

Лабораторные работы кафедры ВТ ИКТИБ ЮФУ для курсов «Проблемно-ориентированные методы и средства цифровой обработки сигналов», «Специализированные методы и средства цифровой фильтрации»

Лабораторная работа №3.

Процедурная реализация быстрого преобразования Фурье на ПЛИС

Цель работы.

Лабораторная работа №3 нацелена на изучение принципов аппаратной процедурной реализации алгоритма быстрого дискретного преобразования Фурье.

Подготовка к работе.

Лабораторная работа предполагает аппаратную реализацию вычислительной структуры быстрого дискретного преобразования Фурье в IDE Quartus 9.2. в виде процедурного блока. Для генерации входного и эталонного массивов данных необходимо использовать реализацию модельной программы целочисленного БПФ на Python из задания №2 лабораторной работы №2.

Ход работы:

1. В САПР Quartus 9.2 реализовать базовую операцию БПФ (бабочку), выполняющую операции с целыми знаковыми числами для вида прореживания и разрядности, выбранных согласно варианту лабораторного задания. Вычислительная структура бабочек приведена на рисунке 1.

Реализацию схемы бабочки необходимо выполнить с помощью мегафункций Quartus. Рекомендуется выбирать следующую латентность (latency, clock cycles) операций: $latency_{mul} = 3$ для целочисленного умножения со знаком (signed); $latency_{add/sub} = 1$ для целочисленного сложения/вычитания со знаком (signed). При реализации схемы не забыть

обрезать младшие $q - 1$ разрядов и один старший биты для сохранения разрядности вычислений (см. задание №2 лабораторной работы №2).

Вместе с этим, схему базовой операции БПФ необходимо реализовать в виде конвейера, обеспечивающего следующее поведение: если входные операнды $\{\text{ReA}, \text{ImA}, \text{ReB}, \text{ImB}, \text{ReW}, \text{ImW}\}$ поступают одновременно, выходные операнды $\{\text{ReA_out}, \text{ImA_out}, \text{ReB_out}, \text{ImB_out}\}$ также выходят одновременно через L тактов. Параметр L рассчитывается исходя суммарной latency длиннейшего пути входных операндов через операции сложения/вычитания и умножения:

$$L = \text{latency}_{mul} + 2 \cdot \text{latency}_{add/sub}$$

Для того, чтобы данные на выходе появились одновременно, необходимо в схему бабочки с прореживанием по времени добавить регистры, задерживающие ReA и ImA на $\text{latency}_{mul} + \text{latency}_{add/sub}$ тактов, а в схему бабочки с прореживанием по частоте добавить два вида регистров: задерживающие ReW и ImW на $\text{latency}_{add/sub}$ тактов, а также задерживающие результат ReA_out и ImA_out на $\text{latency}_{mul} + \text{latency}_{add/sub}$ тактов.

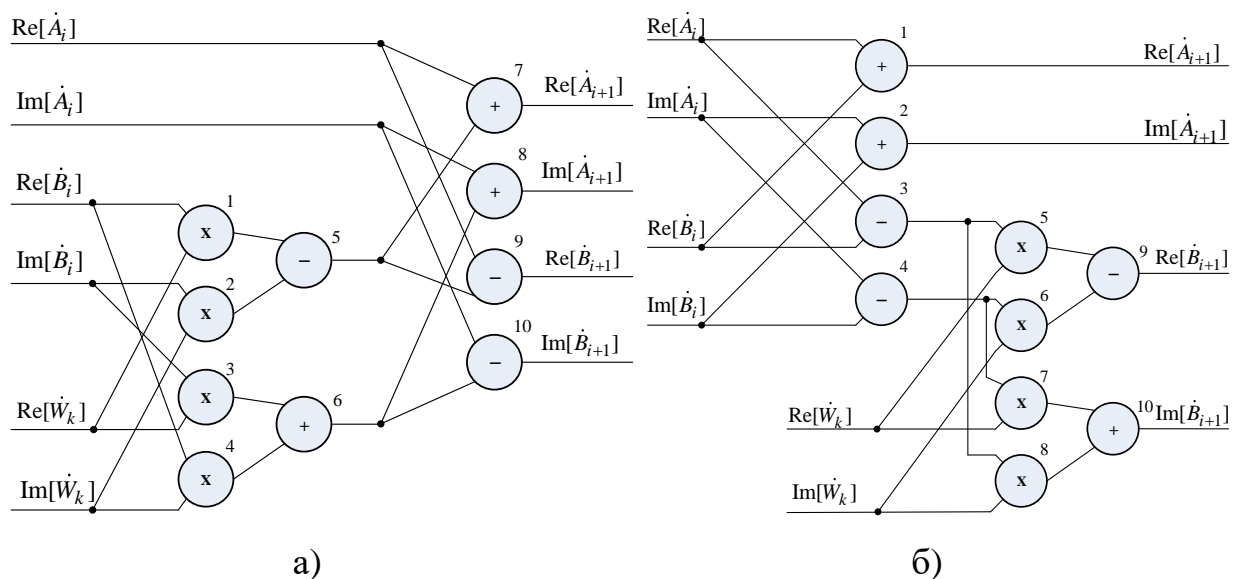


Рисунок 1 – вычислительные структуры базовых операций БПФ для разных видов прореживания: а) с прореживанием по времени; б) с прореживанием по частоте.

2. Реализовать процедуру, схема которой приведена на рисунке 2 или же конвейерную схему из задания повышенной сложности для получения бонусных баллов (по согласованию с преподавателем).

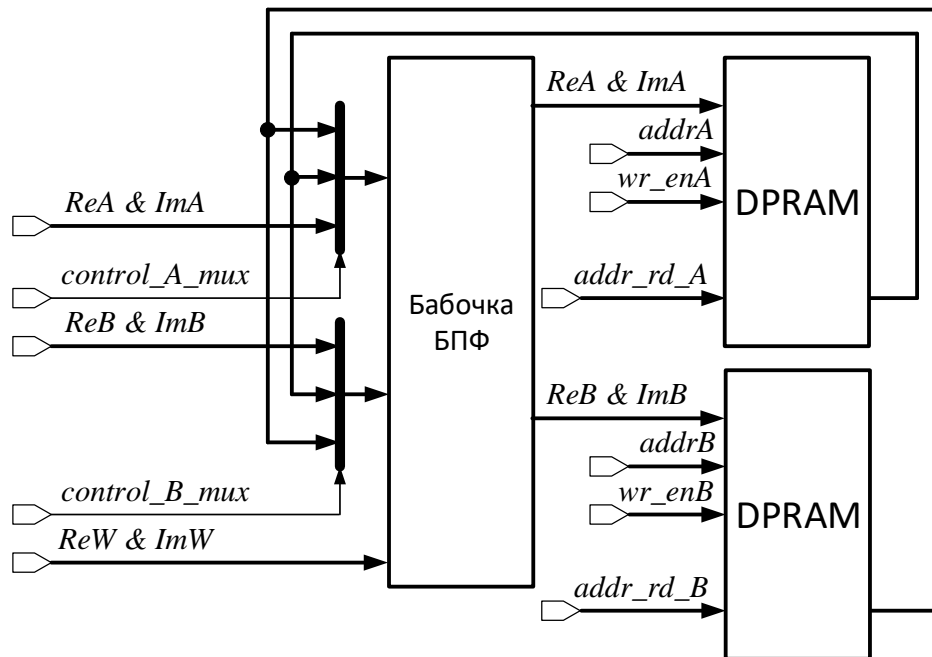


Рисунок 2 – схема процедуры алгоритма БПФ

По входным портам $ReA \& ImA$, $ReB \& ImB$, $ReW \& ImW$ передаются объединенные действительная и мнимая части комплексных входных данных (& – операция конкатенации). В связи с этим, данные порты будут удвоенной разрядности $2 \cdot q$ относительно заданного варианта лабораторной работы.

Данные на вход процедуры подаются последовательно в порядке, заданном графом БПФ (как это реализовано в лабораторной работе №2). Результаты вычислений записываются в двупортовую память DPRAM по первому порту. Адреса записи поступают по входным портам $addrA$, $addrB$, сигналы разрешения записи поступают по входным портам wr_enA , wr_enB .

После обработки данных первого слоя, результаты вычитываются из памяти по входным портам $addr_rd_A$, $addr_rd_B$ и вновь поступают на обработку. Входные порты $control_A_mux$, $control_B_mux$ управляют

поступлением данных на вход бабочки БПФ в порядке, заданном графом БПФ.

3. Сформировать Waveform и подать на входные порты процедуры данные из задания №2 лабораторной работы №2: дискретный сигнал в integer и масштабированные весовые коэффициенты.
4. Выполнить моделирование и получить результат. Выгрузить в виде текстового файла результат вычислений, содержащийся в дампе модели памяти DPRAM в Waveform report. Очистить файл, оставив только результаты последнего слоя БПФ.
5. Подгрузить результат вычислений в программу из задания №2 лабораторной работы №2 любым возможным способом (вручную или автоматически средствами Python).
6. Вывести на одном графике результаты моделей: floating point (задание №1) и integer (задание №2) лабораторной работы №2, а также результат работы модели в ПЛИС.
7. В отчете сделать вывод о точности вычислений в ПЛИС.
8. В отчете привести ресурс ПЛИС, требуемый на реализацию всей схемы.
9. В отчете привести данные о максимально возможной тактовой частоте проекта, взятые из отчета о компиляции проекта.
10. В отчете привести RLT схему из результатов компиляции проекта.

Варианты заданий к лабораторной работе №3

Варианты лабораторного задания выбираются согласно номеру студента в общем списке группы. Нечетный номер реализует БПФ с помощью библиотеки `scipy`, четный – с помощью `numpy`.

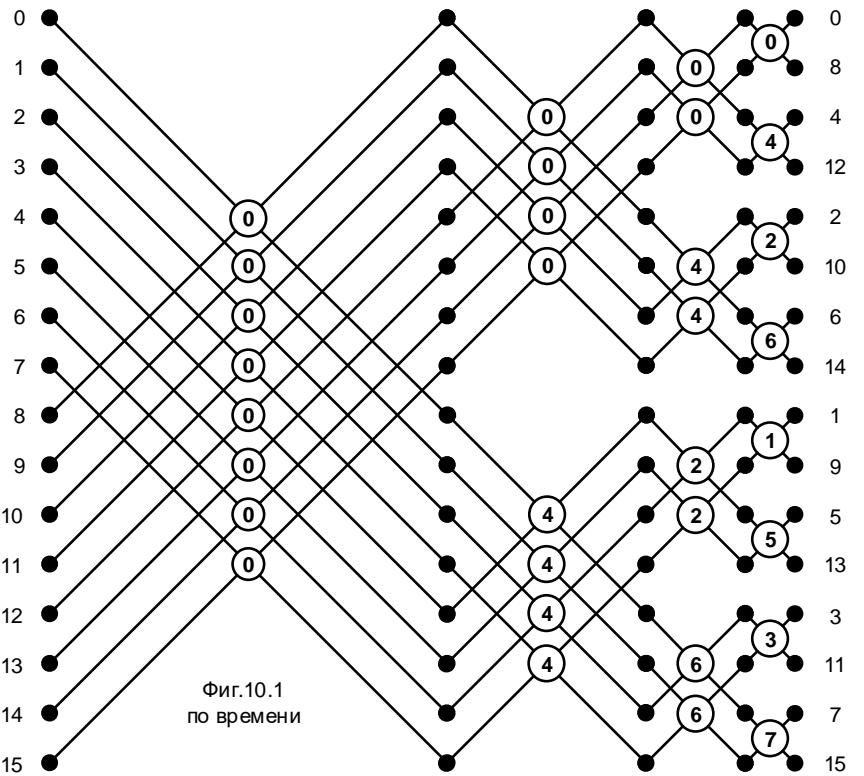
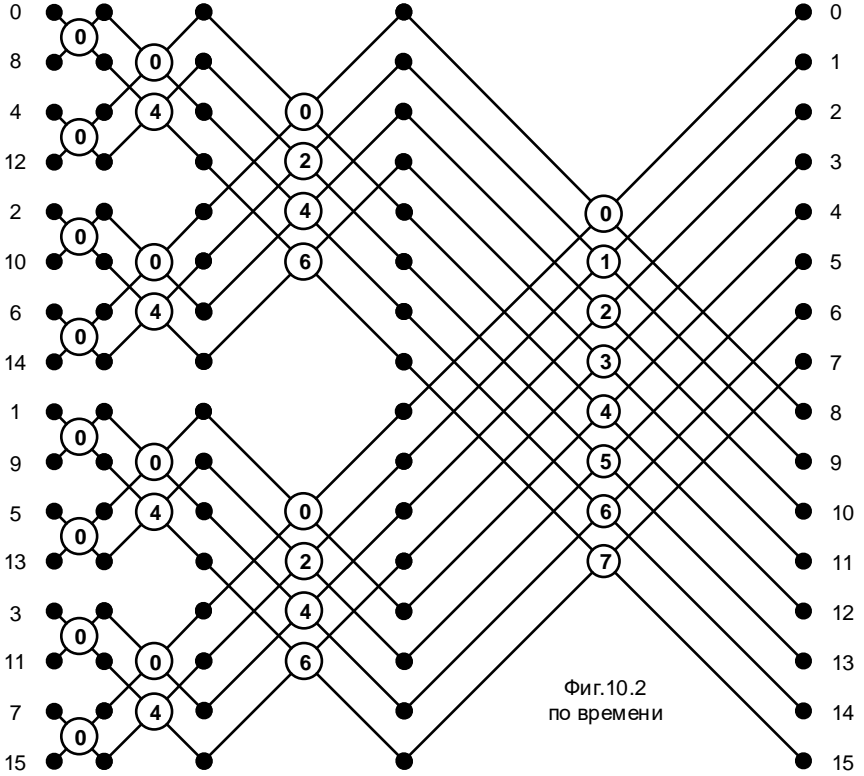
В таблице приведены значения частот и амплитуд для генерации модельного сигнала, схема бабочки алгоритма БПФ и разрядность выполняемых операций.

Таблица 1 – варианты лабораторных заданий

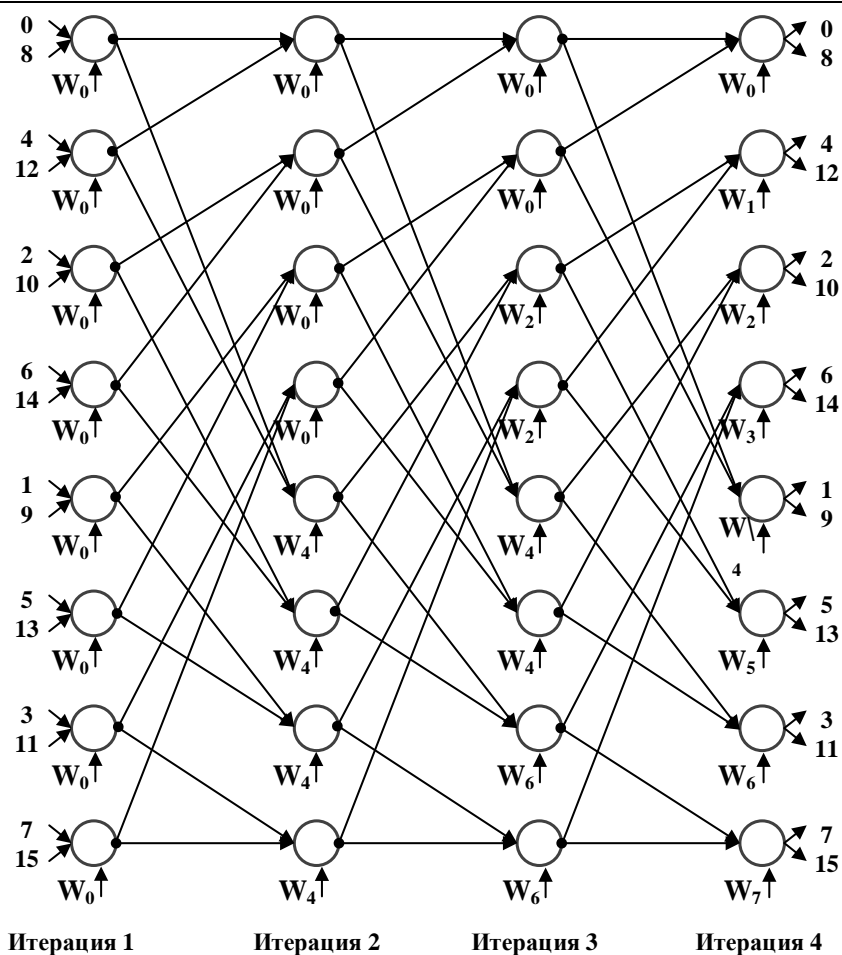
№		1	2	3	4	5	6	7	Схема бабочки:	Разрядность:
1	F, кГц	0,5	1	2	5	7	9	12	С прореживанием по времени 1	8 бит
	A	1	5	3	7	3	2	1		
2	F, кГц	3	5	6	9	12	15	18	С прореживанием по частоте 1	9 бит
	A	7	9	8	12	7	4	6		
3	F, кГц	0,9	1	2	9	12	15	18	С прореживанием по времени 2	10 бит
	A	2	5	3	12	7	4	6		
4	F, кГц	3	5	6	8	9	10	12	С прореживанием по частоте 2	11 бит
	A	7	9	8	7	3	2	5		
5	F, кГц	0,5	1	2	9	12	14	17	С прореживанием по времени 3	12 бит
	A	1	5	3	12	15	4	6		
6	F, кГц	2	4	5	7	9	12	15	С прореживанием по частоте 3	13 бит
	A	11	15	12	13	21	10	9		
7	F, кГц	9	11	15	17	21	29	35	С прореживанием по времени 1	14 бит
	A	17	19	36	34	28	25	40		
8	F, кГц	0,3	0,56	1,2	3,5	5	7	9	С прореживанием по частоте 1	15 бита
	A	21	18	23	19	13	20	11		
9	F, кГц	10	13	14	17	21	22	25	С прореживанием по времени 2	16 бита
	A	9	14	15	19	21	26	30		
10	F, кГц	1	5	7	13	18	19	29	С прореживанием по частоте 2	17 бита
	A	9	11	12	17	21	29	35		
11	F, кГц	10	11	12	13	15	20	25	С прореживанием по времени 3	18 бита
	A	9	11	15	17	21	29	35		
12	F, кГц	21	25	32	33	41	45	50	С прореживанием по частоте 3	17 бита
	A	9	11	15	17	21	29	35		
13	F, кГц	20	21	22	23	27	30	40	С прореживанием по времени 1	16 бита
	A	9	11	15	17	21	29	35		
14	F, кГц	11	15	12	13	21	23	25	С прореживанием по частоте 1	15 бит
	A	19	12	15	27	23	29	25		
15	F, кГц	5	11	12	13	21	23	30	С прореживанием по времени 2	14 бит
	A	9	11	15	17	21	29	35		
16	F, кГц	11	13	15	17	21	23	25	С прореживанием по частоте 2	13 бит
	A	22	23	25	17	21	29	35		
17	F, кГц	1	2	3	7	9	11	13	С прореживанием по времени 3	12 бит
	A	2	12	11	15	3	8	13		
18	F, кГц	3	9	14	16	21	26	32	С прореживанием по частоте 3	11 бит
	A	12	2	17	5	9	15	6		
19	F, кГц	5	7	12	17	24	26	43	С прореживанием по времени 1	10 бит
	A	5	14	3	23	6	20	7		
20	F, кГц	1	4	14	19	20	23	29	С прореживанием по частоте 1	8 бит
	A	8	5	12	7	3	15	8		

Различные виды информационных графов алгоритма БПФ приведены в таблице 2.

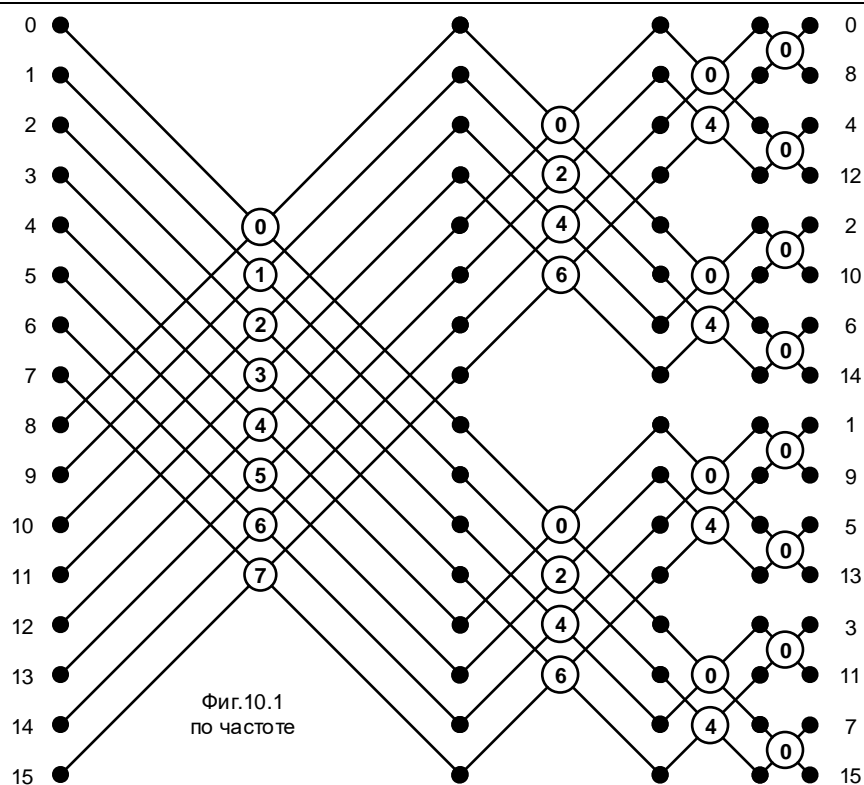
Таблица 2 – варианты информационных графов алгоритма БПФ

Тип и номер графа	Информационный граф алгоритма БПФ
С прореживанием по времени №1	 <p>Фиг.10.1 по времени</p>
С прореживанием по времени №2	 <p>Фиг.10.2 по времени</p>

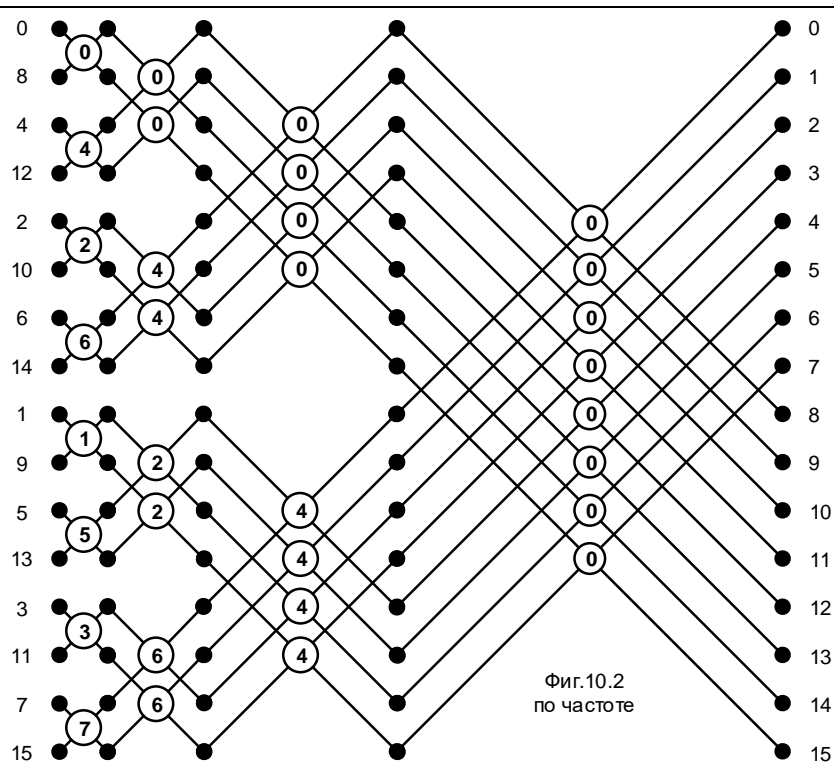
С прореживанием
по времени №3



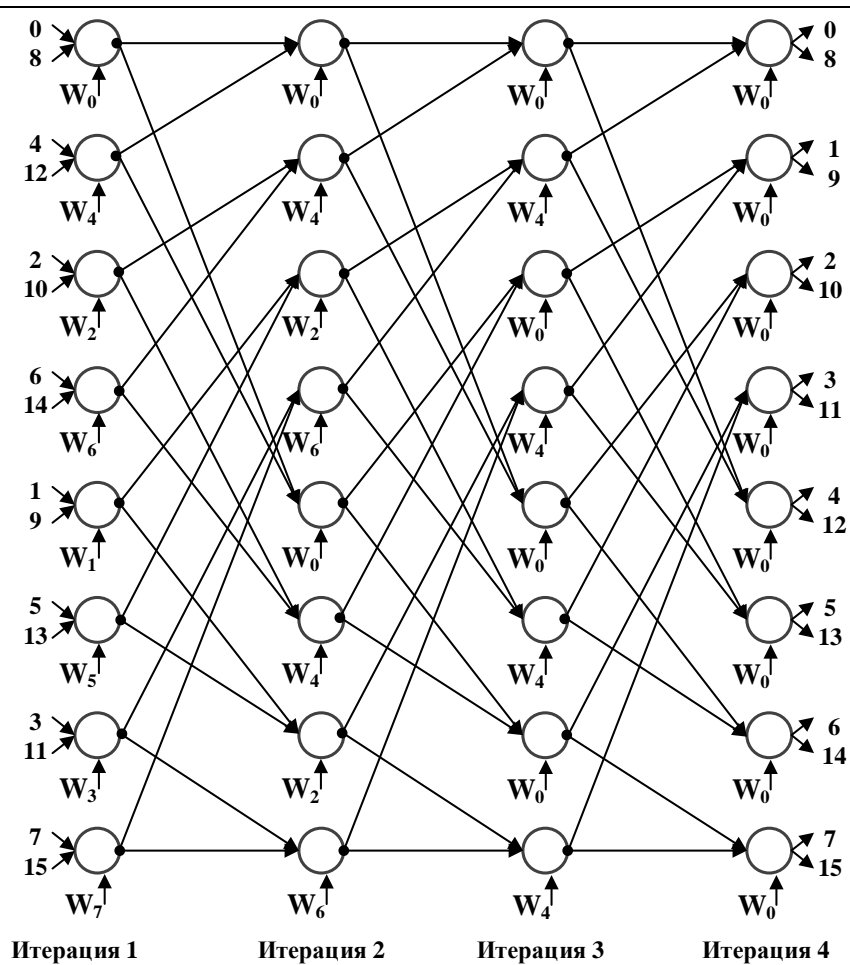
С прореживанием
по частоте №1



С прореживанием
по частоте №2

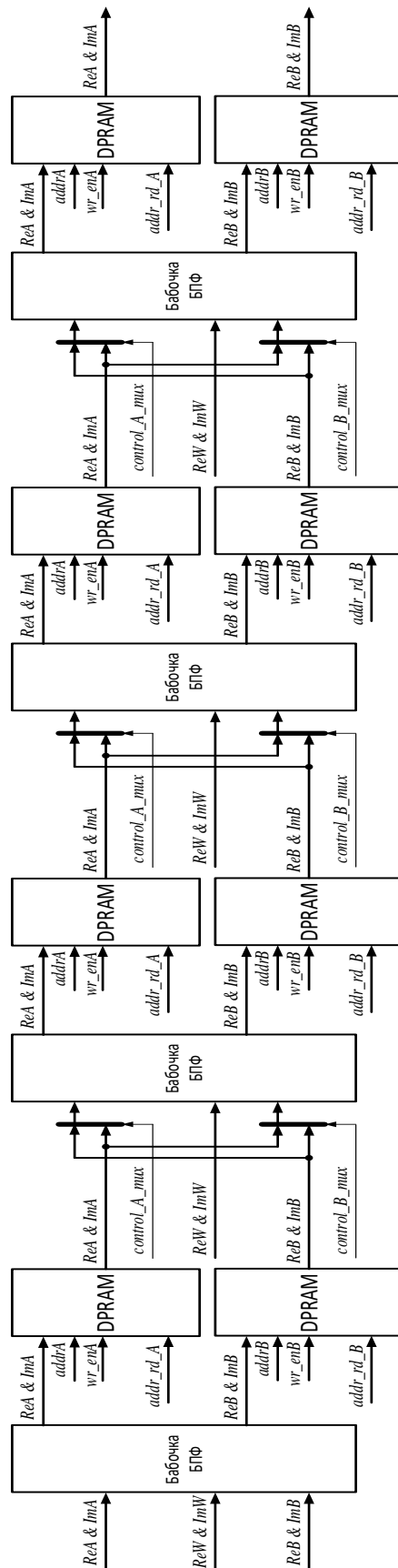


С прореживанием
по частоте №3



Задание повышенной сложности на дополнительные баллы:

Реализовать конвейерную схему, где каждая их бабочек БПФ обчисляет один из слоев графа.



Требования к отчету и защита

Процесс выполнения лабораторной работы документируется с помощью текстового редактора MS Word, полученные сведения служат основой для формирования отчета о выполнении лабораторной работы. Отчет в общем случае должен включать:

- титульный лист;
- описание задач в выбранном варианте лабораторной работы;
- схемы из Quartus 9.2
- отчет о компиляции проекта
- временная диаграмма моделирования проекта
- график исходного сигнала во временной области (задание №1 лабораторной работы №2)
- график исходного сигнала в частотной области в float point (задание №1 лабораторной работы №2)
- график исходного сигнала в частотной области в integer (задание №2 лабораторной работы №2)
- график исходного сигнала в частотной области, полученный в результате моделирования схемы в ПЛИС в integer (лабораторная работа №3)
- выводы
- листинг программы

Защита отчета о выполнении лабораторной работы сопровождается демонстрацией работоспособности кода программ, теоретических знаний и ответов на дополнительные вопросы преподавателя по теме занятия.

ПРИЛОЖЕНИЕ 1.

1. Инструкция по установке IDE PyCharm.

- a. Зайти на страницу загрузки IDE PyCharm с официального сайта JetBrains, опуститься вниз страницы и нажать Download PyCharm Community Edition. Данная версия программного продукта не нуждается в лицензировании.

<https://www.jetbrains.com/pycharm/download/?section=windows>

- b. Установить IDE PyCharm. При установке выбрать все чекбоксы.
- c. С официального сайта Python скачать и установить последнюю версию интерпретатора языка. При установке отметить все чекбоксы (обязательно выбрать «Add python.exe to PATH»).

<https://www.python.org/downloads/>

2. Инструкция по установке VSCode и начальная настройка среды.

- a. Зайти на главную страницу официального сайта VSCode и нажать кнопку Download. VSCode не нуждается в лицензировании.

<https://code.visualstudio.com/>

- b. Установить VSCode. При установке выбрать все чекбоксы.
- c. Открыть среду VSCode и зайти в раздел расширения (Extensions).



- d. Найти и установить расширение «Russian Language Pack for Visual Studio Code» (опционально).
- e. Найти и установить расширение «Python».
- f. С официального сайта Python скачать и установить последнюю версию интерпретатора языка. При установке отметить все чекбоксы (обязательно выбрать «Add python.exe to PATH»).


<https://www.python.org/downloads/>

3. Подготовка работы с IDE.

- a. Проверить, установлена ли на компьютере IDE PyCharm или VSCode. Если нет, выполнить установку согласно описанной выше инструкции.
- b. Создать рабочую директорию с вашей фамилией. Желательно, чтобы папка находилась не на рабочем столе Windows.
- c. Запустить IDE и выполнить команду File => Open Folder, где выбрать путь к созданной вами директории.
- d. Создать новый файл Python File: команда File => New => Python File и задать имя файла с расширением *.py (например, lab1.py).
- e. Файл открылся в правой области окна проекта. Здесь можно набирать код программы.

4. Пример написания и запуска программы в IDE.

Хорошей традицией при изучении первого языка программирования является написание программы, выводящей на экран компьютера приветствие «Hello World!». Для создания такой программы необходимо предварительно создать и открыть новый python файл, как это было описано выше.

- a. Наберите команду: `print('Hello Python!')`
- b. Выполните запуск программы нажав на клавишу RUN , расположенную в правом верхнем углу IDE. Результат работы программы появится мгновенно в окне вывода, расположенном в нижней части IDE.
- c. Для запуска программы в первый раз нужно щелкнуть правой кнопкой мыши в окне с текстом программы и выбрать пункт Run.
- d. Если запуск в IDE PyCharm произошел с ошибкой, проверьте, указан ли интерпретатор языка в окне Run => Edit Configuration.

5. Инструкция по установке пакетов matplotlib, scipy, numpy.

- a. В IDE VSCode зайти в раздел расширения (Extensions); в IDE PyCharm открыть вкладку Python Packages.
- b. Найти и установить библиотеки matplotlib, scipy, numpy.