

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторные работы № 7, 8  
«Программирование полигармоники для измерения АЧХ фильтра. Измерение  
АЧХ цифрового рекурсивного фильтра с помощью БПФ»  
Вариант 12

Проверил:  
Шемаров А.И.

Выполнил:  
ст. гр. 850701  
Филипцов Д. А.

## 1 Цель работы

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).

## 2 Задание

Таблица 2.1 – Условие выполнения задания

Вариант	N (длина 1-й гармоники)	Тип и параметры фильтра					
		Тип	Aa	Fa1	Fp1	Fp2	Fa2
12	256	ФНЧ	60			0,5	0,7

## 3 Ход работы

Assembler-код:

```
.mmregs
.def _c_int00
.text

N_DBC .set 64
K .set 11
N_poli .set 258
GAR .set 64
N .word 1
step .word 64

_c_int00:
    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:
    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 C)
    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
    BC RADER,C ;если в старшем разряде 1, смотрим следующий бит

new_pos:
    ADD AR1,A ;узнаем новую позицию
    LD A,15,B ;сохраняем 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
    BANC 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
        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

```

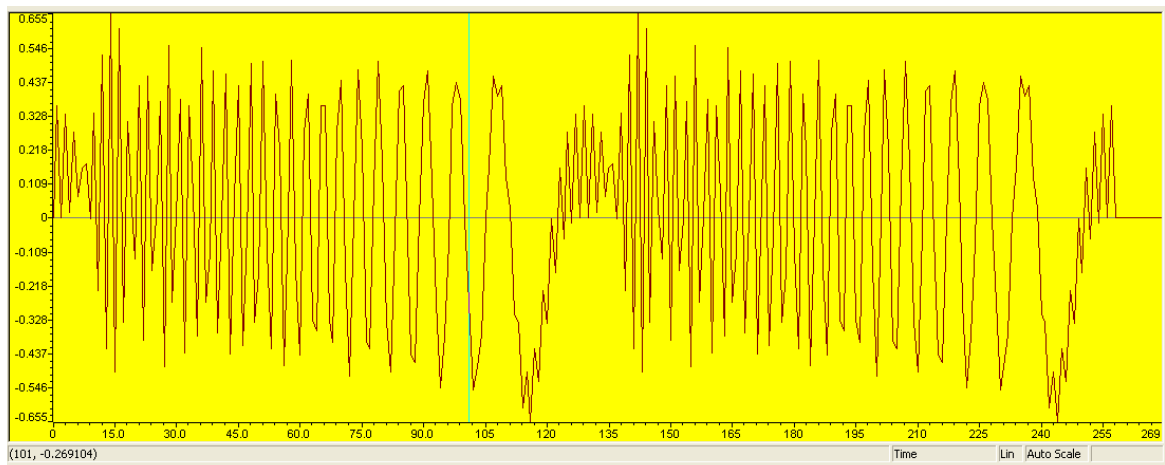
```

SFTA A,15
STH A,*AR2+
BANZ AFC,*AR7-
NOP

.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

```

**Выполнение:**



**Рисунок 3.1 – Полигармонический сигнал**

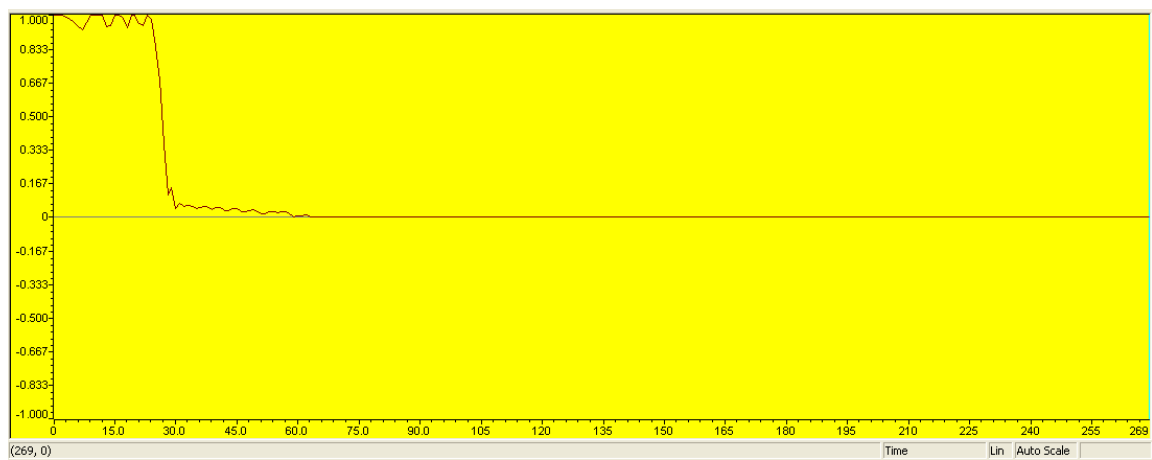


Рисунок 3.2 – АЧХ фильтра

#### 4 Вывод

В данной лабораторной работе был запрограммирован генератор полигармонического сигнала, был измерен АЧХ цифрового рекурсивного фильтра с помощью БПФ.