Министерство науки и высшего образования РФ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по лабораторной работе № 1

**Реализация КИХ-фильтра**

дисциплина «Проектирование устройств и систем на цифровых сигнальных процессорах»

|  |  |
| --- | --- |
| Выполнили: студенты группы ИНБс-5301 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Е.В. Степанов/ |
|  |  |
| Проверил: старший преподаватель кафедры РЭС | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Земцов/ |

Киров 2023

**Цель работы:** исследование основных типов весовых функций (окон);

изучение средств визуализации данных Code Composer Studio.

Текст программы:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "data1.h"

#define PI 3.141592653589793

#define max\_size 20//filter's order

#define delay 20//otschet ot nylya dlya viborki okna

int len = 100;//len if cycle

int N = 100;//len of viborka

float input\_data[100] = {A\_ARRAY};//vxodnie onscheti

float time\_Rect[100];//time window rectangle

float time\_Bartlett[100];//time window Bartlett

float time\_Hann[100];//time window Hann

float output\_data\_Rect[100];// output of rectangle filter (time)

float output\_data\_Bartlett[100];// output of Bartlett filter (time)

float output\_data\_Hann[100];// output of Hann filter (time)

float out\_f1r[100];//real spectrum out signal for rect

float out\_f1i[100];//imag spectrum out signal for rect

float out\_f2r[100];//real spectrum out signal for Bartlett

float out\_f2i[100];//imag spectrum out signal for Bartlett

float out\_f3r[100];//real spectrum out signal for Hann

float out\_f3i[100];//imag spectrum out signal for Hann

float Xr\_in[100];//real input

float Xi\_in[100];//image input

float Xr\_w1[100];//real window 1

float Xr\_w2[100];//real window 2

float Xr\_w3[100];//real window 3

float Xi\_w1[100];//image window 1

float Xi\_w2[100];//image window 2

float Xi\_w3[100];//image window 3

float input\_spectr[100];

float window1\_spectr[100];

float window2\_spectr[100];

float window3\_spectr[100];

float out\_f1[100];//spectrum out signal for rect

float out\_f2[100];//spectrum out signal for Bartlett

float out\_f3[100];//spectrum out signal for Hann

void gen\_Rect(){

int i;

for (i=0; i < N;i++){

if (i >=delay && i <max\_size+delay){

time\_Rect[i]=1;

}

else{

time\_Rect[i] = 0;

}

}

}

void gen\_Bartlett(){

int i;

for (i=0; i < max\_size;i++){

time\_Bartlett[i]=(max\_size-2\*fabs(i-max\_size/2))/max\_size;

}

for (i=0; i < N;i++){

if (i >=delay && i <delay+max\_size){

time\_Bartlett[i]=time\_Bartlett[i-delay];

}

}

for (i=0; i < N;i++){

if (i <=delay || i >delay+max\_size){

time\_Bartlett[i]=0;

}

}

}

void gen\_Hann(){

int i;

for (i=0; i < N;i++){

if (i >=delay && i <=max\_size+delay){

time\_Hann[i]=0.5-0.5\*cos(2\*PI\*i/max\_size);

}

else{

time\_Hann[i]=0;

}

}

}

void do\_window1(){

int i;

for (i=delay;i<delay+max\_size;i++){

output\_data\_Rect[i] = input\_data[i] \* time\_Rect[i];

}

}

void do\_window2(){

int i;

for (i=delay;i<delay+max\_size;i++){

output\_data\_Bartlett[i] = input\_data[i] \* time\_Bartlett[i];

}

}

void do\_window3(){

int i;

for (i=delay;i<delay+max\_size;i++){

output\_data\_Hann[i] = input\_data[i] \* time\_Hann[i];

}

}

void DFT(){

int k,n=0;

for (k = 0; k < N; k++) {

Xr\_w1[k] = 0;

Xi\_w1[k] = 0;

Xr\_w2[k] = 0;

Xi\_w2[k] = 0;

Xr\_w3[k] = 0;

Xi\_w3[k] = 0;

Xr\_in[k] = 0;

Xi\_in[k] = 0;

for (n = 0; n < N; n++) {

Xr\_in[k]= (Xr\_in[k] + input\_data[n] \* cos(2 \* PI \* k \* n / N));//spectrum input data

Xi\_in[k]= (Xi\_in[k] - input\_data[n] \* sin(2 \* PI \* k \* n / N));//

}

input\_spectr[k] = sqrt(Xr\_in[k]\*Xr\_in[k]+Xi\_in[k]\*Xi\_in[k]);

for (n = 0; n < len; n++) {//1st window

Xr\_w1[k]= (Xr\_w1[k] + time\_Rect[n] \* cos(2 \* PI \* k \* n / N));//spectrum 1st window

Xi\_w1[k]= (Xi\_w1[k] - time\_Rect[n] \* sin(2 \* PI \* k \* n / N));//

}

window1\_spectr[k]=sqrt( Xr\_w1[k]\* Xr\_w1[k]+Xi\_w1[k]\*Xi\_w1[k]);

for (n = 0; n < len; n++) {//2nd window

Xr\_w2[k]= (Xr\_w2[k] + time\_Bartlett[n] \* cos(2 \* PI \* k \* n / N));//spectrum 2nd window

Xi\_w2[k]= (Xi\_w2[k] - time\_Bartlett[n] \* sin(2 \* PI \* k \* n / N));//

}

window2\_spectr[k]=sqrt( Xr\_w2[k]\* Xr\_w2[k]+Xi\_w2[k]\*Xi\_w2[k]);

for (n = 0; n < len; n++) {//3rd window

Xr\_w3[k]= (Xr\_w3[k] + time\_Hann[n] \* cos(2 \* PI \* k \* n / N));//spectrum 3rd window

Xi\_w3[k]= (Xi\_w3[k] - time\_Hann[n] \* sin(2 \* PI \* k \* n / N));//

}

window3\_spectr[k]=sqrt( Xr\_w3[k]\* Xr\_w3[k]+Xi\_w3[k]\*Xi\_w3[k]);

for (n = 0; n < N; n++) {//spectrum output signals

out\_f1r[k]= (out\_f1r[k] + output\_data\_Rect[n] \* cos(2 \* PI \* k \* n / N));//spectrum 1st output

out\_f1i[k]= (out\_f1i[k] - output\_data\_Rect[n] \* sin(2 \* PI \* k \* n / N));//

out\_f2r[k]= (out\_f2r[k] + output\_data\_Bartlett[n] \* cos(2 \* PI \* k \* n / N));//spectrum 2nd output

out\_f2i[k]= (out\_f2i[k] - output\_data\_Bartlett[n] \* sin(2 \* PI \* k \* n / N));//

out\_f3r[k]= (out\_f3r[k] + output\_data\_Hann[n] \* cos(2 \* PI \* k \* n / N));//spectrum 3rd output

out\_f3i[k]= (out\_f3i[k] - output\_data\_Hann[n] \* sin(2 \* PI \* k \* n / N));//

}

out\_f1[k] = sqrt(out\_f1r[k]\*out\_f1r[k]+out\_f1i[k]\*out\_f1i[k]);//abs 1st out

out\_f2[k] = sqrt(out\_f2r[k]\*out\_f2r[k]+out\_f2i[k]\*out\_f2i[k]);//abs 2nd out

out\_f3[k] = sqrt(out\_f3r[k]\*out\_f3r[k]+out\_f3i[k]\*out\_f3i[k]);//abs 3rd out

}

}

void calculate(){

gen\_Rect();//Window

do\_window1();

gen\_Bartlett();//Bartlett

do\_window2();

gen\_Hann();//Hann

do\_window3();

DFT();

}

void main()

{

calculate();

}

**1) Изучение прямоугольного окна**

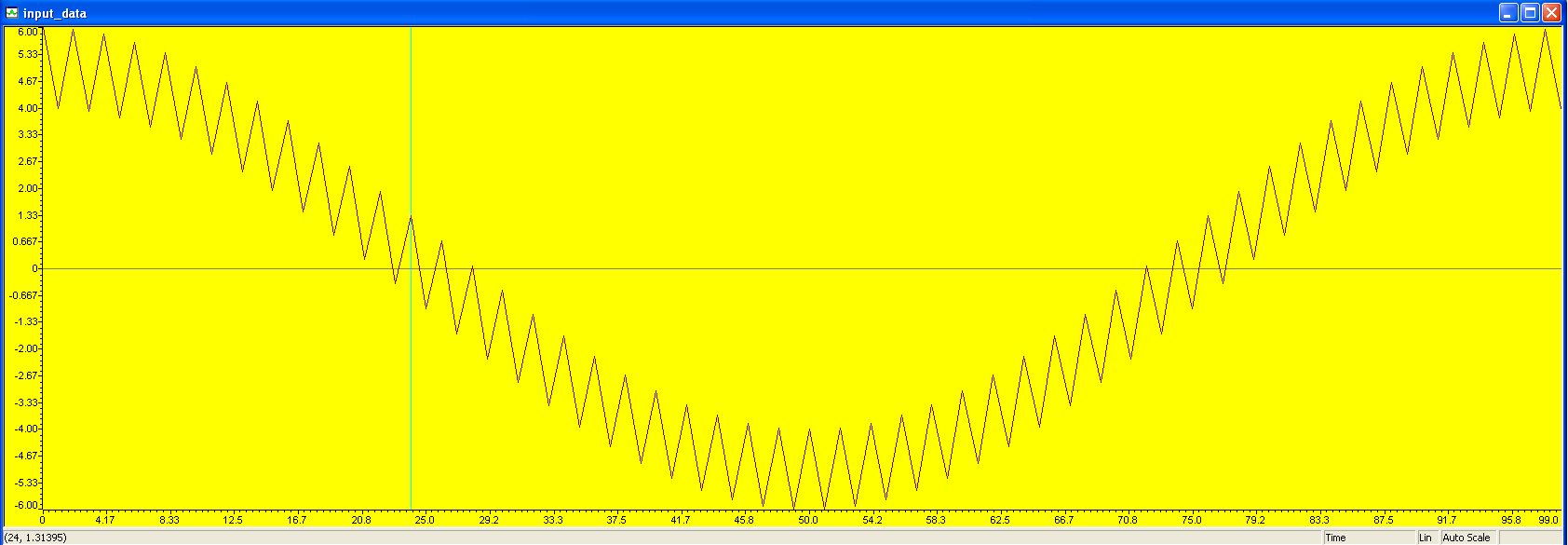


Рисунок 1 – Входные данные (временная область)

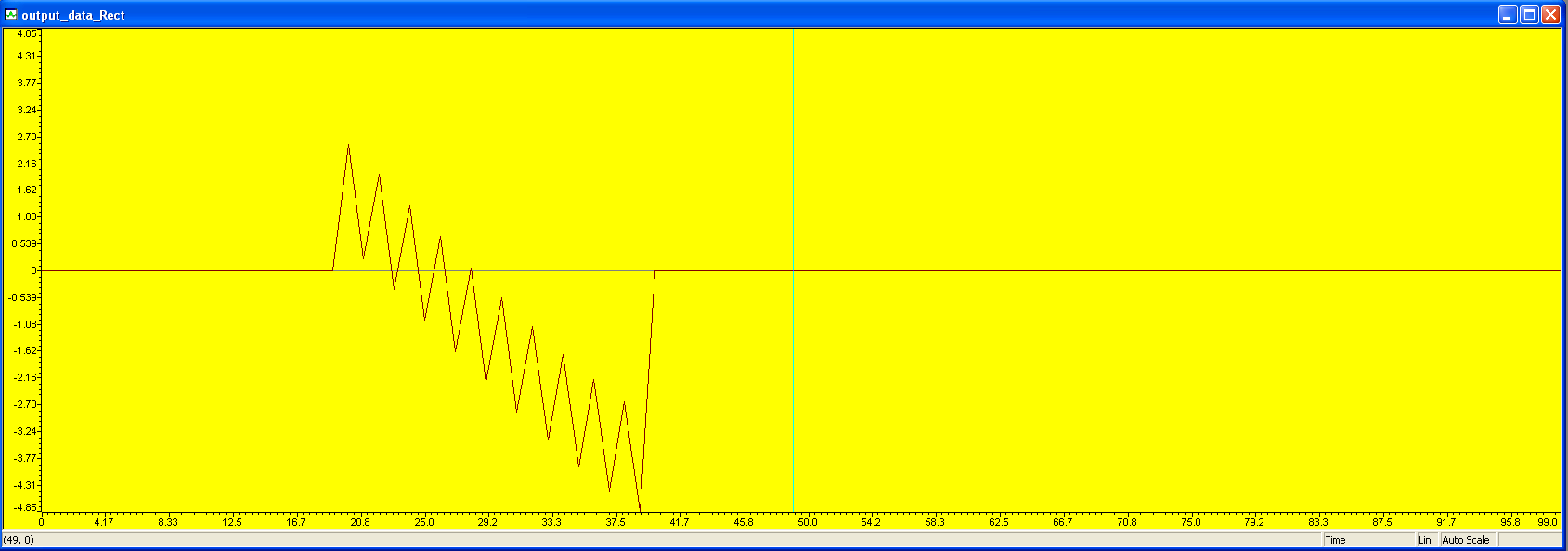


Рисунок 2 – Выходные данные (временная область)

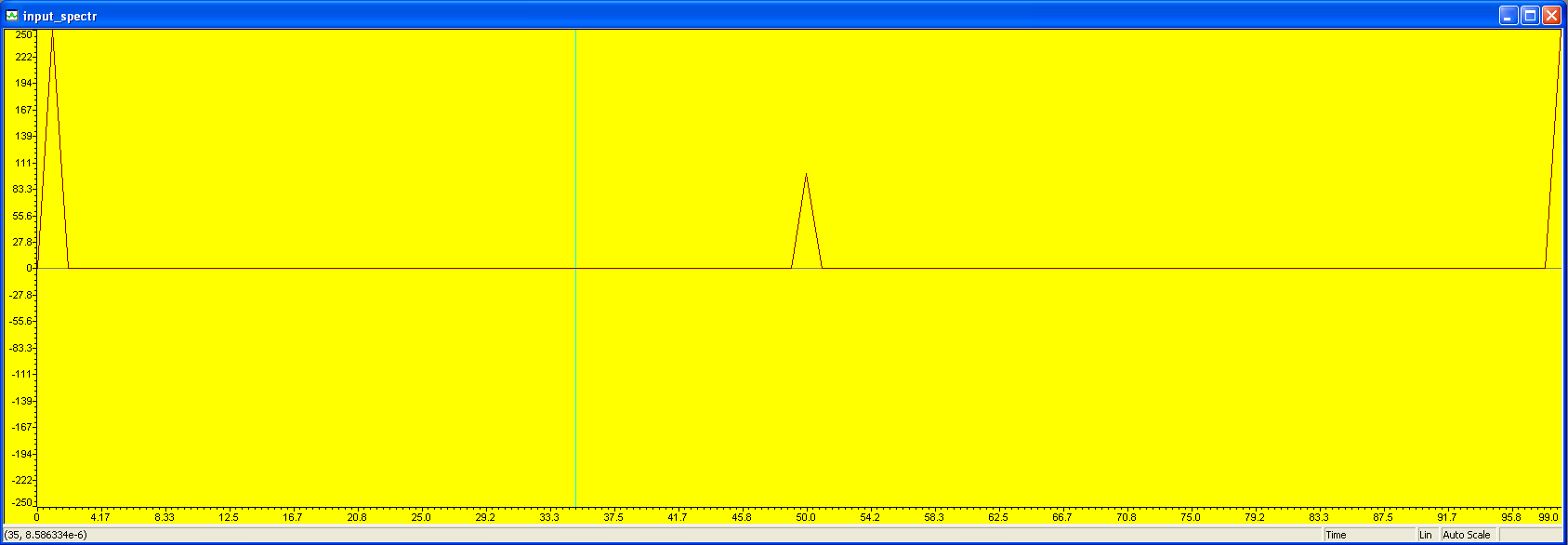


Рисунок 3 – Спектр входного сигнала (в разах)

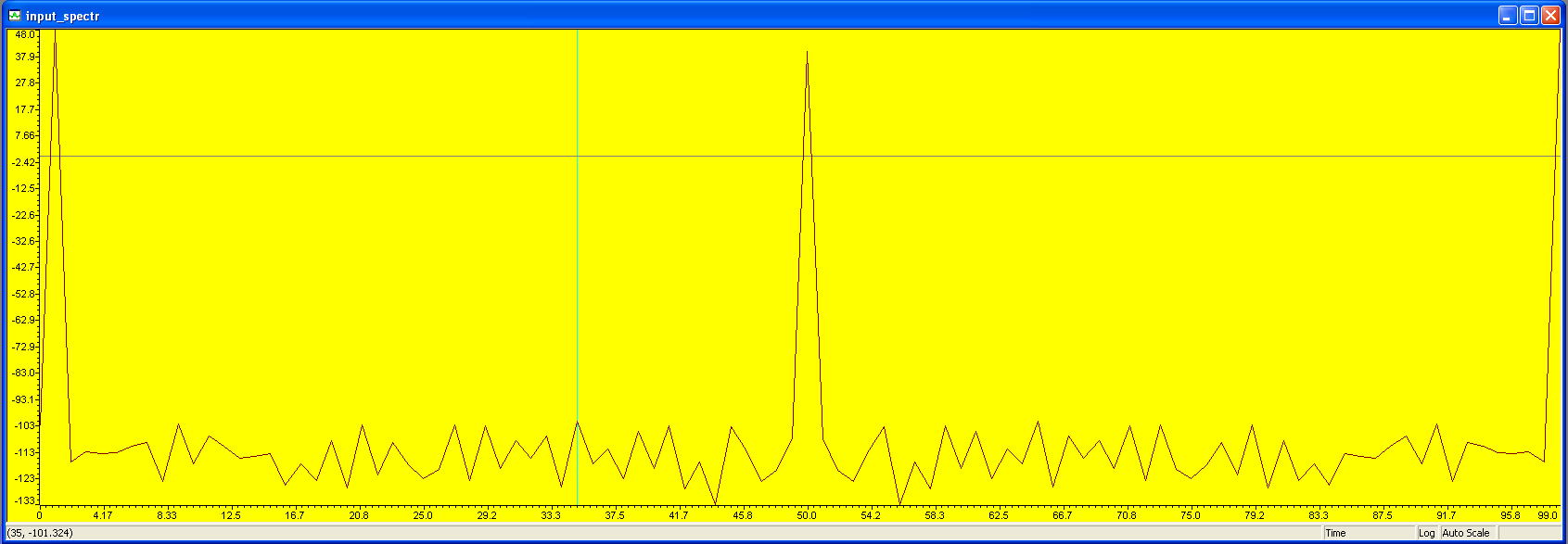


Рисунок 4 – Спектр входного сигнала (в дБ)

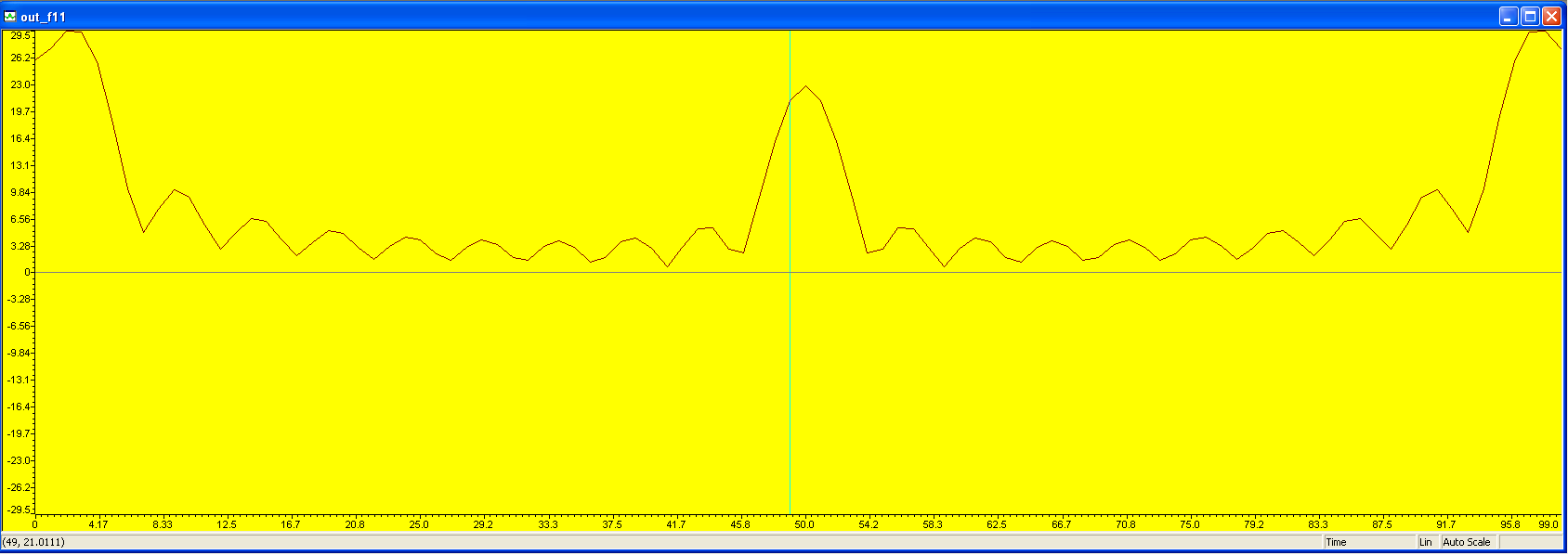


Рисунок 5 – Спектр выходного сигнала (в разах)

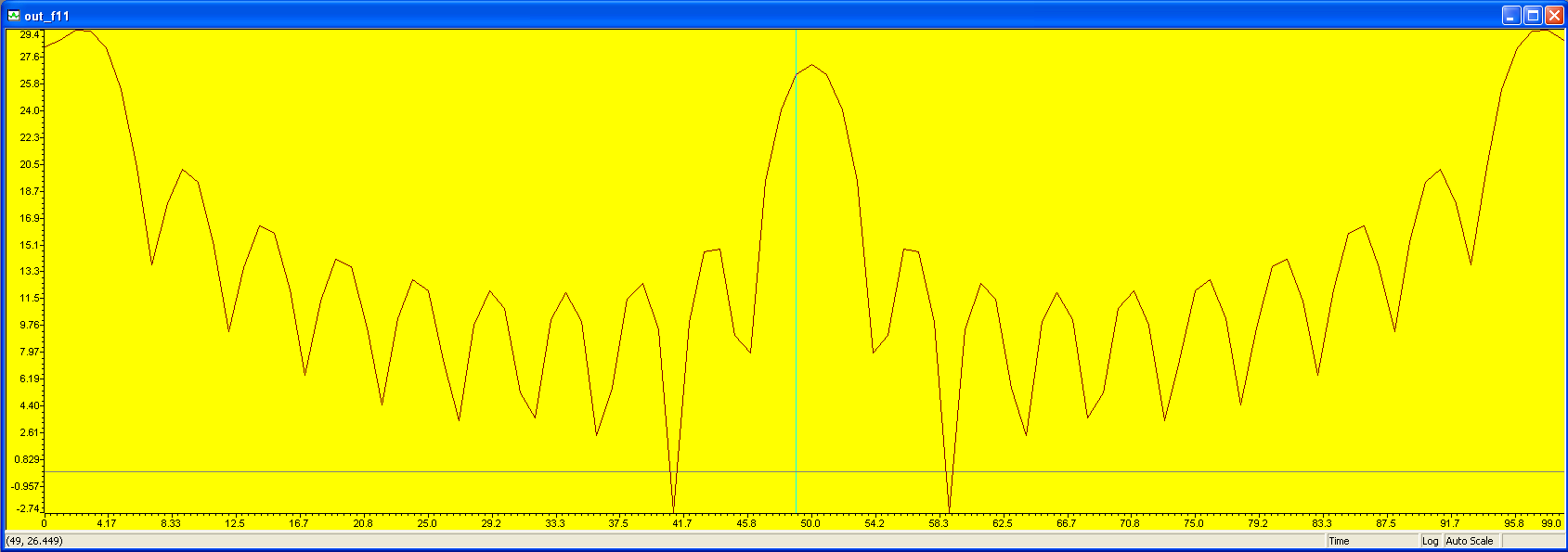


Рисунок 6 – Спектр выходного сигнала (в дБ)

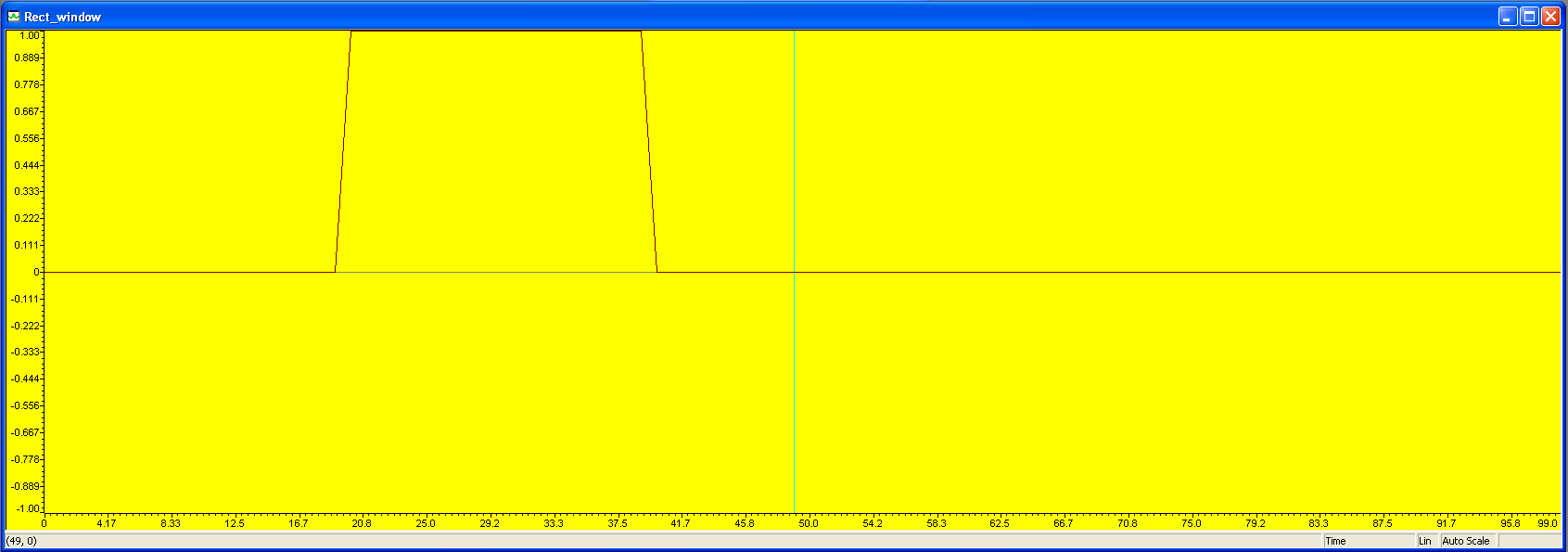
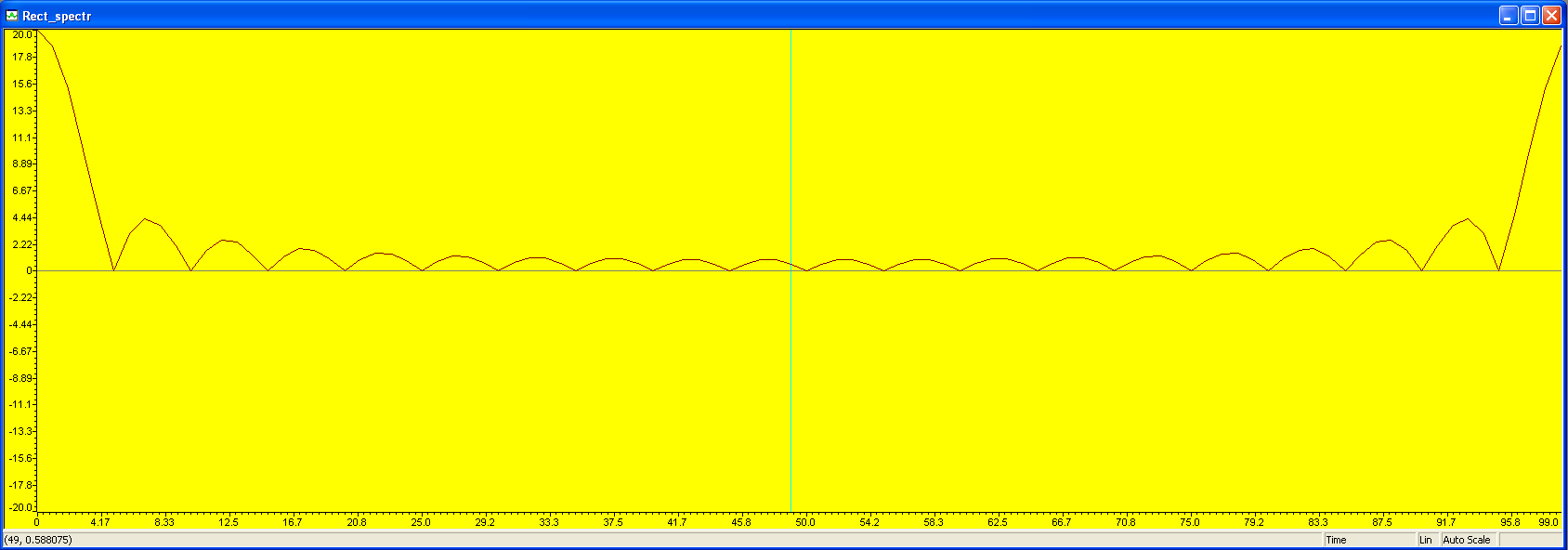


Рисунок 7 – График прямоугольного окна (временная область)



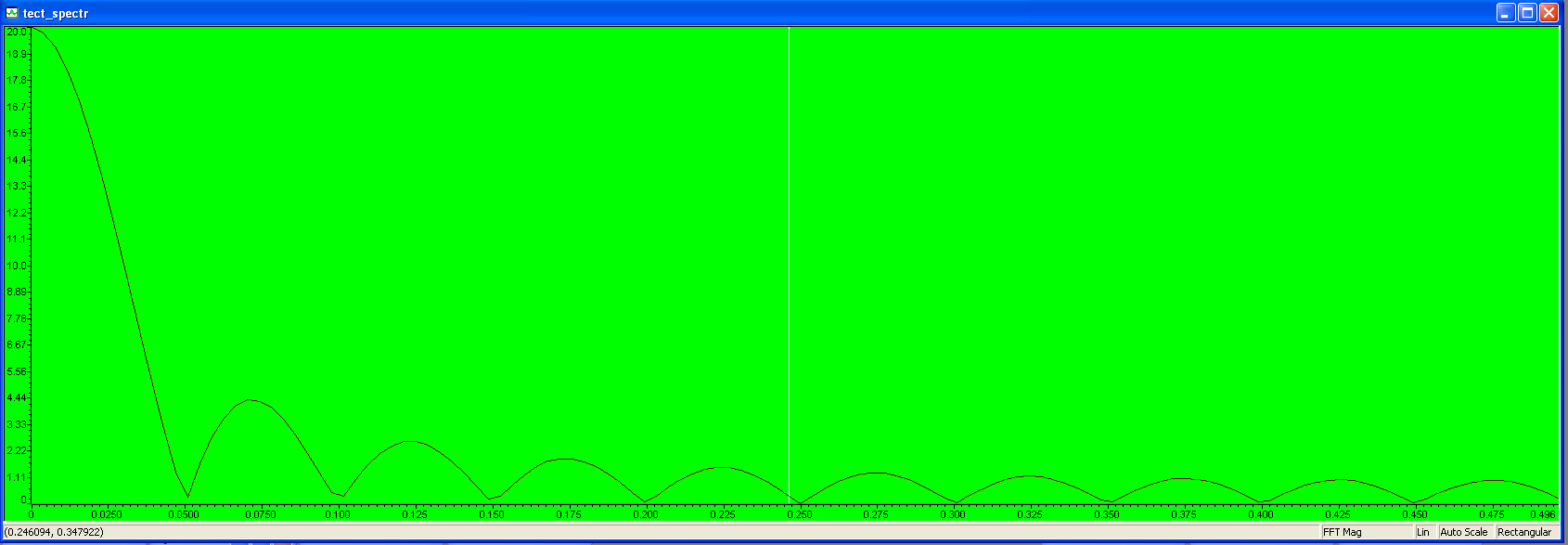


Рисунок 8 – Спектр прямоугольного окна (в разах) (желтый – результат вычисления программы, зеленый – авторассчет Code Composer Studio)

Вывод: графики спектра прямоугольного окна, рассчитанные написанной программой и внутренней функцией Code Composer Studio, сходятся. Желтый график отзеркаливается относительно центра, потому что рассчитывался на интервале 0…fs (fs - частота дискретизации), а зеленый – 0…0.5fs.

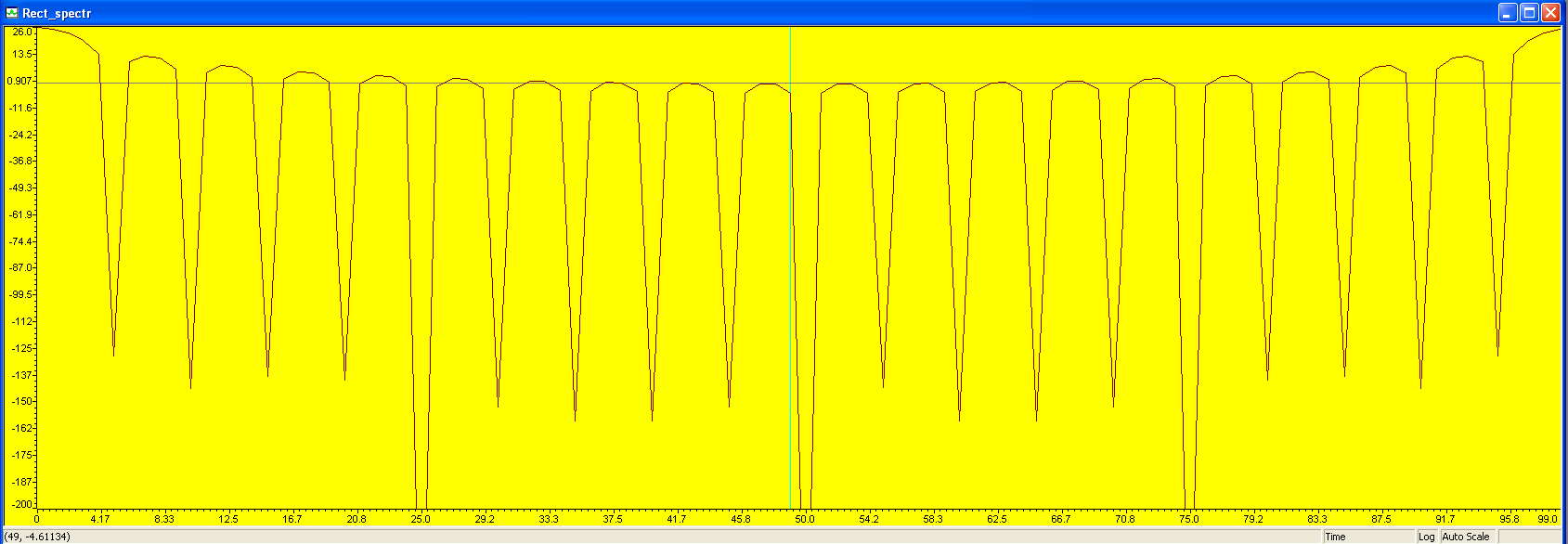


Рисунок 9 – Спектр прямоугольного окна (в дБ)

2) **Изучение окна Бартлетта**

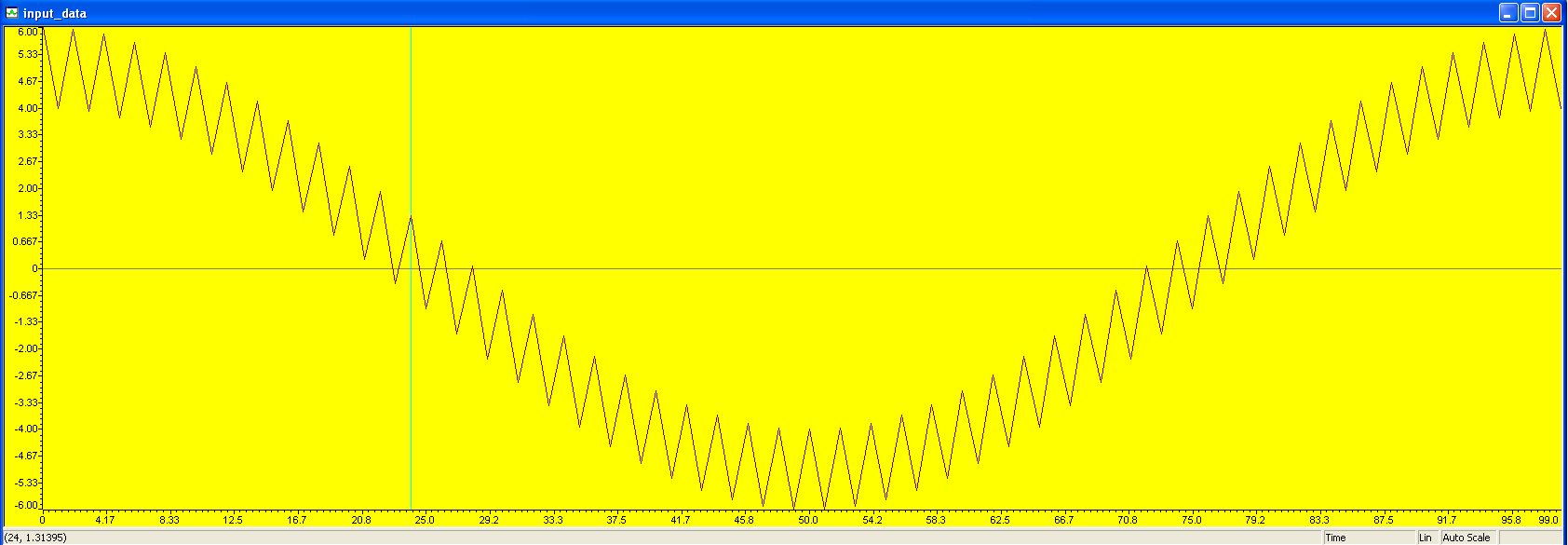


Рисунок 10 – Входные данные (временная область)

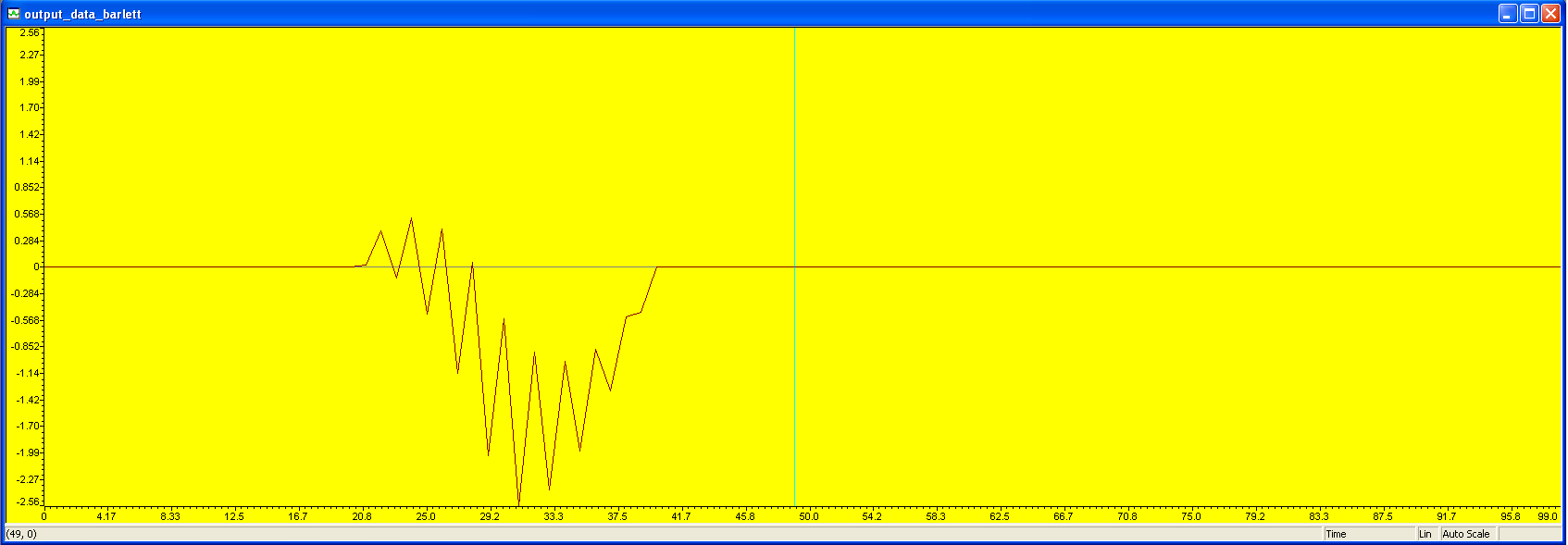


Рисунок 11 – Выходные данные (временная область)

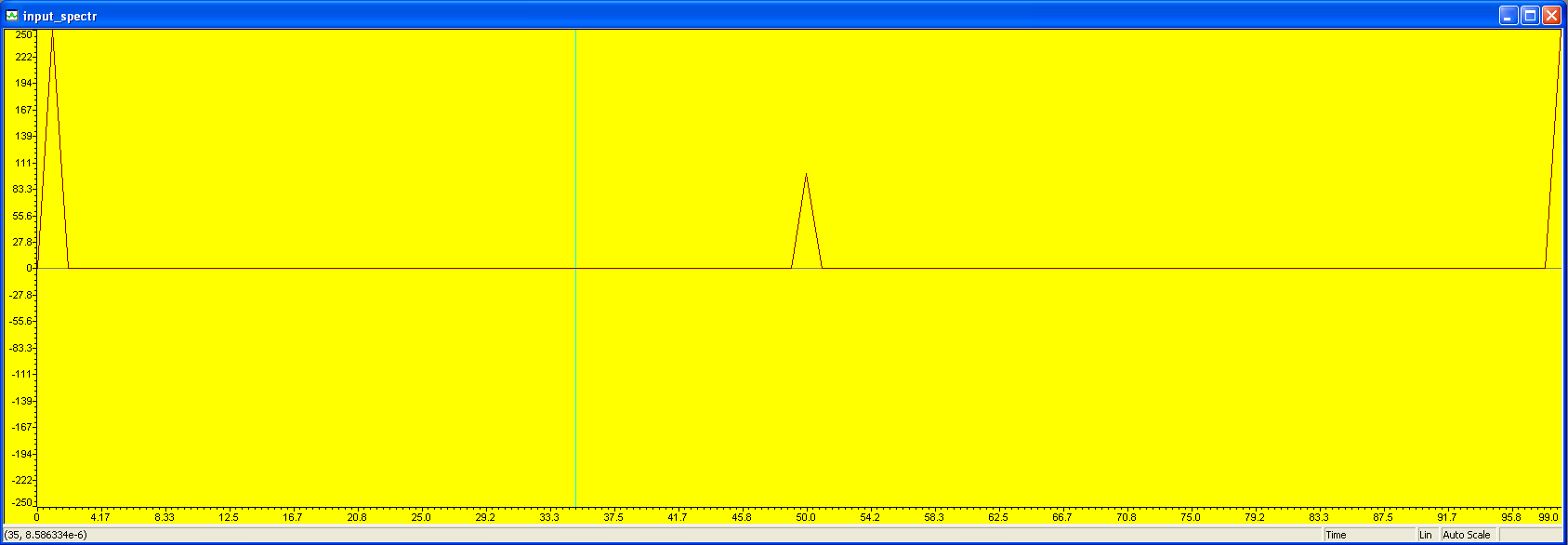


Рисунок 12 – Спектр входного сигнала (в разах)

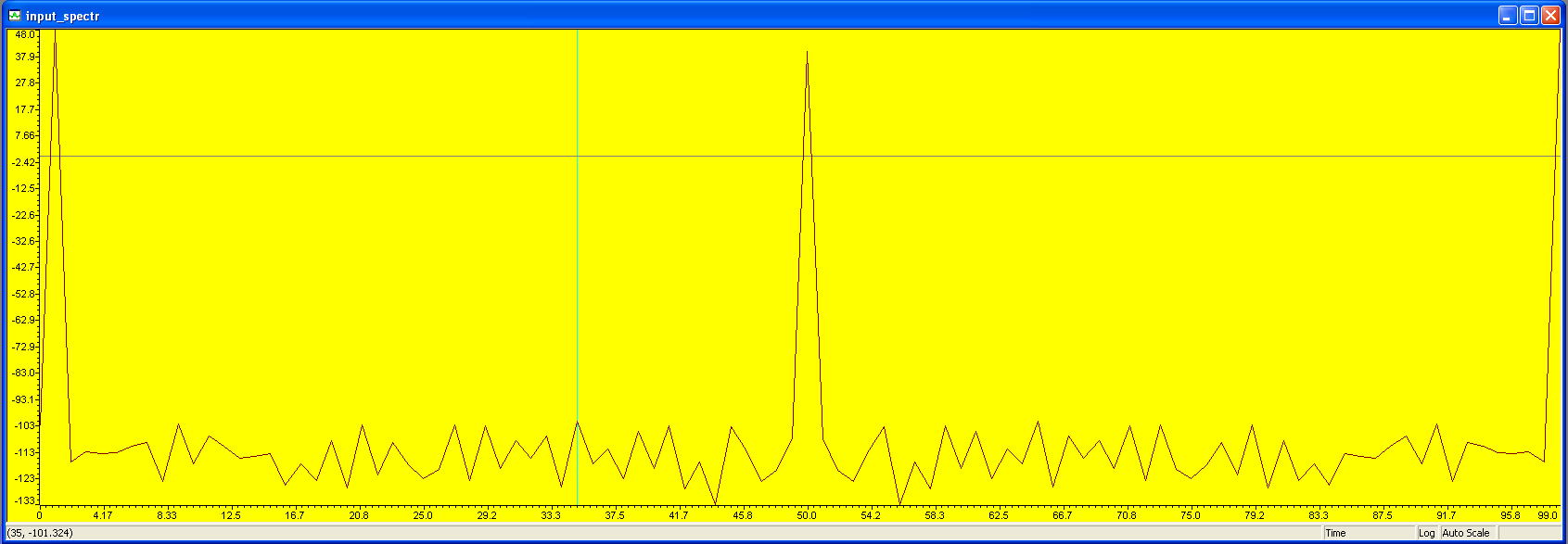


Рисунок 13 – Спектр входного сигнала (в дБ)

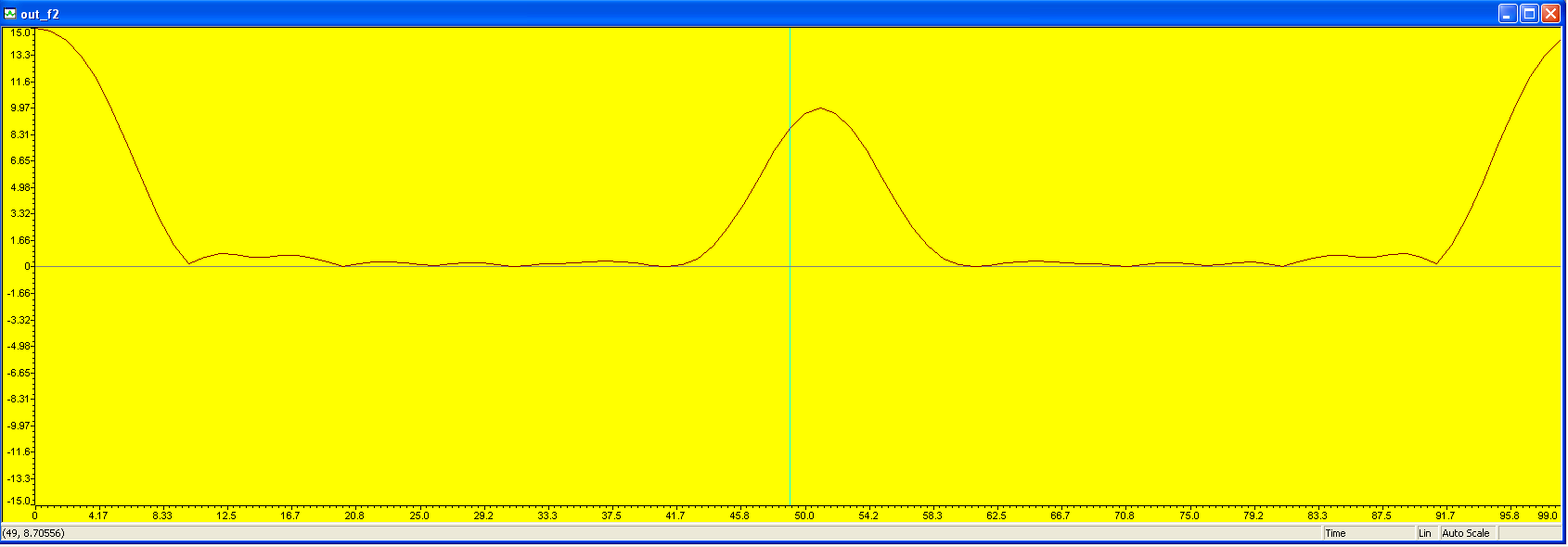


Рисунок 14 – Спектр выходного сигнала (в разах)

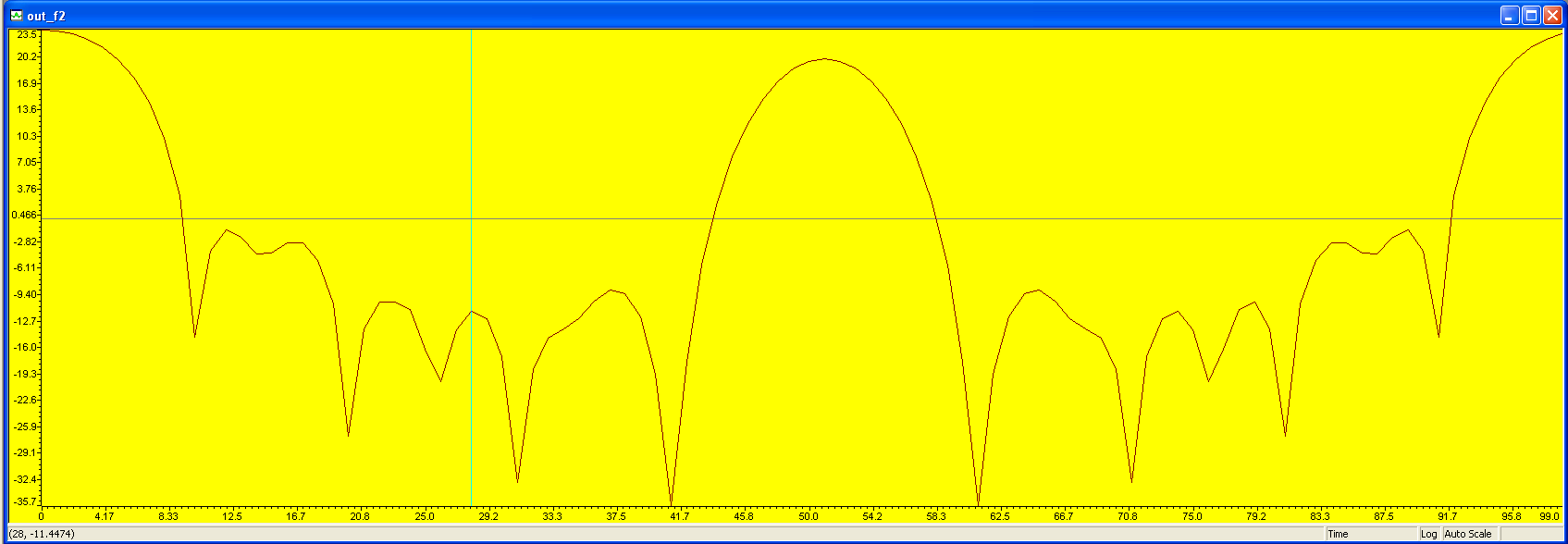


Рисунок 15 – Спектр выходного сигнала (в дБ)

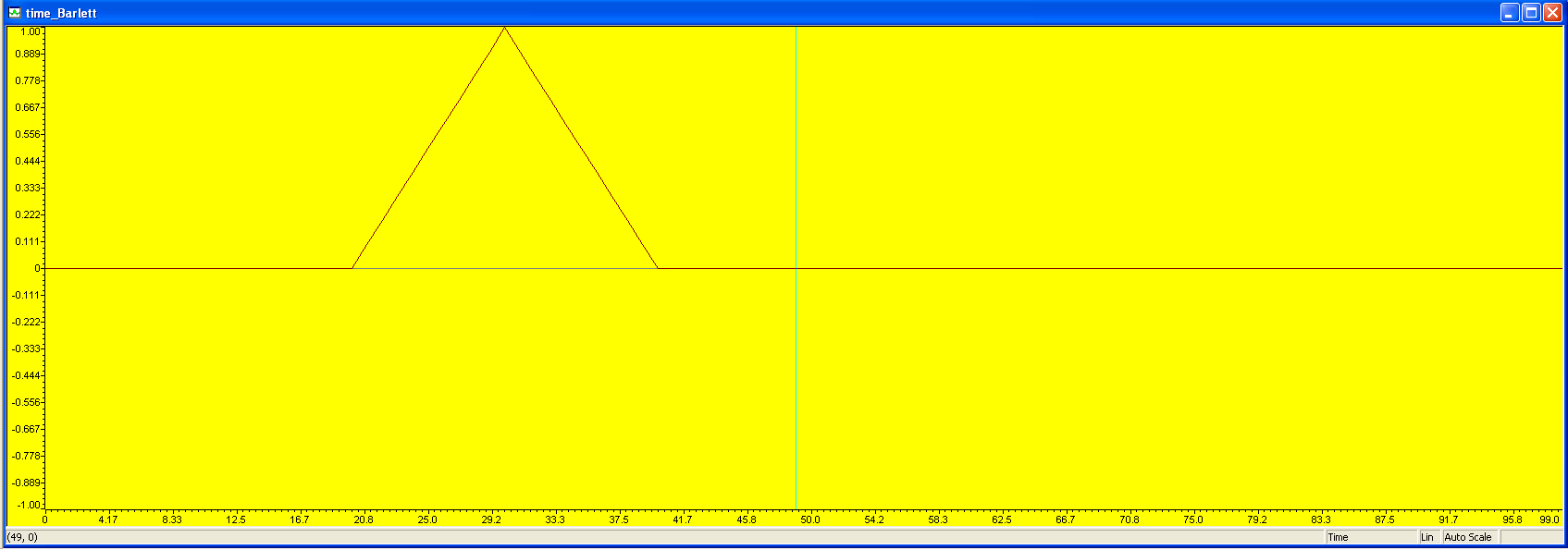
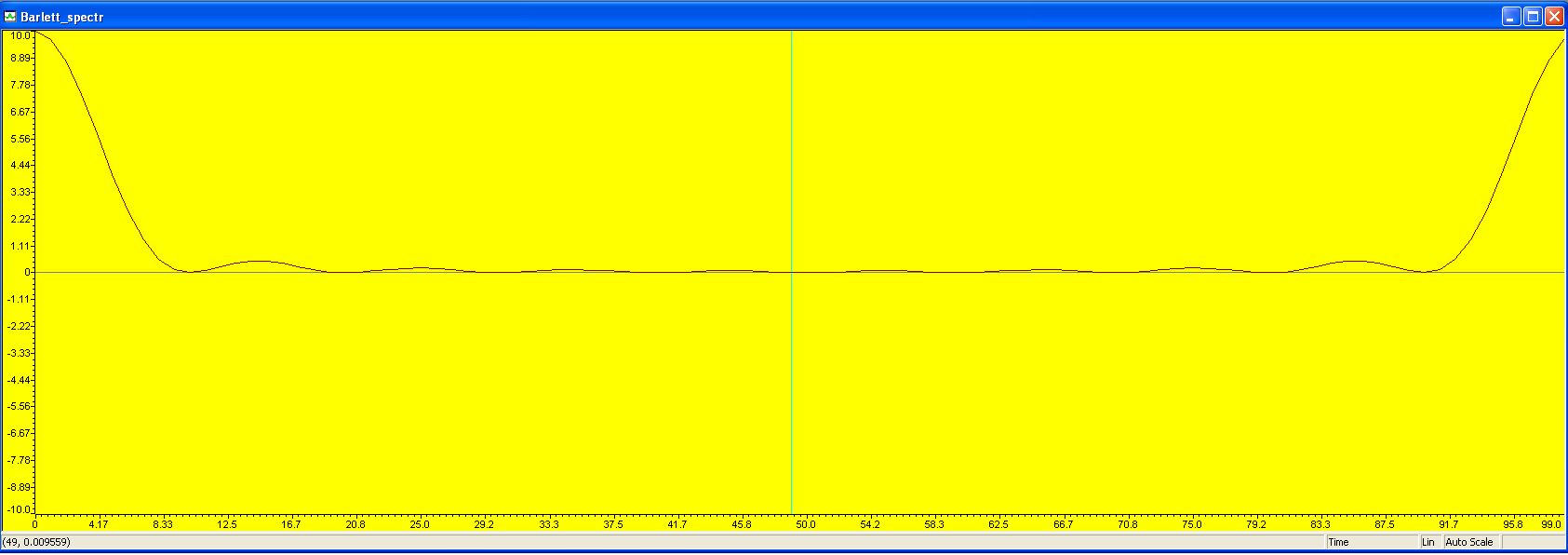


Рисунок 16 – График окна Бартлетта (временная область)



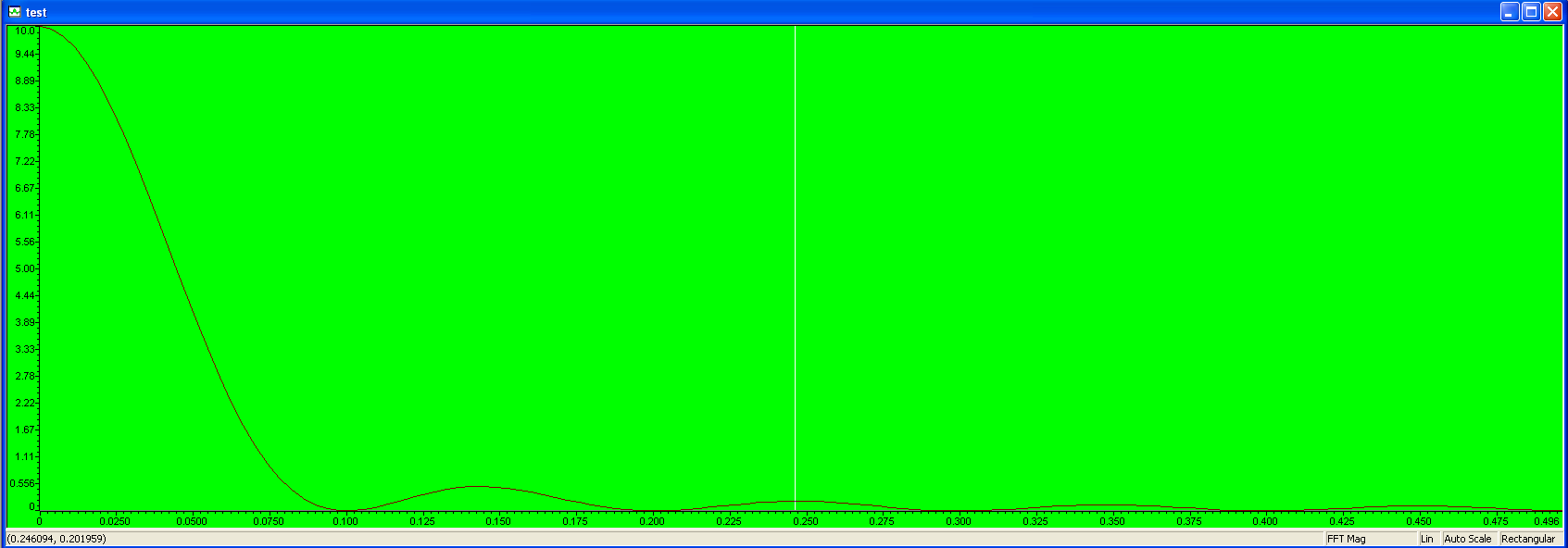


Рисунок 17 – Спектр окна Бартлетта (в разах) (желтый – результат вычисления программы, зеленый – авторассчет Code Composer Studio)

Вывод: графики спектра окна Бартлетта, рассчитанные написанной программой и внутренней функцией Code Composer Studio, сходятся. Желтый график отзеркаливается относительно центра, потому что рассчитывался на интервале 0…fs (fs - частота дискретизации), а зеленый – 0…0.5fs.

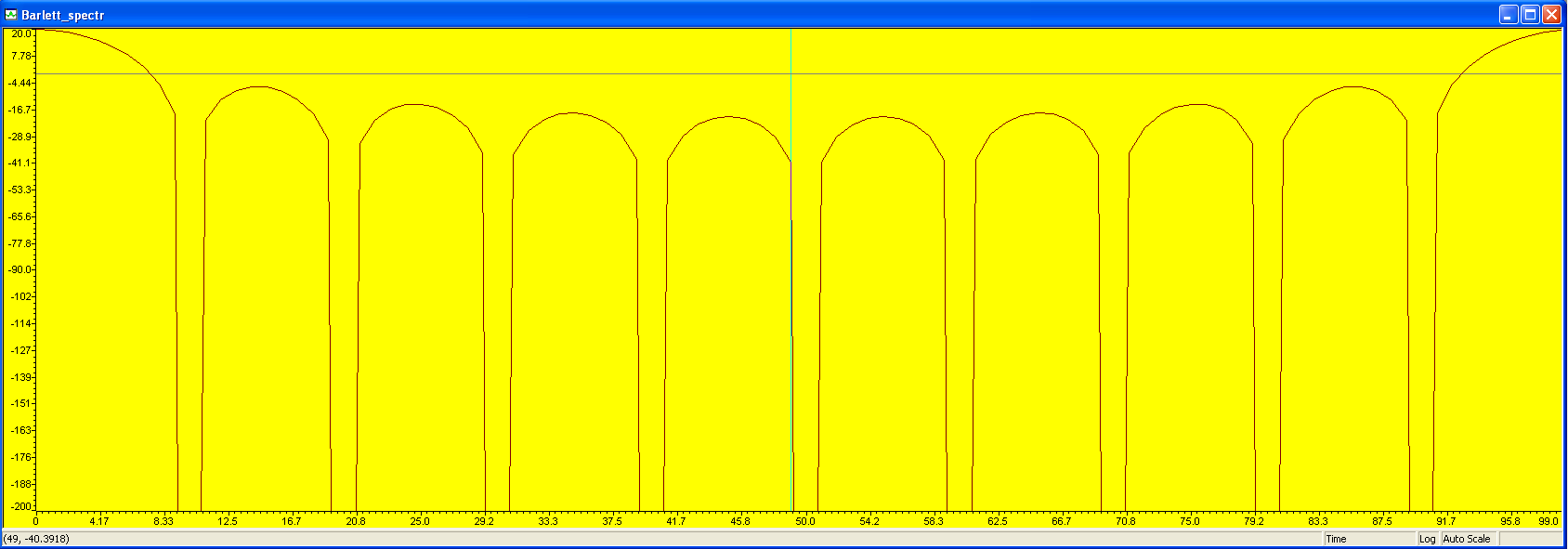


Рисунок 18 – Спектр окна Бартлетта (в дБ)

3) **Изучение окна Ханна**

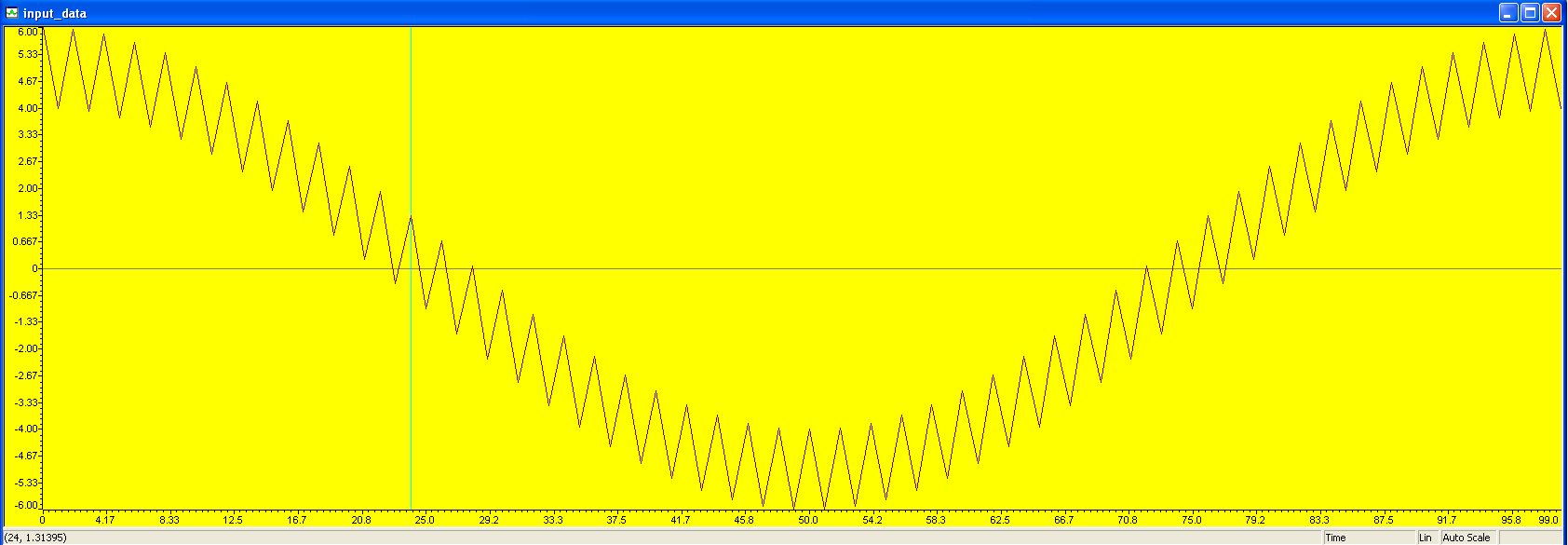


Рисунок 19 – Входные данные (временная область)

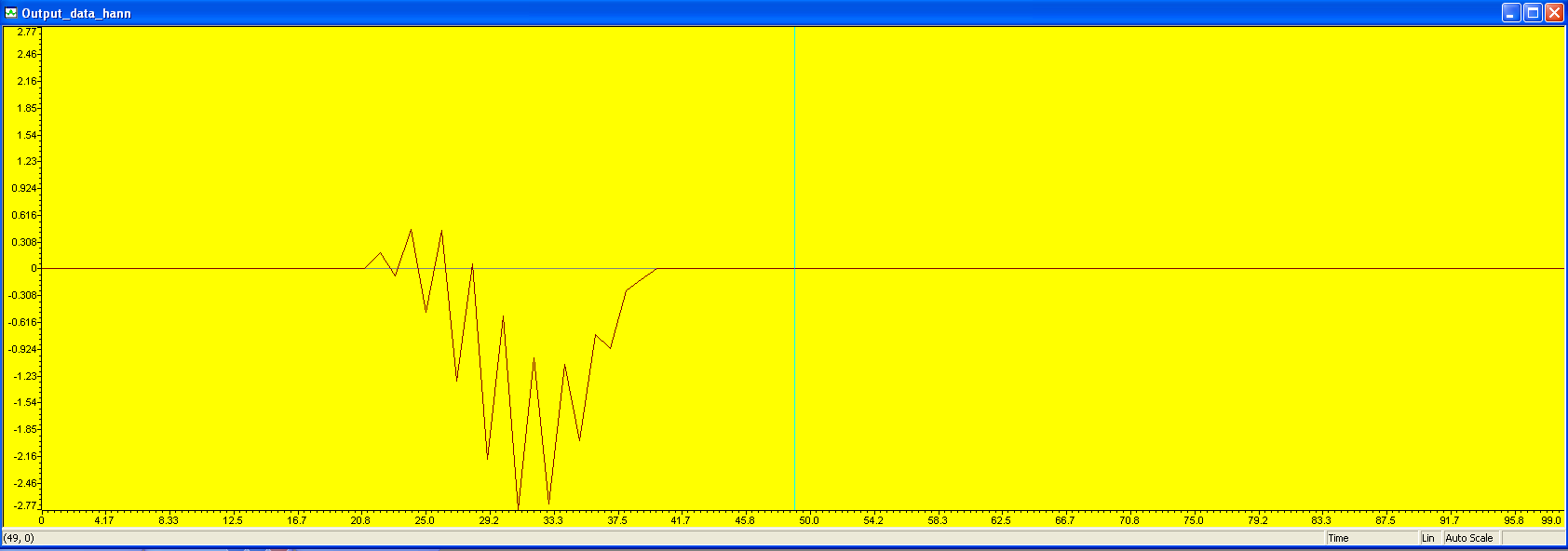


Рисунок 20 – Выходные данные (временная область)

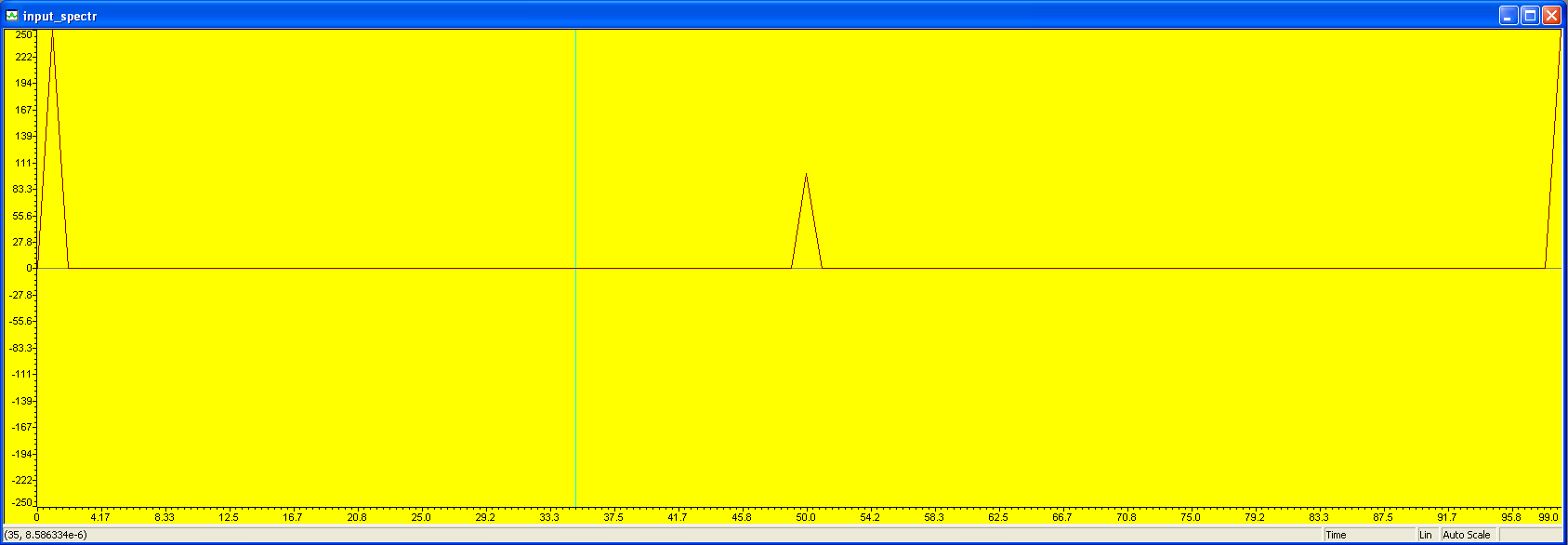


Рисунок 21 – Спектр входного сигнала (в разах)

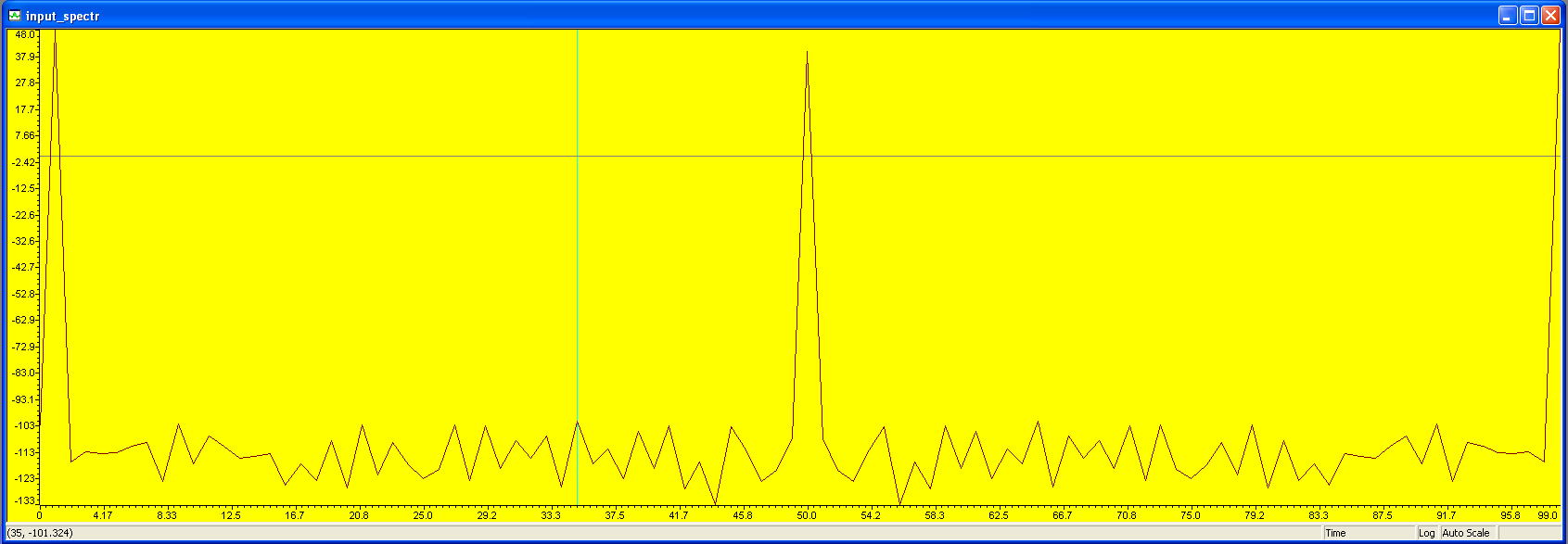


Рисунок 22 – Спектр входного сигнала (в дБ)

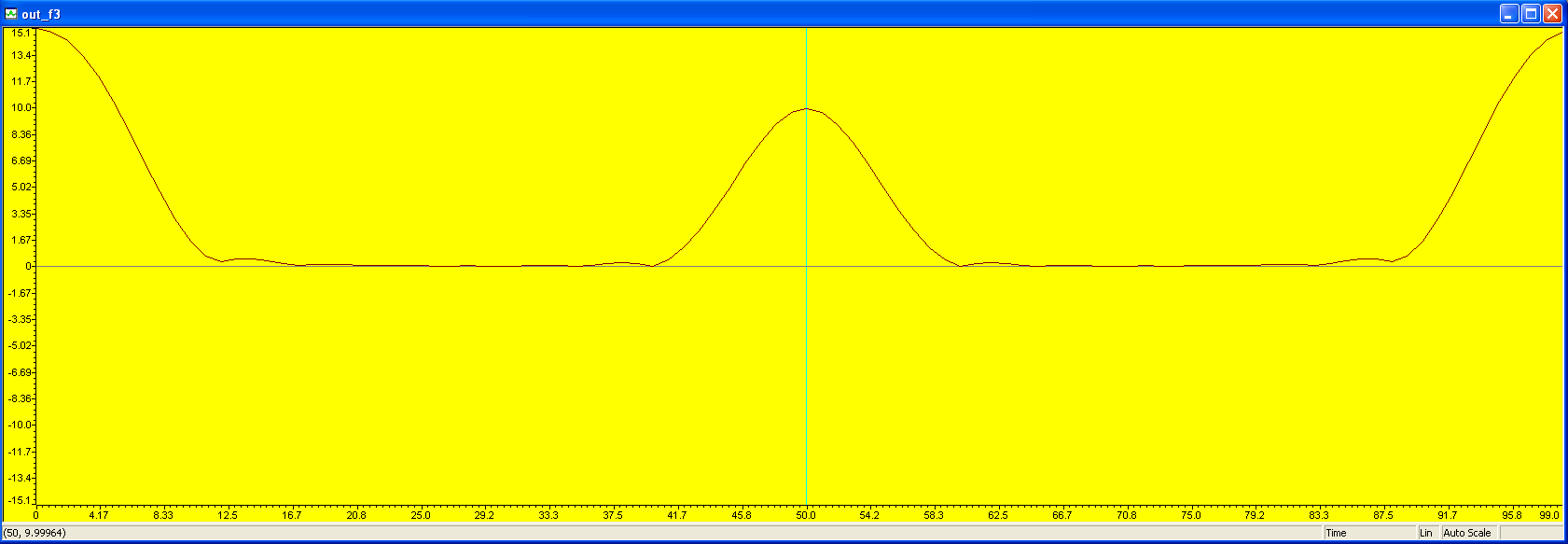


Рисунок 23 – Спектр выходного сигнала (в разах)

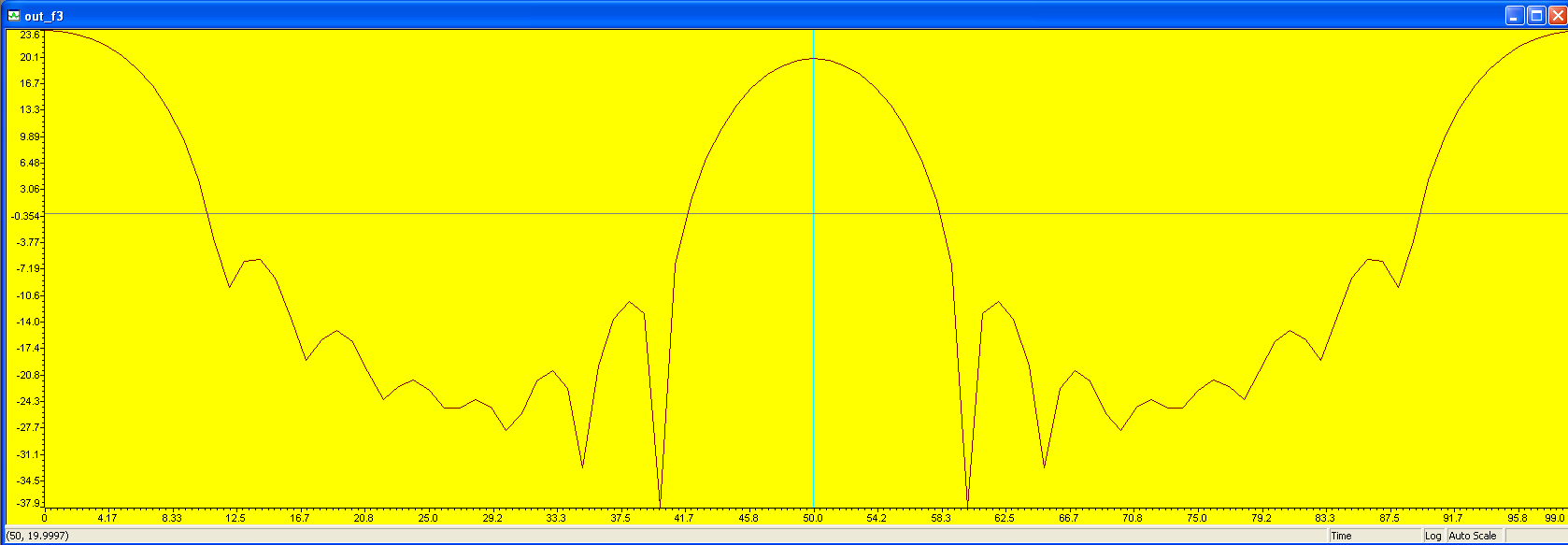


Рисунок 24 – Спектр выходного сигнала (в дБ)

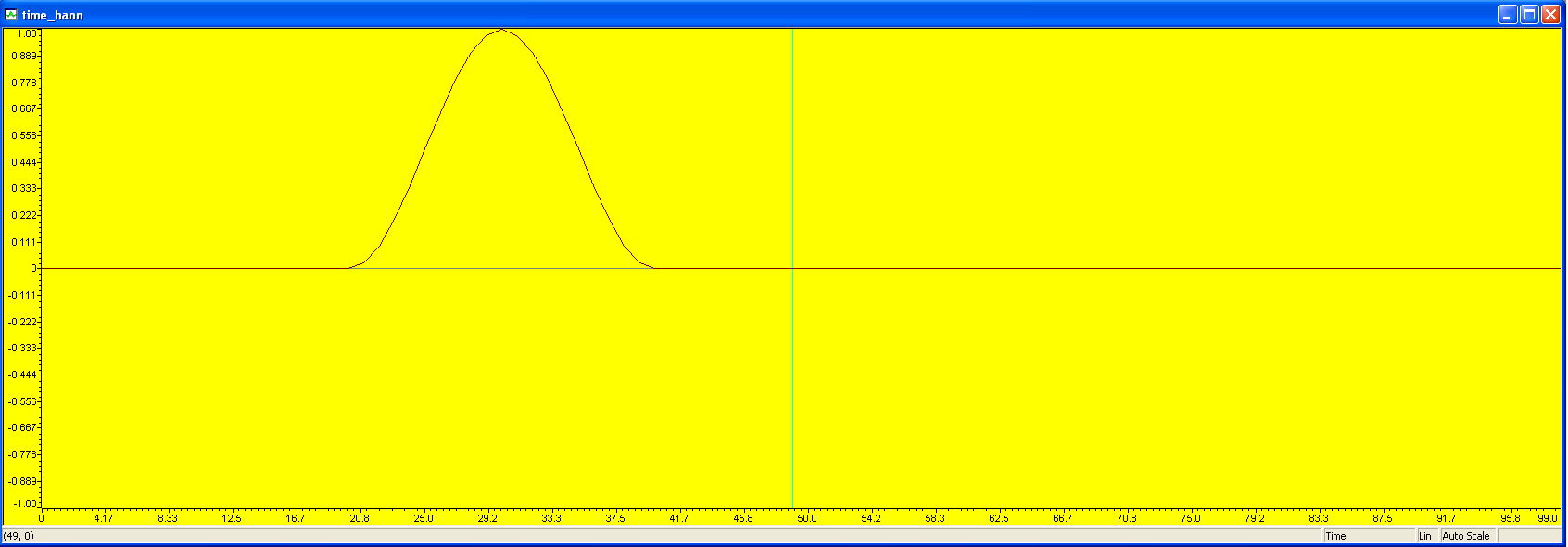
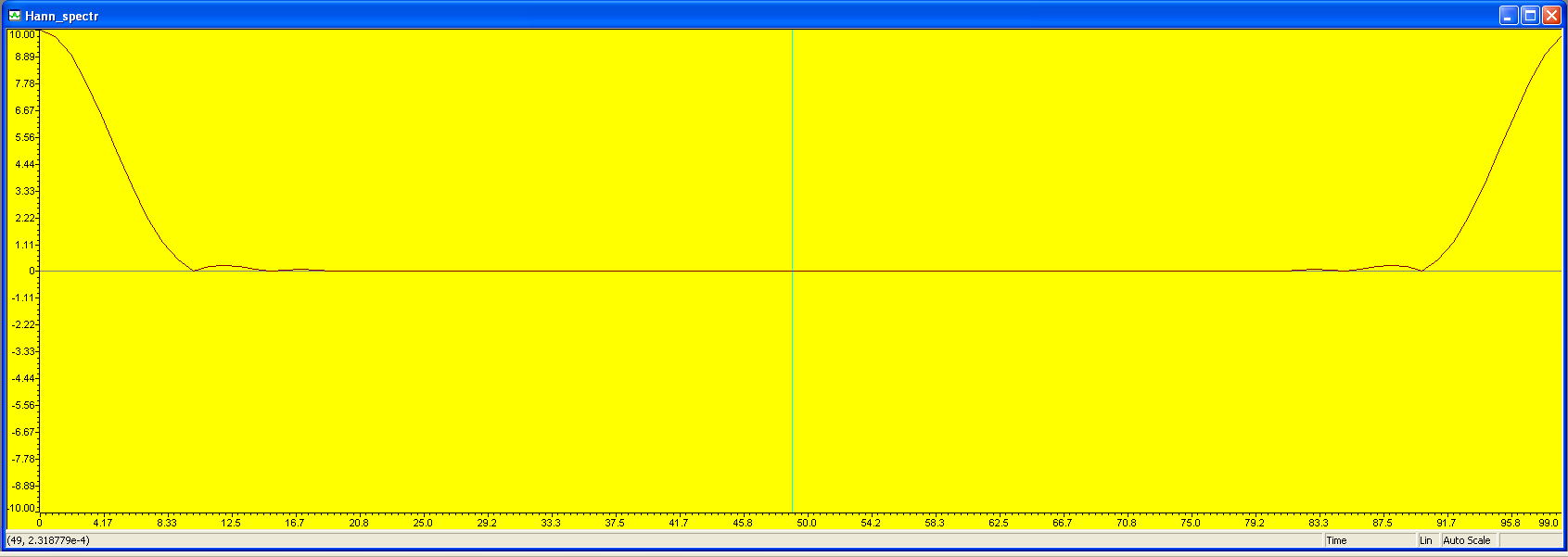


Рисунок 25 – График окна Ханна (временная область)



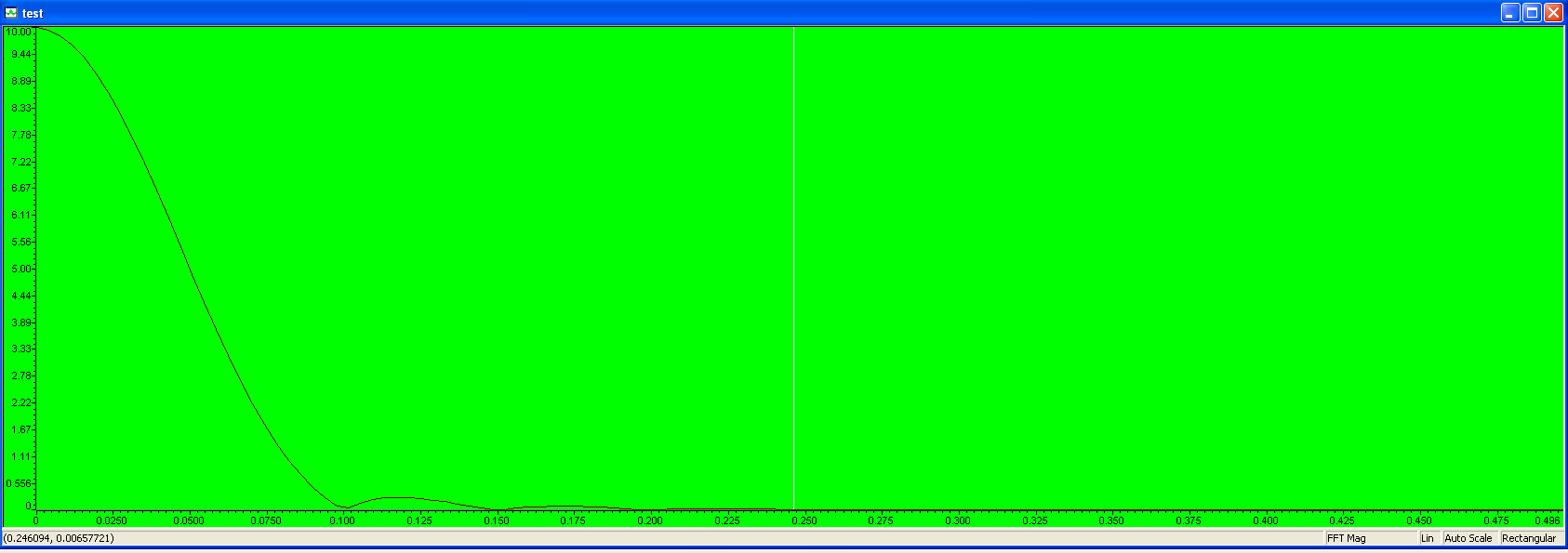


Рисунок 26 – Спектр окна Ханна (в разах) (желтый – результат вычисления программы, зеленый – авторассчет Code Composer Studio)

Вывод: графики спектра окна Ханна, рассчитанные написанной программой и внутренней функцией Code Composer Studio, сходятся. Желтый график отзеркаливается относительно центра, потому что рассчитывался на интервале 0…fs (fs - частота дискретизации), а зеленый – 0…0.5fs.

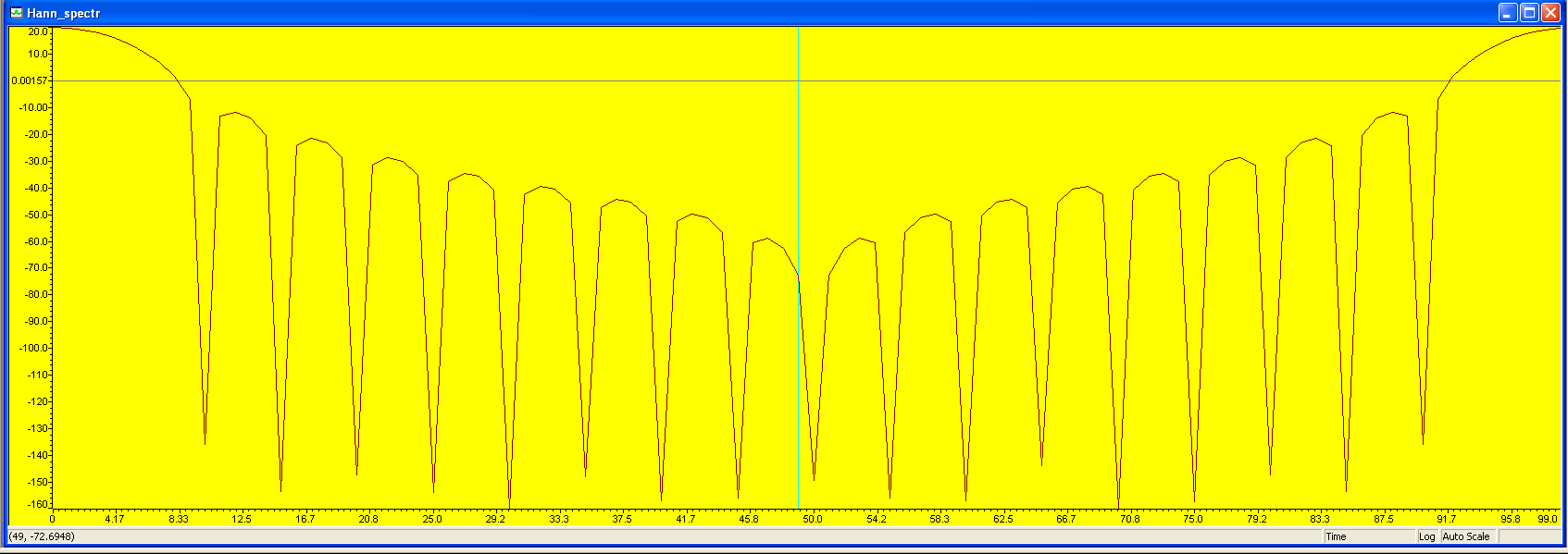


Рисунок 27 – Спектр окна Ханна (в дБ)

Выводы: острые спектральные пики ДПФ синусоиды расширились за счет воздействия копий преобразования окна. Минимальная ширина спектральных пиков взвешенной окном последовательности равна ширине главного лепестка. Боковые лепестки преобразования окна будут изменять амплитуды соседних спектральных пиков и могут маскировать присутствие слабых сигналов.

Характеристики спектрального окна ДПФ:

* уровень максимального бокового лепестка –13,6 дБ;
* скорость спада амплитуды боковых лепестков составляет 6 дБ/октава.

Характеристики спектрального окна Бартлетта:

* относительный уровень максимального бокового лепестка –26,5 дБ;
* скорость спада боковых лепестков 12 дБ/октава.
* Пониженный уровень минимального бокового лепестка и расширение главного лепестка по сравнению с прямоугольным окном.

Характеристики спектрального окна Ханна:

* уровень ближайших боковых лепестков –32 дБ;
* скорость спада амплитуды боковых лепестков 18 дБ/октава.

Наибольшей шириной полосы пропускания по уровню -3 дБ обладает окно Ханна.