7]	-	保留
[6]	BOD25_RST_EN	BOD 2.5 V 复位使能 （写保护位） 1 = BOD25 发生时会产生复位 0 = BOD25 发生时不会产生复位 默认值由用户在配置 flash 控制寄存器config0 bit[20:19] 时设置
[5]	BOD20_RST_EN	BOD 2.0 V 复位使能 （写保护位） 1 = BOD20 发生时会产生复位 0 = BOD20发生时不会产生复位

Bits	描述							
		默认值由用户在配置 flash 控制寄存器config0 bit[20:19] 时设置						
[4]	BOD17_RST_EN	<p>BOD1.7 V 复位使能（写保护位）.</p> <p>1 = BOD1.7发生时会产生复位</p> <p>0 = BOD1.7发生时不会产生复位</p> <p>默认值由用户在配置 flash 控制寄存器config0 bit[20:19] 时设置</p> <p>BOD17_RST_EN 仅当 BOD17_EN 是1的时候有作用.</p> <table><tr><td></td><td>BOD17 RST使能</td></tr><tr><td>BOD17_EN 0</td><td>使能</td></tr><tr><td>BOD17_EN 1</td><td>由BOD17_RST_EN决定</td></tr></table>		BOD17 RST使能	BOD17_EN 0	使能	BOD17_EN 1	由BOD17_RST_EN决定
	BOD17 RST使能							
BOD17_EN 0	使能							
BOD17_EN 1	由BOD17_RST_EN决定							
[3]	-	保留						
[2]	BOD25_EN	<p>BOD2.5 V 使能（写保护位）</p> <p>1 = 使能BOD检测2.5 V 功能</p> <p>0 = 禁止BOD检测2.5 V 功能</p>						
[1]	BOD20_EN	<p>BOD2.0 V 使能（写保护位）</p> <p>1 = 使能BOD检测2.0 V 功能</p> <p>0 = 禁止BOD检测2.0V 功能</p> <p>BOD20_EN默认是使能的，如果软件禁止它，BOD2.0V功能只有在芯片进入掉电模式后才是禁止的。如果系统没有进入掉电模式， BOD20_EN 会由硬件自动使能。</p>						
[0]	BOD17_EN	<p>BOD1.7 V 使能（写保护位）</p> <p>默认值由用户在配置 flash 控制寄存器config0 bit[20:19] 时设置</p> <p>BOD17_EN 可以被禁止，但是只有在系统进入掉电模式才有效，一旦退出掉电模式BOD17会由硬件自动使能。</p> <p>CPU 会读 BOD17 功能 是否使能，换句话说，CPU读到的值总是1.</p> <p>1 = 使能BOD检测1.7 V 功能</p> <p>0 = 禁止BOD检测1.7 V 功能</p> <table><tr><td></td><td>BOD17 使能</td></tr><tr><td>正常模式</td><td>使能</td></tr><tr><td>掉电模式</td><td>由BOD17_EN决定</td></tr></table>		BOD17 使能	正常模式	使能	掉电模式	由BOD17_EN决定
	BOD17 使能							
正常模式	使能							
掉电模式	由BOD17_EN决定							

欠压侦测状态寄存器(BODSTS)

寄存器	偏移量	R/W	描述	复位后的值
BODSTS	GCR_BA+68	R	欠压侦测状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
	BOD25_rise	BOD20_rise	BOD17_rise	BOD25_drop	BOD20_drop	BOD17_drop	BOD_INT

Bits	描述	
[31:7]	-	保留
[6]	BOD25_rise	欠压侦测高于2.5V状态 BOD25_drop被置高, 意味着侦测到的电压高于设定目标电压 (2.5 V)。向该位写 1 清零。
[5]	BOD20_rise	欠压侦测高于2.0V状态 BOD20_drop被置高, 意味着侦测到的电压高于设定目标电压 (2.0 V)。向该位写 1 清零。
[4]	BOD17_rise	欠压侦测高于1.7V 状态 BOD17_drop被置高, 意味着侦测到的高压低于设定目标电压 (1.7 V)。向该位写 1 清零。
[3]	BOD25_drop	欠压侦测低于2.5V状态 BOD25_drop被置高, 意味着侦测到的电压低于设定目标电压 (2.5 V)。向该位写 1 清零。
[2]	BOD20_drop	欠压侦测低于2.0V状态 BOD20_drop被置高, 意味着侦测到的电压低于设定目标电压 (2.0 V)。向该位写 1 清零。
[1]	BOD17_drop	欠压侦测低于1.7V 状态 BOD17_drop被置高, 意味着侦测到的电压低于设定目标电压 (1.7 V)。向该位写 1 清零。
[0]	BOD_INT	欠压侦测中断状态 1= 当电压侦测器侦测到 V _{DD} 降到目标侦测电压以下或升到目标侦测电压以上并且欠压中断使能, 这一位会被置1。 0= 在电压中断使能后, 电压侦测器没有侦测到 V _{DD} 降到目标侦测电压以下或升到目

		标侦测电压以上 向该位写 1 清零
--	--	----------------------

内部参考电压产生器控制寄存器 (Int_VREFCTL)

寄存器	偏移量	R/W	描述	复位后的值
Int_VREFCTL	GCR_BA+0x6C	R/W	内部参考电压控制寄存器	0x0000_0F00

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-				-			
7	6	5	4	3	2	1	0
-				EXT_MODE	SEL25	REG_EN	BGP_EN

Bits	描述	
[31:4]	保留	
[3]	EXT_MODE	<p>稳压器外部模式 (写保护)</p> <p>该位是写保护的, 请参考解锁寄存器RegLockAddr的解锁序列来解锁</p> <p>如果EXT_MODE为1, 用户可以将内部稳压器输出到VREF引脚上</p> <p>0 = 稳压器电压不输出到外部VREF引脚</p> <p>1 = 稳压器电压输出到外部VREF引脚。在AVSS脚连接一个1uF到10uF的电容可以让内部参考电压更稳定</p>
[2]	SEL25	<p>电压调整器输出电压选择 (写保护位)</p> <p>选择内部参考电压值</p> <p>0 = 1.8V</p> <p>1 = 2.5V</p>
[1]	REG_EN	<p>电压调整器使能 (写保护位)</p> <p>使能内部1.8V或2.5V 参考电压.</p> <p>0 = 禁止</p> <p>1 = 使能</p>
[0]	BGP_EN	<p>带宽间隙使能 (写保护位)</p> <p>带宽间隙是内部参考电压, 用户如果想使能内部1.8V 或 2.5V参考电压, 必须使能带宽间隙。</p> <p>0 = 禁止</p> <p>1 = 使能</p>

HIRC 调整控制寄存器(IRCTRIMCTL)

寄存器	偏移量	R/W	描述	复位后的值
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC调整控制寄存器	0x0000_0000

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

Bits	描述											
[31:8]	-	保留										
[8]	ERR_STOP	<p>当检测到32.768KHz 错误，停止调整</p> <p>当检测到32.768KHz 时钟错误时，该位用来控制是否停止 HIRC 调整操作</p> <p>如果该位是高并且 32.768KHz 时钟错误被检测到，状态位32K_ERR_INT 会被置高，HIRC停止调整操作。如果该位是低并且32.768KHz 时钟错误被检测到，状态位23K_ERR_INT会被置高，HIRC继续调整操作。</p> <p>0 = 当32.768KHz 时钟错误被检测到，HIRC继续调整操作</p> <p>1 = 当32.768KHz 时钟错误被检测到，HIRC停止调整操作</p>										
[7:6]	TRIM_RETRY_CNT[1:0]	<p>调整值更新限制次数</p> <p>这部分定义了HIRC的频率被锁定之前，自动调整电路会多少次试着更新HIRC的调整值</p> <p>一旦 HIRC 被锁定, 内部调整值更新计算器会复位</p> <p>如果调整值更新计算器达到限定值并且HIRC的频率仍然没有被锁定。自动调整操作会被禁止 并且TRIM_SEL 会被清 00.</p> <table><tr><th>TRIM_RETRY_CNT</th><th>调整重试计数限制</th></tr><tr><td>00</td><td>调整重试计数限制是 64</td></tr><tr><td>01</td><td>调整重试计数限制是128</td></tr><tr><td>10</td><td>调整重试计数限制是256</td></tr><tr><td>11</td><td>调整重试计数限制是512</td></tr></table>	TRIM_RETRY_CNT	调整重试计数限制	00	调整重试计数限制是 64	01	调整重试计数限制是128	10	调整重试计数限制是256	11	调整重试计数限制是512
TRIM_RETRY_CNT	调整重试计数限制											
00	调整重试计数限制是 64											
01	调整重试计数限制是128											
10	调整重试计数限制是256											
11	调整重试计数限制是512											
[5:4]	TRIM_LOOP[1:0]	<p>调整计算循环</p> <p>这部分定义了调整值计算是在 多少个 32.768KHz 时钟基础上的。</p> <p>例如, 如果TRIM_LOOP 被设置成00, 自动调整电路会在平均频率4 32.768KHz 时钟的基础上计算调整值..</p>										

Bits	描述		
		TRIM_LOOP	平均频率
		00	4 32.768KHz 时钟
		01	8 32.768KHz 时钟
		10	16 32.768KHz 时钟
		11	32 32.768KHz 时钟
[3:2]		保留	
[1:0]	TRIM_SEL[1:0]	调整频率选择 这两个位定义目标HIRC自动调整频率。 如果没有选择任何目标频率(TRIM_SEL 为 00), HIRC自动调整功能被禁止 在自动调整操作期间, 如果32.768KHz 时钟错误或者自动调整次数到达限制次数, 这两个位被自动清00	
		TRIM_SEL	功能
		00	禁止HIRC自动调整功能
		01	使能HIRC自动调整功能 并且调整 HIRC到11.0592MHz
		10	使能HIRC自动调整功能 并且调整 HIRC到12MHz
		11	使能HIRC自动调整功能 并且调整 HIRC到12.288MHz

HIRC调整中断使能寄存器(IRCTRIMIEN)

寄存器	偏移量	R/W	描述	复位后的值
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC调整中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-					32K_ERR_IEN	TRIM_FAIL_IEN	-

Bits	描述	
[31:3]	-	保留
[2]	32K_ERR_IEN	<p>32.768KHz 时钟错误中断使能</p> <p>这个位控制着在自动调整期间，当32.768KHz 不准确的时候，CPU是否会产生中断</p> <p>如果这位是1，并且在自动调整操作期间，32K_ERR_INT 被置1，会产生一个中断通知32.768KHz时钟频率是不准确的。</p> <p>0 = 禁止32K_ERR_INT状态触发CPU中断。</p> <p>1 =使能32K_ERR_INT状态触发CPU中断。</p>
[1]	TRIM_FAIL_IEN	<p>调整失败中断使能</p> <p>这一位控制着当HIRC调整值达到更新次数限制的时候是否产生一个中断，及HIRC频率是否仍然锁定在.TRIM_SEL设定的目标频率上</p> <p>如果这位是1，并且在自动调整操作期间，TRIM_FAIL_INT 被置1,会产生一个中断通知 HIRC 达到自动调整值更新限制次数。</p> <p>0 = 禁止32K_FAIL_INT状态触发CPU中断。</p> <p>1 =使能32K_FAIL_INT状态触发CPU中断。</p>
[0]		保留

HIRC调整中断状态寄存器(IRCTRIMINT)

寄存器	偏移量	R/W	描述	复位后的值
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC调整中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-					32K_ERR_INT	TRIM_FAIL_INT	FREQ_LOCK

Bits	描述	
[31:3]	-	保留.
[2]	32K_ERR_INT	<p>32.768KHz 时钟错误中断状态</p> <p>这位是1的话表明32.768KHz 时钟频率是不正确的. 一旦这位被置1自动调整操作会停止, 并且 TRIM_SEL 会被硬件自动清0</p> <p>如果这位是1并且32K_ERR_IEN是1,会产生一个中断通知 32.768KHz 时钟是不正确的. 向该位写1清零.</p> <p>0 = 32.768KHz时钟频率是正确的</p> <p>1 = 32.768KHz 时钟频率是不正确的</p>
[1]	TRIM_FAIL_INT	<p>调整失败中断状态</p> <p>这位是1的话表明HIRC达到调整值更新限制次数并且HIRC时钟频率仍然没有锁定. 一旦这位被置1, 自动调整操作会停止, 并且 TRIM_SEL 会被硬件自动清0</p> <p>如果这位是1并且TRIM_FAIL_IEN是1, 会产生一个中断通知 HIRC调整值更新限制次数已达到. 向该位写1清零.</p> <p>0 = 调整值更新限制次数没有达到</p> <p>1 =调整值更新限制次数已达到并且HIRC 频率仍然没有锁定</p>
[0]	FREQ_LOCK	<p>HIRC 频率锁定状态</p> <p>这位表明 HIRC频率锁定.</p> <p>这是个状态位不会触发任何中断</p>

寄存器解锁地址寄存器(RegLockAddr)

一些系统控制寄存器需要被保护以避免不注意的写入错误干扰芯片的正常工作。这些系统控制寄存器在上电之后是被锁着的，除非用户把它打开。用户要写这些被保护的寄存器需要一个开锁的顺序。开锁的顺序是连续的写入数据“59h”，“16h”“88h”到解锁控制器地址 0x5000_0100。任何不同的数据值或不同的顺序或者在写入这三个数据期间有任何其他的操作都会破坏整个的解锁过程。

锁被打开之后，用户可以检查锁的状态位RegLockAddr[0]。“1”是解锁状态，“0”是锁状态。之后用户可以更新目标寄存器的值 如果RegUnLock是1，写任何数据到地址“0x5000_0100”可以重新锁住受保护的寄存器。

写这个寄存器是为了锁或解锁，读这个寄存器可以知道当前锁的状态

寄存器	偏移量	R/W	描述	复位后的值
RegLockAddr	GCR_BA+100	R/W	寄存器解锁地址寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[0]	RegUnLock	1 = 写保护的寄存器处于解锁状态 0 = 写保护的寄存器处于锁状态，任何写入动作都是无效的。

5.4 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 (CPU时钟, HCLKx, 和PCLKx) 和所有外围设备时钟。HCLKx 意思是为挂在AHB总线上的外设提供的AHB总线时钟。PCLKx 意思是为挂在APB 总线上的外设提供的APB总线时钟。时钟控制器带有各个时钟开关控制的电源控制功能，时钟源的选择和一个4位的时钟分频器。只有设置了掉电使能位 (PD_EN) 并执行 WFI 指令后才会进入掉电模式。在掉电模式下，控制器始终关闭外部高频晶振，内部高频震荡器，和系统时钟 (CPU 时钟，HCLKx和PCLKx)使系统耗电降到最低。

5.4.2 特征

- 产生系统时钟和所有外设时钟
- 每个外设时钟总可以开关
- 在系统进入掉电模式下，外部高频晶振、内部高频震荡器和系统时钟会被关闭

5.4.3 框图

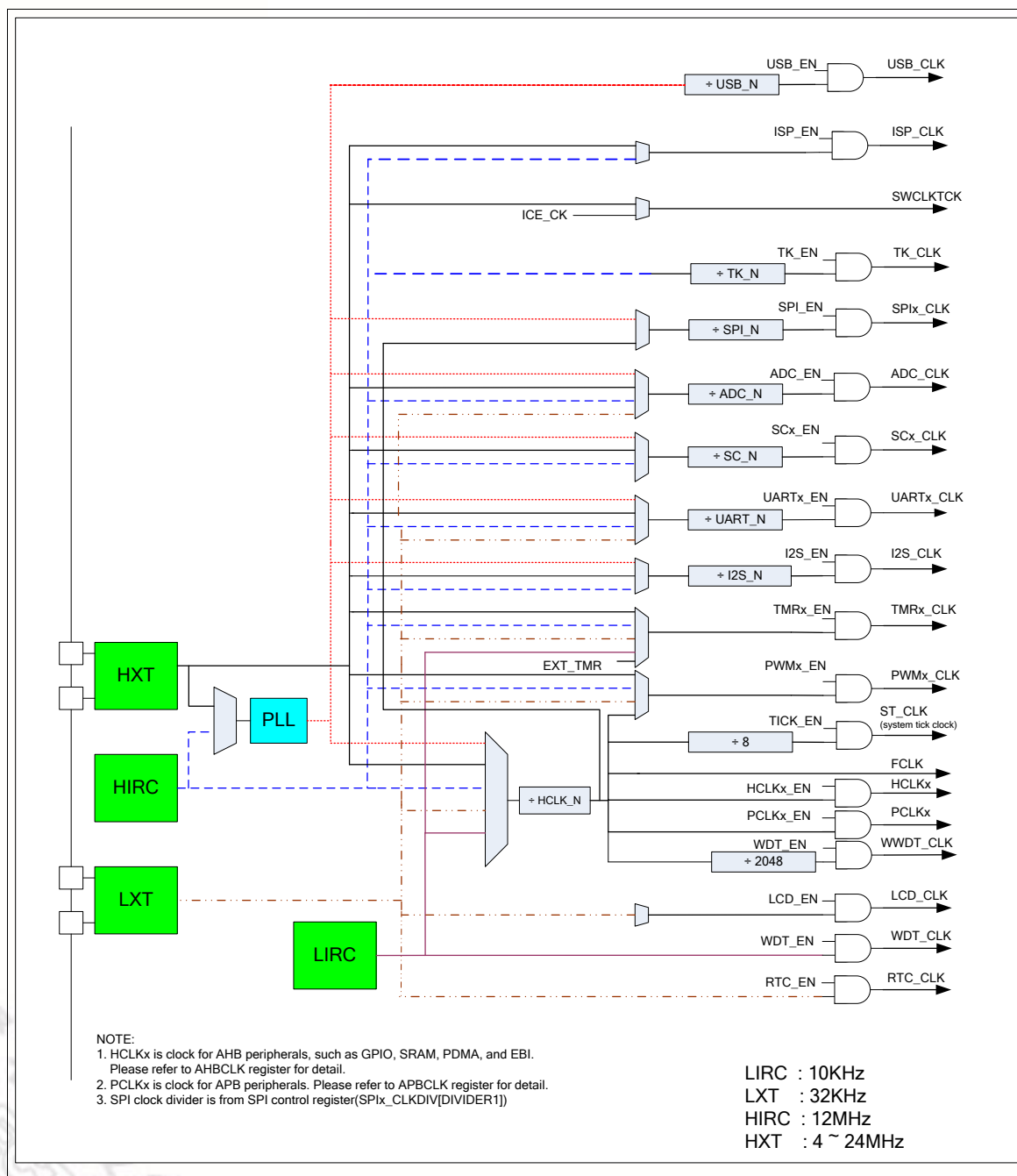


图 5-2 时钟控制器框图

5.4.4 功能描述

5.4.4.1 时钟控制器由下列5个时钟源组成:

- 32768低速外部晶振(LXT)
- 4~ 24MHz 高速外部晶振(HXT)

- 一个可编程的 PLL FOUT (PLL 源由HXT 和HIRC组成)
- 12MHz 高速内部RC震荡器 (HIRC)
- 10KHz 低速内部RC震荡器 (LIRC)

5.4.4.2 外设时钟

每一个外设有不同的时钟源开关设置。请参考CLKSEL1 和CLKSEL2的描述。

	HXT	LXT	HIRC	LIRC	PCLK/HCLK	PLL	Ext. Pin
LCD	-	Yes	-	Yes	-	-	-
TM	Yes	Yes	Yes	Yes	-	-	Yes
PWM	Yes	Yes	Yes	-	-	Yes	-
ADC	Yes	Yes	Yes	-	-	Yes	-
UART	Yes	Yes	Yes	-	-	Yes	-
SC	Yes	-	Yes	-	-	Yes	-
I ² S	Yes	-	Yes	-	-	Yes	-
SPI	-	-	Yes	-	-	Yes	-
I ² C	-	-	-	-	Yes	-	-
USB	-	-	-	-	-	Yes	-

表 5-6 外设时钟

5.4.4.3 掉电模式下的时钟

当芯片进入掉电模式，系统时钟 (CPU 时钟, HCLKx, 和PCLKx), HXT, HIRC 将会被直接关闭., 如果在进入掉电模式之前CPU没有关闭LXT 和LIRC, LXT 和LIRC仍然开启。在掉电模式下，如果LXT 或LIRC没有被禁止，IP 设备仍可以使用这些时钟。

5.4.4.4 频率分频器输出

这个设备是一个16级二进制分频器，选择其中一个输出到GPIOB.12/GPIOF.1，所以输出频率可以为 $F_{in}/(2^1)$ 到 $F_{in}/(2^{16})$ 。

输出的公式是 $F_{out} = F_{in}/2^{(N+1)}$, F_{in} 是输入的时钟频率, F_{out} 是时钟分频器输出的时钟频率, N 是一个4位的值FSEL (FRQDIV[3:0]).

当FDIV_EN (FRQDIV[4]) 被置成1,分频器复位并开始计数，当FDIV_EN被置成0时，计数器会一直计数到输出为0后才停止，并保持输出为0。

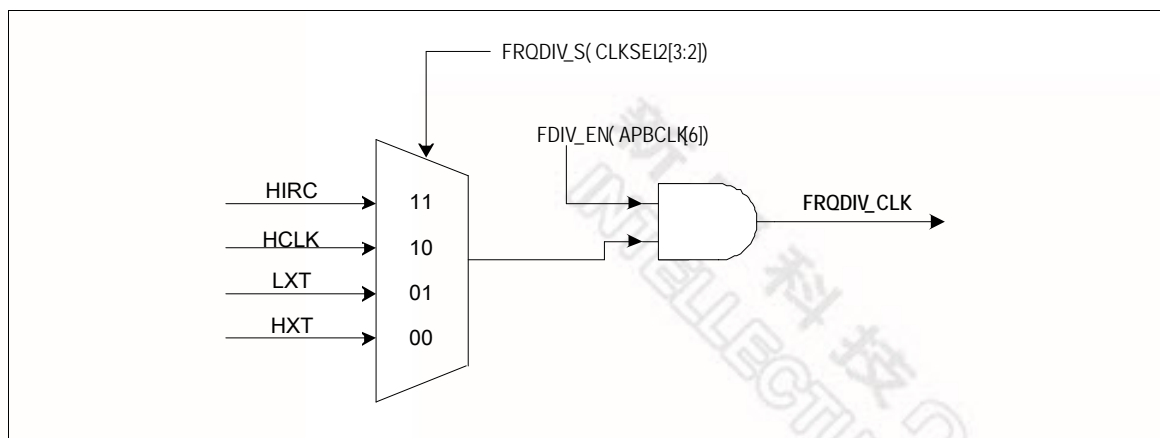


图 5-3 分频器的时钟源

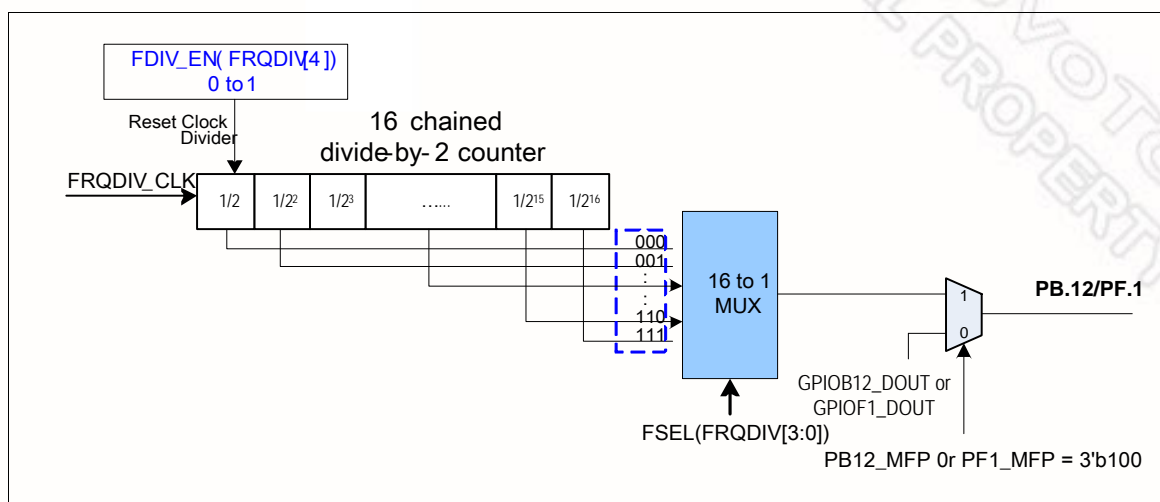


图 5-4 分频器框图

5.4.5 寄存器和内存的映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
CLK_BA = 0x5000_0200				
PWRCTL	CLK_BA + 0x00	R/W	System Power Down Control Register	0x0000_031x
AHBCLK	CLK_BA + 0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0035
APBCLK	CLK_BA + 0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLKSTATUS	CLK_BA + 0x0c	R	Clock status monitor Register	0x0000_001x
CLKSEL0	CLK_BA + 0x10	R/W	Clock Source Select Control Register 0	0x0000_000x
CLKSEL1	CLK_BA + 0x14	R/W	Clock Source Select Control Register 1	0x0007_FFFF
CLKSEL2	CLK_BA + 0x18	R/W	Clock Source Select Control Register 2	0x000F_FFFF
CLKDIV0	CLK_BA + 0x1C	R/W	Clock Divider Number Register 0	0x0000_0000
CLKDIV1	CLK_BA + 0x20	R/W	Clock Divider Number Register 1	0x0000_0000
PLLCTL	CLK_BA + 0x24	R/W	PLL Control Register	0x0003_1120
FRQDIV	CLK_BA + 0x28	R/W	Frequency Divider Control Register	0x0000_0000
WK_INTSTS	CLK_BA + 0x30	R	Wake-up interrupt status	0x0000_0000

5.4.6 寄存器描述

掉电控制寄存器 (PWRCTL)

寄存器	偏移量	R/W	描述	复位后的值
PWRCTL	CLK_BA +0x 00	R/W	系统掉电控制寄存器	0x0000_031x

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
			HXT_HF_ST		LXT_SCNT	HXT_GAIN	HXT_SELXT
7	6	5	4	3	2	1	0
-	PD_EN	PD_WK_IE	WK_DLY	LIRC_EN	HIRC_EN	LXT_EN	HXT_EN

Bits	描述	
[31:10]		保留
[12:11]	HXT_HF_ST	高速外部晶振 (HXT) 频率选择 设置此位以符合 HXT 振频率选择 (建议) 00 = HXT 频率为 4 MHz 至 12 MHz 01 = HXT 频率为 12 MHz 至 16 MHz 10 = HXT 频率为 16 MHz 至 24 MHz 11 = 保留
[10]	LXT_SCNT	LXT稳定时间控制 (写保护位) 0: 在LXT 输出之前延时4096个 LXT 1: 在LXT 输出之前延时8192个 LXT
[9]	HXT_GAIN	HXT 增益控制位 (写保护位) 增益控制是用来扩大晶振的增益确保晶振正常工作 增益使能的情况晶振消耗的电比增益禁止的情况大 0: 禁止增益控制, 那意味着 HXT gain总是高 用于 16MHz 到 24MHz 晶振. 1: 使能增益控制 HXT gain 会持续高 2ms 然后低. 这是为了省电. 用于 4MHz 到 16MHz 晶振.
[8]	HXT_SELXT	HXT SELXT HXT SELXT (写保护位) 0: 高频晶振回路禁止, 这是用在外部震荡器情况下

Bits	描述	
		1: 高频晶振回路使能, 这是用在外部晶振情况下
[7]	-	保留
[6]	PD_EN	<p>芯片掉电模式使能位 (写保护位)</p> <p>当CPU设置了这个位, 芯片掉电使能, 只有CPU睡眠模式激活的情况下才会进入掉电模式</p> <p>当芯片从掉电模式唤醒, 这个位会被自动清0</p> <p>当芯片在掉电模式, LDO, HXT 和HIRC会被禁止, 但是LXT 和LIRC不会被掉电模式控制.</p> <p>在掉电模式下, PLL 和系统时钟 (CPU, HCLKx 和PCLKx) 也是被禁止的无论选择什么时钟源. 如果外围设备时钟来自 LXT 或LIRC, 外围设备时钟不会被这个位控制.</p> <p>在掉电模式下, flash 宏电源是打开的</p> <p>1 = 使能芯片掉电模式</p> <p>0 = 芯片工作在正常模式</p>
[5]	PD_WK_IE	<p>掉电模式唤醒中断使能 (写保护位)</p> <p>0 = 禁止</p> <p>1 = 使能.</p> <p>如果PD_WK_IS 和PD_WK_IE 都为高, PD_WK_INT 会被置位</p>
[4]	WK_DLY	<p>使能唤醒延时计数器 (写保护位)</p> <p>当芯片从掉电模式唤醒, 时钟控制器会延时4096个时钟周期等待HXT稳定或16个时钟周期等待HIRC稳定.</p> <p>1 = 使能时钟延时</p> <p>0 = 禁止时钟延时</p>
[3]	LIRC_EN	<p>LIRC 控制 (写保护位)</p> <p>1 = 使能</p> <p>0 = 禁止</p> <p>LIRC默认是使能</p>
[2]	HIRC_EN	<p>HIRC 控制 (写保护位)</p> <p>1 = 使能</p> <p>0 = 禁止</p> <p>HIRC默认是使能.</p>
[1]	LXT_EN	<p>LXT 控制 (写保护位)</p> <p>1 = 使能</p> <p>0 = 禁止</p> <p>LXT默认是禁止.</p>
[0]	HXT_EN	<p>HXT 控制 (写保护位)</p> <p>这个位的默认值由用户配置flash控制寄存器config0 [26]来设定.</p> <p>1 = 使能</p> <p>0 = 禁止</p>

Bits	描述	
		HXT 默认是禁止的.

模式	PD_EN	CPU 执行WFI 指令	Clock Gating
正常模式	0	NO	由S/W 设定
空闲模式 (CPU进入睡眠模式)	0	YES	仅停CPU时钟
掉电模式	1	YES	除了LXT, LIRC以及使用它们的外 设之外, 其它外设时钟都停止 S/W可以在芯片进入掉电模式之前 关闭LXT和LIRC

表 5-7 电源模式和时钟

AHB设备时钟使能控制寄存器 (AHBCLK)

这些寄存器的位用来使能/禁止 AHB IP HCLK 和engine 时钟

寄存器	偏移量	R/W	描述	复位后的值
AHBCLK	CLK_BA + 0x04	R/W	AHB设备时钟使能控制寄存器	0x0000_0035

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

Bits	描述	
[31:6]	-	保留
[5]	TICK_EN	系统Tick 时钟使能 1 = 使能 0 = 禁止
[4]	SRAM_EN	SRAM 控制时钟使能 1 = 使能 0 = 禁止
[3]	EBI_EN	EBI控制时钟使能 1 = 使能 0 = 禁止
[2]	ISP_EN	Flash ISP控制时钟使能 1 = 使能 0 = 禁止
[1]	DMA_EN	DMA控制时钟使能 1 = 使能 0 = 禁止
[0]	GPIO_EN	GPIO控制时钟使能 1 = 使能 0 = 禁止

APB设备时钟使能控制寄存器 (APBCLK)

这些寄存器位用来使能/禁止APB IP PCLK 和engine 时钟.

寄存器	偏移量	R/W	描述	复位后的值
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
SC1_EN	SC0_EN	I2S_EN	ADC_EN	USBD_EN	LCD_EN	DAC_EN	-
23	22	21	20	19	18	17	16
PWM1_CH23_EN	PWM1_CH01_EN	PWM1_CH23_EN	PWM0_CH01_EN	-		UART1_EN	UART0_EN
15	14	13	12	11	10	9	8
-	SPI2_EN	SPI1_EN	SPI0_EN	-		I2C1_EN	I2C0_EN
7	6	5	4	3	2	1	0
SC2_EN-	FDIV_EN	TMR3_EN	TMR2_EN	TMR1_EN	TMR0_EN	RTC_EN	WDT_EN

Bits	描述	
[31]	SC1_EN	SmartCard 1时钟使能控制 1 = 使能 0 = 禁止
[30]	SC0_EN	SmartCard 0时钟使能控制 1 = 使能 0 = 禁止
[29]	I2S_EN	I2S时钟使能控制 1 = 使能 0 = 禁止
[28]	ADC_EN	模数转换器 (ADC) 时钟使能控制 1 = 使能 0 = 禁止
[27]	USBD_EN	USB FS 设备时钟使能控制 1 = 使能 0 = 禁止
[26]	LCD_EN	LCD控制器时钟使能控制 1 = 使能 0 = 禁止
[25]	DAC_EN	12-bit DAC时钟使能控制

Bits	描述	
		1 = 使能 0 = 禁止
[24]	-	保留
[23]	PWM1_CH23_EN	PWM1 通道2 和通道3 时钟使能控制 1 = 使能 0 = 禁止
[22]	PWM1_CH01_EN	PWM1 通道0 和通道1 时钟使能控制 1 = 使能 0 = 禁止
[21]	PWM0_CH23_EN	PWM0 通道2 和通道3 时钟使能控制 1 = 使能 0 = 禁止
[20]	PWM0_CH01_EN	PWM0 通道0 和通道1 时钟使能控制 1 = 使能 0 = 禁止
[19:18]	-	保留
[17]	UART1_EN	UART1 时钟使能控制 1 = 使能 0 = 禁止
[16]	UART0_EN	UART0 时钟使能控制 1 = 使能 0 = 禁止
[15]	-	保留
[14]	SPI2_EN	SPI2 时钟使能控制 1 = 使能 0 = 禁止
[13]	SPI1_EN	SPI1 时钟使能控制 1 = 使能 0 = 禁止
[12]	SPI0_EN	SPI0 时钟使能控制 1 = 使能 0 = 禁止
[11:10]	-	保留
[9]	I2C1_EN	I2C1 时钟使能控制 1 = 使能

Bits	描述	
		0 = 禁止
[8]	I2C0_EN	I2C0 时钟使能控制 1 = 使能 0 = 禁止
[7]	SC2_EN	SmartCard 2时钟使能控制. 1 =使能 0 =禁止
[6]	FDIV_EN	频率分频器输出时钟使能控制 1 = 使能 0 = 禁止
[5]	TMR3_EN	Timer3 时钟使能控制 1 = 使能 0 = 禁止
[4]	TMR2_EN	Timer2 时钟使能控制 1 = 使能 0 = 禁止
[3]	TMR1_EN	Timer1 时钟使能控制 1 = 使能 0 = 禁止
[2]	TMR0_EN	Timer0 时钟使能控制 1 = 使能 0 = 禁止
[1]	RTC_EN	Real-Time-Clock时钟使能控制 这位仅用来控制 RTC APB 时钟, RTC engine 时钟源来自LXT 1 = 使能 0 = 禁止
[0]	WDT_EN	Watch-Dog Timer 时钟使能控制 这是一个写保护的寄存器, 设置它请参考解锁顺序 这位仅用来控制WDT APB 时钟, WDT engine时钟源来自 LIRC. 1 = 使能 0 = 禁止

时钟状态寄存器 (CLKSTATUS)

这些寄存器位用来侦测芯片的时钟源是否稳定，时钟切换是否失败。

寄存器	偏移量	R/W	描述	复位后的值
CLKSTATUS	CLK_BA+0x0C	R	时钟状态监测寄存器	0x0000_001x

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

Bits	描述	
[31:8]		保留
[7]	CLK_SW_FAIL	时钟切换失败标志 1 = 时钟切换失败 0 = 时钟切换成功 当目标切换时钟源不稳定的时候该位会被置1，向该位写1清0
[6:5]	-	保留
[4]	HIRC_STB	HIRC 时钟源稳定标志 1 = HIRC 时钟是稳定的 0 = HIRC 时钟不稳定或者没有被使能
[3]	LIRC_STB	LIRC 时钟源稳定标志 1 = LIRC 时钟是稳定的 0 = LIRC时钟不稳定或者没有被使能
[2]	PLL_STB	PLL 时钟源稳定标志 1 = PLL时钟是稳定的 0 = PLL 时钟不稳定或者没有被使能
[1]	LXT_STB	LEXT时钟源稳定标志 1 = LXT 时钟是稳定的 0 = LXT时钟不稳定或者没有被使能
[0]	HXT_STB	HEXT 时钟源稳定标志 1 = 时钟是稳定的

Bits	描述	
		0 = HXT时钟不稳定或者没有被使能

时钟源选择控制寄存器0 (CLKSEL0)

寄存器	偏移量	R/W	描述	复位后的值
CLKSEL0	CLK_BA + 0x10	R/W	时钟源选择控制寄存器0	0x0000_000x[

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

Bits	描述													
[31:3]	保留	保留												
[2:0]	HCLK_S[2:0]	HCLK 时钟源选择. （写保护位） 注意： 在时钟源切换之前，相关的时钟源 (预选和新选) 必须被打开。 在任何复位后，这 3 位的缺省值将从Flash 控制器的用户配置寄存器 CFOSC0 (Config0[26:24]) 加载。因此缺省值可能为 000b 或者 111b. .												
		<table><tr><th>HCLK_S</th><th>时钟源</th></tr><tr><td>000</td><td>HXT</td></tr><tr><td>001</td><td>LXT</td></tr><tr><td>010</td><td>PLL clock</td></tr><tr><td>011</td><td>LIRC</td></tr><tr><td>111</td><td>HIRC</td></tr></table>	HCLK_S	时钟源	000	HXT	001	LXT	010	PLL clock	011	LIRC	111	HIRC
		HCLK_S	时钟源											
		000	HXT											
		001	LXT											
		010	PLL clock											
		011	LIRC											
111	HIRC													

时钟源选择控制寄存器1 (CLKSEL1)

寄存器	偏移量	R/W	描述	复位后的值
CLKSEL1	CLK_BA + 0x14	R/W	时钟源选择控制寄存器1	0x007_FFFF

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
					LCD_S	-	
15	14	13	12	11	10	9	8
-	TMR1_S			-	TMR0_S		
7	6	5	4	3	2	1	0
PWM0_CH23_S		PWM0_CH01_S		ADC_S		UART_S	

Bits	描述		
[31:15]		保留	
[18]	LCD_S	LCD 时钟源选择 0 =时钟源来自LXT 1 = 保留	
[17:15]	-	保留	
[14:12]	TMR1_S[2:0]	Timer1时钟源选择	
		TMR1_S	时钟源
		000	HXT
		001	LXT
		010	LIRC
		011	外部管脚
		1xx	HIRC
[11]		保留	
[10:8]	TMR0_S[2:0]	Timer0时钟源选择	
		TMR0_S	时钟源
		000	HXT
		001	LXT
		010	LIRC
		011	external pin

Bits	描述		
		1xx	HIRC
[7:6]	PWM0_CH23_S [1:0]	PWM0通道 2 和通道 3 时钟源选择	
		PWM0 通道2和通道3使用相同的 Engine 时钟源和相同的预分频	
		PWM0_CH23_S	时钟源
		00	HXT
		01	LXT
		10	HCLK
[5:4]	PWM0_CH01_S [1:0]	PWM0通道0 and 通道1 时钟源选择	
		PWM0 通道0和通道1使用相同的 Engine 时钟源和相同的预分频	
		PWM0_CH01_S	时钟源
		00	HXT
		01	LXT
		10	HCLK
[3:2]	ADC_S[1:0]	ADC时钟源选择	
		ADC_S	时钟源
		00	HXT
		01	LXT
		10	PLL clock
		11	HIRC
[1:0]	UART_S[1:0]	UART 0/1时钟源选择. (UART0 和UART1 使用相同的时钟源选择)	
		UART_S	时钟源
		00	HXT
		01	LXT
		10	PLL clock
		11	HIRC

时钟源选择控制寄存器2 (CLKSEL2)

在时钟切换之前，相关的时钟源(预选和新选)，必须打开

寄存器	偏移量	R/W	描述	复位后的值
CLKSEL2	CLK_BA + 0x18	R/W	时钟源选择控制寄存器2	0x00FF_FFFF

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-	SPI2_S	SPI1_S	SPI0_S	SC_S		I2S_S	
15	14	13	12	11	10	9	8
-	TMR3_S			-	TMR2_S		
7	6	5	4	3	2	1	0
PWM1_CH23_S		PWM1_CH01_S		FRQDIV_S		-	

Bits	描述		
[31:23]		保留	
[22]	SPI2_S	SPI2时钟源选择 0 = PLL 1 = HCLK	
[21]	SPI1_S	SPI1时钟源选择 0 = PLL 1 = HCLK	
[20]	SPI0_S	SPI0时钟源选择 0 = PLL 1 = HCLK	
[19:18]	SC_S[1:0]	SC时钟源选择	
		SC_S	时钟源
		00	HXT
		01	PLL 输出
		10	HIRC
		11	HIRC
注意: SC0 和SC1 使用相同的时钟源选择，但是它们有不同的分频值			
[17:16]	I2S_S[1:0]	I ² S 时钟源选择	
		I2S_S	时钟源

Bits	描述	
		0 HXT
		01 PLL 输出
		10 HIRC
		11 HIRC
[15]		保留
[14:12]	TMR3_S[2:0]	Timer3 时钟源选择
		TMR3_S 时钟源
		000 HXT
		001 LXT
		010 LIRC
		011 外部管脚
[11]	TMR2_S[2:0]	1xx HIRC
		保留
		Timer2 时钟源选择
		TMR2_S 时钟源
		000 HXT
		001 LXT
[10:8]	PWM1_CH23_S [1:0]	010 LIRC
		011 外部管脚
		1xx HIRC
		PWM1 通道2 和通道3时钟源选择
		PWM1 通道2 和通道 3 使用相同的Engine时钟源和相同的预分频
		PWM1_CH23_S 时钟源
[7:6]	PWM1_CH01_S [1:0]	00 HXT
		01 LXT
		10 HCLK
		11 HIRC
		PWM1 通道0 和通道1时钟源选择
[5:4]	PWM1_CH01_S [1:0]	PWM1 通道0 和通道1使用相同的 Engine 时钟源和相同的预分频
		PWM1_CH01_S 时钟源
		00 HXT
		01 LXT
		1 HCLK

Bits	描述	
		11 HIRC
[3:2]	FRQDIV_S[1:0]	Clock分频器时钟源选择
		FRQDIV_S 时钟源
		00 HXT
		01 LXT
		10 HCLK
		11 HIRC
[1:0]	-	保留

时钟分频器寄存器0(CLKDIV0)

寄存器	偏移量	R/W	描述	复位后的值
CLKDIV0	CLK_BA_ + 0x1C	R/W	时钟分频器数值寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
SC0_N				-			
23	22	21	20	19	18	17	16
ADC_N							
15	14	13	12	11	10	9	8
I2S_N				UART_N			
7	6	5	4	3	2	1	0
USB_N				HCLK_N			

Bits	描述	
[31:28]	SC0_N[3:0]	SC 0 时钟分频值 $SC\ 0\ 时钟频率 = (SC0\ 时钟源频率) / (SC0_N + 1)$
[27:24]	-	保留
[23:16]	ADC_N[7:0]	ADC时钟分频值 $The\ ADC\ 时钟频率 = (ADC\ 时钟源频率) / (ADC_N + 1)$
[15:12]	I2S_N[3:0]	I ² S 时钟分频值 $The\ I^2S\ 时钟频率 = (I^2S\ 时钟源频率) / (I2S_N + 1)$
[11:8]	UART_N[3:0]	UART 时钟分频值 $UART\ 时钟频率 = (UART\ 时钟源频率) / (UART_N + 1)$
[7:4]	USB_N[3:0]	USB 时钟分频值 $USB\ 时钟频率 = (PLL\ 频率) / (USB_N + 1)$
[3:0]	HCLK_N[3:0]	HCLK 时钟分频值 $HCLK\ 时钟频率 = (HCLK\ 时钟频率) / (HCLK_N + 1)$

时钟分频器寄存器1(CLKDIV1)

寄存器	偏移量	R/W	描述	复位后的值
CLKDIV1	CLK_BA_ + 0x20	R/W	时钟分频器数值寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
SC2_N				SC1_N			

Bits	描述	
[31:4]	-	保留
[7:4]	SC2_N[3:0]	SC 2时钟分频值 The SC 2时钟频率= (SC 2时钟源频率) / (SC2_N + 1)
[3:0]	SC1_N[3:0]	SC 1时钟分频值 SC 1时钟频率= (SC 1时钟源频率) / (SC1_N + 1)

PLL控制寄存器 (PLLCTL)

PLL的参考输入时钟来自 HXT 或HIRC. 这个寄存器用来控制PLL 输出频率和 PLL 操作模式

寄存器	偏移量	R/W	描述	复位后的值
PLLCTL	CLK_BA + 0x24	R/W	PLL控制寄存器	0x0003_1120

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

Bits	描述	
[17]	PLL_SRC	PLL 时钟源选择 1 = PLL 时钟源来自 HIRC 0 = PLL时钟源来自HXT
[16]	PD	掉电模式 如果在 PWR_CTL寄存器中设置 PD_EN 位为“1”， PLL 也会进入掉电模式 0 = PLL 在正常模式 1 = PLL 在掉电模式 (默认)
[15:13]		保留
[12]	OUT_DV	PLL 输出分频控制管脚 参考下表公式
[11:10]		保留
[9:8]	IN_DV[1:0]	PLL 输入分频控制管脚 参考下表公式.
[7:6]		保留
[4:0]	FB_DV[4:0]	PLL 反馈分频控制管脚 参考下表公式. FB_DV 的范围是从 0 到63.

输出时钟频率设定

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

符号	描述
FOUT	输出时钟频率 FOUT 频率必须介于48MHz与 120MHz之间。 48 MHz < FOUT < 120 MHz
FIN	输入(参考) 时钟频率
NR	IN_DV = "00" : NR = 2 IN_DV = "01" : NR = 4 IN_DV = "10" : NR = 8 IN_DV = "11" : NR = 16
NF	反馈分频器(FB_DV + 32)
NO	OUT_DV = 0 → NO = 1 OUT_DV = 1 → NO = 2 (不推荐)

一些推荐的设置

PLL 时钟输入源	PLL 时钟输入频率 (MHz)	PLL 时钟输出频率 (MHz)	PLLCTL 设置
HXT	4	120	0x1C
		96	0x10
		48	0x110
	8	128	0x120
		96	0x110
		48	0x210
	12	120	0x108
		96	0x220
		48	0x320
	16	128	0x220
		96	0x210
		48	0x310
	24	120	0x208
		96	0x200
		48	0x300
HIRC	12	120	0x20108
		96	0x20220
		48	0x20320

频率分频器控制寄存器(FRQDIV)

寄存器	偏移量	R/W	描述	复位后的值
FRQDIV	CLK_BA+ 0x28	R/W	频率分频器控制寄存器	0x0000_0000

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

Bits	描述	
[31:5]	-	保留
[4]	FDIV_EN	频率分频器使能位 0 = 禁止频率分频器 1 = 使能频率分频器
[3:0]	FSEL[3:0]	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} 是输入时钟频率, F_{out} 分频器的输出时钟频率, N是 FSEL[3:0].的4位值

模块时钟输出寄存器 (MCLKO)

寄存器	偏移量	R/W	描述	复位后的值
MCLKO	CLK_BA+ 0x2C	R/W	模块时钟输出寄存器	0x0000_0000

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

Bits	描述																					
[31:5]	-	保留																				
[7]	MCLK_EN	<p>模块时钟输出使能</p> <p>使用者可透过 MCLK_SEL 寄存器选择时钟源，并使能 MCLK_EN 后于 PC.0 管脚获得相应的模块时钟。</p> <p>0 = 模块时钟输出禁能</p> <p>1 = 模块时钟输出使能</p> <p>注：</p> <p>若设置此位，PC.0将会输出模块时钟，而PC0_MFP之设定将无效。</p>																				
[6]	-	保留																				
[5:0]	MCLK_SEL[5:0]	模块时钟输出源选择																				
		<table><tr><th>MCLK_SEL</th><th>Clock Source</th></tr><tr><td>00_0000</td><td>ISP_CLK</td></tr><tr><td>00_0001</td><td>HIRC</td></tr><tr><td>00_0010</td><td>HXT</td></tr><tr><td>00_0011</td><td>LXT</td></tr><tr><td>00_0100</td><td>LIRC</td></tr><tr><td>00_0101</td><td>PLL output</td></tr><tr><td>00_0110</td><td>PLL input</td></tr><tr><td>00_0111</td><td>System Tick</td></tr><tr><td>00_1000</td><td>HCLK clock</td></tr></table>	MCLK_SEL	Clock Source	00_0000	ISP_CLK	00_0001	HIRC	00_0010	HXT	00_0011	LXT	00_0100	LIRC	00_0101	PLL output	00_0110	PLL input	00_0111	System Tick	00_1000	HCLK clock
		MCLK_SEL	Clock Source																			
		00_0000	ISP_CLK																			
		00_0001	HIRC																			
		00_0010	HXT																			
		00_0011	LXT																			
		00_0100	LIRC																			
		00_0101	PLL output																			
		00_0110	PLL input																			
		00_0111	System Tick																			
00_1000	HCLK clock																					

Bits	描述
	00_1010 PCLK clock
	10_0000 TMR0_CLK
	10_0001 TMR1_CLK
	10_0010 UART0_CLK
	10_0011 USB_CLK
	10_0100 ADC_CLK
	10_0101 WDT_CLK
	10_0110 PWM0_CH01_CLK
	10_0111 PWM0_CH23_CLK
	10_1001 LCD_CLK
	11_1000 TMR2_CLK
	11_1001 TMR3_CLK
	11_1010 UART1_CLK
	11_1011 PWM1_CH01_CLK
	11_1100 PWM1_CH23_CLK
	11_1101 I2S_CLK
	11_1110 SC0_CLK
	11_1111 SC1_CLK

掉电唤醒中断状态寄存器(PD_WK_IS)

寄存器	偏移量	R/W	描述	复位后的值
PD_WK_IS	CLK_BA+ 0x30	R	掉电唤醒中断状态寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[0]	PD_WK_IS	<p>在芯片掉电模式下的唤醒中断状态</p> <p>该位表示某事件使芯片从掉电模式中唤醒</p> <p>如果外部中断, UART, GPIO, RTC, USB, SPI, Timer, WDT, 和BOD 唤醒发生, 该位被置1。.</p> <p>向该位写1清0</p>

5.5 模数转换器 (ADC)

5.5.1 概述

这个芯片包含一个12位逐位逼近模数转换器(SAR A/D 转换器)，有12个外部输入通道，6个内部通道。A/D转换器支持三种工作模式：单次模式，单周期扫描模式和连续扫描模式。A/D转换可以被软件，外部管脚 STADC/PB.8和定时事件触发。

注意在使能ADC功能之前，这些用作ADC模拟输入的IO口必须被设置成输入模式，并且要关闭数字功能(GPIOA_OFFD)。

5.5.2 特征

- 模拟输入电压范围: 0~参考值(最大到3.6V).
- 可选择 12-bits, 10-bits, 8-bits 和 6-bits 方案.
- 支持采样时间设置 (用 ADC_CLK 单位)，通道0~11可分别设置采样时间，通道 12~17 共用一个采样时间设置。
- 支持两种掉电模式
 - ◆ Power down 模式
 - ◆ Stand by 模式
- 最大到12 外部模拟输入通道 (通道0 ~ 通道11) 和 6个内部通道(通道12~通道17)。6个内部通道转换6个电压源，分别是 DAC0，DAC1，内部带隙电压，内部温度传感器输出，AVDD和 AVSS。
- 最大 ADC 时钟频率是 42 MHz，每次转换时间是19 个时钟+ 输入阻抗决定的采样时间。
- 三种工作模式
 - ◆ 单一模式：A/D 转换在指定通道完成一次
 - ◆ 单周期模式：A/D 转换在所有指定通道完成一个周期，转换顺序从最小号通道到最大号通道
 - ◆ 连续扫描模式：A/D 转换器持续执行单周期扫描直到软件停止 A/D 转换
- A/D 转换开始条件
 - ◆ 软件向 ADST 位写1
 - ◆ 外部管脚STADC
 - ◆ 选择四个时间事件中的一个 (TMR0, TMR1, TMR2 和 TMR3) 这些时间事件可以使能ADC 和通过PDMA传输AD 转换结果
- 每个通道转换结果存储在数据寄存器内
- 转换结果可和指定的值相比较，当转换值和比较寄存器中的设定值相等时，用户可以选择是否产生一个中断请求。
- 支持校验 和加载校验字能力。

5.5.3 框图

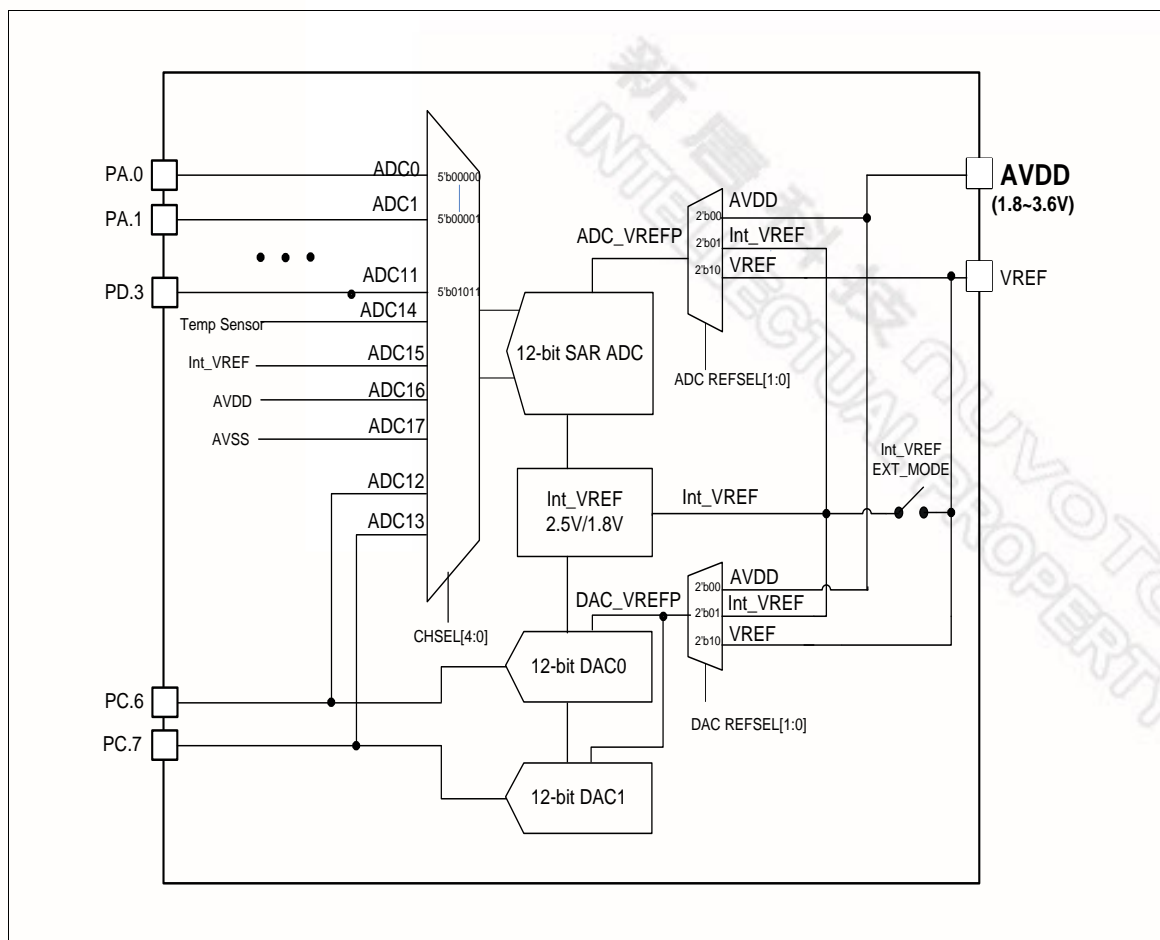


图 5-5 ADC 和 DAC 框图

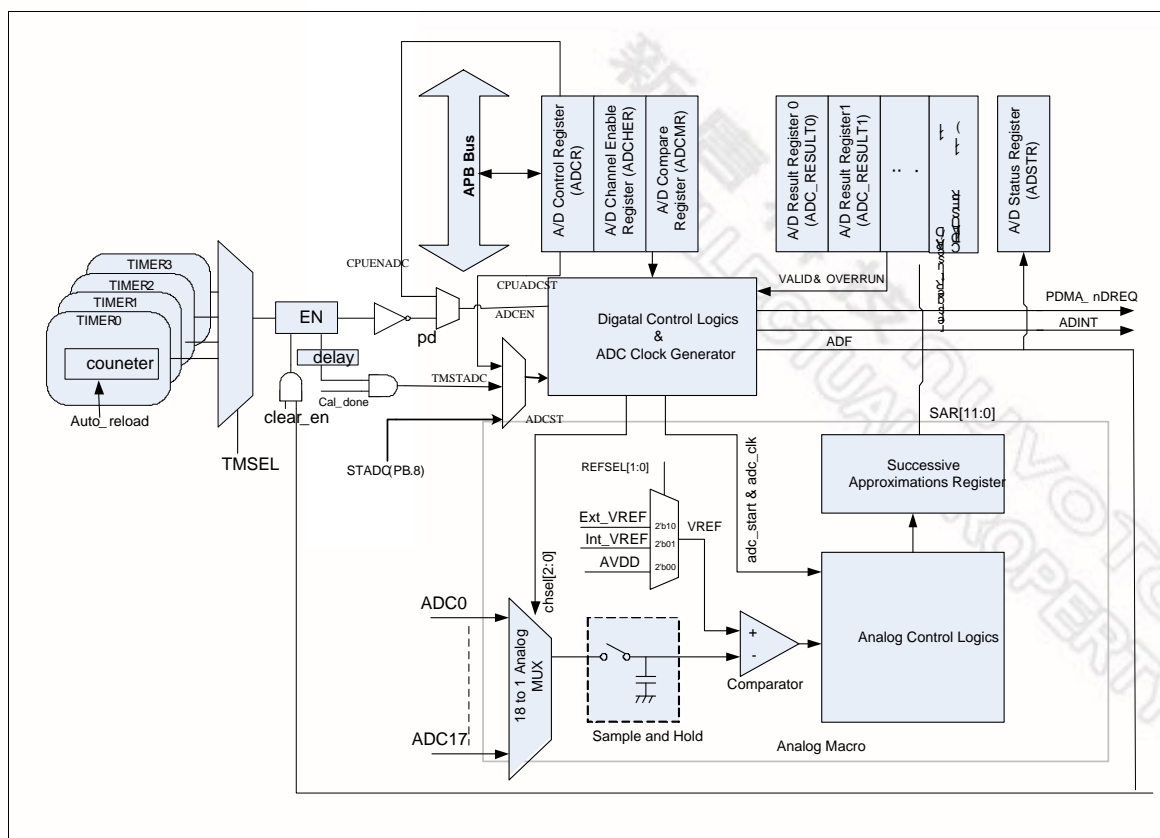


图 5-6 ADC 转换器框图

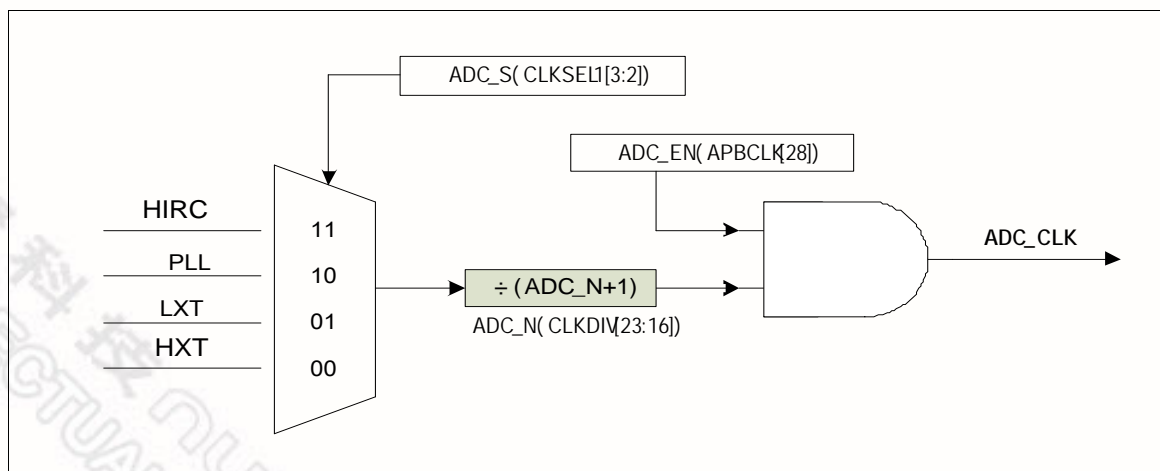


图 5-7 ADC 时钟控制

5.5.4 功能描述

A/D 转换器为 12 位逐次逼近式。ADC 有 3 种操作模式：单一模式、单周期扫描模式和连续扫描模式。当改变运行模式或模拟输入通道时，为了防止错误的操作，软件必须清 ADCR 寄存器中的

ADST 位为 0。在ADST被清0后，A/D转换器会立即放弃当前的转换进入空闲状态。在一些要求省电的应用，ADC可以通过一个时间信号使能并在一个延时间隔之后开始转换，并且在转换完成通过PDMA传送转换数据到内存之后可以进入掉电状态。通过设置ADCON寄存器中TMSEL[1:0]，有四个时钟源(Timer0~3)使能ADC。

5.5.4.1 单一模式

在单一模式下，A/D 转换在指定通道只执行一次，操作如下：

当 ADCR 的 ADST 位被软件或者外部触发输入或是 ADCR 寄存器中TMSEL[1:0]选择的时间结束事件置位时，A/D 转换开始。

当A/D 转换完成，转换值将存储在与通道相对应的 A/D 数据寄存器。

A/D 转换完成，ADSR 的 ADF 位置位。若此时 ADCR 寄存器的 ADIE 位被置位，则将产生 ADC 中断请求。

在 A/D 转换期间，ADST 位保持为1。当 A/D 转换结束，ADST 位自动清 0，A/D 转换器进入空闲状态。如果 ADST位在 A/D 转换期间被软件清0，A/D 会立即停止进入空闲状态。

在前一次转换结束后，下一次转换重复1-4的方法。

PS: 如果 ADC时钟比 PCLK低很多，开始下一次转换前，至少等待一个 ADC 时钟

注：如果在单一模式下，软件使能多于一个通道，则编号最小的通道将被选中，其他通道将被忽略。

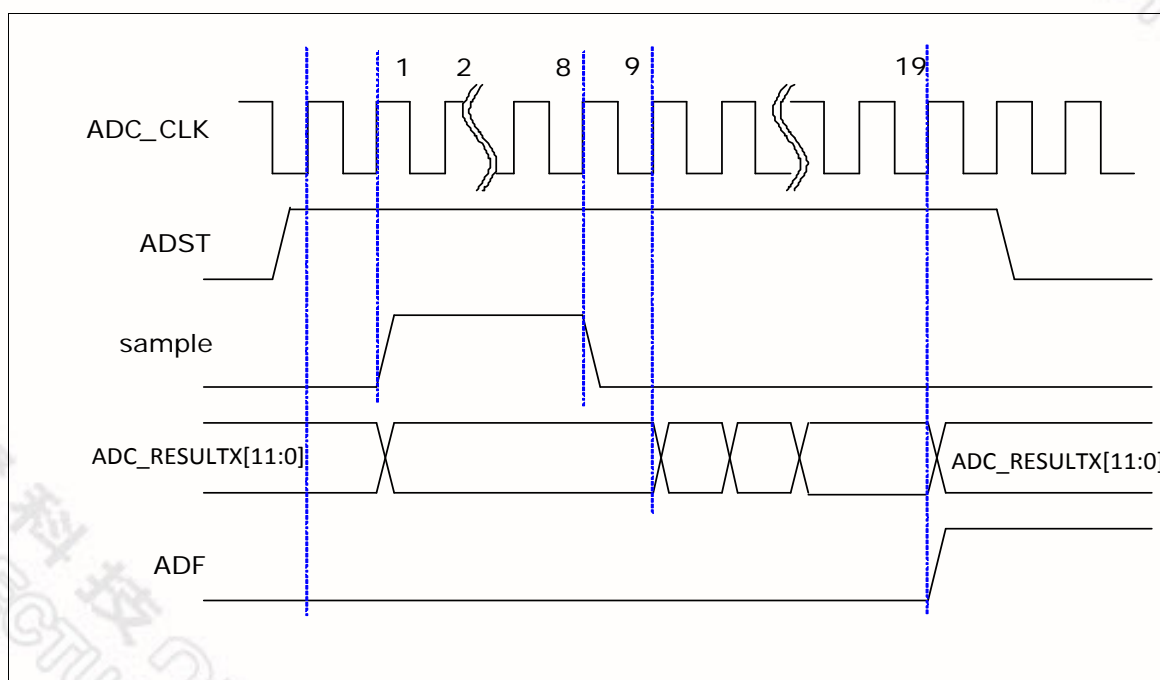


图 5-8 ADC 单一模式转换时序图

5.5.4.2 单周期扫描模式

在单周期扫描模式下，ADC 会对所有指定的通道进行一次采样与转换，且从编号最小的通道到编号最大的通道依次开始。操作如下：

当 ADCR 的 ADST 位被软件或者外部触发输入或是 ADCR 寄存器中TMSEL[1:0]选择的时间结束事件置位时，A/D 转换从最小编号的通道开始。

每路 A/D 转换完成后，A/D 转换数值将传输到相应的 A/D 数据寄存器中。

当所有被选择的通道转换都完成后，ADF 位 (ADSR 寄存器) 置1。若此时 ADC 中断功能使能，则将产生 ADC 中断。

在 A/D 转换期间，ADST 位保持为1。当 A/D 转换结束，ADST 位自动清 0，A/D 转换器进入空闲状态。如果 ADST 位在 A/D 转换期间被软件清0，A/D 会立即停止进入空闲状态。

下面是一个单周期时序图的例子

(在这个例子中，通道0,2,3 和7 被使能)

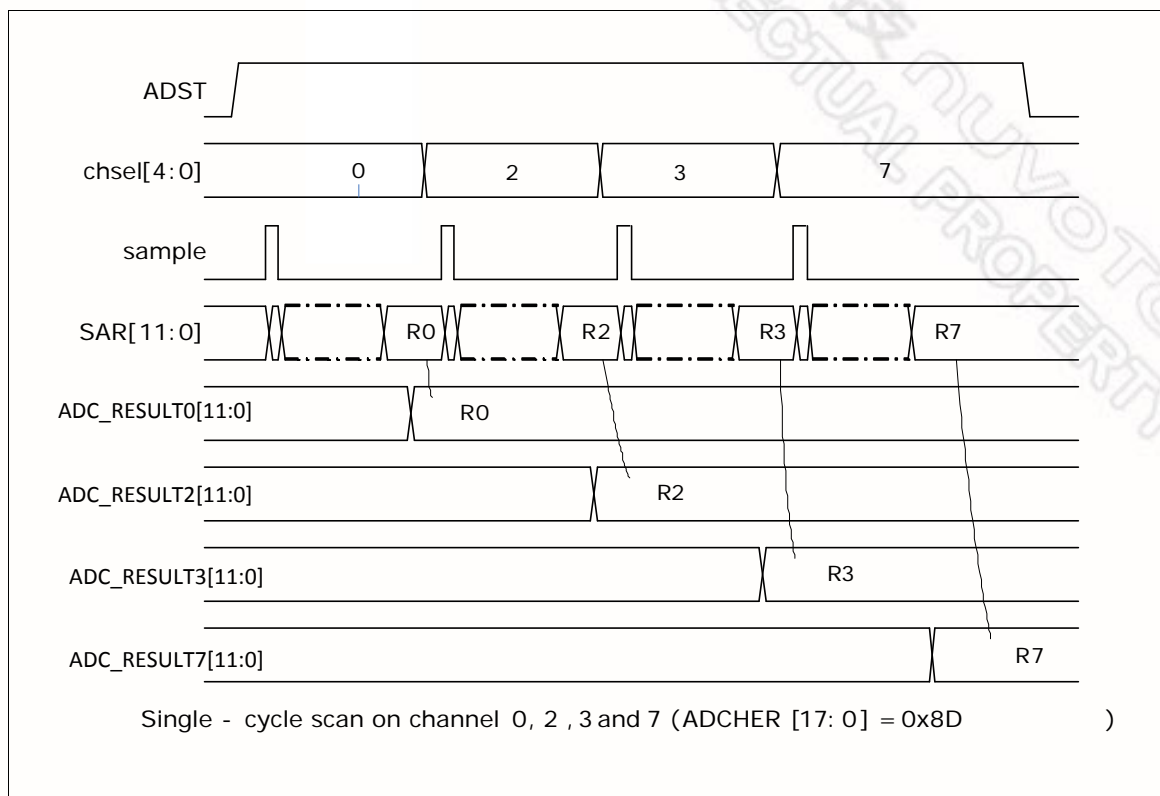


图 5-9 ADC 在使能通道上单周期扫描模式的时序图

5.5.4.3 连续扫描模式

在连续扫描模式，A/D 转换器会连续对所有由 CHEN 位 (ADCHER 寄存器) 使能指定的通道（最多 8 个外部通道，一个用于ADC，DAC0或 DAC1的内部通道）执行转换。操作如下：

1. 当 ADCR 的 ADST 位被软件或者外部触发输入或是 ADCR 寄存器中 TMSEL[1:0]选择的时间结束事件置位时，A/D 转换从最小编号的通道开始。
2. 每路使能的 A/D 转换完成后，A/D 转换数值将装载到相应的 A/D 数据寄存器中。
3. 当所有被使能的通道依次完成一次 A/D 转换时，ADF 位 (ADSR[0]) 将被置位。如果此时 ADC 中断功能已使能，则产生 ADC 中断。
4. 接着步骤3，重新转换使能的最小编号通道。
5. 只要 ADST 保持为1，就重复步骤 2 到步骤 4。当清 ADST 为 0，AD 转换停止，进入空闲状态。

一个使能通道 (0, 2, 3 和 7) 的连续扫描模式时序图示例如下:

(在这个例子中, 通道 0, 2, 3 和 7被使能.)

(这个例子仅对ADC适用.)

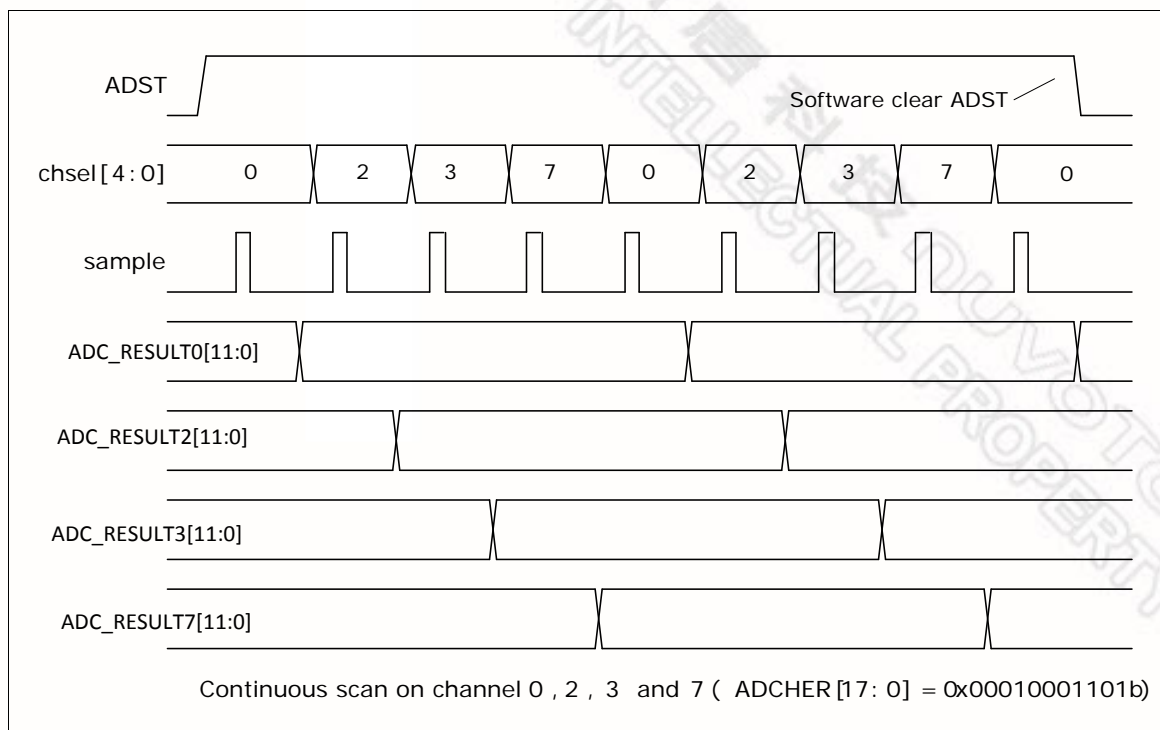


图 5-10 ADC 使能通道上的连续扫描模式时序图

5.5.4.4 外部触发ADC 转换

A/D 转换可由外部管脚请求触发。当 ADCR 的 TRGEN 置为高来使能 ADC外部触发功能, 通过设定 TRGS[1:0] 位为 00b 选择从 STADC 管脚进行外部输入触发, 并可由软件设定 TRGCOND[1:0] 选择上升/下降沿或低/高电平触发。一个8位的计数器用来抗尖峰脉冲。若选择电平触发 (level trigger) 条件, STADC 管脚需要保持默认状态至少 8个 PCLKs。ADST 位将在第九个 PCLK 时被置 1 并且开始转换。在电平触发模式状态下, 如果外部触发输入维持在有效状态, 转换将会持续进行。仅当外部触发条件消失才停止。若选择边缘触发模式, 高和低状态至少需保持 4 个PCLKs, 低于该值的脉冲将被忽略。

5.5.4.5 比较功能监视转换结果

ADC 控制器提供两组比较寄存器 ADCMPR0 和 ADCMPR1, 用于监控 A/D 转换控制器 (最多支持) 2 路通道的转换结果值, 参考图 5-11。软件可通过设定 CMPCH(ADCMPRx[5:0]) 来选择监控哪路通道, 而 CMPCOND 位被用来检查转换值小于指定值或大于 (等于) CMPD[11:0] 的指定值。当 CMPCH 指定的通道转换完成时, 会自动的触发一次比较行为。当比较结果和设定值相匹配, 比较匹配计数器将加1, 否则, 比较匹配计数器将会被清0。当计数器的值和设定值 (CMPMATCNT+1) 匹配, CMPF 位将置 1。如果 CMPIE 置位, 将产生 ADC_INT 中断请求。在扫描模式下无需软件介入就可用其监控外部模拟输入管脚电压变化。具体逻辑框图如下:

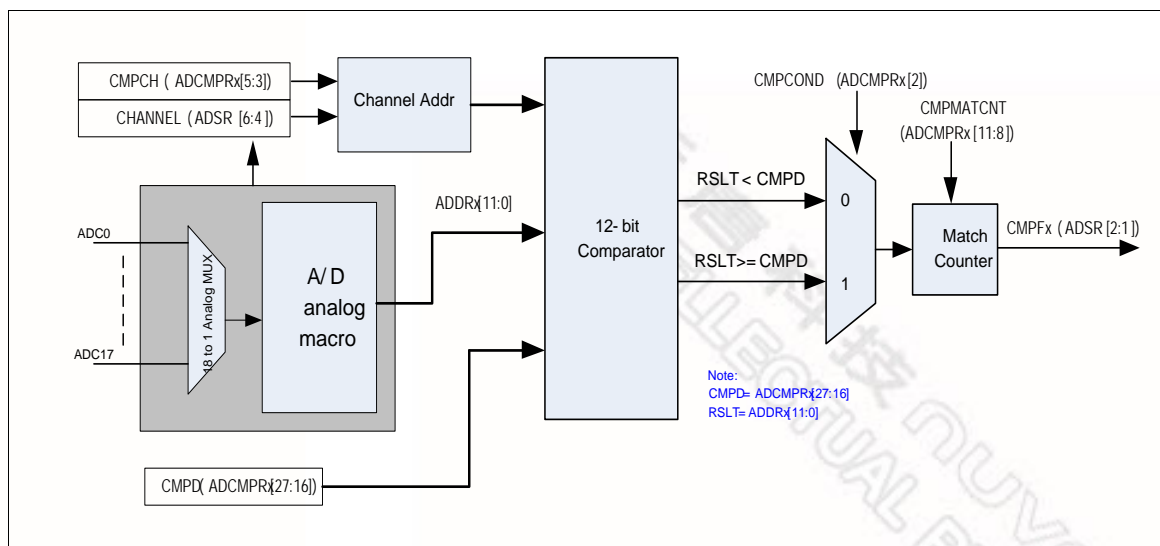


图 5-11 ADC 转换结果监控逻辑图

5.5.4.6 中断源

当某个 ADC 操作模式结束其转换时，A/D 转换的结束标志 ADF 位将被置 1。如果 ADCR 中的 ADIE 位被置 1，会产生一个转换结束中断。当 CMPIE 使能并且 A/D 转换结果满足 ADCMPR 寄存器的设定，产生监测中断。ADINT 会被置位。CPU 清 0 CMPF 和 ADF 停止中断请求。

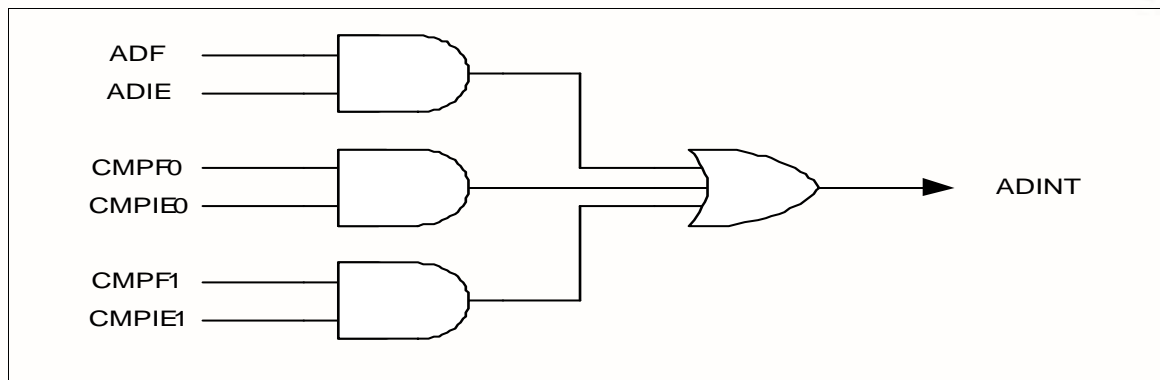


图 5-12 ADC 控制器中断

5.5.4.7 外设 DMA 请求

当 A/D 转换完成时，结果将被装载到 AD_RESULTx (x=0~10) 寄存器且 VALID 位被置 1。如果 ADCR 寄存器的 PTEN 位被置位，ADC 控制器将产生请求到 PDMA。用户可使用 PDMA 将转换结果传输到用户指定的内存空间，而无需 CPU 参与。

5.5.4.8 通过时间结束事件使能 ADC

用户可以通过设置 ADCR 寄存器中的 TMSEL，TMTRGMOD 和 TMPDMACNT 来配置 ADC 使用时间事件触发。当 ADC 进入掉电模式，可以被时间事件使能。时间事件源可以通过 ADCR 寄存器中 TMSEL 选择。

在 ADC 唤醒之后，开始通过 PDMA 传输 ADC_RESULT 到内存。用户应该配置 ADCR 寄存器中 PTEN 使能 PDMA 传输并且指定 ADMD（连续扫描模式，单一模式，单循环模式）来执行 ADC，配置

ADCR 寄存器中TMPDMACNT来设定PDMA一次传输ADC_RESULT 到内存的数量。PDMA传输了设定数量的ADC_RESULT, ADC会进入掉电模式, 直到下次时间事件的到来。

所有的 ADC_RESULT 传输到内存之后, ADC 会进入掉电模式, 这时候ADC不会再被接下来的时间结束事件唤醒。

所有的配置必须系统进入掉电模式之前完成, 因为在整个系统进入掉电模式的时候, CPU不能读写寄存器。在单周期模式和单次模式PDMA传输计数器应该和使能的通道计数器保持一致。

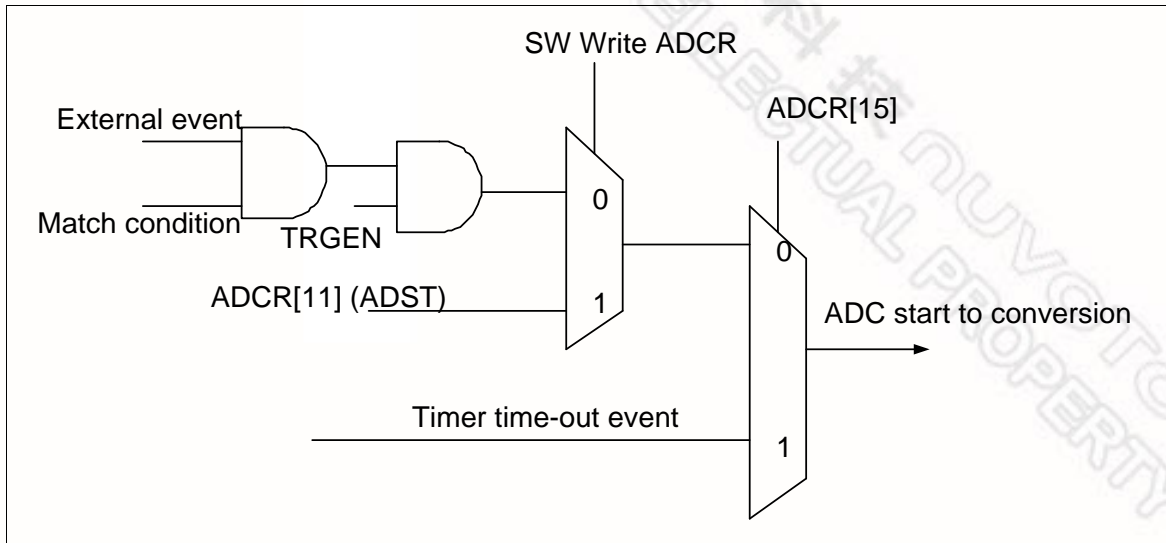


图 5-13 ADC 开始转换的条件

5.5.4.9 ADC 采样时间

下图表示S/H (采样和保持)输入网络的简化等效电路, C_s 是存储电容, R_s 是采样开关的阻抗, R_i 是信号源 V_i 的输出阻抗。图5-16显示在转换周期 n 结束后转换周期 $j+1$ 立即开始的情况。在这种情况下, 采样相位的时间为大约 $1.5 \times \text{ADC_CLK}$ 。 C_s 必须在采样相位被充电, 而且必须保证在采样相位结束时, C_s 的值足够接近 V_i 。为了保证上述要求, R_i 的值不会采用任意大的值。

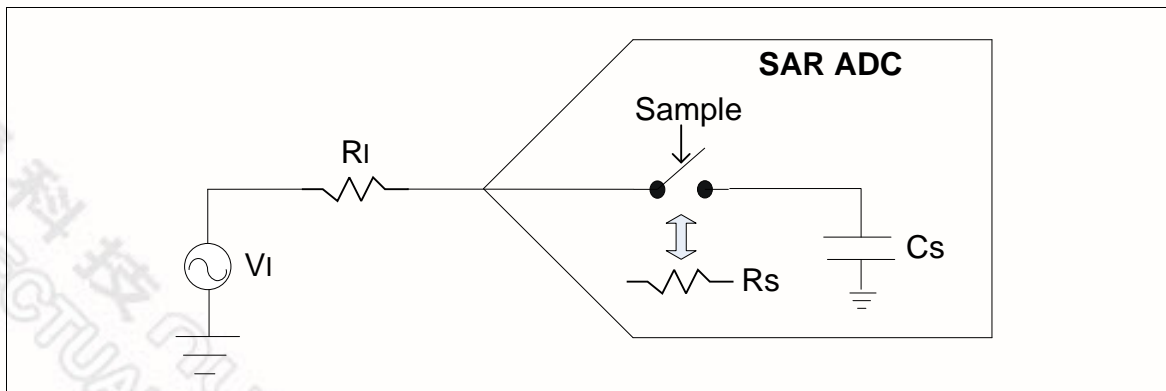


图 5-14 采样网络模型

图5-15表示采样时间如何被增加, 为了允许低驱动能力信号源的操作, `adc_start` 信号被延时需要的时钟周期来确保输入信号采样准确。在此期间, 通道的选择必须保持不变, 注意这个操作会减少有效的采样率。

ADC有两种类型的输入: 通道0~5是快速输入, 通道6~17是慢速输入。所有的输入有相同的功能,

但是在采样期间，通道0~5输入比通道6~17更快。

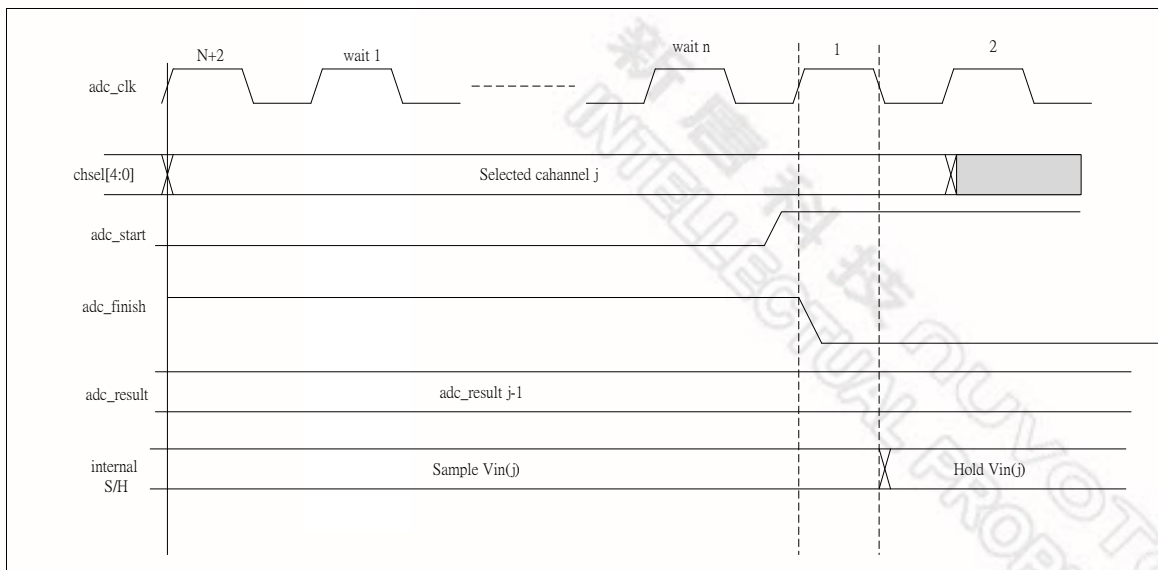


图 5-15 递增采样时间波形

对两种类型的输入，都可能获得最大的采样频率，但是在某种特定的情况下（取决于分辨率模式或信号源的输出阻抗），采样时间应该被延时需要时钟周期来确保采样的准确（上图）。

下图表示额外采样和保持周期数，对一个宽范围的 R_i （信号源输出阻抗）值是必须的，对ADC所有的输入通道取决于分辨率模式（RESSEL）。使用采样计数器寄存器（ADCCHSAMP0 和 ADCCHSAMP1）来增加额外的采样和保持周期。

请注意这些图涉及到的额外采样保持时钟周期，在 $n=0$ 的情况下，采样周期是 $1.5 \times ADC_CLK$ 。

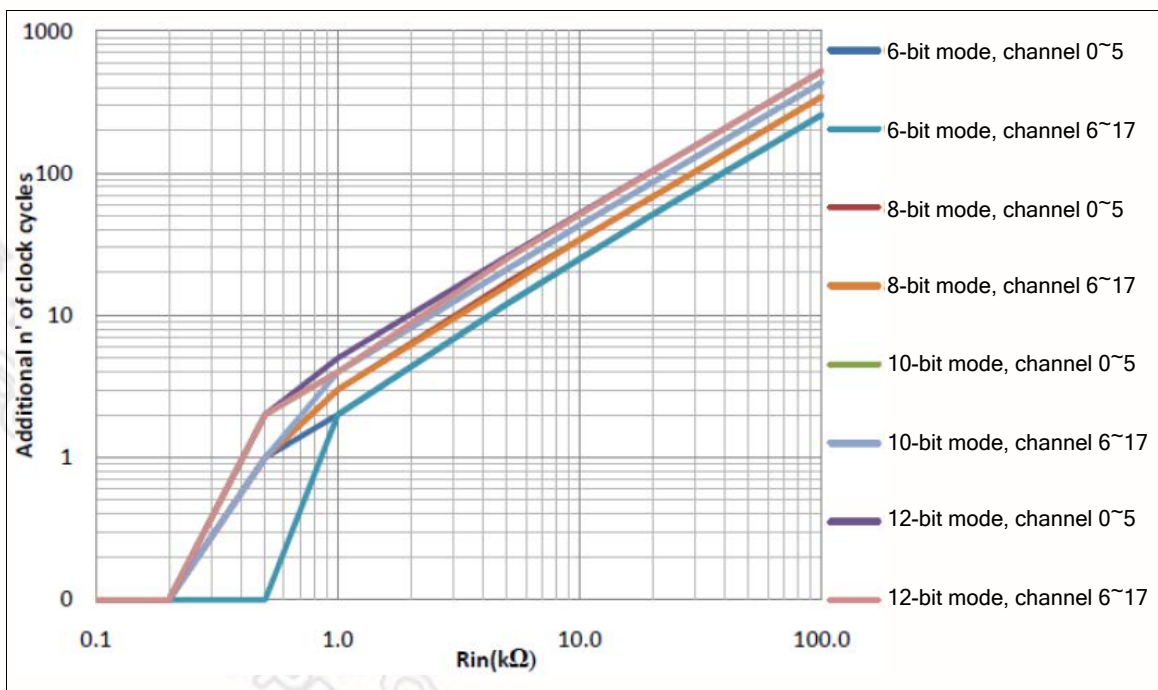


图 5-16 额外采样和保持时钟周期作为信号源输出阻抗 $R_i(k\Omega)$ 的功能。在该图中给出的结果是在额定

条件下测定的 (typical process corner, $V_{DD} = AV_{DD} = 3.3\text{ V}$, LDO output = 1.8 V, $T_{\text{junction}} = 50^{\circ}\text{C}$, $\text{ADC_CLK} = 42\text{ MHz}$, $V_{\text{REF}} = AV_{DD}$)

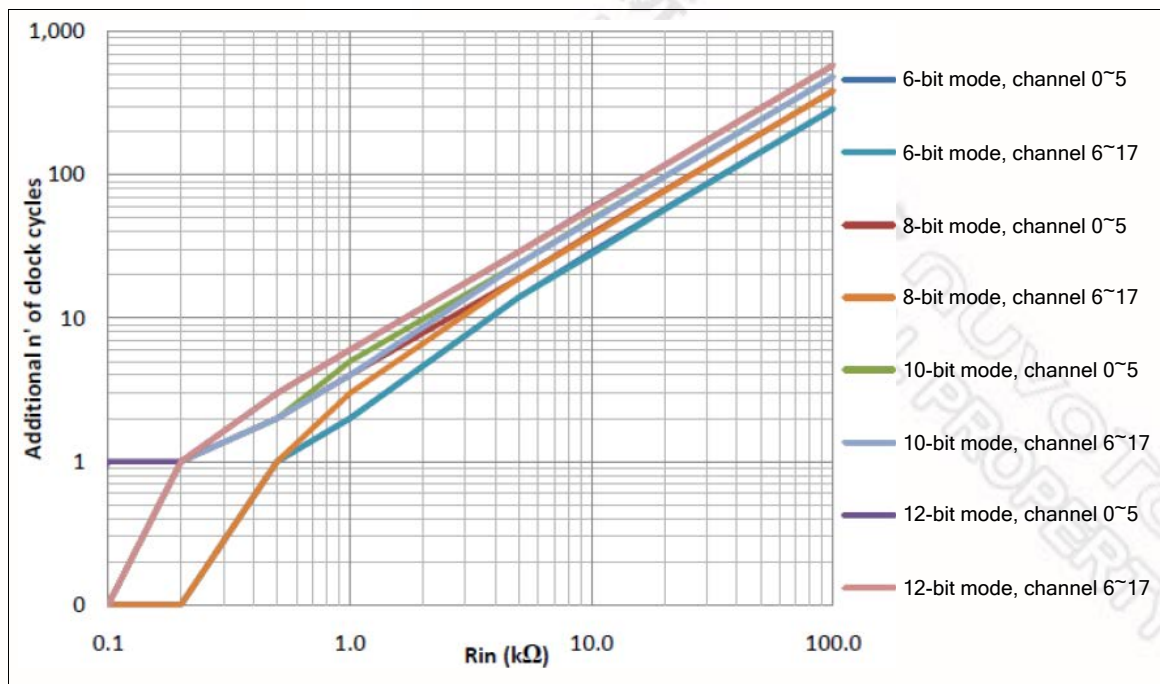


图 5-17 额外采样和保持时钟周期 作为信号源输出阻抗 $R_i(\text{k}\Omega)$ 的功能。在该图中给出的结果是在最坏情况的条件下测定的 (slow process corner, $V_{DD} = AV_{DD} = 1.8\text{ V}$, LDO output = 1.62 V, $T_{\text{junction}} = -40^{\circ}\text{C}$, $\text{ADC_CLK} = 42\text{ MHz}$, $V_{\text{REF}} = AV_{DD}$)

5.5.4.10 ADC 掉电模式

用户可以选择两种掉电模式，分别是power down模式和standby模式，用户可以在禁止ADCR寄存器中ADEN之前，配置ADCPWD寄存器中PWDMOD来决定使用哪种掉电模式。

在不同的掉电模式下，上电顺序也不相同，用户应该知道当前的掉电模式，并在使能ADCR寄存器中ADEN之前，配置ADCPWD寄存器中PWDMOD来决定使用哪种上电顺序。如果上电顺序错误，ADC将会功能错误。

掉电模式之间的区别是耗电量和从各个掉电模式重新开始的稳定时间，最少的耗电量是power down模式，然后是standby模式，稳定时间顺序相反。在ADC进入掉电之前，确保ADC已经停止(通过禁止ADST)，且所有的转换已经完成(通过查询ADF)。

5.5.4.11 ADC 偏移量校准

为了减少电子随机噪音的影响，ADC执行校准来获得测量的平均偏移量。然后，在正常操作下，数字模块把校准得到的校准字应用于内部ADC电容网络，以消除偏移电压。

用户可以设置CALEN位为高，选择CALSEL为1（执行校准），并写1到CALSTART，然后等待CALDONE位为高，当CALDONE为高，校准完成，校准字存放在ADC_CALWORD寄存器中。

用户也可以加载指定的校准字到ADC_CALWORD寄存器，以节省校准完成时间。除了设置CALSEL为1，其他设置与上述方式一样，该方式也需要等待CALDONE位变成高，当CALDONE为高，加载校准字完成，加载的校准字现在可以应用于ADC。

5.5.4.12 可选的分辨率

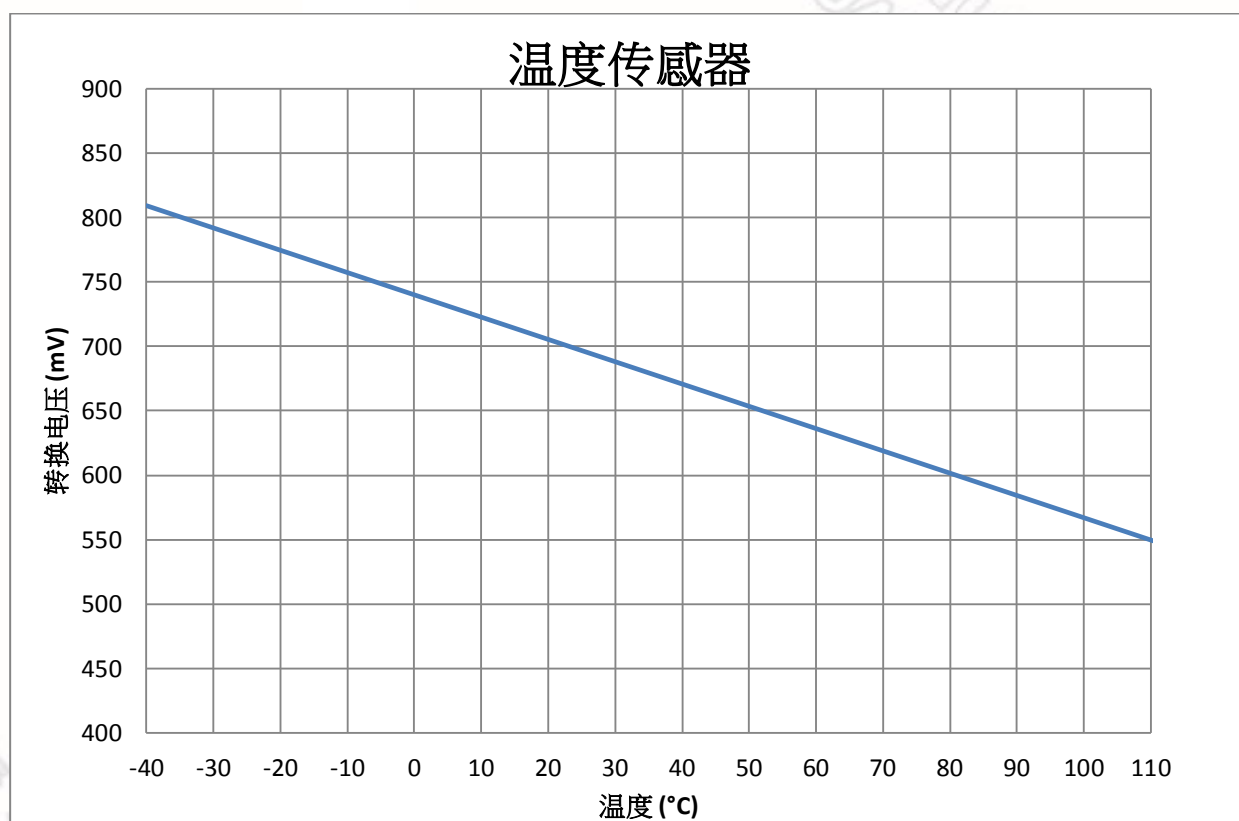
ADC 有 12, 10, 8 和 6 位等四种可选的分辨率，用户可以通过设定 ADCR 寄存器的 RESSEL 位域来选择 ADC 的分辨率。

分辨率选择仅能在转换结束 (ADF 变为高) 之后被更新，不同的分辨率需要不同的转换周期，以 12 位分辨率为例，转换完成一个通道需要 19 个 ADC 时钟周期；而在 10 位分辨率，转换完成一个通道需要 17 个 ADC 时钟周期；在 8 位分辨率需要 15 个 ADC 时钟周期；在 6 位分辨率需要 14 个 ADC 时钟周期。

5.5.4.13 温度传感器

下图显示温度传感器的转换函数结果，输出电压(Vtemp)之转换结果如下公式。

$$V_{temp} (mV) = -1.73 (mV/^{\circ}C) \times Temperature (^{\circ}C) + 740 (mV).$$



5.5.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
ADC_BA = 0x400E_0000				
ADC_RESULT0	ADC_BA+0x00	R	A/D 数据寄存器 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D 数据寄存器 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D 数据寄存器 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0C	R	A/D 数据寄存器 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D 数据寄存器 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D 数据寄存器 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D 数据寄存器 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1C	R	A/D 数据寄存器 7	0x0000_0000
ADC_RESULT8	ADC_BA+0x20	R	A/D 数据寄存器 8	0x0000_0000
ADC_RESULT9	ADC_BA+0x24	R	A/D 数据寄存器 9	0x0000_0000
ADC_RESULT10	ADC_BA+0x28	R	A/D 数据寄存器 10	0x0000_0000
ADC_RESULT11	ADC_BA+0x2C	R	A/D 数据寄存器 11	0x0000_0000
ADC_RESULT12	ADC_BA+0x30	R	A/D 数据寄存器 12	0x0000_0000
ADC_RESULT13	ADC_BA+0x34	R	A/D 数据寄存器 13	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D 数据寄存器 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3C	R	A/D 数据寄存器 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D 数据寄存器 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D 数据寄存器 17	0x0000_0000
ADCR	ADC_BA+0x48	R/W	A/D 控制寄存器	0x0001_0000
ADCHER	ADC_BA+0x4C	R/W	A/D 通道使能寄存器	0x0000_0000
ADCMPR0	ADC_BA+0x50	R/W	A/D 比较寄存器 0	0x0000_0000
ADCMPR1	ADC_BA+0x54	R/W	A/D 比较寄存器 1	0x0000_0000
ADSR	ADC_BA+0x58	R/W	A/D 状态寄存器	0x0000_0000
ADPDMA	ADC_BA+0x60	R	ADC PDMA 当前传输数据	0x0000_0000
ADCPWD	ADC_BA +0x64	R/W	ADC电源管理寄存器	0x0001_E002
ADCCALCTL	ADC_BA +0x68	R/W	ADC 校准控制寄存器	0x0000_0009
ADCCALWORD	ADC_BA +0x6C	R/W	校准字寄存器	0xFFFF_FFFF

ADCCHSAMP0	ADC_BA +0x70	R/W	ADC 通道采样时间计数器寄存器组0	0x0000_0000
ADCCHSAMP1	ADC_BA +0x74	R/W	ADC通道采样时间计数器寄存器组1	0x0000_0000

5.5.6 寄存器 描述

A/D数据寄存器(ADC RESULT0~ ADC RESULT10)

寄存器	偏移量	R/W	描述	复位后的值
ADC_RESULT0	ADC_BA+0x00	R	A/D 数据寄存器 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D 数据寄存器 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D 数据寄存器 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0c	R	A/D 数据寄存器 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D 数据寄存器 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D 数据寄存器 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D 数据寄存器 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1c	R	A/D 数据寄存器 7	0x0000_0000
ADC_RESULT8	ADC_BA+0x20	R	A/D 数据寄存器 8	0x0000_0000
ADC_RESULT9	ADC_BA+0x24	R	A/D 数据寄存器 9	0x0000_0000
ADC_RESULT10	ADC_BA+0x28	R	A/D 数据寄存器 10	0x0000_0000
ADC_RESULT11	ADC_BA+0x2c	R	A/D 数据寄存器 11	0x0000_0000
ADC_RESULT12	ADC_BA+0x30	R	A/D 数据寄存器 12	0x0000_0000
ADC_RESULT13	ADC_BA+0x34	R	A/D 数据寄存器 13	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D 数据寄存器 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3c	R	A/D 数据寄存器 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D 数据寄存器 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D 数据寄存器 17	0x0000_0000

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

Bits	描述	
[31:18]	-	保留
[17]	OVERRUN	Over Run 标志 该位是 ADC_RESULTx中 OVERRUN 位的镜像
[16]	VALID	数据有效标志 该位是 ADC_RESULTx中 VALID 位的镜像
[15:12]	-	保留
[11:0]	RSLT	A/D 转换结果 该域包含 12 位的转换结果.

A/D控制寄存器(ADCR)

寄存器	偏移量	R/W	描述	复位后的值
ADCR	ADC_BA+0x30	R/W	ADC控制寄存器	0x0001_0000

31	30	29	28	27	26	25	24
TMPDMACNT							
23	22	21	20	19	18	17	16
-				RESSEL		REFSEL	
15	14	13	12	11	10	9	8
TMTRGMOD	-	TMSEL[1:0]		ADST	DIFF	PTEN	TRGEN
7	6	5	4	3	2	1	0
TRGCOND		TRGS		ADMD		ADIE	ADEN

Bits	描述									
[31:24]	TMPDMACNT[7:0]	PDMA 计数 当每次定时事件发生PDMA会传输TMPDMACNT +1个 ADC结果 注意: PDMA 传输数据的总数应该被设置在PDMA字节计数寄存器， 当 PDMA 完成被设置，,ADC不会使能并开始传输即使定时事件发生。								
[23:20]	-	保留								
[19:18]	RESSEL	分辨率选择 <table><tr><td>00</td><td>6 bit</td></tr><tr><td>01</td><td>8 bit</td></tr><tr><td>10</td><td>10 bit</td></tr><tr><td>11</td><td>12 bit</td></tr></table>	00	6 bit	01	8 bit	10	10 bit	11	12 bit
00	6 bit									
01	8 bit									
10	10 bit									
11	12 bit									
[17:16]	REFSEL[1:0]	参考电压源选择 <table><tr><td>00</td><td>选择电源作为参考电压</td></tr><tr><td>01</td><td>选择 Int_VREF 作为参考电压</td></tr><tr><td>10</td><td>选择VREF作为参考电压</td></tr></table>	00	选择电源作为参考电压	01	选择 Int_VREF 作为参考电压	10	选择VREF作为参考电压		
00	选择电源作为参考电压									
01	选择 Int_VREF 作为参考电压									
10	选择VREF作为参考电压									
[15]	TMTRGMOD	时间结束事件触发 ADC 转换 1 = 通过时间结束事件使能ADC 设定TMSEL 从timer 0~3选择时间结束事件 0 = 禁止这个功能								
[14]	-	保留								

Bits	描述									
[13:12]	TMSEL[1:0]	<div>选择A/D 使能时间结束源</div> <table><tr><td>00</td><td>TMR0</td></tr><tr><td>01</td><td>TMR1</td></tr><tr><td>10</td><td>TMR2</td></tr><tr><td>11</td><td>TMR3</td></tr></table>	00	TMR0	01	TMR1	10	TMR2	11	TMR3
00	TMR0									
01	TMR1									
10	TMR2									
11	TMR3									
[11]	ADST	<div>A/D 转换开始</div> <div>1 = 转换开始</div> <div>0 = 转换停止 A/D 转换器进入空闲阶段.</div> <div>ADST 位置位有下列 2 种方式：软件设定和外部管脚 STADC 触发。单一模式和单周期模式下，在转换结束后，ADST 将被硬件自动清除。在连续扫描模式下, A/D 转换将一直进行直到软件向该位写 0 或芯片复位。</div> <div>注意:ADC转换完成后，在再次设置这位为高之前，SW需要等待至少一个 ADC 时钟。</div>								
[10]	DIFF	<div>差分模式选择</div> <div>1:ADC 操作在差分模式</div> <div>0:ADC操作在单端模式</div> <div>注意:每次单端和差分模式切换都应该校准</div>								
[9]	PTEN	<div>PDMA 转换使能</div> <div>1 = 使能PDMA 传输数据 ADC_RESULT 0~17</div> <div>0 = 禁止PDMA 传输数据</div> <div>当 A/D转换完成，转换之后的数据被加载到 ADC_RESULT 0~17，软件可以使能这一位产生 PDMA 数据传输请求。</div> <div>当PTEN=1，软件必须设置ADIE=0来禁止中断。PDMA 可以通过时钟或单次传输模式进入 ADC_RESULT 0-17 寄存器。</div>								
[8]	TRGE	<div>外部触发使能</div> <div>使能或禁止触发 A/D 转换通过外部STADC 管脚.</div> <div>1= 使能</div> <div>0= 禁止</div>								
[7:6]	TRGCOND[1:0]	<div>外部触发条件</div> <div>该 2 位决定外部管脚 STADC 触发状态为电平或边沿。该信号必须保持至少 8 PCLKs 的稳定状态用于电平触发，4 PCLKs 的高和低状态用于边沿触发</div> <table><tr><td>00</td><td>低电平</td></tr><tr><td>01</td><td>高电平</td></tr><tr><td>10</td><td>下降沿</td></tr><tr><td>11</td><td>上升沿</td></tr></table>	00	低电平	01	高电平	10	下降沿	11	上升沿
00	低电平									
01	高电平									
10	下降沿									
11	上升沿									

Bits	描述									
[5:4]	TRGS[1:0]	硬件触发源								
		<table><tr><td>00</td><td>A/D 转换由外部 STADC 管脚触发开始</td></tr><tr><td>其他</td><td>保留</td></tr></table>	00	A/D 转换由外部 STADC 管脚触发开始	其他	保留				
		00	A/D 转换由外部 STADC 管脚触发开始							
其他	保留									
改变 TRGS 之前，软件必须禁用 TRGEN 和 ADST。 在硬件触发模式下，在硬件触发模式下，ADST 位由来自 STADC 的外部触发设定，然而软件有最高的优先权，在任何时候都可以置位或清0 ADST 位										
[3:2]	ADMD[1:0]	A/D 转换操作模式								
		<table><tr><td>00</td><td>单一转换</td></tr><tr><td>01</td><td>保留</td></tr><tr><td>10</td><td>单周期扫描</td></tr><tr><td>11</td><td>连续扫描</td></tr></table>	00	单一转换	01	保留	10	单周期扫描	11	连续扫描
		00	单一转换							
		01	保留							
10	单周期扫描									
11	连续扫描									
[1]	ADIE	A/D 中断使能 1 = 使能A/D 中断功能 0 = 禁止A/D 中断功能 如果 ADIE 位被置为1，则 A/D 转换结束将产生中断请求。								
[0]	ADEN	A/D转换器使能 1 = 使能 0 = 禁止 在开始 A/D 转换功能前，该位需置位。清除该位为 0 将禁用 A/D 转换模拟电路从而节省功耗。								

A/D 通道使能寄存器 (ADCHER)

Register	偏移量	R/W	描述	复位后的值
ADCHER	ADC_BA+0x34	R/W	A/D 通道使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-						CHEN17	CHEN16
15	14	13	12	11	10	9	8
CHEN15	CHEN14	CHEN13	CHEN12	CHEN11	CHEN10	CHEN9	CHEN8
7	6	5	4	3	2	1	0
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	描述	
[31:18]	保留	-
[17]	CHEN17	模拟输入通道 17使能(转换 AVSS) 1 = 使能 0 = 禁止
[16]	CHEN16	模拟输入通道 16使能(转换 AVDD) 1 = 使能 0 = 禁止
[15]	CHEN15	模拟输入通道 15使能(转换 Int_VREF) 1 = 使能 0 = 禁止
[14]	CHEN14	模拟输入通道 14使能(转换 VTEMP) 1 = 使能 0 = 禁止
[13]	CHEN13	模拟输入通道 13使能(转换 DAC1 输出电压) 1 = 使能 0 = 禁止
[12]	CHEN12	模拟输入通道 12使能(转换 DAC0 输出电压) 1 = 使能 0 = 禁止

Bits	描述	
[11]	CHEN11	模拟输入通道 11使能(转换输入电压从 PD.3输入) 1 = 使能 0 = 禁止
[10]	CHEN10	模拟输入通道 10使能(转换输入电压从 PD.2输入) 1 = 使能 0 = 禁止
[9]	CHEN9	模拟输入通道 9使能(转换输入电压从 PD.1输入) 1 = 使能 0 = 禁止
[8]	CHEN8	模拟输入通道 8使能(转换输入电压从 PD.0输入) 1 = 使能 0 = 禁止
[7]	CHEN7	模拟输入通道 7使能(转换输入电压从 PA.7输入) 1 = 使能 0 = 禁止
[6]	CHEN6	模拟输入通道 6使能(转换输入电压从 PA.6输入) 1 = 使能 0 = 禁止
[5]	CHEN5	模拟输入通道 5使能(转换输入电压从 PA.5输入) 1 = 使能 0 = 禁止
[4]	CHEN4	模拟输入通道 4使能(转换输入电压从 PA.4输入) 1 = 使能 0 = 禁止
[3]	CHEN3	模拟输入通道 3使能(转换输入电压从 PA.3输入) 1 = 使能 0 = 禁止
[2]	CHEN2	模拟输入通道 2使能(转换输入电压从 PA.2输入) 1 = 使能 0 = 禁止
[1]	CHEN1	模拟输入通道 1使能(转换输入电压从 PA.1输入) 1 = 使能 0 = 禁止

Bits	描述	
[0]	CHEN0	<p>模拟输入通道 0使能(转换输入电压从 PA.0输入)</p> <p>1 = 使能</p> <p>0 = 禁止</p> <p>在单次模式，如果软件使能了多个通道，编号最小的通道被使能，其他通道会被忽略。</p>

A/D比较寄存器0/1 (ADCMPR0/1)

寄存器	偏移量	R/W	描述	复位后的值
ADCMPR0	ADC_BA+0x38	R/W	A/D比较寄存器 0	0x0000_0000
ADCMPR1	ADC_BA+0x3C	R/W	A/D比较寄存器 1	0x0000_0000

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

Bits	描述	
[31:28]	保留	-
[27:16]	CMPD[11:0]	比较数值 此 12-位数据用于和指定通道的转换结果进行比较。 可以用它在扫描模式下监测外部输入电压的波动，不会增加软件的负担
[15:12]	保留	-
[11:8]	CMPMATCNT[3:0]	比较匹配计数 当指定 A/D 通道的模拟转换值和比较条件 CMPCOND[2] 相匹配，内部计数器将加 1。 当内部计数器的值达到设定值 (CMPMATCNT +1)时，将置位 CMPF _x 位。

Bits	描述	
[7:3]	CMPCH[3:0]	比较通道选择
		00000 选择通道 0 转换结果进行比较
		00001 选择通道 1 转换结果进行比较
		00010 选择通道 2 转换结果进行比较
		00011 选择通道 3 转换结果进行比较
		00100 选择通道 4 转换结果进行比较
		00101 选择通道 5 转换结果进行比较
		00110 选择通道 6 转换结果进行比较
		00111 选择通道 7 转换结果进行比较
		01000 选择通道 8 转换结果进行比较
		01001 选择通道 9 转换结果进行比较
		01010 选择通道 10 转换结果进行比较
		01011 选择通道 11 转换结果进行比较
		01100 选择通道 12 转换结果进行比较
		01101 选择通道 13 转换结果进行比较
		01110 选择通道 14 转换结果进行比较
		01111 选择通道 15 转换结果进行比较
		10000 选择通道 16 转换结果进行比较
		10001 选择通道 17 转换结果进行比较
[2]	CMPCOND	比较条件
		<p>1 = 设备比较条件，即当 12- 位 A/D 转换结果大于或等于 12- 位 CMPD (ADCMPrx[27:16])，内部匹配计数器将加 1。</p> <p>0 = 设备比较条件，即当 12- 位 A/D 转换结果小于 12- 位 CMPD (ADCMPrx[27:16])，内部匹配计数器将加 1。</p> <p>注：当内部计数器达到 (CMPMATCNT +1)，CMPF_x 位将被置位。</p>
[1]	CMPIE	<p>比较中断使能</p> <p>1 = 使能比较功能中断</p> <p>0 = 禁用比较功能中断</p> <p>如果比较功能使能，而且比较条件和 CMPCOND 及 CMPMATCNT 的设定匹配，则 CMPF 位将被置位，同时，如果 CMPIE 为 1，将产生比较中断请求。</p>

Bits	描述	
[0]	CMPEN	<p>比较使能</p> <p>1 = 使能比较功能</p> <p>0 = 禁用比较功能</p> <p>当转换的数据载入到 ADC_RESULTx寄存器时，设置该位为1 使能 ADC 控制器去比较 CMPD[11:0] 和指定通道的转换结果。</p> <p>当该位被设置为1，并且 CMPMATCH 是 0，一旦匹配发生CMPF会被置1</p>

A/D状态寄存器(ADSR)

寄存器	偏移量	R/W	描述	复位后的值
ADSR	ADC_BA+0x40	R/W	ADC状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
							INITRDY
15	14	13	12	11	10	9	8
							CHANNEL
7	6	5	4	3	2	1	0
CHANNEL				BUSY	CMPF1	CMPF0	ADF

Bits	描述	
[31:17]	-	保留
[16]	INITRDY	<p>ADC 上电序列完成</p> <p>1 = 从上次系统复位后，ADC 被上电</p> <p>0 = 在系统复位之后，ADC 没有被上电</p> <p>注意: 在系统复位之后，该位被置位; 通过上电事件自动清零</p>
[15:9]	保留	
[8:4]	CHANNEL[4:0]	<p>当前转换通道</p> <p>当 BUSY = 1时，该域反映当前转换通道。当 BUSY=0 时，它表示下一个要转换的通道号。</p> <p>该位只读。</p>
[3]	BUSY	<p>BUSY/IDLE</p> <p>1 = A/D 转换器正在忙着转换</p> <p>0 = A/D 转换器空闲</p> <p>该位是 ADCR 中 ADST 位的镜像，也就是说如果ADST = 1，那么BUSY 也是 1，反之亦然。</p> <p>该位只读。</p>

Bits	描述	
[2]	CMPF1	<p>比较标志</p> <p>当所选择的通道的 A/D 转换结果和 ADCMPR1 的设定条件匹配时，该位置1。该位写1清除。</p> <p>1 = ADC_RESULTx中的转换结果和 ADCMPR1 的设定值匹配</p> <p>0 = ADC_RESULTx中的转换结果和 ADCMPR1 的设定值不匹配</p> <p>向该位写1清0</p> <p>注: 当该位被置位，匹配计数器复位为0，并且当用户向CMPF1写1清0后会继续计数。</p>
[1]	CMPF0	<p>比较标志</p> <p>当所选择的通道的 A/D 转换结果和 ADCMPR0 的设定条件匹配时，该位置1。</p> <p>1 = ADC_RESULTx中的转换结果和 ADCMPR0 的设定值匹配</p> <p>0 = ADC_RESULTx中的转换结果和 ADCMPR0 的设定值不匹配</p> <p>向该位写1清0</p> <p>注: 当该位被置位，匹配计数器复位为0，并且当用户向CMPF0写1清0后会继续计数。</p>
[0]	ADF	<p>A/D 转换结束标志</p> <p>该状态标志指示 A/D 转换的结束。</p> <p>ADF在如下两种条件下被置 1。</p> <ol style="list-style-type: none"> 1. 当在单一模式下的 A/D 转换结束 2. 当在扫描模式下，所有指定的通道的 A/D 转换结束 <p>向该位写1清0</p>

A/D PDMA当前传输数据寄存器(ADPDMA)

寄存器	偏移量	R/W	描述	复位后的值
ADPDMA	ADC_BA+0x60	R	A/D PDMA当前传输数据寄存器	0x0000_0000

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

Bits	描述	
[31:12]	保留	-
[11:0]	AD_PDMA[11:0]	<p>ADC PDMA当前传输数据寄存器</p> <p>当 PDMA 传输时，读该寄存器可以监测当前 PDMA 的传输数据。</p> <p>此为只读寄存器。</p>

A/D 电源管理寄存器(ADCPWD)

寄存器	偏移量	R/W	描述	复位后的值
ADCPWD	ADC_BA+0x64	R	A/D电源控制	0x0001_E002

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

Bits	描述									
[31:4]	保留	-								
[3:2]	PWDMOD[1:0]	<div>掉电模式</div> <table><tr><th>PWDMOD [1:0]</th><th>描述</th></tr><tr><td>00</td><td>Power down</td></tr><tr><td>10</td><td>Stand by 模式</td></tr></table> <div>注意:不同的 PWDMOD 有不同的掉电上电顺序, 为了避免 ADC 上电顺序错误, 用户必须在每次掉电或上电时保持在 PWMOD 一致。</div>	PWDMOD [1:0]	描述	00	Power down	10	Stand by 模式		
PWDMOD [1:0]	描述									
00	Power down									
10	Stand by 模式									
[1]	PWDCALEN	<div>上电校准功能使能</div> <div>1:上电校准</div> <div>0:上电不校准</div> <div>注意: 该位与CALFBKSEL配合起作用 , 见下表</div> <table><tr><th>{PWDCALEN,CALFBSEL}</th><th>描述</th></tr><tr><td>11</td><td>上电校准</td></tr><tr><td>10</td><td>上电加载校准值</td></tr><tr><td>0X</td><td>不需要校准</td></tr></table>	{PWDCALEN,CALFBSEL}	描述	11	上电校准	10	上电加载校准值	0X	不需要校准
{PWDCALEN,CALFBSEL}	描述									
11	上电校准									
10	上电加载校准值									
0X	不需要校准									
[0]	PWUPRDY	<div>ADC 上电顺序完成, 准备转换</div> <div>1:ADC准备转换</div> <div>0:ADC没有准备转换, 可能在掉电状态或上电处理状态</div>								

A/D校准控制寄存器(ADCCALCTL)

寄存器	偏移量	R/W	描述	复位后的值
ADCCALCTL	ADC_BA+0x68	R/W	A/D 校准控制寄存器	0x0000_0009

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

Bits	描述	
[31:4]	-	保留
[3]	CALSEL	选择校准功能模块 1:选择校准功能模块 0:加载校准功能模块
[2]	CALDONE	校准功能模块完成 1: 选择的校准功能模块完成 0:还没完成
[1]	CALSTART	开始校准功能模块 1:开始校准功能模块 0:停止校准功能模块 注意: 该位通过SW置位, 通过 HW清零, 当CALEN = 0,不要写1到该位.
[0]	CALEN	使能校准功能 使能该位打开校准功能模块 1:使能 0:(忽略校准)

A/D 校准字 (ADCCALWORD)

寄存器	偏移量	R/W	描述	复位后的值
ADCCALWORD	ADC_BA+0x6C	R/W	A/D 校准加载字寄存器	0xFFFF_FFFF

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

Bits	描述	
[31:8]	-	保留
[6:0]	CALWORD	<p>校准字寄存器</p> <p>在加载校准之前写入上次的校准值</p> <p>校准结束之后读这个寄存器</p> <p>注意: 校准模块包含两个部分: “校准” 和 “加载校准”。如果校准模块被配置为 “校准”, 该寄存器代表校准结束后的校准结果; 如果配置为 “加载校准”, 设置该寄存器在加载校准之前, 在加载校准完成之后, 被加载的校准字会应用的ADC中, 直到校准完成, 加载值才会等于初始的CALWORD。</p>

A/D 通道 采样寄存器0 (ADCCHSAMP0)

寄存器	偏移量	R/W	描述	复位后的值
ADCCHSAMP0	ADC_BA+0x70	R/W	A/D 通道采样寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
CH7SAMPCNT				CH6SAMPCNT			
23	22	21	20	19	18	17	16
CH5SAMPCNT				CH4SAMPCNT			
15	14	13	12	11	10	9	8
CH3SAMPCNT				CH2SAMPCNT			
7	6	5	4	3	2	1	0
CH1SAMPCNT				CH0SAMPCNT			

Bits	描述	
[31:28]	CH7SAMPCNT	通道 7 采样计数器 和通道0采样计数器表格一样
[27:24]	CH6SAMPCNT	通道 6 采样计数器 和通道0采样计数器表格一样
[23:20]	CH5SAMPCNT	通道 5 采样计数器 和通道0采样计数器表格一样
[19:16]	CH4SAMPCNT	通道 4 采样计数器 和通道0采样计数器表格一样
[15:12]	CH3SAMPCNT	通道 3 采样计数器 和通道0采样计数器表格一样
[11:8]	CH2SAMPCNT	通道 2 采样计数器 和通道0采样计数器表格一样
[7:4]	CH1SAMPCNT	通道1采样计数器 和通道0采样计数器表格一样

Bits	描述	
[3:0]	CH0SAMP CNT	通道0采样计数器
		CH0SAMP CNT ADC clock
		0 0
		1 1
		2 2
		3 4
		4 8
		5 16
		6 32
		7 64
		8 128
		9 256
		10 512
		11 1024
		12 1024
		13 1024
		14 1024
		15 1024

A/D通道采样寄存器1 (ADCCHSAMP1)

寄存器	偏移量	R/W	描述	复位后的值
ADCCHSAMP1	ADC_BA+0x74	R/W	A/D 通道采样寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-				INTCHSAMP CNT			
15	14	13	12	11	10	9	8
CH11SAMP CNT				CH10SAMP CNT			
7	6	5	4	3	2	1	0
CH9SAMP CNT				CH8SAMP CNT			

Bits	描述	
[31:20]	-	保留
[19:16]	INTCHSAMP CNT	内部通道 (VTEMP, AVDD, AVSS, Int_VREF, DAC0, DAC1) 采样计数器 和通道0采样计数器表格一样
[15:12]	CH11SAMP CNT	通道11采样计数器 和通道0采样计数器表格一样
[11:8]	CH10SAMP CNT	通道10采样计数器 和通道0采样计数器表格一样
[7:4]	CH9SAMP CNT	通道9采样计数器 和通道0采样计数器表格一样
[3:0]	CH8SAMP CNT	通道8采样计数器 和通道0采样计数器表格一样

5.6 数模转换器

5.6.1 概述

DAC 是一个12位电压输出型数字模拟转换器。该芯片有两个DAC。

5.6.2 特征

12位电压输出型DAC。可以使用PDMA 控制器。当用到两个DAC，为了同步更新操作，他们组合使用。

特性:

- Int_VREF、VREF 或AVDD 参考电压选择
- 两个DAC可同步更新
- DAC 最大转换速率 500KSPS.

5.6.3 模块图

两个DAC的模块图如下图所示.



5.6.4.1 DAC Core

Revision 1.07

VREF。

5.6.4.2 DAC参考电压

DAC 的参考电压可以设定为外部参考电压或内部参考电压或是带三个可选值(Int_VREF /AVDD/VREF)的 AVDD。

5.6.4.3 DAC操作

用户任何时候更新DACx_DAT的值，DAC都会更新输出电压值。有几种更新DACx_DAT的方法，用户可以通过DACx_CTL 寄存器DACLSEL来配置。

当DACLSEL = 000，用户写数据到DACx_DAT 寄存器时，DAC 会更新输出电压；当DACLSEL = 001，PDMA 控制器发送应答到DAC 控制器时，DAC 会更新输出电压。这个配置明显是为 DAC的PDMA 应用的。在用这个功能的时候，用户应该一起配置PDMA 的其它设定；当DACLSEL = 010, 011, 100 或101时，DAC可以通过定时触发更新DAC输出，用户应该在定时事件到来之前写DACx_DAT。受限于DAC的转换时间，最高的转换速率是500KHz (1/2us)。如果用户更新DACx_DAT寄存器的速度高于500KHz，一些数据会丢失，DAC会输出 DACx_DAT最后更新的值。由于DAC控制器的工作频率 (PCLK)在不同的情况下会有所不同，用户可以调整DAC01_COMCTL 寄存器中WAITDACCONV的值来满足DAC 转换速率的限制。DAC的上电稳定时间也是可以调整的，通过设置 DACx_CTL[DACPWONSTBCNT] 来满足DAC 6 us 的稳定时间。

注：DAC有两个时序要求

- 稳定时间：从 DAC上电到稳定准备转换的时间，用户可以通过设置DACx_CTL[DACPWONSTBCNT]等待6 us来满足稳定时间的要求。
- 转换时间：DAC转换数字量到模拟量的时间。用户可以设置DAC01_COMCTL[WAITDACCONV]来满足2us转换时间的要求

5.6.4.4 组合 DAC0 和 DAC1

两个 DACx 可以组合在一起，用GRP 位来同步更新各个DAC输出。硬件确保这两个DAC同时更新不依赖于任何中断。

DAC0 和DAC1 通过设置DAC01_COMCTL中的GRP位来设置组合。当 DAC0 和 DAC1 被组合：

可以通过DAC0的DACLSEL，配置DAC0与DAC1的触发条件

当DAC0的DACLSEL = 000，在用户写DAC0_DAT 和DAC1_DAT 寄存器之后，无论两个寄存器写的顺序，两个DAC会同步输出各自更新的DAC_DAT。

当DAC0的DACLSEL = 001，在PDMA发送应答信号到DAC控制器后，两个DAC会同步输出各自更新的DAC_DAT。

当DAC0的DACLSEL = 010，在Timer0时间结束事件到来后，两个DAC会同步输出各自更新的DAC_DAT。

当DAC0的DACLSEL = 011，在Timer1时间结束事件到来后，两个DAC会同步输出各自更新的DAC_DAT。

当DAC0的DACLSEL = 100，在Timer2时间结束事件到来后，两个DAC会同步输出各自更新的DAC_DAT。

当DAC0的DACLSEL = 101，在Timer3时间结束事件到来后，两个DAC会同步输出各自更新的DAC_DAT。

注：

当DAC0 和DAC1 被组合在一起并且不带PDMA 操作(DACLSEL ≠ 001),, 两个DACx_DAT 寄存器必须在输出更新之前写, 即使一个或两个DAC的数据没有改变(PDMA 操作会自动地为两个DAC准备DAC_DAT)

下图显示的是组合和未组合的DAC0 和DAC1 latch-update时序范例.

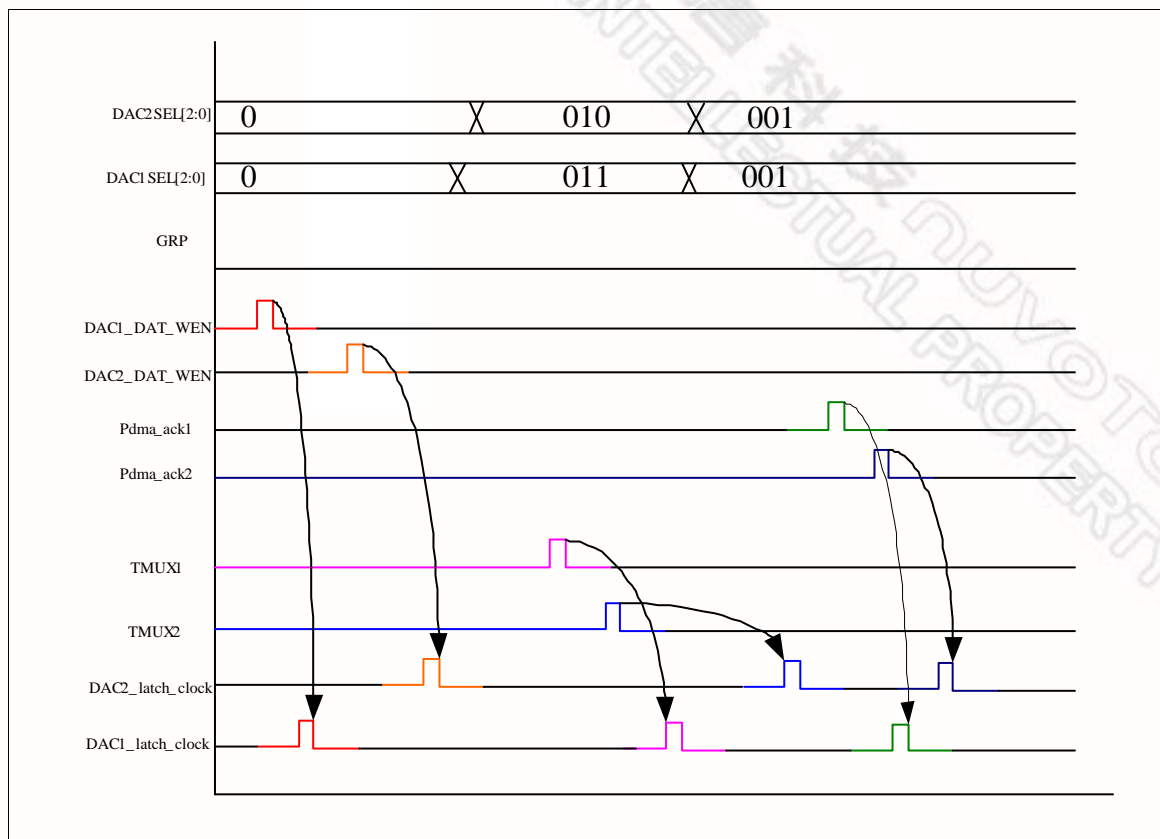


图 5-19 DAC0 和 DAC1 未分组更新范例

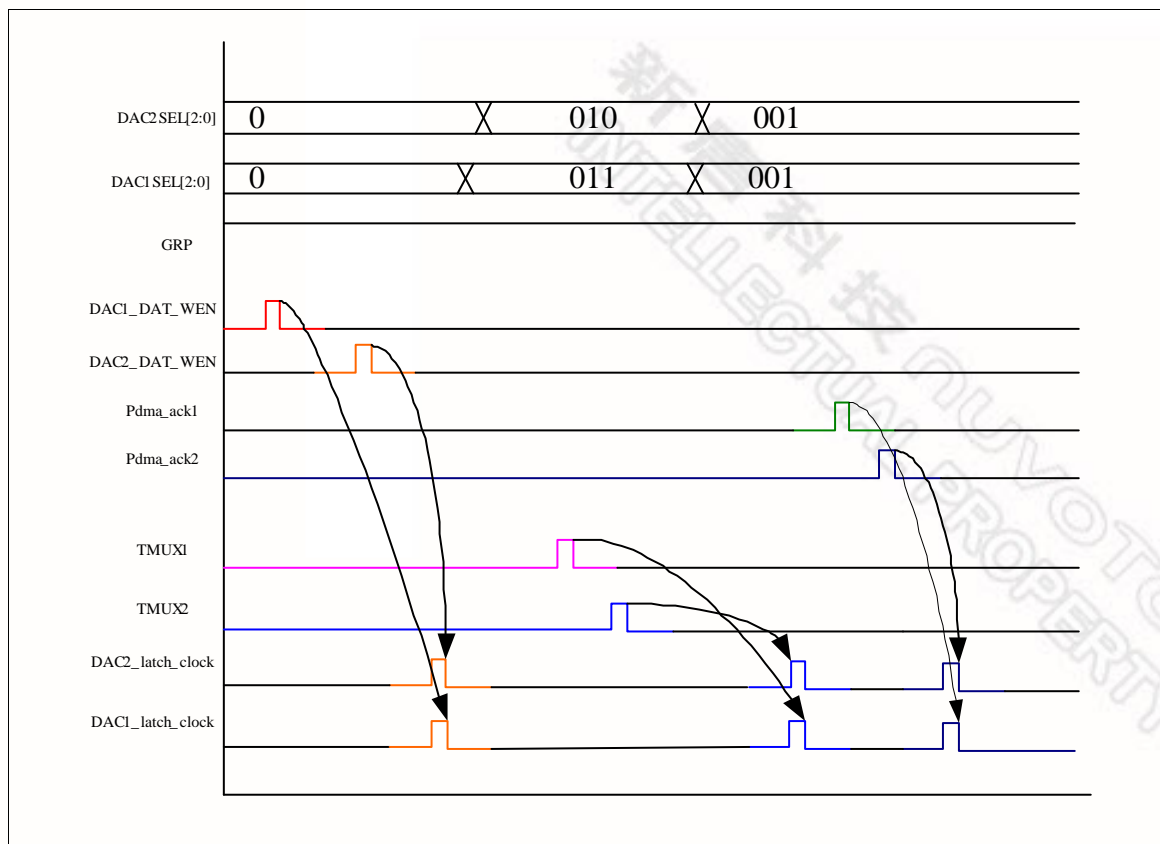


图 5-20 DAC0 和 DAC1 分组更新范例

注: DAC0,1 建立时间

DMA 控制器传输数据到 DAC0,1的速度比 DAC0,1输出处理的速度快。当用DMA 控制器的时候, 用户必须确保DMA数据传输速度不要快于DAC的 处理时间。在设备的规格书中查看参数。

DAC0,1 中断

当DACx_DATA内部锁定，DACIFG被置位。DACIFG位用来指示DACx已经准备好接收新的数据。如果DACIE被置位，DACIFG产生一个中断请求，用户必须向DACIFG写1清0。

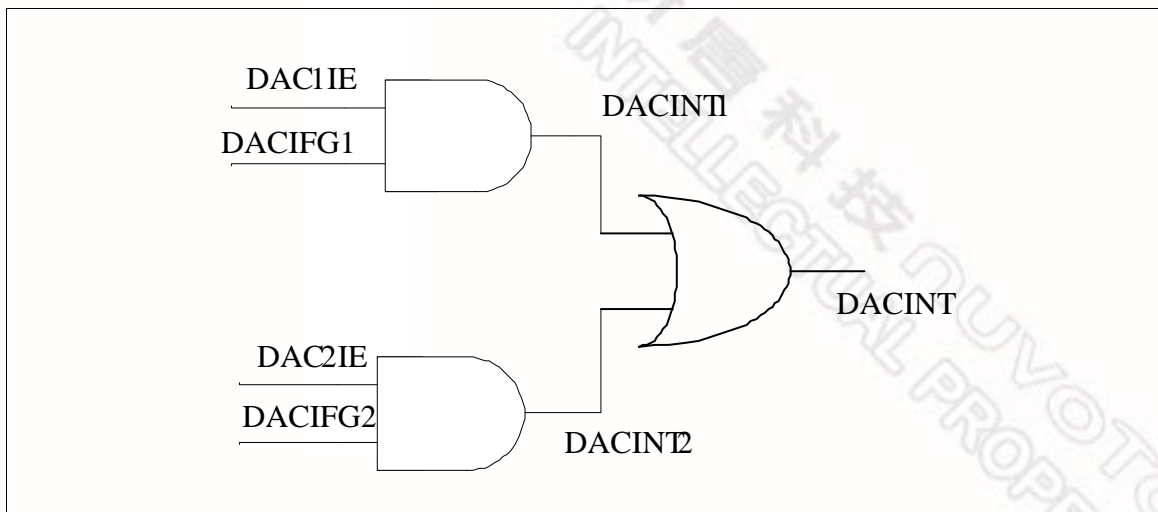


图 5-21 DAC 中断

5.6.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
DAC_BA = 0x400A_0000				
DAC0_CTL	DAC_BA+0x00	R/W	DAC0 控制寄存器	0x0A00_0000
DAC0_DATA	DAC_BA+0x04	R/W	DAC0 数据寄存器	0x0000_0000
DAC0_STS	DAC_BA+0x08	R/W	DAC0 状态寄存器	0x0000_0000
DAC1_CTL	DAC_BA+0x10	R/W	DAC1 控制寄存器	0x0A00_0000
DAC1_DATA	DAC_BA+0x14	R/W	DAC1 数据寄存器	0x0000_0000
DAC1_STS	DAC_BA+0x18	R/W	DAC1 状态寄存器	0x0000_0000
DAC01_COMCTL	DAC_BA+0x20	R/W	DAC01 共用控制寄存器	0x0000_0000

5.6.6 寄存器描述

DAC控制寄存器(DAC control ,x=0,1)

寄存器	偏移量	R/W	描述	复位后的值
DACx_CTL	DAC_BA + 0x00 0x10	R/W	DAC控制寄存器	0x0A00_0000

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

Bits	描述												
[31:7]	- 保留												
[21:8]	DACPWONSTBCNT DAC 自掉电状态上电后需要 6 us可稳定。 此域控制内部计数器(以PCLK单位)确保DAC需要的稳定时间。												
[6:4]	DACLSEL[2:0] DAC加载选择, 为 DAC 锁存器选择加载触发器 <table> <tr> <td>000</td><td>当写DACx_DAT 时, DAC锁存加载</td></tr> <tr> <td>001</td><td>PDMA ACK</td></tr> <tr> <td>010</td><td>TMR0上升沿</td></tr> <tr> <td>011</td><td>TMR1上升沿</td></tr> <tr> <td>100</td><td>TMR2上升沿</td></tr> <tr> <td>101</td><td>TMR3上升沿</td></tr> </table>	000	当写DACx_DAT 时, DAC锁存加载	001	PDMA ACK	010	TMR0上升沿	011	TMR1上升沿	100	TMR2上升沿	101	TMR3上升沿
000	当写DACx_DAT 时, DAC锁存加载												
001	PDMA ACK												
010	TMR0上升沿												
011	TMR1上升沿												
100	TMR2上升沿												
101	TMR3上升沿												
[3:2]	- 保留												
[1]	DACIE DAC 中断使能 1 = 使能 0 = 禁止												

Bits	描述	
[0]	DACEN	<p>DAC 使能</p> <p>1 = DAC上电</p> <p>0 = DAC掉电</p> <p>注: 当DAC上电, DAC在等待DACPWONSTBCNT+1 PCLK 个周期之后会自动开始转换</p>

DACx 数据 寄存器 (DACx DATA, x=0,1)

寄存器	偏移量	R/W	描述	复位后的值
DACx_DATA	DAC_BA+ 0x04 0x14	R/W	DAC数据寄存器	0x0000_0000

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

Bits	描述	
[31:12]	-	保留
[11:0]	DAC Data[11:0]	DAC数据

DAC状态寄存器(DACx STS ,x=0,1)

寄存器	偏移量	R/W	描述	复位后的值
DACx_STS	DAC_BA + 0x08 0x18	R/W	DAC状态寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[2]	BUSY	BUSY 位 1 = DAC忙 0 = DAC空闲
[1]	DACSTFG	DAC 开始标志 1 = DAC已经开始 0 = DAC还没开始 注: 该位只读
[0]	DACIFG	DAC中断标志 1 = 中断挂起 0 = 没有中断挂起 注: 该位只读

DAC共用控制寄存器(DAC01_COMCTL)

寄存器	偏移量	R/W	描述	复位后的值
DAC01_COMCTL	DAC_BA+0x20	R/W	DAC 共用控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-					REFSEL		DAC12GRP
7	6	5	4	3	2	1	0
WAITDACCONV[7:0]							

Bits	描述	
[31:11]	-	保留
[10:9]	REFSEL[1:0]	参考电压选择
		00 AVDD
		01 内部参考电压
		10 外部参考电压
注: 参考图 5-5.		
[8]	DAC01GRP	组合DAC0 和DAC1. 1 = 组合 0 =没有组合
[7:0]	WAITDACCONV	等待DAC 转换完成 当每次数据送到DAC，DAC需要至少2 us来完成转换。这意味着用户每次更新DACx_data 寄存器的速度不能高于2 us，否则数据会丢失。设置这个寄存器可以调整每次写DACx_data 到DAC（以PCLK 为单位）的时间间隔，以便满足 2 us 间隔的需要。

5.7 DMA控制器

5.7.1 概述

DMA控制器包含六个通道的外设存储器直接存取(PDMA) 控制器和一个音频存储器直接存取(VDMA) 控制器和一个循环冗余检查(CRC)发生器。PDMA控制器可以从内存或从APB设备传输数据。8个通道的DMA 包括一个通道VDMA(存储器到存储器)和六个通道的PDMA (外设到存储器或存储器到外设或存储器到存储器)和一个CRC控制器。VDMA通道0支持存储器到存储器的块传输。PDMA (DMA CH1~CH6)在外围 APB 设备和存储器之间有一个字大小的缓存作为传输缓存。通道0 VDMA有两个字大小的缓存。

通过设定禁用 PDMA [PDMACEN] /VDMA [VDMACEN]位, 软件可以停止 DMA 的操作。通过软件轮询或者收到内部的DMA 中断, CPU 可以识别DMA 运作的完成。DMA 控制器可增加源和目的地址或者也能将其固定或回绕。

DMA控制器也包含一个循环冗余检查(CRC)发生器。它可以执行带可编程多项式设定的CRC运算。CRC支持 CPU PIO模式和DMA传输模式。

5.7.2 特征

七个 DMA 通道和一个CRC 发生器: 1 VDMA 通道 和 6 PDMA 通道。每个通道能支持一个单项传输。

AMBA AHB 主/从 接口兼容, 用于数据传输和寄存器读/写。

硬件轮转优先级机制。

- VDMA
 - ◆ 存储器到存储器传输
 - ◆ 支持跨越的块传输
 - ◆ 支持一个字/半个字/字节二进制地址
 - ◆ 支持地址的方向: 递增和递减
- PDMA
 - ◆ 外设到存储器、存储器到外设、存储器到存储器传输
 - ◆ 支持一个字的二进制地址
 - ◆ 在存储器到存储器模式, 支持字对齐传输长度
 - ◆ 在外设到存储器器和存储器到外设模式, 支持字/半字/字节对齐传输长度
 - ◆ 支持字/半字/字节传输数据宽度 从/到外设
 - ◆ 支持地址方向: 递增模式、固定和回绕
- 循环冗余检查(CRC)
 - ◆ 支持四个通用的多项式 CRC-CCITT, CRC-8, CRC-16, 和 CRC-32
 - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - CRC-8: $X^8 + X^2 + X + 1$
 - CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- ◆ 可编程的种子值.
- ◆ 支持对输入数据和 CRC 校验和的可编程的反序设定。
- ◆ 支持对输入数据和 CRC 校验和的可编程的一次补码设定。
- ◆ 支持 CPU PIO 模式或DMA 传输模式
- ◆ 在CPU PIO 模式下，支持 8/16/32-bit 数据宽度
 - 8-bit 写模式: 1-AHB 时钟周期操作.
 - 16-bit写模式: 2-AHB时钟周期操作.
 - 32-bit写模式: 4-AHB时钟周期操作.
- ◆ 在CRC DMA模式下，支持字节对齐传输长度 .

5.7.3 框图

DMA时钟控制框图如下。

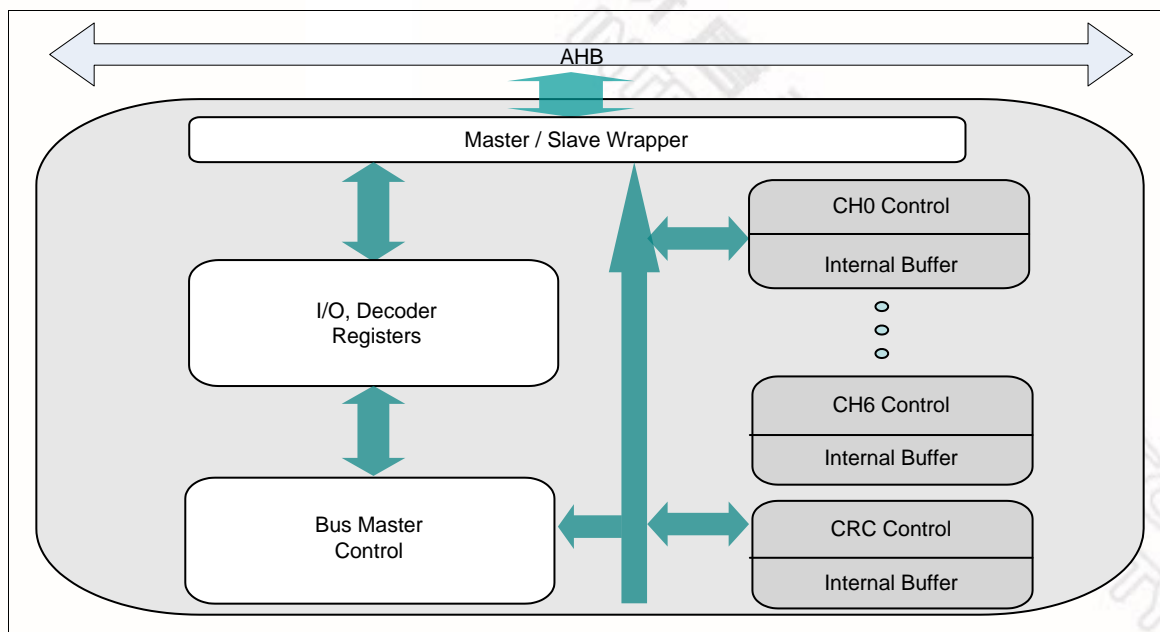


图 5-22 DMA 控制器框图

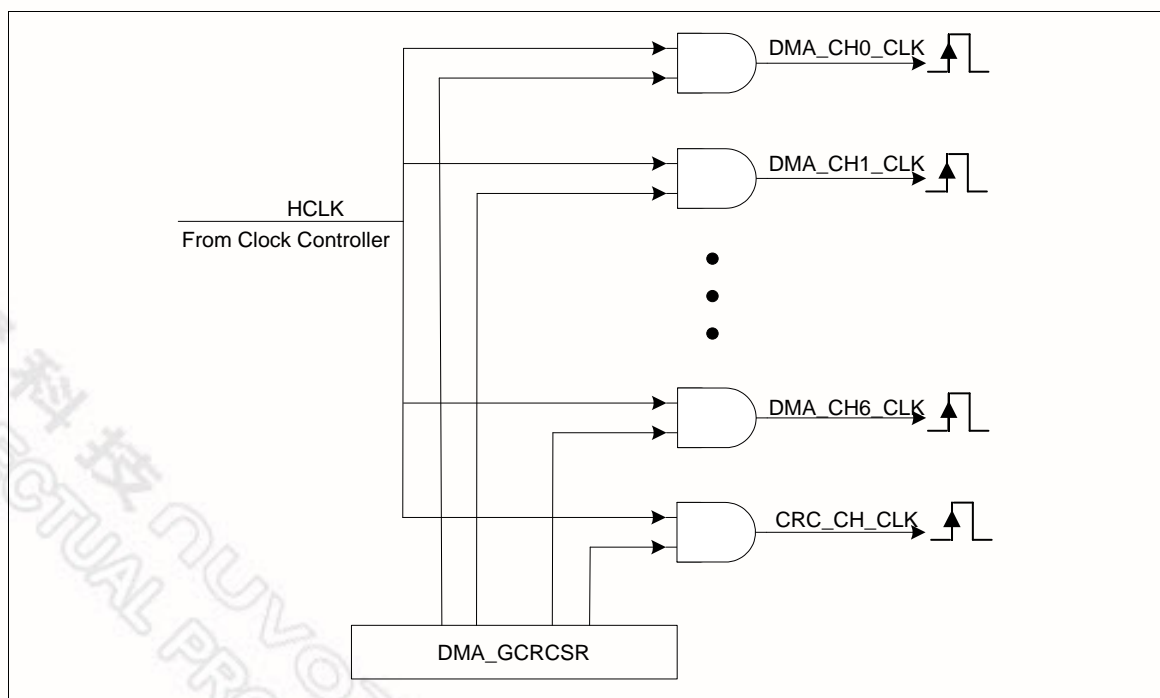


图 5-23 DMA 时钟控制图

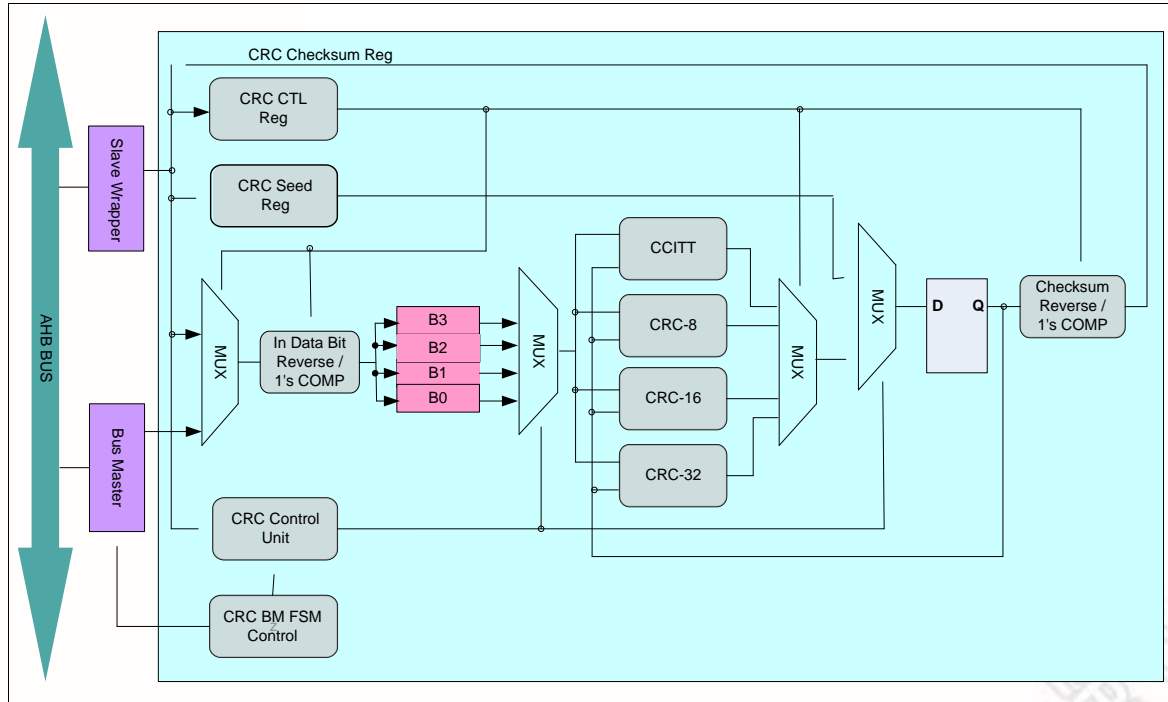


图 5-24 CRC 发生器模块图

5.7.4 功能描述

直接存储器存取(DMA) 控制器模块传输数据从一个地址到另外一个地址，而不需要 CPU 的参与。DMA 控制器有8个通道，包括一个通道VDMA (存储器到存储器) 和六个通道PDMA (外设到存储器或存储器到外设或存储器到存储器) 和一个CRC通道。

通过软件轮询或者收到内部的DMA 中断，CPU 可以识别DMA 运作的完成。对于源和目的地址而言，DMA 控制器有三种不同的模式：递增模式、固定模式和回绕模式。

5.7.4.1 VDMA

DMA控制器包含一个通道VDMA (存储器到存储器)。VDMA 模块可以从一个地址到另外一个地址传输数据 并支持跨越的块传输。当工作在VDMA 模式，传输地址可以连续递增和连续递减。

若不是块传输模式 (禁止 VDMA_CSR [STRIDE_EN])，软件必须使能 DMA 通道VDMA [VDMACEN]，然后写一个有效的源地址到VDMA_SAR 寄存器，一个目标地址到VDMA_DAR 寄存器，一个传输计数到 VDMA_BCR 寄存器。接着触发 VDMA_CSR [TRIG_EN]。DMA 会连续传输直到 VDMA_CBCR 到0。

块传输模式 (使能VDMA_CSR [STRIDE_EN])，软件必须使能 DMA 通道VDMA [VDMACEN]，然后写一个有效的源地址到VDMA_SAR 寄存器，一个源地址偏移量计数到VDMA_SASOCR [SASTOBL] 寄存器，一个目标地址到VDMA_DAR 寄存器，一个目标地址偏移量计数到VDMA_DASOCR [DASTOBL]，一个传输计数到VDMA_BCR 寄存器和一个块字节计数到VDMA_SASOCR [STBC]。接着触发 VDMA_CSR [TRIG_EN]。DMA 会连续传输直到 VDMA_CBCR到0。下图显示源存储器和目标存储器块传输之间的关系。

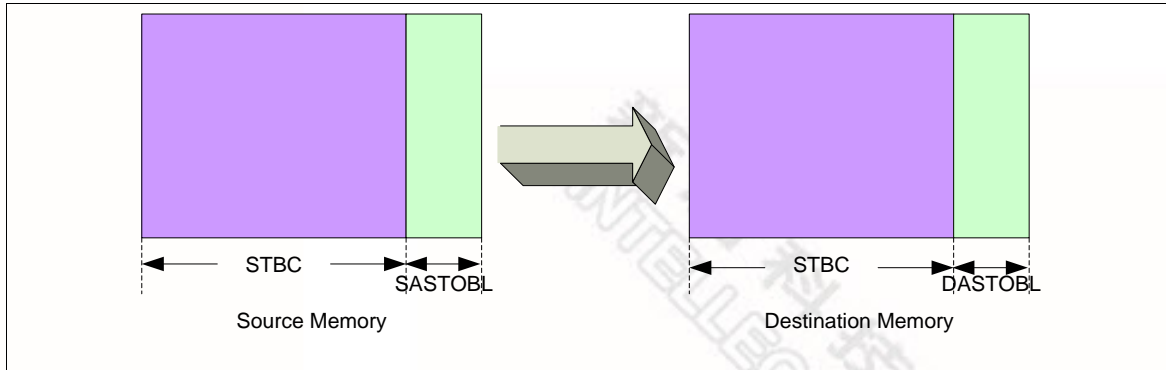


图 5-25 VDMA 块传输

如果在VDMA 操作期间发生一个错误，该通道会停止，直到软件清零错误的条件并设置 VDMA_CSR [SW_RST]复位 VDMA 通道和设置VDMA_CSR [VDMACEN] 和[TRIG_EN]位域重新开始。

5.7.4.2 PDMA

DMA控制器包含六个通道PDMA (外设到存储器或存储器到外设或存储器到存储器)，对于源和目的地址而言，DMA 控制器有三种不同的模式：递增模式、固定模式和回绕模式。

每个 PDMA通道没有预先设定默认值，因此用户必须在开始相关的 PDMA 通道之前通过设定 DMA_DSSR0、DMA_DSSR1进行配置。

软件必须使能 DMA 通道 PDMA [PDMACEN] 并且写有效的源地址到 PDMA_SARx 寄存器，写目的地址到 PDMA_DARx 寄存器，传输数量到 PDMA_BCRx 寄存器。然后触发 DMA_CSRx [TRIG_EN]。PDMA 将继续传输直到 PDMA_CBCRx 降为 0 (在回绕模式，当PDMA_CBCRx 等于 0， PDMA 会重新加载PDMA_CBCRx 并继续工作直到软件禁止 PDMA_CSRx[PDMACEN])。如果在 PDMA 操作期间有错误发生，通道停止直到软件清除错误条件并设置 PDMA_CSRx [SW_RST] 复位 PDMA 通道，并设置PDMA_CSRx [PDMACEN] 和 [TRIG_EN] 位重新开始。

在 PDMA（外设-到-存储器、存储器-到-外设）模式下，DMA 可以在外设 APB IP (ex: UART, SPI, ADC....) 和 存储器间传输数据。

5.7.4.3 CRC

DMA控制器包含一个循环冗余检查(CRC)发生器。它可以执行带可编程多项式设定的CRC运算。CRC 支持 CPU PIO模式和DMA传输模式。多项式操作包括CRC-CCITT, CRC-8, CRC-16 和 CRC-32。通过设置CRC_CTL寄存器中CRC_MODE域，软件可以选择多项式操作模式。

CRC 支持 CPU PIO 模式 (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) 和 DMA 传输 模式 (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 1)。下面是一个编程顺序的例子。

操作在 CPU PIO 模式下的步骤:

设置CRC_CTL寄存器中CRCCEN位使能 CRC。

初始化设置： 设置数据格式 (WDATA_RVS, CHECKSUM_RVS, WDATA_COM 和 CHECKSUM_COM 通过设置 CRC_CTL 寄存器)，初始化种子值(CRC_SEED) 并通过设置 CRC_CTL [CPU_WDLEN] 寄存器选择数据长度。

通过设置CRC_CTL中CRC_RST位，设置 CRC 复位来加载初始种子值到CRC 电路

写数据到 CRC_WDATA执行CRC 运算。

通过读CRC_CHECKSUM寄存器获得 CRC 校验和结果

操作在CRC DMA模式下的步骤:

设置CRC_CTL寄存器中CRCCEN位使能 CRC。 .

初始化设置： 设置数据格式 (WDATA_RVS, CHECKSUM_RVS, WDATA_COM 和 CHECKSUM_COM 通过设置 CRC_CTL 寄存器)，初始化种子值(CRC_SEED)

通过设置CRC_DMASAR 和 CRC_DMABCR给出一个有效的源地址和传输长度

使能 CRC_CTL [TRIG_EN]，然后硬件会复位种子值并读存储器数据来执行CRC计算。

等待 CRC DMA 传输 和 CRC 计算结束，然后通过读CRC_CHECKSUM寄存器获得 CRC 校验和结果。

5.7.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
VDMA Base Address : VDMA_BA = 0x5000_8000				
VDMA_CSR	VDMA_BA + 0x00	R/W	VDMA 控制寄存器	0x0000_0000
VDMA_SAR	VDMA_BA + 0x04	R/W	VDMA 源地址寄存器	0x0000_0000
VDMA_DAR	VDMA_BA + 0x08	R/W	VDMA 目标地址寄存器	0x0000_0000
VDMA_BCR	VDMA_BA + 0x0C	R/W	VDMA 传输字节计数寄存器	0x0000_0000
VDMA_CSAR	VDMA_BA + 0x14	R	VDMA 当前源地址寄存器	0x0000_0000
VDMA_CDAR	VDMA_BA + 0x18	R	VDMA 当前目标地址寄存器	0x0000_0000
VDMA_CBCR	VDMA_BA + 0x1C	R	VDMA 当前传输字节计数寄存器	0x0000_0000
VDMA_IER	VDMA_BA + 0x20	R/W	VDMA 中断使能寄存器	0x0000_0001
VDMA_ISR	VDMA_BA + 0x24	R/W	VDMA 中断状态寄存器	0x0000_0000
VDMA_SASOCR	VDMA_BA + 0x2C	R/W	VDMA 源地址跨越偏移量寄存器	0x0000_0000
VDMA_DASOCR	VDMA_BA + 0x30	R/W	VDMA 目标地址跨越偏移量寄存器	0x0000_0000
PDMA CH1 Base Address : PDMA_BA_CH1 = 0x5000_8100				
PDMA CH2 Base Address : PDMA_BA_CH2 = 0x5000_8200				
PDMA CH3 Base Address : PDMA_BA_CH3 = 0x5000_8300				
PDMA CH4 Base Address : PDMA_BA_CH4 = 0x5000_8400				
PDMA CH5 Base Address : PDMA_BA_CH5 = 0x5000_8500				
PDMA CH6 Base Address : PDMA_BA_CH6 = 0x5000_8600				
PDMA_CSRx	PDMA_BA_CHx + 0x00	R/W	PDMA 控制寄存器	0x0000_0000
PDMA_SARx	PDMA_BA_CHx + 0x04	R/W	PDMA 源地址寄存器	0x0000_0000
PDMA_DARx	PDMA_BA_CHx + 0x08	R/W	PDMA 目标地址寄存器	0x0000_0000
PDMA_BCRx	PDMA_BA_CHx + 0x0C	R/W	PDMA 传输字节计数寄存器	0x0000_0000
PDMA_CSARx	PDMA_BA_CHx + 0x14	R	PDMA 当前源地址寄存器	0x0000_0000
PDMA_CDARx	PDMA_BA_CHx + 0x18	R	PDMA 当前目标地址寄存器	0x0000_0000
PDMA_CBCRx	PDMA_BA_CHx + 0x1C	R	PDMA 当前传输字节计数寄存器	0x0000_0000
PDMA_IERx	PDMA_BA_CHx + 0x20	R/W	PDMA 中断使能寄存器	0x0000_0001
PDMA_ISRx	PDMA_BA_CHx + 0x24	R/W	PDMA 中断状态寄存器	0x0000_0000
PDMA_TCRx	PDMA_BA_CHx + 0x28	R/W	PDMA Timer 计数器设置寄存器	0x0000_0000
CRC Base Address : CRC_BA = 0x5000_8E00				

CRC_CTL	CRC_BA	+ 0x00	R/W	CRC 控制寄存器	0x2000_0000
CRC_DMASAR	CRC_BA	+ 0x04	R/W	CRC DMA 源地址寄存器	0x0000_0000
CRC_DMABCR	CRC_BA	+ 0x0C	R/W	CRC 传输字节计数寄存器	0x0000_0000
CRC_DMACSAR	CRC_BA	+ 0x14	R/W	CRC当前源地址寄存器	0x0000_0000
CRC_DMACBCR	CRC_BA	+ 0x1C	R/W	CRC 当前传输字节计数寄存器	0x0000_0000
CRC_DMAIER	CRC_BA	+ 0x20	R/W	CRC中断使能寄存器	0x0000_0001
CRC_DMAISR	CRC_BA	+ 0x24	R/W	CRC中断状态寄存器	0x0000_0000
CRC_WDATA	CRC_BA	+ 0x80	R/W	CRC 写数据寄存器	0x0000_0000
CRC_SEED	CRC_BA	+ 0x84	R/W	CRC种子寄存器	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA	+ 0x88	R	CRC 校验和寄存器	0x0000_0000
DMA Global Base Address : DMA_BA_GCR = 0x5000_8F00					
DMA_GCRCSR	DMA_BA_GCR + 0x00		R/W	DMA全局控制寄存器	0x0000_0000
DMA_DSSR0	DMA_BA_GCR + 0x04		R/W	DMA服务选择控制寄存器 0	0x1F1F_1F00
DMA_DSSR1	DMA_BA_GCR + 0x08		R/W	DMA服务选择控制寄存器 1	0x001F_1F1F
DMA_GCRISR	DMA_BA_GCR+ 0x0C		R	DMA全局中断寄存器	0x0000_0000

注：PDMA_REGx 中的x代表 PDMA 的通道。

5.7.6 寄存器 描述

VDMA 控制和状态寄存器 (VDMA_CSR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_CSR	VDMA_BA+0x000	R/W	VMAC控制和状态寄存器(CH0)	0x0000_0000

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

Bits	描述	
[31:24]	-	保留.
[23]	TRIG_EN	TRIG_EN 1 = 使能VDMA 数据读或写传输. 0 =不起作用 注1: 当 PDMA 传输完成时, 该位将被自动清除。 注2: 如果总线繁忙, 所有的VDMA传输会停止, 软件必须复位所有VDMA 通道, 然后再试。
[22:12]	-	保留.
[11]	DIR_SEL	传输源/目标地址方向选择 1 =传输地址连续增减. 0 = 传输地址连续递加
[10]	STRIDE_EN	跨越模式使能 1 = 使能跨越传输模式. 0 = 禁止跨越传输模式
[9:2]	-	保留.
[1]	SW_RST	软件引擎复位 1 =写 1 到该位将复位内部状态机, 指针。控制寄存器的内部不会被清除。该位在几个时钟周期之后将自动清除。 0 =写 0 到该位无效
[0]	VDMACEN	VDMA 通道使能

Bits	描述	
		<p>设置该位为 1 使能 VDMA 的操作。如果该位被清除，VDMA 将忽略所有的 VDMA 请求并强制总线主机进入 IDLE 状态。</p> <p>注：SW_RST将清除该位。</p>

VDMA传输源地址寄存器(VDMA_SAR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_SAR	VDMA_BA+0x004	R/W	VDMA 传输源地址寄存器 CH0	0x0000_0000

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

Bits	描述	
[31:0]	VDMA_SAR [31:0]	VDMA 传输源地址寄存器 该域表示一个32位的VDMA源地址

VDMA传输目的地址寄存器(VDMA_DAR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_DAR	VDMA_BA+0x008	R/W	VDMA传输目的地址寄存器CH0	0x0000_0000

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

Bits	描述	
[31:0]	VDMA_DAR [31:0]	VDMA 传输目的地址寄存器 该域表示一个32位的VDMA目的地址

VDMA传输字节计数寄存器(VDMA_BCR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_BCR	VDMA_BA+0x00C	R/W	VDMA传输字节计数寄存器CH0	0x0000_0000

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

Bits	描述	
[31:16]	-	保留。
[15:0]	VDMA_BCR [15:0]	<p>VDMA 传输字节计数寄存器</p> <p>该域表示一个16位的VDMA传输字节计数</p> <p>注：当在跨越使能模式 (VDMA_CSR [10] = "0")，传输字节计数 (VDMA_BCR)必须是STBC (VDMA_SASOCR [31:16]).的整数倍数。.</p>

VDMA当前源地址寄存器(VDMA_CSAR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_CSAR	VDMA_BA+0x014	R	VDMA 当前源地址寄存器CH0	0x0000_0000

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

Bits	描述	
[31:0]	VDMA_CSAR [31:0]	VDMA 当前源地址寄存器 (只读) 该域表示VDMA 正在传输的源地址

VDMA当前目的地址寄存器(VDMA CDAR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_CDAR	VDMA_BA+0x018	R	VDMA 当前目的地址寄存器 CH0	0x0000_0000

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

Bits	描述	
[31:0]	VDMA_CDAR [31:0]	VDMA 当前目的地址寄存器(只读) 该域表示VDMA 正在传输的目的地址.

VDMA当前字节计数寄存器(VDMA_CBCR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_CBCR	VDMA_BA+0x01C	R	VDMA当前字节计数寄存器CH0	0x0000_0000

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

Bits	描述	
[31:16]	-	保留.
[15:0]	VDMA_CBCR [15:0]	VDMA当前字节计数寄存器(只读) 该域表示当前还剩下的 VDMA字节数.

VDMA中断使能控制寄存器(VDMA IER)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_IER	VDMA_BA+0x020	R/W	VDMA中断使能控制寄存器CH0	0x0000_0001

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

Bits	描述	
[31:2]	-	保留
[1]	TD_IE	VDMA传输完成中断使能 1 =使能 VDMA 传输完成中断. 0 = 禁用 VDMA 传输完成中断
[0]	TABORT_IE	VDMA读/写目标中止中断使能 1 =使能 VDMA 传输过程中的目标中止中断 0 =禁止 VDMA 传输过程中的目标中止中断

VDMA中断状态寄存器(VDMA_ISR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_ISR	VDMA_BA+0x024	R/W	VDMA中断状态寄存器CH0	0x0x0x_0000

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

Bits	描述	
[31:2]	-	保留
[1]	TD_IS	<p>传输完成中断状态标志</p> <p>该位指示 VDMA 已经完成了所有传输。</p> <p>1 = 完成</p> <p>0 = 还没完成</p> <p>注： 向该位写1清0</p>
[0]	TABORT_IS	<p>VDMA读/写 目标中止中断标志</p> <p>1 = 收到总线错误应答</p> <p>0 = 没有收到总线错误应答</p> <p>注1: 向该位写1清0</p> <p>注2: VDMA_ISR [TABORT_IF]指示总线主机是否受到 ERROR 响应。如果总线主机收到了 ERROR 响应，则意味着目标中止发生了。VDMA控制器将停止传输和响应该事件到软件，然后进入 IDLE 状态。当目标中止发生，软件必须复位 VDMA，再次传输那些数据。</p>

VDMA源地址跨越偏移量控制寄存器(VDMA SASOCR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_SASOCR	VDMA_BA+0x02C	R/W	VDMA源地址跨越偏移量控制寄存器 CH0	0x0000_0000

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

Bits	描述	
[31:16]	STBC[15:0]	VDMA 跨越传输字节计数 16 位寄存器定义每排的跨越传输字节数
[15:0]	SASTOBL [15:0]	VDMA 源地址跨越偏移量 字节长度 16 位寄存器 定义每排源地址跨越传输偏移量计数.

VDMA目的地址跨越偏移量控制寄存器(VDMA_DASOCR)

寄存器	偏移量	R/W	描述	复位后的值
VDMA_DASOCR	VDMA_BA+0x030	R/W	VDMA目的地址跨越偏移量控制寄存器CH0	0x0000_0000

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

Bits	描述	
[31:16]	-	保留.
[15:0]	DASTOBL [15:0]	VDMA目的地址跨越偏移量字节长度 16 位寄存器定义每排目的地址跨越传输偏移量计数

PDMA控制和状态寄存器(PDMA CSRx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_CSR1	PDMA_BA_ch1+0x00	R/W	PMAC控制和状态寄存器 CH1	0x0000_0000
PDMA_CSR2	PDMA_BA_ch2+0x00	R/W	PMAC控制和状态寄存器CH2	0x0000_0000
PDMA_CSR3	PDMA_BA_ch3+0x00	R/W	PMAC控制和状态寄存器CH3	0x0000_0000
PDMA_CSR4	PDMA_BA_ch4+0x00	R/W	PMAC控制和状态寄存器CH4	0x0000_0000
PDMA_CSR5	PDMA_BA_ch5+0x00	R/W	PMAC控制和状态寄存器CH5	0x0000_0000
PDMA_CSR6	PDMA_BA_ch6+0x00	R/W	PMAC控制和状态寄存器CH6	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
TRIG_EN	-		APB_TWS		-		
15	14	13	12	11	10	9	8
-			TO_EN		-		
7	6	5	4	3	2	1	0
DAD_SEL		SAD_SEL		MODE_SEL		SW_RST	PDMACEN

Bits	描述											
[31:24]	-	保留.										
[23]	TRIG_EN	TRIG_EN 1 = 使能PDMA 数据读或写传输 0 =不起作用 注1：当PDMA 传输完成， 该位会被自动清0 注2：如果总线错误发生，所有的PDMA 传输会被停止。软件必须复位所有的PDMA 通道，然后重新触发。.										
[22:21]	-	保留.										
[20:19]	APB_TWS[1:0]	外设传输宽度选择 <table><tr><th>APB_TWS</th><th>描述</th></tr><tr><td>00</td><td>每个 PDMA 操作传输一个字 (32-bit)</td></tr><tr><td>01</td><td>每个 PDMA 操作传输一个字节 (8-bit)</td></tr><tr><td>1</td><td>每个 PDMA 操作传输一个半字 (16-bit)</td></tr><tr><td>11</td><td>保留.</td></tr></table>	APB_TWS	描述	00	每个 PDMA 操作传输一个字 (32-bit)	01	每个 PDMA 操作传输一个字节 (8-bit)	1	每个 PDMA 操作传输一个半字 (16-bit)	11	保留.
APB_TWS	描述											
00	每个 PDMA 操作传输一个字 (32-bit)											
01	每个 PDMA 操作传输一个字节 (8-bit)											
1	每个 PDMA 操作传输一个半字 (16-bit)											
11	保留.											

Bits	描述											
		注： 当且仅当 MODE_SEL 是IP到存储器模式 (APB-to-Memory) 或存储器到IP模式 (Memory-to-APB) 时，该域才有意义。										
[18:13]	-	保留。										
[12]	TO_EN	时间结束使能 该位会使能PDMA内部计数器。当计数器计数到0， TO_IS 会被置位。 1 = 使能PDMA内部计数器 0 = 禁止PDMA 内部计数器										
[11:8]	-	保留。										
[7:6]	DAD_SEL [1:0]	传输目的地址方向选择 <table><tr><th>DAD_SEL</th><th>描述</th></tr><tr><td>00</td><td>传输目的地址连续递增</td></tr><tr><td>01</td><td>保留。</td></tr><tr><td>10</td><td>传输目的地址固定（该特性可被用于当数据从多个源地址到一个单独的目的地址传输的情况）</td></tr><tr><td>11</td><td>传输目标地址回绕（当 PDMA_CBCR 等于0，PDMA_CDAR 和 PDMA_CBCR 寄存器会被 PDMA_DAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到PDMA_EN 被禁止。当PDMA_EN 禁止， PDMA会结束进行的传输，但是PDMA_BUF中保留的数据不会传输到目标地址）。</td></tr></table>	DAD_SEL	描述	00	传输目的地址连续递增	01	保留。	10	传输目的地址固定（该特性可被用于当数据从多个源地址到一个单独的目的地址传输的情况）	11	传输目标地址回绕（当 PDMA_CBCR 等于0， PDMA_CDAR 和 PDMA_CBCR 寄存器会被 PDMA_DAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到 PDMA_EN 被禁止。当 PDMA_EN 禁止， PDMA会结束进行的传输，但是 PDMA_BUF 中保留的数据不会传输到目标地址）。
DAD_SEL	描述											
00	传输目的地址连续递增											
01	保留。											
10	传输目的地址固定（该特性可被用于当数据从多个源地址到一个单独的目的地址传输的情况）											
11	传输目标地址回绕（当 PDMA_CBCR 等于0， PDMA_CDAR 和 PDMA_CBCR 寄存器会被 PDMA_DAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到 PDMA_EN 被禁止。当 PDMA_EN 禁止， PDMA会结束进行的传输，但是 PDMA_BUF 中保留的数据不会传输到目标地址）。											
[5:4]	SAD_SEL [1:0]	传输源地址方向选择 <table><tr><th>SAD_SEL</th><th>描述</th></tr><tr><td>00</td><td>传输源地连续递增</td></tr><tr><td>01</td><td>保留。</td></tr><tr><td>10</td><td>传输源地址固定（该特性可被用于当数据从一个单独的源地址到多个目的地址传输的情况） .</td></tr><tr><td>11</td><td>传输源地址回绕（当 PDMA_CBCR 等于0，PDMA_CSAR 和 PDMA_CBCR 寄存器会被 PDMA_SAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到PDMA_EN 被禁止。当PDMA_EN 禁止， PDMA会结束进行的传输，但是PDMA_BUF中保留的数据不会传输到目标地址）。</td></tr></table>	SAD_SEL	描述	00	传输源地连续递增	01	保留。	10	传输源地址固定（该特性可被用于当数据从一个单独的源地址到多个目的地址传输的情况） .	11	传输源地址回绕（当 PDMA_CBCR 等于0， PDMA_CSAR 和 PDMA_CBCR 寄存器会被 PDMA_SAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到 PDMA_EN 被禁止。当 PDMA_EN 禁止， PDMA会结束进行的传输，但是 PDMA_BUF 中保留的数据不会传输到目标地址）。
SAD_SEL	描述											
00	传输源地连续递增											
01	保留。											
10	传输源地址固定（该特性可被用于当数据从一个单独的源地址到多个目的地址传输的情况） .											
11	传输源地址回绕（当 PDMA_CBCR 等于0， PDMA_CSAR 和 PDMA_CBCR 寄存器会被 PDMA_SAR 和 PDMA_BCR 自动更新。 . PDMA会开始另外一段传输不需要软件触发，直到 PDMA_EN 被禁止。当 PDMA_EN 禁止， PDMA会结束进行的传输，但是 PDMA_BUF 中保留的数据不会传输到目标地址）。											
[3:2]	MODE_SEL [1:0]	PDMA模式选择										

Bits	描述			
			SAD_SEL	描述
			00	存储器到存储器模式 (Memory-to-Memory).
			01	IP到存储器模式 (APB-to-Memory)
			10	存储器到 IP 模式(Memory-to-APB).
			11	保留.
[1]	SW_RST	软件引擎复位 1 =写 1 到该位将复位内部状态机, 指针和内部缓存。控制寄存器的内部不会被清除。 该位在几个时钟周期之后将自动清除。 0 =写 0 到该位无效		
[0]	PDMACEN	PDMA通道使能 设置该位为 1 使能 PDMA 的操作。如果该位被清除, PDMA 将忽略所有的 PDMA 请求并强制总线主机进入 IDLE 状态。 注: SW_RST将清除该位。		

PDMA传输源地址寄存器(PDMA_SARx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_SAR1	PDMA_BA_ch1+0x04	R/W	PDMA传输源地址寄存器 CH1	0x0000_0000
PDMA_SAR2	PDMA_BA_ch2+0x04	R/W	PDMA 传输源地址寄存器 CH2	0x0000_0000
PDMA_SAR3	PDMA_BA_ch3+0x04	R/W	PDMA 传输源地址寄存器 CH3	0x0000_0000
PDMA_SAR4	PDMA_BA_ch4+0x04	R/W	PDMA传输源地址寄存器 CH4	0x0000_0000
PDMA_SAR5	PDMA_BA_ch5+0x04	R/W	PDMA传输源地址寄存器 CH5	0x0000_0000
PDMA_SAR6	PDMA_BA_ch6+0x04	R/W	PDMA 传输源地址寄存器CH6	0x0000_0000

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

Bits	描述	
[31:0]	PDMA_SAR [31:0]	PDMA 传输源地址寄存器 该域表示一 32-位 PDMA 源地址。 注：源地址必须字对齐。

PDMA 传输目的地址寄存器 (PDMA DARx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_DAR1	PDMA_BA_ch1+0x08	R/W	PDMA 传输目的地址寄存器 CH1	0x0000_0000
PDMA_DAR2	PDMA_BA_ch2+0x08	R/W	PDMA 传输目的地址寄存器 CH2	0x0000_0000
PDMA_DAR3	PDMA_BA_ch3+0x08	R/W	PDMA传输目的地址寄存器 CH3	0x0000_0000
PDMA_DAR4	PDMA_BA_ch4+0x08	R/W	PDMA 传输目的地址寄存器 CH4	0x0000_0000
PDMA_DAR5	PDMA_BA_ch5+0x08	R/W	PDMA 传输目的地址寄存器 CH5	0x0000_0000
PDMA_DAR6	PDMA_BA_ch6+0x08	R/W	PDMA 传输目的地址寄存器 CH6	0x0000_0000

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

Bits	描述	
[31:0]	PDMA_DAR [31:0]	PDMA传输目的地址寄存器 该域表示一 32-位 PDMA 目的地址 注：目的地址必须字对齐

PDMA传输字节计数寄存器(PDMA_BCRx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_BCR1	PDMA_BA_ch1+0x0C	R/W	PDMA传输字节计数寄存器CH1	0x0000_0000
PDMA_BCR2	PDMA_BA_ch2+0x0C	R/W	PDMA传输字节计数寄存器CH2	0x0000_0000
PDMA_BCR3	PDMA_BA_ch3+0x0C	R/W	PDMA传输字节计数寄存器CH3	0x0000_0000
PDMA_BCR4	PDMA_BA_ch4+0x0C	R/W	PDMA传输字节计数寄存器CH4	0x0000_0000
PDMA_BCR5	PDMA_BA_ch5+0x0C	R/W	PDMA传输字节计数寄存器CH5	0x0000_0000
PDMA_BCR6	PDMA_BA_ch6+0x0C	R/W	PDMA传输字节计数寄存器CH6	0x0000_0000

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

Bits	描述	
[31:24]	-	保留.
[15:0]	PDMA_BCR [15:0]	PDMA 传输字节计数寄存器 该域表示一 16-位 PDMA 传输字节计数数目 注：当在存储器到存储器(PDMA_CSR [MODE_SEL] = 00) 模式，传输字节计数数目必须字对齐

PDMA当前源地址寄存器(PDMA CSARx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_CSAR1	PDMA_BA_ch1+0x14	R	PDMA 当前源地址寄存器 CH1	0x0000_0000
PDMA_CSAR2	PDMA_BA_ch2+0x14	R	PDMA 当前源地址寄存器 CH2	0x0000_0000
PDMA_CSAR3	PDMA_BA_ch3+0x14	R	PDMA 当前源地址寄存器 CH3	0x0000_0000
PDMA_CSAR4	PDMA_BA_ch4+0x14	R	PDMA 当前源地址寄存器 CH4	0x0000_0000
PDMA_CSAR5	PDMA_BA_ch5+0x14	R	PDMA 当前源地址寄存器 CH5	0x0000_0000
PDMA_CSAR6	PDMA_BA_ch6+0x14	R	PDMA 当前源地址寄存器 CH6	0x0000_0000

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

Bits	描述	
[31:0]	PDMA_CSAR [31:0]	PDMA 当前源地址寄存器 (只读) 该域指示 PDMA 正在传输的源地址。

PDMA当前目的地址寄存器(PDMA_CDARx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_CDAR1	PDMA_BA_ch1+0x18	R	PDMA 当前目的地址寄存器 CH1	0x0000_0000
PDMA_CDAR2	PDMA_BA_ch2+0x18	R	PDMA 当前目的地址寄存器CH2	0x0000_0000
PDMA_CDAR3	PDMA_BA_ch3+0x18	R	PDMA 当前目的地址寄存器 CH3	0x0000_0000
PDMA_CDAR4	PDMA_BA_ch4+0x18	R	PDMA 当前目的地址寄存器 CH4	0x0000_0000
PDMA_CDAR5	PDMA_BA_ch5+0x18	R	PDMA 当前目的地址寄存器 CH5	0x0000_0000
PDMA_CDAR6	PDMA_BA_ch6+0x18	R	PDMA 当前目的地址寄存器 CH6	0x0000_0000

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

Bits	描述	
[31:0]	PDMA_CDAR [31:0]	PDMA 当前目的地址寄存器（只读） 该域指示 PDMA 正在传输的目的地址。

PDMA 当前字节计数寄存器 (PDMA_CBCRx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_CBCR1	PDMA_BA_ch1+0x1C	R	PDMA当前字节计数寄存器CH1	0x0000_0000
PDMA_CBCR2	PDMA_BA_ch2+0x1C	R	PDMA当前字节计数寄存器CH2	0x0000_0000
PDMA_CBCR3	PDMA_BA_ch3+0x1C	R	PDMA当前字节计数寄存器CH3	0x0000_0000
PDMA_CBCR4	PDMA_BA_ch4+0x1C	R	PDMA当前字节计数寄存器CH4	0x0000_0000
PDMA_CBCR5	PDMA_BA_ch5+0x1C	R	PDMA当前字节计数寄存器CH5	0x0000_0000
PDMA_CBCR6	PDMA_BA_ch6+0x1C	R	PDMA当前字节计数寄存器CH6	0x0000_0000

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

Bits	描述	
[31:24]	-	保留。
[23:0]	PDMA_CBCR [23:0]	PDMA 当前字节计数寄存器（只读） 该域指示 PDMA 当前剩下的字节计数。 注：当PDMA完成数据传输，该域会被改变(数据传输到目的地址)，

PDMA中断使能控制寄存器(PDMA_IERx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_IER1	PDMA_BA_ch1+0x20	R/W	PDMA中断使能控制寄存器CH1	0x0000_0001
PDMA_IER2	PDMA_BA_ch2+0x20	R/W	PDMA中断使能控制寄存器CH2	0x0000_0001
PDMA_IER3	PDMA_BA_ch3+0x20	R/W	PDMA中断使能控制寄存器CH3	0x0000_0001
PDMA_IER4	PDMA_BA_ch4+0x20	R/W	PDMA中断使能控制寄存器CH4	0x0000_0001
PDMA_IER5	PDMA_BA_ch5+0x20	R/W	PDMA中断使能控制寄存器 CH5	0x0000_0001
PDMA_IER6	PDMA_BA_ch6+0x20	R/W	PDMA中断使能控制寄存器CH6	0x0000_0001

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

Bits	描述										
[31:7]	-	保留.									
[6]	TO_IE	时间结束中断使能 1 = 使能时间结束中断 0 = 禁止时间结束中断									
[5:2]	WRA_BCR_IE [3:0]	回绕字节计数中断使能 <table><tr><th>SAD_SEL[1:0]</th><th>描述</th></tr><tr><td>0001</td><td>PDMA_CBCR 等于0的中断使能</td></tr><tr><td>0100</td><td>PDMA_CBCR 等于1/2 PDMA_BCR.的中断使能</td></tr><tr><td>其他</td><td>保留.</td></tr></table>		SAD_SEL[1:0]	描述	0001	PDMA_CBCR 等于0的中断使能	0100	PDMA_CBCR 等于1/2 PDMA_BCR.的中断使能	其他	保留.
SAD_SEL[1:0]	描述										
0001	PDMA_CBCR 等于0的中断使能										
0100	PDMA_CBCR 等于1/2 PDMA_BCR.的中断使能										
其他	保留.										
[1]	TD_IE	PDMA 传输完成中断使能 1 = 使能 PDMA 传输完成中断 0 = 禁用 PDMA 传输完成中断.									
[0]	TABORT_IE	PDMA 读/写 目标中止中断使能									

Bits	描述	
		1 = 使能 PDMA 传输过程中的目标中止中断 0 = 禁用 PDMA 传输过程中的目标中止中断.

PDMA中断状态寄存器(PDMA_ISRx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_ISR1	PDMA_BA_ch1+0x24	R/W	PDMA中断状态寄存器CH1	0x0x0x_0000
PDMA_ISR2	PDMA_BA_ch2+0x24	R/W	PDMA中断状态寄存器CH2	0x0x0x_0000
PDMA_ISR3	PDMA_BA_ch3+0x24	R/W	PDMA中断状态寄存器CH3	0x0x0x_0000
PDMA_ISR4	PDMA_BA_ch4+0x24	R/W	PDMA中断状态寄存器CH4	0x0x0x_0000
PDMA_ISR5	PDMA_BA_ch5+0x24	R/W	PDMA中断状态寄存器CH5	0x0x0x_0000
PDMA_ISR6	PDMA_BA_ch6+0x24	R/W	PDMA中断状态寄存器CH6	0x0x0x_0000

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

Bits	描述	
[31:7]	-	保留
[6]	TO_IS	<p>时间结束中断状态标志</p> <p>该标志表示PDMA已经等待了 外设请求PDMA_TCR 中定义的时间</p> <p>1 = 时间结束标志.</p> <p>0 = 时间没有结束标志</p> <p>注: 向该位写1清0</p>
[5:2]	WRA_BCR_IS [3:0]	<p>回绕传输字节计数中断状态标志</p> <p>WAR_BCR_IS [0] (xxx1) = PDMA_CBCR 等于0标志.</p> <p>WAR_BCR_IS [2] (x1xx) = PDMA_CBCR 等于1/2 PDMA_BCR 标志.</p> <p>注: 向各个位写1清0, 该域段仅在回绕模式有效. (PDMA_CSR[DAD_SEL] =11 或 PDMA_CSR[SAD_SEL] =11)</p>
[1]	TD_IS	<p>传输完成中断状态标志</p> <p>该位表示 PDMA完成所有传输.</p> <p>1 = 完成.</p> <p>0 = 还没完成</p> <p>注: 向该位写1清0</p>

Bits	描述	
[0]	TABORT_IS	<p>PDMA 读/写 目标中止中断标志</p> <p>1 = 收到总线错误应答</p> <p>0 = 没有收到总线错误应答</p> <p>注1: 向该位写1清0</p> <p>注2: PDMA_ISR [TABORT_IF] 指示总线主机是否收到 ERROR 响应。如果总线主机收到了 ERROR 响应, 则意味着目标中止发生了。PDMAC 将停止传输和响应该事件到软件, 然后进入 IDLE 状态。当目标中止发生, 软件必须复位 PDMA, 再次传输那些数据。</p>

PDMA定时器计数设置寄存器(PDMA_TCRx)

寄存器	偏移量	R/W	描述	复位后的值
PDMA_TCR1	PDMA_BA_ch1+0x28	R/W	PDMA 定时器计数设置寄存器 CH1	0x0000_0000
PDMA_TCR2	PDMA_BA_ch2+0x28	R/W	PDMA定时器计数设置寄存器 CH2	0x0000_0000
PDMA_TCR3	PDMA_BA_ch3+0x28	R/W	PDMA定时器计数设置寄存器 CH3	0x0000_0000
PDMA_TCR4	PDMA_BA_ch4+0x28	R/W	PDMA 定时器计数设置寄存器CH4	0x0000_0000
PDMA_TCR5	PDMA_BA_ch5+0x28	R/W	PDMA 定时器计数设置寄存器 CH5	0x0000_0000
PDMA_TCR6	PDMA_BA_ch6+0x28	R/W	PDMA定时器计数设置寄存器 CH6	0x0000_0000

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

Bits	描述	
[31:16]	-	保留.
[15:0]	PDMA_TCR	<p>PDMA 定时器计数设置寄存器</p> <p>每个PDMA 控制器包含一个内部计数器。当设定PDMA_CSR[TO_EN] 寄存器，内部计数器加载PDMA_TCR值并开始下数，当此内部计数器下数至0且PDMA_IER[TO_IE]被设起时，PDMA产生中断，当完成周边程序时，此内部计数器将会重载新的值。</p>

CRC 控制寄存器 (CRC CTL)

寄存器	偏移量	R/W	描述	复位后的值
CRC_CTL	CRC_BA + 0x00	R/W	CRC 控制寄存器	0x2000_0000

31	30	29	28	27	26	25	24
CRC_MODE		CPU_WDLEN		CHECKSUM_COM	WDATA_COM	CHECKSUM_RVS	WDATA_RVS
23	22	21	20	19	18	17	16
TRIG_EN	-						
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-						CRC_RST	CRCEN

Bits	描述	
[31:30]	CRC_MODE [1:0]	CRC 多项式模式
</		

Bits	描述	
		1 = 对校验和进行一次补码 0 = 不对校验和进行一次补码
[26]	WDATA_COM	写数据补码 1 = 对 CRC 写入数据进行一次补码 0 = 不对 CRC 写入数据进行一次补码
[25]	CHECKSUM_RVS	校验和取反 1 = CRC 校验和位顺序倒转 0 = CRC 校验和位顺序不倒转 注意:如果校验和数据是 0XDD7B0F2E, 位顺序颠倒后是 0x74F0DEBB
[23]	WDATA_RVS	写数据顺序倒转 1 = CRC写数据位顺序倒转(每个字节) 0 = CRC写数据位顺序不倒转. 注意: 如果写数据是 0xAABCCDD, 位倒转后 CRC 写数据是0x55DD33BB
[24]	TRIG_EN	触发使能 1 = 使能 CRC DMA数据读或写传输. 0 = 没影响 注意1: 如果该位触发, 表示 CRC 引擎工作在 CRC DMA 模式, 所以不要填任何数载在CRC_WDATA 寄存器. 注意2: 当 CRC DMA 传输完成, 该位会被自动清零. 注意3: 如果总线错误发生, 所有CRC DMA传输会停止. 软件必须复位所有 DMA 通道, 然后再触发.
[22:2]	-	保留.
[1]	CRC_RST	CRC引擎复位 0 = 写 0 到该位没有影响. 1 = 写1到该位会复位内部CRC状态机和内部缓存, 控制寄存器的内容不会被清除, 在几个时钟周期之后该位自动清0. 注: 当工作在 CPU PIO 模式, 设置该位会重新加载初始的种子值.
[0]	CRCEN	CRC 通道使能 设置该位使能CRC的操作. 当工作在 CRC DMA 模式 (TRIG_EN = 1), 如果用户清0该位, DMA操作会继续直到所有CRC DMA操作完成, TRIG_EN位会不确定直到所有CRC DMA操作完成. 但是在这种情况下CRC_DMAISR [BLKD_IF] 标志是不可用的, 当 TRIG_EN = 0, 用户可以通过读CRC_CHECKSUM 寄存器读CRC 结果 当工作在 CRC DMA 模式 (TRIG_EN = 1), 如果用户想立即停止传输, 可以写1到, user CRC_RST位停止传输. .

CRC DMA传输源地址寄存器 (CRC_DMASAR)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMASAR	CRC_BA + 0x04	R/W	CRC DMA 传输源地址寄存器	0x0000_0000

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

Bits	描述	
[31:0]	CRC_DMASAR [31:0]	<p>CRC DMA 传输源地址寄存器</p> <p>该域表示一个 32-bit CRC DMA.的源地址</p> <p>注意：源地址必须字对齐</p>

CRC DMA 传输字节计数寄存器 (CRC_DMABCR)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC DMA 传输字节计数寄存器	0x0000_0000

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

Bits	描述	
[31:0]	-	保留
[15:0]	CRC_DMABCR [15:0]	CRC DMA传输字节计数寄存器 该域表示一个16-bit CRC DMA的传输字节计数值

CRC DMA 当前源地址 寄存器 (CRC_DMACSAR)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMACSAR	CRC_BA + 0x14	R	CRC DMA 当前源地址寄存器	0x0000_0000

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

Bits	描述	
[31:0]	CRC_DMACSAR[31:0]	CRC DMA 当前源地址寄存器 (只读) 该域表示CRC DMA 正在传输的源地址

CRC DMA 当前字节计数寄存器 (CRC_DMACBCR)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMACBCR	CRC_BA + 0x1C	R	CRC DMA当前字节计数寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[15:0]	CRC_DMACBCR[15:0]	CRC DMA 当前字节计数寄存器 (只读) 该域指示CRC_DMA当前剩下的字节计数。 注: CRC_RST 会清零 寄存器 值。

CRC DMA 中断使能控制寄存器 (CRC DMAIER)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMAIER	CRC_BA + 0x20	R/W	CRC DMA中断使能控制寄存器	0x0000_0001

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

Bits	描述	
[31:2]	-	保留.
[1]	BLKD_IE	CRC DMA 传输完成中断使能 1 =当 CRC DMA传输完成, 使能中断发生器. 0 =当 CRC DMA传输完成, 禁止中断发生器.
[0]	TABORT_IE	CRC DMA 读/写 目标中止中断使能 1 = 当 CRC DMA 传输过程中, 使能目标中止中断. 0 =当 CRC DMA 传输过程中, 禁止目标中止中断..

CRC DMA 中断状态 寄存器 (CRC_DMAISR)

寄存器	偏移量	R/W	描述	复位后的值
CRC_DMAISR	CRC_BA + 0x24	R/W	CRC DMA 中断状态寄存器	0x0000_0000

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

Bits	描述	
[31:2]	-	保留.
[1]	BLKD_IF	<p>快传输结束中断标志</p> <p>该位表示CRC DMA 传输完成.</p> <p>1 = 完成</p> <p>0 = 还没完成</p> <p>软件写1清零该位</p>
[0]	TABORT_IF	<p>CRC DMA 读/写目标中止中断标志</p> <p>1 = 收到总线 错误反应</p> <p>0 = 没收到总线错误反应</p> <p>软件写1清零该位</p> <p>注: CRC_DMAISR [TABORT_IF]表示总线主机是否接收到错误反应, 如果总线主机接收到错误反应, 意味着目标中止发生, DMA会停止传输, 软件对这件事做出反应, 然后进入空闲模式。当目标中止发生, 软件必须复位DMA, 然后再传输那些数据。</p>

CRC 写数据寄存器 (CRC_WDATA)

寄存器	偏移量	R/W	描述	复位后的值
CRC_WDATA	CRC_BA + 0x80	R/W	CRC 写数据寄存器	0x0000_0000

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

Bits	描述	
[31:0]	CRC_WDATA [31:0]	<p>CRC 写数据寄存器</p> <p>当工作在CPU PIO (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) 模式, 软件可以写数据到该域执行 CRC 操作;</p> <p>当工作在 CRC DMA 模式 (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0), 该域被用作DMA内部缓存</p> <p>注1: 当工作在 CRC DMA 模式, 不要填任何数据到该域.</p> <p>注2: CRC_CTL [WDATA_COM] 和 CRC_CTL [WDATA_RVS] 的设置会影响该域。例如: 如果WDATA_RVS = 1, 写CRC_WDATA 寄存器的数据是 0xAABBCCDD, 从CRC_WDATA 寄存器 读到的数据会是 0x55DD33BB.</p>

CRC 种子寄存器 (CRC_SEED)

寄存器	偏移量	R/W	描述	复位后的值
CRC_SEED	CRC_BA + 0x84	R/W	CRC 种子寄存器	0xFFFF_FFF F

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

Bits	描述	
[31:0]	CRC_SEED [31:0]	CRC 种子寄存器 该域表示 CRC种子值。

CRC 校验和寄存器 (CRC CHECKSUM)

寄存器	偏移量	R/W	描述	复位后的值
CRC_CHECKSUM	CRC_BA + 0x88	R	CRC 校验和 寄存器	0x0000_0000

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

Bits	描述	
[31:0]	CRC_CHECKSUM[31:0]	CRC 校验和 寄存器 该域表示CRC 校验和

PDMA全局控制和状态寄存器(DMA_GCRCSR)

寄存器	偏移量	R/W	描述	复位后的值
DMA_GCRCSR	DMA_BA_GCR + 0x00	R/W	DMA全局控制和状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							CRC_CLK_EN
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-	CLK6_EN	CLK5_EN	CLK4_EN	CLK3_EN	CLK2_EN	CLK1_EN	CLK0_EN
7	6	5	4	3	2	1	0
-							-

Bits	描述	
[31:25]	-	保留.
[24]	CRC_CLK_EN	CRC控制器时钟使能控制 1 = 使能 0 = 禁用
[23:15]	-	保留
[14]	CLK6_EN	DMA 控制器通道 6 时钟使能控制 1 = 使能 0 = 禁用
[13]	CLK5_EN	DMA 控制器通道 5 时钟使能控制 1 = 使能 0 = 禁用
[12]	CLK4_EN	DMA 控制器通道 4 时钟使能控制 1 = 使能 0 = 禁用
[11]	CLK3_EN	DMA 控制器通道 3 时钟使能控制 1 = 使能 0 = 禁用
[10]	CLK2_EN	DMA 控制器通道 2 时钟使能控制 1 = 使能 0 = 禁用

Bits	描述	
[9]	CLK1_EN	DMA 控制器通道 1 时钟使能控制 1 = 使能 0 = 禁用
[8]	CLK0_EN	DMA 控制器通道 0 时钟使能控制 1 = 使能 0 = 禁用
[7:0]	-	保留.

DMA服务选择控制寄存器0 (DMA_DSSR0)

寄存器	偏移量	R/W	描述	复位后的值
DMA_DSSR0	DMA_BA_GCR +0x04	R/W	DMA服务选择控制寄存器0	0x1F1F_1F00

31	30	29	28	27	26	25	24
-			CH3_SEL				
23	22	21	20	19	18	17	16
-			CH2_SEL				
15	14	13	12	11	10	9	8
-			CH1_SEL				
7	6	5	4	3	2	1	0
-							

Bits	描述				
[31:29]	-	保留.			
[28:24]	CH3_SEL	通道3 选择 该域定义了哪一个外设和PDMA 通道3相连。软件可以通过 CH3_SEL 配置。该通道配置和 CH1_SEL 域相同，请参考CH1_SEL的说明。			
[23:21]	-	保留.			
[20:16]	CH2_SEL	通道2 选择 该域定义了哪一个外设和PDMA 通道2相连。软件可以通过 CH2_SEL 配置。该通道配置和 CH1_SEL 域相同，请参考CH1_SEL的说明。			
[15:13]	-	保留.			
[12:8]	CH1_SEL	通道1 选择 该域定义了哪一个外设和PDMA 通道1相连。软件可以通过 CH1_SEL 配置。该通道配置和 CH1_SEL 域相同，请参考CH1_SEL的说明。.			
		CH1_SEL	连接	CH1_SEL	连接
		00000	连接到 SPI0_TX.	10000	连接到 SPI0_RX.
		00001	连接到 SPI1_TX.	10001	连接到 SPI1_RX.
		00010	连接到 UART0_TX.	10010	连接到 UART0_RX.
		00011	连接到 UART1_TX.	00011	连接到 UART1_RX.
		00100	连接到 USB_TX.	10100	连接到 USB_RX.
		00101	连接到 I ² S_TX.	10101	连接到 I ² S_RX
		00110	连接到 DAC0_TX.	10110	连接到 ADC.

Bits	描述				
		00111	连接到 DAC1_TX.	10111	保留.
		01000	连接到 SPI2_TX.	11000	连接到 SPI2_RX.
		01001	连接到 TMR0.	11001	连接到 PWM0_CH0.
		01010	连接到 TMR1.	11010	连接到 PWM0_CH2.
		01011	连接到 TMR2	11011	连接到 PWM1_CH0.
		01100	连接到 TMR3	11100	连接到 PWM1_CH2.
		其他：禁止连接任何外设			
[7:0]	-	保留.			

DMA服务选择控制寄存器1 (DMA_DSSR1)

寄存器	偏移量	R/W	描述	复位后的值
DAM_DSSR1	DMA_BA_GCR +0x08	R/W	DMA服务选择控制寄存器1	0x001F_1F1F

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-			CH6_SEL				
15	14	13	12	11	10	9	8
-			CH5_SEL				
7	6	5	4	3	2	1	0
-			CH4_SEL				

Bits	描述				
[31:21]	-	保留。			
[20:16]	CH6_SEL	通道6 选择 该域定义了哪一个外设和PDMA 通道6相连。软件可以通过 CH6_SEL 配置。该通道配置和 CH4_SEL 域相同，请参考CH4_SEL的说明。			
[15:13]	-	保留。			
[12:8]	CH5_SEL	通道5 选择 该域定义了哪一个外设和PDMA 通道5相连。软件可以通过 CH5_SEL 配置。该通道配置和 CH4_SEL 域相同，请参考CH4_SEL的说明。			
[7:5]	-	保留。			
[4:0]	CH4_SEL	通道4 选择 该域定义了哪一个外设和PDMA 通道4相连。软件可以通过 CH4_SEL 配置。该通道配置和 CH4_SEL field4'b0000域相同。			
		CH4_SEL	连接	CH4_SEL	连接
		00000	连接到 SPI0_TX.	10000	连接到SPI0_RX.
		00001	连接到SPI1_TX.	10001	连接到SPI1_RX.
		00010	连接到UART0_TX.	10010	连接到UART0_RX.
		00011	连接到UART1_TX.	10011	连接到UART1_RX.
		00100	连接到USB_TX.	10100	连接到USB_RX.
		00101	连接到I ² S_TX.	10101	连接到I ² S_RX.
		00110	连接到DAC0_TX.	10110	连接到ADC.

Bits	描述				
		00111	连接到DAC1_TX.	10111	保留.
		01000	连接到SPI2_TX.	11000	连接到SPI2_RX.
		01001	连接到TMR0.	11001	连接到PWM0_CH0.
		01010	连接到TMR1.	11010	连接到PWM0_CH2.
		01011	连接到TMR2.	11011	连接到PWM1_CH0.
		01100	连接到TMR3.	11100	连接到PWM1_CH2.
		其他：禁止连接任何外设			

DMA全局中断状态寄存器(DMA_GCRISR)

寄存器	偏移量	R/W	描述	复位后的值
DMA_GCRISR	DMA_BA_GCR + 0x0C	R	DMA全局中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-	-						
23	22	21	20	19	18	17	16
-							CRC_INTR
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-	INTR6	INTR5	INTR4	INTR3	INTR2	INTR1	INTR0

Bits	描述	
[31:17]	-	保留
[16]	CRC_INTR	CRC控制器中断管脚状态 该位是CRC控制器的中断状态。 注：该位只读
[15:7]	-	保留
[6]	INTR6	通道 6 中断管脚状态 该位是 DMA 通道 6 的中断状态。 注：该位只读
[5]	INTR5	通道 5 中断管脚状态 该位是 PDMA 通道 5 的中断状态。 注：该位只读
[4]	INTR4	通道 4 中断管脚状态 该位是 DMA 通道 4 的中断状态。 注：该位只读
[3]	INTR3	通道 3 中断管脚状态 该位是 PDMA 通道 3 的中断状态。 注：该位只读
[2]	INTR2	通道 2 中断管脚状态 该位是 PDMA 通道 2 的中断状态。 注：该位只读
[1]	INTR1	通道 1 中断管脚状态

Bits	描述	
		该位是 PDMA 通道 1 的中断状态。 注：该位只读
[0]	INTRO	通道 0 中断管脚状态 该位是 PDMA 通道 0 的中断状态。 注：该位只读

5.8 外部总线接口

5.8.1 概述

该芯片配备了一个外部总线接口 (EBI)，以供外部设备使用。为了节省外部设备与芯片的连接，EBI 支持地址总线与数据总线复用模式。EBI 支持地址总线与数据总线多路复用的模式，地址锁存 (ALE) 信号区分地址与数据。

5.8.2 特征

- 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)
- 支持可变的外部总线基本时钟 (MCLK)
- 支持 8-位 或 16-位 数据宽度
- 支持可变的数据访问时间 (tACC)，地址锁存时间 (tALE) 和地址保持时间 (tAHD)
- 支持地址总线与数据总线多路复用以节省地址管脚
- 支持可配置的空闲周期以用于不同访问条件：写命令完成 (W2X)，连续读(R2R)，连续写(R2W)
- 支持PDMA 和VDMA 传输

5.8.3 框图

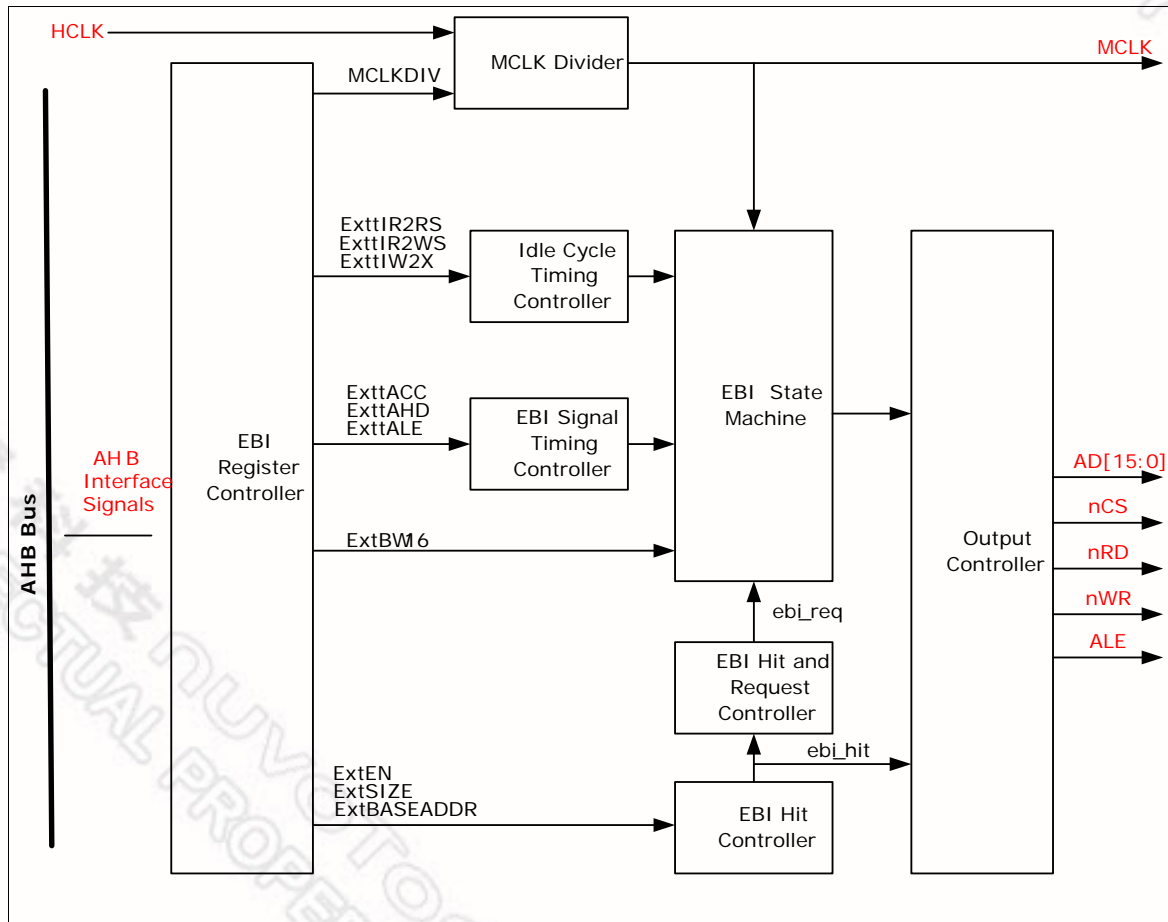


图 5-26 EBI框图

5.8.4 功能描述

5.8.4.1 EBI区域和地址映射

EBI 映射地址分布在 0x6000_0000 ~ 0x6001_FFFF，总共内存空间为 128Kbyte。当系统请求地址在 EBI 的内存空间内时，相应的 EBI 芯片选择信号有效，EBI 状态机工作。

对于 8-位设备 (64Kbyte)，EBI 同时映射该 64Kbyte 设备到 0x6000_0000 ~ 0x6000_FFFF 和 0x6001_0000 ~ 0x6001_FFFF。因为只有低16位地址A0~A15被引出。

5.8.4.2 EBI数据宽度连接

EBI 支持具有多路地址总线和数据总线的设备。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单元锁存地址。这样，管脚 ALE 需要连接到锁存器上锁存地址值。管脚 AD 为锁存器的输入，锁存器的输出连接到外部设备的地址总线上。对于 16-位设备，AD [15:0] 由地址与16-位数据共用。对于 8-位器件，仅 AD [7:0] 由地址与 8-位数据共用，AD [15:8]作地址，直接与 8-位设备连接。

对于 8-位数据宽度，芯片系统地址[15:0] 作为设备地址[15:0]。对于 16-位数据宽度，芯片系统地址[16:1] 作为设备地址[15:0]，芯片系统地址位 [0] 无作用。

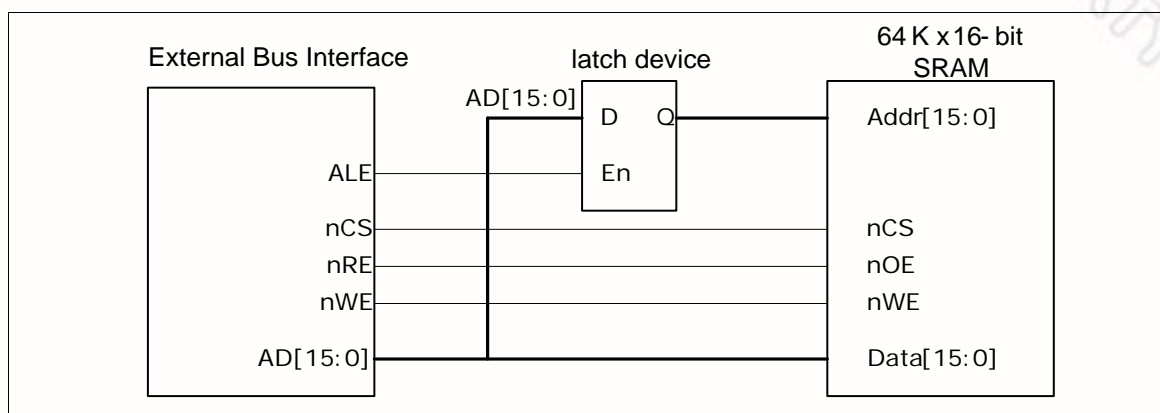


图 5-27 16-位 EBI 数据宽度与 16-位设备的连接

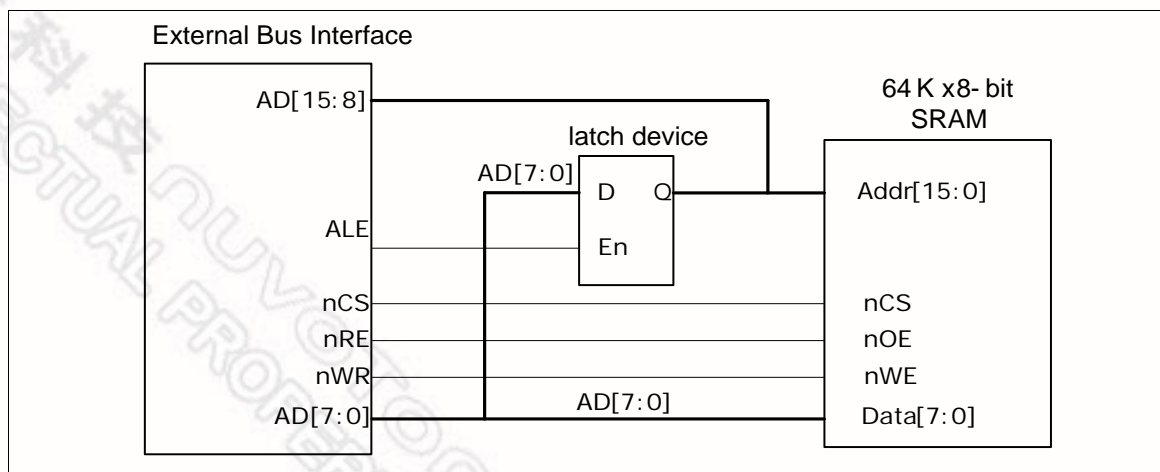


图 5-28 8-位 EBI 数据宽度和 8-位设备的连接

当系统访问数据宽度大于 EBI 的数据宽度，EBI 控制器将访问一次以上以完成操作。例如，如果系统通过 EBI 设备请求 32-位数据，当 EBI 为8-位数据宽度时，EBI 控制器将访问 4 次完成操作。

5.8.4.3 EBI操作控制

MCLK 控制

当 EBI 工作时，芯片内所有的EBI 信号将通过 MCLK 进行同步。当芯片以较低工作频率连接到外部设备，MCLK 可以通过设定寄存器EBICON 的 MCLKDIV 最多分频到 HCLK/32。因此，芯片可以适用于宽频率范围的 EBI 设备。如果 MCLK 设置为 HCLK/1，则 EBI 信号与 MCLK 的正边沿同步，否则与 MCLK 的负边沿同步。

操作与访问时序控制

开始访问时，片选 (nCS) 置低并等待一个 MCLK 时间用于地址建立时间 (tASU) 以使地址稳定。地址稳定后 ALE 置高并保持一段时间 (tALE) 用于地址锁存。地址锁存后，ALE 置低并等待一个 MCLK 的周期用于锁存保持时间 (tLHD) 和另一个插入到地址保持时间之后的 MCLK 的时间 (tA2D) 用于总线转换（地址到数据）。然后当读访问时 nRD 置低或写访问时 nWR 置低。在保持访问时间 (tACC) 后 nRD 或nWR 置高用于读输出稳定或写完成。之后，EBI 信号保持数据访问时间 (tAHD)，片选置高，地址由当前访问控制释放。

EBI 控制器提供了灵活的时序控制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，通过设置寄存器 EXTIME 的 ExttAHD，tAHD 可以在 1~8 MCLK 周期调节，通过设置寄存器 EXTIME 的 ExttACC，tACC可以在 1~32 MCLK 周期调节，通过设置寄存器 EBICON 的 tALE，tALE可以在 1~8 MCLK 周期调节。

参数	值	单元	描述
tASU	1	MCLK	地址锁存建立时间.
tALE	1~8	MCLK	ALE 高时间。由 EBICON 的 ExttALE 控制.
tLHD	1	MCLK	地址锁存保持时间.
tA2D	1	MCLK	地址到数据的延时（总线转换时间）.
tACC	1~32	MCLK	数据访问时间，由 EXTIME 的ExttACC 控制
tAHD	1~8	MCLK	数据访问保持时间，由 EXTIME 的 ExttAHD 控制.
IDLE	1~16	MCLK	空闲周期，由 EXTIME 的 ExtIR2R、ExtIR2W 和 ExtIW2X 控制

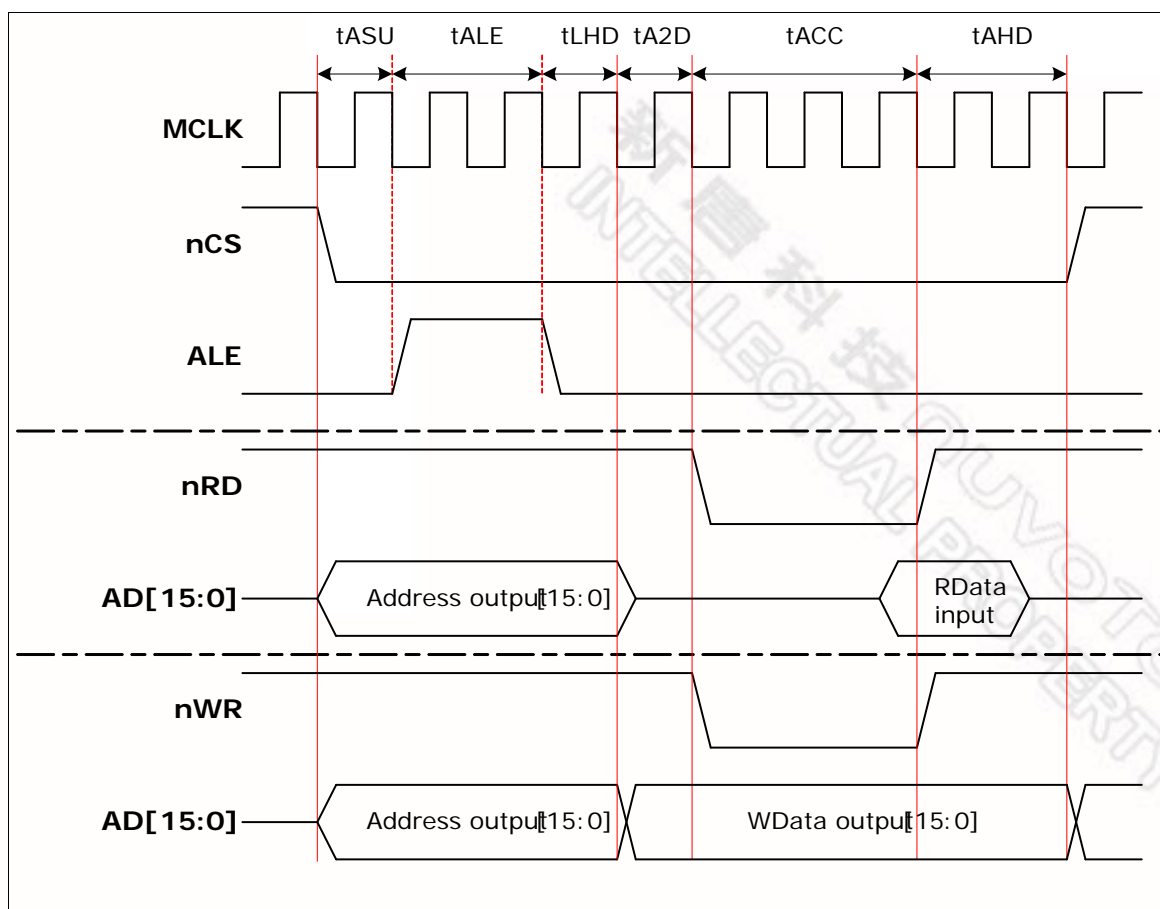


图 5-29 16-位数据宽度的时序控制波形

上图是一个设置 16-位数据宽度的例子。在该例中，AD 总线用作地址 [15:0] 和数据 [15:0]。当 ALE 置高，AD 为地址输出。在地址锁存后，ALE 置低并且在读取访问操作时，AD 总线转换成高阻以等待设备输出数据，或用于写数据输出。

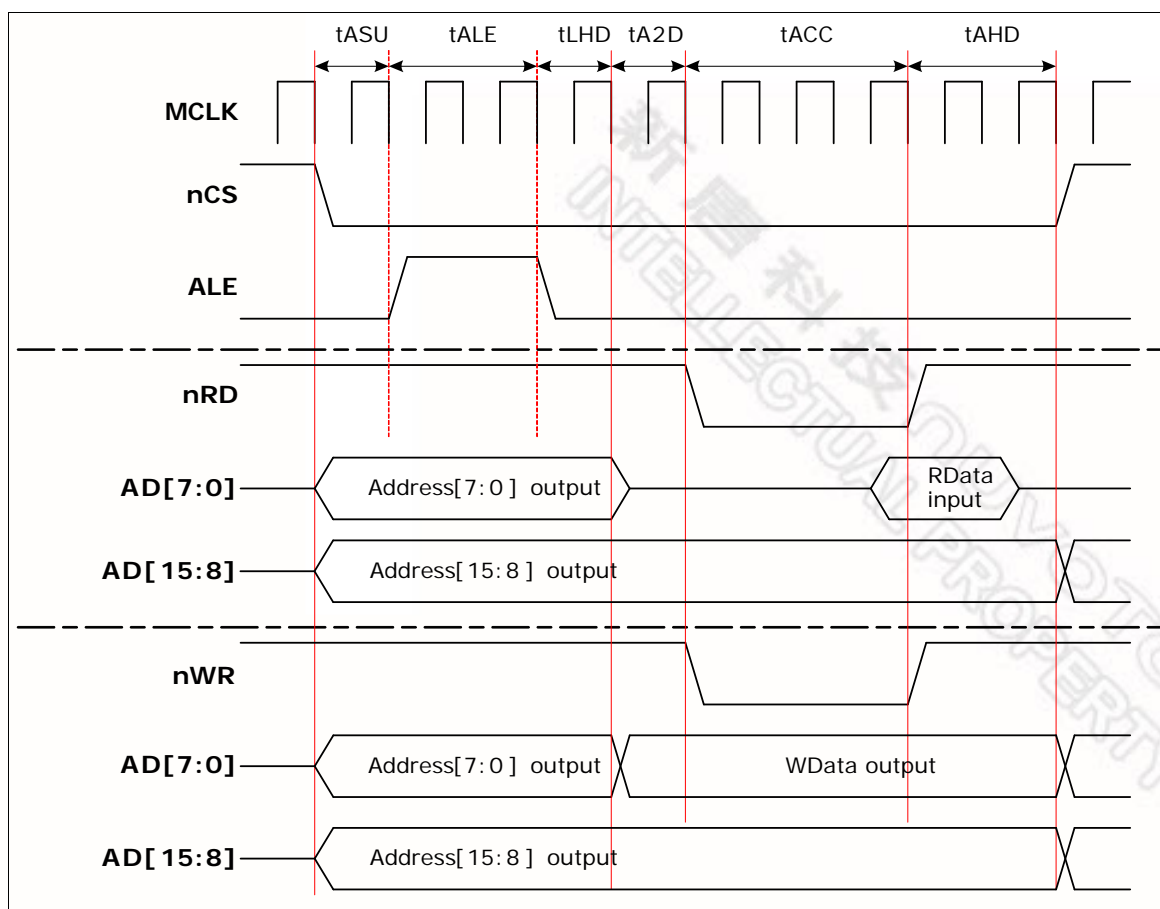


图 5-30 8-位数据宽度时序控制波形

上图是一个设置 8-位数据宽度的例子。8-位和 16-位数据宽度的不同之处在于 AD[15:8]。在 8-位数据宽度的设置中，AD[15:8] 总为地址[15:8]输出，因此外部锁存只需要 8-位宽度。

插入空闲周期

当 EBI 连续访问时，如果器件访问时间比系统操作慢得多，可能会出现总线冲突。EBI 控制器支持额外空闲周期以解决该问题。在空闲周期，所有 EBI 的控制信号无效。下图为空闲周期。

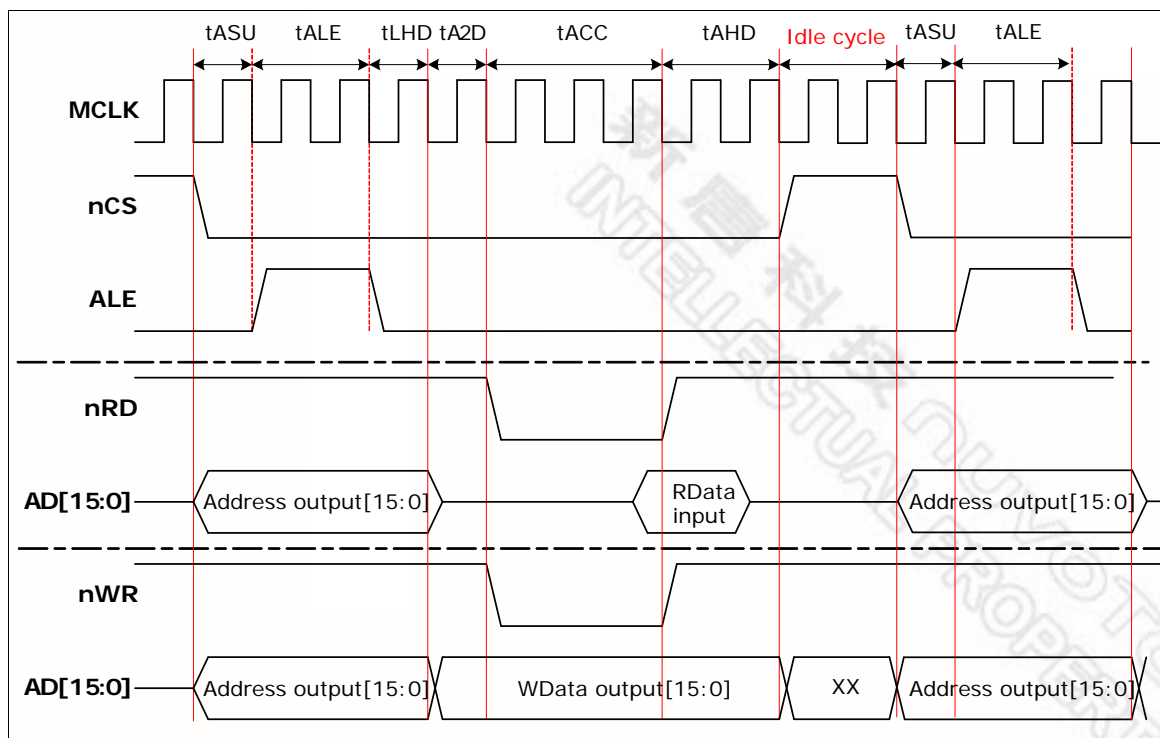


图 5-31 插入空闲周期的时序控制波形

有 3 个条件，EBI 可通过时序控制插入空闲周期：

- 写访问后
- 读访问后及下一个读访问之前
- 读访问后及下一个写访问之前

通过设置寄存器 EXTIME 的 ExtIW2X、ExtIR2R 和 ExtIR2W，空闲周期可被设定在 0~15 MCLK。

5.8.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
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

5.8.6 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	复位后的值
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
-							
23	22	21	20	19	18	17	16
-					ExttALE		
15	14	13	12	11	10	9	8
-				MCLKEN	MCLKDIV		
7	6	5	4	3	2	1	0
-						ExtBW16	ExtEN

Bits	描述	
[31:19]	-	保留
[18:16]	ExttALE	ALE的扩展时间 通过 ExttALE 控制 ALE 宽度 (tALE) 来锁存地址. $tALE = (ExttALE+1)*MCLK$
[15:12]	-	保留
[11]	MCLKEN	外部时钟使能 该位控制EBI 是否产生时钟到外部设备 如果外部设备是同步设备, 需要将该位置为高使能EBI产生时钟到外部设备 如果外部设备是异步设备, 推荐保持该位为低以节省耗电 0 = 禁止EBI产生时钟到外部设备 1 = 使能EBI 产生时钟到外部设备.

Bits	描述																			
[10:8]	MCLKDIV	<p>外部输出时钟分频器</p> <p>EBI输出时钟频率由 MCLKDIV 控制，见下表：</p> <table><tr><th>MCLKDIV</th><th>输出时钟 (MCLK)</th></tr><tr><td>0x0</td><td>HCLK/1</td></tr><tr><td>0x1</td><td>HCLK/2</td></tr><tr><td>0x2</td><td>HCLK/4</td></tr><tr><td>0x3</td><td>HCLK/8</td></tr><tr><td>0x4</td><td>HCLK/16</td></tr><tr><td>0x5</td><td>HCLK/32</td></tr><tr><td>0x6</td><td>Default</td></tr><tr><td>0x7</td><td>Default</td></tr></table> <p>注：输出时钟的默认值是HCLK/1</p>	MCLKDIV	输出时钟 (MCLK)	0x0	HCLK/1	0x1	HCLK/2	0x2	HCLK/4	0x3	HCLK/8	0x4	HCLK/16	0x5	HCLK/32	0x6	Default	0x7	Default
MCLKDIV	输出时钟 (MCLK)																			
0x0	HCLK/1																			
0x1	HCLK/2																			
0x2	HCLK/4																			
0x3	HCLK/8																			
0x4	HCLK/16																			
0x5	HCLK/32																			
0x6	Default																			
0x7	Default																			
[7:2]	-	保留																		
[1]	ExtBW16	<p>EBI 数据宽度 16-位</p> <p>该位定义数据总线是 8-位还是 16-位</p> <p>0 = EBI 数据宽度是 8-位</p> <p>1 = EBI 数据宽度是 16-位</p>																		
[0]	ExtEN	<p>EBI 使能</p> <p>该位 EBI 的功能使能位</p> <p>0 = 禁用 EBI 功能</p> <p>1 = 使能 EBI 功能</p>																		

外部总线接口时序控制寄存器(EXTIME)

寄存器	偏移量	R/W	描述	复位后的值
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
-				ExtIR2R			
23	22	21	20	19	18	17	16
-				ExtIR2W			
15	14	13	12	11	10	9	8
ExtIW2X				-	ExttAHD		
7	6	5	4	3	2	1	0
-				ExttACC			

Bits	描述	
[31:28]	-	保留
[27:24]	ExtIR2R	读-读 之间的空闲状态周期 当读动作完成，下一个动作也是读时，插入空闲状态，nCS 返回高（如果 ExtIR2R 非 0）。 空闲状态周期 = (ExtIR2R * MCLK)
[23:20]	-	保留
[19:16]	ExtIR2W	读-写 之间的空闲状态周期 当读动作完成，下一个动作是写时，插入空闲状态，nCS 返回高（如果 ExtIR2W 非 0）。 空闲状态周期 = (ExtIR2W * MCLK)
[15:12]	ExtIW2X	写之后的空闲状态周期 当写动作完成，插入空闲状态，nCS 返回高（如果 ExtIW2X 非 0）。 空闲状态周期 = (ExtIW2X * MCLK)
[11]	-	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). $tAHD = (ExttAHD + 1) * MCLK$
[7:5]	-	保留
[4:0]	ExttACC	EBI 数据访问时间 ExttACC 定义数据访问时间 (tACC). $tACC = (ExttACC + 1) * MCLK$

NuMicro™ Nano100 系列配置了 32K/64/128K 字节的片上 Flash，用于应用程序内存 (APROM)，用户可通过 ISP/IAP 更新该部分。当芯片焊接在 PCB 上时，在系统编程 (ISP) 功能允许用户更新程序内存。芯片上电后，Cortex-M0 CPU 依据 Config0 中的启动选择 (CBS) 从 APROM 或 LDROM 中读取代码。此外，NuMicro™ Nano100 系列还提供数据 Flash 区域，数据 flash 与原始的程序内存共享，其起始地址是可配置的，由用户在 Config1 中定义。数据 flash 大小根据用户的应用需要来定义。

- 兼容AHB 接口
- 零等待状态，最高可达42MHz的非连续的地址读访问
- 32/64/128KB应用程序内存(APROM)
- 4KB在系统编程 (ISP) 加载程序内存 (LDROM)
- 可配置开始地址的数据 flash， 512 字节页擦除单元
- 在系统编程 (ISP) 用于更新片上Flash EPROM

Flash 内存控制器包括 AHB 从接口, ISP 控制逻辑, 写接口和 flash 宏接口时序控制逻辑。Flash 内存控制器的框图见下图:

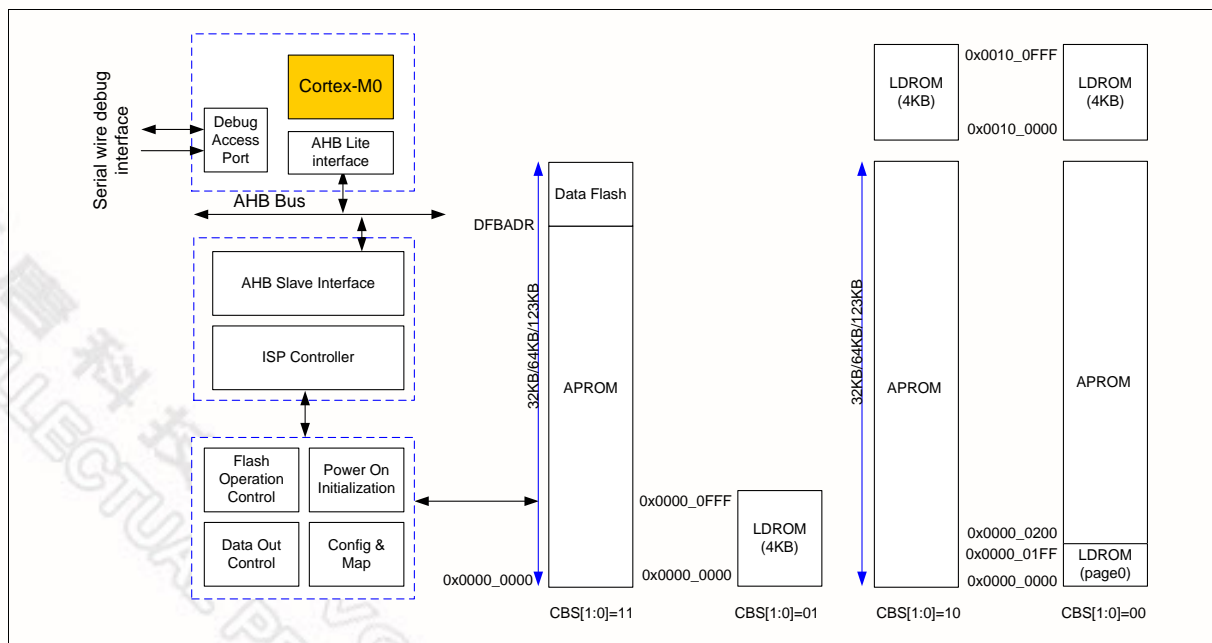


图 5-32 Flash 内存控制器框图

5.9.4 功能描述

5.9.4.1 Flash内存结构

FLASH 内存由程序内存 (64/32/128KB)，数据 FLASH，ISP 载入程序内存，及用户配置区域构成。用户配置区域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 基地址等。它就像通电设置中的熔丝。在芯片上电过程中，用户配置区域被从 FLASH 内存中载入到相应的控制寄存器。在芯片上 PCB 之前，用户可以用烧写器根据应用需求来设置这些位。数据 FLASH 的开始地址和大小，用户根据实际应用来定义。

区域名称	大小	开始地址	结束地址
APROM	(32-0.5*N)KB / (64-0.5*N)KB / (128-0.5*N)KB	0x0000_0000	DFBADR-1 (if DFEN=0)
数据Flash	0.5*N KB	DFBADR	0x0000_7FFF / 0x0000_FFFF / 0x0001_EBFF
保留作将来之用	901KB	0x0001_EC00	0x000F_FFFF
LDROM	4 KB	0x0010_0000	0x0010_0FFF
用户配置	2 words	0x0030_0000	0x0030_0004

表 5-8 内存地址映射

Part Number	NANO1XX-XCXB	NANO1XX-XDXBN	NANO1XX-XEXBN
	Flash ROM:32KB	Flash ROM:64KB	Flash ROM:128KB
APROM 大小	(32-0.5*N) KB	(64-0.5*N) KB	(128-0.5*N) KB
Data Flash 大小	(0.5*N) KB	(0.5*N) KB	(0.5*N) KB
LDROM	4 KB	4 KB	4 KB

表 5-9 Flash 大小

Flash 内存组织结构如下所示.

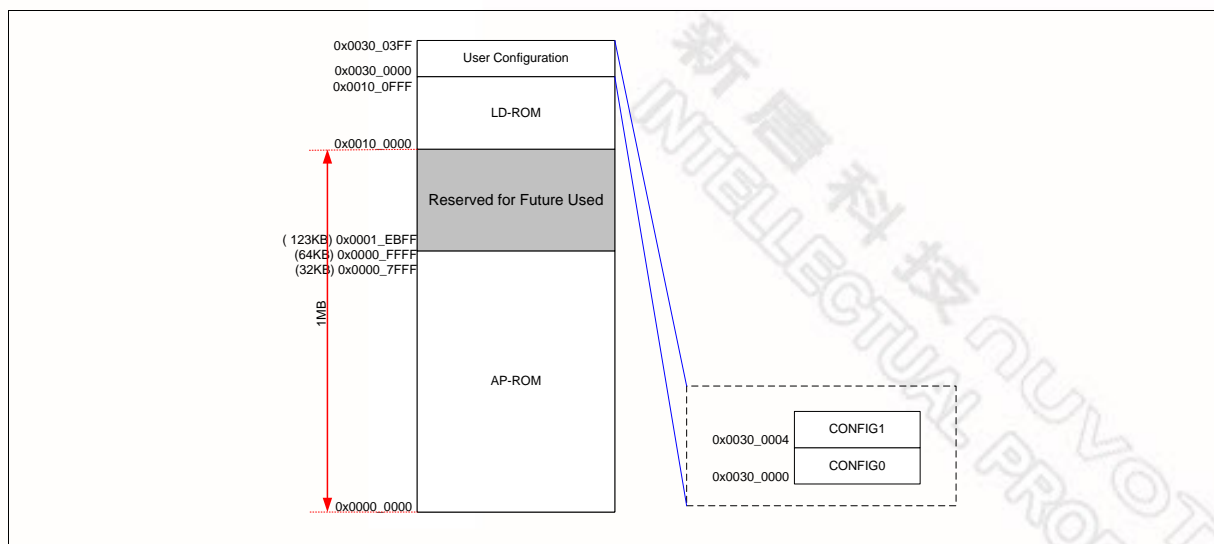


图 5-33 内存组织结构

5.9.4.2 启动选择

NuMicro™ Nano100 系列提供在系统编程 (ISP) 特征，允许用户直接更新 PCB 板上芯片中的程序。专用的 4KB 程序内存用于存储 ISP 固件。用户可以通过设置 Config0 中的 CBS 位来选择从 APROM 还是从 LDROM 取指令来开始程序

CBS[1:0]	启动选择
11	CPU 从 APROM启动， flash访问范围包括APROM 和数据Flash， LDROM不能直接被访问， 除非通过ISP。 在这个模式中APROM 是写保护的.
01	CPU 从LDROM启动， flash访问范围仅4KB LDROM， APROM不能直接被访问， 除非通过ISP。 在这个模式中APROM 可以被更新.
10	CPU 从 APROM启动， flash访问范围包括LDROM 和 APROM 在这个模式中APROM 可以被更新. LDROM 地址映射是0x0010_0000~0x0010_0FFF 通过ISP命令， 地址0x0000_0000 ~ 0x0000_01FF映射可以改变到 LDROM
00	CPU 从 LDROM启动， flash访问范围包括 LDROM 和大部分的 APROM (除了page0, 因为该地址映射到LDROM) 在这个模式中APROM 可以被更新. LDROM 地址映射到0x0010_0000 ~ 0x0010_0FFF， LDROM的前 512 字节映射到地址 0x0000_0000 ~ 0x0000_01FF. 通过ISP命令， 地址0x0000_0000 ~ 0x0000_01FF映射可以改变到 ADROM

表 5-10 启动选择

CBS[1:0]	启动自	向量 重新映射	运行在 LDROM 写 APROM	运行在 APROM 写 LDROM	运行在 LDROM 写 LDROM	运行在 APROM 写 APROM
11	APROM	-	-	Yes	-	-
01	LDROM	-	Yes	-	-	-
10	APROM	Yes	-	Yes	-	Yes
00	LDROM	Yes	Yes	-	Yes	Yes

表 5-11 启动选择和支持功能

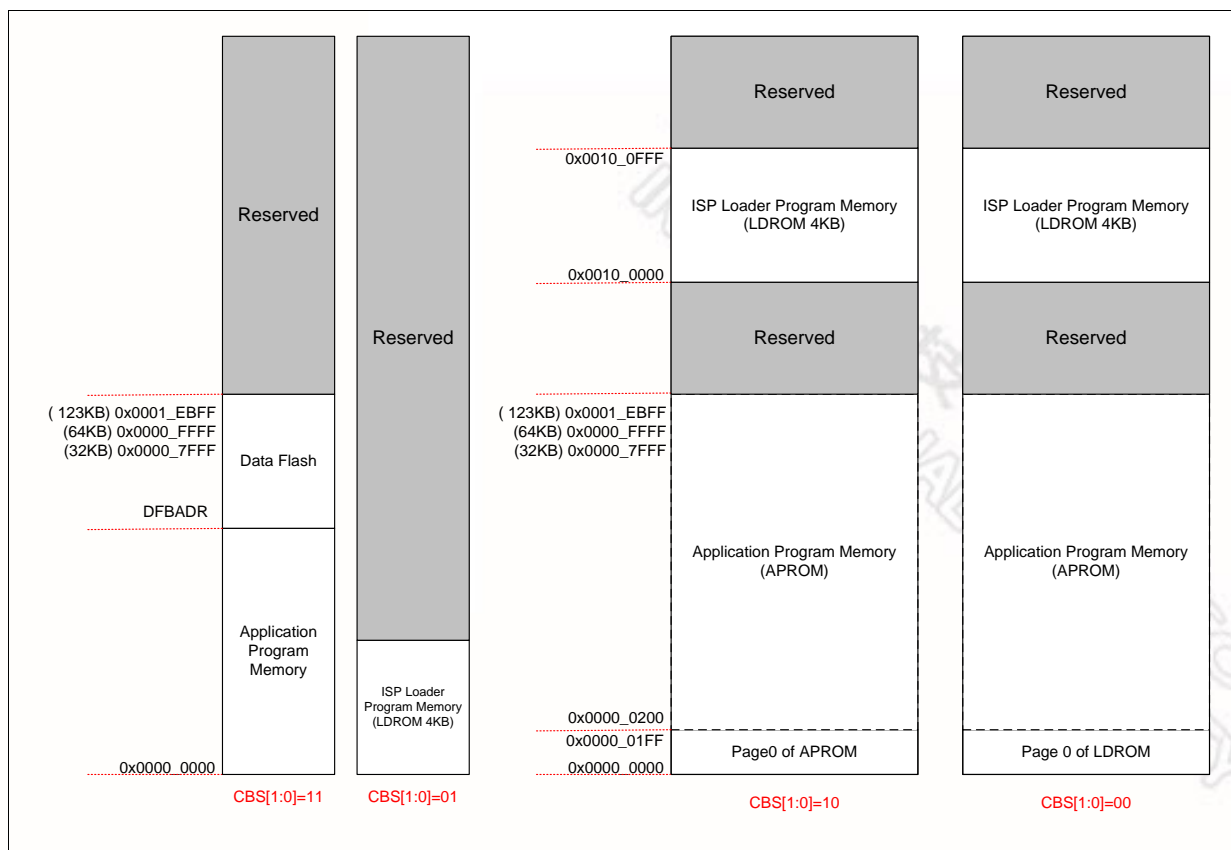


图 5-34 CONFIG0 中 CBS 的 Flash 内存映射

5.9.4.3 数据 Flash

NuMicro™ Nano100 系列为用户提供数据 FLASH 用于存储数据，通过 ISP 过程读/写。每个擦除单元的大小是 512 字节。当要改变一个字节时，需要事先将所有的128个字拷贝到其他页或 SRAM 中。若 Config0 的 DFEN 位使能，数据 FLASH 的基地址由 DFBADR 定义，对64KB flash而言，应用程序内存的大小为 $(64-0.5*N)KB$ ，对32KB flash而言，应用程序内存的大小为 $(32-0.5*N)KB$ 。数据 flash 大小为 $0.5*N KB$ 。

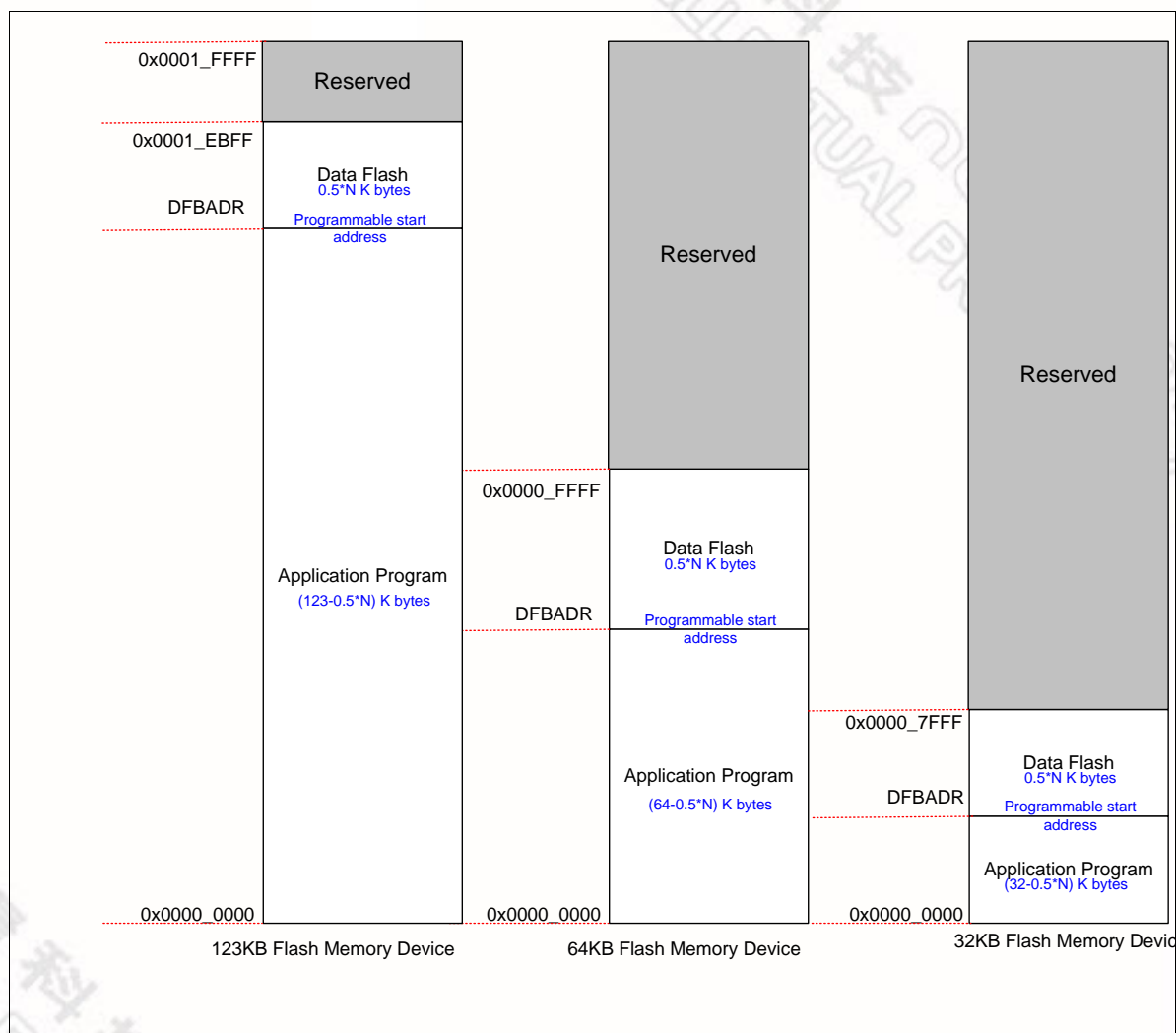


图 5-35 32/64/123KB 内存结构

5.9.4.4 用户配置

Config0 (Address = 0x0030_0000)

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

Bits	描述											
[31]	CWDT_EN	强迫看门狗时钟开启 0 = 强迫看门狗时钟开启，无法透过其他软件设定将此时钟关闭 1 = 不强迫看门狗时钟开启，可透过其他软件设定将此时钟关闭										
[30:27]	-	保留作将来之用										
[26]	CFOSC	复位后，CPU 时钟源选择 <table><tr><th>CFOSC</th><th>时钟源</th></tr><tr><td>0</td><td>外部 12 MHz高速晶振时钟(HXT)</td></tr><tr><td>1</td><td>内部RC 12 MHz高速振荡器时钟k(HIRC)</td></tr></table> 任何复位发生后，CFOSC 的值将被加载到系统寄存器 CLKSEL0.HCLK_S[2]。	CFOSC	时钟源	0	外部 12 MHz高速晶振时钟(HXT)	1	内部RC 12 MHz高速振荡器时钟k(HIRC)				
CFOSC	时钟源											
0	外部 12 MHz高速晶振时钟(HXT)											
1	内部RC 12 MHz高速振荡器时钟k(HIRC)											
[25:21]	-	保留作将来之用										
[20:19]	CBORST	欠压复位使能选择 <table><tr><th>CBORST[1:0]</th><th>欠压复位选择</th></tr><tr><td>00</td><td>BOD17 复位使能</td></tr><tr><td>01</td><td>BOD20复位使能</td></tr><tr><td>10</td><td>BOD25复位使能</td></tr><tr><td>11</td><td>禁止所有BOD 功能</td></tr></table>	CBORST[1:0]	欠压复位选择	00	BOD17 复位使能	01	BOD20复位使能	10	BOD25复位使能	11	禁止所有BOD 功能
CBORST[1:0]	欠压复位选择											
00	BOD17 复位使能											
01	BOD20复位使能											
10	BOD25复位使能											
11	禁止所有BOD 功能											
[18:8]	-	保留作将来之用										

[7:6]	CBS	CONFIG启动选择	
		CBS[1:0]	启动选择
		11	CPU 从 APROM启动, flash访问范围包括APROM 和数据 Flash, LDROM不能被访问, 除非通过ISP。 在这个模式中APROM 是写保护的。
		01	CPU 从LDROM启动, flash访问范围仅4KB LDROM, APROM 不能被访问, 除非通过ISP。 在这个模式中APROM 可以被更新。
		10	CPU 从 APROM 启动, flash 访问范围包括 LDROM 和 APROM 在这个模式中APROM 可以被更新。 LDROM 地址映射是0x0010_0000~0x0010_0FFF 通过ISP命令, 地址0x0000_0000 ~ 0x0000_01FF映射可以改变到 LDROM
		00	CPU 从 LDROM启动, flash访问范围包括 LDROM 和大部分的 APROM (除了page0, 因为该地址映射到LDROM) 在这个模式中APROM 可以被更新。 LDROM 地址映射到 0x0010_0000 ~ 0x0010_0FFF , LDROM 的前 512 字节映射到地址 0x0000_0000 ~ 0x0000_01FF。 通过ISP命令, 地址0x0000_0000 ~ 0x0000_01FF映射可以改变到 ADROM
[5:2]	-	保留作将来之用	
[1]	LOCK	安全加密 0 = 加密 Flash 数据 1 = 解除 Flash 数据加密。 当 flash 数据被加密, 只有设备 ID, Config0 和 Config1 可被烧写器和 ICP 通过串口调试接口读取。其他数据锁定为 0xFFFFFFFF。无论数据是否锁定, ISP 都可以读取数据	
[0]	DFEN	数据 Flash 使能 0 = 使能数据 flash 1 = 禁用数据 flash	

Config1 (Address = 0x0030 0004)

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

Bits	描述	
[31:20]	-	保留 强行给这些保留位写入0x00
[19:0]	DFBA	数据 Flash 基地址 其数据 flash 基地址由用户定义。因为片上 flash 擦除单元为 512 字节，所以强制保持 bit 8-0 为0。

5.9.4.5 在系统编程 (ISP)

程序内存和数据 FLASH 支持硬件编程和在系统编程 (ISP)。硬件编程模式采用 gang-烧写器，以方便量产阶段降低编程成本和时间。但是若产品还在开发阶段或终端用户需要升级固件时，硬件编程模式不是很方便，ISP 模式能更好地适用于这种情况。NuMicro™ Nano100 系列支持 ISP 模式，即通过软件控制来对设备重新编程。而且，这也使得更新固件得以广泛应用。

ISP 可以在没有将微控制器从系统中取下来的情况下执行编程。各种接口使得 LDROM 更容易更新程序代码。最常用的方法是通过 UART 和在 LDROM 中的固件执行 ISP，一般来说，PC 都是通过串口传输新的 APROM 代码。LDROM 固件接收后，通过 ISP 命令重新对 APROM 编程。Nuvoton 提供用于 NuMicro™ Nano100 系列的 ISP 固件和 PC 应用程序。这使得用户可以通过 Nuvoton ISP 工具非常容易的执行 ISP。

ISP 步骤

NuMicro™ Nano100 系列支持根据初始定义的用户配置位 (CBS) 从 APROM 或 LDROM 启动。如果用户想更新 APROM 中的应用程序，可以写 BS=1，软件复位使芯片从 LDROM 启动。ISP 功能的第一步是向 ISPEN 位写 1。在写 ISPCON 寄存器之前，S/W 需要向全局控制寄存器 (GCR, 0x5000_0100) 中的寄存器 RegLockAddr 写 0x59, 0x16 和 0x88。这个过程用于保护 FLASH 内存免受在上电或断电期间无意识的写操作而造成的损坏。

在软件写 ISPGO 位后，要检查几个错误条件。如果错误条件产生，ISP 操作失败，ISP 失败标志置位，ISPFF 标志由 S/W 清零，而不会在下次 ISP 操作时被覆盖，即使 ISPFF 保持为“1”，下一次 ISP 也可以开始。建议在每次 ISP 操作后，由软件检查 ISPFF 位并将其清零。

当 ISPGO 位置位，CPU 将等待 ISP 操作结束。在此期间，外设仍然像往常一样保持工作。如果任

何中断请求发生，CPU 将不会响应中断直到 ISP 执行完成。

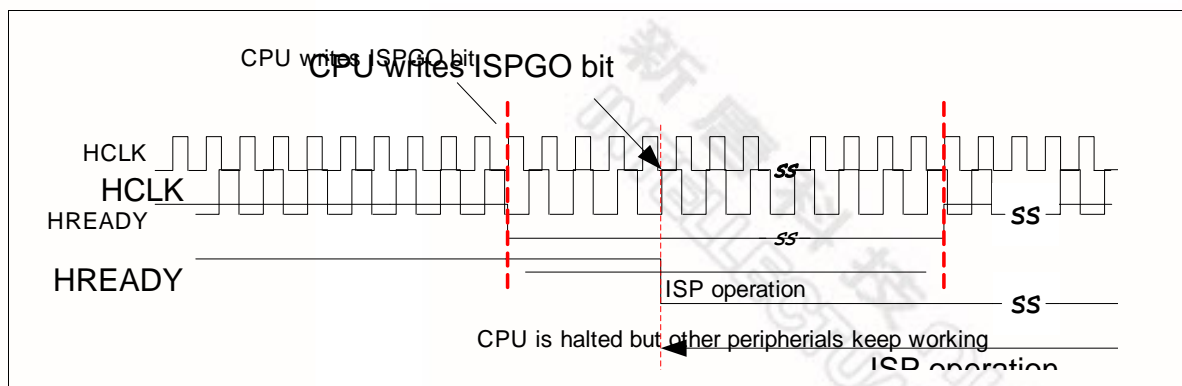


图 5-36 当 ISP 操作时，CPU 暂停工作

注：NuMicro™ Nano100 系列允许用户通过ISP更新CONFIG 值， 但是为了应用程序的安全， s/w 需要在擦除CONFIG之前页擦除 APROM， 否则擦除CONFIG 是不允许的。

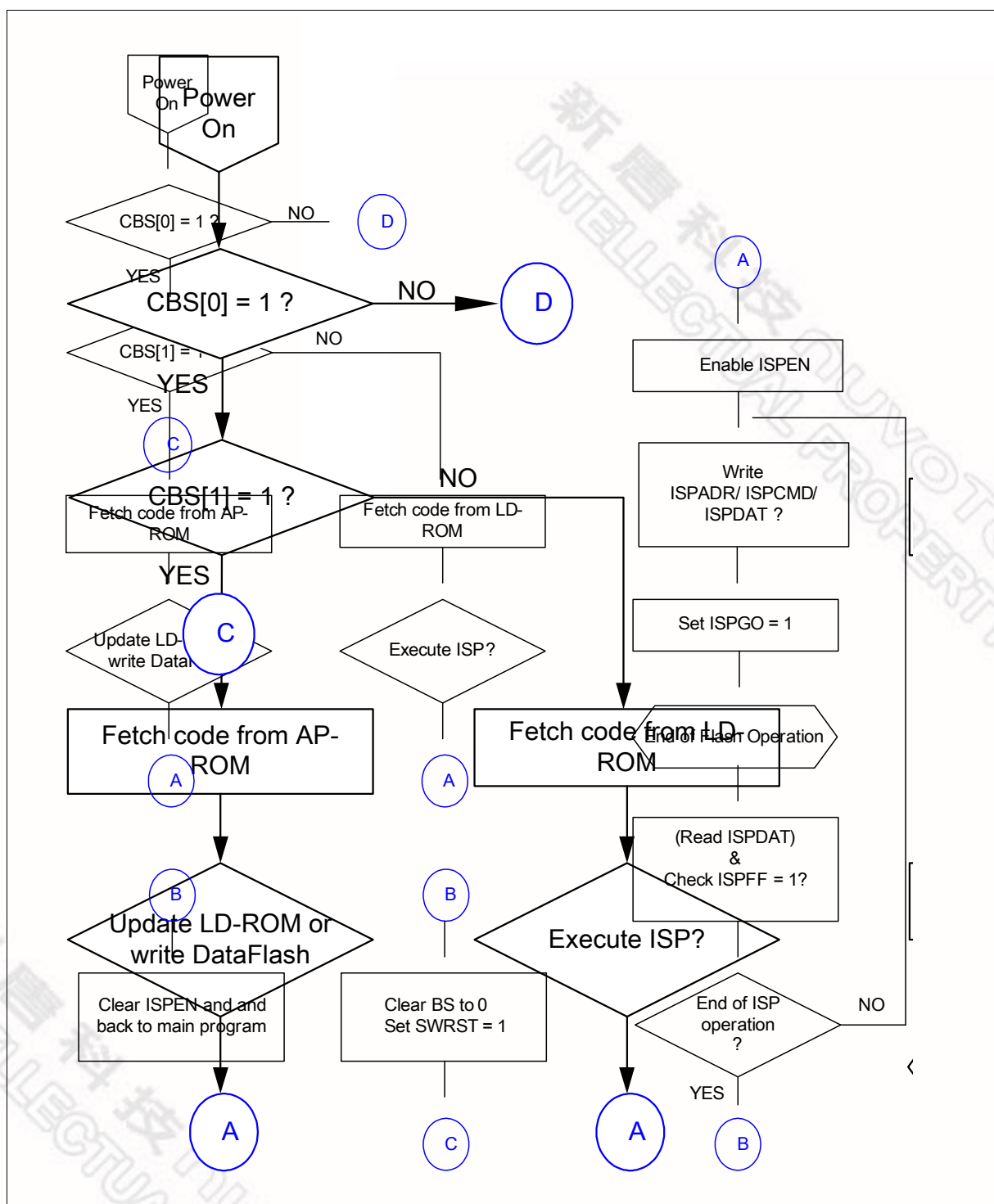


图 5-38 ISP 操作流程(继续)

ISP模式	ISPCMD			ISPADR			ISPDAT
	FOEN	FCEN	FCTRL[3:0]	A21	A20	A[19:0]	D[31:0]
Standby	1	1	X	x	x	x	x
Read Company ID	0	0	1011	x	x	x	Data out D[31:0] = 0x0000_00DA
Read Device ID	0	0	1100	x	x	Address in A[19:0] = 0x00000	Data out D[31:0]= Device ID
Read Unique ID	0	0	0100	x	x	Address in A[19:0] = 0x00000 0x00004 0x00008	Data out D[31:0]= Unique ID
Read Unique Customer ID	0	0	0100	x	x	Address in A[19:0] = 0x00010 0x00014 0x00018 0x0001C	Data out D[31:0]= Unique Customer ID
Vector Page Re-Map	1	0	1110	0	A20	Address in A[19:0]	x
FLASH Page Erase	1	0	0010	0	A20	Address in A[19:0]	x
FLASH Program	1	0	0001	0	A20	Address in A[19:0]	Data in D[31:0]
FLASH Read	0	0	0000	0	A20	Address in A[19:0]	Data out D[31:0]
CONFIG Page Erase	1	0	0010	1	1	Address in A[19:0]	x
CONFIG Program	1	0	0001	1	1	Address in A[19:0]	Data in D[31:0]
CONFIG Read	0	0	0000	1	1	Address in A[19:0]	Data out D[31:0]

表 5-12 ISP 操作命令

* UCID之地址自0x00010至0x0001C，其默认值为0xFFFF。UCID仅由新唐科技提供，关于植入此特定客户标识符，请联系新唐科技或相关代理商。

5.9.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
Base Address (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
ISPSTA	FMC_BA+0x40	R/W	ISP 状态寄存器	0x0000_0000

5.9.6 寄存器描述

ISP控制寄存器(ISPCON)

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

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

Bits	描述
[31:7]	- 保留

[6]	ISPFF	ISP失败标志 （写保护位） 当 ISP 满足下列条件时，该位由硬件置位： (1) APROM 写入本身 (2) LDROM 写入本身 (3) 如果 CFGUEN 设置为 0，CONFIG 被擦除/编程 (4) 目的地址无效，比如超过正常范围 写 1 清除该位。
[5]	LDUEN	LDROM更新使能 （写保护位） LDROM 更新使能位 1 = 当MCU运行在APROM，LDROM 可以被更新。 0 = LDROM不能被更新
[4]	CFGUEN	使能 Config-bits 更新通过 ISP （写保护位） 1 = 使能 ISP 更新config-bits 0 = 禁止 ISP 更新 config-bits
[3]	APUEN	APROM 更新使能 （写保护位） APROM更新使能位。 1 = 当MCU运行在APROM，APROM 可以被更新。 0 = APROM 不能被更新
[2]	-	保留
[1]	BS	启动选择 （写保护位） 置位/清零该位选择下次是由 LDROM 启动还是由 APROM 启动，该位可作为MCU启动状态标志，用于检查芯片是由 LDROM 还是 APROM 启动。这一位在上电复位发生时被初始化为 Config0 的 CBS 位的反转值，在其他复位时保持不变 1 = 从 LDROM 启动 0 = 从 APROM 启动
[0]	ISPEN	ISP 使能 （写保护位） ISP 功能使能位。设置该位使能 ISP 功能。 1 = 使能 ISP 功能 0 = 禁用 ISP 功能

ISP地址(ISPADR)

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

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

Bits	描述	
[31:0]	ISPADR	ISP 地址 该芯片仅支持字编程。执行 ISP 操作时，ISPADR[1:0] 必须保持 00b。向量页重新映射命令时，ISPADR[8:0] 必须保持 0_0000_0000b

ISPDAT (ISP 数据寄存器)

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

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

Bits	描述	
[31:0]	ISPDAT	<p>ISP 数据</p> <p>ISP 操作之前，写数据到该寄存器</p> <p>ISP 读操作后，可从该寄存器读数据</p>

ISP命令(ISPCMD)

寄存器	偏移量	R/W	描述	复位后的值
ISPCMD	FMC_BA+0x0C	R/W	ISP命令寄存器	0x0000_0000

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

Bits	描述						
[31:6]	-	保留					
[5]	FOEN	ISP 命令 ISP 命令表格如下.					
[4]	FCEN						
[3:0]	FCTRL	操作模式	FOEN	FCEN	FCTRL[3:0]		
		读	0	0	0	0	0
		相连页重映射	1	0	1	1	0
		写	1	0	0	0	1
		页擦	1	0	0	0	1
[3:0]	FCTRL	读 CID	0	0	1	0	1
		读 DID	0	0	1	1	0
		读 UID	0	0	0	1	0

ISP触发控制寄存器(ISPTRG)

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

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

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

数据 Flash 基地址寄存器(DFBADR)

寄存器	偏移量	R/W	描述	复位后的值
DFBADR	FMC_BA+0x14	R	数据Flash 基地址	0x0001_F000

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

Bits	描述	
[31:0]	DFBA	<p>数据 Flash 基地址</p> <p>该寄存器为数据 FLASH 开始地址寄存器。该寄存器只读。</p> <p>数据flash开始地址由用户定义，因为片上flash的擦除单元是 512 字节，所以它的 bit 8-0固定为 0。</p>

ISP状态寄存器(ISPSTA)

寄存器	偏移量	R/W	描述	复位后的值
ISPSTA	FMC_BA+0x40	R/W	ISP状态寄存器	0x0000_0000

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

Bits	描述	
[31:21]	-	保留
[20:9]	VECMAP	向量页映射地址 当前 flash 地址空间 0x0000_0000~0x0000_01FF 映射到 {VECMAP[11:0], "000000000b"} ~ {VECMAP[11:0], "111111111b"} 只读
[8:7]	-	保留
[6]	ISPFF	.ISP失败标志 当 ISP 满足下列条件时，该位由硬件置位： (1) APROM 写入本身 (2) LDROM 写入本身 (3) 如果 CFGUEN 设置为 0，CONFIG 被擦除/编程 (4) 目的地址无效，比如超过正常范围 写 1 清除该位。
[5:3]	-	保留

[2:1]	CBS	Config启动选择状态	
		CBS[1:0]	启动选择
		01	芯片从 LDROM;启动, APROM 不可读
		1	芯片从 APROM启动, ;LDROM 不可读
		00	芯片从 LDROM第0页启动; LDROM 和 APROM都可读
		10	芯片从 ADROM第0页启动; LDROM 和 APROM都可读
[0]	ISPBUSY	ISP BUSY	
		1 = ISP 操作忙	
		0 = ISP 操作完成	
		只读	

5.10 通用 I/O 控制器

5.10.1 概述

最大 86 通用 I/O 管脚, 这些管脚可以和其他功能管脚共享, 这取决于芯片的配置。这86个管脚分配在GPIOA、GPIOB、GPIOC、GPIOD、GPIOE 与 GPIOF六个端口上。Port A ~ E每个端口最多16个管脚, Port F 六个管脚。每个管脚都是独立的, 都有相应的寄存器位来控制管脚功能模式与数据。.

I/O 管脚上的 I/O 类型可由软件独立地配置为输入, 输出, 开漏模式。每个 I/O 管脚有一个阻值110 K Ω ~300 K Ω 的弱上拉电阻接到 VDD上, VDD的范围1.8 V到3.6 V.

5.10.2 特征

- 最多到 86 通用I/O 管脚
- 支持输入, 输出, 开漏模式
- 可编程的消抖时间.
- 每个I/O 管脚可以设置成边沿或电平触发中断.
- 每个I/O管脚可以设置成低电平有效或高电平触发中断
- 每个I/O 管脚可以设置成下降沿触发或上升沿触发中断

5.10.3 框图

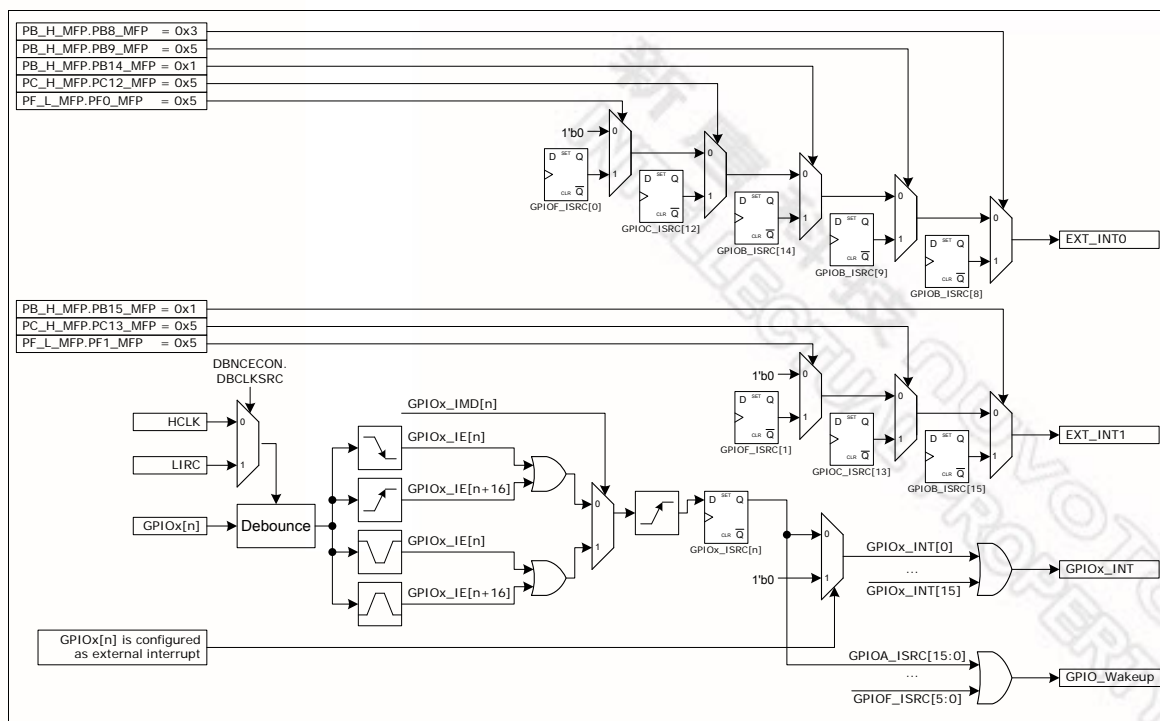


图 5-38 GPIO 框图

5.10.4 功能描述

5.10.4.1 输入模式说明

设置 GPIOx_PMD (PMDn[1:0]) 为 00b, GPIOx port [n] 为输入模式, 则 I/O 管脚为三态 (高阻), 没有输出驱动能力。GPIOx_PIN 的值反映相应端口的状态。

5.10.4.2 输出模式说明

设置 GPIOx_PMD (PMDn[1:0]) 为 01b, GPIOx port [n] 为输出模式, 则 I/O 管脚支持数字输出功能, 有拉/灌电流能力。GPIOx_DOUT 相应位的值被驱动到相应管脚上。

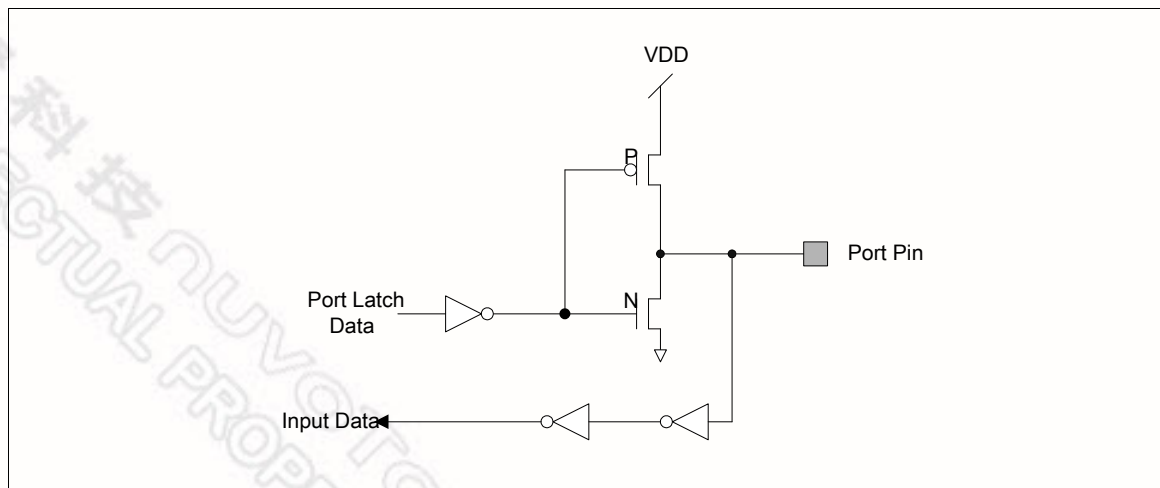


图 5-39推挽输出

5.10.4.3 开漏模式说明

设置 GPIOx_PMD (PMDn[1:0]) 为 10b, GPIOx port [n] 为开漏模式, 则 I/O 管脚数字输出功能仅支持灌电流, 且需要外加一个上拉电阻用于驱动到高电平。如果 GPIOx_DOUT 相应位 bit[n] 为 '0', 管脚上输出低。如果 GPIOx_DOUT 相应位 bit[n] 为 '1', 该管脚则输出为高, 可以由外部上拉电阻控制。

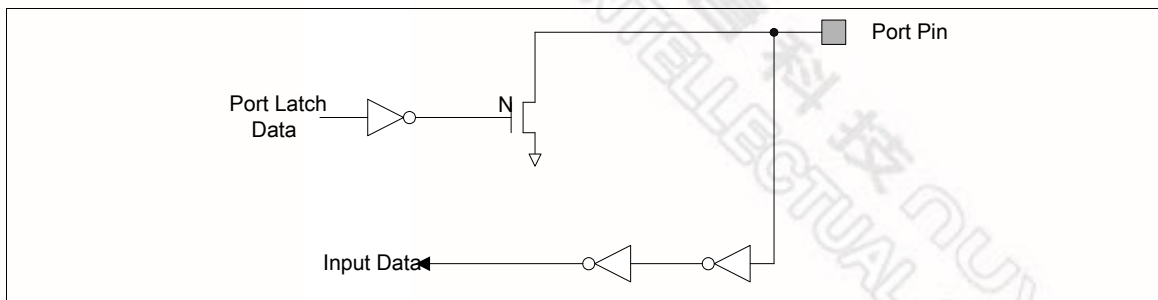


图 5-40 开漏输出

5.10.5 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
GPIO_BA (GP_BA) = 0x5000_4000				
GPIOA_PMD	GP_BA+0x000	R/W	GPIO端口 A 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO端口 A 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO端口 A 数据输出值寄存器	0x0000_FFFF
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO端口 A 数据输出写屏蔽寄存器	0x0000_0000
GPIOA_PIN	GP_BA+0x010	R	GPIO端口 A 管脚数值寄存器	0x0000_XXXX
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO端口 A 去抖动使能	0x0000_0000
GPIOA_IMD	GP_BA+0x018	R/W	GPIO端口 A 中断模式控制寄存器	0x0000_0000
GPIOA_IER	GP_BA+0x01C	R/W	GPIO端口 A 中断使能寄存器	0x0000_0000
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO端口 A 中断源状态寄存器	0xFFFF_XXXX
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO 端口 A 上拉使能寄存器	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO端口 B 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO端口 B 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO端口 B 数据输出值寄存器	0x0000_FFFF
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO端口 B 数据输出写屏蔽寄存器	0x0000_0000
GPIOB_PIN	GP_BA+0x050	R	GPIO端口 B 管脚数值寄存器	0x0000_XXXX
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO端口 B 去抖动使能	0x0000_0000

GPIOB_IMD	GP_BA+0x058	R/W	GPIO端口 B 中断模式控制寄存器	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO端口 B 中断使能寄存器	0x0000_0000
GPIOB_ISRC	GP_BA+0x060	R/W	GPIO端口 B 中断源状态寄存器	0xFFFF_FFFF
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO 端口 B 上拉使能寄存器	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO端口 C 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO端口 C 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO端口 C 数据输出值寄存器	0x0000_FFFF
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO端口 C 数据输出写屏蔽寄存器	0x0000_0000
GPIOC_PIN	GP_BA+0x090	R	GPIO端口 C 管脚数值寄存器	0x0000_XXXX
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO端口 C 去抖动使能	0x0000_0000
GPIOC_IMD	GP_BA+0x098	R/W	GPIO端口 C 中断模式控制寄存器	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO端口 C 中断使能寄存器	0x0000_0000
GPIOC_ISRC	GP_BA+0x0A0	R/W	GPIO端口 C 中断源状态寄存器	0xFFFF_FFFF
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO 端口 C 上拉使能寄存器	0x0000_0000
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO端口 D 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOD_OFFD	GP_BA+0x0C4	R/W	GPIO端口 D 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO端口 D 数据输出值寄存器	0x0000_FFFF
GPIOD_DMASK	GP_BA+0x0CC	R/W	GPIO端口 D 数据输出写屏蔽寄存器	0x0000_0000
GPIOD_PIN	GP_BA+0x0D0	R	GPIO端口 D 管脚数值寄存器	0x0000_XXXX
GPIOD_DBEN	GP_BA+0x0D4	R/W	GPIO端口 D 去抖动使能	0x0000_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO端口 D 中断模式控制寄存器	0x0000_0000
GPIOD_IER	GP_BA+0x0DC	R/W	GPIO端口 D 中断使能寄存器	0x0000_0000
GPIOD_ISRC	GP_BA+0x0E0	R/W	GPIO端口 D 中断源状态寄存器	0xFFFF_FFFF
GPIOD_PUEN	GP_BA+0x0E4	R/W	GPIO 端口 D 上拉使能寄存器	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO端口 E 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO端口 E 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO端口 E 数据输出值寄存器	0x0000_FFFF
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO端口 E 数据输出写屏蔽寄存器	0x0000_0000
GPIOE_PIN	GP_BA+0x110	R	GPIO端口 E 管脚数值寄存器	0x0000_XXXX
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO端口 E 去抖动使能	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO端口 E 中断模式控制寄存器	0x0000_0000

GPIOE_IER	GP_BA+0x11C	R/W	GPIO端口 E 中断使能寄存器	0x0000_0000
GPIOE_ISRC	GP_BA+0x120	R/W	GPIO端口 E 中断源状态寄存器	0xFFFF_XXXX
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO 端口 E 上拉使能寄存器	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO端口 F 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO端口 F 管脚 OFF 数字使能寄存器	0x0000_0000
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO端口 F 数据输出值寄存器	0x0000_003F
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO端口 F 数据输出写屏蔽寄存器	0x0000_0000
GPIOF_PIN	GP_BA+0x150	R	GPIO端口 F 管脚数值寄存器	0x0000_00XX
GPIOF_DBEN	GP_BA+0x154	R/W	GPIO端口 F 去抖动使能	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO端口 F 中断模式控制寄存器	0x0000_0000
GPIOF_IER	GP_BA+0x15C	R/W	GPIO端口 F 中断使能寄存器	0x0000_0000
GPIOF_ISRC	GP_BA+0x160	R/W	GPIO端口 F 中断源状态寄存器	0xFFFF_XXXX
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO 端口 F 上拉使能寄存器	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	去抖动循环控制寄存器	0x0000_0000
GPIOA0	GP_BA+0x200	R/W	GPIO端口 A Bit 0 数据寄存器	0x0000_000X
GPIOA1	GP_BA+0x204	R/W	GPIO端口 A Bit 1 数据寄存器	0x0000_000X
GPIOA2	GP_BA+0x208	R/W	GPIO端口 A Bit 2 数据寄存器	0x0000_000X
GPIOA3	GP_BA+0x20C	R/W	GPIO端口 A Bit 3 数据寄存器	0x0000_000X
GPIOA4	GP_BA+0x210	R/W	GPIO端口 A Bit 4 数据寄存器	0x0000_000X
GPIOA5	GP_BA+0x214	R/W	GPIO端口 A Bit 5 数据寄存器	0x0000_000X
GPIOA6	GP_BA+0x218	R/W	GPIO端口 A Bit 6 数据寄存器	0x0000_000X
GPIOA7	GP_BA+0x21C	R/W	GPIO端口 A Bit 7 数据寄存器	0x0000_000X
GPIOA8	GP_BA+0x220	R/W	GPIO端口 A Bit 8 数据寄存器	0x0000_000X
GPIOA9	GP_BA+0x224	R/W	GPIO端口 A Bit 9 数据寄存器	0x0000_000X
GPIOA10	GP_BA+0x228	R/W	GPIO端口 A Bit 10 数据寄存器	0x0000_000X
GPIOA11	GP_BA+0x22C	R/W	GPIO端口 A Bit 11 数据寄存器	0x0000_000X
GPIOA12	GP_BA+0x230	R/W	GPIO端口 A Bit 12 数据寄存器	0x0000_000X
GPIOA13	GP_BA+0x234	R/W	GPIO端口 A Bit 13 数据寄存器	0x0000_000X
GPIOA14	GP_BA+0x238	R/W	GPIO端口 A Bit 14 数据寄存器	0x0000_000X
GPIOA15	GP_BA+0x23C	R/W	GPIO端口 A Bit 15 数据寄存器	0x0000_000X
GPIOB0	GP_BA+0x240	R/W	GPIO端口 B Bit 0 数据寄存器	0x0000_000X

GPIOB1	GP_BA+0x244	R/W	GPIO端口B Bit 1 数据寄存器	0x0000_000X
GPIOB2	GP_BA+0x248	R/W	GPIO端口B Bit 2 数据寄存器	0x0000_000X
GPIOB3	GP_BA+0x24C	R/W	GPIO端口B Bit 3 数据寄存器	0x0000_000X
GPIOB4	GP_BA+0x250	R/W	GPIO端口B Bit 4 数据寄存器	0x0000_000X
GPIOB5	GP_BA+0x254	R/W	GPIO端口B Bit 5 数据寄存器	0x0000_000X
GPIOB6	GP_BA+0x258	R/W	GPIO端口B Bit 6 数据寄存器	0x0000_000X
GPIOB7	GP_BA+0x25C	R/W	GPIO端口B Bit 7 数据寄存器	0x0000_000X
GPIOB8	GP_BA+0x260	R/W	GPIO端口B Bit 8 数据寄存器	0x0000_000X
GPIOB9	GP_BA+0x264	R/W	GPIO端口B Bit 9 数据寄存器	0x0000_000X
GPIOB10	GP_BA+0x268	R/W	GPIO端口B Bit 10 数据寄存器	0x0000_000X
GPIOB11	GP_BA+0x26C	R/W	GPIO端口B Bit 11 数据寄存器	0x0000_000X
GPIOB12	GP_BA+0x270	R/W	GPIO端口B Bit 12 数据寄存器	0x0000_000X
GPIOB13	GP_BA+0x274	R/W	GPIO端口B Bit 13 数据寄存器	0x0000_000X
GPIOB14	GP_BA+0x278	R/W	GPIO端口B Bit 14 数据寄存器	0x0000_000X
GPIOB15	GP_BA+0x27C	R/W	GPIO端口B Bit 15 数据寄存器	0x0000_000X
GPIOC0	GP_BA+0x280	R/W	GPIO端口C Bit 0 数据寄存器	0x0000_000X
GPIOC1	GP_BA+0x284	R/W	GPIO端口C Bit 1 数据寄存器	0x0000_000X
GPIOC2	GP_BA+0x288	R/W	GPIO端口C Bit 2 数据寄存器	0x0000_000X
GPIOC3	GP_BA+0x28C	R/W	GPIO端口C Bit 3 数据寄存器	0x0000_000X
GPIOC4	GP_BA+0x290	R/W	GPIO端口C Bit 4 数据寄存器	0x0000_000X
GPIOC5	GP_BA+0x294	R/W	GPIO端口C Bit 5 数据寄存器	0x0000_000X
GPIOC6	GP_BA+0x298	R/W	GPIO端口C Bit 6 数据寄存器	0x0000_000X
GPIOC7	GP_BA+0x29C	R/W	GPIO端口C Bit 7 数据寄存器	0x0000_000X
GPIOC8	GP_BA+0x2A0	R/W	GPIO端口C Bit 8 数据寄存器	0x0000_000X
GPIOC9	GP_BA+0x2A4	R/W	GPIO端口C Bit 9 数据寄存器	0x0000_000X
GPIOC10	GP_BA+0x2A8	R/W	GPIO端口C Bit 10 数据寄存器	0x0000_000X
GPIOC11	GP_BA+0x2AC	R/W	GPIO端口C Bit 11 数据寄存器	0x0000_000X
GPIOC12	GP_BA+0x2B0	R/W	GPIO端口C Bit 12 数据寄存器	0x0000_000X
GPIOC13	GP_BA+0x2B4	R/W	GPIO端口C Bit 13 数据寄存器	0x0000_000X
GPIOC14	GP_BA+0x2B8	R/W	GPIO端口C Bit 14 数据寄存器	0x0000_000X
GPIOC15	GP_BA+0x2BC	R/W	GPIO端口C Bit 15 数据寄存器	0x0000_000X

GPIOD0	GP_BA+0x2C0	R/W	GPIO端口D Bit 0 数据寄存器	0x0000_000X
GPIOD1	GP_BA+0x2C4	R/W	GPIO端口D Bit 1 数据寄存器	0x0000_000X
GPIOD2	GP_BA+0x2C8	R/W	GPIO端口D Bit 2 数据寄存器	0x0000_000X
GPIOD3	GP_BA+0x2CC	R/W	GPIO端口D Bit 3 数据寄存器	0x0000_000X
GPIOD4	GP_BA+0x2D0	R/W	GPIO端口D Bit 4 数据寄存器	0x0000_000X
GPIOD5	GP_BA+0x2D4	R/W	GPIO端口D Bit 5 数据寄存器	0x0000_000X
GPIOD6	GP_BA+0x2D8	R/W	GPIO端口D Bit 6 数据寄存器	0x0000_000X
GPIOD7	GP_BA+0x2DC	R/W	GPIO端口D Bit 7 数据寄存器	0x0000_000X
GPIOD8	GP_BA+0x2E0	R/W	GPIO端口D Bit 8 数据寄存器	0x0000_000X
GPIOD9	GP_BA+0x2E4	R/W	GPIO端口D Bit 9 数据寄存器	0x0000_000X
GPIOD10	GP_BA+0x2E8	R/W	GPIO端口D Bit 10 数据寄存器	0x0000_000X
GPIOD11	GP_BA+0x2EC	R/W	GPIO端口D Bit 11 数据寄存器	0x0000_000X
GPIOD12	GP_BA+0x2F0	R/W	GPIO端口D Bit 12 数据寄存器	0x0000_000X
GPIOD13	GP_BA+0x2F4	R/W	GPIO端口D Bit 13 数据寄存器	0x0000_000X
GPIOD14	GP_BA+0x2F8	R/W	GPIO端口D Bit 14 数据寄存器	0x0000_000X
GPIOD15	GP_BA+0x2FC	R/W	GPIO端口D Bit 15 数据寄存器	0x0000_000X
GPIOE0	GP_BA+0x300	R/W	GPIO端口E Bit 0 数据寄存器	0x0000_000X
GPIOE1	GP_BA+0x304	R/W	GPIO端口E Bit 1 数据寄存器	0x0000_000X
GPIOE2	GP_BA+0x308	R/W	GPIO端口E Bit 2 数据寄存器	0x0000_000X
GPIOE3	GP_BA+0x30C	R/W	GPIO端口E Bit 3 数据寄存器	0x0000_000X
GPIOE4	GP_BA+0x310	R/W	GPIO端口E Bit 4 数据寄存器	0x0000_000X
GPIOE5	GP_BA+0x314	R/W	GPIO端口E Bit 5 数据寄存器	0x0000_000X
GPIOE6	GP_BA+0x318	R/W	GPIO端口E Bit 6 数据寄存器	0x0000_000X
GPIOE7	GP_BA+0x31C	R/W	GPIO端口E Bit 7 数据寄存器	0x0000_000X
GPIOE8	GP_BA+0x320	R/W	GPIO端口E Bit 8 数据寄存器	0x0000_000X
GPIOE9	GP_BA+0x324	R/W	GPIO端口E Bit 9 数据寄存器	0x0000_000X
GPIOE10	GP_BA+0x328	R/W	GPIO端口E Bit 10 数据寄存器	0x0000_000X
GPIOE11	GP_BA+0x32C	R/W	GPIO端口E Bit 11 数据寄存器	0x0000_000X
GPIOE12	GP_BA+0x330	R/W	GPIO端口E Bit 12 数据寄存器	0x0000_000X
GPIOE13	GP_BA+0x334	R/W	GPIO端口E Bit 13 数据寄存器	0x0000_000X
GPIOE14	GP_BA+0x338	R/W	GPIO端口E Bit 14 数据寄存器	0x0000_000X

GPIOE15	GP_BA+0x33C	R/W	GPIO端口E Bit 15 数据寄存器	0x0000_000X
GPIOF0	GP_BA+0x340	R/W	GPIO端口F Bit 0 数据寄存器	0x0000_000X
GPIOF1	GP_BA+0x344	R/W	GPIO端口F Bit 1 数据寄存器	0x0000_000X
GPIOF2	GP_BA+0x348	R/W	GPIO端口F Bit 2 数据寄存器	0x0000_000X
GPIOF3	GP_BA+0x34C	R/W	GPIO端口F Bit 3 数据寄存器	0x0000_000X
GPIOF4	GP_BA+0x350	R/W	GPIO端口F Bit 4 数据寄存器	0x0000_000X
GPIOF5	GP_BA+0x354	R/W	GPIO端口F Bit 5 数据寄存器	0x0000_000X

5.10.6 寄存器描述

GPIO端口[A/B/C/D/E/F] 管脚I/O 模式控制寄存器(GPIOx PMD)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_PMD	GP_BA+0x000	R/W	GPIO 端口 A 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO 端口 B 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO 端口 C 管脚 I/O 模式控制寄存器	0x0000_0000
GIOD_PMD	GP_BA+0x0C0	R/W	GPIO 端口 D 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO 端口 E 管脚 I/O 模式控制寄存器	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO 端口 F 管脚 I/O 模式控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PMD15		PMD14		PMD13		PMD12	
23	22	21	20	19	18	17	16
PMD11		PMD10		PMD9		PMD8	
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

Bits	描述	
[2n+1 :2n]	PMDn	<p>GPIO 端口[x] 管脚[n]模式控制</p> <p>决定 GPIOx 管脚[n]的 I/O 类型</p> <p>00 = GPIOx 管脚[n] 为输入模式</p> <p>01 = GPIOx 管脚[n]为输出模式</p> <p>10 = GPIOx 管脚[n]为开漏模式</p> <p>11 = 保留.</p> <p>注: GPIOF_PMD, PMD6 ~ PMD15保留</p>

GPIO端口[A/B/C/D/E/F] 管脚 OFF 数字使能寄存器(GPIOx_OFFD)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO 端口A管脚 OFF 数字使能寄存器	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO 端口B管脚 OFF 数字使能寄存器	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO 端口C管脚 OFF 数字使能寄存器	0x0000_0000
GIOD_OFFD	GP_BA+0x0C4	R/W	GPIO 端口D管脚 OFF 数字使能寄存器	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO 端口E管脚 OFF 数字使能寄存器	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO 端口F管脚 OFF 数字使能寄存器	0x0000_0000

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

Bits	描述	
[n+16]	OFFD[n]	GPIOx Pin[n] OFF 数字输入通道使能 用于控制 bit 对应的 GPIO 的数字输入通道是否禁用。 0 =使能 IO 数字输入通道。 1 =禁用 IO 数字输入通道（数字输入拉低） 注： GPIOF_OFFD, bits [31:22]保留。
[15:0]	-	保留

GPIO端口[A/B/C/D/E/F] 数字输出值寄存器(GPIOx DOUT)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO 端口 A 数据输出值寄存器	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO 端口 B 数据输出值寄存器	0x0000_FFFF
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO 端口 C 数据输出值寄存器寄存器	0x0000_FFFF
GIOD_DOUT	GP_BA+0x0C8	R/W	GPIO 端口 D 数据输出值寄存器寄存器	0x0000_FFFF
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO 端口 E 数据输出值寄存器寄存器	0x0000_FFFF
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO 端口 F 数据输出值寄存器寄存器	0x0000_003F

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

Bits	描述	
[31:16]	-	保留
[n]	DOUT[n]	<p>GPIOx Pin[n] 数据输出值</p> <p>在 GPIO 管脚被配置成输出, 开漏模式时, 控制 GPIO 管脚的状态。</p> <p>0 = GPIO 管脚被配置成输出时, GPIOX Pin[n] 为低。</p> <p>1 = GPIO 管脚被配置成输出时, GPIOX Pin[n] 为高。</p> <p>注: GPIOF_DOUT, bits [15:6]保留。</p>

GPIO端口[A/B/C/D/E/F] 数据输出写屏蔽寄存器(GPIOx_DMASK)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO 端口A数据输出写屏蔽寄存器	0x0000_0000
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO端口B数据输出写屏蔽寄存器	0x0000_0000
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO端口C数据输出写屏蔽寄存器	0x0000_0000
GIOD_DMASK	GP_BA+0x0CC	R/W	GPIO端口D数据输出写屏蔽寄存器	0x0000_0000
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO端口E数据输出写屏蔽寄存器	0x0000_0000
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO端口F数据输出写屏蔽寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[n]	DMASK[n]	<p>GPIO 端口[x] Pin [n] 数据输出写屏蔽</p> <p>用于保护相应寄存器 GPIOx_DOUT bit[n]。在设置 DMASK bit[n] 为'1'，相应的 GPIOx_DOUT[n] bit 被保护。写信号被屏蔽时，不能向保护位写数据。</p> <p>0 = 相应的 GPIOx_DOUT[n] bit 能被更新</p> <p>1 = 相应的 GPIOx_DOUT[n] bit 被保护</p> <p>注： GPIOF_DMASK, bits [15:6]保留。</p> <p>注： 这些位只对CPU写寄存器GPIOx_DOUT起作用，如果 CPU写寄存器 GPIO[x][n]不会起作用</p>

GPIO端口[A/B/C/D/E/F] 管脚数值寄存器(GPIOx_PIN)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_PIN	GP_BA+0x010	R	GPIO 端口A 管脚数值寄存器	0x0000_XXXX
GPIOB_PIN	GP_BA+0x050	R	GPIO 端口B管脚数值寄存器	0x0000_XXXX
GPIOC_PIN	GP_BA+0x090	R	GPIO 端口C管脚数值寄存器	0x0000_XXXX
GIOD_PIN	GP_BA+0x0D0	R	GPIO 端口D管脚数值寄存器	0x0000_XXXX
GPIOE_PIN	GP_BA+0x110	R	GPIO 端口E管脚数值寄存器	0x0000_XXXX
GPIOF_PIN	GP_BA+0x150	R	GPIO 端口F管脚数值寄存器	0x0000_00XX

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

Bits	描述	
[31:16]	-	保留
[n]	PIN[n]	GPIO 端口[x] Pin [n] 数值 这些位的值反映了各个 GPIO 管脚的真实状态。 注： GPIOF_PIN, bits [15:6]保留。

GPIO端口[A/B/C/D/E/F] 去抖动使能寄存器(GPIOx_DBEN)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO 端口A 去抖动使能寄存器	0x0000_0000
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO 端口B 去抖动使能寄存器	0x0000_0000
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO 端口C 去抖动使能寄存器	0x0000_0000
GIOD_DBEN	GP_BA+0x0D4	R/W	GPIO 端口D 去抖动使能寄存器	0x0000_0000
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO 端口E 去抖动使能寄存器	0x0000_0000
GPIOF_DBEN	GP_BA+0x114	R/W	GPIO 端口F 去抖动使能寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[n]	DBEN[n]	<p>GPIO 端口[x] Pin [n]输入信号去抖动使能</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，从而不会触发中断。</p> <p>仅用于边沿触发“edge-trigger”中断，不能用于电平触发 (“level trigger”) 中断。</p> <p>0 =禁止端口[x] Pin [n]去抖动功能</p> <p>1 =使能端口[x] Pin [n]去抖动功能</p> <p>去抖动功能对边沿触发中断有效，对于电平触发中断模式该使能位不起作用。</p> <p>注： GPIOF_DBEN, bits [15:6]保留。</p>

GPIO端口[A/B/C/D/E/F] 中断模式控制寄存器(GPIOx IMD)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_IMD	GP_BA+0x018	R/W	GPIO 端口 A 中断模式控制寄存器	0x0000_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO 端口 B 中断模式控制寄存器	0x0000_0000
GPIOC_IMD	GP_BA+0x098	R/W	GPIO 端口 C 中断模式控制寄存器	0x0000_0000
GIOD_IMD	GP_BA+0x0D8	R/W	GPIO 端口 D 中断模式控制寄存器	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO 端口 E 中断模式控制寄存器	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO 端口 F 中断模式控制寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[n]	IMD[n]	<p>GPIO 端口[x] Pin [n] 边沿或电平检测中断控制</p> <p>MD[n] 用于控制中断是电平触发或边沿触发。若为边沿触发中断，触发源可由去抖动控制；若为电平触发中断，输入源由一个 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>如果设置管脚为电平触发中断，则在寄存器 GPIOx_IER中，只能设置一个电平。若设置了两个电平去触发中断，则设置被忽略，不会产生中断。</p> <p>去抖动功能对于边沿触发中断有效，对于电平触发中断无效。</p> <p>注： GPIOF_IMD, bits [15:6]保留。</p>

GPIO端口[A/B/C/D/E/F] 中断使能寄存器(GPIOx_IER)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_IER	GP_BA+0x01C	R/W	GPIO 端口A 中断使能寄存器	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO 端口B中断使能寄存器	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO 端口C中断使能寄存器	0x0000_0000
GIOD_IER	GP_BA+0x0DC	R/W	GPIO 端口D中断使能寄存器	0x0000_0000
GPIOE_IER	GP_BA+0x11C	R/W	GPIO 端口E中断使能寄存器	0x0000_0000
GPIOF_IER	GP_BA+0x15C	R/W	GPIO 端口F中断使能寄存器	0x0000_0000

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

Bits	描述
[n+16]	<p>RIER[n]</p> <p>GPIO 端口 [x] Pin [n] 输入上升沿或输入高电平中断使能</p> <p>RIER[n] 用来使能相应 GPIO_PIN[n] 输入的中断。该位置 '1' 也能使能管脚的唤醒功能。</p> <p>当设置 RIER [n] 位为 1:</p> <p>如果中断是电平触发模式, 输入 PIN[n] 的状态为高电平时将产生中断。</p> <p>如果中断是边沿触发模式, 输入 PIN[n] 的状态由低电平到高电平变化时将产生中断</p> <p>1 = 使能 PIN[n] 高电平或由低电平到高电平变化的中断</p> <p>0 = 禁用 PIN[n] 高电平或由低电平到高电平变化的中断</p> <p>注: GPIOF_IE, bits [31:22]保留。</p>

Bits	描述	
[n]	FIER[n]	<p>GPIO 端口[x] Pin [n] 输入下降沿或输入低电平中断使能</p> <p>FIER[n] 用来使能相应 GPIO_PIN[n] 输入的中断。该位置 '1' 也能使能管脚的唤醒功能。</p> <p>当设置FIER [n] 位为 1:</p> <p>如果中断是电平触发模式，输入 PIN[n] 的状态为低电平时将产生中断。</p> <p>如果中断是边沿触发模式，输入 PIN[n] 的状态由高电平到低电平变化时将产生中断。</p> <p>1 = 使能 PIN[n] 低电平或由高电平到低电平变化的中断</p> <p>0 = 禁用 PIN[n] 低电平或由高电平到低电平变化的中断</p> <p>注： GPIOF_IER, bits [15:6]保留。</p>

GPIO端口[A/B/C/D/E/F] 中断触发源状态寄存器(GPIOx_ISRC)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_ISRC	GP_BA+0x020	R/WC	GPIO 端口A 中断触发源状态寄存器	0x0000_0000
GPIOB_ISRC	GP_BA+0x060	R/WC	GPIO 端口B 中断触发源状态寄存器	0x0000_0000
GPIOC_ISRC	GP_BA+0x0A0	R/WC	GPIO 端口C 中断触发源状态寄存器	0x0000_0000
GIOD_ISRC	GP_BA+0x0E0	R/WC	GPIO 端口D 中断触发源状态寄存器	0x0000_0000
GPIOE_ISRC	GP_BA+0x120	R/WC	GPIO 端口E 中断触发源状态寄存器	0x0000_0000
GPIOF_ISRC	GP_BA+0x160	R/WC	GPIO 端口F 中断触发源状态寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[n]	ISRC[n]	<p>GPIO 端口[x] Pin [n] 中断触发源标志</p> <p>读:</p> <p>1 = GPIOx[n] 产生中断</p> <p>0 = GPIOx[n] 没有中断</p> <p>写:</p> <p>1= 清除相应的未处理中断</p> <p>0= 无动作</p> <p>注: GPIOF_ISRC, bits [15:6]保留.</p>

GPIO端口[A/B/C/D/E/F] 上拉使能寄存器(GPIOx_PUEN)

寄存器	偏移量	R/W	描述	复位后的值
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO 端口A 上拉使能寄存器	0x0000_0000
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO 端口B上拉使能寄存器	0x0000_0000
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO 端口C上拉使能寄存器	0x0000_0000
GIOD_PUEN	GP_BA+0x0E4	R/W	GPIO 端口D上拉使能寄存器	0x0000_0000
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO 端口E上拉使能寄存器	0x0000_0000
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO 端口F上拉使能寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[n]	PUEN[n]	<p>GPIO端口[x] Pin [n] 上拉使能寄存器</p> <p>读:</p> <p>1 = GPIO 端口[A/B/C/D/E/F] bit [n] 上拉电阻使能</p> <p>0 = GPIO端口[A/B/C/D/E/F] bit [n] 上拉电阻禁止.</p> <p>注: GPIOF_PUEN, bits [15:6]保留.</p>

去抖周期控制寄存器 (DBNCECON)

寄存器	偏移量	R/W	描述	复位后的值
DBNCECON	GP_BA+0x180	R/W	去抖周期控制寄存器	0x0000_0000

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

Bits	描述	
[31:6]	-	保留
[5]	DBCLK_ON	<p>去抖时钟使能</p> <p>该位用来控制去抖时钟是否使能</p> <p>然而， 如果GPIO 管脚的中断被使能，去抖时钟会自动被使能，无论DBCLK_ON 的值是什么。</p> <p>1 = 使能消抖时钟</p> <p>0 = 禁止消抖时钟</p>
[4]	DBCLKSRC	<p>去抖动计数器时钟源选择</p> <p>1 = 去抖动计数器时钟源为内部 10 kHz 振荡器</p> <p>0 = 去抖动计数器时钟源为 HCLK</p>

Bits	描述																																		
[3:0]	DBCLKSEL	去抖动采样周期选择																																	
		DBCLKSEL	描述	0x0	采样中断输入每 1 clocks 1 次	0x1	采样中断输入每 2 clocks 1 次	0x2	采样中断输入每 4 clocks 1 次	0x3	采样中断输入每 8 clocks 1 次	0x4	采样中断输入每 16 clocks 1 次	0x5	采样中断输入每 32 clocks 1 次	0x6	采样中断输入每 64 clocks 1 次	0x7	采样中断输入每 128 clocks 1 次	0x8	采样中断输入每 256 clocks 1 次	0x9	采样中断输入每 2*256 clocks 1 次	0xA	采样中断输入每 4*256clocks 1 次	0xB	采样中断输入每 8*256 clocks 1 次	0xC	采样中断输入每 16*256 clocks 1 次	0xD	采样中断输入每 32*256 clocks 1 次	0xE	采样中断输入每 64*256 clocks 1 次	0xF	采样中断输入每 128*256 clocks 1 次
		DBCLKSEL	描述																																
		0x0	采样中断输入每 1 clocks 1 次																																
		0x1	采样中断输入每 2 clocks 1 次																																
		0x2	采样中断输入每 4 clocks 1 次																																
		0x3	采样中断输入每 8 clocks 1 次																																
		0x4	采样中断输入每 16 clocks 1 次																																
		0x5	采样中断输入每 32 clocks 1 次																																
		0x6	采样中断输入每 64 clocks 1 次																																
		0x7	采样中断输入每 128 clocks 1 次																																
		0x8	采样中断输入每 256 clocks 1 次																																
		0x9	采样中断输入每 2*256 clocks 1 次																																
		0xA	采样中断输入每 4*256clocks 1 次																																
		0xB	采样中断输入每 8*256 clocks 1 次																																
		0xC	采样中断输入每 16*256 clocks 1 次																																
		0xD	采样中断输入每 32*256 clocks 1 次																																
		0xE	采样中断输入每 64*256 clocks 1 次																																
		0xF	采样中断输入每 128*256 clocks 1 次																																

GPIO端口[A/B/C/D/E/F] Bit [n] 数据寄存器(GPIO[A/B/C/D/E/F][n])

寄存器	偏移量	R/W	描述	复位后的值
GPIOAn	GP_BA+0x200 GP_BA+0x23C	R/W	GPIO 端口A Bit [n] 数据寄存器	0x0000_000X
GPIOBn	GP_BA+0x240 GP_BA+0x27C	R/W	GPIO 端口B Bit [n] 数据寄存器	0x0000_000X
GPIOCn	GP_BA+0x280 GP_BA+0x2BC	R/W	GPIO 端口C Bit [n] 数据寄存器	0x0000_000X
GPIODn	GP_BA+0x2C0 GP_BA+0x2FC	R/W	GPIO 端口D Bit [n] 数据寄存器	0x0000_000X
GPIOEn	GP_BA+0x300 GP_BA+0x33C	R/W	GPIO 端口E Bit [n] 数据寄存器	0x0000_000X
GPIOFn	GP_BA+0x340 GP_BA+0x37C	R/W	GPIO 端口F Bit [n] 数据寄存器	0x0000_000X

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

Bits	描述
[31:1]	- 保留

Bits	描述	
[0]	GPIO[x][n]	<p>GPIO 端口[x] Pin [n] I/O 数据</p> <p>该域段支持对相关的GPIO port [x] pin [n].的位操作模式</p> <p>写该域段设置对应的GPIO 端口[x] pin [n] 输出值，读该域段得到对应的GPIO 端口[x] pin [n] 的值.</p> <p>读：</p> <p>1 = 对应的GPIO 端口[x] pin [n] 值为高</p> <p>0 =对应的 GPIO 端口[x] pin [n] 值为低.</p> <p>写：</p> <p>1 =设置对应的 GPIO 端口[x] pin [n]为高</p> <p>0 = 设置对应的 GPIO 端口[x] pin [n]为低.</p> <p>注： 写操作不会被寄存器 GPIOx_DMASK影响</p>

5.11 I²C

5.11.1 概述

I²C 为双线、双向串行总线，通过简单有效的连线方式实现设备间的数据交换。I²C 的标准是一个多主机总线，包括冲突检测和仲裁以防止两个或多个主机试图同时控制总线时发生的数据损坏。串行 8 位双向数据传输最大速度可以到 1.0 Mbps

数据在主机与从机之间通过 SCL 时钟线控制，在 SDA 数据线上按一字节一字节的同步传输。每个字节为 8 位长度，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 线上的跳变视为命令中断 (START or STOP)。

设备的片上 I²C 逻辑提供符合 I²C 总线规范的串行接口。I²C 端口自动处理字节传输，将 I2CON 的 ENS1 设置为 '1'，即可使能该端口。I²C H/W 接口通过 SDA 与 SCL 两个引脚连到 I²C 总线。用于 I²C 操作的两个管脚需要上拉电阻，因为这个两个管脚为开漏脚。

I²C 控制器配备两个从地址寄存器。当 I²C 作为主机时，该寄存器的内容不相关。在从机模式时，7 个最重要的位必须加载用户自己的从地址。如果 I2CADDR 内容和接收到的从地址匹配，I²C 硬件会回应。

该控制器支持“广播呼叫(GC)”功能，如果 GC 位置位，I²C 端口硬件将应答广播呼叫的地址 (00H)。清 GC 位可禁用“广播呼叫”功能。当 GC 位被置且 I²C 处于从机模式时，主机发出广播呼叫地址到 I²C 总线后，从机可以通过地址 00H 接收广播呼叫地址，然后它将跟随 GC 模式的状态。如果工作在主机模式，当它发送广播地址(00H)到 I²C 总线，ACK 位必须清零。

I²C 总线控制器有 2 个地址掩码寄存器，支持多地址识别。当地址掩码寄存器的某位被置 1，表示可以忽略接收到的相应地址位。若该位被置 0，表示接收到的相应地址位应和地址寄存器内的值完全吻合。

5.11.2 特征

- 支持主机/从机 模式
- 主从机之间双向数据传输
- 多主机总线支持（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
- 串行同步时钟可作为握手方式控制总线上数据暂停及恢复传送
- 内建14位溢出计数器，当 I²C 总线中止且定时计数器溢出，产生 I²C 中断
- 可编辑的时钟适用于不同速率控制
- 支持 7 位地址模式
- I²C 总线控制器支持多地址识别（2组从机地址带 mask 选项）
- 支持掉电唤醒功能

5.11.3 功能描述

5.11.3.1 I²C 协议

通常标准的 I²C 通信包含四个部分：

- 起始信号 (START) 或者重复起始信号
- 从机地址传输
- 数据传输
- 停止信号 (STOP)

5.11.3.2 产生STOP 信号

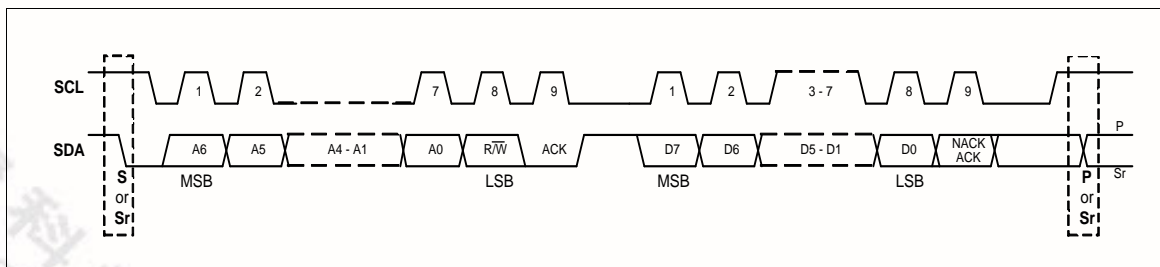


图 5-41 I²C 协议

5.11.3.3 I²C总线上的数据传输

主机发出7 位地址给从机接收者进行寻址。

传输方向未改变

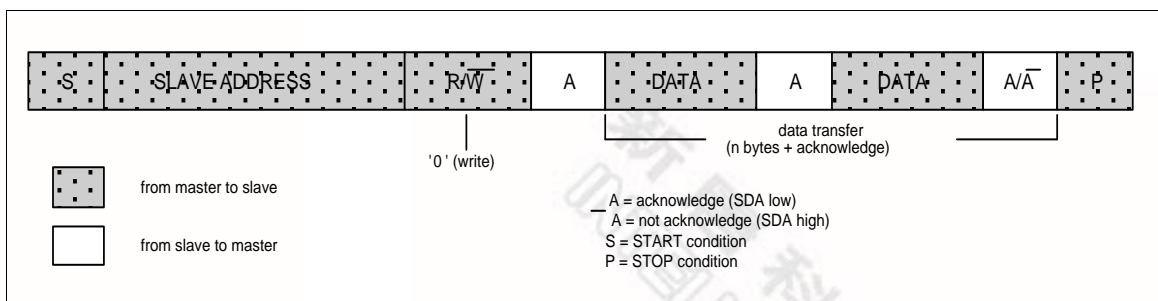


图 5-42 I²C 主机向从机传输数据

第一个字节（从机地址）后，主机紧接着从从机读取数据

传输方向改变

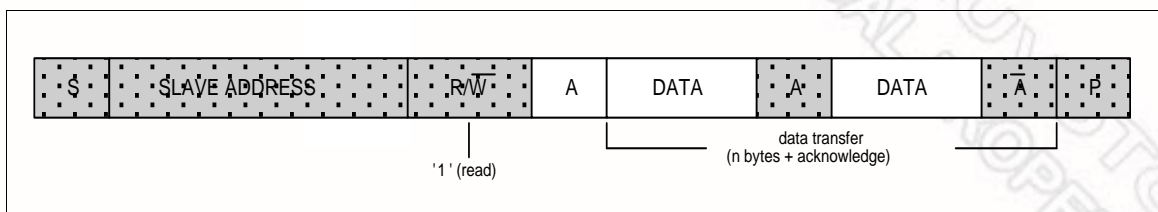


图 5-43 I²C 主机从从机读取数据

5.11.3.4 起始或重复起始信号

当总线处于空闲状态下，说明没有主机设备占用总线（SCL 和 SDA 线同时为高），主机可以通过发送一个 起始 (START) 信号来发起传输请求。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的数据传输的开始。

重复起始信号 (Sr) 即在两个起始 (START) 信号之间没有停止 (STOP) 信号。主机采用这种方法与另一个从机或同一个从机以不同传输方向进行通信（例如：从写设备到读设备）而不释放总线。

5.11.3.5 停止 (STOP) 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。

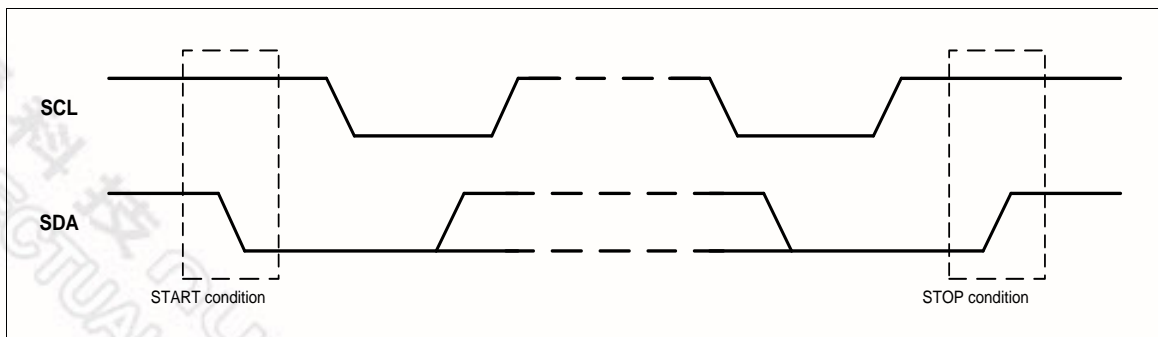


图 5-44 I²C 起始 (START) 和停止 (STOP) 条件

5.11.3.6 从机地址传输

起始信号后立即传输的第一个字节就是从机地址。这是一个7位设备地址跟随一个 RW 位。RW 位标志从机的信号的数据传输方向。系统中没有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 时钟周期时将 SDA 拉低作为应答。

5.11.3.7 数据传输

当从机设备地址被成功识别，就可以根据 RW 位所决定的方向按一字节一字节方式进行数据传输。每个字节传输完后紧接着在第9个 SCL 时钟周期会有一个应答信号位。如果从机上产生无应答信号 (NACK)，主机可以产生停止信号来中止数据传输或者产生重复起始信号开始新一轮数据传输。

当主机作为接收设备时，发生无应答信号 (NACK)，则从机释放 SDA 线，让主机产生停止信号或重复起始信号。

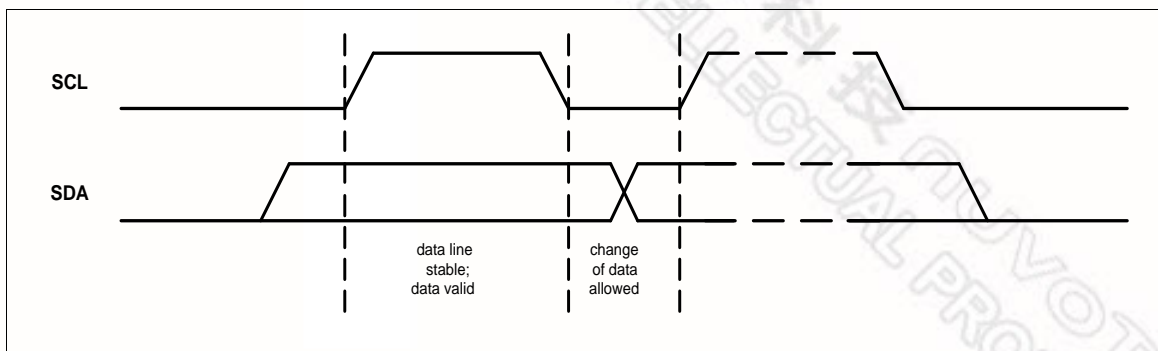


图 5-45 I²C 总线上的位传输

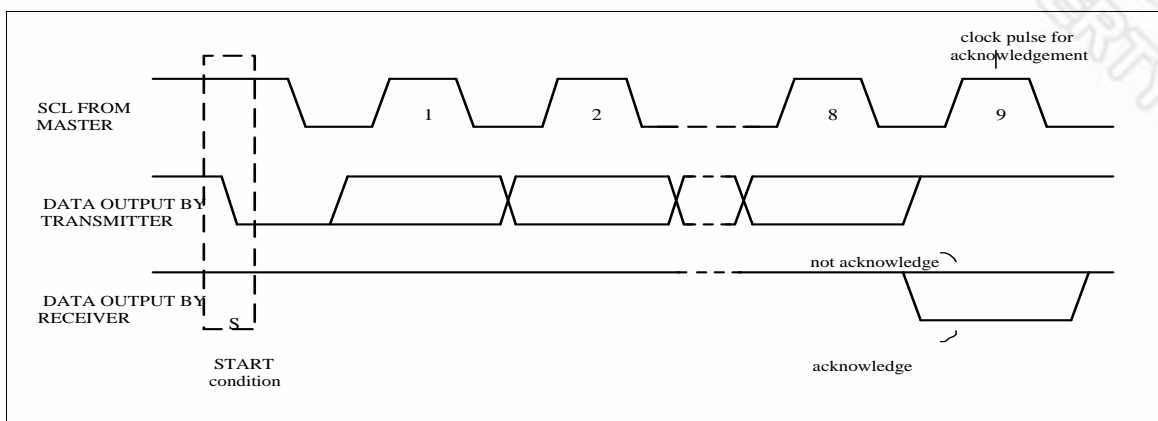


图 5-46 I²C 总线上的应答信号

5.11.3.8 时钟波特率

当 I²C 在主机模式下，I²C 数据的波特率由 CLK_DIV 寄存器设定。其在从机模式下时是不重要的；在从机模式下，I²C 将自动与主机 I²C 设备时钟频率同步。

I²C 数据波特率的设定： $I^2C \text{ 数据波特率} = PCLK / (4 \times (CLK_DIV + 1))$ 。如果 $PCLK = 16 \text{ MHz}$ ， $CLK_DIV = 40$ (28H)，那么 $I^2C \text{ 数据波特率} = 16 \text{ MHz} / (4 \times (40 + 1)) = 97.5 \text{ Kbits/sec}$ 。

5.11.3.9 超时

MCU 提供一个 14 位超时的计数器来处理当 I²C 总线锁死时的情况。当计数功能使能后，计数器开始计数直至溢出 (TIF = 1) 并要求 CPU 产生 I²C 中断或者检测到停止信号。用户也可以通过清除 TOUTEN 为 0 关闭计数功能。当超时计数器使能，对 I2C_STS 和 STAINSTS 标志置高和 I²C 总线的下降沿会使计数器复位，再对 I2C_STS 清零或总线时钟的下降沿之后会重新开始计数。如果 I²C 总线锁死，会使 STATUS 及 I2C_STS 标志一段时间内不再更新，该 14 位超时计数器会发生溢出从而产生 I²C 中断通知 CPU。参考下图 14 位超时计数器。用户可以写 1 清 TIF 为 0。

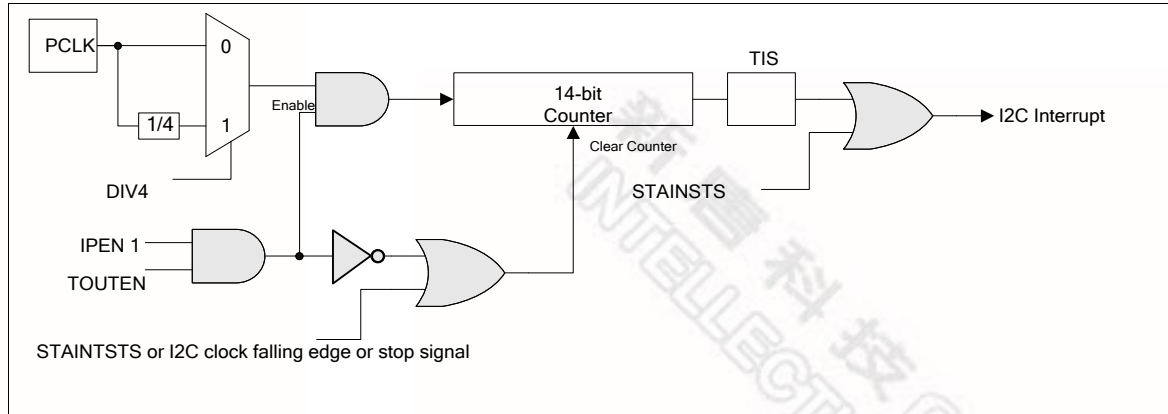


图 5-47 I²C 超时计数器框图

5.11.3.10 I²C 唤醒控制寄存器 (I2CWKUPCON)

当进入掉电模式，其他的 I²C 主机可以通过寻址 I²C 设备唤醒我们的芯片。用户必须在进入掉电模式之前设置 I2CWKUPCON[WKUPEN]。

5.11.3.11 I²C 唤醒状态寄存器 (I2CWKUPSTS)

当系统被其他的 I²C 主机设备唤醒，WKUPIF 被置位表示该事件发生。

5.11.3.12 工作模式

I²C 控制器有 5 种工作模式，主机传送，主机接收，从机传送，从机接收和广播呼叫模式。I2C_STS 位被清零后，I2CCON 寄存器中 START、STOP 和 ACK 位决定控制器的下一个状态，新的操作完成，一个新的状态码会被更新并且 I2C_STS 和 STAINSTS 位会被置位。如果 I²C 中断控制位 INTEN 被置位，可在中断服务子程序中根据新的状态码执行相应的动作。

在实际应用中，I²C 端口可以作为主机和从机。在从机模式，I²C 端口寻找自身从机地址和广播呼叫地址，如果任一地址被检测到，从机准备接收或发送数据（通过设置 ACK 位），应答信号在第 9 个时钟脉冲时发送，此时，如果中断已经使能，则主从机设备发生中断请求。在主机模式，如果总线仲裁失败，I²C 立即切换到从机模式，并检测自身从机地址。

5.11.3.13 主机发送模式

在主机传送模式，当 SCL 线上输出时钟信号时，SDA 线上输出数据。传输的第一个字节包含接收设备的 7 位从机地址和数据方向位。在该模式下，数据方向位为 0，在流程图中用 'W' 表示。因此第一个传输的字节是 SLA+W。串行数据一次传送 8 位。每传送完一个字节后，会收到一个应答位。起始和停止条件的输出分别表示传送的开始和结束。

5.11.3.14 主机接收模式

在该模式下，数据方向为 (R/W) 将为 1，在流程图中用 'R' 表示。因此第一个传输的字节是 SLA+R。当 SCL 输出时钟信号时，串行数据通过 SDA 进行接收。串行数据每次接收 8 位。每接收到一个字节，发送一个响应位。起始和停止条件的输出分别表示串行传输的开始和结束。

5.11.3.15 从机接收模式

通过 SDA 和 SCL 接收串行数据和串行时钟。每接收到一个字节，发送一个响应位。起始和停止条

件将会被分别识别为串行传输的开始和结束。地址识别由硬件在接收从机地址和数据方向后完成。

5.11.3.16 从机发送模式

第一个字节是在从机接收模式接收并处理的，然后传输方向位被反转。当通过 SCL 线接收串行时钟时，I²C 通过 SDA 线传送数据。起始和停止条件将会被分别识别为串行传输的开始和结束。

各模式下的数据传输如下图所示：.

注：

- I2CON = I2CCON
- I2DAT = I2CDATA
- STA = START
- STO = STOP
- AA = ACK

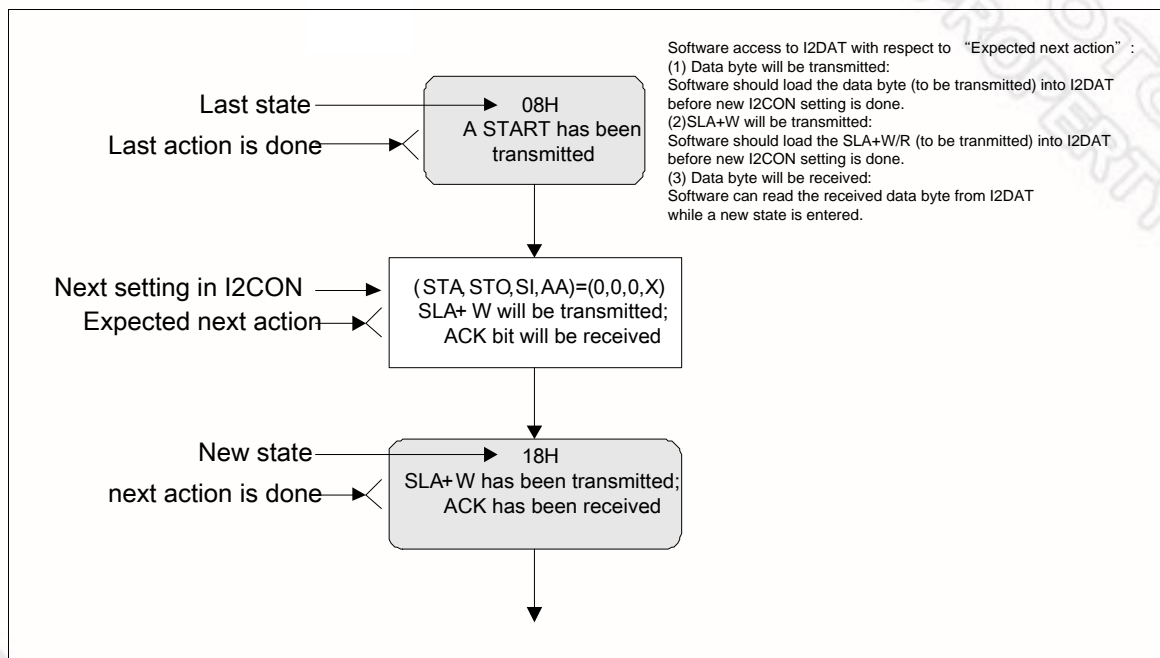


图 5-48 I²C 正常主机传输状态

图 5-49 I²C 主机传送模式

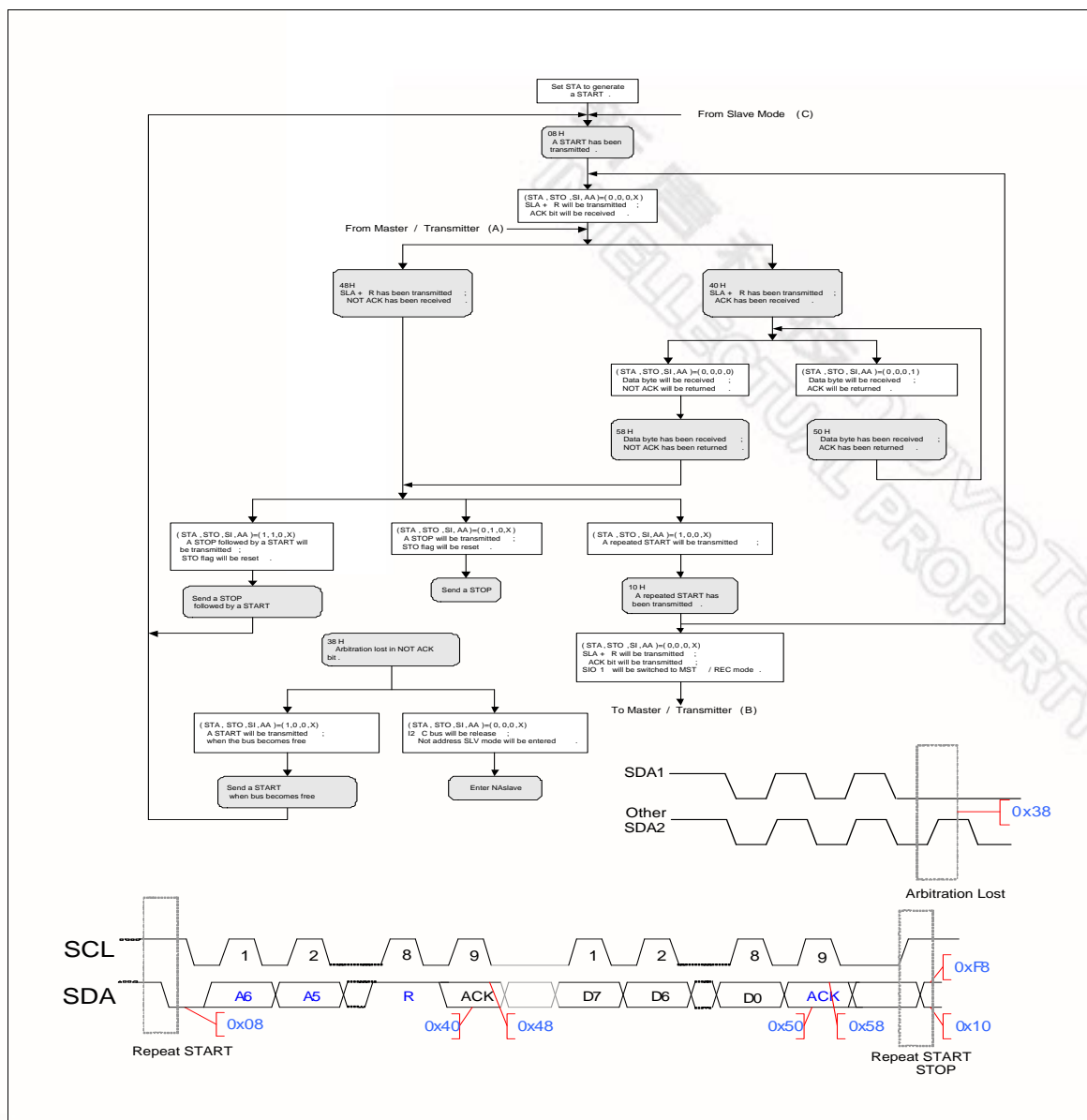


图 5-50 I²C 主机接收模式

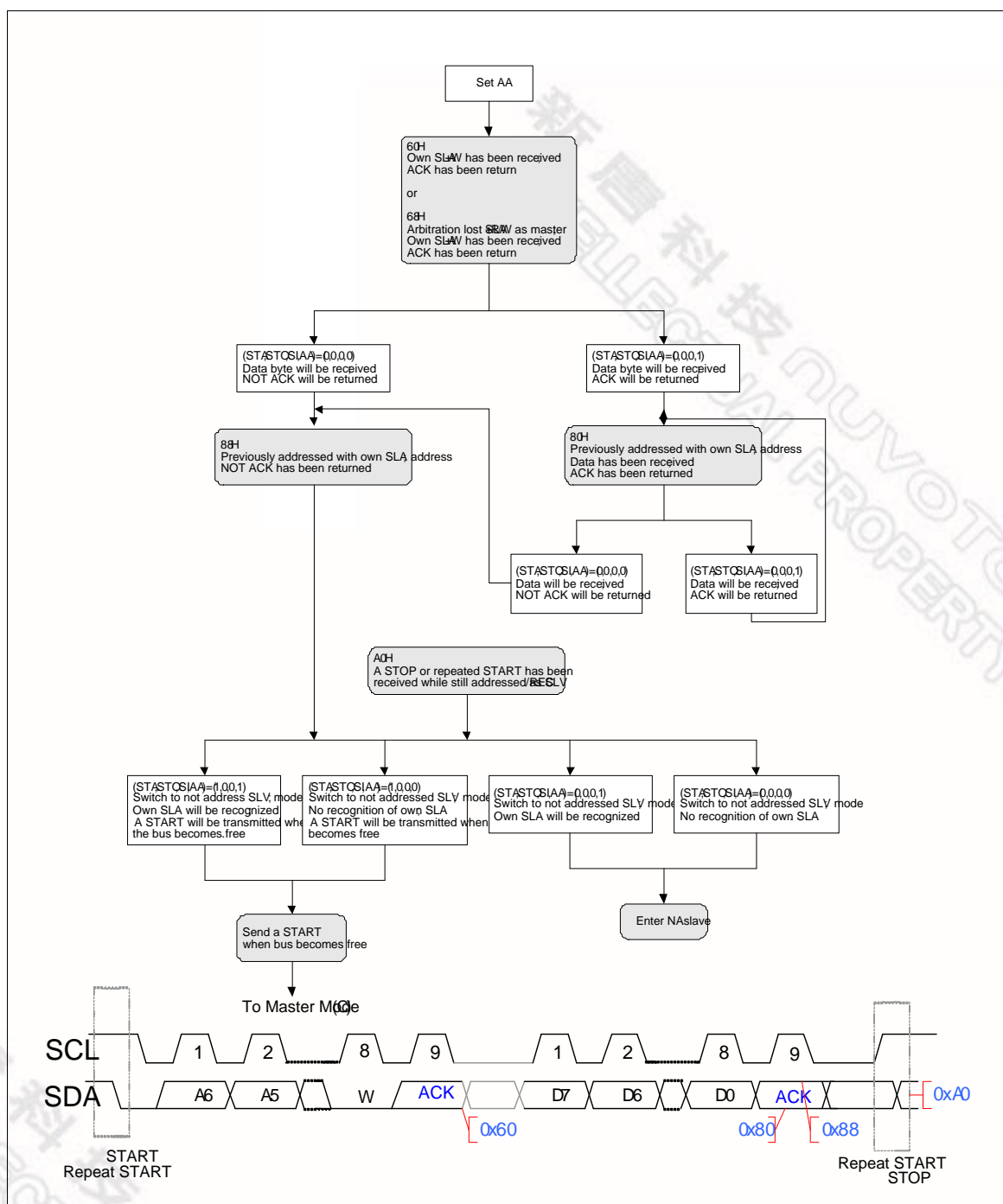


图 5-51 I²C 从机接收模式

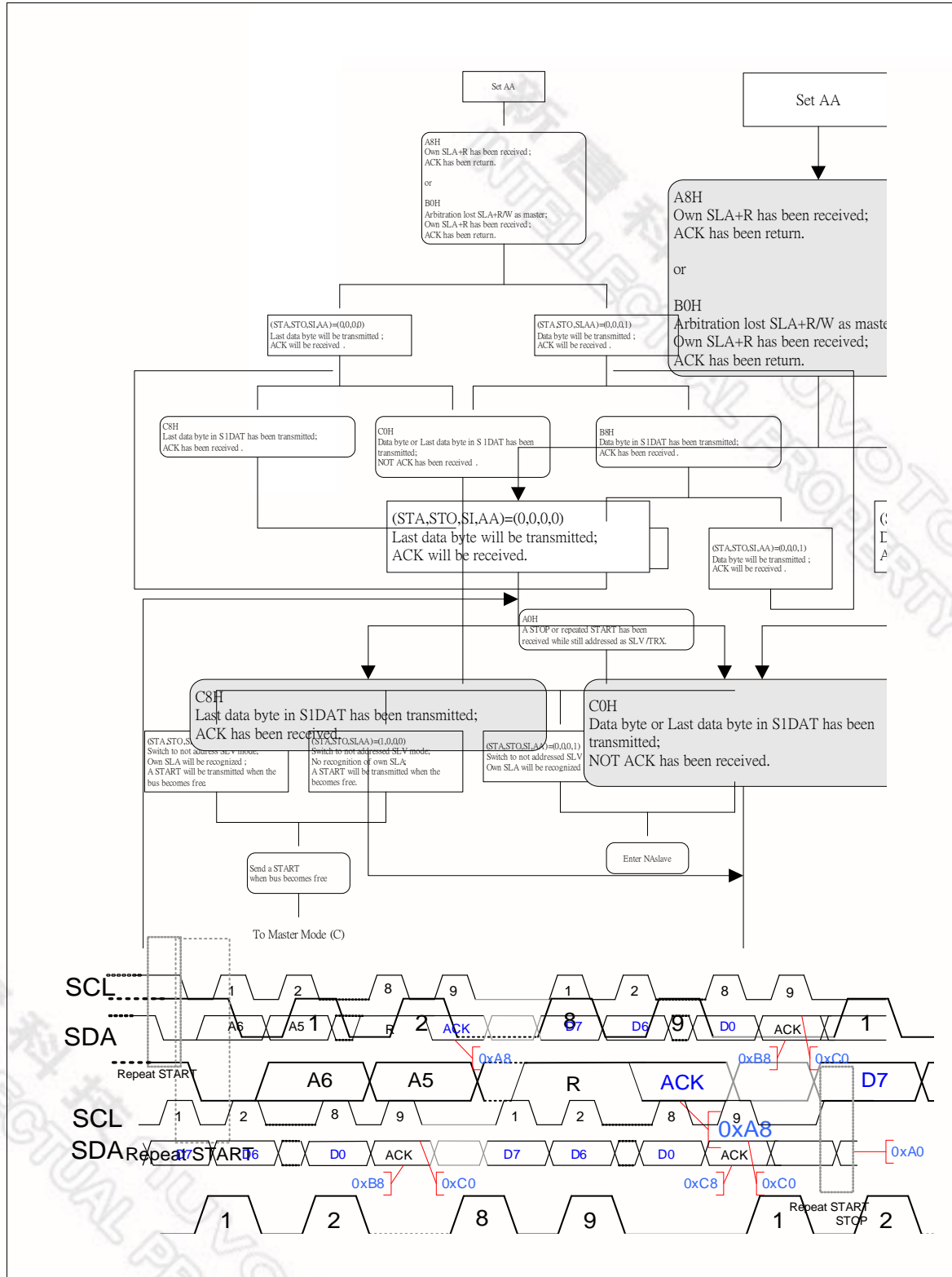


图 5-52 I²C 从机发送模式

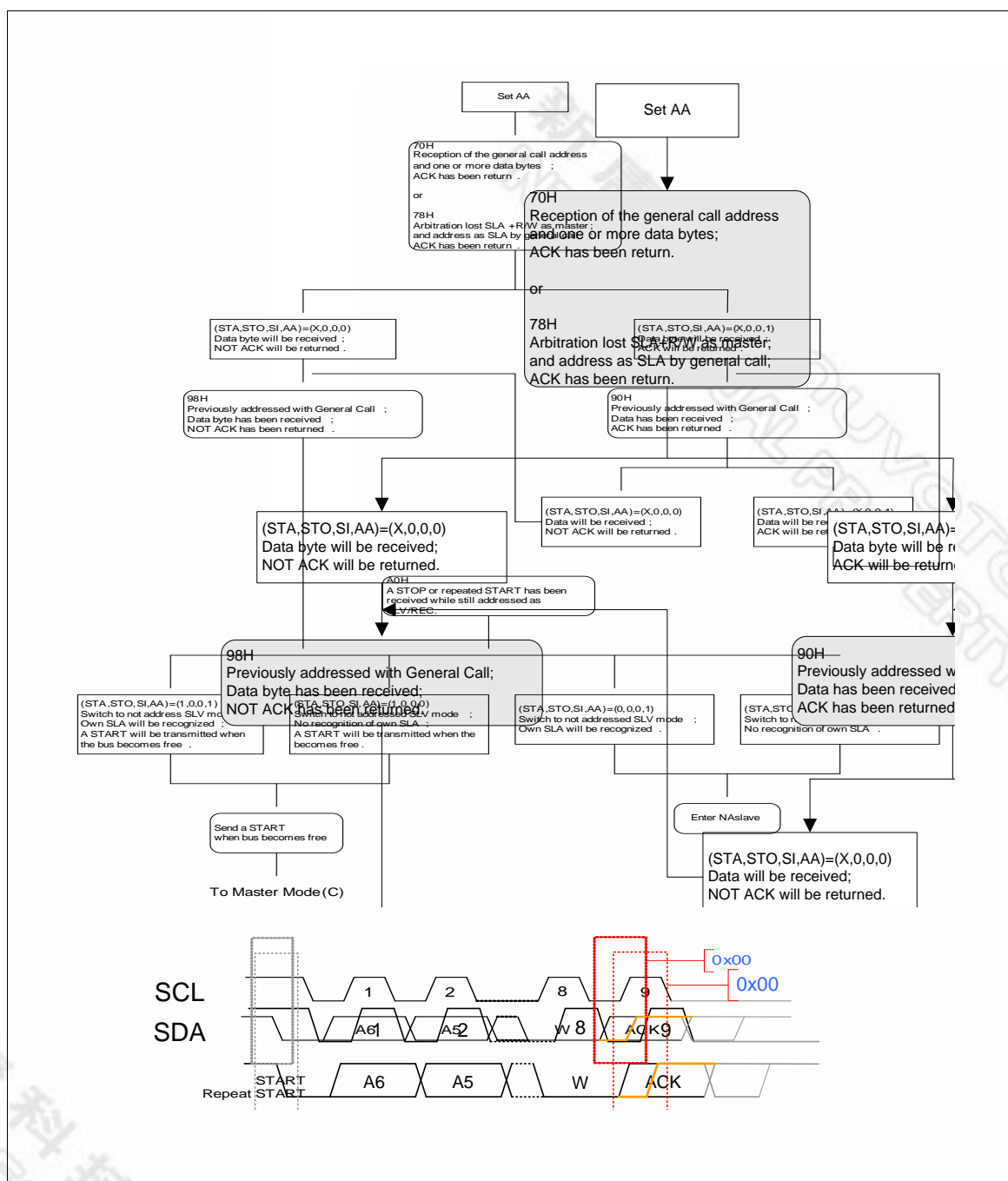


图 5-53 I²C 广播呼叫模式

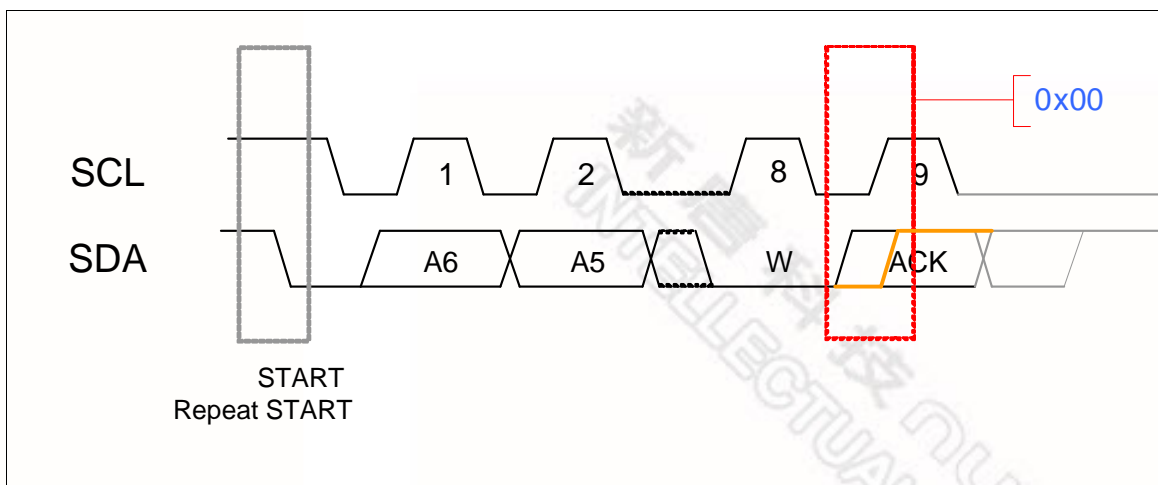


图 5-54 I²C 总线错误时序

5.11.4 寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
I2C0_BA = 0x4002_0000				
I2C1_BA = 0x4012_0000				
I2CCON	I2Cx_BA+0x00	R/W	I ² C 控制寄存器	0x0000_0000
I2CINTSTS	I2Cx_BA+0x04	R/W	I ² C 中断状态寄存器	0x0000_0000
I2CSTATUS	I2Cx_BA+0x08	R	I ² C 状态寄存器	0x0000_00F8
I2CDIV	I2Cx_BA+0x0C	R/W	I ² C 时钟分频寄存器	0x0000_0000
I2CTOUT	I2Cx_BA+0x10	R/W	I ² C 超时控制寄存器	0x0000_0000
I2CDATA	I2Cx_BA+0x14	R/W	I ² C 数据寄存器	0x0000_0000
I2CSADDR0	I2Cx_BA+0x18	R/W	从地址寄存器0	0x0000_0000
I2CSADDR1	I2Cx_BA+0x1C	R/W	从地址寄存器1	0x0000_0000
I2CSAMASK0	I2Cx_BA+0x28	R/W	从地址掩码寄存器0	0x0000_0000
I2CSAMASK1	I2Cx_BA+0x2C	R/W	从地址掩码寄存器1	0x0000_0000
I2CWKUPCON	I2Cx_BA+0x3C	R/W	I ² C 唤醒控制寄存器	0x0000_0000
I2CWKUPSTS	I2Cx_BA+0x40	R	I ² C 唤醒状态寄存器	0x0000_0000

5.11.5 寄存器 描述

I²C控制寄存器 (I2CCON)

寄存器	偏移量	R/W	描述	复位后的值
I2CCON	I2Cx_BA + 0x00	R/W	I ² C 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
INTEN	-		I2C_STS	START	STOP	ACK	IPEN

Bits	描述	
[31:8]	-	保留
[7]	INTEN	中断使能 1 = 使能I2C 中断 0 = 禁止I2C 中断
[6:5]	-	保留
[4]	I2C_STS	I ² C 状态. I2CSTATUS 寄存器中出现一个新的状态, 该位会自动置位, 如果INTEN 位被置位将产生I ² C 中断, 必须软件写1清零该位, 然后I ² C总线时序才会继续收发直到停止位或IPEN禁止 1 = 有一个新的Status状态 0 = Status状态清除了, I ² C总线时序继续.
[3]	START	I ² C 开始命令 设置该位为1, 进入主机模式, 如果总线处于空闲状态, I ² C 硬件会送出 START 或 重复 START 条件. 在START 命令作用之后, 该位会被清零, STATUS 会被更新. 1 = 发送一个 START或重复 START 条件到总线. 0 = 在START或重复 START作用后
[2]	STOP	I ² C 停止控制位. 在主机模式, 设置 STOP来传送一个 STOP 条件到总线, 然后 I ² C 硬件将会检查总线状况, 如果检测到一个 STOP 状况, 这个标志会被硬件自动清除. 在 I ² C 从机模式, 设置 STOP复位 I ² C 硬件来定义“无地址”从机模式, 这表示在从机

Bits	描述	
		接收模式不再接收从主机传送装置发送的数据。 1 =主机模式下传送一个 STOP，从机模式下，让从机地址无效 0 =如果检测到一个 STOP 状况，这个标志会被硬件自动清除。
[1]	ACK	应答控制位 1 = 以下两种情况，若在地址或数据接收之前该位置1，在 SCL 线上的应答时钟脉冲期间将返回一个应答： a. 从机正在应答主机发送的地址 b. 接收设备正在应答发送设备发送的数据 0 =:若在地址或数据接收之前该位清0，，则在 SCL 线上的应答时钟脉冲期间将返回一个非应答（SDA 上的高电平）。
[0]	IPEN	I²C 功能使能 当该位为1，I ² C 串行功能被使能。 1 =使能 I ² C 功能 0 =禁止 I ² C 功能

I²C中断状态寄存器(I2CINTSTS)

寄存器	偏移量	R/W	描述	复位后的值
I2CINTSTS	I2Cx_BA + 0x04	R	I ² C中断状态寄存器	0x0000_0000

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

Bits	描述	
[31:2]	-	保留
[1]	TIF	Time-Out 状态 1 = 定时器超时标志，由硬件置1,若INTEN为1会产生中断，软件写1清0 0 = 无定时器超时。
[0]	INTSTS	I²C STATUS的中断状态 如果在I2CSTATUS中出现了一个新的状态，此位会置1，如果INTEN=1，则会产生中断，必须软件写1清零此位。 1 : I ² C 的 Status 中断标志 0 : 无I ² C Status 中断

I²C 状态寄存器 (I2CSTATUS)

寄存器	偏移量	R/W	描述	复位后的值
I2CSTATUS	I2Cx_BA + 0x08	R	I ² C 状态寄存器	0x0000_00F8

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

Bits	描述	
[31:8]	-	保留
[7:0]	STATUS[7:0]	<p>I²C 状态 寄存器</p> <p>这是只读寄存器，低三位一直为 0。高五位是状态码。当进入其中任一状态时，就会产生状态中断请求(I2C_STS = 1 和 STAINSTS = 1)。在 I2C_STS被硬件置位或 I2C_STS被软件复位后一个机器周期，有效状态码出现在STATUS 中。</p> <p>此外，00H 状态表示总线错误。总线错误发生在 START 或 STOP 条件出现在帧结构不正确的位置。不正确的位置比如是在串行传输地址字节、数据字节或应答位期间。</p> <p>从总线错误中恢复I²C，STOP应该置位，I2C_STS应该清零，进入无地址从模式，然后清零STOP释放总线，等待新的通讯。当总线错误的时候，I²C总线不能识别stop条件。</p>

状态信息:

STATUS	描述	STATUS	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 (AA =1)
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	从机发送最后一个数据 ACK (AA=0)
0xF8	停止	0x60	从机接收地址 ACK
0x38	主机发送仲裁丢失	0x68	从机接收仲裁丢失
0x40	主机地址接收ACK	0x80	从机接收数据 ACK
0x48	主机地址接收NACK	0x88	从机接收数据NACK
0x50	主机数据地址接收 ACK	0x70	广播模式地址ACK
0x58	主机数据接收 NACK	0x78	广播模式仲裁 Lost
0x00	总线错误	0x90	广播模式数据 ACK
		0x98	广播模式数据 NACK (STO no A0)

具体关于 **status**的信息在操作模式部分描述.

I²C 波特率控制寄存器 (I2CDIV)

寄存器	偏移量	R/W	描述	复位后的值
I2CDIV	I2Cx_BA + 0x0C	R/W	I ² C 时钟分频寄存器	0x0000_0000

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

Bits	描述	
[31:8]	-	保留
[7:0]	CLK_DIV[7:0]	<p>I²C 时钟分频控制寄存器</p> <p>I²C 数据波特率 = $PCLK / (4 \times (CLK_DIV + 1))$</p> <p>注: CLK_DIV的最小值是4.</p>

I²C 超时计数器寄存器 (I2CTOUT)

寄存器	偏移量	R/W	描述	复位后的值
I2CTOUT	I2Cx_BA + 0x10	R/W	I ² C超时计数器寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-						DIV4	TOUTEN

Bits	描述	
[31:2]	-	保留
[1]	DIV4	<p>超时计数器输入时钟除以 4</p> <p>1 = 使能</p> <p>0 = 禁用</p> <p>该位使能后，超时时间扩大 4 倍。</p>
[0]	TOUTEN	<p>超时计数器使能/禁用</p> <p>1 = 使能</p> <p>0 = 禁用</p> <p>当使能该位，则 14- 位超时计数器将会在 STAINTSTS 清零后开始计数。设置 STAINTSTS 为高或 I²C 时钟下降沿或停止信号将会复位计数器，在 SI 清零之后，计数器会重新开始计数。</p>

I²C数据寄存器(I2CDATA)

寄存器	偏移量	R/W	描述	复位后的值
I2CDATA	I2Cx_BA + 0x14	R/W	I ² C 数据寄存器	0x0000_0000

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

Bits	描述	
[31:8]	-	保留
[7:0]	DATA[7:0]	<p>I²C 数据寄存器</p> <p>该寄存器包含一字节准备发送或者刚接收到的串行数据。只要不是正在进行移位字节的操作，用户可以直接从该 8 位寄存器读取或者向该 8 位寄存器写入数据。这发生在串行中断标志被置位时，只要 I2C_STS 位被置，该寄存器中的数据就会保持稳定。当数据正在被移出时，在总线上的数据同时被移入；该寄存器总是包含出现在总线上的最后一个字节。这样，即便发生仲裁丢失，从主机发送器到从机接收器的传输中，该寄存器中的数据是正确的。</p> <p>数据和ACK位来自一个9-bit移位寄存器，ACK位被设备硬件控制，不能被用户访问。串行数据在SCL线的上升沿，通过ACK位被移进DATA。当一个字节被移进DATA，DATA中的串行数据才有效，在第九个时钟脉冲返回握手位 (ACK or NACK)。</p>

I²C从机地址寄存器 (I2CSADDRx)

寄存器	偏移量	R/W	描述	复位后的值
I2CSADDR0	I2Cx_BA + 0x18	R/W	I2C从机地址控制寄存器 0	0x0000_0000
I2CSADDR1	I2Cx_BA + 0x1C	R/W	I2C从机地址控制寄存器 1	0x0000_0000

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

Bits	描述	
[31:8]	-	保留
[7:1]	SADDR[6:0]	I ² C 从机地址 寄存器 主机模式下，该寄存器内容没有意义。从机模式下，高七位作为芯片本身的地址。如果地址符合，I ² C 硬件将会自动应答。
[0]	GCALL	广播呼叫功能 该控制器支持“广播呼叫(GC)”功能，如果GCALL位置位，I ² C 端口硬件将应答广播呼叫的地址(00H)。清 GCALL位可禁用“广播呼叫”功能。当 GCALL位被置且 I ² C 处于从机模式时，主机发出广播呼叫地址到 I ² C 总线后，从机可以通过地址 00H 接收广播呼叫地址，然后它将跟随 GCALL模式的状态。如果工作在主机模式，当它发送广播地址(00H)到I ² C总线，ACK位必须清零。 1 = 使能广播呼叫功能。 0 = 禁止广播呼叫功能

从机地址掩码寄存器 (I2CSAMSKx)

寄存器	偏移量	R/W	描述	复位后的值
I2CSAMASK0	I2Cx_BA+ 0x28	R/W	I2C从机地址掩码寄存器 0	0x0000_0000
I2CSAMASK1	I2Cx_BA+ 0x2C	R/W	I2C从机地址掩码寄存器 1	0x0000_0000

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

Bits	描述	
[31:8]	-	保留
[7:1]	SAMASK[6:0]	<p>I²C 从机地址掩码 寄存器</p> <p>1 = 使能掩码（接收到的相应地址位不予辨识）</p> <p>0 = 禁用掩码（接收到的相应地址必须完全符合地址寄存器）</p> <p>I²C 总线控制器有2个地址掩码寄存器，支持多地址识别。当地址掩码寄存器的某位被置 '1'，表示接收到的地址的相应位可忽略。如果该位被置 '0'，则表示接收到的地址的相应位必须和地址寄存器中的完全一致。</p>
[0]		保留

I²C 唤醒控制寄存器 (I2WKUPCON)

寄存器	偏移量	R/W	描述	复位后的值
I2CWKUPCON	I2C_BA+0x3C	R/W	I ² C 唤醒控制寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[0]	WKUPEN	I ² C 唤醒功能使能 1 = I ² C 唤醒 功能使能. 0 = I ² C唤醒功能禁止.

I²C 唤醒状态寄存器 (I2CWKUPSTS)

寄存器	偏移量	R/W	描述	复位后的值
I2CWKUPSTS	I2C_BA+0x40	R	I ² C 唤醒状态寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[0]	WKUPIF	<p>唤醒中断标志</p> <p>1 = 唤醒事件发生.</p> <p>0 = 唤醒事件没有发生.</p> <p>向该位写1清0</p>

5.12 I²S

5.12.1 概述

该音频控制器由I²S协议与外部音频CODEC接口组成。两个 8 字的 FIFO 分别用于接收和发送通道，可以处理 8 ~ 32 位字大小。PDMA 控制器处理 FIFO 和内存之间的数据移动。

5.12.2 特性

- I²S 可工作在主机模式或从机模式。
- 可处理 8, 16, 24 和 32位字大小。
- 支持单声道和立体声的音频数据。
- 支持 I²S 和 MSB 对齐数据格式。
- 提供两个FIFO 数据缓存(每一个32位)，一个用于发送，一个用于接收。
- 当缓存超过可编程边界时，产生中断请求。
- 两个 PDMA通道 请求，一个用于发送，一个用于接收。

5.12.3 框图

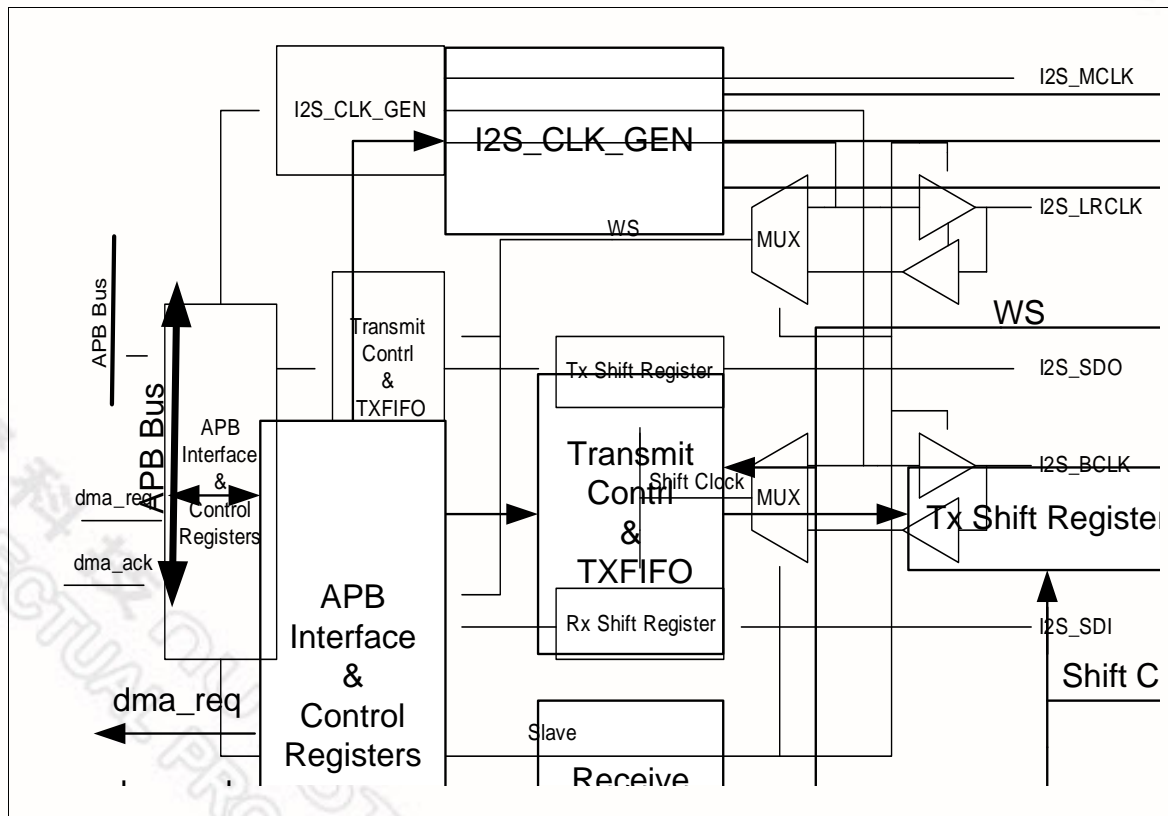


图 5-55 I²S 控制器框图

5.12.4 功能描述

5.12.4.1 I²S 操作

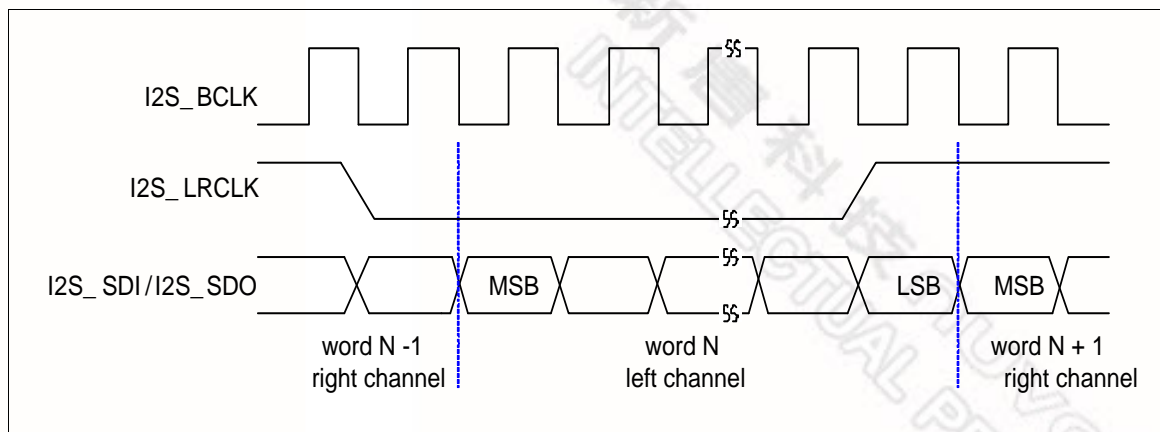


图 5-56 I²S 总线时序图 (Format = 0)

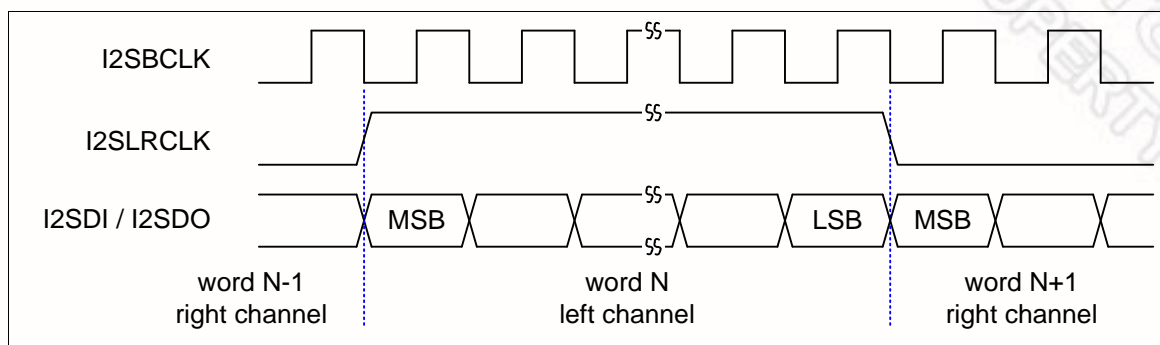


图 5-57 MSB 对齐时序图 (Format = 1)

5.12.4.2 I²S FIFO

8/16/32位的FIFO的数据结构如下所述:

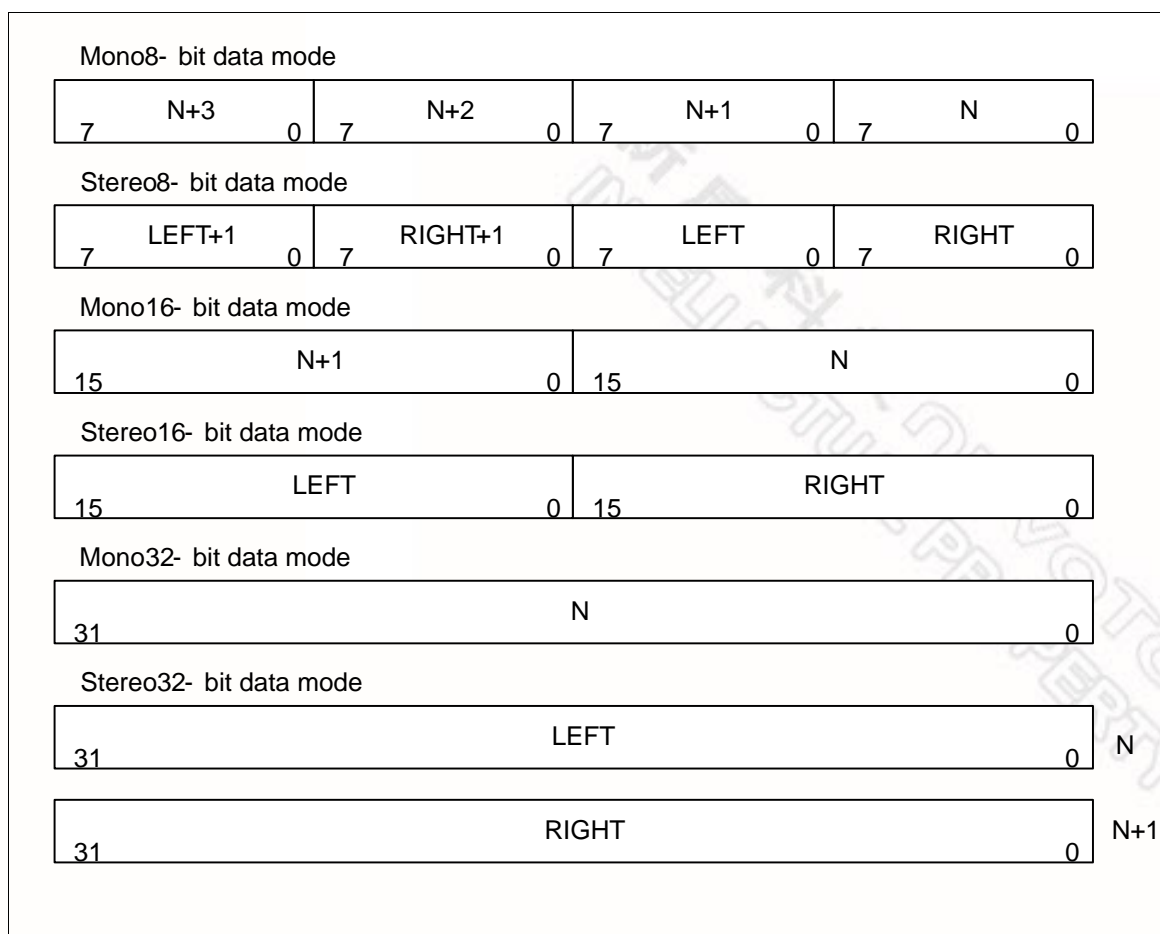


图 5-58 不同 I²S 模式下 FIFO 的内容

5.12.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
I2S_BA = 0x401A_0000.				
I2S_CTRL	I2S_BA+0x00	R/W	I ² S控制寄存器	0x0000_0000
I2S_CLKDIV	I2S_BA+0x04	R/W	I ² S 时钟分频寄存器	0x0000_0000
I2S_INTEN	I2S_BA+0x08	R/W	I ² S 中断使能寄存器	0x0000_0000
I2S_STATUS	I2S_BA+0x0C	R/W	I ² S 状态寄存器	0x0014_1000
I2S_TXFIFO	I2S_BA+0x10	W	I ² S 发送 FIFO 寄存器	0x0000_0000
I2S_RXFIFO	I2S_BA+0x14	R	I ² S 接收 FIFO 寄存器	0x0000_0000

5.12.6 寄存器描述

I²S 控制寄存器 (I2S_CTRL)

寄存器	偏移量	R/W	描述	复位后的值
I2S_CTRL	I2S_BA + 0x00	R/W	I ² S 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
RXLCH	-	RXDMA	TXDMA	CLK_RXFIFO	CLR_TXFIFO	LCHZCEN	RCHZCEN
15	14	13	12	11	10	9	8
MCLKEN	RXTH			TXTH			SLAVE
7	6	5	4	3	2	1	0
FORMAT	MONO	WORDWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	描述	
[31:24]	-	保留
[23]	RXLCH	接收左通道使能 当选择单声道模式(MONO = 1), 如果RXLCH被设置为0, 则I2S 将接收右声道数据, 而如果RXLCH被设置为1, I2S接收左声道数据. 1 = 当选择单声道模式时, 接收左声道数据. 0 = 当选择单声道模式时, 接收右声道数据.
[22]	-	保留
[21]	RXDMA	使能接收 DMA 当 RX DMA 使能时, 如果 FIFO 非空, I ² S 请求 PDMA 从接收 FIFO 向内存传输数据. 1 = 使能 RX DMA. 0 = 禁用 RX DMA.
[20]	TXDMA	使能发送 DMA 当 TX DMA 使能时, 如果 FIFO 没满, I ² S 请求 PDMA 从 内存 向 发送 FIFO 传输数据 1 = 使能 TX DMA. 0 = 禁用 TX DMA.
[19]	CLR_RXFIFO	清除接收 FIFO 写 1 清除接收 FIFO, 内部指针复位指向 FIFO 的起始位置, RX_LEVEL[3:0] 返回 0, 接收 FIFO 变为空. 该位由硬件自动清零, 读时返回0.

[18]	CLR_TXFIFO	<p>清除发送 FIFO</p> <p>写 1 清除发送 FIFO，内部指针复位指向 FIFO 的起始位置，TX_LEVEL[3:0] 返回 0，发送 FIFO 变为空，但在发送 FIFO 中的数据不变。</p> <p>该位由硬件自动清零，读时返回0。</p>																		
[17]	LCHZCEN	<p>使能左声道过零检测</p> <p>如果该位置 1，当左声道数据符号位改变或下一个移位数据的数据位都为0，则寄存器 I2S_STATUS 的 LZCF 标志被置为 1。该功能仅工作在发送模式。</p> <p>1 = 使能左声道过零检测</p> <p>0 = 禁用左声道过零检测</p>																		
[16]	RCHZCEN	<p>使能右声道过零检测</p> <p>如果该位置 1，当右声道数据符号位改变或下一个移位数据的数据位都为0，则寄存器 I2S_STATUS 的 RZCF 标志被置为 1。该功能仅工作在接收模式。</p> <p>1 = 使能右声道过零检测。</p> <p>0 = 禁用右声道过零检测。</p>																		
[15]	MCLKEN	<p>使能主时钟</p> <p>使能主时钟MCLK向外部音频Codec设备输出。输出频率由I2S_CLKDIV寄存器的MCLK_DIV[2:0]设定。</p> <p>1 = 使能主时钟。</p> <p>0 = 禁用主时钟。</p>																		
[14:12]	RXTH[2:0]	<p>接收 FIFO 阈值水平</p> <p>当缓存中的接收字等于或大于阈值水平时，RXTHF 标志置位。</p> <table><tr><th>RXTH</th><th>描述</th></tr><tr><td>000</td><td>接收 FIFO 中有 1 字数据</td></tr><tr><td>001</td><td>接收 FIFO 中有 2 字数据</td></tr><tr><td>010</td><td>接收 FIFO 中有 3 字数据</td></tr><tr><td>011</td><td>接收 FIFO 中有 4 字数据</td></tr><tr><td>100</td><td>接收 FIFO 中有 5 字数据</td></tr><tr><td>101</td><td>接收 FIFO 中有 6 字数据</td></tr><tr><td>110</td><td>接收 FIFO 中有 7 字数据</td></tr><tr><td>111</td><td>接收 FIFO 中有 8 字数据</td></tr></table>	RXTH	描述	000	接收 FIFO 中有 1 字数据	001	接收 FIFO 中有 2 字数据	010	接收 FIFO 中有 3 字数据	011	接收 FIFO 中有 4 字数据	100	接收 FIFO 中有 5 字数据	101	接收 FIFO 中有 6 字数据	110	接收 FIFO 中有 7 字数据	111	接收 FIFO 中有 8 字数据
RXTH	描述																			
000	接收 FIFO 中有 1 字数据																			
001	接收 FIFO 中有 2 字数据																			
010	接收 FIFO 中有 3 字数据																			
011	接收 FIFO 中有 4 字数据																			
100	接收 FIFO 中有 5 字数据																			
101	接收 FIFO 中有 6 字数据																			
110	接收 FIFO 中有 7 字数据																			
111	接收 FIFO 中有 8 字数据																			
[11:9]	TXTH[2:0]	<p>发送 FIFO 阈值水平</p> <p>如果发送 FIFO 中剩下的数据字 (32 bits) 等于或小于阈值水平，则 TXTHF 标志置位。</p> <table><tr><th>TXTH</th><th>描述</th></tr><tr><td>000</td><td>发送 FIFO 中有 1 个字数据</td></tr><tr><td>001</td><td>发送 FIFO 中有 2 个字数据</td></tr><tr><td>010</td><td>发送 FIFO 中有 3 个字数据</td></tr><tr><td>011</td><td>发送 FIFO 中有 4 个字数据</td></tr></table>	TXTH	描述	000	发送 FIFO 中有 1 个字数据	001	发送 FIFO 中有 2 个字数据	010	发送 FIFO 中有 3 个字数据	011	发送 FIFO 中有 4 个字数据								
TXTH	描述																			
000	发送 FIFO 中有 1 个字数据																			
001	发送 FIFO 中有 2 个字数据																			
010	发送 FIFO 中有 3 个字数据																			
011	发送 FIFO 中有 4 个字数据																			

		100	发送 FIFO 中有 5 个字数据										
		101	发送 FIFO 中有 6 个字数据										
		110	发送 FIFO 中有 7 个字数据										
		111	发送 FIFO 中有 8 个字数据										
[8]	SLAVE	从机模式 I ² S 可工作于主机模式或从机模式。主机模式下，I2S_BCLK 和 I2S_LRCLK 管脚都为输出模式并输出 I2S_BCLK和I2S_LRCLK信号到音频 CODEC 芯片。从机模式下，I2S_BCLK 和 I2S_LRCLK 管脚都为输入模式，接收来自外部音频 CODEC 芯片的 I2S_BCLK 和 I2S_LRCLK 信号。 1 = 从机模式 0 = 主机模式											
[7]	FORMAT	数据格式 1 = MSB 对齐数据格式 0 = I ² S 数据格式											
[6]	MONO	单声道数据 1 = 数据为单声道格式，当单声道模式使能，从I2S总线上获取右声道数据。 0 = 数据为立体声格式											
[5:4]	WORDWIDTH[1:0]	字宽度 <table><tr><th>WORDWIDTH</th><th>描述</th></tr><tr><td>00</td><td>数据为 8 位</td></tr><tr><td>01</td><td>数据为 16 位</td></tr><tr><td>10</td><td>数据为 24 位</td></tr><tr><td>11</td><td>数据为 32 位</td></tr></table>		WORDWIDTH	描述	00	数据为 8 位	01	数据为 16 位	10	数据为 24 位	11	数据为 32 位
WORDWIDTH	描述												
00	数据为 8 位												
01	数据为 16 位												
10	数据为 24 位												
11	数据为 32 位												
[3]	MUTE	使能发送静音 1 = 发送数据 0 到发送通道。 0 = 发送缓存中的数据数据到发送通道。											
[2]	RXEN	接收使能 1 = 使能数据接收。 0 = 禁用数据接收。											
[1]	TXEN	发送使能 1 = 使能数据发送。 0 = 禁用数据发送。											
[0]	I2SEN	使能 I²S 控制器 1 = 使能。 0 = 禁用。											

I²S 时钟分频寄存器(I2S_CLKDIV)

寄存器	偏移量	R/W	描述	复位后的值
I2S_CLKDIV	I2S_BA + 0x04	R/W	I ² S时钟分频控制寄存器	0x0000_0000

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

Bits	描述	
[31:16]	-	保留
[15:8]	BCLK_DIV[7:0]	<p>位时钟分频</p> <p>如果 I²S 工作于主机模式，位时钟由该芯片提供。软件可以编程这些位产生采样率时钟频率。</p> $BCLK = I2SCLK / (2 \times (BCLK_DIV + 1))$
[7:3]	-	保留
[2:0]	MCLK_DIV[2:0]	<p>主时钟分频</p> <p>如果芯片的外部晶振频率为 $(2 \times MCLK_DIV) \times 256fs$，则软件可编程这些位产生 256fs 时钟频率到音频 CODEC 芯片。如果 MCLK_DIV 被置为 0，则 MCLK 与外部时钟输入相同</p> <p>例如，采样率为 48 kHz，芯片的外部晶振时钟为 12.288 MHz，设置 MCLK_DIV=0。</p> $MCLK = I2SCLK / (2 \times (MCLK_DIV))$

I²S中断使能寄存器(I2S_INTEN)

寄存器	偏移量	R/W	描述	复位后的值
I2S_INTEN	I2S_BA + 0x08	R/W	I ² S中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-			LZCIE	RZCIE	TXTHIE	TXOVFIE	TXUDFIE
7	6	5	4	3	2	1	0
-					RXTHIE	RXOVFIE	RXUDFIE

Bits	描述	
[31:13]	-	保留
[12]	LZCIE	左声道过零检测中断使能 如果该位设为 1 且左声道有过零信号, 则中断产生. 1 = 使能中断 0 = 禁用中断
[11]	RZCIE	右声道过零检测中断使能 如果该位设为 1 且右声道有过零信号, 则中断产生. 1 = 使能中断. 0 = 禁用中断.
[10]	TXTHIE	发送 FIFO 阈值水平中断使能 如果该位被设为1, 且发送 FIFO 中的数据字小于或等于 TXTH[2:0], 则中断发生. 1 = 使能中断. 0 = 禁用中断.
[9]	TXOVFIE	发送 FIFO 溢出中断使能 如果该位被设为1, 且发送 FIFO 的溢出标志被置为 1, 则中断发生 1 = 使能中断. 0 = 禁用中断.
[8]	TXUDFIE	发送 FIFO 下溢中断使能 如果该位被设为1, 且发送 FIFO 的下溢标志被置为 1, 则中断发生 1 = 使能中断.

		0 = 禁用中断.
[7:3]	-	保留
[2]	RXTHIE	接收 FIFO 阈值水平中断使能 如果该位被设为1, 且接收 FIFO 中的数据字大于 RXTH[2:0], 则中断发生. 1 = 使能中断. 0 = 禁用中断.
[1]	RXOVFIE	接收 FIFO 溢出中断使能 如果该位被设为1, 且接收 FIFO 的溢出标志被置为 1, 则中断发生 1 = 使能中断. 0 = 禁用中断.
[0]	RXUDFIE	接收 FIFO 下溢中断使能 如果该位被设为1, 且接收 FIFO 的下溢出标志被置为 1, 则中断发生. 1 = 使能中断. 0 = 禁用中断.

I²S 状态寄存器 (I2S STATUS)

寄存器	偏移量	R/W	描述	复位后的值
I2S_STATUS	I2S_BA + 0x0C	R/W	I ² S 状态寄存器	0x0014_1000

31	30	29	28	27	26	25	24
TX_LEVEL				RX_LEVEL			
23	22	21	20	19	18	17	16
LZCF	RZCF	TXBUSY	TXEMPTY	TXFULL	TXTHF	TXOVF	TXUDF
15	14	13	12	11	10	9	8
-			RXEMPTY	RXFULL	RXTHF	RXOVF	RXUDF
7	6	5	4	3	2	1	0
-				RIGHT	I2STXINT	I2SRXINT	I2SINT

Bits	描述	
[31:28]	TX_LEVEL[3:0]	发送 FIFO 水平 这些位表示发送 FIFO 中数据字的数目
		TX_LEVEL 描述
		0000 无数据
		0001 发送 FIFO 有 1 个字
		--- ---
[27:24]	RX_LEVEL[3:0]	接收 FIFO 水平 这些位表示接收 FIFO 中数据字的数目
		RX_LEVEL 描述
		0000 无数据
		0001 接收 FIFO 有 1 个字
		--- ---
[23]	LZCF	左声道过零标志 该位表示左声道下一个采样数据符号位改变或所有数据位为 0。 1 =检测到左声道过零 0 =没有过零 软件可以写 1 清除该位。

[22]	RZCF	<p>右声道过零标志</p> <p>该位表示右声道下一个采样数据符号位改变或所有数据位为 0.</p> <p>1 =检测到右声道过零</p> <p>0 =没有过零</p> <p>软件可以写 1 清除该位.</p>
[21]	TXBUSY	<p>发送忙</p> <p>当发送 FIFO 和移位缓存中的所有数据都被移出时, 该位清零。当第一个数据加载到移位缓存时, 该位置 1.</p> <p>1 = 发送移位缓存忙</p> <p>0 = 发送移位缓存为空</p> <p>该位只读.</p>
[20]	TXEMPTY	<p>发送 FIFO 空</p> <p>该位反映发送 FIFO 中的数据字数为 0</p> <p>1 = 非空</p> <p>0 = 空</p> <p>该位只读.</p>
[19]	TXFULL	<p>发送 FIFO 满</p> <p>发送 FIFO 满</p> <p>该位反映发送 FIFO 中的数据字数为8</p> <p>1 = 未滿.</p> <p>0 = 滿.</p> <p>该位只读</p>
[18]	TXTHF	<p>发送 FIFO 阈值标志</p> <p>当发送 FIFO 中的数据字数等于或低于 TXTH[2:0] 中设定的阈值时, TXTHF 位变为 1. 该位将保持 1 直到软件写数据到 TXFIFO 寄存器后 TXFIFO_LEVEL[3:0] 高于 TXTH[2:0].</p> <p>1 =发送FIFO中的数据字数等于或低于阈值水平</p> <p>0 =发送FIFO中的数据字数高于阈值水平</p> <p>该位只读</p>
[17]	TXOVF	<p>发送 FIFO 溢出标志</p> <p>当发送 FIFO 已滿时写数据到该 FIFO, 该位被置为 1</p> <p>1 = 溢出</p> <p>0 = 没有溢出</p> <p>软件可以写 1 清除该位.</p>
[16]	TXUDF	<p>发送 FIFO 下溢标志</p> <p>当发送 FIFO 为空时, 移位逻辑硬件从数据 FIFO 读数据, 会导致该位被置为1.</p> <p>1 = 下溢</p> <p>0 = 没有下溢</p> <p>软件可以写 1 清除该位.</p>

[15:13]	-	保留
[12]	RXEMPTY	接收 FIFO 空 该位反映接收 FIFO 中的数据字数为0 1 = 非空 0 = 空 该位只读.
[11]	RXFULL	接收 FIFO 满 该位反映接收 FIFO 中的数据字数为8 1 = 未滿. 0 = 滿. 该位只读
[10]	RXTHF	接收 FIFO 阈值标志 当接收 FIFO 中数据字等于或高于 RXTH[2:0] 中所设定的阈值时, RXTHF 位 变为 1。 该位将保持 1直到软件读 RXFIFO 寄存器后 RXFIFO_LEVEL[3:0] 低于 RXTH[2:0]. 1 = 接收FIFO 中数据字数等于或高于阈值水平 0 = 接收FIFO 中数据字数低于阈值水平 该位只读
[9]	RXOVF	接收 FIFO 溢出标志 当接收 FIFO 已滿, 接收硬件还尝试写数据到接收 FIFO, 则该位被置为1, 第一个缓存中数据被覆盖。 1 = 发生溢出 0 = 没发生溢出 软件可以写 1 清除该位.
[8]	RXUDF	接收 FIFO 下溢标志 当接收 FIFO 为空时, 读接收 FIFO, 则该位置 1, 表示发生下溢。 1 = 发生下溢 0 = 没发生下溢 软件可以写 1 清除该位.
[7:4]	-	保留
[3]	RIGHT	右声道 该位表示当前发送的数据属于右声道 1 = 右声道 0 = 左声道 该位只读
[2]	I2STXINT	I²S 发送中断 1 = 发送中断发生 0 = 无发送中断发生 该位只读

[1]	I2SRXINT	<p>I²S 接收中断</p> <p>1 = 接收中断发生</p> <p>0 = 无接收中断发生</p> <p>该位只读</p>
[0]	I2SINT	<p>I²S 中断标志</p> <p>1 = I²S 中断发生</p> <p>0 = 无 I²S 中断发生</p> <p>该位为 I2STXINT 和 I2SRXINT 位的“或”。</p> <p>该位只读。</p>

I²S 发送寄存器(I2S_TXFIFO)

寄存器	偏移量	R/W	描述	复位后的值
I2S_TXFIFO	I2S_BA + 0x10	W	I ² S 发送 FIFO	0x0000_0000

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

Bits	描述	
[31:0]	TXFIFO[31:0]	<p>发送 FIFO 寄存器</p> <p>I²S 包含 8 个字 (8x32-bit) 的数据缓存用于数据传送。写数据到该寄存器准备用于发送的数据。剩余的字数目由 I2S_STATUS 的 TX_LEVEL[3:0] 指示。该寄存器只写。</p>

I²S 接收 寄存器(I2S_RXFIFO)

寄存器	偏移量	R/W	描述	复位后的值
I2S_RXFIFO	I2S_BA + 0x14	R	I ² S 接收 FIFO	0x0000_0000

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

Bits	描述	
[31:0]	RXFIFO[31:0]	接收 FIFO 寄存器 I ² S 包含 8 个字 (8x32-bit) 的数据缓存用于数据接收。读该寄存器获取 FIFO 中的数据。剩下的数据字数目由 I2S_STATUS 寄存器的 RX_LEVEL[3:0] 指示。该寄存器只读。

5.13 LCD 显示驱动器

5.13.1 概述

该LCD驱动器可以通过自动地创建交流段和公共电压信号来直接驱动LCD面板。它可以支持静态，1/2，1/3，1/4，1/5 和 1/6 占空比，6x38 (segment 0 作为LCD_COM4、segment 1 作为 LCD_COM5) 或者 4x40(LCD_COM0 ~ LCD_COM3) 的LCD面板。

一个内建的电荷泵功能能够被使能用于提供给 LCD 面板比系统电压更高的电压。LCD 驱动器能产生比阈值电压更高的电压，以便使段变暗，比阈值电压低则可以使段变亮。但是，如果加载的电压有直流成分，则LCD的显示段将退化。为避免这种情况，LCD 驱动器产生的波形被改编成作用于每个段的电压是平均值为 0 的均方根 (RMS(root-mean-square)) 电压，比段阈值电压低的 RMS 电压作用在 LCD 段使得LCD变亮，比段阈值电压高的 RMS 电压作用在 LCD 段使得LCD变暗。

注：ADC与LCD共享的管脚其输出电压不可高于 VDD

(LQFP64 : LCD_SEG17, LCD_SEG19, LCD_SEG20, LCD_SEG21, LCD_SEG22, LCD_SEG23)

(LQFP128 : LCD_SEG36, LCD_SEG37, LCD_SEG38, LCD_SEG39)

5.13.2 特性

- LQFP64 封装支持 180 个点 (6x30) 或者 128 个点 (4x32)，LQFP100/LQFP128 支持 228 个点 (6x38) 或者 160 个点 (4x40) Segment/Com 管脚：
- Common 0-5 与GPIO 管脚多路复用。
- Segment 0-39 与GPIO 管脚多路复用。
- 支持静态，1/2 偏压和 1/3 偏压电压。
- 六种显示模式：静态，1/2 占空比，1/3 占空比，1/4 占空比，1/5 占空比或 1/6 占空比。
- LCD 频率通过频率分频可选择
- 可配置的帧频率
- 内部电荷泵，可调的对比度调整
- 嵌入式LCD 偏压参考梯形(R-Type, 200kohm resisters)
- 可配置电荷泵频率
- 闪烁功能
- 支持 R/C-type 方式。
- LCD 帧中断

5.13.3 框图

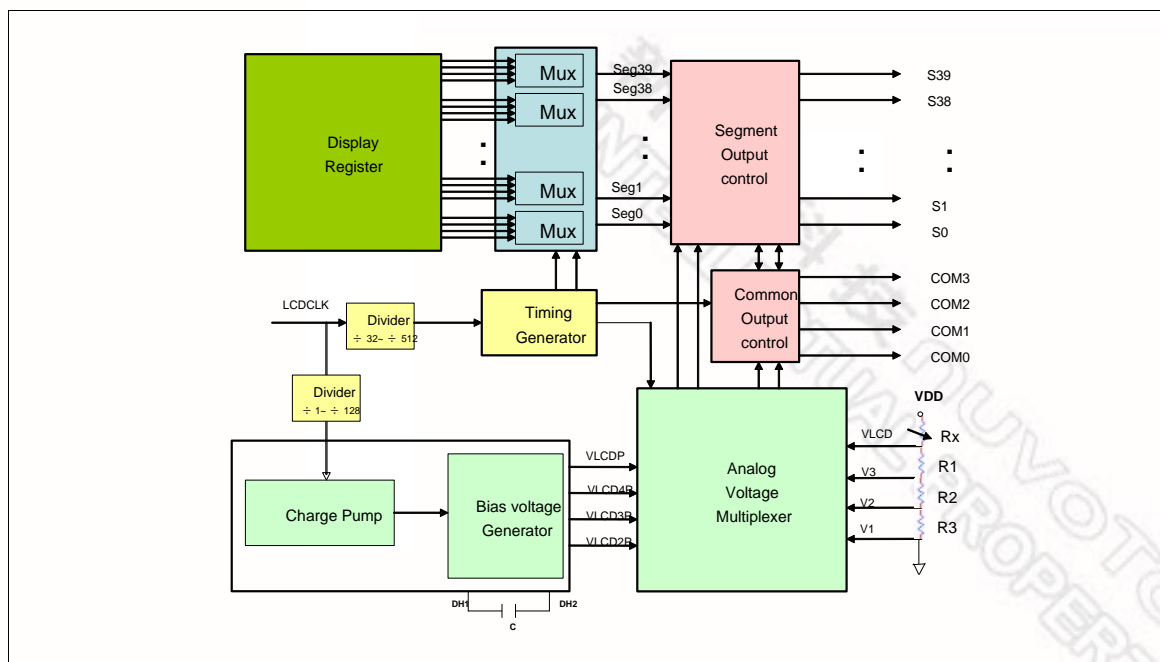


图 5-59 LCD 驱动器框图

5.13.4 功能描述

该LCD 驱动器由显示存储寄存器，段输出控制，公共输出，时序发生器，电荷泵和模拟电压多路转换模块。显示存储寄存器保存 LCD 段暗数据或亮数据。保存在显示存储寄存器中的数据位为“1”将使得 LCD 段为暗，数据位为“0”使得 LCD 段为亮。显示存储寄存器配置为LCD_MEM_0 ~ LCD_MEM_9 寄存器。对 LCD_MEM_0 ~ LCD_MEM_9 寄存器的数据位进行编程能使得相应的LCD 段变暗或变亮。保存在显示存储寄存器中的数据随时序信号发生器模块产生的时钟被依次多路复用到段输出模块。段输出模块负责产生 SEG 0 ~ SEG 39 驱动行，公共输出模块负责产生 COM0 ~ COM3 驱动行。电荷泵模块提供升压电压功能用于 LCD 面板。电荷泵输入电压范围是从 1.8V 到 3.6V。多级偏压电压能够通过设置LCD_DISPCTL 寄存器的 CPUMP_VOL_SET 位进行编程，从 2.6V 到3.3 V 的多级偏压电压可以由电荷泵模块产生。模拟电压多路器可以通过设置 LCD_DISPCTL 寄存器的 BIAS_SEL 位产生静态，1/2 偏压和 1/3 偏压电压输出。用户可以编程 BIAS_SEL 位来产生不同偏压电压用于COM 和 SEG 驱动线驱动 LCD 面板。每个公共信号依据其帧周期指定的时间片依次被选择。例如：在 1/3 占空比情况下，COM0 到 COM2 将输出波形，COM3 将被拉低。然而在 1/6 占空比情况下，COM0 到 COM5 将输出波形。COM 信号的波形如图 5-61 所示。

5.13.4.1 LCD 显示存储映射

DISPLAY Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM	X	X	COM5	COM4	COM3	COM2	COM1	COM0	X	X	COM5	COM4	COM3	COM2	COM1	COM0	X	X	COM5	COM4	COM3	COM2	COM1	COM0	X	X	COM5	COM4	COM3	COM2	COM1	COM0
LCD_MEM_9	X	X	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	X	X	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	X	X	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	X	X	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36
LCD_MEM_8	X	X	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	X	X	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	X	X	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	X	X	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32
LCD_MEM_7	X	X	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	X	X	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	X	X	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	X	X	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCD_MEM_6	X	X	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	X	X	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	X	X	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	X	X	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCD_MEM_5	X	X	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	X	X	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	X	X	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	X	X	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCD_MEM_4	X	X	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	X	X	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	X	X	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	X	X	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCD_MEM_3	X	X	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	X	X	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	X	X	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	X	X	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCD_MEM_2	X	X	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	X	X	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	X	X	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	X	X	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08
LCD_MEM_1	X	X	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	X	X	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	X	X	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	X	X	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04
LCD_MEM_0	X	X	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	X	X	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	X	X	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	X	X	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00

图 5-60 LCD 存储映射

5.13.4.2 帧计数器 (FC) 和闪烁显示

在 6-COM 配置的情况下，COM0, COM1, COM2, COM3, COM4 和 COM5 组成一帧。在 5-COM 配置的情况下，COM0, COM1, COM2, COM3 和 COM4 组成一帧。在 4-COM 配置的情况下，COM0, COM1, COM2 和 COM3 组成一帧。在 3-COM 配置的情况下，COM0, COM1 和 COM2 组成一帧。在 2-COM 配置的情况下，COM0 和 COM1 组成一帧。在 静态配置的情况下，COM0 组成一帧。帧计数器能够通过编程预分频计数器(LCD_FCR[PRESC]) 进行预分频。预分频计数器能够除以1, 2, 4 和 8. 帧计数器从 FCV (LCD_FCR[9:4]) 向下计数到 0. FCV 是帧计数器的峰值。如果 FCEN (LCD_FCR[0]) 和 FCINTEN (LCD_FCR[1]) 被设置为 1, 一旦 帧计数器向下计数到 0, 帧计数器溢出中断产生。同时，帧计数器使用 FCV 的值重新装载。LCD闪烁显示由帧计数器溢出时间控制。在闪烁配置情况下，LCD 段随着帧计数器溢出时间交替打开和关闭。帧计数器溢出中断也可以用于填充数据到 LCD 显示存储寄存器的同步。

5.13.4.3 LCD 显示掉电

如果掉电请求从系统管理器被触发，LCD 控制器将完整执行帧，以避免直流成分。当帧被完全执行后，如果 PDINT_EN (LCD_CTL[9]) 被使能，则LCD 的掉电中断信号将产生，用于通知系统管理器 LCD控制器已经准备好进入掉电状态。反之，如果 PDINT_EN (LCD_CTL[9]) 被禁用，则 LCD 掉电中断信号将会被屏蔽，中断将会被禁用。如果 PDDSIP_EN (LCD_CTL[8]) 被置为1, LCD 在掉电模式下显示，反之，如果 PDDSIP_EN (LCD_CTL[8]) 被清为0, 则 LCD 显示在掉电模式下被关闭。

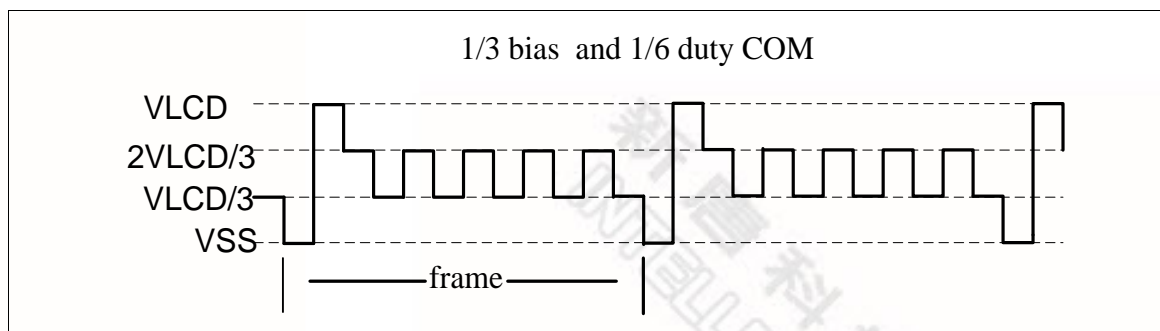


图 5-61 COM 信号波形

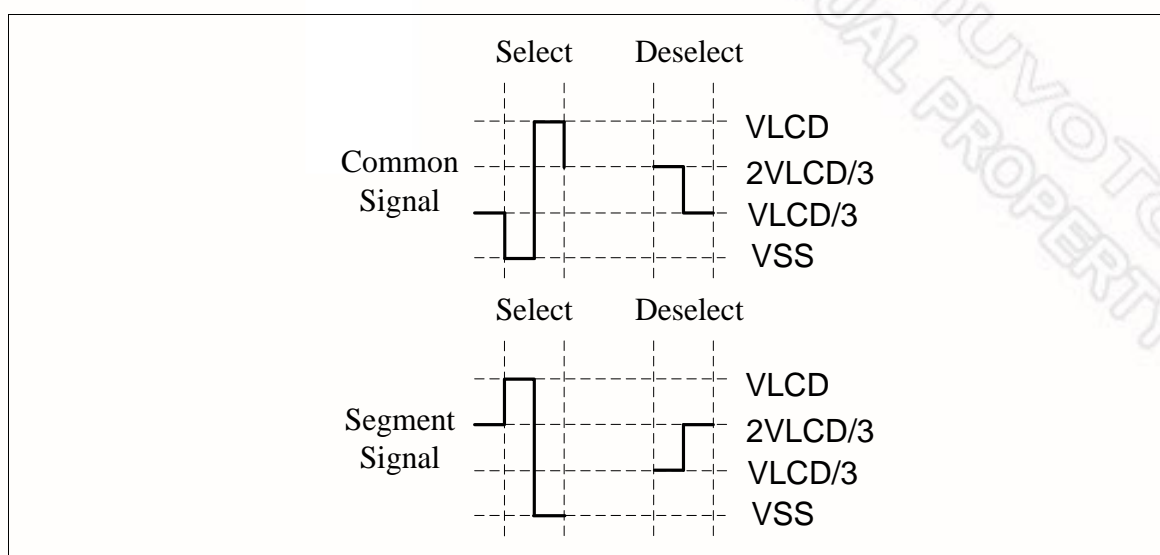


图 5-62 SEG 信号波形

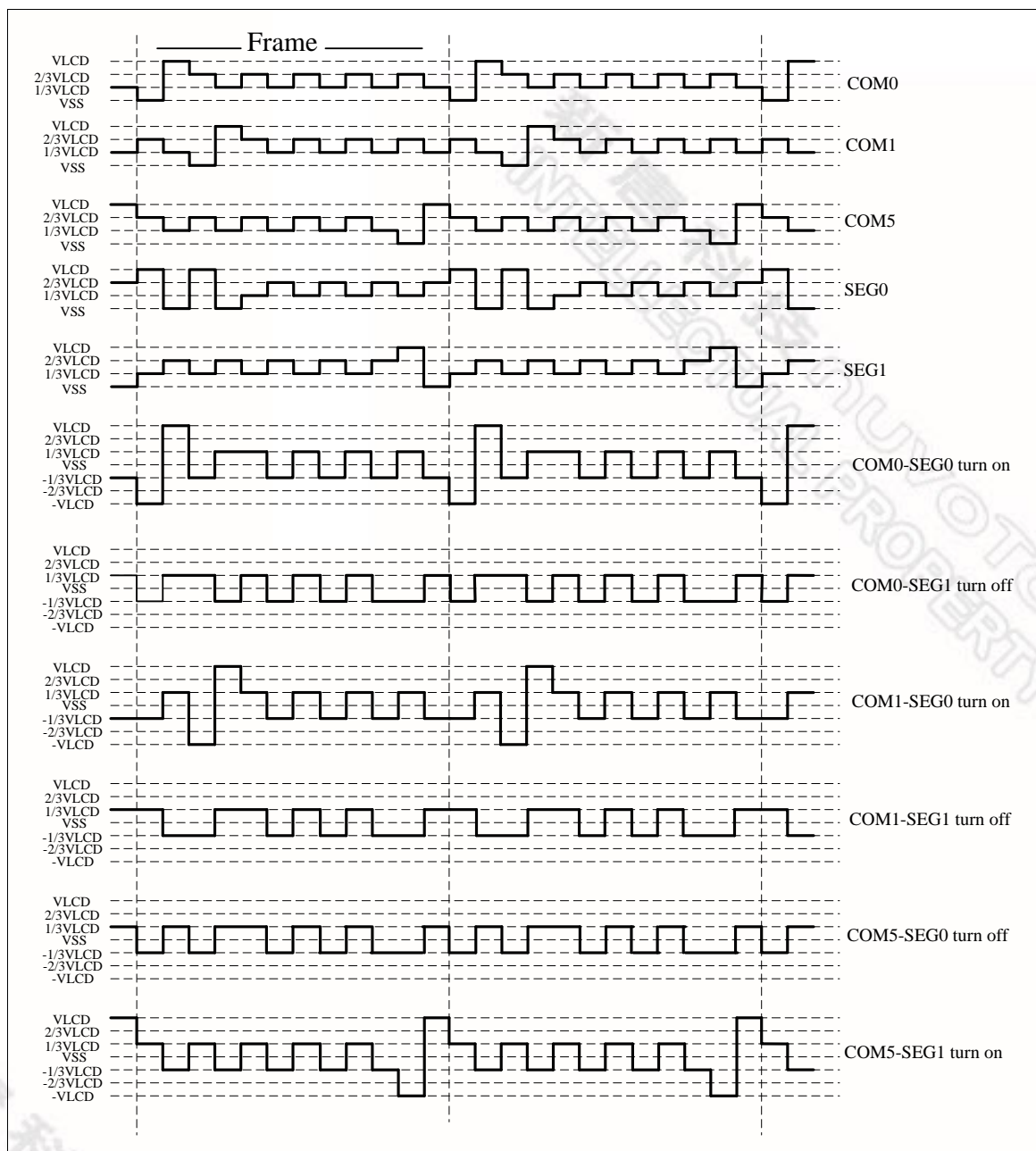


图 5-63 COM-SEG 信号波形，1/6 占空比，1/3 偏压

5.13.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
LCD_BA = 0x400B_0000				
LCD_CTL	LCD_BA+0x00	R/W	LCD 控制寄存器	0x0000_0000
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD 显示控制寄存器	0x0000_0000
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~ SEG0 数据	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~ SEG4 数据	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~ SEG8 数据	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15 ~ SEG12 数据	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~ SEG16 数据	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~ SEG20 数据	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~ SEG24 数据	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~ SEG28 数据	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~ SEG32 数据	0x0000_0000
LCD_MEM_9	LCD_BA+0x2C	R/W	LCD SEG39 ~ SEG36 数据	0x0000_0000
LCD_FCR	LCD_BA+0x30	R/W	LCD 帧计数器控制寄存器	0x0000_0000
LCD_FCSTS	LCD_BA+0x34	R/W	LCD 帧计数器状态	0x0000_0000

5.13.6 寄存器描述

LCD 控制寄存器 (LCD_CTL)

寄存器	偏移量	R/W	描述	复位后的值
LCD_CTL	LCD_BA+0x00	R/W	LCD 控制寄存器	0x0000_0000

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

Bits	描述	
[31:10]	Reserved	保留
[9]	PDINT_EN	掉电中断使能 如果掉电请求从系统管理器被触发，LCD 控制器将完整执行帧，以避免直流成分。当帧被完全执行后，如果 PDINT_EN 被使能，则LCD 的掉电中断信号将产生，用于通知系统管理器 LCD控制器已经准备好进入掉电状态。反之，如果 PDINT_EN 被设为 0，则LCD 掉电中断信号将会被屏蔽，中断将会被禁用。. 0 = 禁用掉电中断 1 = 使能掉电中断
[8]	PDDISP_EN	掉电显示使能 通过设置PDDISP_EN，LCD 能够被编程为在掉电状态下能够显示与否。. 0 = 禁用掉电模式下的 LCD 显示（LCD 不点亮） 1 = 使能掉电模式下的 LCD 显示（LCD 保持显示）
[7]	BLINK	LCD 闪烁使能 0 = 禁用闪烁. 1 = 使能闪烁.
[6:4]	FREQ	LCD 频率选择 000 = LCDCLK 除以 32 001 = LCDCLK 除以 64 010 = LCDCLK 除以 96 011 = LCDCLK 除以 128

		100 = LCDCLK 除以 192 101 = LCDCLK 除以 256 110 = LCDCLK 除以 384 111 = LCDCLK 除以 512																													
[3:1]	MUX	<p>多路选择</p> <p>000 = 静态</p> <p>001 = 1/2 占空比</p> <p>010 = 1/3 占空比</p> <p>011 = 1/4 占空比</p> <p>100 = 1/5 占空比</p> <p>101 = 1/6 占空比</p> <p>110 = 保留</p> <p>111 = 保留</p> <p>注：</p> <p>在 Nano110 与 Nano130 芯片系列中用户不须设置 PD_H_MFP 位域，而仅需设置 MUX 位域使LCD_SEG0 与 LCD_SEG1 切换至 LCD_COM4 与 LCD_COM5。</p> <table><tr><th colspan="5">Nano110 与 Nano130 芯片系列中藉由设置MUX位域改变 LCD_SEG0(LCD_COM4) 与 LCD_SEG1(LCD_COM5) 管脚定义</th></tr><tr><th rowspan="2">MUX</th><th colspan="2">LQFP128</th><th colspan="2">LQFP64</th></tr><tr><th>SEG0(COM4) PD.15</th><th>SEG1(COM5) PD.14</th><th>SEG0(COM4) PB.1</th><th>SEG1(COM5) PB.0</th></tr><tr><td>0b000, 0b001 0b010, 0b011</td><td>LCD_SEG0</td><td>LCD_SEG1</td><td>LCD_SEG0</td><td>LCD_SEG1</td></tr><tr><td>0b100</td><td>LCD_COM4</td><td>LCD_SEG1</td><td>LCD_COM4</td><td>LCD_SEG1</td></tr><tr><td>0b101</td><td>LCD_COM4</td><td>LCD_COM5</td><td>LCD_COM4</td><td>LCD_COM5</td></tr></table>	Nano110 与 Nano130 芯片系列中藉由设置MUX位域改变 LCD_SEG0(LCD_COM4) 与 LCD_SEG1(LCD_COM5) 管脚定义					MUX	LQFP128		LQFP64		SEG0(COM4) PD.15	SEG1(COM5) PD.14	SEG0(COM4) PB.1	SEG1(COM5) PB.0	0b000, 0b001 0b010, 0b011	LCD_SEG0	LCD_SEG1	LCD_SEG0	LCD_SEG1	0b100	LCD_COM4	LCD_SEG1	LCD_COM4	LCD_SEG1	0b101	LCD_COM4	LCD_COM5	LCD_COM4	LCD_COM5
Nano110 与 Nano130 芯片系列中藉由设置MUX位域改变 LCD_SEG0(LCD_COM4) 与 LCD_SEG1(LCD_COM5) 管脚定义																															
MUX	LQFP128		LQFP64																												
	SEG0(COM4) PD.15	SEG1(COM5) PD.14	SEG0(COM4) PB.1	SEG1(COM5) PB.0																											
0b000, 0b001 0b010, 0b011	LCD_SEG0	LCD_SEG1	LCD_SEG0	LCD_SEG1																											
0b100	LCD_COM4	LCD_SEG1	LCD_COM4	LCD_SEG1																											
0b101	LCD_COM4	LCD_COM5	LCD_COM4	LCD_COM5																											
[0]	EN	<p>LCD 使能</p> <p>0 = 禁用 LCD 控制器操作.</p> <p>1 = 使能 LCD 控制器操作.</p>																													

LCD 显示控制寄存器 (LCD_DISPCTL)

寄存器	偏移量	R/W	描述	复位后的值
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD 显示控制寄存器	0x0000_0000

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

Bits	描述	
[31:14]	Reserved	保留
[13:11]	CPUMP_FREQ	电荷泵频率选择: 000 = LCDCLK 001 = LCDCLK/2 010 = LCDCLK/4 011 = LCDCLK/8 100 = LCDCLK/16 101 = LCDCLK/32 110 = LCDCLK/64 111 = LCDCLK/128
[10:8]	CPUMP_VOL_SET	电荷泵电压选择 000 = 2.7V 001 = 2.8V 010 = 2.9V 011 = 3.0V 100 = 3.1V 101 = 3.2V 110 = 3.3V 111 = 3.4V
[7]	-	保留
[6]	BV_SEL	偏压类型选择 0 = C-Type 偏压模式。偏压源是内部偏压发生器

		1 = R-Type 偏压模式。偏压源是外部偏压发生器 注: 外部梯形电阻必须与 V1, V2, V3 和 VSS引脚连接。VLCD引脚也必须连接到VDD.
[5]	-	保留
[4]	IBRL_EN	内部偏压参考梯形使能 0 = 禁用偏压参考梯形. 1 = 使能偏压参考梯形
[3]	-	保留
[2:1]	BIAS_SEL	偏压选择 00 = 静态 01 = 1/2 偏压 10 = 1/3 偏压 11 = 保留
[0]	CPUMP_EN	电荷泵使能 0 = 禁用. 1 = 使能.

LCD MEM x 寄存器 (LCD MEMORY x) (x= 0 ~ 9)

寄存器	偏移量	R/W	描述	复位后的值
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~SEG0 数据	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~SEG4 数据	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~SEG8 数据	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15~SEG12 数据	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~SEG16 数据	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~SEG20 数据	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~SEG24 数据	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~SEG28 数据	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~SEG32 数据	0x0000_0000
LCD_MEM_9	LCD_BA+0x2C	R/W	LCD SEG39 ~SEG36 数据	0x0000_0000

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

Bits	描述	
[31:30]		保留
[29:24]	SEG(3+4x) data	SEG(3+4x) 数据 (x= 0 ~ 9)
[23:22]	-	保留
[21:16]	SEG(2 +4x) data	SEG(2+4x) 数据 (x= 0 ~ 9)
[17:15]	-	保留
[14:8]	SEG(1+4x) data	SEG(1 +4x) 数据 (x= 0 ~ 9)
[7:6]	-	保留
[5:0]	SEG(0+4x) data	SEG(0+4x) 数据 (x= 0 ~ 9)

LCD 帧计数器寄存器 (LCD_FCR)

寄存器	偏移量	R/W	描述	复位后的值
LCD_FCR	LCD_BA+0x30	R/W	LCD 帧计数器控制寄存器	0x0000_0000

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

Bits	描述	
[31:10]	-	保留
[9:4]	FCV	帧计数器峰值 该 6-位为帧计数器的峰值。
[3:2]	PRESCL	帧计数器预分频值 00 = CLKframe/1 01 = CLKframe/2 10 = CLKframe/4 11 = CLKframe/8
[1]	FCINTEN	LCD 帧计数器中断使能 0 = 帧计数器中断禁用. 1 = 帧计数器中断使能.
[0]	FCEN	LCD 帧计数器使能 0 = 禁用. 1 = 使能

LCD 帧计数器状态寄存器 (LCD_INTSTS)

寄存器	偏移量	R/W	描述	复位后的值
LCD_FCSTS	LCD_BA+0x2C	R/W	LCD 帧计数器状态	0x0000_0000

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

Bits	描述	
[31:2]	Reserved	保留
[1]	PDSTS	<p>掉电中断状态</p> <p>0 = 通知系統管理員LCD控制器未準備進入至掉電狀態，直到掉電模式被设起且一個幀未被完全執行，此位會變為1。</p> <p>1 =通知系統管理員LCD控制器已準備進入至掉電狀態，當掉電模式被设起且一個幀已被完全執行。</p>
[0]	FCSTS	<p>LCD 帧计数器状态</p> <p>0 = 帧计数器值未达到 FCV （帧计数峰值）。</p> <p>1 = 帧计数器值达到 FCV （帧计数峰值）。如果 FCINTEN 使能，帧计数器溢出中断产生</p>

5.13.7 应用电路

外部梯形电阻

1. 常用于高 VDD 电压.
2. 使用低成本电阻来创建多级 LCD 电压。无论被激励的像素点的数量上多少，电流值保持恒定。无论是内部还是外部，VLCD 的电压通常被连接到 VDD。
- 3 电阻的值决定于两个要素：
 - a. 显示品质
 - b. 耗电量

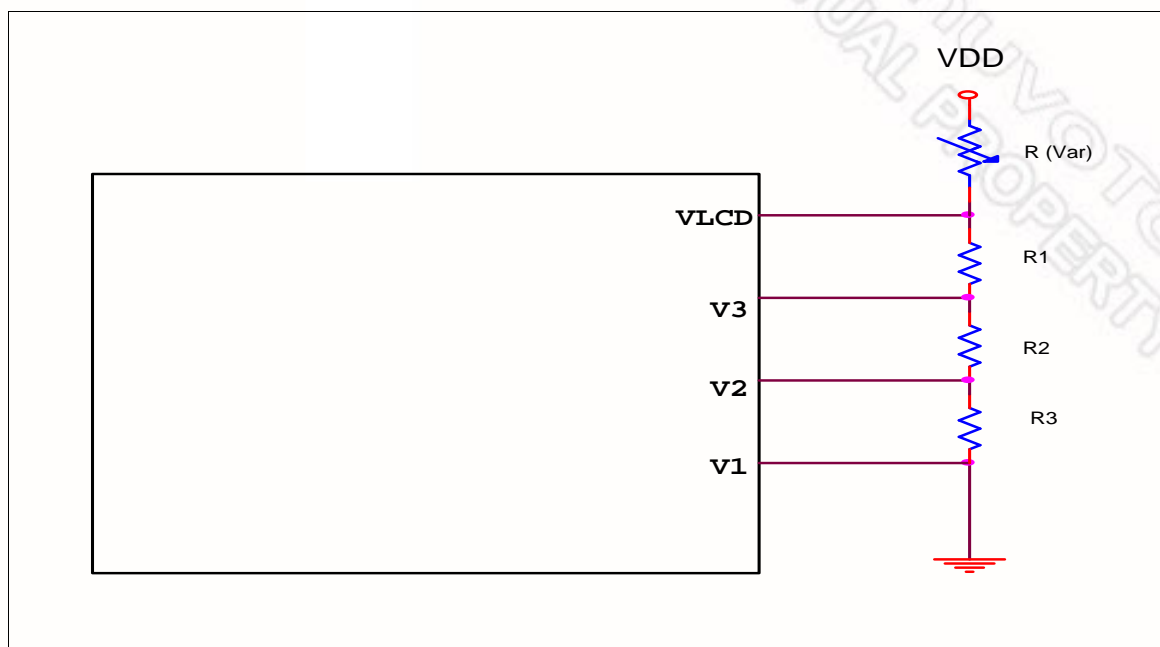


图 5-64 1/3 偏压 (外部 梯形电阻)

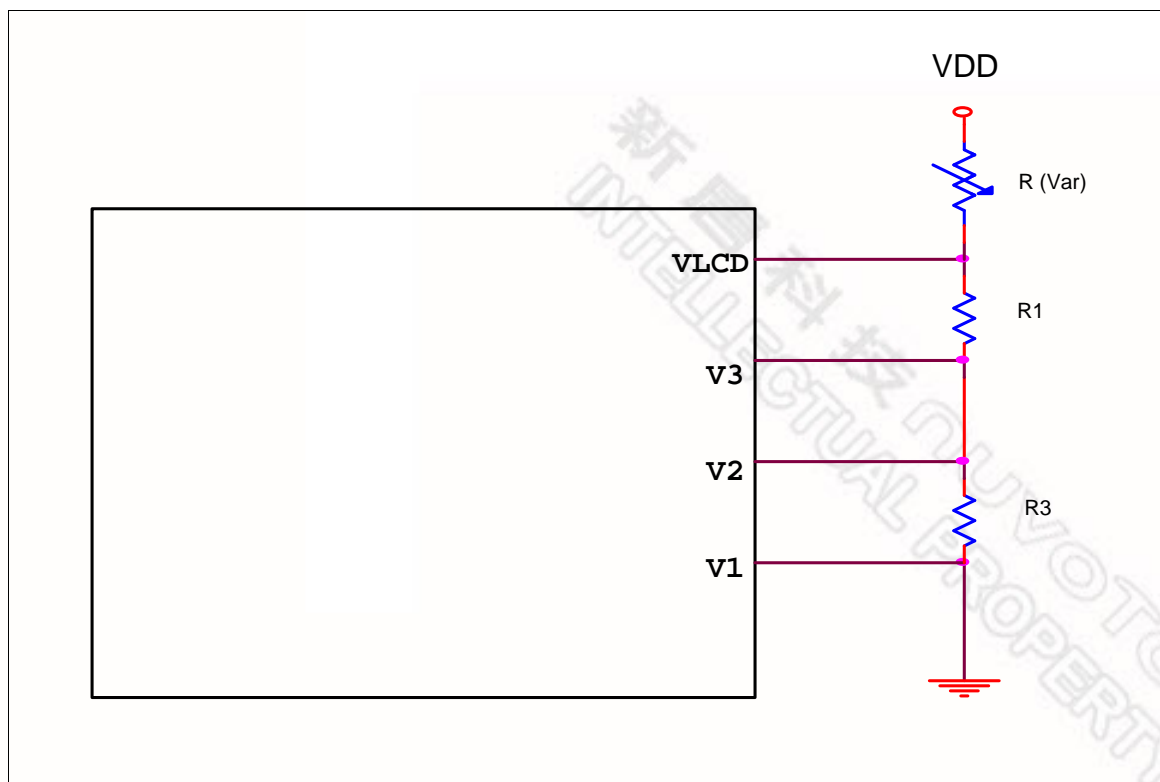


图 5-65 1/2 偏压 (外部 梯形电阻)

带电容的梯形电阻

有时给电阻增加并联电容可以减少因充/放电导致的失真。这种效果是有限的，因为在某些时候大电阻和大电容引起的电压偏移，会影响显示质量。

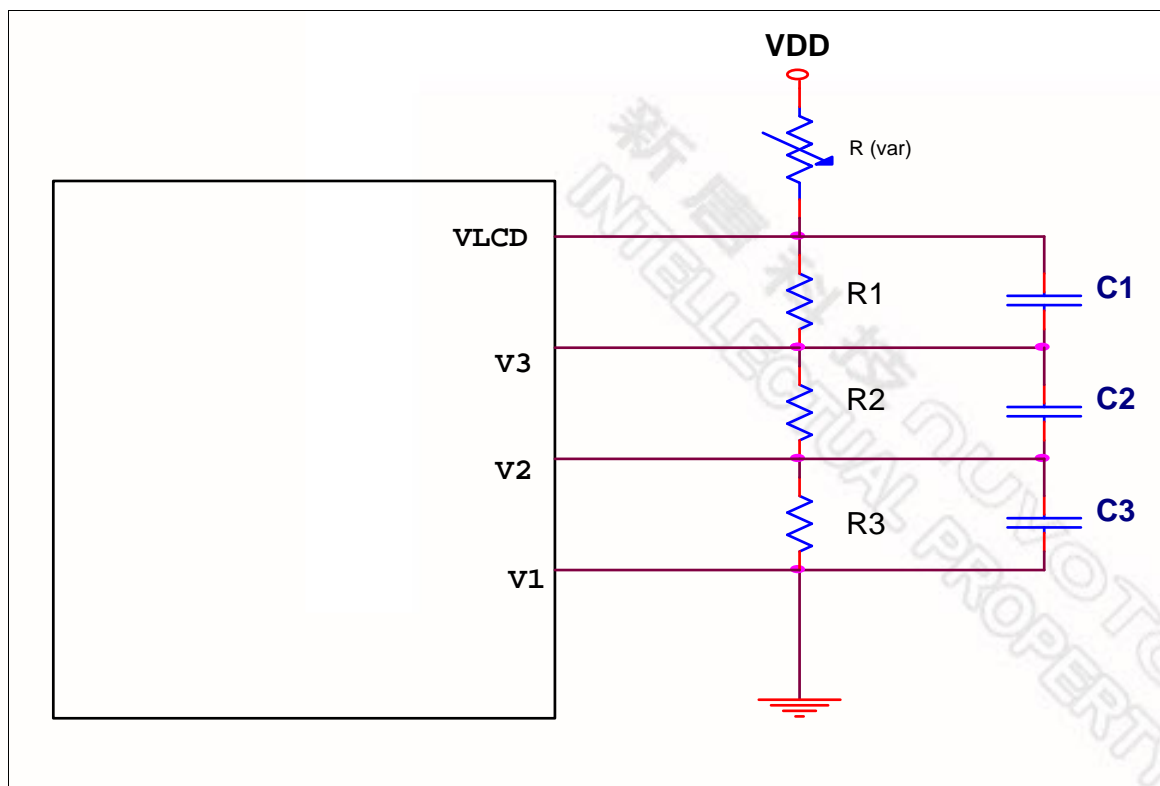


图 5-66 1/3 偏压 (带电容的梯形电阻)

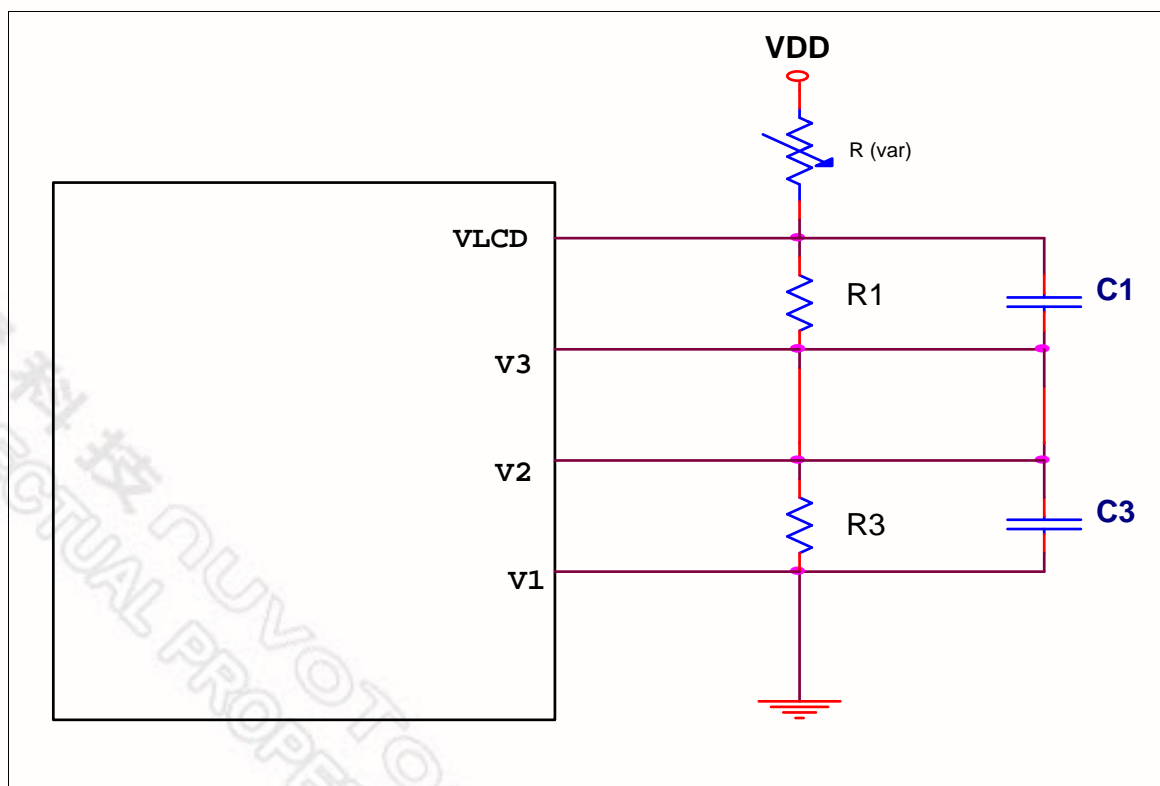


图 5-67 1/2 偏压 (带电容的梯形电阻)

电荷泵

1. 电荷泵是对低电压电池操作的理想应用，因为 VDD 电压能够提高到驱动 LCD 显示控制板。
2. 对于每一个 LCD 电压，电荷泵需要一个充电电容器和过滤电容器。
3. 这些电容是典型的聚酯，聚丙烯或聚苯乙烯材料。
4. 另一个使得电荷泵是个理想的电池应用的特性是电流消耗是和被激励的像素数目成正比的。

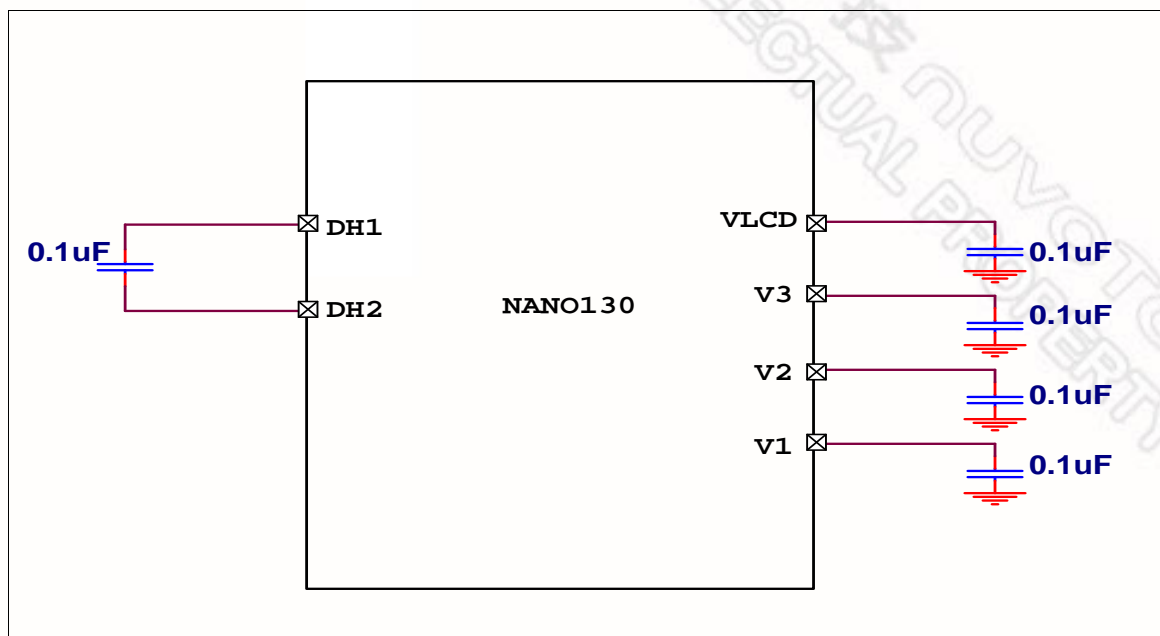


图 5-68 1/3 偏压 (电荷泵)

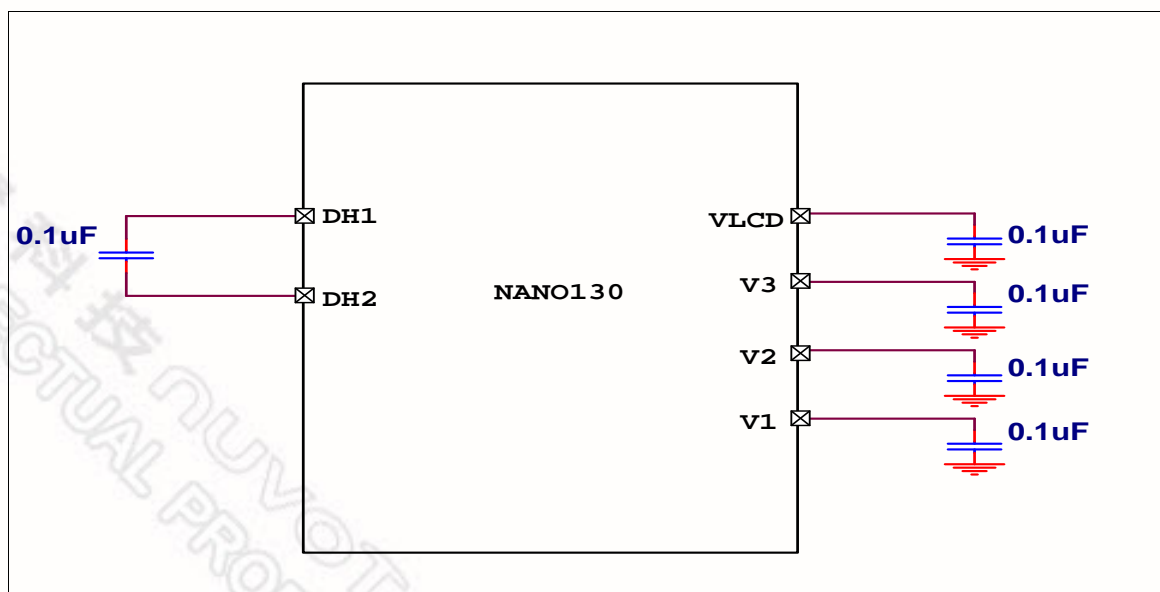


图 5-69 1/2 偏压 (电荷泵)

5.14 脉宽调制(PWM)

5.14.1 概述

该芯片有 2 组 PWM 控制器，每组有 4 个独立的 PWM 输出，CH0~CH3，或者作为两个带有可编程死区发生器的互补的PWM对，(CH0, CH1)，(CH2, CH3)。

每两个PWM输出，(CH0, CH1)，(CH2, CH3)，共享同一个8-位预分频，同一个提供5级分频 (1, 1/2, 1/4, 1/8, 1/16) 的时钟分频器。每个 PWM 输出有一个用于 PWM 周期控制的独立的 16 位 PWM 下数型计数器，和一个用于 PWM 占空比控制的 16 位比较器。每一个死区发生器有两个输出，第一个死区发生器的输出是 CH0 和 CH1，而第二个死区发生器的输出是 CH2 和 CH3。该两组PWM控制器一共提供 8 个独立的PWM中断标志，当某一个 PWM 周期下数计数器计数到 0 时，相应的中断标志由硬件置位。当 PWM 中断源和相应的中断使能位有效时，PWM 中断将会被触发。每一个 PWM 输出都可以被配置为单次模式来产生仅一个 PWM 周期的信号，或者连续模式来连续输出 PWM 波形。

当 PWMx_CTL 寄存器的 DZEN01 位被设置为 1 时，CH0 与 CH1 执行互补的 PWM 对功能，这一对 PWM 的时序，周期，占空比 和死区时间由 PWM 通道 0 定时器和死区发生器 0 决定。同样，当 PWMx_CTL 寄存器的 DZEN23 位被设置为 1 时，互补的 PWM 对 (CH2, CH3) 由 PWM 通道 2 控制。

为防止 PWM 驱动输出引脚输出不稳定波形，16 位周期下数型计数器和 16 位比较器均采用双缓存，当用户向计数器/比较器缓存寄存器内写入值时，只有当下数型计数器的值计数到 0 时，更新的值才会被重新载入下数型计数器/比较器。该双缓存特性可以避免 PWM 输出时产生干扰波形。

当 16 位下数型计数器达到 0 时，中断请求产生。如果 PWM 输出被设置为连续模式，当下数计数器计数到 0 时，下数计数器会重复自动重新装载 PWMx_DUTYy(y=0~3)寄存器中 CN 的值，并开始减计数；如果 PWM 输出被设置为单次模式，当下数计数器计数到 0 时，停止计数，并产生一个中断请求。

PWM 比较器用于脉冲宽度调制，当下数计数器的值与比较寄存器的值匹配时，计数器控制逻辑改变 PWM 输出电平。

PWM 的另一个特性是数字输入捕捉功能。若捕捉功能使能，PWM 输出管脚被切换到捕捉输入模式。捕捉通道 0 和 PWM CH0 共享同一个定时器；捕捉器通道 1 和 PWM CH1 共同一个定时器，以此类推。因此用户在使能捕捉功能之前，必须预先配置 PMW 定时器。捕捉功能使能后，当输入通道有上升沿时，捕捉器总是将 PWM 定时器的值锁存到捕捉上升沿锁存寄存器 (PWMx_CRLy, y=0~3)；而当输入通道有下降沿时，捕捉器总是将 PWM 计数器的值锁存到捕捉下降沿锁存寄存器 (PWMx_CFLy, y=0~3)。捕捉器通道 0 中断可以通过编程 PWMx_CAPINTEN 设定。无论何时，当捕获事件被锁存到通道 0/1/2/3，如果在 CAPCTL 中相应的重载使能位被置位，PWM定时器 0/1/2/3 将会同时被重新装载。

最大的捕捉频率由捕捉中断延迟决定。当捕捉中断发生时，软件至少执行以下三步：第一步，读 PWMINTSTS 获取中断源，第二步，读 PWMx_CRLy/PWMx_CFLy(y=0~3) 获取捕捉值，以及最后写 1 清零 PWMx_INTSTS。如果中断延迟花 T0 完成，捕捉信号在 (T0) 间隔内必须不能改变。此条件下，最大捕捉频率为1/T0。

5.14.2 特性

5.14.2.1 PWM 功能:

- 2 组 PWM 控制器，每组控制器有 4 个独立的 PWM 输出，CH0~CH3，或者 2 个带可编程死区发生器的互补的PWM对，(CH0, CH1)，(CH2, CH3)。
- 多达 8 个PWM通道或者 4 对 PWM 通道

- 高达 16 位的 PWM 计数器宽度
- PWM 中断请求与 PWM 周期同步
- 单次或连续模式
- 四个死区发生器

5.14.2.2 捕捉功能:

- 与 PWM 定时器共享时序控制逻辑
- 8 个捕捉输入通道与 8 个 PWM 输出通道共享
- 每个通道支持 1 个上升沿锁存寄存器 (PWMx_CRLy)，一个下降沿锁存寄存器 (PWMx_CFLy) 和捕捉中断标志 (CAPIFy)，x=0~1, y=0~3.
- 针对 8 个捕获通道，有 8 个 16 位计数器；或者当级联功能 (当 CH01CASKEN 位被置位，原来通道 1 的 16 位计数器将会与通道 0 的 16 位计数器连接起来作为通道 0 的输入捕捉计数器，CH23CASKEN 被置位时，通道 2, 3 的用法与通道 0, 1 一致) 被使能时，针对 4 个捕捉通道，有 4 个 32 位计数器。
- PWMx 的通道 0, 2 支持 PDMA 传输功能

5.14.3 框图

以下图形以组的形式说明 PWM 的架构。(Timer 0&1 为一组, Timer 2/3 为另一组, 依此类推)。

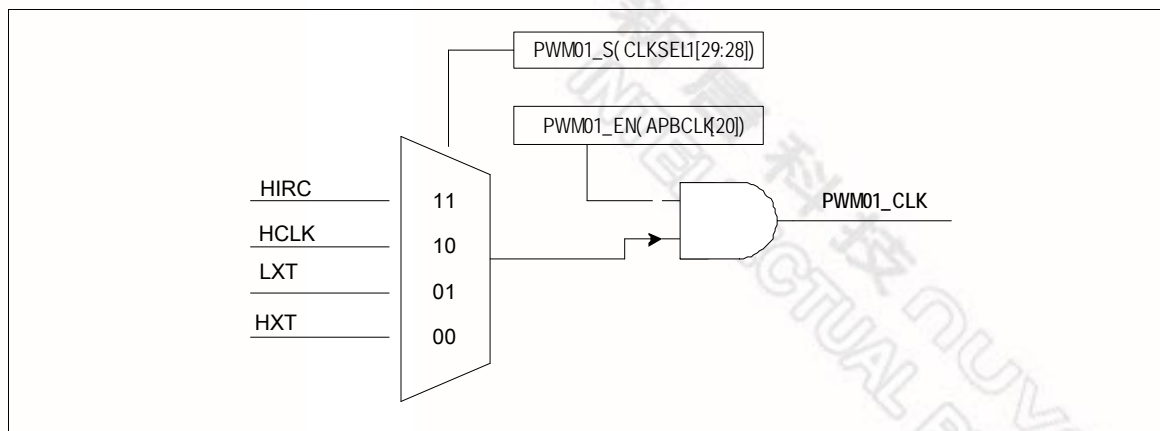


图 5-70 PWM0 时钟

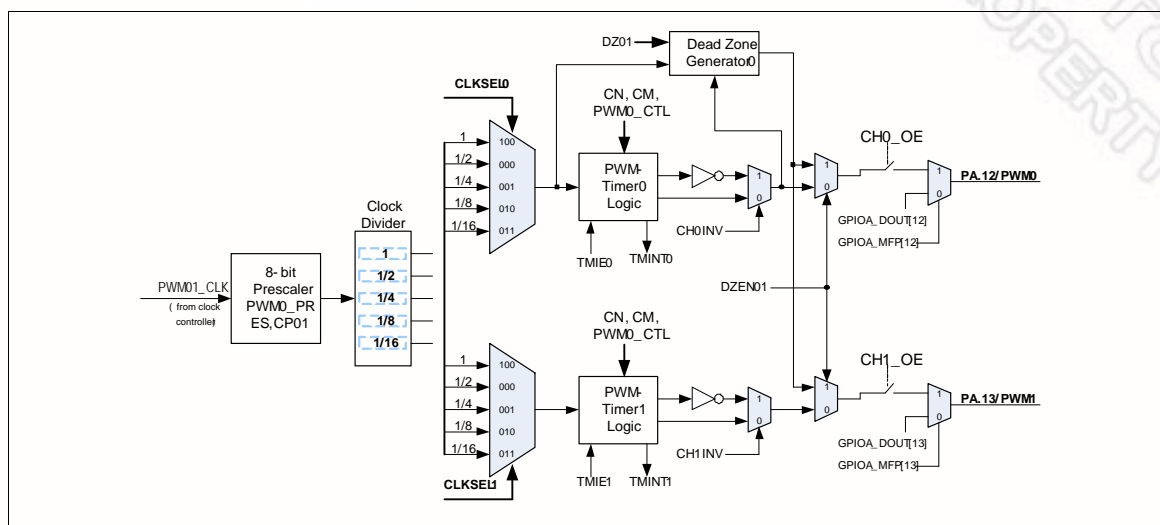


图 5-71 用于通道 0, 1 的 PWM0 发生器

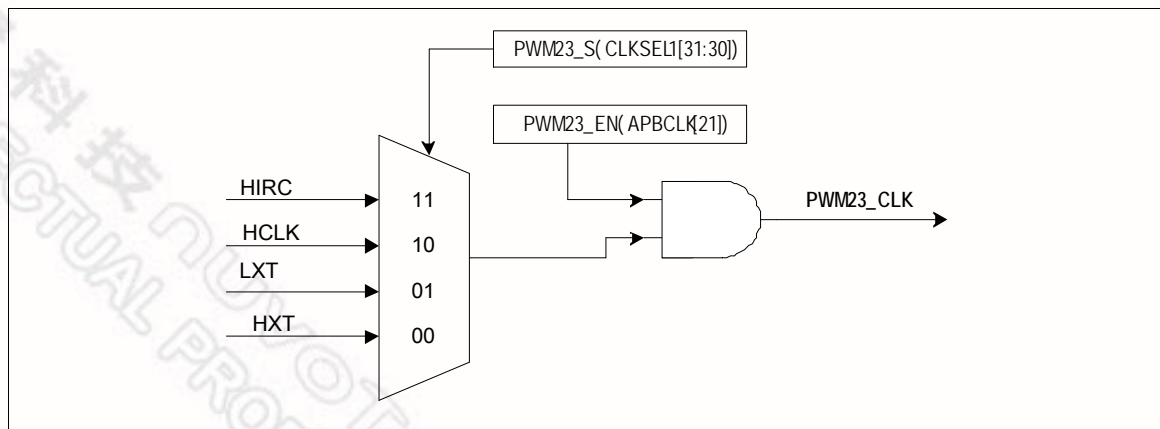


图 5-72 PWM1 时钟

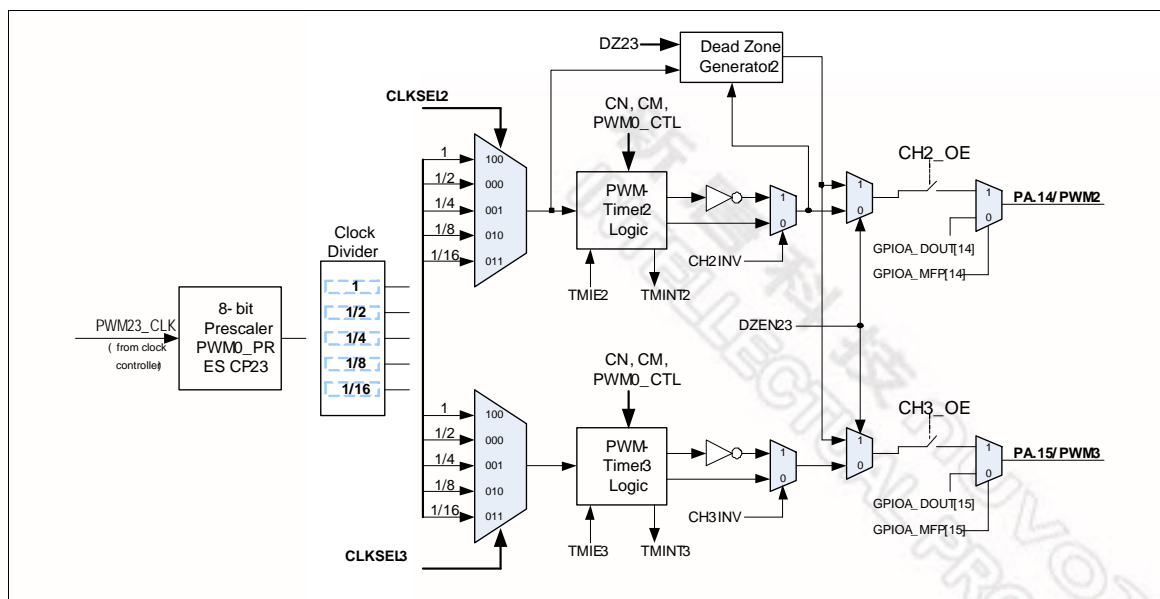


图 5-73 用于通道 2, 3 的 PWM1 发生器

5.14.4 功能描述

5.14.4.1 PWM-定时器 操作

PWM 周期和占空比控制由 PWMx_DUTYy(y=0~3) 寄存器 CN (PWMx_DUTYy[15:0]) 和 CM (PWMx_DUTYy[31:16]) 决定。PWM-定时器工作时序如图 5-75。脉宽调制的公式如下，PWM-定时器的比较器说明如图 5-74。注：在 PWM 功能使能前，相应的 GPIO 管脚必须配置成输出类型。

PWM 频率 = $\text{PWMxy_CLK}/(\text{prescale}+1)*(\text{clock divider})/(\text{CN}+1)$; xy 可以是 01, 23, 取决于所选择的 PWM 通道。

占空比 = $(\text{CM}+1)/(\text{CN}+1)$.

CM >= CN: PWM 输出总是为高

CM < CN: PWM 低脉宽 = (CN-CM) unit1; PWM 高脉宽 = (CM+1) unit.

CM = 0: PWM 低脉宽 = (CN) unit; PWM 高脉宽 = 1 unit

Note: 1. Unit = 一个 PWM 时钟周期。

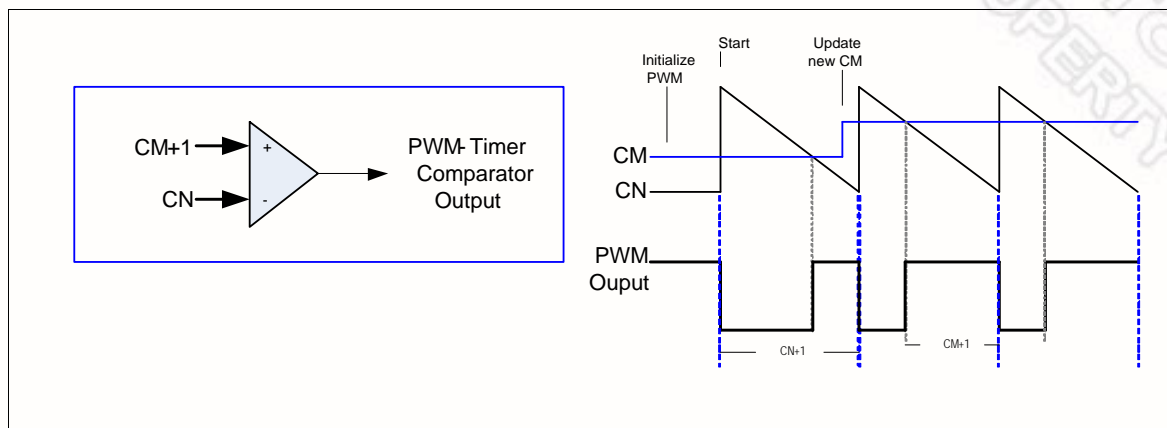


图 5-74 PWM-定时器内部比较器输出

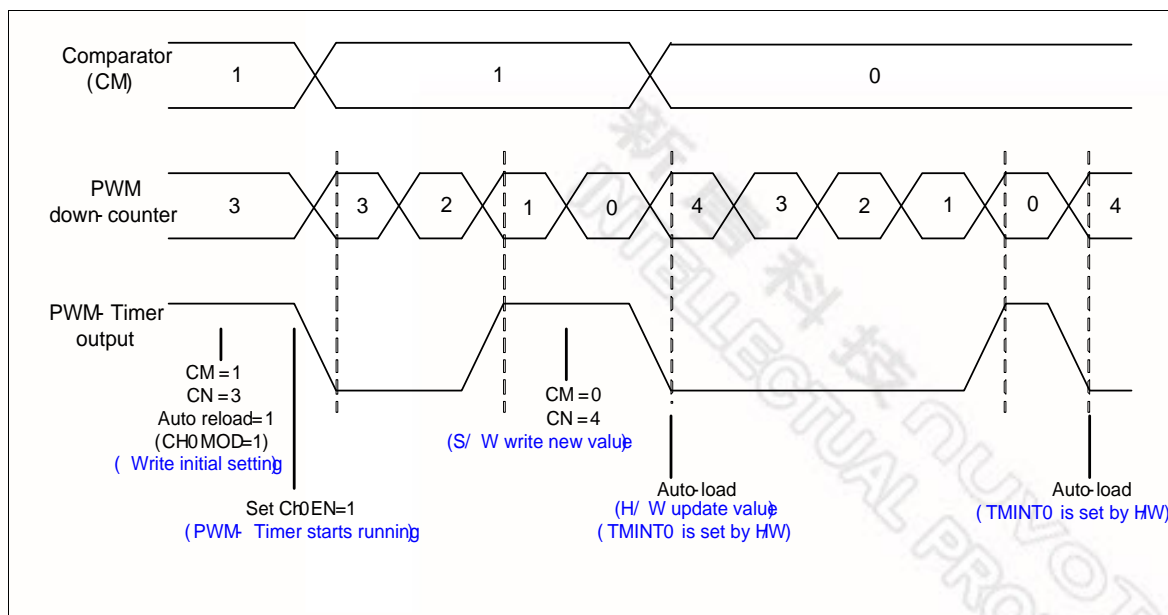


图 5-75 通道 0 PWM-定时器操作时序

5.14.4.2 PWM 双缓存，连续以及单次操作

PWM 具有双缓存功能，重载值只在下一个周期开始时被更新，而不会影响当前定时器工作。PWM 计数器的值可以被写入 PWMx_DUTY0~3的bit[15:0]。

PWM 控制寄存器 (PWMx_CTL) 中 CH0MOD 位定义 PWM0 是连续模式或是单次模式。如果 CH0MOD 被设为 1 (连续模式)，当 PWM 计数器计数到 0，PWM 控制器加载CN 的值到 PWM 计数器。如果 CN 被设为 0，PWM 计数器在计数到0后，将停止运行。

在单次模式下 (CH0MOD=0)，相应的通道将仅仅输出一占空波形，并且如果没有进一步的相应占空寄存器的更新，计数器将停止。当 PWM 计数器正在运行时，更新相应的占空寄存器将会进行下一占空波形输出。

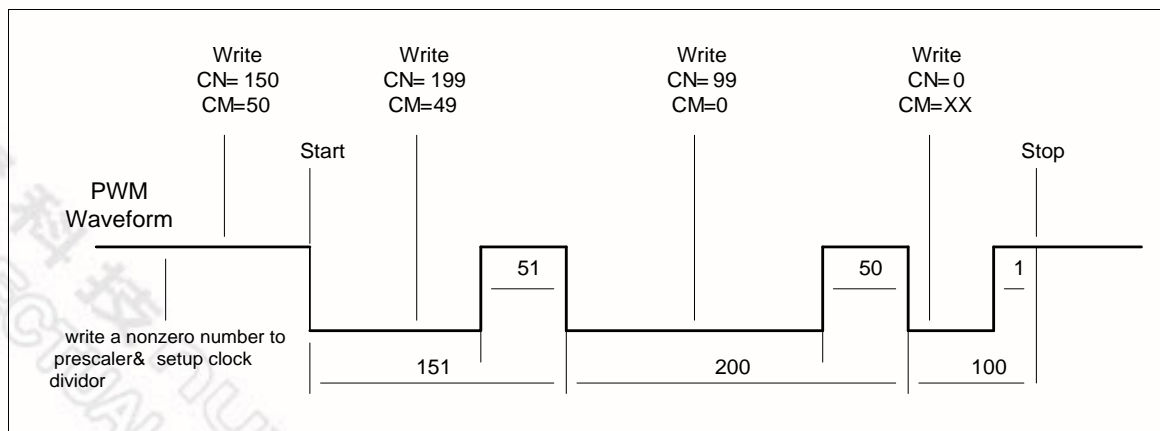


图 5-76 PWM 双缓存说明

5.14.4.3 可调占空比

双缓存功能允许 CM 在当前周期的任意时刻被写入。写入值会在下一个周期生效。

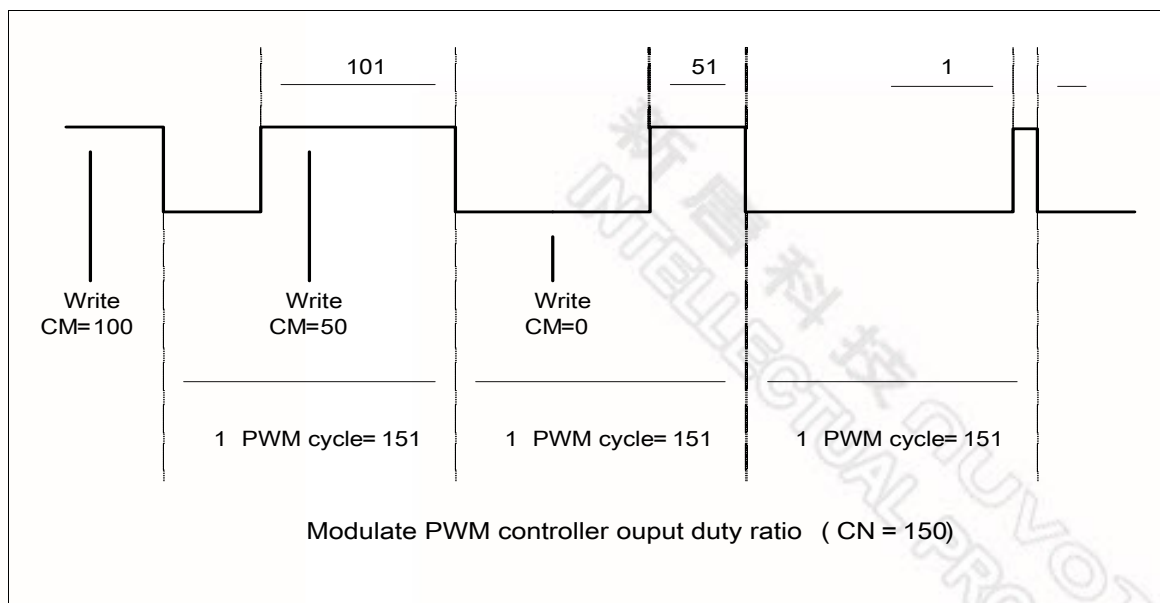


图 5-77 PWM 控制器输出占空比

5.14.4.4 死区发生器

PWM 控制器提供死区发生器，用于功率器件保护。该功能在 PWM 上升沿输出时产生可编程的时隙来延迟 PWM 上升沿输出。用户可通过编程死区计数器来确定死区间隔。

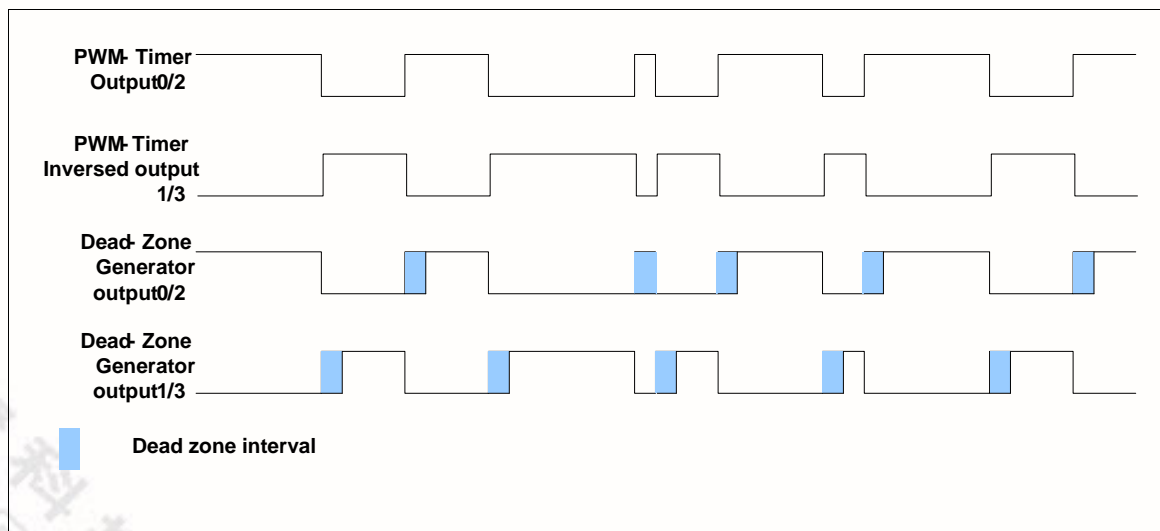


图 5-78 带死区发生器 PWM-对输出操作

5.14.4.5 捕捉操作

捕捉通道 0 和 PWM 通道 0 共享同一个定时器，捕捉通道 1 和 PWM 通道 1 共享另一组定时器，以此类推。当输入信号有上升沿转变时，PWM 定时器的值将被锁存到 PWM_CRLx 寄存器；而当输入信号有下降沿转变时，PWM 定时器的值将被锁存到 PWM_CFLx 寄存器。捕捉通道 0 中断发生的条件可通过编程 PWMx_CAPINTEN[0] (上升沿锁存中断使能) 和 PWMx_CAPINTEN[1] (下降沿锁存中断使能)来设定。无论何时，当捕捉模块触发一个定义在 PWMx_CAPINTSTS 寄存器中的捕捉标志 (上升沿或下降沿锁存)，并且重载使能位 (在 PWMx_CAPCTL 中定义) 被使能，相应的 PWM 定时器将会同时被重新装载为 CN 寄存器的值。注：在捕捉功能被使能之前，相应的 GPIO

管脚必须配置成输入类型。

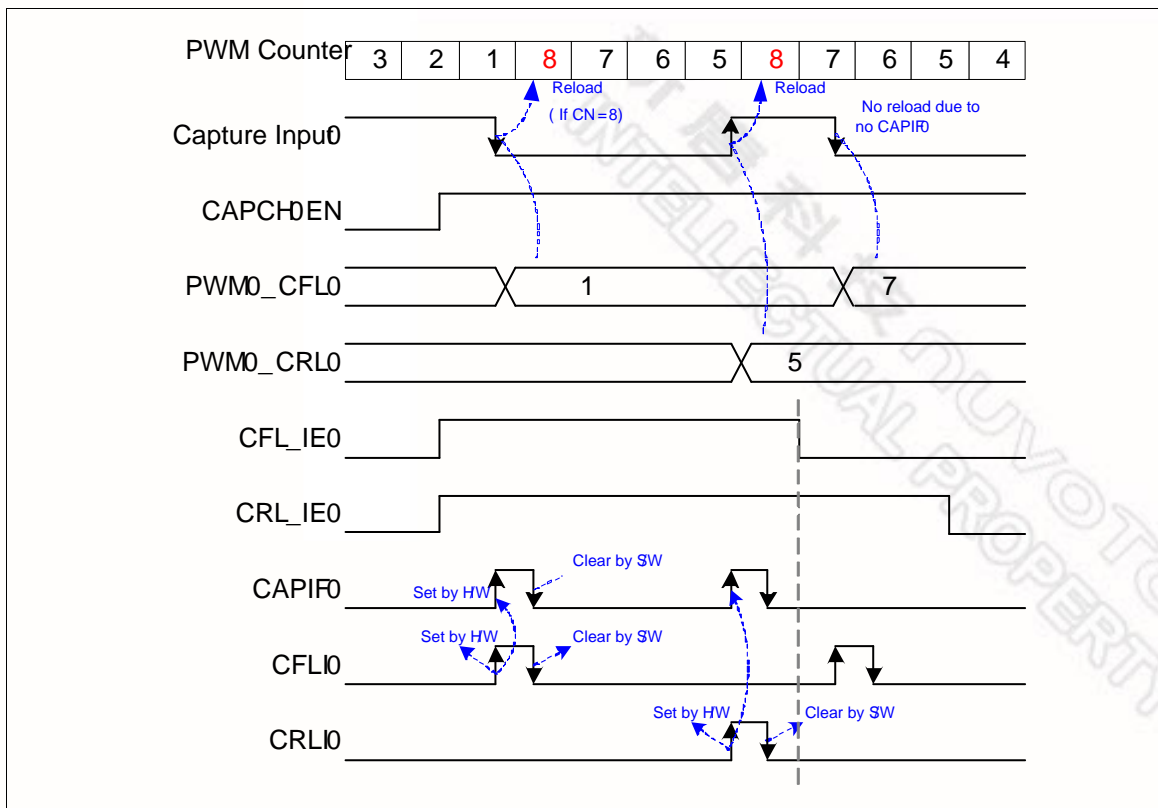


图 5-79 PWM 捕捉操作时序

在这个范例中，CN 为 8：

当捕捉中断标志 (CAPIF0) 被置位时，PWM 定时器将重载为 CN 的值。

通道低脉宽为(CN - CRL)。

通道高脉宽为(CN - CFL)。

对于捕捉功能，在某些需要更宽计数器的情况下，用户可以级联两个 16 位的计数器成为一个 32 位的计数器。

级联方式如下所述：

当使能 PWMx_CAPCTL 寄存器中 CH01CASK 位时，内部级联逻辑将会把 CH0 的 16 位的计数器和 CH1 的 16 位的计数器连接起来作为一个通道 0 的 32 位的计数器；使能 PWMx_CAPCTL 寄存器的 CH23CASK 位可以以同样的方式把 CH2 和 CH3 的计数器级联起来作为 CH2 的 32 位计数器。PWMx_DUTY0 寄存器中 CM 和 CN 级联起来作为通道 0 的 32 位的 CN，而 PWMx_DUTY2 寄存器中 CM 和 CN 级联起来作为通道 2 的 32 位的 CN (CM 位域为 32 位 CN 的高半字，x=0~1)。在这种情况下，CH1 和 CH3 的捕捉功能是无用的。当捕捉标志被设置 (PWMx_CAPINTSTS[1], 上升沿；和 PWMx_CAPINTSTS[2], 下降沿)，在上升沿的捕捉数据被锁存到 PWM_CRL0 和 PWM_CRL1；在下降沿的捕捉数据被锁存到 PWM_CFL0 和 PWM_CFL1。PWM_CRL1 用于锁存高半字，而 PWM_CRL0 用于锁存低半字；对于 PWM_CFL1 和 PWM_CFL0 也是一样的。

当级联被使能时，用户也可以通过直接读 PWM_CRL0 或者 PWM_CFL0 寄存器来获取 32 位的捕捉数据。

注：级联功能仅适用于 PWM 的捕捉功能。

5.14.4.6 PWM PDMA 功能

当工作在捕捉模式时，PWM 支持 PDMA 传输功能，但仅仅在指定的通道上 (通道0, 2) 支持该功能。当相应的 PDMA 使能位 (在 PWMx_CAPCTL 寄存器中定义) 被置位，且之前已经有捕捉事件发生，捕捉模块将会触发一个中断请求到 PDMA 控制器，PDMA 控制器会发出一个 ACK 给捕捉模块，并读 PDMACH0 寄存器的值到内存中。通过设定 PDMACAPMOD0 和 PDMACAPMOD2，PDMA 可以传输上升沿锁存数据或者下降沿锁存数据，或者上升沿和下降沿锁存数据到内存。当使用 PDMA 传输上升沿和下降沿数据时，需要设定 PWMx_CAPCTL 寄存器的CHxRFORORDER 位来决定数据传输顺序(先传输下降沿锁存数据还是先传输上升沿锁存数据)。

5.14.4.7 PWM-定时器中断架构

提供八个 PWM 中断，PWM0CH0_INT~PWM0CH3_INT，PWM1CH0_INT~PWM1CH3_INT，这两组中断分别以或的关系组合成 PWM0_INT 和 PWM1_INT 中断。PWM CH0 和捕捉通道 0 共享同一个中断，PWM CH1 和捕捉通道 1 共享同一个中断，以此类推。因此，不能在同一个通道同时使用 PWM 功能和捕捉功能。

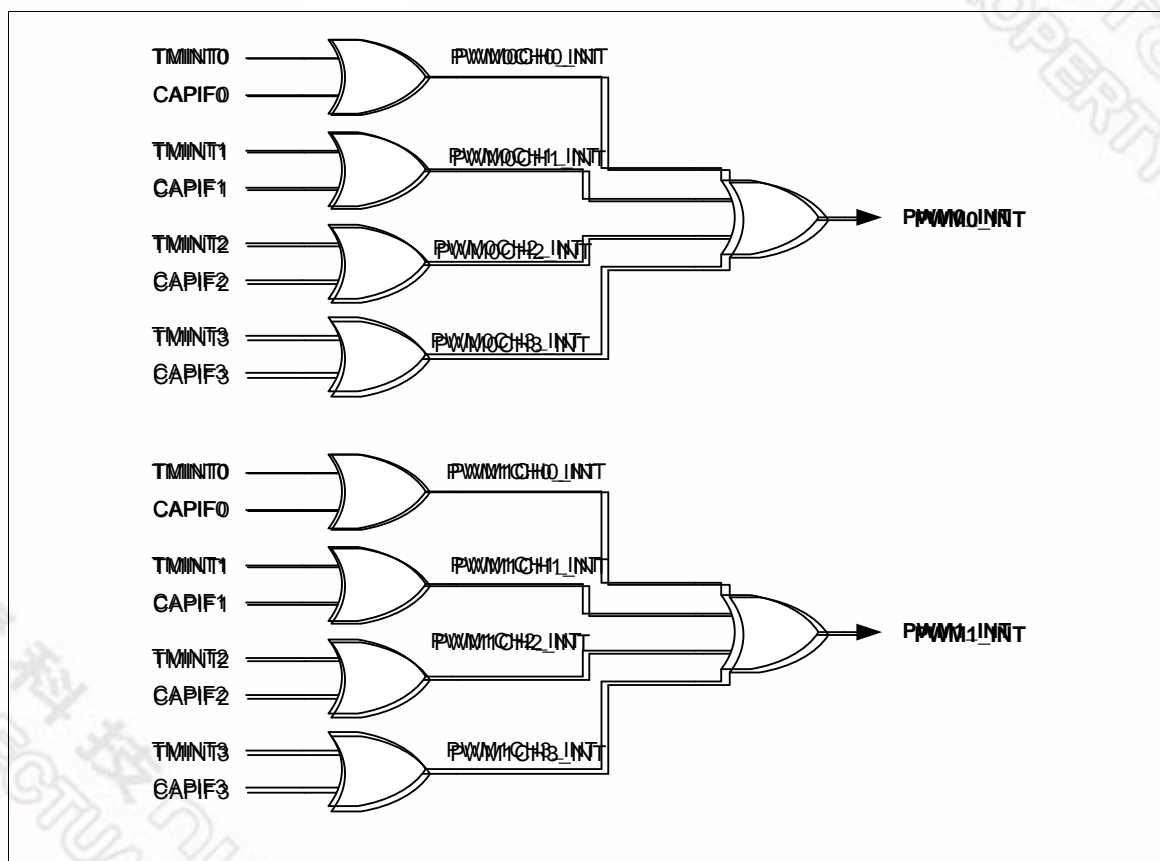


图 5-80 PWM-定时器中断

5.14.4.8 PWM-定时器开启过程

下列步骤用于启动 PWM。

- 设置时钟选择器(PWMx_CLKSEL),x=0~1
- 设置预分频器(PWMx_PRES),x=0~1

- 设置反转打开/关闭，死区发生器打开/关闭，自动重载/单次模式并停止 PWM-定时器 (PWMx_CTL, x=0~1)
- 设置中断使能寄存器(PWMx_INTEN),x=0~1
- 设置 PWM 输出使能(PWMx_OE),x=0~1
- 设置相应 GPIO 管脚为 PWM 功能
- 设置相应 GPIO 管脚为输出类型
- 使能 PWM 下数型计数器开始运行 (设置 PWMx_CTL 中 CHxEN=1, x=0~1)
- 设置 PWMx_DUTYy 寄存器的 CM 和 CN 位域来设定 PWM 占空比, x=0~1, y=0~3。
(当级联被使能, CM位域作为 32 位 CN 位域的高半字)
- 上述步骤 1~8 可以不按照上述的顺序设置, 这对 PWM 定时器的正常工作没有影响

5.14.4.9 PWM-定时器停止过程

以 PWM0，通道 0 为例：

方式 1:

设置 32 位 PWMx_DUTY0 寄存器为 0，并等待 PWM 超时中断 (如果 PWMx_IE 寄存器的 bit0 被置位) 发生，或者轮询相应的超时标志。当 PWM 超时中断发生，或者超时标志被置位，禁止 PWM 定时器 (PWMx_CTL 寄存器中 CH0EN)。(推荐)

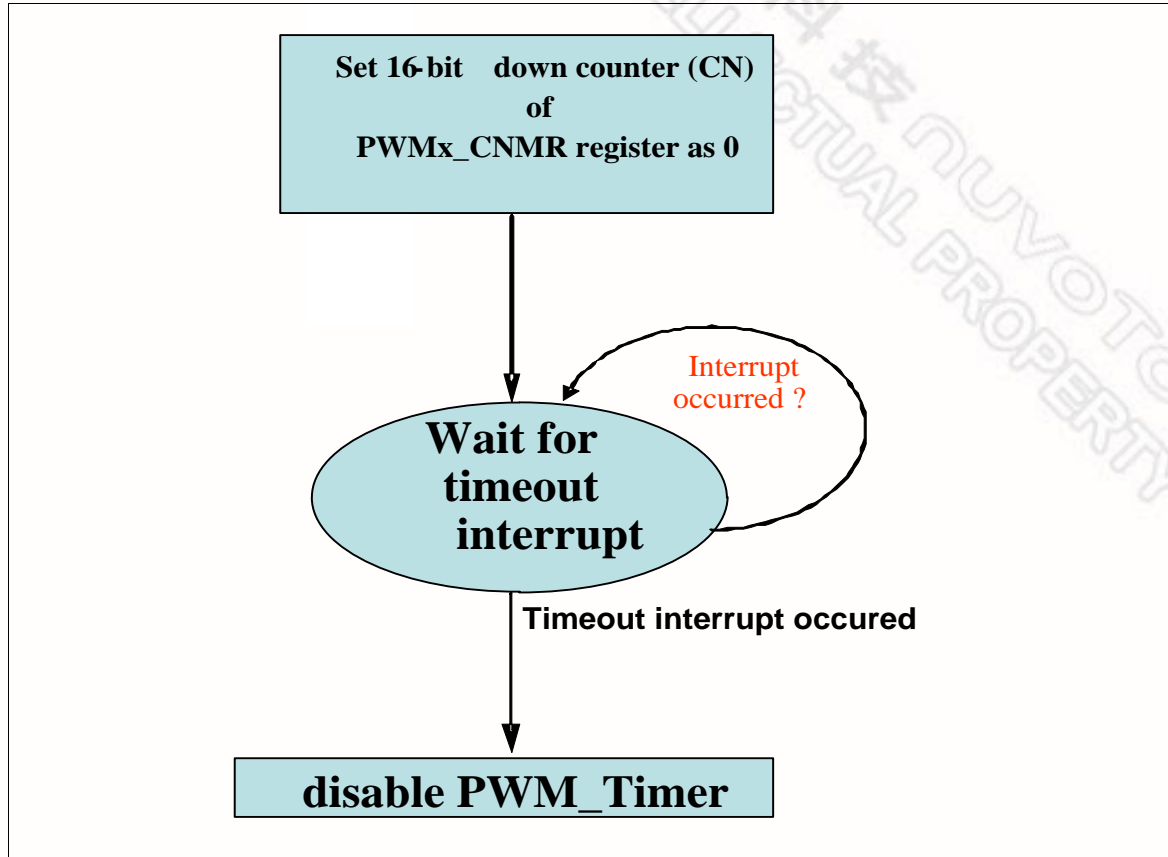


图 5-81 PWM-定时器停止方式 1

方式 2:

直接禁用 PWM 定时器 (PWMx_CTL 寄存器的 CHxEN 位)。(不推荐)

不推荐方式 2 是因为禁止 CHxEN 将立即停止 PWM 输出信号，会导致 PWM 输出的占空比改变，这可能引起电机控制电路的损坏。

5.14.4.10 捕捉开始过程

- 设置时钟选择器(PWMx_CLKSEL),x=0~1
- 设置预分频器 (PWMx_PRES),x=0~1
- 设置通道使能，上升/下降沿中断使能以及输入信号反转打开/关闭 (PWMx_CAPCTL,PWMx_CAPINTEN),x=0~1
- 设置 PWM 下数计数器 (PWMx_DUTYy 寄存器中 CN 位域，x=0~1, y=0~3)
- 设置捕捉输入使能寄存器(PWMx_CAPCTL),x=0~1

- 设置相应的 GPIO 管脚为 PWM 功能
- 设置相应的 GPIO 管脚为输入类型
- 使能 PWM 下数型计数器开始运行 (设置 PWMx_CTL 中的 CHyEN=1, x=0~1, y=0~3)

5.14.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and written

寄存器	偏移量	R/W	描述	复位后的值
PWM0_BA = 0x4004_0000				
PWM1_BA = 0x4014_0000				
PWMx_PRES	PWMx_BA+0x000	R/W	PWM 预分频寄存器	0x0000_0000
PWMx_CLKSEL	PWMx_BA+0x004	R/W	PWM 时钟现在寄存器	0x0000_0000
PWMx_CTL	PWMx_BA+0x008	R/W	PWM 控制寄存器	0x0000_0000
PWMx_INTEN	PWMx_BA+0x00C	R/W	PWM 中断使能寄存器	0x0000_0000
PWMx_INTSTS	PWMx_BA+0x010	R/W	PWM 中断指示寄存器	0x0000_0010
PWMx_OE	PWMx_BA+0x014	R/W	PWM0~PWM3 输出使能	0x0000_0000
PWMx_DUTY0	PWMx_BA+0x01C	R/W	PWM 计数器/比较器寄存器 0	0x0000_0000
PWMx_DATA0	PWMx_BA+0x020	R	PWM 数据寄存器 0	0x0000_0000
PWMx_DUTY1	PWMx_BA+0x028	R/W	PWM 计数器/比较器寄存器 1	0x0000_0000
PWMx_DATA1	PWMx_BA+0x02C	R	PWM 数据寄存器 1	0x0000_0000
PWMx_DUTY2	PWMx_BA+0x034	R/W	PWM 计数器/比较器寄存器 2	0x0000_0000
PWMx_DATA2	PWMx_BA+0x038	R	PWM 数据寄存器 2	0x0000_0000
PWMx_DUTY3	PWMx_BA+0x040	R/W	PWM 计数器/比较器寄存器 3	0x0000_0000
PWMx_DATA3	PWMx_BA+0x044	R	PWM 数据寄存器 3	0x0000_0000
PWMx_CAPCTL	PWMx_BA+0x054	R/W	Capture 控制寄存器	0x0000_0000
PWMx_CAPINTEN	PWMx_BA+0x058	R/W	Capture 中断使能寄存器	0x0000_0000
PWMx_CAPINTSTS	PWMx_BA+0x05C	R/W	Capture 中断指示寄存器	0x0000_0000
PWMx_CRL0	PWMx_BA+0x060	R	Capture 上升沿锁存寄存器 (通道 0)	0x0000_0000
PWMx_CFL0	PWMx_BA+0x064	R	Capture 下降沿锁存寄存器 (通道 0)	0x0000_0000
PWMx_CRL1	PWMx_BA+0x068	R	Capture 上升沿锁存寄存器 (通道 1)	0x0000_0000
PWMx_CFL1	PWMx_BA+0x06C	R	Capture 下降沿锁存寄存器 (通道 1)	0x0000_0000
PWMx_CRL2	PWMx_BA+0x070	R	Capture 上升沿锁存寄存器 (通道 2)	0x0000_0000
PWMx_CFL2	PWMx_BA+0x074	R	Capture 下降沿锁存寄存器 (通道 2)	0x0000_0000
PWMx_CRL3	PWMx_BA+0x078	R	Capture 上升沿锁存寄存器 (通道 3)	0x0000_0000
PWMx_CFL3	PWMx_BA+0x07C	R	Capture 下降沿锁存寄存器 (通道 3)	0x0000_0000
PWMx_PDMACH0	PWMx_BA+0x080	R	PDMA 通道 0 捕捉到的数据	0x0000_0000

寄存器	偏移量	R/W	描述	复位后的值
PWMx_PDMACH2	PWMx_BA+0x084	R	PDMA 通道 2 捕捉到的数据	0x0000_0000

5.14.6 寄存器描述

PWM 预分频寄存器

寄存器	偏移量	R/W	描述	复位后的值
PWMx_PRES x = 0,1	PWMx_BA+0x000	R/W	PWM 预分频寄存器	0x0000_0000

31	30	29	28	27	26	25	24
DZ23							
23	22	21	20	19	18	17	16
DZ01							
15	14	13	12	11	10	9	8
CP23							
7	6	5	4	3	2	1	0
CP01							

Bits	描述	
[31:24]	DZ23[7:0]	CH2 和 CH3 对的死区间隔寄存器 该 8 位决定死区长度 死区长度的单位时间来自时钟选择器 2.
[23:16]	DZ01[7:0]	CH0 和 CH1 对的死区间隔寄存器 该 8 位决定死区长度 死区长度的单位时间来自时钟选择器 0.
[15:8]	CP23[7:0]	应用于 PWM 定时器 2 & 3 的时钟预分频器 2 在输入给计数器 2&3 之前, 时钟输入先被 (CP23 + 1) 除 如果 CP23=0, 则预分频器 2 输出时钟将会停止, 因此 PWM 计数器 2 和 3 也将停止
[7:0]	CP01[7:0]	应用于 PWM 定时器 0 & 1 的时钟预分频器 0 在输入给计数器 0&1 之前, 时钟输入先被 (CP01 + 1) 除 如果 CP01=0, 则预分频器 0 输出时钟将会停止, 因此 PWM 计数器 0 和 1 也将停止

PWM 时钟选择寄存器 (PWMx_CLKSEL)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CLKSEL x = 0,1	PWMx_BA+0x004	R/W	PWM 时钟选择寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-	CLKSEL3			-	CLKSEL2		
7	6	5	4	3	2	1	0
-	CLKSEL1			-	CLKSEL0		

Bits	描述												
[31:15]	-	保留											
[14:12]	CLKSEL3[2:0]	定时器 3 时钟源选择 为定时器 3 选择时钟输入											
		CLKSEL3 [14:12]	输入时钟被除以	100	1	011	16	010	8	001	4	000	2
		CLKSEL3 [14:12]	输入时钟被除以										
		100	1										
		011	16										
		010	8										
		001	4										
000	2												
[10:8]	CLKSEL2[2:0]	定时器 2 时钟源选择 为定时器 2 选择时钟输入 (分频表与 CLKSEL3 一样)											
[6:4]	CLKSEL1[2:0]	定时器 1 时钟源选择 为定时器 1 选择时钟输入 (分频表与 CLKSEL3 一样)											
[2:0]	CLKSEL0[2:0]	定时器 0 时钟源选择 为定时器 0 选择时钟输入 (分频表与 CLKSEL3 一样)											

PWM 控制寄存器 (PWMx_CTL)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CTL x = 0,1	PWMx_BA+0x008	R/W	PWM 控制寄存器 (PWM_CTL)	0x0000_0000

31	30	29	28	27	26	25	24
-				CH3MOD	CH3INV	-	CH3EN
23	22	21	20	19	18	17	16
-				CH2MOD	CH2INV	-	CH2EN
15	14	13	12	11	10	9	8
-				CH1MOD	CH1INV	-	CH1EN
7	6	5	4	3	2	1	0
-		DZEN23	DZEN01	CH0MOD	CH0INV	-	CH0EN

Bits	描述	
[27]	CH3MOD	PWM 定时器 3 连续/单次模式 1 = 连续模式 0 = 单次模式 注: 如果该位的值被从 0 设置为 1, 会导致 PWMx_DUTY3 寄存器的 CN 和 CM 位域被清零
[26]	CH3INV	PWM 定时器 3 输出反向 开/关 1 = 反向开 0 = 反向关
[24]	CH3EN	PWM 定时器 3 使能开始/停止运行 1 = 使能 PWM 定时器 3 开始运行 0 = 停止 PWM 定时器 3
[19]	CH2MOD	PWM 定时器 2 连续/单次模式 1 = 连续模式 0 = 单次模式 注: 如果该位的值被从 0 设置为 1, 会导致 PWMx_DUTY2 寄存器的 CN 和 CM 位域被清零.
[18]	CH2INV	PWM 定时器 2 输出反向 开/关 1 = 反向开 0 = 反向关
[16]	CH2EN	PWM 定时器 2 使能开始/停止运行 1 = 使能 PWM 定时器 2 开始运行

Bits	描述	
		0 = 停止 PWM 定时器 2
[11]	CH1MOD	PWM 定时器 1 连续/单次模式 1 = 连续模式 0 = 单次模式 注: 如果该位的值被从 0 设置为 1, 会导致 PWMx_DUTY1 寄存器的 CN 和 CM 位域被清零
[10]	CH1INV	PWM 定时器 1 输出反向 开/关 1 = 反向开 0 = 反向关
[8]	CH1EN	PWM 定时器 1 使能开始/停止运行 1 = 使能 PWM 定时器 1 开始运行 0 = 停止 PWM 定时器 1
[5]	DZEN23	死区 2 发生器使能/禁止 1 = 使能 0 = 禁止 注: 当死区发生器被使能, PWMx_CH2 和 PWMx_CH3 变成一个互补对
[4]	DZEN01	死区 0 发生器使能/禁止 1 = 使能 0 = 禁止 注: 当死区发生器被使能, PWMx_CH0 和 PWMx_CH1 变成一个互补对.
[3]	CH0MOD	PWM 定时器 0 连续/单次模式 1 = 连续模式 0 = 单次模式 注: 如果该位的值被从 0 设置为 1, 会导致 PWMx_DUTY0 寄存器的 CN 和 CM 位域被清零
[2]	CH0INV	PWM 定时器 0 输出反向 开/关 1 = 反向开 0 = 反向关
[0]	CH0EN	PWM 定时器 0 使能开始/停止运行 1 = 使能 PWM 定时器 0 开始运行 0 = 停止 PWM 定时器 0

PWM 中断使能寄存器 (PWMx INTEN)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_INTEN x=0,1	PWMx_BA+0x00C	R/W	PWM 中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-				TMIE3	TMIE2	TMIE1	TMIE0

Bits	描述	
[31:4]	-	保留。
[3]	TMIE3	PWM 定时器 3 中断使能 1 = 使能 0 = 禁止
[2]	TMIE2	PWM 定时器 2 中断使能 1 = 使能 0 = 禁止
[1]	TMIE1	PWM 定时器 1 中断使能 1 = 使能 0 = 禁止
[0]	TMIE0	PWM 定时器 0 中断使能 1 = 使能 0 = 禁止

PWM 中断标志寄存器 (PWMx_INTSTS)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_INTSTS x =0,1	PWM_BA+0x010	R/W	PWM 中断指示寄存器	0x0000_0010

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							PresSyncFlag
7	6	5	4	3	2	1	0
Duty3Syncflag	Duty2Syncflag	Duty1Syncflag	Duty0Syncflag	TMINT3	TMINT2	TMINT1	TMINT0

Bits	描述	
[8]	PresSyncFlag	<p>预分频同步标志</p> <p>1 = 预分频正在同步于 ECLK</p> <p>0 = 预分频已经同步于 ECLK</p> <p>注: 当写入预分频值时, 软件应当检测该标志, 该标志被置位时, 若用户忽略该标志, 并且去改变预分频值的话, 预分频可能会出现一个分频周期的错误。</p>
[7]	Duty3Syncflag	<p>Duty3 同步标志</p> <p>1 = duty3 正在同步于 ECLK</p> <p>0 = duty3 已经同步于 ECLK</p> <p>注: 当写入 duty3 时, 软件应当检测该标志, 该标志被置位时, 若用户忽略该标志, 并且去改变 duty3 的值的话, 相应的 CN 和 CM 可能会出现一个占空周期的错误。</p>
[6]	Duty2Syncflag	<p>Duty2 同步标志</p> <p>1 = duty2 正在同步于 ECLK</p> <p>0 = duty2 已经同步于 ECLK</p> <p>注: 当写入 duty2 时, 软件应当检测该标志, 该标志被置位时, 若用户忽略该标志, 并且去改变 duty2 的值的话, 相应的 CN 和 CM 可能会出现一个占空周期的错误。</p>
[5]	Duty1Syncflag	<p>Duty1 同步标志</p> <p>1 = duty1 正在同步于 ECLK</p> <p>0 = duty1 已经同步于 ECLK</p> <p>注: 当写入 duty1 时, 软件应当检测该标志, 该标志被置位时, 若用户忽略该标志, 并且去改变 duty1 的值的话, 相应的 CN 和 CM 可能会出现一个占空周期的错误。</p>
[4]	Duty0Syncflag	Duty0 同步标志

Bits	描述	
		<p>1 = duty0 正在同步于 ECLK</p> <p>0 = duty0 已经同步于 ECLK</p> <p>注: 当写入 duty0 时, 软件应当检测该标志, 该标志被置位时, 若用户忽略该标志, 并且去改变 duty0 的值的话, 相应的 CN 和 CM 可能会出现一个占空周期的错误。</p>
[3]	TMINT3	<p>PWM 定时器 3 中断标志</p> <p>当 PWM3 下数型计数器计数到 0 时, 该标志由硬件自动置位, 软件可以通过向该位写 1 来清除该标志</p>
[2]	TMINT2	<p>PWM 定时器 2 中断标志</p> <p>当 PWM2 下数型计数器计数到 0 时, 该标志由硬件自动置位, 软件可以通过向该位写 1 来清除该标志</p>
[1]	TMINT1	<p>PWM 定时器 1 中断标志</p> <p>当 PWM1 下数型计数器计数到 0 时, 该标志由硬件自动置位, 软件可以通过向该位写 1 来清除该标志</p>
[0]	TMINT0	<p>PWM 定时器 0 中断标志</p> <p>当 PWM0 下数型计数器计数到 0 时, 该标志由硬件自动置位, 软件可以通过向该位写 1 来清除该标志</p>

注: 用户可以通过向 PWM_IS 寄存器中某一位写 1 来清除相应的中断标志。

PWM 输出使能寄存器 (PWMx OE)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_OE x =0,1	PWMx_BA+0x014	R/W	PWM CH0 ~ CH3 输出使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-				CH3_OE	CH2_OE	CH1_OE	CH0_OE

Bits	描述	
[3]	CH3_OE	PWM CH3 输出使能 1 = 使能 PWM CH3 输出 0 = 禁止 PWM CH3 输出 注: 相应的 GPIO 引脚必须被切换到 PWM 功能 (参考 GPx_MFP)
[2]	CH2_OE	PWM CH2 输出使能 1 = 使能 PWM CH2 输出 0 = 禁止 PWM CH2 输出 注: 相应的 GPIO 引脚必须被切换到 PWM 功能 (参考 GPx_MFP)
[1]	CH1_OE	PWM CH1 输出使能 1 = 使能 PWM CH1 输出 0 = 禁止 PWM CH1 输出 注: 相应的 GPIO 引脚必须被切换到 PWM 功能 (参考 GPx_MFP)
[0]	CH0_OE	PWM CH0 输出使能 1 = 使能 PWM CH0 输出 0 = 禁止 PWM CH0 输出 注: 相应的 GPIO 引脚必须被切换到 PWM 功能 (参考 GPx_MFP)

PWM DUTY 寄存器 3-0 (PWMx DUTY3~0)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_DUTY0 x =0,1	PWMx_BA+0x01C	R/W	PWM计数器/比较器寄存器 0	0x0000_0000
PWMx_DUTY1 x =0,1	PWMx_BA+0x028	R/W	PWM计数器/比较器寄存器 1	0x0000_0000
PWMx_DUTY2 x =0,1	PWMx_BA+0x034	R/W	PWM计数器/比较器寄存器 2	0x0000_0000
PWMx_DUTY3 x =0,1	PWMx_BA+0x040	R/W	PWM计数器/比较器寄存器 3	0x0000_0000

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

Bits	描述	
[31:16]	CM[15:0]	<p>PWM 比较器寄存器</p> <p>CM 决定 PWM 占空比.</p> <p>PWM 频率 = $PWM_{xy_CLK} / (prescale+1) * (clock\ divider) / (CN+1)$; xy 可以是 01, 23, 取决于选择的 PWM 通道.</p> <p>占空比 = $(CM+1) / (CN+1)$.</p> <p>CM \geq CN: PWM 输出总是为高</p> <p>CM < CN: PWM 低脉冲宽度 = (CN-CM) unit; PWM 高脉冲宽度 = (CM+1) unit.</p> <p>CM = 0: PWM 低脉冲宽度 = (CN) unit; PWM 高脉冲宽度 = 1 unit</p> <p>(Unit = 一个 PWM 时钟周期)</p> <p>注:</p> <p>任何向 CM 的写入都将会在写入之后的下个 PWM 周期生效.</p>
[15:0]	CN[15:0]	<p>PWM 计数器/定时器 加载值</p> <p>CN 决定 PWM 周期.</p> <p>PWM 频率 = $PWM_{xy_CLK} / (prescale+1) * (clock\ divider) / (CN+1)$; xy 可以是 01, 23, 取决于选择的 PWM 通道</p>

Bits	描述
	<p>占空比 = $(CM+1)/(CN+1)$.</p> <p>CM >= CN: PWM 输出总是为高.</p> <p>CM < CN: PWM 低脉冲宽度 = (CN-CM) unit; PWM 高脉冲宽度 = (CM+1) unit.</p> <p>CM = 0: PWM 低脉冲宽度 = (CN) unit; PWM 高脉冲宽度 = 1 unit</p> <p>(Unit = 一个 PWM 时钟周期)</p> <p>注:</p> <p>任何向 CN 的写入都将会在写入之后的下个 PWM 周期生效.</p>

PWM 数据寄存器 (PWMx DATA)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_DATA0	PWMx_BA+0x020	R	PWM 数据寄存器 0	0x0000_0000
PWMx_DATA1	PWMx_BA+0x02C	R	PWM 数据寄存器 1	0x0000_0000
PWMx_DATA2	PWMx_BA+0x038	R	PWM 数据寄存器 2	0x0000_0000
PWMx_DATA3	PWMx_BA+0x044	R	PWM 数据寄存器 3	0x0000_0000

31	30	29	28	27	26	25	24
sync	DATA (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)						
23	22	21	20	19	18	17	16
DATA (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

Bits	描述	
[31]	sync	指示 CN 的值是否同步于 PWM 计数器 1 = CN 的值不同步于 PWM 计数器 0 = CN 的值同步于 PWM 计数器 注: 当相应的级联被使能, 该位不会出现在相应的通道
[30:16]	PWMx_DATAy[30:16] (x=0~1,y=0,2)	PWM 数据寄存器 用户可以通过监视 PWMx_DATAy 的值来查看 32 位下数型计数器的当前值 注: 仅当相应的级联使能位被设置时, 该位域才是有效的
[15:0]	PWMx_DATAy[15:0] (x=0~1,y=0~3)	PWM 数据寄存器 用户可以通过监视 PWMx_DATAy 的值来查看 16 位下数型计数器的当前值

Capture 控制寄存器 (PWMx_CAPCTL)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CAPCTL	PWMx_BA+0x054	R/W	Capture 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
CAPRELOAD FEN3	CAPRELOAD REN3	CH23CASK	CH2RFORDE R	-	CAPCH3PAD EN	CAPCH3EN	INV3
23	22	21	20	19	18	17	16
CAPRELOAD FEN2	CAPRELOAD REN2	PDMACAPMOD2		CH2PDMAEN	CAPCH2PAD EN	CAPCH2EN	INV2
15	14	13	12	11	10	9	8
CAPRELOAD FEN1	CAPRELOAD REN1	CH01CASK	CH0RFORDE R	-	CAPCH1PAD EN	CAPCH1EN	INV1
7	6	5	4	3	2	1	0
CAPRELOAD FEN0	CAPRELOAD REN0	PDMACAPMOD0		CH0PDMAEN	CAPCH0PAD EN	CAPCH0EN	INV0

Bits	描述	
[31]	CAPRELOADFEN3	当 CH3 下降沿捕捉事件发生时, 重载CNR3 1 = 使能 CH3 下降沿捕捉重载 0 = 禁止 CH3 下降沿捕捉重载
[30]	CAPRELOADREN3	当 CH3 上升沿捕捉事件发生时, 重载CNR3 1 = 使能 CH3 上升沿捕捉重载 0 = 禁止 CH3 上升沿捕捉重载
[29]	CH23CASK	级联通道 2 和通道 3 的 PWM 计数器用于捕捉功能
[28]	CH2RFORDER	设置该位用于决定当 PDMACAPMOD2=2'b11 时, 是 PWM_CRL2 还是 PWM_CFL2 作为捕捉到的数据首先通过 PDMA 传输到内存 1 = PWM_CRL2 首先传输到内存 0 = PWM_CFL2 首先传输到内存
[27]	-	保留
[26]	CAPCH3PADEN	Capture 输入使能 0 = OFF 1 = ON
[25]	CAPCH3EN	通道 3 Capture 功能使能/禁止 1 = 使能通道 3 捕捉功能 0 = 禁止通道 3 捕捉功能 当该位使能, 捕捉器锁存 PWM 定时器的值并保存到 PWM_CRL3 (上升沿锁存) 和

Bits	描述									
		PWM_CFL3 (下降沿锁存) 当该位禁止，捕捉器不更新 PWM_CRL3 和 PWM_CFL3，并禁止通道 3 中断.								
[24]	INV3	通道 3 反向开/关 1 = 反向开。在输入给捕捉器定时器之前，反转从 GPIO 输入的信号。 0 = 反向关。								
[23]	CAPRELOADFEN2	当 CH2 下降沿捕捉事件发生时，重载CNR2 1 = 使能 CH2 下降沿捕捉重载 0 = 禁止 CH2 下降沿捕捉重载								
[22]	CAPRELOADREN2	当 CH2 上升沿捕捉事件发生时，重载CNR2 1 = 使能 CH2 上升沿捕捉重载 0 = 禁止 CH2 上升沿捕捉重载								
[21:20]	PDMA CAPMOD2[1:0]	选择 CRL2 或者 CFL2 来进行 PDMA 传输 <table><tr><td>00</td><td>保留</td></tr><tr><td>01</td><td>CRL2</td></tr><tr><td>10</td><td>CFL2</td></tr><tr><td>11</td><td>CRL2 和 CFL2</td></tr></table>	00	保留	01	CRL2	10	CFL2	11	CRL2 和 CFL2
00	保留									
01	CRL2									
10	CFL2									
11	CRL2 和 CFL2									
[19]	CH2PDMAEN	通道 2 PDMA 使能 1 = 使能通道 2 PDMA 功能来传输捕捉到的数据到内存 0 = 禁止通道 2 PDMA 功能								
[18]	CAPCH2PADEN	Capture 输入使能 0 = OFF 1 = ON								
[17]	CAPCH2EN	通道 2 Capture 功能使能/禁止 1 = 使能通道 2 捕捉功能 0 = 禁止通道 2 捕捉功能 当该位使能，捕捉器锁存 PWM 定时器的值并保存到 PWM_CRL2 (上升沿锁存) 和 PWM_CFL2 (下降沿锁存) 当该位禁止，捕捉器不更新 PWM_CRL2 和 PWM_CFL2，并禁止通道 2 中断.								
[16]	INV2	通道 2 反向开/关 1 = 反向开。在输入给捕捉器定时器之前，反转从 GPIO 输入的信号。								

Bits	描述					
		0 = 反向关。				
[15]	CAPRELOADFEN1	当 CH1 下降沿捕捉事件发生时，重载CNR1 1 = 使能 CH1 下降沿捕捉重载 0 = 禁止 CH1 下降沿捕捉重载				
[14]	CAPRELOADREN1	当 CH1 上升沿捕捉事件发生时，重载CNR1 1 = 使能 CH1 上升沿捕捉重载 0 = 禁止 CH1 上升沿捕捉重载				
[13]	CH01CASK	级联通道 0 和通道 1 的 PWM 计数器用于捕捉功能				
[12]	CH0RFORDER	设置该位用于决定当 PDMACAPMOD0=2'b11 时，是 PWM_CRL0 还是 PWM_CFL0 作为捕捉到的数据首先通过 PDMA 传输到内存 1 = PWM_CRL0 首先传输到内存 0 = PWM_CFL0 首先传输到内存				
[11]	-	保留				
[10]	CAPCH1PADEN	Capture 输入使能 0 = OFF 1 = ON				
[9]	CAPCH1EN	通道 1 Capture 功能使能/禁止 1 = 使能通道 1 捕捉功能 0 = 禁止通道 1 捕捉功能 当该位使能，捕捉器锁存 PWM 定时器的值并保存到 PWM_CRL1 (上升沿锁存) 和 PWM_CFL1 (下降沿锁存) 当该位禁止，捕捉器不更新 PWM_CRL1 和 PWM_CFL1，并禁止通道 1 中断。				
[8]	INV1	通道 1 反向开关 1 = 反向开。在输入给捕捉器定时器之前，反转从 GPIO 输入的信号。 0 = 反向关。				
[7]	CAPRELOADFEN0	当 CH0 下降沿捕捉事件发生时，重载CNR0 1 = 使能 CH0 下降沿捕捉重载 0 = 禁止 CH0 下降沿捕捉重载				
[6]	CAPRELOADREN0	当 CH0 上升沿捕捉事件发生时，重载CNR0 1 = 使能 CH0 上升沿捕捉重载 0 = 禁止 CH0 上升沿捕捉重载				
[5:4]	PDMACAPMOD0[1:0]	选择 CRL0 或者 CFL0 来进行 PDMA 传输 <table><tr><td>00</td><td>保留</td></tr><tr><td>01</td><td>CRL0</td></tr></table>	00	保留	01	CRL0
00	保留					
01	CRL0					

Bits	描述		
		10	CFL0
		11	CRL0 和 CFL0
[3]	CH0PDMAEN	通道 0 PDMA 使能 1 = 使能通道 0 PDMA 功能来传输捕捉到的数据到内存 0 = 禁止通道 0 PDMA 功能	
[2]	CAPCH0PADEN	Capture 输入使能 0 = OFF 1 = ON	
[1]	CAPCH0EN	通道 0 Capture 功能使能/禁止 1 = 使能通道 0 捕捉功能 0 = 禁止通道 0 捕捉功能 当该位使能，捕捉器锁存 PWM 定时器的值并保存到 PWM_CRL0 (上升沿锁存) 和 PWM_CFL0 (下降沿锁存) 当该位禁止，捕捉器不更新 PWM_CRL0 和 PWM_CFL0，并禁止通道 0 中断。	
[0]	INV0	通道 0 反向开/关 1 = 反向开。在输入给捕捉器定时器之前，反转从 GPIO 输入的信号。 0 = 反向关。	

Capture 中断使能寄存器 (PWMx_CAPINTEN)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CAPINTEN	PWMx_BA+0x058	R/W	Capture中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-						CFL_IE3	CRL_IE3
23	22	21	20	19	18	17	16
-						CFL_IE2	CRL_IE2
15	14	13	12	11	10	9	8
-						CFL_IE1	CRL_IE1
7	6	5	4	3	2	1	0
-						CFL_IE0	CRL_IE0

Bits	描述	
[31:26]	-	保留
[25]	CFL_IE3	通道 3 下降沿锁存中断使能开/关 1 = 使能下降沿锁存中断 0 = 禁止下降沿锁存中断 当该位使能, 如果捕捉器检测到在通道 3 上有一个下降沿跳变, 捕捉器触发一次中断.
[24]	CRL_IE3	通道 3 上升沿锁存中断使能开/关 1 = 使能上升沿锁存中断 0 = 禁止上升沿锁存中断 当该位使能, 如果捕捉器检测到在通道 3 上有一个上升沿跳变, 捕捉器触发一次中断
[23:18]	-	保留
[17]	CFL_IE2	通道 2 下降沿锁存中断使能开/关 1 = 使能下降沿锁存中断 0 = 禁止下降沿锁存中断 当该位使能, 如果捕捉器检测到在通道 2 上有一个下降沿跳变, 捕捉器触发一次中断.
[16]	CRL_IE2	通道 2 上升沿锁存中断使能开/关 1 = 使能上升沿锁存中断 0 = 禁止上升沿锁存中断 当该位使能, 如果捕捉器检测到在通道 2 上有一个上升沿跳变, 捕捉器触发一次中断
[15:10]	-	保留
[9]	CFL_IE1	通道 1 下降沿锁存中断使能开/关

Bits	描述	
		<p>1 = 使能下降沿锁存中断</p> <p>0 = 禁止下降沿锁存中断</p> <p>当该位使能，如果捕捉器检测到在通道 1 上有一个下降沿跳变，捕捉器触发一次中断。</p>
[8]	CRL_IE1	<p>通道 1 上升沿锁存中断使能开/关</p> <p>1 = 使能上升沿锁存中断</p> <p>0 = 禁止上升沿锁存中断</p> <p>当该位使能，如果捕捉器检测到在通道 1 上有一个上升沿跳变，捕捉器触发一次中断</p>
[7:2]	-	保留
[1]	CFL_IE0	<p>通道 0 下降沿锁存中断使能开/关</p> <p>1 = 使能下降沿锁存中断</p> <p>0 = 禁止下降沿锁存中断</p> <p>当该位使能，如果捕捉器检测到在通道 0 上有一个下降沿跳变，捕捉器触发一次中断。</p>
[0]	CRL_IE0	<p>通道 0 上升沿锁存中断使能开/关</p> <p>1 = 使能上升沿锁存中断</p> <p>0 = 禁止上升沿锁存中断</p> <p>当该位使能，如果捕捉器检测到在通道 0 上有一个上升沿跳变，捕捉器触发一次中断</p>

Capture 中断状态寄存器 (PWMx_CAPINTSTS)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CAPINTSTS	PWMx_BA+0x05C	R/W	Capture 中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-			CAPOVF3	CAPOVR3	CFLI3	CRLI3	CAPIF3
23	22	21	20	19	18	17	16
-			CAPOVF2	CAPOVR2	CFLI2	CRLI2	CAPIF2
15	14	13	12	11	10	9	8
-			CAPOVF1	CAPOVR1	CFLI1	CRLI1	CAPIF1
7	6	5	4	3	2	1	0
-			CAPOVF0	CAPOVR0	CFLI0	CRLI0	CAPIF0

Bits	描述	
[31:29]	-	保留
[28]	CAPOVF3	Capture 通道 3 下降捕捉锁存寄存器更新过快标志 当该标志被置位时，表示 CFLI3 被更新速度快于被软件读取速度 当用户清除 PWM_CAPINTSTS 寄存器的 CFLI3 位时，该标志被自动清除
[27]	CAPOVR3	Capture 通道 3 上升捕捉锁存寄存器更新过快标志 当该标志被置位时，表示 CRLI3 被更新速度快于被软件读取速度 当用户清除 PWM_CAPINTSTS 寄存器的 CRLI3 位时，该标志被自动清除
[26]	CFLI3	PWM_CFLI3 锁存指示标志 当输入通道 3 有一个下降沿跳变，PWM 下数型计数器的值被锁存到 PWM_CFLI3。该标志由硬件自动置位，软件可以向该位写 1 清除该标志
[25]	CRLI3	PWM_CRLI3 锁存指示标志 当输入通道 3 有一个上升沿跳变，PWM 下数型计数器的值被锁存到 PWM_CRLI3。该标志由硬件自动置位，软件可以向该位写 1 清除该标志
[24]	CAPIF3	Capture 通道 3 中断指示标志 如果通道 3 上升沿锁存中断被使能 (CRL_IE3=1)，在输入通道 3 上有一个上升沿跳变发生时，会导致 CAPIF3 被置为高；类似的，如果通道 3 下降沿锁存中断被使能 (CFL_IE3=1)，在输入通道 3 上有一个下降沿跳变发生时，会导致 CAPIF3 被置为高。该标志可以通过软件

Bits	描述	
		向该位写 1 来清除。
[23:21]	-	保留
[20]	CAPOVF2	<p>Capture 通道 2 下降捕捉锁存寄存器更新过快标志</p> <p>当该标志被置位时，表示 CFL2 被更新速度快于被软件读取速度</p> <p>当用户清除 PWM_CAPINTSTS 寄存器的 CFLI2 位时，该标志被自动清除</p>
[19]	CAPOVR2	<p>Capture 通道 2 上升捕捉锁存寄存器更新过快标志</p> <p>当该标志被置位时，表示 CRL2 被更新速度快于被软件读取速度</p> <p>当用户清除 PWM_CAPINTSTS 寄存器的 CRLI2 位时，该标志被自动清除</p>
[18]	CFLI2	<p>PWM_CFL2 锁存指示标志</p> <p>当输入通道 2 有一个下降沿跳变，PWM 下数型计数器的值被锁存到 PWM_CFL2。该标志由硬件自动置位，软件可以向该位写 1 清除该标志</p>
[17]	CRLI2	<p>PWM_CRL2 锁存指示标志</p> <p>当输入通道 2 有一个上升沿跳变，PWM 下数型计数器的值被锁存到 PWM_CRL2。该标志由硬件自动置位，软件可以向该位写 1 清除该标志</p>
[16]	CAPIF2	<p>Capture 通道 2 中断指示标志</p> <p>如果通道 2 上升沿锁存中断被使能 (CRL_IE2=1)，在输入通道 2 上有一个上升沿跳变发生时，会导致 CAPIF2 被置为高；类似的，如果通道 2 下降沿锁存中断被使能 (CFL_IE2=1)，在输入通道 2 上有一个下降沿跳变发生时，会导致 CAPIF2 被置为高。该标志可以通过软件向该位写 1 来清除。</p>
[15:13]	-	保留
[12]	CAPOVF1	<p>Capture 通道 1 下降捕捉锁存寄存器更新过快标志</p> <p>当该标志被置位时，表示 CFL1 被更新速度快于被软件读取速度</p> <p>当用户清除 PWM_CAPINTSTS 寄存器的 CFLI1 位时，该标志被自动清除</p>
[11]	CAPOVR1	<p>Capture 通道 1 上升捕捉锁存寄存器更新过快标志</p> <p>当该标志被置位时，表示 CRL1 被更新速度快于被软件读取速度</p> <p>当用户清除 PWM_CAPINTSTS 寄存器的 CRLI1 位时，该标志被自动清除</p>
[10]	CFLI1	<p>PWM_CFL1 锁存指示标志</p> <p>当输入通道 1 有一个下降沿跳变，PWM 下数型计数器的值被锁存到 PWM_CFL1。该标志由硬件自动置位，软件可以向该位写 1 清除该标志</p>

Bits	描述	
[9]	CRL1	PWM_CRL1 锁存指示标志 当输入通道 1 有一个上升沿跳变，PWM 下数型计数器的值被锁存到 PWM_CRL1。该标志由硬件自动置位，软件可以向该位写 1 清除该标志
[8]	CAPIF1	Capture 通道 1 中断指示标志 如果通道 1 上升沿锁存中断被使能 (CRL_IE1=1)，在输入通道 1 上有一个上升沿跳变发生时，会导致 CAPIF1 被置为高；类似的，如果通道 1 下降沿锁存中断被使能 (CFL_IE1=1)，在输入通道 1 上有一个下降沿跳变发生时，会导致 CAPIF1 被置为高。该标志可以通过软件向该位写 1 来清除。
[7:5]	-	保留
[4]	CAPOVF0	Capture 通道 0 下降捕捉锁存寄存器更新过快标志 当该标志被置位时，表示 CFL0 被更新速度快于被软件读取速度 当用户清除 PWM_CAPINTSTS 寄存器的 CFLI0 位时，该标志被自动清除
[3]	CAPOVR0	Capture 通道 0 上升捕捉锁存寄存器更新过快标志 当该标志被置位时，表示 CRL0 被更新速度快于被软件读取速度 当用户清除 PWM_CAPINTSTS 寄存器的 CRLI0 位时，该标志被自动清除
[2]	CFLR10	PWM_CFL0 锁存指示标志 当输入通道 0 有一个下降沿跳变，PWM 下数型计数器的值被锁存到 PWM_CFL0。该标志由硬件自动置位，软件可以向该位写 1 清除该标志
[1]	CRLI0	PWM_CRL0 锁存指示标志 当输入通道 0 有一个上升沿跳变，PWM 下数型计数器的值被锁存到 PWM_CRL0。该标志由硬件自动置位，软件可以向该位写 1 清除该标志
[0]	CAPIF0	Capture 通道 0 中断指示标志 如果通道 0 上升沿锁存中断被使能 (CRL_IE1=1)，在输入通道 0 上有一个上升沿跳变发生时，会导致 CAPIF0 被置为高；类似的，如果通道 0 下降沿锁存中断被使能 (CFL_IE0=1)，在输入通道 0 上有一个下降沿跳变发生时，会导致 CAPIF0 被置为高。该标志可以通过软件向该位写 1 来清除。

Capture 上升沿锁存寄存器 3-0 (PWMx CRL3-0)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CRL0 x = 0,1	PWMx_BA+0x060	R	Capture 上升沿锁存寄存器 (channel 0)	0x0000_0000
PWMx_CRL1 x = 0,1	PWMx_BA+0x068	R	Capture 上升沿锁存寄存器 (channel 1)	0x0000_0000
PWMx_CRL2 x = 0,1	PWMx_BA+0x070	R	Capture 上升沿锁存寄存器 (channel 2)	0x0000_0000
PWMx_CRL3 x = 0,1	PWMx_BA+0x078	R	Capture 上升沿锁存寄存器 (channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CRL (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)							
23	22	21	20	19	18	17	16
CRL (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)							
15	14	13	12	11	10	9	8
CRL							
7	6	5	4	3	2	1	0
CRL							

Bits	描述	
[31:16]	CRL[31:16]	当级联被使能时, 32 位捕捉数据的高半字 当捕捉通道 0, 2 的级联功能被使能时, 原先的 16 位计数器被扩展为 32 位, 捕捉结果 CRL0 和 CRL2 也被扩展为 32 位。
[15:0]	CRL[15:0]	Capture 上升沿锁存寄存器 当通道 0/1/2/3 上有一个上升沿跳变时, 锁存 PWM 计数器的值。

PWM Capture 下降沿锁存寄存器 3-0 (PWMx CFL3-0)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_CFL0	PWMx_BA+0x064	R	Capture 下降沿锁存寄存器 (channel 0)	0x0000_0000
PWMx_CFL1	PWMx_BA+0x06C	R	Capture 下降沿锁存寄存器 (channel 1)	0x0000_0000
PWMx_CFL2	PWMx_BA+0x074	R	Capture 下降沿锁存寄存器 (channel 2)	0x0000_0000
PWMx_CFL3	PWMx_BA+0x07C	R	Capture 下降沿锁存寄存器 (channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CFL (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)							
23	22	21	20	19	18	17	16
CFL (仅对于通道 0 和 2 在相应的级联使能被设置时有意义)							
15	14	13	12	11	10	9	8
CFL							
7	6	5	4	3	2	1	0
CFL							

Bits	描述	
[31:16]	CFL[31:16]	当级联被使能时, 32 位捕捉数据的高半字 当捕捉通道 0, 2 的级联功能被使能时, 原先的 16 位计数器被扩展为 32 位, 捕捉结果 CFL0 和 CFL2 也被扩展为 32 位。
[15:0]	CFL[15:0]	Capture 下降沿锁存寄存器 当通道 0/1/2/3 上有一个下降沿跳变时, 锁存 PWM 计数器的值。

PWMx_PDMACH0 (Capture 通道 0 PDMA 数据寄存器)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_PDMACH0	PWMx_BA+0x80	R/W	CH0 PDMA 捕捉数据	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH0							
23	22	21	20	19	18	17	16
PDMACH0							
15	14	13	12	11	10	9	8
PDMACH0							
7	6	5	4	3	2	1	0
PDMACH0							

Bits	描述	
[31:24]	Captured data[31:24]	PDMACH0 当 CH01CASK 被禁止时, 该字节为 0 当 CH01CASK 被使能时, 作为通道 0 的 32 位捕捉数据的第四字节
[23:16]	Captured data[23:16]	PDMACH0 当 CH01CASK 被禁止时, 该字节为 0 当 CH01CASK 被使能时, 作为通道 0 的 32 位捕捉数据的第三个字节
[15:8]	Captured data[15:8]	PDMACH0 当 CH01CASK 被禁止时, 作为通道 0 的捕获值 (CFL0/CRL0) 当 CH01CASK 被使能时, 作为通道 0 的 32 位捕捉数据的第二个字节
[7:0]	Captured data[7:0]	PDMACH0 当 CH01CASK 被禁止时, 作为通道 0 的捕获值 (CFL0/CRL0) 当 CH01CASK 被使能时, 作为通道 0 的 32 位捕捉数据的第一个字节

PWMx_PDMACH2 (Capture 通道 0 PDMA 数据寄存器)

寄存器	偏移量	R/W	描述	复位后的值
PWMx_PDMACH2	PWMx_BA+0x84	R/W	CH2 PDMA 捕捉数据	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH2							
23	22	21	20	19	18	17	16
PDMACH2							
15	14	13	12	11	10	9	8
PDMACH2							
7	6	5	4	3	2	1	0
PDMACH2							

Bits	描述	
[31:24]	Captured data[31:24]	PDMACH2 当 CH23CASK 被禁止时, 该字节为 0 当 CH23CASK 被使能时, 作为通道 2 的 32 位捕捉数据的第四字节
[23:16]	Captured data[23:16]	PDMACH2 当 CH23CASK 被禁止时, 该字节为 0 当 CH23CASK 被使能时, 作为通道 2 的 32 位捕捉数据的第三个字节
[15:8]	Captured data[15:8]	PDMACH2 当 CH23CASK 被禁止时, 作为通道 2 的捕获值 (CFL2/CRL2) 当 CH23CASK 被使能时, 作为通道 2 的 32 位捕捉数据的第二个字节
[7:0]	Captured data[7:0]	PDMACH2 当 CH23CASK 被禁止时, 作为通道 2 的捕获值 (CFL2/CRL2) 当 CH23CASK 被使能时, 作为通道 2 的 32 位捕捉数据的第一个字节

5.15 RTC

5.15.1 概述

实时时钟 (RTC) 单元提供给用户实时时间以及日历信息。RTC 的时钟源由外部 32.768 kHz 晶振提供, 连接管脚为 X32I 和 X32O (请参考管脚描述) 或者管脚 X32I 外接 32.768 kHz 振荡器输出信号源。RTC 单元通过时间载入寄存器 (TLR) 提供时间信息 (秒、分、时), 并通过日历载入寄存器 (CLR) 提供日历信息 (日、月、年)。数据信息由 BCD 码格式进行表示。该单元也提供闹钟功能, 用户可以预先在时间闹钟寄存器 (TAR) 中设置闹钟时间、日历闹钟寄存器 (CAR) 中设置闹钟日期来进行闹钟设置。

RTC 单元支持周期时间节拍中断和闹钟匹配中断。通过设定 TTR (TTR[2:0]), 周期中断有 8 个周期选项 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 及 1 秒。当 TLR 和 CLR 中 RTC 计数器的值分别等于闹钟设定时间寄存器 TAR 和 CAR 时, 闹钟中断标志 (RIIR.AIF) 被置位, 如果此时闹钟中断已经被使能 (RIER.AIER=1), 将同时产生闹钟中断请求。如果唤醒 CPU 功能被使能 (RTC_TTR[TWKE] = 1), RTC 时间节拍或者闹钟匹配能够将 CPU 从休眠或掉电模式中唤醒。

5.15.2 特性

- 提供时间计数器 (秒, 分, 时) 和日历计数器 (日, 月, 年), 供用户用来查看时间
- 闹钟寄存器 (秒, 分, 时, 日, 月, 年)
- 12-小时或 24-小时模式可选择
- 闰年自动补偿
- 星期计数器
- 频率补偿寄存器 (FCR)
- 所有时间和日期信息由 BCD 码表示
- 支持周期时间节拍中断, 提供 8 个周期选项: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 和 1 秒
- 支持 RTC 时间节拍和闹钟匹配中断
- 支持从掉电模式下唤醒 CPU
- 支持 80 字节备用寄存器和一个用于清除这些备用寄存器内容的监听引脚

5.15.3 框图

RTC 模块框图如下

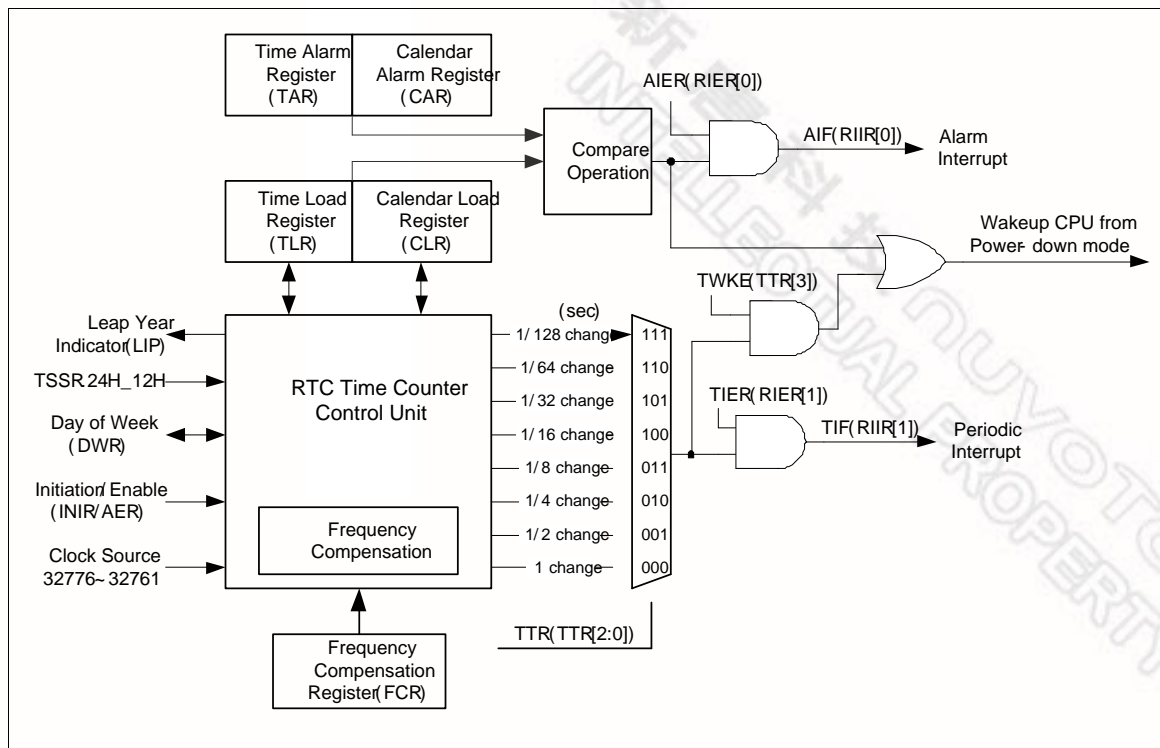


图 5-82 RTC 框图

5.15.4 功能描述

5.15.4.1 访问 RTC 寄存器

由于 RTC 时钟和系统时钟的不同，当用户对任一 RTC 寄存器进行写入时，必须等待2个 RTC 时钟周期 (60us)，寄存器内的值才会被更新。

此外，用户必须注意 RTC 模块不检查载入的数据是否越界。RTC 也不会对 DWR 和 CLR 的合理性进行检查。

5.15.4.2 RTC 初始化

当 RTC 模块上电后，用户必须写数据 (0xa5eb1357) 到 INIR 去复位所有 RTC 逻辑电路。INIR 作为硬件复位电路，一旦 INIR 被设置为 0xa5eb1357，任何值被编程写入 INIR 寄存器，都不会对 RTC 有任何作用。

5.15.4.3 RTC 读/写使能

寄存器 AER 位15~0 作为保护 RTC 寄存器读/写的密码。AER 位15~0 必须被设置为 0xa965 来解除访问限制。一旦该寄存器被设置，将会至少有效 512 个 RTC 时钟 (大约15ms)。用户可以通过读取 RTC 使能标志位 AER.ENF 来确认 RTC 是否开始工作。

5.15.4.4 频率补偿

RTC_FCR 允许软件对时钟输入进行数字补偿。时钟输入的频率必须在 32776 Hz 到 32761 Hz 范围内。RTC频率补偿周期每隔60秒循环一次，对透过RTC_FCR实施晶振输入误差补偿，可实现最高的RTC 频率补偿精度为±0.4768 ppm，对应的时间误差为±0.0412秒/日。用户仅需将32k 晶振实际频率按表列公式填入RTC_FCR寄存器即可，下例为高於或低於32768 Hz对时钟输入进行补偿的示例。

范例1：

频率计数测量值：32773.65 Hz (> 32768 Hz)

整数部分: 32773

$FCR.Integer = (32773 - 32768) + 7 = 12 = 0x0c$

分数部分: 0.65

$FCR.Fraction = 0.65 \times 64 = 41.6 \Rightarrow 0x29$

范例2：

频率计数测量值：32763.25 Hz (< 32768 Hz)

整数部分: 32763

$FCR.Integer = (32763 - 32768) + 7 = 2 = 0x02$

分数部分: 0.25

$FCR.Fraction = 0.25 \times 64 = 16 \Rightarrow 0x10$

注：

当RTC_FCR寄存器为默认值(0x0000_0700)表示RTC尚未进行频率补偿，用户可测量RTC时钟源。因此，在生产过程中，用户可以利用一个通用I/O引脚输出RTC时间节拍中断事件，再透过通用I/O测量RTC 时钟源，并将RTC时钟源数值储存在Flash内存中，用于当产品第一次上电时进行检测。

索。同理，当RTC_FCR寄存器设定之后，用户可利用RTC时间节拍中断事件来检查RTC频率补偿结果。

5.15.4.5 时间和日历计数器

TLR 和 CLR 用于载入时间和日历，TAR 和 CAR 用于闹钟。它们均由 BCD 格式表示。

5.15.4.6 12/24 小时时标选择

根据 TSSR 位 0 选择12/24小时时标。

5.15.4.7 星期计数器

RTC 单元通过星期寄存器 (DWR) 来提供星期信息。定义的值为由 0 至 6，用于表示周日至周六。

5.15.4.8 周期时间节拍中断

周期时间节拍中断有 8 个选项：1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒，通过设置 TTR.TTR[2:0] 来进行选择。当 RIER.TIER 被置 1，周期时间节拍中断使能，周期时间节拍中断将会根据 TTR 寄存器内选定的值周期性的被请求。

5.15.4.9 闹钟中断

当 TLR 和 CLR 中的 RTC 计数器等于 TAR 和 CAR 中的设定值，则闹钟中断标志 (RIIR.AIF) 被置位，如果此时闹钟中断已使能 (RIER.AIER=1)，将同时发出闹钟中断请求。

应用指南

TAR, CAR, TLR 和 CLR 寄存器均为 BCD 格式计数器。

程序设计者必须保证载入值为有效合理的。例如，CLR = 201a (年), 13 (月), 00 (日)，或 CLR 与 DWR 不匹配，等等。

复位状态:

寄存器	复位状态
AER	0
CLR	05/1/1 (年/月/日)
TLR	00:00:00 (时 : 分 : 秒)
CAR	00/00/00 (年/月/日)
TAR	00:00:00 (时 : 分 : 秒)
TSSR	1 (24 小时制)
DWR	6 (星期六)
RIER	0
RIIR	0
LIR	0
TTR	0

在 TLR 和 TAR 中，仅 2 位 BCD 码用于指示“年”。我们假设两位 BCD 码 xY 表示20xY年，而

不是 19xY 或 21xY 年。

5.15.4.10 备用寄存器和监听引脚

RTC 配备有 80 字节的备用寄存器，用于存储用户的重要信息，并且配备有一个监听功能用于检测在监听引脚上的电平跳变，一旦在监听引脚上检测到在 RTC_SPRCTL[1] 中定义的跳变信号，80 字节的备用寄存器将会被 RTC 自动清除。

因为这 80 字节的备用寄存器位于 LXT (32.768KHz) 的时钟域内 (该时钟域异步于系统时钟域)，所以当写数据到这 80 字节的备用寄存器时，一个同步延时是必须的。一旦 CPU 写 20 个备用寄存器 (RTC_SPR0 ~ RTC_SPR19) 中的任意一个时，用户需要去查询 SPRRDY (RTC_SPRCTL[7]) 位来确认数据是否已经写入该寄存器。CPU 只有在 SPRRDY 为 1 时，才能再次访问 (读或写) 这些备用寄存器，在 SPRRDY 为 0 时，任何对备用寄存器的访问 (读或写) 都是未定义的。

5.15.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write, C: write 1 to clear

寄存器	偏移量	R/W	描述	复位后的值
RTC_BA = 0x4000_8000				
RTC_INIR	RTC_BA+0x00	R/W	RTC 初始化寄存器	0x0000_0000
RTC_AER	RTC_BA+0x04	R/W	RTC 访问使能寄存器	0x0000_0000
RTC_FCR	RTC_BA+0x08	R/W	RTC 频率补偿寄存器	0x0000_0700
RTC_TLR	RTC_BA+0x0C	R/W	时间载入寄存器	0x0000_0000
RTC_CLR	RTC_BA+0x10	R/W	日历载入寄存器	0x0005_0101
RTC_TSSR	RTC_BA+0x14	R/W	时间格式选择寄存器	0x0000_0001
RTC_DWR	RTC_BA+0x18	R/W	星期寄存器	0x0000_0006
RTC_TAR	RTC_BA+0x1C	R/W	时间闹钟寄存器	0x0000_0000
RTC_CAR	RTC_BA+0x20	R/W	日历闹钟寄存器	0x0000_0000
RTC_LIR	RTC_BA+0x24	R	闰年指示寄存器	0x0000_0000
RTC_RIER	RTC_BA+0x28	R/W	RTC 中断使能寄存器	0x0000_0000
RTC_RIIR	RTC_BA+0x2C	R/W	RTC 中断指示寄存器	0x0000_0000
RTC_TTR	RTC_BA+0x30	R/W	RTC 时间节拍寄存器	0x0000_0000
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC 备用功能控制寄存器	0x0000_0080
RTC_SPR0	RTC_BA+0x40	R/W	RTC 备用寄存器 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC 备用寄存器 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC 备用寄存器 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC 备用寄存器 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC 备用寄存器 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC 备用寄存器 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC 备用寄存器 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC 备用寄存器 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC 备用寄存器 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC 备用寄存器 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC 备用寄存器 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC 备用寄存器 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC 备用寄存器 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC 备用寄存器 13	0x0000_0000

寄存器	偏移量	R/W	描述	复位后的值
RTC_SPR14	RTC_BA+0x78	R/W	RTC 备用寄存器 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC 备用寄存器 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC 备用寄存器 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC 备用寄存器 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC 备用寄存器 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC 备用寄存器 19	0x0000_0000

5.15.6 寄存器描述

RTC 初始化寄存器 (RTC INIR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_INIR	RTC_BA+0x00	R/W	RTC 初始化寄存器	0x0000_0000

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

Bits	描述	
[31:1]	INIR[31:1]	RTC 初始化 当 RTC 模块上电后, RTC 处于复位状态。用户需要写入数据 (0xa5eb1357) 到 INIR 使得 RTC 离开复位状态。一旦 INIR 被写入 0xa5eb1357, RTC 将会一直处于非复位状态。 INIR[31:1] 是只写位域, 读取该位域将总是返回 0
[0]	INIR[0]/ACTIVE	RTC 活动状态 (只读) 0 = RTC 处在复位状态 1 = RTC 处在正常有效状态

RTC 访问使能寄存器 (RTC_AER)

寄存器	偏移量	R/W	描述	复位后的值
RTC_AER	RTC_BA+0x04	R/W	RTC 访问使能寄存器	0x0000_0000

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

Bits	描述																																											
[31:17]	-	保留																																										
[16]	ENF	<p>RTC 寄存器访问使能标志 (只读)</p> <p>1 = RTC 寄存器读/写使能</p> <p>0 = RTC 寄存器读/写禁用</p> <p>对 AER[15:0] 载入 0xA965 后, 该位将被置。在 512 个 RTC 时钟后, 或者当 AER[15:0] 不等于 0xA965 时, 该位自动被清除。</p> <table> <tr> <th>Register</th><th>AER[16] = 1</th><th>AER[16] = 0</th></tr> <tr> <td>INIR</td><td>R/W</td><td>R/W</td></tr> <tr> <td>AER</td><td>R/W</td><td>R/W</td></tr> <tr> <td>FCR</td><td>R/W</td><td>-</td></tr> <tr> <td>TLR</td><td>R/W</td><td>R</td></tr> <tr> <td>CLR</td><td>R/W</td><td>R</td></tr> <tr> <td>TSSR</td><td>R/W</td><td>R/W</td></tr> <tr> <td>DWR</td><td>R/W</td><td>R</td></tr> <tr> <td>TAR</td><td>R/W</td><td>-</td></tr> <tr> <td>CAR</td><td>R/W</td><td>-</td></tr> <tr> <td>LIR</td><td>R</td><td>R</td></tr> <tr> <td>RIER</td><td>R/W</td><td>R/W</td></tr> <tr> <td>RIIR</td><td>R/W</td><td>R/W</td></tr> <tr> <td>TTR</td><td>R/W</td><td>-</td></tr> </table>	Register	AER[16] = 1	AER[16] = 0	INIR	R/W	R/W	AER	R/W	R/W	FCR	R/W	-	TLR	R/W	R	CLR	R/W	R	TSSR	R/W	R/W	DWR	R/W	R	TAR	R/W	-	CAR	R/W	-	LIR	R	R	RIER	R/W	R/W	RIIR	R/W	R/W	TTR	R/W	-
Register	AER[16] = 1	AER[16] = 0																																										
INIR	R/W	R/W																																										
AER	R/W	R/W																																										
FCR	R/W	-																																										
TLR	R/W	R																																										
CLR	R/W	R																																										
TSSR	R/W	R/W																																										
DWR	R/W	R																																										
TAR	R/W	-																																										
CAR	R/W	-																																										
LIR	R	R																																										
RIER	R/W	R/W																																										
RIIR	R/W	R/W																																										
TTR	R/W	-																																										

		SPRCTL	R/W	-	
		SPR0~SPR19	R/W	-	
[15:0]	AER[15:0]	RTC 寄存器访问使能密码 (只写) 0xA965 = 使能 RTC 读写 Others = 禁用 RTC 读写			

RTC 频率补偿寄存器 (RTC_FCR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_FCR	RTC_BA+0x08	R/W	频率补偿寄存器	0x0000_0700

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

Bits	描述			
[31:12]	-	保留		
[11:8]	INTEGER[3:0]	整数部分		
		检测到的值的整数部分	FCR[11:8]	检测到的值的整数部分
		32776	1111	32768
		32775	1110	32767
		32774	1101	32766
		32773	1100	32765
		32772	1011	32764
		32771	1010	32763
		32770	1001	32762
		32769	1000	32761
[7:6]	-	保留		
[5:0]	FRACTION[5:0]	小数部分		
		公式 = (检测到的值的小数部分) x 64		
		注: FCR 的值必须按照 16 进制格式表示。		

注: 在 RTC 访问通过 AER 被使能后, 该寄存器的值可以被读回。

RTC 时间载入寄存器 (RTC TLR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_TLR	RTC_BA+0x0C	R/W	时间载入寄存器	0x0000_0000

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

Bits	描述	
[31:22]	-	保留
[21:20]	10HR	10 Hour Time Digit (0~2)
[19:16]	1HR	1 Hour Time Digit (0~9)
[14:12]	10MIN	10 Min Time Digit (0~5)
[11:8]	1MIN	1 Min Time Digit (0~9)
[6:4]	10SEC	10 Sec Time Digit (0~5)
[3:0]	1SEC	1 Sec Time Digit (0~9)

注: TLR 为 BCD 码格式的计数器, RTC 不会对载入值进行检测.

括号内列出的为可接受的值.

RTC 日历载入寄存器 (RTC_CLR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_CLR	RTC_BA+0x10	R/W	日历载入寄存器	0x0005_0101

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

Bits	描述	
[31:24]	-	保留
[23:20]	10YEAR	10 Year Calendar Digit (0~9)
[19:16]	1YEAR	1 Year Calendar Digit (0~9)
[12]	10MON	10 Month Calendar Digit (0~1)
[11:8]	1MON	1 Month Calendar Digit (0~9)
[5:4]	10DAY	10 Day Calendar Digit (0~3)
[3:0]	1DAY	1 Day Calendar Digit (0~9)

注: CLR 为 BCD 码格式的计数器, RTC 不会检测载入值

括号内列出的为可接受的值.

RTC 时间格式选择寄存器 (RTC_TSSR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_TSSR	RTC_BA+0x14	R/W	时间格式选择寄存器	0x0000_0001

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

Bits	描述				
[31:1]	-	保留			
[0]	24hr/12hr	24-小时/ 12-小时 模式选择 用于表示 TLR 和 TAR 为 24-小时 或 12-小时模式 1 = 选择 24-小时制 0 = 选择 12-小时制，带AM 和 PM 指示			
		24-小时制	12-小时制	24-小时制	12-小时制 (PM 时间 + 20)
		00	12(AM12)	12	32(PM12)
		01	01 (AM01)	13	21 (PM01)
		02	02(AM02)	14	20(PM02)
		03	03(AM03)	15	23(PM03)
		04	04 (AM04)	16	24 (PM04)
		05	05(AM05)	17	25(PM05)
		06	06(AM06)	18	26(PM06)
		07	07(AM07)	19	27(PM07)
		08	08(AM08)	20	28(PM08)
		09	09(AM09)	21	29(PM09)
		10	10 (AM10)	22	30 (PM10)
		11	11 (AM11)	23	31 (PM11)

RTC 星期寄存器 (RTC_DWR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_DWR	RTC_BA+0x18	R/W	星期寄存器	0x0000_0006

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

Bits	描述																	
[31:3]	-	保留																
[2:0]	DWR[2:0]	星期寄存器																
		<table><tr><th>DWR</th><th>星期</th></tr><tr><td>0</td><td>星期日</td></tr><tr><td>1</td><td>星期一</td></tr><tr><td>2</td><td>星期二</td></tr><tr><td>3</td><td>星期三</td></tr><tr><td>4</td><td>星期四</td></tr><tr><td>5</td><td>星期五</td></tr><tr><td>6</td><td>星期六</td></tr></table>	DWR	星期	0	星期日	1	星期一	2	星期二	3	星期三	4	星期四	5	星期五	6	星期六
		DWR	星期															
		0	星期日															
		1	星期一															
		2	星期二															
		3	星期三															
		4	星期四															
		5	星期五															
6	星期六																	

RTC 时间闹钟寄存器 (RTC_TAR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_TAR	RTC_BA+0x01C	R/W	时间闹钟寄存器	0x0000_0000

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

Bits	描述	
[31:22]	-	保留
[21:20]	10HR	10 Hour Time Digit of Alarm Setting (0~2)
[19:16]	1HR	1 Hour Time Digit of Alarm Setting (0~9)
[14:12]	10MIN	10 Min Time Digit of Alarm Setting (0~5)
[11:8]	1MIN	1 Min Time Digit of Alarm Setting (0~9)
[6:4]	10SEC	10 Sec Time Digit of Alarm Setting (0~5)
[3:0]	1SEC	1 Sec Time Digit of Alarm Setting (0~9)

注:

TAR 为 BCD 码格式的计数器, RTC 不会对载入值检测

括号内列出的为可接受的值

在 RTC 访问通过 AER 被使能后, 该寄存器的值可以被读回

RTC 日历闹钟寄存器 (RTC_CAR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_CAR	RTC_BA+0x20	R/W	日历闹钟寄存器	0x0000_0000

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

Bits	描述	
[31:24]	-	保留
[23:20]	10YEAR	10 Year Calendar Digit of Alarm Setting (0~9)
[19:16]	1YEAR	1 Year Calendar Digit of Alarm Setting (0~9)
[12]	10MON	10 Month Calendar Digit of Alarm Setting (0~1)
[11:8]	1MON	1 Month Calendar Digit of Alarm Setting (0~9)
[5:4]	10DAY	10 Day Calendar Digit of Alarm Setting (0~3)
[3:0]	1DAY	1 Day Calendar Digit of Alarm Setting (0~9)

注:

CAR 为 BCD 码格式的计数器, RTC 不会对载入值检测.

括号内列出的为可接受的值.

在 RTC 访问通过 AER 被使能后, 该寄存器的值可以被读回

RTC 闰年指示寄存器 (RTC LIR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_LIR	RTC_BA+0x24	R	RTC 闰年指示寄存器.	0x0000_0000

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

Bits	描述	
[31:1]	-	保留
[0]	LIR	闰年指示寄存器 (只读) 1 = 该年为闰年 0 = 该年非闰年

RTC 中断使能寄存器 (RTC_RIER)

寄存器	偏移量	R/W	描述	复位后的值
RTC_RIER	RTC_BA+0x28	R/W	RTC 中断使能寄存器	0x0000_0000

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

Bits	描述	
[31:3]	-	保留
[2]	SNOOPIER	监听引脚事件检测中断使能 1 = 监听引脚事件检测中断被使能 0 = 监听引脚事件检测中断被禁止.
[1]	TIER	时间节拍中断和时间节拍唤醒功能使能 1 = RTC 时间节拍中断被使能 0 = RTC 时间节拍中断被禁止.
[0]	AIER	闹钟中断使能 1 = RTC 闹钟中断被使能 0 = RTC 闹钟中断被禁止

RTC 中断指示寄存器 (RTC_IIR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_RIIR	RTC_BA+0x2C	R/W	RTC 中断指示寄存器	0x0000_0000

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

Bits	描述	
[31:3]	-	保留
[2]	SNOOPIS	<p>监听引脚事件检测中断状态</p> <p>当 SNOOPEN 为 1，并且由 SNOOPEDGE 定义的事件在监听引脚上被检测到时，该标志将会被置位，当该位被置位时，如果 SNOOPIER 为 1，RTC 将会向 CPU 产生一个中断</p> <p>向该位写 1 清除该标志</p> <p>1 = 在监听引脚上检测到了由 SNOOPEDGE 定义的事件。</p> <p>0 = 在监听引脚上从来没有检测到过由 SNOOPEDGE 定义的事件。</p>
[1]	TIS	<p>RTC 时间节拍中断状态</p> <p>RTC 会根据通过 TTR[2:0] 选定的时间节拍周期性的设置 TIS 标志为 1。当该标志被置位时，如果 TIER 为 1，RTC 将会向 CPU 产生一个中断</p> <p>向该位写 1 清除该标志</p> <p>1 = 表示 RTC 时间节拍中断条件已经发生</p> <p>0 = 表示 RTC 时间节拍中断条件从来没有发生过。</p>
[0]	AIS	<p>RTC 闹钟中断状态</p> <p>一旦 RTC 实时计数器 TLR 和 CLR 分别达到 闹钟设定时间寄存器 TAR 和 CAR，AIS 标志将被置位。当该标志被置位时，如果 AIER 为 1，RTC 将会向 CPU 产生一个中断。</p> <p>向该位写 1 清除该标志</p> <p>1 = 表示 RTC 闹钟中断条件已经发生</p> <p>0 = 表示 RTC 闹钟中断条件从来没有发生过。</p>

RTC 时间节拍寄存器 (RTC_TTR)

寄存器	偏移量	R/W	描述	复位后的值
RTC_TTR	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

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

Bits	描述																			
[31:1]	-	保留																		
[3]	TWKE	<p>RTC 定时器唤醒功能使能</p> <p>如果芯片在进入掉电模式之前，TWKE 被置位，当 RTC 时钟节拍或闹钟匹配发生时，芯片会被 RTC 唤醒。</p> <p>1 = 使能 RTC 定时器通过时钟节拍或闹钟匹配将芯片从掉电模式唤醒的功能</p> <p>0 = 禁用 RTC 定时器唤醒功能</p> <p>注：节拍定时器根据 TTR[2:0] 的描述设定。</p>																		
[2:0]	TTR[2:0]	<p>时钟节拍寄存器</p> <p>设定用于产生周期时间节拍中断的时间节拍周期。</p> <table><tr><th>TTR[2:0]</th><th>时间节拍 (秒)</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>1/2</td></tr><tr><td>2</td><td>1/4</td></tr><tr><td>3</td><td>1/8</td></tr><tr><td>4</td><td>1/16</td></tr><tr><td>5</td><td>1/32</td></tr><tr><td>6</td><td>1/64</td></tr><tr><td>7</td><td>1/128</td></tr></table> <p>注: 在 RTC 访问通过 AER 被使能后，该寄存器的值可以被读回</p>	TTR[2:0]	时间节拍 (秒)	0	1	1	1/2	2	1/4	3	1/8	4	1/16	5	1/32	6	1/64	7	1/128
TTR[2:0]	时间节拍 (秒)																			
0	1																			
1	1/2																			
2	1/4																			
3	1/8																			
4	1/16																			
5	1/32																			
6	1/64																			
7	1/128																			

RTC 备用功能控制寄存器 (RTC_SPRCTL)

寄存器	偏移量	R/W	描述	复位后的值
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC 备用功能控制寄存器	0x0000_0080

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

Bits	描述	
[31:8]	-	保留
[7]	SPRRDY	<p>SPR 寄存器准备好</p> <p>该位表示寄存器 SPR0 ~ SPR19 是否已经准备好被读</p> <p>在 CPU 写寄存器 SPR0 ~ SPR19 后, 需要轮询该位以确认 SP0 ~ SPR19 是否已经更新完成.</p> <p>该位是只读位, 任何对该位的写入不会产生任何作用</p> <p>1 = SPR0 ~ SPR19 已经更新完成, 准备好被读.</p> <p>0 = SPR0 ~ SPR19 正在更新.</p>
[6:2]	-	保留
[1]	SNOOPEDGE	<p>监听有效边沿选择</p> <p>该位定义在监听引脚上何种边沿信号会产生监听引脚被检测到的事件, 用于清除 20 个备用寄存器。</p> <p>1 = 在监听引脚上的下降沿将产生监听引脚检测到的事件</p> <p>0 = 在监听引脚上的上升沿将产生监听引脚检测到的事件.</p>
[0]	SNOOPEN	<p>监听引脚事件检测使能</p> <p>该位使能监听引脚事件检测</p> <p>当该位被置位, 并且在监听引脚上检测到了一个由 SNOOPEDGE 定义的事件, 20 个备用寄存器将被硬件自动清零, 而且, SNOOPIS 也将置位。此外, RTC 将会产生唤醒事件来唤醒系统。</p> <p>1 = 使能监听引脚事件检测功能</p> <p>0 = 禁止监听引脚事件检测功能.</p>

RTC 备用寄存器 X (RTC_SPRx)

寄存器	偏移量	R/W	描述	复位后的值
RTC_SPR0	RTC_BA+0x40	R/W	RTC 备用寄存器 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC 备用寄存器 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC 备用寄存器 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC 备用寄存器 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC 备用寄存器 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC 备用寄存器 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC 备用寄存器 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC 备用寄存器 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC 备用寄存器 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC 备用寄存器 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC 备用寄存器 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC 备用寄存器 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC 备用寄存器 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC 备用寄存器 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC 备用寄存器 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC 备用寄存器 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC 备用寄存器 16	0x0000_0000
RTC_SPR16	RTC_BA+0x84	R/W	RTC 备用寄存器 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC 备用寄存器 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC 备用寄存器 19	0x0000_0000

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

SPARE

Bits	描述	
[31:0]	SPARE[31:0]	<p>SPARE</p> <p>该位域用于存储由软件定义的备份信息</p> <p>一旦监听引脚事件被检测到，该位域将由硬件自动清零</p>

5.16 智能卡主机接口 (SC)

5.16.1 概述

智能卡接口控制器 (SC controller) 是基于 ISO/IEC 7816-3 标准并完全兼容 PC/SC 规格。它也提供卡插入/移除的状态。

5.16.2 特性

- ISO-7816-3 T = 0, T = 1 兼容.
- EMV2000 兼容
- 支持 3 个 ISO-7816-3 端口
- 用于数据负载的独立的接收/发送 4 字节入口缓存
- 可编程的发送时钟频率
- 可编程的接收器缓存触发水平.
- 可编程的保护时间选择 (11 ETU ~ 266 ETU).
- 一个 24-位和两个 8-位计数器用于请求应答 (Answer to Reset (ATR)) 和等待时间处理
- 支持自动反向约定功能
- 支持停止时钟电平和时钟停止 (时钟保持) 功能
- 支持传送器和接收器错误重试和错误数目限制功能
- 支持硬件激活序列处理
- 支持硬件暖复位序列处理
- 支持硬件释放序列处理.
- 支持当检测到卡移除时, 硬件自动释放序列
- 支持 UART 模式
 - ◆ 半双工, 异步通信.
 - ◆ 用于数据负载的独立的接收/发送 4 字节入口缓存
 - ◆ 在每个通道上支持可编程的波特率发生器
 - ◆ 支持可编程的接收器缓存触发水平
 - ◆ 从最后一个停止位被从 TX-FIFO 发出到释放的发送数据延迟时间通过设定 SCx_EGTR[EGT] 可编程
 - ◆ 可编程的偶, 奇或者无校验位生成和检测.
 - ◆ 可编程的停止位, 1 或 2 停止位生成

5.16.3 框图

SC 时钟控制和框图如下所示。SC 控制器是和时钟域，PCLK 和引擎时钟完全异步的设计，特别注意 PCLK 必须高于或等于引擎时钟的频率。

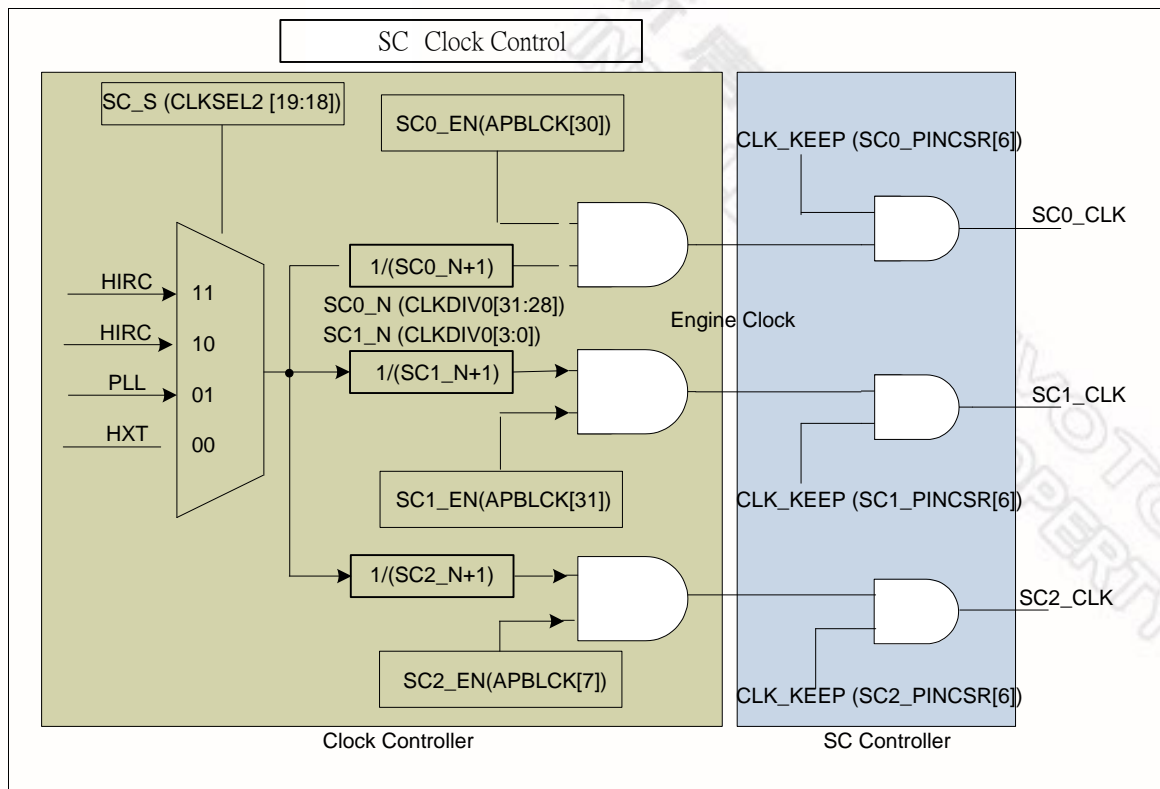


图 5-83 SC 时钟控制框图（时钟控制器中的 4-bit 预分频计数器）

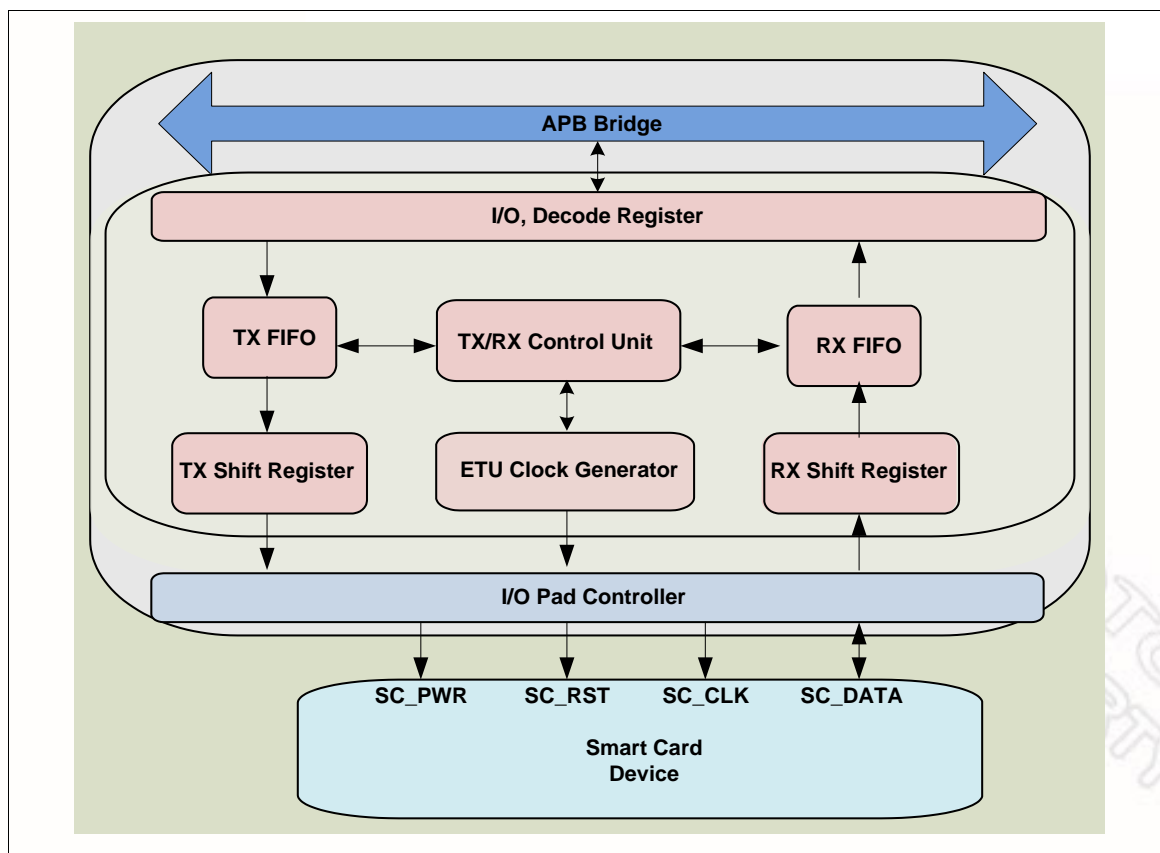


图 5-84 SC 控制器框图

5.16.4 功能描述

基本上，智能卡接口扮演的是一个半双工异步通信端口的角色，它的数据格式如下图所示的10个连续位组成。

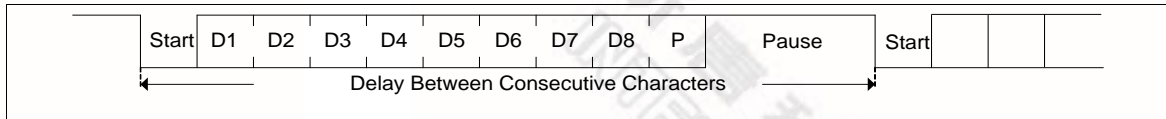


图 5-85 SC 数据字符

5.16.4.1 激活，暖复位和释放序列

激活

智能卡接口控制器支持硬件激活，暖复位和释放序列。激活序列如下：

- 设置 SC_RST 为低
- 设置 SC_PWR 在高电平，SC_DATA 在接收模式.
- 使能 SC_CLK 时钟
- 释放 SC_RST 到高电平

激活序列能够被软件或硬件控制。如果软件想控制，软件能够控制 SC_PINCSR 和 SC_TMRx 寄存器去处理激活序列或者设置 SC_ALTCTL [ACT_EN] 寄存器，然后接口将执行硬件激活序列

如下是硬件激活模式下的激活控制序列：

- 通过设置 SC_ALTCTL [INIT_SEL] 设置激活时序
- 当 SC_CTL [TMR_SEL] 为 01, 10 或 11 时，TMR0 可以被选择
- 设置操作模式 SC_TMR0 [MODE] 为 0011，通过设置 SC_TMR0 [CNT] 寄存器设定请求应答的值
- 当硬件释放 SC_RST 到高，硬件将同时产生一个中断 INT_INIT 到 CPU (SC_IER[INIT_IE] = "1")
- 如果 TMR0 减小计数值到 "1"（从 SC_RST 开始），而且在此之前卡没有回应 ATR，硬件将产生 INT_TMR0 中断到 CPU

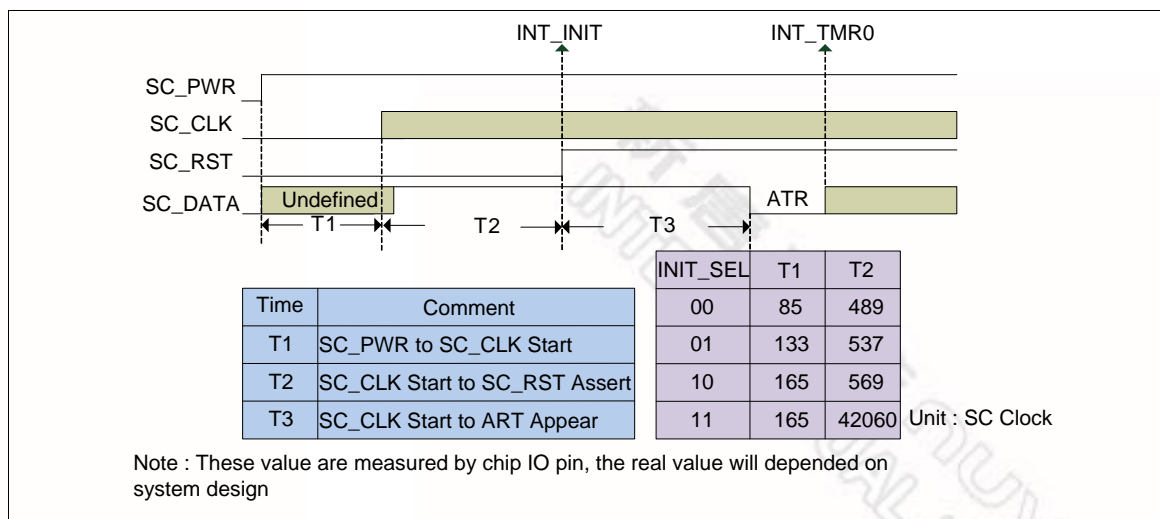


图 5-86 SC 激活序列

暖复位

暖复位序列如下：

暖复位序列能够被软件和硬件控制。如果软件想控制该顺序，软件能够控制 SC_PINCSR 和 SC_TMRx 寄存器去处理暖复位序列或者设置 SC_ALTCTL [WARST_EN] 寄存器，然后接口将执行硬件暖复位序列。

如下是硬件暖复位模式下的暖复位控制序列。

- 通过设置 SC_ALTCTL [INIT_SEL] 设定暖复位时序
- 通过设置 SC_CTL [TMR_SEL] 寄存器（TMR_SEL 可以是 01, 10, 或者 11)选择 TMR0
- 设定操作模式 SC_TMR0 [MODE] 为 011，通过设置 SC_TMR0 [CNT] 寄存器设定请求应答的值。
- 通过设置 SC_ALTCTL 寄存器设置 TMR0_SEN 和 WARST_EN 开始计数
- 当硬件释放 SM_RST 到高，硬件将同时产生INT_INIT 中断到 CPU (SC_IER[INIT_IE] = “1”)
- 如果 TMR0 减小计数器到 “1”（从 SC_RST开始），而且在此之前卡没有回应 ATR，硬件将产生 INT_TMR0 中断到 CPU。

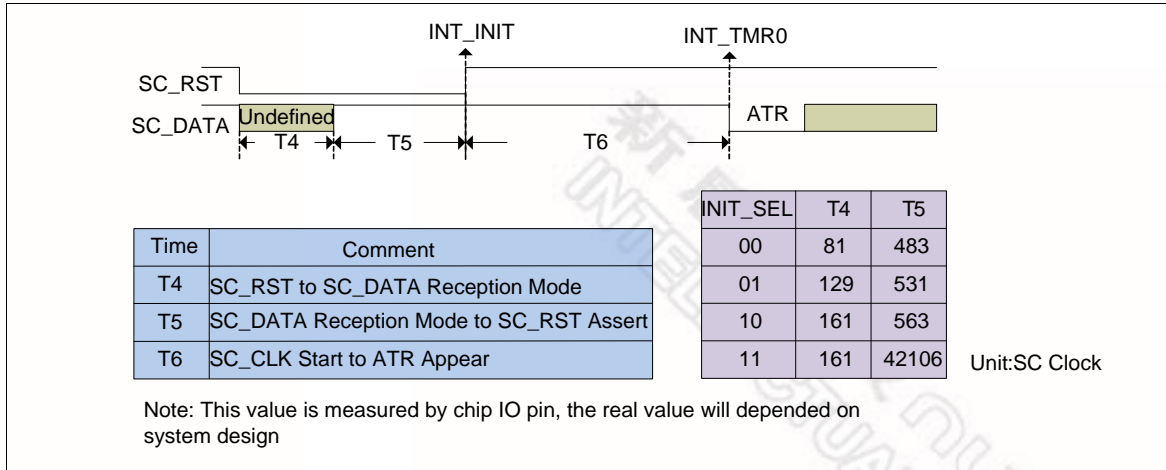


图 5-87 SC 暖复位序列

释放

释放序列如下：

- 设置 SC_RST 到低
- 停止 SC_CLK
- 设置 SC_DATA 到状态“低”
- 释放 SC_PWR

释放序列可以被软件或者硬件控制。如果软件想控制的话，软件能够控制 SC_PINCSR 和 SC_TMR0 寄存器去处理释放序列或者设置 SC_ALTCTL [DACT_EN] 寄存器，然后接口将执行硬件释放序列

当设置了卡移除检测(SC_PINCSR [ADAC_CDEN])时，SC 控制器也支持自动释放序列

如下是硬件释放模式下的释放控制序列：

通过设置 SC_ALTCTL [INIT_SEL] 设置释放时序

通过设置 SC_ALTCTL 寄存器设置 DACT_EN 以开始计数

当硬件释放 SC_PWR 为低，控制器将同时产生一个 INT_INIT 中断到 CPU (SC_IER[INIT_IE] = “1”)

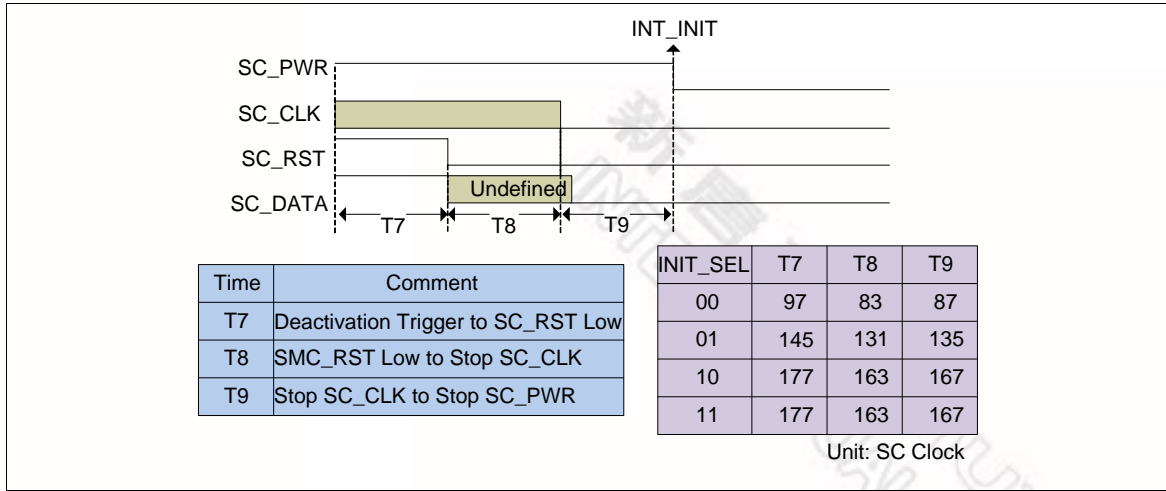


图 5-88 SC 释放序列

5.16.4.2 初始化字符 TS

依据 7816-3，初始化 ATR 的字符 TS 有如下两个可能的模式（如下图所示）。如果 TS 模式是 0_1100_0000_1，则是反向约定。当按反向约定进行解码后，则传送的字节等于 '3F'。如果 TS 模式是 0_1101_1100_1，则是直接约定。当按直接约定解码后，则传送的字节等于 '3B'。软件可以设置 SC_CTL [AUTO_CON_EN]，由硬件来决定操作的约定。软件也可以设置 SC_CTL [CON_SEL] 寄存器（设为 00 或 11），在 SC 收到 ATR 的 TS 后去改变操作约定。

如果软件通过设置 SC_CTL [AUTO_CON_EN] 寄存器使能自动约定功能，则设置步骤必须在 ATR 状态之前完成，而且第一个数据必须是 0x3B 或者 0x3F。在硬件收到第一个数据并存放缓存中之后，硬件将决定约定模式并自动改变 SC_CTL [CON_SEL] 寄存器。如果第一个数据既不是 0x3B 也不是 0x3F，则硬件将产生一个 INT_ACON_ERR 中断给 CPU（如果 SC_IER [ACON_ERR_IE] = "1"）。

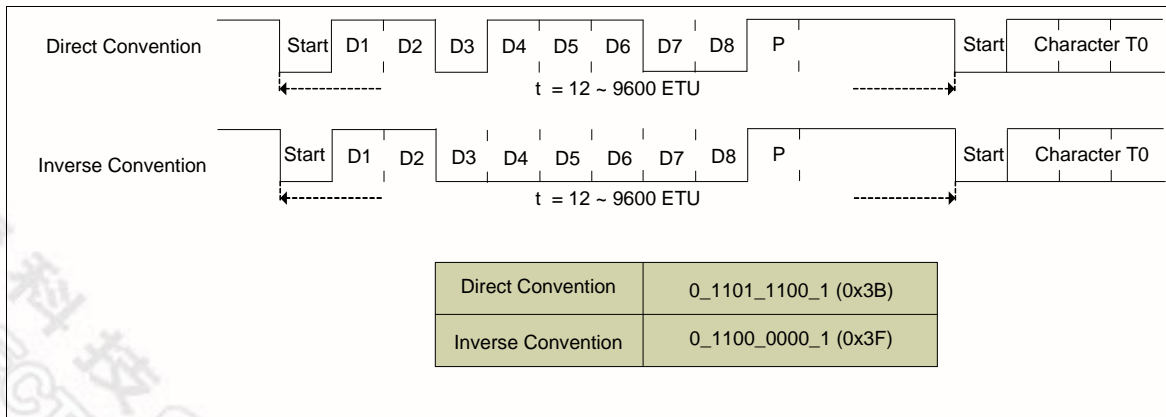


图 5-89 初始化字符 TS

5.16.4.3 错误信号和字符重复

依据 7816-3 T=0 模式描述，如下所示，如果接收器接收到一个错误的校验位，接收器将会拉低 SC_DATA 1 到 2 个位周期去通知发送器校验错误。然后发送器将重传该字符。SC 接口控制器支持接收器硬件错误检测功能和发送器硬件重传功能。软件能通过设置 SC_CTL [TX_ERETRY_EN] 来使能重传功能。软件也能够通过 SC_CTL [TX_ERETRY] 寄存器中定义重传的次数限制。如果重传的次数大于 SC_CTL [TX_ERETRY]，发送器将传送下一笔新的数据给设备，并产生一个

INT_TERR 中断给 CPU（如果此时 SC_IER [TERR_IE] = “1”）。如果接收器接收到错误的次数大于 SC_CTL [RX_ERETRY]，接收器将接收该错误数据到缓存，并产生一个 INT_TERR 中断给 CPU（如果 SC_IER [TERR_IE] = “1”）。

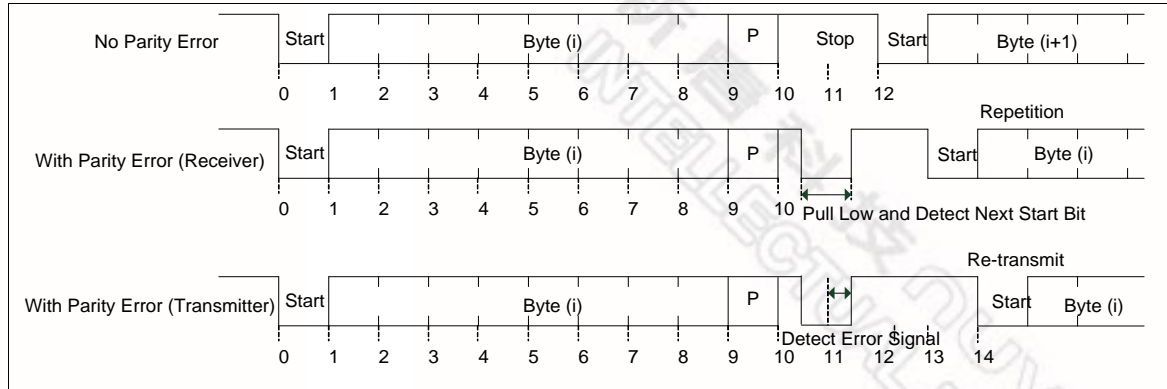


图 5-90 SC 错误信号

5.16.4.4 内部超时计数器

智能卡接口包括一个24-位超时计数器和两个 8-位 超时计数器。这些计数器帮助控制器处理不同的实时间隔 (ATR, WWT, BWT, 等待)。每个计数器可以被设置成一旦触发使能位被写或者检测到一个START 位就开始进行计数。

如下是编程流程：

- 通过设置 SC_CTL[TMR_SEL] 使能/禁用 计数器
- 通过设置SC_TMRx 寄存器选择操作模式 ([MODE]) 并设置一个计数值 ([CNT])
- 设置 TMRx_SEN 开始进行计数。

5.16.4.5 UART 模式

当 SCx_UACTL [UA_MODE_EN] 被置位，智能卡接口控制器也可以作为一个基本的 UART 功能来使用，此时不具有自动流控制。在UART模式下，智能接口数据(SCx_DATA)管脚将会作为UART RXD，而智能接口时钟(SCx_CLK)管脚将会作为UART TXD。如下是 UART 模式下的编程示例

编程示例

1. 软件可以通过设定 SCx_UACTL[UA_MODE_EN] 位来进入 UART 模式。
2. 通过设定 SCx_ALTCTL[RX_RST] 和 SCx_ALTCTL[TX_RST] 位进行软件复位，以确保所有的状态机处于休眠状态。
3. 填充 “0” 到 SCx_CTL [CON_SEL] 和 SCx_CTL [AUTO_CON_EN] 域。(当工作在 UART 模式时，这些位必须为 “0”)
4. 通过设定 SCx_ETUCR [ETU_RDIV] 来选择 UART 波特率。
 - 波特率 = $f / (ETU_RDIV + 1)$ ，此时f为智能卡控制器接口运行时钟频率(SCx_CLK)，有效的ETU_RDIV的有效值介于0x04与0xFFF之间，当其值小于0x04将会被视为0x04。
 - 例如当用户欲设定波特率115200，智能卡时钟频率为12MHz，ETU_RDIV应设为0x67其错误率大约为0.16%。
5. 选择数据格式，包括数据长度 (通过设定 SCx_UA_CTL [DATA_LEN])，校验位格式 (通过设定 SCx_UA_CTL [OPB] 和 SCx_UA_CTL [PBDIS] 位) 和停止位长度 (通过设定 SCx_CTL [SLEN])

or SCx_EGTR [EGT]).

6. 通过设定 SCx_CTL [RX_FTRI_LEV] 域来选择接收器缓存触发水平，并通过设定 SCx_RFTMR [RFTMR] 域来选择接收器缓存超时间隔。
7. 写 SCx_THR (TX) 寄存器或者读 SC_RBR (RX) 寄存器可以执行 UART 功能。

5.16.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
SC 基地址: SC0_BA = 0x4019_0000 SC1_BA = 0x401B_0000 SC2_BA = 0x401C_0000				
SCx_RBR	SCx_BA + 0x00	R	SC 接收缓存寄存器 (只读)	未定义
SCx_THR	SCx_BA + 0x00	W	SC 发送保持寄存器	未定义
SCx_CTL	SCx_BA + 0x04	R/W	SC 控制寄存器	0x0000_0000
SCx_ALTCTL	SCx_BA + 0x08	R/W	SC 交替控制寄存器	0x0000_0000
SCx_EGTR	SCx_BA + 0x0C	R/W	SC 扩展保护时间寄存器	0x0000_0000
SCx_RFTMR	SCx_BA + 0x10	R/W	SC 接收缓存超时寄存器	0x0000_0000
SCx_ETUCR	SCx_BA + 0x14	R/W	SC ETU 控制寄存器	0x0000_0173
SCx_IER	SCx_BA + 0x18	R/W	SC 中断使能寄存器	0x0000_0000
SCx_ISR	SCx_BA + 0x1C	R/W	SC 中断状态寄存器	0x0000_0002
SCx_TRSR	SCx_BA + 0x20	R/W	SC 传输状态寄存器	0x0000_0202
SCx_PINCSR	SCx_BA + 0x24	R/W	SC 管脚控制状态寄存器	0x0000_00x0
SCx_TMR0	SCx_BA + 0x28	R/W	SC 内部时间控制寄存器0	0x0000_0000
SCx_TMR1	SCx_BA + 0x2C	R/W	SC 内部时间控制寄存器1	0x0000_0000
SCx_TMR2	SCx_BA + 0x30	R/W	SC 内部时间控制寄存器2	0x0000_0000
SCx_UACTL	SCx_BA + 0x34	R/W	SC UART 模式控制寄存器.	0x0000_0000
SCx_TDRA	SCx_BA + 0x38	R	SC 定时器当前数据寄存器A	0x0000_07FF
SCx_TDRB	SCx_BA + 0x3C	R	SC 定时器当前数据寄存器B	0x0000_7F7F

注: SCx_REG 中的 x 代表 SC 是 通道.

5.16.6 寄存器描述

SC 接收缓存寄存器 (SCx_RBR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_RBR	SCx_BA + 0x00	R	SC 接收缓存寄存器.	未定义

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

Bits	描述	
[31:8]	-	保留.
[7:0]	RBR[7:0]	接收缓存寄存器 通过读该寄存器, SC 将返回一个8-位 的接收到的数据

SC 发送保持寄存器 (SCx THR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_THR	SCx_BA + 0x00	W	SC 发送保持寄存器.	未定义

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

Bits	描述	
[31:8]	-	保留.
[7:0]	THR[7:0]	<p>发送保持寄存器</p> <p>通过写该寄存器, SC 将发送出一个 8-位 数据。</p> <p>注: 如果 SC_CTL [SC_CEN] 未使能, 该寄存器将不能被编程。</p>

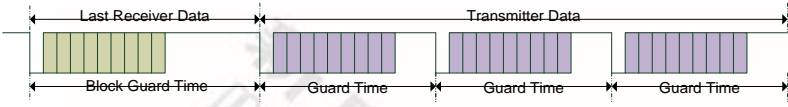
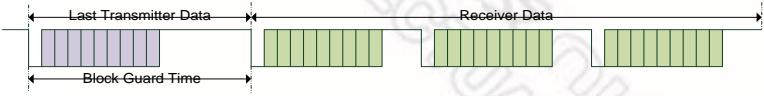
SC 控制寄存器 (SCx CTL)

寄存器	偏移量	R/W	描述	复位后的值
SCx_CTL	SCx_BA + 0x04	R/W	SC 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-						CD_DEB_SEL	
23	22	21	20	19	18	17	16
TX_ERETRY_EN	TX_ERETRY			RX_ERETRY_EN	RX_ERETRY		
15	14	13	12	11	10	9	8
SLEN	TMR_SEL		BGT				
7	6	5	4	3	2	1	0
RX_FTRI_LEV		CON_SEL		AUTO_CON_EN	DIS_TX	DIS_RX	SC_CEN

Bits	描述											
[30:26]	-	保留。										
[25:24]	CD_DEB_SEL [1:0]	卡检测去抖选择寄存器 该域指示卡检测去抖选择。										
		<table><tr><th>CD_DEB_SEL[1:0]</th><th>描述</th></tr><tr><td>00</td><td>每 384 (128 * 3) 个引擎时钟去抖采样卡插入一次，每 128 个引擎时钟去抖采样卡移除一次。</td></tr><tr><td>01</td><td>每 192 (64 * 3) 个引擎时钟去抖采样卡插入一次，每64 个引擎时钟去抖采样卡移除一次。</td></tr><tr><td>10</td><td>每 96 (32 * 3) 个引擎时钟去抖采样卡插入一次，每32 个引擎时钟去抖采样卡移除一次。</td></tr><tr><td>11</td><td>每 48 (16 * 3) 个引擎时钟去抖采样卡插入一次，每16 个引擎时钟去抖采样卡移除一次。</td></tr></table>	CD_DEB_SEL[1:0]	描述	00	每 384 (128 * 3) 个引擎时钟去抖采样卡插入一次，每 128 个引擎时钟去抖采样卡移除一次。	01	每 192 (64 * 3) 个引擎时钟去抖采样卡插入一次，每64 个引擎时钟去抖采样卡移除一次。	10	每 96 (32 * 3) 个引擎时钟去抖采样卡插入一次，每32 个引擎时钟去抖采样卡移除一次。	11	每 48 (16 * 3) 个引擎时钟去抖采样卡插入一次，每16 个引擎时钟去抖采样卡移除一次。
		CD_DEB_SEL[1:0]	描述									
		00	每 384 (128 * 3) 个引擎时钟去抖采样卡插入一次，每 128 个引擎时钟去抖采样卡移除一次。									
		01	每 192 (64 * 3) 个引擎时钟去抖采样卡插入一次，每64 个引擎时钟去抖采样卡移除一次。									
10	每 96 (32 * 3) 个引擎时钟去抖采样卡插入一次，每32 个引擎时钟去抖采样卡移除一次。											
11	每 48 (16 * 3) 个引擎时钟去抖采样卡插入一次，每16 个引擎时钟去抖采样卡移除一次。											
[23]	TX_ERETRY_EN	TX 错误重传使能寄存器 当校验错误发生时，该位使能发送器重传功能。 1 = 使能 TX 错误重传功能 0 = 禁用 TX 错误重传功能 注：用户必须在使能该位前填充 TX_ERETRY 的值。										
[22:20]	TX_ERETRY [2:0]	TX 错误重传寄存器 该域表明当校验错误发送后，发送器允许尝试的最多重传次数。 注1：实际的重传次数是 TX_ERETRY + 1，所以8次是最多的重传次数。										

Bits	描述		
		注2：当 TX_ERETRY_EN 使能时，该域不能被修改。修改该域的流程是先禁用 TX_ETRTY_EN，然后填充新的重传值。	
[19]	RX_ERETRY_EN	RX 错误重接收使能寄存器 当校验错误发生时，该位使能接收器重接收功能。 1 = 使能 RX 错误重接收功能 0 = 禁用 RX 错误重接收功能 注：在使能该位之前，用户必须填充 RX_ERETRY。	
[18:16]	RX_ERETRY [2:0]	RX 错误重接收寄存器 该域表示当校验错误发生时，接收器允许重接收的最多次数。 注1：实际的重接收次数是 RX_ERETRY + 1，所以 8 次是最多的重接收次数。 注2：当 RX_ERETRY_EN 使能时，该域不能被修改。修改该域的流程是先禁用 RX_ETRTY_EN，然后填充新的重接收值。	
[15]	SLEN	停止位长度 该域表示停止位的长度。 1 = 停止位长度是 1 ETU。 0 = 停止位的长度是 2 ETU。 注：默认停止位的长度是2.	
[14:13]	TMR_SEL[1:0]	定时器选择	
		TMR_SEL[1:0]	描述
		00	禁用所以内部定时器功能
		01	使能内部 24-位定时器。软件能够通过设置 SC_TMR0 [23:0] 配置定时器。SC_TMR1 和 SC_TMR2 在该模式下将被忽略。
		10	使能内部 24-位定时器和 8-位内部定时器。软件可以通过设置SC_TMR0 [23:0] 来配置24 位定时器，设置 SC_TMR1 [7:0] 来配置8 位定时器。SC_TMR2 该模式下将被忽略。
		11	使能内部 24-位定时器和两个8-位定时器。软件可以通过设置SC_TMR0 [23:0], SC_TMR1 [7:0]和 SC_TMR2 [7:0] 配置定时器。
[12:8]	BGT [4:0]	块保护时间 (BGT) 该域表示用于块保护的时间的计数器。依据 ISO7816-3, T=0 的模式下，软件必须填充该域为 15（实际的块保护时间 = 16），T=1 的模式，软件必须填充 该域为 21（实际的块保护时间 = 22）。 在 TX 模式，硬件将自动阻挡住第一个字符直到 BGT 的时间走完，而不管 TX 数据是什么。	

Bits	描述										
	 <p>Note1 : Hardware will control the transmit block guard time by SC_CTL [BGT] register setting. Note2 : Hardware will control the transmit guard time by SC_EGTR [EGT] register setting.</p> <p>RX 模式下，软件可以启用 SC_ALTCTL [RX_BGT_EN] 来检测第一个过来的字符时序。如果来的数据时间小于 BGT，将产生中断。</p>  <p>Note : If the incoming data timing less than SC_CTL [BGT], an interrupt will be generated (SC_ALTCTL [RX_BGT_EN] enable)</p> <p>注：实际的块保护时间是 BGT + 1.</p>										
[7:6]	<p>RX_FTRI_LEV [1:0]</p> <p>RX 缓存触发水平</p> <p>当接收缓存的字节数目等于 RX_FTRI_LEV，RDA_IF 将被置位（如果使能了 IER [RDA_IEN]，将产生中断）。</p> <table border="1"> <thead> <tr> <th>RX_FTRI_LEV</th><th>INTR_RDA 触发水平（字节）</th></tr> </thead> <tbody> <tr> <td>00</td><td>01</td></tr> <tr> <td>01</td><td>02</td></tr> <tr> <td>10</td><td>03</td></tr> <tr> <td>11</td><td>保留</td></tr> </tbody> </table>	RX_FTRI_LEV	INTR_RDA 触发水平（字节）	00	01	01	02	10	03	11	保留
RX_FTRI_LEV	INTR_RDA 触发水平（字节）										
00	01										
01	02										
10	03										
11	保留										
[5:4]	<p>CON_SEL [1:0]</p> <p>约定选择</p> <table border="1"> <thead> <tr> <th>CON_SEL</th><th>约定</th></tr> </thead> <tbody> <tr> <td>00</td><td>直接约定</td></tr> <tr> <td>01</td><td>保留</td></tr> <tr> <td>10</td><td>保留</td></tr> <tr> <td>11</td><td>反向约定</td></tr> </tbody> </table> <p>注：如果 AUTO_CON_EN 使能，该域将被忽略。</p>	CON_SEL	约定	00	直接约定	01	保留	10	保留	11	反向约定
CON_SEL	约定										
00	直接约定										
01	保留										
10	保留										
11	反向约定										
[3]	<p>AUTO_CON_EN</p> <p>自动约定使能</p> <p>0 = 禁用自动约定。</p> <p>1 = 使能自动约定。当硬件在 ATR 状态收到 TS，且 TS 是直接约定，CON_SEL 将被自动设为 00；否则，如果 TS 是反向约定，CON_SEL 将被设为 11。</p> <p>如果软件使能自动约定功能，设置步骤必须在 ATR 状态之前完成，而且第一个数据必须是 0x3B 或者 0x3F。在硬件收到第一个数据并保存数据到缓存后，硬件将决定何种约定并自动改变 SC_CTL[CON_SEL] 寄存器的值。如果第一个数据既不是 0x3B 也不是 0x3F，则硬件将产生 INT_ACON_ERR 中断到 CPU（如果 SC_IER [ACON_ERR_IE] = “1”）。</p>										
[2]	<p>DIS_TX</p> <p>禁用 TX 转换</p>										

Bits	描述	
		1 = 发送器禁用 0 = 发送器使能
[1]	DIS_RX	禁用 RX 转换 1 = 接收器禁用 0 = 接收器使能
[0]	SC_CEN	SC 引擎使能 设置该位为“1”使能 SC 操作。如果该位被清除，SC 将强制所有转换到 IDLE 状态。

SC 交替控制寄存器 (SCx_ALTCTL)

寄存器	偏移量	R/W	描述	复位后的值
SCx_ALTCTL	SCx_BA + 0x08	R/W	SC 交替控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
TMR2_ATV	TMR1_ATV	TMR0_ATV	RX_BGT_EN	-		INIT_SEL	
7	6	5	4	3	2	1	0
TMR2_SEN	TMR1_SEN	TMR0_SEN	WARST_EN	ACT_EN	DACT_EN	RX_RST	TX_RST

Bits	描述	
[31:16]	-	保留
[15]	TMR2_ATV	内部定时器 2 激活状态（只读） 该位指示定时器 2 的定时器计数器状态 1 = Timer2 激活 0 = Timer2 未激活
[14]	TMR1_ATV	内部定时器 1 激活状态（只读） 该位指示定时器 1 的定时器计数器状态 1 = Timer1 激活 0 = Timer1 未激活
[13]	TMR0_ATV	内部定时器 0 激活状态（只读） 该位指示定时器 0 的定时器计数器状态 1 = Timer0 激活 0 = Timer0 未激活
[12]	RX_BGT_EN	接收器块保护时间功能使能 1 = 使能接收器块保护时间功能 0 = 禁用接收器块保护时间功能
[11:10]	-	保留
[9:8]	INIT_SEL [1:0]	初始时序选择 该域表示硬件初始状态的时序（激活或 暖复位 或是 释放）。 单位：SC 时钟 激活：涉及激活数字

Bits	描述																							
		<table><tr><th>INIT_SEL</th><th>T1</th><th>T2</th></tr><tr><td>00</td><td>85</td><td>489</td></tr><tr><td>01</td><td>133</td><td>537</td></tr><tr><td>10</td><td>165</td><td>569</td></tr><tr><td>11</td><td>165</td><td>42060</td></tr></table>	INIT_SEL	T1	T2	00	85	489	01	133	537	10	165	569	11	165	42060							
INIT_SEL	T1	T2																						
00	85	489																						
01	133	537																						
10	165	569																						
11	165	42060																						
		暖复位：涉及暖复位数字																						
		<table><tr><th>INIT_SEL</th><th>T4</th><th>T5</th></tr><tr><td>00</td><td>81</td><td>483</td></tr><tr><td>01</td><td>129</td><td>531</td></tr><tr><td>10</td><td>161</td><td>563</td></tr><tr><td>11</td><td>161</td><td>42106</td></tr></table>	INIT_SEL	T4	T5	00	81	483	01	129	531	10	161	563	11	161	42106							
INIT_SEL	T4	T5																						
00	81	483																						
01	129	531																						
10	161	563																						
11	161	42106																						
		释放：涉及释放数字																						
		<table><tr><th>INTI_SEL</th><th>T7</th><th>T8</th><th>T9</th></tr><tr><td>00</td><td>97</td><td>83</td><td>87</td></tr><tr><td>01</td><td>145</td><td>131</td><td>135</td></tr><tr><td>10</td><td>177</td><td>163</td><td>167</td></tr><tr><td>11</td><td>177</td><td>163</td><td>167</td></tr></table>	INTI_SEL	T7	T8	T9	00	97	83	87	01	145	131	135	10	177	163	167	11	177	163	167		
INTI_SEL	T7	T8	T9																					
00	97	83	87																					
01	145	131	135																					
10	177	163	167																					
11	177	163	167																					
[7]	TMR2_SEN	<p>内部定时器 2 开始使能</p> <p>该位使能定时器 2 开始计数。软件能够填充“0”来停止计数器，设置“1”重载和计数。</p> <p>1 = 开始计数</p> <p>0 = 停止计数</p> <p>注1：当 SC_CTL [TMR_SEL] == 11 时，该域用于内部 8-位 定时器。当 SC_CTL [TMR_SEL] == 00 或 01 或 10 时，不要填充TMR2_SEN。</p> <p>注2：如果操作模式不是自动重载模式 (SC_TMR2 [26] = “0”)，则该位将被硬件自动清除。</p> <p>注3：该位会被 TX_RST 和 RX_RST 清除。所以不要同时设置该位，TX_RST 和 RX_RST。</p> <p>注4：如果 SC_CTL [SC_CEN] 未使能，不能对该位进行编程。</p>																						
[6]	TMR1_SEN	<p>内部定时器 1 开始使能</p> <p>该位使能定时器 1 开始计数。软件能够填充“0”来停止计数器，设置“1”重载和计数。</p> <p>1 = 开始计数</p> <p>0 = 停止计数</p> <p>注1：当 SC_CTL [TMR_SEL] == 01 或10 时，该域用于内部 8-位 定时器。当 SC_CTL [TMR_SEL] == 00 或 11时，不要填充TMR1_SEN。</p> <p>注2：如果操作模式不是自动重载模式 (SC_TMR2 [26] = “0”)，则该位将被硬件自动清除。</p>																						

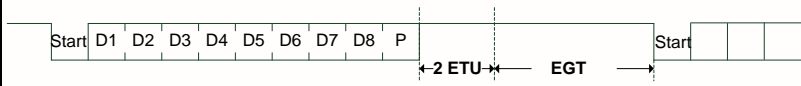
Bits	描述	
		<p>除。</p> <p>注3: 该位会被 TX_RST 和 RX_RST 清除。所以不要同时设置该位, TX_RST 和 RX_RST。</p> <p>注4: 如果 SC_CTL [SC_CEN] 未使能, 不能对该位进行编程。</p>
[5]	TMR0_SEN	<p>内部定时器 0 开始使能</p> <p>该位使能定时器 0 开始计数。软件能够填充“0”来停止计数器, 设置“1”重载和计数。</p> <p>1 = 开始计数</p> <p>0 = 停止计数</p> <p>注1: 当 SC_CTL [TMR_SEL] == 01 时, 该域用于内部 24-位 定时器。</p> <p>注2: 如果操作模式不是自动重载模式 (SC_TMR2 [26] = “0”), 则该位将被硬件自动清除。</p> <p>注3: 该位会被 TX_RST 和 RX_RST 清除。所以不要同时设置该位, TX_RST 和 RX_RST。</p> <p>注4: 如果 SC_CTL [SC_CEN] 未使能, 不能对该位进行编程。</p>
[4]	WARST_EN	<p>暖复位序列发生器使能</p> <p>该位使能 SC 控制器通过暖复位序列初始化卡</p> <p>1 = 使能暖复位序列发生器</p> <p>0 = 不起作用</p> <p>注1: 当暖复位序列完成后, 该位将被自动清除, SC_ISR [INIT_IS] 将被设置为“1”。</p> <p>注2: 该域将被 TX_RST 和 RX_RST 清除, 所以不要同时填充该位, TX_RST 和 RX_RST。</p> <p>注3: 如果 SC_CTL [SC_CEN] 未使能, 该域将不能被编程。</p>
[3]	ACT_EN	<p>激活序列发生器使能</p> <p>该位使能 SC 控制器通过激活序列初始化卡</p> <p>1 = 使能激活序列发生器</p> <p>0 = 不起作用</p> <p>注1: 当激活序列完成后, 该位将被自动清除, SC_ISR [INIT_IS] 将被设置为“1”。</p> <p>注2: 该域将被 TX_RST 和 RX_RST 清除, 所以不要同时填充该位, TX_RST 和 RX_RST。</p> <p>注3: 如果 SC_CTL [SC_CEN] 未使能, 该域将不能被编程。</p>
[2]	DACT_EN	<p>释放序列发生器使能</p> <p>该位使能 SC 控制器通过释放序列初始化卡</p> <p>1 = 使能释放序列发生器</p> <p>0 = 不起作用</p> <p>注1: 当释放序列完成后, 该位将被自动清除, SC_ISR [INIT_IS] 将被设置为“1”。</p> <p>注2: 该域将被 TX_RST 和 RX_RST 清除, 所以不要同时填充该位, TX_RST 和 RX_RST。</p> <p>注3: 如果 SC_CTL [SC_CEN] 未使能, 该域将不能被编程。</p>
[1]	RX_RST	RX 软件复位

Bits	描述	
		<p>当 RX_RST 被置位，接收缓存中的所有数据和 RX 内部状态机将被清除。</p> <p>1 = 复位 RX 内部状态机和指针。</p> <p>0 = 不起作用。</p> <p>注：该位将被自动清除，需要至少 3 个 SC 引擎时钟周期。</p>
[0]	TX_RST	<p>TX 软件复位</p> <p>当 TX_RST 被置位，发送缓存中的所有数据和 TX 内部状态机将被清除。</p> <p>1 = 复位 TX 内部状态机和指针。</p> <p>0 = 不起作用。</p> <p>注：该位将被自动清除，需要至少 3 个 SC 引擎时钟周期。</p>

SC 扩展保护时间寄存器 (SCx EGTR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_EGTR	SCx_BA + 0x0C	R/W	SC 扩展保护时间寄存器。	0x0000_0000

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

Bits	描述	
[31:8]	-	保留。
[7:0]	EGT[7:0]	<p>扩展保护时间</p> <p>该域表示扩展保护时间的值。</p>  <p>注: 该计数器是基于 ETU, 实际的扩展时间是 EGT。</p>

SC 接收器缓存超时寄存器 (SCx_RFTMR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_RFTMR	SCx_BA + 0x10	R/W	SC 接收器缓存超时寄存器	0x0000_0000

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

Bits	描述	
[31:9]	-	保留。
[8:0]	RFTM[8:0]	<p>SC 接收缓存超时寄存器 (基于 ETU)</p> <p>无论什么时候 RX 缓存收到一个新的数据字，超时计数器复位并开始计数。一旦计数器减小到 “1” 而且没有接收到新的数据或者 CPU 读 SC_RBR 寄存器没有可读数据，则将产生一个接收器超时中断 INT_RTMR（如果 SC_IER[RTMR_IE] 为高）。</p> <p>注1：计数器是基于 ETU，实际的计数值是 RFTM + 1</p> <p>注2：填充 “0” 到该域来禁用该功能。</p>

SC ETU 控制寄存器 (SCx ETUCR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_ETUCR	SCx_BA + 0x14	R/W	SC ETU 控制寄存器	0x0000_0173

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

Bits	描述	
[31:16]	-	保留。
[15]	COMPEN_EN	<p>补偿模式使能</p> <p>该位使能时钟补偿功能。当该位使能，硬件将在 n 时钟周期和 (n-1) 时钟周期之间交替其中 n 为写入到 ETU_RDIV 寄存器的值。</p> <p>1 = 使能补偿功能</p> <p>0 = 禁用补偿功能</p>
[14:12]	-	保留。
[11:0]	ETU_RDIV [11:0]	<p>ETU 速率除法器</p> <p>该域表示时钟速率除法器</p> <p>实际的 ETU 为 ETU_RDIV + 1.</p> <p>注1：软件可以配置该域，但是该域必须大于 0x04.</p> <p>注2：软件可以配置该域，但是如果错误速率等于2%，该域必须大于 0x040.</p>

SC 中断控制寄存器 (SCx_IER)

寄存器	偏移量	R/W	描述	复位后的值
SCx_IER	SCx_BA + 0x18	R/W	SC 中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-					ACON_ERR_I E	RTMR_IE	INIT_IE
7	6	5	4	3	2	1	0
CD_IE	BGT_IE	TMR2_IE	TMR1_IE	TMR0_IE	TERR_IE	TXBE_IE	RDA_IE

Bits	描述	
[31:11]	-	保留。
[10]	ACON_ERR_IE	自动约定错误中断使能 该域用于自动约定错误中断使能。 1 = 使能 INT_ACON_ERR. 0 = 禁用 INT_ACON_ERR.
[9]	RTMR_IE	接收器缓存超时中断使能 该域用于接收器缓存超时中断使能。 1 = 使能 INT_RTMR 0 = 禁用 INT_RTMR
[8]	INIT_IE	初始化结束中断使能 该域用于激活 (SC_ALTCTL [ACT_EN]), 释放 (SC_ALTCTL [DACT_EN]) 和暖复位 (SC_ALTCTL [WARST_EN]) 序列中断使能。 1 = 使能 INT_INIT 0 = 禁用 INT_INIT
[7]	CD_IE	卡检测中断使能 该域用于卡检测中断使能。卡检测状态寄存器是 SC_PINCSR [CD_CH] 和 SC_PINCSR[CD_CL]。 1 = 使能 INT_CD 0 = 禁用 INT_CD

Bits	描述	
[6]	BGT_IE	<p>块保护时间中断使能</p> <p>该域用于块保护时间中断使能。</p> <p>1 = 使能 INT_BGT</p> <p>0 = 禁用 INT_BGT</p>
[5]	TMR2_IE	<p>Timer2 中断使能</p> <p>该域用于 TMR2 中断使能。</p> <p>1 = 使能 INT_TMR2</p> <p>0 = 禁用 INT_TMR2</p>
[4]	TMR1_IE	<p>Timer1 中断使能</p> <p>该域用于 TMR1 中断使能</p> <p>1 = 使能 INT_TMR1</p> <p>0 = 禁用 INT_TMR1</p>
[3]	TMR0_IE	<p>Timer0 中断使能</p> <p>该域用于 TMR0 中断使能。</p> <p>1 = 使能 INT_TMR0</p> <p>0 = 禁用 INT_TMR0</p>
[2]	TERR_IE	<p>传输错误中断使能</p> <p>该域用于传输错误中断使能。传输错误状态在 SC_TRSR 寄存器，包括接收器断开错误(RX_EBR_F)，帧错误(RX_EFR_F)，校验错误(RX_EPA_F)，接收器缓存溢出错误(RX_OVER_F)，传送缓存溢出错误 (TX_OVER_F)，接收器重接收超过限制错误(RX_OVER_ERETRY) 和 传送器重传超过限制错误 (TX_OVER_ERETRY)。</p> <p>1 = 使能 INT_TERR</p> <p>0 = 禁用 INT_TERR</p>
[1]	TBE_IE	<p>传送缓存空中断使能</p> <p>该域用于传送缓存中断使能。</p> <p>1 = 使能 INT_THRE</p> <p>0 = 禁用 INT_THRE</p>
[0]	RDA_IE	<p>接收数据到达中断使能</p> <p>该域用于接收数据达到触发水平 (SC_CTL [RX_FTRI_LEV]) 中断使能。</p> <p>1 = 使能 INT_RDR</p> <p>0 = 禁用 INT_RDR</p>

SC 中断状态寄存器 (SCx_ISR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_ISR	SCx_BA + 0x1C	R/W	SC 中断状态寄存器 (只读)	0x0000_0002

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-					ACON_ERR_IS	RTMR_IS	INIT_IS
7	6	5	4	3	2	1	0
CD_IS	BGT_IS	TMR2_IS	TMR1_IS	TMR0_IS	TERR_IS	TBE_IS	RDA_IS

Bits	描述	
[31:11]	-	保留。
[10]	ACON_ERR_IS	自动约定错误中断状态标志 (只读) 该域指示自动约定序列错误。如果在 ATR 状态下收到的 TS 不是 0x3B 或者 0x3F, 该位将被置位。 注: 该位只读, 但是可以写“1”清除。
[9]	RTMR_IS	接收器缓存超时中断状态标志 (只读) 该域用于接收器缓存超时中断状态标志。 注: 该域是接收器缓存超时状态的状态标志。如果软件要清除该位, 软件必须通过读 SC_RBR 寄存器读取接收器缓存中剩下的数据。
[8]	INIT_IS	初始化结束中断状态标志 (只读) 该位用于激活 (SC_ALTCTL [ACT_EN]), 释放 (SC_ALTCTL [DACT_EN]) 和暖复位 (SC_ALTCTL [WARST_EN]) 顺序中断状态标志。 注: 该位只读, 但是可以写“1”清除。
[7]	CD_IS	卡检测中断状态标志 (只读) 该域用于卡检测中断状态标志。卡检测状态寄存器是 SC_PINCSR [CD_INS_F] 和 SC_PINCSR [CD_REM_F]。 注: 该域是 SC_PINCSR [CD_INS_F] 或 SC_PINCSR [CD_REM_F] 的状态标志。所以软件想要清除该位, 必须写“1”该位。
[6]	BGT_IS	块保护时间中断状态标志 (只读) 该域用于块保护时间中断状态标志。 注: 该位只读, 可以写“1”清除。

Bits	描述	
[5]	TMR2_IS	<p>Timer2 中断状态标志（只读）</p> <p>该域用于 TMR2 中断状态标志。</p> <p>注：该位只读，可以写“1”清除。</p>
[4]	TMR1_IS	<p>Timer1 中断状态标志（只读）</p> <p>该域用于 TMR1 中断状态标志。</p> <p>注：该位只读，可以写“1”清除。</p>
[3]	TMR0_IS	<p>Timer0 中断状态标志（只读）</p> <p>该域用于 TMR0 中断状态标志。</p> <p>注：该位只读，可以写“1”清除。</p>
[2]	TERR_IS	<p>传输错误中断状态标志（只读）</p> <p>该域用于传输错误中断状态标志。传输错误状态在 SC_TRSR 寄存器，包括接收器断开错误 (RX_EBR_F)，帧错误 (RX_EFR_F)，校验错误 (RX_EPA_F) 和接收器缓存溢出错误 (RX_OVER_F)，传输缓存溢出错误 (TX_OVER_F)，接收器重接收超过限制错误 (RX_OVER_ERETRY) 和发送器重传超过限制错误 (TX_OVER_ERETRY)。</p> <p>注：该域是 SC_TRSR [RX_EBR_F]，SC_TRSR [RX_EFR_F]，SC_TRSR [RX_EPA_F]，SC_TRSR [RX_OVER_F]，SC_TRSR [TX_OVER_F]，SC_TRSR [RX_OVER_ERETRY] 或 SC_TRSR [TX_OVER_ERETRY] 的状态标志。如果软件想清除该位，写“1”到该位清除。</p>
[1]	TBE_IS	<p>传输缓存空中断状态标志（只读）</p> <p>该域用于传输缓存空中断状态标志。该位不同于 SC_TRSR [TX_EMPTY_F] 标志和 SC_TRSR [TX_ATV] 标志。当最后一个字节数据被读到移位寄存器后，TX_EMPTY_F 会被置位；TX_ATV 标志指示传送器是否激活（最后的数据是否已经被发送），而当最后一个字节数据被读取到转移寄存器或最后一笔数据已经被传送了，TBE_IS 将被置位。当该位被置，软件可以写1~4 字节数据到 SC_THR 寄存器。</p> <p>注：如果软件想清除该位，软件必须写数据到 SC_THR 寄存器，然后该位将被自动清除。</p>
[0]	RDA_IS	<p>接收数据到达中断状态标志（只读）</p> <p>该域用于接收数据到达触发水平 (SC_CTL [RX_FTRI_LEV]) 中断状态标志。</p> <p>注：该域是接收数据达到SC_CTL [RX_FTRI_LEV] 的状态标志。如果软件从 SC_RBR 读取数据而且接收器的指针小于 SC_CTL [RX_FTRI_LEV]，该位自动清除。</p>

SC 传输状态寄存器 (SCx_TRSR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TRSR	SCx_BA + 0x20	R/W	SC传输状态寄存器 (只读)	0x0000_0202

31	30	29	28	27	26	25	24
TX_ATV	TX_OVER_E RETRY	TX_ERETRY_ F	-		TX_POINT_F		
23	22	21	20	19	18	17	16
RX_ATV	RX_OVER_E RETRY	RX_ERETRY_ F	-		RX_POINT_F		
15	14	13	12	11	10	9	8
-					TX_FULL_F	TX_EMPTY_F	TX_OVER_F
7	6	5	4	3	2	1	0
-	RX_EBR_F	RX_EFR_F	RX_EPA_F	-	RX_FULL_F	RX_EMPTY_ F	RX_OVER_F

Bits	描述	
[31]	TX_ATV	传输激活状态标志 (只读) 当 TX 传输处于激活状态或者最后一个字节传输还没有完成, 该位被硬件置位。 当 TX 传输完成而且 STOP 位 (包括保护时间) 已经被发送, 则该位自动清除。
[30]	TX_OVER_ERETRY	发送器重传超过限制次数错误 (只读) 当发送器重传超过重传次数的限制时, 该位被置位。 注: 该位只读, 但是能够写 “1” 清除该位。
[29]	TX_ERETRY_F	发送器重传错误 (只读) 当发送器重传时, 该位被置位。 注1: 该位只读, 可以写 “1” 清除。 注2: 该位只是一个标志位, 不能产生任何中断给 CPU。
[28:27]	-	保留。
[26:24]	TX_POINT_F [2:0]	发送缓存指针状态标志 (只读) 该域指示 TX 缓存指针状态标志。当 CPU 写数据到 SC_THR, TX_POINT_F 增加1。 当 TX 缓存的一个字节被传输到发送器移位寄存器时, TX_POINT_F 减少1。
[23]	RX_ATV	接收器激活状态标志 (只读) 当 RX 传输处于激活时, 该位被置位。 当 RX 传输完成时, 该位被自动清除。
[22]	RX_OVER_ERETRY	接收器重接收次数超过限制错误 (只读) 当 RX 传输错误重接收超过重接收次数限制时, 该位被置位。

Bits	描述	
		<p>注1: 该位只读, 但是可以写“1”清除。</p> <p>注2: 如果 CPU 通过设置SC_CTL [RX_ERETRY_EN] 寄存器使能接收器重接收功能, RX_EPA_F 标志将被忽略(硬件将不会设置 RX_EPA_F)。</p>
[21]	RX_ERETRY_F	<p>接收器重传错误(只读)</p> <p>当 RX 有任何错误并且重传时, 该位被置位。</p> <p>注1: 该位只读, 但是可以写“1”清除。</p> <p>注2: 该位是标志, 不能产生任何中断给 CPU。</p> <p>注3: 如果 CPU 通过设置SC_CTL [RX_ERETRY_EN] 寄存器使能接收器重试功能, RX_EPA_F 标志将被忽略(硬件将不能设置 RX_EPA_F)。</p>
[20:19]	-	保留。
[18:16]	RX_POINT_F [2:0]	<p>接收器缓存指针状态标志(只读)</p> <p>该域指示 RX 缓存指针状态标志。当 SC 从外部设备收到一个字节, RX_POINT_F 加1。当 RX 缓存被 CPU 读取了一个字节, RX_POINT_F 减1。</p>
[15:11]	-	保留。
[10]	TX_FULL_F	<p>发送缓存满状态标志(只读)</p> <p>该位指示 TX 缓存是否满了。</p> <p>当 TX 指针等于 4 时, 该位被置位。否则会被硬件清除。</p>
[9]	TX_EMPTY_F	<p>发送缓存空状态标志(只读)</p> <p>该位指示 TX 缓存是否为空。</p> <p>当 TX 缓存的最后一个字节已经被传送到发送器移位寄存器后, 硬件将置该位为高。当写数据到 SC_THR 时(TX 缓存不为空), 该位被清除。</p>
[8]	TX_OVER_F	<p>TX 溢出错误中断状态标志(只读)</p> <p>如果 TX 缓存为满(TX_FULL_F = “1”), 一个额外的写数据到 SC_THR 将导致该位被置为逻辑“1”。</p> <p>注1: 该位只读, 但是可以写“1”清除该位。</p> <p>注2: 额外的写数据被忽略。</p>
[7]	-	保留。
[6]	RX_EBR_F	<p>接收器断开错误状态标志(只读)</p> <p>每当接收到的输入数据(RX)保持在“spacing state”(逻辑“0”)状态的时间长过一个完整的字传输时间(也就是“start bit”+ data bits + parity + stop bits 的总时间), 该位被设置成逻辑“1”。</p> <p>注1: 该位只读, 可以写“1”清除。</p> <p>注2: 如果 CPU 通过设置寄存器 SC_CTL [RX_ERETRY_EN] 设置接收器重接收功能, 硬件将不会设置该标志位。</p>
[5]	RX_EFR_F	<p>接收器帧错误状态标志(只读)</p> <p>每当接收的字符没有一个有效的“stop bit”(也就是, 紧跟在最后一位数据位或校验位后的停止位被检测为逻辑“0”), 该位被置为逻辑“1”。</p> <p>注1: 该位只读, 不过可以写“1”清除。</p> <p>注2: 如果 CPU 通过设置 SC_CTL [RX_ERETRY_EN] 寄存器设置接收器重接收功</p>

Bits	描述	
		能，硬件将不会设置该标志位。
[4]	RX_EPA_F	<p>接收器校验错误状态标志（只读）</p> <p>每当收到的字符没有有效的“parity bit”，该位设置为逻辑“1”。</p> <p>注1：该位只读，可以写“1”清除。</p> <p>注2：如果 CPU 通过设置SC_CTL [RX_ERETRY_EN] 寄存器设置接收器重接收功能，硬件将不会设置该标志位。</p>
[3]	-	保留。
[2]	RX_FULL_F	<p>接收器缓存满状态标志（只读）</p> <p>该位指示 RX 缓存是否为满。</p> <p>当 RX 指针等于 4 时该位被置位，否则该位将被硬件清除。</p>
[1]	RX_EMPTY_F	<p>接收器缓存空状态标志（只读）</p> <p>该位指示 RX 缓存是否为空。</p> <p>当 RX 缓存的最后一个字节被 CPU 读取后，硬件将设置该位为高。当 SC 收到任何新的数据，该位将被清除。</p>
[0]	RX_OVER_F	<p>RX 溢出错误状态标志（只读）</p> <p>当 RX 缓存溢出时，该位被置位。</p> <p>如果收到的数据个数大于 RX 缓存 (SC_RBR) 的大小，4 字节，该位将被置位。</p> <p>注1：该位只读，可以写“1”清除。</p> <p>注2：覆盖的数据将被忽略。</p>

SC 管脚控制状态寄存器 (SCx PINCSR)

寄存器	偏移量	R/W	描述	复位后的值
SCx_PINCSR	SCx_BA + 0x24	R/W	SC管脚控制状态寄存器	0x0000_00x0

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							SC_DATA_I_ST
15	14	13	12	11	10	9	8
-					CD_LEV	SC_DATA_O	SC_OEN_ST
7	6	5	4	3	2	1	0
ADAC_CD_EN	CLK_KEEP	CLK_STOP_LEV	CD_PIN_ST	CD_INS_F	CD_REM_F	SC_RST	POW_EN

Bits	描述	
[31:17]	-	保留。
[16]	SC_DATA_I_ST	SC 数据输入管脚状态（只读） 该位是 SC_DATA_I 的管脚状态。 1 = SC_DATA_I 管脚为高。 0 = SC_DATA_I 管脚为低。
[15:11]	-	保留。
[10]	CD_LEV	卡检测电平 1 = 当硬件检测到卡检测管脚从低到高，该位指示检测到一张卡。 0 = 当硬件检测到卡检测管脚从高到低，该位指示检测到一张卡。  注：软件必须在智能卡引擎使能之前，选择卡检测的电平。
[9]	SC_DATA_O	SC 数据输出管脚 该位是 SC_DATA_O 的管脚状态，但是用户可以通过设置该位驱动 SC_DATA_O 管脚为高或低。 1 = 驱动 SC_DATA_O 管脚到高 0 = 驱动 SC_DATA_O 管脚到低 注：当 SC 处于激活，暖复位或者释放模式时，该位将被自动改变，所以 SC 在这些模

Bits	描述	
		式下，不要设置该位。
[8]	SC_OEN_ST	SC 数据输出使能管脚状态（只读） 1 = SC_DATA_OEN 管脚为高 0 = SC_DATA_OEN 管脚为低
[7]	ADAC_CD_EN	当卡移除时，自动释放 1 = 使能当硬件检测到卡移除时自动释放 0 = 禁用当硬件检测到卡移除时自动释放 注1：当卡被移除时，硬件将停止所有处理然后进行释放序列（如果该位置位的话）。当该过程完成后，硬件将产生一个 INT_INIT 给 CPU。
[6]	CLK_KEEP	SC 时钟使能 1 = SC 时钟总是保持自由运行 0 = 禁用 SC 时钟产生 注：当操作在激活，暖复位或者释放模式时，该位将会自动改变。所以当操作在这些模式时，不要填充该位。
[5]	CLK_STOP_LEVEL	SC 时钟停止电平 该域指示在时钟停止模式时的时钟极性控制。 1 = 使得 SC_CLK 停止在高电平 0 = 使得 SC_CLK 停止在低电平
[4]	CD_PIN_ST	SC_CD 管脚状态（只读） 该位为 SC_CD 的管脚状态。 1 = SC_CD 管脚状态为高 0 = SC_CD 管脚状态为低
[3]	CD_INS_F	SC_CD 管脚卡检测插入状态（只读） 每当卡插入时，该位被置位。 1 = 卡插入 0 = 不起作用 注1：该位只读，可以写“1”清除。 注2：卡检测引擎将在 SC_CTL [SC_CEN] 设置后开启。
[2]	CD_REM_F	SC_CD 管脚卡检测移除状态（只读） 每当卡移除后，该位被置位。 1 = 卡移除 0 = 不起作用 注1：该位只读，可以写“1”清除。 注2：卡检测引擎将在 SC_CTL [SC_CEN] 设置后开启。
[1]	SC_RST	SC_RST 管脚信号 该位是 SC_RST 的管脚状态，但是用户可以通过设置该位驱动 SC_RST 管脚为高或低。

Bits	描述	
		<p>1 = 驱动 SC_RST 管脚为高</p> <p>0 = 驱动 SC_RST 管脚为低</p> <p>注：当操作在激活，暖复位或释放模式时，该位会自动改变。所以当操作在这些模式时，不要填充该位。</p>
[0]	POW_EN	<p>SC_POW_EN 管脚信号</p> <p>该位是SC_POW_EN 的管脚状态，但是用户可以通过设置该位驱动 SC_POW_EN 管脚为高或低。</p> <p>1 = 驱动 SC_POW_EN 管脚为高</p> <p>0 = 驱动 SC_POW_EN 管脚为低</p> <p>注：当操作在激活，暖复位或释放模式时，该位会自动改变。所以当操作在这些模式时，不要填充该位。</p>

SC 定时器控制寄存器 0 (SC_TMR0)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TMR0	SCx_BA + 0x28	R/W	SC 内部定时器控制寄存器 0	0x0000_0000

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

Bits	描述			
[31:28]	-	保留。		
[27:24]	MODE[3:0]	Timer 0 操作模式选择 该位指示内部24-位 定时器的操作选择。		
		TMR0_SEL	操作描述	
		0000	向下计数器开始于SC_ALTCTL [TMR0_SEN] 使能，结束于计数器超时。超时值为 CNT+1。	
			开始	当 SC_ALTCTL [TMR0_SEN] 使能，开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR0_IS 并清除 SC_ALTCTL [TMR0_SEN]。
[23:0]	MODE[3:0]	0001	向下计数器开始于第一个 START 位被检测到，结束于计数器超时。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR0_SEN] 设为 “1”后，当检测到第一个 START 位（接收或发送），开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR0_IS 并清除 SC_ALTCTL [TMR0_SEN]。
		0010	向下计数器开始于第一个 START 位（接收）被检测到，结束于计数器超时出现。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR0_SEN] 设为 “1”后，当检测到第一个 START 位（接收），开始计数。
结束	当向下计数器等于 “0”，硬件将自动设置 TMR0_IS 并清除			

Bits	描述		
			SC_ALTCTL [TMR0_SEN]。
		0011	<p>向下计数器只用于硬件激活，暖复位序列来测量 ATR 时序。</p> <p>时间开始于SC_RST 释放，结束于 ATR 应答收到或超时。</p> <p>如果在 ATR 应答收到之前，计数器减到“0”，则硬件将产生一个中断给 CPU。超时值为 CNT+1。</p>
			<p>开始</p> <p>在SC_ALTCTL [TMR0_SEN] 设为“1”后，当SC_RST 释放，开始计数。</p> <p>用于硬件激活，暖复位模式。</p>
			<p>结束</p> <p>如果在 ATR 应答收到之前，计数器减到“0”，则硬件将自动设置 TMR0_IS 并清除 SC_ALTCTL [TMR0_SEN]。</p> <p>当 ATR 接收到而且向下计数器没有等于“0”，硬件将自动清除 SC_ALTCTL [TMR0_SEN]。</p>
		0100	<p>和模式 0000 一样，但是当向下计数器等于“0”时，硬件将设置 TMR0_IS 而且计数器将重载 SC_TMR0 [CNT] 的值，并重新计数直到软件清除 SC_ALTCTL [TMR0_SEN]。</p> <p>当 SC_ALTCTL [TMR0_ATV] = “1”，软件能在任何时候改变 SC_TMR0 [CNT] 的值。当向下计数器等于“0”，计数器将重载 SC_TMR0 [CNT] 的新值并重新计数。</p> <p>超时值为 CNT+1。</p>
		0101	<p>和模式 0001 一样，但是当向下计数器等于“0”时，硬件将设置 TMR0_IS 而且计数器将重载 SC_TMR0 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR0_SEN]。</p> <p>当 SC_ALTCTL [TMR0_ATV] = “1”，软件能在任何时候改变 SC_TMR0 [CNT] 的值。当向下计数器等于“0”，计数器将重载 SC_TMR0 [CNT] 的新值并重新计数。</p> <p>超时值为 CNT+1。</p>
	MODE[3:0]	0110	<p>和模式 0010 一样，但是当向下计数器等于“0”时，硬件将设置 TMR0_IS 而且计数器将重载 SC_TMR0 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR0_SEN]。</p> <p>当 SC_ALTCTL [TMR0_ATV] = “1”，软件能在任何时候改变 SC_TMR0 [CNT] 的值。当向下计数器等于“0”，计数器将重载 SC_TMR0 [CNT] 的新值并重新计数。</p> <p>超时值为 CNT+1。</p>
		0111	<p>向下计数器开始于第一个 START 位被检测到，结束于软件清除 SC_ALTCTL [TMR0_SEN] 位。如果下一个START 位被检测到，计数器将重载 SC_TMR0 [CNT] 的新值并重新计数。</p> <p>如果在 下一个 START 位被检测到之前，计数器减到“0”，则硬件将产生一个中断给CPU。超时值为 CNT+1。</p>
			<p>开始</p> <p>在SC_ALTCTL [TMR0_SEN] 设为“1”后，当检测到第一个 START 位，开始计数。</p>
			<p>结束</p> <p>SC_ALTCTL [TMR0_SEN] 被设为“0”后，停止计数。</p>

Bits	描述						
		1000	<p>向上计数器开始于SC_ALTCTL [TMR0_SEN] 使能，结束于 SC_ALTCTL [TMR0_SEN] 禁用。计数值将被保存在 SC_TDRA [23:0]。该模式下，硬件将不能产生任何中断给 CPU。实际计数值将是 SC_TDRA [23:0] +1。</p> <table><tr><td>开始</td><td>在SC_ALTCTL [TMR0_SEN] 设为 “1”后，开始计数，开始计数的值为“0”（硬件将忽略 CNT的值）。</td></tr><tr><td>结束</td><td>SC_ALTCTL [TMR0_SEN] 被设为 “0”后，停止计数，并保存值到 SC_TDRA [23:0] 寄存器。</td></tr></table>	开始	在SC_ALTCTL [TMR0_SEN] 设为 “1”后，开始计数，开始计数的值为“0”（硬件将忽略 CNT的值）。	结束	SC_ALTCTL [TMR0_SEN] 被设为 “0”后，停止计数，并保存值到 SC_TDRA [23:0] 寄存器。
开始	在SC_ALTCTL [TMR0_SEN] 设为 “1”后，开始计数，开始计数的值为“0”（硬件将忽略 CNT的值）。						
结束	SC_ALTCTL [TMR0_SEN] 被设为 “0”后，停止计数，并保存值到 SC_TDRA [23:0] 寄存器。						
[23:0]	CNT[23:0]	<p>Timer 0 计数器值寄存器（基于 ETU）</p> <p>该域指示内部定时器操作值。</p>					

SC 定时器控制寄存器 1 (SCx_TMR1)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TMR1	SCx_BA + 0x2C	R/W	SC 内部定时器控制寄存器 1	0x0000_0000

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

Bits	描述			
[31:28]	-	保留。		
[27:24]	MODE[3:0]	Timer 1 操作模式选择 该域指示内部 8 位定时器操作模式。		
		TMR1_SEL	操作描述	
		0000	向下计数器开始于SC_ALTCTL [TMR1_SEN] 使能，结束于计数器超时。超时值为 CNT+1.	
			开始	当 SC_ALTCTL [TMR1_SEN] 使能，开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR1_IS 并清除 SC_ALTCTL [TMR1_SEN]。
[27:24]	MODE[3:0]	0001	向下计数器开始于第一个 START 位被检测到，结束于计数器超时。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR1_SEN] 设为 “1”后，当检测到第一个 START 位（接收或发送），开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR1_IS 并清除 SC_ALTCTL [TMR1_SEN]。
		0010	向下计数器开始于第一个 START 位（接收）被检测到，结束于计数器超时出现。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR1_SEN] 设为 “1”后，当检测到第一个 START 位（接收），开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR1_IS 并清除 SC_ALTCTL [TMR1_SEN]。

Bits	描述		
		0100	和模式 0000 一样，但是当向下计数器等于“0”时，硬件将设置 TMR1_IS 而且计数器将重载 SC_TMR1 [CNT] 的值，并重新计数直到软件清除 SC_ALTCTL [TMR1_SEN]。 当 SC_ALTCTL [TMR1_ATV] = “1”，软件能在任何时候改变 SC_TMR1 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR1 [CNT] 的新值并重新计数。 超时值为 CNT+1.
		0101	和模式 0001 一样，但是当向下计数器等于“0”时，硬件将设置 TMR1_IS 而且计数器将重载 SC_TMR1 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR1_SEN] 。 当 SC_ALTCTL [TMR1_ATV] = “1”， 软件能在任何时候改变 SC_TMR1 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR1 [CNT] 的新值并重新计数。 超时值为 CNT+1.
[27:24]	MODE[3:0]	0110	和模式 0010 一样，但是当向下计数器等于“0”时，硬件将设置 TMR1_IS 而且计数器将重载 SC_TMR1 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR1_SEN]。 当 SC_ALTCTL [TMR1_ATV] = “1”， 软件能在任何时候改变 SC_TMR1 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR1 [CNT] 的新值并重新计数。 超时值为 CNT+1
		0111	向下计数器开始于第一个 START 位被检测到，结束于软件清除 SC_ALTCTL [TMR1_SEN] 位。如果下一个START 位被检测到，计数器将重载 SC_TMR1 [CNT] 的新值并重新计数。 如果在 下一个 START 位被检测到之前，计数器减到“0”，则硬件将产生一个中断给CPU。超时值为 CNT+1。
			开始 在SC_ALTCTL [TMR1_SEN] 设为 “1”后，当检测到第一个 START 位，开始计数。
			结束 SC_ALTCTL [TMR1_SEN] 被设为 “0”后，停止计数。
		1000	向上计数器开始于SC_ALTCTL [TMR1_SEN] 使能，结束于 SC_ALTCTL [TMR1_SEN] 禁用。计数值将被保存在 SC_TDRB [7:0]。该模式下，硬件将不能产生任何中断给 CPU。实际计数值将是 SC_TDRB [7:0] +1。
			开始 在SC_ALTCTL [TMR1_SEN] 设为 “1”后，开始计数，开始计数的值为“0”（硬件将忽略 CNT的值）。 结束 SC_ALTCTL [TMR1_SEN] 被设为 “0”后，停止计数，并保存值到 SC_ TDRB [7:0] 寄存器。
[23:8]	-	保留。	
[7:0]	CNT[7:0]	Timer 1 计数器值寄存器（基于 ETU） 该域指示内部定时器操作值	

SC 定时器控制寄存器 2 (SCx_TMR2)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TMR2	SCx_BA + 0x30	R/W	SC 内部定时器控制寄存器 2	0x0000_0000

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

Bits	描述			
[31:28]	-	保留。		
[27:24]	MODE[3:0]	Timer 2 操作模式选择 该域指示内部 8-位定时器操作选择。		
		TMR2_SEL	操作描述	
		0000	向下计数器开始于SC_ALTCTL [TMR2_SEN] 使能，结束于计数器超时。超时值为 CNT+1。	
			开始	当 SC_ALTCTL [TMR2_SEN] 使能，开始计数
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR2_IS 并清除 SC_ALTCTL [TMR2_SEN]。
[27:24]	MODE[3:0]	0001	向下计数器开始于第一个 START 位被检测到，结束于计数器超时。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR2_SEN] 设为 “1”后，当检测到第一个 START 位（接收或发送），开始计数。
			结束	当向下计数器等于 “0”，硬件将自动设置 TMR2_IS 并清除 SC_ALTCTL [TMR2_SEN]。
		0010	向下计数器开始于第一个 START 位（接收）被检测到，结束于计数器超时出现。超时值为 CNT+1。	
			开始	在SC_ALTCTL [TMR2_SEN] 设为 “1”后，当检测到第一个 START 位（接收），开始计数。
		结束	当向下计数器等于 “0”，硬件将自动设置 TMR2_IS 并清除 SC_ALTCTL [TMR2_SEN]。	

Bits	描述		
		0100	和模式 0000 一样，但是当向下计数器等于“0”时，硬件将设置 TMR2_IS 而且计数器将重载 SC_TMR2 [CNT] 的值，并重新计数直到软件清除 SC_ALTCTL [TMR2_SEN]。 当 SC_ALTCTL [TMR2_ATV] = “1”，软件能在任何时候改变 SC_TMR2 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR2 [CNT] 的新值并重新计数。 超时值为 CNT+1.
		0101	和模式 0001 一样，但是当向下计数器等于“0”时，硬件将设置 TMR2_IS 而且计数器将重载 SC_TMR2 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR2_SEN] 。 当 SC_ALTCTL [TMR2_ATV] = “1”， 软件能在任何时候改变 SC_TMR2 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR2 [CNT] 的新值并重新计数。 超时值为 CNT+1.
[27:24]	MODE[3:0]	0110	和模式 0010 一样，但是当向下计数器等于“0”时，硬件将设置 TMR2_IS 而且计数器将重载 SC_TMR2 [CNT] 的值。当检测到下一个 START 位，计数器将重新计数直到软件清除SC_ALTCTL [TMR2_SEN]。 当 SC_ALTCTL [TMR2_ATV] = “1”， 软件能在任何时候改变 SC_TMR2 [CNT] 的值。当向下计数器等于 “0”，计数器将重载 SC_TMR2 [CNT] 的新值并重新计数。 超时值为 CNT+1.
		0111	向下计数器开始于第一个 START 位被检测到，结束于软件清除 SC_ALTCTL [TMR2_SEN] 位。如果下一个START 位被检测到，计数器将重载 SC_TMR2 [CNT] 的新值并重新计数。 如果在 下一个 START 位被检测到之前，计数器减到“0”，则硬件将产生一个中断给CPU。超时值为 CNT+1。
			开始 在SC_ALTCTL [TMR2_SEN] 设为 “1”后，当检测到第一个 START 位，开始计数。
			结束 SC_ALTCTL [TMR2_SEN] 被设为 “0”后，停止计数。
		1000	向上计数器开始于SC_ALTCTL [TMR2_SEN] 使能，结束于 SC_ALTCTL [TMR2_SEN] 禁用。计数值将被保存在 SC_TDRB [15:8]。该模式下，硬件将不能产生任何中断给 CPU。实际计数值将是 SC_ TDRB [15:8] +1。
			开始 在SC_ALTCTL [TMR2_SEN] 设为 “1”后，开始计数，开始计数的值为“0”（硬件将忽略 CNT的值）。 结束 SC_ALTCTL [TMR2_SEN] 被设为 “0”后，停止计数，并保存值到 SC_ TDRB [15:8] 寄存器。
[23:8]	-	保留。	
[7:0]	CNT[7:0]	Timer 2 计数器值寄存器（基于 ETU） 该域指示内部定时器操作值	

SC UART 模式控制寄存器 (SCx_UACTL)

寄存器	偏移量	R/W	描述	复位后的值
SCx_UACTL	SCx_BA + 0x34	R/W	SC UART 控制寄存器	0x0000_0000

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

Bits	描述											
[31:8]	-	保留。										
[7]	OPE	奇校验使能 1 = 奇数个逻辑 1 被发送, 或者在接收模式下检查数据位和校验位 0 = 偶数个逻辑 1 被发送, 或者在接收模式下检查数据位和校验位 注: 该位仅在 PBDIS 位为 '0'时起作用。										
[6]	PBDIS	校验位禁止 1 = 校验位不产生 (发送数据) 或者在传输过程中不被检查 (接收数据). 0 = 校验位产生或者在串行数据的 “最后一个数据位” 和 “停止位” 之间被检查 注: 当工作在智能卡模式时, 该位必须被设置为 '0' (默认设置带校验位)										
[5:4]	DATA_LEN	数据长度 <table><tr><th>WLS[1:0]</th><th>字符 Length</th></tr><tr><td>00</td><td>8 位</td></tr><tr><td>01</td><td>7 位</td></tr><tr><td>10</td><td>6 位</td></tr><tr><td>11</td><td>5 位</td></tr></table> 注: 当工作在智能卡模式时, 该位域必须被设置为 '00'	WLS[1:0]	字符 Length	00	8 位	01	7 位	10	6 位	11	5 位
WLS[1:0]	字符 Length											
00	8 位											
01	7 位											
10	6 位											
11	5 位											
[3:1]	-	保留										
[0]	UA_MODE_EN	UART 模式使能 1 = UART 模式.										

Bits	描述	
		<p>0 = 智能卡模式.</p> <p>注 1: 当工作在 UART 模式, 用户必须设置 SCx_CTL [CON_SEL] 和 SCx_CTL [AUTO_CON_EN] 为 "0".</p> <p>注 2: 当工作在智能卡模式, 用户必须设置 SCx_UACTL [7:0] 为 "0".</p> <p>注 3: 当 UART 功能被使能, 硬件将产生一个复位信号来复位内部缓存和内部状态机。</p>

SC 定时器当前数据寄存器 A (SCx_TDRA)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TDRA	SCx_BA + 0x38	R	SC 定时器当前数据寄存器 A	0x0000_07FF

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
TDR0							
15	14	13	12	11	10	9	8
TDR0							
7	6	5	4	3	2	1	0
TDR0							

Bits	描述	
[31:24]	-	保留。
[23:0]	TDR0 [23:0]	Timer0 当前数据寄存器（只读） 该位指示 timer0 当前的计数值。

SC 定时器当前数据寄存器 B (SCx TDRB)

寄存器	偏移量	R/W	描述	复位后的值
SCx_TDRB	SCx_BA + 0x3C	R	SC 定时器当前数据寄存器 B	0x0000_7F7F

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
TDR2							
7	6	5	4	3	2	1	0
TDR1							

Bits	描述	
[31:16]	-	保留。
[15:8]	TDR2[7:0]	Timer2 当前数据寄存器（只读） 该位指示 timer2 当前的计数值。
[7:0]	TDR1[7:0]	Timer1 当前数据寄存器（只读） 该位指示 timer1 当前的计数值。

5.17 SPI

5.17.1 概述

串行外围设备接口 (SPI) 是一个同步串行数据通讯协议。设备可工作在主/从模式，利用 4 线双向接口进行相互通讯。当从一个外围设备接收数据时，SPI 执行串-并的转换，而在数据向外围设备发送时执行并-串的转换。SPI 控制器可以被配置为主设备或从设备。

SPI 控制器支持唤醒功能。当该芯片处于掉电模式时，可以被片外设备唤醒。

该控制器支持可变串行时钟以适应特殊的应用，也支持 2 位传输模式，可同时连接两个片外从机设备。SPI 控制器也支持使用 PDMA 功能访问数据缓冲区。

5.17.2 特性

- 支持主机 (最大 32 MHz) 或从机 (最大 16 MHz) 模式操作
- 支持 1 位或 2 位传输模式
- 支持双 IO 传输模式
- 一个事务传输的数据长度可配置为 8 到 32 位
- 支持 MSB 或 LSB 优先传输
- 主机模式下支持 2 条从机选择线
- 可配置的字节或字休眠模式
- 支持字节重排序
- 主机模式下，支持可变串行时钟频率
- 提供独立的 8 级深度发送和接收 FIFO 缓存。
- 支持唤醒功能
- 支持 PDMA 传输
- 支持三线，没有从机选择信号的双向接口

5.17.3 框图

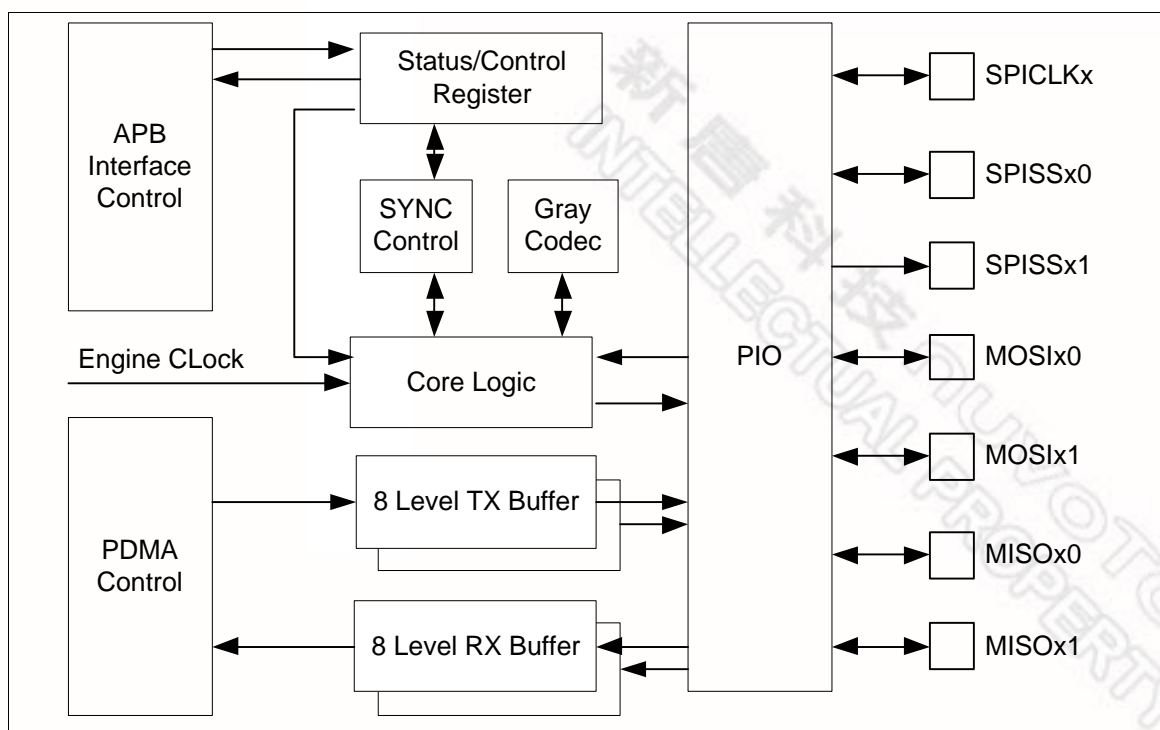


图 5-91 SPI 框图

“x”表示SPI 模块号。例如：SPICLK0 是第一个SPI 块串行时钟端口。

5.17.4 功能描述

5.17.4.1 SPI 引擎时钟和串行时钟

SPI 控制器需要 SPI 引擎时钟来驱动 SPI 逻辑单元执行数据传输。SPI 引擎时钟速率通过设置时钟源和时钟分频器来设定。CLKSEL2 寄存器的 SPIx_S 位决定 SPI 引擎时钟的时钟源。时钟源可以是 HCLK 或者 PLL 输出时钟。SPI_DIVIDER 寄存器中 DIVIDER 的设定值决定时钟速率计算时的分频值。

在主机模式下，如果可变时钟功能禁止，串行时钟输出引脚的输出频率等于 SPI 引擎时钟速率。在从机模式下，SPI 串行时钟由片外主机设备提供。从机设备的 SPI 引擎时钟速率必须快于连接在一起的主机设备的串行时钟速率。不论工作在主机还是从机模式，SPI 引擎时钟频率不能快于 APB 时钟速率。

5.17.4.2 主机/从机模式

SPI 控制器可通过设置 SLAVE 位 (SPI_CTL[18])，配置为主机或从机模式，来与片外从机或主机通信。主机模式与从机模式的应用框图如下：

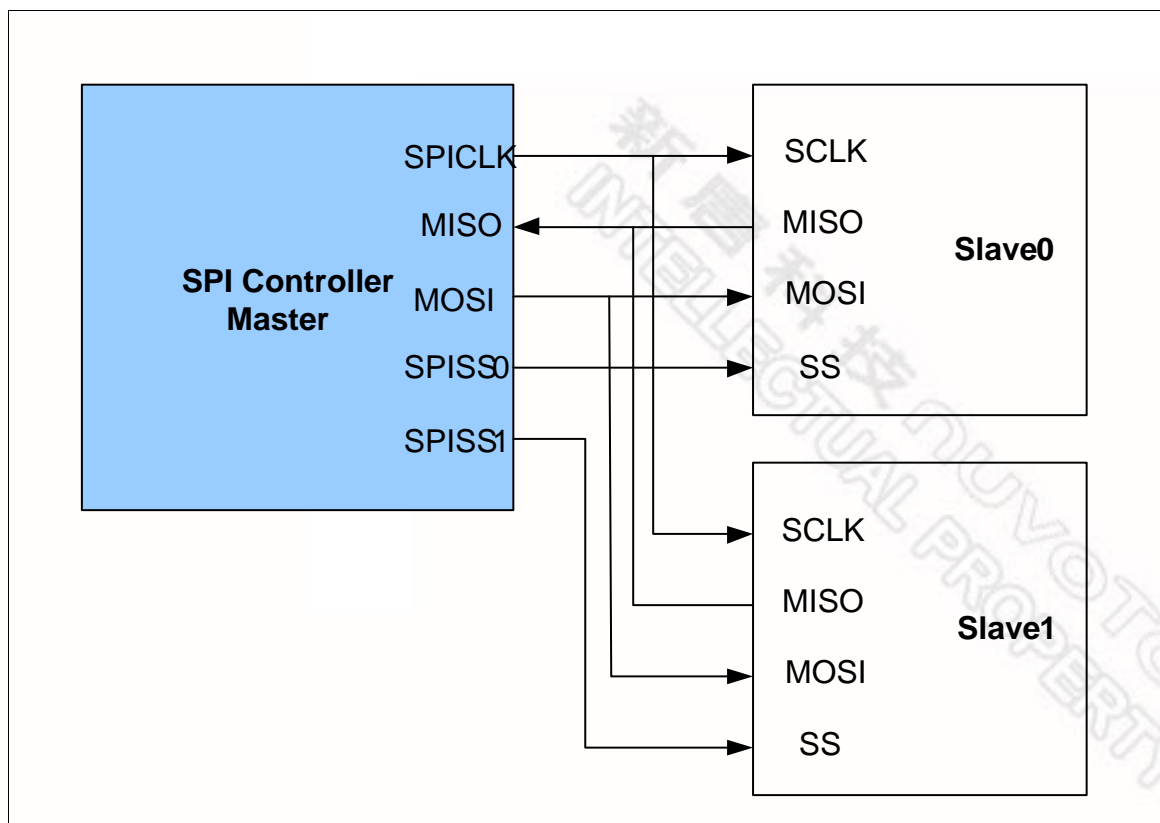


图 5-92 SPI 主机模式应用框图

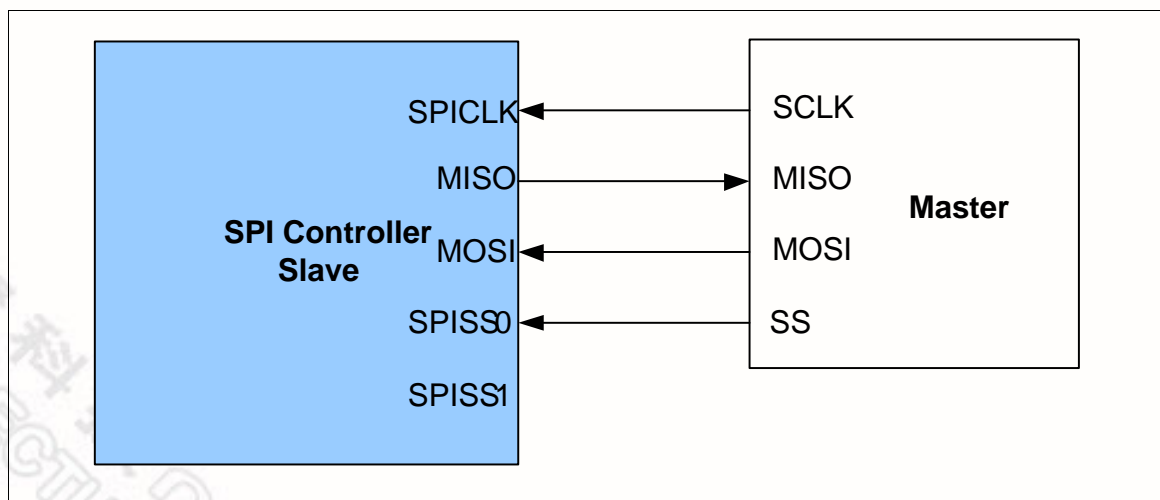


图 5-93 SPI 从机模式应用框图

5.17.4.3 从机选择

在主机模式下，该 SPI 控制器能通过从机选择输出脚 SPIxSS0 与 SPIxSS1 来驱动多达两个片外从机设备，但是这种驱动两个片外从机设备的操作是一种分时操作，不能同时与两个外围从设备进行通讯。

在从机模式下，片外主机设备通过 **SPIxSS0** 输入端口驱动从机选择信号到 **SPI** 控制器。在主机/从机模式下，从机选择信号的有效电平可以通过编程 **SS_LVL** 位 (**SPI_SSR[2]**) 来设定为低有效或高有效，从机选择信号 **SPIxSS0/1** 为电平触发还是边沿触发由 **SS_LTRIG** 位 (**SPI_SSR[4]**) 定义。触发条件的选择取决于所连接的从机/主机设备的类型。

从机模式下，如果 **SS_LTRIG** 位被配置成电平触发，则 **LTRIG_FLAG** 位 (**SPI_STATUS[4]**) 用来表示在一个 **SPI** 中，接收到的数据位数是否满足 **TX_BIT_LEN** 的设定值

5.17.4.4 电平触发/边沿触发

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，数据传输从一个有效的边沿开始，到一个无效的边沿结束。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，如下两个条件会终结传输过程，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数与 **TX_BIT_LEN** 的设定值匹配，从机的单元传输中断标志将会被置位。第二个条件，如果主机在传输过程正在进行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** 状态位来检查数据是否已被全部传输。

5.17.4.5 自动从机选择

在主机模式下，如果置位 **AUTOSS** (**SPI_SSR[3]**)，将自动产生从机选择信号，并根据 **SSR[0]** 位 (**SPI_SSR[0]**) 与 **SSR[1]** 位 (**SPI_SSR[1]**) 是否使能，将从机选择信号输出到 **SPIxSS0** 与 **SPIxSS1** 管脚上。这意味着当通过设置 **GO_BUSY** 位 (**SPI_CTL[0]**) 来开始数据传输时，在 **SPI_SSR[1:0]** 中使能的从机选择信号将由 **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为无效状态。当 **AUTOSS** 位被清零时，从机选择输出信号需要通过手工置位/清除 **SPI_SSR[1:0]** 相关位，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** 位 (**SPI_SSR[2]**) 定义。

在主机模式下，如果 **SP_CYCLE[3:0]** 的值小于 3 且 **AUTOSS** 被置位，在两个连续的事务之间，从机选择线好将保持有效。

在从机模式下，为了识别从机选择信号的无效状态，在两个连续的事务之间，从机选择信号无效状态持续时间必须大于或等于 6 个引擎时钟周期。

5.17.4.6 无从机选择模式 (3 线模式)

在从机模式下，当软件设置 **NOSLVSEL** 位来使能 3 线模式时，**SPI** 控制器可以在无从机选择信号的情况下工作。**NOSLVSEL** 位仅在从机模式下起作用。该模式仅需要 3 个管脚，**SPICLK**，**SPI_MISO**，和 **SPI_MOSI**，与 **SPI** 主机通信。**SPISS** 管脚可以被配置为 **GPIO**。当 **NOSLVSEL** 位被设置为 1，**SPI** 将在 **GO_BUSY** 位被设置为 1 之后准备好发送/接收数据。在 3 线模式下，**SS_LTRIG**，**SPI_SSR[4]**，必须被设置为 1。

正常情况下，当发送开始后，中断标志 **SLV_START_INTSTS** 将会被置位；而且，当接收到的数据满足由 **TX_BIT_LEN** 定义的位数时，也会有一个中断事件发生。在无从机选择模式下，当接收到的数据位数少于由 **TX_BIT_LEN** 定义的位数，并且在由用户定义的时间周期内，也再没有串行时钟输入的情况下，用户可以通过设置 **SLV_ABORT** 位来强制结束当前传输，然后用户可以获取到一个传输结束中断事件。

5.17.4.7 可变时钟功能

在主机模式下，如果 **VARCLK_EN**(**SPI_CTL[23]**) 被设置为 1，串行时钟输出可以被编程为可变频率模式。每个串行时钟周期的长度取决于 **SPI_VARCLK** 寄存器的设定值。当可变时钟功能使能。**TX_BIT_LEN** 必须被设定为 0x10 来配置数据传输为 16 位传输模式。**VARCLK[31]** 决定第一个时

钟周期的长度。如果 VARCLK[31] 为 0，第一个时钟周期的长度取决于 DIVIDER1 的设定值，如果为 1，第一个时钟周期的长度取决于 DIVIDER2 的设定值。在 VARCLK[30:1] 中两个连续的位定义一个时钟周期。位域 VARCLK[30:29] 定义一个事务中 SPI 串行时钟的第二个周期，位域 VARCLK[28:27] 定义第三个周期，等等。VARCLK[0] 无意义。串行时钟 (SPICLK)，VARCLK，DIVIDER1 和 DIVIDER2 寄存器之间的时序关系如下图所示。

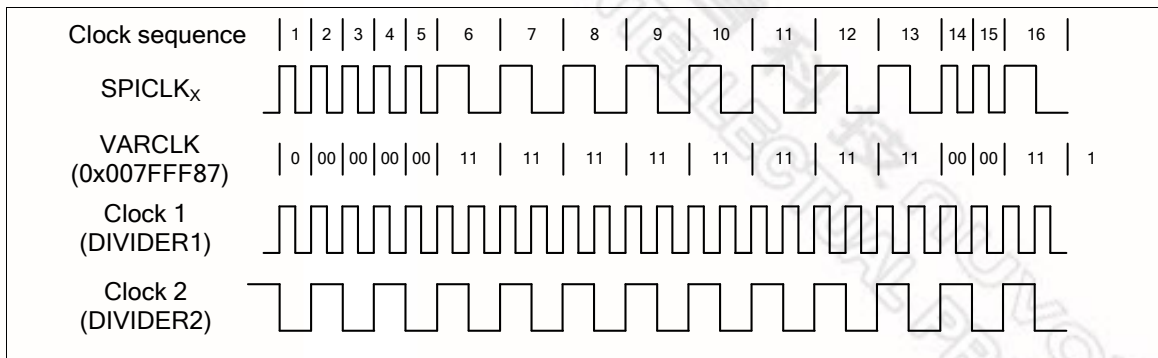


图 5-94 SPI 可变时钟频率

5.17.4.8 时钟极性

在主机模式下，CLKP 位 (SPI_CTL[11]) 定义串行时钟的空闲状态。如果 CLKP = 1，输出 SPICLK 为空闲高电平状态，否则如果 CLKP = 0，输出 SPICLK 为空闲低电平状态。对于可调串行时钟，仅工作于 CLKP = 0 的情况下。

5.17.4.9 发送/接收位长

一个事务的位长在 Tx_BIT_LEN 位域 (SPI_CTL[7:3]) 中定义。对于发送和接收，在一个事务中，位长可被配置为多达 32 位。

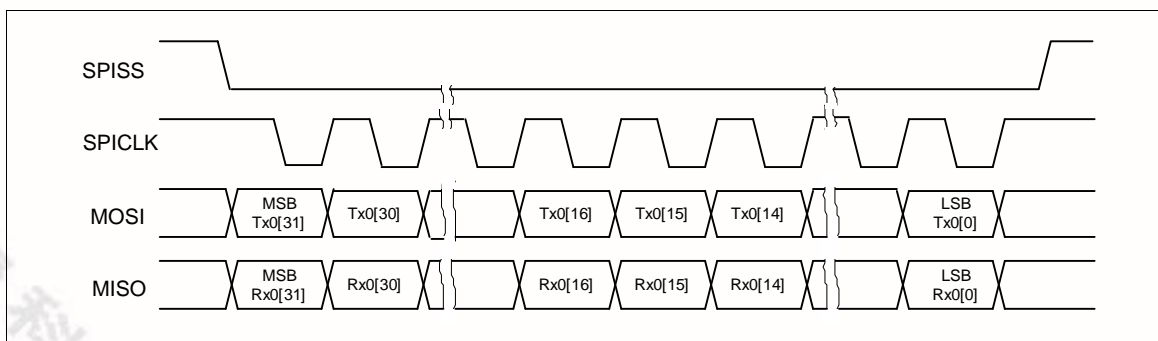


图 5-95 SPI 一个事务中数据位长为 32 位

5.17.4.10 LSB 优先

LSB (SPI_CTL[10]) 位定义一个事务中位传输序列。如果设定 LSB 位为 1，传输序列为 LSB 优先。位 0 首先被传输。如果清除 LSB 位为 0，传输序列是 MSB 优先 (参考上图)。

5.17.4.11 发送边沿

TX_NEG (SPI_CTL[2]) 位定义数据在串行时钟 SPICLK 的下降沿还是上升沿被发送出去。

5.17.4.12 接收边沿

RX_NEG (SPI_CTL[1]) 定义数据在串行时钟 SPICLK 的下降沿还是上升沿被接收。注意 TX_NEG

和 RX_NEG 必须是互斥的。

5.17.4.13 字休眠

主机模式下，4 位位域 SP_CYCLE (SPI_CTL[15:12]) 提供一个在两个连续的事务之间可配置为 0.5 ~ 15.5 个串行时钟周期的休眠间隔。休眠间隔指从前一个传输事务的最后一个时钟沿到下一个传输事务的第一个时钟沿。SP_CYCLE 的缺省值是 0x3 (3.5 串行时钟周期)。如果软件禁止 FIFO 模式，该 SP_CYCLE 设定值对字休眠间隔不起作用。

如果 VARCLK_EN, SPI_CNTRL[23], 和 FIFO 位, SPI_CNTRL[21], 都被设置为 1, 最小字休眠间隔是 $(6.5 + SP_CYCLE) * SPI$ 串行时钟周期。

5.17.4.14 字节重排序

当一次传输被设置为 MSB 优先 (LSB = 0), 且 REORDER 被使能, 在 32 位 (TX_BIT_LEN = 0) 传输模式下, 存储在 SPI_TX FIFO 与 SPI_RX FIFO 中的数据将会按 [BYTE0, BYTE1, BYTE2, BYTE3] 的顺序被重新排列, 数据也将以 BYTE0, BYTE1, BYTE2, 和 BYTE3 的顺序进行发送/接收。如果 TX_BIT_LEN 被设置为 24 位模式, 存储在 SPI_TX 缓存与 SPI_RX 缓存中的数据将会按 [unknown byte, BYTE0, BYTE1, BYTE2] 的顺序被重新排列, 数据也将按照 BYTE0, BYTE1, BYTE2, 每个字节以 MSB 优先的顺序依次发送/接收数据。16 位模式的规则与上述相同。字节重排序功能仅当 TX_BIT_LEN 被配置为 16, 24, 和 32 位时有效

注: 可变时钟功能使能时, 不支持字节重排序功能。

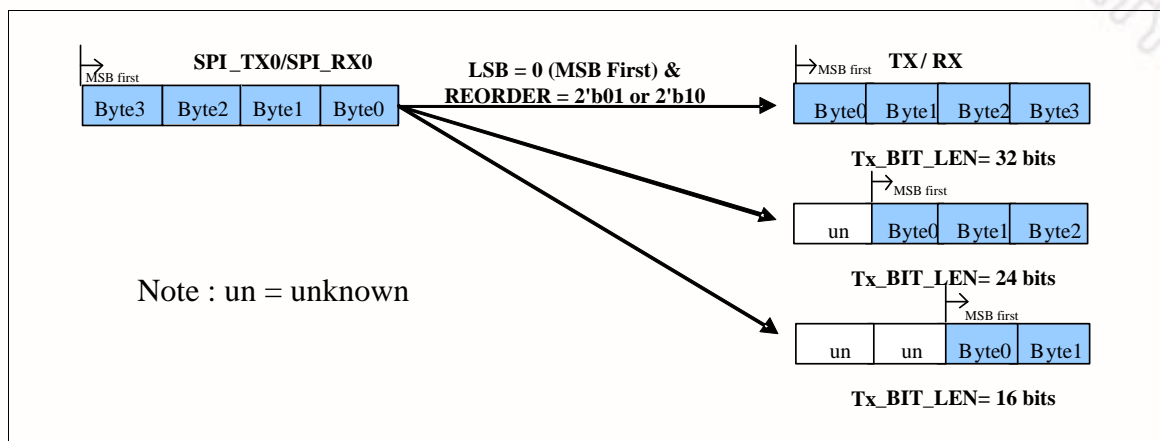


图 5-96 SPI 字节重排序

5.17.4.15 字节休眠

主机模式下, 如果 SPI_CTL[19] 被设置为 1, 在一次事务传输中, 硬件会在连续的两个字节之间插入一个 0.5~15.5 个串行时钟周期的休眠间隔。字节休眠和字休眠间隔的值均在 SP_CYCLE 中配置。

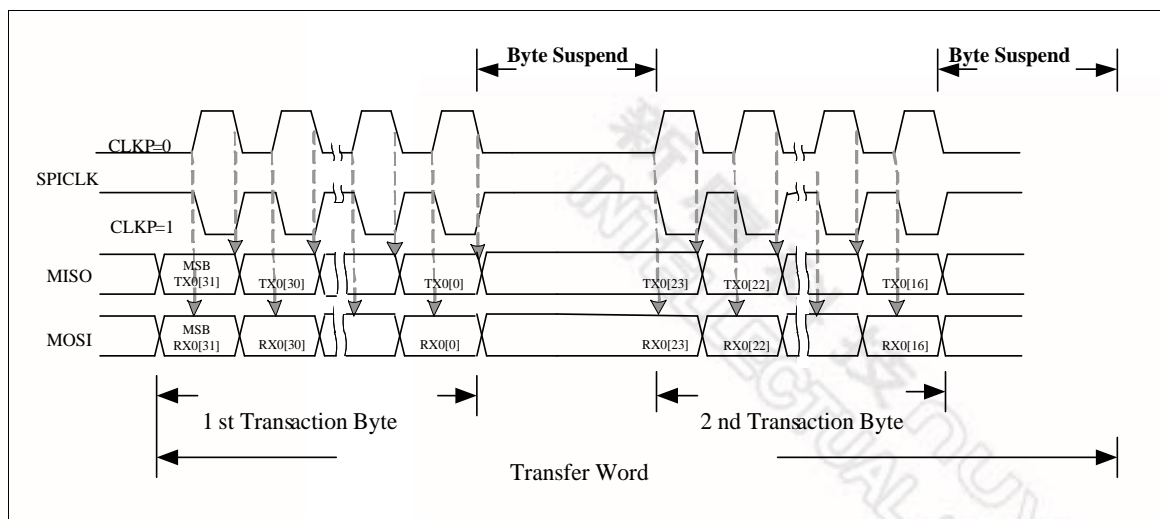


图 5-97 SPI 字节 休眠 模式

5.17.4.16 中断

当数据传输完成，或者 FIFO 达到阈值水平，每个 SPI 控制器可以产生独立的中断，各自的中断事件标志 INTSTS (SPI_STATUS[7], TX_INTSTS (SPI_STATUS[10]), RX_INTSTS (SPI_STATUS[8]) 将会被置位。如果中断使能位 INTEN (SPI_CTL[17]) 被置位，中断事件标志将会向 CPU 产生一个中断。中断事件标志 INTSTS, TX_INTSTS, 和 RX_INTSTS 仅可以通过向自身写 1 清除。

当 SPI 控制器完成一次单元传输，单元传输中断标志 IF (SPI_STS[7]) 将会被设置为 1。如果单元传输中断使能位 IE (SPI_CNTRL[17]) 被置位，单元传输中断事件将会向 CPU 产生一个中断。单元传输标志仅可以通过向自身写 1 清除。

在 3 线模式下，当从机检测到 SPI 时钟信号，3 线模式开始中断标志，SLV_START_INTSTS, 将会被置为 1。如果 SSTA_INTEN 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 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 位来中止当前传输。如果软件设置 SLV_ABORT 位为 1，单元传输中断标志，IF, 将会被置位。

在 FIFO 模式下，可以利用超时功能通知用户。如果在 FIFO 中有一个接收到的数据，且在主机模式下，该数据在 64 个 SPI 引擎时钟周期内没有被软件读取；或者在从机模式下，该数据在 576 个引擎时钟周期内没有被读取，如果超时中断使能位，FIFO_CTL[7], 被设置为 1，将会向系统发送一个超时中断。

5.17.4.17 两位传输模式

当设置 TWOB 位 (SPI_CTL[22]) 为 1 时，SPI 控制器支持两位 (two-bit) 传输模式。即当 TWOB 位被使能时，一个串行时钟周期可以同时发送或者接收两位串行数据。

第一位串行数据通过 MOSI0 或者 MISO0 口从 SPI_TX0 寄存器被发送或者被接收到 SPI_RX0 寄存器；第二位串行数据通过 MOSI1 或者 MISO1 口从 SPI_TX1 寄存器被发送或者被接收到 SPI_RX1 寄存器。下图是两位传输模式的系统模块和时序图。

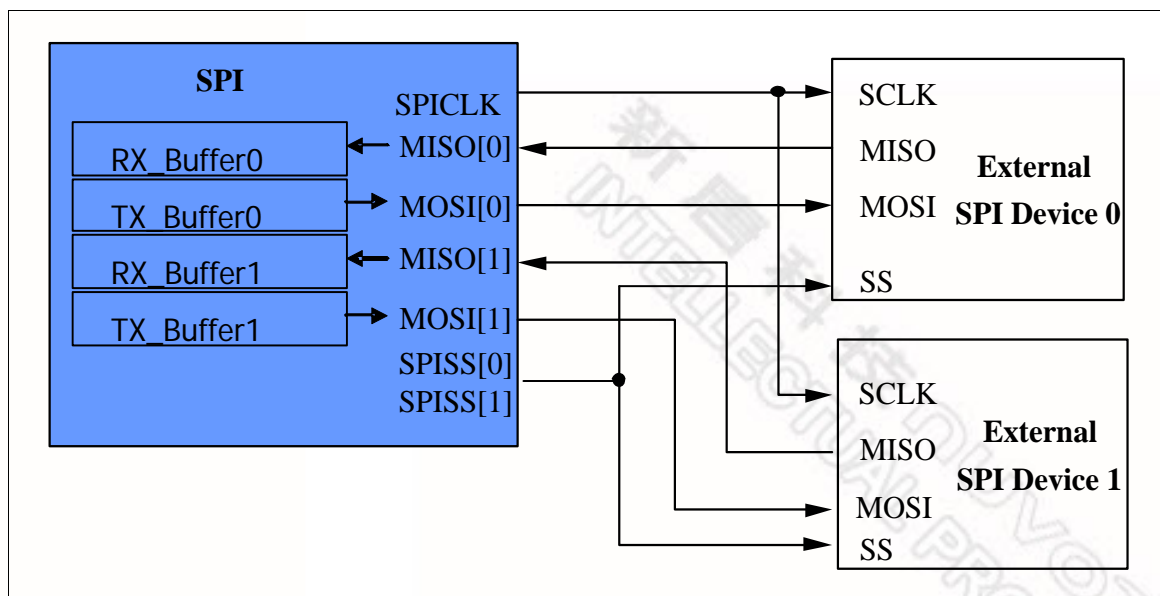


图 5-98 SPI 两位传输模式

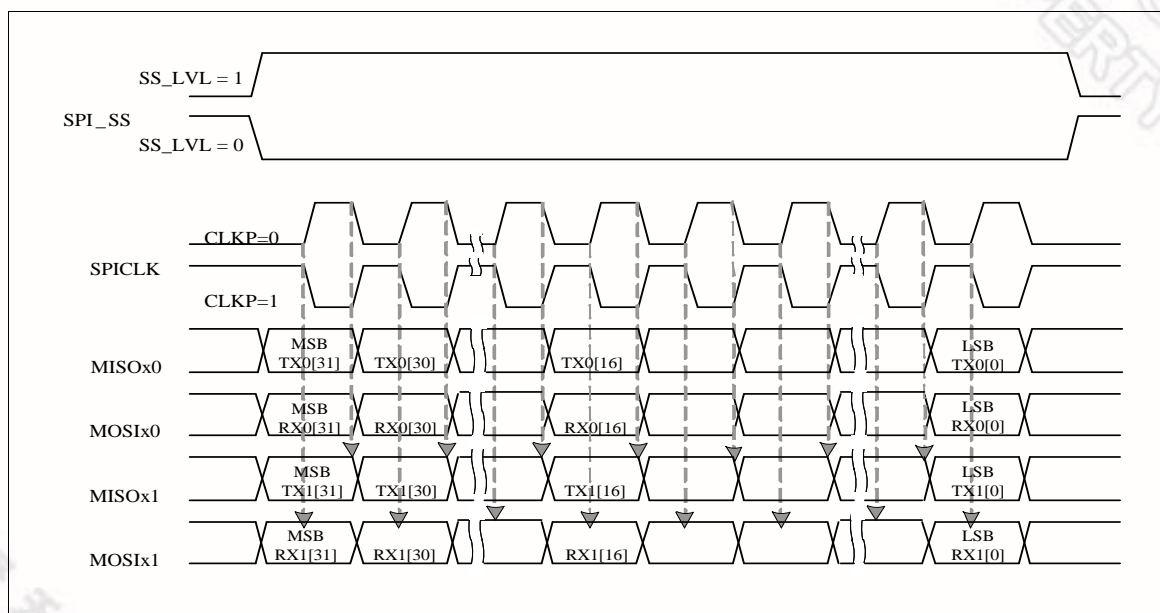


图 5-99 SPI 两位传输模式时序图

5.17.4.18 双 IO 模式

当 DUAL_IO_EN 位 (SPI_CTL[29]) 被设置为 1 时，SPI 控制器也支持双 IO 传输。DUAL_IO_DIR 位 (SPI_CTL2[28]) 用于定义数据传输的方向。当设置 DUAL_IO_DIR 位为 1，SPI 控制器向外部设备发送数据；当设置 DUAL_IO_DIR 位为 0，SPI 控制器从外部设备读取数据。该功能支持 8, 16, 24, 和 32 位数据长度。

当工作于 3 线模式，或者字节重排序功能被使能，不支持 SPI 控制器双 IO 模式。

如果 DUAL_IO_EN 和 DUAL_IO_DIR 位都被设置为 1，MOSI0 输出偶数位数据，而 MISO0 将会被作为奇数位数据输出管脚。如果 DUAL_IO_EN 被设置为 1 且 DUAL_IO_DIR 被设置为 0，MISO0 和 MOSI0 将会被设置为数据输入端口。

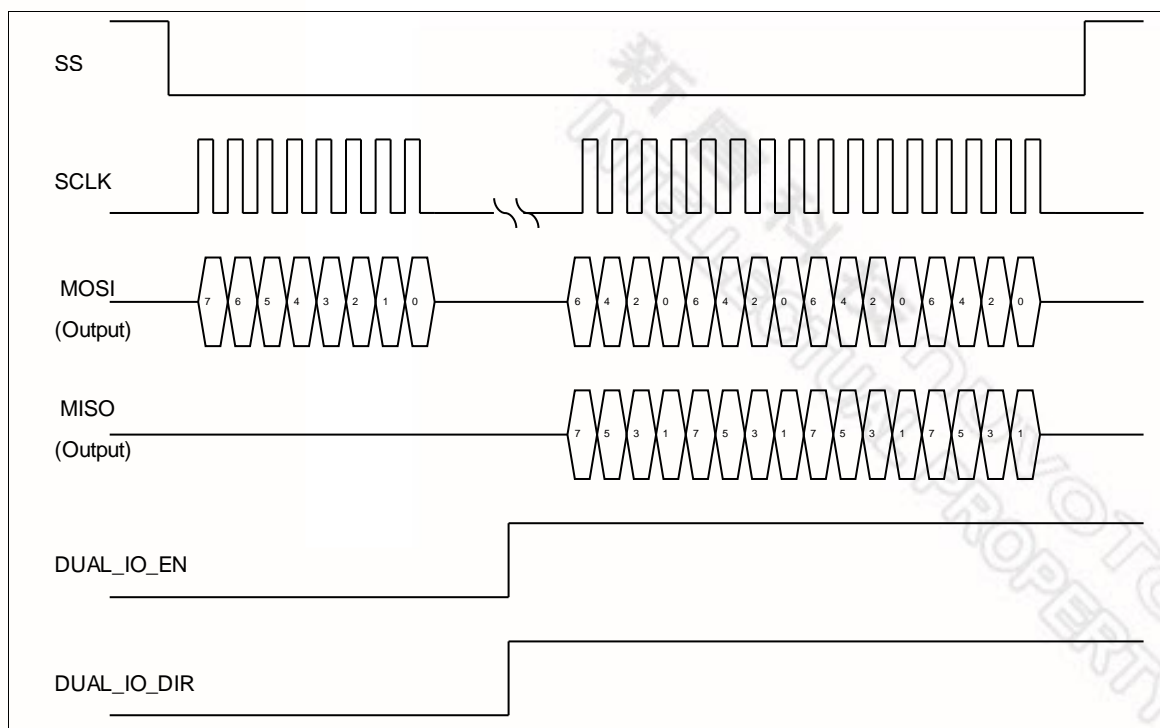


图 5-100 SPI DUAL-I/O 输出序列

注: 在 DUAL I/O 模式下不支持字节重排序和无从机选择功能。

5.17.4.19 超时

在 FIFO 模式下, 有一个超时功能用于通知用户。如果有一个接收到的数据在 FIFO 中, 且在主机模式下在 64 个 SPI 引擎时钟周期内没有被读取; 或者在从机模式下, 在 514 个 SPI 引擎时钟内没有被读取, 如果超时使能位, FIFO_CTL[7] 被设置为 1, 将会向系统发送一个中断。

5.17.4.20 FIFO 模式

当 FIFO 位 (SPI_CTL[21]) 被设为 1 时, SPI 控制器支持 FIFO 模式。SPI 控制器配备有 8 个 32 位宽的发送和接收 FIFO 缓存。

发送 FIFO 缓存是 8 级深度, 32 位宽, 先进先出的寄存器缓存。软件可以通过写 SPI_TX0 寄存器向发送 FIFO 缓存写入数据。存储在发送 FIFO 缓存中的数据将会被发送控制逻辑读取和发送出去。如果 8 级 FIFO 缓存满, TX_FULL 位将会被设置为 1。当 SPI 发送逻辑将 FIFO 缓存中的最后一个数据发送出去, 8 级发送 FIFO 缓存变为空, TX_EMPTY 位将会被设置为 1。注意 TX_EMPTY 标志是在最后一个事务仍在处理的过程中被设置为 1 的。在主机模式下, 软件应当同时检测 GO_BUSY 位和 TX_EMPTY 位以确认 SPI 是否处在空闲状态。

接收 FIFO 缓存也是 8 级深度, 32 位宽, 先进先出的寄存器缓存。接收控制逻辑将会存储接收到的数据到这个缓存。软件可以从 SPI_RX0 寄存器来读取该 FIFO 缓存的数据。有 FIFO 相关的状态位, 像 RX_EMPTY 和 RX_FULL 用于指示 FIFO 的当前状态。

在 FIFO 模式下, 软件可以通过设置 TX_THRESHOLD 和 RX_THRESHOLD 来设定发送和接收阈值。当存储在发送 FIFO 缓存中的有效数据数量小于或等于 TX_THRESHOLD 的设定值, TX_INTSTS 位将会被置为 1。当存储在接收 FIFO 缓存中的有效数据数量大于 RX_THRESHOLD

的设定值，RX_INTSTS 位将会被置为 1。

在 FIFO 模式下，软件可以事先写 8 个数据到 SPI 发送 FIFO 缓存。当 SPI 控制器操作在 FIFO 模式，SPI_CNTRL 寄存器中 GO_BUSY 位由硬件控制，软件除了清除 FIFO 位来禁止 FIFO 模式外，不能修改 SPI_CNTRL 寄存器的内容。

在主机发送操作中，当 FIFO 位被设置为 1，TX_EMPTY 标志将会被清除为 0，软件写第一个数据到 SPI_TX0 寄存器。只要发送 FIFO 缓存非空，发送操作立即开始，用户可以立即写接下来的数据到 SPI_TX0 寄存器。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_CNTRL [15:12]) 的设定值决定。只要 TX_FULL 标志为 0，用户就可以写数据到 SPI_TX0 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 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 将会被清除为 0。只要 RX_EMPTY 标志为 0，软件就可以从 SPI_RX0 寄存器读取接收到的数据。如果接收 FIFO 缓存包含 8 个未读数据，RX_FULL 标志将会被设置为 1。SPI 控制器将会停止接收数据，直到软件读取 SPI_RX0 寄存器。

在从机模式下，当 FIFO 位被设置为 1，GO_BUSY 位将会被硬件自动设置为 1。如果用户想停止从机模式下的 SPI 数据传输，软件必须把 FIFO 位和 GO_BUSY 都清除为 0。

在从机发送操作中，当软件写数据到 SPI_TX0 寄存器，数据将会被加载到发送 FIFO 缓存，且 TX_EMPTY 标志将会被设置为 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 标志为 0，软件就可以写数据到 SPI_TX0 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 寄存器，TX_EMPTY 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 寄存器。接收机制与主机模式接收操作类似。

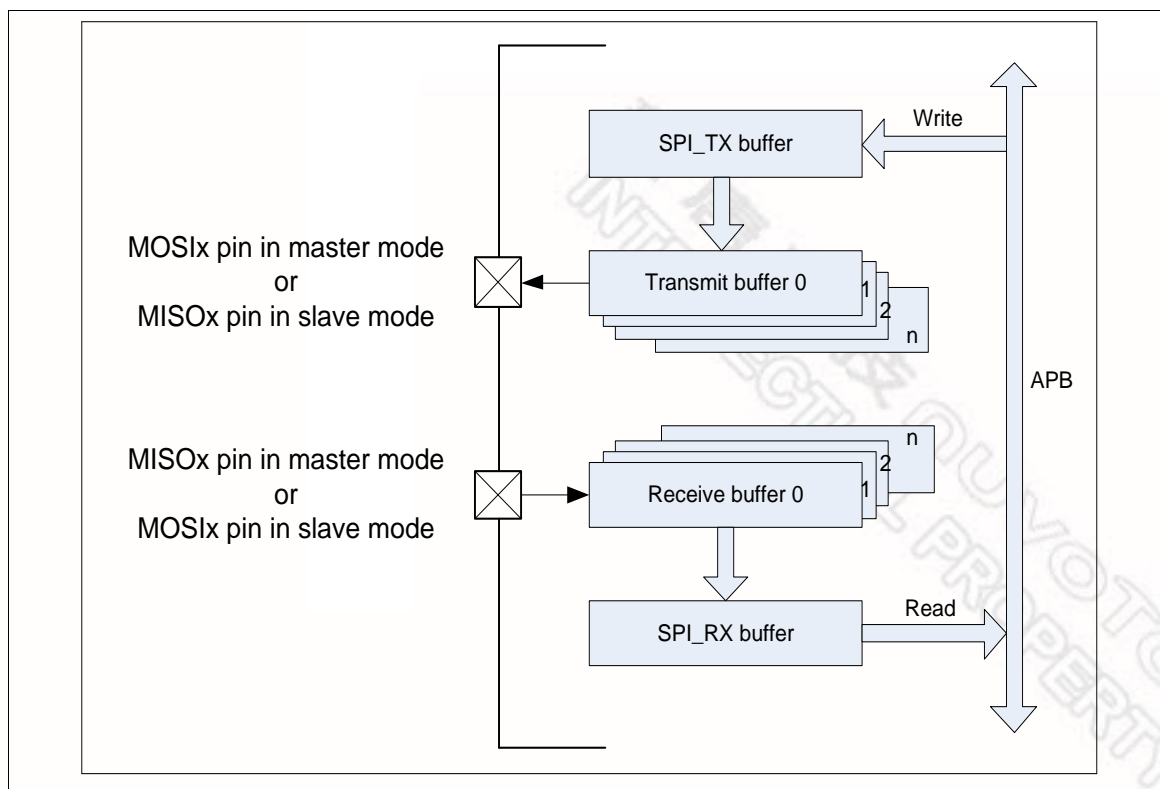


图 5-101 SPI FIFO 模式控制时序

5.17.4.21 唤醒

SPI 控制器支持唤醒功能。当系统处于掉电模式时，当在 SPICLK 管脚上接收到一个外部设备发过来的翻转信号时，CPU 将会被唤醒。

5.17.4.22 SPI 编程

在主机/从机模式，设备/从机选择信号 (SPI_SS) 可以在 SS_LVL 位 (SPI_SSR[2]) 被编程为低电平有效或者高电平有效，但是 SPISS0/1 是电平触发还是边沿触发是在 SS_LTRIG 位 (SPI_SSR[4]) 定义。串行时钟 (SPICLK) 空闲状态可以通过设定 CLKP 位 (SPI_CTL[11]) 来配置为高电平状态还是低电平状态。SPI 也提供对一个事务的位长通过位域 TX_BIT_LEN (SPI_CTL[7:3]) 进行设置，一次数据传输的事务数由位 TX_NUM (SPI_CTL[8]) 定义，是从 MSB 还是 LSB 开始发送/接收数据由 LSB 位 (SPI_CTL[10]) 定义。用户也可以通过设置位域 TX_NEG/RX_NEG (SPI_CTL[2:1]) 来选择是上升沿还是下降沿来发送/接收数据。主机/从机模式下的四种 SPI 时序操作和相关的设定如下图所示。

注：Tx_NEG = TX_NEG; Rx_NEG = RX_NEG 和 Tx_BIT_LEN = TX_BIT_LEN

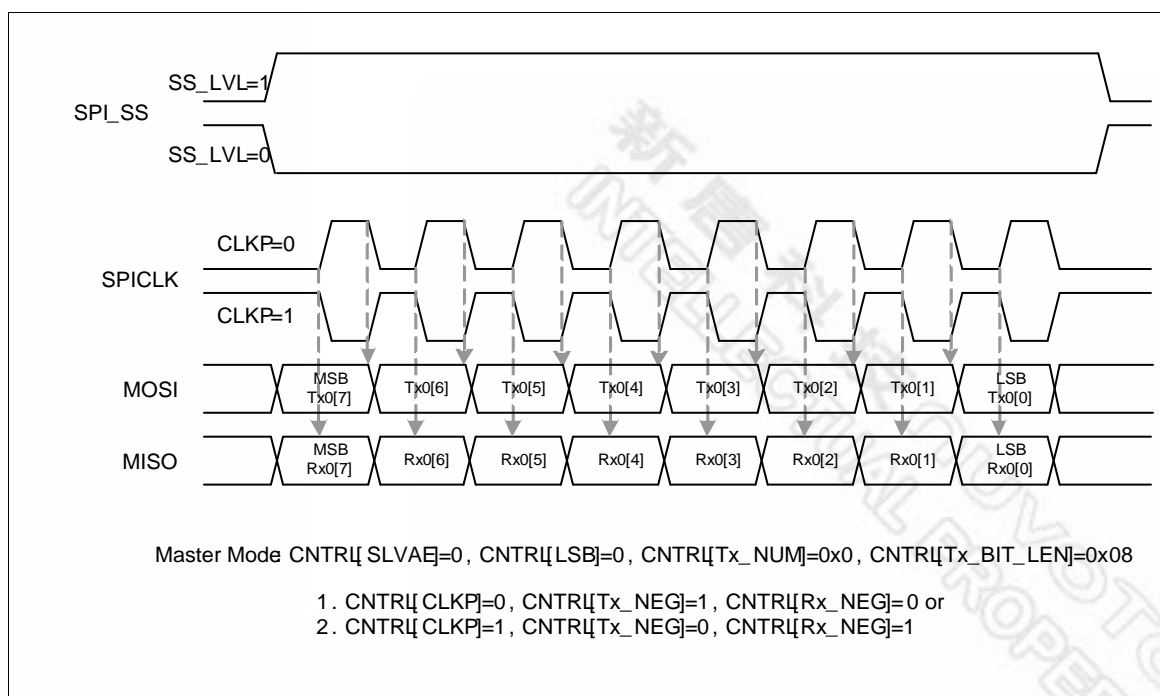


图 5-102 主机模式下 SPI 时序

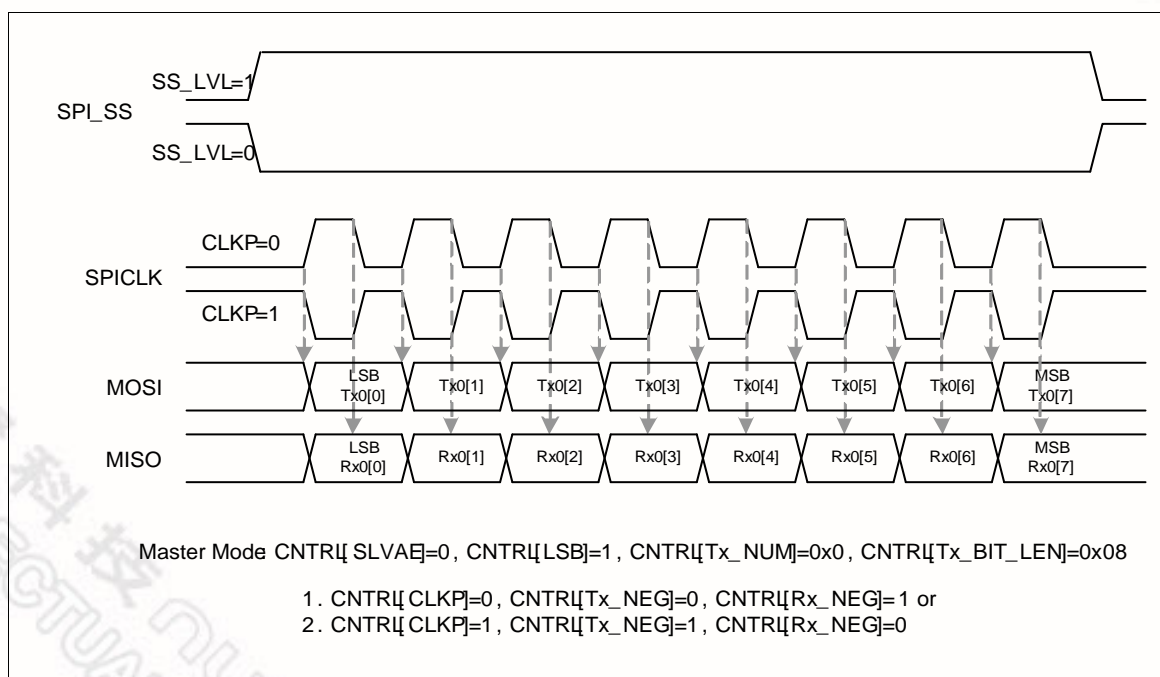


图 5-103 主机模式下 SPI 时序(Alternate Phase of SPICLK & LSB = 1)

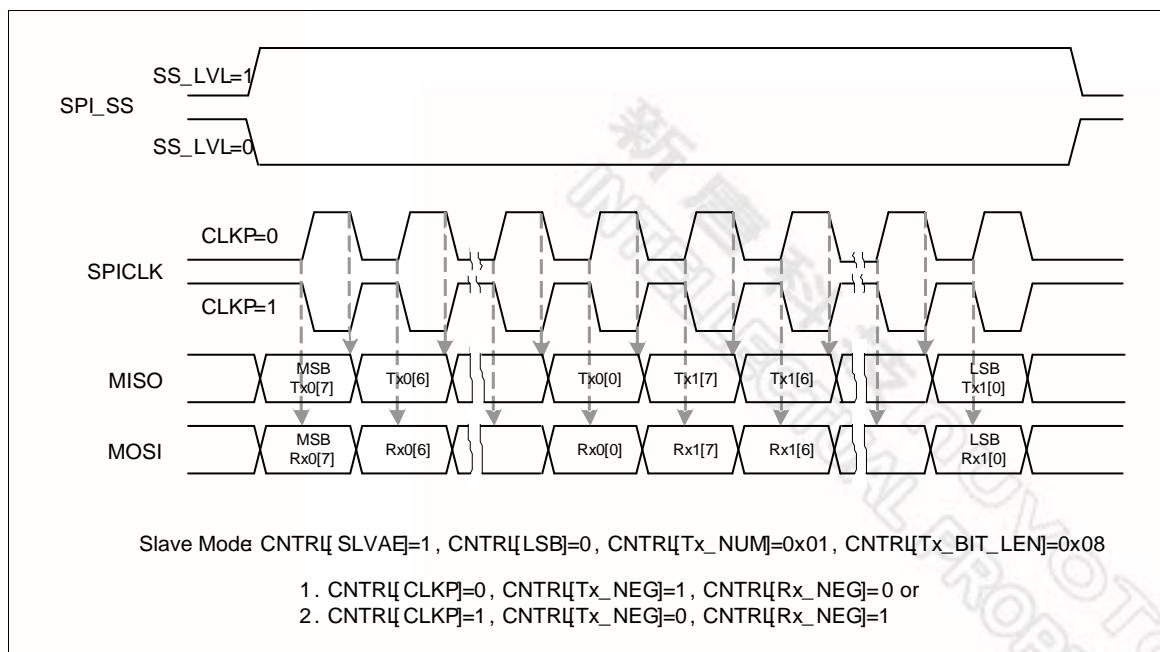


图 5-104 主机模式下 SPI 时序(Alternate Phase of SPICLK & LSB = 0)

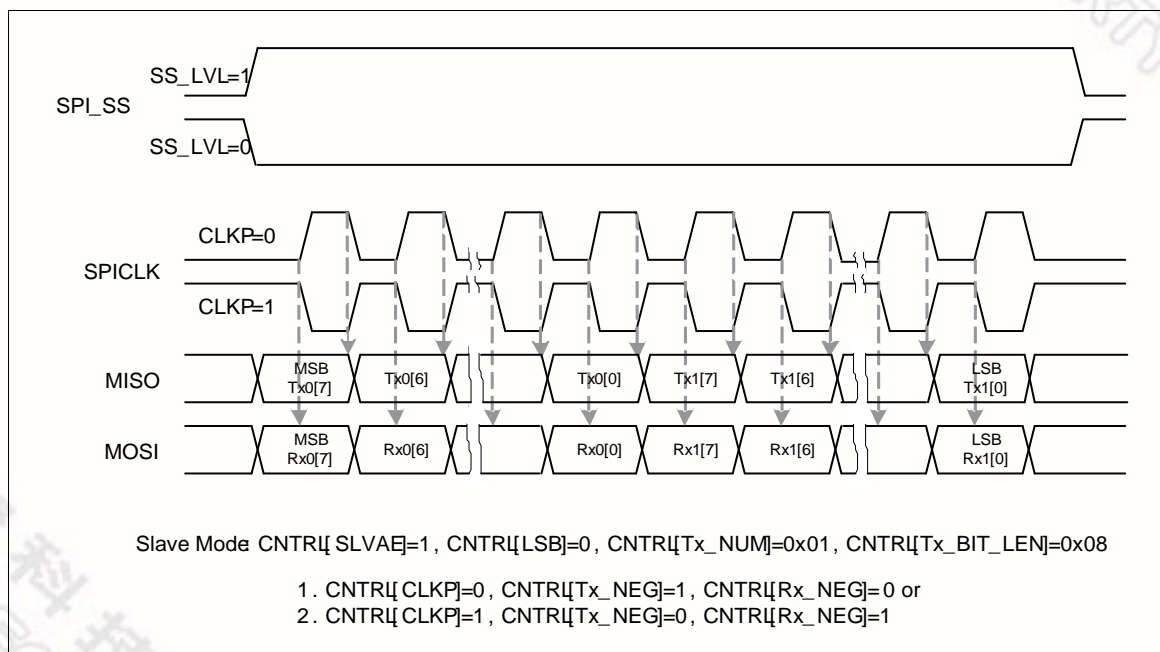


图 5-105 从机模式下 SPI 时序

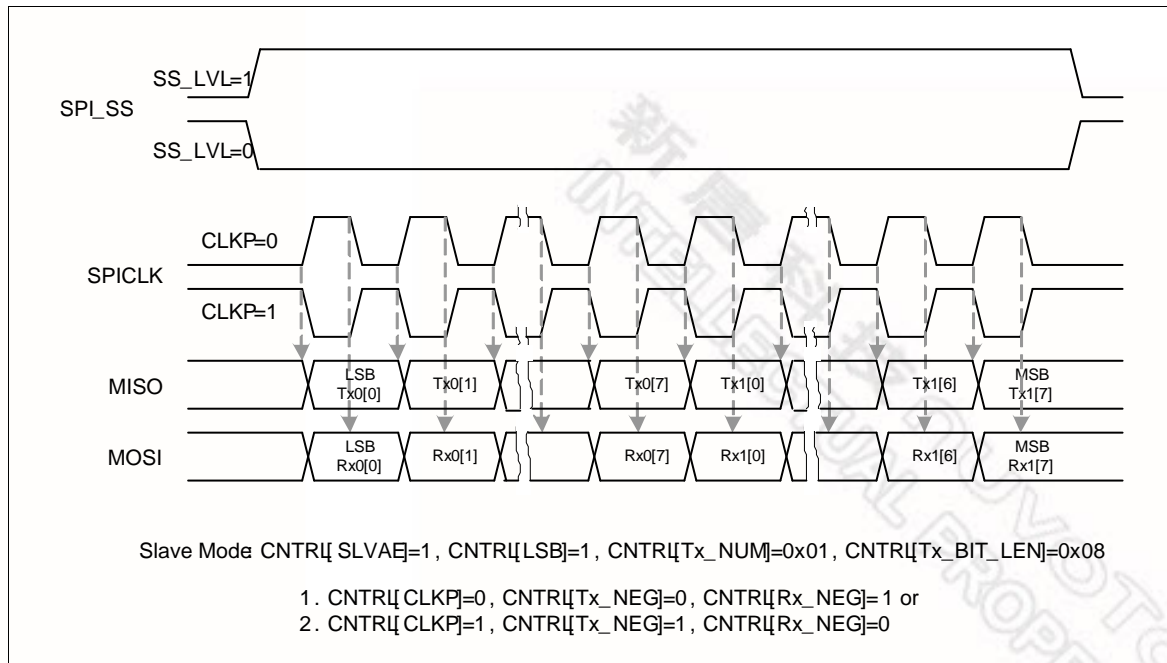


图 5-106 从机模式下 SPI 时序(Alternate Phase of SPICLK)

5.17.4.23 SPI 编程示例

例 1, SPI 控制器被设置为主机去访问片外从设备, 规格如下:

- 数据在串行时钟上升沿被接收
- 数据在串行时钟下降沿被发送
- MSB 先传输
- SPICLK 空闲状态为低电平
- 每次数据传输只发送/接收一个字节
- 从机选择信号为低电平有效

基本上, 在进行如下步骤之前, 应当详细参考一下所连接的片外从机设备的规格书:

1. 设定寄存器 DIVIDER1 (SPI_CLKDIV[7:0]) 来决定串行时钟输出频率.
2. 写一个合适的值到 SPI_SSR 寄存器来对主模式相关配置进行设定
 - (a). 禁止自动从选择位 AUTOSS (SPI_SSR[3] = 0), (b). 在从机选择有效电平位 SS_LVL (SPI_SSR[2] = 0) 和从机选择电平触发位 SS_LTRIG 设定从机选择为低电平触发输出. (c). 通过设定从机选择寄存器位 SSR[0] 或 SSR[1] (SPI_SSR[1:0]) 选择哪一个从机选择信号会在相应的 IO 管脚上输出, 以激活片外从设备.
3. 写相关设定值到 SPI_CTL 寄存器 (1). 在 SLAVE 位 (SPI_CTL[18] = 0) 中设定该 SPI 控制器工作在主机模式. (2). 在 CLKP 位 (SPI_CTL[11] = 0) 设定串行时钟空闲状态为低电平, (3). 在 TX_NEG 位 (SPI_CTL[2] = 1) 选择在串行时钟的下降沿发送数据. (4). 在 RX_NEG 位 (SPI_CTL[1] = 0) 选择在串行时钟的上升沿接收数据. (5). 在 TX_BIT_LEN 位域 (SPI_CTL[7:3] = 0x08) 设定事务位长为 8. (6). 在 LSB 位 (SPI_CTL[10] = 0) 设定 MSB 优先传输.
4. 如果该 SPI 主机准备发送 (写) 一个字节数据到片外从机设备, 该字节数据将会被写入 SPI_TX0 寄存器. .

5. 如果该 SPI 主机只是从片外从机设备接收 (读) 一个字节数据, 用户不用关心什么样的数据将会被发送, 软件不需要去更新 SPI_TX0 寄存器。
6. 使能 GO_BUSY 位 (SPI_CTL[0] = 1) 开始 SPI 数据传输。
7. 等待 SPI 中断发生 (如果中断使能位 INTEN 被置位), 或者轮询 GO_BUSY 位, 直到该位被硬件自动清零。
8. 从 RDATA0[7:0] (SPI_RX0[7:0]) 寄存器读出接收到的一个字节数据。
9. 跳转到步骤 4) 继续另一次数据传输, 或者设置 SSR[0] 或 SSR[1] 为 0 来置片外从机设备为非激活状态。

例 2, SPI 控制器作为从机设备, 由一片外主机设备控制, 假设片外主机设备访问片内 SPI 从机控制器的 SPI 接口设定如下:

- 数据在串行时钟上升沿被接收
 - 数据在串行时钟下降沿被发送
 - LSB 先传输
 - SPICLK 空闲状态为高电平
 - 在一次数据传输中仅发送/接收一个字节
 - 从机选择信号为高电平触发
1. 基本上, 在进行如下步骤之前, 应当详细参考一下所连接的片外主机设备的规格书: 在从机选择有效电平位 SS_LVL (SPI_SSR[2] = 1) 和从机选择电平触发位 SS_LTRIG (SPI_SSR[4] = 1) 设定从机选择信号为高电平和电平触发。
 2. 写相关的设定值到 SPI_CTL 寄存器来控制该 SPI 从机动作。通过 SLAVE 位 (SPI_CTL[18] = 1) 来设定 SPI 控制器作为从机设备。通过 CLKP 位 (SPI_CTL[11] = 1) 来设定串行时钟空闲状态为高电平, 通过 TX_NEG 位 (SPI_CTL[2] = 1) 来设定数据在串行时钟下降沿被发送, 通过 RX_NEG 位 (SPI_CTL[1] = 0) 来设定数据在串行时钟的上升沿被接收。通过 TX_BIT_LEN 位域 (SPI_CTL[7:3] = 0x08) 来设置一个传输事务的数据长度为 8 位。通过 LSB 位 (SPI_CTL[10] = 1) 设定 LSB 优先传输。SP_CYCLE 位域 (SPI_CTL[15:12]) 不需要关心, 因为本例中没有使用突发模式。
 3. 如果该 SPI 从机准备发送 (被读) 一个字节数据到片外主机设备, 该字节数据将会被写入 SPI_TX0 寄存器。
 4. 如果该 SPI 从机只是从片外主机设备接收 (被写) 一个字节数据, 用户不用关心什么样的数据将会被发送, 软件不需要更新 SPI_TX0 寄存器。
 5. 使能 GO_BUSY 位 (SPI_CTL[0] = 1), 等待从机选择触发信号和串行时钟从片外主机设备输入来开始 SPI 数据传输。
 6. 等待 SPI 中断发生 (如果中断使能位 INTEN 被置位), 或者轮询 GO_BUSY 位, 直到该位被硬件自动清零。
 7. 从 SPI_RX0[7:0] 寄存器读出接收到的一个字节数据。
 8. 跳转到步骤 3) 继续另一次数据传输, 或者停止数据传输。

5.17.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write, C: Write 1 Clear

寄存器	偏移量	R/W	描述	复位后的值
SPI0_BA = 0x4003_0000 SPI1_BA = 0x4013_0000 SPI2_BA = 0x400D_0000 .				
SPI_CTL	SPIx_BA+0x00	R/W	SPI 控制寄存器	0x0000_0004
SPI_STATUS	SPIx_BA+0x04	R/W/C	SPI 状态寄存器	0x0000_0005
SPI_CLKDIV	SPIx_BA+0x08	R/W	SPI 时钟分频寄存器	0x0000_0000
SPI_SSR	SPIx_BA+0x0C	R/W	SPI 从机选择寄存器	0x0000_0000
SPI_RX0	SPIx_BA+0x10	R	SPI 接收数据缓冲寄存器 0	0x0000_0000
SPI_RX1	SPIx_BA+0x14	R	SPI 接收数据缓冲寄存器 1	0x0000_0000
SPI_TX0	SPIx_BA+0x20	W	SPI 发送数据缓冲寄存器 0	0x0000_0000
SPI_TX1	SPIx_BA+0x24	W	SPI 发送数据缓冲寄存器 1	0x0000_0000
SPI_VARCLK	SPIx_BA+0x34	R/W	SPI 可变时钟模式标志寄存器	0x007F_FF87
SPI_DMA	SPIx_BA+0x38	R/W	SPI PDMA 控制寄存器	0x0000_0000
SPI_FFCLT	SPIx_BA+0x3C	R/W	SPI FIFO 控制寄存器	0x0000_0000
SPI_INTERNAL	SPIx_BA+0x50	R/W	SPI 内部寄存器	0x0000_0000

注: 在接下来的寄存器描述中, 对每个 SPI 模块来说, “x” 表示 0~2。例如, SPIx_BA 包括 SPI0_BA, SPI1_BA 和 SPI2_BA。

5.17.6 寄存器描述

SPI 控制寄存器 (SPI_CTL)

寄存器	偏移量	R/W	描述	复位后的值
SPI_CTL	SPIx_BA + 0x00	R/W	SPI 控制寄存器	0x0000_0004

31	30	29	28	27	26	25	24
WKEUP_EN	-	DUAL_IO_EN	DUAL_IO_DIR			-	
23	22	21	20	19	18	17	16
VARCLK_EN	TWOB	FIFOM		REORDER	SLAVE	INTEN	-
15	14	13	12	11	10	9	8
SP_CYCLE				CLKP	LSB		
7	6	5	4	3	2	1	0
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

Bits	描述	
[31]	WKEUP_EN	<p>唤醒使能</p> <p>1 = 使能唤醒功能.</p> <p>0 = 当系统进入掉电模式时禁止唤醒功能</p> <p>系统进入掉电模式时, 在该位被使能的情况下, 若在 SPICLK 管脚上有电平翻转, 系统可以由 SPI 控制器唤醒。在系统被唤醒之后, 该位必须由用户清除, 以禁止唤醒需求.</p>
[30]	-	保留
[29]	DUAL_IO_EN	<p>双 IO 模式使能</p> <p>1 = 使能双 I/O 模式</p> <p>0 = 禁用双 I/O 模式</p>
[28]	DUAL_IO_DIR	<p>双 IO 模式方向</p> <p>1 = 双 I/O 模式数据写</p> <p>0 = 双 I/O 模式数据读</p>
[27:24]	-	保留
[23]	VARCLK_EN	<p>可变时钟使能</p> <p>1 = 串行时钟频率可变。输出频率由 VARCLK (SPI_VARCLK), DIVIDER1 和 DIVIDER2 的值决定。</p> <p>0 = 串行时钟频率固定, 并且仅由 DIVIDER1 的值决定。注: 当 VARCLK_EN 被设置为 1, TX_BIT_LEN 必须被编程为 0x10 (16 位模式)。</p>
[22]	TWOB	<p>两位传输模式使能</p> <p>1 = 使能两位传输模式</p>

		0 = 禁止两位传输模式。 注： 当使能 TWOB，两位串行数据从 SPI_TX1/0 被发送，被接收到 SPI_RX1/0。						
[21]	FIFOM	FIFO 模式使能 1 = FIFO 模式。 0 = 正常模式 注： 1. 在使能 FIFO 模式前，其他相关的设定必须先设定 2. 2. 在主机模式下，如果 FIFO 模式被使能，在数据被写入 8 级深度 FIFO 中之后，GO_BUSY 将会自动被硬件设置为 1。用户可以在发送 FIFO 状态为空且 GO_BUSY 恢复到 0 之后，清除该 FIFO 位。						
[19]	REORDER	字节重排序功能使能 <table border="1"><thead><tr><th>REORDER</th><th>描述</th></tr></thead><tbody><tr><td>0</td><td>禁止字节重排序功能</td></tr><tr><td>1</td><td>使能字节重排序功能，并且在每两个字节之间插入一个字节休眠间隔。TX_BIT_LEN 必须被配置为 00b (32 bits/ word)</td></tr></tbody></table> 休眠间隔在 SP_CYCLE 中定义。 注： 1. 字节重排序仅在 TX_BIT_LEN 被定义为 16 位，24 位和 32 位时有效。 2. 在电平触发的从机模式下，如果字节休眠功能被使能，从机选择管脚必须在连续的四个字节传输之间保持有效 3. 当可变时钟功能或者双 I/O 模式被使能时，不支持字节重排序功能。	REORDER	描述	0	禁止字节重排序功能	1	使能字节重排序功能，并且在每两个字节之间插入一个字节休眠间隔。TX_BIT_LEN 必须被配置为 00b (32 bits/ word)
REORDER	描述							
0	禁止字节重排序功能							
1	使能字节重排序功能，并且在每两个字节之间插入一个字节休眠间隔。TX_BIT_LEN 必须被配置为 00b (32 bits/ word)							
[18]	SLAVE	从机模式 1 = SPI 控制器被设置为从机模式 0 = SPI 控制器被设置为主机模式。						
[17]	INTEN	中断使能 1 = 使能 SPI 中断。 0 = 禁止 SPI 中断。						
[16]	-	保留						
[15:12]	SP_CYCLE[3:0]	休眠间隔(Master Only) 该四位提供在一次数据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。如果 CLKP =0，休眠间隔从当前事务的最后一个串行时钟的下降沿开始到接下来的事务的串行时钟的第一个上升沿结束；而如果 CLKP = 1，休眠间隔从当前事务的最后一个串行时钟的上升沿开始到接下来的事务的串行时钟的第一个下降沿结束。 缺省值是 0x3. 要求的休眠间隔根据如下公式获得： (SP_CYCLE[3:0] + 0.5) * period of SPICLK 例： SP_CYCLE = 0x0 ... 0.5 SPICLK 时钟周期						

		<div>SP_CYCLE = 0x1 ... 1.5 SPICLK 时钟周期</div> <div>.....</div> <div>SP_CYCLE = 0xE ... 14.5 SPICLK 时钟周期</div> <div>SP_CYCLE = 0xF ... 15.5 SPICLK 时钟周期</div> <div>如果可变时钟功能使能，在连续的两个事务之间最小休眠间隔时间 (在 FIFO 缓存中发送数据非空) 是 (6.5 + SP_CYCLE) * SPICLK clock cycle.</div>												
[11]	CLKP	<div>时钟极性</div> <div>1 = SPICLK 空闲状态的电平默认为高</div> <div>0 = SPICLK 空闲状态的电平默认为低.</div>												
[10]	LSB	<div>LSB 优先</div> <div>1 = LSB, SPI_TX0/1 的位 0，首先被发送到 SPI 数据输出管脚；从 SPI 数据输入管脚上接收到的第一个数据位将会被放置到 SPI_RX 寄存器 (SPI_RX0/1) LSB 的位置.</div> <div>0 = MSB, 具体是发送/接收寄存器的哪一位，首先被发送/接收，取决于 TX_BITLEN 的设定值。</div>												
[9:8]		保留												
[7:3]	TX_BIT_LEN[4:0]	<div>传输位长</div> <div>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。</div> <table><tr><th>TX_BIT_LEN</th><th>描述</th></tr><tr><td>01000</td><td>在一个事务中，8 位数据被传输</td></tr><tr><td>01001</td><td>在一个事务中，9 位数据被传输</td></tr><tr><td>-----</td><td>-----</td></tr><tr><td>11111</td><td>在一个事务中，31 位数据被传输</td></tr><tr><td>00000</td><td>在一个事务中，32 位数据被传输</td></tr></table>	TX_BIT_LEN	描述	01000	在一个事务中，8 位数据被传输	01001	在一个事务中，9 位数据被传输	-----	-----	11111	在一个事务中，31 位数据被传输	00000	在一个事务中，32 位数据被传输
TX_BIT_LEN	描述													
01000	在一个事务中，8 位数据被传输													
01001	在一个事务中，9 位数据被传输													
-----	-----													
11111	在一个事务中，31 位数据被传输													
00000	在一个事务中，32 位数据被传输													
[2]	TX_NEG	<div>在下降沿发送</div> <div>1 = 在 SPICLK 的下降沿发送数据</div> <div>0 = 在 SPICLK 的上升沿发送数据</div>												
[1]	RX_NEG	<div>Receive At Negative Edge</div> <div>1 = 在 SPICLK 的下降沿锁存数据.</div> <div>0 = 在 SPICLK 的上升沿锁存数据.</div>												
[0]	GO_BUSY	<div>SPI 传输控制位和忙状态</div> <div>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</div> <div>0 = 如果 SPI 传输正在进行，写 0 到该位停止数据传输.</div> <div>如果 FIFO 模式被禁止，在整个数据传输过程中，该位保持为 ‘1’。当传输结束，该位自动被清除。软件可以读取该位来检查 SPI 是否处于忙状态。</div> <div>在 FIFO 模式下，该位由硬件控制。在软件不能修改该位的值。从机模式下，软件读该寄存器总是返回 1。在主机模式下，该位反映 SPI 是处于忙，还是空闲状态。</div> <div>注:</div>												

		<ol style="list-style-type: none"> 1. 当 FIFO 模式被禁止，在写 1 到 GO_BUSY 位之前，所有配置必须事先在 SPI_CTL 寄存器中被设定好。 2. 当 FIFO 被禁止，且软件使用 TX 或 RX PDMA 功能来传输数据，在 PDMA 控制器结束数据传输之后，该位将会被清除。
--	--	--

SPI 状态寄存器 (SPI_STATUS)

寄存器	偏移量	R/W	描述	复位后的值
SPI_STATUS	SPIx_BA + 0x04	R/W/C	SPI 状态寄存器	0x0000_0005

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
TX_FIFO_CNT				RX_FIFO_CNT			
15	14	13	12	11	10	9	8
-			TIME_OUT_STS	-	TXINT_STS	RX_OVER_RUN	RXINT_STS
7	6	5	4	3	2	1	0
INTSTS	SLV_START_INTSTS	-	LTRIG_FLAG	TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY

Bits	描述	
[31:24]	-	保留
[23:20]	TX_FIFO_CNT	TX FIFO 中的数据量(只读)
[19:16]	RX_FIFO_CNT	RX FIFO 中的数据量(只读)
[15:13]	-	保留
[12]	TIME_OUT_STS	<p>TIMEOUT 中断标志</p> <p>1 = RX FIFO 非空, 且在主机模式下, 在 64 个 SPI 引擎时钟周期内没有被读; 或者在从机模式下, 在 576 个 SPI 引擎时钟周期内没有被读。当接收 FIFO 被用户读取, 超时状态将会被自动清除。</p> <p>0 = 接收缓存没有超时事件。</p> <p>注: 该位写 1 清除</p>
[11]	-	保留
[10]	TXINT_STS	<p>TX FIFO 阈值中断状态 (只读)</p> <p>1 = TX 有效数据数量小于或等于 TXTHRESHOLD.</p> <p>0 = TX 有效数据数量大于 TXTHRESHOLD.</p>
[9]	RX_OVER_RUN	<p>RX FIFO 溢出标志</p> <p>如果 SPI 在 RX FIFO 满时接收数据, 该位将会被设置为 1。接收到的数据将会被丢弃。</p> <p>注: 该位写 1 清除</p>
[8]	RXINT_STS	RX FIFO 阈值中断状态 (只读)

		<p>1 = RX 有效数据数量大于 RXTHRESHOLD.</p> <p>0 = RX 有效数据数量小于或等于 RXTHRESHOLD.</p> <p>注: 如果 RXINT_EN = 1 且 RX_INTSTS = 1, SPI 将产生中断.</p>
[7]	INTSTS	<p>中断状态</p> <p>1 = 表示数据传输完成。当 INTEN 位被使能, 数据传输完成中断将被请求</p> <p>0 = 表示数据传输尚未完成</p> <p>注: 该位只读, 可以由软件向该位写 1 清除该标志</p>
[6]	SLV_START_INTSTS	<p>从机开始中断状态</p> <p>用于在 3 线从机模式下, 指示数据传输已经开始。</p> <p>1 = 表示在 3 线从机模式下, 数据传输已经开始。该标志可以在数据传输完成时自动被清除, 或者由软件写 1 清除。</p> <p>0 = 表示从机数据传输开始没有被激活。</p>
[4]	LTRIG_FLAG	<p>电平触发完成标志 (INTERNAL ONLY)</p> <p>在从机模式下, 该位指示在当前传输完成之后, 接收到的数据位数是否满足要求。</p> <p>1 = 传输位长满足在 TX_BIT_LEN 定义的要求。</p> <p>0 = 一个事务中传输位长不满足要求</p> <p>注: 该位只读。当软件设置 GO_BUSY 位为 1, 在 4 个 SPI 引擎周期和一个系统时钟周期之后, LTRIG_FLAG 将会被清除为 0。在 FIFO 模式下, 该位无意义。</p>
[3]	TX_FULL	<p>发送 FIFO_FULL 状态</p> <p>1 = 表示在 FIFO 模式下, 发送数据 FIFO 满。</p> <p>0 = 表示在 FIFO 模式下, 发送数据 FIFO 未滿</p>
[2]	TX_EMPTY	<p>发送 FIFO_EMPTY 状态</p> <p>1 = 表示在 FIFO 模式下, 发送数据 FIFO 空。</p> <p>0 = 表示在 FIFO 模式下, 发送数据 FIFO 非空</p>
[1]	RX_FULL	<p>接收 FIFO_FULL 状态</p> <p>1 = 表示在 FIFO 模式下, 接收数据 FIFO 满。</p> <p>0 = 表示在 FIFO 模式下, 接收数据 FIFO 未滿。</p>
[0]	RX_EMPTY	<p>接收 FIFO_EMPTY 状态</p> <p>1 = 表示在 FIFO 模式下, 接收数据 FIFO 空。</p> <p>0 = 表示在 FIFO 模式下, 接收数据 FIFO 非空。</p>

SPI 串行时钟分频寄存器 (SPI_CLKDIV)

寄存器	偏移量	R/W	描述	复位后的值
SPI_CLKDIV	SPIx_BA + 0x08	R/W	串行时钟分频寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
DIVIDER2							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
DIVIDER1							

Bits	描述	
[31:24]	-	保留
[23:16]	DIVIDER2[7:0]	<p>时钟除频器 2 寄存器</p> <p>该位域的值是外设时钟 PCLK 的第二个频率除频值，用于产生串行时钟 SPICLK。要求的频率值根据如下等式获得：</p> $f_{sclk} = \frac{f_{eclk}}{(DIVIDER2 + 1) * 2}$
[15:8]	-	保留
[7:0]	DIVIDER1[7:0]	<p>时钟除频器 1 寄存器</p> <p>该位域的值是外设时钟 PCLK 的第一个频率除频值，用于产生串行时钟 SPICLK。要求的频率值根据如下等式获得：</p> $f_{sclk} = \frac{f_{eclk}}{(DIVIDER1 + 1)}$ <p>f_{eclk} 是 SPI 引擎时钟源，在 CLK_SEL1 中定义。</p>

SPI 从机选择寄存器 (SPI SSR)

寄存器	偏移量	R/W	描述	复位后的值
SPI_SSR	SPIx_BA + 0x0C	R/W	从机选择寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							SS_INT_OPT
15	14	13	12	11	10	9	8
-						SSTA_INTEN	SLV_ABORT
7	6	5	4	3	2	1	0
-		NOSLVSEL	SS_LTRIG	AUTOSS	SS_LVL	SSR	

Bits	描述	
[31:10]	-	保留
[16]	SS_INT_OPT	<p>从机选择中断选项</p> <p>该位用于在从机模式下，当传输完成时使能从机选择中断。</p> <p>1 = 当从机选择无效时，有一个中断事件，用于通知用户事务已经结束，从选择进入无效状态。</p> <p>0 = 无任何中断，即便有从机选择无效事件。</p>
[9]	SSTA_INTEN	<p>从机开始中断使能</p> <p>该位用于使能在 3 线从机模式下传输已经开始中断。如果在传输开始以后，在由用户定义的时间内，没有传输完成中断，用户可以设置 SLV_ABORT 位来强制传输完成。</p> <p>1 = 事务开始中断使能。当当前传输完成或 SLV_START_INTSTS 位被清除时(写 1 清除)，该位被清除。</p> <p>0 = 传输开始中断禁止。</p>
[8]	SLV_ABORT	<p>Abort in Slave Mode with No Slave Selected</p> <p>在正常操作模式下，当接收到的数据位数满足在 TX_BIT_LEN 位域中定义的需求时，会有一个中断事件。</p> <p>在 3 线从机模式下，如果接收到的数据位数小于在 TX_BIT_LEN 位域中定义的需求，并且在由用户定义的时间内都没有另外的串行时钟输入的情况下，用户可以通过设置该位来强制结束当前传输，这种情况下，用户可以检测到一个传输完成中断事件。</p> <p>注：当退出事件被激活，该位由硬件自动清 0..</p>
[5]	NOSLVSEL	<p>从机模式下无从机选择信号 (3 线从机模式)</p> <p>在从机模式下，使能该位，从机选择信号将被忽略，SPI 控制器可以工作在包括 SPICLK, SPI_MISO 和 SPI_MOSI 在内的 3 线 SPI 接口模式。</p> <p>1 = SPI 控制器工作在 3 线双向接口从机模式。在该位被设置为 1 的情况下，当 GO_BUSY 被置位，且有串行时钟通过 SPICLK 引脚输入进来时，SPI 控制器开始</p>

		<p>发送/接收数据。</p> <p>0 = SPI 控制器工作在 4 线双向接口模式。</p> <p>注：在无从机选择信号的模式下，SS_LTRIG 位 (SPI_SSR[4]) 必须被设置为 1。</p>
[4]	SS_LTRIG	<p>从机选择电平触发</p> <p>1 = 从机选择信号为电平触发。从机选择信号是低有效还是高有效决定于 SS_LVL 位的值。</p> <p>0 = 从机选择信号为边沿触发。</p>
[3]	AUTOSS	<p>自动从机选择则 (master only)</p> <p>1 = 如果该位被置位，SPISS[1:0] 信号将会自动被产生，这意味着在 SSR[1:0] 中设定的设备/从机选择信号，在开始数据发送/接收时自动被 SPI 控制器触发，在数据传输结束后自动被 SPI 控制器置为无效。</p> <p>0 = 如果该位被设置为 0，从机选择信号通过置位和清除位域 SSR[1:0] 来被置位有效和无效。</p>
[2]	SS_LVL	<p>从机选择有效电平</p> <p>该位定义设备/从机选择信号的有效电平 (SPISS[1:0])</p> <p>1 = SPI_SS 从机选择信号高电平有效。</p> <p>0 = SPI_SS 从机选择信号低电平有效</p>
[1:0]	SSR[1:0]	<p>从机选择有效寄存器 (master only)</p> <p>如果 AUTOSS 位被清除，写 1 到 SSR[0] 位设置 SPISS[0] 引脚输出有效状态，写 0 设置该引脚恢复到无效状态 (SSR[1] 对于 SPISS[1] 引脚的作用也一样)</p> <p>如果 AUTOSS 位被设置，写 1 到该位域的任一位，在数据传输过程中，相应的 SPISS[1:0] 引脚都会被自动的置为有效状态，而在其他时间里被自动置为无效状态。(SPISS[1:0] 的有效电平在 SS_LVL 位定义)。</p> <p>注：</p> <ol style="list-style-type: none"> 该接口在一次给定的时间内只能驱动一个设备/从机，因此，在开始读或写传输之前，被选择的设备的从机选择信号必须被设置为激活状态。 在从机模式下，SPISS[0] 也被定义为设备/从机选择输入，并且从机选择输入必须由边沿触发激活 (至于是上升沿还是下降沿触发激活由 SS_LVL 定义)，否则，SPI 从机控制器将会进入僵死状态，直到有一个边沿激活触发信号，或者软件复位 SPI 控制器。

SPI 接收 FIFO 寄存器 (SPI_RX)

寄存器	偏移量	R/W	描述	复位后的值
SPI_RX0	SPIx_BA + 0x10	R	接收数据 FIFO 寄存器 0	0x0000_0000
SPI_RX1	SPIx_BA + 0x14	R	接收数据 FIFO 寄存器 1	0x0000_0000

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

Bits	描述	
[31:0]	RDATA[31:0]	<p>接收 FIFO 寄存器</p> <p>可以从该寄存器读取接收到的数据。如果 FIFO 位被设置为 1，用户也可以通过检查 RX_EMPTY, SPI_STATUS[0]，来确认是否有另外的数据被接收到。</p> <p>注：该寄存器只读。</p>

SPI 发送数据 FIFO 寄存器 (SPI_TX)

寄存器	偏移量	R/W	描述	复位后的值
SPI_TX0	SPIx_BA + 0x20	W	发送数据 FIFO 寄存器 0	0x0000_0000
SPI_TX1	SPIx_BA + 0x24	W	发送数据 FIFO 寄存器 1	0x0000_0000

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

Bits	描述
[31:0]	<p>发送数据 FIFO 寄存器</p> <p>发送数据缓存寄存器保持将要在下一次数据传输时发送到数据，该寄存器的有效位数取决于 SPI_CTL 寄存器中由 TX_BIT_LEN 位域定义的事务的位长。</p> <p>例如，如果 TX_BIT_LEN 被设置为 0x08, 位域 SPI_TX[7:0] 将会在下次传输时被发送。如果 TX_BIT_LEN 被设置为 0x00, SPI 控制器将会执行一次 32 位数据传输。</p> <p>注：当 SPI 控制器被配置为从机模式，且 FIFO 模式被禁止，如果 SPI 控制器试图向主机发送数据，如见必须在设置 GO_BUSY 位为 1 之前，更新发送数据寄存器。</p>

SPI 可变时钟寄存器 (SPI VARCLK)

寄存器	偏移量	R/W	描述	复位后的值
SPI_VARCLK	SPIx_BA + 0x34	R/W	可变时钟模式标志寄存器	0x007F_FF87

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

Bits	描述	
[31:0]	VARCLK[31:0]	<p>可变时钟模式标志</p> <p>该位域的值用于指示 SPICLK 的模式，如果 VARCLK 的某一位为 0，SPICLK 的输出频率取决于 DIVIDER1 的值；如果 VARCLK 的某一位为 1，SPICLK 的输出频率取决于 DIVIDER2 的值。</p> <p>注：该位域仅在 CLKP = 0 情况下可用</p>

SPI DMA 控制寄存器 (SPI DMA)

寄存器	偏移量	R/W	描述	复位后的值
SPI_DMA	SPIx_BA + 0x38	R/W	SPI DMA 控制寄存器	0x0000_0000

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

Bits	描述	
[31:3]	-	保留。
[2]	PDMA_RST	<p>PDMA 复位</p> <p>该位用于复位 SPI PDMA 功能为默认状态。</p> <p>1 = 复位 PDMA 功能</p> <p>0 = PDMA 功能复位完成，或者 SPI 处于正常操作模式。</p> <p>注：在 PDMA 复位完成之后，该位被自动清零</p>
[1]	RX_DMA_EN	<p>接收 PDMA 使能 (PDMA 读 SPI 数据到内存)</p> <p>设置该位为 1 将会开始接收 PDMA 处理过程。当有数据被写入接收缓存，或者在 FIFO 模式下 RX_EMPTY 的状态被设置为 0 时，SPI 控制器将会自动向 PDMA 控制器触发请求。</p> <p>如果使用 RX_PDMA 去接收数据，但是 TX_DMA 被禁止，GO_BUSY 位必须由用户设定。</p> <p>当 PDMA 传输完成后，硬件自动清除该位为 0</p> <p>在从机模式，且 FIFO 位被禁止的情况下，如果接收 PDMA 被使能，但是发送 PDMA 被禁止，在边沿触发模式，在两个连续事务输入之间的最小休眠间隔需要大于 9 个 SPI 从机引擎时钟 + 4 APB 时钟；而在电平触发模式，该休眠间隔需要大于 9.5 个 SPI 从机引擎时钟 + 4 APB 时钟。</p>
[0]	TX_DMA_EN	<p>发送 PDMA 使能 (PDMA 写数据到 SPI)</p> <p>设置该位为 1 将会开始发送 PDMA 处理过程。SPI 控制器将会自动向 PDMA 控制器触发请求。</p> <p>如果使用 PDMA 模式来传输数据，不要去设定 SPI_CTL 寄存器中 GO_BUSY 位。当需要的时候，SPI 控制器中的 DMA 控制器将会自动的设置 GO_BUSY 位。</p> <p>注：1. 主机模式，在边沿模式下，两个事务需要最小 18 APB 时钟 + 8 SPI 串行时钟的休眠间隔；而在电平模式下，两个事务需要最小 18 APB 时钟 + 9.5 串行时钟。</p> <p>2. 如果两位功能使能，需要的时序应当在上述时钟周期的基础上加上 18 APB 时钟</p>

		<p>钟。</p> <p>当 PDMA 传输完成后，硬件自动清除该位为 0.</p>
--	--	--

SPI FIFO 控制寄存器 (SPI_FFCTL)

寄存器	偏移量	R/W	描述	复位后的值
SPI_FFCTL	SPIx_BA + 0x3C	R/W	SPI FIFO 控制寄存器	0x4400_0000

31	30	29	28	27	26	25	24
TX_THRESHOLD				RX_THRESHOLD			
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
TIMEOUT_EN			RXOVF_INTEN	TXINT_EN	RX_INTEN	TX_CLR	RX_CLR

Bits	描述	
[31]	-	保留。
[30:28]	TX_THRESHOLD	发送 FIFO 阈值 3 位寄存器，取值范围是 0 ~7。 如果 TX 有效数据数量小于或者等于 TXTHRESHOLD, TXINT_STS 将会被设置为 1, 否则 TXINT_STS 将会被设置为 0.
[26:24]	RX_THRESHOLD	接收 FIFO 阈值 3 位寄存器，取值范围是 0 ~7。 如果 RX 有效数据数量大于 RXTHRESHOLD, RXINT_STS 将会被设置为 1, 否则 RXINT_STS 将会被设置为 0.
[23:8]	-	保留。
[7]	TIMEOUT_EN	RX 读超时功能使能 1 = 使能 RX 读超时功能。 0 = 禁用RX 读超时功能
[6:5]	-	保留。
[4]	RXOVINT_EN	RX FIFO 溢出中断使能 1 = 使能 RX FIFO 溢出中断 0 = 禁用 RX FIFO 溢出中断
[3]	TXINT_EN	TX 阈值知道使能 1 = 使能 TX 阈值中断 0 = 禁用 Tx 阈值中断

[2]	RXINT_EN	RX 阈值中断使能 1 = 使能 RX 阈值中断 0 = 禁用 Rx 阈值中断.
[1]	TX_CLR	发送 FIFO 计数器清零 该位用于在 FIFO 模式下，清除发送计数器。可以通过向该位写 1 来清除发送计数器，当发送计数器清除完成之后，该位被自动清零。在清除计数器操作之后，TX_EMPTY 标志 (SPI_STATUS[2]) 将被置 1.
[0]	RX_CLR	接收 FIFO 计数器清零 该位用于在 FIFO 模式下，清除接收计数器。可以通过向该位写 1 来清除接收计数器，当接收计数器清除完成之后，该位被自动清零。在清除计数器操作之后，RX_EMPTY 标志 (SPI_STATUS[0]) 将被置 1.

5.18 定时器控制器

5.18.1 概述

该芯片带有四个定时器模块，TIMER0, TIMER1, TIMER2与TIMER3 (TIMER0/1 在 APB1总线上，TIMER2/3 在 APB2总线上)，用户可以很方便的使用这些定时器执行计数或时间控制机制。定时器模块可执行像频率测量，事件计数，间隔时间测量，时钟产生，延迟时间等功能。定时器可在计时溢出时产生中断信号，也可在操作过程中提供计数的当前值。

5.18.2 特性

- 每个定时器都有独立的时钟源 (TMRx_CLK, x=0,1,2,3)
- 超时周期 = (输入给定时器的时钟周期) * (8 位预分频计数器 + 1) * (24 位 TCMP)
- 计数周期 = $(1 / \text{TMRx_CLK}) * (2^8) * (2^{24})$
- 内部 8 位预分频计数器
- 内部 24 位上数型计数器通过 TDR 可读 (定时器数据寄存器)
- 支持单次、周期、输出翻转和连续计数操作模式
- 支持外部引脚输入捕捉功能用于时间间隔测量
- 支持外部引脚输入捕捉功能用于定时器计数器复位
- 支持定时器间互触发
- 支持内部触发 ADC, DAC 和 PDMA。

5.18.3 框图

每个定时器带一个 8 位预分频计数器，一个 24 位上数型计数器，一个 24 位比较寄存器和一个中断请求信号。参考图 5-107定时器控制框图。每个定时器有 5 个可选的时钟源，图 5-108 说明了时钟源控制功能。

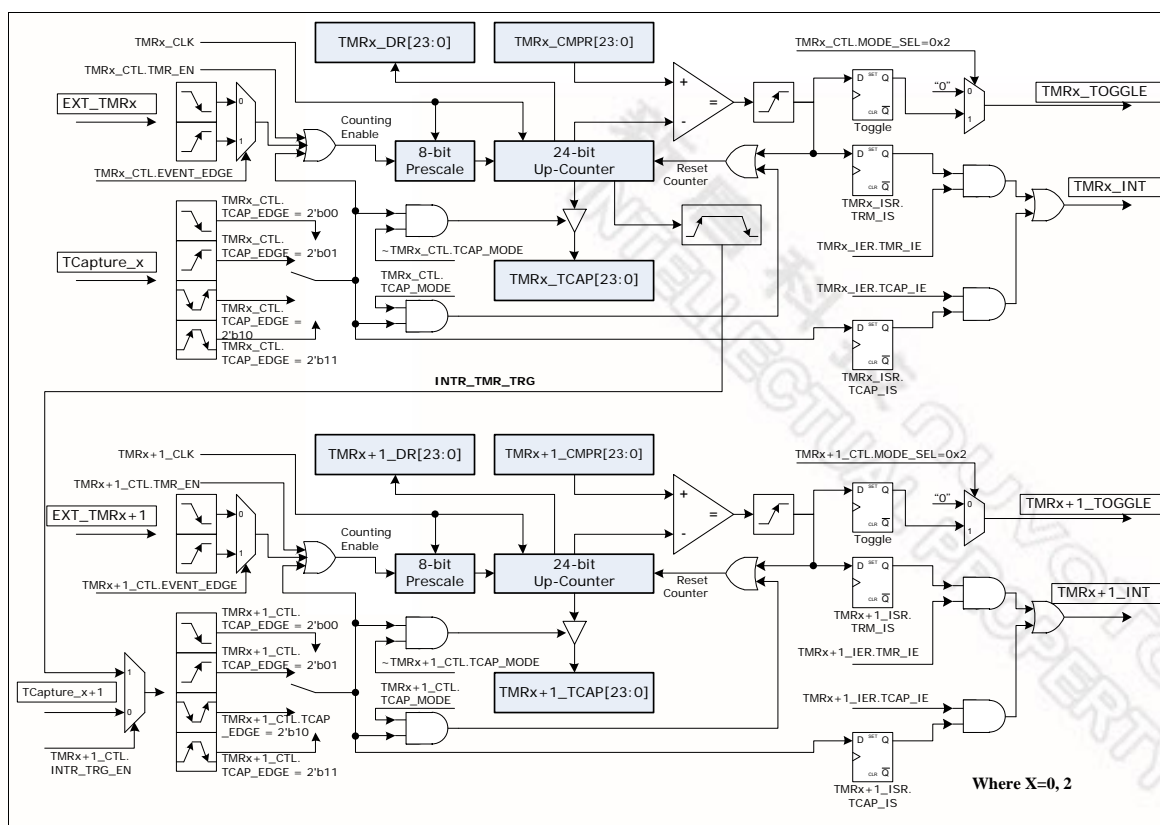


图 5-107 定时器控制器框图

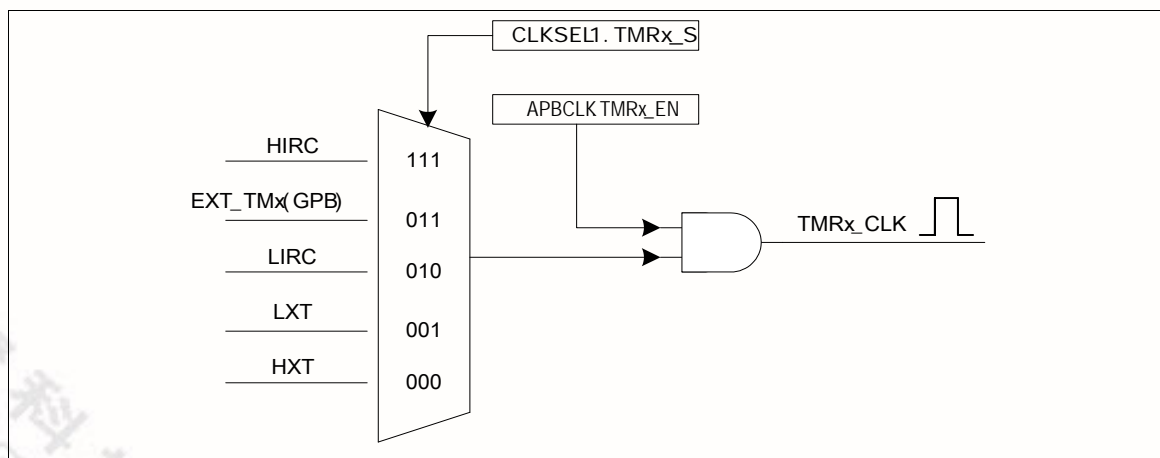


图 5-108 定时器时钟控制器图

5.18.4 功能描述

定时器控制器提供单次，周期，翻转输出和连续计数等操作模式。也提供事件计数功能用于记录外部引脚的事件数，提供外部引脚捕捉功能用于时间间隔测量或者复位定时器计数器。另外，定时器控制器提供定时器间互触发模式用于精确测量输入频率。每个操作功能模式如下：

5.18.4.1 单次模式

如果定时器控制器工作在单次模式 (MODE_SEL[1:0] 为 0x0) 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 1, 定时器控制器开始计数。一旦定时器计数器的值 (TMRx_DR 值) 达到定时

器比较器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1. 如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 1, 且 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 1, 中断信号将会被产生并被发送到 NVIC 来通知 CPU 定时器计数溢出发生。如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 0, 没有中断信号产生。

在这种操作模式下, 一旦定时器计数器的值 (TMRx_DR 的值) 达到定时器比较器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1, 定时器计数操作停止, 定时器计数器的值 (TMRx_DR 的值) 恢复到计数初始值, 然后 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被定时器控制器自动清除为 0. 也即是说, 在编程定时器比较寄存器 (TMRx_CMPR) 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 之后, 定时器计数和与 TMRx_CMPR 的值比较大操作只进行一次。所以, 该操作模式叫做单次模式。

5.18.4.2 周期模式

如果定时器工作在周期模式 (MODE_SEL[1:0] 为 0x1) 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 1, 定时器计数器开始计数。一旦计数值 (TMRx_DR 的值) 达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1. 如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 1, 且 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 1, 中断信号产生并被发送到 NVIC 来通知 CPU 定时器计数溢出发生。如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 0, 无中断信号产生。

在这种操作模式下, 一旦定时器计数器的值 (TMRx_DR 的值) 达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1, 定时器计数器的值 (TMRx_DR 的值) 恢复到计数初值, 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 保持为 1 (继续计数使能), 定时器计数器再一次开始向上计数。如果 TMR_IS (TMRx_ISR[0] 定时器中断状态) 被软件清除, 一旦定时器计数值 (TMRx_DR 的值) 再一次达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被再次设置为 1。也即是说, 定时器计数和与 TMRx_CMPR 的值比较的功能是周期性的。定时器计数操作不会停止, 直到 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 0. 中断信号也周期性的被产生。所以, 该操作模式叫做周期模式。

5.18.4.3 翻转模式

如果定时器工作在翻转模式 (MODE_SEL[1:0] 为 0x2) 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 1, 定时器计数器开始向上计数。一旦定时器计数器的值 (TMRx_DR 的值) 达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1. 如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 1, 且 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 1, 中断信号产生并被发送到 NVIC 来通知 CPU 定时器计数溢出发生。如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 0, 无中断信号产生。

在该操作模式下, 一旦定时器计数器的值 (TMRx_DR 的值) 达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1, 翻转输出信号被设置为 1, 定时器计数器的值 (TMRx_DR 的值) 恢复到计数初值, 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 保持为 1 (继续计数使能), 定时器计数器再一次开始向上计数。如果 TMR_IS (TMRx_ISR[0] 定时器中断状态) 被软件清除为 0, 一旦定时器计数器的值 (TMRx_DR 的值) 再一次达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将再一次被设置为 1, 翻转输出信号被设置为 0. 定时器计数操作不会停止, 直到 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 0. 这样, 翻转输出信号输出 50% 占空比的周期信号。所以, 该操作模式叫做翻转模式。

5.18.4.4 连续计数模式

如果定时器工作在连续计数模式 (MODE_SEL[1:0] 为 0x3) 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 被设置为 1, 定时器计数器开始向上计数。一旦定时器计数器的值 (TMRx_DR 的值) 达到

定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1. 如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 1, 且 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 1, 中断信号产生并被发送到 NVIC 来通知 CPU 定时器计数溢出发生。如果 TMR_IE (TMRx_IER[0] 定时器中断使能位) 被设置为 0, 无中断信号产生。

在该操作模式下, 一旦定时器计数器的值 (TMRx_DR 的值) 达到定时器比较寄存器 (TMRx_CMPR) 的值, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1, 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 保持为 1 (继续计数使能), 定时器计数器继续计数, 而不用重载定时器计数器的值 (TMRx_DR 的值) 到计数初值。用户立即可以改变不同的定时器比较寄存器 (TMRx_CMPR) 的值, 而不用禁止定时器计数器和重启定时器计数器。

例如, 定时器比较寄存器 (TMRx_CMPR) 被设置为 80 (定时器比较寄存器 (TMRx_CMPR) 的值必须大于 1, 且小于 2^{24}). 一旦定时器计数器的值 (TMRx_DR 的值) 达到 80, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会被设置为 1, 且 TMR_EN (TMRx_CTL[0] 定时器计数器使能位) 保持为 1 (继续计数使能), 定时器计数器的值 (TMRx_DR 的值) 将不会恢复到 0, 而是继续计数, 81, 82, 83, ... to ($2^{24} - 1$), 然后再一次 0, 1, 2, 3, ... 到 $2^{24} - 1$, 如此往复。接下来, 如果用户编程定时器比较寄存器 (TMRx_CMPR) 的值为 200, 且 TMR_IS (TMRx_ISR[0] 定时器中断状态) 被清除为 0, 当定时器比较寄存器的值 (TMRx_DR 的值) 达到 200, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会再一次被设置为 1。最后, 用户编程定时器比较寄存器 (TMRx_CMPR) 的值为 500, 并清除 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 0, 当定时器计数器的值 (TMRx_DR 的值) 达到 500, TMR_IS (TMRx_ISR[0] 定时器中断状态) 将会再一次被设置为 1。在这种模式下, 定时器计数器的值 (TMRx_DR 的值) 总是保持持续向上计数, 即便 TMR_IS (TMRx_ISR[0] 定时器中断状态) 为 1。所以, 该操作模式叫做连续计数模式。

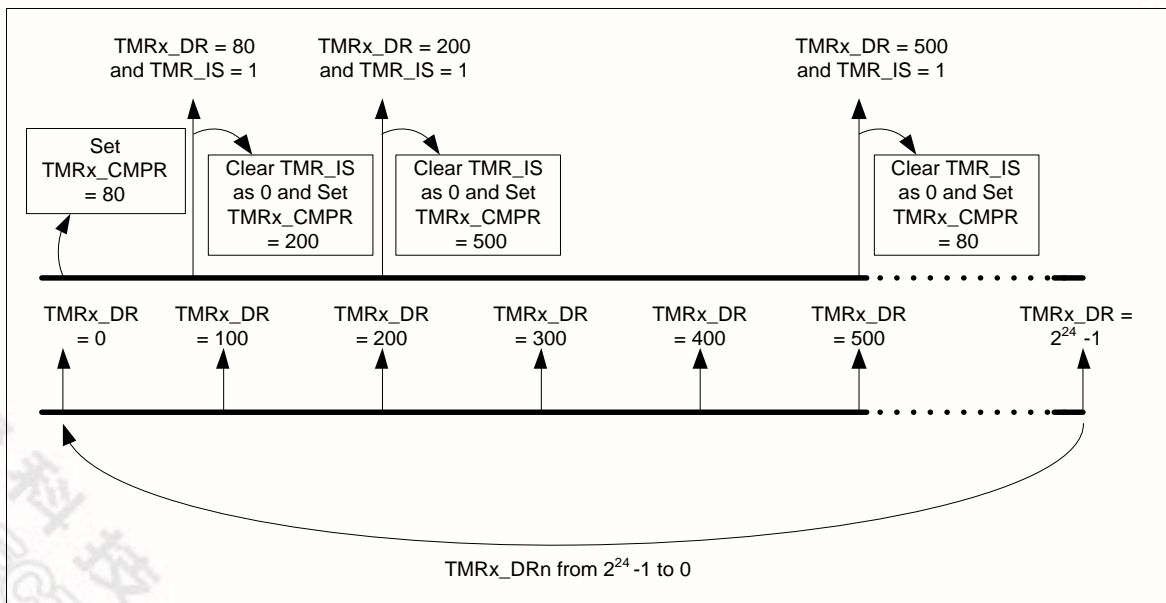


图 5-109 定时器时钟控制器

5.18.4.5 事件计数模式

能计数从外部事件输入引脚输入的事件的应用叫做事件计数功能。在该模式下, 除了 EVENT_EN (TMRx_CTL[12] 事件计数模式使能) 必须被设置为 1 之外, 定时器控制寄存器中大部分的设定值与定时器操作功能模式下的一样。当外部事件输入管脚上的状态改变, 事件计数器的值 (TMRx_DR 的值) 将会根据在 EVENT_EDGE (TMRx_CTL[13] 事件计数模式边沿选择) 中的设定值来进行计数。EVENT_DEB_EN (TMRx_CTL[14] 外部事件去抖使能) 用于使能或禁止外部事件输入管脚的边沿检测去抖电路。如果 EVENT_DEB_EN (TMRx_CTL[14] 外部事件去抖使能) 被设置为 0, 外部事

件输入管脚上事件计数源的最大频率必须小于 $1/4 \text{ TRMx_CLK}$ ；而如果 `EVNT_DEB_EN` (`TMRx_CTL[14]` 外部事件去抖使能) 被设置为 1，外部事件输入管脚上事件计数源的最大频率必须小于 $1/10 \text{ TRMx_CLK}$ 。否则，事件计数值 (`TMRx_DR` 的值) 将不会正常计数。

5.18.4.6 定时器计数器捕捉/复位功能

在该模式下，定时器会监视外部引脚上的电平翻转，来保存 24 位计数器的值或者复位 24 位计数器。

如果 `TCAP_MODE` 为 0，外部引脚上的电平翻转被用于定时器计数器捕捉功能，在这种模式下，如果 `CAP_CNT_MOD` 为 0，定时器计数器工作在自由计数模式，24 位上数型计数器将会保持连续不停的计数，当外部引脚上的电平翻转与在 `TCAP_EDGE` 位中设定的翻转条件相匹配时，24 位上数型计数器的值会被保存到 `TMRx_TCAP` 寄存器。如果 `CAP_CNT_MOD` 为 1，定时器计数器工作在触发计数模式，24 位上数型计数器的值会保持为 0，一旦外部引脚上的电平翻转与 `TCAP_EDGE` 位域中设定的第一个翻转条件相匹配的话，24 位上数型计数器将会开始计数；当外部引脚上再一次电平翻转与在 `TCAP_EDGE` 位域中设定的第二个翻转条件相匹配的话，24 位上数型计数器将停止计数，而计数器的当前值也将被保存到 `TMRx_TCAP` 寄存器中。

如果 `TCAP_MODE` 为 1，外部引脚上的电平翻转被用于复位定时器计数器。在这种模式下，一旦外部引脚上的电平翻转与在 `TCAP_EDGE` 位域中设定的条件相匹配时，24 位上数型计数器将会被复位。

为了检测外部引脚的电平翻转，定时器电路对外部引脚执行去抖操作。基于去抖电路的结果，外部引脚上的上升沿或下降沿可以被检测到。去抖电路的复位值为 0，且仅当 `TCAP_DEB_EN` 和 `TCAP_EN` 都被使能的情况下，去抖功能才是有效的。所以，如果外部引脚电平为 1，且 `TCAP_EDGE` 被设置为检测外部引脚的上升沿，然后设置去抖电路有效 (`TCAP_DEB_EN` 和 `TCAP_EN` 均设置为 1)，一个伪上升沿将会被检测到，这会导致在第一次 `TCAP_IS` 被置位时，捕捉的数据不正确 (`TMRx_TCAP`)，为了避免这种不正确的捕捉数据影响捕捉应用，推荐忽略第一次捕捉到的数据，同时，这也是有必要的。

5.18.4.7 定时器间互触发模式

在该模式下，`TMRx` ($x=0$ 或 2) 将会被强制工作在计数器模式，每有一个外部事件计数一次，并会在特定的时刻产生相应的信号 (`INTR_TMR_TRG`) 触发 `TMRx+1` ($x=0$ 或 2)，而 `TMRx+1` 被强制工作在捕捉功能的触发计数模式。

在 `INTR_TRG_EN` 被设置为 1 的情况下，当 `TMRx` 的计数器从 `0x0` 计数到 `0x1` 时，`TMRx` 将会向 `TMRx+1` 产生一个上升沿触发信号 (`INTR_TMR_TRG`)。当 24 位计数器的值达到 24 位比较寄存器 `TCMPR` 的值时，`TMRx` 将会向 `TMRx+1` 产生一个下降沿触发信号 (`INTR_TMR_TRG`)。

当 `INTR_TMR_TRG` 信号从低到高 (上升沿) 翻转时，`TMRx+1` 的 24 位计数器将开始计数；而当 `INTR_TMR_TRG` 从高到低 (下降沿) 翻转时，`TMRx+1` 的 24 位计数器将停止计数，同时，该计数器的值会被保存到 `TMRx+1_TCAP` 寄存器。

下述示例演示了定时器间互操作是如何工作的。当定时器间互触发模式被使能 (`TMRx_CTL.INTR_RTG_EN = 1`)，`TMRx_Counter` 开始计数，每次在 `TMRx` 引脚上检测到一个外部事件时计数一次。当 `TMRx_Counter` 从 `0x0` 计数到 `0x1` 时，`INTR_TMR_TRG` 被设置为高，`TMRx+1_Counter` 开始计数，每一个 `TMRx+1_CLK` 时钟计数一次。当 `TMRx_Counter` 计数到 `TMRx_CMPR` 时，`TMRx_Counter` 停止计数，`INTR_TMR_TRG` 被设置为低，然后 `TMRx+1_Counter` 也停止计数，同时，`TMRx+1_Counter` 的值被保存到 `TMRx+1_TCAP` 寄存器中，中断状态 `TMRx_ISR.TMR_IS` 被置位，此外，`TMRx_CTL.INTR_TRG_EN` 也会被硬件自动清零。

通过使用定时器间互触发模式，对外部事件 (`TMRx`) 的周期的测量会更加精确。

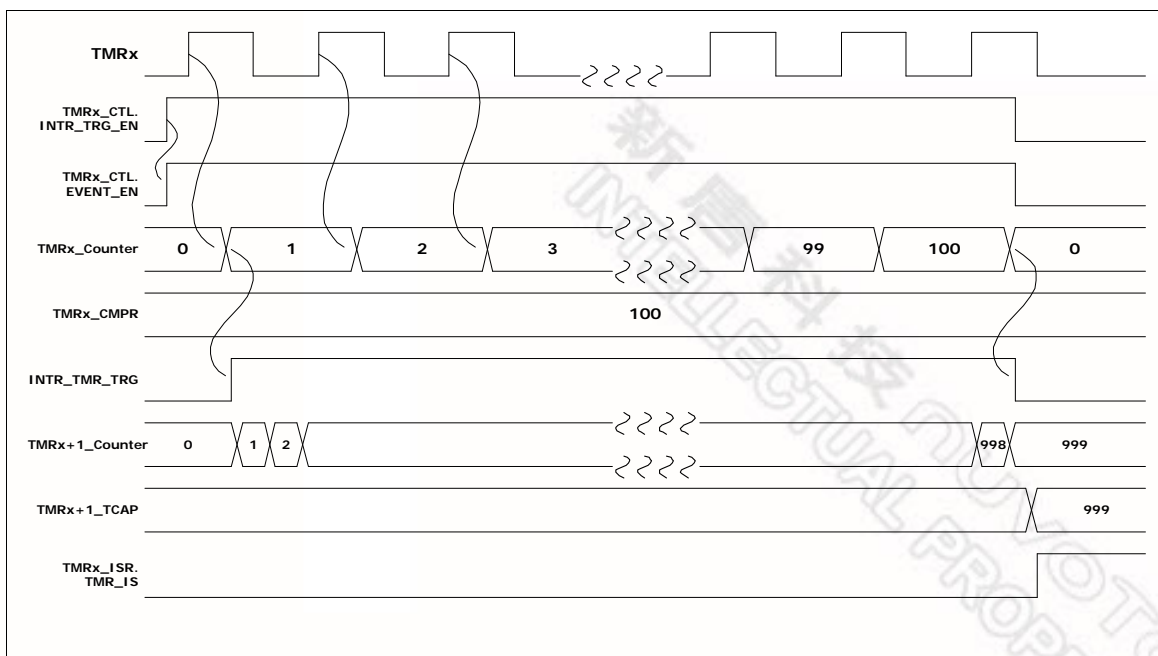


图 5-110 定时器间互触发模式

5.18.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
TIMER Base Address :				
TMR0_BA = 0x4001_0000				
TMR1_BA = 0x4001_0100				
TMR2_BA = 0x4011_0000				
TMR3_BA = 0x4011_0100				
TMRx_CTL x=0,1,2,3	TMRx_BA+0x000	R/W	Timer x 控制寄存器	0x0000_0000
TMRx_PRECNT x=0,1,2,3	TMRx_BA+0x004	R/W	Timer x 预分频器寄存器	0x0000_0000
TMRx_CMPR x=0,1,2,3	TMRx_BA+0x008	R/W	Timer x 比较寄存器	0x0000_0000
TMRx_IER x=0,1,2,3	TMRx_BA+0x00C	R/W	Timer x 中断使能寄存器	0x0000_0000
TMRx_ISR x=0,1,2,3	TMRx_BA+0x010	R/W	Timer x 中断状态寄存器	0x0000_0000
TMRx_DR x=0,1,2,3	TMRx_BA+0x014	R	Timer x 数据寄存器	0x0000_0000
TMRx_TCAP x=0,1,2,3	TMRx_BA+0x018	R	Timer x 捕捉数据寄存器	0x0000_0000

5.18.6 寄存器描述

Timer x 控制寄存器 (TMRx CTL)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_CTL x= 0,1,2,3	TMRx_BA+0x000	R/W	Timer x 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							INTR_TRG_EN
23	22	21	20	19	18	17	16
-	TCAP_DEB_EN	-	CAP_CNT_MOD	TCAP_EDGE		TCAP_MODE	TCAP_EN
15	14	13	12	11	10	9	8
-	EVNT_DEB_EN	EVENT_EDGE	EVENT_EN	CAP_TRG_EN	PDMA_TEEN	DAC_TEEN	ADC_TEEN
7	6	5	4	3	2	1	0
TMR_ACT	-	MODE_SEL		DBGACK_EN	WAKE_EN	SW_RST	TMR_EN

Bits	描述	
[31:25]	-	保留
[24]	INTR_TRG_EN	<p>定时器间互触发模式使能</p> <p>该位控制定时器间互触发模式是否被使能</p> <p>如果定时器间互触发模式被使能，TMRx 将会工作在计数器模式，在外部计数输入引脚 TMRx 上每有一个外部时钟周期，或者一个事件，TMRx+1 的计数器会计数一次。这种模式下，TMRx+1 将会工作在捕捉功能的触发计数模式。</p> <p>0 = 定时器间互触发模式禁止。</p> <p>1 = 定时器间互触发模式使能。</p> <p>注：对于 TMRx+1_CTL，该位被忽略，读取时总是返回 1'b0。</p>
[23]	-	保留
[22]	CAP_DEB_EN	<p>捕捉引脚 TCapture 去抖使能</p> <p>当 CAP_DEB_EN 被置位，捕捉引脚去抖电路将会被使能，用于消除信号抖动。</p> <p>去抖电路会每四个 TMRx_CLK 对捕捉引脚信号采样一次。</p> <p>0 = 禁止去抖电路。</p> <p>1 = 使能去抖电路。</p> <p>注：当 TCAP_EN 被使能，推荐同时使能该位；而当 TCAP_EN 被禁止时，为了减少功率损耗，推荐同时禁止该位。</p>
[21]	-	保留

Bits	描述																					
[20]	CAP_CNT_MOD	<p>Timer 捕捉计数模式选择</p> <p>该位在 TCAP_EN 被置位时，指示 24 位上数型计数器的工作方式。</p> <p>如果该位为 0，计数器工作在自由计数模式，24 位上数型计数器的工作方式在 MODE_SEL 位域中定义。当 TCAP_EN 被置位，TCAP_MODE 为 0 时，若在捕捉引脚上的电平翻转与在 TCAP_EDGE 位域中定义的翻转条件相匹配，24 位上数型计数器的值将会被保存到 TMRx_TCAPn 寄存器。</p> <p>如果该位为 1，计数器工作在触发计数模式，正常情况下，24 位上数型计数器不会计数，并一直保持为 0，当 TCAP_EN 被置位，TCAP_MODE 为 0 时，一旦外部捕捉引脚上的电平翻转与在 TCAP_EDGE 位域中定义的第一个翻转条件相匹配，24 位上数型计数器将开始计数，当在该引脚上再一次出现与在 TCAP_EDGE 位域中定义的第二个翻转条件相匹配的电平翻转时，24 位上数型计数器停止计数，同时，计数器的值将会被保存到 TMRx_TCAPn 寄存器中。</p> <p>0 = 捕捉模式下定时器计数器工作在自由计数模式。 1 = 捕捉模式下定时器计数器工作在触发计数模式</p> <p>注：对于 TMRx+1_CTL，如果 INTR_TRG_EN 被置位，CAP_CNT_MOD 将会被强制为1，捕捉模式下定时器计数器工作在触发计数模式。</p>																				
[19:18]	TCAP_EDGE	<p>TCapture 引脚边沿检测选择</p> <p>该位域用于在定时器工作在捕捉功能或者计数器复位功能时，定义捕捉引脚 TCapture 的有效边沿。</p> <p>对于定时器计数器复位功能和计数器工作在自由计数模式时的定时器捕捉功能，配置如下表：</p> <table><tr><th>TCAP_EDGE</th><th>Output clock (MCLK)</th></tr><tr><td>0x0</td><td>TCapture 引脚上的下降沿 (从 1 到 0 翻转) 是有效边沿。</td></tr><tr><td>0x1</td><td>TCapture 引脚上的上升沿 (从 0 到 1 翻转) 是有效边沿。</td></tr><tr><td>0x2</td><td>TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。</td></tr><tr><td>0x3</td><td>TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。</td></tr></table> <p>对于计数器工作在触发计数模式时的定时器捕捉功能，配置如下表：</p> <table><tr><th>TCAP_EDGE</th><th>Output clock (MCLK)</th></tr><tr><td>0x0</td><td>在TCapture 引脚上的第一个下降沿触发 24 位计数器开始计数，第二个下降沿触发 24 位计数器停止计数。</td></tr><tr><td>0x1</td><td>在TCapture 引脚上的第一个上升沿触发 24 位计数器开始计数，第二个上升沿触发 24 位计数器停止计数</td></tr><tr><td>0x2</td><td>在TCapture 引脚上的下降沿触发 24 位计数器开始计数，上升沿触发 24 位计数器停止计数。</td></tr><tr><td>0x3</td><td>在TCapture 引脚上的上升沿触发 24 位计数器开始计数，下降沿触发 24 位计数器停止计数。</td></tr></table> <p>注：对于 TMRx+1_CTL，如果 INTR_TRG_EN 被置位，TCAP_EDGE 将会被强制为 0b11。</p>	TCAP_EDGE	Output clock (MCLK)	0x0	TCapture 引脚上的下降沿 (从 1 到 0 翻转) 是有效边沿。	0x1	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 是有效边沿。	0x2	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。	0x3	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。	TCAP_EDGE	Output clock (MCLK)	0x0	在TCapture 引脚上的第一个下降沿触发 24 位计数器开始计数，第二个下降沿触发 24 位计数器停止计数。	0x1	在TCapture 引脚上的第一个上升沿触发 24 位计数器开始计数，第二个上升沿触发 24 位计数器停止计数	0x2	在TCapture 引脚上的下降沿触发 24 位计数器开始计数，上升沿触发 24 位计数器停止计数。	0x3	在TCapture 引脚上的上升沿触发 24 位计数器开始计数，下降沿触发 24 位计数器停止计数。
TCAP_EDGE	Output clock (MCLK)																					
0x0	TCapture 引脚上的下降沿 (从 1 到 0 翻转) 是有效边沿。																					
0x1	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 是有效边沿。																					
0x2	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。																					
0x3	TCapture 引脚上的上升沿 (从 0 到 1 翻转) 和下降沿 (从 1 到 0 翻转) 都是有效边沿。																					
TCAP_EDGE	Output clock (MCLK)																					
0x0	在TCapture 引脚上的第一个下降沿触发 24 位计数器开始计数，第二个下降沿触发 24 位计数器停止计数。																					
0x1	在TCapture 引脚上的第一个上升沿触发 24 位计数器开始计数，第二个上升沿触发 24 位计数器停止计数																					
0x2	在TCapture 引脚上的下降沿触发 24 位计数器开始计数，上升沿触发 24 位计数器停止计数。																					
0x3	在TCapture 引脚上的上升沿触发 24 位计数器开始计数，下降沿触发 24 位计数器停止计数。																					

Bits	描述	
[17]	TCAP_MODE	<p>TCapture 引脚功能模式选择</p> <p>该位指示在 TCapture 引脚上的电平翻转用于定时器计数器复位功能还是定时器捕捉功能。</p> <p>0 = 在 TCapture 引脚上的电平翻转用于定时器捕捉功能。</p> <p>1 = 在 TCapture 引脚上的电平翻转用于定时器计数器复位功能。</p> <p>注：对于 TMRx+1_CTL，如果 INTR_TRG_EN 被置位，TCAP_MODE 将会被强制为 0</p>
[16]	TCAP_EN	<p>TCapture 引脚功能使能</p> <p>该位控制在 TCapture 引脚上的电平翻转是否用于定时器计数器复位功能或者定时器捕捉功能</p> <p>0 = 在 TCapture 引脚上的电平翻转被忽略</p> <p>1 = 在 TCapture 引脚上的电平翻转会导致捕捉或 24 位定时器计数器的复位。</p> <p>注：对于 TMRx_CTL，如果 INTR_TRG_EN 被置位，TCAP_EN 将会被强制为低，并且在 TCapture 引脚上的电平翻转被忽略。</p> <p>注：对于 TMRx+1_CTL，如果 INTR_TRG_EN 被置位，TCAP_EN 将会被强制为高</p>
[15]	-	保留
[14]	EVNT_DEB_EN	<p>外部事件去抖使能</p> <p>当 EVNT_DEB_EN 被置位，外部事件引脚去抖电路将会被使能，用于消除信号抖动。去抖电路会每 4 个 TMRx_CLK 对外部事件引脚采样一次。</p> <p>0 = 禁止去抖电路</p> <p>1 = 使能去抖电路。</p> <p>注：当 EVENT_EN 被使能，推荐同时使能该位；当 EVENT_EN 被禁止，为了减少功率损耗，推荐同时禁止该位。</p>
[13]	EVENT_EDGE	<p>事件计数模式边沿选择</p> <p>该位指示在外部事件引脚上的上升沿还是下降沿使能定时器计数器的值加 1。</p> <p>0 = 在外部事件引脚上的下降沿使能定时器计数器的值加 1。</p> <p>1 = 在外部事件引脚上的上升沿使能定时器计数器的值加 1。</p>
[12]	EVENT_EN	<p>事件计数模式使能</p> <p>当 EVENT_EN 被置位时，24 位上数型计数器的值的增加由外部事件引脚控制。</p> <p>当外部事件引脚上的电平翻转与在 EVENT_EDGE 位域中定义的翻转条件相匹配时，24 位上数型计数器的值加 1，否则，24 位上数型计数器的值保持不变。</p> <p>0 = 定时器计数器的计数值不由外部事件引脚控制。</p> <p>1 = 定时器计数器的计数值由外部事件引脚控制。</p> <p>注：当 EVENT_EN 被使能，不可选择 EXT_TMx(GPB) 作为时钟源，并且被选择的时钟源速度须高于 3 倍的 EXT_TMx 的时钟速度。</p>

Bits	描述	
[11]	CAP_TRG_EN	<p>TCAP_IS 触发模式使能</p> <p>该位用于当 TMR_IS 或者 TCAP_IS 被置位时，控制是TMR_IS还是 TCAP_IS 用于触发 PDMA, DAC 和 ADC</p> <p>如果该位被置为 0，且 TMR_IS 被置位，而相关的触发使能位 (PDMA_TEEN, DAC_TEEN 或者 ADC_TEEN) 也被使能，定时器将产生一个内部触发事件到 PDMA, DAC 或者 ADC.</p> <p>如果该位被置为 1，且 TCAP_IS 被置位，而相关的触发使能位 (PDMA_TEEN, DAC_TEEN 或者 ADC_TEEN) 也被使能，定时器将产生一个内部触发事件到 PDMA, DAC 或者 ADC.</p> <p>0 = TMR_IS 被用于触发 PDMA, DAC 和 ADC.</p> <p>1 = TCAP_IS 被用于触发 PDMA, DAC 和 ADC.</p>
[10]	PDMA_TEEN	<p>TMR_IS 或者 TCAP_IS 触发 PDMA 使能</p> <p>该位控制 TMR_IS 或者 TCAP_IS 是否可以用于触发 PDMA.</p> <p>当 PDMA_TEEN 被置位，TMR_IS 被置位，且 CAP_TRG_EN 为 0 时，定时器控制器将会产生一个内部触发事件到 PDMA 控制器.</p> <p>当 PDMA_TEEN 被置位，TCAP_IS 被置位，且 CAP_TRG_EN 为 1 时，定时器控制器将会产生一个内部触发事件到 PDMA 控制器.</p> <p>0 = TMR_IS 或者 TCAP_IS 触发 PDMA 禁止.</p> <p>1 = TMR_IS 或者 TCAP_IS 触发 PDMA 使能.</p>
[9]	DAC_TEEN	<p>TMR_IS 或者 TCAP_IS 触发 DAC 使能</p> <p>该位控制 TMR_IS 或者 TCAP_IS 是否可以用于触发 DAC.</p> <p>当 DAC_TEEN 被置位，TMR_IS 被置位，且 CAP_TRG_EN 为 0 时，定时器控制器将会产生一个内部触发事件到 DAC 控制器.</p> <p>当 DAC_TEEN 被置位，TCAP_IS 被置位，且 CAP_TRG_EN 为 1 时，定时器控制器将会产生一个内部触发事件到 DAC 控制器.</p> <p>0 = TMR_IS 或者 TCAP_IS 触发 DAC 禁止.</p> <p>1 = TMR_IS 或者 TCAP_IS 触发 DAC 使能.</p>
[8]	ADC_TEEN	<p>TMR_IS 或者 TCAP_IS 触发 ADC 使能</p> <p>该位控制 TMR_IS 或者 TCAP_IS 是否可以用于触发 ADC.</p> <p>当 ADC_TEEN 被置位，TMR_IS 被置位，且 CAP_TRG_EN 为 0 时，定时器控制器将会产生一个内部触发事件到 ADC 控制器.</p> <p>当 ADC_TEEN 被置位，TCAP_IS 被置位，且 CAP_TRG_EN 为 1 时，定时器控制器将会产生一个内部触发事件到 ADC 控制器.</p> <p>0 = TMR_IS 或者 TCAP_IS 触发 DAC 禁止.</p> <p>1 = TMR_IS 或者 TCAP_IS 触发 DAC 使能.</p>
[7]	TMR_ACT	<p>Timer 激活状态位 (只读)</p> <p>该位指示定时器计数器的状态.</p> <p>0 = Timer 处于非激活状态.</p> <p>1 = Timer 处于激活状态.</p>
[6]	-	保留

Bits	描述											
[5:4]	MODE_SEL	Timer 操作模式选择										
		<table><tr><th>ODE_SEL</th><th>Output clock (MCLK)</th></tr><tr><td>0x0</td><td>Timer 工作在单次模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会被产生, 同时 TMR_CTL [TMR_EN] 将会由硬件自动清除。</td></tr><tr><td>0x1</td><td>Timer 工作在周期模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。</td></tr><tr><td>0x2</td><td>Timer 工作在周期模式且带输出翻转功能. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。 同时, 定时器控制器切换输出引脚 TMRx_TOG_OUT 到相反的电平 (从低到高或者从高到低) 注: 在复位后 TMRx_TOG_OUT 默认为低电平</td></tr><tr><td>0x3</td><td>Timer 工作在連續計數模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 當 TMR_DR = TMR_CMPR, 相关的中断将会被产生, 然而 24 位上数型计数器繼續計數而不重置。</td></tr></table>	ODE_SEL	Output clock (MCLK)	0x0	Timer 工作在单次模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会被产生, 同时 TMR_CTL [TMR_EN] 将会由硬件自动清除。	0x1	Timer 工作在周期模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。	0x2	Timer 工作在周期模式且带输出翻转功能. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。 同时, 定时器控制器切换输出引脚 TMRx_TOG_OUT 到相反的电平 (从低到高或者从高到低) 注: 在复位后 TMRx_TOG_OUT 默认为低电平	0x3	Timer 工作在連續計數模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 當 TMR_DR = TMR_CMPR, 相关的中断将会被产生, 然而 24 位上数型计数器繼續計數而不重置。
		ODE_SEL	Output clock (MCLK)									
		0x0	Timer 工作在单次模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会被产生, 同时 TMR_CTL [TMR_EN] 将会由硬件自动清除。									
		0x1	Timer 工作在周期模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。									
0x2	Timer 工作在周期模式且带输出翻转功能. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 每次当 24 位上数型计数器的值达到 TMRx_CMPR 的值时, 相关的中断将会周期性的被产生, 然后, 24 位计数器将会被复位, 并从 0 重新开始计数。 同时, 定时器控制器切换输出引脚 TMRx_TOG_OUT 到相反的电平 (从低到高或者从高到低) 注: 在复位后 TMRx_TOG_OUT 默认为低电平											
0x3	Timer 工作在連續計數模式. 在该模式下, 如果 TMRx_IER[TMR_IE] 位被使能, 當 TMR_DR = TMR_CMPR, 相关的中断将会被产生, 然而 24 位上数型计数器繼續計數而不重置。											
[3]	DBGACK_EN	ICE 调试模式响应无效使能 0 = ICE 调试模式响应影响 TIMER 计数, 当 ICE 调试模式响应时, TIMER 计数器将会保持当前值, 而停止计数. 1 = ICE 调试模式响应不影响 TIMER 计数, 不管 ICE 调试模式是否响应, TIMER 计数器都将会正常计数.										
[2]	WAKE_EN	唤醒使能 当 WAKE_EN 被置位, 且 TMR_IS 或者 TCAP_IS 也被置位, 定时器控制器将会向 CPU 产生一个唤醒触发事件. 0 = 唤醒触发事件禁止. 1 = 唤醒触发事件使能.										
[1]	SW_RST	软件复位 设置该位将会复位定时器计数器和预分频器, 并强制 TMR_CTL[TMR_EN] 为 0. 0 = 没有影响 1 = 复位定时器的预分频器, 内部 24 位上数型计数器和 TMR_CTL [TMR_EN] 位. 注: 该位将持续至少 3 个 TMRx_CLK 时钟周期, 并且会自动清零										

Bits	描述	
[0]	TMR_EN	<p>Timer 计数器使能位</p> <p>0 = 停止/挂起计数</p> <p>1 = 开始计数</p> <p>注 1: 设置 TMR_EN 为 1 使能 24 位计数器继续从上一次计数停止时的计数器的值开始计数。</p> <p>注 2: 在定时器单次模式 (MODE_SEL [5:4] =2'b00) 下, 一旦上数型计数器的值达到 TMRx_CMPR, 该位由硬件自动清零.</p>

Timer x 预分频寄存器 (TMRx PRECNT)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_PRECNT x=0,1,2,3	TMRx_BA+0x004	R/W	Timer x 预分频寄存器	0x0000_0000

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

Bits	描述	
[31:8]	-	保留
[7:0]	PRESCALE_CNT	预分频计数器 在输入给计数器之前, 时钟输入被 PRESCALE_CNT + 1 除。如果 PRESCALE_CNT = 0, 则时钟输入不分频。

Timer x 比较寄存器 (TMRx_CMPR)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_CMPR x=0,1,2,3	TMRx_BA+0x008	R/W	Timer x 比较寄存器	0x0000_0000

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

Bits	描述	
[31:24]	-	保留
[23:0]	TMR_CMP	<p>Timer 比较值</p> <p>TMR_CMP 是一个 24 位比较寄存器。当内部 24 位上数型计数器计数达到 TMR_CMP 的值时，如果定时器中断被使能(TMR_IER[TMR_IE])，一个定时器中断将会被请求。TMR_CMP 的值定义定时器计数周期时间。</p> <p>Time out period = (Period of timer clock input) * (8-bit PRESCALE_CNT + 1) * (24-bit TMR_CMP)</p> <p>注 1：绝对不能写 0x0 或 0x1 到 TMR_CMP 寄存器，否则 CPU 会进入一个未知的状态。</p> <p>注 2：不管 TMR_CTL[TMR_EN] 是 0 还是 1，任何时刻当软件写一个新值到该寄存器，TIMER 计数器将丢弃之前的计数值，重新开始计数。</p>

Timer x 中断使能寄存器 (TMRx_IER)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_IER x=0,1,2,3	TMRx_BA+00C	R/W	Timer x 通道 0 中断使能寄存器	0x0000_0000

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

Bits	描述	
[31:2]	-	保留
[1]	TCAP_IE	Timer 捕捉功能中断使能 0 = 禁止定时器外部引脚捕捉功能中断 1 = 使能定时器外部引脚捕捉功能中断 注：如果定时器外部引脚捕捉功能被使能，当 TCAP_EN 被置位，且外部引脚上电平翻转与在 TCAP_EDGE 位域中设定的翻转条件相匹配时，定时器会触发一个中断信号。
[0]	TMR_IE	Timer 中断使能 0 = 禁止定时器中断。 1 = 使能定时器中断。 注：如果定时器中断被使能，当相关的计数器的值等于 TMR_CMPR 的值时，定时器会触发一个中断。

Timer x 中断状态寄存器 (TMRx_ISR)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_ISR X=0,1,2,3	TMRx_BA+0x010	R/W	Timer x 中断状态寄存器	0x0000_0000

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

Bits	描述	
[31:6]	-	保留
[5]	NCAP_DET_STS	<p>新捕获检测状态</p> <p>该状态指示在 CPU 清除 TCAP_IS 状态之前，有一个新的捕捉事件被检测到。</p> <p>如果如上条件发生，定时器会保持寄存器 TMRx_CAPn 的值不变，而把新的捕获值丢掉。</p> <p>当 TCAP_IS 被清除时，该位也会被清除。</p> <p>0 = 在 CPU 清除 TCAP_IS 状态之前，没有检测到新的捕捉事件</p> <p>1 = 在 CPU 清除 TCAP_IS 状态之前，检测到了新的捕捉事件</p>
[4]	TMR_Wake_STS	<p>Timer 唤醒状态</p> <p>如果定时器从掉电模式把 CPU 唤醒，该位将会被设置为 1。必须由软件向该位写 1 来清除该状态。</p> <p>0 = 定时器没有唤醒系统</p> <p>1 = 定时器定时时间到事件把系统从掉电模式唤醒。</p>
[3:2]	-	保留
[1]	TCAP_IS	<p>Timer 捕捉功能中断状态</p> <p>该位指示定时器外部引脚捕捉功能的中断状态</p> <p>当 TCAP_EN 被设置为 1，且外部引脚上有与在 TCAP_EDGE 位域中设定的翻转条件相匹配的电平翻转时，该位被置位。写 1 到该位可以清除该标志。</p> <p>如果该位置位，且 TCAP_IE 被使能，定时器会向 CPU 触发一个中断。</p>

Bits	描述	
[0]	TMR_IS	<p>Timer 中断状态</p> <p>该位指示定时器的中断状态</p> <p>当内部 24 位上数型计数器的值满足设定的定时器比较值 (TMR_CMPR) 时，该位将会被置位。写 1 到该位可以清除该标志。</p> <p>如果该位置位，且 TMR_IE 被使能，定时器会向 CPU 触发一个中断。</p>

Timer x 数据寄存器 (TMRx_DR)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_DR x=0,1,2,3	TMRx_BA+0x014	R	Timer x 数据寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
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							

Bits	描述	
[31:24]	-	保留
[23:0]	TDR	Timer 数据寄存器 用户可以通过读该寄存器来获得内部 24 位上数型计数器的值

Timer x 捕捉数据寄存器 (TMRx TCAP)

寄存器	偏移量	R/W	描述	复位后的值
TMRx_TCAP x=0,1,2,3	TMRx_BA+0x018	R	Timer x 捕捉数据寄存器	0x0000_0000

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

Bits	描述	
[31:24]	-	保留
[23:0]	CAP	Timer 捕捉数据寄存器 当 TCAP_EN 被置位, TCAP_MODE 为 0, 且外部捕捉引脚上的电平翻转与在 TCAP_EDGE 位域中设定的翻转条件相匹配时, 24 位上数型计数器的值会被保存到 TMRx_TCAP 寄存器。用户可以读该寄存器来获取捕捉到的值

GPIO 端口 x 引脚值影子寄存器 (GPx_SHADOW)

寄存器	偏移量	R/W	描述	复位后的值
GPA_SHADOW	TMR0_BA+0x200	R	GPIO 端口 A 引脚值影子寄存器	0x0000_XXXX
GPB_SHADOW	TMR0_BA+0x 204	R	GPIO 端口 B 引脚值影子寄存器	0x0000_XXXX
GPC_SHADOW	TMR0_BA+0x 208	R	GPIO 端口 C 引脚值影子寄存器	0x0000_XXXX
GPD_SHADOW	TMR0_BA+0x 20C	R	GPIO 端口 D 引脚值影子寄存器	0x0000_XXXX
GPE_SHADOW	TMR0_BA+0x 210	R	GPIO 端口 E 引脚值影子寄存器	0x0000_XXXX
GPF_SHADOW	TMR0_BA+0x 204	R	GPIO 端口 F 引脚值影子寄存器	0x0000_XXXX

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

Bits	描述	
[31:16]	-	保留
[n]	PIN[n]	<p>GPIO 端口 [A/B/C/D/E/F] 引脚值</p> <p>从该寄存器读取到的每一位的值反映了相应 GPIO 的实际状态。</p> <p>这些寄存器是 GPIOx_PIN 寄存器的影子寄存器</p> <p>注：对于 GPF_SHADOW，位 [15:9] 是保留位。</p>

5.19 UART 控制器

5.19.1 概述

UART 控制器提供多达两个通道的通用异步收发器模块，UART0 和 UART1。UART0 在 APB1 总线上，而 UART1 在 APB2总线上。

通用异步收发器 (UART) 在从外设接收数据时执行串行到并行的转换，从 CPU 发送数据时执行并行到串行的转换。该 UART 控制器同时支持 IrDA (SIR) 功能，LIN 主机/从机功能和 RS-485 功能模式。每个 UART 通道支持 9 种类型的中断，包括接收阈值到达中断 (INT_RDA)，发送 FIFO 空中断 (INT_THRE)，线状态中断 (break 错误，校验错误，格式错误或者 RS-485 中断) (INT_RLS)，超时中断 (INT_TOUT)，MODEM 状态中断 (INT_MODEM)，缓存错误中断 (INT_BUF_ERR)，唤醒中断 (INT_WAKE)，自动波特率检测或自动波特率计数器溢出标志中断

(INT_ABAUD) 和 LIN 功能中断 (INT_LIN)。

UART0 和 UART1 中均内嵌一个 16 位发送 FIFO (TX_FIFO) 和一个 16 位接收 FIFO (RX_FIFO) 来降低向 CPU 申请中断的次数。在操作过程中 CPU 可以随时读 UART 的状态。报告的状态信息包括已经被 UART 执行的传输操作的类型和条件, 也包括当接收数据可能发生的 3 种错误条件 (校验错误, 格式错误和 break 中断)。UART 控制器支持自动波特率检测, 自动波特率检测控制为波特率发生器所进行的对传入的时钟/数据速率的测量进程, 并可按照用户的意愿被读写。UART 控制器也支持数据传入或 CTSn 唤醒功能, 当系统处于掉电模式时, 一个传入的数据或者 CTSn 信号将会把 CPU 从掉电模式唤醒。UART 包括一个可编程的波特率发生器, 它可以将输入晶振除以一个除数来得到收发器需要的串行时钟。波特率公式为波特率 = $UART_CLK / [BRD + 1]$, 其中 BRD 在波特率分频寄存器 (UARTx_BAUD) 中定义。下表列举了不同条件下的等式和 UART 波特率设置表。

DIV_16_EN	BRD	Baud rate equation
Disable (Mode 0)	A	$UART_CLK / (A+1)$, A must >8
Enable (Mode 1)	A	$UART_CLK / [16 * (A+1)]$

表 5-13 UART 波特率等式

System clock =12 MHz		
Baud rate	Mode 0	Mode 1
921600	A=12	Not Supported
460800	A=25	Not Supported
230400	A=51	A=2
115200	A=103	A=6
57600	A=207	A=12
38400	A=311	A=19
19200	A=624	A=38
9600	A=1249	A=77
4800	A=2499	A=155

表 5-14 UART 波特率设置

5.19.1.1 自动流控:

UART0 和 UART1 控制器用 2 种低电平信号, CTSn (clear-to-send) 和 RTSn (request-to-send), 来支持自动流控制功能, 用来控制 UART 和外部设备 (如: Modem) 之间的数据流传输。当自动流控被使能时, UART 将不允许接收数据直到 UART 向外部设备置 RTSn (RTSn 为高) 为有效, 当 RX FIFO 内的字节数等于 $UART_TLCTL[RTS_TRI_LEV]$ 的值时, RTSn 信号被置为无效。当 UART 控制器从外部设备检测到 CTSn 有效信号 (CTSn 为高) 时, UART 控制器向外发送数据。如果有有效的 CTSn 信号未被检测到, UART 控制器将不会向外发送数据。

5.19.1.2 自动波特率检测:

UART0 和 UART1 控制器支持自动波特率检测。自动波特率检测功能可用于测量接收器传入数据的波特率。如果自动波特率特性被使能, UART 控制器将会测量接收到的数据流的位时间, 并设置波特率除数寄存器 UART_BAUD。。自动波特率检测可通过设置 UART_CTL[ABAUD_EN] 位域来开始。

5.19.1.3 UART 唤醒功能:

UART0 和 UART1 控制器支持唤醒系统功能。唤醒功能包括 CTSn 唤醒功能 (UART_CTL[WAKE_CTS_EN]) 和数据唤醒功能 (UART_CTL[WAKE_DATA_EN])。当系统处于掉电模式时, UART 可以通过 CTSn 引脚和传入数据来唤醒系统。

5.19.1.4 IrDA 功能模式:

UART 控制器提供串行 IrDA (SIR, 串行红外) 功能 (用户必须设置 UA_FUN_SEL 来选择 IrDA 功能)。SIR 定义一种短距离红外异步串行传输模式, 该模式有 1 个起始位, 8 个数据位和 1 个停止位。最大数据速率为 115.2 Kbps (半双工)。IrDA SIR 模块包括 IrDA SIR 协议编码/解码器。IrDA SIR 协议是一个半双工协议, 所以不能同时发送和接收数据。IrDA SIR 物理层规定在发送和接收之间至少有 10ms 传输延时, 当 UART 控制器工作在 IrDA 模式时, UART_BAUD 设定值必须被设定为模式 1 (UART_BAUD[DIV_16_EN] = 1)。

5.19.1.5 RS-485 功能模式:

UART 控制器另一个可选的功能是 RS-485 9位模式功能, 该模式下的方向控制可以由 RTSn 引脚或 GPIO 控制。RS-485 功能模式可通过设定 UART_FUN_SEL 寄存器来选择。RS-485 驱动器控制可以通过使用来自一个异步串行口的控制信号来实现。在 RS-485 模式下, RX 和 TX 的很多特性跟在 UART 模式下一样。

5.19.1.6 LIN 功能模式:

LIN 模式可以通过设定 UART_FUN_SEL 寄存器中的 LIN_EN 位来选择。在 LIN 模式下, 依照 LIN 标准, 要求数据格式为 1 个起始位, 8 个数据位和 1 个停止位。

5.19.2 特征

- 全双工, 异步通信
- 独立的接收/发送 16 字节 FIFO, 用于数据装载。
- 支持硬件自动流控制/流控制功能 (CTS_n, RTS_n) 和可编程的 (CTS_n, RTS_n) 流控制触发电平
- 对于每个通道, 支持可编程的波特率发生器
- 支持自动波特率检测功能
- 支持可编程的接收缓存触发极限值
- 支持传入数据和 CTS_n 唤醒功能
- 支持 9 位接收缓存定时溢出检测功能
- 所有 UART 通道都可以使用 PDMA 控制器
- 通过设置寄存器 UART_TMCTL[DLY] 可编程设定在上一次数据传输的停止位与下一次数据传输的开始位之间发送数据的延迟时间
- 支持 break error, frame error, parity error 和接收/发送缓存溢出检测功能。
- 完全可编程的串口特性

- 可编程为 5-, 6-, 7-, 8-位的数据位
- 可编程的校验位, even, odd, no parity 或 stick parity bit 产生和侦测
- 可编程为 1, 1.5, 或 2 位的停止位
- 支持 IrDA SIR 功能模式
 - 支持 3/16 位周期调制
- 支持 LIN 功能模式
 - 支持 LIN 主机/从机模式
 - 发送器支持可编程的 break 产生功能
 - 接收器支持 break 侦测功能
- 支持 RS-485 功能模式
 - 支持 RS-485 9 位模式
 - 支持硬件或软件控制 RTSn 或者软件控制 GPIO 来控制传输方向

5.19.3 框图

UART 时钟控制和框图见图 5-112 和图 5-113。UART 控制器是完全异步设计，带有两个时钟域，PCLK 和 UART 时钟，注意 PCLK 的频率值应当大于或等于 UART 时钟的频率值。

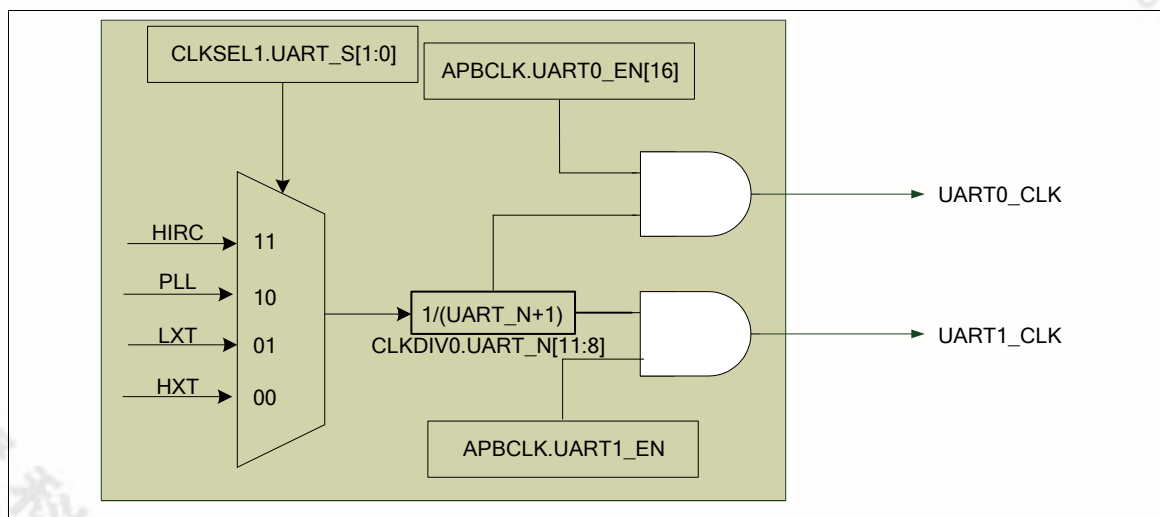


图 5-111 UART 时钟控制图

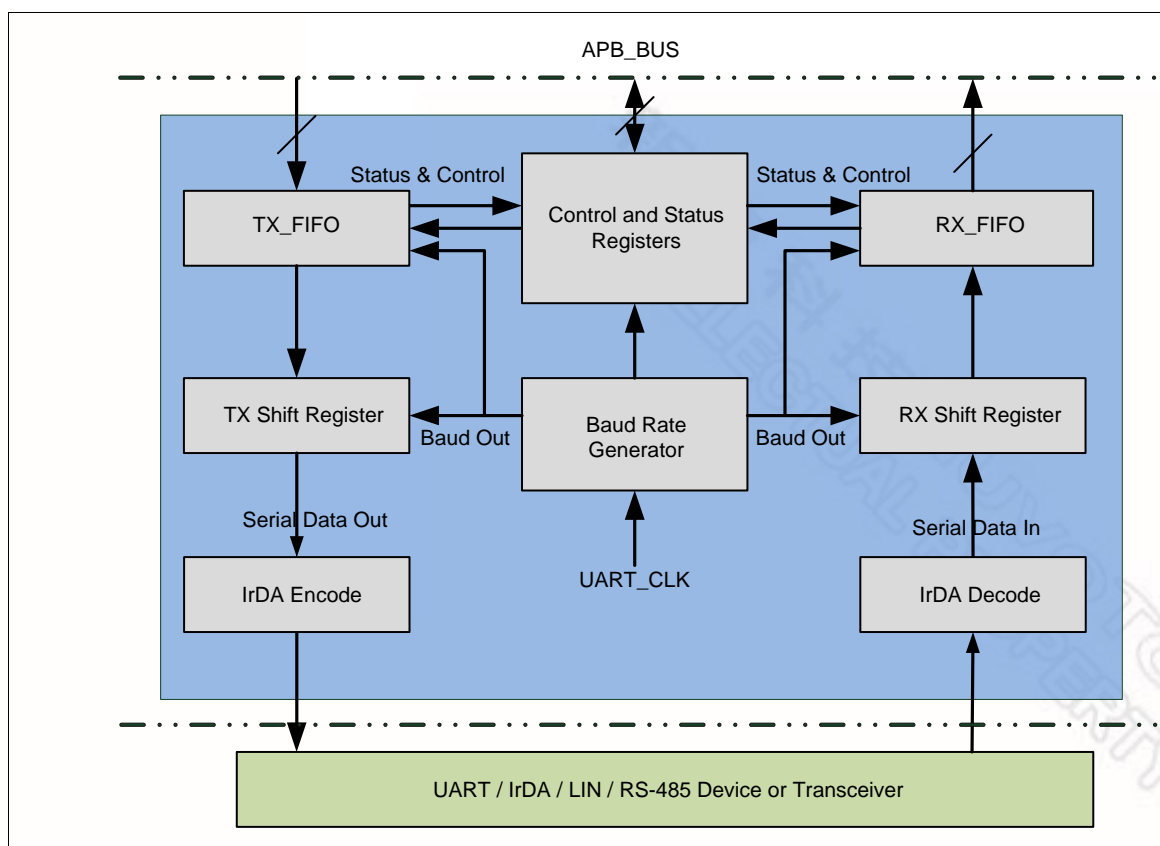


图 5-112 UART 框图

TX_FIFO

发送器用一个 16 字节的 FIFO 缓存来减少向 CPU 申请中断的次数。

RX_FIFO

接收器用一个 16 字节的 FIFO (每个字节加 3 个比特的纠错位) 缓存来减少向 CPU 申请中断的次数

TX 移位寄存器

移位发送数据串行输出的控制模块

RX 移位寄存器

将接收到的数据串行移入的控制模块

波特率发生器

将外部时钟或内部时钟除以一个除数来产生要求的波特率时钟。参考波特率公式。

IrDA 编码

IrDA 编码控制模块。

IrDA 解码

IrDA 解码控制模块

控制和状态寄存器

这是一个应用于发生器和接收器的寄存器组，包括传输线控制寄存器 (UART_TLCTL)，传输状态寄存器 (UART_TRSR) 和控制寄存器 (UART_CTL)。超时控制寄存器 (UART_TMCTL) 标识超时中断发生的条件。该寄存器组也包括用于使能和禁止相应中断，并标识相应中断发生与否的中断使能寄存器 (UART_IER) 和中断状态寄存器 (UART_ISR)，一共有 9 种类型的中断，包括接收阈值到达中断 (INT_RDA)，发送 FIFO 空中断 (INT_THRE)，线状态中断 (break 错误，校验错误，格式错误

或者 RS-485 中断) (INT_RLS)，超时中断 (INT_TOUT)，MODEM 状态中断 (INT_MODEM)，缓存错误中断(INT_BUF_ERR)，唤醒中断 (INT_WAKE)，自动波特率检测或自动波特率计数器溢出标志中断 (INT_ABAUD) 和 LIN 功能中断 (INT_LIN)。

5.19.4 功能描述

5.19.4.1 自动流控:

下图为自动流控框图: .

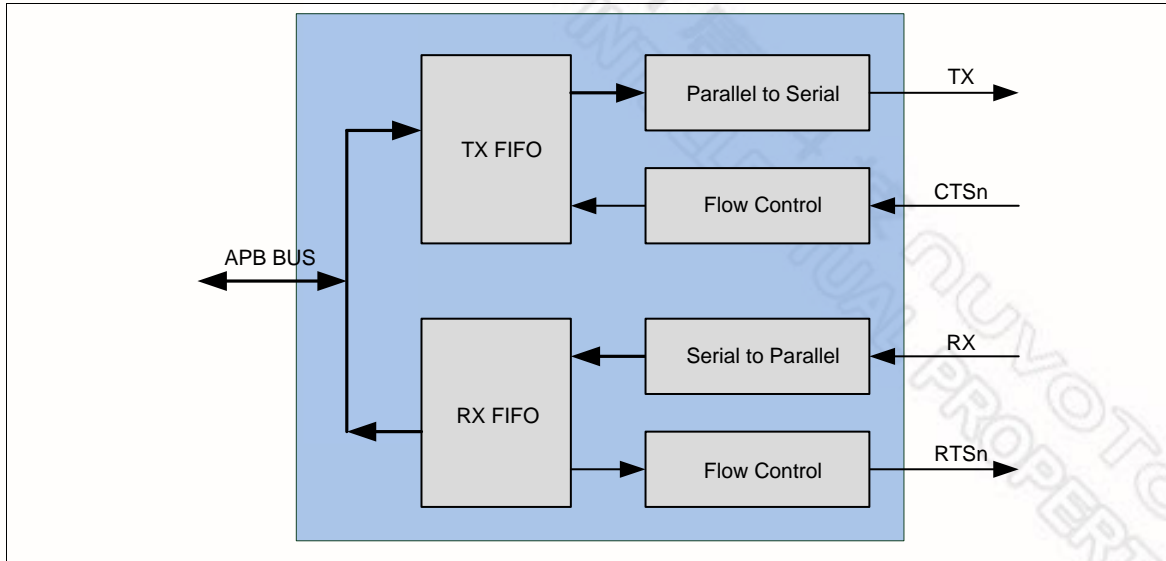


图 5-113 UART 自动流控框图

5.19.4.2 自动波特率检测:

UART 支持自动波特率检测。如果自动波特率检测特性使能，控制器将测量接收到的数据流 (LSB 必须是 1) 的位时间，并设置除数锁存寄存器 UART_BAUD。自动波特率检测通过设定 UART_CTL[ABAUD_EN] 开始工作。当自动波特率检测流程结束时，ABAUD_EN 位将会被自动清除，且 UART_ISR[ABAUD_IS] 位和 UART_ISR[ABAUD_F]位 也将会被置位。如果发生了超时 (波特率计数器溢出)，UART_ISR[ABAUD_IS] 位和 UART_TRSR[ABAUD_TOUT_F] 位将会被置位。下图为自动波特率检测功能框图: .

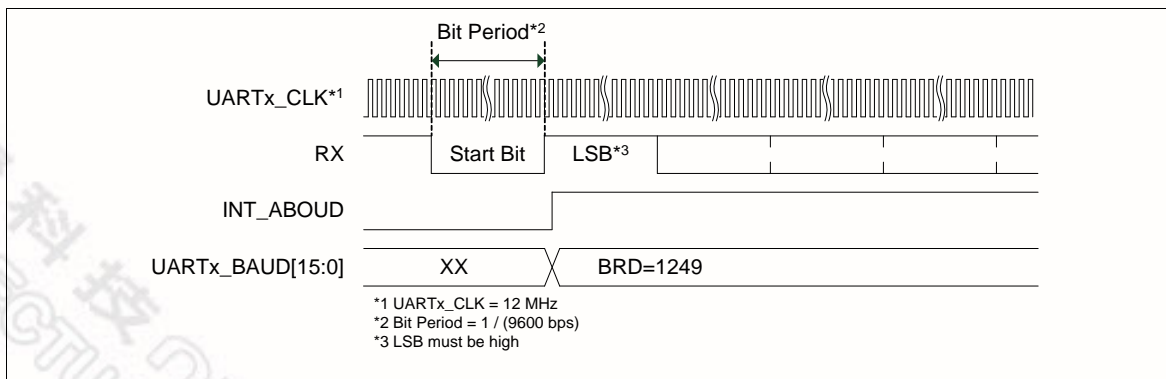


图 5-114 UART 自动波特率框图

5.19.4.3 唤醒功能:

UART0 和 UART1 控制器支持唤醒系统的功能。唤醒功能包括 CTS_n 唤醒功能 (UART_CTL[WAKE_CTS_EN]) 和传入数据唤醒功能 (UART_CTL[WAKE_DATA_EN])。当系统处于掉电模式时，UART 可以通过 CTS_n 引脚和传入数据来唤醒系统。当传入数据唤醒系统时，传入的数据会被接收并存储到接收 FIFO 中，并且控制器会自动清除 UART_CTL[WAKE_DATA_EN]

位。下图为唤醒功能框图：

CTSn唤醒情况1

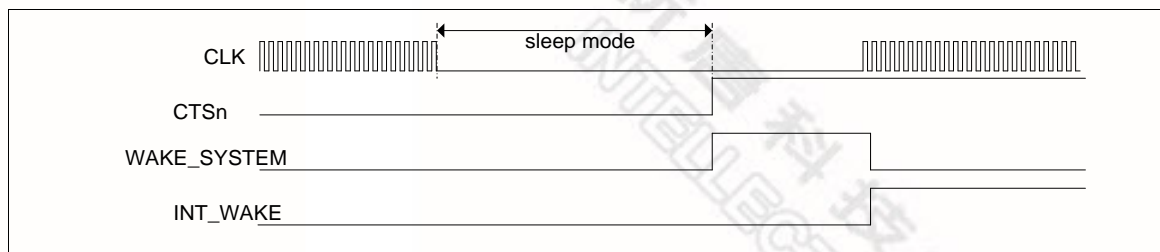


图 5-115 UART CTSn 唤醒情况 1

CTSn唤醒情况2

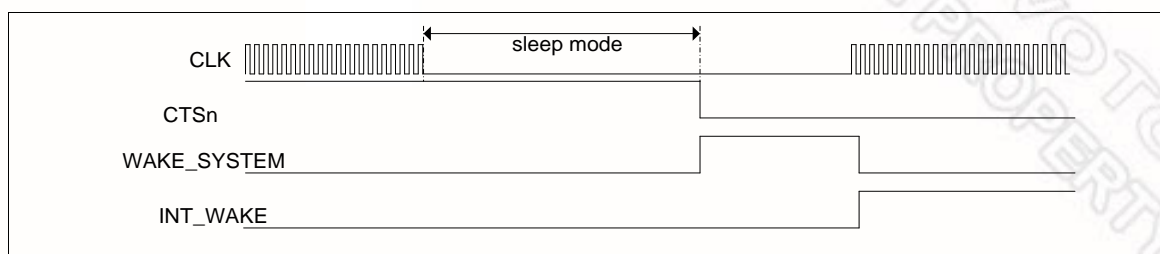


图 5-116 UART CTSn 唤醒情况 2

数据唤醒

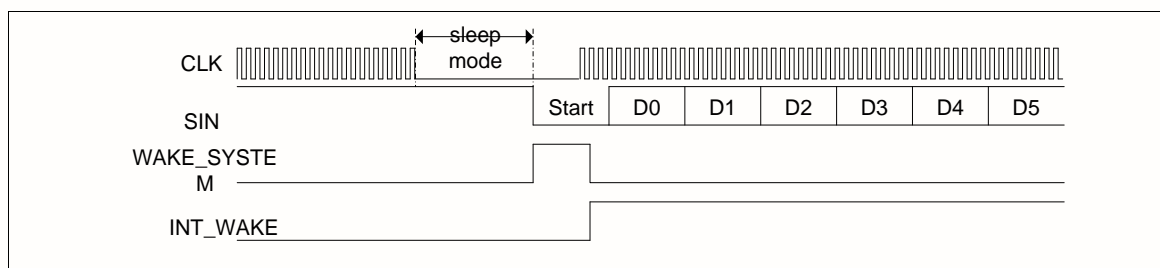


图 5-117 UART DATA 唤醒

5.19.4.4 IrDA 功能模式：

UART 支持 IrDA SIR (串行红外) 发送编码器和接收解码器，IrDA 模式可通过设定 UART_FUN_SEL 寄存器中的 FUN_SEL 位来选择，当 UART 控制器工作在 IrDA 模式时，必须通过设定 UART_TRSR[RFITL] = 0 来把接收 FIFO 触发阈值设置为 1。

IrDA 模式下，UART_BAUD[DIV_16_EN] 位必须被使能。

Baud Rate = Clock / (16 * (BRD + 1))，其中 BRD 是在波特率分频寄存器 UART_BAUD 中定义的波特率除数。

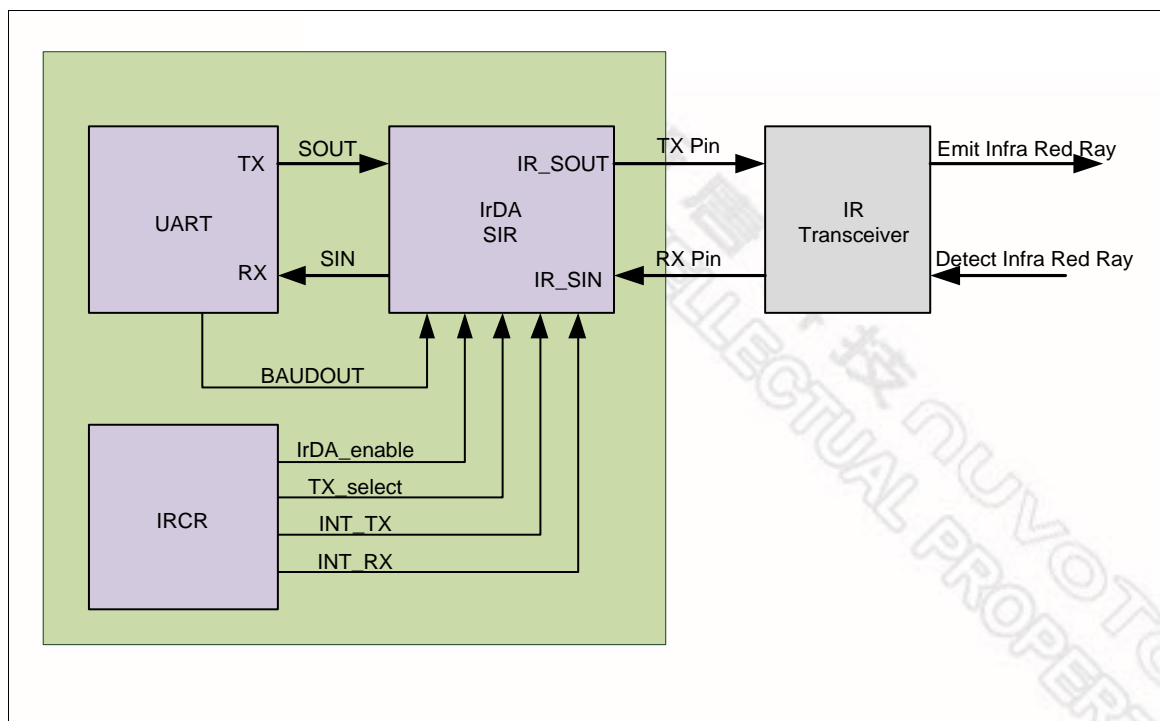


图 5-118 IrDA 框图

IrDA SIR 发送编码器

IrDA SIR 发送编码器调制 Non-Return-to Zero (NRZ) 发送位流从 UART 输出。IrDA SIR 物理层指定使用归零反向调制机制 (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线发光二极管。

发送脉冲的宽度被规定为 3/16 波特率周期。

IrDA SIR 接收解码器

IrDA SIR 接收解码器对来自输入探测器的 Return-to-Zero 比特流进行解调, 并将接收到的 NRZ 串行比特流输出到 UART。在空闲状态下, 解码器输入通常为高。(因此, IRCR 的 bit 6 默认为1)。

当解码器输入为低时, 表明检测到一个起始位。

IrDA SIR 操作

IrDA SIR 编码/解码器提供 UART 数据流和半双工串行 SIR 之间的转换。IrDA编码/解码波形图如下:

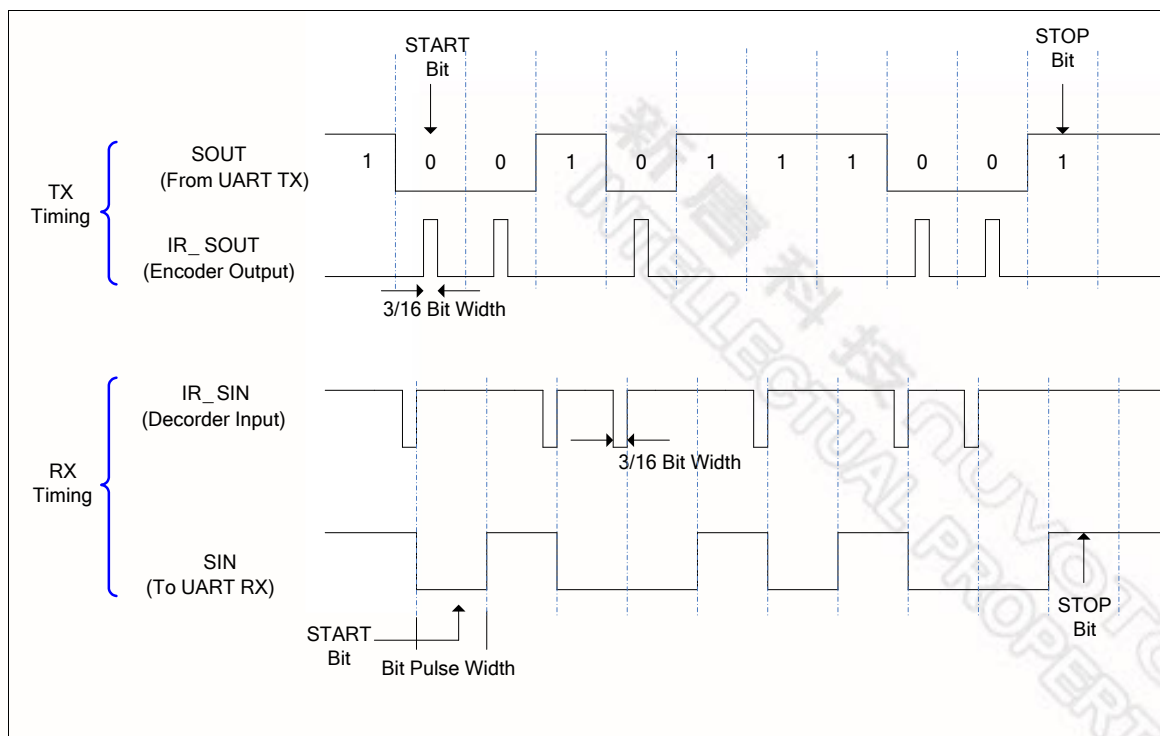


图 5-119 IrDA TX/RX 时序图

5.19.4.5 RS-485 功能模式:

UART 支持 RS-485 9 位模式功能。RS-485 模式可以通过设置寄存器 UART_FUN_SEL 来选择。当 UART 控制器工作在 RS-485 模式时，必须通过设定 UART_TRSR[RFITL] = 0 来把接收 FIFO 触发阈值设置为 1。RS-485 驱动器控制可以通过使用来自一个异步串行口的 RTSn 控制信号来实现。在 RS-485 模式下，RX 和 TX 的很多特性跟在 UART 模式下一样。

在 RS-485 功能模式下，第 9 位将会被配置为地址位，对于数据字符，第 9 位要被设置为 0，软件可以编程 UART_TLCTL 寄存器来控制第 9 位 (当 PBE, EPE 和 SPE 被置位，第 9 位作为 0 被发送；而当 PBE 和 SPE 被置位，EPE 被清零时，第 9 位作为 1 被发送)。该模式下，UART 控制器支持三种操作模式，分别是 RS-485 普通多点操作模式 (RS-485 NMM 模式)，RS-485 自动地址检测操作模式 (RS-485 AAD 模式) 和 RS-485 自动方向控制操作模式 (RS-485 AUD 模式)，可通过编程 UART_ALT_CTL 寄存器选择这三种模式中的其中一种，并且软件可以通过设置 UART_TMCTL[DLY] 寄存器来在上一次数据传输的停止位与下一次数据传输的开始位之间插入一个发送延时。

RS-485 普通多点操作模式 (NMM)

在 RS-485 普通多点操作模式下，在检测到地址字节之前 (bit 9 = "1")，软件可以决定接收器是否忽略数据。当地址字节被硬件检测到 (bit 9 = "1")，地址字节数据将会被存储到 RX-FIFO，软件可以通过设定 UART_CTL[RX_DIS] 位来决定是使能还是禁止接收器接受接下来的数据字节。如果接收器被使能 (UART_CTL[RX_DIS] 位为 0)，所有接收到的数据都将会被接受并存储到 RX-FIFO；如果接收器被禁止 (UART_CTL[RX_DIS] 位为 1)，所有接收到的字节数据都会被忽略，直到下一个地址字节被检测到。如果软件通过设定 UART_CTL[RX_DIS] 位为 1 来禁止接收器，当下一个地址字节被检测到，UART 控制器会清零 UART_CTL[RX_DIS] 位，地址字节数据将会被存储到 RX-FIFO。

编程流程示例:

1. 编程 UART_FUN_SEL 寄存器的 FUN_SEL 位来选择 RS-485 功能。
2. 编程 UART_CTL 寄存器的 RX_DIS 位来决定在地址字节被检测到之前 (bit 9 = “1”), 是否存储接收到的数据。
3. 通过设定 UART_ALT_CTL 寄存器来编程设定 RS-485_NMM。
4. 当一个地址字节被检测到 (bit 9 = “1”), 硬件会置位 UART_ISR [RLS_IS] 和 UART_TRSR [RS-485_ADDET_F] 标志。
5. 软件可以通过设定 UART_CTL [RX_DIS] 来决定是否接受接下来的数据。
6. 重复步骤 4 和步骤 5。

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 UART_ALT_CTL[ADDR_MATCH] 的值相匹配之前, 将忽略所有数据。地址字节数据将存储在 RX-FIFO, 接下来的所有数据将被接受并存储于 RX-FIFO, 直到地址字节与 UART_ALT_CTL [ADDR_MATCH] 的值不匹配。当处于 RS-485 AAD 模式时, 不要写任何值到 UART_CTL [RX_DIS] 位。

编程流程示例:

1. 编程 UART_FUN_SEL 寄存器中的 FUN_SEL 位来选择 RS-485 功能。
2. 通过设定 UART_ALT_CTL 寄存器来编程设定 RS-485_AAD。
3. 当地址字节被检测到 (bit9 = “1”), 硬件会比较地址字节与 UART_ALT_CTL[ADDR_MATCH] 的值。
4. 如果地址字节与 UART_ALT_CTL[ADDR_MATCH] 的值相匹配, 硬件会置位 UART_ISR [RLS_IS] 和 UART_TRSR [RS-485_ADDET_F] 标志, 接收器会存储地址字节到 RX-FIFO, 接受接下来的数据, 并把它们存储到 RX-FIFO, 直到下一个地址字节被检测到。

然而, 如果地址字节与 UART_ALT_CTL[ADDR_MATCH] 的值不匹配, 硬件将忽略地址字节, 并忽略接下来的数据传输。

5. 重复步骤 3 和步骤 4。

RS-485 自动方向模式 (AUD)

RS-485 控制器的另一个功能是 RS-485 自动方向控制。RS-485 驱动器控制可以通过使用来自一个异步串行口的 RTSn 控制信号来实现。RTSn 线被连接到 RS-485 驱动器使能引脚, 设置 RTSn 线为高 (逻辑1) 将使能 RS-485 驱动器; 设置 RTSn 线为低 (逻辑0), 将会使驱动器进入高阻态。用户可以通过设置寄存器 UART_MCSR 中的 LEV_RTS 位来改变 RTSn 驱动电平。

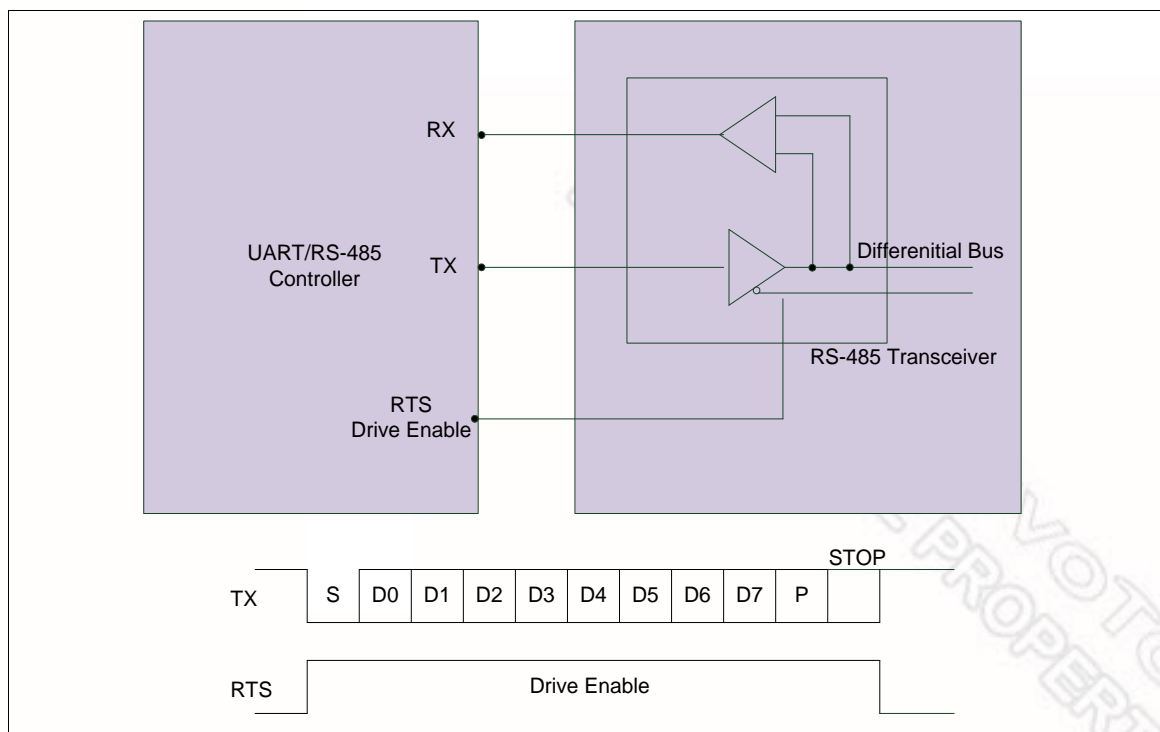


图 5-120 RS-485 帧结构

5.19.4.6 LIN (Local Interconnection Network) 功能模式:

UART 支持 LIN 功能，LIN 模式可以通过设定 UART_FUN_SEL 寄存器的来选择。在 LIN 模式下，依照 LIN 的标准，每个字节域初始由一个值为 0 的开始位 (显性)，后跟 8 个数据位 (LSB 优先)，最后是值为 1 的 1 个停止位 (隐性)。

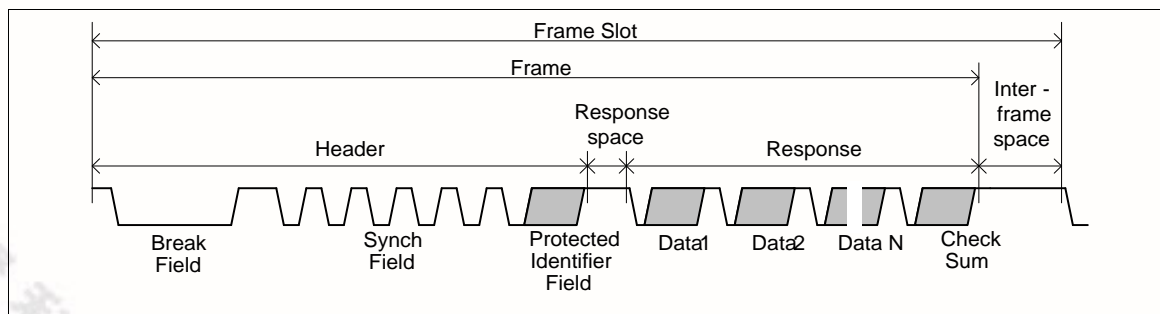


图 5-121 LIN 帧结构

LIN 总线发送传输 (TX) 编程流程:

LIN 总线发送传输 (TX) 的编程流程如下所示:

Case 1: Header 域选择为 “Break”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式
2. 通过设定 UART_ALT_CTL [LIN_HEAD_SEL] 来选择数据 header 为 “break field”。
3. 使能 UART_ALT_CTL [BIT_ERR_EN] 位，当 SIN 引脚不等于 SOUT 引脚时，硬件会向 CPU 产生一个中断)，如果用户想同时接收数据， 用户必须使能 UART_ALT_CTL [LIN_RX_EN] 位。

4. 写位域 LIN_TX_BCNT 来选择 break 域的长度 (break 域的长度是 LIN_TX_BCNT + 8)。
5. 置位 UART_ALT_CTL 寄存器中的 LIN_TX_EN 位来开始发送 break 域, 且当 break 域的操作完成后, 硬件将置位 UART_ISR[LIN_IS] 和 UART_TRSR[LIN_TX_F] 标志, 且 UART_ALT_CTL [LIN_TX_EN] 位将会被自动清零。
6. 写 0x55 到 UART_THR 请求发送 synch 域。
7. 写被保护的标识符值 (PID) 到 UART_THR。
8. 写 N 字节数据和 Checksum 到 UART_THR, 然后重复步骤 4 ~ 步骤 8 来发送数据。

Case 2: Header 域选择为 “Break + Sync”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式。
2. 通过设置 UART_ALT_CTL [LIN_HEAD_SEL]) 来选择数据 header 为 “break + sync” 域。
3. 使能 UART_ALT_CTL [BIT_ERR_EN] 位, 当 SIN 引脚不等于 SOUT 引脚时, 硬件会向 CPU 产生一个中断), 如果用户想同时接收数据, 用户必须使能 UART_ALT_CTL [LIN_RX_EN] 位。
4. 写位域 LIN_TX_BCNT 来选择 break 域的长度 (break 域的长度是 LIN_TX_BCNT + 8)。
5. 置位 UART_ALT_CTL 寄存器中的 LIN_TX_EN 位来开始发送 break 和 sync 域, 且当 break 和 sync 域的操作完成后, 硬件将置位 UART_ISR[LIN_IS] 和 UART_TRSR[LIN_TX_F] 标志, 且 UART_ALT_CTL [LIN_TX_EN] 位将会被自动清零。
6. 写被保护的标识符值 (PID) 到 UART_THR。
7. 写 N 字节数据和 Checksum 到 UART_THR, 然后重复步骤 4 ~ 步骤 7 来发送数据。

Case 3: Header 域选择为 “Break + Sync + PID”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式。
2. 通过设置 UART_ALT_CTL [LIN_HEAD_SEL]) 来选择数据 header 为 “break + sync + PID” 域。
3. 使能 UART_ALT_CTL [BIT_ERR_EN] 位, 当 SIN 引脚不等于 SOUT 引脚时, 硬件会向 CPU 产生一个中断), 如果用户想同时接收数据, 用户必须使能 UART_ALT_CTL [LIN_RX_EN] 位。
4. 写位域 LIN_TX_BCNT 来选择 break 域的长度 (break 域的长度是 LIN_TX_BCNT + 8)。
5. 置位 UART_ALT_CTL 寄存器中的 LIN_TX_EN 位来开始发送 break, sync 和 PID 域, 且当 break, sync 和 PID 域的操作完成后, 硬件将置位 UART_ISR[LIN_IS] 和 UART_TRSR[LIN_TX_F] 标志, 且 UART_ALT_CTL [LIN_TX_EN] 位将会被自动清零。
6. 写 N 字节数据和 Checksum 到 UART_THR, 然后重复步骤 4 ~ 步骤 6 来发送数据。

LIN 总线接收器传输 (RX) 编程流程

LIN 总线接收器传输 (RX) 编程流程如下所示:

Case 1: header 域选择为 “Break”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式
2. 通过设定 UART_ALT_CTL [LIN_HEAD_SEL]) 来选择数据 header 为 “break field”。
3. 设置 UART_ALT_CTL 寄存器的 LIN_RX_EN 位来使能 LIN RX 模式。
4. 等待 UART_TRSR 寄存器的 LIN_RX_F 标志来确定 RX 是否接收到了 break 域 (break 域不会被

存储到 FIFO)

5. 等待 UART_ISR 寄存器中 RDA_IF 标志置位，然后读 UART_RBR 寄存器。

Case 2: header 域选择为 “Break + Sync”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式

2. 通过设定 UART_ALT_CTL [LIN_HEAD_SEL] 来选择数据 header 为 “break + sync field”。

3. 设置 UART_ALT_CTL 寄存器的 LIN_RX_EN 位来使能 LIN RX 模式。

4. 等待 UART_TRSR 寄存器的 LIN_RX_F 标志来确定 RX 是否接收到了 break 和 sync 域，如果已经接收到了 break 和 sync 域，硬件会置位 UART_TRSR[LIN_RX_F] 标志，如果 break 域被接收到了，但是接收到的 sync 域不等于 0x55，硬件会置位 UART_TRSR[LIN_RX_F] 和 UART_TRSR[LIN_RX_SYNC_ERR_F] 标志。break 和 sync 域 (等于或不等于 0x55) 不会被存储到 FIFO

5. 等待 UART_ISR 寄存器中 RDA_IF 标志置位，然后读 UART_RBR 寄存器。

Case 3: header 域选择为 “Break + Sync + PID”

1. 通过设置 UART_FUN_SEL 寄存器来选择 LIN 功能模式

2. 通过设定 UART_ALT_CTL [LIN_HEAD_SEL] 来选择数据 header 为 “break + sync + PID field”。

3. 设置 UART_ALT_CTL 寄存器的 LIN_RX_EN 位来使能 LIN RX 模式。

4. 在这种情况下，硬件会自动控制数据传输。在接收到与在 UART_ALT_CTL[ADDR_MATCH] 中定义的值相匹配的 break + sync (0x55) + PID 之前 (break + sync + PID 不会被存储到 FIFO)，硬件忽略任何接收到的数据。当接收到的 break + sync (0x55) + PID 与在 UART_ALT_CTL [ADDR_MATCH] 中定义的值相匹配时，硬件会置位 UART_TRSR[LIN_RX_F] 标志，并且接下来的数据都将被接受并存储到 RX-FIFO 中，直到检测到下一个 break 域。如果接收器接收到了 break + 错误的 sync (不等于 0x55) + PID，硬件会置位 UART_TRSR[LIN_RX_F] 和 UART_TRSR [LIN_RX_SYNC_ERR_F] 标志，并且接收器将被禁止。如果接收器接收到了 break + sync (0x55) + 错误的 PID，硬件会置位 UART_TRSR[LIN_RX_F] 标志，并且接收器将被禁止。

5. 等待 UART_ISR 寄存器中 RDA_IF 标志置位，然后读 UART_RBR 寄存器。.

5.19.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
UART Base Address : UART0_BA = 0x4005_0000 UART1_BA = 0x4015_0000				
UARTx_RBR	UARTx_BA + 0x00	R	UART 接收缓存寄存器.	Undefined
UARTx_THR	UARTx_BA + 0x00	W	UART 发送保持寄存器.	Undefined
UARTx_CTL	UARTx_BA + 0x04	R/W	UART 控制状态寄存器.	0x0000_0000
UARTx_TLCTL	UARTx_BA + 0x08	R/W	UART传输线控制寄存器	0x0000_0000
UARTx_IER	UARTx_BA + 0x0C	R/W	UART 中断使能寄存器.	0x0000_0000
UARTx_ISR	UARTx_BA + 0x10	R/W	UART 中断状态寄存器	0x0000_0002
UARTx_TRSR	UARTx_BA + 0x14	R/W	UART 传输状态状态寄存器.	0x0000_0000
UARTx_FSR	UARTx_BA + 0x18	R/W	UART FIFO 状态状态寄存器	0x0000_0A02
UARTx_MCSR	UARTx_BA + 0x1C	R/W	UART Modem 状态状态寄存器.	0x0002_0002
UARTx_TMCTL	UARTx_BA + 0x20	R/W	UART 超时控制状态寄存器.	0x0000_01FF
UARTx_BAUD	UARTx_BA + 0x24	R/W	UART 波特率除数寄存器	0x0000_0000
UARTx_IRCR	UARTx_BA + 0x30	R/W	UART IrDA 控制寄存器	0x0000_0040
UARTx_ALT_CTL	UARTx_BA + 0x34	R/W	UART 可选控制状态寄存器	0x0000_0000
UARTx_FUN_SEL	UARTx_BA+0x38	R/W	UART 功能选择寄存器	0x0000_0000

注: UARTx_REG 中的 x 代表 UART 通道.

5.19.6 寄存器描述

UART接收缓存寄存器(UARTx_RBR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_RBR	UARTx_BA + 0x00	R	UART接收缓存寄存器.	未定义

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

Bits	描述	
[31:8]	-	保留.
[7:0]	RBR[7:0]	接收缓存寄存器 读该寄存器，UART 会返回一个从 RX 引脚接收到的 8 位数据 (LSB 优先)

UART发送保持寄存器(UARTx_THR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_THR	UARTx_BA + 0x00	W	UART发送保持寄存器	未定义

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

Bits	描述	
[31:8]	-	保留.
[7:0]	THR[7:0]	发送保持寄存器 写该寄存器，UART 将通过 TX 引脚发送出一个 8 位 (LSB 优先).

UART Control 寄存器 (UARTx CTL)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_CTL	UARTx_BA + 0x04	R/W	UART Control 寄存器.	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-			ABAUD_EN	-		WAKE_DATA_EN	WAKE_CTS_EN
7	6	5	4	3	2	1	0
DMA_TX_EN	DMA_RX_EN	AUTO_CTS_EN	AUTO_RTS_EN	TX_DIS	RX_DIS	TX_RST	RX_RST

Bits	描述	
[31:13]	-	保留.
[12]	ABAUD_EN	<p>自动波特率检测使能</p> <p>1 = 使能自动波特率检测功能.</p> <p>0 = 禁止自动波特率检测功能</p> <p>注: 当自动波特率检测操作结束时, 硬件会清除该位, 如果 UART_IER[ABAUD_IE] 位被使能, 相关的中断 (INT_ABAUD) 将会产生。</p>
[11:10]	-	保留.
[9]	WAKE_DATA_EN	<p>传入数据唤醒功能使能</p> <p>1 = 使能传入数据唤醒功能。当系统处于掉电模式时, 传入数据将会把系统从掉电模式唤醒。</p> <p>0 = 禁止传入数据唤醒系统功能</p> <p>注: 当传入数据唤醒操作完成, 并且系统时钟工作稳定之后, 硬件会清除该位。</p>
[8]	WAKE_CTS_EN	<p>CTS_n 唤醒功能使能</p> <p>1 = 使能唤醒功能, 当系统处于掉电模式时, 一个在 CTS_n 引脚上的电平变化会将系统从掉电模式唤醒。</p> <p>0 = 禁止 CTS_n 唤醒系统功能</p>
[7]	DMA_TX_EN	<p>TX DMA 使能</p> <p>该位可以使能或禁止 TX PDMA 服务</p> <p>1 = 使能 TX PDMA 服务功能</p> <p>0 = 禁止 TX PDMA 服务功能</p>

Bits	描述	
[6]	DMA_RX_EN	RX DMA 使能 该位可以使能或禁止 RX PDMA 服务 1 = 使能RX PDMA 服务功能. 0 = 禁止RX PDMA 服务功能..
[5]	AUTO_CTS_EN	CTS_n 自动流控使能 1 = 使能 CTS _n 自动流控. 0 = 禁止 CTS _n 自动流控 注: 当 CTS _n 自动流控被使能, 如果 CTS _n 输入被触发, UART 将发送数据到外部设备。(在 CTS _n 被触发之前, UART 不会发送数据到外部设备).
[4]	AUTO_RTS_EN	RTS_n 自动流控功能使能 1 = 使能 RTS _n 自动流控. 0 = 禁止 RTS _n 自动流控. 注: 当 RTS _n 自动流控被使能, 如果在 RX-FIFO 中的字节数等于在 UART_FCR [RTS_Tri_Lev] 中设定的值, UART 会重新触发 RTS _n 信号。
[3]	TX_DIS	发送器禁止寄存器. 发送器是否被禁止 (置 “1” 禁止发送器) 1 = 禁止发送器 0 = 使能发送器
[2]	RX_DIS	接收器禁止寄存器. 接收器是否被禁止 (置 “1” 禁止接收器) 1 = 禁止接收器 0 = 使能接收器 注 1: 当使用 RS-485 NMM 模式, 用户可以设定该位来决定在检测到地址字节之前是否接收数据。. 注 2: 当处于 RS-485 AAD 模式下, 该位会被自动置位。 注 3: 当处于 RS-485 AUD 模式, 或者 LIN “break + sync +PID” header 模式下, 硬件会自动控制数据, 因此不要向该位写任何值。
[1]	TX_RST	TX 软件复位 当 TX_RST 被置位, 在发送 FIFO 中的所有字节数据, 和 TX 内部状态机都将被清除。 1 = 写 “1” 到该位将复位 TX 内部状态机和指针。 0 = 写 “0” 到该位没有影响。 注: 该位会自动清除, 会起至少 3 个 UART 时钟周期的作用。
[0]	RX_RST	RX 软件复位 当 RX_RST 被置位, 在接收 FIFO 中的所有字节数据, 和 RX 内部状态机都将被清除。 1 = 写 “1” 到该位将复位 RX 内部状态机和指针。 0 = 写 “0” 到该位没有影响。 注: 该位会自动清除, 会起至少 3 个 UART 时钟周期的作用。

UART传输线控制寄存器(UARTx TLCTL)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_TLCTL	UARTx_BA + 0x08	R/W	UART传输线控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-		RTS_TRI_LEV			-		RFITL
7	6	5	4	3	2	1	0
-	BCB	SPE	EPE	PBE	NSB	DATA_LEN	

Bits	描述											
[31:14]	-	保留.										
[13:12]	RTS_TRI_LEV [1:0]	RTSn 触发阈值 (用于自动流控)										
		<table><tr><th>RTS_TRI_LEV</th><th>Trigger Level (Bytes)</th></tr><tr><td>00</td><td>01</td></tr><tr><td>01</td><td>04</td></tr><tr><td>10</td><td>08</td></tr><tr><td>11</td><td>14</td></tr></table>	RTS_TRI_LEV	Trigger Level (Bytes)	00	01	01	04	10	08	11	14
		RTS_TRI_LEV	Trigger Level (Bytes)									
		00	01									
		01	04									
10	08											
11	14											
注：该位域用于自动 RTSn 流控。												
[11:10]	-	保留.										
[9:8]	RFITL[1:0]	RX-FIFO 中断 (INT_RDA) 触发阈值										
		当接收 FIFO 中断字节数等于 RFITL 中断设定值时，RDA_IF 将会被置位，如果 IER[RDA_IEN] 被置位，将会发生一个接收中断。										
		<table><tr><th>RFITL</th><th>INTR_RDA Trigger Level (Bytes)</th></tr><tr><td>00</td><td>01</td></tr><tr><td>01</td><td>04</td></tr><tr><td>10</td><td>08</td></tr><tr><td>11</td><td>14</td></tr></table>	RFITL	INTR_RDA Trigger Level (Bytes)	00	01	01	04	10	08	11	14
		RFITL	INTR_RDA Trigger Level (Bytes)									
		00	01									
01	04											
10	08											
11	14											
注：当工作在 IrDA 模式或 RS-485 模式，RFITL 必须被设置为“0”。												

Bits	描述														
[7]	-	保留。													
[6]	BCB	Break 控制位 当该位被设置为逻辑“1”，串行数据输出 (TX) 被强制为 Spacing 状态 (逻辑“0”)。该位仅对 TX 引脚起作用，对发送逻辑没有任何影响。													
[5]	SPE	Stick 校验使能 1 = 当 PBE, EPE 和 SPE 都被置位，发送数据时，校验位总是为“0”；接收数据时，校验位被检测到为“0”才认为接收到的数据是正确的。当 PBE 和 SPE 被置位，EPE 被清零，发送数据时，校验位总是为“1”；接收数据时，校验位被检测到为“1”才认为接收到的数据是正确的。当处于 RS-485 模式下，PBE, EPE 和 SPE 可以控制第 9 位，第 9 位的设定值如下所示： 0 = 禁止 stick 校验 <table><tr><td rowspan="3">RS-485 mode</td><td>SPE</td><td>EPE</td><td>PBE</td><td>Bit9</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	RS-485 mode	SPE	EPE	PBE	Bit9	1	1	1	0	1	0	1	1
RS-485 mode	SPE	EPE		PBE	Bit9										
	1	1		1	0										
	1	0	1	1											
[4]	EPE	偶校验使能 1 = 在发送模式下，偶数个 1 将被发送；在接收模式下，检验数据字和校验位。 0 = 在发送模式下，奇数个 1 将被发送；在接收模式下，检验数据字和校验位。 注：该位仅在 PBE 位 (校验位使能) 被置位时起作用。													
[3]	PBE	校验位使能 1 = 校验位在串行数据的最后一个数据位和停止位之间被产生或者被检验 0 = 在传输过程中，校验位不会被产生 (发送数据) 或被检验 (接收数据)。													
[2]	NSB	停止位长度 1 = 当发送数据长度被设置为 5 位时，在发送数据时，1.5 “STOP bit” 被产生；而当发送数据长度被设置为 6, 7 或者 8 位时，在发送数据时，2 “STOP bit” 被产生 0 = 发送数据时，1 “STOP bit” 被产生													
[1:0]	DATA_LEN [1:0]	数据长度 <table><tr><td>WLS[1:0]</td><td>Character length</td></tr><tr><td>00</td><td>5 bits</td></tr><tr><td>01</td><td>6 bits</td></tr><tr><td>10</td><td>7 bits</td></tr><tr><td>11</td><td>8 bits</td></tr></table>	WLS[1:0]	Character length	00	5 bits	01	6 bits	10	7 bits	11	8 bits			
WLS[1:0]	Character length														
00	5 bits														
01	6 bits														
10	7 bits														
11	8 bits														

UART Interrupt Enable 寄存器 (UARTx_IER)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_IER	UARTx_BA + 0x0C	R/W	UART中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							LIN_IE
7	6	5	4	3	2	1	0
ABAUD_IE	WAKE_IE	BUF_ERR_IE	RTO_IE	MODEM_IE	RLS_IE	THRE_IE	RDA_IE

Bits	描述	
[31:9]	-	保留。
[8]	LIN_IE	LIN 中断使能 1 = 使能 INT_LIN 0 = 禁止 INT_LIN
[7]	ABAUD_IE	自动波特率中断使能 1 = 使能 INT_ABAUD 0 = 禁止 INT_ABAUD
[6]	WAKE_IE	唤醒中断使能 1 = 使能 INT_WAKE 0 = 禁止 INT_WAKE
[5]	BUF_ERR_IE	缓存错误中断使能 1 = 使能 INT_BUF_ERR 0 = 禁止 INT_BUF_ERR
[4]	RTO_IE	RX 超时中断使能 1 = 使能 INT_TOUT 0 = 禁止 INT_TOUT
[3]	MODEM_IE	Modem 状态中断使能 1 = 使能 INT_MOS 0 = 禁止 INT_MOS
[2]	RLS_IE	接收线状态中断使能

Bits	描述	
		1 = 使能 INT_RLS 0 = 禁止 INT_RLS
[1]	THRE_IE	发送保持寄存器空中断使能 1 = 使能 INT_THRE 0 = 禁止 INT_THRE
[0]	RDA_IE	接收数据有效中断使能. 1 = 使能 INT_RDA 0 = 禁止 INT_RDA

UART Interrupt Status Control 寄存器 (UARTx_ISR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_ISR	UARTx_BA + 0x10	R/W	UART中断状态寄存器.	0x0000_0002

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							LIN_IS
7	6	5	4	3	2	1	0
ABAUD_IS	WAKE_IS	BUF_ERR_IS	RTO_IS	MODEM_IS	RLS_IS	THRE_IS	RDA_IS

Bits	描述	
[31:9]	-	保留.
[8]	LIN_IS	<p>LIN 中断状态标志 (只读)</p> <p>当 LIN TX header 被发送, RX header 被接收或者 SIN 不等于 SOUT 时, 该位被置位, 如果 IER[LIN_IE] 被设置为 1, LIN 中断将会被产生.</p> <p>注 1: 该位只读, 但是可以通过向 UART_TRSR[BIT_ERR_F], UART_TRSR[BIT_TX_F] 或 UART_TRSR[LIN_RX_F] 写 1 清除该标志. .</p> <p>注 2: 当 BIT_ERR_F, BIT_TX_F 和 LIN_RX_F 都被清除时, 该标志被清除</p>
[7]	ABAUD_IS	<p>自动波特率中断状态标志 (只读)</p> <p>当自动波特率检测功能完成, 或者自动波特率计数器溢出时, 该位被置位, 如果 IER[ABAUD_IE] 被置位, 自动波特率中断将会被产生.</p> <p>注 1: 该位只读, 但是可以通过向 UART_TRSR[ABAUD_TOUT_F] 或 UART_TRSR[ABAUD_F] 写 1 清除该标志. .</p> <p>注 2: 当 ABAUD_TOUT_F 和 ABAUD_F 都被清除时, 该标志被清除</p>
[6]	WAKE_IS	<p>唤醒中断状态标志 (只读)</p> <p>系统处于掉电模式下, 当接收器接收到数据或者 CTSn 引脚上有电平变化时, 该位被置位. 如果 IER[WAKE_IE] 被置位, 唤醒中断将会被产生. .</p> <p>注: 该位只读, 但是可以向该位写 1 清除该标志.</p>
[5]	BUF_ERR_IS	<p>缓存错误中断状态标志 (只读)</p> <p>当 TX 或者 RX FIFO 溢出时, 该位被置位. 当 BUF_ERR_IS 被置, 数据传输可能不正确, 如果 IER[BUF_ER_IEN] 被置位, 缓存错误中断将会被产生. .</p> <p>注 1: 该位只读, 但是可以通过向 UART_FSR[TX_OVER_F] 或 UART_FSR[RX_OVER_F] 写 1 清除该标志. .</p> <p>注 2: 当 TX_OVER_F 和 RX_OVER_F 都被清除时, 该标志被清除</p>

Bits	描述	
[4]	RTO_IS	RX 超时中断状态标志 (只读) 当 RX-FIFO 非空, 并且在由 TOIC 定义的超时时间内, 都没有新的数据被接收到 RX-FIFO, 而 CPU 也没有从 RX-FIFO 读取接收到的数据的情况下, 该位被置位, 如果 IER [Tout_IEN] 被置位, 超时中断将会被产生。 注: 该位只读, 用户可以通过读 UART_RBR (RX 处于激活状态) 来清除该位。
[3]	MODEM_IS	MODEM 中断状态标志 (只读) 当 CTSn 引脚上有电平变化 (DCTSF = 1)时, 该位被置位, 如果 IER [MODEM_IEN] 被置位, modem 中断将会被产生。 注: 该位只读, 但是可以通过写 “1” 到 UART_MCSR [DCT_F] 来清除该标志。
[2]	RLS_IS	接收线中断状态标志 (只读). 当接收到的数据有校验位错误 (UART_FSR[PE_F]), 格式错误 (UART_FSR[FE_F]), break 错误 (UART_FSR [BI_F]) 或者 RS-485 检测到地址字节 (UART_TRSR[RS-485_ADDDET_F]) 时, 该位被置位, 如果 IER [RLS_IEN] 被置位, RLS 中断将会被产生。 注 1: 该位只读, 但是可以通过向 UART_FSR[BI_F], UART_FSR[FE_F], UART_FSR [PE_F] 或者 UART_TRSR [RS-485_ADDDET_F] 写 “1” 清除该标志。 注 2: 当 BI_F, FE_F, PE_F 和 RS-485_ADDDET_F 均被清除时, 该位被清除
[1]	THRE_IS	发送保持寄存器空中断标志 (只读). 当 TX-FIFO 被传送到发送器的移位寄存器时, 该位被置位, 如果 IER[THRE_IEN] 被置位, THRE 中断将会被产生 注: 该位只读, 当写数据到 THR (TX-FIFO 非空), 该标志将会被清除
[0]	RDA_IS	接收数据有效中断标志 (只读). 当接收到 RX-FIFO 中的字节数据个数等于由 RFITL 设定的值时, RDA_IF 将会被置位, 如果 IER[RDA_IEN] 被置位, RDA 中断将会被产生 注: 该位只读, 当 RX-FIFO 中剩余的未读的字节数据个数小于接收阈值 (RFITL) 时, 该标志将会被清除。

Interrupt Indicator	中断源	中断使能	中断标志	标志清除
INT_LIN	LIN 功能 中断	BUF_ERR_IE	LIN_IS = UART_TRSR [BIR_ERR_F] or UART_TRSR [LIN_RX_F] or UART_TRSR [LIN_TX_F]	Write “1” to UART_TRSR [BIR_ERR_F] or UART_TRSR [LIN_RX_F] or UART_TRSR [LIN_TX_F].
INT_ABAUD	自动波特率中断	ABAUD_IE	ABAUD_IS = UART_TRSR [ABAUD_TOUT_F] or UART_TRSR [ABAUD_F]	Write “1” to UART_TRSR [ABAUD_TOUT_F] UART_TRSR [ABAUD_F].
INT_WAKE	唤醒中断	WAKE_IE	WAKE_IS	Write “1” to

				UART_ISR[WAKE_IS]
INT_BUF_ERR	缓存错误中断	BUF_ERR_IE	BUF_ERR_IS = UART_FSR [RX_OVER_F] or UART_FSR [TX_OVER_F].	Write "1" to UART_FSR [RX_OVER_F] or UART_FSR [TX_OVER_F].
INT_RTO	RX Time-Out 中断	RTO_IE	RTO_IS	Read UART_RBR
INT_MODEM	Modem状态中断	MODEM_IE	MODME_IS = UART_MCSR[DCT_F]	Write "1" to UART_MCSR[DCT_F]
INT_RLS	接收线状态中断	RLS_IE	RLS_IS = UART_FSR [BI_F] or UART_FSR [FE_F] or UART_FSR [PE_F] or UART_TRSR [RS- 485_ADDDET_F]	Write "1" to UART_FSR [BI_F] or UART_FSR [FE_F] or UART_FSR [PE_F] or UART_TRSR [RS- 485_ADDDET_F]
INT_THRE	发送保持寄存器空中断	THRE_IE	THRE_IS	Write UART_THR
INT_RDA	接收数据有效中断	RDA_IE	RDA_IS	Read UART_RBR

表 5-15 UART 中断源和标志

UART传输状态寄存器(UARTx TRSR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_TRSR	UARTx_BA + 0x14	R/W	UART传输状态寄存器.	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							LIN_RX_SYNC_ERR_F
7	6	5	4	3	2	1	0
-		BIT_ERR_F	LIN_RX_F	LIN_TX_F	ABAUD_TOUT_F	ABAUD_F	RS-485_ADDDET_F

Bits	描述	
[31:9]	-	保留.
[8]	LIN_RX_SYNC_ERR_F	<p>LIN RX SYNC错误标志 (只读)</p> <p>当 LIN 接收到一个错误的 SYNC 域时, 该位被置 1.</p> <p>用户可以通过设定 UART_ALT_CTL[LIN_HEAD_SEL] 寄存器来选择 LIN header.</p> <p>如果 LIN header 包括 “break field + sync field”, 且 sync 不等于 0x55, LIN_RX_F 和 LIN_RX_SYNC_ERR_F 将会被置位, 而错误的 sync 数据将会被忽略. 控制器将会接收下一个的数据, 并存储到 FIFO.</p> <p>如果 LIN header 包括 “break field + sync field + PID field”, 且 sync 不等于 0x55, LIN_RX_F 和 LIN_RX_SYNC_ERR_F 将会被置位, 而错误的 sync 数据将会被忽略. 控制器将会接收下一个的数据, 并存储到 FIFO.</p> <p>注: 该位只读, 但是可以通过向 LIN_RX_F 写 1 来清除该标志.</p>
[7:6]	-	保留.
[5]	BIT_ERR_F	<p>位错误检测状态标志 (只读)</p> <p>在发送状态下, 硬件会监视总线状态, 如果输入引脚 (SIN) 状态不等于输出引脚 (SOUT) 状态, BIT_ERR_F 将会被置位.</p> <p>当发生了位错误, 硬件会向 CPU 产生一个中断 (INT_LIN).</p> <p>注 1: 该位只读, 但是可以向该位写 1 来清除该标志</p> <p>注 2: 该位仅在使能位错误检测功能 (UART_ALT_CTL [BIT_ERR_EN] = “1”) 时有效</p>
[4]	LIN_RX_F	<p>LIN RX 中断标志 (只读)</p> <p>当接收到 LIN header 域, 该位被置位, header 域可以是 “break field” 或 “break field + sync field” 或 “break field + sync field + PID field”, 可以通过设定 UART_ALT_CTL [LIN_HEAD_SEL] 寄存器来选择.</p> <p>如果 header 域包括 “break field”, 当接收器接收到 break 域, LIN_RX_F 将会被置</p>

Bits	描述	
		<p>位，控制器会接收下一个数据，并存储到 FIFO。</p> <p>如果 header 域包括 “break field + sync field”，硬件会等待 UART_TRSR 寄存器中 LIN_RX_F 标志来检查 RX 已经收到了 break 域和 sync 域。如果 break 域和 sync 域已经被接收到，硬件会置位 UART_TRSR[LIN_RX_F] 位，如果 break 已经被接收到，但是 sync 域不等于 0x55，硬件会置位 UART_TRSR[LIN_RX_F] 和 UART_TRSR[LIN_RX_SYNC_ERR_F] 标志。break 和 sync 数据 (等于或不等于 0x55) 不会被存储到 FIFO。</p> <p>如果 header 域包括 “break field + sync field + PID field”，当工作在该模式下，硬件会自动控制数据，在接收到与由 UART_ALT_CTL [ADDR_MATCH] 定义的值相匹配的 break + sync (0x55) + PID (break + sync + PID 不会被存储到 FIFO) 之前，硬件会忽略任何数据，当接收到的 break + sync (0x55) + PID 的值与由 UART_ALT_CTL [ADDR_MATCH] 定义的值相匹配时，硬件会置位 UART_TRSR [LIN_RX_F]，接下来的所有数据将会被接受并存储到 RX-FIFO，直到检测到下一个 break 域；如果接收器接收到了 break + 错误的 sync (不等于 0x55) + PID 值，硬件会置位 UART_TRSR [LIN_RX_F] 和 UART_TRSR[LIN_RX_SYNC_ERR_F] 标志，接收器将会被禁止；如果接收器接收到了 break + sync (0x55) + 错误的 PID 值，硬件会置位 UART_TRSR [LIN_RX_F] 标志，接收器将会被禁止。</p> <p>注：该位只读，但是可以通过向该位写 1 清除该标志。</p>
[3]	LIN_TX_F	<p>LIN TX 中断标志 (只读)</p> <p>当 LIN 发送出 header 域之后，该位被置位。header 域可以是 “break field” 或 “break field + sync field” 或 “break field + sync field + PID field”，可以通过设定 UART_ALT_CTL [LIN_HEAD_SEL] 寄存器来选择。</p> <p>注：该位只读，但是可以通过向该位写 1 清除该标志。</p>
[2]	ABAUD_TOUT_F	<p>自动波特率超时中断 (只读)</p> <p>当工作在自动波特率检测模式，且波特率计数器溢出时，该位被置位。</p> <p>注：该位只读，但是可以通过向该位写 1 清除该标志。</p>
[1]	ABAUD_F	<p>自动波特率中断 (只读)</p> <p>当自动波特率检测功能完成时，该位被置位。</p> <p>注：该位只读，但是可以通过向该位写 1 清除该标志。</p>
[0]	RS-485_ADDDET_F	<p>RS-485 地址字节检测状态标志 (只读)</p> <p>无论何时，当在 RS-485 模式下，接收器检测到任何地址字节数据 (bit 9 = “1”)，该位被置位，且 UART_ALT_CTL[RS-485_ADD_EN] 位被置位。当 CPU 向该位写 1 时，该位被复位。</p> <p>注 1：该位域用于 RS-485 模式。</p> <p>注 2：该位只读，但是可以通过向该位写 1 清除该标志。</p>

UART FIFO状态寄存器(UART_FSR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_FSR	UARTx_BA + 0x18	R/W	UART FIFO状态寄存器.	0x0000_0A02

31	30	29	28	27	26	25	24
-			TX_POINTER_F				
23	22	21	20	19	18	17	16
-			RX_POINTER_F				
15	14	13	12	11	10	9	8
-				TE_F	TX_FULL_F	TX_EMPTY_F	TX_OVER_F
7	6	5	4	3	2	1	0
-	BI_F	FE_F	PE_F	-	RX_FULL_F	RX_EMPTY_F	RX_OVER_F

Bits	描述	
[31:28]	-	保留.
[28:24]	TX_POINTER_F[4:0]	TX-FIFO 指针 (只读) 该位表示 TX-FIFO 缓存指针。每当 CPU 写一个字节数据到 UART_THR，TX_POINTER_F 加 1。当 TX-FIFO 中一个字节的数据被传送到发送器移位寄存器，TX_POINTER_F 减 1。
[23:21]	-	保留.
[20:16]	RX_POINTER_F[4:0]	RX-FIFO 指针 (只读) 该位表示 RX-FIFO 缓存指针，当 UART 从外设接收一个字节，RX_POINTER_F 加 1。当 RX-FIFO 中断一个字节被 CPU 读出，RX_POINTER_F 减 1。
[15:12]	-	保留.
[11]	TE_F	发送器空状态标志 (只读) 当 TX 未激活时 (TX 移位寄存器中没有数据)，该位被硬件置位。 当 TX-FIFO 传送数据到 TX 移位寄存器，或者 TX 为空，但是数据传输尚未完成时，该位被自动清除。
[10]	TX_FULL_F	发送器 FIFO 满 (只读) 该位指示 TX-FIFO 是否满。 当 TX_POINTER_F 等于 16 时，该位被置位，否则该位被硬件清除。
[9]	TX_EMPTY_F	发送器 FIFO 空 (只读) 该位表示 TX-FIFO 是否空。 当 TX-FIFO 中最后一个字节数据已经被传送到发送器移位寄存器后，硬件置位该位。 当写数据到 THR (TX-FIFO 非空) 时，该位被清除。

Bits	描述	
[8]	TX_OVER_F	TX 溢出错误中断状态标志 (只读) 如果 TX-FIFO (UART_THR) 满, 向 UART_THR 一个额外的写入会导致该位被置位。 注: 该位只读, 但是可以通过向该位写 1 清除该标志。
[7]	-	保留。
[6]	BI_F	Break 状态标志 (只读) 当接收到的数据输入 (RX) 被保持在 “spacing state” (逻辑 0) 超过一个字传输时间 (整个 “start bit” + data bits + parity + stop bits 的时间), 该位被置位。当 CPU 向该位写 1, 该位被复位。 注: 该位只读, 但是可以通过向该位写 1 清除该标志。
[5]	FE_F	格式错误状态标志 (只读) 当接收到的字符没有一个有效的停止位 (即, 跟随在最后一个数据位或校验位之后的停止位被检测到为逻辑 0) 时, 该位被置位。当 CPU 向该位写 1, 该位被复位。 注: 该位只读, 但是可以通过向该位写 1 清除该标志。
[4]	PE_F	校验位错误状态标志 (只读) 当接收到的字符没有一个有效的校验位时, 该位被置位。当 CPU 向该位写 1, 该位被复位。 注: 该位只读, 但是可以通过向该位写 1 清除该标志。
[3]	-	保留。
[2]	RX_FULL_F	接收器 FIFO 满 (只读) 该位表示 RX-FIFO 是否满。 当 RX_POINTER_F 等于 16 时, 该位被置位, 否则该位被硬件清除。
[1]	RX_EMPTY_F	接收器 FIFO 空 (只读) 该位表示 RX-FIFO 是否空。 当 RX-FIFO 中最后一个字节已经被 CPU 读出, 硬件设置该位为高, 当 UART 接收到一个新的数据时, 该位被清除。
[0]	RX_OVER_F	RX 溢出错误状态标志 (只读) 当 RX-FIFO 溢出时, 该位被置位。 如果接收到的字节数据个数大于 RX-FIFO (UART_RBR) 的大小, 16 字节, 该位将会被置位。 注: 该位只读, 但是可以通过向该位写 1 清除该标志。

UART MODEM控制寄存器(UARTx_MCSR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_MCSR	UARTx_BA + 0x1C	R/W	UART Modem控制状态寄存器.	0x0002_0002

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

Bits	描述					
[31:19]	-	保留.				
[18]	DCT_F	检测 CTSn 状态变化状态标志 (只读) 当 CTSn 输入状态改变时, 该位被置位, 如果 UART_IER [Modem_IEN] 被置位, 将会向 CPU 产生 Modem 中断。. 注: 该位只读, 但是可以通过向该位写 1 清除该标志。				
[17]	CTS_ST	CTSn 引脚状态 (只读) 该位反映 CTSn 引脚状态.				
[16]	LEV_CTS	CTSn 触发电平 该位可以改变 CTSn 触发电平. 1 = 高电平触发 0 = 低电平触发				
		工作模式	LEV_CTS	CTSn 管脚输入	CTS_ST	发送状态
		CTS 自动流控模式	0	0	0	STOP
			0	1	1	ACTIVE
			1	0	0	ACTIVE
			1	1	1	STOP
[15:12]	-	保留.				
[1]	RTS_ST	RTSn 引脚状态 (只读) 该位反映 RTSn 引脚状态.				

Bits

描述

[0]

LEV_RTS

RTSn 触发电平

该位可以改变 RTSn 触发电平.

1 = 高电平触发

0 = 低电平触发

例如，在 LEV_RTS 和 RTSn 之间，相关的波形如下所示：

工作模式	LEV_RTS	RTS_ST (默认输出状态)
RS-485 AUD 模式 (Note)	0	0
	1	1
RTS 自动流控模式 (Note)	0	1
	1	0
正常模式	0	1
	1	0

注：当处于 RS-485 AUD 模式和 RTS 自动流控模式下，硬件会自动控制输出 RTS 引脚，因此该表表示默认值。

UART Mode :

LEV_RTS

RTS_ST

RS-485 Mode : LEV_RTS = 0

TX

Start bit

D0

D1

D2

D3

D4

D5

D6

D7

P

STOP

RTS_ST (TX_EN)

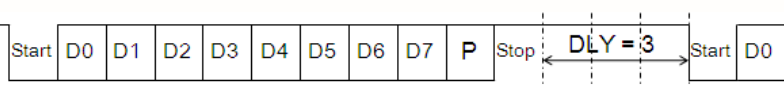
Drive Enable

注：在 UART 模式下，默认的设置是 LEV_RTS = “0” 和 RTS_ST = “1”.

UART超时寄存器(UARTx_TMCTL)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_TMCTL	UARTx_BA + 0x20	R/W	UART0超时控制寄存器	0x0000_01FF

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

Bits	描述	
[31:24]	-	保留.
[23:16]	DLY[7:0]	<p>TX 延迟时间值</p> <p>该位用于编程在最后一个字节数据的停止位到下一个字节数据的开始位之间的传输延迟时间, 通过 UART_TMCTL[DLY] 寄存器来设定。</p>  <p>注 1: 向该位域写 0 表示禁止该功能.</p> <p>注 2: 实际延迟值是 DLY.</p> <p>注 3: 计数时钟是波特率时钟.</p>
[15:9]	-	保留.
[8:0]	TOIC[8:0]	<p>超时比较器</p> <p>当 RX-FIFO 接收到一个新数据时, 超时计数器复位并开始计数 (计数时钟 = 波特率时钟)。一旦超时计数器 (TOUT_CNT) 的值等于超时中断比较器 (TOIC), 如果 UART_IER[RTO_IEN] 被置位, 一个接收器超时中断 (INT_TOUT) 将会被产生。一个新的传输数据或者 RX-FIFO 空清除 INT_TOUT 中断。</p> <p>注 1: 向该位域写 0 表示禁止该功能.</p> <p>注 2: 实际超时值是 TOIC + 1.</p> <p>注 3: 计数时钟是波特率时钟.</p> <p>注 4: UART 数据格式是 start bit + 8 bit data + parity bit + stop bit, 所以, 尽管软件可以配置该域为任意值, 还是推荐写到该位域的值大于 0xA.</p>

UART波特率除数寄存器(UARTx BAUD)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_BAUD	UARTx_BA + 0x24	R/W	UART波特率除数寄存器	0x0000_0000

31	30	29	28	27	26	25	24
DIV_16_EN	-						
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
BRD							
7	6	5	4	3	2	1	0
BRD							

Bits	描述										
[31]	DIV_16_EN	<p>除 16 使能</p> <p>BRD = 波特率除数, 波特率公式为: $\text{Baud Rate} = \text{UART_CLK} / [16 * (\text{BRD} + 1)]$;</p> <p>1 = 波特率计算公式为 $\text{UART_CLK} / [16 * (\text{BRD} + 1)]$</p> <p>0 = 波特率计算公式为 $\text{UART_CLK} / [(\text{BRD} + 1)]$</p> <p>注: 当处于 IrDA 模式时, 该位必须被设置为 0.</p>									
[30:16]	-	保留.									
[15:0]	BRD[15:0]	<p>波特率除数</p> <p>波特率除数的低字节</p> <table> <tr> <th>DIV_16_EN</th><th>BRD</th><th>波特率等式</th></tr> <tr> <td>禁止 (Mode 0)</td><td>A</td><td>$\text{UART_CLK} / (A + 1)$, A must > 8</td></tr> <tr> <td>使能 (Mode 1)</td><td>A</td><td>$\text{UART_CLK} / [16 * (A + 1)]$</td></tr> </table>	DIV_16_EN	BRD	波特率等式	禁止 (Mode 0)	A	$\text{UART_CLK} / (A + 1)$, A must > 8	使能 (Mode 1)	A	$\text{UART_CLK} / [16 * (A + 1)]$
DIV_16_EN	BRD	波特率等式									
禁止 (Mode 0)	A	$\text{UART_CLK} / (A + 1)$, A must > 8									
使能 (Mode 1)	A	$\text{UART_CLK} / [16 * (A + 1)]$									

UART IrDA控制寄存器(UARTx_IRCR)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_IRCR	UARTx_BA + 0x30	R/W	UART IrDA控制寄存器	0x0000_0040

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

Bits	描述	
[31:7]	-	保留.
[6]	INV_RX	INV_RX 1 = 反转 RX 输入信号 0 = 不反转
[5]	INV_TX	INV_TX 1 = 反转 TX 输出信号 0 = 不反转
[4:2]	-	保留.
[1]	TX_SELECT	TX_SELECT 1 = 使能 IrDA 发送器 0 = 使能 IrDA 接收器 注: 当处于 IrDA 模式时, UART_BAUD [DIV_16_EN] 寄存器必须被置位 (波特率公式必须为 Clock / 16 * (BRD))
[0]	-	保留.

UART可选控制状态寄存器(UARTx_ALT_CTL)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_ALT_CSR	UARTx_BA + 0x34	R/W	UART可选控制状态寄存器.	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_PID_MATCH							
23	22	21	20	19	18	17	16
-				RS-485_ADD_EN	RS-485_AUD	RS-485_AAD	RS-485_NMM
15	14	13	12	11	10	9	8
-							BIT_ERR_EN
7	6	5	4	3	2	1	0
LIN_TX_EN	LIN_RX_EN	LIN_HEAD_SEL		-	LIN_TX_BCNT		

Bits	描述	
[31:24]	ADDR_PID_MATCH[7:0]	<p>Address / PID 匹配值寄存器</p> <p>当处于 RS-485 模式下, 该位域包含 RS-485 的地址匹配值.</p> <p>当处于 LIN 功能模式下, 该位域包含 LIN 被保护标识符域, 软件写 ID0 ~ ID5 (ADDR_PID_MATCH[5:0]), 硬件会计算 P0 和 P1.</p> <div style="text-align: center;"> <p>PID ——— Start bit ID0 ID1 ID2 ID3 ID4 ID5 P0 P1</p> <p>$P0 = ID0 \text{ xor } ID1 \text{ xor } ID2 \text{ xor } ID4$</p> <p>$P1 = \sim(ID1 \text{ xor } ID3 \text{ xor } ID4 \text{ xor } ID5)$</p> </div> <p>注: 该位域用于 RS-485 自动地址检测模式或者 LIN 被保护标识符域 (PID)</p>
[23:20]	-	保留.
[19]	RS-485_ADD_EN	<p>RS-485 地址检测使能</p> <p>该位用于使能 RS-485 硬件地址检测模式, 如果硬件检测到地址字节, 控制器会置位 UART_TRSR[RS_485_ADDDET_F]</p> <p>1 = 使能地址检测模式</p> <p>0 = 禁止地址检测模式</p> <p>注: 该位域可用于 RS-485 任意操作模式</p>
[18]	RS-485_AUD	<p>RS-485 自动方向模式 (RS-485 AUD Mode)</p> <p>1 = 使能 RS-485 自动方向模式 (AUD)</p> <p>0 = 禁止 RS-485 自动方向模式 (AUD)</p> <p>注: 可以跟 RS-485_AAD 或 RS-485_NMM 操作模式同时使用</p>

Bits	描述											
[17]	RS-485_AAD	RS-485 自动地址检测操作模式 (RS-485 AAD Mode) 1 = 使能 RS-485 自动地址检测操作模式 (AAD) 0 = 禁止 RS-485 自动地址检测操作模式 (AAD) 注：不可以跟 RS-485_NMM 操作模式同时使用。										
[16]	RS-485_NMM	RS-485 普通多点操作模式 (RS-485 NMM Mode) 1 = 使能 RS-485 普通多点操作模式 (NMM) 0 = 禁止 RS-485 普通多点操作模式 (NMM) 注：不可以跟 RS-485_AAD 操作模式同时使用。										
[15:9]	-	保留。										
[8]	Bit_ERR_EN	位错误检测使能 1 = 使能位错误检测功能。 0 = 禁止位错误检测功能。 注：在 LIN 功能模式下，当位错误发生时，硬件将向 CPU 产生一个中断 (INT_LIN)。										
[7]	LIN_TX_EN	LIN TX Header 触发使能 1 = 使能 LIN TX Header 触发。 0 = 禁止 LIN TX Header 触发。 注 1：当 TX header 域 (“break field” 或 “break + sync field” 或 “break + sync + PID field”) 传输操作完成，该位将自动被清除，并将向 CPU 产生一个中断 (INT_LIN)。 注 2：如果用户想接收发送的数据，需要使能 LIN_RX_EN 位。										
[6]	LIN_RX_EN	LIN RX 使能 当 LIN RX 模式被使能，且接收到了一个 break field 或 sync field 或 PID field (通过 LIN_Header_SEL 选择)，控制器将向 CPU 产生一个中断 (INT_LIN) 1 = 使能 LIN RX 模式。 0 = 禁止 LIN RX 模式。										
[5:4]	LIN_HEAD_SEL[1:0]	LIN Header 选择 <table><tr><th>LIN_HEADER_SEL</th><th>描述</th></tr><tr><td>00</td><td>LIN header 包含 “break field”。</td></tr><tr><td>01</td><td>LIN header 包含 “break field + sync field”。</td></tr><tr><td>10</td><td>LIN header 包含 “break field + sync field + PID field”。</td></tr><tr><td>11</td><td>保留</td></tr></table>	LIN_HEADER_SEL	描述	00	LIN header 包含 “break field”。	01	LIN header 包含 “break field + sync field”。	10	LIN header 包含 “break field + sync field + PID field”。	11	保留
LIN_HEADER_SEL	描述											
00	LIN header 包含 “break field”。											
01	LIN header 包含 “break field + sync field”。											
10	LIN header 包含 “break field + sync field + PID field”。											
11	保留											
[3]	-	保留。										
[2:0]	LIN_TX_BCNT[2:0]	LIN TX Break 域数量寄存器 该域包含 3-bit LIN TX break 域数量。 注：break 域长度是 LIN_TX_BCNT + 8。										

UART功能选择寄存器(UARTx FUN_SEL)

寄存器	偏移量	R/W	描述	复位后的值
UARTx_FUN_SEL	UARTx_BA + 0x38	R/W	UART功能选择寄存器	0x0000_0000

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

Bits	描述											
[31:2]	-	保留.										
[1:0]	FUN_SEL[1:0]	功能选择										
		<table><tr><th>FUN_SEL</th><th>描述</th></tr><tr><td>00</td><td>UART 功能模式.</td></tr><tr><td>01</td><td>LIN 功能模式.</td></tr><tr><td>10</td><td>IrDA 功能模式.</td></tr><tr><td>11</td><td>RS-485 功能</td></tr></table>	FUN_SEL	描述	00	UART 功能模式.	01	LIN 功能模式.	10	IrDA 功能模式.	11	RS-485 功能
		FUN_SEL	描述									
		00	UART 功能模式.									
		01	LIN 功能模式.									
		10	IrDA 功能模式.									
11	RS-485 功能											

5.20 USB

5.20.1 概述

该 USB 控制器是一组 USB 2.0 全速设备控制器，符合 USB 2.0 全速设备规范，支持控制/批量/中断/等时传输类型。

在该设备控制器中，包含两个主接口：APB 总线和来自 USB PHY 收发器的 USB 总线。CPU 能够通过 APB 总线编程控制寄存器。在该控制器中内置有 512-字节的 SRAM 作为数据缓存。对于 IN 令牌或 OUT 令牌传输，需要通过 APB 接口向 SRAM 写数据或从 SRAM 读数据。用户需要通过缓存分段寄存器 (BUFSEG) 为每个端点缓存分配有效的 SRAM 起始地址。

USB 设备控制器共有 8 个可配置的端点。每个端点可以配置为 IN 或者 OUT 类型。设备的功能地址和每个端点的端点号必须事先正确的配置好以便用于正确的接收和发送数据包。每个端点发送/接收到长度定义在最大负载寄存器 (MXPLD)，主机和设备间的握手也由端点来处理。

该控制器有 4 种不同的中断事件，分别是唤醒功能，设备插拔事件，USB 事件（如 IN ACK, OUT ACK 等）和 BUS 事件（如 挂起 和 恢复 等）。任何事件都将会引发一个中断，用户只需要在中断事件状态寄存器 (USB_INTSTS) 中检查相关事件标志以获知发生何种中断，然后检测相关的 USB 端点状态寄存器 (USB_EPSTS) 以获知在该端点上发生何种事件。

USB 设备控制器有一个软件禁用功能，用于模拟设备从主机分离的情况。如果用户使能 DRVSE0 位 (USB_CTL[4])，USB 控制器将强制 USB_DP 和 USB_DM 输出低电平，USB 设备功能被禁止 (断开)。在禁用 DRVSE0 位之后，主机将重新枚举 USB 设备。

参考文献：通用串行总线规范修订版 2.0

5.20.2 特性

该通用串行总线 (USB) 为一个带有单个连接器的串行接口，可以连接所有 USB 外设到主机系统。下面是 USB 的一些特性。

- 兼容 USB 2.0 全速规范
- 提供一个 1 个中断向量，4 个中断事件 (WAKEUP, FLDET, USB 和 BUS)
- 支持 控制/批量/中断/等时传输类型
- 支持在没有总线活动超过 3 ms 之后的挂起功能
- 提供 8 个可配置为控制/批量/中断/等时传输类型的端点
- 512-字节 SRAM 内置缓存
- 提供远程唤醒功能

5.20.4 功能描述

SIE是设备控制器的前端，处理大多数 USB 协议包。SIE 一般包括向上发送信号到事务处理层面。处理功能包括：

- #### 5.20.4.2 端点控制

该控制器有 8 端点。每个端点可以被配置成 IN 或者 OUT 端点，所有操作包括 控制，批量，中断和等时传输都在该模块内执行。端点控制模块也用于管理数据序列同步，端点状态，当前端点起始

地址，当前事务状态和每个端点的数据缓存状态。

5.20.4.3 缓存控制

控制器中有 512-字节 SRAM，8 个端点共享该缓存。在功能激活前，用户必须在相应的缓存段寄存器中配置每个端点的有效起始地址。缓存控制模块 (BUFFER CONTROL) 用于控制每个端点的有效起始地址，SRAM 的大小在 MXPLD 寄存器中定义。

下图描述了依据 BUFSEG 和 MXPLD 寄存器的内容定义的每个端点起始地址。如果 BUFSEG0 被编程设为 0x08h，MXPLD0 为 0x40h，则端点0 的SRAM 大小是从 USB_BASE + 0x108h 开始，结束于 USB_BASE + 0x148h，因为 USB SRAM 的基地址是 USB_BASE + 0x100h。

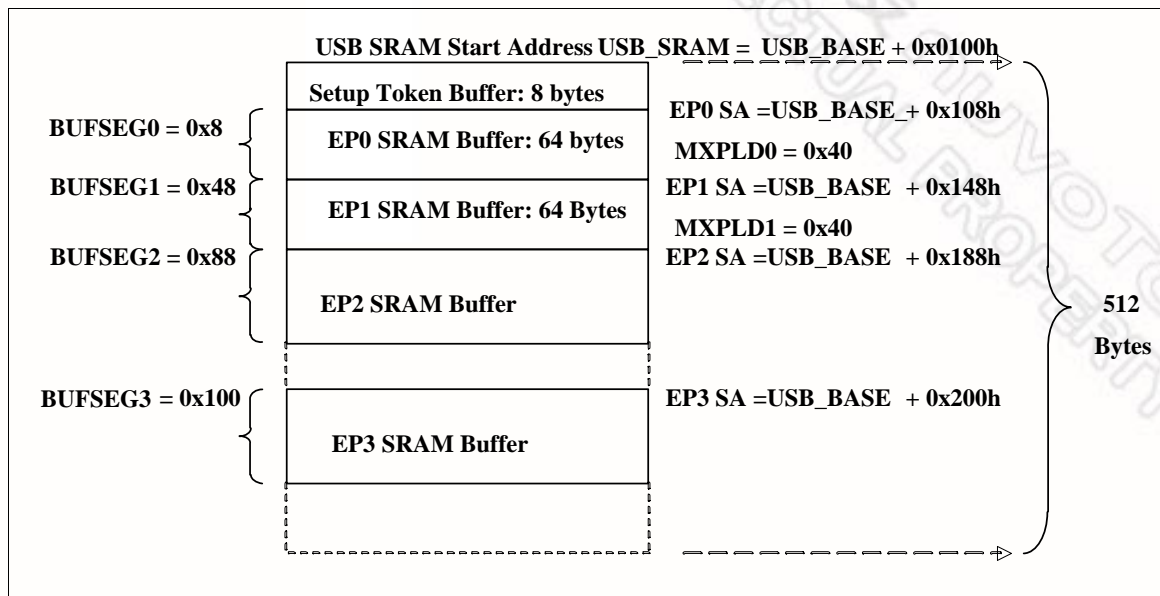


图 5-123 SRAM 中各端点的分配

5.20.4.4 DPLL (数字锁相环)

USB 数据的比特率为 12 MHz。DPLL 采用由时钟控制器产生的 48 MHz 频率来锁定 RXDP 和 RXDM 上的数据。12 MHz 的比特率时钟也是由 DPLL 转换而来。

5.20.4.5 插/拔去抖

USB 设备可以进行热插拔操作，为了监测USB设备被拔出时的状态，设备控制器为 USB 悬空检测中断提供了硬件去抖动以防止在 USB 插拔时产生的抖动问题。悬空检测中断产生于 USB 进行插拔操作的 10 ms后，用户可以通过读取 USB_BUSSTS 寄存器中“FLDET”的值，来应答 USB 的插拔。“FLDET”标志代表当前总线上没有过去抖动处理的状态。如果 FLDET 为 1，则意味着控制器已经插入 USB。如果用户要通过这个标志轮询来检测 USB 的状态，则必要时需要添加软件去抖动功能

5.20.4.6 中断和唤醒

该 USB 提供唤醒功能和 1 个带有 4 个中断事件 (WAKEUP, FLDET, USB 和 BUS) 的中断向量。WAKEUP 事件被用来在掉电模式下唤醒系统时钟。（掉电模式功能在系统掉电控制寄存器 PWRCON 中定义）。FLDET 事件用于 USB 插拔。USB 事件告知用户一些 USB 请求，如 IN ACK, OUT ACK 等。BUS 事件告知用户一些总线事件，如挂起、恢复等。用户必须在 NVIC 和 USB 设备控制器的中断使能寄存器 (USB_INTEN) 设置相应的位以使能 USB 中断。

唤醒中断只会出现在芯片进入掉电模式且唤醒事件发生时。在芯片进入掉电模式后，在 USB_DP，USB_DM 和设备悬空引脚上的任何变化都能唤醒芯片（假定 USB 唤醒功能被使能）。若这个变化不是有意而为的，例如，在悬空检测引脚上的噪声，那么只有唤醒中断会发生。若 USB 唤醒超过 20 ms 后，没有其他 USB 中断事件发生，唤醒中断将发生。下图为唤醒中断的控制流程。

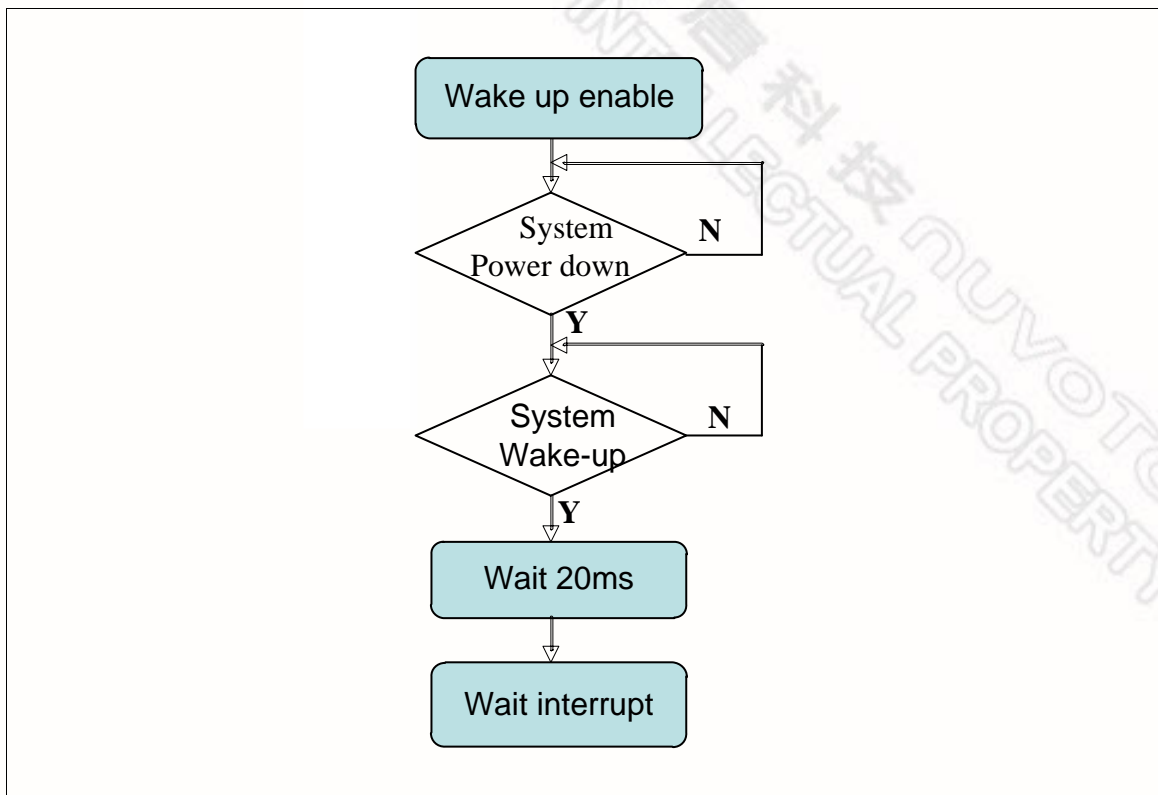


图 5-124 USB 唤醒中断操作流程

USB 中断用于告知用户总线上的所有 USB 事件，用户可以通过读寄存器 USB_EPSTS 和 USB_INTSTS 来获知是哪类请求，在哪个端点上，并采取必要的响应。

与 USB 中断类似，BUS 中断告知用户一些总线事件，如 USB 复位、挂起、超时和恢复。用户可以读寄存器 USB_BUSSTS 获取总线事件。

5.20.4.7 省电

当芯片进入掉电模式时，USB 自动关闭 PHY 收发器以减少功耗。此外，在一些特殊情况下，如挂起，用户可以向 PHY_EN 位 (USB_CTL[1]) 写入“0”关闭 PHY 来进入省电状态。

5.20.4.8 与 USB 外设通信事务处理

用户可以采用中断或轮询检测 USB_INTSTS 来监测 USB 事务，当一个 USB 事务发生时，USB_INTSTS 由硬件设置，并向 CPU 发送中断请求（如果相关中断使能）。若中断未使能，用户可以轮询检测 USB_INTSTS 寄存器来获取这些事件。以下是使能中断时的控制流程。

当 USB 主机向设备控制器请求数据时，用户需要预先准备相关的数据到指定的端点缓存。在将数据写入缓冲区后，用户需要写入实际数据长度到指定的 MAXPLD 寄存器。一旦这个寄存器被写入数据，内部信号 “In_Rdy” 将会被触发，当收到主机发送的相关 IN token 之后，缓冲数据将被立刻发送。注意在发送指定数据之后，信号 “In_Rdy” 会由硬件自动清除。

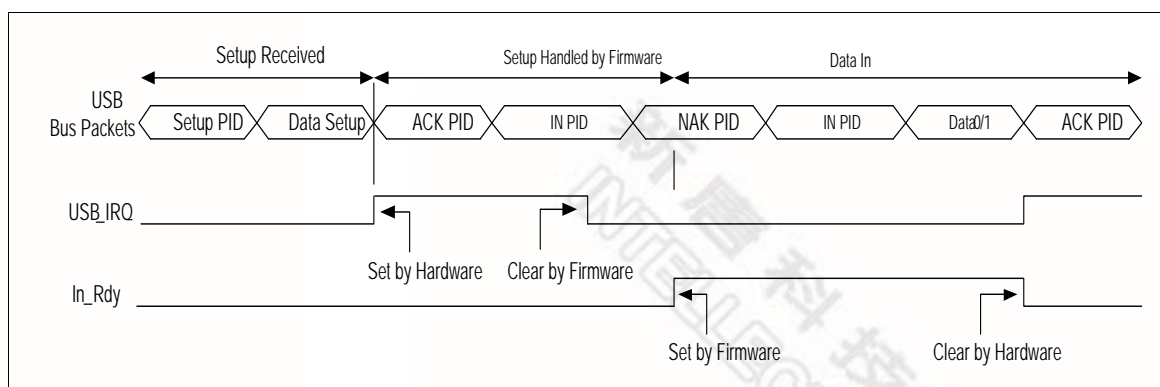


图 5-125 USB 数据 IN 事务紧跟 Setup 事务

USB 主机要发送数据到设备控制器的 OUT 端点，硬件将这些数据存在指定的端点缓存里，事务完成后，硬件在相关的 MAXPLD 寄存器记录数据长度，并清除“Out_Rdy”信号，这可以避免硬件在用户没有取走当前数据时接收下一个事务。一旦用户处理了这次事务，相关的寄存器 “MAXPLD” 需要由软件写入来再一次设置 “Out_Rdy” 信号，以接收下一个事务。

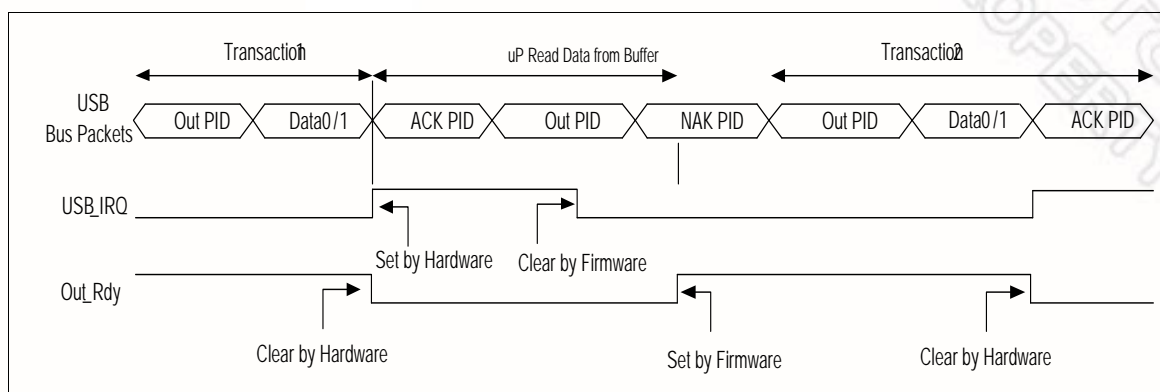


图 5-126 USB 数据输出事务

5.20.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write, W/C: Write 1 Clear

寄存器	偏移量	R/W	描述	复位后的值
USB_BA = 0x4006_0000				
USB_CTL	USB_BA+0x000	W, R/W	USB 控制寄存器	0x0000_0900
USB_BUSSTS	USB_BA+0x004	R	USB 总线状态寄存器	0x0000_0000
USB_INTEN	USB_BA+0x008	W, R/W	中断使能寄存器	0x0000_0000
USB_INTSTS	USB_BA+0x00C	R/W&C	中断事件状态寄存器	0x0000_0000
USB_FADDR	USB_BA+0x010	R/W	设备功能地址寄存器	0x0000_0000
USB_EPSTS	USB_BA+0x014	R	端点状态寄存器	0x0000_0000
USB_BUFSEG	USB_BA+0x018	R/W	Setup 令牌 (Token) 缓存段寄存器	0x0000_0000
USB_BUFSEG0	USB_BA+0x020	R/W	端点 0 缓存段寄存器	0x0000_0000
USB_MXPLD0	USB_BA+0x024	R/W	端点 0 最大负载寄存器	0x0000_0000
USB_CFG0	USB_BA+0x028	R/W/C	端点 0 配置寄存器	0x0000_0000
USB_BUFSEG1	USB_BA+0x030	R/W	端点 1 缓存段寄存器	0x0000_0000
USB_MXPLD1	USB_BA+0x034	R/W	端点 1 最大负载寄存器	0x0000_0000
USB_CFG1	USB_BA+0x038	R/W/C	端点 1 配置寄存器	0x0000_0000
USB_BUFSEG2	USB_BA+0x040	R/W	端点 2 缓存段寄存器	0x0000_0000
USB_MXPLD2	USB_BA+0x044	R/W	端点 2 最大负载寄存器	0x0000_0000
USB_CFG2	USB_BA+0x048	R/W/C	端点 2 配置寄存器	0x0000_0000
USB_BUFSEG3	USB_BA+0x050	R/W	端点 3 缓存段寄存器	0x0000_0000
USB_MXPLD3	USB_BA+0x054	R/W	端点 3 最大负载寄存器	0x0000_0000
USB_CFG3	USB_BA+0x058	R/W/C	端点 3 配置寄存器	0x0000_0000
USB_BUFSEG4	USB_BA+0x060	R/W	端点 4 缓存段寄存器	0x0000_0000
USB_MXPLD4	USB_BA+0x064	R/W	端点 4 最大负载寄存器	0x0000_0000
USB_CFG4	USB_BA+0x068	R/W/C	端点 4 配置寄存器	0x0000_0000
USB_BUFSEG5	USB_BA+0x070	R/W	端点 5 缓存段寄存器	0x0000_0000
USB_MXPLD5	USB_BA+0x074	R/W	端点 5 最大负载寄存器	0x0000_0000
USB_CFG5	USB_BA+0x078	R/W/C	端点 5 配置寄存器	0x0000_0000
USB_BUFSEG6	USB_BA+0x080	R/W	端点 6 缓存段寄存器	0x0000_0000
USB_MXPLD6	USB_BA+0x084	R/W	端点 6 最大负载寄存器	0x0000_0000
USB_CFG6	USB_BA+0x088	R/W/C	端点 6 配置寄存器	0x0000_0000

USB_BUFSEG7	USB_BA+0x090	R/W	端点 7 缓存段寄存器	0x0000_0000
USB_MXPLD7	USB_BA+0x094	R/W	端点 7 最大负载寄存器	0x0000_0000
USB_CFG7	USB_BA+0x098	R/W/C	端点 7 配置寄存器	0x0000_0000
USB_BIST	USB_BA+0x0A0	R/W	USB 缓存自测寄存器	0x0000_0000
USB_PDMA	USB_BA+0x0A4	R/W	USB PDMA 控制寄存器	0x0000_0000

寄存器	偏移量	R/W	描述	复位后的值
USB_BA = 0x4006_0000				
SRAM	USB_BA + 0100h ~ USB_BA + 02FFh	512-字节	USB SRAM 缓存大小	

5.20.6 寄存器描述

USB 控制寄存器 (USB_CTL)

寄存器	偏移量	R/W	描述	复位后的值
USB_CTL	USB_BA+0x00	R/W	USB 控制寄存器	0x0000_0090

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-	-					WAKEUP_EN	RWAKEUP
7	6	5	4	3	2	1	0
-			DRVSE0	DPPU_EN	PWRDB	PHY_EN	USB_EN

Bits	描述	
[31:10]	-	保留
[9]	WAKEUP_EN	唤醒功能使能 1 = 使能 USB 唤醒功能 0 = 禁用 USB 唤醒功能
[8]	RWAKEUP	远程唤醒 1 = 强制 USB 总线到 K (USB_DP 低, USB_DM: 高) 状态, 用于远程唤醒 0 = 不强制 USB 总线到 K 状态
[7:5]	-	保留
[4]	DRVSE0	强制 USB PHY 收发器去驱动 SE0 (单端 0) 当 USB_DP 和 USB_DM 都被拉低时, 为 SE0. 1 = 强制 USB PHY 收发器驱动 SE0 0 = 无 默认值为“1”.
[3]	DPPU_EN	USB_DP 的上拉电阻使能 1 = 使能 USB_DP 总线的上拉电阻 0 = 禁用 USB_DP 总线的上拉电阻
[2]	PWRDB	PHY 收发器掉电, 低电平有效 1 = 打开 PHY 收发器的相关电路 0 = 掉电 PHY 收发器的相关电路

[1]	PHY_EN	PHY 收发器使能 1 = 使能 PHY 收发器 0 = 禁用 PHY 收发器
[0]	USB_EN	USB 功能使能 1 = 使能 USB 0 = 禁用 USB

USB 总线状态寄存器 (USB BUSSTS)

寄存器	偏移量	R/W	描述	复位后的值
USB_BUSSTS	USB_BA+0x004	R	USB 总线状态	0x0000_0000

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

Bits	Description	
[31: 5]	-	保留
[4]	FLDET	设备悬空检测 1 = 当控制器连接到 USB, 该位被置位 "1" 0 = 控制器没有连接到 USB
[3]	TIMEOUT	超时标志 1 = 总线无响应超过 18 bits 时间。该位只读。
[2]	RESUME	恢复状态 1 = 从挂起状态恢复。该位只读。
[1]	SUSPEND	挂起状态 1 = 总线空闲超过 3ms, 或线缆拔出或主机休眠。该位只读。
[0]	USBRST	USB 复位状态 1 = 当 SE0 (single-ended 0) 超过 2.5us, 总线复位。该位只读。

USB 中断使能寄存器 (USB_INTEN)

寄存器	偏移量	R/W	描述	复位后的值
USB_INTEN	USB_BA+0x08	R/W	USB 中断使能	0x0000_0000

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

Bits	描述	
[31: 4]	-	保留
[3]	WAKEUP_IE	USB 唤醒中断使能 1 = 使能唤醒中断 0 = 禁用唤醒中断
[2]	FLDET_IE	悬空检测中断使能 1 = 使能悬空检测中断 0 = 禁用悬空检测中断
[1]	USBEVT_IE	USB 事件中断使能 1 = 使能 USB 事件中断 0 = 禁用 USB 事件中断
[0]	BUSEVT_IE	总线事件中断使能 1 = 使能总线事件中断 0 = 禁用总线事件中断

USB 中断状态寄存器 (USB_INTSTS)

寄存器	偏移量	R/W	描述	复位后的值
USB_INTSTS	USB_BA+0x0C	R/W/C	中断事件状态	0x0000_0000

31	30	29	28	27	26	25	24
SETUP	-						
23	22	21	20	19	18	17	16
EPEVT7	EPEVT6	EPEVT5	EPEVT4	EPEVT3	EPEVT2	EPEVT1	EPEVT0
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-				WKEUP_STS	FLD_STS	USB_STS	BUS_STS

Bits	描述	
[31]	SETUP	Setup 事件状态 1 = Setup 事件发生, 向 USB_INTSTS[31] 写“1” 清标志 0 = 无 Setup 事件
[30: 24]	-	保留
[23]	EPEVT7	端点 7 的 USB 事件状态 1 = 端点 7 上的 USB 事件, 检查 USB_EPSTS2[6:4]可知发生了何种 USB 事件, 向 USB_INTSTS[23] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 7 没有事件发生
[22]	EPEVT6	端点 6 的 USB 事件状态 1 = 端点 6 上的 USB 事件, 检查 USB_EPSTS2[2:0]可知发生了何种 USB 事件, 向 USB_INTSTS[22] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 6 没有事件发生
[21]	EPEVT5	端点 5 的 USB 事件状态 1 =端点 5 上的 USB 事件, 检查 USB_EPSTS[31:28]可知发生了何种 USB 事件, 向 USB_INTSTS[21] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 5 没有事件发生
[20]	EPEVT4	端点 4 的 USB 事件状态 1 = 端点 4 上的 USB 事件, 检查 USB_EPSTS[27:24]可知发生了何种 USB 事件, 向 USB_INTSTS[20] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 4 没有事件发生
[19]	EPEVT3	端点 3 的 USB 事件状态 1 = 端点 3 上的 USB 事件, 检查 USB_EPSTS[23:20]可知发生了何种 USB 事件, 向

		USB_INTSTS[19] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 3 没有事件发生
[18]	EPEVT2	端点 2 的 USB 事件状态 1 = 端点 2 上的 USB 事件, 检查 USB_EPSTS[19:16]可知发生了何种 USB 事件, 向 USB_INTSTS[18] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 2 没有事件发生
[17]	EPEVT1	端点 1 的 USB 事件状态 1 = 端点 1 上的 USB 事件, 检查 USB_EPSTS[15:12]可知发生了何种 USB 事件, 向 USB_INTSTS[17] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 1 没有事件发生
[16]	EPEVT0	端点 0 的 USB 事件状态 1 = 端点 0 上的 USB 事件, 检查 USB_EPSTS[11:8]可知发生了何种 USB 事件, 向 USB_INTSTS[16] 或者 USB_INTSTS[1] 写“1” 清零 0 = 端点 0 没有事件发生
[15:4]	-	保留
[3]	WKEUP_STS	唤醒中断状态 1 = 唤醒事件发生, 向 USB_INTSTS[3] 写 1 清零 0 = 无唤醒事件发生
[2]	FLD_STS	悬空中断状态 1 = USB 总线上有连接事件, 向 USB_INTSTS[2] 写“1” 清零 0 = USB 总线上无连接事件
[1]	USB_STS	USB 中断状态 USB 事件包括 Setup Token, IN Token, OUT ACK, ISO IN, 或 ISO OUT, 该位用于指示是否有其中之一的事件在总线上。 1 = USB 事件发生, 检查USB_EPSTS [31:8]中的 EPSTS0~7[3:0] 可知发生了何种 USB 事件, 向 USB_INTSTS [1] 或 USB_INTSTS[31] 或 EPEVT0~7 写 “1” 清除该位。 0 = 无 USB 事件发生
[0]	BUS_STS	总线中断状态 总线事件意味着总线上存在着暂停或者恢复功能。该位用于指示总线上有其中之一的事件发生。 1 = 总线事件发生; 检查 USB_BUSSTS[3:0] 可知发生了何种总线事件, 向 USB_INTSTS[0] 写“1” 清零。 0 = 无总线事件发生

USB 设备地址寄存器 (USB_FADDR)

在 USB 总线上作为设备地址的 7-位值。

寄存器	偏移量	R/W	描述	复位后的值
USB_FADDR	USB_BA+0x010	R/W	USB 设备功能地址	0x0000_0000

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

Bits	描述	
[31:7]	-	保留
[6:0]	FADDR	USB 设备功能地址

USB 端点状态寄存器 (USB EPSTS)

寄存器	偏移量	R/W	描述	复位后的值
USB_EPSTS	USB_BA+0x014	R	端点总线状态	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS5				EPSTS4			
23	22	21	20	19	18	17	16
EPSTS3				EPSTS2			
15	14	13	12	11	10	9	8
EPSTS1				EPSTS0			
7	6	5	4	3	2	1	0
OVERRUN	-						

Bits	描述															
[31:28]	EPSTS5	端点 5 总线状态 这些位用于表示该端点的当前状态。														
		<table><tr><th>EPSTS5</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS5	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
		EPSTS5	描述													
		000	INACK													
		001	保留													
		010	OUT Packet Data0 ACK													
		110	OUT Packet Data1 ACK													
		011	Setup ACK													
111	等时传输结束															
[27:24]	EPSTS4	端点 4 总线状态 这些位用于表示该端点的当前状态。														
		<table><tr><th>EPSTS4</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS4	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
		EPSTS4	描述													
		000	INACK													
		001	保留													
		010	OUT Packet Data0 ACK													
		110	OUT Packet Data1 ACK													
		011	Setup ACK													
111	等时传输结束															
[23:20]	EPSTS3	端点 3 总线状态														

		<div>这些位用于表示该端点的当前状态。</div> <table><tr><th>EPSTS3</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS3	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
EPSTS3	描述															
000	INACK															
001	保留															
010	OUT Packet Data0 ACK															
110	OUT Packet Data1 ACK															
011	Setup ACK															
111	等时传输结束															
[19:16]	EPSTS2	<div>端点 2 总线状态</div> <div>这些位用于表示该端点的当前状态。</div> <table><tr><th>EPSTS2</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS2	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
EPSTS2	描述															
000	INACK															
001	保留															
010	OUT Packet Data0 ACK															
110	OUT Packet Data1 ACK															
011	Setup ACK															
111	等时传输结束															
[15:12]	EPSTS1	<div>端点 1 总线状态</div> <div>这些位用于表示该端点的当前状态。</div> <table><tr><th>EPSTS1</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS1	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
EPSTS1	描述															
000	INACK															
001	保留															
010	OUT Packet Data0 ACK															
110	OUT Packet Data1 ACK															
011	Setup ACK															
111	等时传输结束															
[11:8]	EPSTS0	<div>端点 0 总线状态</div> <div>这些位用于表示该端点的当前状态。</div> <table><tr><th>EPSTS0</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr></table>	EPSTS0	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK		
EPSTS0	描述															
000	INACK															
001	保留															
010	OUT Packet Data0 ACK															
110	OUT Packet Data1 ACK															
011	Setup ACK															

		111	等时传输结束	
[7]	OVERRUN	Overrun 表示接收到数据是否大于有效负载最大值。 1 = 表明主机 Out 数据多于 MXPLD 寄存器中定义的最大负载或 Setup 数据多于 8 字节 0 = 没有溢出		
[6:0]	-	保留		

USB 缓存段寄存器 (USB BUFSEG)

仅用于 Setup 令牌

寄存器	偏移量	R/W	描述	复位后的值
USB_BUFSEG	USB_BA+0x018	R/W	Setup 令牌缓存段	0x0000_0000

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

Bits	描述	
[31:9]	-	保留
[8:3]	BUFSEG[5:0]	该寄存器仅用于 Setup 令牌。该部分用来表示 Setup 令牌相对 USB SRAM 起始地址的偏移量。 Setup 令牌在 USB SRAM 中的物理起始地址是 USB_SRAM 地址 + { BUFSEG[5:0], 000} USB_SRAM 地址 = USB_BASE+0x100h
[2:0]	-	保留

USB 端点状态 2 寄存器 (USB EPSTS2)

寄存器	偏移量	R/W	描述	复位后的值
USB_EPSTS2	USB_BA+0x01C	R	端点总线状态	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
EPSTS7				EPSTS6			

Bits	描述															
[31:8]	-	保留														
[6:4]	EPSTS7	端点 7 总线状态 这些位用于表示该端点的当前状态。														
		<table><tr><th>EPSTS7</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS7	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
		EPSTS7	描述													
		000	INACK													
		001	保留													
		010	OUT Packet Data0 ACK													
		110	OUT Packet Data1 ACK													
		011	Setup ACK													
111	等时传输结束															
[2:0]	EPSTS6	端点 6 总线状态 这些位用于表示该端点的当前状态。														
		<table><tr><th>EPSTS6</th><th>描述</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>保留</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>等时传输结束</td></tr></table>	EPSTS6	描述	000	INACK	001	保留	010	OUT Packet Data0 ACK	110	OUT Packet Data1 ACK	011	Setup ACK	111	等时传输结束
		EPSTS6	描述													
		000	INACK													
		001	保留													
		010	OUT Packet Data0 ACK													
		110	OUT Packet Data1 ACK													
		011	Setup ACK													
111	等时传输结束															

USB 缓存段寄存器 (USB BUFSEGx) x = 0~7

寄存器	偏移量	R/W	描述	复位后的值
USB_BUFSEG0	USB_BA+0x020	R/W	端点 0 缓存段	0x0000_0000
USB_BUFSEG1	USB_BA+0x030	R/W	端点 1 缓存段	0x0000_0000
USB_BUFSEG2	USB_BA+0x040	R/W	端点 2 缓存段	0x0000_0000
USB_BUFSEG3	USB_BA+0x050	R/W	端点 3 缓存段	0x0000_0000
USB_BUFSEG4	USB_BA+0x060	R/W	端点 4 缓存段	0x0000_0000
USB_BUFSEG5	USB_BA+0x070	R/W	端点 5 缓存段	0x0000_0000
USB_BUFSEG6	USB_BA+0x080	R/W	端点 6 缓存段	0x0000_0000
USB_BUFSEG7	USB_BA+0x090	R/W	端点 7 缓存段	0x0000_0000

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

Bits	描述	
[31:9]	-	保留
[8:3]	BUFSEG[5:0]	该部分用来定义每个端点相对USB SRAM 起始地址的偏移地址。端点的起始物理地址是 USB_SRAM 地址 + { BUFSEG[5:0], 000} USB_SRAM 地址 = USB_BASE+0x100h. 参考章节 5.4.3.3 中关于端点 SRAM 结构与描述
[2:0]	-	保留

USB 最大负载寄存器 (USB_MXPLDx) x = 0~7

寄存器	偏移量	R/W	描述	复位后的值
USB_MXPLD0	USB_BA+0x024	R/W	端点 0 最大负载	0x0000_0000
USB_MXPLD1	USB_BA+0x034	R/W	端点 1 最大负载	0x0000_0000
USB_MXPLD2	USB_BA+0x044	R/W	端点 2 最大负载	0x0000_0000
USB_MXPLD3	USB_BA+0x054	R/W	端点 3 最大负载	0x0000_0000
USB_MXPLD4	USB_BA+0x064	R/W	端点 4 最大负载	0x0000_0000
USB_MXPLD5	USB_BA+0x074	R/W	端点 5 最大负载	0x0000_0000
USB_MXPLD6	USB_BA+0x084	R/W	端点 6 最大负载	0x0000_0000
USB_MXPLD7	USB_BA+0x094	R/W	端点 7 最大负载	0x0000_0000

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

Bits	描述	
[31:9]	-	保留
[8:0]	MXPLD[8:0]	<p>最大负载</p> <p>用于定义发送到主机 (IN token) 的数据长度或从主机接收到的 (OUT token) 实际数据长度。也用于表示端点是否准备好发送 (IN token) 或接收 (OUT token)。</p> <p>(1). 当 CPU 写该寄存器</p> <p>IN token, MXPLD 的值用于定义要发送的数据长度并表示数据缓存以已经准备好了。</p> <p>OUT token, 表示控制器准备接收主机的数据, MXPLD 的值表示从主机发送过来的最大数据长度。</p> <p>(2). 当 CPU 读该寄存器</p> <p>IN token, MXPLD 的值表示发生到主机的数据长度</p> <p>OUT token, MXPLD 的值表示从主机接收到的实际数据长度</p> <p>注: 一旦 MXPLD 被写, 数据包将在 IN/OUT token 到达后立即发送/接收。</p>

USB 配置寄存器 (USB_CFGx) x = 0~7

寄存器	偏移量	R/W	描述	复位后的值
USB_CFG0	USB_BA+0x028	R/W/C	端点 0 配置寄存器	0x0000_0000
USB_CFG1	USB_BA+0x038	R/W/C	端点 1 配置寄存器	0x0000_0000
USB_CFG2	USB_BA+0x048	R/W/C	端点 2 配置寄存器	0x0000_0000
USB_CFG3	USB_BA+0x058	R/W/C	端点 3 配置寄存器	0x0000_0000
USB_CFG4	USB_BA+0x068	R/W/C	端点 4 配置寄存器	0x0000_0000
USB_CFG5	USB_BA+0x078	R/W/C	端点 5 配置寄存器	0x0000_0000
USB_CFG6	USB_BA+0x088	R/W/C	端点 6 配置寄存器	0x0000_0000
USB_CFG7	USB_BA+0x098	R/W/C	端点 7 配置寄存器	0x0000_0000

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

Bits	描述	
[31:110]	-	保留
[9]	SSTALL	设置 STALL 响应 1 = 设置设备自动应答 STALL 0 = 禁用设备应答 STALL
[8]	CSTALL	清除 STALL 响应 1 = 清除设备在 setup 阶段响应 STALL，此后设备将会向主机回 ACK/NAK。 0 = 不清除设备在 setup 阶段响应 STALL，设备继续向主机回 STALL。
[7]	DSQ_SYNC	数据序列同步 1 = DATA1 PID 0 = DATA0 PID 指明在当前的传输中是 DATA0 或 DATA1 PID。对于 IN token，在主机应答 ACK 后，它将自动切换。在其他 token，用户必须处理该位，以保证 PID 的正确性。
[6:5]	EPMODE[1:0]	端点模式

		EPMODE	描述
		00	端点禁用
		01	Out 端点
		10	IN 端点
		11	未定义
[4]	ISOCH	等时 端点 该位用来设置端点为等时端点，无握手。	
[3:0]	EP_NUM[3:0]	端点号 用于定义当前端点的端点号。	

USB PDMA 控制寄存器 (USB_PDMA)

寄存器	偏移量	R/W	描述	复位后的值
USB_PDMA	USB_BA+0x0A4	R/W	PDMA 控制寄存器	0x0000_0000

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

Bits	描述	
[31:2]	-	保留
[3]	PDMA_RST	PDMA 复位 该位用于复位 USB PDMA 功能到默认状态。 1 = 复位 PDMA 功能 0 = PDMA 复位功能禁用 注：在复位功能完成后，该位自动清零。
[2]	BYTEM	CPU 访问 USB SRAM 大小模式选择 1 = 字节模式：从 CPU 到 SUB SRAM 传输的大小是字节排序 0 = 字模式：从 CPU 到 SUB SRAM 传输的大小是字排序
[1]	PDMA_TRG	激活 PDMA 功能 1 = USB PDMA 功能激活 0 = PDMA 功能没有激活 在 PDMA 传输结束后，该位自动清除。
[0]	PDMA_RW	PDMA_RW 1 = PDMA 将从 USB 缓存读数据到内存 0 = PDMA 将从内存读数据到 USB 缓存

5.21 看门狗定时器控制器

5.21.1 概述

看门狗定时器的用途是在软件出问题执行系统复位功能，这可以防止系统无限期地挂起。除此之外，看门狗定时器还支持将 CPU 从掉电模式唤醒的功能。看门狗定时器包含一个 18 位的自由运行

计数器，定时溢出间隔可编程。

5.2.1.2 特征

- 18-位自由运行 WDT 计数器用于看门狗定时器超时间隔。
- 可选的超时间隔 ($2^4 \sim 2^{18}$)，超时间隔为 104 ms ~ 26.316 s (如果 WDT_CLK = 10 kHz)。
- 复位周期 = $(1 / 10 \text{ kHz}) * 63$ ，如果 WDT_CLK = 10 kHz。

5.2.1.3 框图

看门狗定时器时钟控制和框图如下所示：

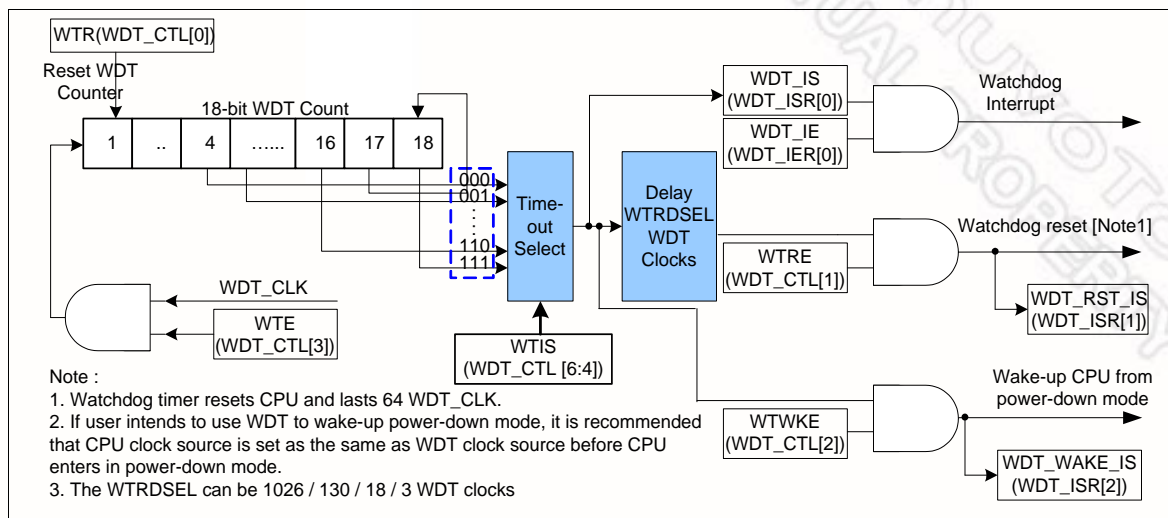


图 5-127 看门狗定时器框图

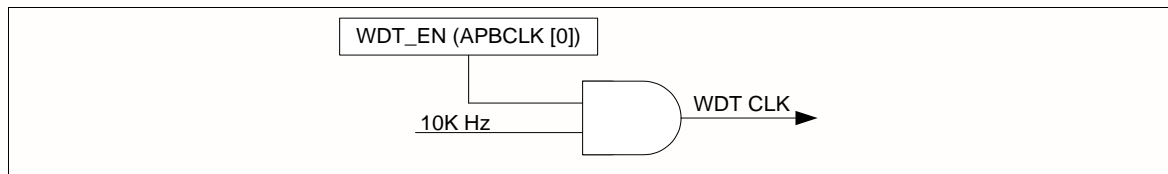


图 5-128 看门狗定时器时钟控制图

5.2.1.4 功能描述

看门狗定时器的用途是在软件出问题执行系统复位功能，这可以防止系统无限期地挂起。除此之外，看门狗定时器还支持将 CPU 从掉电模式唤醒的功能，此外当 CPU 进入掉电模式时，WDT 计数器会自动重置。看门狗定时器包含一个 18 位的自由运行计数器，定时溢出间隔可编程。接下来的表格给出了看门狗超时间隔选择，而接下来的图片给出了看门狗中断信号和复位信号的时序。

设置 WTE (WDT_CTL[3]) 使能看门狗定时器和 WDT 计数器开始计数。当计数器达到选择的超时间隔，看门狗定时器中断标志 WDT_IS 将被立即被置位，并请求 WDT 中断 (如果看门狗定时器中断使能位 WDT_IE 置位)，同时紧接着超时事件会有一个指定的延时可透过 WTRDSEL 设置，用户必须在该指定延时过期前设置 WTR (WDT_CTL[0]) (看门狗定时器复位) 为高来复位 18 位 WDT 计数器，以防止芯片复位。WTR 位在 WDT 计数器复位后由硬件自动清零。通过设置 WTIS (WDT_CTL[6:4])，有 8 种带指定延时时间的超时间隔可供选择。如果在指定延迟时间过期后，WDT 计数器没有被清零，看门狗定时器将置位看门狗定时器复位标志 (WDT_RST_IS)，并复位

CPU。这个复位将持续 63 个 WDT 时钟 (T_{RST})，然后芯片重启，并从复位向量 (0x0000_0000) 处重新开始执行程序。WDT_RST_IS 不会被看门狗复位清零。用户可用软件轮询WDT_RST_IS 来识别复位源是否 WDT。

WTIS	WTR Time-out Interval	Interrupt Period T_{INT}	Time-out Interval WDT_CLK = 10K Hz T_{TIS}	Reset Interval WDT_CLK = 10K Hz T_{WTR}
000	$2^4 * T_{WDT}$	$1024 * T_{WDT}$	1.6 ms	104 ms
001	$2^6 * T_{WDT}$	$1024 * T_{WDT}$	6.4 ms	108.8 ms
010	$2^8 * T_{WDT}$	$1024 * T_{WDT}$	25.6 ms	128 ms
011	$2^{10} * T_{WDT}$	$1024 * T_{WDT}$	102.4 ms	204.8 ms
100	$2^{12} * T_{WDT}$	$1024 * T_{WDT}$	407 ms	512 ms
101	$2^{14} * T_{WDT}$	$1024 * T_{WDT}$	1.638 s	1.741 s
110	$2^{16} * T_{WDT}$	$1024 * T_{WDT}$	6.553 s	6.656 s
111	$2^{18} * T_{WDT}$	$1024 * T_{WDT}$	26.214 s	26.316 s

表 5-16 看门狗超时时间间隔选择

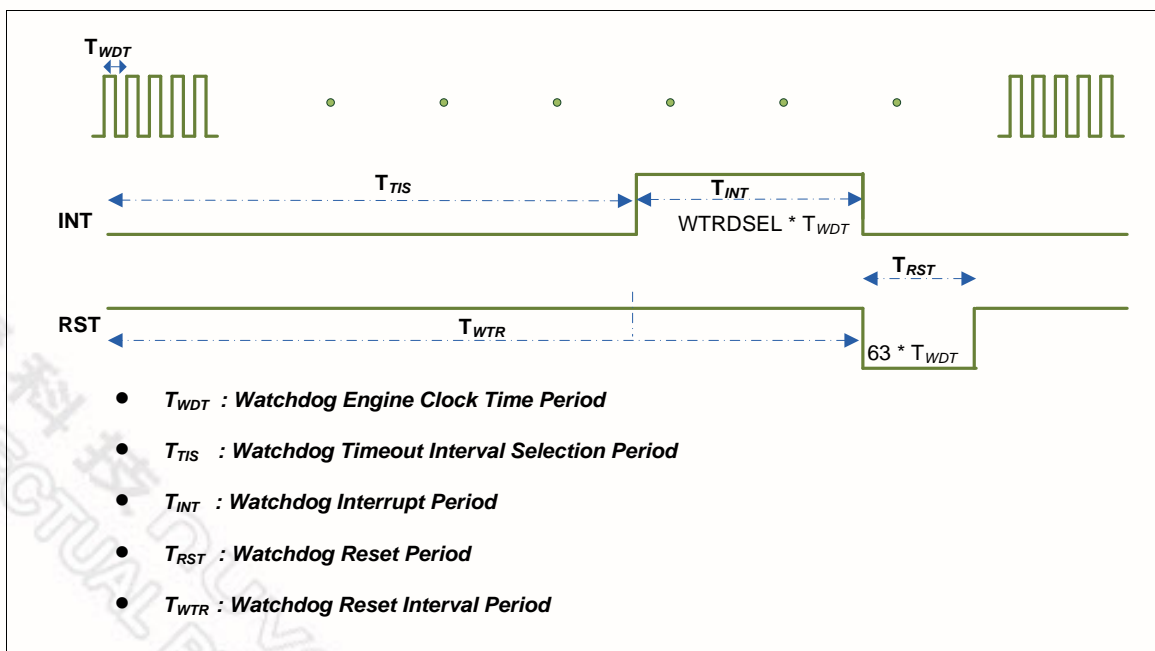


图 5-129 看门狗中断和复位信号时序

5.21.5 寄存器和存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
WDT Base Address : WDT_BA = 0x4000_4000				
WDT_CTL	WDT_BA0+0x00	R/W	看门狗定时器控制寄存器	0x0000_0070
WDT_IER	WDT_BA0+0x04	R/W	看门狗定时器中断使能寄存器	0x0000_0000
WDT_ISR	WDT_BA0+0x08	R/W	看门狗定时器中断状态寄存器	0x0000_0000

5.21.6 寄存器 描述

看门狗控制寄存器(WDT CTL)

寄存器	偏移量	R/W	描述	复位后的值
WDT_CTL	WDT_BA+0x00	R/W	看门狗定时器控制寄存器	0x0000_0070

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

Bits	描述											
[31:10]	-	保留.										
[9:8]	WTRDSEL[1:0]	<p>看门狗定时器复位延时选择（写保护位）</p> <p>当看门狗定时器超时发生，软件有一个叫看门狗复位延时周期的时间来清除看门狗定时器防止复位发生。软件可以为不同的看门狗超时周期选择一个合适的看门狗复位延时周期。</p> <table><tr><th>WTRDSEL[1:0]</th><th>描述</th></tr><tr><td>00</td><td>看门狗复位延时周期是 1026 个看门狗时钟</td></tr><tr><td>01</td><td>看门狗复位延时周期是 130 个看门狗时钟</td></tr><tr><td>10</td><td>看门狗复位延时周期是 18 个看门狗时钟</td></tr><tr><td>11</td><td>看门狗复位延时周期是 3 个看门狗时钟</td></tr></table> <p>如果看门狗复位这个寄存器也会复位</p>	WTRDSEL[1:0]	描述	00	看门狗复位延时周期是 1026 个看门狗时钟	01	看门狗复位延时周期是 130 个看门狗时钟	10	看门狗复位延时周期是 18 个看门狗时钟	11	看门狗复位延时周期是 3 个看门狗时钟
WTRDSEL[1:0]	描述											
00	看门狗复位延时周期是 1026 个看门狗时钟											
01	看门狗复位延时周期是 130 个看门狗时钟											
10	看门狗复位延时周期是 18 个看门狗时钟											
11	看门狗复位延时周期是 3 个看门狗时钟											
[7]	-	保留										
[6:4]	WTIS[2:0]	<p>看门狗定时器间隔选择（写保护位）</p> <p>这三位选择看门狗定时器的超时间隔，该计数器运行于自由计数模式 请参考 表 5-16</p>										
[3]	WTE	<p>看门狗定时器使能（写保护位）</p> <p>1 = 使能看门狗定时器.</p>										

		0 = 禁止看门狗定时器 (该操作会复位内部计数器).
[2]	WTWKE	看门狗定时器唤醒功能使能 (写保护位) 1 = 使能唤醒功能, 看门狗定时器超时可以把 CPU 从掉电模式唤醒 0 = 禁止看门狗定时器唤醒 CPU 功能
[1]	WTRE	看门狗定时器复位功能使能 (写保护位) 置位该位将使能看门狗定时器复位功能. 1 = 使能看门狗定时器复位功能 0 = 禁止看门狗定时器复位功能
[0]	WTR	清零看门狗定时器 (写保护位) 置位该位将清零看门狗定时器. 1 = 复位看门狗定时器的计数器值 0 = 写 0 到该位没有影响 注: 该位在一些时钟周期后会被自动清零

看门狗中断使能寄存器(WDT_IER)

寄存器	偏移量	R/W	描述	复位后的值
WDT_IER	WDT_BA+0x04	R/W	看门狗定时器中断使能寄存器	0x0000_0000

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

Bits	描述	
[31:1]	-	保留.
[0]	WDT_IE	看门狗定时器中断使能 1 = 使能看门狗定时器中断 0 = 禁止看门狗定时器中断

看门狗定时器中断状态寄存器(WDT_ISR)

寄存器	偏移量	R/W	描述	复位后的值
WDT_ISR	WDT_BA+0x08	R	看门狗定时器中断状态寄存器	0x0000_0000

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

Bits	描述	
[31:3]	-	保留。
[2]	WDT_WAKE_IS	<p>看门狗定时器唤醒状态 (只读)</p> <p>如果看门狗定时器导致系统从掉电模式唤醒，该位将会被置位。该位必须由软件向该位写 1 清除。</p> <p>1 = 看门狗定时器超时把系统从掉电模式唤醒</p> <p>0 = 看门狗定时器没有导致系统唤醒。</p> <p>注 1：当系统处于掉电模式，且看门狗超时，硬件将置位 WDT_WAKE_IS 和 WDT_IS。</p> <p>注 2：在一个 WDT 时钟过后，该位可以通过写 1 清除。</p>
[1]	WDT_RST_IS	<p>看门狗定时器复位状态 (只读)</p> <p>当看门狗定时器引发了一次复位，硬件将置位该位，该标志可以被软件读取，用于决定 CPU 的复位源。软件负责通过向该位写 1 来手动清除该标志。如果 WDTRE 被禁止，看门狗定时器对该位不产生影响。</p> <p>1 = 发生了看门狗定时器复位</p> <p>0 = 看门狗定时器复位没有发生</p> <p>注：该位只读，但是可以通过向该位写 1 来清除该标志。</p>
[0]	WDT_IS	<p>看门狗定时器中断状态 (只读)</p> <p>如果看门狗定时器中断被使能，硬件会置位该位来表明发生了看门狗定时器中断。如果看门狗定时器中断没有被使能，该位表明发生了超时。</p> <p>1 = 发生了看门狗定时器中断</p> <p>0 = 看门狗定时器中断没有发生</p> <p>注：该位只读，但是可以通过向该位写 1 清除该标志。</p>

Revision 1.07

PERIODSEL	预分频值	超时周期	超时间隔 12M/2048 = 5.859 kHz WWDT_CLK=5.859 kHz
0000	1	$1 * 64 * T_{WWDT}$	10.9 ms
0001	2	$2 * 64 * T_{WWDT}$	21.8 ms
0010	4	$4 * 64 * T_{WWDT}$	43.7 ms
0011	8	$8 * 64 * T_{WWDT}$	87.4 ms
0100	16	$16 * 64 * T_{WWDT}$	174.7 ms
0101	32	$32 * 64 * T_{WWDT}$	349.5 ms
0110	64	$64 * 64 * T_{WWDT}$	699.1 ms
0111	128	$128 * 64 * T_{WWDT}$	1.3981 s
1000	192	$192 * 64 * T_{WWDT}$	2.0971 s
1001	256	$256 * 64 * T_{WWDT}$	2.7962 s
1010	384	$384 * 64 * T_{WWDT}$	4.1943 s
1011	512	$512 * 64 * T_{WWDT}$	5.5924 s
1100	768	$768 * 64 * T_{WWDT}$	8.3886 s
1101	1024	$1024 * 64 * T_{WWDT}$	11.1848 s
1110	1536	$1536 * 64 * T_{WWDT}$	16.7772 s
1111	2048	$2048 * 64 * T_{WWDT}$	22.3696 s

表 5-17 窗口看门狗预分频值选择

窗口看门狗定时器可以通过设置WWDTEN (WWDTCR[0])为1使能。当窗口看门狗定时器使能，下数计数器开始从0x3F下数，不能被软件停止。

在. WWDT下数期间，如果计数器值等于窗口看门狗定时器比较值WINCMP (WWDTCR[21:16])且INTEN(WWDT_IER[0])置1，会发生中断。如果WWDT计数器值到0，WWDT发生复位。在WWDT下数到0之前，软件可以写特定值(0x00005AA5) 到寄存器WWDTRLD来重新加载0x3F到WWDT计数器防止WWDT复位发生。这个重新加载的动作仅在WWDT计数器值小于或等于WINCMP有效。如果软件写WWDTRLD 在WWDT计数器大于WINCMP期间，会导致芯片复位。

当软件写特定值(0x00005AA5)到寄存器 WWDTRLD重载WWDT计数器，需要3个窗口看门狗时钟来同步重载命令实现真正执行重载操作。这意味着如果软件设置窗口时钟分频器为1，比较值WINCMP (WWDTCR[21:16])应该大于2或软件不能在WWDT复位之前重载WWDT计数器。

为防止程序跑到非预期代码禁止窗口看门狗，控制寄存器WWDTCR 和 WWDT_IER在芯片上电或复位之后仅能写一次。一旦窗口看门狗被软件使能，软件不能禁止窗口看门狗，改变预分频周期或改变窗口比较值，除非芯片复位。当CPU在睡眠模式(WFI 或 WFE)，CPU会停止但系统时钟会继续，所以窗口看门狗定时器会仍然计数，但是当系统进入掉电模式，系统时钟也会禁止，所以窗口看门狗时钟会停止。

5.22.5 寄存器存储器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
WWDT Base Address : 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
WWDT_IER	WWDT_BA+0x08	R/W	窗口看门狗定时器中断使能寄存器	0x0000_0000
WWDTSTS	WWDT_BA+0x0C	R/W	窗口看门狗定时器状态 寄存器	0x0000_0000
WWDTVAL	WWDT_BA+0x10	R	窗口看门狗计数器值 寄存器	0x0000_003F

5.22.6 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	复位后的值
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							

Bits	描述
[31:0]	<p>窗口看门狗定时器重载计数器寄存器</p> <p>写 0x00005AA5 到该寄存器 会重载窗口看门狗定时器计数器值为 0x3F.</p> <p>注：软件仅能写WWDTRLD 当 WWDT计数器值在 0 和 WINCMP之间。如果软件写 WWDT 当 WWDT 计数器值大于 WINCMP，WWDT会产生 RESET 信号。.</p>

窗口看门狗定时器控制寄存器 (WWDTCR)

寄存器	偏移量	R/W	描述	复位后的值
WWDTCR	WWDT_BA+0x04	R/W	窗口看门狗定时器控制寄存器	0x003F_0800

注: 芯片上电或复位后, 该寄存器只能被写一次.

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

Bits	描述	
[31]	DBGEN	WWDT Debug 使能 1 = WWDT在 debug 模式仍然计数. 0 = WWDT在 debug 模式停止计数
[30:22]	-	保留.
[21:16]	WINCMP[5:0]	WWDT窗口比较 寄存器 设置该寄存器调整有效的重载窗口. 注: 软件仅能写WWDTRL D 当 WWDT计数器值在 0 和 WINCMP之间. 如果软件写WWDTRL D 当 WWDT 计数器值大于 WINCMP, WWDT会产生 RESET 信号. .
[15:12]	-	保留.
[11:8]	PERIODSEL[3:0]	WWDT 预分频周期选择 这三个位为WWDT计数周期选择预分频 请参考 表 5-17
[7:1]	-	保留.
[0]	WWDTEN	窗口看门狗使能 设置该位使能窗口看门狗定时器. 1 = 使能窗口看门狗定时器功能 0 =禁止窗口看门狗定时器功能

窗口看门狗定时器中断使能寄存器 (WWDT_IER)

寄存器	偏移量	R/W	描述	复位后的值
WWDT_IER	WWDT_BA+0x08	R/W	窗口看门狗定时器中断使能寄存器	0x0000_0000

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

注: 芯片上电或复位后, 该寄存器只能被写一次..

Bits	描述	
[31:1]	-	保留.
[0]	WWDTIE	WWDT中断使能 设置该位使能窗口看门狗定时器中断功能. 1 =使能窗口看门狗定时器中断功能 0 =禁止窗口看门狗定时器中断功能

窗口看门狗定时器状态寄存器 (WWDTSTS)

寄存器	偏移量	R/W	描述	复位后的值
WWDTSTS	WWDT_BA+0x0C	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
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-						WWDTRF	WWDTIF

Bits	描述	
[31:2]	-	保留.
[1]	WWDTRF	WWDT 复位标志 当WWDT计数器下数到0或当 WWDT 计数器值大于 WINCMP写WWDTRLD，芯片会复位且该位会置1。软件写1清0该位。
[0]	WWDTIF	WWDT 比较匹配中断标志 当 WWCMP 和 WWDT 计数器匹配，该位置1，软件写1清0该位。

窗口看门狗定时器计数器值寄存器 (WWDTVAL)

寄存器	偏移量	R/W	描述	复位后的值
WWDTVAL	WWDT_BA+0x10	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
-							
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-		WWDTVAL					

Bits	描述	
[31:6]	-	保留.
[5:0]	WWDTVAL[5:0]	WWDT计数器值 该寄存器表示窗口看门狗计数器值，该寄存器只读

6 ARM® CORTEX™-M0 内核

6.1 概述

Cortex™-M0 处理器是32位可配置的多级流水线RISC处理器。它有 AMBA、AHB-Lite 接口和嵌套向量中断控制器（NVIC），具有可选的硬件调试功能，可以执行Thumb指令，并与其它Cortex-M系列兼容。支持两种模式-Thread 模式与 Handler 模式。异常时系统进入 Handler 模式。从 Handler 模式返回时，执行异常返回。复位时系统进入Thread 模式。Thread 模式也可由异常返回时进入。下图为处理器的功能控制器。

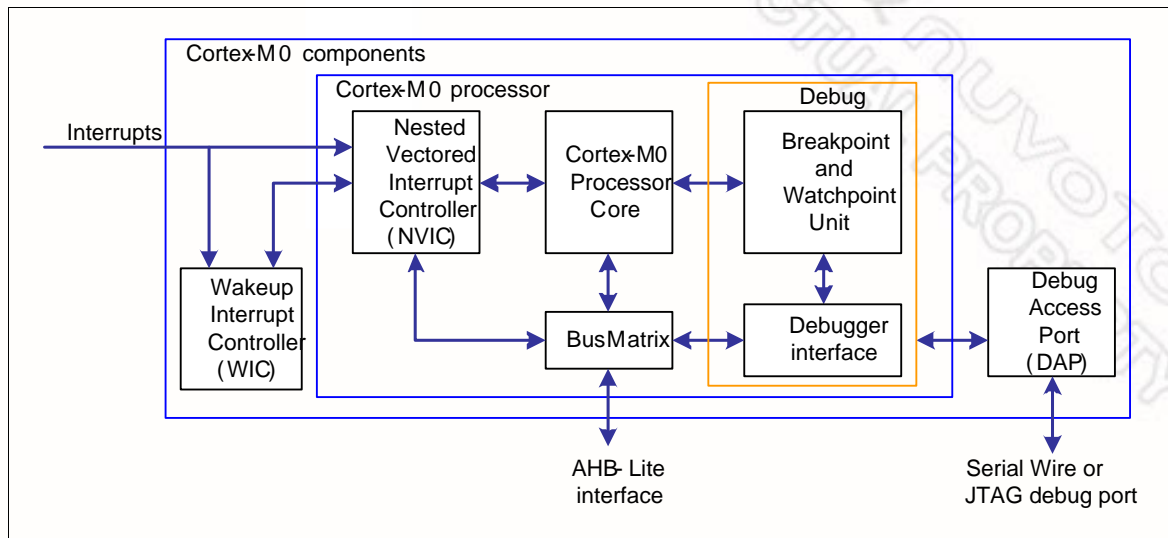


图 6-1 M0 功能模块

6.2 特性

- 低门数处理器特征：
 - ◆ ARMv6-M Thumb® 指令集
 - ◆ Thumb-2 技术
 - ◆ ARMv6-M 兼容 24-位 SysTick 定时器
 - ◆ 一个 32-位 硬件乘法器
 - ◆ 系统接口支持小端（little-endian）数据访问
 - ◆ 准确而及时的中断处理能力
 - ◆ 加载/存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
 - ◆ C 应用程序二进制接口的异常兼容模式（C-ABI）。这个 ARMv6-M C-ABI的兼容异常模式允许用户使用纯C函数实现中断处理
 - ◆ 使用中断唤醒（WFI）与事件唤醒（WFE）指令进入低功耗的休眠模式，或者从中断退出休眠模式
- NVIC 特性：

- ◆ 32个外部中断，每个中断具有4级优先级
- ◆ 专用的不可屏蔽中断 (NMI) 输入
- ◆ 支持电平和脉冲中断触发
- ◆ 中断唤醒控制器 (WIC)，支持极低功耗休眠模式
- 调试支持
 - ◆ 四个硬件断点
 - ◆ 两个观察点
 - ◆ 用于非侵入式代码分析的程序计数采样寄存器 (PCSR)
 - ◆ 单步和向量捕获能力
- 总线接口：
 - ◆ 通过简单的集成到所有系统外设和存储器的单一32位 AMBA-3 ABH-Lite 系统接口
 - ◆ 支持DAP (Debug Access Port) 的单一32位的从机端口

6.3 系统定时器(SysTick)

Cortex-M0 包含系统定时器 SysTick. SysTick 提供一个简单的24-位写清零、递减、自装载同时具有可灵活控制机制的计数器。该计数器有不同的使用方法。例如：

一个 RTOS 的滴答定时器，运行在可编程的速率（比如 100Hz），调用一个 SysTick 例程。

一个使用系统时钟的高速闹钟定时器。

一个可变速率的闹钟或信号定时器 – 持续时间范围取决于使用的参考时钟和计数器的动态范围。

一个简单的计数器。软件可以使用这个来测量任务完成时间。

一个内置的时钟源控制基于丢失/命中的持续时间。控制和状态寄存器的 COUNTFLAG 位域可以被用于决定在一定的设置时间内，作为一个动态时钟管理控制循环的一部分的某个动作是否完成，。

当系统定时器使能后，将从 SysTick 的当前值寄存器 (SYST_CVR) 的值向下计数到0，并在下一个时钟周期，重新加载 SysTick 重新加载值寄存器 (SYST_RVR) 的值。当计数器减到0时，标志位 COUNTFLAG置位，读 COUNTFLAG 位使其清零。

复位后，SYST_CVR 的值未知。使能前，软件应该向寄存器写入值清零。这样确保定时器以 SYST_RVR 的值计数，而非任意值。

若 SYST_RVR 为0，在重新加载后，定时器将保持当前值0。这个功能可以在计数器使能后用来禁用独立的功能。

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

6.3.1 系统定时器控制寄存器映射

R: read only, W: write only, R/W: both read and write, W&C: Write 1 clear

寄存器	偏移量	R/W	描述	复位后的值
SCS_BA = 0xE000_E000				

SYST_CTL	SCS_BA + 010	R/W	SysTick控制与状态	0x0000_0004
SYST_RVR	SCS_BA + 014	R/W	SysTick重新加载值	0xFFFF_FFFF
SYST_CVR	SCS_BA + 018	R/W	SysTick当前值	0xFFFF_FFFF

6.3.2 系统定时器控制寄存器描述

SysTick控制和状态 (SYST_CTL)

寄存器	偏移量	R/W	描述	复位后的值
SYST_CTL	SCS_BA + 0x10	R/W	SysTick控制和状态	0x0000_0004

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

Bits	描述	
[31:17]	-	保留
[16]	COUNTFLAG	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，COUNTFLAG 置位。 在读或者写当前值寄存器 (Current Value register)时，COUNTFLAG 被清零。
[15:3]	-	保留
[2]	CLKSRC	1: 如果没有外部时钟提供，内核时钟用作 SysTick，读取该位将为1，忽略写操作。 0: 时钟源为外部参考时钟（可选）
[1]	TICKINT	1: 向下计数到 0 将引起 SysTick 异常而挂起。软件清 SysTick 当前值寄存器 (SYST_CVR) 将不会导致 SysTick 挂起。 0: 向下计数到 0 不会引起 SysTick 异常而挂起。软件根据 COUNTFLAG 来确定是否已经发生计数到 0。
[0]	ENABLE	1: 计数器运行于连拍方式 (multi-shot manner) 0: 禁用计数器

SysTick重新加载值寄存器(SYST_RVR)

寄存器	偏移量	R/W	描述	复位后的值
SYST_RVR	SCS_BA + 0x14	R/W	SysTick重新加载值寄存器	0xFFFF_FFFF

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

Bits	描述	
[31:24]	-	保留。
[23:0]	RELOAD[23:0]	当计数器达到 0 时，这个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	SCS_BA + 0x18	R/W	SysTick当前值寄存器	0xFFFF_XXXX

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

Bits	描述	
[31:24]	-	保留
[23:0]	CURRENT[23:0]	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，该寄存器为写-清除 (write-clear)。软件写入任何值将清寄存器为 0。不支持位RAZ（读为 0，写忽略）（参见SysTick 重新加载值寄存器）。

6.4 系统控制寄存器

Cortex-M0 关键控制和状态特性由带有系统控制寄存器的系统控制模块集中管理。

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

6.4.1 系统控制寄存器映射

R: read only, W: write only, R/W: both read and write

寄存器	偏移量	R/W	描述	复位后的值
SCS_BA = 0xE000_E000				
CPUID	SCS_BA + 0xD00	R	CPUID 基础寄存器	0x410C_C200
ICSR	SCS_BA + 0xD04	R/W	中断控制状态寄存器	0x0000_0000
SCR	SCS_BA + 0xD10	R/W	系统控制寄存器	0x0000_0000
SHPR2	SCS_BA + 0xD1C	R/W	系统处理优先级寄存器2	0x0000_0000
SHPR3	SCS_BA + 0xD20	R/W	系统处理优先级寄存器3	0x0000_0000

6.4.2 系统控制寄存器描述

CPUID基础寄存器(CPUID)

寄存器	偏移量	R/W	描述	复位后的值
CPUID	SCS_BA + 0xD00	R	CPUID基础寄存器	0x 410CC200

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

Bits	描述	
[31:24]	IMPLEMENTER [7:0]	ARM 分配的执行码 (ARM = 0x41)
[23:20]	-	保留
[19:16]	PART[3:0]	ARMv6-M 读取值 0xC
[15:4]	PARTNO[11:0]	读取值 0xC20.
[3:0]	REVISION[3:0]	读取值 0x0

中断控制和状态寄存器(ICSR)

寄存器	偏移量	R/W	描述	复位后的值
ICSR	SCS_BA + 0xD04	R/W	中断控制和状态寄存器	0x 00000000

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

Bits	描述	
[31]	NMIPENDSET	设置该位将激活一个 NMI。因为NMI 是优先级最高的异常，它将在其一登记就激活。读返回当前状态（1：挂起；0：没有挂起）
[28]	PENDSVSET	设置挂起 PendSV中断。这一般用于请求上下文交换。读返回当前状态（1：挂起；0：没有挂起）
[27]	PENDSVCLR	写 1 清除一个挂起的 PendSV 中断
[26]	PENDSTSET	设置一个挂起的 SysTick。读返回当前状态（1：挂起；0：没有挂起）
[25]	PENDSTCLR	写 1 清除一个挂起的 SysTick。
[23]	ISRPREEMPT	若置位，则挂起的异常将从调试停止状态退出进入活动状态。
[22]	ISRPENDING	指示外部配置的中断 (NVIC 产生的) 是否挂起
[20:12]	VECTPENDING [8:0]	表示挂起异常中最高优先级异常的异常号。挂起状态包括内存映射使能和屏蔽寄存器的结果。它并不包括 PRIMASK 特定用途寄存器限定。0表示没有异常挂起。
[8:0]	VECTACTIVE[8:0]	0: Thread 模式 如果 VECTACTIVE > 1: 当前执行异常处理的异常号。

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	0x 00000000

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

Bits	描述	
[4]	SEVONPEND	当使能，中断从闲置到挂起的变换将包含在 WFE 指令的唤醒事件列表。
[2]	SLEEPDEEP	指示从睡眠中唤醒可能需要更长的时间。
[1]	SLEEPONEXIT	当设为1，内核进入睡眠状态异常返回到 Thread 模式。这是复位时的模式和异常水平，执行的基准面。

系统处理优先级寄存器2 (SHPR2)

寄存器	偏移量	R/W	描述	复位后的值
SHPR2	SCS_BA + 0xD1C	R/W	系统处理优先级寄存器2	0x 00000000

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

Bits	描述	
[31:30]	PRI_11[1:0]	系统处理的优先级 11 – SVCall “0” 表示最高优先级 & “3” 表示最低优先级

系统处理优先级寄存器3 (SHPR3)

寄存器	偏移量	R/W	描述	复位后的值
SHPR3	SCS_BA + 0xD20	R/W	系统处理优先级寄存器3	0x00000000

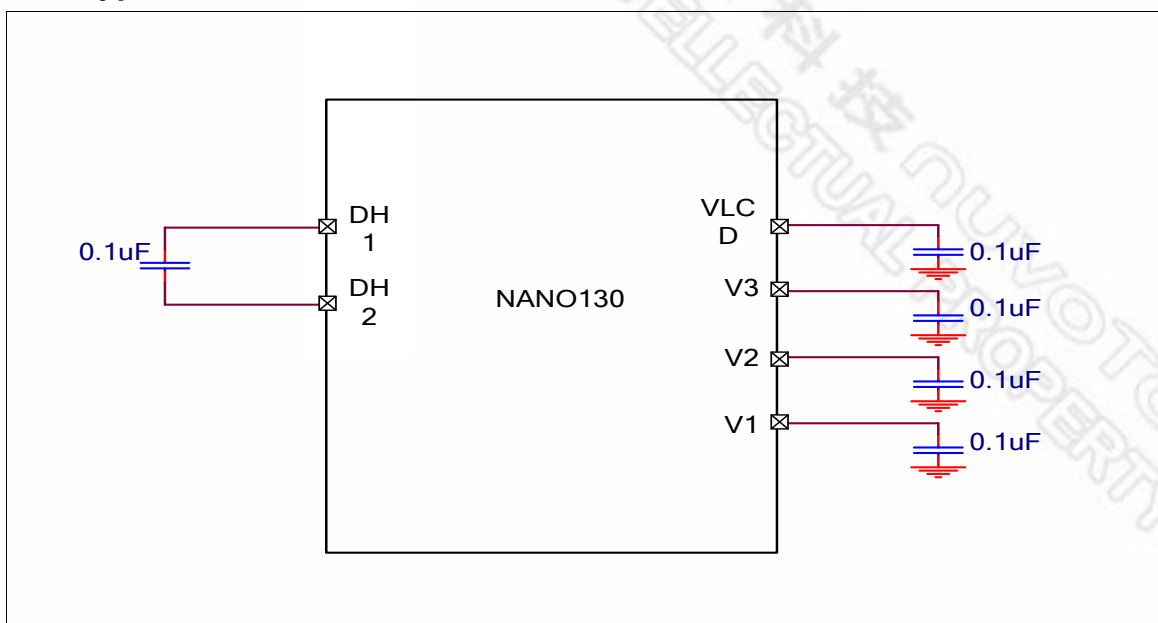
31	30	29	28	27	26	25	24
PRI_15		-					
23	22	21	20	19	18	17	16
PRI_14		-					
15	14	13	12	11	10	9	8
-							
7	6	5	4	3	2	1	0
-							

Bits	描述	
[31:30]	PRI_15[1:0]	系统处理的优先级 15 – SysTick “0”表示最高优先级 & “3”表示最低优先级
[23:22]	PRI_14[1:0]	系统处理的优先级 14 – PendSV “0”表示最高优先级 & “3”表示最低优先级

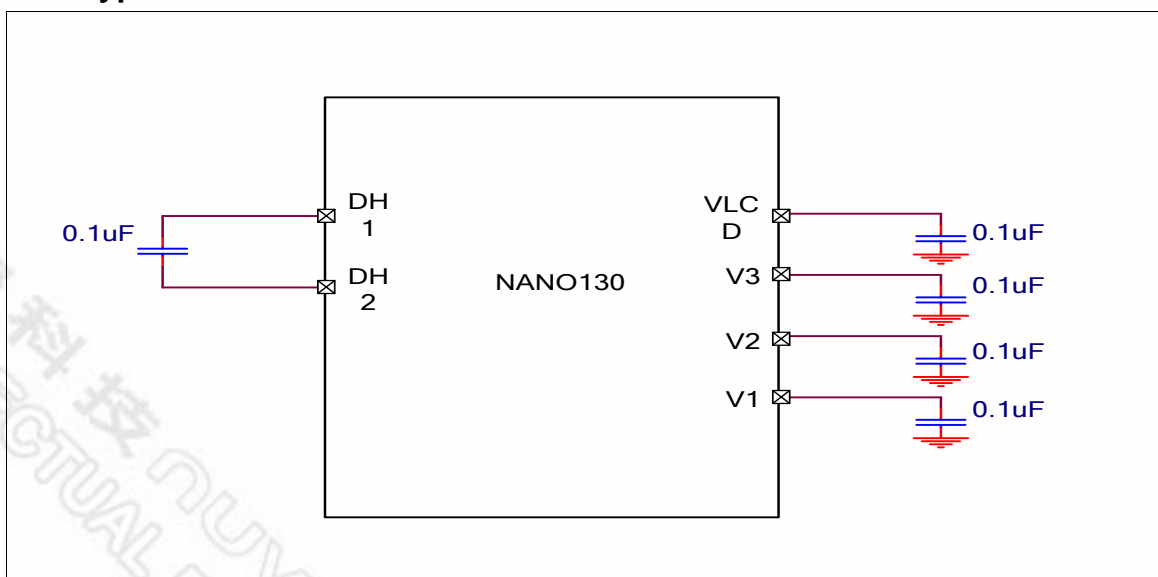
7 应用电路

7.1 LCD Charge Pump

7.1.1 C-type 1/3 Bias

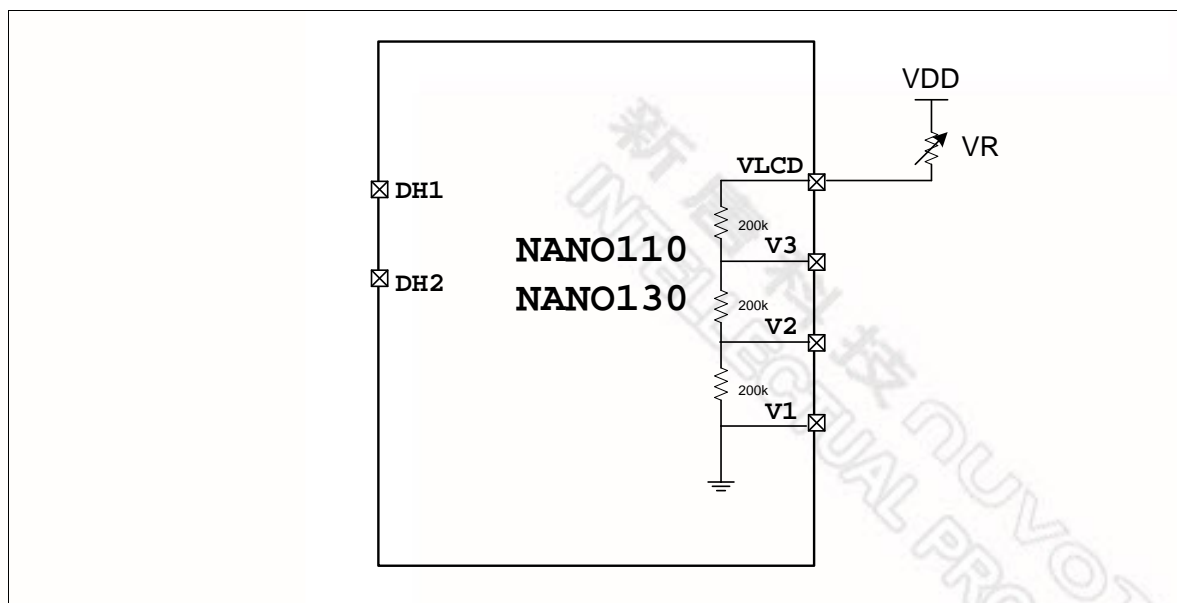


7.1.2 C-type 1/2 Bias



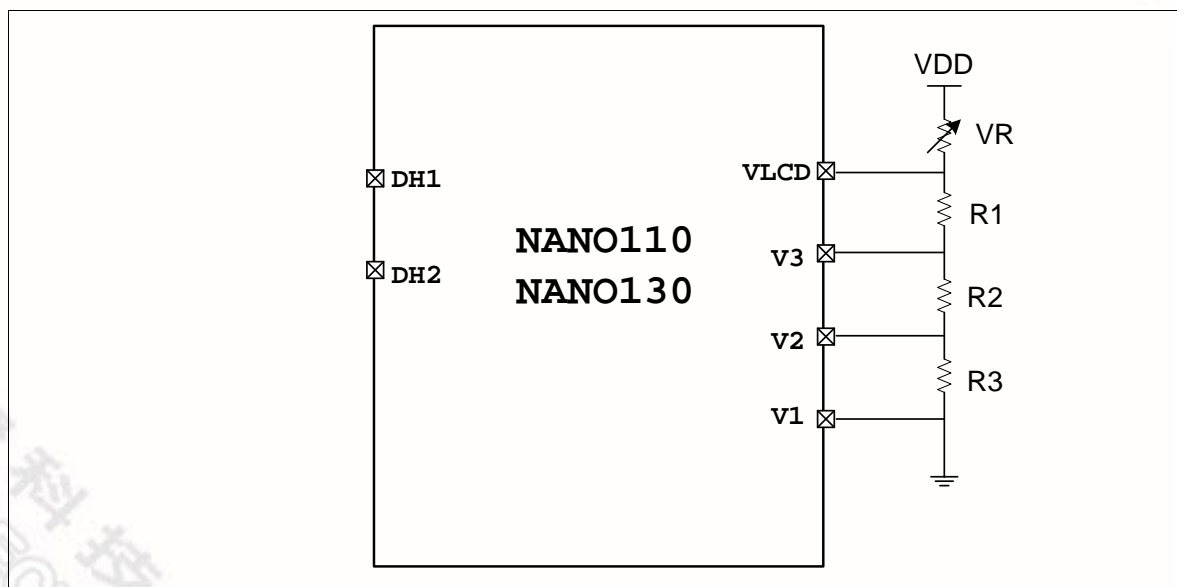
7.1.3 内部R-type

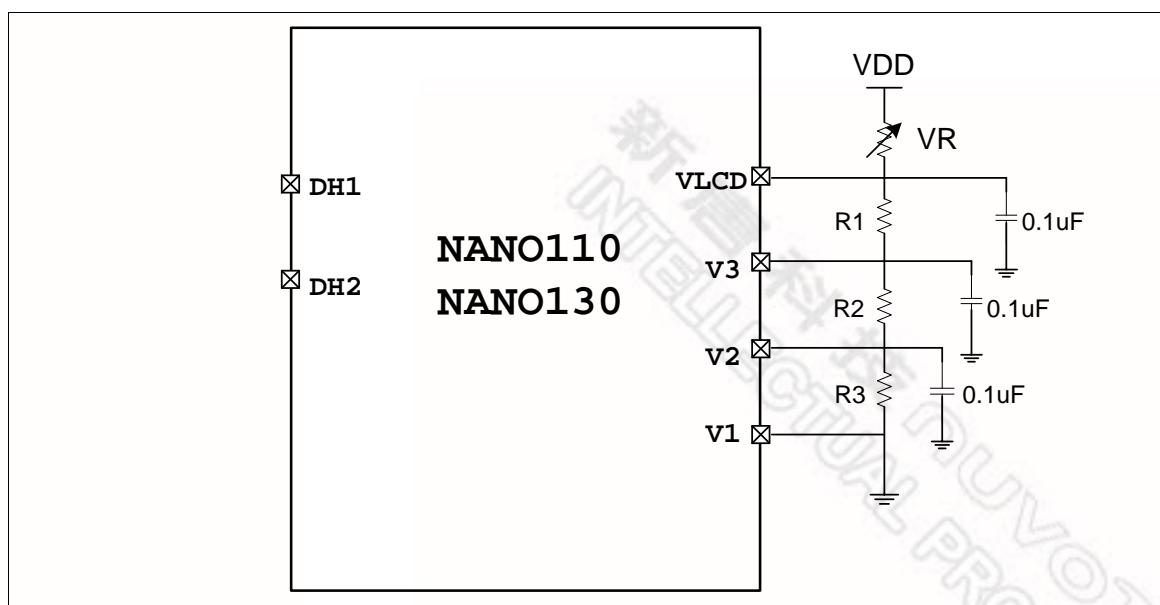
Nano110/130 系列芯片也提供外部R-type (旁通内部R-type)来降低电流消耗。对于外部R-type应用，VLCD一般与系统VDD连接或可透过外部可变电阻(VR)与VDD连接，其中可变电阻用来调整LCD的对比度。



7.1.4 外部R-type

为降低电流值，可以提升电阻梯之电阻值。当电阻梯之电阻值提升时，在某些点的对比度会受影响且其波形也会发生变动，因此根据玻璃面板像素的对比度与尺寸，选择约0.1uF 的电容值，且将其放置距离电阻梯较近的位置。

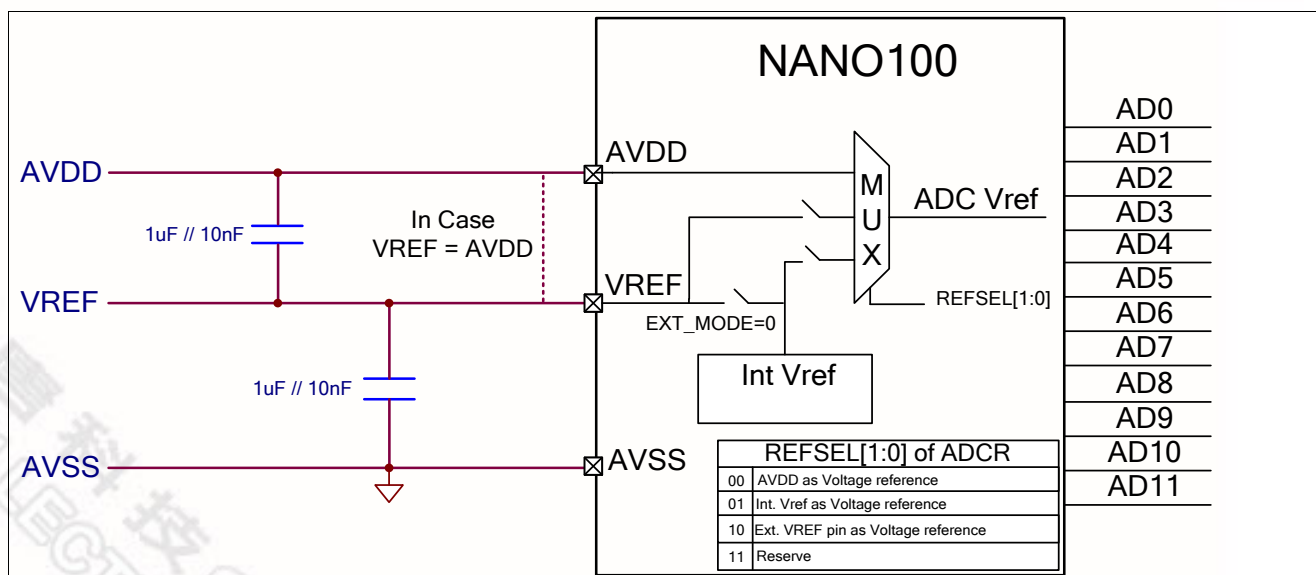




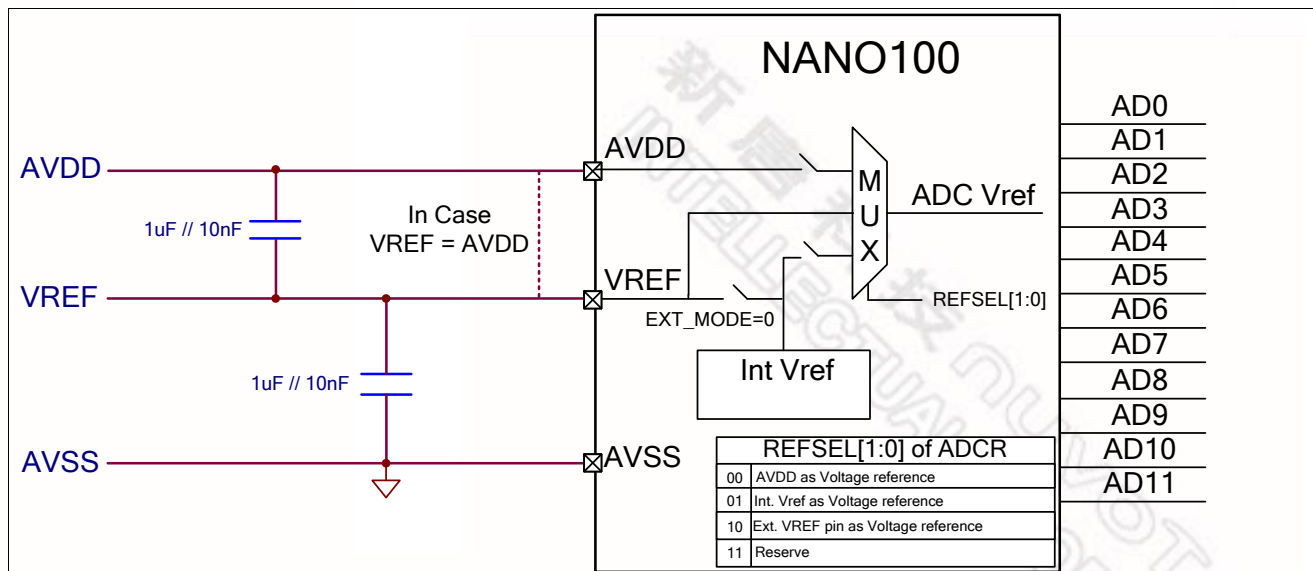
7.2 ADC 应用电路

7.2.1 电压参考源

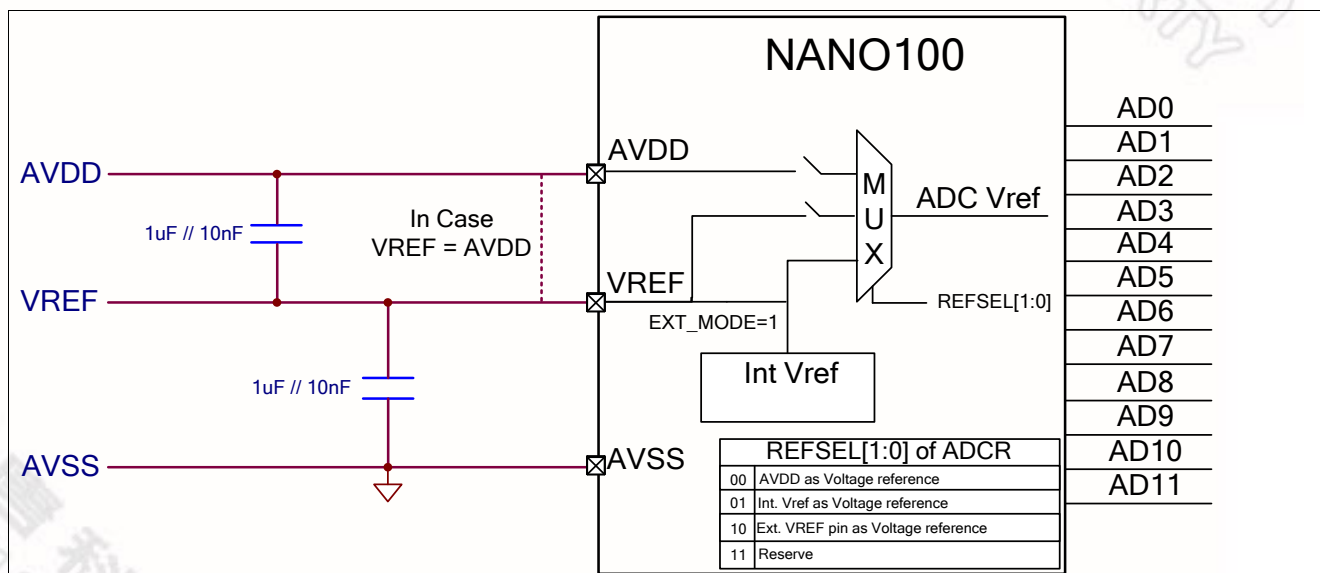
7.2.1.1 AVDD



7.2.1.2 Vref Pin



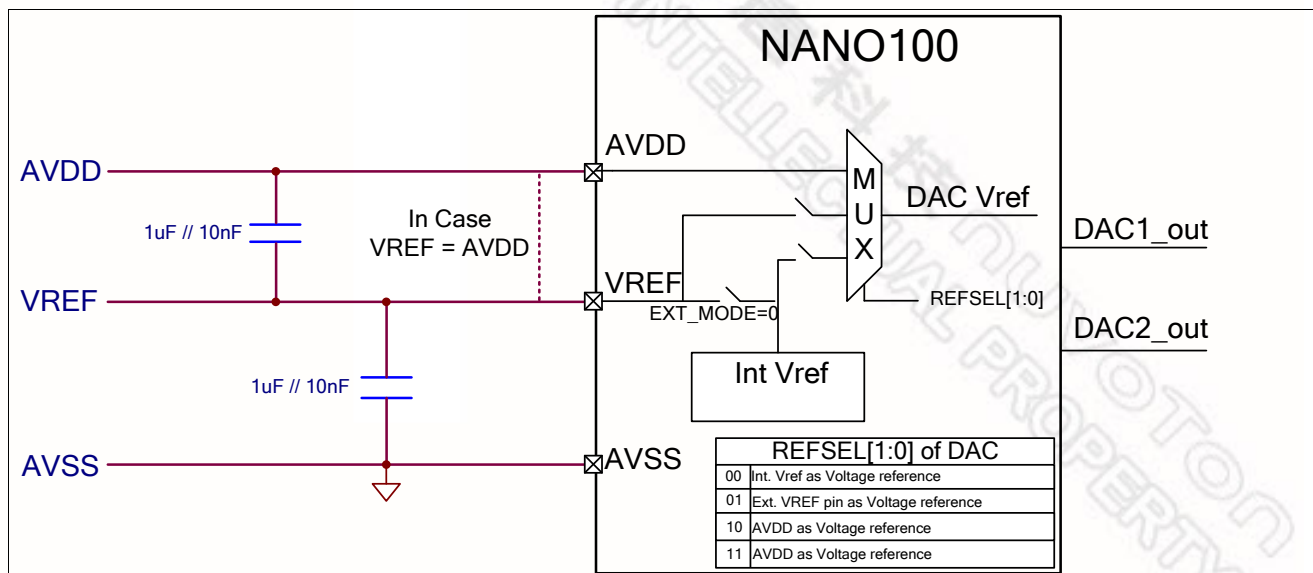
7.2.1.3 Int Vref



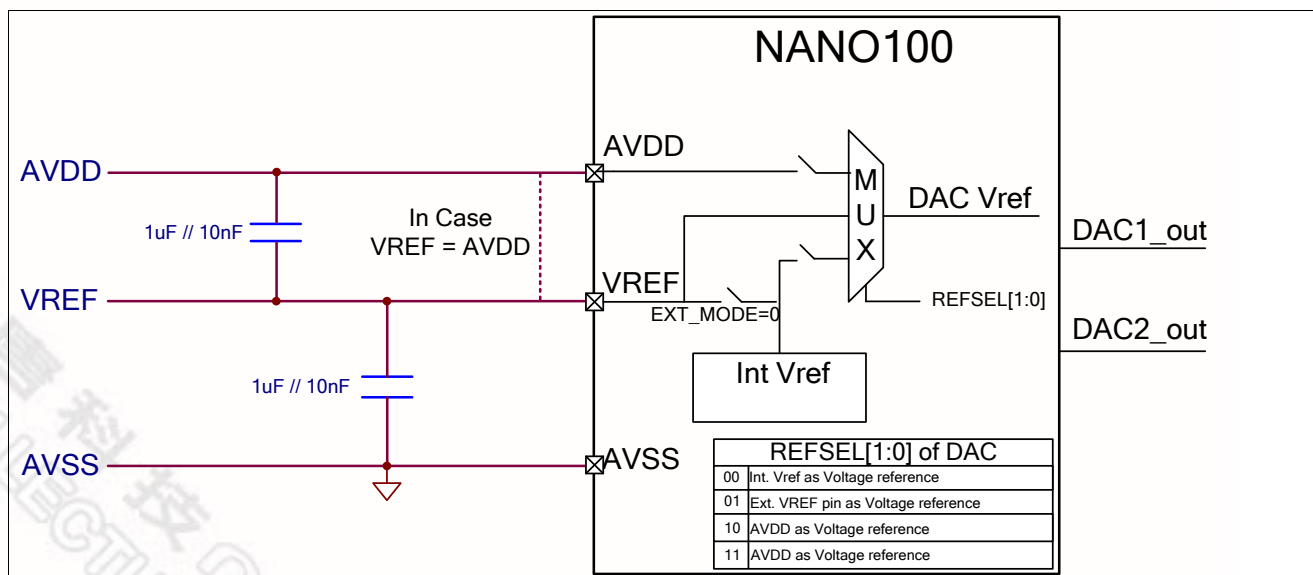
7.3 DAC 应用电路

7.3.1 电压参考源

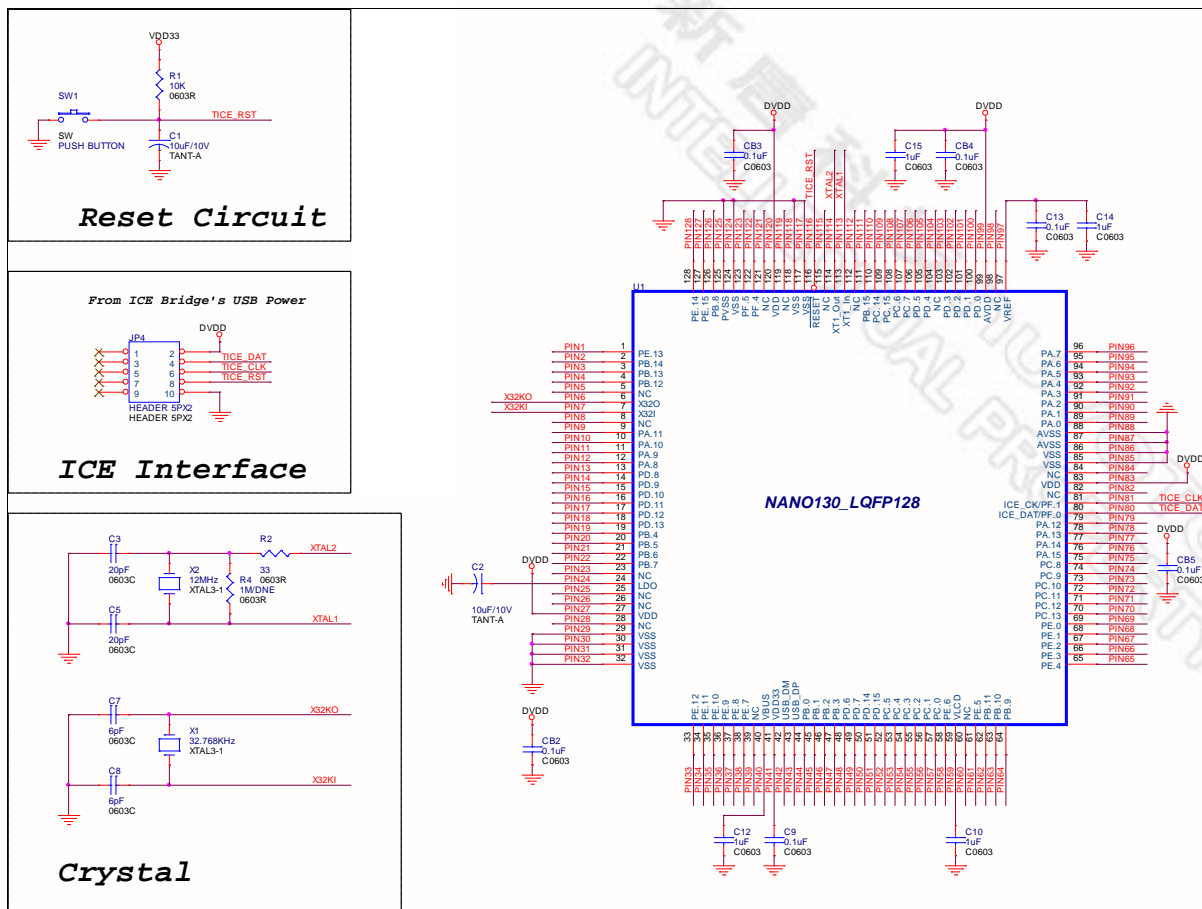
7.3.1.1 AVDD



7.3.1.2 Vref Pin



7.4 整个芯片应用电路



8 功耗

Part No	测试条件		VDD	CPU 时钟	电流
Nano100 (B) series 128KB Flash 16KB RAM	工作模式: CPU run while(1) in FLASH ROM Clock = 12MHz Crystal Oscillator 禁止所有外设		3.3V	12MHz	2.41mA 200uA/MHz
			1.8V	12MHz	N/A
	空闲模式: CPU stop Clock = 12MHz Crystal Oscillator 禁止所有外设		3.3V	12MHz	900uA 75uA/MHz
			1.8V	12MHz	N/A
	RTC + LCD 模式: (RAM 保持) (Power down with 32K and LCD enable) CPU stop Clock = 32.768KHz Crystal Oscillator 禁止除 RTC 和 LCD 电路外的所有外设 Without panel loading	C-type	3.3V	-	10uA
		Internal R-type (With 200kΩ Resistor ladder)			8.5uA
		External R-type (With 1MΩ Resistor ladder)			4.5uA
		C-type/R-type	1.8V	-	N/A
	RTC 模式: (RAM 保持) (Power down with 32K enable) CPU stop Clock = 32.768KHz Crystal Oscillator 禁止除 RTC 电路外的所有外设		3.3V	-	2.5uA
			1.8V	-	2.0uA
	掉电模式: (RAM 保持) CPU 和所有时钟停止		3.3V	-	1uA
			1.8V	-	0.8uA
	从掉电模式唤醒		3.3V	7us	N/A

9 电气特性

参数	符号	最小值	最大值	单位
直流电源电压	$V_{DD}-V_{SS}$	-0.3	+4.0	V
5V 管脚上的容忍输入电压	V_{IN}	$V_{SS} - 0.3$	$V_{DD} + 3.7$	V
非 5V 管脚上的容忍输入电压	V_{IN}	$V_{SS} - 0.3$	$V_{DD} + 0.3$	V
振荡器频率	$1/t_{CLCL}$	4	24	MHz
工作温度	T_A	-40	+85	°C
贮存温度	T_{ST}	-55	+150	°C
VDD 最大流入电流		-	150	mA
VSS 最大流出电流		-	150	mA
单一 I/O 管脚最大灌电流		-	25	mA
单一 I/O 管脚最大拉电流		-	25	mA
所有 I/O 管脚最大灌电流总和		-	100	mA
所有 I/O 管脚最大拉电流总和		-	100	mA

9.1 绝对最大额定值

Note : ADC与LCD共享的管脚其输出电压不可高于 VDD。

(LQFP64 : LCD_SEG17, LCD_SEG19, LCD_SEG20, LCD_SEG21, LCD_SEG22, LCD_SEG23)

(LQFP128 : LCD_SEG36, LCD_SEG37, LCD_SEG38, LCD_SEG39)

9.2 Nano100/Nano110/Nano120/Nano130 DC电气特性

(VDD-VSS=3.3V, TA = 25°C, FOSC = 32 MHz除非其他特别说明.)

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	V _{DD}	1.8	-	3.6	V	V _{DD} = 1.8V up to 42 MHz
电源地	V _{SS} AV _{SS}	-0.3	-		V	
LDO输出电压	V _{LDO1}	1.62	1.8	1.98	V	MCU operating in Run or Idle mode
	V _{LDO2}	1.49	1.66	1.83	V	MCU operating in Power-down mode
模拟工作电压	AV _{DD}		V _{DD}		V	
模拟参考电压	V _{ref}	1.8	-	AV _{DD}	V	
运行模式下的工作电流 @ XTAL 12MHz, HCLK = 42 MHz	I _{DD1}		20.5		mA	V _{DD} = 3.6V at 42 MHz, all IP and PLL enabled [^{*5}]
	I _{DD2}		10.6		mA	V _{DD} = 3.6V at 42 MHz all IP disabled and PLL enabled
	I _{DD3}		19.1		mA	V _{DD} = 1.8V at 42 MHz all IP and PLL enabled [^{*5}]
	I _{DD4}		10.3		mA	V _{DD} = 1.8V at 42 MHz all IP disabled and PLL enabled
运行模式下的工作电流 @ XTAL 12MHz, HCLK = 32 MHz	I _{DD5}		16.2		mA	V _{DD} = 3.6V at 32 MHz, all IP and PLL enabled [^{*5}]
	I _{DD6}		8.3		mA	V _{DD} = 3.6V at 32 MHz all IP disabled and PLL enabled
	I _{DD7}		15.3		mA	V _{DD} = 1.8V at 32 MHz all IP and PLL enabled [^{*5}]
	I _{DD8}		8.0		mA	V _{DD} = 1.8V at 32 MHz all IP disabled and PLL enabled
运行模式下的工作电流	I _{DD9}		6.4		mA	V _{DD} = 3.6V at 12 MHz, all IP enabled and PLL disabled

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
@ XTAL 12MHz, HCLK = 12MHz	I _{DD10}		2.8		mA	V _{DD} = 3.6V at 12 MHz, all IP and PLL disabled
	I _{DD11}		6.3		mA	V _{DD} = 1.8V at 12 MHz, all IP enabled and PLL disabled
	I _{DD12}		2.8		mA	V _{DD} = 1.8V at 12 MHz, all IP and PLL disabled
运行模式下的工作电流 @ IRC 12MHz, HCLK = 12MHz	I _{DD13}		6.7		mA	V _{DD} = 3.6V at 12MHz, all IP enabled and PLL disabled
	I _{DD14}		3.0		mA	V _{DD} = 3.6V at 12 MHz, all IP and PLL disabled
	I _{DD15}		6.6		mA	V _{DD} = 1.8V at 12 MHz, all IP enabled and PLL disabled
	I _{DD16}		3.0		mA	V _{DD} = 1.8V at 12 MHz, all IP and PLL disabled
运行模式下的工作电流 @ XTAL 4MHz, HCLK = 4MHz	I _{DD17}		3.3		mA	V _{DD} = 3.6V at 4 MHz, all IP enabled and PLL disabled
	I _{DD18}		1.3		mA	V _{DD} = 3.6V at 4 MHz, all IP and PLL disabled
	I _{DD19}		3.2		mA	V _{DD} = 1.8V at 4 MHz, all IP enabled and PLL disabled
	I _{DD20}		1.3		mA	V _{DD} = 1.8V at 4 MHz, all IP and PLL disabled
运行模式下的工作电流 @ XTAL 32.768 kHz, HCLK = 32.768 kHz	I _{DD21}		82		uA	V _{DD} = 3.6V at 32.768 kHz all IP enabled and PLL disabled,
	I _{DD22}		74		uA	V _{DD} = 3.6V at 32.768 kHz all IP and PLL disabled
	I _{DD23}		77		uA	V _{DD} = 1.8V at 32.768 kHz all IP enabled and PLL disabled
	I _{DD24}		68		uA	V _{DD} = 1.8V at 32.768 kHz all IP and PLL disabled
运行模式下的工作电流	I _{DD25}		70		uA	V _{DD} = 3.6V at 10 kHz all IP enabled and PLL disabled

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
@ IRC 10kHz, HCLK = 10kHz	I _{DD26}		68		uA	V _{DD} = 3.6V at 10 kHz all IP and PLL disabled
	I _{DD27}		65		uA	V _{DD} = 1.8V at 10 kHz all IP enabled and PLL disabled
	I _{DD28}		62		uA	V _{DD} = 1.8V at 10 kHz all IP and PLL disabled
空闲模式下的工作电流 @ XTAL 12MHz, HCLK = 42MHz	I _{IDLE1}		14.5		mA	V _{DD} = 3.6V at 42 MHz all IP and PLL enabled [⁵]
	I _{IDLE2}		4.6		mA	V _{DD} =3.6V at 42 MHz all IP disabled and PLL enabled
	I _{IDLE3}		13.8		mA	V _{DD} = 1.8V at 42MHz all IP and PLL enabled [⁵]
	I _{IDLE4}		4.5		mA	V _{DD} = 1.8V at 42 MHz all IP disabled and PLL enabled
空闲模式下的工作电流 @ XTAL 12MHz, HCLK = 32MHz	I _{IDLE5}		11.6		mA	V _{DD} = 3.6V at 32 MHz all IP and PLL enabled [⁵]
	I _{IDLE6}		3.6		mA	V _{DD} =3.6V at 32 MHz all IP disabled and PLL enabled
	I _{IDLE7}		11.1		mA	V _{DD} = 1.8V at 32MHz all IP and PLL enabled [⁵]
	I _{IDLE8}		3.6		mA	V _{DD} = 1.8V at 32 MHz all IP disabled and PLL enabled
空闲模式下的工作电流 @ XTAL 12MHz, HCLK = 12MHz	I _{IDLE9}		4.7		mA	V _{DD} = 3.6V at 12 MHz, all IP enabled and PLL disabled
	I _{IDLE10}		0.99		mA	V _{DD} = 3.6V at 12 MHz, all IP and PLL disabled
	I _{IDLE11}		4.6		mA	V _{DD} = 1.8V at 12 MHz, all IP enabled and PLL disabled
	I _{IDLE12}		0.94		mA	V _{DD} = 1.8V at 12 MHz, all IP and PLL disabled
空闲模式下的工作电流 @ IRC 12MHz,	I _{IDLE13}		5.9		mA	V _{DD} = 3.6V at 12 MHz, all IP enabled and PLL disabled

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
HCLK = 12MHz	I _{IDLE14}		1.3		mA	V _{DD} = 3.6V at 12 MHz, all IP and PLL disabled
	I _{IDLE15}		4.9		mA	V _{DD} = 1.8V at 12 MHz, all IP enabled and PLL disabled
	I _{IDLE16}		1.3		mA	V _{DD} = 1.8V at 12 MHz, all IP and PLL disabled
空闲模式下的工作电流 @ XTAL 4MHz, HCLK = 4MHz	I _{IDLE17}		2.7		mA	V _{DD} = 3.6V at 4 MHz, all IP enabled and PLL disabled
	I _{IDLE18}		0.66		mA	V _{DD} = 3.6V at 4 MHz, all IP and PLL disabled
	I _{IDLE19}		2.7		mA	V _{DD} = 1.8V at 4 MHz, all IP enabled and PLL disabled
	I _{IDLE20}		0.64		mA	V _{DD} = 1.8V at 4 MHz, all IP and PLL disabled
空闲模式下的工作电流 @ XTAL 32.768kHz, HCLK = 32.768kHz	I _{IDLE21}		78		uA	V _{DD} = 3.6V at 32.768 kHz all IP enabled and PLL disabled
	I _{IDLE22}		69		uA	V _{DD} = 3.6V at 32.768 kHz all IP and PLL disabled
	I _{IDLE23}		72		uA	V _{DD} = 1.8V at 32.768 kHz all IP enabled and PLL disabled
	I _{IDLE24}		63		uA	V _{DD} = 1.8V at 32.768 kHz all IP and PLL disabled
空闲模式下的工作电流 @ IRC 10kHz, HCLK = 10kHz	I _{IDLE25}		69		uA	V _{DD} = 3.6V at 10 kHz all IP enabled and PLL disabled
	I _{IDLE26}		66		uA	V _{DD} = 3.6V at 10 kHz all IP and PLL disabled
	I _{IDLE27}		63		uA	V _{DD} = 1.8V at 10 kHz all IP enabled and PLL disabled
	I _{IDLE28}		61		uA	V _{DD} = 1.8V at 10 kHz all IP and PLL disabled
掉电模式下的工作电流	I _{PWD1}		1.2		μA	V _{DD} = 3.6V, RTC OFF, all clock stop With RAM Retention, IO no loading

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
	IPWD2		0.8		μA	V _{DD} = 1.8V, RTC OFF, all clock stop With RAM Retenstion, IO no loading
	IPWD3		2.8		μA	V _{DD} = 3.6V, RTC ON, all clock stop except 32.768 kHz With RAM Retenstion, IO no loading
	IPWD4		2.0		μA	V _{DD} = 1.8V, RTC ON, all clock stop except 32.768 kHz With RAM Retenstion, IO no loading
PA, PB, PC, PD, PE, PF 输入上拉电阻	R _{IN}		40		KΩ	V _{DD} = 3.3V
			98		KΩ	V _{DD} = 1.8V
PA, PB, PC, PD, PE, PF 输入漏电流	I _{LK}	-0.1	-	+0.1	μA	V _{DD} = 3.3V, 0 < V _{IN} < V _{DD}
PA, PB, PC, PD, PE, PF 输入低电压 (Schmitt input)	V _{IL1}		-	0.4V _{DD}	V	
PA, PB, PC, PD, PE, PF 输入高电压 (Schmitt input)	V _{IH1}	0.6V _{DD}		5.5	V	ADC and DAC shared pins without Input 5V tolerance.
XT1 ^[*2] 输入低电压	V _{HY}		0.2V _{DD}		V	
XT1 ^[*2] 输入高电压	V _{IL2}	0	-	0.4		V _{DD} = 3.3V
X32I ^[*2] 输入低电压	V _{IH2}	1.5	-	V _{DD} +0.2	V	V _{DD} = 3.3V
X32I ^[*2] 输入高电压	V _{IL4}	0	-	0.3	V	
Input High Voltage X32I ^[*2]	V _{IH4}	1.5	-	1.98	V	
/RESET 管脚负向阈值 电压 (Schmitt 输入),	V _{ILS}	1.28	1.33	1.37	V	V _{DD} = 3.3V
/RESET 管脚正向阈值 电压 (Schmitt 输入),	V _{IHS}	1.75	1.98	2.25	V	V _{DD} = 3.3V
PA, PB, PC, PD, PE, PF 拉电流 (推挽模式)	I _{SR21}	-10	-14	-	mA	V _{DD} = 3.3V, V _S = V _{DD} -0.7V
	I _{SR22}	-3	-5	-	mA	V _{DD} = 1.8V, V _S = V _{DD} -0.45V

参数	SYM.	规格				测试条件
		最小值	典型值	最大值	单位	
PA, PB, PC, PD, PE, PF 灌电流 (推挽模式)	I _{SK21}	10	15	-	mA	V _{DD} = 3.3V, V _S = 0.7V
	I _{SK22}	3	6	-	mA	V _{DD} = 1.8V, V _S = 0.45V

注：

1. /RESET管脚为 Schmitt 触发输入
2. 晶振输入为 CMOS 输入
3. 建议一个10uF 或更高的电容和一个 100nF 旁路电容连接在 VDD 和设备最近的 VSS 管脚之间。
4. 为保证电源的稳定性，一个 4.7uF 或更高的电容必须连接在 LDO 管脚和设备最近的 VSS 管脚之间。另外一个 100nF 旁路电容在 LDO 和 VSS 之间帮助抑制输出噪声。
5. 所有外设的时钟源皆来自HXT(12MHz)，除了SPI是来自HCLK。

9.3 AC电气特性

9.3.1 外部输入时钟

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
时钟高电平时间	t_{CHCX}	10	-		nS	
时钟低电平时间	t_{CLCX}	10	-		nS	
时钟上升沿时间	t_{CLCH}	2	-	15	nS	
时钟下降沿时间	t_{CHCL}	2	-	15	nS	

注：占空比为50 %

9.3.2 外部 4~24MHz XTAL 振荡器

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
振荡器频率	f_{HXT}	4	12	24	MHz	$V_{DD} = 1.8V \sim 3.6V$
温度	T_{HXT}	-40	-	+85	°C	
工作电流	I_{HXT}		0.3		mA	$V_{DD} = 3.0V$

9.3.2.1 典型晶振应用电路

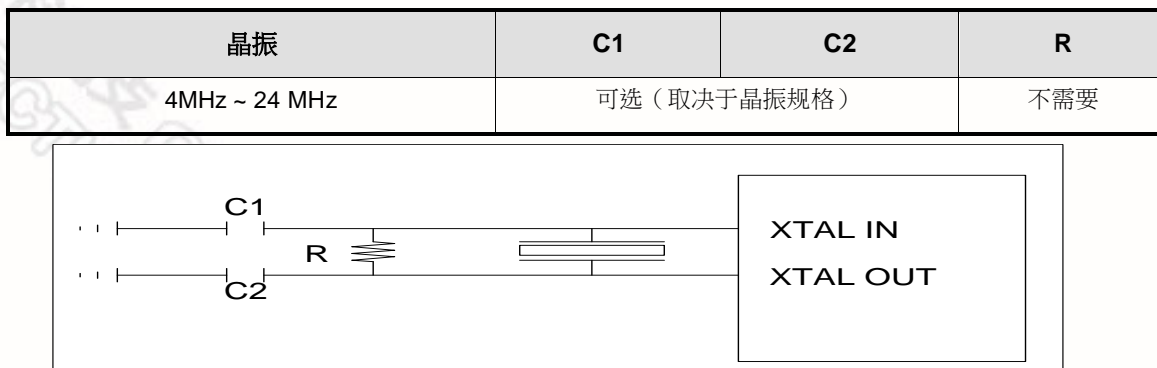


图 9-1 典型晶振应用电路

9.3.3 外部 32.768 kHz 晶振

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
振荡器频率	f_{LXT}		32.768		kHz	$V_{DD} = 1.8V \sim 3.6V$
温度	T_{LXT}	-40	-	+85	°C	
工作电流	I_{LXT}		1.2		μA	$V_{DD} = 3.0V$

9.3.4 内部 12 MHz 振荡器

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压 ^[1]	V_{HRC}		1.8		V	
标定内部振荡器频率	F_{HRC}	11.88	12	12.12	MHz	25°C, $V_{DD} = 3V$
		11.76	12	12.24	MHz	-40°C~+85°C, $V_{DD} = 1.8V \sim 3.6V$
		11.97	12	12.03	MHz	-40°C~+85°C, $V_{DD} = 1.8V \sim 3.6V$ Enable 32.768K crystal oscillator and set TRIM_SEL[1:0]="10"
工作电流	I_{HRC}		450		μA	

注：内部振荡器工作电压来自LDO。

9.3.5 内部 10 kHz 振荡器

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压 ^[1]	V_{LRC}		1.8		V	
中心频率	F_{LRC}	7	10	13	kHz	25°C, $V_{DD} = 3V$
		5	10	15	kHz	-40°C~+85°C, $V_{DD} = 1.8V \sim 3.6V$
工作电流	I_{LRC}		0.7		μA	$V_{DD} = 3V$

注：内部振荡器工作电压来自LDO。

9.4 模拟量特性

9.4.1 12-bit ADC

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	AV_{DD}	1.8		3.6	V	$AV_{DD} = V_{DD}$
工作电流	I_{ADC42}		147		μA	$AV_{DD} = V_{DD} = 3.0V$ $ADC_VREF = AV_{DD}$ $ADC\ Clock\ Rate = 42\ MHz$
	I_{ADC12}		50		μA	$AV_{DD} = V_{DD} = 3.0V$ $ADC_VREF = AV_{DD}$ $ADC\ Clock\ Rate = 12\ MHz$
分辨率	R_{ADC}			12	Bit	
参考电压	V_{REF}	1.8		AV_{DD}	V	
参考输入电流(Avg.)	I_{REF}			10	μA	
ADC 输入电压	V_{IN}	0		V_{REF}	V	
转换时间	T_{CONV}	0.5			μS	
采样率	F_{SPS}			2M	Hz	$V_{DD} = 3V$
非线性积分误差	INL		± 1	± 2	LSB	V_{REF} is external Vref pin
非线性差分误差	DNL		± 0.8	-1~+1.5	LSB	V_{REF} is external Vref pin
增益误差	E_G		-	± 2	LSB	V_{REF} is external Vref pin
偏移量误差	E_{OFFSET}		-	± 3	LSB	V_{REF} is external Vref pin
绝对误差	E_{ABS}		-	± 6	LSB	V_{REF} is external Vref pin
ADC 时钟频率	F_{ADC}	0.25		42	MHz	
时钟周期	AD_{CYC}	20			Cycle	
内部电容	C_{IN}	-	5	-	pF	
一致性	-	Guaranteed			-	

9.4.2 欠压检测

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	V_{BOD}	1.8		3.6	V	

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
BOD17 静态电流	I _{BOD17}		1		μA	AV _{DD} = 3.0V, enable BOD17
BOD20 静态电流	I _{BOD20}		1		μA	AV _{DD} = 3.0V, enable BOD20
BOD25 静态电流	I _{BOD25}		1		μA	AV _{DD} = 3.0V, enable BOD25
BOD17 侦测电平	V _{B17dt}	1.6	1.7	1.8	V	25°C
BOD20 侦测电平	V _{B20dt}	1.9	2.0	2.1	V	25°C
BOD25 侦测电平	V _{B25dt}	2.4	2.5	2.6	V	25°C

9.4.3 上电复位

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
复位电压	V _{POR}	-	1.6	-	V	
静态电流	I _{POR}	-	1	-	nA	LDO 输出 > 复位电压

9.4.4 温度传感器

参数	符号	规格				测试条件 (提供电压 = 3.36V)
		最小值	典型值	最大值	单位	
检测温度	T _{DET}	-40		+110	°C	
工作电流	I _{TEMP}	-	5	-	μA	
增益	V _{TG}	-1.80	-1.73	-1.65	mV/°C	
偏移量	V _{TO}	730	740	750	mV	温度在 0 °C

注：内部振荡器工作电压来自LDO。

9.4.5 12-bit DAC

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	AV _{DD}	2.0		3.6	V	AV _{DD} = V _{DD}

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电流	I _{DAC}		2.20		mA	AV _{DD} = V _{DD} = 3.0V, DAC_VREF = AV _{DD} 500kHz 转换速率
分辨率	R _{ADC}			12	Bit	
参考电压	V _{REF}	1.8		A _{VDD}	V	
参考输入电流 (Avg.)	I _{REF}		0.85		mA	AV _{DD} = V _{DD} = 3.0V DAC_VREF=Ext_Vref 500kHz 转换速率
DAC 输出范围	V _{OUT}	0.1 x V _{REF}	-	0.9 x V _{REF}	V	
转换速率	F _{SPS}			500	kHz	V _{DD} = 3V
非线性积分误差	INL		±4	±5	LSB	V _{REF} is external Vref pin Not include 偏移量 and gain error
非线性差分误差	DNL		±1	±2	LSB	V _{REF} is external Vref pin Not include 偏移量 and gain error
增益误差	E _G		290		LSB	
偏移量误差	E _{OFFSET}		150		LSB	

9.4.6 LCD

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	V _{DD}	1.8	-	3.6	V	
VLCD 电压	V _{LCD34}	-	3.4	-	V	CPUMP_VOL_SET=111, no loading
VLCD 电压	V _{LCD33}	-	3.3	-	V	CPUMP_VOL_SET=110, no loading
VLCD 电压	V _{LCD32}	-	3.2	-	V	CPUMP_VOL_SET=101, no loading
VLCD 电压	V _{LCD31}	-	3.1	-	V	CPUMP_VOL_SET=100, no loading
VLCD 电压	V _{LCD30}	-	3.0	-	V	CPUMP_VOL_SET=011, no loading
VLCD 电压	V _{LCD29}	-	2.9	-	V	CPUMP_VOL_SET=010, no loading
VLCD 电压	V _{LCD28}	-	2.8	-	V	CPUMP_VOL_SET=001, no loading
VLCD 电压	V _{LCD27}	-	2.7	-	V	CPUMP_VOL_SET=000, no loading
工作电流	I _{LCD}	-	10	-	μA	V _{DD} = 3V, frame rate = 32Hz Without loading

9.4.7 内部参考电压

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
工作电压	AV_{DD}	1.8	-	3.6	V	
1.8V 参考电压	V_{REF1}	1.69	1.8	1.87	V	$AV_{DD} \geq 2.0V$ (-40°C~85°C)
2.5V 参考电压	V_{REF2}	2.35	2.5	2.60	V	$AV_{DD} \geq 2.8V$ (-40°C~85°C)
稳定时间	T_{REFTAB}	-	1	-	ms	
工作电流	I_{VREF}	-	30	-	μA	$AV_{DD} = 3V$

9.4.8 USB PHY 规格

9.4.8.1 USB PHY DC 电气特性

符号	参数	条件	最小值	典型值	最大值	单位
V_{IH}	输入高(driven)		2.0	-		V
V_{IL}	输入低			-	0.8	V
V_{DI}	差分输入	PADP-PADM	0.2	-		V
V_{CM}	差分同模范围	Includes V_{DI} range	0.8	-	2.5	V
V_{SE}	单端接收器阈值		0.8	-	2.0	V
	接收器迟滞			200		mV
V_{OL}	输出低 (driven)		0	-	0.3	V
V_{OH}	输出高 (driven)		2.8	-	3.6	V
V_{CRS}	输出信号串扰电压		1.3	-	2.0	V
R_{PU}	上拉电阻		1.425	-	1.575	k Ω
R_{PD}	下拉电阻		14.25	-	15.75	k Ω
V_{TRM}	上行端口上的上拉电阻的极限电压 (RPU)		3.0	-	3.6	V
Z_{DRV}	驱动输出阻抗	稳态驱动*		10		Ω
C_{IN}	发射器电容	Pin to GND		-	20	pF

*驱动输出阻抗不包括串联电阻阻抗。

9.4.8.2 USB PHY全速驱动器电气特性

符号	参数	条件	最小值	典型值	最大值	单位
T_{FR}	上升时间	$C_L=50p$	4	-	20	ns
T_{FF}	下降时间	$C_L=50p$	4	-	20	ns
T_{FRFF}	上升和下降时间比值	$T_{FRFF}=T_{FR}/T_{FF}$	90	-	111.11	%

9.4.8.3 USB PHY电源功耗

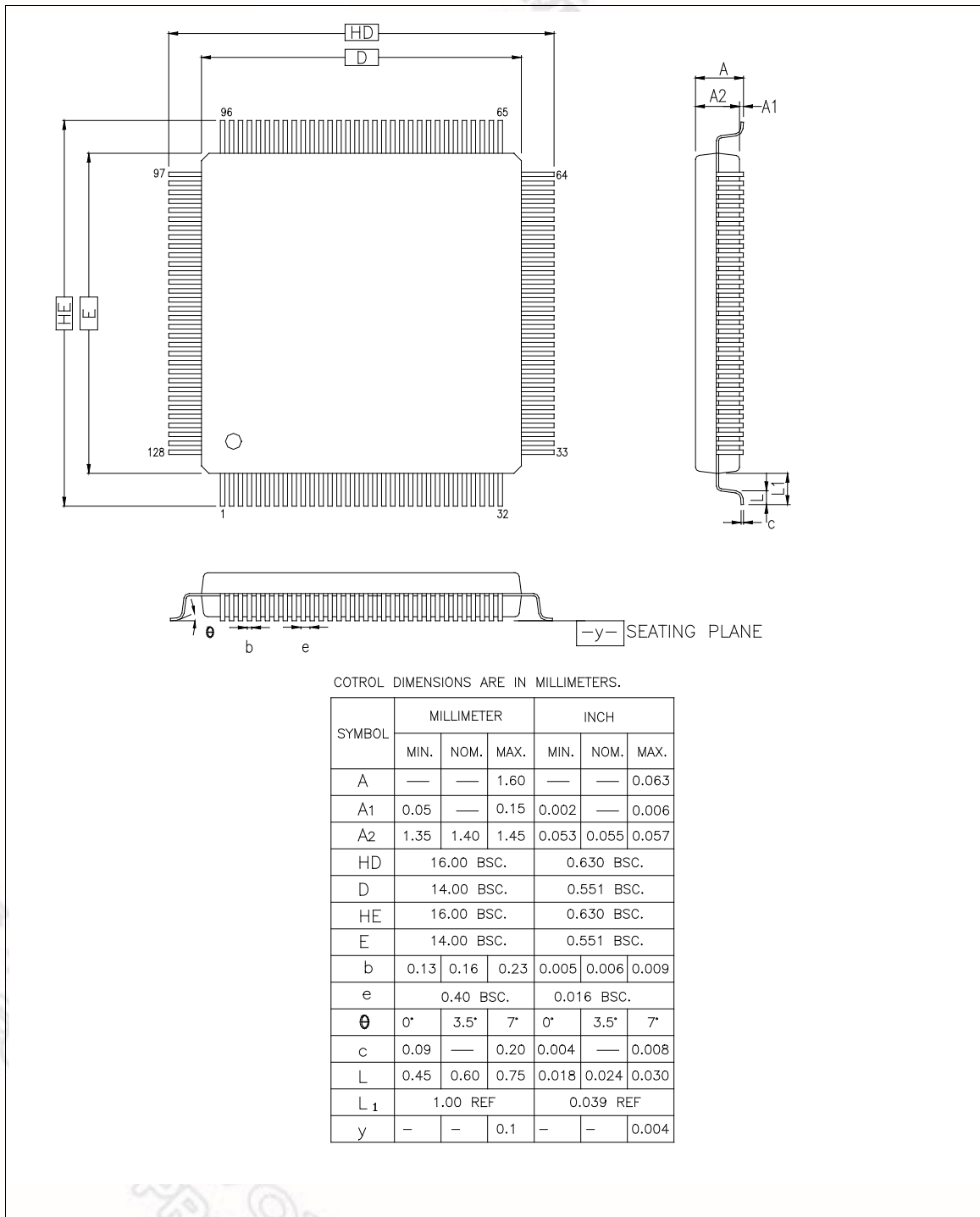
符号	参数	条件	最小值	典型值	最大值	单位
I_{VDDREG} (Full Speed)	VDDD 和 VDDREG 供给电流 (稳态)	待机		50		uA

9.4.8.4 USB LDO DC电气特性

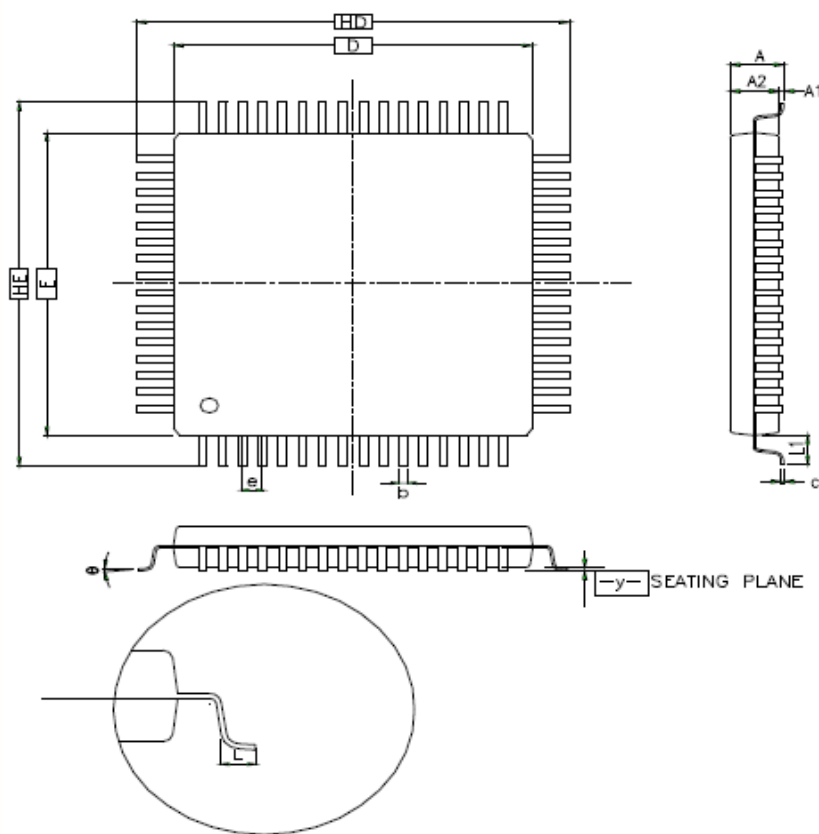
符号	参数	条件	最小值	典型值	最大值	单位
VBUS				5		V
V33	输出电压	VBUS = 5V, 25°C	2.97	3.3	3.63	V
Iop	工作电流			100		uA

10 封装尺寸

10.1 LQFP128 (14x14x1.4 mm footprint 2.0 mm)

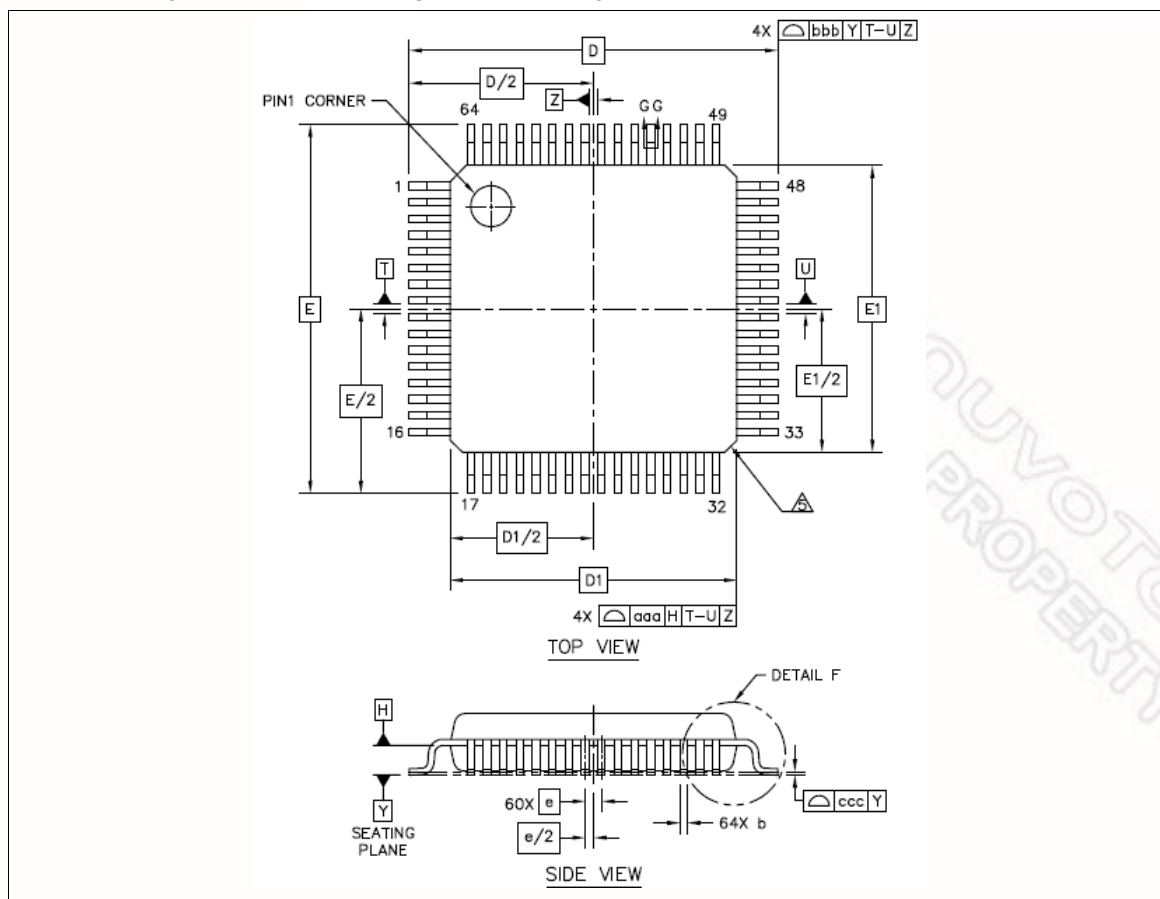


10.2 LQFP64 (10x10x1.4 mm footprint 2.0 mm)

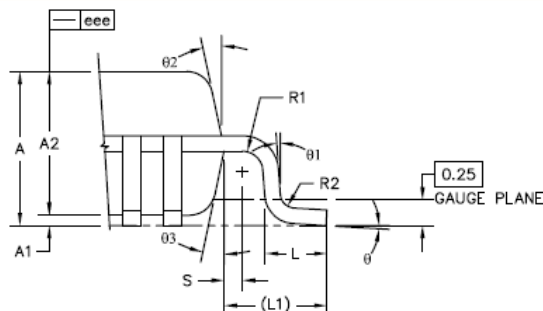
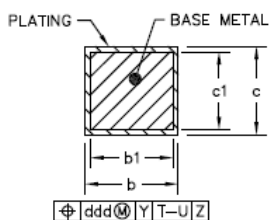


Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.063	—	—	1.60
A ₁	0.002	—	0.006	0.05	—	0.15
A ₂	0.053	0.055	0.057	1.35	1.40	1.45
b	0.007	0.008	0.011	0.17	0.20	0.27
c	0.004	—	0.008	0.09	—	0.20
D	—	0.393	—	—	10.00	—
E	—	0.393	—	—	10.00	—
e	—	0.020	—	—	0.50	—
H _D	—	0.472	—	—	12.00	—
H _E	—	0.472	—	—	12.00	—
L	0.018	0.024	0.030	0.45	0.60	0.75
L ₁	—	0.039	—	—	1.00	—
y	—	0.004	—	—	0.10	—
θ	0°	3.5°	7°	0°	3.5°	7°

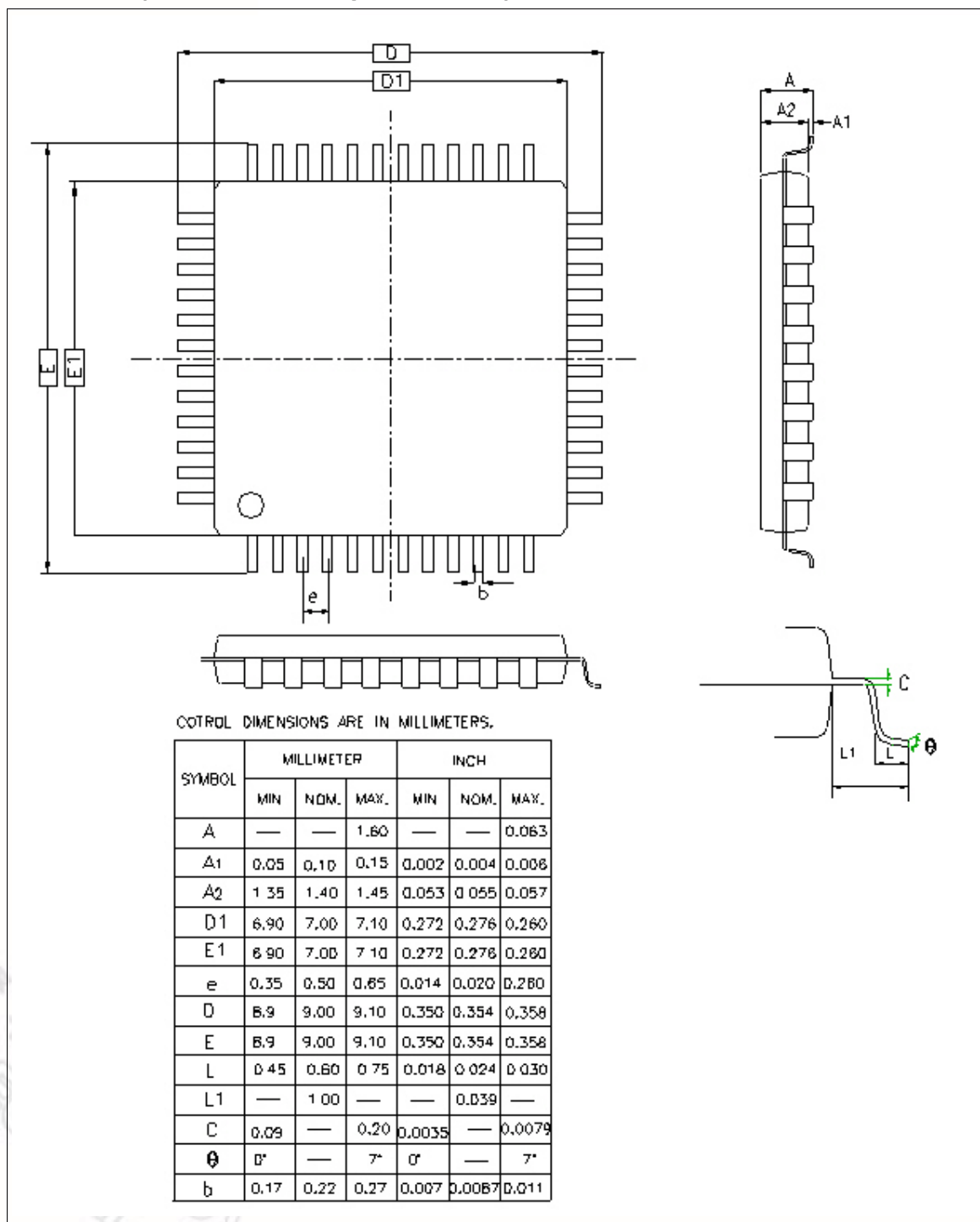
10.3 LQFP64 (7x7x1.4 mm footprint 2.0 mm)



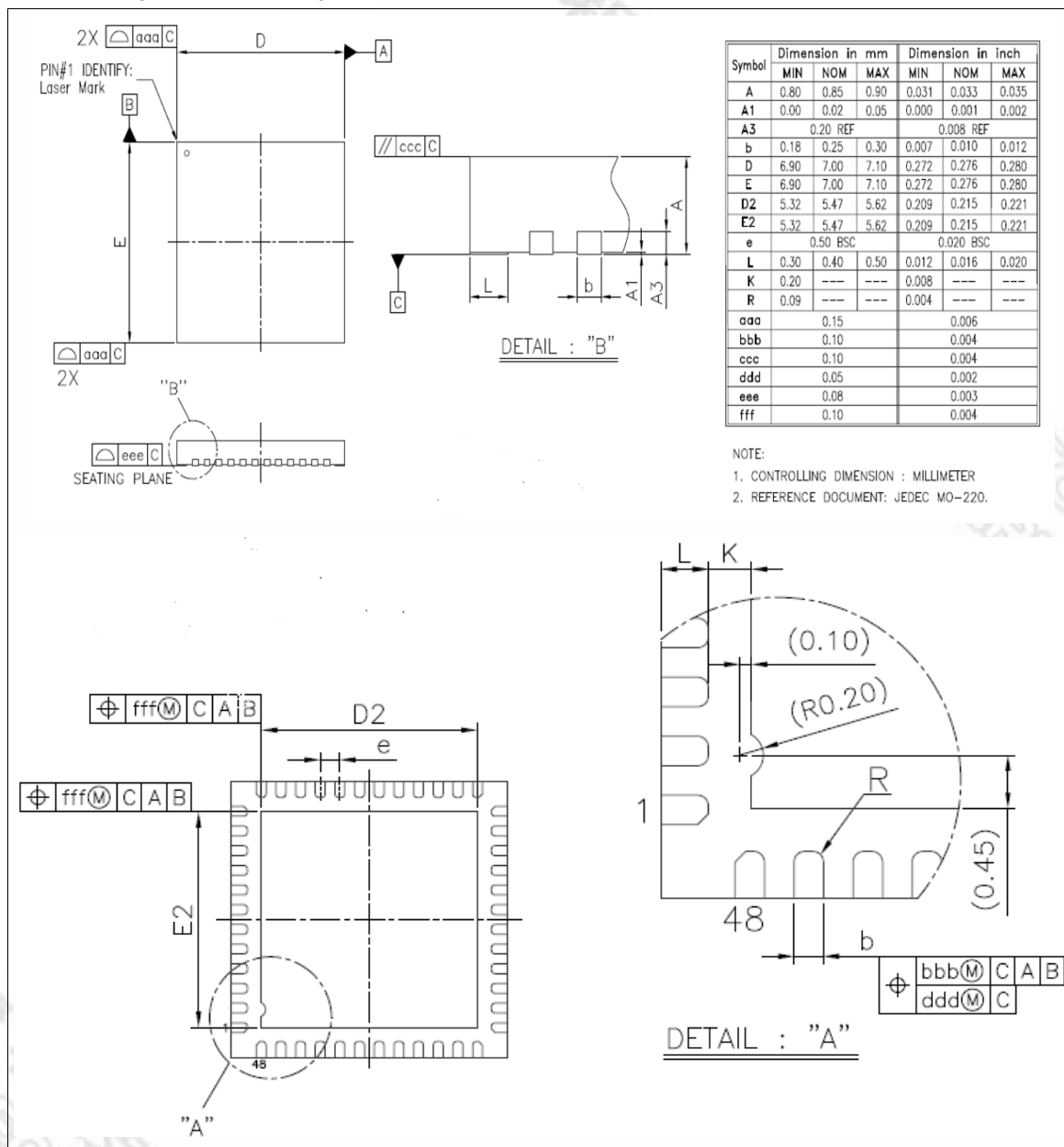
		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	----	----	1.6
STAND OFF		A1	0.05	----	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.13	0.18	0.23
LEAD WIDTH		b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)		c	0.09	----	0.2
L/F THICKNESS		c1	0.09	----	0.16
BODY SIZE	X	D	9 BSC		
	Y	E	9 BSC		
	X	D1	7 BSC		
	Y	E1	7 BSC		
LEAD PITCH		e	0.4 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	----	----
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	----	----
		R2	0.08	----	0.2
		S	0.2	----	----
PACKAGE EDGE TOLERANCE		aaa	0.2		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.07		
MOLD FLATNESS		eee	0.05		



10.4 LQFP48 (7x7x1.4 mm footprint 2.0 mm)



10.5 QFN48 (7x7x0.85 mm)



11 版本历史

日期	版本	描述
2012.10.11	1.00	初次发布
2012.12.11	1.01	<ol style="list-style-type: none"> 1. 引脚描述部分添加SmartCard 用作UART 时的引脚描述. 2. 统一定时器缩写为 TMR. 3. 修正外部输入时钟的说明. 4. 于管脚功能描述与LCD总览中加入LCD COM4与COM5的说明 5. 修改ADC章节中由定时器事件使能之描述 6. 修订定时器章节(由TIMER0/1_CH0/1 改为 TMR_x (x=0,1, 2,3)) 7. 增加定时器连续计数模式于寄存器中的说明
2012.12.17	1.02	<ol style="list-style-type: none"> 1. 增加UCID于ISP模式的说明。
2012.12.28	1.03	<ol style="list-style-type: none"> 1. 于5.3.3节增加各外设可再次进入掉电模式的对照表。 2. 于7.1节增加LCD的R-type描述。 3. 于9.2节修正每一时钟源于运行与空闲模式下的运行电流数值，并增加42 MHz的相关数据。
2013.01.02	1.04	<ol style="list-style-type: none"> 1. 更新耗电章节表格。
2013.03.05	1.05	<ol style="list-style-type: none"> 1. 更新5.5.4.8节与5.20.4.5节描述。 2. 更新5.13.2节显示模式由4组改为6组。 3. 修正3.4节之管脚说明。 4. 更新5.9.4.4节Config0 第31位 (CWDT_EN) 的描述。 5. 更新图5-16与图 5-17量测条件。 6. 新增5.5.4.13节温度传感器相关描述。 7. 更新5.4.3节时钟控制器。 8. 更新9.4.4节中温度传感器之仿真量特性。 9. 修正5.20.6节DMA_TX_EN与DMA_RX_EN寄存器设定说明。 10. 修正5.15.4.4节中RTC 频率补偿说明与范例。 11. 修正5.15.6节中FRACTION寄存器之计算公式。 12. 修正5.5.4.9节快速输入与慢速输入之通道说明。 13. 修正5.7.6节PDMA_TCR与TO_EN寄存器说明。 14. 修正5.13.6节PDSTS寄存器说明。 15. 修正5.19.3节UART时钟控制图。 16. 修正5.19.4.4节IrDA TX/RX 时序图。

		<p>17. 修正5.16.2节智能卡特性，在UART模式下为半双工异步通信。</p> <p>18. 新增5.21.4节当系统进掉电模式，WDT 计数器会自动重置之描述。</p>
2013.05.28	1.06	<p>1. 新增5.13.6节 MUX (LCD_CTL[3:1])设定 LCD_SEG0/1 或 LCD_COM4/5寄存器的相关描述。</p> <p>2. 修正3.3.2节 Nano110 LQFP128 管脚图。</p> <p>3. 新增5.4.6节模块时钟输出寄存器(MCLK)等相关设定说明。</p> <p>4. 修正2.1至2.4节特性「在全温度范围 12MHz OSC 会有 2 % 偏差」。</p> <p>5. 修正5.7.6节 CH1_SEL(DMA_DSSR0[12:8])与 CH4_SEL(DMA_DSSR1[12:8]) 寄存器相关描述。</p> <p>6. 修正9.4.5节 DAC模拟量特性。</p> <p>7. 修正5.4.6节 HXT_HF_ST (PWRCTL[12:11])寄存器相关描述。</p> <p>8. 新增Nano110RC2BN至Nano110 LCD Line 产品选型表。</p> <p>9. 修正5.3.5节PA2_MFP與PA3_MFP寄存器描述。</p>
2013.12.04	1.07	<p>1. 更新3.1节Nano100系列选型编码。</p> <p>2. 新增3.2节Nano100 QFN48 与第10章 QFN48的封装尺寸。</p> <p>3. 修正5.5.6节ADCPWD寄存器初始值为0x1E002与PWDMOD位说明。</p> <p>4. 修正5.4.6节PLLOUT输出时钟频率「频率必须介于48MHz与 120MHz 之间」$48\text{ MHz} < \text{FOUT} < 120\text{ MHz}$。</p> <p>5. 修正9.4.7节LCD模拟量特性表格中参数符号与其说明。</p> <p>6. 新增5.18.6节TMR_CTL[12]之ENENT_EN位注记，「注：当 EVENT_EN 被使能，不可选择EXT_TMx(GPB)作为时钟源，并且被选择的时钟源速度须高于3倍的EXT_TMx的时钟速度。」</p> <p>7. 新增3.4节、5.13.1节与9.1节说明对于LCD显示驱动器「ADC与LCD 共享的管脚其输出电压不可高于 VDD」。</p> <p>8. 修正7.2与7.3节ADC与DAC应用电路图。</p> <p>9. 修正5.21.4节以下说明，设置 WTE (WDT_CTL[3]) 使能看门狗定时器和 WDT 计数器开始计数。当计数器达到选择的超时间隔，看门狗定时器中断标志 WDT_IS 将被立即被置位，并请求 WDT 中断 (如果看门狗定时器中断使能位 WDT_IE 置位)，同时紧接着超时事件会有一个指定的延时可透过 WTRDSEL 设置。</p> <p>10. 修正5.13.4.2节6-COM、5-COM、4-COM 配置说明。</p> <p>11. 修正5.4.6节PLLCON[16]之描述，「如果在 PWR_CTL寄存器中设置 PD_EN 位为“1”，PLL 也会进入掉电模式」。</p>

新唐科技 NUVOTON
INTELLECTUAL PROPERTY

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.

新唐科技 NUVOTON
INTELLECTUAL PROPERTY