

VILNIAUS UNIVERSITETAS
FIZIKOS FAKULTETAS
KIETO KŪNO ELEKTRONIKOS KATEDRA

Mindaugas Kurmauskas

CORTEX R4 MIKROVALDIKLIO ARCHITEKTŪROS TYRIMAS

Pagrindinių studijų kursinis darbas

(studijų programa – TAIKOMOJI FIZIKA)

Studentas

Darbo vadovas

Katedros vedėjas

Mindaugas Kurmauskas

dr. Mindaugas Vilūnas

dr.(HP) Kęstutis Arlauskas

Vilnius 2013

Turiny

Įvadas	3
1 Naudojamos įrangos ir programos	4
2 Naudoti testavimo algoritmai	5
2.1 Slankaus kabelio algoritmai	5
2.2 Neslankaus kabelio algoritmas	6
2.3 Bitu testavimo algoritmas	6
2.4 Greitas diskretinis Furjė eilutės transformavimo testas	7
3 Rezultatai	9
4 Išvados	9
5 Priedai	10
Mokslinės publikacijos ir konferencijų pranešimai	11
Literatūros sąrašas	13
Santrauka	13
Summary	14

Ivadas

Čia yra įvado tekstas!

1. Pirmas
2. Antras

1 Naudojamos įrangos ir programos

Naudojamas TMDXRM48USB paleisties rinkinys¹. Jame yra:

- Mikrovaldiklis xRM48L950AZWTT:
 - Du 32-bitu ARM Cortex-R4F procesoriai, veikiantys kartu²
 - 3MB flash, 256kB RAM
 - greitis iki 220MHz
 - neslankaus(32bitu) ir dvigubo tikslumo (64bitu) slankaus kabelio aritmetikos modulis
- Integruotas XDS100v2 emuliatorius programavimui per usb
- LED'ai, temperatūros sensorius, šviesos sensorius, akselerometras
- prožektoriukas

Palyginimui naudojamas STM32F4DISCOVERY paleisties rinkinys:

- mikrovaldiklis STM32F407VGT:
 - 32-bitu ARM Cortex-M4F procesorius
 - 1MB flash, 192 kB RAM
 - greitis iki 168MHz
 - neslankaus (32bitu) ir slankaus (32bitu) kabelio aritmetikos modulis
- Integruotas ST-LINK/V2 emuliatorius programavimui per usb
- LED'ai, akselerometras, skaitmeninis mikrofonas
- CS43L22- SAK garsui su integruotu D klases garso stiprinimu

¹development kit

²lockstep - kartu atlieka tas pačias komandas

2 Naudoti testavimo algoritmai

2.1 Slankaus kablelio algoritmai

Slankaus kablelio Gauss Legendre algoritmas π skaičiavimui:

$$\begin{aligned}a_0 &= 1 \quad b_0 = \frac{1}{\sqrt{2}} \quad t_0 = \frac{1}{4} \quad p_0 = 1 \\a_{n+1} &= \frac{a_n + b_n}{2}, \\b_{n+1} &= \sqrt{a_n b_n}, \\t_{n+1} &= t_n - p_n(a_n - a_{n+1})^2, \\p_{n+1} &= 2p_n. \\\pi &\approx \frac{(a_n + b_n)^2}{4t_n}\end{aligned}\tag{1}$$

Naudojant dvigubo tikslumo kintamųjų testą buvo ieškomas 1000 narys, suskaičiuota konstanta nuo pasirinktosios skiriasi $3.55271 * 10^{-15}$.

```
int doubleTest() {
    volatile int i;
    volatile double an,bn,tn,pi;
    volatile double a,b,t,p;
    a = 1.0; b = 1/sqrt(2);
    t = 1/4; p = 1.0;
    for (i = 0; i < 1000; i++) {
        an = (a+b)/2;
        bn = sqrt(a*b);
        tn = t - p*(a-an)*(a-an);
        p *= 2;
        pi = (an+bn)*(an+bn)/(4*tn);
        a = an; b = bn; t = tn;
    }
    if ((pi - 3.14159265358979) <= 3.55271e-15)
        return 1; //testas atliktas sekmingai
    return 0; //teste ivyko klaida
}
```

Naudojant viengubo tikslumo slankaus kablelio kintamuosius buvo ieškomas 120 narys. Konstanta nuo pasirinktosios skiriasi $8.74228 * 10^{-8}$

```
int floatTest() {
    volatile int i;
```

```

volatile float an,bn,tn,pi;
volatile float a,b,t,p;
for(volatile int j = 0; j <9; j++) {
    a = 1.0; b = sqrt(0.5);
    t = 0.25; p = 1.0;
    for (i = 0; i < 120; i++) {
        an = (a+b)/2;
        bn = sqrt(a*b);
        tn = t - (p*(a-an)*(a-an));
        p *= 2;
        pi = (an+bn)*(an+bn)/(4*tn);
        a = an; b = bn; t = tn;
    }
}
if ((pi - 3.14159265358979) <= 8.74228e-8)
    return 1; //testas atliktas sekmingai
return 0; //teste ivyko klaida
}

```

2.2 Neslankaus kablelio algoritmas

Fiksuoto tikslumo algoritmas:

$$b = \sum_{i=0}^{100000} (i * (-1)^{i+1}) \quad (2)$$

```

void intTest() {
    volatile int a,b;
    a = 1;
    b = 0;
    for(volatile int i = 0; i < 100000; i++) {
        b += i * a;
        a *= -1;
    }
}

```

2.3 Bitu testavimo algoritmas

```

short test_bits() {
    volatile unsigned sum = 0x55555555;
    // 0101 0101 0101 0101 0101 0101 0101 0101
    for (volatile int i = 0; i < 0x020000; i++) {

```

```

    sum = sum << 1;          /// sum = 0xAAAAAAAA
    sum &= 0x0000FFFF;       /// sum = 0x0000AAAA
    sum ^= 0xFFFFFFFF;       /// sum = 0xFFFF5555
    sum = sum << 16;         /// sum = 0x55550000
    sum = sum | (sum >> 16);  /// sum = 0x55555555
    sum = ~sum;              /// sum = 0xAAAAAAAA
    sum = sum >> 1;          /// sum = 0x55555555
}
if (0x55555555 == sum)
    return 1; //testas ivykditas sekmingai
return 0; // teste ivyko klaida
}

```

2.4 Greitas diskretinis Furjė eilutės transformavimo testas

Naudotos arm dsp³ bibliotekos, atskirai optimizuotos R4 bei M4 mikrovaldikliams. Naudota 1024 kompleksinių taškų, kurie masyve išdėstyti vienas po kito ir bendras masyvo ilgis 2048. Slankaus kabelio testui masyvas užpildomas funkcija:

$$Re(z) = (\sin(v * i) + \cos(v * 7)), t = 0, 2, 4...2046$$

$$Im(z) = 0 \tag{3}$$

```

#define TEST_LENGTH_SAMPLES 2048
static float32_t testInput[TEST_LENGTH_SAMPLES];
static float32_t testOutput[TEST_LENGTH_SAMPLES/2];
uint32_t fftSize = 1024;
uint32_t ifftFlag = 0;
uint32_t doBitReverse = 1;

void prepareTestFloat(void) {
    for(volatile int t = 0; t < 2048; t+=2) {
        testInput[i] = sin(t*3) + cos(t*7);
        testInput[i+1] = 0;
    }
}

int32_t testCfftFloat(void) {
    arm_status status;
    arm_cfft_radix4_instance_f32 S;
    float32_t maxValue;

```

³Dsp - digital signal processing (skaitmeninis signalų apdorojimas)

```

arm_cfft_radix4_init_f32(&S, fftSize, ifftFlag, doBitReverse);

arm_cfft_radix4_f32(&S, testInput);

arm_cmplx_mag_f32(testInput, testOutput, fftSize);

arm_max_f32(testOutput, fftSize, &maxValue, &testIndex);

if(testIndex == refIndex) {
    return 1;
}
return 0;
}

```

Fiksuoto tikslumo Furijė transformacijai naudoti ARM Q31 fiksuoto kablelio kintamieji. Iš slankaus kablelio paversti į Q31 naudota funkcija:

$$Q31(x) = ROUND(((2^{31}) * (10^{(x/20)}))) \quad (4)$$

Masyvo užpildo funkcija

$$Re(z) = Q31(1000 * (\sin(v * i) + \cos(v * 7))), \quad t = 0, 2, 4 \dots 2046$$

$$Im(z) = 0 \quad (5)$$

```

#define TEST_LENGTH_SAMPLES 2048
uint32_t fftSize = 1024;
uint32_t ifftFlag = 0;
uint32_t doBitReverse = 1;
static q31_t testInputQ[TEST_LENGTH_SAMPLES];
static q31_t testOutputQ[TEST_LENGTH_SAMPLES/2];
uint32_t refQIndex = 0;

void prepareTestQ31(void) {
    double skaicius=0;
    int nulis = (int)round((pow(2,31)) * pow(10,(0/20)));
    for(volatile int32_t i = 0; i < 2048; i+=2) {
        skaicius = 1000*sin(i*3) + 1000*cos(i*7);
        testInputQ[i] = (int)round((pow(2,31)) * pow(10,(skaicius/20)));;
        testInputQ[i+1] = nulis;
    }
}

```



```

int32_t testCfftQ31(void) {
    arm_status status;
    arm_cfft_radix4_instance_q31 S;
    q31_t maxValue;

    status = arm_cfft_radix4_init_q31(&S, fftSize, ifftFlag, doBitReverse);

    arm_cfft_radix4_q31(&S, testInputQ);

    arm_cmplx_mag_q31(testInputQ, testOutputQ, fftSize);

    arm_max_q31(testOutputQ, fftSize, &maxValue, &testIndex);

    if(testIndex == refIndex) {
        return 1;
    }
    return 0;
}

```

3 Rezultatai

4 Išvados

1. Pirmas
2. Antras
3. Trečias
4. Ketvirtas

5 Priedai

Bet kokia reikalinga papildoma informacija: paveikslėliai, grafikai ir t.t.

Mokslinės publikacijos ir konferencijų pranešimai

Šiame skyriuje pateikiamos mokslinės publikacijos bei žodiniai ir stendiniai pranešimai konferencijose, kurie yra susiję su "mano darbo pavadinimas".

Mokslinių publikacijų, įtrauktų į mokslinės informacijos instituto (ISI) pagrindinių žurnalų duomenų bazę, sąrašas

- 1.
- 2.

Kitos mokslinės publikacijos

- 1.
- 2.

Žodiniai pranešimai konferencijose

- 1.
- 2.

Stendiniai pranešimai konferencijose

- 1.
- 2.

Santrauka

Mindaugas Kurmauskas

„CORTEX R4 MIKROVALDIKLIO ARCHITEKTŪROS TYRIMAS”

Tekstas

Summary

Mindaugas Kurmauskas

„TITLE”

Text