
N32H7xx系列CORDIC使用指南

简介

此文档的目的在于让使用者能够快速熟悉 N32H7xx 系列微控制器（MCU）的 CORDIC 外设基本功能及使用方法，用于指导用户更容易的了解和使用本产品。

目录

目录I

1. 概述	2
2. CORDIC 外设介绍	3
2.1. CORDIC 简介	3
2.2. CORDIC 功能框图	3
3. CORDIC 配置	4
3.1. CORDIC 配置说明	4
3.2. CORDIC 初始化	5
3.3. CORDIC 中断与 DMA	6
4. CORDIC 工作模式	7
5. 版本历史	8
6. 声明	9

1. 概述

CORDIC 通过硬件进行加法+移位的迭代操作，替代常用数学函数复杂的软件运算。每个时钟周期可进行 4 次迭代，大大节约了时间，同时提供可接受的数据精度。

本文档介绍 N32H7xx 系列微控制器（MCU）CORDIC 模块的基本使用方法，详细功能说明、参数、注意事项等等请参照用户手册 CN_UM_N32H7xx Series User Manual。

2. CORDIC 外设介绍

2.1. CORDIC 简介

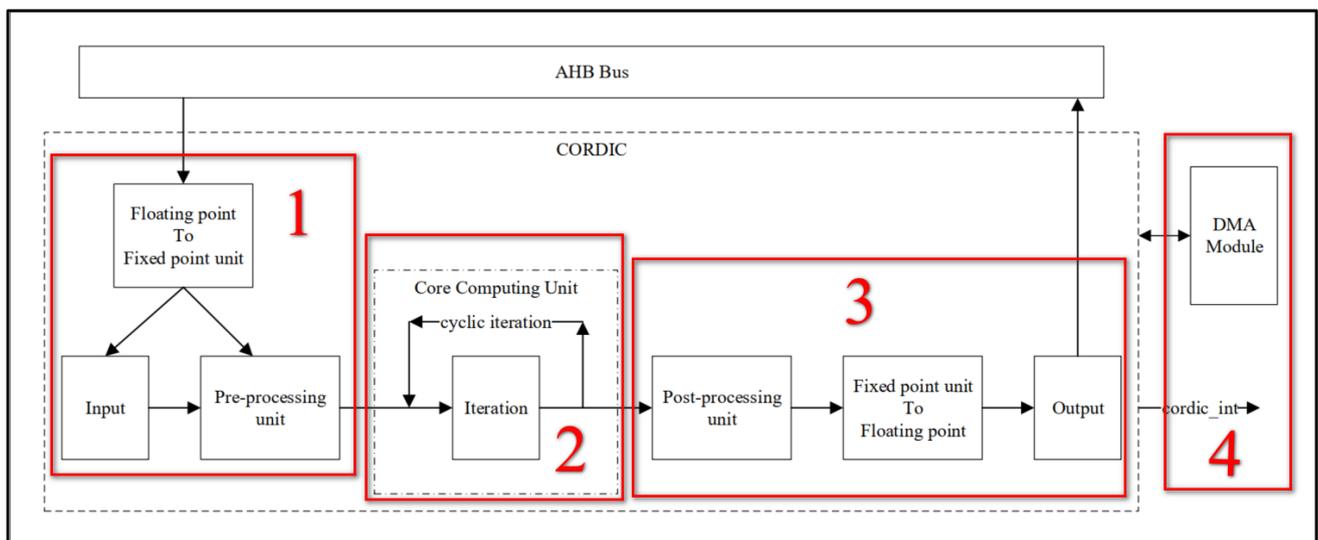
CORDIC 支持对以下 10 种常用数学函数进行加速运算：

sin, cosine, sinh, cosh, atan, atan2, atanh, modulus, square root, natural logarithm。

初始化后，向 WDAT 寄存器写入符合要求的参数（格式与数据范围），CORDIC 自动开始转换。转换完成后，直接从 RDAT 寄存器获取结果。

2.2. CORDIC 功能框图

CORDIC 模块结构框图如下图所示，大致可分为 4 个部分：



各部分功能如下：

- 1) 数据输入：包括输入数据转换单元与预处理单元。其中数据转换单元可将输入浮点数据（单精度或半精度）转换成定点数据（q1.31 或 q1.15）。
- 2) 计算核心单元：将输入数据进行迭代计算得到计算结果，精度由可配置的迭代次数决定。
- 3) 数据输出：包括数据后处理单元与输出数据转换单元。其中输出数据转换单元可将输出的定点数据（q1.31 或 q1.15）转换成浮点数据（单精度或半精度）。
- 4) 中断与 DMA：中断请求输出与 DMA 读写接口。

3. CORDIC 配置

3.1. CORDIC 配置说明

CORDIC 配置通过操作控制状态寄存器 CORDIC_CTRLSTS 实现，主要配置项如下：

- 1) 输入数据格式：
 - 可配置为定点或浮点格式。
 - 输入数据位宽可配置为 32-bit 或 16-bit。
 - 每次计算前输入参数时，需要写数据寄存器 WDAT 的次数，可配置为 1 次或 2 次，比如：
 - 2 次写入+32-bit 位宽：需要 1 个 32-bit 主参数和 1 个 32-bit 副参数，依次写入数据寄存器。
 - 1 次写入+32-bit 位宽：仅需写入 1 个 32-bit 主参数，副参数保持默认值 1 或上一次的设定值。
 - 1 次写入+16-bit 位宽：需要 1 个由主参数和副参数组合而成的 32-bit 数据，一次写入寄存器，其中低 16 bits 为主参数。
 - 2 次写入+16-bit 位宽：第 1 次写入由主参数和副参数组合而成的 32-bit 数据，其中低 16 bits 为主参数。第 2 次写入数据被忽略。
- 2) 输出数据格式
 - 可配置为定点或浮点格式。
 - 输出数据位宽可配置为 32-bit 或 16-bit。
 - 计算完需要读取结果寄存器 RDAT 的次数，可配置为 1 次或 2 次，比如：
 - 2 次读取+32-bit 位宽：第 1 次读取得到 32-bit 主结果，第 2 次读取得到 32-bit 副结果。
 - 1 次读取+32-bit 位宽：仅 1 次读取，得到 32-bit 主结果。
 - 1 次读取+16-bit 位宽：仅 1 次读取，获取 32-bits 结果，其中：低 16 位为主结果，高 16 位为副结果。
 - 2 次读取+16-bit 位宽：第 1 次读取得到一个 32-bit 结果，其中：低 16 位为主结果，高 16 位保持上一次的数据。第 2 次读取得到另一个 32-bit 结果，其中：低 16 位为主结果，高 16 位为副结果。
- 3) 函数功能：在 10 种函数中选择其中一种。
- 4) 精度：迭代次数，建议在 12、16、20、24 次迭代中选择。
- 5) 缩放因子 SCALE：

部分函数支持对结果进行放大（*2^{SCALE}），但需要在写入寄存器前对输入参数进行缩小（/2^{SCALE}）。不同函数对缩放因子要求不同，实际输入、输出数据范围也有所不同，详见用户手册。
- 6) DMA：读写均支持 DMA，可分别配置。
- 7) 支持 2 种中断，共用 1 个中断入口：
 - 输入参数溢出。
 - 计算结果就绪。
- 8) 限幅：超出上限的以上限值替换，低于下限的以下限值替换。
 - 坐标输出结果限幅。
 - 相位输出结果限幅。

3.2. CORDIC 初始化

以下对 CORDIC 初始化结构体进行补充说明：

```
typedef struct
{
    uint32_t Function;    /* Select function */
    uint32_t Precision;  /* Set the number of iterations */
    uint32_t Scale;      /* Select Scaling Factor */
    uint32_t NbWrite;    /* Select the number of CORDIC_WDAT register to write */
    uint32_t NbRead;     /* Select the number of CORDIC_RDAT register to read */
    uint32_t InSize;     /* Select input data width */
    uint32_t OutSize;    /* Select output data width */
    uint32_t InSelect;   /* Select floating-point input or fixed-point input */
    uint32_t OutSelect;  /* Select floating-point output or fixed-point output */
    uint32_t CodinLimit; /* Enables or disables Coordinate Limit */
    uint32_t PhaseLimit; /* Enables or disables Phase Limit */
} CORDIC_InitType;
```

其中：

- **Function:** 选择当前需要进行计算的数学函数，十选一。
例如：CORDIC_FUNCTION_SINE。
- **Precision:** 运算精度，实际写入寄存器的是计算所需要的时钟周期（迭代次数/4）。
例如：CORDIC_PRECISION_6CYCLES，表示 24 次迭代。
- **Scale:** 缩放因子。
例如：CORDIC_SCALE_2，表示缩放因子为 2^2 。
- **NbWrite:** 单次转换需要写数据寄存器 WDAT 的次数。
例如：CORDIC_NBWRITE_1，表示只需写 1 次数据寄存器。
- **NbRead:** 转换结束后，需要读取结果寄存器 RDAT 的次数。
例如：CORDIC_NBREAD_1，表示每次转换结束后只需读 1 次结果寄存器。
- **InSize:** 输入数据位宽。
例如：CORDIC_INSIZE_16BITS，表示输入数据有效位宽为 16 位。
- **OutSize:** 输出结果位宽。
例如：CORDIC_OUTSIZE_32BITS，表示转换结果有效位宽为 32 位。
- **InSelect:** 输入数据格式。
例如：CORDIC_INPUT_FIX，表示输入数据为定点格式。
- **OutSelect:** 输出数据格式，定点或浮点。
例如：CORDIC_OUTPUT_FLOAT，表示输出结果为浮点格式。

-
- **CodinLimit:** 坐标输出结果限幅，将输出结果限制在 $[-1,1)$ 。
例如：CORDIC_CODIN_LIMIT_ENABLE，表示坐标输出结果限制在 $[-1,1)$ 范围内。
 - **PhaseLimit:** 相位输出结果限幅，将输出结果限制在 $[-\pi,\pi)$ 。
例如：CORDIC_PHASE_LIMIT_DISABLE，表示相位输出结果不受限于 $[-\pi,\pi)$ ，超出范围将得不到正确结果。

3.3. CORDIC 中断与 DMA

CORDIC 初始化不包括中断与 DMA 配置，需要额外配置：

- 支持输入参数溢出中断与计算结果就绪中断，共用一个中断入口，可分别配置使能或禁用。
用户可通过下面的函数配置中断：
 - void CORDIC_InterruptCmd(uint32_t Interrupt, FunctionalStatus Cmd)
- 数据输入及结果输出均支持 DMA，可分别配置。
用户可通过下面的函数配置 DMA 功能：
 - void CORDIC_DMAReadRequestCmd(FunctionalStatus Cmd)
 - void CORDIC_DMAWriteRequestCmd(FunctionalStatus Cmd)

4. CORDIC 工作模式

CORDIC 共支持 4 种工作模式，不同工作模式的详细说明详见用户手册：

- 零开销模式：
在写入参数后，直接读取结果，如果结果未就绪，会插入总线等待状态，直至就绪并返回结果。
- DMA 模式：
通过 DMA 输入参数及读取结果。
- 轮询模式：
轮询 CORDIC_CTRLSTS 中的 RRF 标志，标志置 1 后读取结果。
- 中断模式：
使能中断，当 RRF 置位后产生计算结果就绪中断并读取结果。

因 CORDIC 计算只需要约 3~6 个时钟周期，其中零开销模式速度最快：

- DMA 模式：数据就绪后，需要 DMA 请求及数据传输等额外开销。
- 轮询模式：查询 RRF 标志需要额外的软件操作。
- 中断模式：需要额外的中断响应及中断服务函数调用。

实际应用中，用户可根据需要选择适合的工作模式。

5. 版本历史

版本	日期	备注
V1.0.0	2025-6-24	创建文档
V1.0.1	2025-12-5	修改部分描述。

6. 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。