应用笔记

Application Note

文档编号：**AN1132**

G32R501 Zidian 应用笔记

版本：**V1.0**

1 引言

本应用笔记介绍 G32R5xx Zidian 的相关内容，包括 Zidian 的基本介绍及应用程序的软件设计实例。

目录

2 Zidian 概述

Zidian 是为 G32R5xx 系列实时控制 MCU 设计的一套数学扩展指令集，能有效提高数学运算的性能。

2.1 ICAU 与 FCAU

在 G32R5xx 中，包括两类指令扩展：CDE 和 FPCDE。

对于扩展 CDE 指令，指令操作通用寄存器，zidian 中实施了一些特定指令以提高整数运算的性能，包括：

- 需要 SIMD 的计算：FFT，复数运算等
- CRC 算法

这组指令被称为整数计算加速单元（ICAU）。

对于扩展 FPCDE 指令，指令仅操作浮点寄存器，Zidian 中实施了一些特定的指令，以提高浮点运算的性能，包括：

- 三角函数
- 平方根
- 除法

这组指令被称为浮点计算加速单元（FCAU）。

2.2 CDE 内置函数

通过将 CDE 的标准化内置函数引入，作为 ARM C 语言扩展的一部分，表 1 和表 2 列出了 Zidian 支持的内置函数。

表格 1 ICAU 支持的内置函数

指令类型	内置函数
CX2	uint32_t __arm_cx2(int coproc, uint32_t n, uint32_t imm);
CX2A	uint32_t __arm_cx2a(int coproc, uint64_t acc, uint32_t n, uint32_t imm);
CX2DA	uint64_t __arm_cx2da(int coproc, uint64_t acc, uint32_t n, uint32_t imm);
CX3	uint32_t __arm_cx3(int coproc, uint32_t n, uint32_t m, uint32_t imm);
CX3D	uint64_t __arm_cx3d(int coproc, uint32_t n, uint32_t m, uint32_t imm);

指令类型	内置函数
CX3DA	uint64_t __arm_cx3da(int coproc, uint64_t acc, uint32_t n, uint32_t m, uint32_t imm);
CX2	uint32_t __arm_cx2(int coproc, uint32_t n, uint32_t imm);

表格 2 FACU 支持的内置函数

指令类型	内置函数
VCX2	uint32_t __arm_vcx2(int coproc, uint32_t n, uint32_t imm);
VCX3	uint32_t __arm_vcx3(int coproc, uint32_t n, uint32_t m, uint32_t imm);

通过这些内置函数，G32R5xx 在 Zidian 中对这些指令进行了重新命名。表 3 以 VCX3 指令为例，列举了 Zidian 中重新命名的 VCX3 函数。

表格 3 VCX3 重命名函数

指令	Zidian 函数
VCX3 0, Sd, Sn, Sm, #0x0	DIVF32 Sd, Sn, Sm
VCX3 0, Sd, Sn, Sm, #0x1	QUADF32 Sd, Sn, Sm
VCX3 0, Sd, Sn, Sm, #0x2	DIVF32_ATAN2 Sd, Sn

3 使用 Zidian 加速运算

3.1 zidian_math.h

在文件“zidian_cde.h”中，通过宏定义的方式对内置函数进行了重新命名。若需要在 driverlib 库中使用 Zidian 进行常规数学函数进行运算，则需要包含文件“zidian_math.h”，具体请查看 3.3.1 中内容。

文件“zidian_math.h”对常用的数学函数进行了重新封装。表 4 列举了“zidian_math.h”文件中所封装的数学函数。

表格 4 zidian_math 函数

Zidian 函数	描述
__sinpuf32	计算正弦值
__sin	将其归一化到 [0, 2π) 区间，然后再计算正弦值
__cospuf32	计算余弦值
__cos	将其归一化到 [0, 2π) 区间，然后再计算余弦值
__atanpuf32	计算反正切值
__atan	计算反正切值
__mpy2pif32	单精度浮点乘法
__div2pif32	单精度浮点除法
__sqrtf32	单精度浮点数平方根
__divf32	浮点除法
__quadf32	计算 X 和 Y 的象限值
__divf32_atan2	计算 X 和 Y 的比值
__atan2puf32	atan2
__atan2	atan2

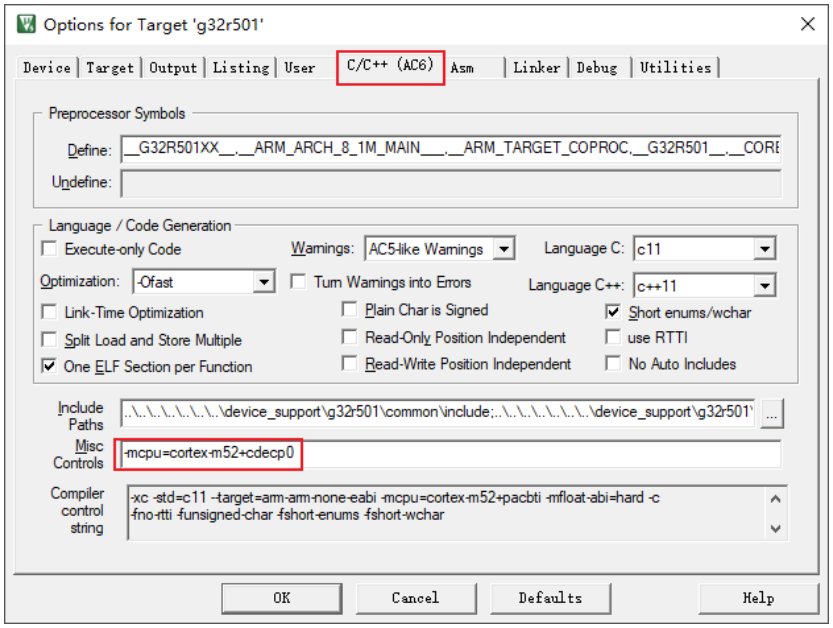
3.2 编译器支持

目前 SDK 中提供了 MDK-ARM 和 IAR 两种环境，不同的编译器下开启 Zidian 支持有些许不一样。

3.2.1 MDK-ARM（AC6）

在选项配置卡中，C/C++ （AC6）选项卡下的 Misc Controls 选项卡下加入“-mcpu=cortex-m52+cdecp0”即可开启支持。

图 1 AC6 开启 CDE

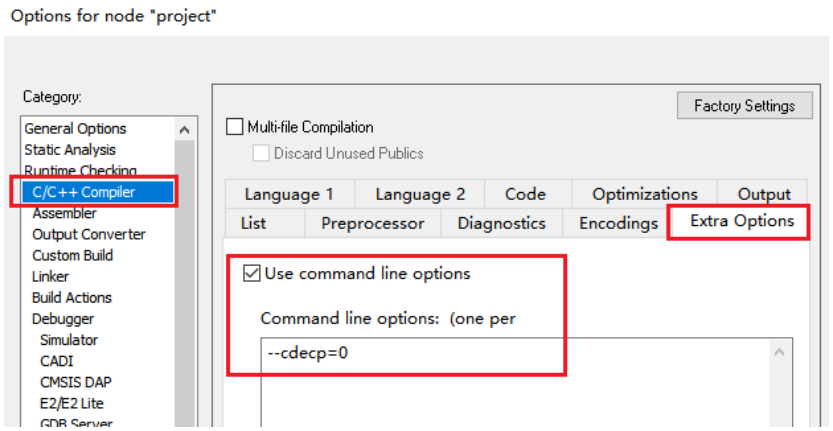


3.2.2 IAR（ICC）

IAR EW for Arm 支持分为编译支持。

编译支持的开启方式是在 C/C++ Compiler 下的 Extra Options，勾选 Use command line options 后，在命令行窗口添加“--cdecp=0”，即可开启支持。

图 2 ICC 开启 CDE 编译



3.3 软件设计示例

本节以 MDK-ARM 环境为例，介绍如何使用 Zidian 加速运算。

3.3.1 driverlib 结合 Zidian 加速运算

在 SDK-driverlib 例程中使用 Zidian 进行计算的具体步骤如下：

- 开启编译器支持
- 在工程配置的 C/C++选项卡下加入“__ZIDIAN_FCAU__”宏，以重定向 Zidian 支持的三角函数。
- 在需要 Zidian 参与运算的源码中包含“zidian_math.h”。

本应用参考 SDK-driverlib 例程 zidian_ex1_math，进行详细说明。

例程使用 Zidian，对 sqrt、sin、cos、atan、atan2 进行计算测试，具体代码如下：

由于在“zidian_math.h”文件中已经封装了这些函数，所以若想要使用 Zidian 计算这些函数，只需直接在头文件中引用“zidian_math.h”即可。

```
#include "zidian_math.h"

//
// Main
//
void example_main(void)
{
    volatile float x, y;
    uint16_t i;

    //
    // Test1 sqrtf
    //
    x = 0.81f;

    GET_DWT_CYCLE_COUNT(dwtCycleCounts[0],trace1Result[0] = sqrtf(x));

    for(i = 0; i < COUNT; i++)
    {
        trace1Result[i] = sqrtf(x);
    };

    //
    // Test2 sinf
    //
    x = PI / 2;
    GET_DWT_CYCLE_COUNT(dwtCycleCounts[1],trace2Result[1] = sinf(x));

    for(i = 0; i < COUNT; i++)
    {
        trace2Result[i] = sinf(x);
```

```
};

//
// Test3 cosf
//
x = PI / 4;
GET_DWT_CYCLE_COUNT(dwtCycleCounts[2], trace3Result[2] = cosf(x));

for(i = 0; i < COUNT; i++)
{
    trace3Result[i] = cosf(x);
};

//
// Test4 atanf
//
x = 5.0f;
GET_DWT_CYCLE_COUNT(dwtCycleCounts[3], trace4Result[3] = atanf(x));

for(i = 0; i < COUNT; i++)
{
    trace4Result[i] = atanf(x);
};

//
// Test5 atan2f
//
x = 5.0f;
y = 10.0f;
GET_DWT_CYCLE_COUNT(dwtCycleCounts[4], trace5Result[4] = atan2f(y, x));

for(i = 0; i < COUNT; i++)
{
    trace5Result[i] = atan2f(y, x);
};

//
// Loop
//
for(;;)
{
}
```

```
}
```

3.3.2 数学库结合 Zidian 加速运算

在数学库使用 Zidian 加速功能，对计算加速，具体步骤如下：

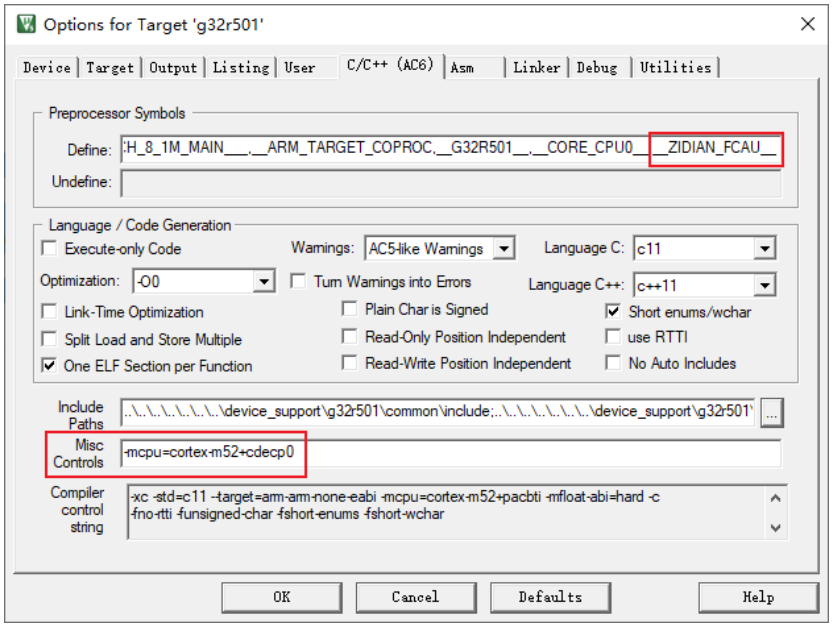
- 开启编译器支持
- 在工程配置的 C/C++选项卡下加入“__ZIDIAN_FCAU__”宏，以重定向 Zidian 支持的数学库函数。

以 FPUfastRTS 例程为例，使用 Zidian 进行计算加速。

由于在 FPUfastRTS 库中对 sin、cos 等函数进行了封装，若不在工程配置中加入“__ZIDIAN_FCAU__”宏，调用此类函数会使用 FPUfastRTS 库中的函数。

若要使用 Zidian 加速计算，需在相应工程配置的 C/C++选项卡下的 option 中添加相应宏,再调用目标函数即可。

图 3 数学库使用 Zidian 加速



开启编译器支持和添加__ZIDIAN_FCAU__宏后，直接调用相应函数，具体代码如下：

```
    in.f32   = test_input[i];

    // Run the calculation function and measure the DWT cycle count
simultaneously
    GET_DWT_CYCLE_COUNT(dwtCycleCounts[0], out.f32  = atanf(in.f32));
```

```
    test_output[i] = out.f32;
```

4 版本历史

表格 5 文件版本历史

日期	版本	变更历史
2025.01	1.0	新建

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。