

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук, доц.

должность, уч. степень, звание

подпись, дата

О.О. Жаринов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

РАЗРАБОТКА РЕКУРСИВНОГО ЦИФРОВОГО ФИЛЬТРА НА ПЛИС

по курсу: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4143

подпись, дата

Е.Д. Тегай

инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Изучить методологию расчёта цифровых фильтров и способ их реализации на ПЛИС с использованием языков описания аппаратуры.

Индивидуальное задание

Всего имеется 60 вариантов заданий. Данные вариантов заданий с 1 по 20 приведены в таблице 1. Варианты 21 – 40 и 41 - 60 отличаются от соответствующих им вариантов 1- 20 (принцип соответствия очевиден) только типом фильтра-прототипа. Для вариантов 1 – 20 следует использовать прототип Баттерворта, для вариантов 21- 40 – прототип Бесселя, для вариантов 41- 60 – прототип Чебышева.

Значение периода дискретизации T_{Δ} выбирается студентом самостоятельно, так, чтобы частота дискретизации превышала бы верхнюю частоту среза фильтра как минимум на порядок. В принципе, использованное в примере, рассмотренном в теоретической части методических указаний, значение $T_{\Delta} = 2$ мкс подходит для всех вариантов.

Задание на лабораторную работу включается 2 части:

1) расчет и моделирование фильтра заданного типа: вычисление значений коэффициентов уравнения фильтра, построение графика АЧХ и переходной характеристики;

2) реализация алгоритма рассчитанного цифрового фильтра на ПЛИС в виде проекта на языке Verilog и получение переходной характеристики.

Для выполнения расчетов рекомендуется использовать компьютерный пакет MathCAD или MatLAB. Создание и моделирование проекта фильтра для ПЛИС производится в среде Quartus.

Таблица 1

Вар.	Тип фильтра	Граничные частоты полосы пропускания фильтра	
		$f_{\text{ниж}}, \text{Гц}$	$f_{\text{верх}}, \text{Гц}$
1	полосовой фильтр	10	30
2	фильтр нижних частот	0	3000
3	режекторный фильтр	10	30
4	фильтр высоких частот	100	-
5	полосовой фильтр	100	200
6	режекторный фильтр	40	60
7	фильтр нижних частот	0	500
8	фильтр высоких частот	1000	-
9	полосовой фильтр	350	450
10	фильтр нижних частот	0	400
11	полосовой фильтр	15	25
12	фильтр нижних частот	0	500
13	режекторный фильтр	15	25
14	фильтр высоких частот	300	-
15	полосовой фильтр	1000	2000
16	режекторный фильтр	45	55
17	фильтр нижних частот	0	50
18	фильтр высоких частот	350	-
19	полосовой фильтр	35	45
20	фильтр нижних частот	0	10

*Примечание: граничные значения частот заданы в Гц, тогда как в расчетных формулах используются круговые частоты, изменяемые в рад/с. При проведении расчетов следует учитывать, что $\omega=2\pi f$.

Следует также отметить, что так как изначально был задан вариант №55, то, исходя из конечной длины таблицы 1 вариантом в рамках данной лабораторной работы является вариант №15. В качестве прототипа используется фильтр Чебышёва первого порядка.

Ход работы

В качестве языка описания аппаратуры был выбран Verilog.

Для начала стоит привести краткие теоретические сведения о задачах фильтрации сигналов и их практическом применении, а также о методах фильтрации во временной области, и рекурсивных цифровых фильтрах.

Вообще говоря, **фильтрация сигналов** – процесс обработки сигнала с целью изменения его спектральных характеристик. Фильтрация выполняет несколько задач:

- **Удаление шума.** Шум может негативно влиять на качество сигнала и усложнять его анализ. Фильтрация позволяет уменьшить влияние шума на сигнал.

- **Извлечение интересных частот.** Иногда в сигнале присутствуют несколько частотных компонентов, и нужно выделить только определённые частоты для анализа или дальнейшей обработки.

- **Улучшение качества сигнала.** Фильтрация может использоваться для улучшения качества сигнала путём удаления помех или нежелательных частот.

Фильтры имеют широкий спектр практического применения в различных областях: телекоммуникации, аудио и звукозапись, медицинская техника, радиоэлектроника, обработка изображений и видео, а также промышленная автоматизация.

Во временной области существует несколько методов фильтрации, включая:

- **Скользящее среднее.** Этот метод использует окно заданной ширины, которое перемещается по входному сигналу. Для каждой позиции окна вычисляется среднее значение отсчетов в этом окне. Этот метод эффективен для сглаживания шумов и удаления высокочастотных компонентов сигнала.

- **Экспоненциальное сглаживание.** Этот метод использует взвешенное среднее значение предыдущих отсчетов сигнала с уменьшающимся коэффициентом веса для каждого последующего отсчета. Это позволяет сгладить сигнал, сохраняя при этом его основные тренды.

- **Фильтр Калмана.** Этот метод используется для оценки состояния динамических систем на основе последовательности измерений с учетом статистических характеристик шума в системе и измерениях. Он широко

применяется в системах управления и оценке состояния для фильтрации шумов и улучшения оценок состояния системы.

- **Медианный фильтр.** Этот метод использует медианное значение отсчетов в окне заданной ширины для замены текущего значения. Он эффективен для удаления выбросов и импульсных шумов в сигналах.

- **Фильтр конечного импульсного отклика.** Этот метод основан на свертке входного сигнала с коэффициентами фильтра, которые определяются его импульсной характеристикой. Эти фильтры могут иметь линейную фазу и точно настраиваемую частотную характеристику.

Рекурсивные цифровые фильтры являются одним из типов цифровых фильтров, который использует рекурсивные связи для обработки входного сигнала. Они также известны как ИР-фильтры из-за их бесконечной импульсной характеристики. Рекурсивные фильтры часто описываются разностными уравнениями вида, как показано на рисунке 1.

$$y[k] = \sum_{m=1}^P a_m y[k-m] + \sum_{v=0}^Q b_v x[k-v],$$

Рисунок 1 – Разностное уравнение ЦРФ

где $y[k]$ - текущий выходной отсчет, $x[k]$ - текущий входной отсчет, b_v и a_m - коэффициенты фильтра, P и Q - порядки фильтра.

Рекурсивные фильтры могут иметь различные характеристики, такие как полосовые, низкочастотные, высокочастотные, полосово-пропускные и полосово-заграждающие. Кроме того, они могут быть реализованы для фильтрации сигналов как в аналоговой, так и в цифровой области.

Программа, моделирующая АЧХ и переходную характеристику

Для реализации данного пункта было решено использовать программу Matlab. Соответственно, используемый язык – MATLAB.

Программа начинается с объявления начальных параметров, а именно: частоты дискретизации, значений нижней и верхней частот (ещё пока в Гц), а также порядка фильтра. Затем, с помощью специальной функции `cheb1ar`,

идёт расчёт коэффициентов аналогового фильтра Чебышёва. В качестве первого аргумента идёт порядок фильтра, а далее – число 0.5. Это параметр, который указывает на максимальную допустимую амплитудную риппл-полосу в децибелах в полосе пропускания. Функция возвращает 3 массива: z , p и k , которые содержат нули, полюса и коэффициент усиления аналогового фильтра Чебышёва соответственно (потом эти же массивы будут использованы для создания передаточной функции фильтра).

После этого с помощью функции `zp2tf` идёт преобразование нулей, полюсов и коэффициента усиления (z , p , k соответственно) в коэффициенты передаточной функции фильтра. Функция возвращает два массива коэффициентов передаточной функции: b – массив коэффициентов числителя передаточной функции и a – массив коэффициентов знаменателя передаточной функции. Эти коэффициенты определяют форму и характеристики фильтра, которые будут использоваться для фильтрации сигнала.

Далее используется функция `lp2bp` для преобразования передаточной функции нижеполосового фильтра в передаточную функцию полосового фильтра с заданными нижней и верхней частотами. Она возвращает два массива коэффициентов новой передаточной функции полосового фильтра: bt – массив коэффициентов числителя новой передаточной функции, at – массив коэффициентов знаменателя новой передаточной функции.

Затем с помощью функции `tf2zp` идёт преобразование передаточной функции в форму нулей и полюсов. На выходе образуются массивы нулей, полюсов и коэффициентов передаточной функции полосового фильтра в новой форме.

Используя функцию `bilinear` происходит преобразование нулей и полюсов фильтра из дискретной области частот в непрерывную с использованием метода билинейного преобразования. Результатом являются новые нули, полюсы и коэффициент передаточной функции фильтра в непрерывной области частот.

После этого с помощью функции `zp2sos` идёт преобразование нулей, полюсов и коэффициентов передаточной функции в форму каскада второго порядка. Эта форма представляет собой более устойчивый способ представления фильтров. Потом идёт преобразование этого каскада в форму передаточной функции в виде коэффициентов числителя и знаменателя.

Наконец, строится график АЧХ и переходная характеристика.

Листинг программы

```
clc, clear, close all

Td = 50000;
low_freq = 1000;
high_freq = 2000;
filter_poryad = 1;

[z,p,k] = cheb1ap(filter_poryad,0.5);

[b,a] = zp2tf(z,p,k);

[bt,at] = lp2bp(b,a,low_freq*2*pi, high_freq*2*pi);

[z_new,p_new,k_new] = tf2zp(bt,at);

[zd,pd,kd] = bilinear(z_new,p_new,k_new,Td);

sos = zp2sos(zd,pd,kd);

[b,a] = sos2tf(sos);

N = 1000;
[H, F] = freqz(b,a,N,'whole',Td);
f = 0:(Td/N):Td/2;
W = H(1:length(f));

figure(1)
semilogx(f, 20*log(abs(W))./log(10)), grid % график АЧХ
ylim([-80 10])
xlim([10 2800])

for i = 1:40
    input_signal(i) = 100;
end

output_signal = filter(b,a,input_signal);
disp('Выходной сигнал:')
disp(output_signal)

figure(2)
plot(output_signal(1:40))
```

Результаты моделирования

Искомые результаты представлены на рисунках 2 – 4.

Выходной сигнал:												
Columns 1 through 12												
72.0610	89.5331	53.5032	52.4763	36.3302	31.9623	23.7793	19.8680	15.3092	12.4781	9.7802	7.8769	
Columns 13 through 24												
6.2251	4.9848	3.9552	3.1584	2.5109	2.0024	1.5934	1.2698	1.0109	0.8054	0.6413	0.5108	
Columns 25 through 36												
0.4068	0.3240	0.2581	0.2055	0.1637	0.1304	0.1038	0.0827	0.0659	0.0525	0.0418	0.0333	
Columns 37 through 40												
0.0265	0.0211	0.0168	0.0134									

Рисунок 2 – Результаты моделирования

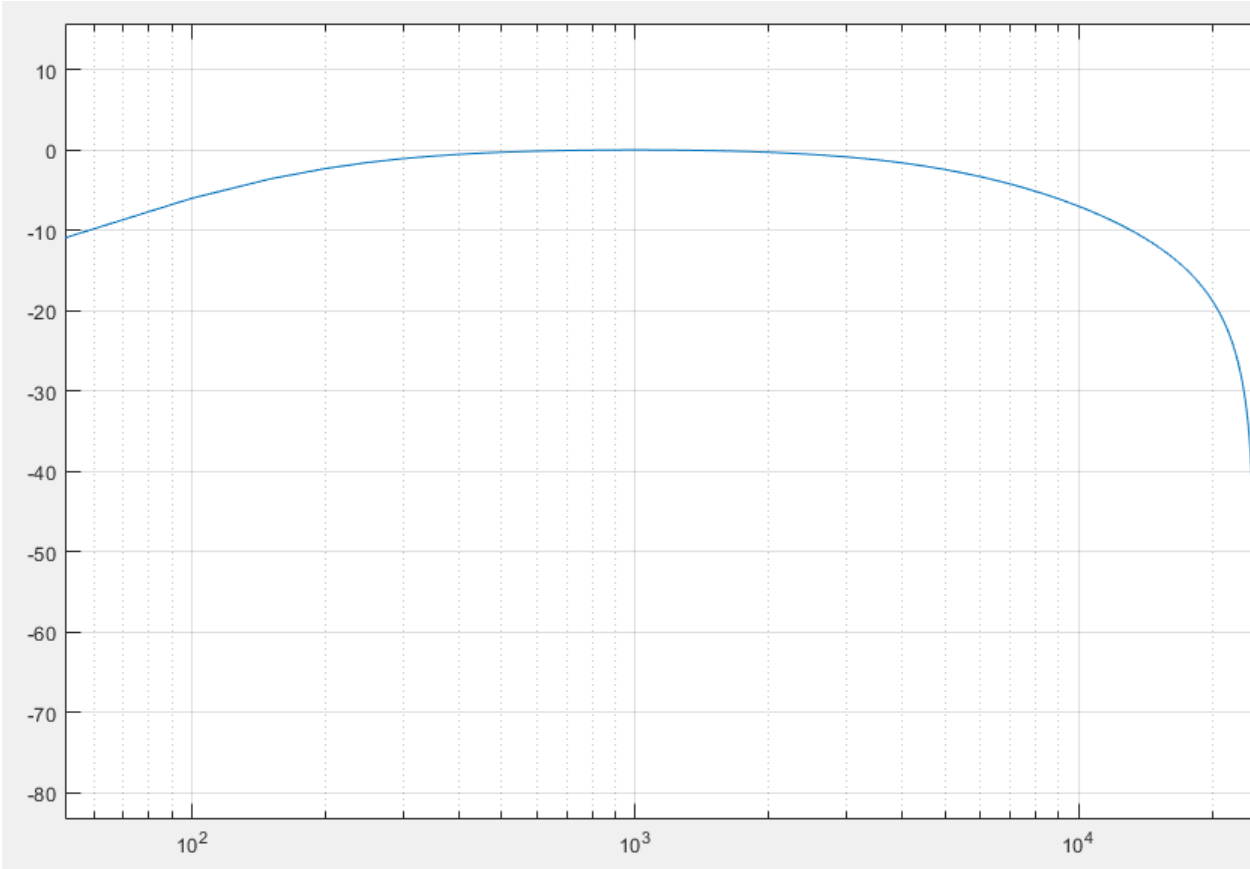


Рисунок 3 – АЧХ

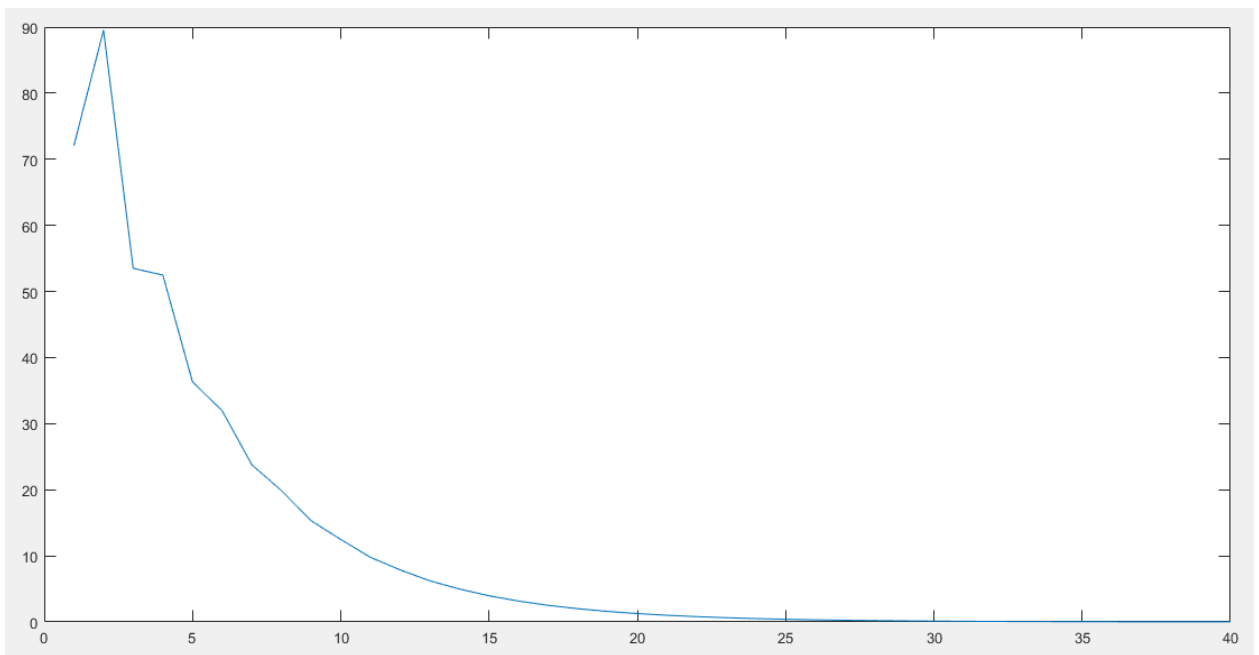


Рисунок 4 – Переходная характеристика

Программа

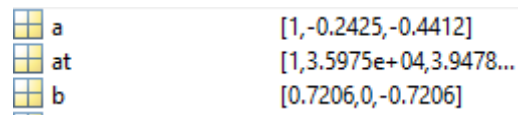
Перейдём к разработке программы на языке Verilog. Модуль будет состоять из следующих сигналов:

- **c** - обычный тактовый сигнал
- **TD** – сигнал с частотой дискретизации
- **X** – последовательность входных отсчётов
- **Y** – последовательность выходных отсчётов

Далее идёт описание входов и выходы для модуля. Таким образом описываются порты **c**, **TD**, **X** и **Y**, обозначенные ранее. После этого определяются различные регистры, а именно 16-битные регистры, которые используются для хранения истории входных отсчётов, 32-битные регистры, которые используются для хранения входных значений в формате с плавающей точкой, 32-битные регистры, которые используются для хранения выходных значений с плавающей точкой, 32-битные регистры, которые используются для хранения коэффициентов и промежуточных значений, связанных с рекурсивным цифровым фильтром, 32-битные регистры, которые используются для хранения коэффициентов числителя передаточной функции, а также 32-битные регистры, которые используются для хранения

коэффициентов знаменателя передаточной функции.

Далее идёт блок `always`, внутри которого идёт инициализация значений регистров коэффициентов передаточной функции рекурсивного цифрового фильтра. Значения были получены в MATLAB. Это показано на рисунке 5 (а и b).



a	[1,-0.2425,-0.4412]
at	[1,3.5975e+04,3.9478...
b	[0.7206,0,-0.7206]

Рисунок 5 – Полученные коэффициенты

После этого идёт формирование рекурсивных связей между входными и выходными отсчётами внутри рекурсивного цифрового фильтра. Затем идёт преобразование целочисленных значений вещественного формата, умножение вещественных чисел на коэффициенты нерекурсивной части разностного уравнения, последующее умножение вещественных чисел на коэффициенты рекурсивной части разностного уравнения и сложение компонентов разностного уравнения. Завершается код преобразованием выходного отсчёта.

Помимо этого, в проект была добавлена схема с расширением `bdf`. Это сделано для эффективной реализации на ПЛИС. В проекте применяются специальные блоки, способные выполнять арифметические операции с числами с плавающей точкой. Эти действия необходимы в силу того, что при вычислениях в контексте фильтров необходима высокая точность расчётов. Таким образом были настроены блоки:

- **altfp_convert** – блок преобразования типов данных
- **altfp_mult** – блок выполнения операции умножения
- **altfp_add_sub** – блок выполнения операции сложения-вычитания

Схема представлена на рисунке 6.

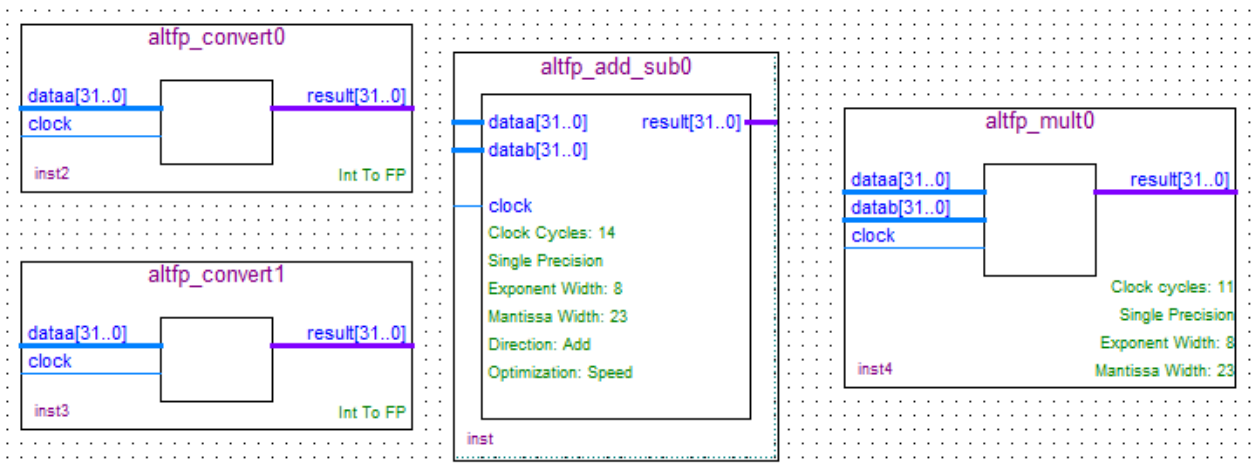


Рисунок 6 – Схема

Листинг программы

```
module lab7
```

```
(
```

```
c,
```

```
TD,
```

```
X,
```

```
Y
```

```
);
```

```
input wire c;
```

```
input wire TD;
```

```
input wire [15:0] X;
```

```
output wire [15:0] Y;
```

```
reg [15:0] Xn, Xn_1, Xn_2;
```

```
reg [31:0] XFn, XFn_1, XFn_2;
```

```
reg [31:0] YF, YFn, YFn_1, YFn_2;
```

```
reg [31:0] B0_XFn, B1_XFn_1, B2_XFn_2, _A1_YFn_1, _A2_YFn_2, B0_B1_X,  
B0_B1_B2_X, B0_B1_B2_A1_XY;
```

```
reg [31:0] BF0, BF1, BF2;
```

```
reg [31:0] AF1, AF2;
```

```
always @*
```

```
begin
```

```
BF0 = 32'h3F38793E;
```

```
BF1 = 32'h00000000;
```

```
BF2 = 32'hBF38793E;
```

```
AF1 = 32'h3E7851EC;
```

```
AF2 = 32'h3EE1E4F7;
```

```
end
```

```
always @ (posedge TD)
```

```
begin
```

```
Xn_2 <= Xn_1; Xn_1 <= Xn; Xn <= X;
```

```
YFn_2 <= YFn_1; YFn_1 <= YFn;
```

```
YF <= YFn;
```

```
end
```

```
altfp_convert0 int2float_Xn (.clock(c), .dataa(Xn), .result(XFn));
```

```
altfp_convert0 int2float_Xn_1 (.clock(c), .dataa(Xn_1), .result(XFn_1));
```

```
altfp_convert0 int2float_Xn_2 (.clock(c), .dataa(Xn_2), .result(XFn_2));
```

```
altfp_mult0 float_multiplication_B0X (.clock(c), .dataa(XFn), .datab(BF0),  
.result(B0_XFn));
```

```
altfp_mult0 float_multiplication_B1X (.clock(c), .dataa(XFn_1), .datab(BF1),  
.result(B1_XFn_1));
```

```
altfp_mult0 float_multiplication_B2X (.clock(c), .dataa(XFn_2), .datab(BF2),  
.result(B2_XFn_2));
```

```
altfp_mult0 float_multiplication_A1Y (.clock(c), .dataa(YFn_1), .datab(AF1),  
.result(_A1_YFn_1) );
```

```
altfp_mult0 float_multiplication_A2Y (.clock(c), .dataa(YFn_2), .datab(AF2),  
.result(_A2_YFn_2) );
```

```

altfp_add_sub0 altfp_add_sub0_B0B1 (.clock(c), .dataa(B0_XFn),
.datab(B1_XFn_1), .result(B0_B1_X) );
altfp_add_sub0 altfp_add_sub0_B0B1B2 (.clock(c), .dataa(B0_B1_X),
.datab(B2_XFn_2), .result(B0_B1_B2_X) );
altfp_add_sub0 altfp_add_sub0_BOB1B2A1 (.clock(c), .dataa(B0_B1_B2_X),
.datab(_A1_YFn_1), .result(B0_B1_B2_A1_XY) );
altfp_add_sub0 altfp_add_sub0_Yn (.clock(c), .dataa(B0_B1_B2_A1_XY),
.datab(_A2_YFn_2), .result(YF) );

altfp_convert1 float2int_Y (.clock(c), .dataa(YF), .result(Y) );

endmodule

```

Результаты моделирования

Искомые результаты продемонстрированы на рисунках 7 – 9.

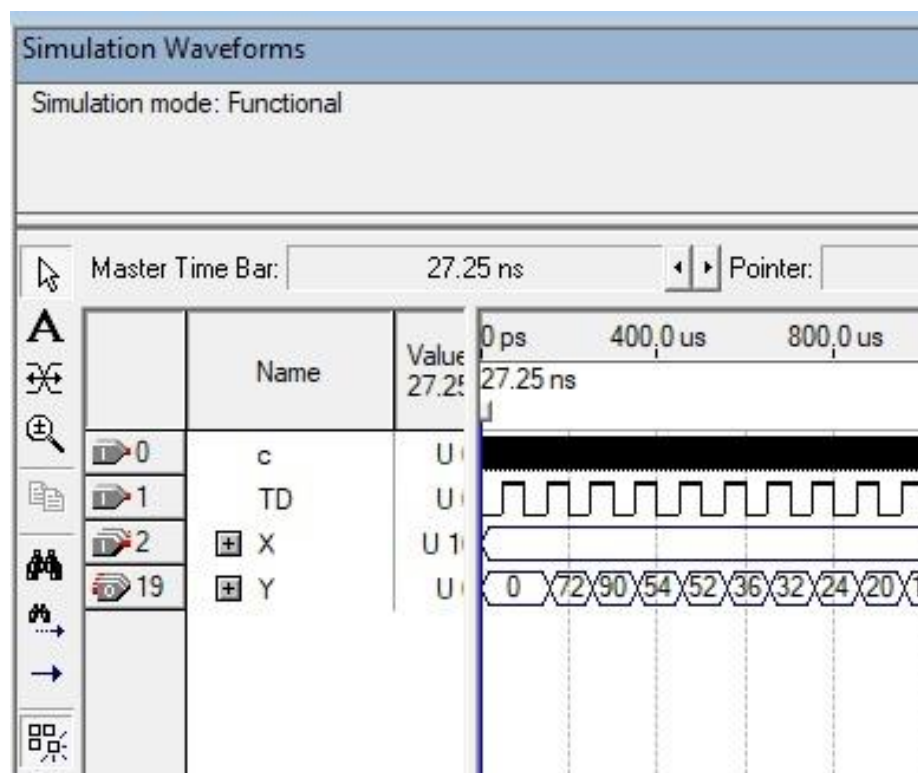


Рисунок 7 – Результат моделирования (начало)

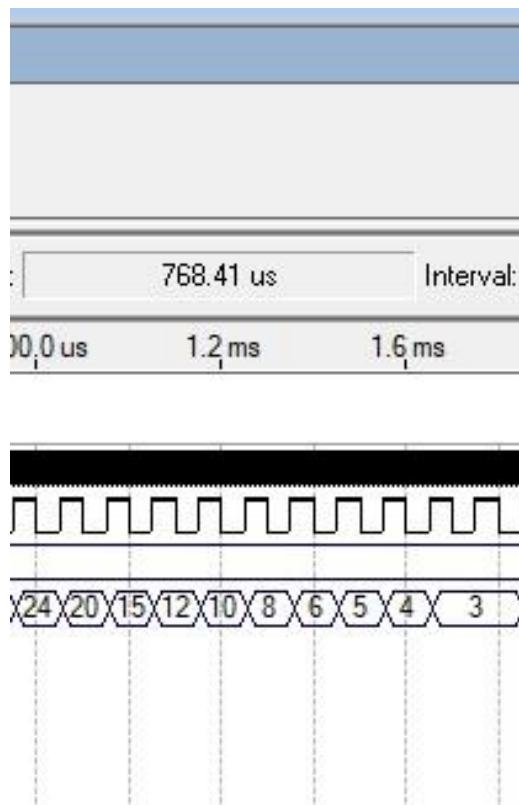


Рисунок 8 – Результат моделирования (середина)

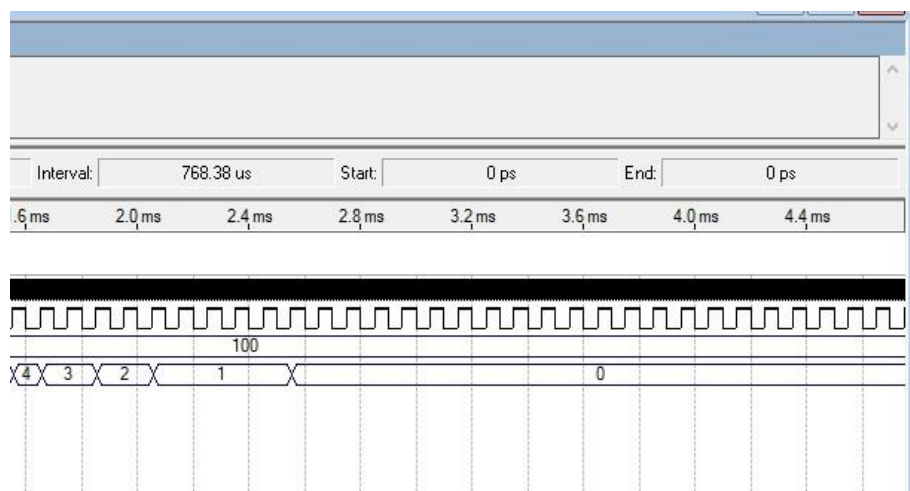


Рисунок 9 – Результат моделирования (окончание)

Обратим внимание на полученные выходные отсчёты. Как видим из рисунков 7 – 9 и 2, значения совпали. Это говорит о корректном построении фильтра.

Выводы

В данной лабораторной работе были изучены методология расчёта цифровых фильтров и способ их реализации на ПЛИС с использованием языков описания аппаратуры.

Список используемых источников

1. Проектирование встраиваемых систем на ПЛИС. / З.Наваби; перев. с англ. В.В. Соловьева. – М.: ДМК Пресс, 2016. - 464 с.
2. Проектирование цифровых устройств на ПЛИС: учеб. пособие / И.В. Ушенина. - СПб: Лань, 2022. - 408 с.
3. Цифровая схемотехника и архитектура компьютера / Д.М. Харрис, С.Л. Харрис; пер. с англ. Imagination Technologies. – М.: ДМК Пресс, 2018. - 792 с.
4. Учебно-методические материалы к выполнению лабораторной работы №7 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://pro.guap.ru/inside/student/tasks/ad082e0862c9584d5144a1bd553cf8b0/download>. (Дата обращения: 27.04.24).
5. Лекция №4 от 25 марта 2024 года по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://bbb2.guap.ru/playback/presentation/2.3/8df832b35e59b7b17bb499a557e56942cdbfdf0b-1711367744513>. (Дата обращения: 27.04.24).
6. Лекция №5 от 8 апреля 2024 года по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://bbb2.guap.ru/playback/presentation/2.3/a6ff35572040a1e35c40a0f1dce7d285f414ed63-1712576737388>. (Дата обращения: 27.04.24).