北京大学
PEKING UNIVERSITY

FFT accelerater实验

王俊棋

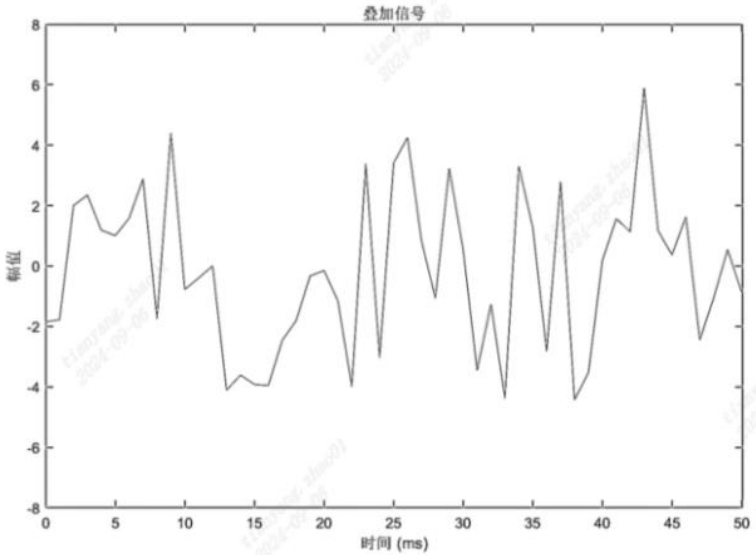
2024 11 13

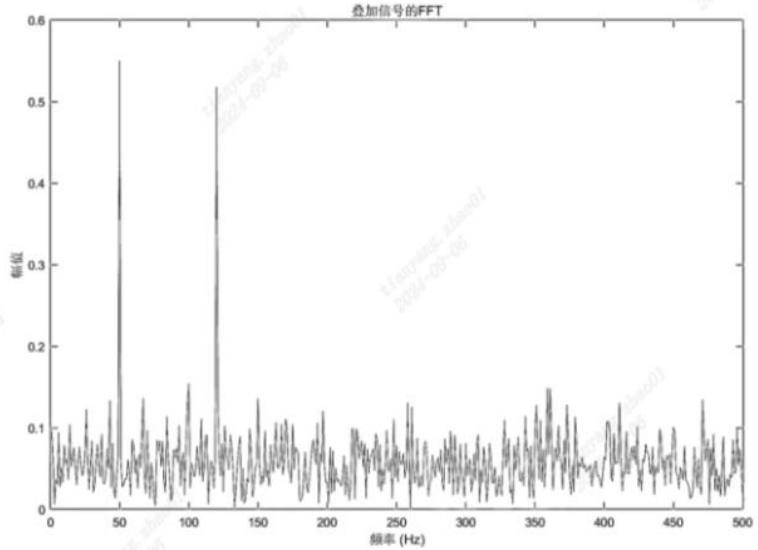
Outline

- DFT及FFT背景介绍
- FFT accelerater介绍
- 代码解析

DFT及FFT背景介绍

傅里叶变换的背景：





傅里叶变换的基本思想：

一个函数可以用无穷个周期函数的线性组合来逼近，组合函数在保有原函数几乎全部信息的同时，还直接反映了该函数的“频域特征”

DFT及FFT背景介绍

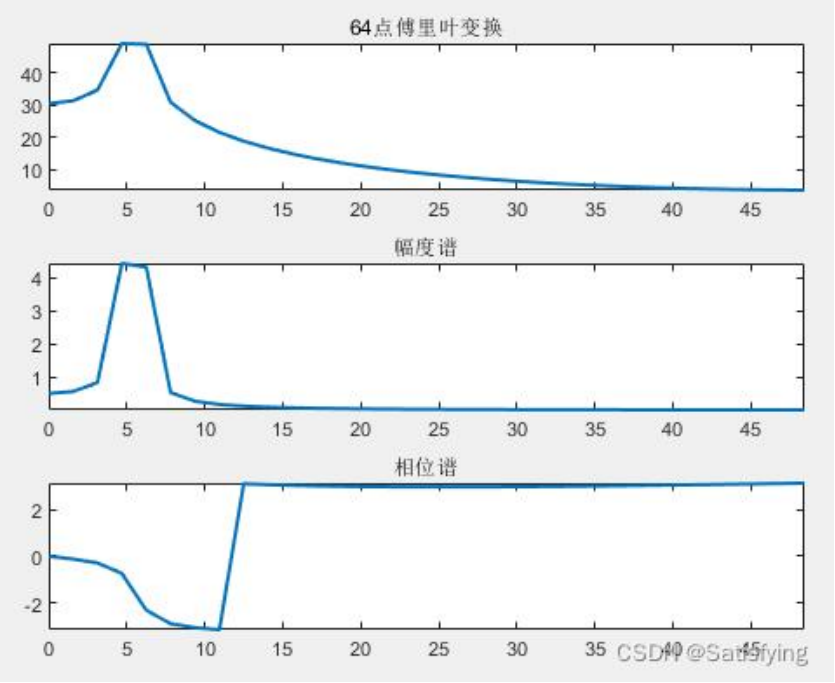
北京大学 PEKING UNIVERSITY

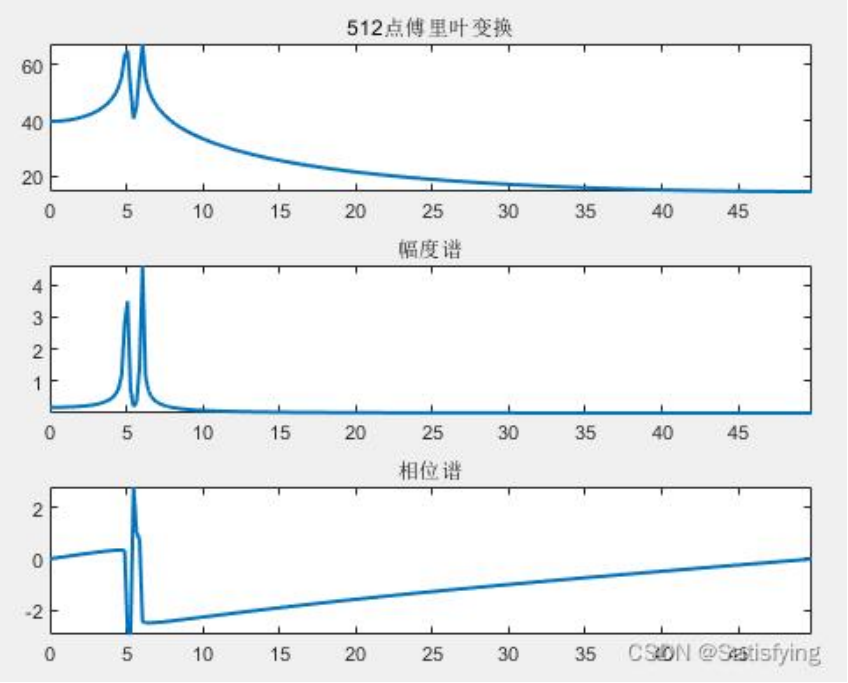
DFT的数学表达：

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi kn/N}$$

- (x(n)) 是时域信号的第 (n) 个样本。
- (X(k)) 是频域信号的第 (k) 个样本。
- (N) 是总样本数。
- (k) 是频率索引，从 0 到 (N-1)。
- (e^{-i 2 \pi k n / N}) 是一个复数，表示在 (n) 时刻，频率为 (k) 的复指数函数的值。

FFT（快速傅里叶变换）：一类快速计算离散傅里叶变换的方法，可以将计算N点序列DFT的复乘数降低到（$(N/2)\log_2 N$）。





FFT accelerater介绍

北京大学 PEKING UNIVERSITY

功能介绍：

①支持64点、128点、256点、512点的FFT及IFFT；

②支持可配的输入数据位宽：32位及64位；

③支持可配置的数据排列方式：虚部实部交替、纯实部、实虚部分离三种数据排列方式；

④支持DMA传输。

配置流程：

①初始化模块：配置FFT点数、选择模式（FFT、IFFT）、数据输入格式、使能模块；

②写入需要计算的数据；

③计算完成，读出结果。

API介绍

9.3.1 fft_complex_uint16_dma

```
void fft_complex_uint16_dma(dmac_channel_number_t dma_send_channel_num,
    dmac_channel_number_t dma_receive_channel_num, uint16_t shift, fft_direction_t
    direction, const uint64_t *input, size_t point_num, uint64_t *output);
```

参数名称	描述	输入输出
dma_send_channel_num	发送数据使用的 DMA 通道号	输入
dma_receive_channel_num	接收数据使用的 DMA 通道号	输入
shift	FFT 模块 16 位寄存器导致数据溢出 (-32768~32767)，FFT 变换有 9 层，shift 决定哪一层需要移位操作 (如 0x1ff 表示 9 层都做移位操作；0x03 表示第第一层与第二层做移位操作)，防止溢出。如果移位了，则变换后的幅值不是正常 FFT 变换的幅值，对应关系可以参考 fft_test 测试 demo 程序。包含了求解频率点、相位、幅值的示例	输入
direction	FFT 正变换或是逆变换	输入
input	输入的数据序列，格式为 RIRI..，实部与虚部的精度都为 16bit	输入
point_num	待运算的数据点数，只能为 512/256/128/64	输入
output	运算后结果。格式为 RIRI..，实部与虚部的精度都为 16bit	输出

库文件及参数


```c
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include "encoding.h"
#include "dmac.h"
#include "fft.h"
#include "encoding.h"
#include "sysctl.h"
#include "fft_soft.h"

#define FFT_N                512U
#define FFT_FORWARD_SHIFT    0x0U
#define FFT_BACKWARD_SHIFT   0x1ffU
#define PI                   3.14159265358979323846

uint64_t buffer_input[FFT_N];
uint64_t buffer_output[FFT_N];
```

```c
typedef struct{double real, imag;} complex;

void fft_soft(complex *data, int n);
void ifft_soft(complex *data, int n);
```

```c
typedef struct _complex_hard
{
    int16_t real;
    int16_t imag;
} complex_hard_t;

typedef struct _fft_data
{
    int16_t I1;
    int16_t R1;
    int16_t I2;
    int16_t R2;
} fft_data_t;

typedef enum _fft_direction
{
    FFT_DIR_BACKWARD,
    FFT_DIR_FORWARD,
    FFT_DIR_MAX,
} fft_direction_t;
```

FFT代码部分实现

北京大学 PEKING UNIVERSITY

生成FFT输入数据，通过硬件和软件分别计算，并获取计算时间

```
//生成需要进行fft的数据
    for (i = 0; i < FFT_N; i++)
    {
        tempf1[0] = 0.3 * cosf(2 * PI * i / FFT_N + PI / 3) * 256;
        tempf1[1] = 0.1 * cosf(16 * 2 * PI * i / FFT_N - PI / 9) * 256;
        tempf1[2] = 0.5 * cosf((19 * 2 * PI * i / FFT_N) + PI / 6) * 256;
        data_hard[i].real = (int16_t)(tempf1[0] + tempf1[1] + tempf1[2] + 10);
        data_hard[i].imag = (int16_t)0;
        data_soft[i].real = data_hard[i].real;
        data_soft[i].imag = data_hard[i].imag;
    }
    for (int i = 0; i < FFT_N / 2; ++i)//将生成的数据转移到输入信号中（buffer_input）
    {
        input_data = (fft_data_t *)&buffer_input[i];
        input_data->R1 = data_hard[2 * i].real;
        input_data->I1 = data_hard[2 * i].imag;
        input_data->R2 = data_hard[2 * i + 1].real;
        input_data->I2 = data_hard[2 * i + 1].imag;
    }
//进行硬件和软件计算，并获取计算时间
    cycle[FFT_HARD][FFT_DIR_FORWARD] = read_cycle();
    fft_complex_uint16_dma(DMAC_CHANNEL0, DMAC_CHANNEL1, FFT_FORWARD_SHIFT, FFT_DIR_FORWARD, buffer_input, FFT_N, buffer_output);
    cycle[FFT_HARD][FFT_DIR_FORWARD] = read_cycle() - cycle[FFT_HARD][FFT_DIR_FORWARD];
    cycle[FFT_SOFT][FFT_DIR_FORWARD] = read_cycle();
    fft_soft(data_soft, FFT_N);
    cycle[FFT_SOFT][FFT_DIR_FORWARD] = read_cycle() - cycle[FFT_SOFT][FFT_DIR_FORWARD];
```

IFFT代码部分实现

北京大学 PEKING UNIVERSITY

通过硬件和软件分别计算IFFT，并获取计算时间

```
//重新计算IFFT
    for (int i = 0; i < FFT_N / 2; ++i)
    {
        input_data = (fft_data_t *)&buffer_input[i];
        input_data->R1 = data_hard[2 * i].real;
        input_data->I1 = data_hard[2 * i].imag;
        input_data->R2 = data_hard[2 * i + 1].real;
        input_data->I2 = data_hard[2 * i + 1].imag;
    }
    cycle[FFT_HARD][FFT_DIR_BACKWARD] = read_cycle();
    fft_complex_uint16_dma(DMAC_CHANNEL0, DMAC_CHANNEL1, FFT_BACKWARD_SHIFT, FFT_DIR_BACKWARD, buffer_input, FFT_N, buffer_output);
    cycle[FFT_HARD][FFT_DIR_BACKWARD] = read_cycle() - cycle[FFT_HARD][FFT_DIR_BACKWARD];
    cycle[FFT_SOFT][FFT_DIR_BACKWARD] = read_cycle();
    ifft_soft(data_soft, FFT_N);
    cycle[FFT_SOFT][FFT_DIR_BACKWARD] = read_cycle() - cycle[FFT_SOFT][FFT_DIR_BACKWARD];
    for (i = 0; i < FFT_N / 2; i++)
    {
        output_data = (fft_data_t*)&buffer_output[i];
        data_hard[2 * i].imag = output_data->I1 ;
        data_hard[2 * i].real = output_data->R1 ;
        data_hard[2 * i + 1].imag = output_data->I2 ;
        data_hard[2 * i + 1].real = output_data->R2 ;
    }
```

结果分析

北京大学 PEKING UNIVERSITY

	[hard fft real]	[soft fft real]	[hard fft imag]	[soft fft imag]
0:	5109	5109	0	0
1:	9805	9801	16977	16982
2:	4	0	-5	-4
3:	4	0	2	-3
4:	-5	-3	9	9
5:	7	-2	-1	0
6:	-5	-6	13	6
7:	0	0	-6	-3
8:	4	4	-1	0
9:	6	3	3	0
10:	3	-1	2	2
11:	3	0	-6	-3
12:	1	1	-12	-12
13:	1	0	-9	-5
14:	-6	-7	-6	-2
15:	-4	-3	2	1
16:	6140	6140	-2238	-2237
17:	-4	-5	-16	-14
18:	0	0	2	3
19:	28246	28247	16306	16307
20:	-7	-7	0	4

	hard power	soft power:
0 :	9.978516	9.978516
1 :	76.582085	76.592094
2 :	0.025012	0.019302
3 :	0.017469	0.012297
4 :	0.040217	0.038868
5 :	0.027621	0.009179
6 :	0.054408	0.035621
7 :	0.023438	0.011783
8 :	0.016106	0.017883
9 :	0.026204	0.014552
10 :	0.014084	0.011185
11 :	0.026204	0.012172
12 :	0.047037	0.049996
13 :	0.035373	0.021000
14 :	0.033146	0.029610
15 :	0.017469	0.013857
16 :	25.527946	25.528057
17 :	0.064424	0.060285
18 :	0.007812	0.015604
19 :	127.401382	127.410881

	hard phase	soft phase:
0 :	0.000000	0.000000
1 :	59.991585	60.008694
2 :	-51.340191	-83.773621
3 :	26.565050	-82.332520
4 :	119.054604	108.723869
5 :	-8.130102	-172.541611
6 :	111.037506	135.838745
7 :	-90.000000	-92.087402
8 :	-14.036242	8.737115
9 :	26.565050	-12.262694
10 :	33.690067	119.307495
11 :	-63.434948	-78.735786
12 :	-85.236351	-82.604240
13 :	-83.659805	-86.469994
14 :	-135.000000	-159.547562
15 :	153.434952	157.457642
16 :	-20.026550	-20.021925
17 :	-104.036240	-111.489159
18 :	90.000000	96.845062
19 :	29.997206	29.998243
20 :	180.000000	150.203659

结果分析

北京大学 PEKING UNIVERSITY

```
[before FFT]
[hard fft real][soft fft real][hard fft imag][soft fft imag]
 0:     183     183       0       0
 1:     165     165       0       0
 2:     142     142       0       0
 3:     114     114       0       0
 4:      82      82       0       0
 5:      49      49       0       0
 6:      16      16       0       0
 7:     -14     -14       0       0
 8:     -42     -42       0       0
 9:     -66     -66       0       0
10:     -83     -83       0       0
11:     -94     -94       0       0
12:     -98     -98       0       0
13:     -95     -95       0       0
14:     -84     -84       0       0
15:     -67     -67       0       0
16:     -45     -45       0       0
17:     -20     -20       0       0
18:       8       8       0       0
19:      37      37       0       0
20:      66      66       0       0
```

```
[hard ifft real][soft ifft real][hard ifft imag][soft ifft imag]
 0:     183     183       0       0
 1:     166     164       0       0
 2:     143     142       0       0
 3:     114     113       0       0
 4:      82      81       0       0
 5:      50      48       0       0
 6:      16      15       0       0
 7:     -14     -13       0       0
 8:     -42     -41       0       0
 9:     -66     -66       0       0
10:     -83     -82       1       0
11:     -94     -93       0       0
12:     -98     -98       0       0
13:     -95     -94       0       0
14:     -84     -83       0       0
15:     -67     -67       0       0
16:     -45     -44       0       0
17:     -20     -19       0       0
18:       8       8       0       0
19:      37      37       0       0
20:      66      66       0       0
```

结果分析

第一次运行：

```
[hard fft test] [512 bytes] forward time = 231603 us, backward time = 139 us
[soft fft test] [512 bytes] forward time = 40487 us, backward time = 41026 us
```

```
[hard fft test] [256 bytes] forward time = 193411 us, backward time = 111 us
[soft fft test] [256 bytes] forward time = 17501 us, backward time = 17784 us
```

```
[hard fft test] [128 bytes] forward time = 193554 us, backward time = 101 us
[soft fft test] [128 bytes] forward time = 7366 us, backward time = 7522 us
```

```
[hard fft test] [64 bytes] forward time = 193702 us, backward time = 95 us
[soft fft test] [64 bytes] forward time = 3012 us, backward time = 3086 us
```

第二次运行：

```
[hard fft test] [512 bytes] forward time = 131 us, backward time = 137 us
[soft fft test] [512 bytes] forward time = 40910 us, backward time = 41371 us
```

北京大学
PEKING UNIVERSITY

THANK YOU !