Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных систем и сетей

Кафедра ЭВС

Отчёт по лабораторной работе №7,8

**«Программирование полигармоники для измерения АЧХ фильтра. Измерение АЧХ цифрового рекурсивного фильтра с помощью БПФ»**

**Вариант 3**

Выполнили: Проверил:

студенты гр. 550701 Герасимович В.Ю.

Шимко М.Д.

Кудрявцев П.Д.

Богданович В.В.

Минск 2018

**Цель:**

Запрограммировать генератор полигармонического сигнала на 2N отсчетов (сумма гармоник от 1-й до N/2) табличным методом для измерения АЧХ фильтра. При программировании фазу очередной гармоники необходимо сдвигать на 2 отсчета относительно предыдущей. Проще всего это сделать формируя 2N+2 отсчета сигнала и не изменяя аргумента только увеличивать шаг в таблице на 1 при переходе к очередной гармонике. БПФ применяется ко второй половине полигармоники на выходе фильтра (последние N точек из 2N точек).

**Исходные данные:**

|  |  |
| --- | --- |
| **Вариант** | **N (длина 1-й гармоники)** |
|
| 3 | 128 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тип и параметры фильтра** | | | | | |
| **Тип** | **Aa** | **Fa1** | **Fp1** | **Fp2** | **Fa2** |
| ФНЧ | 60 |  |  | 0,4 | 0,6 |

Исходный код:  
 .mmregs

.def \_c\_int00

.text

\_c\_int00:

N\_DBC .set 64

K .set 11

N\_poli .set 258

GAR .set 64

N .word 1

step .word 64

ssbx OVM

stm #sinus, AR3

st #N\_poli-2,BK

stm #0,AR0

stm #GAR-1, AR7

new\_polyharmonic:

stm #polyharmonic, AR5

addm #1, AR0

stm #N\_poli-1,BRC

rptb new\_harmonic

ld \*AR3+0%,-4,A

add \*AR5,A

stl A,\*AR5+

new\_harmonic:

nop

banz new\_polyharmonic,\*AR7-

nop

st #6, AR0

stm #filter-6, AR4 ;y[i]

rptz A,#6

stl A,\*AR4+

stm #polyharmonic-6, AR5 ;результат X

stm #filter-6, AR4 ;y[i]

stm #N\_poli-1, BRC

rptb IIR

RSBX OVA ; сброс бита переполнения

xor A,A

stm #koef, AR3

rpt #6

mac \*AR5+,\*AR3+, A,A ;x[i-6:i]

rpt #5

mas \*AR4+,\*AR3+, A,A ;y[i-6:i]

sfta A,3

sth A, \*AR4+ ;y[i]

ld \*AR4-0,A

ld \*AR5-0,A

IIR:

nop

xor B,B

stm #126,AR7

stm #filter+128,AR3

stm #sinus\_real, AR5

ld \*AR3+,A

;DBC

DBC:

ld #N\_DBC,A ;A=N/2, для алгоритма Рэйдера

sth B,1,AR1 ;сохранение текущей позиции

sftl B,#K ;сдвиг этой позиции, чтобы узнать ее старший бит(бит С)

bc new\_pos,NC ;если в старшем разряде 1, смотрим следующий бит

RADER:

ld A,-1,A ;алгоритм Рэйдера

sub #N\_DBC,A ;алгоритм Рэйдера

sftl B,1 ;сдвиг позиции еще на 1 разряд, чтобы узнать ее старший бит(бит С)

bc RADER,C ;если в старшем разряде 1, смотрим следующий бит

new\_pos:

add AR1,A ;узнаем новую позицию

ld A,15,B ;сохряняем новую позицию в B

stlm A,AR0 ;загружаем ее в AR0

ld \*AR5+0,A ;устанавливаем указатель в новом массиве на adr+AR0

mvdd \*AR3+,\*AR5 ;перенос значения из adr+1 старого массива в adr+AR0 нового массива

stm #sinus\_real, AR5 ;устанавливаем указатель в новом массиве на adr = 0

banz DBC,\*AR7-

nop

stm #sinus\_imagine, AR3

stm #sinus\_real, AR5

stm #SIN,AR4

stm #COS, AR2

ld #N,DP

stm #6, AR1

ld step,A

sub #1,A

stlm A,AR7

ld N,A

stlm A,AR0

sub #1,A

rpt #128

st #0,\*AR3+

stm #sinus\_imagine, AR3

nop

ld \*AR3+0,B

ld \*AR5+0,B

nop

block\_step:

stlm A,BRC

nop

rptb BPF

mpy \*AR5,\*AR2,B ;B = QR\*cosx

mac \*AR3,\*AR4,B,B ;B = (QR\*cosx+QI\*sinx)/2

sftl B,-16

ld \*AR5-0,A

ld \*AR5,-1,A

sub B,A ;A = PR/2-B/2 = QR

add \*AR5,-1,B ;B = PR/2+B/2 = PR

stl B,\*AR5+0

mpy \*AR3,\*AR2,B ;B = QI\*cosx

mas \*AR5,\*AR4,B,B ;B = (QI\*cosx-QR\*sinx)/2

sftl B,-16

stl A,\*AR5+

ld \*AR3-0,A

ld \*AR3,-1,A

sub B,A ;A = PI/2-B/2 = QI

add \*AR3,-1,B ;B = PI/2+B/2 = PI

stl B,\*AR3+0

stl A,\*AR3+

mvdm step,AR0 ;переход к следующему поворотному коэффициенту

nop

ld \*AR2+0,A

ld \*AR4+0,A

ld N,A

stlm A,AR0

BPF:

nop

nop

ld \*AR3+0,B

ld \*AR5+0,B

stm #COS, AR2

stm #SIN, AR4

sub #1,A

banz block\_step,\*AR7-

nop

ld step,A

sfta A,-1

stl A,step

sub #1,A

stlm A,AR7

ld N,A

sfta A,1

stl A,N

stlm A,AR0

stm #sinus\_real, AR5

stm #sinus\_imagine, AR3

ld \*AR3+0,B

ld \*AR5+0,B

stm #COS, AR2

stm #SIN, AR4

sub #1,A

banz block\_step,\*AR1-

nop

;AFC

stm #output, AR2

stm #sinus\_imagine+1, AR3

stm #sinus\_real+1, AR5

stm #GAR-1,AR7

AFC:

rsbx OVA ;overflow

squr \*AR5+,A ;A = ReX(k)^2

squra \*AR3+,A ;A = A + ImX(k)^2 = ReX(k)^2 + ImX(k)^2 = P(мощность сигнала на k гармонике)

sfta A,-5 ;A = A\*2 = 2\*P(-15+8+1+1)

st #0,\*AR4 ;Первое приближение корня из A

sqrt\_block:

mas \*AR4,\*AR4,A,B

bc sqrt\_find,bleq

addm #1,\*AR4

b sqrt\_block

sqrt\_find:

nop

mpy #362, A

sfta A,15

sth A,\*AR2+

banz AFC,\*AR7-

nop

;AFC

.data

.align 512

sinus .include SIN256.asm

SIN .include SIN256.asm

COS .include COS128.asm

koef .include koef.asm

.space 10\*16

polyharmonic

.space 270\*16

filter .space N\_poli\*16 ;под фильтр

sinus\_real

.space 128\*16

sinus\_imagine

.space 128\*16

output .space GAR\*16

**Вывод:** В данной лабораторной работе мы запрограммировали генератор полигармонического сигнала. Измерили АЧХ цифрового рекурсивного фильтра с помощью БПФ.