

# NuMicro™ NUC100 系列 NUC100/NUC120 规格书

*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](http://www.nuvoton.com)

目录

1	概述 .....	7
2	特征 .....	8
2.1	NuMicro™ NUC100 特征 – Advanced Line.....	8
2.2	NuMicro™ NUC120特征– USB Line .....	12
2.3	NuMicro™ NUC100/120xxxDN产品选型指南 .....	16
2.3.1	NuMicro™ NUC100 Advanced Line选型指南.....	16
2.3.2	NuMicro™ NUC120 USB Line选型指南 .....	16
2.4	管脚配置.....	18
2.4.1	NuMicro™ NUC100 管脚图.....	18
2.4.2	NuMicro™ NUC120 管脚图.....	21
2.5	管脚描述.....	24
2.5.1	NuMicro™ NUC100 管脚描述 .....	24
2.5.2	NuMicro™ NUC120 管脚描述 .....	31
3	框图 .....	38
3.1	NuMicro™ NUC100 框图.....	38
3.2	NuMicro™ NUC120 框图.....	39
4	功能描述 .....	40
4.1	ARM® Cortex™-M0 内核.....	40
4.2	系统管理器 .....	42
4.2.1	概述 .....	42
4.2.2	系统复位.....	42
4.2.3	系统电源分配.....	43
4.2.4	系统内存映射 .....	45
4.2.5	系统管理器控制寄存器.....	47
4.2.6	寄存器描述 .....	48
4.2.7	系统定时器 (SysTick).....	87
4.2.8	嵌套向量中断控制器 (NVIC) .....	92
4.2.9	系统控制寄存器(SCS).....	116
4.3	时钟控制器 .....	117
4.3.1	概述 .....	117
4.3.2	时钟发生器 .....	119
4.3.3	系统时钟 & SysTick 时钟 .....	120
4.3.4	外围设备时钟.....	121
4.3.5	掉电模式时钟.....	121
4.3.6	分频器输出 .....	122
4.4	USB 设备控制器 (USB).....	123
4.4.1	概述 .....	123
4.4.2	特征 .....	123
4.5	通用 I/O (GPIO).....	124

4.5.1	概述	124
4.5.2	特征	124
4.6	I <sup>2</sup> C 串行接口控制器 (Master/Slave) (I <sup>2</sup> C)	125
4.6.1	概述	125
4.6.2	特征	126
4.7	PWM 发生器和捕捉定时器 (PWM)	127
4.7.1	概述	127
4.7.2	特征	127
4.8	实时时钟 (RTC)	129
4.8.1	概述	129
4.8.2	特征	129
4.9	串行外围设备接口(SPI)	130
4.9.1	概述	130
4.9.2	特征	130
4.10	定时器控制器 (TMR)	131
4.10.1	概述	131
4.10.2	特征	131
4.11	看门狗定时器(WDT)	132
4.11.1	概述	132
4.11.2	特征	132
4.12	窗口看门狗定时控制器	133
4.12.1	概述	133
4.12.2	特征	133
4.12.3	框图	133
4.13	UART接口控制器(UART)	134
4.13.1	概述	134
4.14	PS/2 设备控制器 (PS2D)	135
4.14.1	概述	135
4.14.2	特征	135
4.15	I <sup>2</sup> S 控制器 (I <sup>2</sup> S)	136
4.15.1	概述	136
4.15.2	特征	136
4.16	模拟数字转换(ADC)	137
4.16.1	概述	137
4.16.2	特征	137
4.17	模拟比较器(ACMP)	138
4.17.1	概述	138
4.17.2	特征	138
4.18	PDMA控制器(PDMA)	139
4.18.1	概述	139
4.18.2	特征	139

4.19	外部总线接口 (EBI) .....	139
4.19.1	简介 .....	140
4.19.2	特性 .....	140
4.20	FLASH 内存控制器 (FMC) .....	141
4.20.1	概述 .....	141
4.20.2	特征 .....	141
5	电气特性 .....	142
5.1	绝对最大额定值 .....	142
5.2	DC电气特性 .....	143
5.3	AC电气特性 .....	147
5.3.1	外部 4~24 MHz 高速振荡器 .....	148
5.3.2	外部 4~24 MHz 高速晶振 .....	148
5.3.3	外部 32.768 kHz 低速晶振 .....	149
5.3.4	内部 22.1184 MHz 高速振荡器 .....	149
5.3.5	内部 10 kHz 低速振荡器 .....	149
5.4	模拟量特性 .....	149
5.4.1	12-bit SARADC规格 .....	149
5.4.2	LDO规格和电源管理 .....	150
5.4.3	低压复位说明 .....	151
5.4.4	欠压检测说明 .....	151
5.4.5	上电复位说明 .....	151
5.4.6	温度传感器说明 .....	152
5.4.7	比较器说明 .....	152
5.4.8	USB PHY说明 .....	153
5.5	Flash DC 电气特性 .....	155
6	封装定义 .....	156
6.1	100-pin LQFP (14x14x1.4 mm footprint 2.0 mm) .....	156
6.2	64-pin LQFP (10x10x1.4 mm footprint 2.0 mm) .....	157
6.3	48-pin LQFP (7x7x1.4 mm footprint 2.0 mm) .....	158
7	版本历史 .....	159

图目录

图 2-1 NuMicro™ NUC100系列选型码 ..... 17

图 2-2 NuMicro™ NUC100VxxDN LQFP 100-pin管脚图..... 18

图 2-3 NuMicro™ NUC100SxxDN LQFP 64-pin 管脚图..... 19

图 2-4 NuMicro™ NUC100LxxDN LQFP 48-pin 管脚图 ..... 20

图 2-5 NuMicro™ NUC120VxxDN LQFP 100-pin 管脚图..... 21

图 2-6 NuMicro™ NUC120SxxDN LQFP 64-pin 管脚图..... 22

图 2-7 NuMicro™ NUC120LxxDN LQFP 48-pin 管脚图 ..... 23

图 3-1 NuMicro™ NUC100框图 ..... 38

图 3-2 NuMicro™ NUC120 框图 ..... 39

图 4-1 功能框图 ..... 40

图 4-2 NuMicro™ NUC100电源分配图 ..... 43

图 4-3 NuMicro™ NUC120电源分配图 ..... 44

图 4-4 时钟发生器全局框图 ..... 118

图 4-5 时钟发生器框图 ..... 119

图 4-6 系统时钟框图 ..... 120

图 4-7 SysTick 时钟控制框图 ..... 120

图 4-8 分频器的时钟源 ..... 122

图 4-9 分频器的框图 ..... 122

图 4-10 I<sup>2</sup>C 总线时序 ..... 125

图 5-1 典型晶振应用电路..... 149

表目录

表 1-1 所支持的接口列表..... 7

表 4-1 片上控制器的地址空间分配 ..... 46

表 4-2 异常模式 ..... 93

表 4-3 系统中断映射 ..... 94

表 4-4 向量表格式..... 95

## 1 概述

NuMicro™ NUC100系列是32位的内嵌ARM® Cortex™-M0核的微控制器，适用于工业控制和需要丰富的通信接口的应用领域，Cortex™-M0是ARM最新的32位嵌入式处理器，拥有与传统8051单片机之匹敌的价格优势。NuMicro™ NUC100 系列包括NUC100和NUC120。

NuMicro™ NUC100 Advanced Line 内嵌Cortex™-M0内核，最高可运行至50 MHz，内建32K/64K/128K字节的Flash存储器，以及8K/16K字节SRAM，4K字节用于存储ISP引导代码的ROM。另外还有丰富的外设，如定时器，看门狗定时器，窗口看门狗定时器，RTC，带CRC计算单元的PDMA，UART，SPI，I<sup>2</sup>C，I<sup>2</sup>S，PWM 定时器，GPIO，PS/2，12位ADC，模拟比较器，低电压复位控制和欠压检测功能。

NuMicro™ NUC120 USB Line 带全速USB 2.0设备控制器，内嵌Cortex™-M0内核，最高可运行至50 MHz，内建32K/64K/128K字节的Flash存储器，以及8K/16K字节SRAM，4K字节用于存储ISP引导代码的ROM。另外还有丰富的外设，如定时器，看门狗定时器，窗口看门狗定时器，RTC，带CRC计算单元的PDMA，UART，SPI，I<sup>2</sup>C，I<sup>2</sup>S，PWM 定时器，GPIO，EBI，PS/2，USB 2.0 FS 设备，12位ADC，模拟比较器，低电压复位控制和欠压检测功能。

Product Line	UART	SPI	I <sup>2</sup> C	USB	EBI	PS/2	I <sup>2</sup> S
NUC100	•	•	•		•	•	•
NUC120	•	•	•	•	•	•	•

表 1-1 所支持的接口列表

## 2 特征

该器件的功能依赖于产品线和他们的子系统产品。

### 2.1 NuMicro™ NUC100 特征 – Advanced Line

- 内核
  - ARM® Cortex™-M0 内核最高允许 50 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
- 内建 LDO, 宽电压工作范围 2.5 V 到 5.5 V
- Flash 存储器
  - 32K/64K/128K 字节 Flash 用于存储程序代码
  - 4KB flash 用于存储ISP引导代码
  - 支持在系统编程 (ISP)和在应用编程(IAP)方式更新应用程序
  - 支持512 字节单页擦除
  - 在128K字节系统中可配置数据FLASH地址和大小, 在32K字节和64K字节系统中固定为4K字节数据Flash
  - 通过SWD/ICE接口, 支持2 线 ICP升级方式
  - 支持外部编程器并行高速编程模式
- SRAM存储器
  - 8K/16K 字节内建 SRAM
  - 支持 PDMA 模式
- PDMA (Peripheral DMA)
  - 支持9通道 PDMA 用于SRAM和周边设备的自动数据传输
  - 支持带4个通用多项式的CRC 运算, CRC-CCITT, CRC-8, CRC-16 和 CRC-32
- 时钟控制
  - 针对不同应用可灵活选择时钟
  - 内部 22.1184 MHz 高速振荡器可用于系统运行
    - ◆ 在+25 °C, VDD = 5.0 V时, 精度校正到± 1 %
    - ◆ 在-40 °C ~ +85 °C 和 VDD = 2.5 V ~ 5.5 V范围内, 精度为± 3 %
  - 内部低功耗 10 KHz 低速振荡器用于看门狗及掉电模式唤醒等功能
  - 支持一组PLL, 高至 50 MHz, 用于高性能的系统运行
  - 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
  - 四种 I/O 模式:
    - ◆ 准双向模式
    - ◆ 推挽输出模式
    - ◆ 开漏输出模式
    - ◆ 高阻输入模式
  - TTL/Schmitt 触发输入可选
  - I/O 管脚可被配置为边沿/电平触发模式的中断源
- 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 one-shot, periodic, toggle and continuous 计数操作模式
- 支持事件计数功能
- 支持输入捕捉功能
- Watchdog Timer
  - 多路时钟源
  - 从1.6ms 到26.0sec 有8个可选的定时溢出周期(取决于所选的时钟源)
  - WDT 可用作掉电模式/空闲模式的唤醒
  - 看门狗定时溢出的中断/复位选择
- Window Watchdog Timer
  - 带11-bit预分频的6-bit下数计数器，用于宽范围的窗口选择
- 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
  - 支持唤醒功能
- PWM/Capture
  - 内建四个16位PWM产生器，可输出8路PWM或4组互补配对PWM
  - 每个PWM产生器配有一个时钟源选择器，一个时钟分频器，一个8位时钟预分频和一个用于互补配对PWM的死区发生器
  - 8路16位捕捉定时器（共享PWM定时器）提供8路输入的上升/下降沿的捕捉功能
  - 支持捕捉(Capture)中断
- UART
  - 最多三组UART控制器
  - 支持流控 (TXD, RXD, CTS 和 RTS)
  - UART0 带 64-字节 FIFO 用于高速模式
  - UART1/2 (可选) 带16-字节 FIFO 用于标准模式
  - 支持 IrDA (SIR) 和 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频率高至1/16系统时钟
  - 支持 PDMA 模式
- SPI
  - 最多四组SPI控制器
  - 主机速率高至36 MHz (芯片工作在 5V 状态时)
  - 从机速率高至18 MHz (芯片工作在 5V 状态时)
  - 支持 SPI 主机/从机模式
  - 全双工同步串行数据传输
  - 可变数据长度 (从8位到 32 位) 传输模式
  - 可设置MSB 或LSB 在前的传输模式
  - 在时钟上升沿或下降沿接收还是发送是独立配置的
  - 当作为主机时2条从机片选线，作为从机时1条从机片选线
  - 支持 32-bit 传输模式下的字节睡眠模式
  - 支持 PDMA 模式

- 支持三线无从机选择信号的双向接口
- I<sup>2</sup>C
  - 最多支持2组 I<sup>2</sup>C 设备
  - 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
  - 可编程的时钟适用于不同速率控制
  - I<sup>2</sup>C总线上支持多地址识别（4个从机地址带mask选项）
  - 支持唤醒功能
- I<sup>2</sup>S
  - 外部音频 CODEC 接口
  - 可作主机也可作从机模式
  - 能处理 8, 16, 24 和 32 位字长的数据
  - 支持单声道和立体声的音频数据
  - 支持I<sup>2</sup>S 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DMA请求，一组用于发送，另一组用于接收
- PS/2 设备控制器
  - 禁止 Host 通信和请求发送检测
  - 接收帧错误检测
  - 可编程的 1 到 16 字节的发送缓冲以减少CPU的负担
  - 数据接收的双缓冲
  - 软件可控总线
- ADC
  - 12位ADC，转换速率达760 kSPS
  - 最多 8 通道单端模式输入或4 通道差分模式输入
  - 单一扫描模式/单周期扫描模式/连续扫描模式
  - 每个通道有独立的结果寄存器
  - 扫描使能通道
  - 阈电压侦测
  - 软件编程或外部管脚触发开始转换
  - 支持 PDMA 模式
- 模拟比较器(Analog Comparator)
  - 2组模拟比较器模块
  - 负端电位可选择外部输入或内部频带间隙电压
  - 比较结果改变可作为中断触发条件
  - 支持掉电模式唤醒功能
- 96-bit 唯一 ID (UID)
- 内建温度传感器， 1°C 分辨率
- 欠压检测(Brown-Out detector)
  - 支持四级检测电压： 4.4 V/3.7 V/2.7 V/2.2 V

- 支持欠压中断和复位选择
- 低压复位
  - 阈电压: 2.0 V
- 工作温度: -40°C~85°C
- 封装:
  - 无铅封装(RoHS)
  - LQFP 100-pin / 64-pin / 48-pin

## 2.2 NuMicro™ NUC120 特征– USB Line

- 内核
  - ARM® Cortex™-M0 内核最高运行 50 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
- 内建 LDO, 宽电压工作范围 2.5 V 到 5.5 V
- Flash 存储器
  - 32K/64K/128K 字节 Flash 用于存储程序代码
  - 4KB flash 用于存储ISP引导代码
  - 支持在系统编程 (ISP)和在应用编程(IAP)方式更新应用程序
  - 支持512 字节单页擦除
  - 在128K字节系统中可配置数据FLASH地址和大小, 在32K字节和64K字节系统中固定为 4K字节数据Flash
  - 通过SWD/ICE接口, 支持2 线 ICP升级方式
  - 支持外部编程器并行高速编程模式
- SRAM 存储器
  - 8K/16K 字节内建 SRAM
  - 支持 PDMA 模式
- PDMA (Peripheral DMA)
  - 支持9通道 PDMA 用于SRAM和周边设备的自动数据传输
  - 支持带4个通用多项式的CRC 运算, CRC-CCITT, CRC-8, CRC-16 和 CRC-32
- 时钟控制
  - 针对不同应用可灵活选择时钟
  - 内部 22.1184 MHz 高速振荡器可用于系统运行
    - ◆ 在+25 °C, VDD = 5.0 V时, 精度校正到± 1 %
    - ◆ 在-40 °C ~ +85 °C 和 VDD = 2.5 V ~ 5.5 V范围内, 精度为± 3 %
  - 内部低功耗 10 KHz 低速振荡器用于看门狗及掉电模式唤醒等功能
  - 支持一组PLL, 高至 50 MHz, 用于高性能的系统运行
  - 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
  - 四种 I/O 模式:
    - ◆ 准双向模式
    - ◆ 推挽输出模式
    - ◆ 开漏输出模式
    - ◆ 高阻输入模式
  - TTL/Schmitt 触发输入可选
  - I/O 管脚可被配置为边沿/电平触发模式的中断源
- 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 one-shot, periodic, toggle and continuous 计数操作模式

- 支持事件计数功能
- 支持输入捕捉功能
- Watchdog Timer
  - 多路时钟源
  - 从1.6ms 到26.0sec 有8个可选的定时溢出周期(取决于所选的时钟源)
  - WDT 可用作掉电模式/空闲模式的唤醒
  - 看门狗定时溢出的中断/复位选择
- Window Watchdog Timer
  - 带11-bit预分频的6-bit下数计数器，用于宽范围的窗口选择
- 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
  - 支持唤醒功能
- PWM/Capture
  - 内建四个16位PWM产生器，可输出8路PWM或4组互补配对PWM
  - 每个PWM产生器配有一个时钟源选择器，一个时钟分频器，一个8位时钟预分频和一个用于互补配对PWM的死区发生器
  - 8路16位捕捉定时器（共享PWM定时器）提供8路输入的上升/下降沿的捕捉功能
  - 支持捕捉(Capture)中断
- UART
  - 最多三组UART控制器
  - 支持流控 (TXD, RXD, CTS 和 RTS)
  - UART0 带 64-字节 FIFO 用于高速模式
  - UART1/2 （可选）带16-字节 FIFO 用于标准模式
  - 支持 IrDA (SIR) 和 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频率高至1/16系统时钟
  - 支持 PDMA 模式
- SPI
  - 最多支持4组SPI
  - 主机速率高至36 MHz (芯片工作在 5V 状态时)
  - 从机速率高至18 MHz (芯片工作在 5V 状态时)
  - 支持 SPI 主机/从机模式
  - 全双工同步串行数据传输
  - 可变数据长度（从8位到 32 位）传输模式
  - 可设置MSB 或LSB 在前的传输模式
  - 在时钟上升沿或下降沿接收还是发送是独立配置的
  - 当作为主机时2条从机片选线，作为从机时1条从机片选线
  - 支持 32-bit 传输模式下的字节睡眠模式
  - 支持 PDMA 模式
  - 支持三线无从机选择信号的双向接口

- I<sup>2</sup>C
  - 最多支持2组 I<sup>2</sup>C 设备
  - 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
  - 可编程的时钟适用于不同速率控制
  - I<sup>2</sup>C总线上支持多地址识别（4个从机地址带mask选项）
  - 支持唤醒功能
- I<sup>2</sup>S
  - 外部音频 CODEC 接口
  - 可作主机也可作从机模式
  - 能处理 8, 16, 24 和 32 位字长的数据
  - 支持单声道和立体声的音频数据
  - 支持I<sup>2</sup>S 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DMA请求，一组用于发送，另一组用于接收
- PS/2 设备控制器
  - 禁止 Host 通信和请求发送检测
  - 接收帧错误检测
  - 可编程的 1 到 16 字节的发送缓冲以减少CPU的负担
  - 数据接收的双缓冲
  - 软件可控总线
- USB 2.0全速设备
  - 一组12Mbps的USB 2.0 FS 设备
  - 片内集成USB收发模块
  - 提供1组中断源，提供四个中断事件
  - 支持控制传输(Control)，批量传输(Bulk In/Out)，中断传输(Interrupt)及同步传输
  - 当总线上无信号达到3ms时，具有自动暂停的功能
  - 支持6组可编程端点(endpoints)
  - 512 字节内部 SRAM 作为 USB 的缓存区
  - 支持远程唤醒功能
- ADC
  - 12位ADC，转换速率达760 kSPS
  - 最多 8 通道单端模式输入或4 通道差分模式输入
  - 单一扫描模式/单周期扫描模式/连续扫描模式
  - 每个通道有独立的结果寄存器
  - 扫描使能通道
  - 阈电压侦测
  - 软件编程或外部管脚触发开始转换
  - 支持 PDMA 模式
- 模拟比较器(Analog Comparator)
  - 2组模拟比较器模块
  - 负端电位可选择外部输入或内部频带间隙电压

- 比较结果改变可作为中断触发条件
- 支持掉电模式唤醒功能
- 96-bit 唯一 ID (UID)
- 内建温度传感器，1°C 分辨率
- 欠压检测(Brown-Out detector)
  - 支持四级检测电压：4.4 V/3.7 V/2.7 V/2.2 V
  - 支持欠压中断和复位选择
- 低压复位
  - 阈电压：2.0 V
- 工作温度：-40°C~85°C
- 封装：
  - 无铅封装(RoHS)
  - LQFP 100-pin / 64-pin / 48-pin

编号信息列表及管脚名称定义

### 2.3 NuMicro™ NUC100/120xxxDN 产品选型指南

#### 2.3.1 NuMicro™ NUC100 Advanced Line 选型指南

Part number	APROM	RAM	Data Flash	ISP Loader ROM	I/O	Timer	Connectivity						I <sup>2</sup> S	Comp.	PWM	ADC	RTC	EBI	ISP ICP	Package
							UART	SPI	I <sup>2</sup> C	USB	LIN	CAN								
NUC100LC1DN	32 KB	4 KB	4 KB	4 KB	up to 37	4x32-bit	2	1	2	-	-	-	1	1	6	8x12-bit	v	-	v	LQFP48
NUC100LD2DN	64 KB	8 KB	4 KB	4 KB	up to 37	4x32-bit	2	1	2	-	-	-	1	1	6	8x12-bit	v	-	v	LQFP48
NUC100LE3DN	128 KB	16 KB	Definable	4 KB	up to 37	4x32-bit	2	1	2	-	-	-	1	1	6	8x12-bit	v	-	v	LQFP48
NUC100RC1DN	32 KB	4 KB	4 KB	4 KB	up to 51	4x32-bit	3	2	2	-	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC100RD1DN	64 KB	4 KB	4 KB	4 KB	up to 51	4x32-bit	3	2	2	-	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC100RD2DN	64 KB	8 KB	4 KB	4 KB	up to 51	4x32-bit	3	2	2	-	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC100RE3DN	128 KB	16 KB	Definable	4 KB	up to 51	4x32-bit	3	2	2	-	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC100VE3DN	128 KB	16 KB	Definable	4 KB	up to 84	4x32-bit	3	4	2	-	-	-	1	2	8	8x12-bit	v	v	v	LQFP100

#### 2.3.2 NuMicro™ NUC120 USB Line 选型指南

Part number	APROM	RAM	Data Flash	ISP Loader ROM	I/O	Timer	Connectivity						I <sup>2</sup> S	Comp.	PWM	ADC	RTC	EBI	ISP ICP	Package
							UART	SPI	I <sup>2</sup> C	USB	LIN	CAN								
NUC120LC1DN	32 KB	4 KB	4 KB	4 KB	up to 33	4x32-bit	2	1	2	1	-	-	1	1	4	8x12-bit	v	-	v	LQFP48
NUC120LD2DN	64 KB	8 KB	4 KB	4 KB	up to 33	4x32-bit	2	1	2	1	-	-	1	1	4	8x12-bit	v	-	v	LQFP48
NUC120LE3DN	128 KB	16 KB	Definable	4 KB	up to 33	4x32-bit	2	1	2	1	-	-	1	1	4	8x12-bit	v	-	v	LQFP48
NUC120RC1DN	32 KB	4 KB	4 KB	4 KB	up to 47	4x32-bit	2	2	2	1	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC120RD2DN	64 KB	8 KB	4 KB	4 KB	up to 47	4x32-bit	2	2	2	1	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC120RE3DN	128 KB	16 KB	Definable	4 KB	up to 47	4x32-bit	2	2	2	1	-	-	1	2	6	8x12-bit	v	v	v	LQFP64
NUC120VE3DN	128 KB	16 KB	Definable	4 KB	up to 80	4x32-bit	3	4	2	1	-	-	1	2	8	8x12-bit	v	v	v	LQFP100



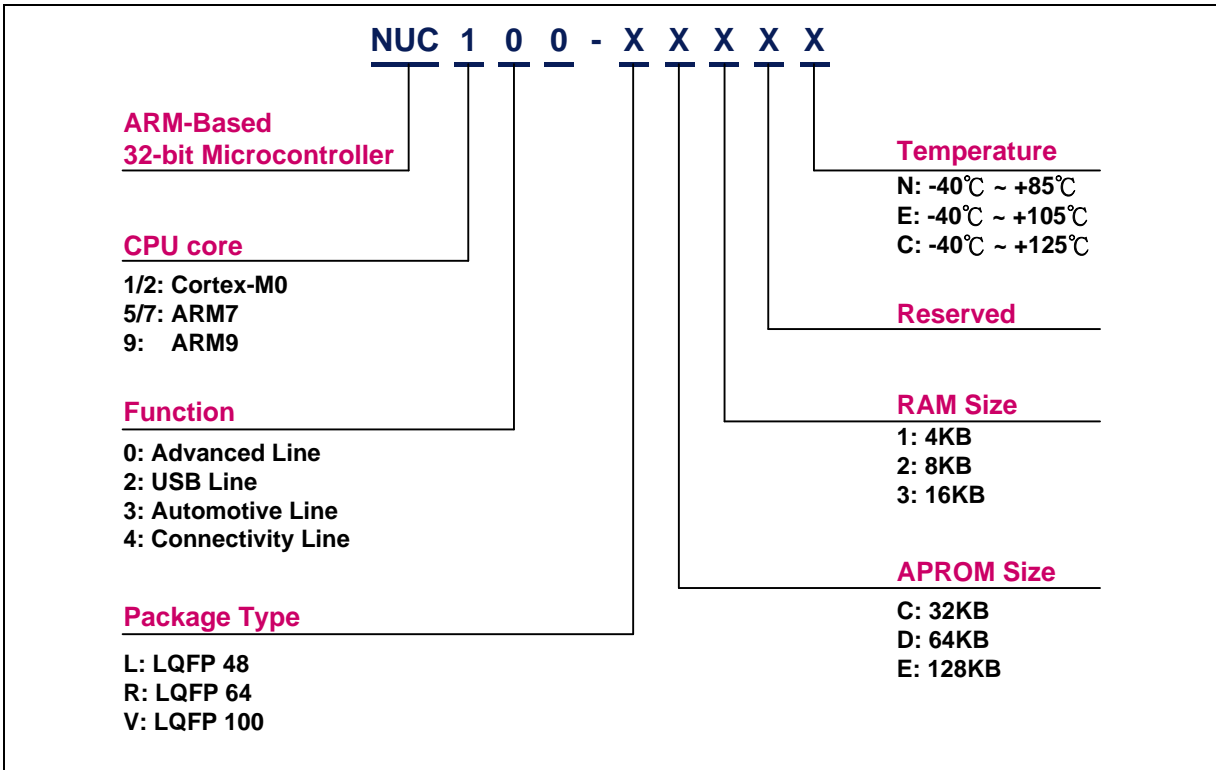


图 2-1 NuMicro™ NUC100 系列选型码

2.4 管脚配置

2.4.1 NuMicro™ NUC100 管脚图

2.4.1.1 NuMicro™ NUC100VxxDN LQFP 100-pin

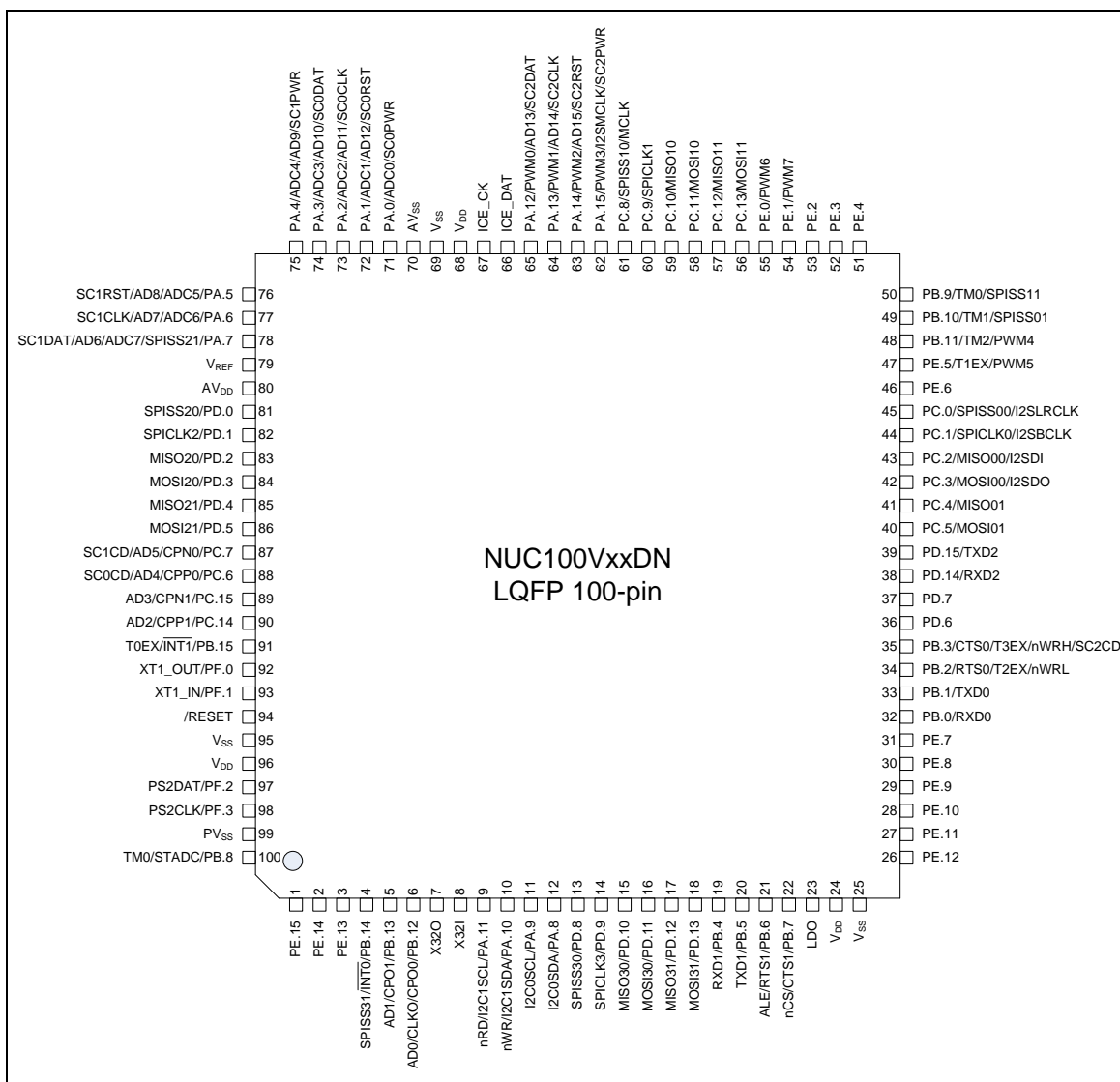


图 2-2 NuMicro™ NUC100VxxDN LQFP 100-pin 管脚图

2.4.1.2 NuMicro™ NUC100RxxDN LQFP 64-pin

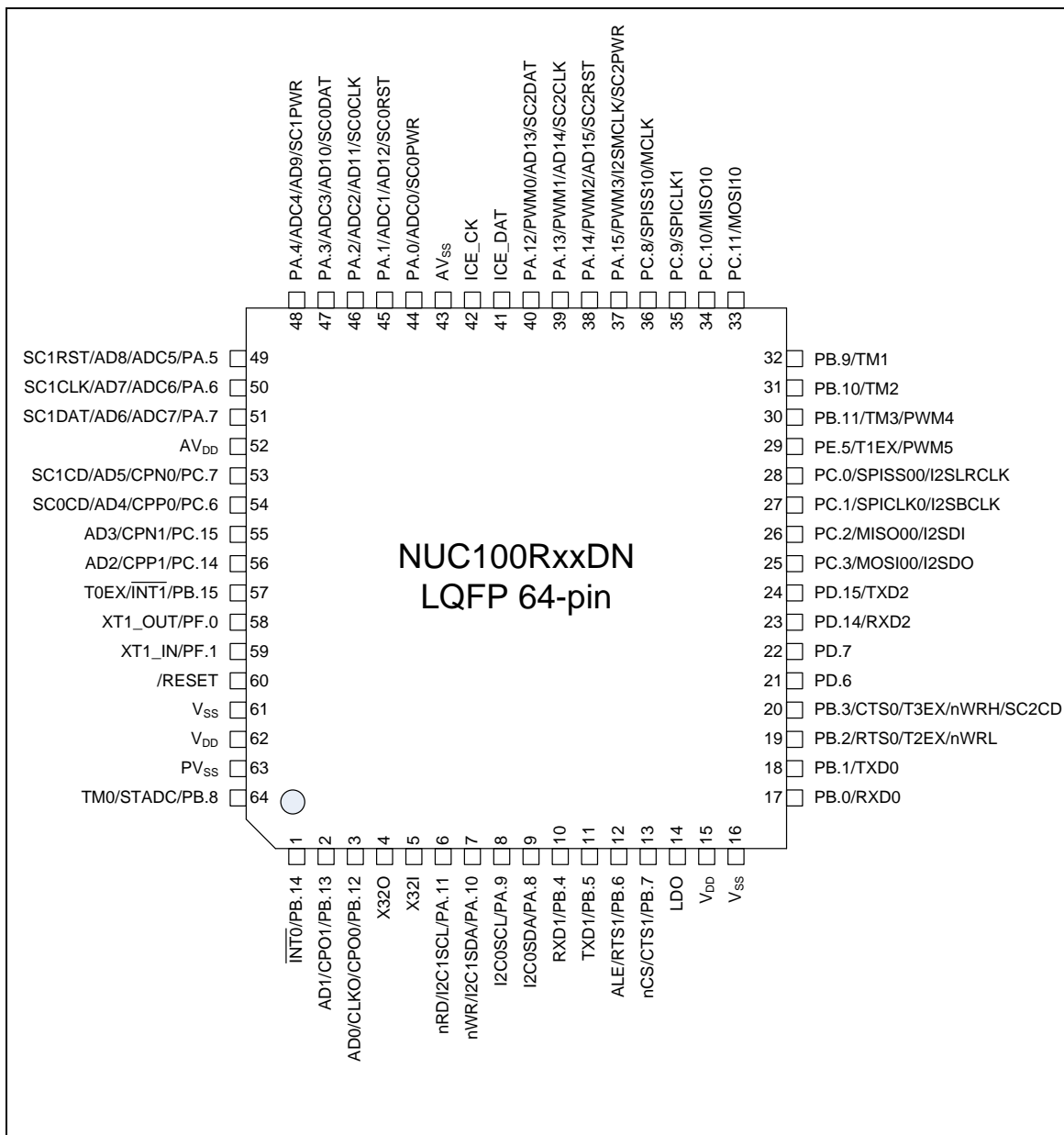


图 2-3 NuMicro™ NUC100SxxDN LQFP 64-pin 管脚图

2.4.1.3 NuMicro™ NUC100LxxDN LQFP 48-pin

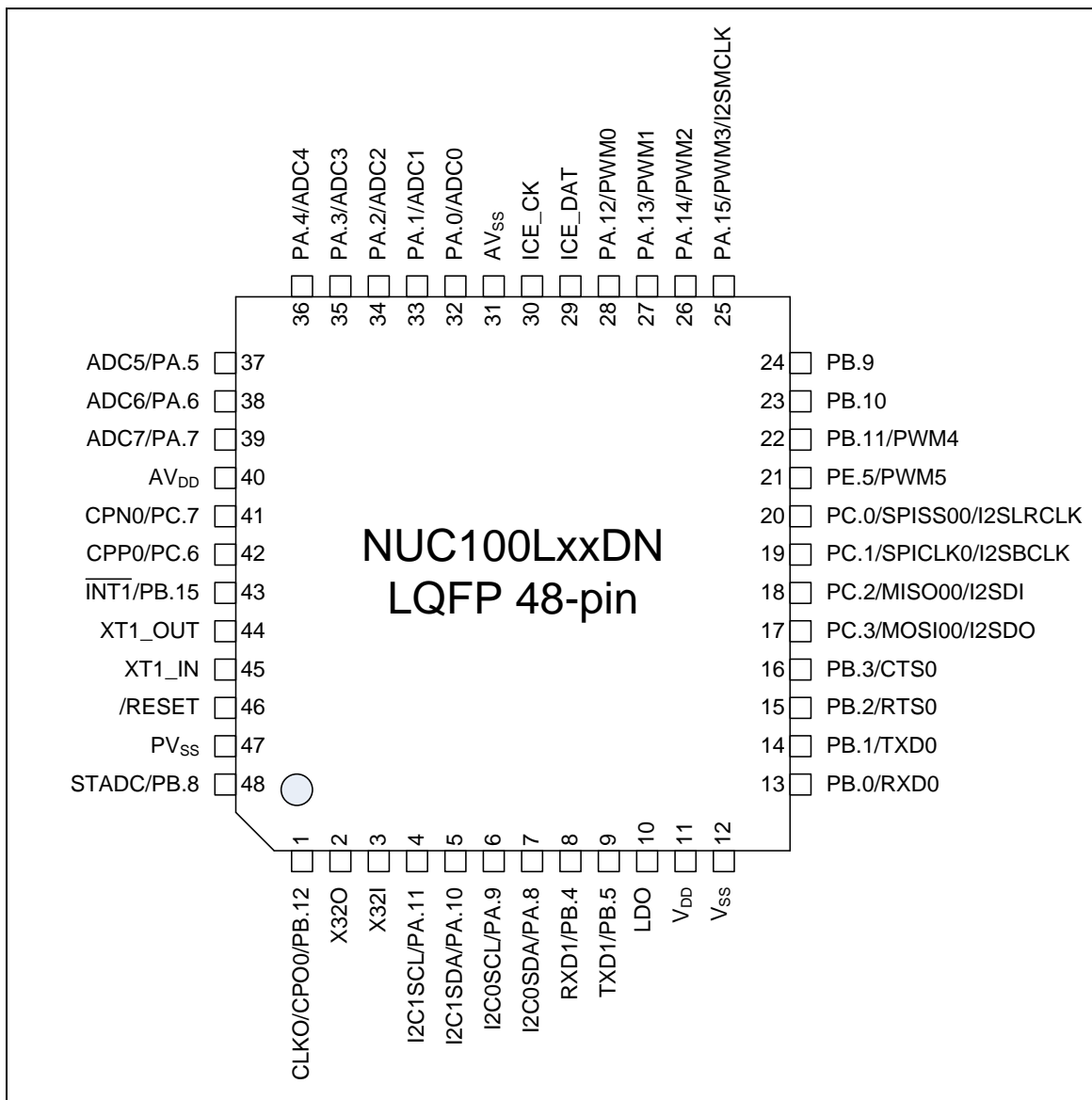


图 2-4 NuMicro™ NUC100LxxDN LQFP 48-pin 管脚图

2.4.2 NuMicro™ NUC120 管脚图

2.4.2.1 NuMicro™ NUC120VxxDN LQFP 100-pin

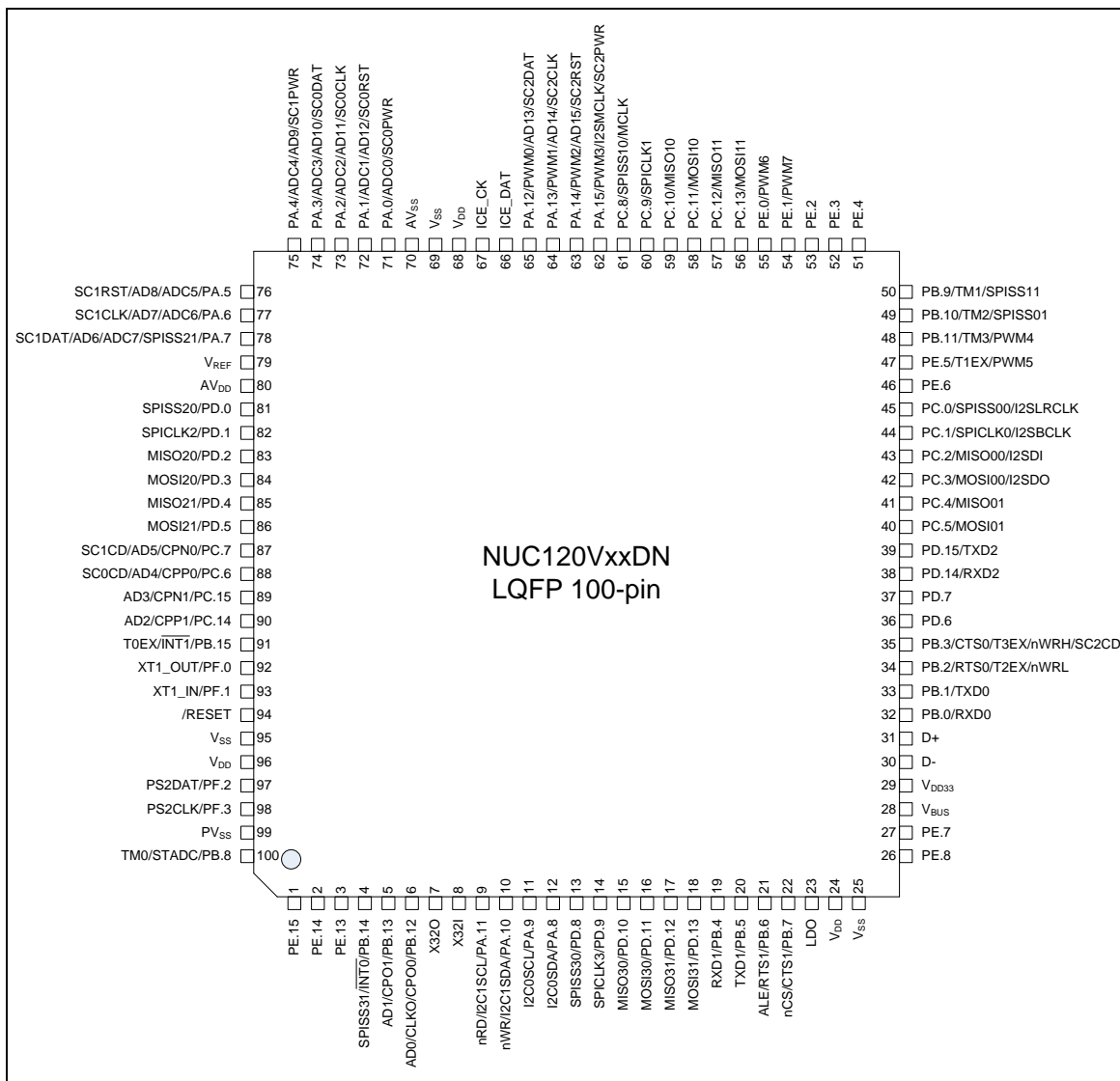


图 2-5 NuMicro™ NUC120VxxDN LQFP 100-pin 管脚图

2.4.2.2 NuMicro™ NUC120RxxDN LQFP 64-pin

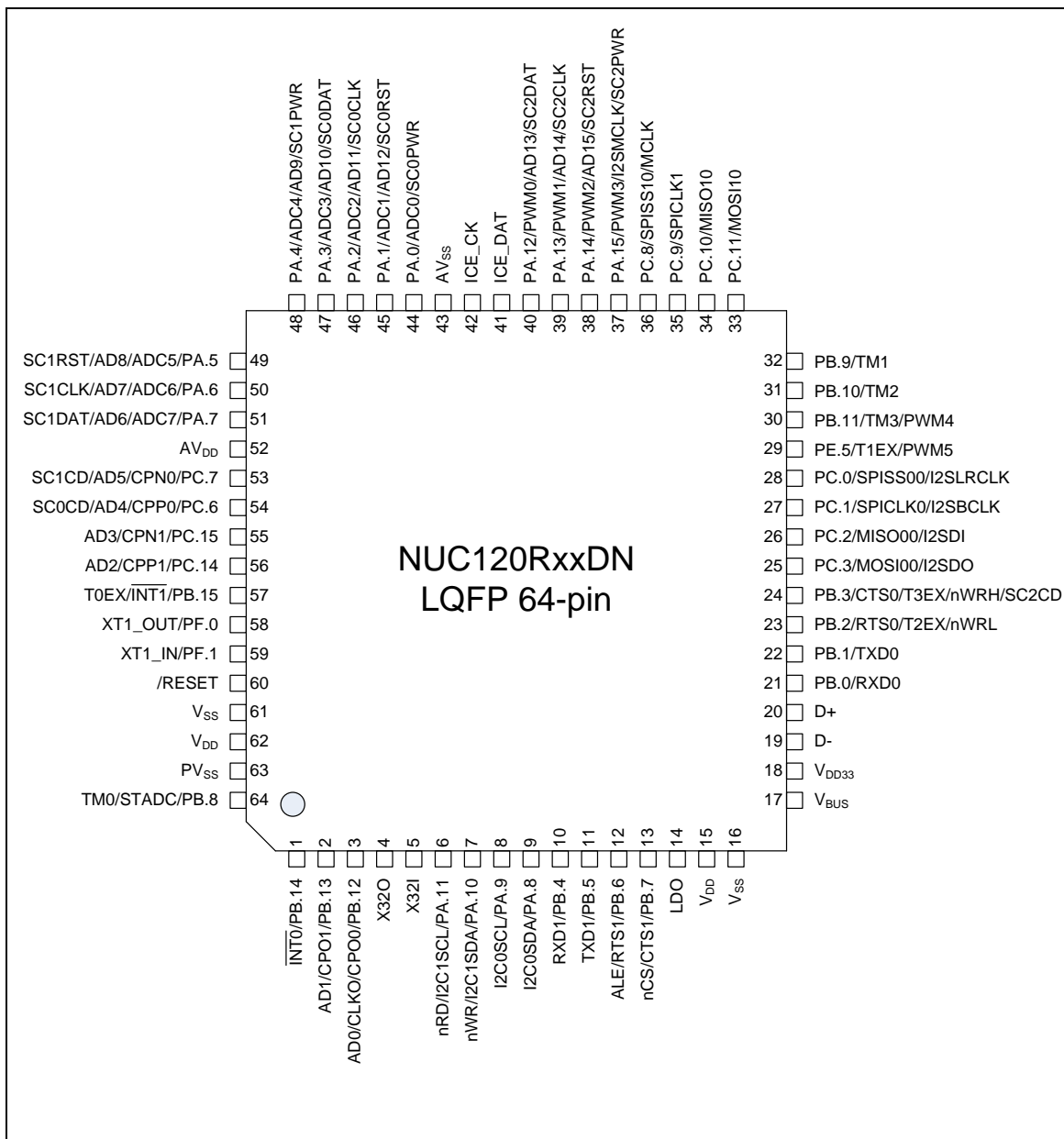


图 2-6 NuMicro™ NUC120SxxDN LQFP 64-pin 管脚图

2.4.2.3 NuMicro™ NUC120LxxDN LQFP 48-pin

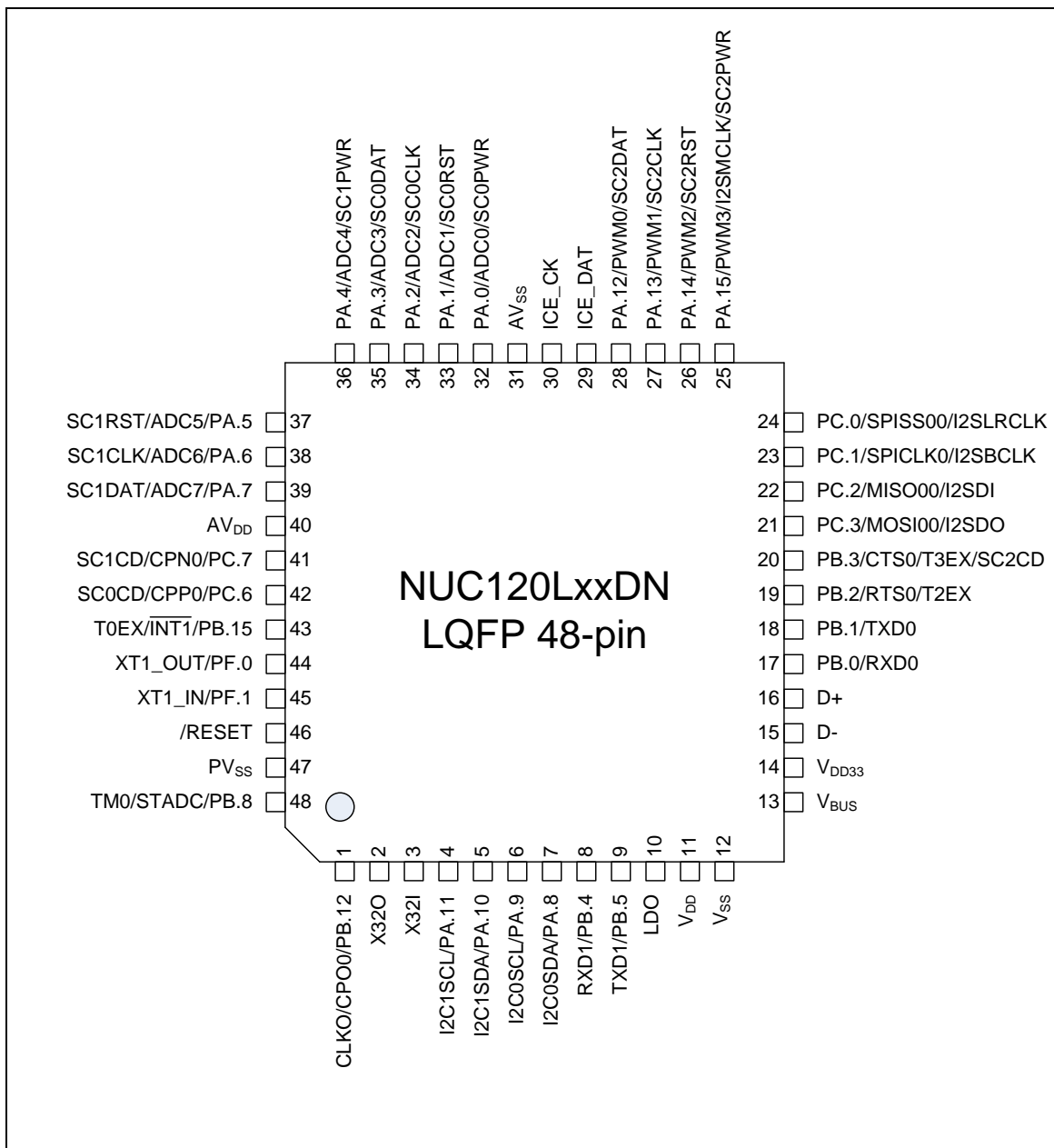


图 2-7 NuMicro™ NUC120LxxDN LQFP 48-pin 管脚图

2.5 管脚描述

2.5.1 NuMicro™ NUC100 管脚描述

管脚号			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
1			PE.15	I/O	通用数字输入/输出管脚
2			PE.14	I/O	通用数字输入/输出管脚
3			PE.13	I/O	通用数字输入/输出管脚
4	1		PB.14	I/O	通用数字输入/输出管脚
			/INT0	I	/INT0: 外部中断1输入脚
			SPISS31	I/O	SPISS31: SPI3 2 <sup>nd</sup> 从机选择脚
5	2		PB.13	I/O	通用数字输入/输出管脚
			CPO1	O	比较器1 输出脚
6	3	1	PB.12	I/O	通用数字输入/输出管脚
			CPO0	O	比较器0 输出脚
			CLKO	O	分频器输出脚
7	4	2	X32O	O	32.768 kHz crystal输出脚
8	5	3	X32I	I	32.768 kHz crystal 输入脚
9	6	4	PA.11	I/O	通用数字输入/输出管脚
			I2C1SCL	I/O	I2C1SCL: I <sup>2</sup> C1时钟脚
10	7	5	PA.10	I/O	通用数字输入/输出管脚
			I2C1SDA	I/O	I2C1SDA: I <sup>2</sup> C1数据输入/输出脚
11	8	6	PA.9	I/O	通用数字输入/输出管脚
			I2C0SCL	I/O	I2C0SCL: I <sup>2</sup> C0时钟脚
12	9	7	PA.8	I/O	通用数字输入/输出管脚
			I2C0SDA	I/O	I2C0SDA: I <sup>2</sup> C0数据输入/输出脚
13			PD.8	I/O	通用数字输入/输出管脚
			SPISS30	I/O	SPISS30: SPI3从机选择脚
14			PD.9	I/O	通用数字输入/输出管脚
			SPICLK3	I/O	SPICLK3: SPI3 串行时钟脚
15			PD.10	I/O	通用数字输入/输出管脚
			MISO30	I	MISO30: SPI3 MISO(主机输入, 从机输出) pin



管脚号			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
16			PD.11	I/O	通用数字输入/输出管脚
			MOSI30	O	MOSI30: SPI3 MOSI(主机输出, 从机输入) pin
17			PD.12	I/O	通用数字输入/输出管脚
			MISO31	I	MISO31: SPI3 2 <sup>nd</sup> MISO(主机输入, 从机输出) pin
18			PD.13	I/O	通用数字输入/输出管脚
			MOSI31	O	MOSI31: SPI3 2 <sup>nd</sup> MOSI(主机输出, 从机输入) pin
19	10	8	PB.4	I/O	通用数字输入/输出管脚
			RXD1	I	RXD1: UART1数据接收输入脚
20	11	9	PB.5	I/O	通用数字输入/输出管脚
			TXD1	O	TXD1: 数据发送输出脚
21	12		PB.6	I/O	通用数字输入/输出管脚
			RTS1		RTS1: UART1请求发送输出脚
22	13		PB.7	I/O	通用数字输入/输出管脚
			CTS1		CTS1: UART1清发送输入脚
23	14	10	LDO	P	LDO输出脚
24	15	11	VDD	P	电源供应引脚, 为IO端口、内部PLL电路LDO源和数字功能提供电源
25	16	12	VSS	P	地
26			PE.12	I/O	通用数字输入/输出管脚
27			PE.11	I/O	通用数字输入/输出管脚
28			PE.10	I/O	通用数字输入/输出管脚
29			PE.9	I/O	通用数字输入/输出管脚
30			PE.8	I/O	通用数字输入/输出管脚
31			PE.7	I/O	通用数字输入/输出管脚
32	17	13	PB.0	I/O	通用数字输入/输出管脚
			RXD0	I	RXD0: UART0数据接收输入脚
33	18	14	PB.1	I/O	通用数字输入/输出管脚
			TXD0	O	TXD0: UART0数据发送输出脚
34	19	15	PB.2	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 100	LQFP 64	LQFP 48			
			RTS0		RTS0: UART0请求发送输出脚
35	20	16	PB.3	I/O	通用数字输入/输出管脚
			CTS0		CTS0: UART0清发送输入脚
36	21		PD.6	I/O	通用数字输入/输出管脚
37	22		PD.7	I/O	通用数字输入/输出管脚
38	23		PD.14	I/O	通用数字输入/输出管脚
			RXD2	I	RXD2: UART2数据接收输入脚
39	24		PD.15	I/O	通用数字输入/输出管脚
			TXD2	O	TXD2: UART2数据发送输出脚
40			PC.5	I/O	通用数字输入/输出管脚
			MOSI01	O	MOSI01: SPI0 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
41			PC.4	I/O	通用数字输入/输出管脚
			MISO01	I	MISO01: SPI0 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
42	25	17	PC.3	I/O	通用数字输入/输出管脚
			MOSI00	O	MOSI00: SPI0 MOSI (主机输出, 从机输入) pin
			I2SDO	O	I2SDO: I <sup>2</sup> S数据输出
43	26	18	PC.2	I/O	通用数字输入/输出管脚
			MISO00	I	MISO00: SPI0 MISO (主机输入, 从机输出) pin
			I2SDI	I	I2SDI: I <sup>2</sup> S数据输入
44	27	19	PC.1	I/O	通用数字输入/输出管脚
			SPICLK0	I/O	SPICLK0: SPI0串行时钟脚
			I2SBCLK	I/O	I2SBCLK: I <sup>2</sup> S bit时钟脚
45	28	20	PC.0	I/O	通用数字输入/输出管脚
			SPISS00	I/O	SPISS00: SPI0从机选择脚
			I2SLRCLK	I/O	I2SLRCLK: I <sup>2</sup> S左右声道时钟
46			PE.6	I/O	通用数字输入/输出管脚
47	29	21	PE.5	I/O	通用数字输入/输出管脚
			PWM5	O	PWM5: PWM输出

管脚号			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
48	30	22	PB.11	I/O	通用数字输入/输出管脚
			TM3	O	TM3: Timer3外部计数器输入
			PWM4	O	PWM4: PWM输出
49	31	23	PB.10	I/O	通用数字输入/输出管脚
			TM2	O	TM2: Timer2外部计数器输入
			SPISS01	I/O	SPISS01: SPI0 2 <sup>nd</sup> 从机选择脚
50	32	24	PB.9	I/O	通用数字输入/输出管脚
			TM1	O	TM1: Timer1外部计数器输入
			SPISS11	I/O	SPISS11: SPI1 2 <sup>nd</sup> 从机选择脚
51			PE.4	I/O	通用数字输入/输出管脚
52			PE.3	I/O	通用数字输入/输出管脚
53			PE.2	I/O	通用数字输入/输出管脚
54			PE.1	I/O	通用数字输入/输出管脚
			PWM7	O	PWM7: PWM输出
55			PE.0	I/O	通用数字输入/输出管脚
			PWM6	O	PWM6: PWM输出
56			PC.13	I/O	通用数字输入/输出管脚
			MOSI11	O	MOSI11: SPI1 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
57			PC.12	I/O	通用数字输入/输出管脚
			MISO11	I	MISO11: SPI1 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
58	33		PC.11	I/O	通用数字输入/输出管脚
			MOSI10	O	MOSI10: SPI1 MOSI (主机输出, 从机输入) pin
59	34		PC.10	I/O	通用数字输入/输出管脚
			MISO10	I	MISO10: SPI1 MISO (主机输入, 从机输出) pin
60	35		PC.9	I/O	通用数字输入/输出管脚
			SPICLK1	I/O	SPICLK1: SPI1串行时钟脚
61	36		PC.8	I/O	通用数字输入/输出管脚
			SPISS10	I/O	SPISS10: SPI1从机选择脚
62	37	25	PA.15	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 100	LQFP 64	LQFP 48			
			PWM3	O	PWM3: PWM输出脚
			I2SMCLK	O	I2SMCLK: I <sup>2</sup> S主时钟输出脚
63	38	26	PA.14	I/O	通用数字输入/输出管脚
			PWM2	O	PWM2: PWM输出
64	39	27	PA.13	I/O	通用数字输入/输出管脚
			PWM1	O	PWM1: PWM输出
65	40	28	PA.12	I/O	通用数字输入/输出管脚
			PWM0	O	PWM0: PWM输出
66	41	29	ICE_DAT	I/O	调试器的串行数据脚
67	42	30	ICE_CK	I	调试器的串行时钟脚
68			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
69			VSS	P	地
70	43	31	AVSS	AP	模拟电路地
71	44	32	PA.0	I/O	通用数字输入/输出管脚
			ADC0	AI	ADC0: ADC模拟输入
72	45	33	PA.1	I/O	通用数字输入/输出管脚
			ADC1	AI	ADC1: ADC模拟输入
73	46	34	PA.2	I/O	通用数字输入/输出管脚
			ADC2	AI	ADC2: ADC模拟输入
74	47	35	PA.3	I/O	通用数字输入/输出管脚
			ADC3	AI	ADC3: ADC 模拟输入
75	48	36	PA.4	I/O	通用数字输入/输出管脚
			ADC4	AI	ADC4: ADC 模拟输入
76	49	37	PA.5	I/O	通用数字输入/输出管脚
			ADC5	AI	ADC5: ADC模拟输入
77	50	38	PA.6	I/O	通用数字输入/输出管脚
			ADC6	AI	ADC6: ADC模拟输入
78	51	39	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 100	LQFP 64	LQFP 48			
			ADC7	AI	ADC7: ADC模拟输入
			SPISS21	I/O	SPISS21: SPI2 2 <sup>nd</sup> 从机选择脚
79			Vref	AP	ADC参考电压输入
80	52	40	AVDD	AP	内部模拟电路电源
81			PD.0	I/O	通用数字输入/输出管脚
			SPISS20	I/O	SPISS20: SPI2从机选择脚
82			PD.1	I/O	通用数字输入/输出管脚
			SPICLK2	I/O	SPICLK2: SPI2串行时钟脚
83			PD.2	I/O	通用数字输入/输出管脚
			MISO20	I	MISO20: SPI2 MISO (主机输入, 从机输出) pin
84			PD.3	I/O	通用数字输入/输出管脚
			MOSI20	O	MOSI20: SPI2 MOSI (主机输出, 从机输入) pin
85			PD.4	I/O	通用数字输入/输出管脚
			MISO21	I	MISO21: SPI2 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
86			PD.5	I/O	通用数字输入/输出管脚
			MOSI21	O	MOSI21: SPI2 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
87	53	41	PC.7	I/O	通用数字输入/输出管脚
			CPN0	I	CPN0: Comparator0负端输入脚
88	54	42	PC.6	I/O	通用数字输入/输出管脚
			CPP0	I	CPP0: Comparator0正端输入脚
89	55		PC.15	I/O	通用数字输入/输出管脚
			CPN1	I	CPN1: Comparator1负端输入脚
90	56		PC.14	I/O	通用数字输入/输出管脚
			CPP1	I	CPP1: Comparator1正端输入脚
91	57	43	PB.15	I/O	通用数字输入/输出管脚
			/INT1	I	/INT1: 外部中断0输入脚
92	58	44	XT1_OUT	O	外部 4~24 MHz 高速晶振输出脚
93	59	45	XT1_IN	I	外部 4~24 MHz 高速晶振输入脚

管脚号			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
94	60	46	/RESET	I	外部复位输入：低有效，置低时复位MCU至初始状态，带内部上拉。
95	61		VSS	P	地
96	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
97			PS2DAT	I/O	PS/2 数据脚
98			PS2CLK	I/O	PS/2 时钟脚
99	63	47	PVSS	I/O	PLL地
100	64	48	PB.8	I/O	通用数字输入/输出管脚
			STADC	I	STADC: ADC外部触发输入
			TM0	O	TM0: Timer0 外部计数器输入

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); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)

## 2.5.2 NuMicro™ NUC120 管脚描述

管脚号.			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
1			PE.15	I/O	通用数字输入/输出管脚
2			PE.14	I/O	通用数字输入/输出管脚
3			PE.13	I/O	通用数字输入/输出管脚
4	1		PB.14	I/O	通用数字输入/输出管脚
			/INT0	I	/INT0: 外部中断1输入脚
			SPISS31	I/O	SPISS31: SPI3 2 <sup>nd</sup> 从机选择脚
5	2		PB.13	I/O	通用数字输入/输出管脚
			CPO1	O	比较器1输出脚
6	3	1	PB.12	I/O	通用数字输入/输出管脚
			CPO0	O	比较器0输出脚
			CLKO	O	分频器输出脚
7	4	2	X32O	O	32.768 kHz crystal 输出脚
8	5	3	X32I	I	32.768 kHz crystal 输入脚
9	6	4	PA.11	I/O	通用数字输入/输出管脚
			I2C1SCL	I/O	I2C1SCL: I <sup>2</sup> C1时钟脚
10	7	5	PA.10	I/O	通用数字输入/输出管脚
			I2C1SDA	I/O	I2C1SDA: I <sup>2</sup> C1数据输入/输出脚
11	8	6	PA.9	I/O	通用数字输入/输出管脚
			I2C0SCL	I/O	I2C0SCL: I <sup>2</sup> C0时钟脚
12	9	7	PA.8	I/O	通用数字输入/输出管脚
			I2C0SDA	I/O	I2C0SDA: I <sup>2</sup> C0数据输入/输出脚
13			PD.8	I/O	通用数字输入/输出管脚
			SPISS30	I/O	SPISS30: SPI3从机选择脚
14			PD.9	I/O	通用数字输入/输出管脚
			SPICLK3	I/O	SPICLK3: SPI3串行时钟脚
15			PD.10	I/O	通用数字输入/输出管脚
			MISO30	I	MISO30: SPI3 MISO (主机输入, 从机输出) pin
16			PD.11	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 100	LQFP 64	LQFP 48			
			MOSI30	O	MOSI30: SPI3 MOSI (主机输出, 从机输入) pin
17			PD.12	I/O	通用数字输入/输出管脚
			MISO31	I	MISO31: SPI3 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
18			PD.13	I/O	通用数字输入/输出管脚
			MOSI31	O	MOSI31: SPI3 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
19	10	8	PB.4	I/O	通用数字输入/输出管脚
			RXD1	I	RXD1: UART1数据接收输入脚
20	11	9	PB.5	I/O	通用数字输入/输出管脚
			TXD1	O	TXD1: UART1数据发送输出脚
21	12		PB.6	I/O	通用数字输入/输出管脚
			RTS1		RTS1: UART1请求发送输出脚
22	13		PB.7	I/O	通用数字输入/输出管脚
			CTS1		CTS1: UART1清发送输入脚
23	14	10	LDO	P	LDO输出脚
24	15	11	VDD	P	电源供应引脚, 为IO端口、内部PLL电路LDO源和数字功能提供电源
25	16	12	VSS	P	地
26			PE.8	I/O	通用数字输入/输出管脚
27			PE.7	I/O	通用数字输入/输出管脚
28	17	13	VBUS	USB	USB HOST或HUB提供电源脚
29	18	14	VDD33	USB	内部3.3V电压输出引脚通用
30	19	15	D-	USB	USB 差分信号 D-
31	20	16	D+	USB	USB 差分信号 D+
32	21	17	PB.0	I/O	通用数字输入/输出管脚
			RXD0	I	RXD0: UART0数据接收输入脚
33	22	18	PB.1	I/O	通用数字输入/输出管脚
			TXD0	O	TXD0: UART0数据发送输出脚
34	23	19	PB.2	I/O	通用数字输入/输出管脚
			RTS0		RTS0: UART0请求发送输出脚



管脚号.			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
35	24	20	PB.3	I/O	通用数字输入/输出管脚
			CTS0		CTS0: UART0清发送输入脚
36			PD.6	I/O	通用数字输入/输出管脚
37			PD.7	I/O	通用数字输入/输出管脚
38			PD.14	I/O	通用数字输入/输出管脚
			RXD2	I	RXD2: UART2数据接收输入脚
39			PD.15	I/O	通用数字输入/输出管脚
			TXD2	O	TXD2: UART2数据发送输出脚
40			PC.5	I/O	通用数字输入/输出管脚
			MOSI01	O	MOSI01: SPI0 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
41			PC.4	I/O	通用数字输入/输出管脚
			MISO01	I	MISO01: SPI0 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
42	25	21	PC.3	I/O	通用数字输入/输出管脚
			MOSI00	O	MOSI00: SPI0 MOSI (主机输出, 从机输入) pin
			I2SDO	O	I2SDO: I <sup>2</sup> S数据输出
43	26	22	PC.2	I/O	通用数字输入/输出管脚
			MISO00	I	MISO00: SPI0 MISO (主机输入, 从机输出) pin
			I2SDI	I	I2SDI: I <sup>2</sup> S数据输入
44	27	23	PC.1	I/O	通用数字输入/输出管脚
			SPICLK0	I/O	SPICLK0: SPI0串行时钟脚
			I2SBCLK	I/O	I2SBCLK: I <sup>2</sup> S bit 时钟脚
45	28	24	PC.0	I/O	通用数字输入/输出管脚
			SPISS00	I/O	SPISS00: SPI0 从机选择脚
			I2SLRCLK	I/O	I2SLRCLK: I <sup>2</sup> S左右声道时钟
46			PE.6	I/O	通用数字输入/输出管脚
47	29		PE.5	I/O	通用数字输入/输出管脚
			PWM5	O	PWM5: PWM 输出
48	30		PB.11	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 100	LQFP 64	LQFP 48			
			TM3	O	TM3: Timer3 外部计数器输入
			PWM4	O	PWM4: PWM 输出
49	31		PB.10	I/O	通用数字输入/输出管脚
			TM2	O	TM2: Timer2 外部计数器输入
			SPISS01	I/O	SPISS01: SPI0 2 <sup>nd</sup> 从机选择脚
50	32		PB.9	I/O	通用数字输入/输出管脚
			TM1	O	TM1: Timer1 外部计数器输入
			SPISS11	I/O	SPISS11: SPI1 2 <sup>nd</sup> 从机选择脚
51			PE.4	I/O	通用数字输入/输出管脚
52			PE.3	I/O	通用数字输入/输出管脚
53			PE.2	I/O	通用数字输入/输出管脚
54			PE.1	I/O	通用数字输入/输出管脚
			PWM7	O	PWM7: PWM output
55			PE.0	I/O	通用数字输入/输出管脚
			PWM6	O	PWM6: PWM output
56			PC.13	I/O	通用数字输入/输出管脚
			MOSI11	O	MOSI11: SPI1 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
57			PC.12	I/O	通用数字输入/输出管脚
			MISO11	I	MISO11: SPI1 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
58	33		PC.11	I/O	通用数字输入/输出管脚
			MOSI10	O	MOSI10: SPI1 MOSI (主机输出, 从机输入) pin
59	34		PC.10	I/O	通用数字输入/输出管脚
			MISO10	I	MISO10: SPI1 MISO (主机输入, 从机输出)pin
60	35		PC.9	I/O	通用数字输入/输出管脚
			SPICLK1	I/O	SPICLK1: SPI1串行时钟脚
61	36		PC.8	I/O	通用数字输入/输出管脚
			SPISS10	I/O	SPISS10: SPI1从机选择脚
62	37	25	PA.15	I/O	通用数字输入/输出管脚
			PWM3	O	PWM3: PWM 输出脚

管脚号.			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
			I2SMCLK	O	I2SMCLK: I <sup>2</sup> S主机时钟输出脚
63	38	26	PA.14	I/O	通用数字输入/输出管脚
			PWM2	O	PWM2: PWM 输出
64	39	27	PA.13	I/O	通用数字输入/输出管脚
			PWM1	O	PWM1: PWM 输出
65	40	28	PA.12	I/O	通用数字输入/输出管脚
			PWM0	O	PWM0: PWM 输出
66	41	29	ICE_DAT	I/O	调试器的串行数据脚
67	42	30	ICE_CK	I	调试器的串行时钟脚
68			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
69			VSS	P	地
70	43	31	AVSS	AP	模拟电路地脚
71	44	32	PA.0	I/O	通用数字输入/输出管脚
			ADC0	AI	ADC0: ADC模拟输入
72	45	33	PA.1	I/O	通用数字输入/输出管脚
			ADC1	AI	ADC1: ADC 模拟输入
73	46	34	PA.2	I/O	通用数字输入/输出管脚
			ADC2	AI	ADC2: ADC模拟输入
74	47	35	PA.3	I/O	通用数字输入/输出管脚
			ADC3	AI	ADC3: ADC模拟输入
75	48	36	PA.4	I/O	通用数字输入/输出管脚
			ADC4	AI	ADC4: ADC模拟输入
76	49	37	PA.5	I/O	通用数字输入/输出管脚
			ADC5	AI	ADC5: ADC模拟输入
77	50	38	PA.6	I/O	通用数字输入/输出管脚
			ADC6	AI	ADC6: ADC模拟输入
78	51	39	PA.7	I/O	通用数字输入/输出管脚
			ADC7	AI	ADC7: ADC模拟输入

管脚号.			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
			SPISS21	I/O	SPISS21: SPI2 2 <sup>nd</sup> 从机选择脚
79			Vref	AP	ADC参考电压输入
80	52	40	AVDD	AP	内部模拟电路电源
81			PD.0	I/O	通用数字输入/输出管脚
			SPISS20	I/O	SPISS20: SPI2从机选择脚
82			PD.1	I/O	通用数字输入/输出管脚
			SPICLK2	I/O	SPICLK2: SPI2串行时钟脚
83			PD.2	I/O	通用数字输入/输出管脚
			MISO20	I	MISO20: SPI2 MISO (主机输入, 从机输出) pin
84			PD.3	I/O	通用数字输入/输出管脚
			MOSI20	O	MOSI20: SPI2 MOSI (主机输出, 从机输入) pin
85			PD.4	I/O	通用数字输入/输出管脚
			MISO21	I	MISO21: SPI2 2 <sup>nd</sup> MISO (主机输入, 从机输出) pin
86			PD.5	I/O	通用数字输入/输出管脚
			MOSI21	O	MOSI21: SPI2 2 <sup>nd</sup> MOSI (主机输出, 从机输入) pin
87	53	41	PC.7	I/O	通用数字输入/输出管脚
			CPN0	I	CPN0: Comparator0 负端输入脚
88	54	42	PC.6	I/O	通用数字输入/输出管脚
			CPP0	I	CPP0: Comparator0 正端输入脚
89	55		PC.15	I/O	通用数字输入/输出管脚
			CPN1	I	CPN1: Comparator1 负端输入脚
90	56		PC.14	I/O	通用数字输入/输出管脚
			CPP1	I	CPP1: Comparator1 正端输入脚
91	57	43	PB.15	I/O	通用数字输入/输出管脚
			/INT1	I	/INT1: 外部 中断0输入脚
92	58	44	XT1_OUT	O	外部 4~24 MHz 高速晶振输出脚
93	59	45	XT1_IN	I	外部 4~24 MHz 高速晶振输入脚
94	60	46	/RESET	I	外部复位输入: 低有效, 置低时复位MCU为初始状态, 带内部上拉.

管脚号.			管脚名称	管脚类型	描述
LQFP 100	LQFP 64	LQFP 48			
95	61		VSS	P	地
96	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
97			PS2DAT	I/O	PS/2 数据脚
98			PS2CLK	I/O	PS/2 时钟脚
99	63	47	PVSS	I/O	PLL 地
100	64	48	PB.8	I/O	通用数字输入/输出管脚
			STADC	I	STADC: ADC 外部触发输入.
			TM0	O	TM0: Timer0 外部计数器输入

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); P=电源管脚 (Power Pin); AP= 模拟电源 (Analog Power)

3 框图

3.1 NuMicro™ NUC100 框图

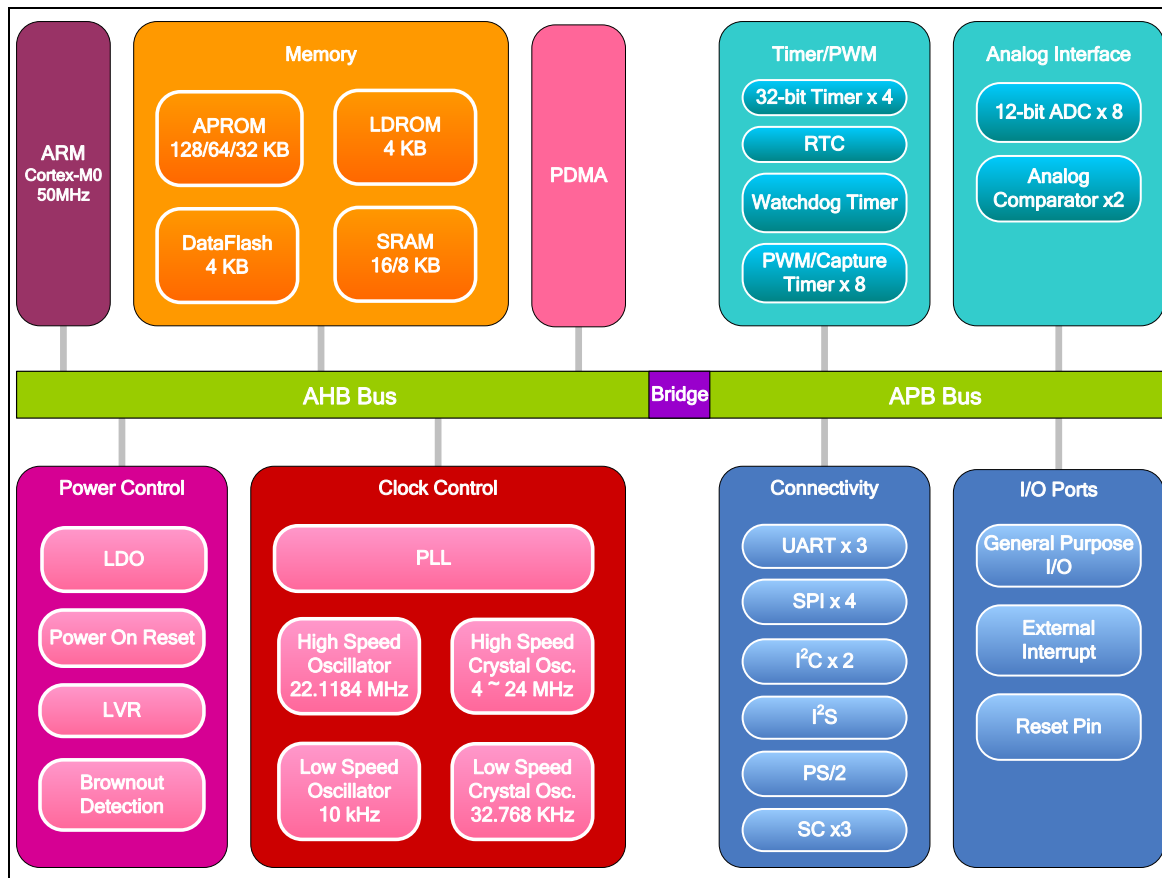


图 3-1 NuMicro™ NUC100 框图

3.2 NuMicro™ NUC120 框图

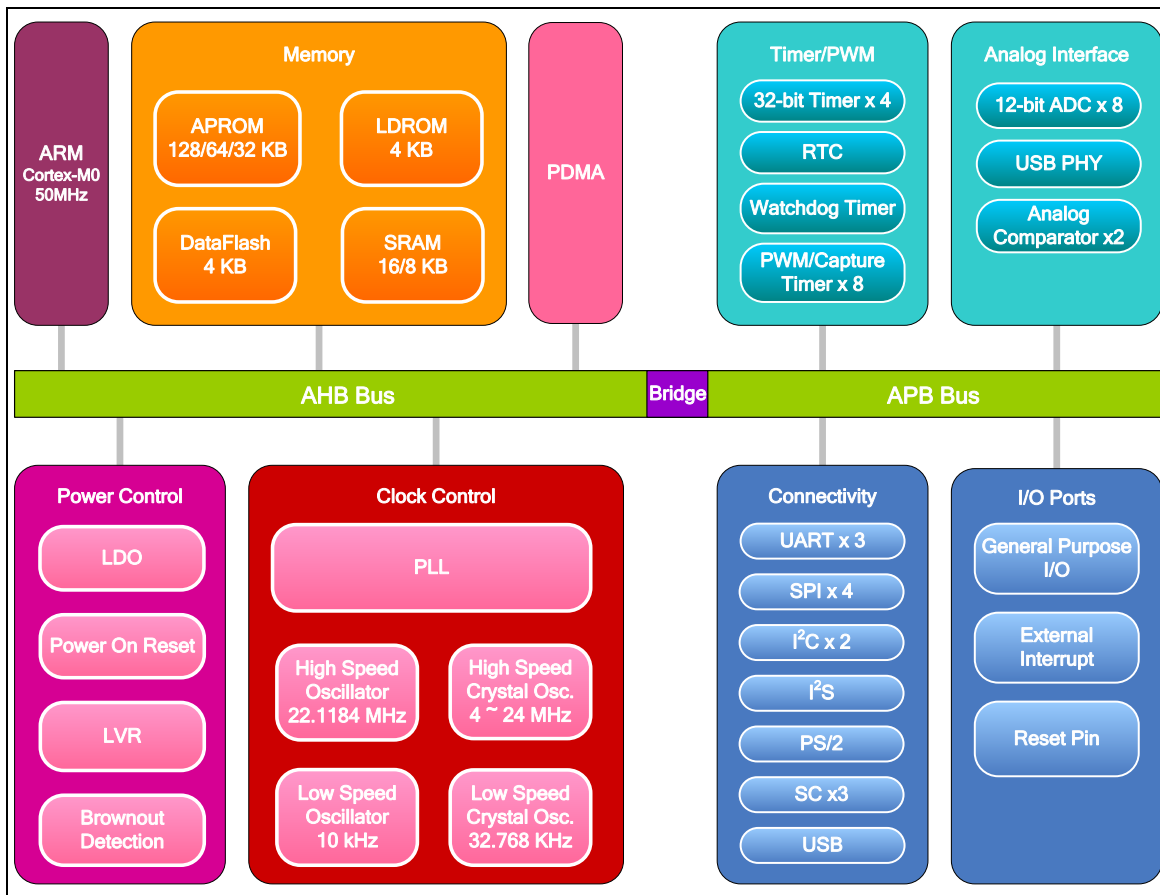


图 3-2 NuMicro™ NUC120 框图

## 4 功能描述

### 4.1 ARM® Cortex™-M0 内核

The 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。图 4-1为处理器的功能图。.

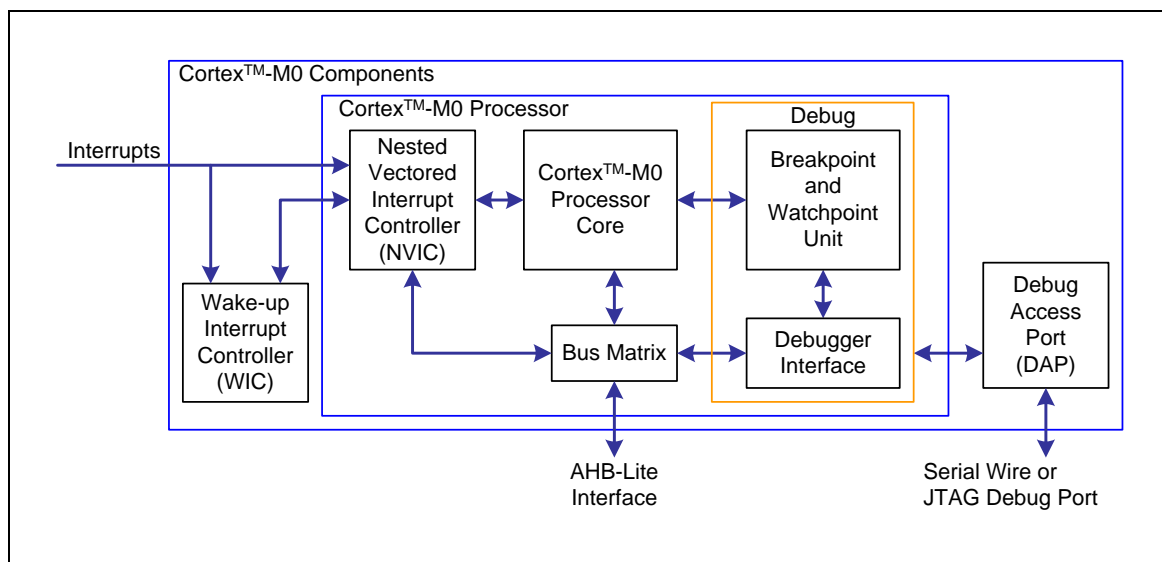


图 4-1 功能框图

设备提供:

- 低门数处理器特征:
  - ◆ ARMv6-M Thumb® 指令集
  - ◆ Thumb-2 技术
  - ◆ ARMv6-M 兼容 24-bit SysTick 定时器
  - ◆ 32-bit 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 的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 特征:



- ◆ 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）
- ◆ 同时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：
  - ◆ 提供简单的集成到所有系统外设和存储器的单一32位 AMBA-3 ABH-Lite 系统接口
  - ◆ 支持DAP (Debug Access Port) 的单一32位的从机端口

## 4.2 系统管理器

### 4.2.1 概述

系统管理器包括如下功能：

- 系统复位
- 系统内存映射
- 产品ID、芯片复位、模块功能复位和多功能管脚控制的系统管理寄存器
- 系统定时器 (SysTick)
- 嵌套向量中断控制器 (NVIC)
- 系统控制寄存器

### 4.2.2 系统复位

下列任一情况发生时，系统复位，复位标志由寄存器 RSTSRC 读出。

- 上电复位
- 复位脚（/RESET）上有低电平
- 看门狗复位
- 低压复位
- 欠压检测器复位
- CPU 复位
- 系统复位

系统复位和上电复位使整个芯片复位，包括所有外设。系统复位与上电复位的区别在于外部晶振电路与 ISPCON.BS 位。系统复位不复位外部晶振电路和 ISPCON.BS 位，上电复位可以。

### 4.2.3 系统电源分配

该器件的电源分为四个部分：

- 由 AVDD 和 AVSS 提供的模拟电源，为模拟部分工作提供电压。
- 由 VDD 和 VSS 提供的数字电源，提供一个固定的 1.8V 的数字电源，用于数字操作和 I/O 管脚的内部稳压电源。
- VBUS 提供给 USB 的电源，用于 USB 模块传输操作。

内部电压调节器输出，LDO 和 VDD33，需要在相应的管脚上外接电容，并尽量靠近管脚摆放。模拟电源（AVDD）的电压电平必须和数字电源（VDD）的一样。图 4-2 为 NuMicro™ NUC100 的电源分配图，图 4-3 为 NuMicro™ NUC120 的电源分配图。

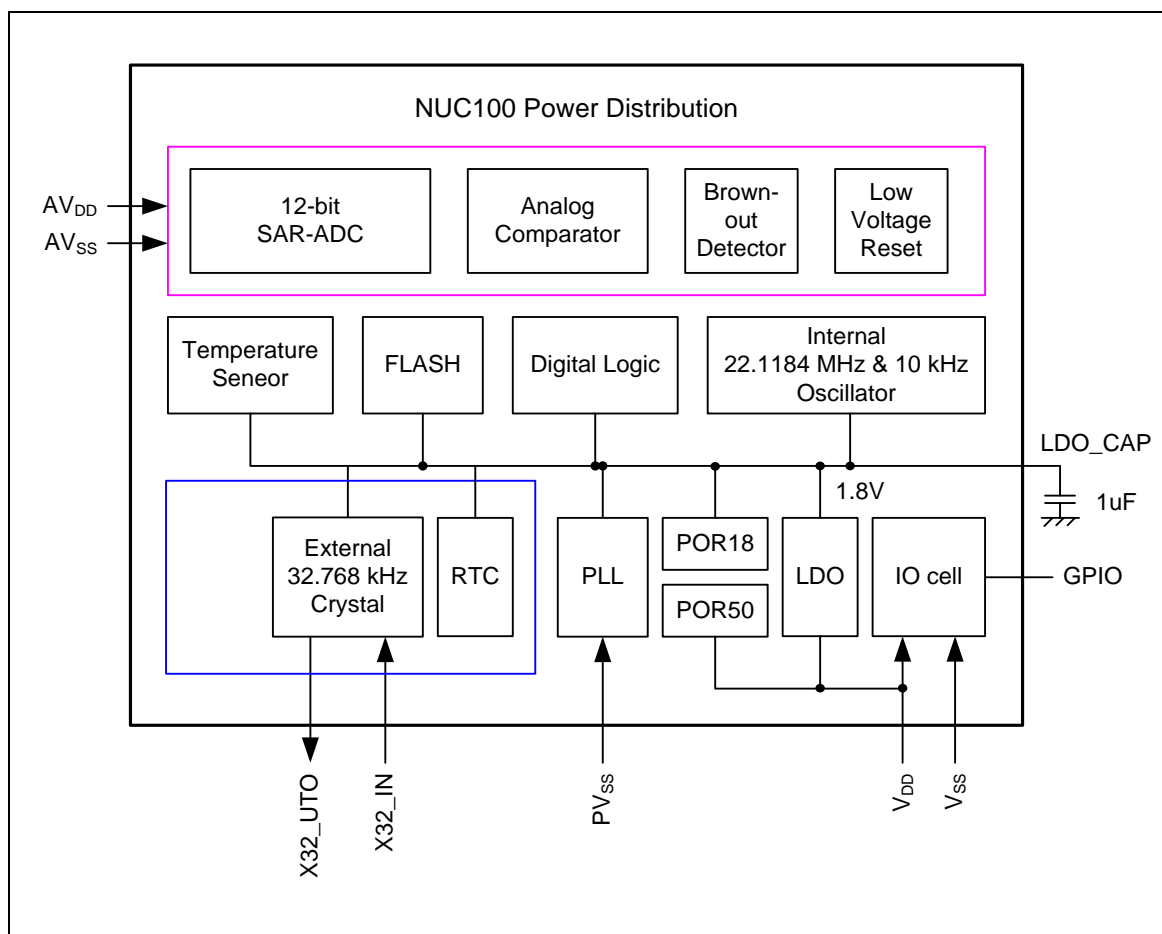


图 4-2 NuMicro™ NUC100 电源分配图

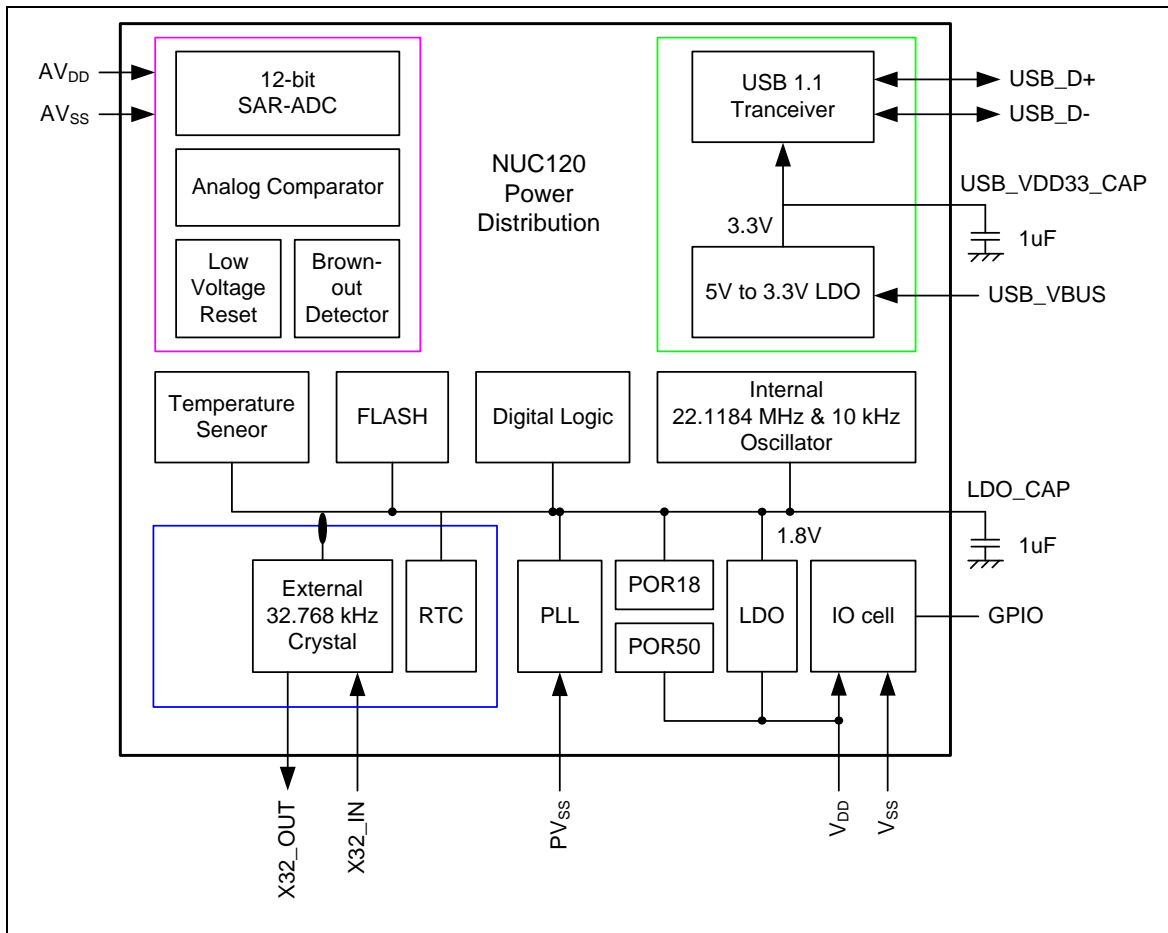


图 4-3 NuMicro™ NUC120 电源分配图

#### 4.2.4 系统内存映射

NuMicro™ NUC100 系列提供 4G字节的寻址空间。内存地址分配情况见下表。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。NuMicro™ NUC100 系列仅支持小端数据格式。

地址空间	标志	控制器
<b>Flash &amp; SRAM 内存空间</b>		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH内存空间(128 KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM内存空间(16 KB)
0x6000_0000 – 0x6001_FFFF	EXTMEM_BA	External Memory Space (128KB)
<b>AHB控制器空间(0x5000_0000 – 0x501F_FFFF)</b>		
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	PDMA_BA	外设 DMA 控制寄存器
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash 内存控制寄存器
<b>APB1控制器空间(0x4000_0000 ~ 0x400F_FFFF)</b>		
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 <sup>2</sup> 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
0x4003_4000 – 0x4003_7FFF	SPI1_BA	带主/从功能的SPI1 控制寄存器
0x4004_0000 – 0x4004_3FFF	PWMA_BA	PWM0/1/2/3 控制寄存器
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 控制寄存器
0x4006_0000 – 0x4006_3FFF	USBD_BA	USB 2.0 FS 设备控制寄存器
0x400D_0000 – 0x400D_3FFF	ACMP_BA	模拟比较控制寄存器
0x400E_0000 – 0x400E_FFFF	ADC_BA	ADC 控制寄存器
<b>APB2控制器空间(0x4010_0000 ~ 0x401F_FFFF)</b>		
0x4010_0000 – 0x4010_3FFF	PS2_BA	PS/2 接口控制寄存器
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2/Timer3 控制寄存器
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I <sup>2</sup> C1 接口控制寄存器
0x4013_0000 – 0x4013_3FFF	SPI2_BA	带主/从功能的SPI2 控制寄存器

0x4013_4000 – 0x4013_7FFF	SPI3_BA	带主/从功能的SPI3 控制寄存器
0x4014_0000 – 0x4014_3FFF	PWMB_BA	PWM4/5/6/7 控制寄存器
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 控制寄存器
0x4015_4000 – 0x4015_7FFF	UART2_BA	UART2 控制寄存器
0x401A_0000 – 0x401A_3FFF	I2S_BA	I <sup>2</sup> S接口控制寄存器
<b>系统控制器空间(0xE000_E000 ~ 0xE000_EFFF)</b>		
0xE000_E010 – 0xE000_E0FF	SYST_BA	System 定时器控制寄存器
0xE000_E100 – 0xE000_ECFF	NVIC_BA	外部中断控制器控制寄存器
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System 控制寄存器

表 4-1 片上控制器的地址空间分配

4.2.5 系统管理器控制寄存器

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
GCR 基地址: GCR_BA = 0x5000_0000				
PDID	GCR_BA+0x00	R	器件 ID 寄存器	0x2014_0018 <sup>[1]</sup>
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	0x0000_0000
IPRSTC2	GCR_BA+0x0C	R/W	外设复位控制寄存器2	0x0000_0000
IPRSTC3	GCR_BA+0x10	R/W	外设复位控制寄存器3	0x0000_0000
BODCR	GCR_BA+0x18	R/W	欠压检测控制寄存器	0x0000_008X
TEMPCR	GCR_BA+0x1C	R/W	温度传感器控制寄存器	0x0000_0000
PORCR	GCR_BA+0x24	R/W	上电复位控制寄存器	0x0000_00XX
GPA_MFP	GCR_BA+0x30	R/W	GPIOA 多功能和输入类型控制寄存器	0x0000_0000
GPB_MFP	GCR_BA+0x34	R/W	GPIOB 多功能和输入类型控制寄存器	0x0000_0000
GPC_MFP	GCR_BA+0x38	R/W	GPIOC 多功能和输入类型控制寄存器	0x0000_0000
GPD_MFP	GCR_BA+0x3C	R/W	GPIOD 多功能和输入类型控制寄存器	0x0000_0000
GPE_MFP	GCR_BA+0x40	R/W	GPIOE 多功能和输入类型控制寄存器	0x0000_0000
GPF_MFP	GCR_BA+0x44	R/W	GPIOF 多功能和输入类型控制寄存器	0x0000_000X
ALT_MFP	GCR_BA+0x50	R/W	复用多功能管脚控制寄存器	0x0000_0000
ALT_MFP1	GCR_BA+0x58	R/W	复用多功能管脚控制寄存器1	0x0000_0000
IRCTIMCTL	GCR_BA+0x80	R/W	IRC 校准控制寄存器	0x0000_0000
IRCTIMIEN	GCR_BA+0x84	R/W	IRC 校准中断使能寄存器	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	IRC 校准中断状态寄存器	0x0000_0000
REGWRPROT	GCR_BA+0x100	R/W	寄存器写保护寄存器	0x0000_0000

Note: [1] 依赖于产品型号.

4.2.6 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	0x2014_0018 <sup>[1]</sup>

[1] 每个器件具有一个独一无二的默认复位值。.

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

Bits	描述	
[31:0]	PDID	产品器件识别码 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。



**系统复位源寄存器(RSTSRC)**

该寄存器提供一些信息用于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
RSTSRC	GCR_BA+0x04	R/W	系统复位源寄存器	0x0000_00XX

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

Bits	描述	
[31:8]	Reserved	保留
[7]	RSTS_CPU	如果软件写 1 到 CPU_RST (IPRSTC1[1])，复位Cortex-M0 CPU的核和 Flash 控制器，硬件会把 RSTS_CPU 标志位置 ‘1’ 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]	Reserved	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 = Cortex-M0 因为软件向 SYSRESETREQ(AIRCR[2]) 写 1，发出复位信号而复位系统 (AIRCR[2]寄存器的地址是 0xE000ED0C)。 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]	RSTS_LVR	RSTS_LVR 标志位由低压复位模块的“复位信号”置位，用于表示导致之前复位的复位源。 1 = 低压 LVR 模块发出复位信号使系统复位。 0 = LVR 无复位 向该位写 1 清零。
[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。

		<p>1 = 看门狗模块发出复位信号使系统复位。</p> <p>0 = 看门狗无复位</p> <p>向该位写 1 清零。</p>
[1]	<b>RSTS_RESET</b>	<p>RSTS_RESET 标志位由 /RESET 管脚的“复位信号”置位，用于表示导致之前复位的复位源。</p> <p>1 = /RESET 管脚发出复位信号使系统复位</p> <p>0 = /RESET 管脚无复位</p> <p>向该位写 1 清零。</p>
[0]	<b>RSTS_POR</b>	<p>RSTS_POR 标志位由 上电复位 (POR) 的“复位信号”或 CHIP_RST (IPRSTC1[0]) 位置位，用于表示导致之前复位的复位源。</p> <p>1 = 上电复位 (POR) 或 CHIP_RST 发出复位信号使系统复位</p> <p>0 = 上电复位 (POR) 或 CHIP_RST 无复位</p> <p>向该位写 1 清零。</p>

外设复位控制寄存器1 (IPRSTC1)

寄存器	偏移量	R/W	描述	复位后的值
IPRSTC1	GCR_BA+0x08	R/W	外设复位控制寄存器1	0x0000_0000

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

Bits	描述	
[31:3]	Reserved	保留
[2]	PDMA_RST	<p><b>PDMA 控制器复位</b> (写保护位)</p> <p>该位置 1，产生复位信号到 PDMA，用户需要置 0 才能释放复位状态。</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”，该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100)</p> <p>1 = PDMA 控制器复位</p> <p>0 = PDMA 控制器正常工作</p>
[1]	CPU_RST	<p><b>CPU 内核复位</b> (写保护位)</p> <p>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</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”，该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100)</p> <p>1 = CPU 复位</p> <p>0 = CPU 正常工作</p>
[0]	CHIP_RST	<p><b>CHIP 复位</b> (写保护位)</p> <p>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。</p> <p>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。</p> <p>CHIP_RST 和 SYSRESETREQ 的区别，请参考章节 5.2.2</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”，该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100)</p> <p>1 = CHIP 复位</p> <p>0 = CHIP 正常工作</p>

**外设复位控制寄存器2 (IPRSTC2)**

置 1 会产生异步复位信号给相应的 IP 控制器。用户需要将该位置 0 才能将相应的 IP 控制器从复位状态恢复。

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

31	30	29	28	27	26	25	24
Reserved		I2S_RST	ADC_RST	USBD_RST	Reserved		
23	22	21	20	19	18	17	16
PS2_RST	ACMP_RST	PWM47_RST	PWM03_RST	Reserved	UART2_RST	UART1_RST	UART0_RST
15	14	13	12	11	10	9	8
SPI3_RST	SPI2_RST	SPI1_RST	SPIO_RST	Reserved	Reserved	I2C1_RST	I2C0_RST
7	6	5	4	3	2	1	0
Reserved		TMR3_RST	TMR2_RST	TMR1_RST	TMR0_RST	GPIO_RST	Reserved

Bits	描述
[31:30]	Reserved 保留
[29]	I2S_RST <b>I<sup>2</sup>S 控制器复位</b> 1 = I <sup>2</sup> S 控制器复位 0 = I <sup>2</sup> S 控制器正常工作
[28]	ADC_RST <b>ADC 控制器复位</b> 1 = ADC 控制器复位 0 = ADC 控制器正常工作
[27]	USBD_RST <b>USB 设备控制器复位</b> 1 = USB 设备控制器复位 0 = USB 设备控制器正常工作
[26:24]	Reserved 保留
[23]	PS2_RST <b>PS/2 控制器复位</b> 1 = PS/2 控制器复位 0 = PS/2 控制器正常工作
[22]	ACMP_RST <b>模拟比较控制器复位</b> 1 = 模拟比较控制器复位 0 = 模拟比较控制器正常工作
[21]	PWM47_RST <b>PWM47 控制器复位</b> 1 = PWM47 控制器复位 0 = PWM47 控制器正常工作
[20]	PWM03_RST <b>PWM03 控制器复位</b>

		1 = PWM03 控制器复位 0 = PWM03 控制器正常工作
[19]	Reserved	保留
[18]	UART2_RST	<b>UART2 控制器复位</b> 1 = UART2 控制器复位 0 = UART2 控制器正常工作
[17]	UART1_RST	<b>UART1 控制器复位</b> 1 = UART1 控制器复位 0 = UART1 控制器正常工作
[16]	UART0_RST	<b>UART0 控制器复位</b> 1 = UART0 控制器复位 0 = UART0 控制器正常工作
[15]	SPI3_RST	<b>SPI3 控制器复位</b> 1 = SPI3 控制器复位 0 = SPI3 控制器正常工作
[14]	SPI2_RST	<b>SPI2 控制器复位</b> 1 = SPI2 控制器复位 0 = SPI2 控制器正常工作
[13]	SPI1_RST	<b>SPI1 控制器复位</b> 1 = SPI1 控制器复位 0 = SPI1 控制器正常工作
[12]	SPI0_RST	<b>SPI0 控制器复位</b> 1 = SPI0 控制器复位 0 = SPI0 控制器正常工作
[11:10]	Reserved	保留
[9]	I2C1_RST	<b>I<sup>2</sup>C1 控制器复位</b> 1 = I <sup>2</sup> C1 控制器复位 0 = I <sup>2</sup> C1 控制器正常工作
[8]	I2C0_RST	<b>I<sup>2</sup>C0 控制器复位</b> 1 = I <sup>2</sup> C0 控制器复位 0 = I <sup>2</sup> C0 控制器正常工作
[7:6]	Reserved	保留
[5]	TMR3_RST	<b>Timer3 控制器复位</b> 1 = Timer3 控制器复位 0 = Timer3 控制器正常工作
[4]	TMR2_RST	<b>Timer2 控制器复位</b> 1 = Timer2 控制器复位 0 = Timer2 控制器正常工作
[3]	TMR1_RST	<b>Timer1 控制器复位</b> 1 = Timer1 控制器复位

		0 = Timer1 控制器正常工作
[2]	TMR0_RST	<b>Timer0 控制器复位</b> 1 = Timer0 控制器复位 0 = Timer0 控制器正常工作
[1]	GPIO_RST	<b>GPIO 控制器复位</b> 1 = GPIO 控制器复位 0 = GPIO 控制器正常工作
[0]	Reserved	保留

**外设复位控制寄存器 3 (IPRSTC3)**

置 1 会产生异步复位信号给相应的 IP 控制器。用户需要将该位置 0 才能将相应的 IP 控制器从复位状态恢复。

寄存器	偏移量	R/W	描述	复位后的值
IPRSTC3	GCR_BA+0x10	R/W	外设复位控制寄存器 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					SC2_RST	SC1_RST	SC0_RST

Bits	描述	
[31:3]	Reserved	保留
[2]	SC2_RST	<b>SC2 控制器复位</b> 1 = SC2 控制器复位 0 = SC2 控制器正常工作
[1]	SC1_RST	<b>SC1 控制器复位</b> 1 = SC1 控制器复位 0 = SC1 控制器正常工作
[0]	SC0_RST	<b>SC0 控制器复位</b> 1 = SC0 控制器复位 0 = SC0 控制器正常工作

**欠压检测控制寄存器 (BODCR)**

BODCR 控制寄存器的部分位在 flash 配置时 (config0 寄存器) 已经被初始化, 部分位是受保护的位。编程这些写保护的位时, 需要向地址 0x5000\_0100 依次写入“59h”, “16h”, “88h”。该位操作请参考寄存器 REGWRPROT (地址 GCR\_BA+0x100)

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

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

Bits	描述	
[31:8]	Reserved	保留
[7]	LVR_EN	<p><b>低压复位使能</b>（写保护位）</p> <p>当输入电源电压低于 LVR 电路设置时，LVR 复位芯片。默认使能低电压复位功能。</p> <p>1 = 使能低电压复位功能，使能该位100us后，低电压复位输出稳定，LVR功能生效（默认）</p> <p>0 = 禁用低电压复位功能</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入“59h”，“16h”，“88h”，该操作参考寄存器 REGWRPROT（地址 GCR_BA+0x100）</p>
[6]	BOD_OUT	<p><b>欠压检测输出状态位</b></p> <p>1 = 欠压检测输出状态为 1。表示检测到的电压低于 BOD_VL 的设置。若 BOD_EN 是 '0'，禁用 BOD 功能，该位通常响应必定为 '0'。</p> <p>0 = 欠压检测输出状态为 0。表示检测到的电压高于 BOD_VL 的设置或者 BOD_EN 为 0。</p>
[5]	BOD_LPM	<p><b>低压模式下的欠压检测</b>（写保护位）</p> <p>1 = 使能 BOD 的低功耗模式</p> <p>0 = BOD 工作在正常模式（默认）</p> <p>BOD 在正常模式下消耗电流约为100 uA，低功耗模式下能减小到当前的约1/10，但BOD响应速度变慢。</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入“59h”，“16h”，“88h”，该操作参考寄存器 REGWRPROT（地址 GCR_BA+0x100）</p>
[4]	BOD_INTF	<p><b>欠压检测中断标志</b></p> <p>1 = 当欠压检测到 VDD 下降到 BOD_VL 的设定电压或者 VDD 上升到 BOD_VL 的设定电压，该位设为1，如果欠压电压中断被使能，则发生欠压中断。</p> <p>0 = 欠压检测没有检测到任何电压由 VDD 下降或者上升到 BOD_VL 设定值。</p> <p>向该位写 1 清零。</p>
[3]	BOD_RSTEN	<p><b>欠压复位使能</b>（写保护位）</p> <p>1 = 使能欠压复位功能</p> <p>当同时使能欠压检测功能（BOD_EN 为高）和 BOD 复位功能（BOD_RSTEN 为高）时，</p>



		<p>如果检测到电压低于阙电压 (BOD_OUT 为高), BOD 将发送信号复位芯片。</p> <p>0 = 使能欠压中断功能</p> <p>当同时使能 BOD 功能 (BOD_EN 为高) 和 BOD 中断功能 (BOD_RSTEN 为低), 如果 BOD_OUT 为高, 则 BOD 将产生中断。BOD 中断将保持直到 BOD_EN 被设置为 0。可以通过禁用 NVIC BOD 中断或者禁用 BOD 功能 (设置 BOD_EN 为低) 封锁 BOD 中断。</p> <p>默认值由用户在配置 flash 控制寄存器 (config0 bit[20]) 时设置。</p> <p>该位受保护, 编程该位时, 需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”, 该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100)</p>															
[2:1]	BOD_VL	<p>欠压检测阙电压选择 (写保护位)</p> <p>默认值由用户在配置 flash 控制寄存器 (config0 bit[22:21]) 时设置。</p> <p>该位受保护, 编程该位时, 需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”, 该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100) .</p> <table border="1"> <thead> <tr> <th>BOD_V [1]</th> <th>BOD_VL[ ]</th> <th>Brown-out Voltage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>4.4 V</td> </tr> <tr> <td>1</td> <td>0</td> <td>3.7 V</td> </tr> <tr> <td>0</td> <td>1</td> <td>2.7 V</td> </tr> <tr> <td>0</td> <td>0</td> <td>2.2 V</td> </tr> </tbody> </table>	BOD_V [1]	BOD_VL[ ]	Brown-out Voltage	1	1	4.4 V	1	0	3.7 V	0	1	2.7 V	0	0	2.2 V
BOD_V [1]	BOD_VL[ ]	Brown-out Voltage															
1	1	4.4 V															
1	0	3.7 V															
0	1	2.7 V															
0	0	2.2 V															
[0]	BOD_EN	<p>欠压检测使能 (写保护位)</p> <p>默认值由用户在配置 flash 控制寄存器 (config0 bit[23]) 时设置。</p> <p>1 = 使能欠压检测功能</p> <p>0 = 禁用欠压检测功能</p> <p>该位受保护, 编程该位时, 需要依次向地址 0x5000_0100 写入 “59h”, “16h”, “88h”, 该操作参考寄存器 REGWRPROT (地址 GCR_BA+0x100)</p>															

**温度传感器控制寄存器(TEMPCR)**

寄存器	偏移量	R/W	描述	复位后的值
TEMPCR	GCR_BA+0x1C	R/W	温度传感器控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
----	----	----	----	----	----	----	----

Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							VTEMP_EN

Bits	描述	
[31:1]	Reserved	保留
[0]	VTEMP_EN	<p><b>温度传感器使能</b></p> <p>该位用于使能/禁用温度传感器功能。</p> <p>1 = 使能温度传感器功能</p> <p>0 = 禁用温度传感器功能</p> <p>该位置 1 后，温度值可以从 ADC 转换结果获得，这之前需要把 ADC channel 选择设为 channel 7，多功能复用通道选择到温度传感器选项。具体的 ADC 转换功能请参考 ADC 功能章节。</p>

上电复位控制寄存器(PORCR)

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

31	30	29	28	27	26	25	24
Reserved							

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

Bits	描述	
[31:16]	Reserved	保留
[15:0]	POR_DIS_CODE	<p><b>该寄存器用于上电复位使能控制（写保护位）</b></p> <p>上电时，POR 电路产生复位信号使整个芯片复位，但是电源部分的干扰可能引起 POR 重新有效。用户可以将 POR_DIS_CODE 设置为 0x5AA5，禁用 POR 内部电路，以免造成不可预知的干扰。</p> <p>当设置 POR_DIS_CODE 为其他值时，或者由芯片的其他复位功能引起复位时，POR 功能重新有效。这些复位功能包括： /RESET管脚复位，看门狗，LVR 复位，BOD 复位，ICE 复位命令和软件复位。</p> <p>该位受保护，编程该位时，需要依次向地址 0x5000_0100 写入“59h”，“16h”，“88h”，该操作参考寄存器 REGWRPROT（地址 GCR_BA+0x100）</p>

**GPIOA 多功能管脚控制寄存器(GPA\_MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPA_MFP	GCR_BA+0x30	R/W	GPIOA多功能与输入类型控制寄存器	0x0000_0000

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

Bits	描述																					
[31:16]	GPA_TYPEn	1 = 使能 GPIOA[15:0] I/O Schmitt 触发输入 0 = 禁用 GPIOA[15:0] I/O Schmitt 触发输入																				
[15]	GPA_MFP15	<p><b>PA.15管脚功能选择</b> 该管脚功能取决于PA15_SC2PWR (ALT_MFP1[12]), PA15_I2SMCLK (ALT_MFP[9]) 和 GPA_MFP[15]。</p> <table border="1"> <thead> <tr> <th>PA15_SC2PWR</th> <th>PA15_I2SMCLK</th> <th>GPA_MFP[15]</th> <th>PA.15 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>PWM3 (PWM)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>I2S_MCLK</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>SC2_PWR</td> </tr> </tbody> </table>	PA15_SC2PWR	PA15_I2SMCLK	GPA_MFP[15]	PA.15 功能	0	0	0	GPIO	0	0	1	PWM3 (PWM)	0	1	1	I2S_MCLK	1	0	1	SC2_PWR
PA15_SC2PWR	PA15_I2SMCLK	GPA_MFP[15]	PA.15 功能																			
0	0	0	GPIO																			
0	0	1	PWM3 (PWM)																			
0	1	1	I2S_MCLK																			
1	0	1	SC2_PWR																			
[14]	GPA_MFP14	<p><b>PA.14管脚功能选择</b> 该管脚功能取决于PA14_SC2RST (ALT_MFP1[13]) 和 GPA_MFP[14]。</p> <table border="1"> <thead> <tr> <th>PA14_SC2RST</th> <th>GPA_MFP[14]</th> <th>PA.14 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>PWM2 (PW )</td> </tr> <tr> <td>1</td> <td>1</td> <td>SC2_RST</td> </tr> </tbody> </table>	PA14_SC2RST	GPA_MFP[14]	PA.14 功能	0	0	GPIO	0	1	PWM2 (PW )	1	1	SC2_RST								
PA14_SC2RST	GPA_MFP[14]	PA.14 功能																				
0	0	GPIO																				
0	1	PWM2 (PW )																				
1	1	SC2_RST																				
[13]	GPA_MFP13	<p><b>PA.13管脚功能选择</b> 该管脚功能取决于PA13_SC2CLK (ALT_MFP1[10]) 和 GPA_MFP[13]。</p> <table border="1"> <thead> <tr> <th>PA13_SC2CLK</th> <th>GPA_MFP[13]</th> <th>PA.13 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>PWM1 (PWM)</td> </tr> <tr> <td>1</td> <td>1</td> <td>SC2_CLK</td> </tr> </tbody> </table>	PA13_SC2CLK	GPA_MFP[13]	PA.13 功能	0	0	GPIO	0	1	PWM1 (PWM)	1	1	SC2_CLK								
PA13_SC2CLK	GPA_MFP[13]	PA.13 功能																				
0	0	GPIO																				
0	1	PWM1 (PWM)																				
1	1	SC2_CLK																				

[12]	GPA_MFP12	<b>PA.12管脚功能选择</b> 该管脚功能取决于PA12_SC2DAT (ALT_MFP1[11]) 和 GPA_MFP[12]			
		<b>PA12_SC2DAT</b>	<b>GPA_MFP[12]</b>	<b>PA.12 功能</b>	
		0	0	GP O	
		0	1	PWM0 (P )	
		1	1	SC2_DAT	
[11]	GPA_MFP11	<b>PA.11管脚功能选择</b> 该管脚功能取决于GPA_MFP[11]			
		<b>GPA_MFP[11]</b>	<b>PA.11 功能</b>		
		0	GPIO		
		1	I2C1_SCL		
[10]	GPA_MFP10	<b>PA.10管脚功能选择</b> 该管脚功能取决于GPA_MFP[10]			
		<b>G A_MFP 10]</b>	<b>PA.10 功能</b>		
		0	GPI		
		1	I2C1_ DA		
[9]	GPA_MFP9	<b>PA.9管脚功能选择</b> 该管脚功能取决于GPA_MFP[9]			
		<b>GPA_MFP[9]</b>	<b>PA.9 功能</b>		
		0	GPIO		
		1	I2C0_SCL		
[8]	GPA_MFP8	<b>PA.8管脚功能选择</b> 该管脚功能取决于 GPA_MFP[8]			
		<b>GPA_MFP[8]</b>	<b>PA.8 功能</b>		
		0	GPIO		
		1	I2C0_SDA		
[7]	GPA_MFP7	<b>PA.7管脚功能选择</b> 该管脚功能取决于PA7_SC1DAT (ALT_MFP1[6]), PA7_S21 (ALT_MFP[2]) 和 GPA_MFP[7]			
		<b>PA7_SC1DAT</b>	<b>PA _S21</b>	<b>GPA_MFP[7]</b>	<b>PA.7 功能</b>
			0		GPIO
		0	0	1	ADC7
		0	1	1	SPISS21 (SPI2)
		1	0	1	SC1_DAT

[6]	GPA_MFP6	<b>PA.6管脚功能选择</b> 该管脚功能取决于PA6_SC1CLK (ALT_MFP1[5]) 和 GPA_MFP[6]		
		<b>PA6_SC1CLK</b>	<b>GPA_MFP[6]</b>	<b>PA.6 功能</b>
		0	0	GPIO
		0	1	ADC6
		1	1	SC1_CLK
[5]	GPA_MFP5	<b>PA.5管脚功能选择</b> 该管脚功能取决于PA5_SC1RST (ALT_MFP1[8]) 和 GPA_MFP[5].		
		<b>PA5_SC1RST</b>	<b>GPA_MFP[5]</b>	<b>PA.5 功能</b>
		0	0	GPIO
		0	1	ADC5
		1	1	SC1_RST
[4]	GPA_MFP4	<b>PA.4管脚功能选择</b> 该管脚功能取决于PA4_SC1PWR (ALT_MFP1[7]) 和 GPA_MFP[4].		
		<b>PA4_SC1PWR</b>	<b>A_MFP[4]</b>	<b>PA.4 功能</b>
		0	0	GPI
		0	1	ADC4
		1	1	SC1_PWR
[3]	GPA_MFP3	<b>PA.3管脚功能选择</b> 该管脚功能取决于PA3_SC0DAT (ALT_MFP1[1]) 和 GPA_MFP[3]		
		<b>PA3_SC0DAT</b>	<b>GPA_MFP[3]</b>	<b>PA.3 功能</b>
		0	0	GPIO
		0	1	ADC3
		1	1	SC0_DAT
[2]	GPA_MFP2	<b>PA.2管脚功能选择</b> 该管脚功能取决于PA2_SC0CLK (ALT_MFP1[0]) 和 GPA_MFP[2]		
		<b>PA2_SC0CLK</b>	<b>GPA_MFP[2]</b>	<b>PA.2 功能</b>
		0	0	GPIO
		0		ADC2
		1	1	SC0_CLK

[1]	GPA_MFP1	<b>PA.1管脚功能选择</b> 该管脚功能取决于PA1_SC0RST (ALT_MFP1[3]) 和 GPA_MFP[1]		
		<b>PA1_SC0RST</b>	<b>GPA_MFP[1]</b>	<b>PA.1 功能</b>
		0	0	GPIO
		0	1	ADC1
		1	1	SC0_RST
[0]	GPA_MFP0	<b>PA.0管脚功能选择</b> 该管脚功能取决于PA0_SC0PWR (ALT_MFP1[2]) 和 GPA_MFP[0]		
		<b>PA_SC0PWR</b>	<b>GPA_MFP[0]</b>	<b>PA.0 功能</b>
		0	0	GPIO
		0	1	ADC0
		1	1	SC0_PWR

**GPIOB 多功能管脚控制寄存器 (GPB\_MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPB_MFP	GCR_BA+0x34	R/W	GPIOB 多功能与输入类型控制寄存器	0x0000_0000

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

Bits	描述													
[31:16]	GPB_TYPEn	1 = 使能 GPIOB[15:0] I/O Schmitt 触发输入 0 = 禁用 GPIOB[15:0] I/O Schmitt 触发输入												
[15]	GPB_MFP15	<p><b>PB.15 管脚功能选择</b> 该管脚功能取决于 PB15_T0EX (ALT_MFP[24]) 和 GPB_MFP[15].</p> <table border="1"> <thead> <tr> <th>PB15_T0EX</th> <th>GPB_MFP[15]</th> <th>PB.15 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>INT1</td> </tr> <tr> <td>1</td> <td>1</td> <td>TM0_EXT</td> </tr> </tbody> </table>	PB15_T0EX	GPB_MFP[15]	PB.15 功能	0		GPIO	0	1	INT1	1	1	TM0_EXT
PB15_T0EX	GPB_MFP[15]	PB.15 功能												
0		GPIO												
0	1	INT1												
1	1	TM0_EXT												
[14]	GPB_MFP14	<p><b>PB.14管脚功能选择</b> 该管脚功能取决于PB14_S31 (ALT_MFP[3]) 和 GPB_MFP[14]</p> <table border="1"> <thead> <tr> <th>PB14_S31</th> <th>GPB_MFP[14]</th> <th>PB.14 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>INT0</td> </tr> <tr> <td>1</td> <td>1</td> <td>SPISS31 (SPI3)</td> </tr> </tbody> </table>	PB14_S31	GPB_MFP[14]	PB.14 功能	0	0	GPIO	0	1	INT0	1	1	SPISS31 (SPI3)
PB14_S31	GPB_MFP[14]	PB.14 功能												
0	0	GPIO												
0	1	INT0												
1	1	SPISS31 (SPI3)												
[13]	GPB_MFP13	<p><b>PB.13管脚功能选择</b> 该管脚功能取决于GPB_MFP[13].</p> <table border="1"> <thead> <tr> <th>GPB_MFP 13]</th> <th>PB.13 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>GP O</td> </tr> <tr> <td></td> <td>CMP1_O</td> </tr> </tbody> </table>	GPB_MFP 13]	PB.13 功能	0	GP O		CMP1_O						
GPB_MFP 13]	PB.13 功能													
0	GP O													
	CMP1_O													
[12]	GPB_MFP12	保留												



[11]	GPB_MFP11	<p><b>PB.11管脚功能选择</b> 该管脚功能取决于PB11_PWM4 (ALT_MFP[4]) 和 GPB_MFP[11]</p> <table border="1"> <thead> <tr> <th>PB11_PWM4</th> <th>GPB_MFP[11]</th> <th>PB.11 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>TM3</td> </tr> <tr> <td>1</td> <td></td> <td>PWM4 (PWM)</td> </tr> </tbody> </table>	PB11_PWM4	GPB_MFP[11]	PB.11 功能	0	0	GPIO	0	1	TM3	1		PWM4 (PWM)
PB11_PWM4	GPB_MFP[11]	PB.11 功能												
0	0	GPIO												
0	1	TM3												
1		PWM4 (PWM)												
[10]	GPB_MFP10	<p><b>PB.10管脚功能选择</b> 该管脚功能取决于PB10_S01 (ALT_MFP[0]) 和 GPB_MFP[10]</p> <table border="1"> <thead> <tr> <th>PB10_S 1</th> <th>GPB_MFP[10]</th> <th>PB.10 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>TM2</td> </tr> <tr> <td>1</td> <td></td> <td>SPISS 1 (SPI0)</td> </tr> </tbody> </table>	PB10_S 1	GPB_MFP[10]	PB.10 功能	0	0	GPIO	0	1	TM2	1		SPISS 1 (SPI0)
PB10_S 1	GPB_MFP[10]	PB.10 功能												
0	0	GPIO												
0	1	TM2												
1		SPISS 1 (SPI0)												
[9]	GPB_MFP9	<p><b>PB.9管脚功能选择</b> 该管脚功能取决于PB9_S11 (ALT_MFP[1]) 和 GPB_MFP[9].</p> <table border="1"> <thead> <tr> <th>PB9_S11</th> <th>GPB_MFP[9]</th> <th>PB.9 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>T 1</td> </tr> <tr> <td>1</td> <td>1</td> <td>SPISS11 (SPI1)</td> </tr> </tbody> </table>	PB9_S11	GPB_MFP[9]	PB.9 功能	0		GPIO	0	1	T 1	1	1	SPISS11 (SPI1)
PB9_S11	GPB_MFP[9]	PB.9 功能												
0		GPIO												
0	1	T 1												
1	1	SPISS11 (SPI1)												
[8]	GPB_MFP8	<p><b>PB.8管脚功能选择</b> 该管脚功能取决于PB8_CLKO (ALT_MFP[29]) 和 GPB_MFP[8]</p> <table border="1"> <thead> <tr> <th>PB8_CLKO</th> <th>GPB_MFP[8]</th> <th>PB.8 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>TM0</td> </tr> <tr> <td>1</td> <td>1</td> <td>CLKO (Clock Driver output)</td> </tr> </tbody> </table>	PB8_CLKO	GPB_MFP[8]	PB.8 功能	0		GPIO	0	1	TM0	1	1	CLKO (Clock Driver output)
PB8_CLKO	GPB_MFP[8]	PB.8 功能												
0		GPIO												
0	1	TM0												
1	1	CLKO (Clock Driver output)												
[7]	GPB_MFP7	<p><b>PB.7管脚功能选择</b> 1 = PB.7作为UART1_nCST 0 = PB.7作为GPIOB[7]</p>												
[6]	GPB_MFP6	<p><b>PB.6管脚功能选择</b> 1 = PB.6作为UART1_nRST 0 = PB.6作为GPIOB[6]</p>												
[5]	GPB_MFP5	<p><b>PB. 5管脚功能选择</b> 1 = PB.5作为UART1_TXD 0 = PB.5作为GPIOB[5]</p>												
[4]	GPB_MFP4	<p><b>PB.4管脚功能选择</b> 1 = PB.4作为UART1_RXD 0 = PB.4作为GPIOB[4]</p>												

[3]	GPB_MFP3	<p><b>PB.3管脚功能选择</b></p> <p>该管脚功能取决于 PB3_SC2CD (ALT_MFP1[14]), PB3_T3EX (ALT_MFP[27]) 和 GPB_MFP[3]</p> <table border="1"> <thead> <tr> <th>PB3_SC2CD</th> <th>PB3_T3EX</th> <th>GPB_MFP[3]</th> <th>PB.3 功能</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>UART0_nCTS</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>TM3_EXT</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>SC2_CD</td> </tr> </tbody> </table>	PB3_SC2CD	PB3_T3EX	GPB_MFP[3]	PB.3 功能		0	0	GPIO	0	0	1	UART0_nCTS	0	1	1	TM3_EXT	1	0	1	SC2_CD
PB3_SC2CD	PB3_T3EX	GPB_MFP[3]	PB.3 功能																			
	0	0	GPIO																			
0	0	1	UART0_nCTS																			
0	1	1	TM3_EXT																			
1	0	1	SC2_CD																			
[2]	GPB_MFP2	<p><b>PB.2管脚功能选择</b></p> <p>该管脚功能取决于 PB2_CPO0 (ALT_MFP[30]), PB2_T2EX (ALT_MFP[26]) 和 GPB_MFP[2]</p> <table border="1"> <thead> <tr> <th>PB2_CPO0</th> <th>PB2_T2EX</th> <th>GPB_MFP[2]</th> <th>PB.2 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>UART0_nRTS</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>TM2_EXT</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>CMPO_O</td> </tr> </tbody> </table>	PB2_CPO0	PB2_T2EX	GPB_MFP[2]	PB.2 功能	0	0	0	GPIO	0	0	1	UART0_nRTS	0	1	1	TM2_EXT	1	0	1	CMPO_O
PB2_CPO0	PB2_T2EX	GPB_MFP[2]	PB.2 功能																			
0	0	0	GPIO																			
0	0	1	UART0_nRTS																			
0	1	1	TM2_EXT																			
1	0	1	CMPO_O																			
[1]	GPB_MFP1	<p><b>PB.1管脚功能选择</b></p> <p>1 = PB.1作为UART0_TXD 0 = PB.1作为GPIOB[1]</p>																				
[0]	GPB_MFP0	<p><b>PB.0管脚功能选择</b></p> <p>1 = PB.0作为UART0_RXD 0 = PB.0作为GPIOB[0]</p>																				

**GPIOC多功能管脚控制寄存器 (GPC MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPC_MFP	GCR_BA+0x38	R/W	GPIOC多功能与输入类型控制寄存器	0x0000_0000

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

Bits	描述							
[31:16]	GPC_TYPEn	1 = 使能 GPIOC[15:0] I/O Schmitt 触发输入 0 = 禁用 GPIOC[15:0] I/O Schmitt 触发输入						
[15]	GPC_MFP15	<b>PC.15 管脚功能选择</b> 该管脚功能取决于GPC_MFP[15]						
		<table border="1"> <thead> <tr> <th>GPC_MFP[15]</th> <th>PC.15 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>GPIO</td> </tr> <tr> <td>1</td> <td>CMP1_N</td> </tr> </tbody> </table>	GPC_MFP[15]	PC.15 功能	0	GPIO	1	CMP1_N
		GPC_MFP[15]	PC.15 功能					
0	GPIO							
1	CMP1_N							
<table border="1"> <thead> <tr> <th>GPC_MFP[15]</th> <th>PC.15 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>GPIO</td> </tr> <tr> <td>1</td> <td>CMP1_N</td> </tr> </tbody> </table>	GPC_MFP[15]	PC.15 功能	0	GPIO	1	CMP1_N		
GPC_MFP[15]	PC.15 功能							
0	GPIO							
1	CMP1_N							
[14]	GPC_MFP14	<b>PC.14管脚功能选择</b> 该管脚功能取决于GPC_MFP[14]						
		<table border="1"> <thead> <tr> <th>GPC_MFP[14]</th> <th>PC.14 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>GPIO</td> </tr> <tr> <td>1</td> <td>CMP1_P</td> </tr> </tbody> </table>	GPC_MFP[14]	PC.14 功能	0	GPIO	1	CMP1_P
		GPC_MFP[14]	PC.14 功能					
0	GPIO							
1	CMP1_P							
<table border="1"> <thead> <tr> <th>GPC_MFP[14]</th> <th>PC.14 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>GPIO</td> </tr> <tr> <td>1</td> <td>CMP1_P</td> </tr> </tbody> </table>	GPC_MFP[14]	PC.14 功能	0	GPIO	1	CMP1_P		
GPC_MFP[14]	PC.14 功能							
0	GPIO							
1	CMP1_P							
[13]	GPC_MFP13	<b>PC.13管脚功能选择</b> 1 = PC.13作为SPI1_MOSI1 (主机输出, 从机输入脚-1) 0 = PC.13作为GPIOC[13]						
[12]	GPC_MFP12	<b>PC.12管脚功能选择</b> 1 = PC.12作为SPI1_MISO1 (主机输入, 从机输出脚-1) 0 = PC.12作为GPIOC[12]						
[11]	GPC_MFP11	<b>PC.11管脚功能选择</b> 1 = PC.11作为SPI1_MOSI0 (主机输出, 从机输入脚-0) 0 = PC.11作为GPIOC[11].						

[10]	GPC_MFP10	<p><b>PC.10管脚功能选择</b></p> <p>1 = PC.10作为SPI1_MISO0 (主机输入, 从机输出脚-0).</p> <p>0 = PC.10作为GPIOC[10].</p>												
[9]	GPC_MFP9	<p><b>PC.9管脚功能选择</b></p> <p>1 = PC.9作为SPI1_CLK.</p> <p>0 = PC.9作为GPIOC[9].</p>												
[8]	GPC_MFP8	<p><b>PC.8管脚功能选择</b></p> <p>1 = PC.8作为SPI1_SS0.</p> <p>0 = PC.8作为GPIOC[8].</p>												
[7]	GPC_MFP7	<p><b>PC.7管脚功能选择</b></p> <p>该管脚功能取决于 PC7_SC1CD (ALT_MFP1[9]) 和 GPC_MFP[7]</p> <table border="1"> <thead> <tr> <th>PC7_SC1CD</th> <th>GPC_MFP[7]</th> <th>PC.7 功能</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>GPI</td> </tr> <tr> <td>0</td> <td>1</td> <td>CMP0_N</td> </tr> <tr> <td>1</td> <td>1</td> <td>SC1_CD</td> </tr> </tbody> </table>	PC7_SC1CD	GPC_MFP[7]	PC.7 功能		0	GPI	0	1	CMP0_N	1	1	SC1_CD
PC7_SC1CD	GPC_MFP[7]	PC.7 功能												
	0	GPI												
0	1	CMP0_N												
1	1	SC1_CD												
[6]	GPC_MFP6	<p><b>PC.6管脚功能选择</b></p> <p>该管脚功能取决于 PC6_SC0CD (ALT_MFP1[4]) 和 GPC_MFP[6]</p> <table border="1"> <thead> <tr> <th>PC6_SC0CD</th> <th>GPC_FP[6]</th> <th>PC.6 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>CMP0_P</td> </tr> <tr> <td>1</td> <td>1</td> <td>SC0_CD</td> </tr> </tbody> </table>	PC6_SC0CD	GPC_FP[6]	PC.6 功能	0	0	GPIO	0	1	CMP0_P	1	1	SC0_CD
PC6_SC0CD	GPC_FP[6]	PC.6 功能												
0	0	GPIO												
0	1	CMP0_P												
1	1	SC0_CD												
[5]	GPC_MFP5	<p><b>PC.5管脚功能选择</b></p> <p>1 = PC.5作为SPI0_MOSI1 (主机输出, 从机输入脚-1)</p> <p>0 = PC.5作为GPIOC[5]</p>												
[4]	GPC_MFP4	<p><b>PC.4管脚功能选择</b></p> <p>1 = PC.4作为SPI0_MISO1 (主机输入, 从机输出脚-1)</p> <p>0 = PC.4作为GPIOC[4]</p>												
[3]	GPC_MFP3	<p><b>PC.3管脚功能选择</b></p> <p>该管脚功能取决于 PC3_I2SDO (ALT_MFP[8]) 和 GPC_MFP[3]</p> <table border="1"> <thead> <tr> <th>PC3_I2SDO</th> <th>GPC_MFP[3]</th> <th>PC.3 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>SPI0_MOSI0</td> </tr> <tr> <td>1</td> <td>1</td> <td>I2S_DO</td> </tr> </tbody> </table>	PC3_I2SDO	GPC_MFP[3]	PC.3 功能	0	0	GPIO	0	1	SPI0_MOSI0	1	1	I2S_DO
PC3_I2SDO	GPC_MFP[3]	PC.3 功能												
0	0	GPIO												
0	1	SPI0_MOSI0												
1	1	I2S_DO												

[2]	GPC_MFP2	<b>PC.2管脚功能选择</b> 该管脚功能取决于 PC2_I2SDI (ALT_MFP[7]) 和 GPC_MFP[2]		
		<b>PC2_I2SDI</b>	<b>GPC_MFP[2]</b>	<b>PC.2 功能</b>
		0	0	GPI
		0	1	SPI0_MISO0
		1	1	I2S_DI
[1]	GPC_MFP1	<b>PC.1管脚功能选择</b> 该管脚功能取决于 PC1_I2SBCLK (ALT_MFP[6]) 和 GPC_MFP[1]		
		<b>PC1_I2SBCLK</b>	<b>GPC_MFP[1]</b>	<b>PC.1 功能</b>
		0	0	GPIO
		0	1	SPI0_C K
		1	1	I2S_BCLK
[0]	GPC_MFP0	<b>PC.0管脚功能选择</b> 该管脚功能取决于 PC0_I2SLRCLK (ALT_MFP[5]) 和 GPC_MFP[0]		
		<b>PC0_I2_LRCLK</b>	<b>GPC_MFP[0]</b>	<b>PC.0 功能</b>
		0		GPIO
		0	1	SPI0_SS0
		1	1	I2S_LRCK

**GPIOD 多功能管脚控制寄存器 (GPD\_MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPD_MFP	GCR_BA+0x3C	R/W	GPIOD多功能与输入类型控制寄存器	0x0000_0000

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

Bits	描述	
[31:16]	GPD_TYPEn	1 = 使能 GPIOD[15:0] I/O Schmitt 触发输入 0 = 禁用 GPIOD[15:0] I/O Schmitt 触发输入
[15]	GPD_MFP15	<b>PD.15 管脚功能选择</b> 1 = PD.15作为UART2_TXD 0 = PD.15作为GPIOD[15]
[14]	GPD_MFP14	<b>PD.14管脚功能选择</b> 1 = PD.14作为UART2_RXD 0 = PD.14 GPIOD[14]
[13]	GPD_MFP13	<b>PD.13 管脚功能选择</b> 1 = PD.13 作为 SPI3 MOSI1 (主机输出, 从机输入管脚-1) 0 = PD.13 作为 GPIOD[13]
[12]	GPD_MFP12	<b>PD.12 管脚功能选择</b> 1 = PD.12 作为 SPI3 MISO1 (主机输入, 从机输出管脚-1) 0 = PD.12 作为 GPIOD[12]
[11]	GPD_MFP11	<b>PD.11 管脚功能选择</b> 1 = PD.11 作为 SPI3 MOSI0 (主机输出, 从机输入管脚-0) 0 = PD.11 作为 GPIOD[11]
[10]	GPD_MFP10	<b>PD.10 管脚功能选择</b> 1 = PD.10 作为 SPI3 MISO0 (主机输入, 从机输出管脚-0) 0 = PD.10 作为 GPIOD[10]

[9]	GPD_MFP9	<b>PD.9 管脚功能选择</b> 1 = PD.9 作为 SPI3 SPICLK 0 = PD.9 作为 GPIOD[9]
[8]	GPD_MFP8	<b>PD.8 管脚功能选择</b> 1 = PD.8 作为 SPI3 SS30 0 = PD.8 作为 GPIOD[8]
[7]	GPD_MFP7	<b>PD.7管脚功能选择</b> 保留
[6]	GPD_MFP6	<b>PD.6管脚功能选择</b> 保留
[5]	GPD_MFP5	<b>PD.5 管脚功能选择</b> 1 = PD.5 作为 SPI2 MOSI1（主机输出，从机输入管脚-1） 0 = PD.5 作为 GPIOD[5]
[4]	GPD_MFP4	<b>PD.4 管脚功能选择</b> 1 = PD.4 作为 SPI2 MISO1（主机输入，从机输出管脚-1） 0 = PD.4 作为 GPIOD[4]
[3]	GPD_MFP3	<b>PD.3 管脚功能选择</b> 1 = PD.3 作为 SPI2 MOSI0（主机输出，从机输入管脚-0） 0 = PD.3 作为 GPIOD[3]
[2]	GPD_MFP2	<b>PD.2 管脚功能选择</b> 1 = PD.2 作为 SPI2 MISO0（主机输入，从机输入管脚-0） 0 = PD.2 作为 GPIOD[2]
[1]	GPD_MFP1	<b>PD.1 管脚功能选择</b> 1 = PD.1 作为 SPI2 SPICLK 0 = PD.1 作为 GPIOD[1]
[0]	GPD_MFP0	<b>PD.0 管脚功能选择</b> 1 = PD.0 作为 SPI2 SS20 0 = PD.0 作为 GPIOD[0]

**GPIOE 多功能管脚控制寄存器 (GPE\_MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPE_MFP	GCR_BA+0x40	R/W	GPIOE多功能与输入类型控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
GPE_TYPE[15:8]							
23	22	21	20	19	18	17	16
GPE_TYPE[7:0]							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		GPE_MFP5	Reserved			GPE_MFP1	GPE_MFP0

Bits	描述													
[31:16]	GPE_TYPE <sub>n</sub>	1 = 使能 GPIOE[15:0] I/O Schmitt 触发输入 0 = 禁用 GPIOE[15:0] I/O Schmitt 触发输入												
[15:6]	Reserved	保留												
[5]	GPE_MFP5	<p><b>PE.5 管脚功能选择</b> 该管脚功能取决于 PE5_T1EX (ALT_MFP[25]) 和 GPE_MFP5</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>PE5_T1EX</th> <th>GPE_MFP[5]</th> <th>PE.5 功能</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>GPIO</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>PWM5</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>TM1_EXT</td> </tr> </tbody> </table>	PE5_T1EX	GPE_MFP[5]	PE.5 功能	0	0	GPIO	0	1	PWM5	1	1	TM1_EXT
PE5_T1EX	GPE_MFP[5]	PE.5 功能												
0	0	GPIO												
0	1	PWM5												
1	1	TM1_EXT												
[4:2]	Reserved	保留												
[1]	GPE_MFP1	<p><b>PE.1管脚功能选择</b> 1 = PE.1 作为 PWM7 0 = PE.1 作为 GPIOE[1]</p>												
[0]	GPE_MFP0	<p><b>PE.0管脚功能选择</b> 1 = PE.0 作为 PWM6 0 = PE.0 作为 GPIOE[0]</p>												



**GPIOF 多功能管脚控制寄存器(GPF\_MFP)**

寄存器	偏移量	R/W	描述	复位后的值
GPF_MFP	GCR_BA+0x44	R/W	GPIOF多功能与输入类型控制寄存器	0x0000_000X

注: GPF\_MFP[3]/GPF\_MFP[2] 的默认值是 1。GPF\_MFP[1]/GPF\_MFP[0] 的默认值由用户配置CGPFMFP 决定。

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				GPF_TYPE[3:0]			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				GPF_MFP3	GPF_MFP2	GPF_MFP1	GPF_MFP0

Bits	描述	
[31:20]	Reserved	保留
[19:16]	GPF_TYPEn	1 = 使能GPIOF[3:0] I/O Schmitt 触发输入 0 = 禁用GPIOF[3:0] I/O Schmitt 触发输入.
[15:4]	Reserved	保留
[3]	GPF_MFP3	<b>PF.3管脚功能选择</b> 1 = PF.3作为PS2_CLK. 0 = PF.3作为GPIOF[3].
[2]	GPF_MFP2	<b>PF.2管脚功能选择</b> 1 = PF.2作为PS2_DAT. 0 = PF.2作为GPIOF[2].
[1]	GPF_MFP1	<b>PF.1管脚功能选择</b> 1 = PF.1作为XT1_IN. 0 = PF.1作为GPIOF[1]. 注: 该位仅读并且由用户配置 CGPFMFP (Config0[27])决定.
[0]	GPF_MFP0	<b>PF.0管脚功能选择</b> 1 = PF.0作为XT1_OUT. 0 = PF.0作为GPIOF[0]. 注: 该位仅读并且由用户配置 CGPFMFP (Config0[27])决定.

复用多功能管脚控制寄存器(ALT\_MFP)

寄存器	偏移量	R/W	描述	复位后的值
ALT_MFP	GCR_BA+0x50	R/W	复用多功能管脚控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PB2_CPO0	PB8_CLKO	Reserved	PB3_T3EX	PB2_T2EX	PE5_T1EX	PB15_T0EX
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					PB12_CLKO	PA15_I2SMCLK	PC3_I2SDO
7	6	5	4	3	2	1	0
PC2_I2SDI	PC1_I2SBCLK	PC0_I2SLRCLK	PB11_PWM4	PB14_S31	PA7_S21	PB9_S11	PB10_S01

Bits	描述																				
[31]	Reserved 保留																				
[30]	<p>PB2_CPO0 (ALT_MFP[30]), PB2_T2EX (ALT_MFP[26]) 和 GPB_MFP[2] 决定了PB.2 功能.</p> <table border="1"> <thead> <tr> <th>PB2_CPO0</th> <th>PB2_T2EX</th> <th>GPB_MFP[2]</th> <th>PB.2 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>UART0_nRTS</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>TM2_EXT</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>CMPO_O</td> </tr> </tbody> </table>	PB2_CPO0	PB2_T2EX	GPB_MFP[2]	PB.2 功能	0	0	0	GPIO	0	0	1	UART0_nRTS	0	1	1	TM2_EXT	1	0	1	CMPO_O
PB2_CPO0	PB2_T2EX	GPB_MFP[2]	PB.2 功能																		
0	0	0	GPIO																		
0	0	1	UART0_nRTS																		
0	1	1	TM2_EXT																		
1	0	1	CMPO_O																		
[29]	<p>PB8_CLKO (ALT_MFP[29]) 和 GPB_MFP[8] 决定了PB.8 功能.</p> <table border="1"> <thead> <tr> <th>PB8_CLKO</th> <th>GPB_MFP[8]</th> <th>PB.8 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>TM0</td> </tr> <tr> <td>1</td> <td>1</td> <td>CLKO (Clock Driver Output)</td> </tr> </tbody> </table>	PB8_CLKO	GPB_MFP[8]	PB.8 功能	0	0	GPIO	0	1	TM0	1	1	CLKO (Clock Driver Output)								
PB8_CLKO	GPB_MFP[8]	PB.8 功能																			
0	0	GPIO																			
0	1	TM0																			
1	1	CLKO (Clock Driver Output)																			
[28]	Reserved 保留																				

[27]	PB3_T3EX	PB3_SC2CD (ALT_MFP1[14]), PB3_T3EX (ALT_MFP[27]) 和 GPB_MFP[3] 决定了PB.3 功能.			
		<b>PB3_SC2CD</b>	<b>PB3_T3EX</b>	<b>GPB_MFP[3]</b>	<b>PB.3 功能</b>
		0	0	0	GPIO
		0	0	1	UART0_nCTS
		0		1	TM3_EXT
		1	0	1	SC2_CD
[26]	PB2_T2EX	PB2_CPO0 (ALT_MFP[30]), PB2_T2EX (ALT_MFP[26]) 和 GPB_MFP[2] 决定了PB.2 功能.			
		<b>PB2_CPO0</b>	<b>PB2_T2EX</b>	<b>GPB_MFP[2]</b>	<b>PB.2 功能</b>
		0	0	0	PIO
		0	0	1	UART0_nRTS
		0	1	1	TM2_EXT
		1	0	1	CMP0_O
[25]	PE5_T1EX	GPE_MFP5 和 PE5_T1EX (ALT_MFP[25]) 决定了PE.5 功能.			
		<b>PE5_1EX</b>	<b>GPE_MFP[5]</b>	<b>PE.5 功能</b>	
		0	0	PIO	
		0	1	PWM5	
		1	1	TM1_EXT	
[24]	PB15_T0EX	PB15_T0EX (ALT_MFP[24]) 和 GPB_MFP[15] 决定了PB.15 功能.			
		<b>PB15_T0EX</b>	<b>GPB_MFP[15]</b>	<b>PB.15 功能</b>	
		0	0	GPIO	
		0	1	INT1	
		1	1	TM0_EXT	
[23:10]	Reserved	保留. (软件必须保持这位为 0.)			
[9]	PA15_I2SMCLK	PA15_SC2PWR (ALT_MFP1[12]), PA15_I2SMCLK (ALT_MFP[9]) 和 GPA_MFP[15] 决定了PA.15 功能.			
		<b>PA15_SC2PWR</b>	<b>PA15_I2SMCLK</b>	<b>GPA_MFP[15]</b>	<b>PA.15 功能</b>
		0	0	0	GPIO
		0	0	1	PWM3 (PWM)
		0	1	1	I2S_MCLK
		1	0	1	SC2_PWR

[8]	PC3_I2SDO	PC3_I2SDO 和 GPC_MFP[3] 决定了PC.3 功能.		
		<b>P 3_I2SD</b>	<b>PC_MFP[3]</b>	<b>PC.3 功能</b>
		0	0	GPIO
		0	1	SPI0_MO I0
		1	1	I2S_DO
[7]	PC2_I2SDI	PC2_I2SDI 和 GPC_MFP[2] 决定了PC.2 功能.		
		<b>PC2_I2 DI</b>	<b>GPC_MFP[2]</b>	<b>PC.2 功能</b>
		0	0	GPIO
		0	1	SPI0_MISO
		1	1	I2S_DI
[6]	PC1_I2SBCLK	PC1_I2SBCLK 和 GPC_MFP[1] 决定了PC.1 功能.		
		<b>PC1_I2SBCLK</b>	<b>GPC_ FP[1]</b>	<b>PC.1 功能</b>
		0	0	GPIO
		0	1	SPI0_CLK
		1	1	I2S_BC K
[5]	PC0_I2SLRCLK	PC0_I2SLRCLK 和 GPC_MFP[0] 决定了PC.0 功能.		
		<b>PC0_I2SLRCLK</b>	<b>GPC_MFP[0]</b>	<b>PC.0 功能</b>
		0	0	GPIO
		0	1	SPI0_SS0
		1		I2S_LRC
[4]	PB11_PWM4	PB11_PWM4 和 GPB_MFP[11] 决定了PB.11 功能.		
		<b>PB1 _PWM4</b>	<b>PB_MFP[11]</b>	<b>PB.11 功能</b>
		0	0	GPIO
		0	1	M3
		1	1	PWM4 (PWM)
[3]	PB14_S31	PB14_S31 和 GPB_MFP[14] 决定了PB.14 功能.		
		<b>PB14_S31</b>	<b>GPB_MFP[14]</b>	<b>PB.14 功能</b>
		0	0	GPIO
		0	1	INT0
		1	1	SPI3_SS1

[2]	PA7_S21	PA7_SC1DAT (ALT_MFP1[6]), PA7_S21 (ALT_MFP[2]) 和 GPA_MFP[7] 决定了PA.7 功能.			
		<b>PA7_SC1DAT</b>	<b>PA7_S 1</b>	<b>GPA_MFP[7]</b>	<b>PA.7 功能</b>
		0	0	0	GPIO
		0	0	1	ADC7
		0	1	1	SPI2_SS1
		1	0	1	SC1_DAT
[1]	PB9_S11	PB9_S11 和 GPB_MFP[9] 决定了PB.9 功能.			
		<b>PB9_S11</b>	<b>GPB_MFP[9]</b>	<b>PB.9 功能</b>	
		0	0	GP O	
		0	1	T 1	
		1	1	SPI1_SS1	
[0]	PB10_S01	PB10_S01 和 GPB_MFP[10] 决定了PB.10 功能.			
		<b>PB10_S01</b>	<b>GPB MFP[10]</b>	<b>PB.10 功能</b>	
		0	0	G IO	
		0	1	TM2	
		1	1	SPI0_SS1	

复用多功能管脚控制寄存器1 (ALT\_MFP1)

寄存器	偏移量	R/W	描述	复位后的值
ALT_MFP1	GCR_BA+0x58	R/W	复用多功能管脚控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	PB3_ SC2CD	PA14_ SC2RST	PA15_ SC2PWR	PA12_ SC2DAT	PA13_ SC2CLK	PC7_ SC1CD	PA5_ SC1RST
7	6	5	4	3	2	1	0
PA4_ SC1PWR	PA7_ SC1DAT	PA6_ SC1CLK	PC6_ SC0CD	PA1_ SC0RST	PA0_ SC0PWR	PA3_ SC0DAT	PA2_ SC0CLK

Bits	描述																					
[31:15]	Reserved	保留																				
[14]	PB3_SC2CD	<p>PB3_SC2CD (ALT_MFP1[14]), PB3_T3EX (ALT_MFP[27]) 和 GPB_MFP[3] 决定了PB.3 功能.</p> <table border="1"> <thead> <tr> <th>PB3_SC2CD</th> <th>PB3_T3EX</th> <th>GP_MFP[3]</th> <th>PB.3 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>UART0_nCTS</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>TM3_EXT</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>SC2_CD</td> </tr> </tbody> </table>	PB3_SC2CD	PB3_T3EX	GP_MFP[3]	PB.3 功能	0	0	0	GPIO	0	0	1	UART0_nCTS	0	1	1	TM3_EXT	1	0	1	SC2_CD
PB3_SC2CD	PB3_T3EX	GP_MFP[3]	PB.3 功能																			
0	0	0	GPIO																			
0	0	1	UART0_nCTS																			
0	1	1	TM3_EXT																			
1	0	1	SC2_CD																			
[13]	PA14_SC2RST	<p>PA14_SC2RST (ALT_MFP1[13]) 和 GPA_MFP[14] 决定了PA.14 功能.</p> <table border="1"> <thead> <tr> <th>PA14_SC2RST</th> <th>GPA_FP[14]</th> <th>PA.14 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GP O</td> </tr> <tr> <td>0</td> <td>1</td> <td>PWM2 (PWM)</td> </tr> <tr> <td>1</td> <td>1</td> <td>SC2_RST</td> </tr> </tbody> </table>	PA14_SC2RST	GPA_FP[14]	PA.14 功能	0	0	GP O	0	1	PWM2 (PWM)	1	1	SC2_RST								
PA14_SC2RST	GPA_FP[14]	PA.14 功能																				
0	0	GP O																				
0	1	PWM2 (PWM)																				
1	1	SC2_RST																				
[12]	PA15_SC2PWR	<p>PA15_SC2PWR (ALT_MFP1[12]), PA15_I2SMCLK (ALT_MFP[9]) 和 GPA_MFP[15] 决定了PA.15 功能.</p> <table border="1"> <thead> <tr> <th>PA15_SC2PWR</th> <th>PA15_I2SMCLK</th> <th>GPA_MFP[15]</th> <th>PA.15 功能</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>PWM3 (PWM)</td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td>I2S_MCLK</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>SC2_PWR</td> </tr> </tbody> </table>	PA15_SC2PWR	PA15_I2SMCLK	GPA_MFP[15]	PA.15 功能	0		0	GPIO	0	0	1	PWM3 (PWM)	0	1		I2S_MCLK	1	0	1	SC2_PWR
PA15_SC2PWR	PA15_I2SMCLK	GPA_MFP[15]	PA.15 功能																			
0		0	GPIO																			
0	0	1	PWM3 (PWM)																			
0	1		I2S_MCLK																			
1	0	1	SC2_PWR																			

[11]	PA12_SC2DAT	Bits PA12_SC2DAT (ALT_MFP1[11]) 和 GPA_MFP[12] 决定了 PA.12 功能.			
		<b>PA12_SC2DAT</b>	<b>GPA_MFP[12]</b>	<b>PA.12 功能</b>	
		0	0	GPIO	
		0	1	PWM0 (PWM)	
		1	1	SC2_DAT	
[10]	PA13_SC2CLK	Bits PA13_SC2CLK (ALT_MFP1[10]) 和 GPA_MFP[13] 决定了 PA.13 功能.			
		<b>PA13_SC2CLK</b>	<b>GPA_MF [13]</b>	<b>PA.13 功能</b>	
		0	0	GPIO	
		0		PWM1 (PWM)	
		1	1	SC2_CLK	
[9]	PC7_SC1CD	PC7_SC1CD (ALT_MFP1[9]) 和 GPC_MFP[7] 决定了 PC.7 功能.			
		<b>PC7_SC1CD</b>	<b>GPC_MFP[7]</b>	<b>PC.7 功能</b>	
		0	0	GPIO	
		0	1	CMP0_N	
		1	1	SC1_CD	
[8]	PA5_SC1RST	PA5_SC1RST (ALT_MFP1[8]) 和 GPA_MFP[5] 决定了 PA.5 功能.			
		<b>PA5_SC1RST</b>	<b>PA_MFP[5]</b>	<b>PA.5 功能</b>	
		0	0	GPIO	
		0	1	ADC5	
		1	1	SC1_RST	
[7]	PA4_SC1PWR	PA4_SC1PWR (ALT_MFP1[7]) 和 GPA_MFP[4] 决定了 PA.4 功能.			
		<b>PA4_SC1PWR</b>	<b>GPA_MFP[4]</b>	<b>PA.4 功能</b>	
		0	0	GPIO	
		0	1	ADC4	
		1	1	SC1_PWR	
[6]	PA7_SC1DAT	PA7_SC1DAT (ALT_MFP1[6]), PA7_S21 (ALT_MFP[2]) 和 GPA_MFP[7] 决定了 PA.7 功能.			
		<b>PA7_SC1DAT</b>	<b>PA7_S21</b>	<b>GPA_FP 7]</b>	<b>PA.7 功能</b>
		0	0	0	GPIO
			0	1	ADC7
		0	1	1	SPISS21 SPI2)
		1	0	1	SC1_DAT

[5]	PA6_SC1CLK	PA6_SC1CLK (ALT_MFP1[5]) 和 GPA_MFP[6] 决定了PA.6 功能.		
		<b>PA6_SC1CLK</b>	<b>GPA_MFP[6]</b>	<b>PA.6 功能</b>
		0	0	GPIO
		0		ADC6
		1	1	SC1_CLK
[4]	PC6_SC0CD	PC6_SC0CD (ALT_MFP1[4]) 和 GPC_MFP[6] 决定了PC.6 功能.		
		<b>PC6_SC0CD</b>	<b>G C_MFP[6]</b>	<b>PC.6 功能</b>
		0	0	GPIO
		0	1	CMP0_P
		1	1	SC0_CD
[3]	PA1_SC0RST	PA1_SC0RST (ALT_MFP1[3]) 和 GPA_MFP[1] 决定了 PA.1 功能.		
		<b>PA1_SC0RST</b>	<b>GPA_MFP[1]</b>	<b>PA.1 功能</b>
		0	0	GPI
		0	1	ADC1
		1	1	SC0_RST
[2]	PA0_SC0PWR	PA0_SC0PWR (ALT_MFP1[2]) 和 GPA_MFP[0] 决定了 PA.0 功能		
		<b>PA0_C0PWR</b>	<b>GPA_MFP[0]</b>	<b>PA.0 功能</b>
		0	0	GPIO
		0	1	ADC0
		1	1	SC0_PWR
[1]	PA3_SC0DAT	PA3_SC0DAT (ALT_MFP1[1]) 和 GPA_MFP[3] 决定了PA.3 功能.		
		<b>PA3_SC0DAT</b>	<b>GPA_MFP[3]</b>	<b>PA.3 功能</b>
		0	0	GPIO
		0	1	ADC3
			1	SC0_DAT
[0]	PA2_SC0CLK	PA2_SC0CLK (ALT_MFP1[0]) 和 GPA_MFP[2]决定了 PA.2 功能.		
		<b>PA2_SC0CLK</b>	<b>GPA_MFP[2]</b>	<b>PA.2 功能</b>
		0	0	GPIO
		0	1	ADC2
		1	1	SC0_CLK



**HIRC 校准控制寄存器 (IRCTRMCTL)**

寄存器	偏移量	R/W	描述	复位后的值
IRCTRMCTL	GCR_BA+0x80	R/W	IRC 校准控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							CLKERR_ STOP_EN

7	6	5	4	3	2	1	0
TRIM_RETRY_CNT		TRIM_LOOP		Reserved		TRIM_SEL	

Bits	描述	
[31:9]	Reserved	保留
[8]	CLKERR_STOP_EN	<p>当侦测到32.768KHz 错误, 停止校准</p> <p>0 =当32.768KHz 时钟错误被侦测到, HIRC继续校准操作</p> <p>1 =当32.768KHz 时钟错误被侦测到, HIRC停止校准操作</p>
[7:6]	TRIM_RETRY_CNT	<p><b>校准值更新限制次数</b></p> <p>这部分定义了了在HIRC的频率被锁定之前, 自动校准电路会多少次试着更新HIRC的校准值</p> <p>一旦 HIRC 被锁定, 内部校准值更新计算器会复位</p> <p>如果校准值更新计算器达到限定值并且HIRC的频率仍然没有被锁定。 自动校准操作会被禁止 并且TRIM_SEL 会被清 00..</p> <p>00 =校准重试计数限制是 64</p> <p>01 =校准重试计数限制是128</p> <p>10 =校准重试计数限制是256</p> <p>11 =校准重试计数限制是512</p>
[5:4]	TRIM_LOOP	<p><b>校准计算循环</b></p> <p>这部分定义了校准值计算是在 多少个 32.768KHz 时钟基础上的.</p> <p>例如, 如果TRIM_LOOP 被设置成00, 自动校准电路会在平均频率偏差4 个32.768KHz 时钟的基础上计算校准值.</p> <p>00 = 校准值在平均偏差 4 个clocks基础上计算.</p> <p>01 = 校准值在平均偏差 8 个clocks基础上计算</p> <p>10 = 校准值在平均偏差 16个 clocks基础上计算.</p> <p>11 = 校准值在平均偏差 32 个clocks基础上计算</p>
[3:2]	Reserved	保留
[1:0]	TRIM_SEL	<p><b>校准频率选择</b></p> <p>该域表明内部22.1184 MHz 高速振荡器的目标频率会自动校准到精确的22.1184MHz或24MHz</p> <p>如果没有选择任何目标频率(TRIM_SEL 为 00), HIRC自动校准功能被禁止</p> <p>在自动校准操作期间, 如果因为CLKERR_STOP_EN置1, 时钟错误被侦测到或者自动校准次数到达限制次数, 这两个位被自动清00</p> <p>00 =禁止HIRC自动校准功能.</p> <p>01 =使能HIRC自动校准功能 并且校准 HIRC到22.1184 MHz.</p> <p>10 =使能HIRC自动校准功能 并且校准 HIRC到24MHz</p> <p>11 = 保留.</p>

**HIRC校准中断使能寄存器(IRCTRIMIEN)**

寄存器	偏移量	R/W	描述	复位后的值
IRCTRIMIEN	GCR_BA+0x84	R/W	IRC校准中断使能寄存器	0x0000_0000

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

Bits	描述	
[31:3]	Reserved	保留
[2]	CLKERR_IEN	<p><b>时钟错误中断使能</b></p> <p>这个位控制着在自动校准期间，当时钟 不准确的时候，CPU是否会产生中断</p> <p>如果这位是1，并且在自动校准操作期间，CLKERR_INT 被置1，会产生一个中断通知时钟频率是不准确的。</p> <p>1 =使能CLKERR_INT 状态触发CPU中断</p> <p>0 = 禁止CLKERR_INT 状态触发CPU中断.</p>
[1]	TRIM_FAIL_IEN	<p><b>校准失败中断使能</b></p> <p>这一位控制着当HIRC校准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>如果这一位是1，并且在自动校准操作期间，TRIM_FAIL_INT 被置1,会产生一个中断通知HIRC 达到自动校准值更新限制次数。</p> <p>0 = 禁止TRIM_FAIL_INT状态触发CPU中断.</p> <p>1 =使能TRIM_FAIL_INT状态触发CPU中断.</p>
[0]	Reserved	保留

**HIRC 校准中断状态寄存器(IRCTRIMINT)**

寄存器	偏移量	R/W	描述	复位后的值
IRCTRIMINT	GCR_BA+0x88	R/W	IRC 校准中断状态寄存器	0x0000_0000

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

Bits	描述
------	----

[31:3]	Reserved	保留
[2]	CLKERR_INT	<p><b>时钟错误中断状态</b></p> <p>当外部32.768 kHz低速晶振或内部22.1184 MHz高速振荡器的频率移动到大于不合理的值，该位被置1，表明时钟频率是不准确的。</p> <p>一旦该位被置1，如果CLKERR_STOP_EN置1，自动校准操作会停止并且TRIM_SEL会被硬件自动清0。</p> <p>如果这位是1并且CLKERR_IEN是1，会产生一个中断通知时钟频率是不正确的。向该位写1清零。</p> <p>0 = 时钟频率是正确的 1 = 时钟频率是不正确的</p>
[1]	TRIM_FAIL_INT	<p><b>校准失败中断状态</b></p> <p>该位表明内部22.1184 MHz高速振荡器校准值达到更新限制次数并且内部22.1184 MHz高速振荡器时钟频率仍然没有锁定。一旦这位被置1，自动校准操作会停止，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校准值更新限制次数已达到。向该位写1清零。</p> <p>0 = 校准值更新限制次数没有达到 1 =校准值更新限制次数已达到并且内部22.1184 MHz高速振荡器频率仍然没有锁定</p>
[0]	FREQ_LOCK	<p><b>HIRC频率锁定状态</b></p> <p>这位表明内部22.1184 MHz高速振荡器频率锁定。</p> <p>这是个状态位不会触发任何中断</p>

**寄存器写保护控制寄存器 (REGWRPROT)**

有些系统控制寄存器需要被保护起来，以防止误操作而影响芯片运行，这些寄存器在上电复位到用户解锁之前是锁定的。用户可以连续依次写入“59h” “16h” “88h” 到寄存器 REGWRPROT（地址：0x5000\_0100）解锁定。在这三个数据之间写入任何其他数据，不同时序或写入其他地址都会中止整个时序，导致无法解锁。

解锁后，用户可以检测解锁指示位：0x5000\_0100 的bit0，“1”表示已经解锁定，“0”表示锁定。用户可以更新目标寄存器的值，向“0x5000\_0100”写入任何值，就可以重新锁定保护寄存器。

该位寄存器用于禁用/使能保护寄存器，读取得到 REGPROTDIS 状态。

寄存器	偏移量	R/W	描述	复位后的值
REGWRPROT	GCR_BA+0x100	R/W	寄存器写保护控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8

Reserved							
7	6	5	4	3	2	1	0
REGWRPROT[7:1]							REGWRPROT [0] REGPROTDIS

Bits	描述	
[31:16]	Reserved	保留
[7:0]	REGWRPROT	<p><b>寄存器写保护码 (Write Only)</b></p> <p>一些寄存器具有写保护功能。写这些保护寄存器必须通过依次写入“59h”, “16h”, “88h” 到 REGWRPROT 解除锁定状态。这个时序完成之后, REGPROTDIS 位将被置1, 写保护寄存器可以正常写入数据。</p>
[0]	REGPROTDIS	<p><b>寄存器写保护禁用标志 (Read only)</b></p> <p>1 = 写保护已禁用。可以写入受保护寄存器 0 = 写保护已使能。不能写入受保护寄存器</p> <p>受保护的寄存器有:</p> <p><b>IPRSTC1:</b> 地址 0x5000_0008  <b>BODCR:</b> 地址 0x5000_0018  <b>PORCR:</b> 地址 0x5000_0024  <b>PWRCON:</b> 地址 0x5000_0200 (在电源唤醒中断被清时, bit[6]不受保护)  <b>APBCLK bit[0]:</b> 地址 0x5000_0208 (bit[0] 是看门狗时钟使能)  <b>CLKSEL0:</b> 地址 0x5000_0210 (选择HCLK 和 CPU STCLK 时钟源)  <b>CLKSEL1 bit[1:0]:</b> 地址 0x5000_0214 (用于看门狗时钟源选择)  <b>NMI_SEL bit[8]:</b> 地址0x5000_0380 (用于 NMI_EN时钟源选择)  <b>ISPCON:</b> 地址 0x5000_C000 (Flash ISP 控制寄存器)  <b>WTCR:</b> 地址 0x4000_4000  <b>FATCON:</b> 地址 0x5000_C018</p>

#### 4.2.7 系统定时器 (SysTick)

Cortex-M0 包含系统定时器: SysTick。SysTick 提供一种简单的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。该计数器可用作实时系统(RTOS) 的滴答定时器或一个简单的计数器。

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”。

4.2.7.1 系统定时器控制寄存器映射

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

寄存器	偏移量	R/W	描述	复位后的值
<b>SYST 基地址:</b> <b>SYST_BA = 0xE000_E000</b>				
<b>SYST_CSR</b>	SYST_BA+0x10	R/W	SysTick 控制与状态寄存器	0x0000_0000
<b>SYST_RVR</b>	SYST_BA+0x14	R/W	SysTick 重新加载值寄存器	0xFFFF_FFFF
<b>SYST_CVR</b>	SYST_BA+0x18	R/W	SysTick 当前值寄存器	0xFFFF_FFFF



4.2.7.2 系统定时器控制寄存器描述

**SysTick控制与状态寄存器(SYST\_CSR)**

寄存器	偏移量	R/W	描述	复位后的值
SYST_CSR	SYST_BA+0x10	R/W	SysTick控制与状态寄存器	0x0000_0000

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

Bits	描述	
[31:17]	Reserved	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(SYST_CVR)时，COUNTFLAG 被清零。
[15:3]	Reserved	保留
[2]	CLKSRC	1 = 内核时钟用作 SysTick. 0 = 时钟源为外部参考时钟
[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	SYST_BA+0x14	R/W	SysTick重新加载值寄存器	0xFFFF_FFFF

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

Bits	描述	
[31:24]	Reserved	保留
[23:0]	RELOAD	当计数器达到 0 时，这个值加载到当前值寄存器 (SYST_CVR)。

**SysTick当前值寄存器(SYST\_CVR)**

寄存器	偏移量	R/W	描述	复位后的值
SYST_CVR	SYST_BA+0x18	R/W	SysTick当前值寄存器	0xFFFF_FFFF

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

Bits	描述	
[31:24]	Reserved	保留
[23:0]	CURRENT	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 (见 SysTick 重载值寄存器)。

#### 4.2.8 嵌套向量中断控制器 (NVIC)

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：

- 支持嵌套和向量中断
- 自动保存和恢复处理器状态
- 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。当开始地址取得时，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”。

4.2.8.1 异常模式和系统中断映射

NuMicro™ NUC100 系列支持表 4-2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
Reserved	4 ~ 10	保留
SVCall	11	可配置
Reserved	12 ~ 13	保留
PendSV	14	可配置
SysTick	15	可配置
Interrupt (IRQ0 ~ IRQ31)	16 ~ 47	可配置

表 4-2 异常模式

向量号	中断号 (Bit In Interrupt Registers)	中断名称	源 IP	中断描述
0 ~ 15	-	-	-	系统异常
16	0	<b>BOD_INT</b>	Brown-out	欠压检测中断
17	1	<b>WDT_INT</b>	WDT	看门狗定时器中断
18	2	<b>EINT0</b>	GPIO	PB.14 管脚上的外部信号中断
19	3	<b>EINT1</b>	GPIO	PB.15 管脚上的外部信号中断
20	4	<b>GPAB_INT</b>	GPIO	PA[15:0]/PB[13:0] 的外部信号中断
21	5	<b>GPCDEF_INT</b>	GPIO	PC[15:0]/PD[15:0]/PE[15:0]/ PF[3:0] 的外部信号中断
22	6	<b>PWMA_INT</b>	PWM0~3	PWM0, PWM1, PWM2 与 PWM3 中断
23	7	<b>PWMB_INT</b>	PWM4~7	PWM4, PWM5, PWM6 与 PWM7 中断
24	8	<b>TMR0_INT</b>	TMR0	Timer 0 中断
25	9	<b>TMR1_INT</b>	TMR1	Timer 1 中断
26	10	<b>TMR2_INT</b>	TMR2	Timer 2 中断
27	11	<b>TMR3_INT</b>	TMR3	Timer 3 中断
28	12	<b>UART02_INT</b>	UART0/2	UART0 与 UART2 中断
29	13	<b>UART1_INT</b>	UART1	UART1 中断

30	14	<b>SPI0_INT</b>	SPI0	SPI0 中断
31	15	<b>SPI1_INT</b>	SPI1	SPI1 中断
32	16	<b>SPI2_INT</b>	SPI2	SPI2 中断
33	17	<b>SPI3_INT</b>	SPI3	SPI3 中断
34	18	<b>I2C0_INT</b>	I <sup>2</sup> C0	I <sup>2</sup> C0 中断
35	19	<b>I2C1_INT</b>	I <sup>2</sup> C1	I <sup>2</sup> C1 中断
36	20	<b>Reserved</b>	-	-
37	21	<b>Reserved</b>	-	-
38	22	<b>SC012_INT</b>	SC0/1/2	SC0, SC1 和 SC2 中断
39	23	<b>USB_INT</b>	USBD	USB 2.0 FS 设备中断
40	24	<b>PS2_INT</b>	PS/2	PS/2 中断
41	25	<b>ACMP_INT</b>	ACMP	模拟比较器-0 或 模拟比较器-1 中断
42	26	<b>PDMA_INT</b>	PDMA	PDMA 中断
43	27	<b>I2S_INT</b>	I <sup>2</sup> S	I <sup>2</sup> S 中断
44	28	<b>PWRWU_INT</b>	CLKC	从掉电状态唤醒的时钟控制器中断
45	29	<b>ADC_INT</b>	ADC	ADC 中断
46	30	<b>IRC_INT</b>	IRC	IRC TRIM 中断
47	31	<b>RTC_INT</b>	RTC	RTC 中断

表 4-3 系统中断映射

4.2.8.2 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器的入口地址。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 4-4 向量表格式

4.2.8.3 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 Register 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。寄存器 Clear-Pending 在中断响应时的不影响执行状态。

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。

4.2.8.4 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
NVIC 基地址: NVIC_BA = 0xE000_E000				
NVIC_ISER	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31 设置使能控制寄存器	0x0000_0000
NVIC_ICER	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31 清使能控制寄存器	0x0000_0000
NVIC_ISPR	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	0x0000_0000
NVIC_ICPR	NVIC_BA+0x280	R/W	IRQ0 ~ IRQ31 清挂起控制寄存器	0x0000_0000
NVIC_IPR0	NVIC_BA+0x400	R/W	IRQ0 ~ IRQ3 优先级控制寄存器	0x0000_0000
NVIC_IPR1	NVIC_BA+0x404	R/W	IRQ4 ~ IRQ7 优先级控制寄存器	0x0000_0000
NVIC_IPR2	NVIC_BA+0x408	R/W	IRQ8 ~ IRQ11 优先级控制寄存器	0x0000_0000
NVIC_IPR3	NVIC_BA+0x40C	R/W	IRQ12 ~ IRQ15 优先级控制寄存器	0x0000_0000
NVIC_IPR4	NVIC_BA+0x410	R/W	IRQ16 ~ IRQ19 优先级控制寄存器	0x0000_0000
NVIC_IPR5	NVIC_BA+0x414	R/W	IRQ20 ~ IRQ23 优先级控制寄存器	0x0000_0000
NVIC_IPR6	NVIC_BA+0x418	R/W	IRQ24 ~ IRQ27 优先级控制寄存器	0x0000_0000
NVIC_IPR7	NVIC_BA+0x41C	R/W	IRQ28 ~ IRQ31 优先级控制寄存器	0x0000_0000



4.2.8.1 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	NVIC_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[31:24]							
23	22	21	20	19	18	17	16
SETENA [23:16]							
15	14	13	12	11	10	9	8
SETENA [15:8]							
7	6	5	4	3	2	1	0
SETENA[7:0]							

Bits	描述
[31:0]	<p>SETENA</p> <p>使能一个或者多个中断，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                      写 0 无效                      读取该寄存器返回当前使能状态。</p>

**IRQ0 ~ IRQ31清使能控制寄存器(NVIC\_ICER)**

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

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

Bits	描述	
[31:0]	CLRENA	禁用一个或者多个中断，每位代表 IRQ0 ~ IRQ31 的中断号（向量号：从16到 47）。 写 1 禁用相应中断 写 0 无效 读取该寄存器返回当前使能状态。

**IRQ0 ~ IRQ31设置挂起控制寄存器(NVIC\_ISPR)**

寄存器	偏移量	R/W	描述	复位后的值
NVIC_ISPR	NVIC_BA+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[31:24]							
23	22	21	20	19	18	17	16
SETPEND [23:16]							
15	14	13	12	11	10	9	8
SETPEND [15:8]							
7	6	5	4	3	2	1	0
SETPEND [7:0]							

Bits	描述	
[31:0]	SETPEND	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）。 写 0 无效 读取该寄存器返回当前等待处理的中断状态。

**IRQ0 ~ IRQ31清挂起控制寄存器(NVIC\_ICPR)**

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

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

Bits	描述	
[31:0]	CLRPEND	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）。 写 0 无效 读取该寄存器返回当前等待处理的中断状态。

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_2		Reserved					
15	14	13	12	11	10	9	8
PRI_1		Reserved					
7	6	5	4	3	2	1	0
PRI_0		Reserved					

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

**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	NVIC_BA+0x404	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		Reserved					
23	22	21	20	19	18	17	16
PRI_6		Reserved					
15	14	13	12	11	10	9	8
PRI_5		Reserved					
7	6	5	4	3	2	1	0
PRI_4		Reserved					

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

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_10		Reserved					
15	14	13	12	11	10	9	8
PRI_9		Reserved					
7	6	5	4	3	2	1	0
PRI_8		Reserved					

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

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_14		Reserved					
15	14	13	12	11	10	9	8
PRI_13		Reserved					
7	6	5	4	3	2	1	0
PRI_12		Reserved					

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



**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_18		Reserved					
15	14	13	12	11	10	9	8
PRI_17		Reserved					
7	6	5	4	3	2	1	0
PRI_16		Reserved					

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

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_22		Reserved					
15	14	13	12	11	10	9	8
PRI_21		Reserved					
7	6	5	4	3	2	1	0
PRI_20		Reserved					

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

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_26		Reserved					
15	14	13	12	11	10	9	8
PRI_25		Reserved					
7	6	5	4	3	2	1	0
PRI_24		Reserved					

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

**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	NVIC_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		Reserved					
23	22	21	20	19	18	17	16
PRI_30		Reserved					
15	14	13	12	11	10	9	8
PRI_29		Reserved					
7	6	5	4	3	2	1	0
PRI_28		Reserved					

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

4.2.8.2 中断源控制寄存器映射

除了 NVIC 相关的中断控制寄存器外，NuMicro™ NUC100 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：

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
<b>INT 基地址:</b>				
<b>INT_BA = 0x5000_0300</b>				
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) 中断源识别	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) 中断源识别	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0) 中断源识别	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1) 中断源识别	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (GPA/B) 中断源识别	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (GPC/D/E/F) 中断源识别	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWMA) 中断源识别	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (PWMB) 中断源识别	0xXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) 中断源识别	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) 中断源识别	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (TMR2) 中断源识别	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (TMR3) 中断源识别	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (URT0/2) 中断源识别	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (URT1) 中断源识别	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI0) 中断源识别	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	IRQ15 (SPI1) 中断源识别	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	IRQ16 (SPI2) 中断源识别	0xXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	R	IRQ17 (SPI3) 中断源识别	0xXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I <sup>2</sup> C0) 中断源识别	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I <sup>2</sup> C1) 中断源识别	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	IRQ20 (保留) 中断源识别	0xXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	R	IRQ21 (保留) 中断源识别	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	IRQ22 (SC0/1/2) 中断源识别	0xXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	R	IRQ23 (USBD) 中断源识别	0xXXXX_XXXX

<b>IRQ24_SRC</b>	INT_BA+0x60	R	IRQ24 (PS/2) 中断源识别	0xFFFF_XXXX
<b>IRQ25_SRC</b>	INT_BA+0x64	R	IRQ25 (ACMP) 中断源识别	0xFFFF_XXXX
<b>IRQ26_SRC</b>	INT_BA+0x68	R	IRQ26 (PDMA) 中断源识别	0xFFFF_XXXX
<b>IRQ27_SRC</b>	INT_BA+0x6C	R	IRQ27 (I <sup>2</sup> S) 中断源识别	0xFFFF_XXXX
<b>IRQ28_SRC</b>	INT_BA+0x70	R	IRQ28 (PWRWU) 中断源识别	0xFFFF_XXXX
<b>IRQ29_SRC</b>	INT_BA+0x74	R	IRQ29 (ADC) 中断源识别	0xFFFF_XXXX
<b>IRQ30_SRC</b>	INT_BA+0x78	R	IRQ30 (IRC) 中断源识别	0xFFFF_XXXX
<b>IRQ31_SRC</b>	INT_BA+0x7C	R	IRQ31 (RTC) 中断源识别	0xFFFF_XXXX
<b>NMI_SEL</b>	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	0x0000_0000
<b>MCU_IRQ</b>	INT_BA+0x84	R/W	MCU 中断请求源寄存器	0x0000_0000

4.2.8.1 中断源寄存器 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 ..... INT_BA+0x7C	R	IRQ0 (BOD) 中断源识别 : IRQ31 (RTC) 中断源识别	0xXXXX_XXXX

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

Bits	地址	INT-Num	描述
[2:0]	INT_BA+0x00	0	Bit2: 0 Bit1: 0 Bit0: BOD_INT
[2:0]	INT_BA+0x04	1	Bit2: 0 Bit1: WWDT_INT Bit0: WDT_INT
[2:0]	INT_BA+0x08	2	Bit2: 0 Bit1: 0 Bit0: EINT0 – PB.14 上的外部中断 0
[2:0]	INT_BA+0x0C	3	Bit2: 0 Bit1: 0 Bit0: EINT1 – PB.15 上的外部中断 1
[2:0]	INT_BA+0x10	4	Bit2: 0 Bit1: GPB_INT Bit0: GPA_INT
[3:0]	INT_BA+0x14	5	Bit3: GPF_INT Bit2: GPE_INT Bit1: GPD_INT Bit0: GPC_INT
[3:0]	INT_BA+0x18	6	Bit3: PWM3_INT Bit2: PWM2_INT

			Bit1: PWM1_INT Bit0: PWM0_INT
[3:0]	INT_BA+0x1C	7	Bit3: PWM7_INT Bit2: PWM6_INT Bit1: PWM5_INT Bit0: PWM4_INT
[2:0]	INT_BA+0x20	8	Bit2: 0 Bit1: 0 Bit0: TMR0_INT
[2:0]	INT_BA+0x24	9	Bit2: 0 Bit1: 0 Bit0: TMR1_INT
[2:0]	INT_BA+0x28	10	Bit2: 0 Bit1: 0 Bit0: TMR2_INT
[2:0]	INT_BA+0x2C	11	Bit2: 0 Bit1: 0 Bit0: TMR3_INT
[2:0]	INT_BA+0x30	12	Bit2: 0 Bit1: URT2_INT Bit0: URT0_INT
[2:0]	INT_BA+0x34	13	Bit2: 0 Bit1: 0 Bit0: URT1_INT
[2:0]	INT_BA+0x38	14	Bit2: 0 Bit1: 0 Bit0: SPI0_INT
[2:0]	INT_BA+0x3C	15	Bit2: 0 Bit1: 0 Bit0: SPI1_INT
[2:0]	INT_BA+0x40	16	Bit2: 0 Bit1: 0 Bit0: SPI2_INT
[2:0]	INT_BA+0x44	17	Bit2: 0 Bit1: 0 Bit0: SPI3_INT
[2:0]	INT_BA+0x48	18	Bit2: 0 Bit1: 0 Bit0: I2C0_INT
[2:0]	INT_BA+0x4C	19	Bit2: 0 Bit1: 0 Bit0: I2C1_INT
[2:0]	INT_BA+0x50	20	Bit2: 0 Bit1: 0



			Bit0: 0
[2:0]	INT_BA+0x54	21	Bit2: 0 Bit1: 0 Bit0: 0
[2:0]	INT_BA+0x58	22	Bit2: SC2_INT Bit1: SC1_INT Bit0: SC0_INT
[2:0]	INT_BA+0x5C	23	Bit2: 0 Bit1: 0 Bit0: USB_INT
[2:0]	INT_BA+0x60	24	Bit2: 0 Bit1: 0 Bit0: PS2_INT
[2:0]	INT_BA+0x64	25	Bit2: 0 Bit1: 0 Bit0: ACMP_INT
[2:0]	INT_BA+0x68	26	Bit2: 0 Bit1: 0 Bit0: PDMA_INT
[2:0]	INT_BA+0x6C	27	Bit2: 0 Bit1: 0 Bit0: I2S_INT
[2:0]	INT_BA+0x70	28	Bit2: 0 Bit1: 0 Bit0: PWRWU_INT
[2:0]	INT_BA+0x74	29	Bit2: 0 Bit1: 0 Bit0: ADC_INT
[2:0]	INT_BA+0x78	30	Bit2: 0 Bit1: 0 Bit0: IRC_INT
[2:0]	INT_BA+0x7C	31	Bit2: 0 Bit1: 0 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
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							NMI_EN
7	6	5	4	3	2	1	0
Reserved			NMI_SEL[4:0]				

Bits	描述	
[31:8]	Reserved	保留
[8]	NMI_EN	NMI 中断使能（写保护位） 1 = 使能 NMI 中断 0 = 禁用 NMI 中断 该位为写保护位，写该位需要写“59h”，“16h”，“88h” 到地址 0x5000_0100禁止寄存器写保护功能，参考寄存器REGWRPROT，地址为 GCR_BA+0x100。
[7:5]	Reserved	保留
[4:0]	NMI_SEL	NMI 中断源选择 通过设置 NMI_SEL 可以在外围设备中断中选择 Cortex-M0 的 NMI 中断。

**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[31:24]							
23	22	21	20	19	18	17	16
MCU_IRQ[23:16]							
15	14	13	12	11	10	9	8
MCU_IRQ[15:8]							
7	6	5	4	3	2	1	0
MCU_IRQ[7:0]							

Bits	描述	
[31:0]	MCU_IRQ	<p><b>MCU_IRQ 源寄存器</b></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>MCU_IRQ 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>

#### 4.2.9 系统控制寄存器(SCS)

系统控制寄存器控制了 Cortex-M0 的状态和操作模式，包括 CPUID，Cortex-M0 的中断优先级和 Cortex-M0 的电源管理。

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

## 4.3 时钟控制器

### 4.3.1 概述

时钟控制器为整个芯片提供时钟源，包括系统时钟和所有外围设备时钟，该控制器还通过个别时钟的开或关，时钟源选择和分频器来进行功耗控制。CPU 使能 PWR\_DOWN\_EN(PWRCON[7]) 和 PD\_WAIT\_CPU (PWRCON[8]) 位后，同时CPU Cortex-M0 内核执行 WFI 指令，芯片将进入掉电模式，直到唤醒中断发生，芯片才会退出掉电模式。在掉电模式下，时钟控制器关闭外部 4~24 MHz 晶振和内部 22.1184 MHz 振荡器，以降低整个系统的功耗。

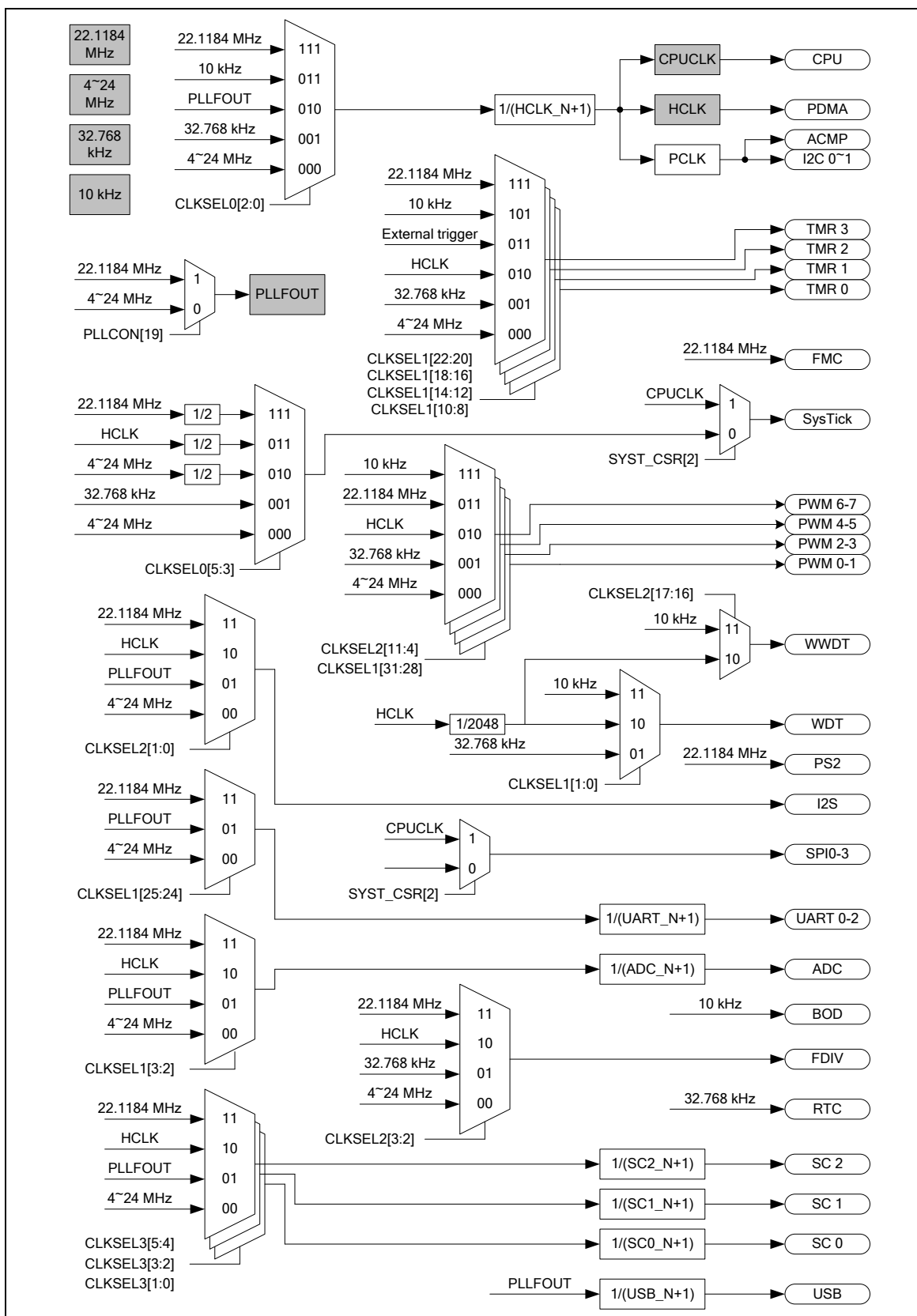


图 4-4 时钟发生器全局框图

### 4.3.2 时钟发生器

时钟发生器由如下 5 个时钟源组成：

- 一个外部 32.768 kHz 晶振
- 一个外部 4~24 MHz 晶振
- 一个可编程的 PLL FOUT（PLL 由外部 4~24 MHz 晶振和内部 22.1184 MHz 振荡器提供时钟源）
- 一个内部 22.1184 MHz 振荡器
- 一个内部 10 kHz 振荡器

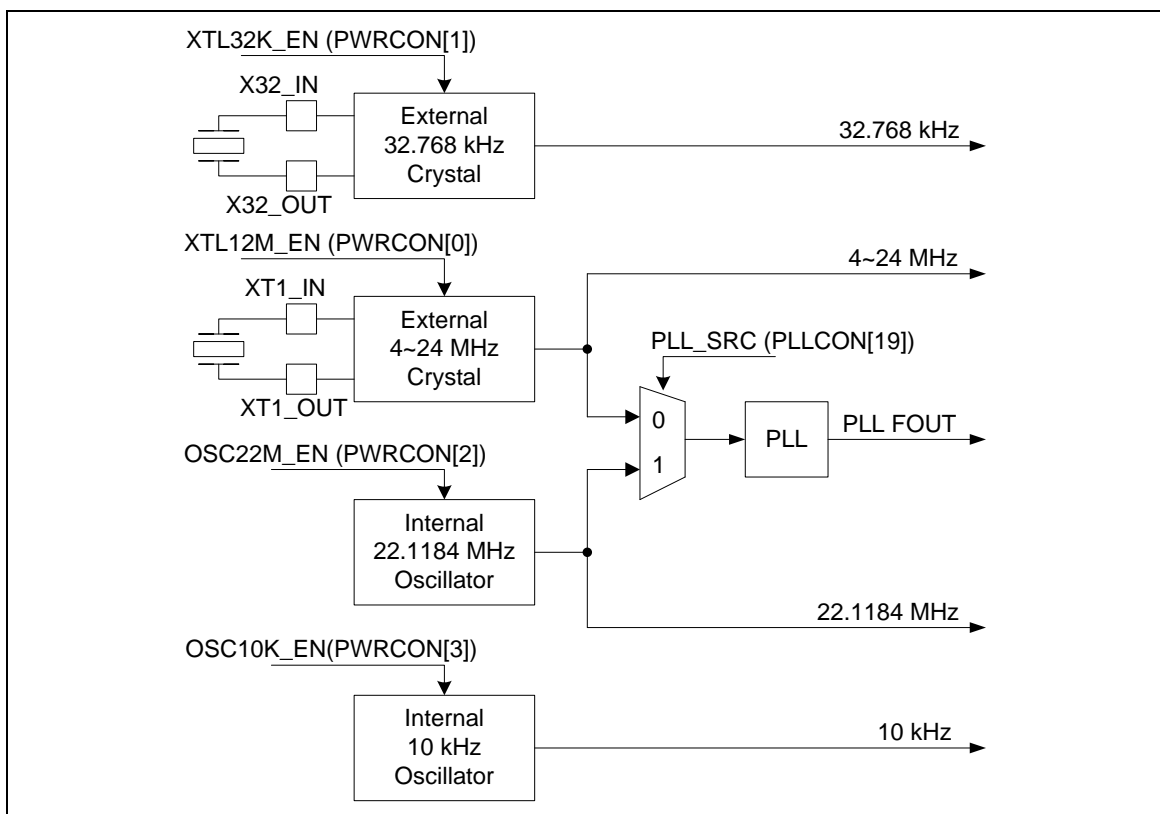


图 4-5 时钟发生器框图

### 4.3.3 系统时钟 & SysTick 时钟

系统时钟有 5 个时钟源，由时钟发生器发生。时钟源切换取决于寄存器 HCLK\_S (CLKSEL0[2:0])，如图 4-6 所示。

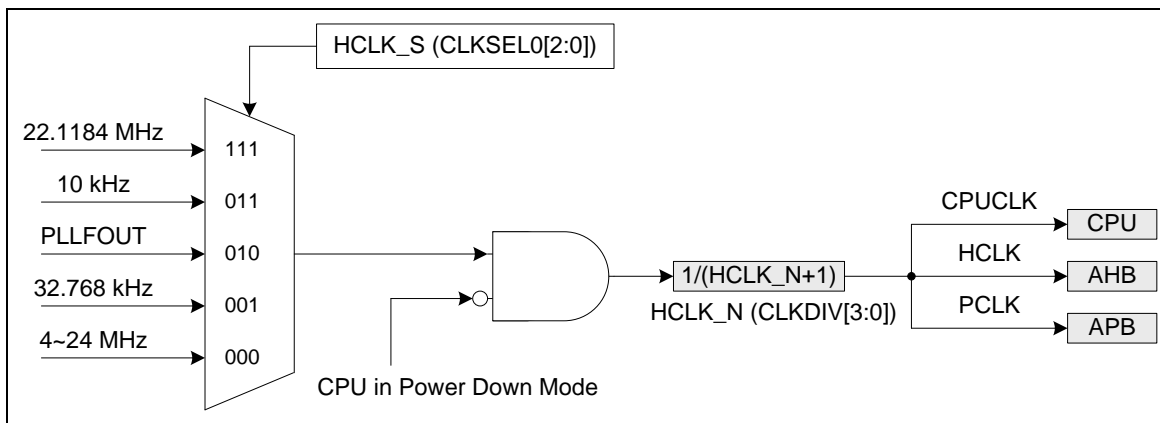


图 4-6 系统时钟框图

Cortex-M0 内核的 SysTick 时钟源可以选择 CPU 时钟或外部时钟 (SYST\_CSR[2])。如果使用外部时钟，SysTick 时钟 (STCLK) 有 5 个时钟源，由时钟发生器产生。时钟源切换取决于寄存器 STCLK\_S (CLKSEL0[5:3])。框图如图 4-7。

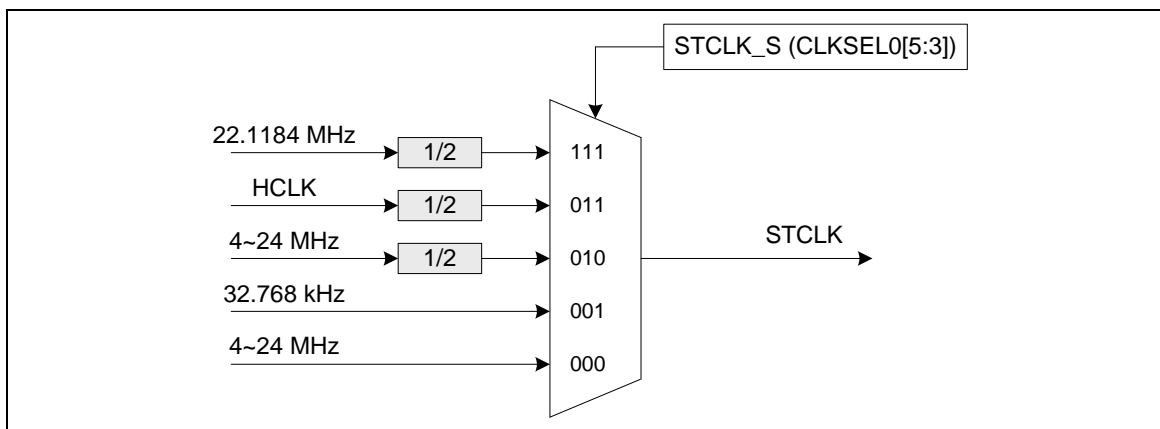


图 4-7 SysTick 时钟控制框图



#### 4.3.4 外围设备时钟

不同的外设，其外围设备时钟有不同的时钟源切换。请参考 5.3.8 节的寄存器 CLKSEL1, CLKSEL2 和 CLKSEL3。

#### 4.3.5 掉电模式时钟

当芯片进入掉电模式后，一些时钟源、外设时钟和系统时钟将被关闭，也有一些时钟源与外设时钟仍在工作。

如下时钟仍在工作：

- 时钟发生器
  - ◆ 内部 10 kHz 振荡器时钟
  - ◆ 外部 32.768 kHz 晶振时钟
- 外设时钟（当这些 IP 采用外部 32.768 kHz 晶振或者 10 kHz 振荡器作为时钟源时）

4.3.6 分频器输出

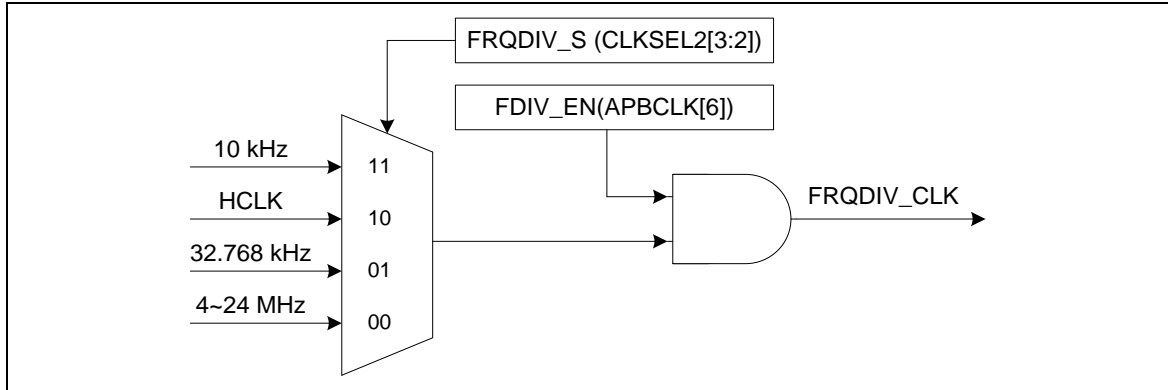


图 4-8 分频器的时钟源

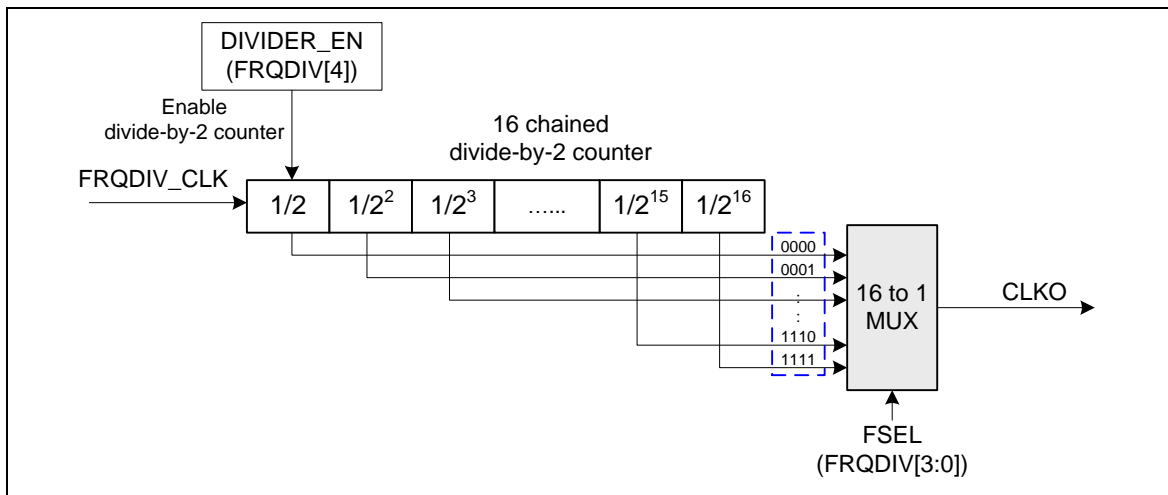


图 4-9 分频器的框图

## 4.4 USB 设备控制器 (USB)

### 4.4.1 概述

该器件有一组 USB 2.0 全速设备控制器和收发器，符合 USB 2.0 全速设备规范，支持 control/bulk/interrupt/ isochronous 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。输入或输出传输，需要通过 APB 接口或 SIE 向 SRAM 写数据或从 SRAM 读数据。用户需要通过缓存分段寄存器 (BUFSEGx) 为每个端点缓存设置有效的 SRAM 地址。

USB 设备控制器共有 6 个可配置的端点。每个端点可以配置为 IN 或者 OUT 类型。所有的操作包括 Control, Bulk, Interrupt 和 Isochronous transfer 传输都由端点模块来执行。端点控制模块还可以用来管理数据同步时序，端点状态控制，当前起始地址，当前事务状态和每个端点的数据缓存状态。

该控制器有 4 种不同的中断事件，分别是唤醒事件，设备插拔事件，USB 事件（如 IN ACK, OUT ACK 等）和 BUS 事件（如 suspend 和 resume 等）。任何时间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\_DRVSE0)，USB 控制器将使 USB\_DP 和 USB\_DM 输出低电平禁止其功能。在禁用 DRVSE0 位之后，主机将重新枚举 USB 设备。

参考文献：USB规范修订版 1.1

### 4.4.2 特征

该通用串行总线 (USB) 为一个带有单独连接器的串行接口，可以连接所有 USB 外设到主机系统。下面是 USB 的一些特征。

- 兼容 USB 2.0 全速规范
- 提供 1 个中断向量，4 个中断事件 (WAKEUP, FLDET, USB and BUS)
- 支持 Control/Bulk/Interrupt/Isochronous 传输类型
- 支持在没有总线活动超过 3 ms 之后的暂停功能
- 为可配置的 Control/Bulk/Interrupt/Isochronous 传输类型提供 6 个端点和最大 512 字节的缓存
- 提供远程唤醒功能

## 4.5 通用 I/O (GPIO)

### 4.5.1 概述

NuMicro™ NUC100 有 80 个通用 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。80 个管脚分配在 GPIOA, GPIOB, GPIOC, GPIOD, GPIOE 与 GPIOF 六个端口上，GPIOA/B/C/D/E 端口最多 16 个管脚，GPIOF 最多 4 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，开漏或准双向模式。复位之后，所有管脚的 I/O 管脚类型取决于 Config0[10] 的设置。在准双向模式，每个 I/O 管脚有一个阻值 110 K $\Omega$ ~300 K $\Omega$  的弱上拉电阻接到 V<sub>DD</sub> 上，V<sub>DD</sub> 的范围从 5.0 V 到 2.5 V。

### 4.5.2 特征

- 四种 I/O 模式：
  - ◆ 准双向模式
  - ◆ 推挽输出
  - ◆ 开漏输出
  - ◆ 高阻态输入
- 通过 GPx\_MFP[31:16] 中 GPx\_TYPE[15:0]，可选 TTL/Schmitt 触发输入。
- I/O 管脚可配置为 边沿/电平触发的中断源。
- 通过 Config0[10] 的设置，可配置所有 I/O 复位之后的默认模式。
  - ◆ 如果 Config[10] 是 0, 所有 GPIO 在芯片复位之后是输入三态模式
  - ◆ 如果 Config[10] 是 1, 所有 GPIO 在芯片复位之后是准双向模式
- 仅在准双向模式，内部上拉电阻才使能。
- 使能管脚的中断功能会同时使能唤醒功能。

## 4.6 I<sup>2</sup>C 串行接口控制器 (Master/Slave) (I<sup>2</sup>C)

### 4.6.1 概述

I<sup>2</sup>C 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。I<sup>2</sup>C 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。

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)。更多关于 I<sup>2</sup>C 总线时序的细节请参考图 4-10。

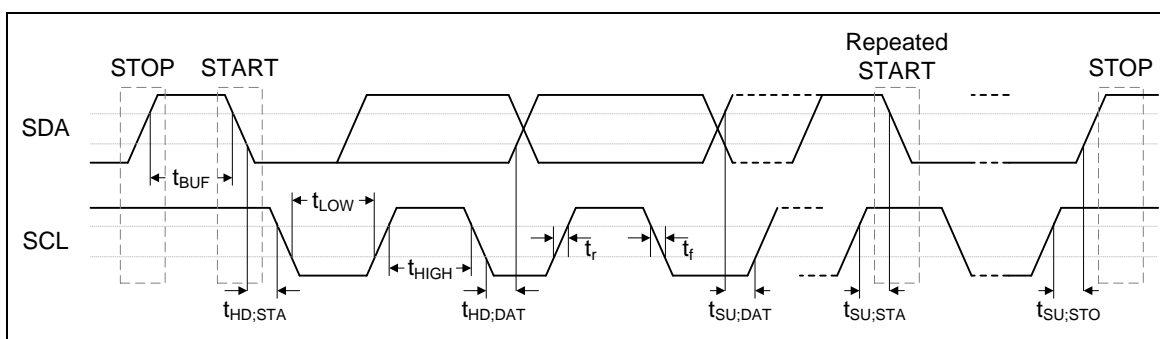


图 4-10 I<sup>2</sup>C 总线时序

设备的片上 I<sup>2</sup>C 逻辑提供符合 I<sup>2</sup>C 总线规范的串行接口。I<sup>2</sup>C 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。I<sup>2</sup>C H/W 接口通过 SDA 与 SCL 两个管脚连到 I<sup>2</sup>C 总线。用于 I<sup>2</sup>C 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。当 I/O 管脚作为 I<sup>2</sup>C 端口使用时，用户必须事先设定 I/O 管脚功能为 I<sup>2</sup>C 功能。

#### 4.6.2 特征

I<sup>2</sup>C 总线通过两条线 (SDA 和 SCL) 在连接到总线上的设备之间传输信息。总线的主要特征:

- 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 I<sup>2</sup>C 总线中止且定时计数器溢出, 产生 I<sup>2</sup>C 中断
- 需要外部上拉确保高电平输出
- 可编辑的时钟适用于不同速率控制
- 支持 7 位地址模式
- I<sup>2</sup>C 总线控制器支持多地址识别 (4组从机地址带 mask 选项)

## 4.7 PWM 发生器和捕捉定时器 (PWM)

### 4.7.1 概述

NuMicro™ NUC100系列有2组PWM，每组有4个PWM，这些发生器可以配置成8个独立的PWM输出PWM0~PWM7，或4组互补的PWM对，(PWM0, PWM1), (PWM2, PWM3), (PWM4, PWM5) 和 (PWM6, PWM7) 4对PWM具有可编程的死区设定。

每组PWM发生器带有一个8-位预分频，一个提供5级时钟源(1, 1/2, 1/4, 1/8, 1/16)的时钟除频器，两个包含2个时钟选择的PWM定时器，两个用于PWM周期控制的16位PWM向下计数计数器，两个16位比较器用于PWM占空比控制及死区发生器。4组PWM发生器提供8个独立的由硬件置位的PWM中断标志，当相应的PWM向下计数周期达到0时触发中断。每个PWM中断源通过相应的使能位允许CPU来请求PWM中断。PWM发生器可以定义为产生单周期PWM周期信号的单触发模式或连续输出PWM波形的连续模式。

当PCR.DZEN01置位，PWM0与PWM1形成互补的PWM功能，这一对PWM的周期，占空比和死区时间由PWM0定时器和死区发生器0决定。同样，PWM互补对(PWM2, PWM3), (PWM4, PWM5)与(PWM6, PWM7)分别由PWM2, PWM4与PWM6定时器和死区发生器2, 4, 6控制。参考 **錯誤! 找不到參照來源**。到图5-43查看PWM定时器的结构。

为防止PWM输出抖动不稳定波形，16位向下计数计数器和16位比较器采用两组缓存。当用户向计数器/比较器寄存器内写入值时，只有当计数器/比较器的值计数到0后，写入值才会被重新写入计数器/比较器。该两组缓存可以避免PWM输出时产生干扰波形。

当16位向下计数计数器达到0时，中断请求产生。如果PWM定时器被定义为连续模式，当向下计数器达到0时，会自动重新导入PWM计数寄存器(CNRx)的设定值并重新开始运行下一个周期。如果定时器设为单触发模式，向下计数器达到0时停止计数，并产生中断请求。

比较器数据用于脉冲高宽度调节，计数控制逻辑在向下计数器计数到比较值时将PWM输出变高。

PWM-定时器其中之一特性是数字输入捕捉功能。若捕捉功能使能，则PWM的输出管脚会被切换成捕捉输入模式。捕捉器0和PWM0共用PWM0中的定时器；捕捉器1和PWM1共用PWM1定时器，以此类推。因此用户在使用捕捉功能之前，必须预先配置PWM定时器。使能捕捉器特性后，当输入通道有上升沿时，捕捉器总是将PWM-计数器的值锁存到捕捉上升沿锁存寄存器(CRLR)，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(CFLR)。捕捉器通道0中断可以通过编程设定CCR0.CRL\_IE0[1]（上升沿触发中断有效）和CCR0.CFL\_IE0[2]（下降沿触发中断有效）来决定中断发生的条件。同样设定CCR0.CRL\_IE1[17]和CCR0.CFL\_IE1[18]，可以设定捕捉器通道1。类似的捕捉器每组的通道2到3通过设定CCR2的相应位也有同样的特性。对每组而言，每当捕捉器触发中断0/1/2/3时，PWM计数器0/1/2/3也会同时被重置。

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PIIR获取中断源，第二步，读CRLRx/CFLRx(x=0~3)获取捕捉值，以及最后写1清PIIR为0。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。例如：

HCLK = 50 MHz, PWM\_CLK = 25 MHz, 中断延时为 900 ns

因此最大捕捉频率为 1/900ns ≈ 1000 kHz

### 4.7.2 特征

#### 4.7.2.1 PWM 功能:

- 最高2个PWM组(PWMA/PWMB)可支持8路PWM通道或4对PWM通道

- PWM 组有两个PWM发生器。每个PWM支持一个8-位预分频器，两个时钟除频器，两个 PWM 定时器（向下计数），一个死区发生器和两个 PWM 输出。
- 最高 16-位 解析度
- PWM 中断请求与 PWM 周期同步
- 单次或自动重载模式
- 边沿对齐模式或中心对齐模式选择

#### 4.7.2.2 捕捉功能:

- 与 PWM 发生器共用定时器模块
- 支持 8 个捕捉输入通道，共享 8 个 PWM 输出通道
- 每个通道支持 1 个上升沿锁存寄存器 (CRLR)，一个下降沿锁存寄存器 (CFLR) 和捕捉中断标志 (CAPIFx)



## 4.8 实时时钟 (RTC)

### 4.8.1 概述

实时时钟 (RTC) 控制器用于记录实时时间及日历功能。RTC 的时钟源由外部 32.768 kHz 晶振提供，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 控制器提供时间信息（秒、分、时）在时间载入寄存器 (TLR) 以及通过日历载入寄存器 (CLR) 提供日历信息（日、月、年）。时间信息由 BCD 码格式进行表示。该控制器也提供闹钟功能，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铃设置。

RTC 控制器支持周期时间节拍和闹钟匹配中断。通过设定时间节拍寄存器 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。当闹钟中断使能 (RIER.AIER=1) 的情况下，当 TLR 和 CLR 里面的值分别等于 TAR（闹钟设定时间寄存器）和 CAR（闹钟日历设定寄存器）时，如果 AIER (RIER[0] 闹钟中断使能位) 使能，中断标志 AIF (RIIR[0]) 将被置 1，并产生闹钟中断请求。在芯片进入空闲或掉电模式之前，如果相应的中断使能位 (AIER 或 TIER) 被置 1，当 IC 处于空闲模式或掉电模式，RTC 时间节拍和闹钟定时中断将唤醒 IC。

### 4.8.2 特征

- 支持时间计数（秒，分，时）和日历计数（日，月，年），用户访问 TLR 可以用来查看时间，用户访问 CLR 可以用来查看日历
- 闹钟寄存器（秒，分，时，日，月，年）
- 12-小时或 24-小时模式可选择
- 闰年自动识别
- 一周天数计数器
- 频率补偿寄存器 (FCR)
- 所有时间日期由 BCD 码表示
- 支持周期时间节拍中断，提供 8 个周期选项供选择 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 及 1 秒
- 支持 RTC 定时节拍和闹钟定时中断
- 支持从掉电模式下或空闲模式下唤醒芯片

## 4.9 串行外围设备接口(SPI)

### 4.9.1 概述

SPI 接口是工作于全双工模式下的同步串行数据通信接口。主/从模式使用四线双向接口通信。NuMicro™ NUC100系列包括4组 SPI 控制器，将从外设得到的数据进行串并转换, 或将数据进行并串转换，发送到外设。每组 SPI 控制器可以被作为一个主机，还可以被设置为外围设备的从机。

该控制器支持不同串行时钟以适应不同应用，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 PDMA 功能访问数据缓冲，也支持双I/O传输模式。

### 4.9.2 特征

- 最多支持四组 SPI 控制器
- 支持主机或从机模式操作
- 支持2 位传输模式
- 支持双 I/O 传输模式
- 可配置传输字长从 8 位到 32-位
- 提供独立 8-层深度发送和接收FIFO 缓存
- 支持 MSB 或 LSB 为最先传输模式
- 主机模式下有 2 条从机选择线
- 支持数据寄存器字节重排序
- 支持字节或字休眠模式
- 主机模式下，支持输出不同串行时钟频率
- 支持 PDMA 传输
- 支持三线，没有从机选择信号、双向接口

## 4.10 定时器控制器 (TMR)

### 4.10.1 概述

定时器控制器包含 4 组 32-位定时器，TIMER0~TIMER3，提供用户便捷的计数定时功能。定时器模块可支持例如频率测量，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。

### 4.10.2 特征

- 4 组 32-位定时器，带24位向上定时器和一个8位的预分频计数器
- 每个定时器都有独立的时钟源
- 提供 one-shot, periodic, toggle 和 continuous 计数操作模式
- 超时周期 = (输入的定时器时钟周期) \* (8-位预分频计数器 + 1) \* (24-位 TCMP)
- 最大计数周期 =  $(1 / T \text{ MHz}) * (2^8) * (2^{24})$ ，T 是定时器周期
- 通过 TDR（定时器数据寄存器）可读取内部 24 位向上计数器的值
- 支持事件计数功能，可用于外部管脚事件的计数
- 支持输入捕捉功能，可用于捕捉或复位计数值
- 支持外部管脚捕捉功能用来复位定时器计数
- 如果定时器中断信号产生，支持芯片从空闲/掉电模式唤醒(TIF 置 1)

## 4.11 看门狗定时器(WDT)

### 4.11.1 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，可以防止系统无限止地挂机，除此之外，看门狗定时器还可将芯片由掉电模式唤醒。

### 4.11.2 特征

- 18-位自由运行 WDT 计数器用于看门狗定时器超时间隔。
- 可选择的超时间隔 ( $2^4 \sim 2^{18}$ )，超时间隔为 104 ms ~ 26.316 s (如果 WDT\_CLK = 10 kHz)。
- 系统保持复位状态的周期为  $(1 / \text{WDT\_CLK}) * 63$
- 支持可选的看门狗定时器复位延时周期，包括 (1024+2)、(128+2)、(16+2) 或 (1+2)个 WDT\_CLK的复位延时周期。
- 当CWDTEN(Config0[31] 看门狗使能)置0，支持芯片上电或复位之后，强制看门狗定时器使能。
- 当看门狗时钟源选择10 kHz低速振荡器，支持看门狗定时器超时唤醒功能。

## 4.12 窗口看门狗定时控制器

### 4.12.1 概述

窗口看门狗定时器的目的是在一个指定的窗口周期中执行系统复位，防止软件在任何不可预知的条件下进入不可控制的状态。

### 4.12.2 特征

- 一个6-bit 下数计数器和一个6-bit比较值使窗口周期可调。
- 可选的 WWDT 时钟预分频计数器使WWDT溢出间隔可变

### 4.12.3 框图

窗口看门狗定时器框图如下。

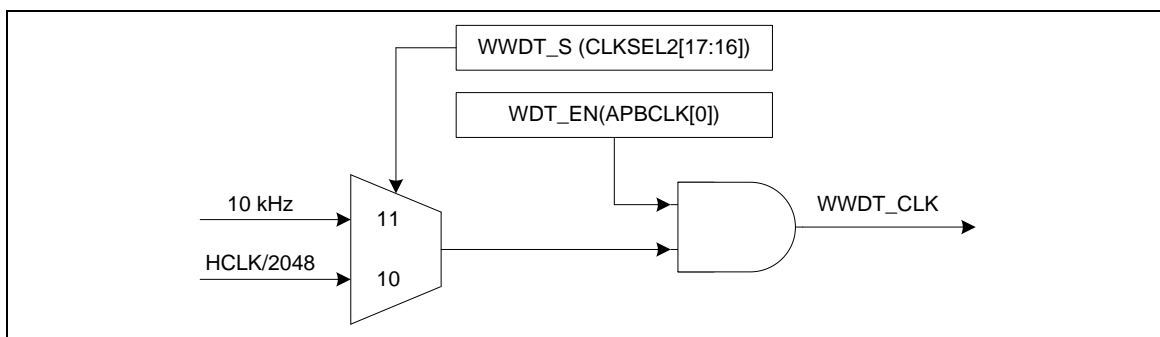


图 4-78 窗口看门狗定时器时钟控制

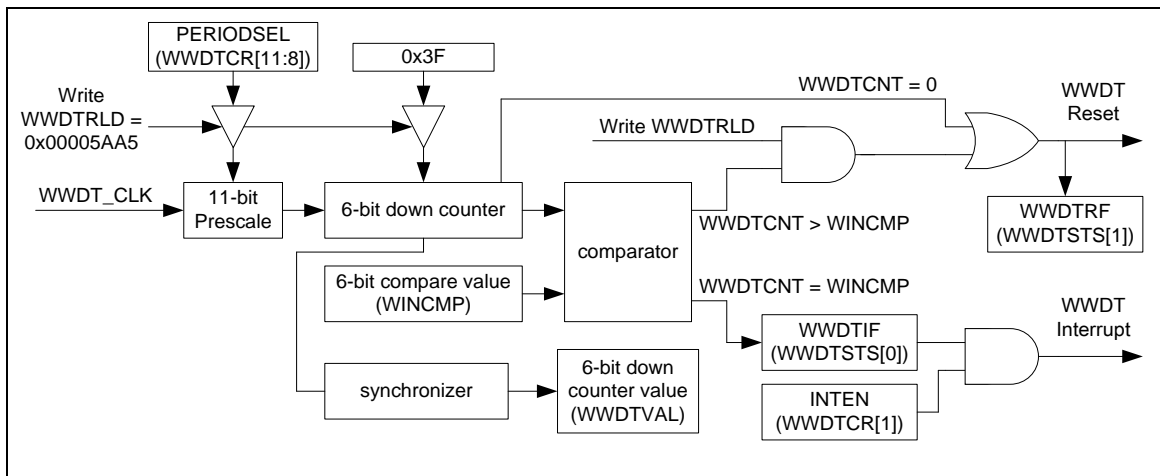


图 4-79 窗口看门狗定时器框图

### 4.13 UART 接口控制器(UART)

NuMicro™ NUC100 提供最多 3 个通用异步收发器 (UART) 通道。UART0 支持高速 UART，UART1~2 支持普通速度 UART，此外，只有 UART0 和 UART1 支持流控制功能。

#### 4.13.1 概述

通用异步收发器 (UART) 在从外设收到数据的时候执行串行到并行的转换，从 CPU 发送数据的时候执行并行到串行的转换。该串口控制器同时支持 IrDA SIR 功能，LIN 主机/从机模式功能和 RS-485 模式功能。每个 UART 通道支持 7 种类型的中断, 包括:

- 发送FIFO 空中断(INT\_THRE);
- 接收数据个数达到阈值设定长度中断(INT\_RDA);
- 线状态中断 (奇偶校验错误, 帧错误或者break 中断) (INT\_RLS);
- 接收缓存超时中断 (INT\_TOUT);
- MODEM/唤醒状态中断 (INT\_MODEM);
- 缓冲错误中断(INT\_BUF\_ERR);
- LIN 中断 (INT\_LIN).

## 4.14 PS/2 设备控制器 (PS2D)

### 4.14.1 概述

PS/2 设备控制器为 PS/2 通讯提供基本的时序控制。所有在设备和主机之间的通讯都是通过PS2\_CLK 和PS2\_DAT 管脚管理的。不同于 PS/2 键盘和鼠标设备控制器，接收/传送代码需要固件进行代码转换成有意义的代码。在接收到发送请求后设备控制器启动发送PS\_CLK 信号，但是在通信过程中主机拥有最终的控制权。主机发送到设备的数据 (DATA) 是在上升沿读取，设备发送到主机的数据在上升沿之后被改变.设备向主机发送数据。一个 16 字节的 FIFO 被应用于减少 CPU 的介入。S/W 可选择 1 ~ 16 字节的连续传输。

### 4.14.2 特征

- 主机通讯禁止和请求发送状态侦测
- 接收帧错误侦测
- 可编程的 1~16 字节传送缓存用以减少 CPU 干扰
- 双数据缓冲功能
- S/W override 总线

## 4.15 I<sup>2</sup>S 控制器 (I<sup>2</sup>S)

### 4.15.1 概述

I<sup>2</sup>S 控制器由 IIS 协议与外部音频 CODEC 接口组成。两个 8 字的 FIFO 分别用于读和写通道，可以处理 8-、16-、24- 和 32- 位字大小。PDMA 控制器处理 FIFO 和 内存之间的数据移动。

### 4.15.2 特征

- I<sup>2</sup>S 支持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 I<sup>2</sup>S 和 MSB 校正数据格式
- 提供两个 8 字的 FIFO 数据缓存，一个用于发送，一个用于接收
- 当缓存超出时，产生中断请求
- 两个 PDMA 请求，一个用于发送，一个用于接收



## 4.16 模拟数字转换(ADC)

### 4.16.1 概述

NuMicro™ NUC100系列包含一个12-位8通道逐次逼近式的模拟-数字转换器 (SAR A/D converter)。A/D转换器支持三种操作模式：单一 (single)，单周期扫描 (single-cycle scan) 和连续扫描模式 (continuous scan mode)。A/D转换器可由软件和外部 STADC 管脚启动转换。

### 4.16.2 特征

- 模拟输入电压范围：0~ $V_{REF}$
- 12-位分辨率和 10-位精确度保证
- 多达 8 路单端模拟输入通道或 4 路差分模拟输入通道
- 当ADC时钟频率是16 MHz，最大转换速率为760 kSPS (芯片工作在 5V)
- 三种操作模式
  - 单一模式：对指定的一个通道只进行一次A/D转换
  - 单周期扫描模式：对所有指定通道完成一次A/D转换，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
  - PWM 中心对齐触发
  - 外部管脚STADC
- 每个通道转换结果存储在数据寄存器内，并带有 valid/overrun 标志
- 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
- 通道 7 支持 3 路输入源：外部模拟电压，内部带隙电压和内部温度传感器输出

## 4.17 模拟比较器(ACMP)

### 4.17.1 概述

NuMicro™ NUC100 系列包含 2 路模拟比较器。该比较器可应用于不同的配置。当正极输入大于负极输入时，比较器输出为逻辑 1，反之为逻辑 0。当比较器输出值改变时，每一路比较器都可通过配置产生中断。该模块系统框图见錯誤! 找不到参照來源。

### 4.17.2 特征

- 模拟输入电压范围: 0~  $V_{DDA}$
- 支持迟滞功能
- 2 路模拟比较器，其负极可选内部参考电压输入

## 4.18 PDMA 控制器(PDMA)

### 4.18.1 概述

NuMicro™ NUC100 系列DMA包含九个通道外设直接存储器存取 (PDMA) 控制器和一个循环冗余检查(CRC)发生器。

PDMA可以传输数据到存储器和从存储器传输数据或传输数据到APB设备和从APB设备传输数据。PDMA (DMA CH0~CH8)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。软件可以停止PDMA通过禁止PDMA PDMA\_CSRx[PDMACEN]位。通过软件轮询或者收到内部的PDMA 中断, CPU 可以识别PDMA 运作的完成。PDMA 控制器可增加源和目的地址或者也能将其固定。

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。

### 4.18.2 特征

- 支持9个PDMA通道和一个CRC通道。每个通道能支持一个单向传输。
- AMBA AHB主机/从机接口兼容, 用于数据传输和寄存器读/写。
- 硬件通道优先级。DMA 通道 0 拥有最高优先级, 通道 8 拥有最低优先级。
- PDMA 操作
  - 外设到存储器、存储器到外设、存储器到存储器传输。
  - 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。
- 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CRC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, 支持下面数据宽度
    - 8-bit 写模式 (字节): 1-AHB时钟周期操作
    - 16-bit写模式 (半字): 2-AHB时钟周期操作。
  - - 32-bit 写模式 (字): 4-AHB时钟周期操作
  - 在CRC DMA模式下, 支持字节对齐传输长度和字对齐传输源地址

## 4.19 外部总线接口 (EBI)

#### 4.19.1 简介

NuMicro™ NUC100/NUC120 Low Density 系列LQFP-64 封装配备一个外部总线接口 (EBI) , 以供外部设备使用.

为节省外部设备与芯片的连接线, EBI支持地址总线与数据总线多路复用. 且地址锁存使能 (ALE)信号支持地址与数据周期的差别.

#### 4.19.2 特性

外部总线接口有下列功能:

- 支持外部设备最大64K-byte (8 位数据宽度)/128K-byte (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), 地址锁存使能时间(tALE) 和地址保持时间(tAHD)
- 支持地址总线和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(W2X), Read-to-Read (R2R)

## 4.20 FLASH 内存控制器 (FMC)

### 4.20.1 概述

NuMicro™ NUC100 系列配置了128/64/32K 字节的片上 Flash，用于应用程序内存 (APROM)，用户可通过 ISP 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 LDRROM 中读取代码。此外，NuMicro™ NUC100 系列还提供额外的数据 Flash用于用户在芯片断电之前存储一些应用所需的数据。对于 128K 字节的 APROM 设备，数据 flash与原始的 128K 程序内存共享，其起始地址是可配置的，由用户在 Config1 中定义。对于 64K/32K 字节的 APROM 设备，数据 flash 固定为 4K。

### 4.20.2 特征

- 零等待状态，可达50 MHz的连续的地址读访问
- 所有嵌入的flash 存储器 支持 512字节页擦除单元
- 128/64/32 KB应用程序内存(APROM)
- 4KB 在系统编程 (ISP) 加载程序内存 (LDRROM)
- 对于 64/32 KB APROM 设备，固定4KB数据flash
- 对于128KB APROM 设备，可配置数据flash的大小
- 可配置或固定 4KB 的数据 flash，带有 512 字节页擦除单元
- 支持在应用编程 (IAP)，在APROM 和 LDRROM之间切换代码，不需要复位
- 在系统编程 (ISP) 用于更新片上 Flash

## 5 电气特性

### 5.1 绝对最大额定值

参数	符号	最小值	最大值	单位
直流电源电压	$V_{DD}-V_{SS}$	-0.3	+7.0	V
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	TA	-40	+85	°C
贮存温度	TST	-55	+150	°C
VDD 最大流入电流		-	120	mA
VSS 最大流出电流			120	mA
单一 I/O 管脚最大灌电流			35	mA
单一 I/O 管脚最大拉电流			35	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

注：上表所列的条件中，其极限值可能对器件的提升和稳定有反作用。

## 5.2 DC 电气特性

( $V_{DD}-V_{SS}=5.5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ ,  $F_{OSC} = 50\text{ MHz}$ 除非其他特别说明.)

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	$V_{DD}$	2.5		5.5	V	$V_{DD} = 2.5\text{V} \sim 5.5\text{V}$ up to 50 MHz
电源地	$V_{SS}$ $AV_{SS}$	-0.3			V	
LDO 输出电压	$V_{LDO}$	1.62	1.8	1.98	V	$V_{DD} > 2.5\text{V}$
模拟工作电压	$AV_{DD}$		$V_{DD}$		V	当系统使用模拟功能, 请参考6.4节对应的模拟操作电压。
Operating Current Normal Run Mode at 50 MHz	$I_{DD1}$		34		mA	$V_{DD} = 5.5\text{V}$ , 所有 IP 和 PLL 使能, XTAL = 12 MHz
	$I_{DD2}$		15		mA	$V_{DD} = 5.5\text{V}$ , All IP disabled and PLL enabled, XTAL = 12 MHz
	$I_{DD3}$		32		mA	$V_{DD} = 3.3\text{V}$ , All IP and PLL enabled, XTAL = 12 MHz
	$I_{DD4}$		14		mA	$V_{DD} = 3.3\text{V}$ , All IP disabled and PLL enabled, XTAL = 12 MHz
Operating Current Normal Run Mode at 12 MHz	$I_{DD5}$		8.5		mA	$V_{DD} = 5.5\text{V}$ , All IP enabled and PLL disabled, XTAL = 12 MHz
	$I_{DD6}$		3.6		mA	$V_{DD} = 5.5\text{V}$ , All IP and PLL disabled, XTAL = 12 MHz
	$I_{DD7}$		7.5		mA	$V_{DD} = 3.3\text{V}$ , All IP enabled and PLL disabled, XTAL = 12 MHz
	$I_{DD8}$		2.6		mA	$V_{DD} = 3.3\text{V}$ , All IP and PLL disabled, XTAL = 12 MHz
普通模式下的工作电流 @ 4 MHz	$I_{DD9}$		3.6		mA	$V_{DD} = 5.5\text{V}$ , All IP enabled and PLL disabled, XTAL = 4 MHz

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
	I <sub>DD10</sub>		2		mA	V <sub>DD</sub> = 5.5V, All IP and PLL disabled, XTAL = 4 MHz
	I <sub>DD11</sub>		2.8		mA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL disabled, XTAL = 4 MHz
	I <sub>DD12</sub>		1.2		mA	V <sub>DD</sub> = 3.3V, All IP and PLL disabled, XTAL = 4 MHz
普通模式下的工作电流 @ 32.768 kHz	I <sub>DD13</sub>		141		μA	V <sub>DD</sub> = 5.5V, All IP enabled and PLL disabled, XTAL = 32.768 kHz
	I <sub>DD14</sub>		129		μA	V <sub>DD</sub> = 5.5V, All IP and PLL disabled, XTAL = 32.768 kHz
	I <sub>DD15</sub>		138		μA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL disabled, XTAL = 32.768 kHz
	I <sub>DD16</sub>		125		μA	V <sub>DD</sub> = 3.3V, All IP and PLL disabled, XTAL = 32.768 kHz
普通模式下的工作电流 @10 kHz	I <sub>DD17</sub>		125		μA	V <sub>DD</sub> = 5.5V, All IP enabled and PLL disabled, LIRC10 kHz enabled
	I <sub>DD18</sub>		120		μA	V <sub>DD</sub> = 5.5V, All IP and PLL disabled, LIRC10 kHz enabled
	I <sub>DD19</sub>		125		μA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL disabled, LIRC10 kHz enabled
	I <sub>DD20</sub>		120		μA	V <sub>DD</sub> = 3.3V, All IP and PLL disabled, LIRC10kHz enabled
空闲模式下的工作电流 @50 MHz	I <sub>IDLE1</sub>		28		mA	V <sub>DD</sub> = 5.5V, All IP and PLL enabled, XTAL = 12 MHz
	I <sub>IDLE2</sub>		10		mA	V <sub>DD</sub> = 5.5V, All IP disabled and PLL enabled, XTAL = 12 MHz



参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
	I <sub>IDLE3</sub>		27		mA	V <sub>DD</sub> = 3.3V, All IP and PLL enabled, XTAL = 12 MHz
	I <sub>IDLE4</sub>		9		mA	V <sub>DD</sub> = 3.3V All IP disabled and PLL enabled, XTAL = 12 MHz
空闲模式下的工作电流 @ 12 MHz	I <sub>IDLE5</sub>		7.5		mA	V <sub>DD</sub> = 5.5V, All IP enabled and PLL disabled, XTAL = 12 MHz
	I <sub>IDLE6</sub>		2.4		mA	V <sub>DD</sub> = 5.5V, All IP and PLL disabled, XTAL = 12 MHz
	I <sub>IDLE7</sub>		6.5		mA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL enabled, XTAL = 12 MHz
	I <sub>IDLE8</sub>		1.5		mA	V <sub>DD</sub> = 3.3V, All IP and PLL disabled, XTAL = 12 MHz
空闲模式下的工作电流 @ 4 MHz	I <sub>IDLE9</sub>		3.3		mA	V <sub>DD</sub> = 5.5V, All IP enabled and PLL disabled, XTAL = 4 MHz
	I <sub>IDLE10</sub>		1.7		mA	V <sub>DD</sub> = 5.5V All IP and PLL disabled, XTAL = 4 MHz
	I <sub>IDLE11</sub>		2.4		mA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL disabled, XTAL = 4 MHz
	I <sub>IDLE12</sub>		0.8		mA	V <sub>DD</sub> = 3.3V, All IP and PLL disabled, XTAL = 4 MHz
空闲模式下的工作电流 @ 32.768 kHz	I <sub>IDLE13</sub>		133		μA	V <sub>DD</sub> = 5.5V, All IP enabled and PLL disabled, XTAL = 32.768 kHz
	I <sub>IDLE14</sub>		120		μA	V <sub>DD</sub> = 5.5V, All IP and PLL disabled, XTAL = 32.768 kHz
	I <sub>IDLE15</sub>		133		μA	V <sub>DD</sub> = 3.3V, All IP enabled and PLL disabled, XTAL = 32.768 kHz

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
	$I_{IDLE16}$		120		$\mu A$	$V_{DD} = 3.3V$ , All IP and PLL disabled, XTAL = 32.768 kHz
空闲模式下的工作电流 @ 10 kHz	$I_{IDLE13}$		122		$\mu A$	$V_{DD} = 5.5V$ , All IP enabled and PLL disabled, LIRC10 kHz enabled
	$I_{IDLE14}$		118		$\mu A$	$V_{DD} = 5.5V$ , All IP and PLL disabled, LIRC10 kHz enabled
	$I_{IDLE15}$		122		$\mu A$	$V_{DD} = 3.3V$ , All IP enabled and PLL disabled, LIRC10 kHz enabled
	$I_{IDLE16}$		118		$\mu A$	$V_{DD} = 3.3V$ All IP and PLL disabled, LIRC10 kHz enabled
掉电模式下的Standby电流 (深度睡眠模式)	$I_{PWD1}$		15		$\mu A$	$V_{DD} = 5.5V$ , RTC disabled, When BOD function disabled
	$I_{PWD2}$		15		$\mu A$	$V_{DD} = 3.3V$ , RTC disabled, When BOD function disabled
	$I_{PWD3}$		17		$\mu A$	$V_{DD} = 5.5V$ , RTC enabled , When BOD function disabled
	$I_{PWD4}$		17		$\mu A$	$V_{DD} = 3.3V$ , RTC enabled , When BOD function disabled
PA, PB, PC, PD, PE, PF 输入电流 (准双向模式)	$I_{IN1}$		-50	-60	$\mu A$	$V_{DD} = 5.5V$ , $V_{IN} = 0V$ or $V_{IN} = V_{DD}$
nRESET <sup>[1]</sup> 管脚输入电流	$I_{IN2}$	-55	-45	-30	$\mu A$	$V_{DD} = 3.3V$ , $V_{IN} = 0.45V$
PA, PB, PC, PD, PE, PF输入漏电流	$I_{LK}$	-2	-	+2	$\mu A$	$V_{DD} = 5.5V$ , $0 < V_{IN} < V_{DD}$
PA~PF 逻辑 1至 0 转换时的电流 (准双向模式)	$I_{TL}^{[3]}$	-650	-	-200	$\mu A$	$V_{DD} = 5.5V$ , $V_{IN} < 2.0V$
PA, PB, PC, PD, PE, PF输入低电压 (TTL 输入)	$V_{IL1}$	-0.3	-	0.8	V	$V_{DD} = 4.5V$
		-0.3	-	0.6		$V_{DD} = 2.5V$
PA, PB, PC, PD, PE, PF输入高电压 (TTL 输入)	$V_{IH1}$	2.0	-	$V_{DD} + 0.2$	V	$V_{DD} = 5.5V$
		1.5	-	$V_{DD} + 0.2$		$V_{DD} = 3.0V$
PA, PB, PC, PD, PE, PF输入低电压 (Schmitt 输入)	$V_{IL2}$	-0.3	-	$0.3V_{DD}$	V	
PA, PB, PC, PD, PE, PF输入高电压 (Schmitt 输入)	$V_{IH2}$	$0.7V_{DD}$	-	$V_{DD} + 0.2$	V	

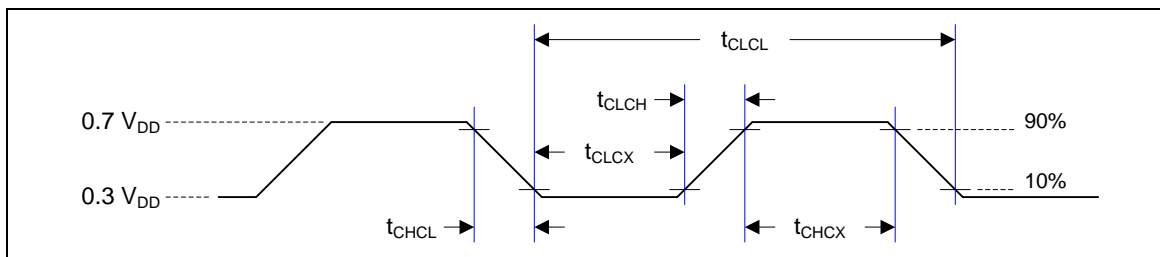
参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
PA, PB, PC, PD, PE, PF 迟滞电压 (Schmitt 输入)	$V_{HY}$		0.2V <sub>DD</sub>		V	
XT1_IN <sup>[2]</sup> 管脚输入低电压	$V_{IL3}$	0	-	0.8	V	V <sub>DD</sub> = 4.5V
		0	-	0.4		V <sub>DD</sub> = 3.0V
XT1_IN <sup>[2]</sup> 管脚输入高电压	$V_{IH3}$	3.5	-	V <sub>DD</sub> + 0.2	V	V <sub>DD</sub> = 5.5V
		2.4	-	V <sub>DD</sub> + 0.2		V <sub>DD</sub> = 3.0V
X32_IN <sup>[2]</sup> 管脚输入低电压	$V_{IL4}$	0	-	0.4	v	
X32_IN <sup>[2]</sup> 管脚输入高电压	$V_{IH4}$	1.2		1.8	V	
/RESET 管脚负向阈值电压 (Schmitt 输入)	$V_{ILS}$	-0.5	-	0.2V <sub>DD</sub> - 0.2	V	
/RESET 管脚正向阈值电压 (Schmitt 输入)	$V_{IHS}$	0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V	
PA, PB, PC, PD, PE, PF 拉电流 (准双向模式)	$I_{SR11}$	-300	-370	-450	μA	V <sub>DD</sub> = 4.5V, V <sub>S</sub> = 2.4V
	$I_{SR12}$	-50	-70	-90	μA	V <sub>DD</sub> = 2.7V, V <sub>S</sub> = 2.2V
	$I_{SR12}$	-40	-60	-80	μA	V <sub>DD</sub> = 2.5V, V <sub>S</sub> = 2.0V
PA, PB, PC, PD, PE, PF 拉电流 (推挽模式)	$I_{SR21}$	-24	-28	-32	mA	V <sub>DD</sub> = 4.5V, V <sub>S</sub> = 2.4V
	$I_{SR22}$	-4	-6	-8	mA	V <sub>DD</sub> = 2.7V, V <sub>S</sub> = 2.2V
	$I_{SR22}$	-3	-5	-7	mA	V <sub>DD</sub> = 2.5V, V <sub>S</sub> = 2.0V
PA, PB, PC, PD, PE, PF 灌电流 (准双向和推挽模式)	$I_{SK1}$	10	16	20	mA	V <sub>DD</sub> = 4.5V, V <sub>S</sub> = 0.45V
	$I_{SK1}$	7	10	13	mA	V <sub>DD</sub> = 2.7V, V <sub>S</sub> = 0.45V
	$I_{SK1}$	6	9	12	mA	V <sub>DD</sub> = 2.5V, V <sub>S</sub> = 0.45V
BOV_VL [1:0] = 00b 的欠压电压	$V_{BO2.2}$	2.1	2.2	2.3	V	
BOV_VL [1:0] = 01b 的欠压电压	$V_{BO2.7}$	2.6	2.7	2.8	V	
BOV_VL [1:0] = 10b 的欠压电压	$V_{BO3.8}$	3.5	3.7	3.9	V	
BOV_VL [1:0] = 11b 的欠压电压	$V_{BO4.5}$	4.2	4.4	4.6	V	
BOD 电压的迟滞范围	$V_{BH}$	30	-	150	mV	V <sub>DD</sub> = 2.5V~5.5V

注:

1. /RESET 管脚为 Schmitt 触发输入。
2. 晶振输入为 CMOS 输入。
3. 管脚 PA, PB, PC, PD, PE 及 PF 被外部由 1 拉成 0 时可以提供一个输出电流, 当 V<sub>DD</sub> = 5.5 V 时电流最大, 这时 V<sub>IN</sub> 接近 2V。

### 5.3 AC 电气特性

5.3.1 外部 4~24 MHz 高速振荡器



注：占空比为 50%.

符号	参数	条件	最小值	典型值	最大值	单位
t <sub>CHCX</sub>	时钟高电平时间		10	-	-	nS
t <sub>CLCX</sub>	时钟低电平时间		10	-	-	nS
t <sub>CLCH</sub>	时钟上升沿时间		2	-	15	nS
t <sub>CHCL</sub>	时钟下降沿时间		2	-	15	nS

5.3.2 外部 4~24 MHz 高速晶振

参数	条件	最小值	典型值	最大值	单位
操作电压 V <sub>DD</sub>	-	2.5	-	5.5	V
温度	-	-40	-	85	°C
操作电流	12 MHz at V <sub>DD</sub> = 5V	-	1	-	mA
输入时钟频率	外部晶振	4		24	MHz

5.3.2.1 典型晶振应用电路

晶振	C1	C2	R
4 MHz ~ 24 MHz	10~20pF	10~20pF	不需要

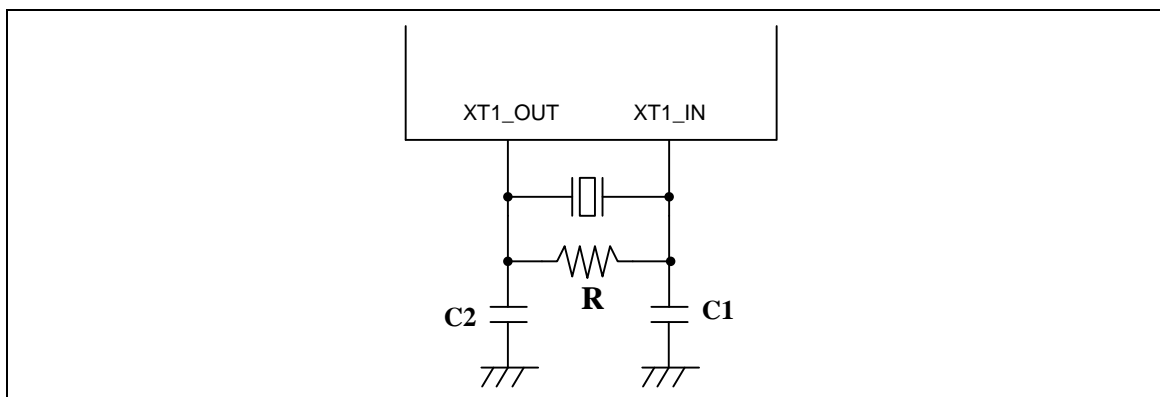


图 5-1 典型晶振应用电路

5.3.3 外部 32.768 kHz 低速晶振

参数	条件	最小值	典型值	最大值	单位
操作电压V <sub>DD</sub>	-	2.5	-	5.5	V
温度	-	-40	-	85	°C
操作电流	32.768KHz at V <sub>DD</sub> =5V		1.5		μA
输入时钟频率	外部晶振	-	32.768	-	kHz

5.3.4 内部 22.1184 MHz 高速振荡器

参数	条件	最小值	典型值	最大值	单位
工作电压V <sub>DD</sub>	-	2.5	-	5.5	V
中心频率	-	-	22.1184	-	MHz
校验内部振荡器频率	+25°C; V <sub>DD</sub> =5 V	-1	-	+1	%
	-40°C~+85°C; V <sub>DD</sub> =2.5 V~5.5 V	-3	-	+3	%
工作电流	V <sub>DD</sub> =5 V	-	500	-	uA

5.3.5 内部 10 kHz 低速振荡器

参数	条件	最小值	典型值	最大值	单位
工作电压V <sub>DD</sub>	-	2.5	-	5.5	V
中心频率	-	-	10	-	kHz
校验内部振荡器频率	+25°C; V <sub>DD</sub> =5 V	-30	-	+30	%
	-40°C~+85°C; V <sub>DD</sub> =2.5 V~5.5 V	-50	-	+50	%

5.4 模拟量特性

5.4.1 12-bit SARADC 规格

符号	参数	最小值	典型值	最大值	单位
-	分辨率	-	-	12	Bit
DNL	非线性差分误差	-	-1~2	-1~4	LSB
INL	非线性积分误差	-	±2	±4	LSB
EO	补偿误差	-	±1	10	LSB

EG	增益误差 (传输增益)	-	1	1.005	-
-	一致性	Guaranteed			
F <sub>ADC</sub>	ADC时钟频率(AV <sub>DD</sub> = 5V/3V)	-	-	16/8	MHz
F <sub>S</sub>	采样率	-	-	760	kSPS
V <sub>DDA</sub>	工作电压	3	-	5.5	V
I <sub>DD</sub>	工作电流 (平均)	-	0.5	-	mA
I <sub>DDA</sub>		-	1.5	-	mA
V <sub>REF</sub>	参考电压	3	-	V <sub>DDA</sub>	V
I <sub>REF</sub>	参考电流 (平均)	-	1	-	mA
V <sub>IN</sub>	输入电压	0	-	V <sub>REF</sub>	V

#### 5.4.2 LDO 规格和电源管理

参数	最小值	典型值	最大值	单位	备注
输入电压	2.5		5.5	V	V <sub>DD</sub> input voltage
输出电压	1.62	1.8	1.98	V	V <sub>DD</sub> > 2.5 V
温度	-40	25	85	°C	
Cbp	-	1	-	μF	R <sub>ESR</sub> = 1 Ω

注:

1. 建议接一颗 10uF 或更大的电容和一颗 100nF 旁路电容在 VDD 与 VSS 之间。
2. 为保证电源稳定, 要在 LDO 与 VSS 之间接一颗 1uF 或更大的电容。

5.4.3 低压复位说明

参数	条件	最小值	典型值	最大值	单位
工作电压	-	0	-	5.5	V
静态电流	V <sub>DD</sub> =5.5 V	-	1	5	μA
温度	-	-40	25	85	℃
阈值电压	温度=-40~85℃	1.7	2.0	2.3	V
迟滞	-	0	0	0	V

5.4.4 欠压检测说明

参数	条件	最小值	典型值	最大值	单位
工作电压	-	0	-	5.5	V
静态温度	-	-40	25	85	℃
静态电流	A <sub>VDD</sub> =5.5 V	-	-	125	μA
欠压电压	BOD_VL[1:0]=11	4.2	4.4	4.6	V
	BOD_VL [1:0]=10	3.5	3.7	3.9	V
	BOD_VL [1:0]=01	2.6	2.7	2.8	V
	BOD_VL [1:0]=00	2.1	2.2	2.3	V
迟滞	-	30	-	150	mV

5.4.5 上电复位说明

参数	条件	最小值	典型值	最大值	单位
温度	-	-40	25	85	℃
复位电压	V+	-	2	-	V
静态电流	V <sub>in</sub> > reset voltage	-	1	-	nA

5.4.6 温度传感器说明

参数	条件	最小值	典型值	最大值	单位
电源 <sup>[1]</sup>		2.5	-	5.5	V
温度		-40	-	85	°C
电流消耗		6.4	-	10.5	μA
增益			-1.76		mV/°C
偏移量	Temp=0 °C		720		mV

注：内部工作电压来自 LDO。

5.4.7 比较器说明

参数	条件	最小值	典型值	最大值	单位
工作电压 AV <sub>DD</sub>	-	2.5		5.5	V
工作温度	-	-40	25	85	°C
工作电流	V <sub>DD</sub> =3.0 V	-	20	40	μA
输入偏移量电压	-	-	5	15	mV
输出漂移	-	0.1	-	V <sub>DDA</sub> -0.1	V
共模输入范围	-	0.1	-	V <sub>DDA</sub> -0.1	V
DC 增益	-	-	70	-	dB
传播延时	V <sub>CM</sub> = 1.2 V and V <sub>DIFF</sub> = 0.1 V	-	200	-	ns
比较电压	20 mV at V <sub>CM</sub> =1 V 50 mV at V <sub>CM</sub> =0.1 V 50 mV at V <sub>CM</sub> =V <sub>DD</sub> -1.2 10 mV for non-hysteresis	10	20	-	mV
迟滞	V <sub>CM</sub> =0.4 V ~ V <sub>DD</sub> -1.2 V	-	±10	-	mV
稳定时间	C <sub>INP</sub> = 1.3 V C <sub>INN</sub> = 1.2 V	-	-	2	μs



### 5.4.8 USB PHY 说明

#### 5.4.8.1 USB DC电气特性

符号	参数	条件	最小值	典型值	最大值	单位
V <sub>IH</sub>	输入高 (driven)		2.0			V
V <sub>IL</sub>	输入低				0.8	V
V <sub>DI</sub>	差分输入灵敏度	PADP-PADM	0.2			V
V <sub>CM</sub>	差分共模范围	Includes V <sub>DI</sub> range	0.8		2.5	V
V <sub>SE</sub>	单端接收器阈值		0.8		2.0	V
	接收器迟滞			200		mV
V <sub>OL</sub>	输出低 (driven)		0		0.3	V
V <sub>OH</sub>	输出高 (driven)		2.8		3.6	V
V <sub>CRS</sub>	输出信号串扰电压		1.3		2.0	V
R <sub>PU</sub>	上拉电阻		1.425		1.575	kΩ
V <sub>TRM</sub>	上行端口上的上拉电阻的极限电压(RPU)		3.0		3.6	V
Z <sub>DRV</sub>	驱动输出阻抗	稳态驱动*		10		Ω
C <sub>IN</sub>	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

#### 5.4.8.2 USB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 <sub>FR</sub>	上升时间	C <sub>L</sub> =50p	4		20	ns
T <sub>FF</sub>	下降时间	C <sub>L</sub> =50p	4		20	ns
T <sub>FRFF</sub>	上升和下降时间比值	T <sub>FRFF</sub> =T <sub>FR</sub> /T <sub>FF</sub>	90		111.11	%

#### 5.4.8.3 USB电源功耗

符号	参数	条件	最小值	典型值	最大值	单位
I <sub>VBUS</sub>	USB_VBUS 电流 (稳态)	Standby		50		μA

5.4.8.4 USB LDO 规格

符号	参数	条件	最小值	典型值	最大值	单位
USB_VBUS	USB_VBUS 管脚输入电压		4.0	5.0	5.5	V
USB_VDD33_C AP	LDO 输出电压		3.0	3.3	3.6	V
C <sub>bp</sub>	外部旁路电容			1.0	-	uF

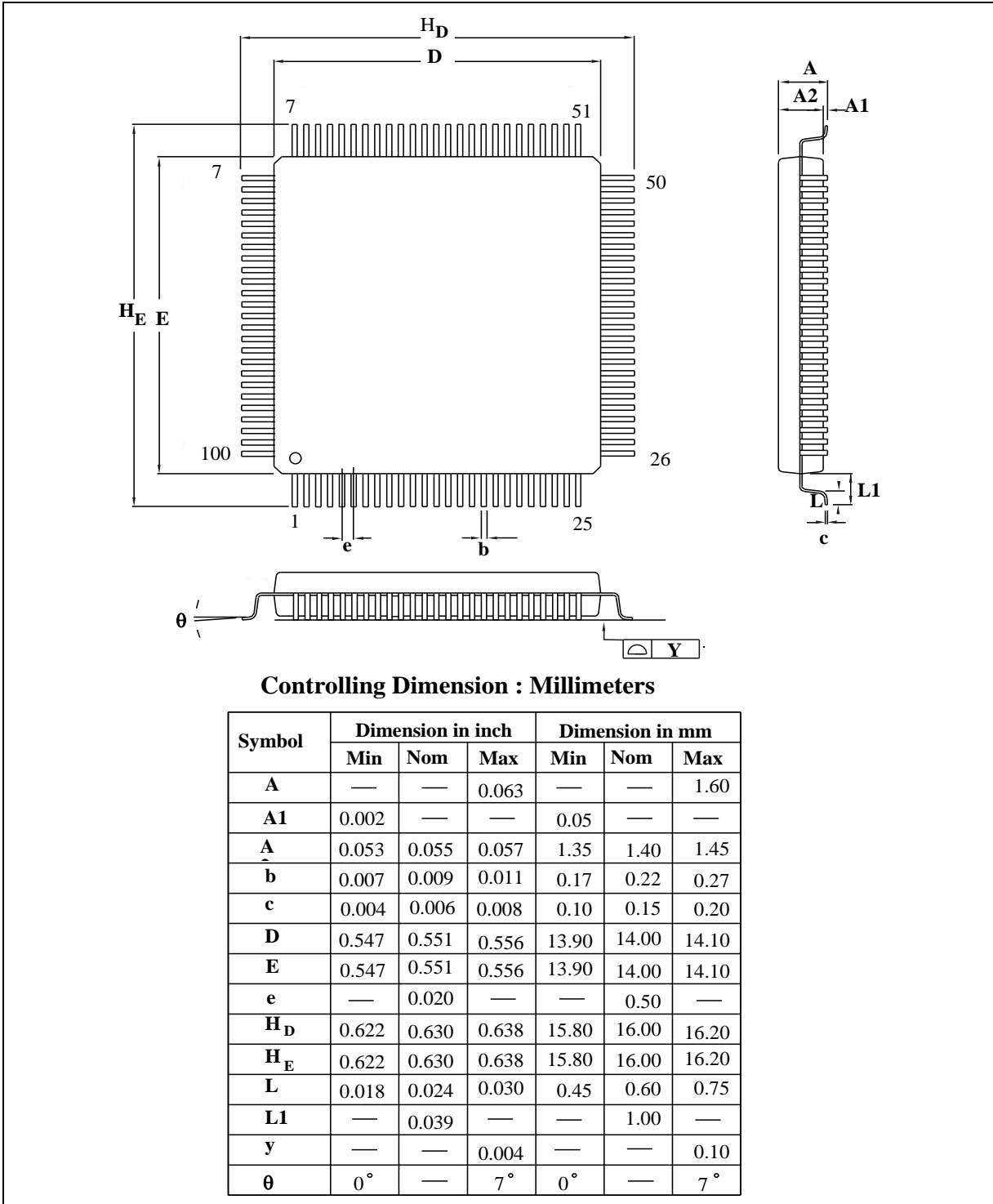
5.5 Flash DC 电气特性

符号	参数	条件	最小值	典型值	最大值	单位
V <sub>DD</sub>	电源		1.62	1.8	1.98	V <sup>[2]</sup>
T <sub>RET</sub>	数据保存	At 85°C	10			year
T <sub>ERASE</sub>	页擦除时间			2		ms
T <sub>MER</sub>	批量擦除时间			10		ms
T <sub>PROG</sub>	编程时间			20		μs
I <sub>DD1</sub>	读电流		-	0.15	0.5	mA/MHz
I <sub>DD2</sub>	编程/擦除 电流				7	mA
I <sub>PD</sub>	掉电电流		-	1	20	μA

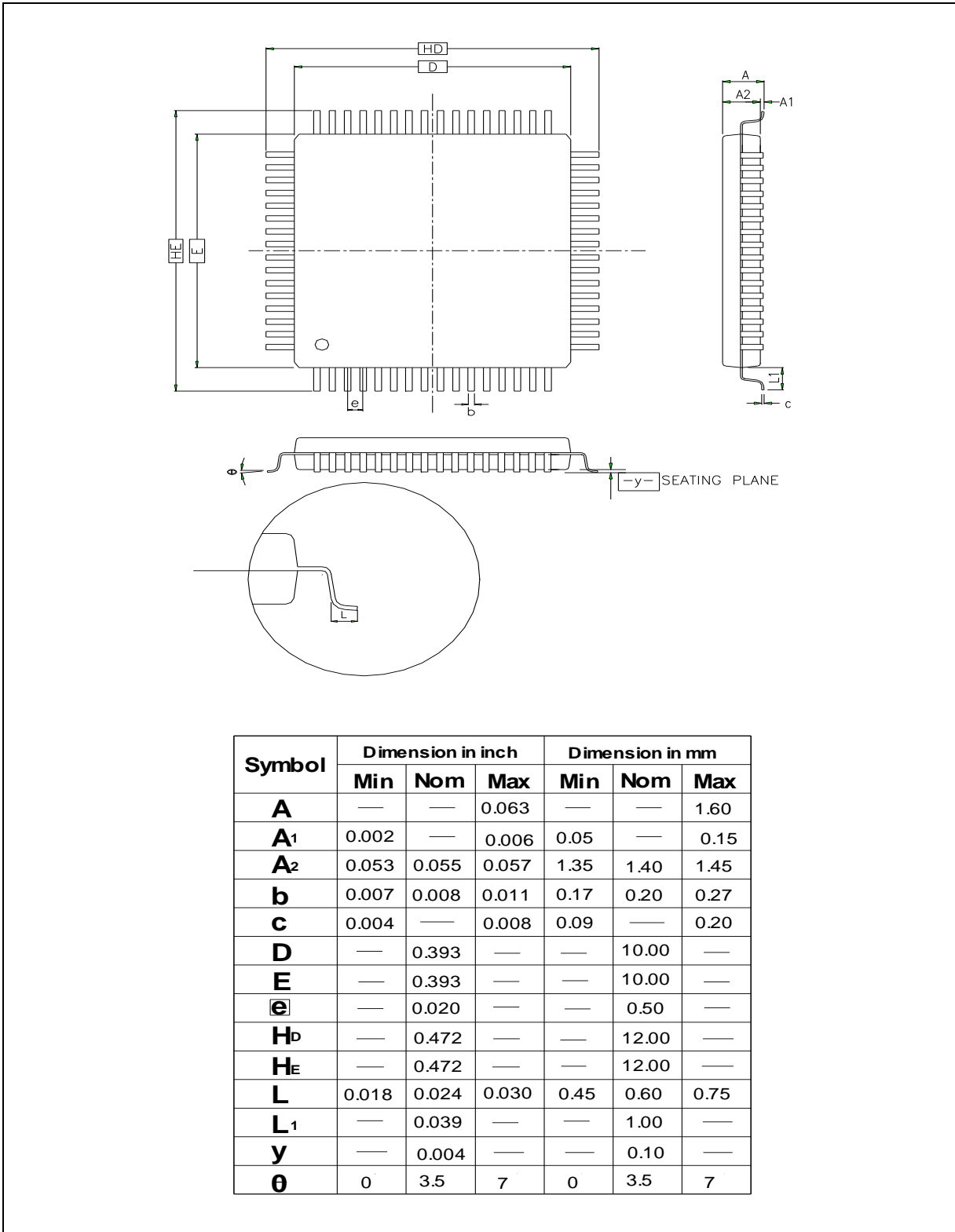
1. V<sub>DD</sub> 来自芯片 LDO 输出电压.
2. 该表是设计保证，产品中没有测量

6 封装定义

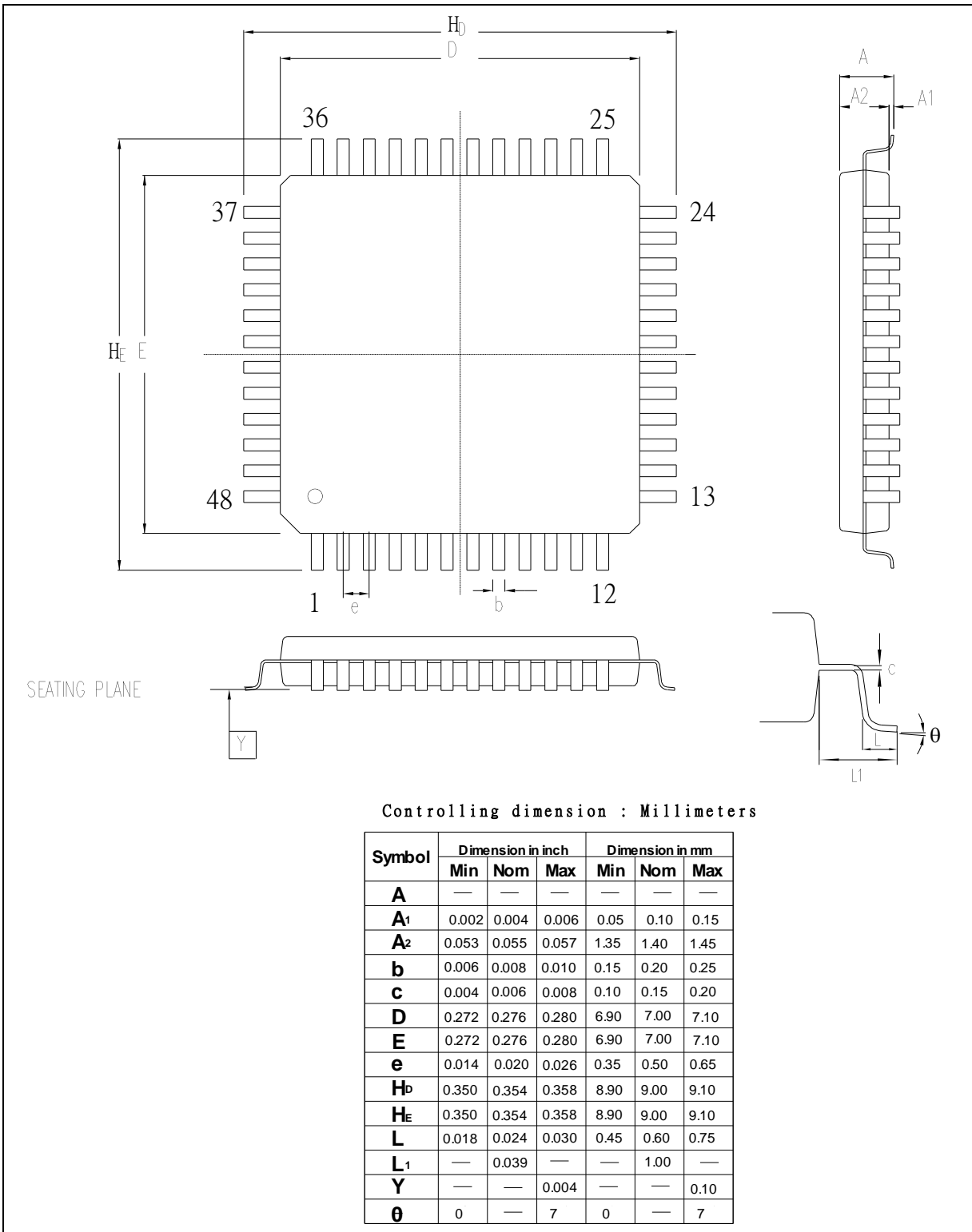
6.1 100-pin LQFP (14x14x1.4 mm footprint 2.0 mm)



6.2 64-pin LQFP (10x10x1.4 mm footprint 2.0 mm)



6.3 48-pin LQFP (7x7x1.4 mm footprint 2.0 mm)



## 7 版本历史

版本	日期	描述
V1.00	Aug. 24, 2014	1. 第一版
V1.01	Jul. 1, 2015	1. 修正章节6.2 封装大小为10x10mm
V1.02	Mar. 02 2017	<ol style="list-style-type: none"> <li>1. 更新章节2.3 NuMicro™ NUC100/120xxxDN产品选型指南</li> <li>2. 更新章节5.4.3 低压复位说明.</li> <li>3. 更新章节5.4.7 比较器说明.</li> </ol>

### 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.*