GigaDevice Semiconductor Inc.

GD32G5x3 FFT 模块使用指南

应用笔记

AN248

1.0 版本

（2024 年 12 月）

目录

图索引

表索引

1. 前言

本文是专门为基于 GD32 MCU 开发的工程设计人员提供，主要介绍了 GD32G5x3 系列器件内 FFT 模块的内部结构、功能配置、以及使用时的注意事项，旨在帮助 GD32 MCU 开发者正确快速的使用 FFT 模块，缩短开发周期。

2. FFT 模块简介

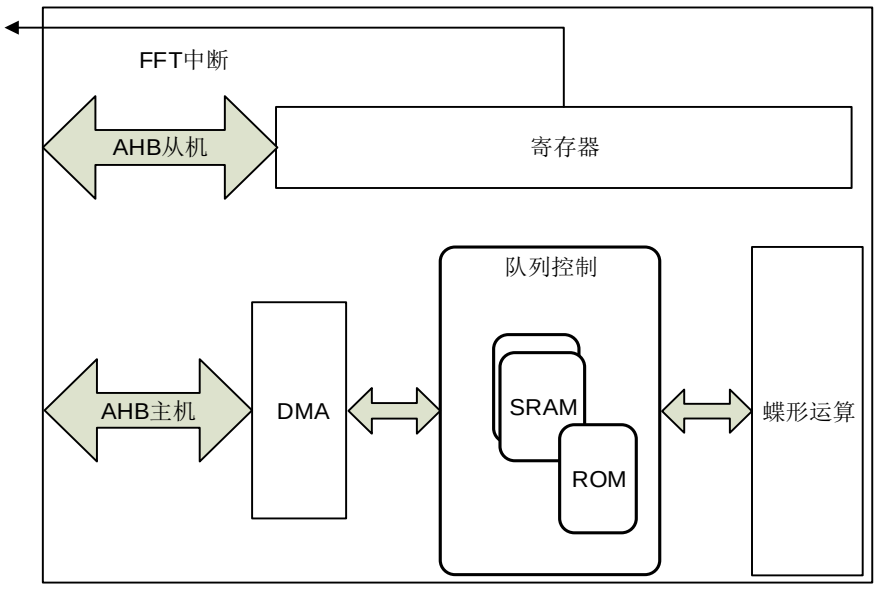
快速傅里叶变换（FFT）是离散傅里叶变换（DFT）的高效计算方法。该模块可以进行 FFT 运算，减轻了 CPU 负担。与软件实现相比，该模块可以加速了 FFT 的计算时间。该模块支持 6 个可配置的 FFT 点数，最多 1024，输入和输出数据应为 IEEE-754 单精度浮点复数。

FFT 模块主要特性如下：

- 支持1024/512/256/128/64/32点FFT；
- 支持IFFT模式；
- IEEE-754单精度浮点复数输入和输出数据；
- DMA主机加载和存储数据；
- 支持在内存中配置的窗函数功能；
- 支持输入下采样。

2.1. 运行过程

图 2-1. FFT 模块框图



FFT 模块有 DMA 主机、队列控制子模块、蝶形运算子模块和寄存器组成，如*图 2-1. FFT 模块框图*所示。该模块是使用的基-2 FFT 算法。先从存储中加载 IEEE754 单精度浮点格式的输入数据（包含实部和虚部）和窗函数数据，经过窗函数运算和输入位反运算之后，将结果写入内部 SRAM。旋转因子已经存储在 ROM 中。

蝶形运算子模块在 FFT 计算中被重复使用，并计算 $N / 2 * \log_2 N$ 次。每次计算是相同的地址，也就是说，每次计算的输出数据被写回之前存储输入数据的相同地址。

所有蝶形计算迭代完成后，DMA 主机将输出数据从内部 SRAM 传输到内存。

2.2. 操作指南

该部分介绍 FFT 的推荐操作指南。

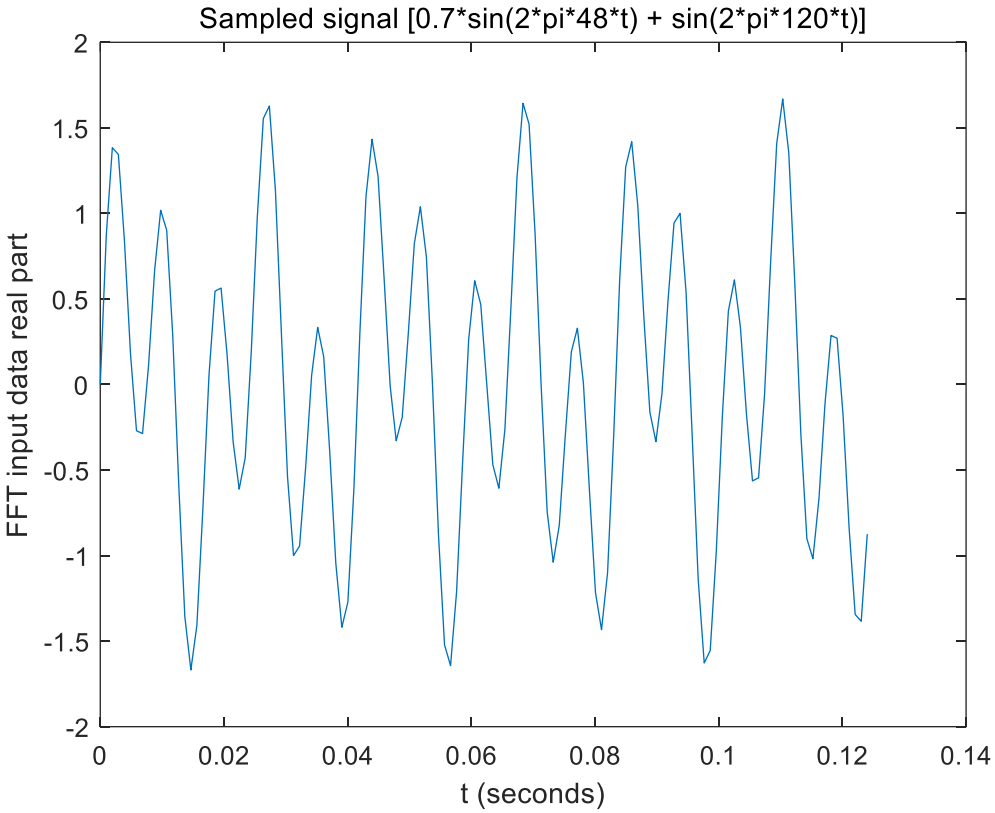
1. 如果需要，配置FFT_IMSADDR寄存器来设置FFT虚部的基地址，即虚部数据不为零。
2. 配置FFT_RESADDR寄存器来设置FFT实部基地址。
3. 如果需要，配置WINEN位和FFT_WSADDR寄存器来设置FFT窗函数的基地址。
4. 配置FFT_OSADDR寄存器来设置FFT输出结果的基地址。
5. 如果需要，配置FFT_LOOPLEN寄存器。
6. 如果需要，配置IFFTMODE。
7. 配置FFT_CSR寄存器中的NUMPT[2:0]位域来设置FFT点的数量。
8. 配置FFTEN位。
9. 等待FFTEN清零，或者CCF位置1。

3. FFT 配置代码示例

示例主要展示使用 FFT 模块进行 128 个输入数据的 FFT 变换（无加窗），FFT 模块的输入数据、窗函数和输出数据都是复数，包含实部和虚部。

128 个输入数据是从信号[0.7* sin(2π*48*t) + sin(2π*120*t)]上采集的，采集频率为 1024Hz，采样时长为 0.125s。可以明显看出，这个信号是两个正弦信号的叠加，其中一个正弦信号频率为 48Hz，幅值为 0.7；另外一个正弦信号频率为 120Hz，幅值为 1.0，如*图 3-1. FFT 实部输入数据*所示。其虚部默认是 0。

图 3-1. FFT 实部输入数据



输入数据的实部定义在数组 *fft_real_buf[FFT_DATA_LENGTH]*中，虚部默认是 0，代码如下：

表 3-1. FFT 输入数据代码

```
/* Specify the parameters of a signal with a sampling frequency of 1024 Hz and a signal duration of 0.125 seconds.Form a signal containing a 48 Hz sinusoid of amplitude 0.7 and a 120 Hz sinusoid of amplitude 1 input data = 0.7*sin(2*pi*48*t) + sin(2*pi*120*t) */
#define FERQ_SAMPLE             1024            //uint:Hz
#define PI                      3.1415926f
#define FFT_DATA_LENGTH         128U

float fft_real_buf[FFT_DATA_LENGTH] = {0};
```

```
for(index=0; index<FFT_DATA_LENGTH; index++){
fft_real_buf[index] = 0.7*sin(2*PI*48*index/FERQ_SAMPLE) + sin(2*PI*120*index/FERQ_SAMPLE);
}};
```

FFT 模块配置代码如下：

表 3-2. FFT 模块配置代码

```
/* fft parameter variable */
fft_parameter_struct fftconfig;
/* enable FFT clock */
rcu_periph_clock_enable(RCU_FFT);

/* reset the FFT */
fft_deinit();
/* initialize the FFT filter parameter struct */
fft_struct_para_init(&fftconfig);
/* setting FFT parameter */
fftconfig.mode_sel          = FFT_MODE;
fftconfig.point_num         = FFT_POINT_128;
fftconfig.window_enable = FFT_WINDOW_DISABLE;
fftconfig.downsamp_sel    = FFT_DOWNSAMPLE_1;
fftconfig.image_source    = FFT_IM_ZERO;
fftconfig.loopbuf_len     = 0U;
fftconfig.loopbuf_index = 0U;
fftconfig.real_addr         = (uint32_t)fft_real_buf;
fftconfig.image_addr       = NULL;
fftconfig.window_addr      = NULL;
fftconfig.output_addr     = (uint32_t)fft_output_buf;
/* config FFT parameter */
fft_init(&fftconfig);
/* start FFT calculation */
fft_calculation_start();
```

FFT 变换后的结果是 128 个输出数据，存储在 *fft_output_buf[FFT_DATA_LENGTH*2]*数组中，每个输出数据由两个元素表示，偶数元素是实部，奇数元素是虚部。比如第一个输出数据的实部是 *fft_output_buf[0]*，虚部是 *fft_output_buf[1]*，其他数据以此类推。输出数据的代码如下：

表 3-3. FFT 输出数据代码

```
/* output data */
float fft_output_buf[FFT_DATA_LENGTH*2] = {0};
```

通过串口可以将 FFT 变换结果输出，如下：

表 3-4. FFT 输出结果

```
FFT input data:
```

0.000000+0*i, 0.874758+0*i, 1.384084+0*i, 1.344315+0*i, 0.841806+0*i, 0.182527+0*i, -0.270391+0*i, -0.286645+0*i, 0.112291+0*i, 0.666867+0*i, 1.018484+0*i, 0.901419+0*i, 0.287692+0*i, -0.590805+0*i, -1.355038+0*i, -1.668654+0*i, -1.407107+0*i, -0.718927+0*i, 0.052364+0*i, 0.545101+0*i, 0.563592+0*i, 0.174369+0*i, -0.334832+0*i, -0.611566+0*i, -0.428905+0*i, 0.189789+0*i, 0.976833+0*i, 1.554358+0*i, 1.627501+0*i, 1.136808+0*i, 0.290884+0*i, -0.537751+0*i, -1.000000+0*i, -0.944151+0*i, -0.486918+0*i, 0.054591+0*i, 0.334070+0*i, 0.161101+0*i, -0.396262+0*i, -1.044898+0*i, -1.418853+0*i, -1.271523+0*i, -0.607962+0*i, 0.311590+0*i, 1.099346+0*i, 1.433252+0*i, 1.216425+0*i, 0.620792+0*i, -0.007105+0*i, -0.328937+0*i, -0.190983+0*i, 0.297341+0*i, 0.823447+0*i, 1.038642+0*i, 0.745359+0*i, 0.006915+0*i, -0.877654+0*i, -1.521333+0*i, -1.643492+0*i, -1.210734+0*i, -0.451630+0*i, 0.262099+0*i, 0.606286+0*i, 0.468361+0*i, 0.000004+0*i, -0.468355+0*i, -0.606285+0*i, -0.262103+0*i, 0.451622+0*i, 1.210728+0*i, 1.643489+0*i, 1.521336+0*i, 0.877661+0*i, -0.006908+0*i, -0.745356+0*i, -1.038644+0*i, -0.823453+0*i, -0.297348+0*i, 0.190978+0*i, 0.328938+0*i, 0.007110+0*i, -0.620784+0*i, -1.216419+0*i, -1.433251+0*i, -1.099350+0*i, -0.311597+0*i, 0.607956+0*i, 1.271521+0*i, 1.418857+0*i, 1.044904+0*i, 0.396268+0*i, -0.161095+0*i, -0.334071+0*i, -0.054595+0*i, 0.486914+0*i, 0.944146+0*i, 1.000000+0*i, 0.537754+0*i, -0.290873+0*i, -1.136801+0*i, -1.627500+0*i, -1.554359+0*i, -0.976842+0*i, -0.189795+0*i, 0.428903+0*i, 0.611570+0*i, 0.334841+0*i, -0.174363+0*i, -0.563589+0*i, -0.545103+0*i, -0.052370+0*i, 0.718921+0*i, 1.407099+0*i, 1.668653+0*i, 1.355043+0*i, 0.590809+0*i, -0.287685+0*i, -0.901416+0*i, -1.018487+0*i, -0.666871+0*i, -0.112300+0*i, 0.286641+0*i, 0.270392+0*i, -0.182518+0*i, -0.841798+0*i, -1.344311+0*i, -1.384085+0*i, -0.874766+0*i,

---

FFT calculation completion!

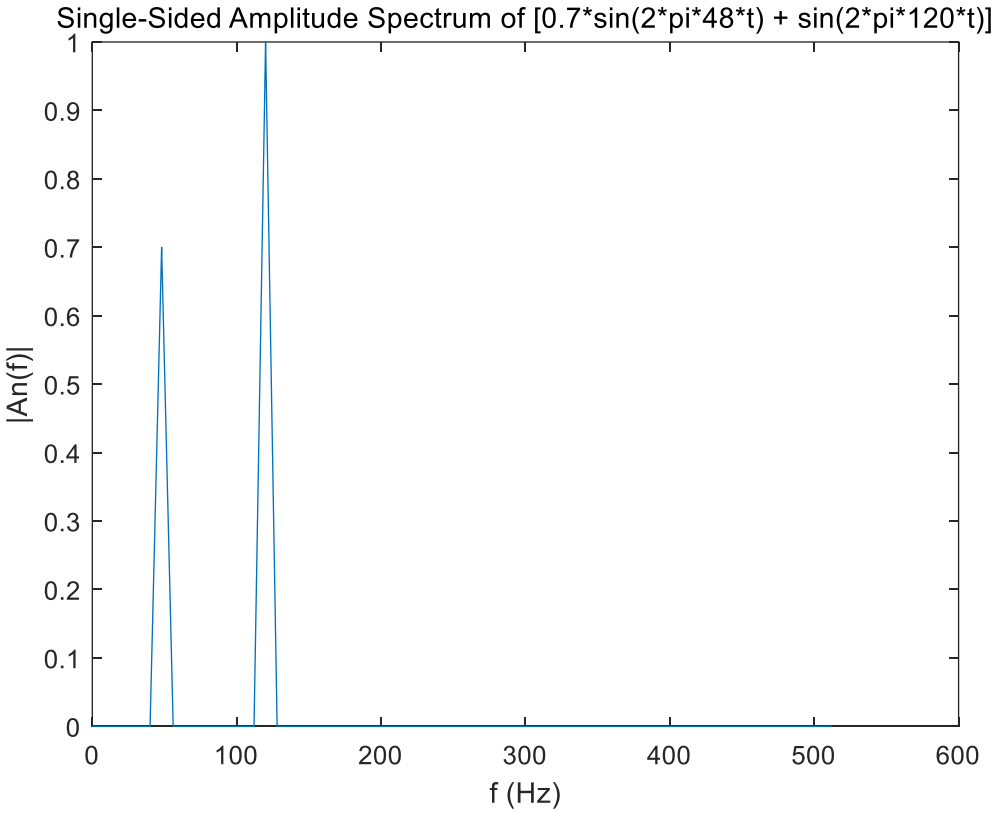
FFT calculation result:

0.000018+(0.000000*i), 0.000004+(0.000004*i), -0.000007+(-0.000003*i), 0.000009+(0.000004*i), 0.000010+(0.000009*i), -0.000001+(0.000007*i), 0.000001+(-44.799988*i), -0.000004+(0.000008*i), -0.000003+(0.000004*i), -0.000002+(0.000012*i), -0.000000+(0.000003*i), 0.000004+(0.000013*i), 0.000006+(0.000018*i), 0.000004+(0.000042*i), 0.000001+(0.000074*i), -0.000234+(-63.999992*i), -0.000004+(-0.000078*i), -0.000004+(-0.000035*i), 0.000003+(-0.000031*i), -0.000009+(-0.000018*i), -0.000018+(-0.000024*i), -0.000003+(-0.000044*i), 0.000021+(-0.000026*i), 0.000012+(-0.000005*i), -0.000001+(-0.000018*i), 0.000008+(-0.000012*i), 0.000009+(-0.000014*i), 0.000010+(-0.000004*i), 0.000012+(-0.000012*i), 0.000011+(0.000004*i), 0.000010+(-0.000008*i), 0.000001+(0.000005*i), -0.000006+(-0.000009*i), -0.000002+(-0.000003*i), 0.000011+(-0.000010*i), 0.000007+(-0.000003*i), -0.000004+(0.000004*i), 0.000002+(-0.000014*i), 0.000007+(-0.000005*i), 0.000000+(0.000011*i), -0.000007+(-0.000022*i), 0.000002+(-0.000002*i), 0.000012+(-0.000021*i), 0.000010+(0.000004*i), 0.000010+(-0.000006*i), 0.000011+(0.000003*i), 0.000007+(-0.000000*i), 0.000000+(0.000003*i), 0.000001+(-0.000003*i), -0.000009+(0.000008*i), -0.000015+(-0.000014*i), 0.000009+(-0.000026*i), 0.000026+(-0.000009*i), 0.000014+(0.000010*i), 0.000000+(-0.000004*i), 0.000011+(-0.000010*i), 0.000017+(0.000011*i), 0.000006+(-0.000000*i), 0.000002+(0.000008*i), 0.000006+(-0.000009*i), 0.000011+(0.000001*i), 0.000003+(0.000016*i), -0.000000+(-0.000003*i), 0.000001+(0.000008*i), -0.000003+(0.000000*i), 0.000001+(-0.000008*i), -0.000000+(0.000003*i), 0.000003+(-0.000016*i), 0.000011+(-0.000001*i), 0.000006+(0.000009*i), 0.000002+(-0.000008*i), 0.000006+(0.000000*i), 0.000017+(-0.000011*i), 0.000011+(0.000010*i), 0.000000+(0.000004*i), 0.000014+(-0.000010*i), 0.000026+(0.000009*i),

0.000009+(0.000026*i), -0.000015+(0.000014*i), -0.000009+(-0.000008*i), 0.000001+(0.000003*i), 0.000000+(-0.000003*i), 0.000007+(0.000000*i), 0.000011+(-0.000003*i), 0.000010+(0.000006*i), 0.000010+(-0.000004*i), 0.000012+(0.000021*i), 0.000002+(0.000002*i), -0.000007+(0.000022*i), 0.000000+(-0.000011*i), 0.000007+(0.000005*i), 0.000002+(0.000014*i), -0.000004+(-0.000004*i), 0.000007+(0.000003*i), 0.000011+(0.000010*i), -0.000002+(0.000003*i), -0.000006+(0.000009*i), 0.000001+(-0.000005*i), 0.000010+(0.000008*i), 0.000011+(-0.000004*i), 0.000012+(0.000012*i), 0.000010+(0.000004*i), 0.000009+(0.000014*i), 0.000008+(0.000012*i), -0.000001+(0.000018*i), 0.000012+(0.000005*i), 0.000021+(0.000026*i), -0.000003+(0.000044*i), -0.000018+(0.000024*i), -0.000009+(0.000018*i), 0.000003+(0.000031*i), -0.000004+(0.000035*i), -0.000004+(0.000078*i), -0.000234+(63.999992*i), 0.000001+(-0.000074*i), 0.000004+(-0.000042*i), 0.000006+(-0.000018*i), 0.000004+(-0.000013*i), -0.000000+(-0.000003*i), -0.000002+(-0.000012*i), -0.000003+(-0.000004*i), -0.000004+(-0.000008*i), 0.000001+(44.799988*i), -0.000001+(-0.000007*i), 0.000010+(-0.000009*i), 0.000009+(-0.000004*i), -0.000007+(0.000003*i), 0.000004+(-0.000004*i)

上述输出数据 *fft_output_buf* 对应的幅频图如所示

图 3-2. FFT 输出数据幅频特性



4. 版本历史

表 4-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2024 年 12 月 31 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.