

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Инфокоммуникационные технологии»

РАБОТА ПРОВЕРЕНА

Рецензент

_____/Г.А. Непомнящий/

« ____ » _____ 2016 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____/С.Н. Даровских/

« ____ » _____ 2016 г.

**«Аппаратно-программное устройство определения параметров радиосигнала
на основе частотно-временной обработки»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ-Д.11.05.01.2016.284.00 ПЗ ВКР

Консультанты

Безопасность жизнедеятельности

_____/И.С. Окраинская/

« ____ » _____ 2016г.

Руководитель проекта

Главный инженер ЧНТЦ ЦТОРС

_____/А.Н. Николаев/

« ____ » _____ 2016 г.

Организационно - экономический
раздел

_____/Р.Ш. Закиров/

« ____ » _____ 2016 г.

Автор проекта

студент группы КЭ-665

_____/Н.С. Никитин/

« ____ » _____ 2016 г.

Нормоконтролер

_____/В.Д. Спицина/

« ____ » _____ 2016 г.

Челябинск 2016

РЕФЕРАТ

Никитин Н.С. Аппаратно-программное устройство определения параметров радиосигнала на основе частотно-временной обработки – Челябинск: ЮУрГУ, ВШЭЖН; 2016, 91 с., 28 ил., 10 табл., библиогр. список 8 наим.

В данной дипломной работе рассматривается процесс разработки аппаратно-программного устройства определения параметров радиосигнала (далее по тексту - устройство) на основе частотно-временной обработки принимаемой информации. Основными объектами исследования является алгоритм определения параметров сигнала по результатам частотно-временного анализа, формирование дескрипторов для обнаруженных сигналов и передача их на персональный компьютер.

В первом разделе приводится анализ технического задания на устройство. Приводятся результаты обзора литературы, в том числе и зарубежной, по схожим устройствам. Обосновывается выбор ПЛИС для реализации алгоритма.

Во втором разделе описаны принципы частотно-временной обработки, а также явления, которые следует учитывать при анализе полученных данных и с которыми пришлось столкнуться на этапе испытаний устройства.

В третьем разделе представлено описание модели алгоритма, разработанной в среде Matlab Simulink а также результаты работы модели. В четвертом разделе приведены результаты испытаний. В приложении имеются коды исходных файлов программы для ПЛИС, разработанного алгоритма. Также рассматриваются вопросы экономической эффективности разработки. В разделе БЖД приводятся основные нормативные документы и правила безопасности при эксплуатации устройства

В заключительной части сделаны выводы по эффективности разработки и оценены перспективы её дальнейшего развития.

ЮУрГУ-Д.11.05.01.2016.284.00 ПЗ ВКР

Лит	Изм.	№ докум.	Подп.	Дата				
	Разраб.	Никитин Н.С.			Аппаратно-программное устройство определения параметров радиосигнала на основе частотно- временной обработки	Лит	Лист	Листов
	Пров.	Николаев А.Н.				Д	2	91
	Т. контр.					ЮУрГУ Кафедра ИКТ		
	Н. контр.	Спицына В.Д.						
	Утв.	Даровских С.Н.						

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

ОГЛАВЛЕНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	4
ВВЕДЕНИЕ	5
1 Анализ технического задания	
1.1 Анализ исходных данных.....	7
1.2 Анализ требуемых результатов	8
1.3 Обзор литературных источников	9
2 Разработка структуры устройства	
2.1 Метод частотно-временной обработки.....	11
2.2 Кратковременное преобразование Фурье.....	14
2.3 Структура многоканального приемника с частотно-временным процессором.....	16
2.4 Разработка подробной структуры устройства	17
2.5 Выбор весового окна анализа для БПФ	20
2.6 Исследование характеристик обнаружителя на основе БПФ.....	23
2.7 Разработка алгоритмов измерения частоты	27
2.8 Разработка алгоритмов определения параметров сигнала	35
3 Разработка модели устройства	
3.1 Назначение и условия выполнения модели устройства	
3.2 Описание внешнего вида модели	38
3.3 Работа с моделью	42
3.4 Сообщения оператору.....	44
4 Результаты испытаний устройства.....	46
5 Безопасность жизнедеятельности.....	49
6 Организационно-экономический раздел	
6.1 Сетевое планирование.	54
6.2 Расчет параметров работ сетевого графика.....	57
6.3 Расчет затрат на проведение НИОКР.....	62
6.4 Экономическая эффективность	65
ЗАКЛЮЧЕНИЕ	66
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	67
ПРИЛОЖЕНИЕ А	68

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПЛИС – программируемая логическая интегральная схема;

БПФ – быстрое преобразование Фурье;

КПФ – кратковременное преобразование Фурье;

АЦП – аналого-цифровой преобразователь;

СВЧ – сверх высокие частоты;

ЦОС – цифровая обработка сигналов;

SGMII – Serial Gigabit Media Independent Interface;

UDP – User Datagram Protocol;

ARP – Address Resolution Protocol;

ICMP – Internet Control Message Protocol;

CORDIC – COordinate Rotation Digital Computer;

VHSIC – Very high speed integrated circuits;

VHDL – VHSIC Hardware Description Language;

НС – непрерывный сигнал;

ИС – импульсный сигнал;

$N_{\text{шум}}$ – спектральная плотность мощности шума;

P_f – вероятность ложной тревоги;

P_d – вероятность правильного обнаружения;

ЛЧМ – линейная частотная модуляция;

ФКМ – фазо-кодовая манипуляция;

ВВЕДЕНИЕ

На сегодняшний день количество систем передачи радиосигнала становятся все более распространенными. Современные связные системы постоянно совершенствуют элементную базу и захватывают все более высокочастотный диапазон. Несмотря на то, что частотный диапазон четко регламентируется для каждого типа устройств (системы связи, системы управления воздушным движением, радиолобительские станции), постоянно появляются незаконные источники излучения, которые могут вмешиваться в работу систем. И если для радиолобительской станции это вмешательство не наносит сильного ущерба, то нарушение работы системы управления воздушным движением аэропорта может стать причиной катастрофы. По этой причине как нельзя актуальным становятся системы радиомониторинга для обнаружения и фиксации незаконных источников излучения в конкретном частотном диапазоне.

С развитием элементной базы системы радиомониторинга получили возможность увеличить мгновенный анализируемый частотный диапазон. Применение современных АЦП, с большой частотой дискретизации, позволяют использовать метод прямого преобразования частоты для анализа промежуточной частоты приемника. Этот факт позволяет использовать цифровую обработку сигналов для выполнения основной функции радиомониторинга – обнаружение и классификация источников излучения.

Целью данной дипломной работы является разработка аппаратно-программного устройства на основе алгоритмов обнаружения и определения параметров радиосигнала, которое будет применяться в цифровом приемнике системы радиомониторинга. В системах приема обработка сигналов разделяется на этап первичной, вторичной и третичной обработки. Алгоритмы, разрабатываемые в данной работе, относятся к этапу вторичной обработки. Также данное устройство можно использовать в системах радиотехнической разведки. Существующие системы радиомониторинга и имеют ряд недостатков:

а. системы, работающие в широком частотном диапазоне, быстро реагируют на сигнальную обстановку, но не может зафиксировать узкополосные изменения;

б. системы, работающие в узкой полосе, имеют медленное время реагирования, но они не могут обнаруживать широкополосные сигналы, имеющие внутриимпульсную модуляцию.

Задачами данной работы является:

- разработка алгоритма обнаружения и определения параметров сигнала по результатам частотно-временного анализа;

- разработка имитационной модели устройства;

- исследование разработанных алгоритмов на имитационной модели устройства;

- написание программного обеспечения для ПЛИС, реализующее данные алгоритмы;

- испытание разработанного устройства на макетном образце.

Пояснительная записка организована следующим образом: в первом разделе проводится анализ технического задания, отмечаются основные проблемы, с которыми придется столкнуться при разработке устройства, также приводятся результаты анализа литературных источников, на основе которых выбирается прототип разрабатываемого устройства; во втором разделе описывается этапы разработки структуры устройства, а также результаты испытания на имитационной модели; в третьем разделе приводится описание имитационной модели устройства; в четвертом разделе приводятся результаты испытания программного обеспечения на ПЛИС на макетном образце.

1 Анализ технического задания

В данном разделе проводится анализ задания на дипломную работу. Рассматриваются исходные данные для работы, требуемые результаты, а также возможные проблемы при разработке устройства.

1.1 Анализ исходных данных

Перед началом разработки устройства необходимо проанализировать, какие данные ему придется обрабатывать, и в каком виде необходимо предоставлять результаты работы. Согласно ТЗ, Устройства будет размещаться на ПЛИС и осуществлять анализ данных от входного аналогового СВЧ тракта. Использование высокоскоростных АЦП позволяет оцифровывать выборки непосредственно с СВЧ тракта. Требуемая разрядность АЦП составляет 10 бит, частота дискретизации должна быть равна 1 ГГц. Исходя из этого, можно сказать, что на ПЛИС должно идти два потока данных с разрядностью 10 бит и частотой следования 1 ГГц. Даже современные ПЛИС не могут полноценно обрабатывать такой поток данных. Для облегчения обработки скоростных потоков применяются специализированные приемники-десериализаторы, основная задача которых понизить скорость передаваемой информации до приемлемого уровня для обработки. Такой эффект достигается за счет распараллеливания одного потока на несколько, идущих с меньшей скоростью. Следовательно, необходимо проработать вопрос частоты работы схемы обработки данных в ПЛИС, для грамотной настройки таких приемников.

В среде разработки Vivado Design Suite, фирмы Xilinx, находится большая библиотека примеров работы с аппаратными частями ПЛИС. Поэтому для упрощения процесса разработки был использован готовый блок управления входными приемниками из примера и модифицирован под требуемые частоты работы.

Следующим пунктом задания, которое следует проработать, является требование к анализу данных от блоков БПФ. Сам механизм БПФ не представляет

собой никаких трудностей, производители ПЛИС в свободном доступе предоставляют готовые блоки БПФ, являющимися для разработчиков «черными ящиками». Одно из главных ограничений этих блоков является требования по тактовой частоте работы. Блоки фирмы Xilinx имеют максимальную тактовую частоту 450 МГц, для БПФ размером 64 точки и при входных данных с разрядностью 8 бит. Эти параметры полностью удовлетворяют нашим требованиям.

Согласно ТЗ устройство должно производить обнаружение сигнала во всей заданной полосе частот, а также определять параметры этих сигналов. Перед разработкой структуры устройства необходимо провести анализ литературы по существующим методам определения параметров сигнала по спектру.

По заданию для устройства отведено не более 40% от общего объема ПЛИС. Это обуславливается наличием различных интерфейсов работы с внешними устройствами. Для каждого интерфейса реализуется собственный контроллер. Наличие этого требования гарантирует, что хватит места для реализации этих контроллеров. Сам процесс разработки контроллера не относится к теме данной работе, поэтому информация о них не приводится.

1.2 Анализ требуемых результатов

Результатом работы устройства является информация об обнаруженных сигналах. Определяемые параметры сигнала должны помещаться в специальную структуру – дескрипторы, которые затем передаются в центральный процессор для дальнейшей обработки. По согласованию с руководителем, минимальная информация о сигнале, при которой дескриптор считается действительным, является поле несущей частоты. Таким образом, разделение сигналов по частоте является наиболее важным параметром. Перед разработкой алгоритмов определения частоты были проанализированы несколько литературных источников по существующим методам измерения частоты.

Дескрипторы должны передаваться в центральный процессор через интерфейс Ethernet, стандарта SGMII. Для работы с интерфейсом Ethernet на уровне пакетов,

у фирмы Xilinx присутствует готовый блок, который может конфигурироваться и для работы со стандартом SGMII. Данные в пакеты Ethernet будут передаваться по протоколу UDP, поскольку он обеспечивает наибольшее быстродействие и простоту реализации. Другим протоколом, поддержка которого требуется, является протокол ARP, для возможности автоматической связи устройства с процессором или ПЭВМ по сети. Для реализации этих протоколов в устройстве используется готовый блок с открытым кодом [- ссылка на блок]. Данный блок поддерживает следующие протоколы – UDP, ARP, ICMP.

Также, согласно ТЗ одним из результатов работы должна являться имитационная модель устройства, в которую входят блоки формирования сигналов, блоки БПФ и блок обнаружения и определения параметров. Модель должна быть реализована в виде конфигурируемых блоков Matlab Simulink. Для реализации модели используется версия Matlab R2015b, одной из отличительных особенностей данной версии является большое число блоков цифровой обработки, которые поддерживают формирования VHDL-кода сразу из модели. Таким образом, для создания модели необходимо придерживаться такого стиля моделирования, который позволял бы с минимальными доработками перевести модель в VHDL-код.

1.3 Обзор литературных источников

Разрабатываемое устройство представляют собой цифровой многоканальный приемник. Многоканальность обеспечивается применением БПФ, которое можно рассматривать в виде банка фильтров, с одинаковой полосой.

Аналогом устройства в литературе можно назвать цифровую систему приема и обработки радиосигналов [1]. В статье описывается структура модуля цифровой обработки – в неё входит блок БПФ, набор цифровых преобразователей частоты, набор цифровых фильтров и блоки обнаружителя-измерителя параметров. Данные со спектра используются только для настройки преобразователей частоты, выходы которых в дальнейшем анализируются. Отличие данного модуля от устройства, разрабатываемого в данной работе, заключается в более узкой

полосе пропускания входного тракта. В статье указывается полоса ± 20 МГц, с центральной частотой 70 МГц. Требования к устройству в данной работе соответствуют полосе в 500 МГц, при частоте дискретизации 1 ГГц. К тому имеется возможность использовать для анализа вторую зону Найквиста, при условии в аналоговом тракте будет установлен блок аналогового измерения частоты для избавления от неопределенности при дискретизации сигнала в устройстве. Также стоит отметить ограничение числа одновременно анализируемых сигналов. К недостаткам схемы можно отнести наличие использования корреляционной обработки, для определения временных параметров. Данный метод требует серьезных вычислительных затрат и не отвечает требованиям по скорости реакции системы на сигнальную обстановку на входе.

Также в зарубежной статье [2] описывается структура цифрового многоканального приемника с применением частотно-временного процессора. Главной особенностью данного приемника является применение метода частотно-временной обработки для анализа сигнала, с использованием данных о спектре. В статье приводятся краткое описание основных блоков приемника, таких как обнаружитель, классификатор и формирователь дескрипторов импульсов. Заявленная частотная полоса входного тракта приемника составляет 125 МГц. Частотно-временной процессор основывается на блоке БПФ с числом точек 64. Описываемая структура приемника наиболее близко подходит под задание на проектирование данной работы. Поэтому я решил более подробно изучить информацию по методу частотно-временной обработки.

В следующей статье [3] проводится обзор метода частотно-временной обработки, в частности сравниваются инструменты, которые в нем применяются. Описывается алгоритм кратковременного преобразования Фурье, преобразование Габора, преобразование Вигнера - Вилла, а также параметрические методы анализа спектра. Расписываются их достоинства и недостатки.

2 Разработка структуры устройства

В предыдущем разделе были рассмотрены наиболее тонкие моменты технического задания, были представлены результаты обзора литературы по анализу спектр. В результате в качестве прототипа устройства была выбрана схема многоканального приемника с частотно-временным процессором. В данном разделе представлено теоретическое описание принципов частотно-временной обработки сигналов, также перечислены основные инструменты, которые применяются в такой обработке. Также в разделе проводится проработка структуры устройства с учетом её дальнейшей реализации на ПЛИС.

2.1 Метод частотно-временной обработки

Широко известно, что любой сигнал можно представить бесчисленным количеством способов. Различные представления сигналов можно использовать в различных приложениях. Например, в большинстве инженерно-технических приложениях сигнал рассматривается как функция времени. Но в процессе исследования или разработки системы сигналы часто рассматриваются в частотной области. Это связано тем, что множество характеристик сигнала или системы проще описать в частотной области, нежели во временной.

В области радиотехники наиболее важными переменными являются частота и время. Функции времени описывают изменение амплитуды сигнала с течением времени, а функции частоты говорят, нам как часто эти изменения имеют место. Наиболее часто используемым инструментом для связи частоты и времени в радиотехнике является преобразование Фурье.

Основным недостатком преобразования Фурье заключается в том, что в результате теряется информация о времени. Для примера рассмотрим отрезок импульсной последовательности размером 16384 отсчета (рисунок 2.1). Частота дискретизации последовательности равна 1 ГГц, длительность импульсов равна 1 мкс, период повторения импульсов равен 3 мкс. Частота заполнения импульсов составляет 400 МГц. На рисунке 2.2 представлен результат БПФ данного отрезка,

количество точек в БПФ равно 16384 - соответствует размеру выборки. Как видно на графике спектра, мы наблюдаем высокое разрешение по частоте из-за большого количества точек. Также мы можем определить длительность и период повторения импульсов после обработки гармоник спектра. Но мы полностью потеряли информацию о времени прихода импульсов, а в системах радиомониторинга этот параметр сигнала является важным. К тому же анализ спектра большого размера представляет серьезные вычислительные затраты.

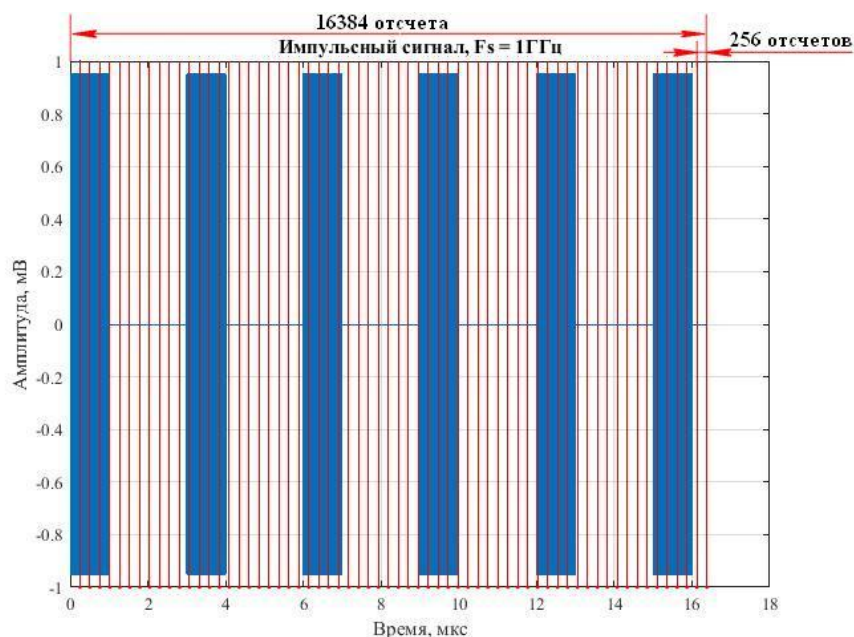


Рисунок 2.1 – Отрезок импульсной последовательности

Эта проблема обычного частотного анализа. Для его преодоления научным сообществом было предложен метод, который анализирует частотный состав сигнала вместе с сохранением информации о времени. Такой подход и называется частотно-временным анализом.

Существует множество инструментов для частотно-временного анализа, их можно грубо разделить на линейные преобразования: преобразование Габора, вейвлеты, «оконное» преобразование Фурье и квадратичные преобразования: например разложение Вигнера-Вилле.

Все эти инструменты имеют свои недостатки: преобразование Габора требует серьезных вычислительных затрат на расчет коэффициентов Габора, особенность

вейвлетов, связанная с непостоянным распределением шума на частотах, ограничивает их применение в практике, разложение Вигнера-Вилле имеет высокое частотно-временное разрешение, но порождают ложные компоненты в спектре из-за квадратичной формы преобразования. Что касается кратковременного преобразования Фурье, то ему присущи все проблемы обычного преобразования Фурье: Необходимость компромисса между разрешающей способностью по времени и по частоте, явление утечки спектра.

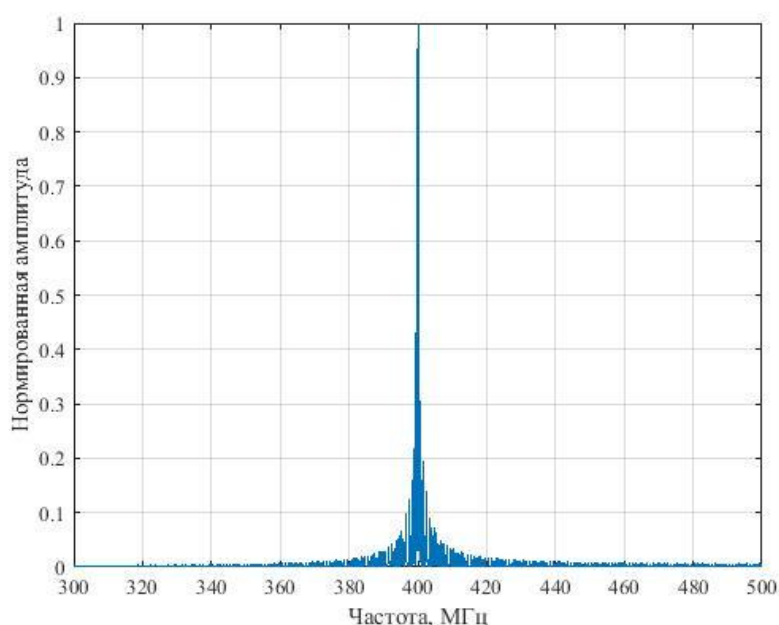


Рисунок 2.2 – Спектр импульсной последовательности
(БПФ на 16384 отсчетов)

В отличие от других перечисленных инструментов частотно-временного анализа кратковременное преобразование Фурье имеет одну положительную особенность. Его можно реализовать с использованием обычного БПФ. Это позволяет упростить практическое применение частотно-временного анализа.

Возвращаясь к примеру с импульсной последовательностью, рассмотрим частотно-временное представление сигнала с использованием БПФ на 256 точек. На рисунке 2.1 показано деление отрезка на интервалы в 256 отсчетов для лучшего представления сути операции. На рисунке 2.3 мы наблюдаем эволюцию спектра сигнала на данном отрезке. По этим данным можно четко сказать, когда

начался импульс, измерить его длительность и период повторения импульсов, подсчитать количество импульсов. Но из-за уменьшения количества точек БПФ мы ухудшили частотное разрешение, поэтому здесь следует искать компромисс между разрешением по частоте и времени.

С использованием данного метода появилась возможность разрабатывать частотно-временные процессоры – блок, реализующий частотно-временной анализ, и выводящий полученные результаты в удобном для системы обработки виде.

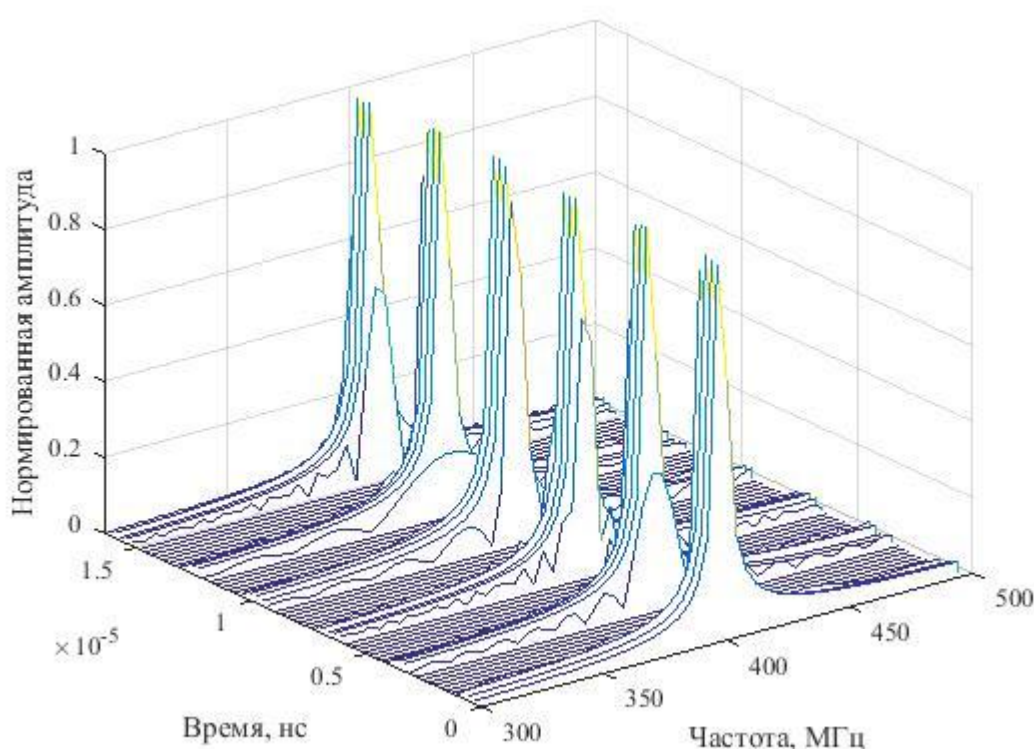


Рисунок 2.3 – Частотно-временное представление отрезка импульсной последовательности (размер БПФ 256 точки)

2.2 Кратковременное преобразование Фурье

Кратковременное преобразование Фурье (КПФ) представляет собой модификацию традиционного преобразования Фурье. Вместо обработки всей выборки сигнала за раз, КПФ производит преобразование последовательно блок за блоком (рисунок 2.4). Таким образом, результирующее преобразование можно

рассматривать как поведение частот сигнала в течении рассматриваемого блока данных.

Выражение для расчета КПФ имеет следующий вид:

$$STFT[mT, n\Omega] = \int_{-\infty}^{+\infty} s(t)\gamma(t - mT)e^{-jn\Omega t} dt, \quad (2.1)$$

где mT – дискретное время сигнала;

$n\Omega$ – дискретная частота сигнала;

$s(t)$ – рассматриваемый сигнал;

$\gamma(t)$ – весовой функции окна.

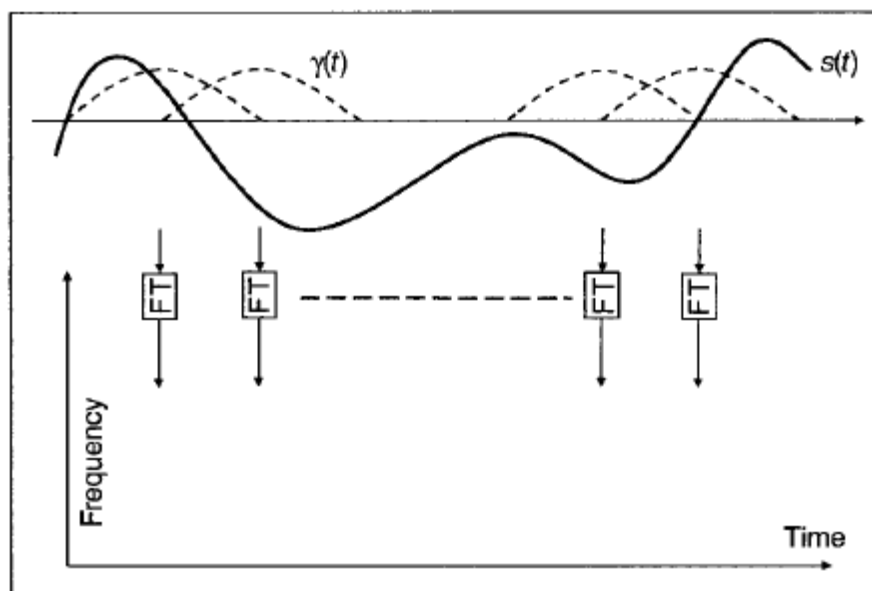


Рисунок 2.4 – Принцип кратковременного преобразования Фурье

Весовая функция окна предназначена для балансировки разрешения по времени и по частоте. Чем меньше время весовой функции, тем лучше временное разрешение и тем хуже частотное разрешение. Блоки сигнала могут перекрываться, или не пересекаться. Процент перекрытия между каждым блоком определяется шагом дискретизации и длиной весовой функции окна $\gamma(t)$.

2.3 Структура многоканального приемника с частотно-временным процессором

После анализа литературы, в которой описываются схожие по назначению средств, за прототип разрабатываемого мной устройства я решил взять цифровой многоканальный приемник с частотно-временным процессором, который описывается в статье [2]. Структура приемника, описанная в статье, представлена на рисунке 2.5.

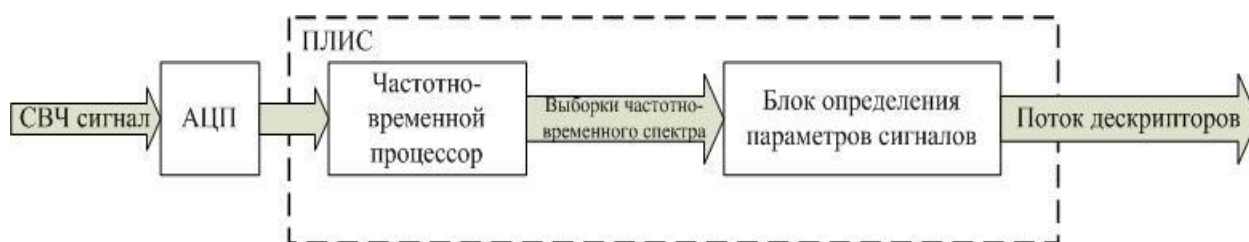


Рисунок 2.5 – Схема приемника с частотно-временным процессором

Частотно-временной процессор представляет собой блок БПФ, который используется для формирования кратковременного преобразования Фурье. Длина преобразования блока БПФ составляет 64 точки. Весовая функция окна рассчитывалась по методу Паркса-Макклирена [4], который рассчитывает оптимальный фильтр с линейной ФЧХ, который удовлетворяет требуемой маске фильтра. Параметры весовой функции окна указывались следующие: для 64-канального приемника, с неравномерностью в полосе пропускания ± 1 дБ и уровнем полосы затухания 60 дБ окно имеет длину в 256 отсчетов. Внешний вид окна показан на рисунке 2.6.

Поскольку длина весового окна больше длины блока БПФ, то в статье применяется метод наложения во времени. Он подразумевает деление большого временного интервала на меньшие части, по размеру совпадающие с длиной БПФ, после чего эти части суммируют и отправляются на блок БПФ. В результате, получается вычислять БПФ с сохранением преимуществ временного окна большей длины. Более того данные на БПФ подаются с децимацией во времени.

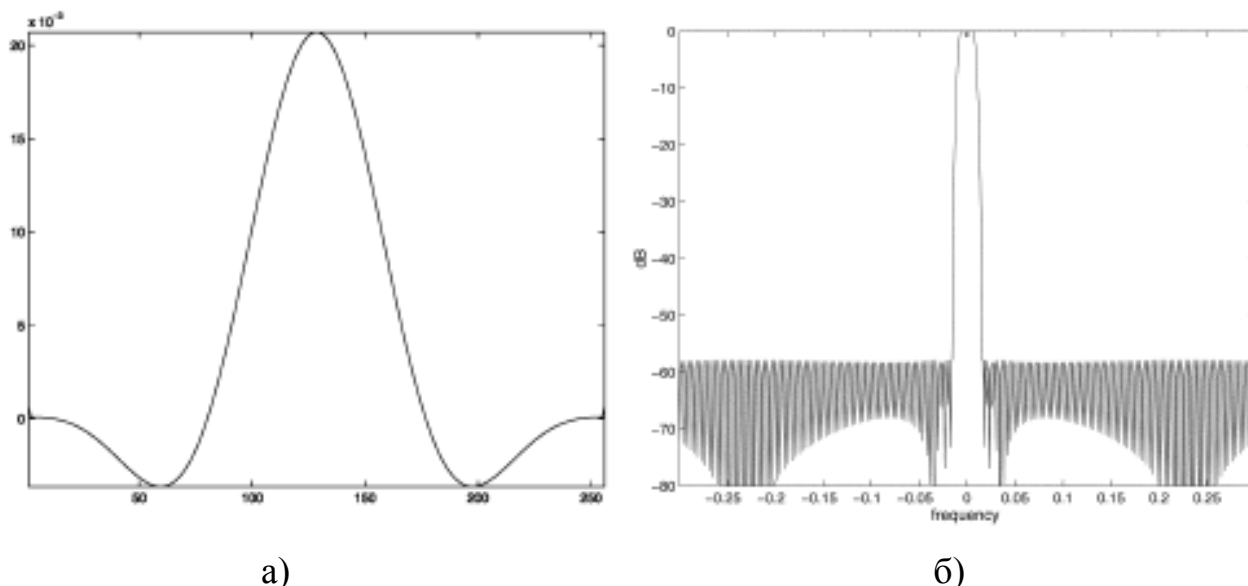


Рисунок 2.6 – Весовая функция окна для 64-канального приемника во временной области (а) и в частотной области (б)

Коэффициент децимации составляет 32 отсчета. Это позволило снизить вычислительные затраты для расчета БПФ, поскольку окно анализа сдвигается на 32 отсчета вместо одного, как в формуле (2.1).

Несмотря на то, что структура приемника, описанная в статье, подходит требуемому заданию, необходимо проработать множество деталей, которые не описываются в статье. К тому же я решил пересмотреть некоторые решения авторов статьи. В частности, я решил отказаться от децимации во времени, поскольку используется современная ПЛИС, с большим количеством элементов для цифровой обработки сигналов (ЦОС).

2.4 Разработка подробной структуры устройства

Первое, что нужно сделать для модификации выбранного прототипа, это проработать более подробную структуру разрабатываемого устройства. Для облегчения дальнейшего моделирования и разработки программы для ПЛИС, следует продумать рациональное деление структуры на модули.

На входе устройства располагается частотно-временной процессор, который представляет собой блок БПФ с определенным весовым окном. Поскольку по ТЗ необходимо измерять разность фаз, то в устройстве должно быть предусмотрено

два АЦП, на которые поступают данные от двух антенн. Таким образом, частотно-временной процессор включает себя два блока БПФ.

На выходе БПФ появляются данные в декартовых координатах (действительная и мнимая часть), при этом гораздо удобнее проводить обработку в полярных координатах (амплитуда и фаза). Наиболее известным способом перевода из декартовых координат в полярные является применение алгоритма CORDIC – coordinate rotation digital computer, цифровой вычислитель поворота системы координат. Он представляет собой итерационный метод сведения прямых вычислений сложных функций к выполнению простых операций сложения и сдвига. Фирма Xilinx предлагает к использованию готовый блок алгоритма CORDIC [5].

Первое, что должно выполнять устройство с выборками спектра – это проводить обнаружение, а также измерение частоты сигнала. Самая простая в реализации схема обнаружения представляет собой пороговый обнаружитель с одним порогом. В сети есть различные статьи, в которых приводятся характеристики обнаружения для систем с БПФ [6]. Наиболее часто выбирают порог по фиксированному значению вероятности ложной тревоги обнаружителя. Используемая в статье схема БПФ не соответствует разрабатываемой в данной работе, поэтому одним из этапов работы было исследование характеристик обнаружения для схемы БПФ из 256 точек.

После определения параметров обнаружителя, необходимо проработать алгоритмы определения параметров сигнала. Согласно ТЗ необходимо определять такие параметры как несущая частота, длительность импульса, период повторения импульса, пиковая мощность, время прихода сигнала, разность фаз между каналами. При этом следует учесть, что необходимо анализировать данные о спектре сигнала. При этом логично в качестве приоритета использовать параметр несущей частоты сигнала, этот параметр будет являться минимально необходимым для формирования дескриптора сигнала. Таким образом, зная, что на входе БПФ поступает действительный сигнал, и число каналов равно 256,

можно просчитать эффективное число каналов. Оно составляет 127 каналов, поскольку 0 канал не несет полезной информации и практически никогда не используется при анализе спектральных данных. В устройстве я решил использовать многоканальную систему обработки. При этом число каналов равно эффективному количеству каналов. При такой реализации параметр несущая частота представляет собой просто номер канала обработки.

После обнаружения и измерения несущей частоты сигнала устройство начинает анализировать другие измеряемые параметры. Прежде всего, проходит классификация импульсного сигнала (ИС) и непрерывного сигнала (НС). Критерием для непрерывного сигнала является длительность импульса, превышающая 600 мкс. Поскольку часть параметров относятся только к ИС, то я решил разделить обработку ИС и НС в разные блоки. Следующим шагом будет установка правил формирования дескрипторов импульса, которые подробно расписываются в описании блока формирования дескрипторов.

Сформированные дескрипторы необходимо передать по интерфейсу Ethernet в ПЭВМ или ЦП. Поскольку в устройстве реализована многоканальная обработка, то важной необходимостью стала создание алгоритма формирования выходной очереди дескрипторов. Алгоритм заключается в кольцевом опрашивании всех каналов на наличие дескрипторов, при наличии дескрипторов на выходе канала он укладывается в выходную очередь.

Структура разрабатываемого устройства представлена на рисунке 2.7.

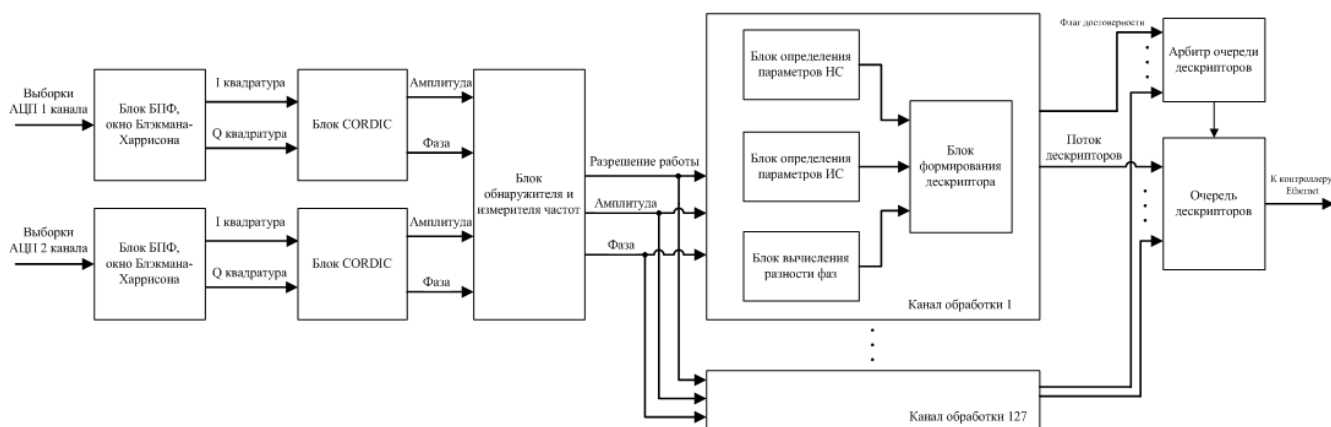


Рисунок 2.7 – Структура разрабатываемого устройства

В дальнейших подразделах представлено более подробное описание основных блоков устройства.

2.5 Выбор весового окна анализа для БПФ

Основная функция весового окна заключается в уменьшении эффекта утечки спектра, за счет уменьшения разрешения по частоте блока БПФ. Основным параметром временного окна является уровень подавления боковых лепестков, который определяет динамический диапазон системы на выходе БПФ. Этот уровень не должен быть меньше динамического диапазона на входе блока БПФ, иначе боковые лепестки от весового окна будут искажать информацию о сигнале.

Динамический диапазон на входе устройства определяется разрядностью АЦП. По заданию количество разрядов АЦП – 10 бит. Формула для расчета динамического диапазона исходя из числа разрядов приведена в (2.2):

$$D_c = 6N = 6 \cdot 10 = 60 \text{ [дБ]}, \quad (2.2)$$

где D_c – динамический диапазон системы в дБ; N – количество разрядов.

Таким образом, минимальный уровень боковых лепестков для весового окна должен быть не ниже 60 дБ. В статье [7] проводится обзор наиболее популярных видов весовых окон для применения в БПФ. Приводятся параметры, нормированные относительно длины окна, для каждого вида окна. Таблица со списком параметров представлена в таблице 2.1. В таблице указаны следующие параметры: $\Delta F_{0.5}$ – нормированная ширина главного лепестка по уровню 0.5; ΔF_0 – нормированная ширина главного лепестка по уровню 0; γ_{max} – максимальный уровень боковых лепестков, дБ; K – коэффициент расширения главного лепестка по сравнению с прямоугольным лепестком; β – коэффициент ослабления окна, дБ.

Согласно таблице 2.1, по динамическому диапазону подходит окно Блэкмана-Харриса – 92 дБ. Из параметров окна мне также потребуются коэффициент ослабления окна, для учета измеренной мощности сигнала, и коэффициент расширения главного лепестка по сравнению с прямоугольным окном. По заданию количество точек БПФ в устройстве равно 256, таким же должна быть

длина весового окна. Внешний вид во временной и частотной области показан на рисунке 2.8.

Таблица 2.1 – Свойства оконных функций

Наименование окна	$\Delta F_{0,5}$	ΔF_0	K	γ_{max} , дБ	β , дБ
Прямоугольное окно (rectangle window)	2	0,89	1	-13	0
Синус-окно	3	1,23	1,5	-23	-3,93
Окно Ланцоша (Lanczos window), или sinc - окно	3,24	1,3	1,62	-26,4	-4,6
Окно Барлетта (Bartlett window), или треугольное окно	4	1,33	2	-26,5	-6
Окно Ханна (Hann window)	4	1,5	2	-31,5	-6
Окно Барлетта — Ханна (Bartlett–Hann window)	4	1,45	2	-35,9	-6
Окно Хемминга (Hamming window)	4	1,33	2	-42	-5,37
Окно Блэкмана (Blackman window)	6	1,7	3	-58	-7,54
Окно Блэкмана — Харриса (Blackman–Harris window)	8	1,97	4	-92	-8,91
Окно Наталла (Nuttall window)	8	1,98	4	-93	-9
Окно Блэкмана — Наталла (Blackman–Nuttall window)	8	1,94	4	-98	-8,8
Окно с плоской вершиной (Flat top window)	10	3,86	5	-69	0
Окно Гаусса (Gaussian window) $\sigma = 0,3$	8	1,82	4	-65	-8,52
Окно Гаусса (Gaussian window) $\sigma = 0,5$	3,4	1,2	1,7	-31,5	-4,48
Окно Гаусса (Gaussian window) $\sigma = 1,2$	2,2	0,94	1,1	-15,5	-0,96

Коэффициент $K = 4$ показывает, что ширина главного лепестка для данного окна в четыре раза больше, по сравнению со случаем без окна. Это влечет за собой некоторые трудности, связанные с различением сигнала в спектре. В частности, если на вход БПФ подать отрезок с одним сигналом, частота которого соответствует центру одного канала БПФ – влияние утечки минимальное, тогда для прямоугольного окна будет зафиксирована гармоника в одном канале. В случае применения данного окна будет наблюдаться гармоники в нескольких каналах. Пример этой ситуации показан на рисунках 2.8 - 2.9. Данная проблема повлияла на разработку алгоритмов измерения частоты, об этом подробнее расписано в дальнейших разделах.

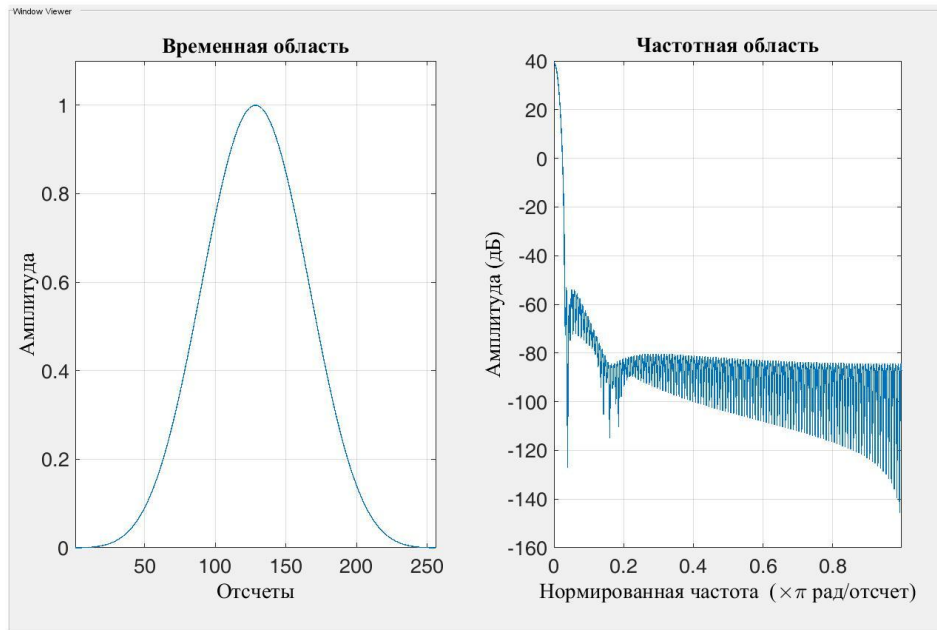


Рисунок 2.8 – Внешний вид весового окна Блэкмана-Харриса с длиной 256 отсчетов

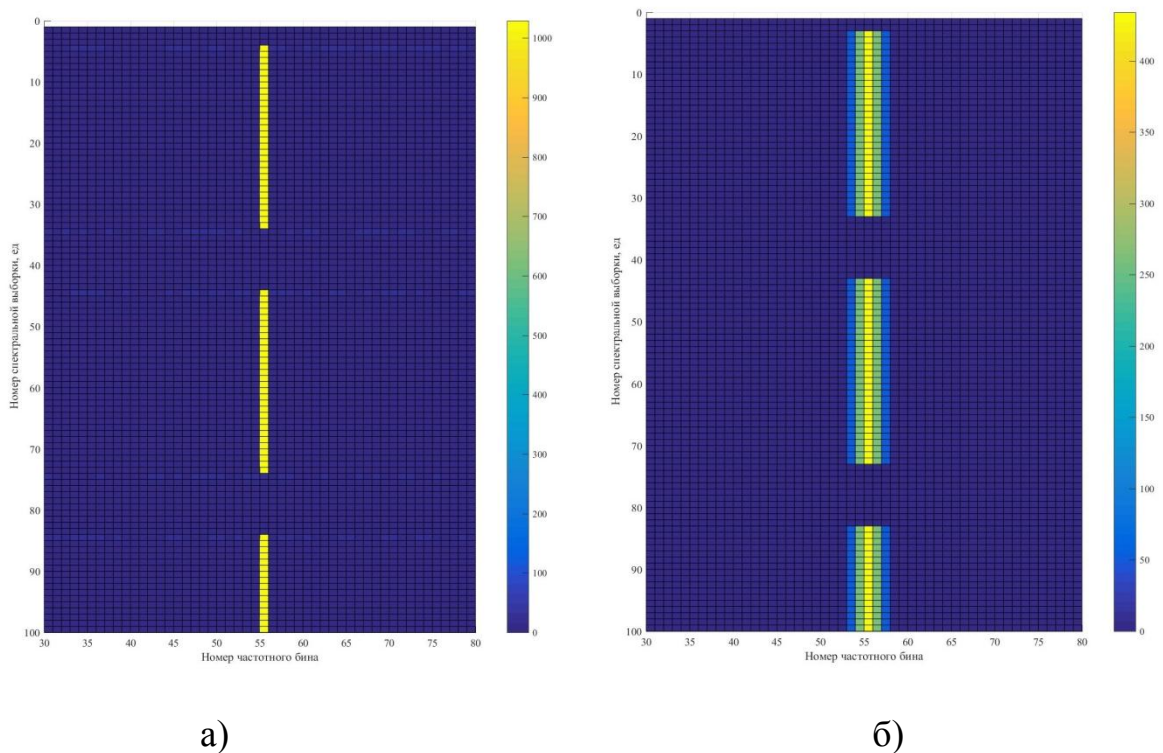


Рисунок 2.9 – Частотно-временное представление сигнала для прямоугольного окна (а) и для окна Блэкмана-Харрисона (б)

2.6 Исследование характеристик обнаружителя на основе БПФ

Основная задача блока обнаружения заключается в анализе выборок спектра и получение информации о присутствии сигнала в конкретной частоте. На вход блока поступают значения амплитуды и фазы с двух каналов обработки, для обнаружения сигнала будет использоваться только информация об амплитуде сигнала на выходе БПФ, данные о фазе передаются на выход блока с учетом задержки на обнаружение и измерение частоты сигнала. На выходе блок передает служебные сигналы, такие как строб обнаружения, разрешение работы для каналов. Помимо служебных сигналов на выход транслируется информация с БПФ для каждого канала обработки.

В качестве схемы обнаружителя была взята самая простая схема с одним порогом срабатывания. Выбор порога определяется требуемыми значениями вероятности ложной тревоги и правильного обнаружения. Теоретический расчет уровня порога подробно расписан в литературе [8]. В радиолокации наиболее часто при расчете обнаружителей пользуются критерием Неймана-Пирсона. Согласно этому критерию выбирается такое правило обнаружения, которое обеспечивает максимальную вероятность правильного обнаружения при условии, что вероятность ложной тревоги не превышает заданной величины. Задача обнаружения в данном случае представляет собой случай двух гипотез. Описание этих гипотез представлено в (2.3 и 2.4).

$$H_0: x = n, \quad (2.3)$$

$$H_1: x = s + n, \quad (2.4)$$

где n – белый гауссовский шум с нулевым математическим ожиданием и известной мощности. Мощность шума известна, поскольку данный шум в основном определяется внутренними шумами входного тракта приемника.

Ключевой особенностью БПФ является равномерность шумовой характеристики в каждом бине. В литературе [2] представлена следующая схема обнаружения на основе БПФ – на выходе каждого канала проводится отдельное обнаружение, значение порога в каждом канале выбирается с учетом общей

вероятности ложной тревоги во всем рассматриваемом диапазоне частот (P_{FAg}).

Формула для обнаружителя в каждом канале представлена в (2.3):

$$A_i(m, i) \underset{H_0}{\overset{H_1}{>}} q_i, \quad i = 1, 2, \dots, N \quad (2.3)$$

где A_i – амплитуда сигнала на выходе i -го бина БПФ;

m – дискретное время выборки спектра;

i – номер бина БПФ;

q_i – значение порога в i -м бине БПФ. Требование для уровня порогов представлено в (2.4):

$$P_{FAg} = P_{H_0} \left(\bigcup_{i,m} \{A_i(m, i) > q_i\} \right). \quad (2.3)$$

Поскольку в ТЗ не обговаривается требуемый уровень ложной тревоги, необходимо исследовать возможности обнаружителя на основе БПФ для различных уровней порога. Также в свободном доступе в литературе не представлены характеристики обнаружителя на основе БПФ, поэтому было интересно получить эти характеристики для данного устройства. Исследования проводились на разработанной модели, описание модели представлено в разделе 3. Порядок исследований был следующий: сначала получили зависимость вероятности ложной тревоги и правильного обнаружения от уровня порога срабатывания, при различных значениях спектральной плотности мощности, затем рассчитали зависимость вероятности правильного обнаружения от отношения сигнал-шум. На рисунке 2.10 представлена полученная зависимость вероятности ложной тревоги от нормированного порога срабатывания, для различных спектральных плотностей мощности шума $N_{\text{шум}} = 10^{-2}, 10^{-3}, 10^{-4} \left[\frac{\text{Вт}}{\text{Гц}} \right]$. Порог нормировался относительно спектральной плотности мощности шума.

По рисунку видно, что вероятность ложной тревоги на уровне 10^{-3} обеспечивается при выборе нормированного порога на уровне минус 4.5... минус 3.5 дБ. Отрицательное значение нормированного порога можно объяснить

фильтрующей способностью блока БПФ, накопление данных позволяет выделять полезную информацию даже при малом отношении сигнал-шум.

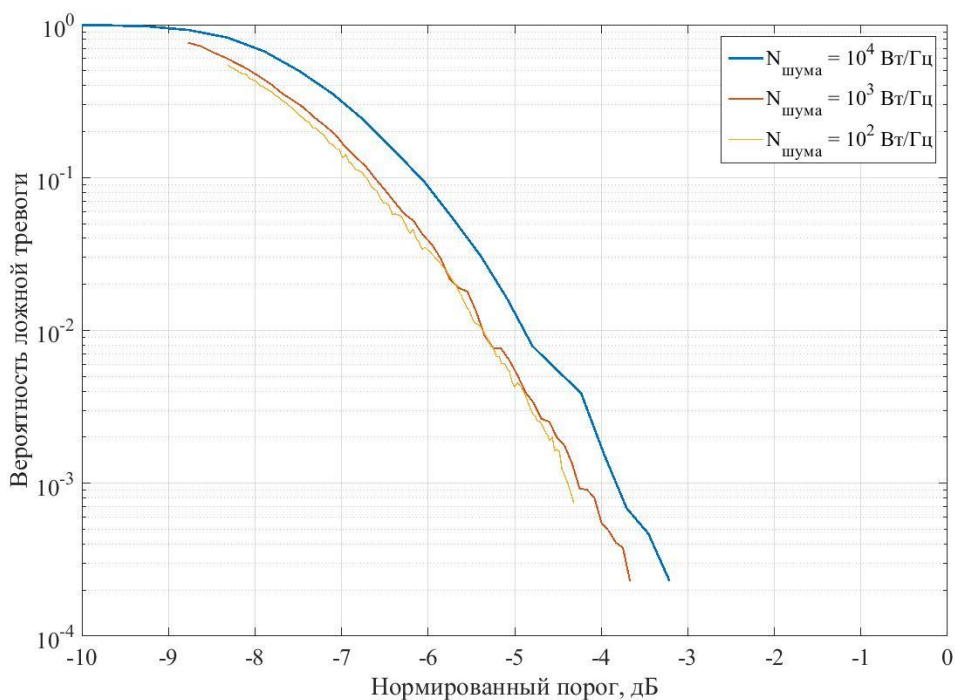


Рисунок 2.10 – Зависимость вероятности ложной тревоги от порога срабатывания

На рисунке 2.11 показана зависимость вероятности правильного обнаружения от уровня порога срабатывания, для спектральной плотности мощности $N_{\text{шум}} = 10^{-2} \left[\frac{\text{Вт}}{\text{Гц}} \right]$. Отношение сигнал-шум было зафиксировано на уровне 0 дБ. Как видно по рисунку на уровне порога минус 4.5... минус 3.5 дБ вероятность правильного обнаружения достигает на уровне 0.9.

На рисунке 2.12 представлена зависимость вероятности правильного обнаружения от отношения сигнал-шум для двух значений вероятности ложной тревоги $P_f = 10^{-3}, 10^{-2}$. Данный график представляет собой характеристику обнаружения и является итогом исследования блока обнаружения.

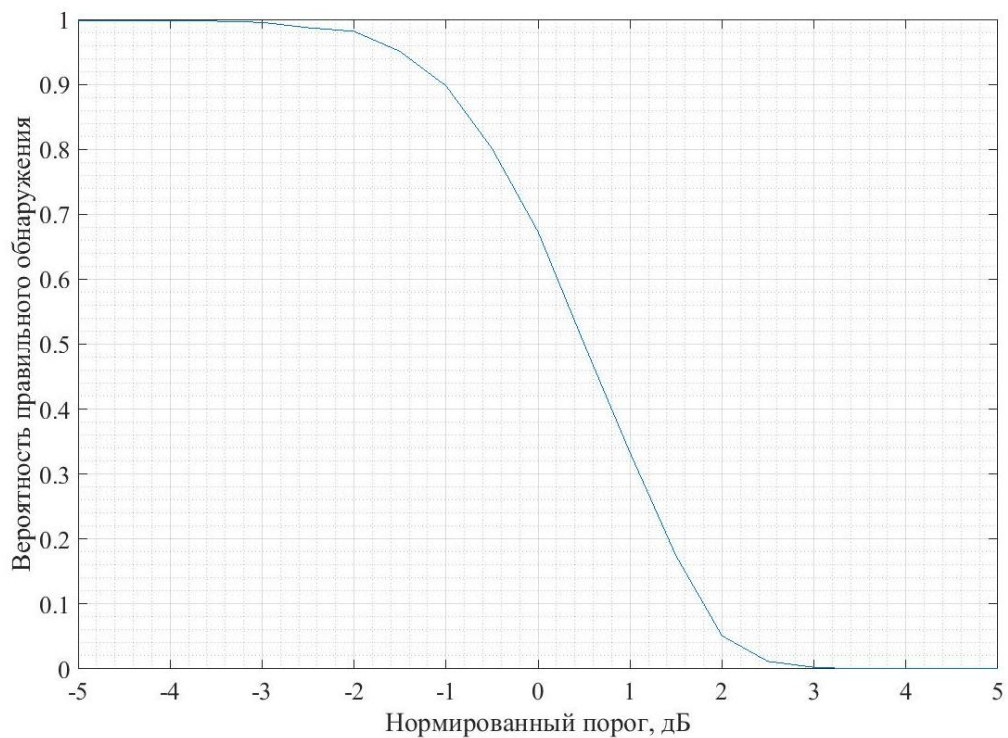


Рисунок 2.11 – Зависимость вероятности правильного обнаружения от порога срабатывания

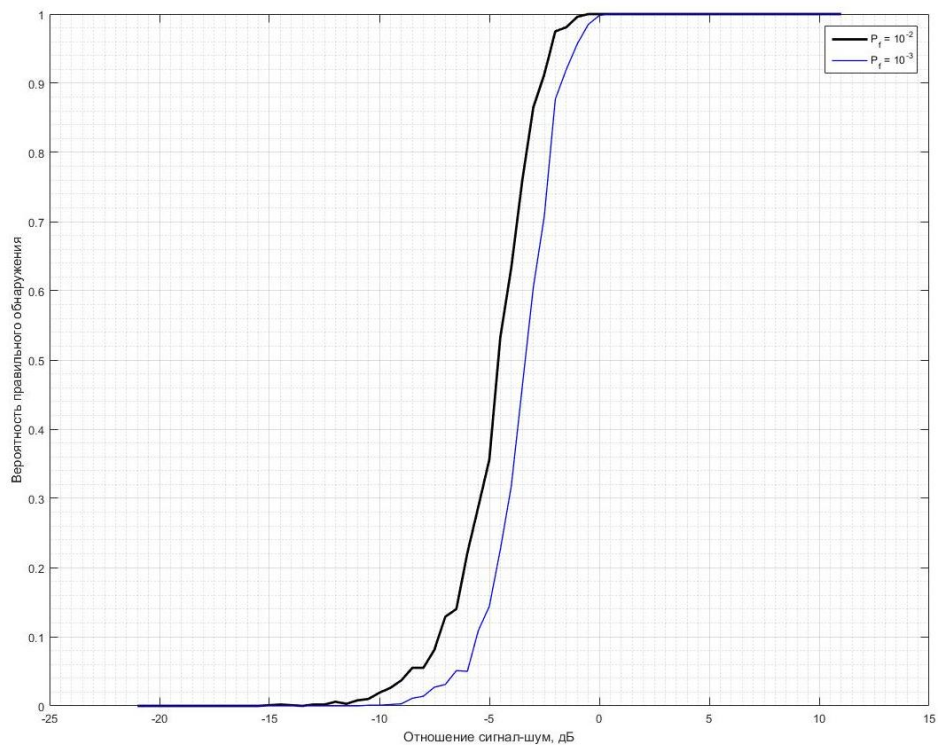


Рисунок 2.12 – Зависимость вероятности правильного обнаружения от отношения сигнал-шум

Подводя итог исследованию обнаружителя можно отметить следующее: общая вероятность ложной тревоги, которой должны удовлетворять обнаружители во всех каналах составляет $P_f = 10^{-3}$; нормированный порог срабатывания для данной вероятности составляет минус 4 дБ; вероятность правильного обнаружения для данной вероятности ложной тревоги достигает значения 0,9 при отношении сигнал-шум минус 2,5 дБ. Работа обнаружителя при отрицательном отношении сигнал-шум объясняется эффектом накопления информации в блоке БПФ.

2.7 Разработка алгоритмов измерения частоты

Для разрабатываемого устройства было принято решение реализовать многоканальную систему с параллельной обработкой данных от всех эффективных бинов БПФ. Использование современной ПЛИС позволяет реализовывать полностью параллельную систему обработки.

После исследования характеристик обнаружения обнаружена следующая особенность работы БПФ. Она заключается в том, что всегда в спектре будут присутствовать множественные превышения порога обнаружения, даже при наличии одного сигнала. Эта особенность связана с эффектом утечки спектра, а также с использованием весового окна анализа для БПФ. Утечка спектра заключается в просачивании энергии сигнала в соседние каналы БПФ. На рисунке 2.13 показано частотно-временное представление ситуации, когда присутствует один импульсный сигнал. Использовано прямоугольное временное окно анализа. Частота сигнала совпадает с центром бина, поэтому утечка спектра минимальна. Но это частный случай, который в реальности практически не будет появляться. На рисунке 2.14 показан случай, когда частота сигнала не совпадает с центром бина БПФ. Здесь явно видны последствия утечки спектра. Из-за прямоугольного окна спектр растекается по всем бинам БПФ. Для уменьшения эффекта утечки применяется временное окно анализа. Оно позволяет исключить растекание энергии сигнала по всему спектру. Недостатком применения окна является эффект расширения главного лепестка, то есть даже один сигнал будет

давать отклик в нескольких бинах БПФ. Пример представлен на рисунках 2.15, 2.16.

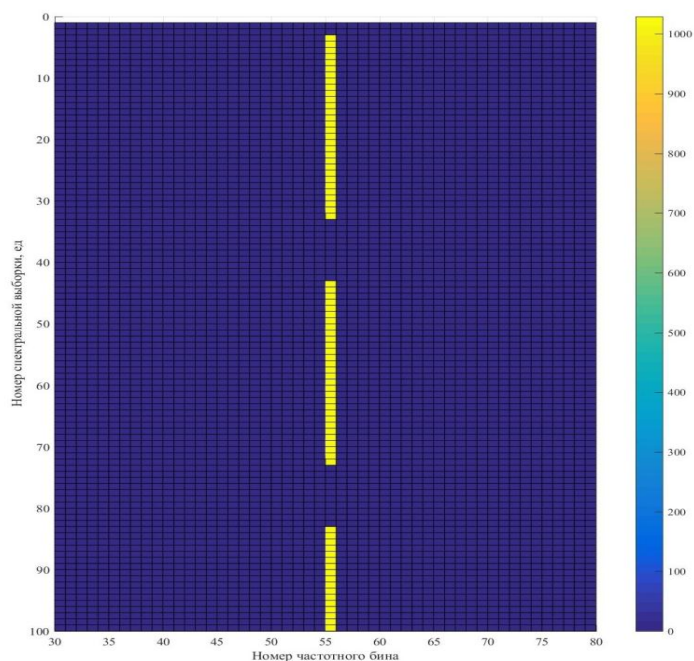


Рисунок 2.13 – Частотно-временное представление для одного импульсного сигнала, частота совпадает с центром 54 бина (прямоугольного окна)

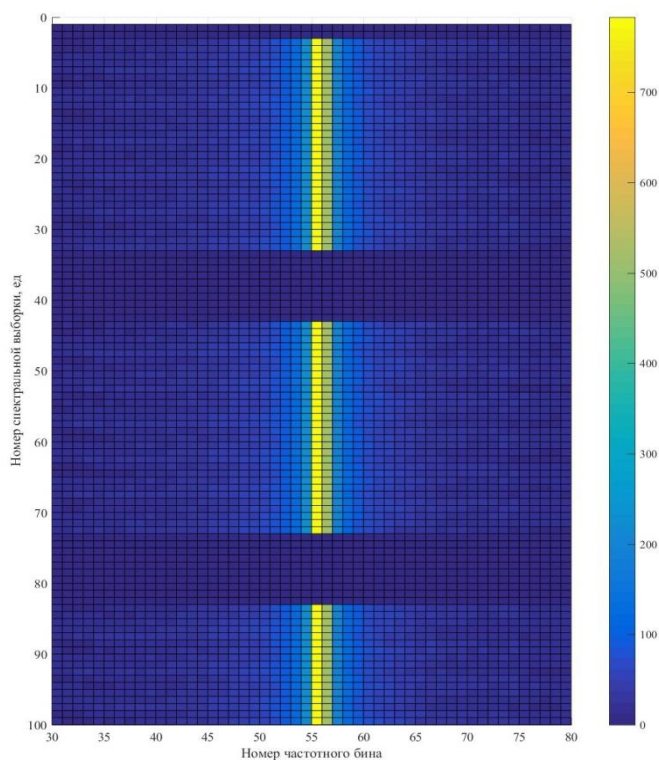


Рисунок 2.14 – Частотно-временное представление для одного ИС, частота не совпадает с центром 54 бина (прямоугольное окно)

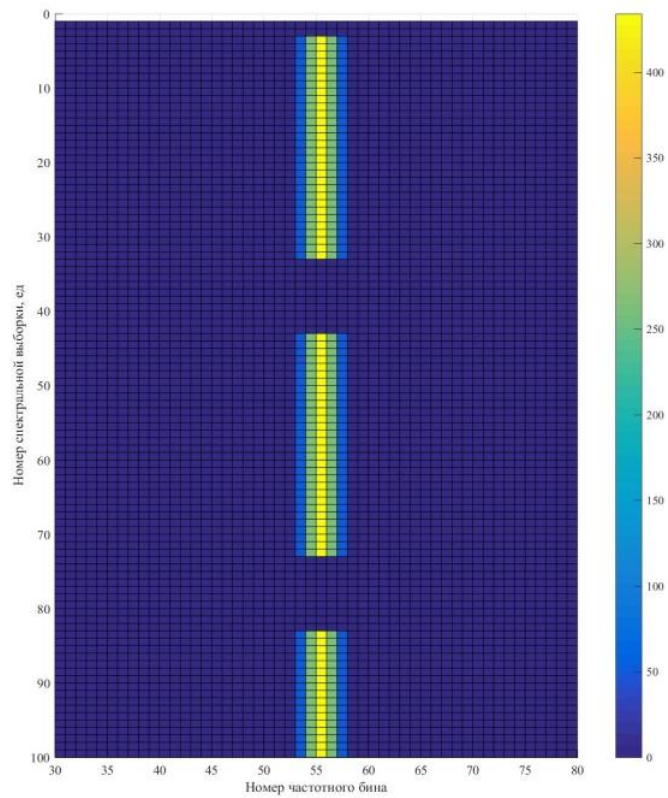


Рисунок 2.15 – Частотно-временное представление для одного импульсного сигнала, частота совпадает с центром 54 бина (окно Блэкмана-Харрисона)

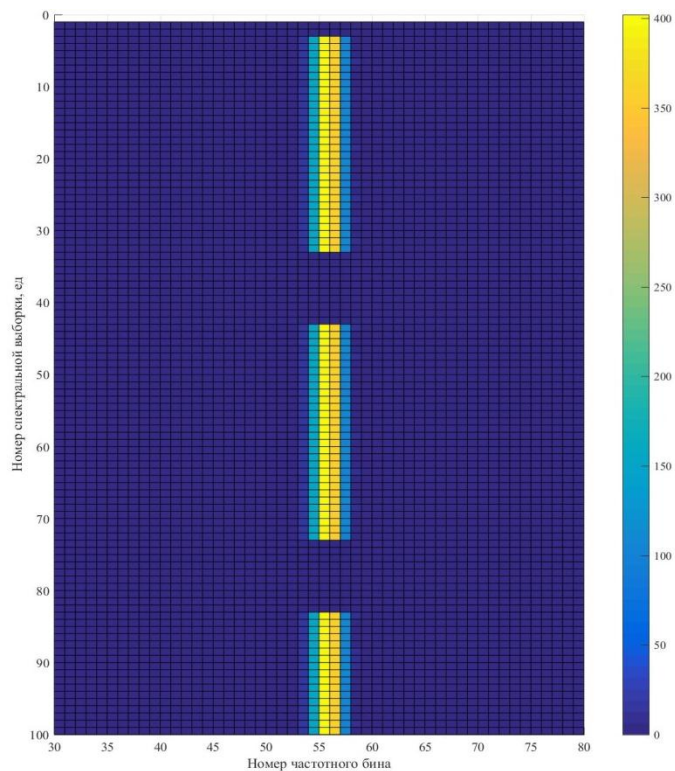


Рисунок 2.16 – Частотно-временное представление для одного импульсного сигнала, частота не совпадает с центром 54 бина (окно Блэкмана-Харрисона)

Как видно по рисунку 2.16 даже при попадании частоты сигнала на стык каналов растекания спектра не наблюдается. Но в то же время становится невозможным измерение частоты с помощью схемы обнаружения с одним порогом. Для решения этой проблемы я решил использовать дополнительную обработку данных об амплитуде сигнала.

Применение окна Блэкмана-Харрисона, (рисунок 2.15), расширяет один сигнал на четыре соседних бина БПФ. При этом анализируя распределения амплитуд составляющих спектра можно получить более точное значение частоты сигнала. Самый простой способ анализа амплитуды сигнала заключается в определении максимального значения из превышающих порог гармоник. Это значение принимается за истинную частоту сигнала.

Для облегчения реализации на ПЛИС, алгоритм измерения частоты должен работать последовательно, в скользящем окне, пробегая по всей выборке спектра, поступающей от БПФ. Алгоритм работы измерителя частоты был предложен следующий:

а. В обработку поступают те бины БПФ, в которых зафиксировано превышение порога сигнала. При этом происходит запоминание текущего бина, предыдущего бина и последующего бина;

б. Значение амплитуды в текущем бине сравнивается с двумя другими, если оно больше остальных, то этот бин принимается за максимальный и выставляется флаг разрешения работы соответствующего канал обработки;

в. Если текущий бин имеет номер 0 или 127 (началу и концу спектра соответственно), тогда он сравнивается только с одним из двух бинов. Нулевой сравнивается с последующим, а 127-й сравнивается с предыдущим.

В процессе моделирования данный алгоритм показал исправную работу на идеальных тестовых сигналах. Но в процессе отладки программы на макете обнаружили недостатки в данном алгоритме. Прежде всего, схема измерения частоты не правильно работала, когда частота сигнала попадала по центру между двумя бинами БПФ. В этом случае происходило смещение максимума амплитуды,

в результате схема давала разрешение то одному каналу обработки, то другому. Данная ситуация показана на рисунке 2.17. Информация на рисунке соответствует сигналу разрешения для каналов обработки.

Как итог, в устройстве происходило неправильное подключение каналов обработки. Данная проблема является широко известной для любых многоканальных систем, в которых есть перекрытие каналов. Для решения этой проблемы я решил модернизировать схему определения максимума. Модификация алгоритма работы схемы заключается в анализе разности амплитуд соседних каналов, если она не превышает порогового значения, тогда включать в обработку оба канала. Это позволит правильно фиксировать временные параметры сигналов, пусть и в двух каналах обработки одновременно, а задачу объединения двух сигналов в один можно поручить ЦП. Такая ситуация гораздо проще фиксируется и разрешается средствами ЦП. Результат работы модифицированной схемы показан на рисунке 2.18.

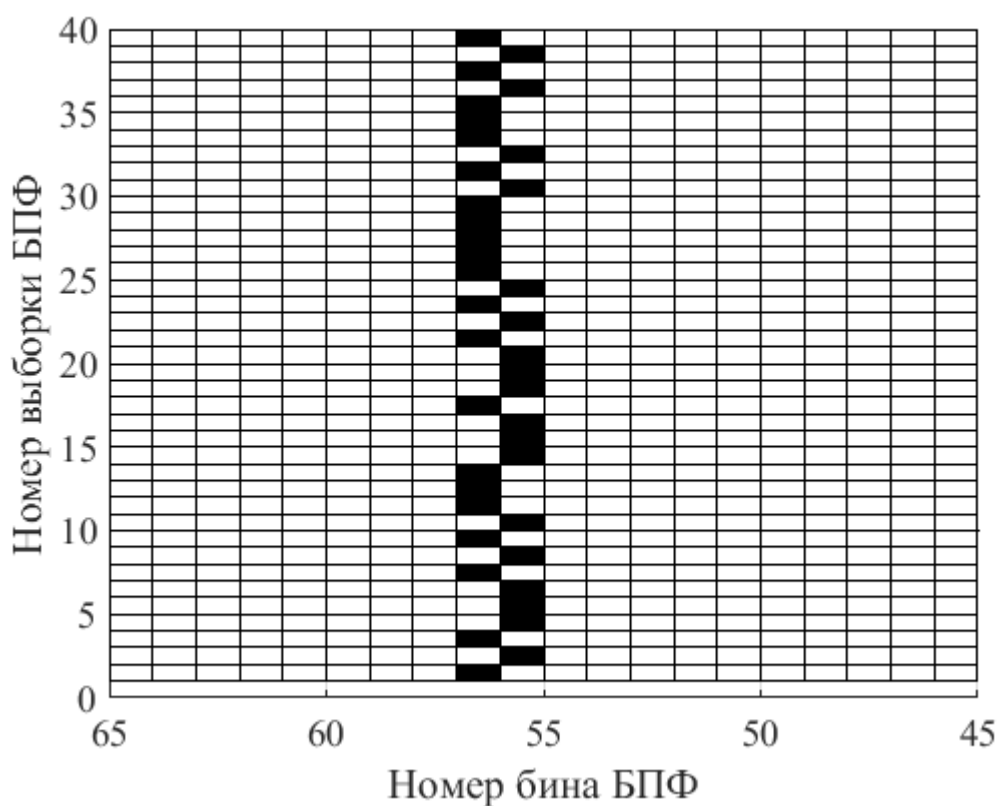


Рисунок 2.17 – Результат работы схемы определения максимума (черным показан зафиксированный максимум)

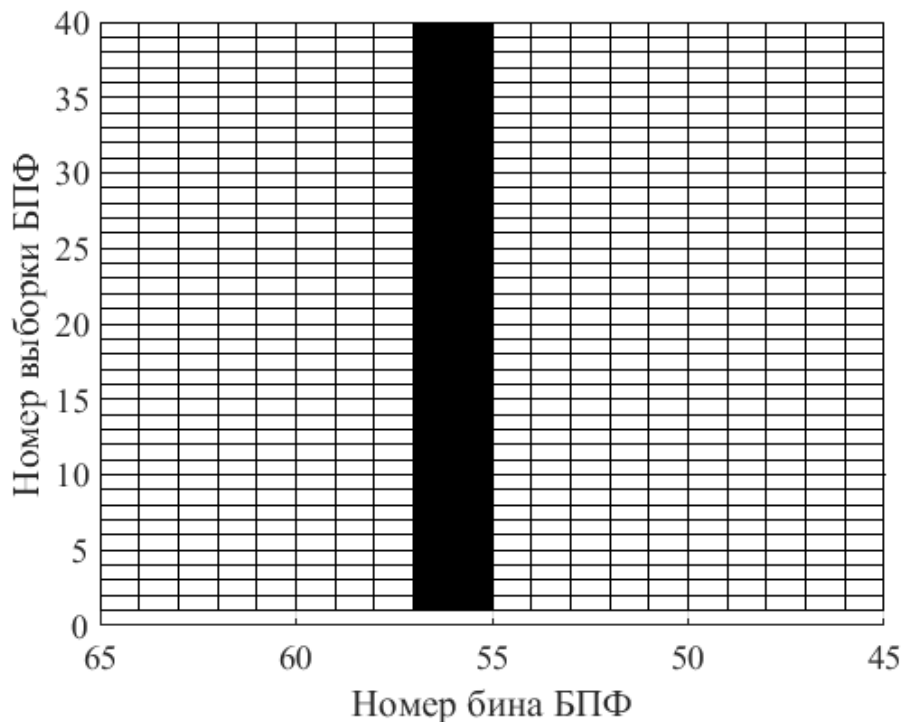


Рисунок 2.18 – Результат работы модифицированной схемы определения максимума (черным показан зафиксированный максимум)

Следующая ситуация, в которой схема работала не эффективно, происходила при подаче на вход импульсного сигнала, длительность которого не попадает на целое число выборок БПФ. В результате передний или задний фронт импульса в спектре расплывался, что приводило к сбою в определении максимума, и даже к ложному срабатыванию, когда фиксировались несколько максимумов в спектре одного сигнала. Это явление объясняется тем фактом, что фронты сигнала не полностью заполняют выборку БПФ, таким образом, мы имеем на входе импульс с длительностью меньше чем окно анализа. Как известно из теории, чем короче длительность импульса, тем шире его спектр. Это явление в зарубежной литературе называется *rabbit-ear effect* – эффект заячьих ушей. Название происходит от того факта, что если провести обратное преобразование Фурье над полученной выборкой, то фронты импульса будут превышать амплитуду сигнала по центру импульса. На рисунке 2.19 приведена ситуация для импульсного сигнала, смещенного на половину выборки спектра.

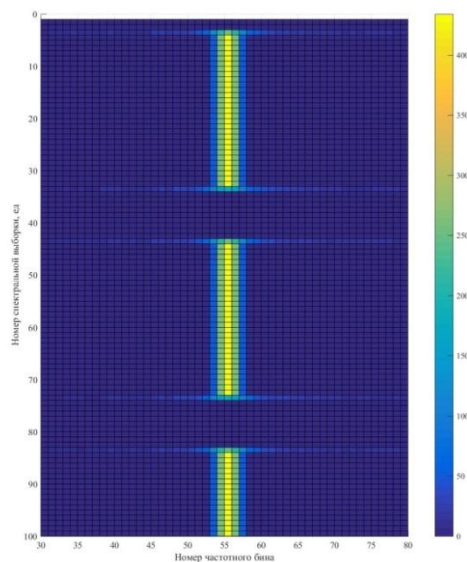


Рисунок 2.19 – Частотно-временное представление ситуации эффекта просачивания фронтов ИС

Как видно из рисунка, растекание спектра на фронтах импульса приводит к уменьшению амплитуды по центру скопления частот. Таким образом, возможны ситуации, когда максимум смещается относительно реальной частоты сигнала. В этом случае будут активизироваться лишние каналы обработки информации. И в конечном итоге устройство будет формировать ложные дескрипторы сигнала. Результат работы схемы определения максимума представлен на рисунке 2.20.

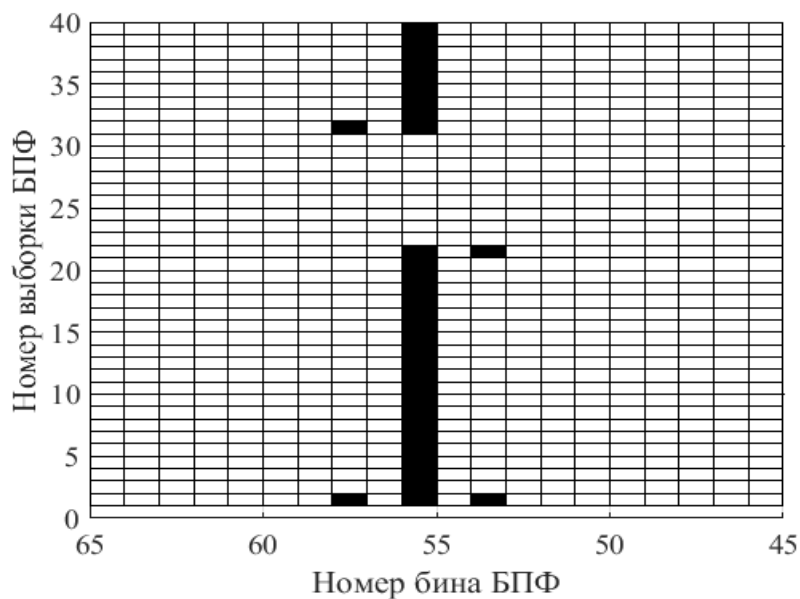


Рисунок 2.20 – Результат работы схемы определения максимума для ИС (черным показан зафиксированный максимум)

Эти ложные максимумы в спектре появляются постоянно на фронтах сигнала, следовательно, можно предсказывать момент их появления и реагировать на них следующим образом. Во-первых, эти ложные срабатывания всегда будут занимать не более одной выборки спектра, во-вторых, это явление не связано с утечкой спектра, поэтому весовое окно не будет влиять на эти максимумы и, при большом уровне сигнала, всплески будут фиксироваться во всей полосе частот. В связи с этими особенностями, я решил ограничить минимальную длительность обнаружения, при которой сигнал поступает в обработку, двумя выборками спектра. Результат работы модифицированного алгоритма показан на рисунке 2.21.

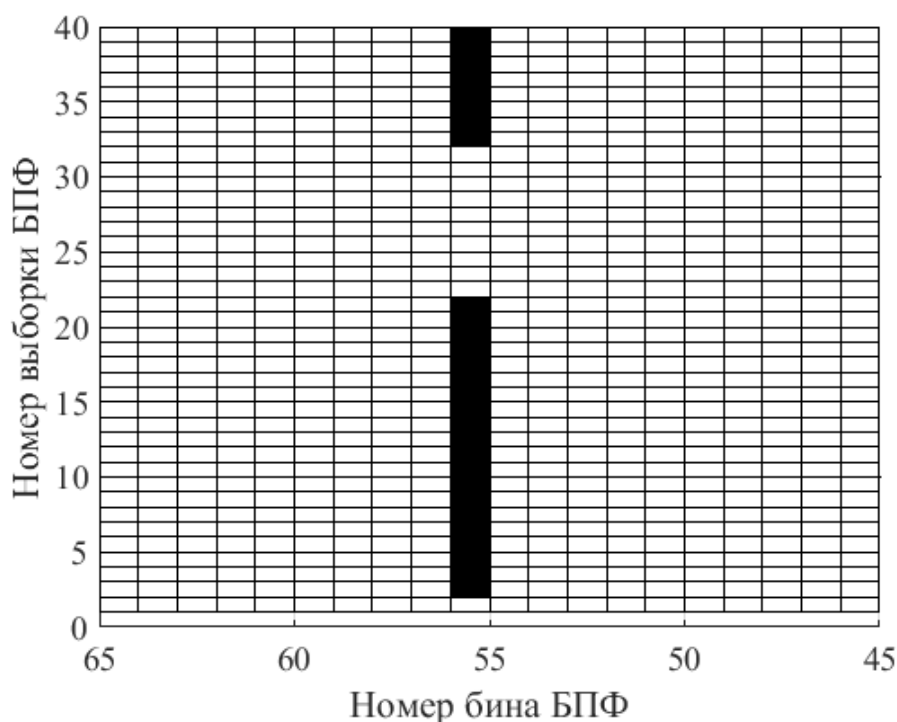


Рисунок 2.21 – Результат работы схемы определения максимума для ИС (черным показан зафиксированный максимум)

Таким образом, поле несущей частоты сигнала соответствует номеру канала обработки, который сформировал дескриптор. Поскольку эффективное число каналов равно 127, то разрядность параметра несущей частоты в дескрипторе составляет семь бит.

2.8 Разработка алгоритмов определения параметров сигнала

Согласно ТЗ кроме несущей частоты требуется определять следующие параметры: длительность импульса, период повторения импульсов в пачке, длительность пачки импульсов, время прихода сигнала, пиковую мощность сигнала, а также разность фаз между двумя входными каналами. Далее будут описаны алгоритмы определения данных параметров.

Каждая выборка БПФ соответствует своему временному интервалу. Для наших параметров частоты дискретизации 1 ГГц и 256 точек БПФ, одна выборка соответствует 256 нс, и она определяет минимальную дискрету по времени, с которой можно измерять параметры сигнала. В устройстве был реализован счетчик времени, с дискретой 256 нс, который формирует метки времени для каждой выборки спектра, с момента начала работы устройства. В ТЗ указаны два параметра времени сигнала – точное и грубое время прихода сигнала. Такое разделение времени позволяет использовать два счетчика времени с меньшей разрядностью, нежели один многоразрядный счетчик. Чем больше разрядность счетчика, тем меньше максимальная частота работы данного счетчика. В параметры времени прихода сигнала записываются значения счетчиков времени, которое было на момент начала обнаружения сигнала в канале обработки. Параметр точного времени имеет разрядность 18 бит.

Поскольку параметр грубого времени прихода сигнала имеет дискрету 1 мс, а одна выборка спектра соответствует 256 нс, то разрешение работы счетчика грубого времени дается на каждом четвертом такте работы счетчика точного времени. Это обеспечивает синхронизацию работы двух счетчиков. Параметр грубого времени имеет разрядность 16 бит.

Параметр пиковой мощности измеряется следующим образом: пока активен сигнал обнаружения запоминается максимальное значение амплитуды в канале обработки. При этом при сбрасывании сигнала обнаружения в случае ИС происходит сброс запомненного значения амплитуды, в случае НС параметр обновляется каждую миллисекунду. Запомненное значение пиковой амплитуды

переводится в децибелы с помощью заранее просчитанной таблицы логарифмов. Таким образом, параметр заносится в дескриптор в децибелах. Размерность параметра пиковой мощности 10 бит.

Параметр разности фаз между двумя каналами входных данных вычисляется, когда активен сигнал обнаружения в канале обработки. Значение фазы с выхода блока CORDIC на обработку поступает в формате градусов. Размерность фазы составляет 10 разрядов. Согласно ТЗ требуется вычислять значение разности фаз в диапазоне минус 180...179, с шагом один градус. Поскольку набор значений не охватывает всю возможную разрядность, необходимо при вычислении разности фазы корректировать результат, чтобы он не превышал требуемый диапазон.

Измерение длительности сигнала реализовано с помощью счетчика, который увеличивается всегда, когда в канале есть сигнал обнаружения. Счетчик длительности находится в сбросе при отсутствии сигнала обнаружения. Параметр длительности импульса имеет разрядность 14 бит. На основе параметра длительности в устройстве происходит классификация типа сигнала – НС или ИС.

В канале обработки НС фиксируется при длительности импульса более 400 мкс. Когда счетчик длительности досчитывает до этого порогового значения, формируется флаг НС, который запрещает дальнейшее увеличение счетчика длительности. При активном флаге НС, в поле длительности импульса записывается нулевое значение, затем формируется дескриптор сигнала с измеренными параметрами. Сброс НС в канале происходит, когда обнаружение в канале пропадает на время больше определенного интервала времени. По результатам моделирования значение этого интервала времени выбрано не меньше 400 мкс. После пропадания сигнала обнаружения в канале, в котором активен флаг НС, запускается счетчик времени пропуска НС.

В канале обработки фиксируется ИС в момент пропадания обнаружения, при значении длительности импульса менее 400 мкс. При этом формируется дескриптор сигнала, в который записываются измеренные параметры импульса. Поскольку на вход устройства может сигнал, состоящий из единичных импульсов

или состоящий из пачек импульсов, то необходимо проработать критерий определения пачки импульсов. В литературе по радиотехнике отличительной особенностью пачек радиоимпульсов, используемых в РЛС, заключается в одинаковом значении длительности импульсов в пачке. С технической точки зрения для передатчика сложно реализовать изменение длительности формируемого СВЧ сигнала, гораздо проще менять период повторения импульсов в пачке. Исходя из этих соображений, проще фиксировать пачку импульсов, отслеживая длительность импульсов в пачке. Схема определения пачки импульсов следующая – в устройстве запоминается длительность трех последних зафиксированных импульсов, при этом фиксируется разница трех длительностей. Если разница оказывается меньше заданного порога, то принимается решение о наличии пачки импульсов в канале обработки, после чего запоминается среднее значение измеренных длительностей. На это событие формируется дескриптор для сигнала, в котором указывается значение среднего значения длительности импульсов в пачке, а также добавляется измеренное значение периода импульсов. После фиксирования пачки в канале, длительности новых импульсов сравниваются с запомненным средним значением в пачке. Если новый импульс отличается от среднего значения, тогда формируется один сигнал ошибки обнаружения пачки. Критерием окончания пачки является наличие двух из трех возможных сигналов ошибки, а также, аналогично критерию пропуска НС, отсутствие обнаружения на определенный интервал времени. Но, в отличие от определения НС, время пропуска пачки радиоимпульсов выбирается равным трем значениям периода импульсов в пачке. При срабатывании события окончания пачки импульса формируется дескриптор, содержащий параметр длительности пачки импульсов. Размерность параметра длительности пачки импульсов 10 бит.

Параметр периода импульсов в пачке измеряется после фиксирования пачки импульсов. Значение периода рассчитывается как разность времени прихода текущего импульса и предыдущего импульсов. Размерность параметра периода импульсов в пачке 16 бит.

3 Разработка модели устройства

3.1 Назначение и условия выполнения модели устройства

Основное назначение имитационной модели заключается в отладке, корректировке и тестированию работы устройства. Модель содержит блок формирования входных радиосигналов, с их последующей оцифровкой, двухканальную систему БПФ и модуль обнаружения и определения параметров радиосигнала.

Модель представлена в виде конфигурируемых блоков Simulink системы Matlab, версии R2015b (8.6.0.267246), работающей под управлением операционной системы Windows 7 SP1/8/8.1/10.

Требования к ПЭВМ:

- таковая частота процессора 1 ГГц и выше (рекомендуется 2...3 ГГц);
- объем ОЗУ не менее 2 Гб;
- место на жестком диске не менее 5 Гб;

3.2 Описание внешнего вида модели

Запуск модели осуществляется путем открытия из среды Simulink файла Model_Obnaг.m. Верхний уровень модели показан на рисунке 3.1.

Обозначения на рисунке 3.1:

А – Блок формирования входных радиосигналов в двух каналах. Он позволяет формировать до четыре сигнала в каналах. Реализовано четыре возможных типа сигналов: непрерывный сигнал, импульсный сигнал с ЛЧМ модуляцией, импульсный сигнал с ФКМ модуляцией и обычная последовательность радиоимпульсов. Параметры сигналов при запуске задаются с помощью файла Signals_Parameters.m.

Б, В – Блок системы БПФ для первого и второго канала соответственно. Перед БПФ к выборкам сигнала применяется окно. Тип окна выбирается из списка в маске блока. Внешний вид маски представлен на рисунке 3.2;

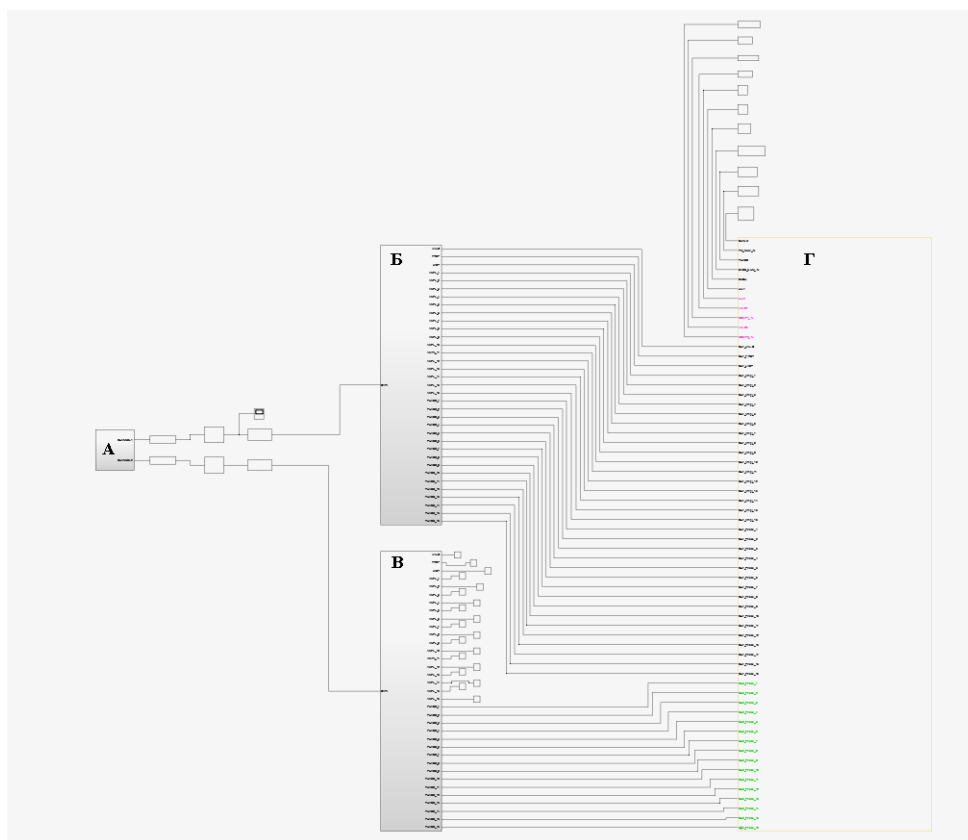


Рисунок 3.1 – Верхний уровень модели

Г – Блок обнаружителя, над которым расположены блоки разовых команд. Он имеет набор параметров, которые можно менять в маске: «Максимальное значение счетчика точного времени», «Максимальное значение счетчика грубого времени», «Максимальное значение счетчика 1 мс», «Значение порога фиксации пачки импульсов», «Максимальная длительность импульсов», «Интервал времени принятия решения о конце пачки», «Интервал времени принятия решения об окончании НС». Внешний вид маски блока обнаружителя представлен на рисунке 3.3. По умолчанию в маску передаются переменные, задаваемые с помощью конфигурационного скрипт-файла `Model_config.m`. Чтобы внести изменения в любой параметр, достаточно заменить имя переменной в поле на требуемое значение. Или можно изменить значение переменной в конфигурационном файле, затем запустить его на выполнение.

В файле `Model_config.m`, помимо основных параметров блока обнаружителя, также присутствуют переменные разовых команд. Для изменения выставленных значений разовых команд необходимо скорректировать нужную переменную в

конфигурационном файле и запустить его. После чего необходимо перезапустить симуляцию модели.

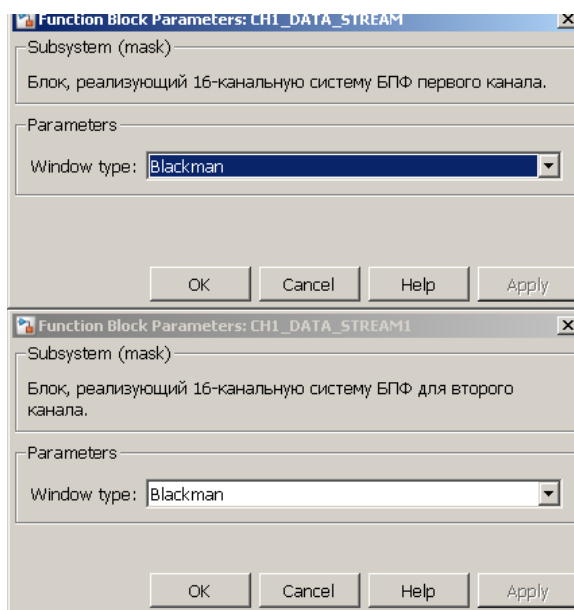


Рисунок 3.2 – Внешний вид маски блоков системы БПФ для двух каналов

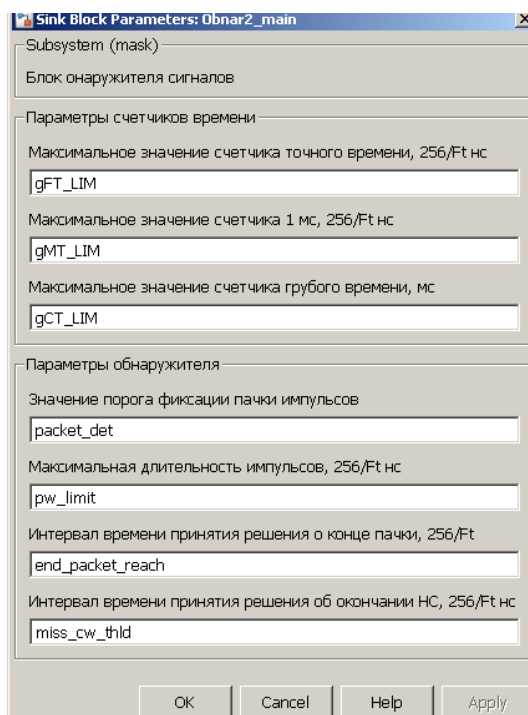


Рисунок 3.3 – Внешний вид маски блока обнаружителя

Для облегчения процесса компиляции модели, в ней реализовано пять каналов обнаружителя, с седьмого канала по 11-й. Номер канала соответствует бину БПФ. Номера каналов можно сменить, для этого нужно зайти под маску блока обнаружителя. Чтобы это сделать следует выделить блок, щелкнуть правой

кнопкой мыши по блоку и в контекстном меню выбрать Mask/Look under mask, или нажать комбинацию клавиш Ctrl+U. Также под маску можно зайти, нажав на стрелочку в левом нижнем углу блока. После этого откроется внутренний уровень блока обнаружителя, в котором явно обозначены пять блоков каналов. У блока канала есть вход «CHANNEL_NUM», на него подается сигнал с блока константы, в которой записан номер канала. Изменение значения этого блока константы приведет к смене канала, которые будут симулироваться. Внешний вид внутреннего уровня блока обнаружителя с указанным расположением констант с номерами каналов показан на рисунке 3.4.

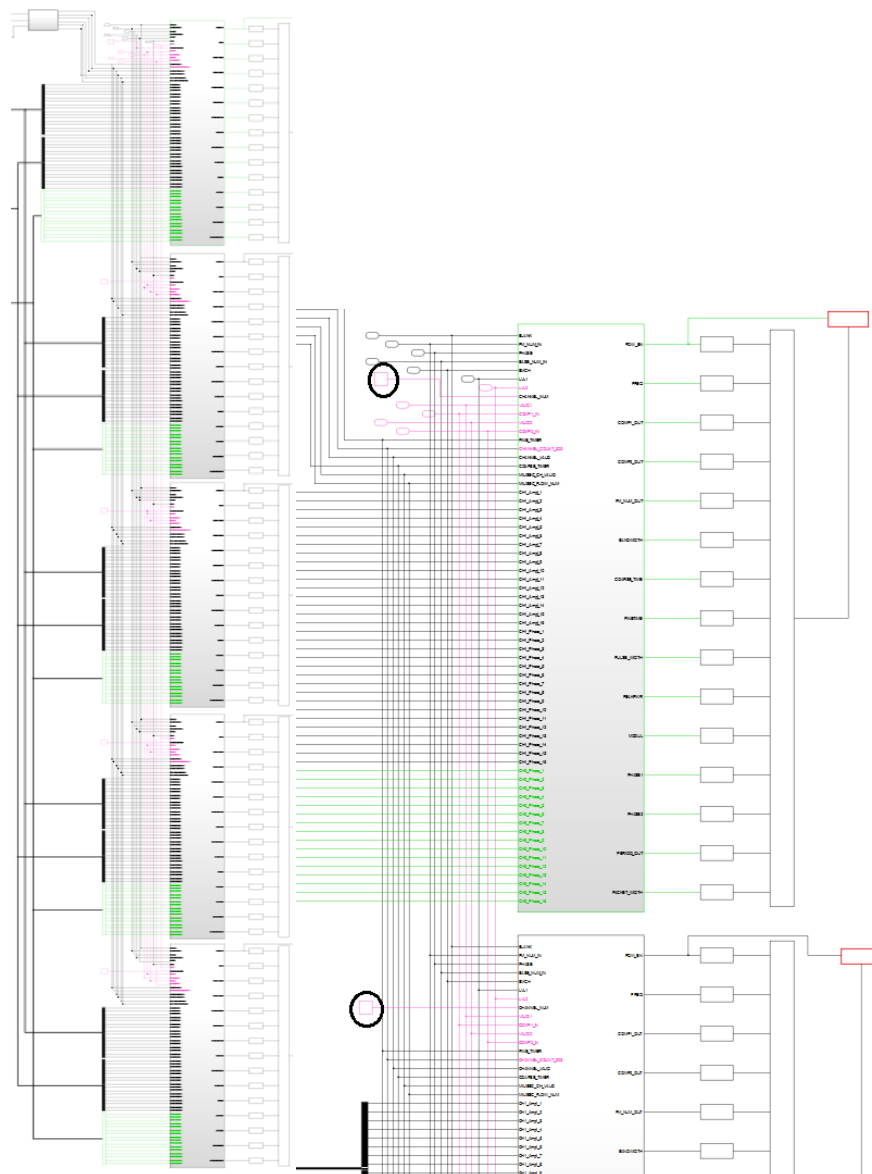


Рисунок 3.4 – Внутренний уровень блока обнаружения (слева), расположение блоков с номером канала (справа)

3.3 Работа с моделью

Для начала работы с моделью необходимо запустить среду Simulink, затем открыть файл модели Model_Obnar.slx. При запуске модели автоматически произойдет выполнение конфигурационного скрипт-файла Model_config.m, который очистит командную строку Matlab, освободит рабочее пространство от созданных ранее переменных, после чего создаст переменные, заданные в скрипте.

Параметры формируемых сигналов записаны в скрипт-файле Signals_Parameters.m. Он формирует пять переменных-векторов:

x – содержит параметры, определяющие количество формируемых сигналов, их типы, разность фаз между каналами, а также параметры шума добавляемого к сигналу;

$s1$ – содержит параметры 1-го сигнала всех возможных типов;

$s2$ – содержит параметры 2-го сигнала всех возможных типов;

$s3$ – содержит параметры 3-го сигнала всех возможных типов;

$s4$ – содержит параметры 4-го сигнала всех возможных типов.

Всего в модели имеется возможность подавать в канал до 4 сигналов одновременно. Каждый из них имеет один из 4 возможных типов: непрерывный гармонический сигнал, радиоимпульсы с ЛЧМ, радиоимпульсы с ФКМ и обычные радиоимпульсы. Выбор типа и количества сигналов в канале определяется значением переменной $x[1...8]$.

При открытии модели происходит автоматическое выполнение скрипт-файла с параметрами, выполняющего инициализацию всех параметров в блоке генератора сигналов. После этого для внесения изменений в формируемые сигналы достаточно внести их в скрипт-файл и перезапустить его. В результате обновятся переменные-вектора и следующая симуляция модели будет формировать новый сигнал.

После инициализации модели обнаружителя и генератора сигналов можно запускать симуляцию, для этого нужно нажать круглую зеленую кнопку

проигрывания (Run) на верхней панели окна модели. Начнется компиляция модели, после чего запустится процесс симуляции. Текущий статус работы модели, а также процент выполнения симуляции показывается на панели состояний внизу окна.

Следует отметить, что симуляция занимает значительное количество времени, поэтому не рекомендуется запускать симуляцию более чем на 1 мс ($1e-3$ с). При таком интервале симуляции может длиться больше 1 часа, к тому же в процессе Matlab может значительно увеличить объем занимаемой оперативной памяти (вплоть до 3 ГГб), что негативно сказывается на работе других приложений. Для уменьшения времени симуляции, например, непрерывных сигналов с длительностью больше 600 мкс, можно уменьшить параметр «Максимальная длительность импульса». В результате обнаружитель воспримет сигнал с длительностью меньше 600 мкс как непрерывный сигнал. Тот же механизм работает с другими параметрами блока обнаружителя.

В процессе симуляции сформированные дескрипторы в каждом канале записываются в переменные с именем «PDW_ch\$», где \$ - номер канала.

После окончания симуляции будет выполнен скрипт-файл form_table.m, который обрабатывает сформированные каждым каналом дескрипторы. Обработка заключается в формировании очереди дескрипторов, отсортированной по времени их появления. Отсортированная очередь дескрипторов выводится в окне в виде таблицы. Внешний вид окна представлен на рисунке 3.5. Если по окончании симуляции не было сформировано ни одного дескриптора, то в командной строке выводится сообщение «Нет сформированных дескрипторов».

	Частота, Гц	Грубое время, мс	Точное время, нс	Длительность импульса	Мощность	Тип модуляции	Разность фаз, база 1	Период импульсов	Длительность пачки
1	7	0	2	9	214	0	-256	0	0
2	7	0	2	9	214	0	-256	60	0
3	7	0	2	9	214	0	-256	60	0
4	8	0	2	14	214	0	-256	0	0
5	8	0	2	14	214	0	-256	60	0
6	8	0	2	14	214	0	-256	60	0
7	9	0	3	18	214	0	-256	0	0
8	9	0	3	18	214	0	-261	60	0
9	9	0	3	18	214	0	-256	60	0
10	10	0	8	18	214	0	-256	0	0
11	10	0	8	18	214	0	-256	60	0
12	10	0	8	18	214	0	-256	60	0
13	11	0	13	18	213	0	-256	0	0
14	11	0	13	18	213	0	-256	60	0
15	11	0	13	18	213	0	-254	60	0

Рисунок 3.5 – Внешний вид таблицы дескрипторов, сформированных по результату симуляции модели

3.4 Сообщения оператору

При работе с моделью могут возникнуть сообщения, связанные с ошибочными действиями пользователя, неправильной инициализацией модели, предупреждения системы Simulink. Все сообщения выводятся в отдельном окне «Diagnostic Viewer» (рисунок 3.6). Общее число сообщений, включающих предупреждения и ошибки выводится в панели состояния в нижней части окна модели. Типовые сообщения об ошибках приведены в таблице 3.1.

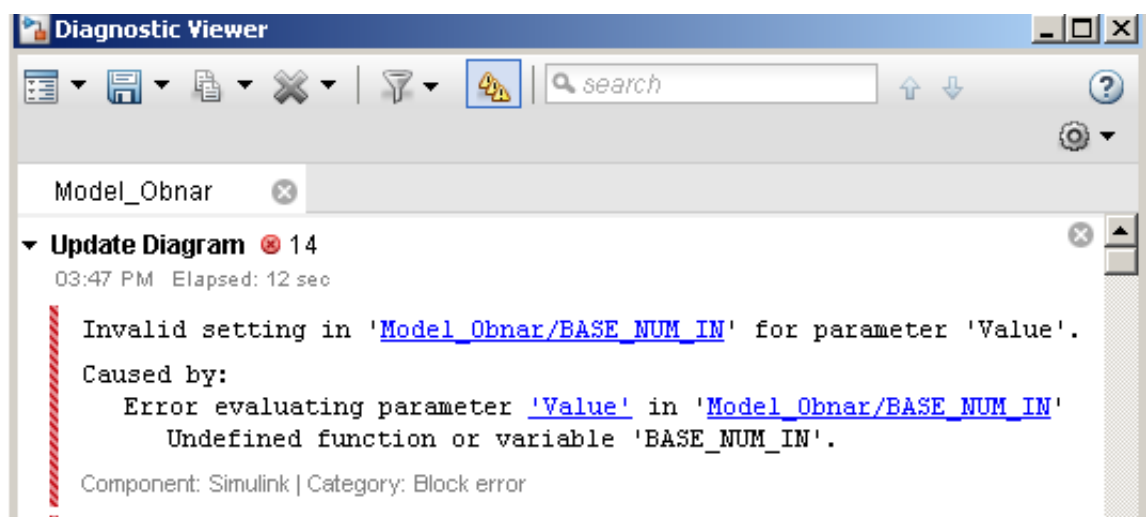


Рисунок 6 – Сообщения об отсутствии внутренних переменных

Таблица 3.1 – Ошибки, возникающие при работе с моделью

Сообщение	Содержание сообщения	Действия оператора
Invalid setting in 'Model_Obna/XXXXX' for parameter 'Value'. Caused by Undefined function or variable 'YYYYYY'	В один или несколько параметризуемых блоков введены неправильные, или не введены значения параметров.	Чаще всего, такая ошибка возникает, если запустить модель до запуска файла конфигурации. Или если в процессе работы была произведена очистка рабочего пространства Matlab. Необходимо перезапустить файл конфигурации.
Pulse width must be less than or equal to the period of the Pulse Generator in 'Model_Obna/Модуль генератора сигналов/Модель XXXXX канала/Блок сигналов X/XXXXXXXX'	Заданное в блоке формирования прямоугольной огибающей значение длительности импульса больше, чем период импульса.	Проверить правильность выставления параметров длительности и периода для всех возможных значений. После исправления перезапустить файл с параметрами сигналов.
Error in 'Model_Obna/Модуль генератора сигналов/Модель XXXX канала/Блок сигналов X/Генератор ЛЧМ сигнала': The 'Target time' parameter must be less or equal to the 'Sweep time' parameter.	Заданный параметр «Время достижения конечной частоты» для ЛЧМ сигнала имеет значение больше чем параметр «Время достижения начальной частоты».	Проверить правильность задания параметров ЛЧМ сигнала, после чего перезапустить файл с параметрами сигналов.
Error in '...': The 'Initial phase' parameter must be greater than or equal to 0.	Заданное значение разности фаз в каналах, или начальная фаза сигнала не должны быть отрицательными.	Значение разности фаз в каналах или начальной фазы сигнала, выраженное в радианах должно быть положительным числом. Для задания противоположного значения фазы, нужно прибавить к требуемому значению π , после чего перезапустить файл с параметрами сигнала

4 Результаты испытаний устройства

В предыдущих разделах была описана разработка алгоритмов устройства, а также представлено описание имитационной модели устройства. Следующим шагом стало разработка программного обеспечения на ПЛИС для макета устройства. Согласно ТЗ языком написания ПО для ПЛИС должен быть VHDL, а среда программирования стала Vivado Design Suite, фирмы Xilinx.

В процессе написания ПО для устройства было принято решение придерживаться структуры имитационной модели. Такое разделение на блоки обработки позволило упростить процесс отладки ПО, в процессе написания. В устройстве было предусмотрено несколько частотных доменов, для работы разных логических блоков. Первый домен имеет частоту 300 МГц, на таком такте работают входные блоки БПФ и CORDIC, а также блоки обнаружения и измерения частоты. Такой высокий такт формируется из высокоскоростных приемников данных от АЦП, поэтому для исключения возможной потери данных был выбран такой такт. Следующий домен имеет частоту 125 МГц, эта частота работы блоков, связанных с интерфейсом Ethernet, также на данной частоте работают все каналы обработки сигналов. Такое понижение частоты работы обусловлено наличием запаса по частоте в данных, поступающих от БПФ. Каждый канал обработки принимает данные только от своего канала БПФ, при этом скорость поступления данных в один канал в 256 раз медленнее, чем общая скорость работы БПФ.

Каждый канал может формировать дескрипторы, которые складываются в буфер на выходе канала. Все выходы канала контролирует арбитр дескрипторов, который по кольцу опрашивает все каналы обработки на наличие готовых дескрипторов. Данный арбитр перемещает дескрипторы из очереди каналов в выходную очередь дескрипторов. Длина этой очереди составляет 128 дескрипторов.

Выходная очередь дескрипторов непосредственно взаимодействует с контроллером Ethernet, по приходу запроса от ЦП или ПЭВМ, из очереди

извлекается действительный дескриптор и посылается в контроллер Ethernet. Если дескрипторы отсутствуют, тогда посылается дескриптор с нулевым содержимым, что является знаком пустой очереди. Каждый дескриптор в очереди имеет свое «время жизни», по истечении которого он удаляется из очереди. Это время задается программно через интерфейс Ethernet. В приложении А приведен исходный код ПО для устройства на языке VHDL. Требования ТЗ по занимаемым ресурсам были выполнены, отчет среды программирования, по размещенному проекту устройства представлен в таблице 4.1.

Таблица 4.1 – Отчет о занимаемых ресурсах ПЛИС

Название параметра	Значение
Логических ячеек	39 %
Количество регистров	35 %
Блоков памяти	15 %
Число ячеек ЦОС	20 %
Количество входов	80 %

После разработки ПО были проведены испытания устройства в лабораторных условиях. Для испытаний устройство подключалось к ПЭВМ, которое исполняло роль ЦП. На вход устройства с генераторов подавались различные сигналы, а на ПЭВМ наблюдался поток дескрипторов. Программа для ПЭВМ проводила первоначальную настройку устройства, после чего производила чтение дескрипторов и отображение их в удобном для оператора виде. Написание программы для ПЭВМ не входило в задачи данной работы. Внешний вид программы представлен на рисунке 4.1.

По результатам испытаний было установлено, что устройство определяет параметры сигнала согласно техническому заданию. Определение параметров сигналов происходит во всей полосе работы устройства, которая составляет 500 МГц. Для случая двух сигналов на входе устройства, различение этих сигналов

происходит при минимальном разнесении частот сигналов на расстояние одного канала, то есть 3,9 МГц.

	1	2	3	4	5	6	7	8	9	10
Несущая частота	9	5	9	5	9	5	9	5	9	5
Грубое время прихода сигнала	ffa6	ffae	ffa7	ffae	ffa8	ffae	ffa9	ffae	ffaa	ffae
Точное время прихода сигнала	1165b	783e	125b5	783e	1350f	783e	14469	783e	153c3	783e
Длительность импульса	0	3e	0	3e	0	3e	0	3e	0	3e
Пиковая амплитуда	19b	195	1a0	192	1a2	193	1a2	193	19f	194
Разница фаз (град.)	200	200	200	200	200	200	200	200	200	200
Период повторения импульсов в пачке	0	7a	0	7a	0	7a	0	7a	0	7a
Длительность пачки импульсов	0	0	0	0	0	0	0	0	0	0

Рисунок 4.1 – Внешний вид интерфейса программы для ПЭВМ

По результатам испытаний были сформированы пути дальнейшего развития работы:

- необходимо исследовать влияние перекрытия во времени пакетов сигналов перед БПФ, согласно литературным источникам [2] это позволяет улучшить разрешение во времени;
- поскольку реализованная схема обнаружителя с одним порогом оказалась недостаточно эффективной, то нужно исследовать влияние на качество работы устройства, схемы с адаптивным порогом;
- необходимо провести более детальное исследование работы устройства в многосигнальной обстановке;
- необходимо разработать алгоритм обнаружения широкополосных сигналов, таких как ЛЧМ, ФКМ и БПЧ;
- также возможна дальнейшая оптимизация программы для ПЛИС.

5 Безопасность жизнедеятельности

В дипломной работе разрабатывается аппаратно-программное устройство определения параметров радиосигнала, которое представляет собой блок цифрового приемника сверхвысокочастотного излучения. Поэтому в данном разделе необходимо рассмотреть вопросы, связанные с наличием опасных и вредных производственных факторов, условиями эксплуатации устройства, воздействием электромагнитного излучения на организм человека.

Устройство предполагается устанавливать на борту летательного аппарата. Оно является автономной системой, в штатном режиме работы не требуется работа оператора. Разработанный макет предполагается настраивать в лабораторных условиях с применением поверенных приборов.

Устройство, как электротехническое устройство, соответствует требованиям ГОСТ 12.2.007.0-75 ССБТ. «Изделия электротехнические. Общие требования безопасности».

Основной частью макета устройства является отладочная плата Virtex-7 VC707 Evaluation Kit. В ней используется безопасное сверхнизкое напряжение плюс 5 В. Блок питания отладочной платы располагается отдельно и преобразует напряжение сети 220 В в безопасное напряжение плюс 5 В.

По способу защиты человека от поражения электрическим током устройство относится к III классу. Оно не имеет ни внешних, ни внутренних электрических цепей, работающих при напряжении отличном от плюс 5 В, а также присоединяется непосредственно к блоку питания, преобразующему напряжение сети 220 В в безопасное сверхнизкое напряжение плюс 5 В), что осуществляется посредством преобразователя с отдельными обмотками.

Электрическая схема отладочной платы исключает возможность его самопроизвольного включения и отключения. Конструкция отладочной платы исключает возможность неправильного присоединения его сочленяемых токоведущих частей. Ввод проводов в корпус устройства (питания, сигнальные кабели) осуществляется через изоляционные детали. Конструкция и материал

вводных устройств исключает возможность случайного прикосновения к токоведущим частям. На отладочной плате предусмотрена предупредительная сигнализация, указывающая на включенное состояние устройства. Отладочная плата не оснащена отдельным корпусом, поэтому требуется соблюдать меры безопасности при работе с платой. На плате предусмотрена цветовая маркировка, а также предупреждающие надписи для частей, которые могут находиться под напряжением. Все элементы управления на плате разнесены с элементами питания, что исключает случайное замыкание или прикосновение к цепям питания.

Устройство не является источником шума, вибраций, теплового, оптического или рентгеновского излучения.

Анализ воздействия электромагнитного поля на организм человека

Вследствие того, что разрабатываемое устройство является приемником сверхвысокочастотного излучения в диапазоне сигналов до 500 МГц, рассмотрим влияние сверхвысокочастотного излучения на организм человека.

По воздействию на организм человека сверхвысокочастотное излучение условно делится на два вида.

Тепловое воздействие – за счет нагрева тканей организма человека, проявляется при наибольших уровнях излучения. Наиболее подвержены тепловому воздействию глаза (хрусталик) и яички у мужчин. Это связано с тем, что в этих органах мало кровеносных сосудов, поэтому из-за крайне низкого теплоотвода глаза и яички поражаются в первую очередь.

Нетепловое (информационное) воздействие – проявляется на небольших уровнях излучения, как результат взаимодействия высокочастотного излучения с биополем человека. Проявляется косвенно, как дополнительный стресс организма, в комплексе с другими негативными воздействиями (экология, продукты питания, психическое напряжение жителей мегаполисов). Воздействие неионизирующего излучения имеет тенденцию накапливаться в организме.

Чувствительность организма к сверхвысокочастотному излучению начинается

при уровнях много меньше теплового воздействия. Начиная порядка долей микроватт на сантиметр квадратный; до единиц милливатт продолжается фаза угнетения организма, далее наступает фаза стимуляции - улучшение под влиянием сверхвысокочастотного излучения общего состояния организма или чувствительности его отдельных органов, а на плотности более 10 мВт/см^2 снова наступает фаза угнетения организма.

Поглощение электромагнитной энергии биологической тканью при постоянной частоте определяется средними значениями диэлектрической проницаемости, электропроводности и магнитной проницаемости тканей. В итоге, все биологические ткани организма человека можно представить в виде двух групп: к первой относятся подвижные среды (кровь, спинномозговая жидкость, воздух в легких и др., количество которых непостоянно для данной области биологического объекта); ко второй – все остальные среды (мышцы, костная и жировая ткань и т. д.). Диэлектрическая проницаемость, электропроводность и магнитная проницаемость для всех тканей различны, что определяет глубину проникновения электромагнитной энергии в биологическую ткань и биологический эффект облучения ЭМИ. Если механизм терморегуляции организма способен путем рассеивания избыточного тепла предупреждать перегрев, то его температура остается нормальной, если нет - происходит непрерывный подъем температуры, который приводит к воспалению или даже разрушению ткани.

На границе раздела тканей первой и второй групп в результате отражения происходит образование стоячих волн, обуславливающих образование так называемых «горячих пятен».

Наличие отражения на границе воздух - ткань приводит к уменьшению теплового эффекта на всех частотах. Перераспределение тепловой энергии между соседними тканями через кровь наряду с конвекционной отдачей энергии теплоиспусканием в окружающее пространство во многом определяет температуру нагреваемых участков тела. Именно из-за ухудшения системы отвода

тепла от некоторых участков (например, глаза, ткань семенников, в которых мало кровеносных сосудов) эти органы наиболее уязвимы для облучения.

Анализ материалов клинических и экспериментальных исследований свидетельствует, что в результате воздействия ЭМИ развивается характерный синдром раннего старения организма: ухудшение памяти и снижение работоспособности, полипатология, снижение иммунологической реактивности и компенсаторно-приспособительных возможностей организма, раннее нарушение обмена липидов и прогрессирующее накопление дисульфидных групп, нарушение репродуктивной функции, развитие возрастной патологии в ранние годы. Можно констатировать общность клинической симптоматики последствий электромагнитного облучения и старения организма.

Разрабатываемый макет устройства предполагается использовать в лабораторных условиях. В этом случае источником излучения могут быть только приборы и внешние генераторы, которые применяются на стадии настройки прибора и в процессе проведения испытаний макета.

Согласно СанПиН 2.2.4/1191-03 «Электромагнитные поля в производственных условиях» оценка воздействия ЭМИ РЧ на людей осуществляется по значению плотности потока энергии (ППЭ). В диапазоне частот 300 МГц ... 300 ГГц она рассчитывается по формуле:

$$ЭЭ_{\text{ППЭ}} = \text{ППЭ} \cdot T, \quad (5.1)$$

где $ЭЭ_{\text{ППЭ}}$ – энергетическая экспозиция плотности потока энергии, (В/м²) ч;

ППЭ – плотность потока энергии, Вт/м²;

T – время экспозиции, ч.

Имеющиеся на предприятии лабораторные генераторы тестовых сигналов 9 кГц ... 3 ГГц не создают на рабочем месте персонала электромагнитных полей, превышающих предельно допустимые уровни, которые указаны в таблице 1 согласно ГОСТ 12.1.006-84 (СТ СЭВ 5801-86) ССБТ «Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению

контроля». Используемые генераторы не имеют возможности работать на открытый волновод, излучение передается с помощью экранированных кабелей, что исключает возможность превышения предельно-допустимых уровней излучения.

Таблица 5.1 – Уровни излучения

Параметр	Диапазон частот
	300 МГц ...300 ГГц
Предельно допустимое значение ЭЭ-пээ, (мкВт/см ²) · ч	200
Максимальный предельно допустимый уровень ПЭЭ, мкВт/см ²	1000

Каждый лабораторный прибор на предприятии прошел метрологическую аттестацию, в том числе на создаваемый ими уровень излучения, что гарантирует уровень ниже, указанного в таблице 5.1.

Контроль электромагнитного излучения на рабочих местах необходимо проводить также в соответствии с СанПиН 2.2.4.1191-03.

6 Организационно-экономический раздел

В данном разделе приводится экономический анализ и сетевое планирование разрабатываемого устройства. Оно заключается в оценке экономического эффекта от разрабатываемой продукции. В процессе выполнения данного раздела выделены и рассчитаны основные этапы экономического планирования, по итогам которого построен сетевой график выполнения работ. Также проведен расчет сметы разработки и себестоимости конечного изделия.

Разрабатываемое устройство представляет собой цифровой приемник, который может являться составной частью комплекса радиотехнической разведки РТР.

6.1 Сетевое планирование.

Метод сетевого планирования основан на моделировании процессов с помощью сетевого графика и представляет собой совокупность расчетных методов, организационных и контрольных мероприятий. Метод применяется, когда необходимо выполнить значительный объем работ в ограниченные сроки с привлечением значительного числа работников в условиях ограниченности ресурсов. В сетевом планировании используется метод графического изображения определенного комплекса работ, отражающего их логическую последовательность, взаимосвязь и длительность.

Элементы сетевого графика

Событие – основной элемент сетевого графика, это момент времени, означающий окончание какой-либо работы и начало следующей.

Событие может быть трех видов:

исходное событие – событие, означающие начало работы, является единственным событием;

завершающее событие – событие, означающие окончание всех работ и достижение конечной цели работ, также единственное событие в сетевом графике;

промежуточное событие – результат одной или нескольких работ, позволяющих начать одну или несколько следующих за данной работой.

Событие не имеет протяженности во времени и случается мгновенно, как только все предшествующие ему работы будут выполнены. Пока не свершится данное событие, ни одна из работ, следующих за ним, не может начаться.

Работа – процесс, требующий определенного количества работников и времени. Любая работа связывает два события.

Путь – определенная последовательность ряда выполненных работ. Продолжительность любого пути равна сумме работ, входящих в данный путь. Полный путь имеет максимальную продолжительность и его называют критическим. Продолжительность критического пути определяет общую продолжительность выполнения всех работ.

Ожидаемая продолжительность каждой работы определяется по формуле:

$$t_{ij\text{ож}} = 0.6 \cdot t_{ij\text{мин}} + 0.4 \cdot t_{ij\text{макс}}, \quad (6.1)$$

где $t_{ij\text{ож}}$ - ожидаемая продолжительность выполнения работ;

$t_{ij\text{мин}}$ – минимальная оценка продолжительности выполнения работ;

$t_{ij\text{макс}}$ - максимальная оценка продолжительности выполнения работ.

Среднеквадратичное отклонение продолжительности выполнения работ в двухоценочной методике рассчитывается по формуле:

$$\delta_{ij} = 0.2 \cdot (t_{ij\text{макс}} - t_{ij\text{мин}}). \quad (6.2)$$

Дисперсия определяется по формуле:

$$D_{ij} = 0.04 \cdot (t_{ij\text{макс}} - t_{ij\text{мин}})^2. \quad (6.3)$$

Результаты вычисления по формулам (6.1, 6.2, 6.3) приведены в таблице 6.1.

В качестве исполнителей работ привлечем одного руководителя и двух инженеров.

Таблица 6.1 – Перечень, параметры и вероятностные характеристики работ СГ

Код новой работы	Наименование работы	Продолжительность, дн			Исполнители, чел.			СКО, дн	Дисперсия, дн ²
		мин.	макс.	ожд.	рук.	инж.	инж.		
(0,1)	Получение и анализ ТЗ	1	1	1	1	1	1	0	0
(9,10)	Подбор литературы по охране труда	1	3	2	0	1	0	0,4	0,16
(1,2)	Подбор технической документации	2	4	3	0	1	1	0,4	0,16
(2,3)	Анализ технической документации	1	3	2	1	1	1	0,4	0,16
(10,12)	Выбор мероприятий по БЖД	1	3	2	0	0	1	0,4	0,16
(3,4)	Разработка модели устройства	8	11	10	1	1	0	0,6	0,36
(4,5)	Испытание модели устройства	5	7	6	1	1	1	0,4	0,16
(5,6)	Корректировка модели по результатам испытаний	4	6	5	0	1	1	0,4	0,16
(4,6)	Разработка РО для модели	2	4	3	0	0	1	0,4	0,16
(6,7)	Написание программного обеспечения для ПЛИС	15	23	21	1	0	1	1,6	2,56
(7,8)	Испытание ПО на отладочной плате	7	11	10	1	1	1	0,8	0,64
(8,9)	Корректировка ПО по результатам испытаний	6	9	8	1	0	1	0,6	0,36
(14,16)	Оформление ПЗ	4	8	7	0	1	1	0,8	0,64
(14,15)	Оформление графического материала	3	5	4	0	1	1	0,4	0,16

Продолжение таблицы 6.1

Код новой работы	Наименование работы	Продолжительность, дн			Исполнители, чел.			СКО, дн	Дисперсия, дн ²
		мин.	макс.	ожд.	рук.	инж.	инж.		
(9,11)	Расчет экономических показателей	1	3	2	0	1	0	0,4	0,16
(11,13)	Расчет сетевого графика	1	3	2	0	0	1	0,4	0,16
(13,14)	Утверждение экономического раздела	1	1	1	0	1	0	0	0
(12,14)	Утверждение раздела по охране труда	1	1	1	0	0	1	0	0
(15,16)	Согласование графических материалов	1	1	1	1	1	1	0	0
(16,17)	Сдача проекта	1	1	1	0	1	1	0	0

6.2 Расчет параметров работ сетевого графика

1) Ранний срок начала работы $T_{pni j}$ совпадает с ранним сроком свершения ее начального события T_{pi} :

$$T_{pni j} = T_{pi}. \quad (6.4)$$

2) Поздний срок начала работы $T_{nni j}$ можно получить, если из позднего срока свершения ее конечного события T_{nj} вычесть ее ожидаемую продолжительность t_{ij} :

$$T_{nni j} = T_{nj} - t_{ij}. \quad (6.5)$$

3) Ранний срок окончания работы $T_{poi j}$ образуется прибавлением ее продолжительности t_{ij} к раннему сроку свершения ее начального события T_{pi} :

$$T_{poi j} = T_{pi} + t_{ij}. \quad (6.6)$$

4) Поздний срок окончания работы T_{noij} совпадает с поздним сроком свершения ее конечного события T_{nj} :

$$T_{noij} = T_{nj}. \quad (6.7)$$

Для всех работ критического пути, как не имеющих резервов времени, ранний срок начала совпадает с поздним сроком начала, а ранний срок окончания с поздним сроком окончания.

Работы, не лежащие на критическом пути, обладают резервами времени.

5) Полный резерв времени работы R_{nij} образуется вычитанием из позднего срока свершения ее конечного события раннего срока свершения ее начального события и ее ожидаемой продолжительности:

$$R_{nij} = T_{nj} - T_{pi} - t_{ij}. \quad (6.8)$$

6) Частный резерв времени первого рода $R_{¹ij}$ равен разности поздних сроков свершения ее конечного и начального событий за вычетом ее ожидаемой продолжительности:

$$R_{¹ij} = T_{nj} - T_{ni} - t_{ij}. \quad (6.9)$$

7) Частный резерв времени второго рода $R_{²ij}$ равен разности ранних сроков свершения ее конечного T_{pj} и начального событий T_{pi} за вычетом её ожидаемой продолжительности:

$$R_{²ij} = T_{pj} - T_{pi} - t_{ij}. \quad (6.10)$$

8) Свободный (независимый) резерв времени работы R_{cij} образуется вычитанием из раннего срока свершения ее конечного события T_{pj} позднего срока свершения ее начального события T_{ni} и ее ожидаемой продолжительности:

$$R_{cij} = T_{pj} - T_{ni} - t_{ij}. \quad (6.11)$$

Свободный резерв времени может быть отрицательным.

Для работ, лежащих на критическом пути, никаких резервов времени нет и, следовательно, коэффициент напряженности k_{nij} таких работ равен единице.

Критический путь – максимальный по продолжительности полный путь.

Критическая работа – любая работа на критическом пути. Особенность критических работ состоит в том, чтобы каждая из них начиналась точно в момент времени, когда закончилась предыдущая и, кроме того, продолжаться она должна не более того времени, которое ей отведено по плану. В противном случае критический путь увеличится. Следовательно, критический путь должен быть всегда под контролем руководителей работ, ибо от выполнения критических работ целиком зависит выполнение плана проектирования.

В зависимости от коэффициента напряженности все работы попадают в одну из трех зон напряженности:

- критическую, $k_{nij} > 0.8$;
- промежуточную, $0.5 \leq k_{nij} \leq 0.8$;
- резервную, $k_{nij} < 0.5$.

Результаты расчётов сводятся в таблицу 6.2

Таблица 6.2 - Параметры работ сетевого графика

В днях

Код работы	Ожидаемая продолжительность	Срок начала		Срок окончания		Резерв времени			
		ранний	поздний	ранний	поздний	полный	частный 1 рода	частный 2 рода	свободный
1	2	3	4	5	6	7	8	9	10
-0,1	1	0	0	1	1	0	0	0	0
-1,2	3	1	1	4	4	0	0	0	0
-2,3	2	4	4	6	6	0	0	0	0
-3,4	10	6	6	16	16	0	0	0	0
-4,5	6	16	16	22	22	0	0	0	0
-4,6	3	16	24	19	27	8	8	8	8
-5,6	5	22	22	27	27	0	0	0	0
-6,7	21	27	27	48	48	0	0	0	0
-7,8	10	48	48	58	58	0	0	0	0
-8,9	8	58	58	66	66	0	0	0	0
-9,1	2	66	66	68	68	0	0	0	0

Продолжение таблицы 6.2

Код работы	Ожидаемая продолжительность	Срок начала		Срок окончания		Резерв времени			
		ранний	поздний	ранний	поздний	полный	частный 1 рода	частный 2 рода	свободный
1	2	3	4	5	6	7	8	9	10
-9,11	2	66	66	68	68	0	0	0	0
-10,12	2	68	68	70	70	0	0	0	0
-11,13	2	68	68	70	70	0	0	0	0
-12,14	1	70	70	71	71	0	0	0	0
-13,14	1	70	70	71	71	0	0	0	0
-14,15	4	71	73	75	77	2	2	0	0
-14,16	7	71	71	78	78	0	0	0	0
-15,16	1	75	77	76	78	2	0	2	0
-16,17	1	78	78	79	79	0	0	0	0

Количество событий $n_c = 18$. Количество работ $n_p = 20$. Критический путь составляет $L_{кр} = 79$ дней.

Коэффициент сложности СГ рассчитывается как отношение количество работ к количеству событий:

$$k_{сл} = \frac{n_p}{n_c} = \frac{20}{18} = 1.11. \quad (6.12)$$

Сетевой график представлен на рисунке 6.1.

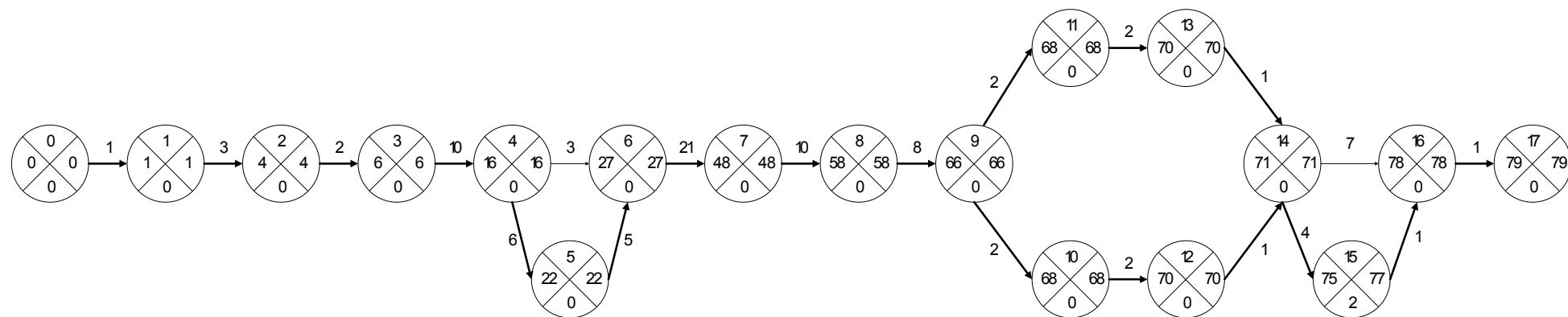


Рисунок 6.1 – Сетевой график

6.3 Расчет затрат на проведение НИОКР

Дипломная работа относится к НИОКР. В соответствии с приказом от 11 сентября 2014г. №1788 «Об утверждении методики определения начальной (максимальной) цены государственных контрактов на выполнение научно-исследовательских, опытно-конструкторских и технологических работ» сметная стоимость НИОКР складывается из следующих статей затрат:

- затраты на оплату труда работников;
- отчисления на социальные нужды;
- материалы;
- прочие прямые расходы;
- накладные расходы.

Рассмотрим каждую статью затрат подробнее.

Таблица 6.3 – Затраты на материалы

В рублях

Наименование материала	Единица измерения	Количество, шт	Цена за единицу	Сумма
1 Бумага формата А4	упаковка	1	170	170
2 Ватман формата А1	лист	6	50	300
Итого				470
С учетом транспортно-заготовительных расходов				500

Рассчитаем расходы на основную заработную плату.

Введём должностные оклады персонала и вычислим коэффициенты перерасчёта. Результаты занесем в таблицу 6.4.

Таблица 6.4 – Должностные оклады и коэффициенты перерасчёта

Категория персонала	Месячный должностной оклад, руб./месяц	Коэффициент пересчета
Руководители	30000,0	2,0
Инженеры	15000,0	1,0

Прямую заработную плату (ПЗП) можно рассчитать, разделив месячный должностной оклад на среднее число рабочих дней в месяце, установленное в законодательном порядке (22 дня), и умножив на количество дней, в которых занят специалист (трудоемкость).

Основная заработная плата (ОЗП) рассчитывается суммированием прямой заработной платы и уральского коэффициента (составляет 15% от ПЗП).

Затраты на основную заработную плату представлены в таблице 5.6.

Таблица 6.5 – Расчет основной заработной платы

В рублях

Категории работников	Оклад	Трудоемкость, Чел/дн	Однодневная з/п	Прямая з/п	Уральский коэффициент	Основная з/п
Руководитель	30000	59	1363,00	80454,00	12068,18	92522,73
1 Инженер	15000	55	681,80	37500,00	5625,00	43125,00
2 Инженер	15000	77	681,80	52500,00	7875,00	60375,00
					Итого	196022,7

Прочие прямые затраты (ПЗП) В данную статью расходов включаются затраты на оформление технической.

Таблица 5.7 – Прочие прямые затраты

Наименование расходов	Единица измерения	Количество, шт.	Цена за единицу	Сумма, руб.
1 Оформление технической документации	Лист А4	120	2,5	300
2 Оформление технической документации	Лист А1	6	50	300
Итого				600

Накладные расходы (НР). Накладными расходами в разработке устройства являются затраты на управление и хозяйственное обслуживание, которые не могут быть отнесены прямым счетом на конкретную тему:

- расходы на аренду помещения;
- амортизацию.

Рассчитаем расходы на аренду помещения HP .

$$HP = P_{A,Д} + A. \quad (6.13)$$

Рассчитаем амортизацию A .

$$A = K \cdot n, \quad (6.13)$$

где K – ежемесячные амортизационные отчисления,
 n – срок полезной эксплуатации оборудования.

Срок полезного использования компьютера составляет пять лет. Тогда, процент амортизации равен $100/5 = 20\%$. Стоимость компьютера 30000 руб.

Соответственно, амортизационные отчисления составят : $30000 \cdot 20/100 = 6000$ (руб.)

Ежемесячная амортизация равна: $K = 6000 \text{ руб.}/12 \text{ мес.} = 500$ (руб./мес.)

Разработка проекта составит 4 месяца. Количество месяцев полезной эксплуатации $n = 4$ (мес.).

Таким образом, $A = 500 \cdot 4 = 2000$ (руб.).

Рассчитаем расходы на аренду помещения.

Исходя из стоимости аренды помещения 8000 руб., получаем

$$P_{A,П} = 8000 \cdot 4 = 32000 \text{ (руб.)}$$

Таким образом, накладные расходы равны

$$HP = P_{A,П} + A = 32000 + 2000 = 34000 \text{ (руб.)}$$

Итоговая смета затрат на проведение НИОКР представлена в таблице 5.8.

Таблица 5.8 – Расчет затрат на НИОКР

В рублях

Наименование статьи затрат	Сумма,
материалы	500,00
основная заработная плата	196022,70
отчисления на социальное страхование	-
прочие прямые затраты	600,00
накладные расходы	34000,00
ИТОГО	283549,44

6.4 Экономическая эффективность

Разработанный в данной работе алгоритм испытывался на отладочной плате Xilinx VC707 Evaluation board, стоимость данной платы составляет 3495 \$, что по текущему курсу составляет 227175 руб. Если провести обзор схожих разработок в свободном доступе, то ближайшим аналогом разрабатываемого в дипломной работе устройства является цифровой приемник [5], разрабатываемый для системы на кристалле (SoC). Отличие приведенного в статье алгоритма заключается в применении обработки информации с помощью ресурсов микропроцессора, который входит в систему на кристалле. Основной идеей устройства является накопление информации на большом интервале времени, после чего проводится БПФ на большом интервале времени. Устройство, разрабатываемое в данной дипломной работе, отличается более быстрым временем реагирования, поскольку не предполагается использование микропроцессора, все основные алгоритмы основаны на цифровой обработке в ПЛИС. Рассматриваемое в статье устройство также испытывалось на отладочной плате фирмы Altera Aria 10 SoC Development Kit. Стоимость одной этой отладки составляет 4495 \$, что в переводе в рубли составляет 292175 руб.

Таким образом разрабатываемый макет устройства получается выгоднее на 65000 руб.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной дипломной работы разработано аппаратно-программное устройство определения параметров радиосигнала на основе частотно-временной обработки. Была создана имитационная модель устройства в системе Matlab Simulink, проведены испытания ПО для ПЛИС на макетном образце.

Были проведены исследования возможностей определения параметров радиосигнала на основе частотно-временной обработки. Обнаружена сложность в определении частоты сигнала, когда его истинная частота располагается между двумя соседними бинами БПФ, также обнаружен факт формирования дескрипторов в результате расширения спектра импульсного сигнала на фронтах. По результатам исследований проведена модификация алгоритмов измерения частоты, которая привела к устранению ложных срабатываний для перечисленных случаев.

По результатам испытания разработанное устройство в целом удовлетворяет требованиям ТЗ. Из-за указанных выше модификаций, изменилась минимальная определяемая длительность импульсного сигнала.

Планируется дальнейшее развитие разработки, усовершенствование и оптимизация алгоритмов программы, а также имитационной модели устройства. В планах дальнейшее исследование частотно-временного метода обработки сигнала с использованием перекрытия выборок во времени. Исследование преимущества схемы обнаружения с адаптивным порогом, по сравнению с фиксированным порогом. Исследование работы устройства для случая многосигнальной обстановки.

Рассчитанные экономические показатели позволят определить стоимостную оценку работ для создания серийной продукции, а описанные в разделе безопасности жизнедеятельности мероприятия оградить человека от опасных и вредных производственных факторов, возникающих в процессе эксплуатации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Иванников, Д. Реализация цифровой системы приема и обработки радиосигналов пассивной моноимпульсной радиолокационной станции / Д. Иванников, Н. Потапов, П. Тупиков // Современные технологии автоматизации. Раздел «Цифровая обработка сигналов». – 2010. – № 2. – С. 52–58.
2. Lopez-Risueno, G. Digital channelized receiver based on time-frequency analysis for signal interception / G. Lopez-Risueno, J. Grajal, A. Sanz-Osorio // IEEE transactions on aerospace and electronic system. – 2005. – V. 41. – № 3. – P. 879–898.
3. Shie, Qian Joint time-frequency analysis / Shie Qian, Dapang Chen // IEEE signal processing magazine. – 1999. – V. 16. – № 2. – P. 52–67.
4. Oppenheim, A.V. Discrete-time signal processing / A.V. Oppenheim, R.W. Schaffer // Englewood Cliffs, NJ: Prentice-Hall, 1989.
5. PG105. CORDIC v. 6.0, LogiCORE IP product guide for Vivado design suite. – https://www.xilinx.com/support/documentation/ip_documentation/cordic/v6_0/pg105-cordic.pdf.
6. Maranda, B.H. On the false alarm probability for an overlapped FFT processor / B.H. Maranda // IEEE transactions on aerospace and electronic systems. – 1996. – V. 32. – № 4. – P. 1452–1456.
7. Спектральный анализ на ограниченном интервале времени. Оконные функции. – <http://www.dsplib.ru/content/win/win.html>.
8. Тихонов, В.И. Статистическая радиотехника / В.И. Тихонов. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1982. – 624 с.

ПРИЛОЖЕНИЕ А

А1 Исходный код программы для плис. Блок измерения частоты

```
-----  
-- Блок определения несущей частоты сигнала  
-- Имя файла: peak_finder.vhd  
-- Дата создания: 2016-07-06 00:41:44  
-- Автор: Никитин Н.С.  
--  
-----  
-- Частота работы схемы: 300 МГц  
-----  
--  
-----  
LIBRARY IEEE;  
USE IEEE.std_logic_1164.ALL;  
USE IEEE.numeric_std.ALL;  
  
ENTITY Obnar2_Peak_Finder IS  
    PORT(clk      : IN  std_logic;  
         reset    : IN  std_logic;  
         Data_in  : IN  std_logic_vector(11 DOWNTO 0);  
         Peak     : OUT  std_logic;  
         Data_out : OUT  std_logic_vector(11 DOWNTO 0)  
    );  
END Obnar2_Peak_Finder;  
  
ARCHITECTURE rtl OF Obnar2_Peak_Finder IS  
-- Signals  
SIGNAL Data_in_unsigned      : unsigned(11 DOWNTO 0);  
SIGNAL sData_reg1_out1      : unsigned(11 DOWNTO 0);  
SIGNAL sData_reg2_out1      : unsigned(11 DOWNTO 0);  
SIGNAL sData_reg3_out1      : unsigned(11 DOWNTO 0);  
SIGNAL sData_reg4_out1      : unsigned(11 DOWNTO 0);  
SIGNAL sD2_more_D3_relop1   : std_logic;  
SIGNAL sD2_more_D1_relop1   : std_logic;  
SIGNAL sPeak_1_out1         : std_logic;  
SIGNAL sD2_more_D3n_out1    : std_logic;  
SIGNAL sD2_eqmore_D3_relop1 : std_logic;  
SIGNAL switch_compare_1     : std_logic;  
SIGNAL RAdd2_out1           : unsigned(11 DOWNTO 0);  
SIGNAL RAdd_out1            : unsigned(11 DOWNTO 0);  
SIGNAL sOVFLW_sw_out1       : unsigned(11 DOWNTO 0);  
SIGNAL sDELTA_D_out1        : std_logic;
```

```

SIGNAL sPeak_2_out1           : std_logic;
SIGNAL sD2_more_D1n_out1     : std_logic;
SIGNAL sD2_eqmore_D1_relop1  : std_logic;
SIGNAL switch_compare_1_1    : std_logic;
SIGNAL RAdd4_out1            : unsigned(11 DOWNT0 0);
SIGNAL RAdd3_out1            : unsigned(11 DOWNT0 0);
SIGNAL sOVFLW_sw1_out1       : unsigned(11 DOWNT0 0);
SIGNAL sDELTA_D1_out1        : std_logic;
SIGNAL sPeak_3_out1          : std_logic;
SIGNAL sPeak_out_out1        : std_logic;
SIGNAL sPeak_out_out2        : std_logic;
SIGNAL sPeak_out_reg1        : std_logic;
SIGNAL sPeak_out_reg2        : std_logic;
SIGNAL sPeak_out_reg_out1    : std_logic;

```

```
BEGIN
```

```

-- %%%%%%%%%%
-- Входные регистры данных
-- %%%%%%%%%%
Data_in_unsigned <= unsigned(Data_in);

```

```
sData_reg1_process : PROCESS (clk, reset)
```

```
BEGIN
```

```
IF reset = '0' THEN
```

```
    sData_reg1_out1 <= to_unsigned(16#000#, 12);
```

```
ELSIF clk'EVENT AND clk = '1' THEN
```

```
    sData_reg1_out1 <= Data_in_unsigned;
```

```
END IF;
```

```
END PROCESS sData_reg1_process;
```

```
sData_reg2_process : PROCESS (clk, reset)
```

```
BEGIN
```

```
IF reset = '0' THEN
```

```
    sData_reg2_out1 <= to_unsigned(16#000#, 12);
```

```
ELSIF clk'EVENT AND clk = '1' THEN
```

```
    sData_reg2_out1 <= sData_reg1_out1;
```

```
END IF;
```

```
END PROCESS sData_reg2_process;
```

```
sData_reg3_process : PROCESS (clk, reset)
```

```
BEGIN
```

```
IF reset = '0' THEN
```

```
    sData_reg3_out1 <= to_unsigned(16#000#, 12);
```

```
ELSIF clk'EVENT AND clk = '1' THEN
```

```

    sData_reg3_out1 <= sData_reg2_out1;
END IF;
END PROCESS sData_reg3_process;

```

```

sData_reg4_process : PROCESS (clk, reset)
BEGIN
    IF reset = '0' THEN
        sData_reg4_out1 <= to_unsigned(16#000#, 12);
    ELSIF clk'EVENT AND clk = '1' THEN
        sData_reg4_out1 <= sData_reg3_out1;
    END IF;
END PROCESS sData_reg4_process;

```

```

--%%%%%%%%%
-- Схема определения максимальной амплитуды
--%%%%%%%%%

```

```

sD2_more_D3_reg_process: process(clk,reset)
begin
    if reset = '0' then
        sD2_more_D3_relop1 <= '0';
    elsif clk'event and clk = '1' then
        if (sData_reg2_out1 > sData_reg3_out1) then
            sD2_more_D3_relop1 <= '1';
        else
            sD2_more_D3_relop1 <= '0';
        end if;
    end if;
end process sD2_more_D3_reg_process;

```

```

--sD2_more_D1_relop1 <= '1' WHEN sData_reg2_out1 > sData_reg1_out1 ELSE '0';
sD2_more_D1_reg_process: process(clk,reset)
begin
    if reset = '0' then
        sD2_more_D1_relop1 <= '0';
    elsif clk'event and clk = '1' then
        if (sData_reg2_out1 > sData_reg1_out1) then
            sD2_more_D1_relop1 <= '1';
        else
            sD2_more_D1_relop1 <= '0';
        end if;
    end if;
end process sD2_more_D1_reg_process;

```

```

sPeak_1_out1 <= sD2_more_D3_relop1 AND sD2_more_D1_relop1;

```

```

sD2_more_D3n_out1 <= NOT sD2_more_D3_relop1;
sD2_eqmore_D3_relop1 <= '1' WHEN sData_reg2_out1 >= sData_reg3_out1 ELSE '0';
switch_compare_1 <= '1' WHEN sD2_eqmore_D3_relop1 > '0' ELSE '0';
RAdd2_out1 <= sData_reg3_out1 - sData_reg2_out1;
RAdd_out1 <= sData_reg2_out1 - sData_reg3_out1;

--sOVFLW_sw_out1 <= RAdd2_out1 WHEN switch_compare_1 = '0' ELSE
RAdd_out1;
sOVFLW_sw_out1_reg_process: process(clk,reset)
begin
    if reset = '0' then
        sOVFLW_sw_out1 <= (others => '0');
    elsif clk'event and clk = '1' then
        if switch_compare_1 = '0' then
            sOVFLW_sw_out1 <= RAdd2_out1;
        else
            sOVFLW_sw_out1 <= RAdd_out1;
        end if;
    end if;
end process sOVFLW_sw_out1_reg_process;

sDELTA_D_out1 <= '1' WHEN sOVFLW_sw_out1 <
to_unsigned(16#005#, 12) ELSE '0';
sPeak_2_out1 <= sDELTA_D_out1 AND
(sD2_more_D3n_out1 AND sD2_more_D1_relop1);
sD2_more_D1n_out1 <= NOT sD2_more_D1_relop1;
sD2_eqmore_D1_relop1 <= '1' WHEN sData_reg2_out1 >= sData_reg1_out1 ELSE '0';
switch_compare_1_1 <= '1' WHEN sD2_eqmore_D1_relop1 > '0' ELSE '0';
RAdd4_out1 <= sData_reg1_out1 - sData_reg2_out1;
RAdd3_out1 <= sData_reg2_out1 - sData_reg1_out1;

sOVFLW_sw1_out1_reg_process: process(clk,reset)
begin
    if reset = '0' then
        sOVFLW_sw1_out1 <= (others => '0');
    elsif clk'event and clk = '1' then
        if switch_compare_1_1 = '0' then
            sOVFLW_sw1_out1 <= RAdd4_out1;
        else
            sOVFLW_sw1_out1 <= RAdd3_out1;
        end if;
    end if;
end process sOVFLW_sw1_out1_reg_process;

```



```

sDELTA_D1_out1 <= '1' WHEN sOVFLW_sw1_out1 <
to_unsigned(16#005#, 12) ELSE '0';
sPeak_3_out1 <= sDELTA_D1_out1 AND
(sD2_more_D1n_out1 AND sD2_more_D3_relop1);
sPeak_out_out1 <= sPeak_3_out1 OR (sPeak_1_out1 OR sPeak_2_out1);

sPeak_out_reg1_process: process(clk,reset)
begin
  if reset = '0' then
    sPeak_out_reg1 <= '0';
    sPeak_out_reg2 <= '0';
  elsif clk'event and clk = '1' then
    sPeak_out_reg1 <= sPeak_out_out1;
    sPeak_out_reg2 <= sPeak_out_reg1;
  end if;
end process;

process(sPeak_out_out1,sPeak_out_reg1,sPeak_out_reg2)
begin
  if (sPeak_out_out1 = '1') and (sPeak_out_reg1 = '1') and (sPeak_out_reg2 = '1') then
    sPeak_out_out2 <= '0';
  else
    sPeak_out_out2 <= sPeak_out_out1;
  end if;
end process;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-- Выходной регистр на сигналы
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sPeak_out_reg_process : PROCESS (clk, reset)
BEGIN
  IF reset = '0' THEN
    sPeak_out_reg_out1 <= '0';
  ELSIF clk'EVENT AND clk = '1' THEN
    sPeak_out_reg_out1 <= sPeak_out_out2;
  END IF;
END PROCESS sPeak_out_reg_process;
Data_out <= std_logic_vector(sData_reg4_out1);
Peak <= sPeak_out_reg_out1;
END rtl;

```

A2 Исходный код программы для плис. Арбитр выходной очереди дескрипторов

-- Арбитр выходной очереди дескрипторов сигналов
-- Имя файла: Obnar2_FIFO_BUFFER_OUT
-- Дата создания: 19.04.2016 13:51:26
-- Автор: Никитин Н.С.

-- Частота работы схемы: 125 МГц
--

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;

entity Obnar2_FIFO_BUFFER_OUT is

-- Параметр «времени жизни» дескриптора в очереди, в тактах

Generic (gTIMEOUT : integer := 3906250);

Port (clk : in std_logic;

reset : in std_logic;

data_in : in STD_LOGIC_VECTOR (144 downto 0);

wr_en : in STD_LOGIC;

rd_en : in STD_LOGIC;

data_out : out STD_LOGIC_VECTOR (143 downto 0);

test_out : out std_logic_vector (15 downto 0);

valid : out STD_LOGIC

);

end Obnar2_FIFO_BUFFER_OUT;

architecture Behavioral of Obnar2_FIFO_BUFFER_OUT is

COMPONENT OUTPUT_FIFO_1

PORT (

clk : IN STD_LOGIC;

rst : IN STD_LOGIC;

din : IN STD_LOGIC_VECTOR(144 DOWNT0 0);

wr_en : IN STD_LOGIC;

rd_en : IN STD_LOGIC;

dout : OUT STD_LOGIC_VECTOR(144 DOWNT0 0);

full : OUT STD_LOGIC;

almost_full : OUT STD_LOGIC;

wr_ack : OUT STD_LOGIC;

empty : OUT STD_LOGIC;

```

    almost_empty : OUT STD_LOGIC;
    valid : OUT STD_LOGIC;
    data_count : OUT STD_LOGIC_VECTOR(5 DOWNTO 0)
);
END COMPONENT;

```

```

COMPONENT blk_rom_log
PORT (
    clka : IN STD_LOGIC;
    ena : IN STD_LOGIC;
    addr : IN STD_LOGIC_VECTOR(11 DOWNTO 0);
    dout : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END COMPONENT;

```

```

signal sVALID : std_logic;
signal sRD_EN1 : std_logic;
signal sRD_EN2 : std_logic;
signal sRD_EN3 : std_logic;
signal sENB : std_logic;
signal sdina : std_logic_vector(144 downto 0);
signal sdoutb : std_logic_vector(144 downto 0);
signal sdoutb_reg : std_logic_vector(144 downto 0);
signal sdoutb_reg1 : std_logic_vector(144 downto 0);
signal sPEAKPWR : std_logic_vector(7 downto 0);
signal sDSCR_TIMEOUT : unsigned(21 downto 0);
signal sTIMEOUT : std_logic;
signal sFIFO_data_count_reg : std_logic_vector(5 downto 0);
signal sFIFO_data_count : std_logic_vector(5 downto 0);
signal sRD_reg : std_logic;
signal sRD_reg1 : std_logic;
signal sRD_reg2 : std_logic;
signal sSRV_RD : std_logic;
signal sFIFO_RD : std_logic;
signal sFIFO_full : std_logic;
signal sFIFO_empty : std_logic;
signal sFIFO_aempty : std_logic;
signal sFIFO_afull : std_logic;
signal sFIFO_valid : std_logic;
signal sFIFO_wrack : std_logic;
signal sRESETn : std_logic;
signal sTEST_OUT : std_logic_vector(15 downto 0);

```

```
begin
```

```
sRESETn <= not reset;
```

```
-- Тестовые сигналы для контроля арбитра
```

```
process(clk,reset)
```

```
begin
```

```
  if reset = '0' then
```

```
    sTEST_OUT <= (others => '0');
```

```
  elsif clk'event and clk = '1' then
```

```
    sTEST_OUT(0) <= sFIFO_full;
```

```
    sTEST_OUT(1) <= sFIFO_afull;
```

```
    sTEST_OUT(2) <= sFIFO_empty;
```

```
    sTEST_OUT(3) <= sFIFO_aempty;
```

```
    sTEST_OUT(4) <= sFIFO_valid;
```

```
    sTEST_OUT(5) <= sFIFO_RD;
```

```
    sTEST_OUT(6) <= sRD_EN1;
```

```
    sTEST_OUT(7) <= sVALID;
```

```
    sTEST_OUT(15 downto 8) <= sdoutb(143 downto 136);
```

```
  end if;
```

```
end process;
```

```
test_out <= sTEST_OUT;
```

```
valid <= sVALID;
```

```
sFIFO_RD <= (sRD_reg or sSRV_RD);
```

```
-- %%%%%%%%%%%%%%%
```

```
-- Счетчик «времени жизни» дескриптора и формирование stroba удаление из
```

```
-- очереди
```

```
-- %%%%%%%%%%%%%%%
```

```
process(clk,reset)
```

```
begin
```

```
  if reset = '0' then
```

```
    sDSCR_TIMEOUT <= to_unsigned(0,22);
```

```
    sTIMEOUT <= '0';
```

```
  elsif clk'event and clk = '1' then
```

```
    if sDSCR_TIMEOUT = to_unsigned(gTIMEOUT,22) then
```

```
      sDSCR_TIMEOUT <= to_unsigned(0,22);
```

```
      sTIMEOUT <= '1';
```

```
    else
```

```
      if ((rd_en = '1') or (wr_en = '1')) or (sFIFO_empty = '1') then
```

```
        sDSCR_TIMEOUT <= to_unsigned(0,22);
```

```
        sTIMEOUT <= '0';
```

```
      else
```

```
        sDSCR_TIMEOUT <= sDSCR_TIMEOUT + to_unsigned(1,22);
```

```

        sTIMEOUT <= '0';
    end if;
end if;
end if;
end process;
-- %%%%%%%%%%%%%%
-- Выходной регистр данных дескриптора, формирование сигнала достоверности
-- данных на выходе
-- %%%%%%%%%%%%%%
process(reset,clk,sRD_reg2,sdoutb_reg,sPEAKPWR)
begin
    if reset = '0' then
        data_out <= (others => '0');
    elsif clk'event and clk = '1' then
--        if sFIFO_valid = '1' then
--            data_out <= sdoutb(143 downto 0);
--        else
--            data_out <= (others => '0');
--        end if;
        if sRD_reg2 = '1' then
            data_out(143 downto 36) <= sdoutb_reg1(143 downto 36);
            data_out(35 downto 24) <= ("0000" & sPEAKPWR);
            data_out(23 downto 0) <= sdoutb_reg1(23 downto 0);
        else
            data_out <= (others => '0');
        end if;
    end if;
end process;

process(clk,reset)
begin
    if reset = '0' then
        sdoutb_reg <= (others => '0');
        sdoutb_reg1 <= (others => '0');
    elsif clk'event and clk = '1' then
        sdoutb_reg <= sdoutb;
        sdoutb_reg1 <= sdoutb_reg;
    end if;
end process;

```

fifo_ram: OUTPUT_FIFO_1

```

port map (
  clk      => clk,
  rst      => reset,
  din      => data_in,
  wr_en    => wr_en,
  rd_en    => sFIFO_RD,
  dout     => sdoutb,
  full     => sFIFO_full,
  almost_full => sFIFO_afull,
  wr_ack   => sFIFO_wrack,
  empty    => sFIFO_empty,
  almost_empty => sFIFO_aempty,
  valid    => sFIFO_valid,
  data_count => sFIFO_data_count
);
-- %%%%%%%%%%
-- Регистр-защелка для стробов чтения и записи очереди дескрипторов
-- %%%%%%%%%%
process(clk,reset)
begin
  if reset = '0' then
    sVALID <= '0';
    sRD_EN1 <= '0';
    sRD_EN2 <= '0';
    sRD_EN3 <= '0';
    sRD_reg1 <= '0';
    sRD_reg2 <= '0';
    sFIFO_data_count_reg <= (others => '0');
  elsif clk'event and clk = '1' then
    sRD_EN1 <= rd_en;
    sRD_EN2 <= sRD_EN1;
    sRD_EN3 <= sRD_EN2;
    sVALID <= sRD_EN3;
    sRD_reg1 <= sFIFO_valid;
    sRD_reg2 <= sRD_reg1;
    sFIFO_data_count_reg <= sFIFO_data_count;
  end if;
end process;

-- Блок памяти для перевода амплитуды в дБ

```

```

BLK_ROM_LOG_1 : blk_rom_log
PORT MAP (
  clka => clk,
  ena => sFIFO_valid,
  addr_a => sdoutb(35 downto 24),
  dout_a => sPEAKPWR
);

-- Формирование сигнала чтения из очереди
process(reset,rd_en,wr_en,sFIFO_empty,sFIFO_full,sTIMEOUT)--
sRD_AD_unsgn,sWR_AD_unsgn,sdoutb,sEMPTY3)
begin
  if reset = '0' then
    sRD_reg <= '0';
    sSRV_RD <= '0';
  else
    if ((sFIFO_empty = '1') and (rd_en = '1')) then
      sRD_reg <= '0';
    else
      sRD_reg <= rd_en;
    end if;

    if ((sFIFO_full = '1') and (wr_en = '1')) or (sTIMEOUT = '1') then
      sSRV_RD <= '1';
    else
      sSRV_RD <= '0';
    end if;
  end if;
end process;

end Behavioral;

```

А3 Исходный код программы для плис. Блок определения параметров сигнала

```
-----  
-- Блок определения параметров сигнала,  
-- верхний уровень.  
-- Имя файла: Obnar2_Channel_handler.vhd  
-- Дата создания: 2016-03-28 12:17:55  
-- Автор: Никитин Н.С.
```

```
-----  
-- Частота работы схемы: 125 МГц  
-----  
--  
-- Имя модуля: Obnar2_Channel_handler  
--  
-----
```

```
LIBRARY IEEE;  
USE IEEE.std_logic_1164.ALL;  
USE IEEE.numeric_std.ALL;
```

```
ENTITY Obnar2_Channel_handler IS
```

```
    GENERIC ( gCHANNEL_NUM      : std_logic_vector(7 downto 0) := X"01";  
              gCW_MISS_LIM     : integer := 12188;  
              gPW_LIM          : integer := 12188;  
              gPACKET_LIM      : integer := 3;  
              gEND_PACKET_LIM  : integer := 101563  
            );
```

```
    PORT(  
        clk          : IN  std_logic;  
        reset        : IN  std_logic;  
        clk_en       : IN  std_logic;  
        PEAK_in      : IN  std_logic;  
        PHASE        : IN  std_logic;  
        BASE_NUM     : IN  std_logic;  
        FINECOUNT   : IN  std_logic_vector(17 DOWNT0 0);  
        COARSECOUNT  : IN  std_logic_vector(15 DOWNT0 0);  
        BLANK        : IN  std_logic;  
        BMCH         : IN  std_logic;  
        LVL1         : IN  std_logic_vector(11 DOWNT0 0);  
        LVL2         : IN  std_logic_vector(11 DOWNT0 0);  
        HYST         : IN  std_logic_vector(11 downto 0);  
        milisec_hit  : IN  std_logic;  
        AMPL_INT     : IN  std_logic_vector(11 DOWNT0 0);  
        CH1_PHASE_INT : IN  std_logic_vector(9 DOWNT0 0);  
        channel_data_valid : IN  std_logic;  
        FREQ         : OUT  std_logic_vector(7 downto 0);
```



```

        PA_OUT          : OUT  std_logic_vector(11 DOWNT0 0);
        PDW_EN          : OUT  std_logic;
        PW_OUT          : OUT  std_logic_vector(13 DOWNT0 0);
        FINETIME_OUT    : OUT  std_logic_vector(17 DOWNT0 0);
        COARSETIME_OUT  : OUT  std_logic_vector(15 DOWNT0 0);
        PACKET_WIDTH_OUT: OUT  std_logic_vector(9 DOWNT0 0);
        PERIOD_OUT      : OUT  std_logic_vector(15 DOWNT0 0);
        PHASE1          : OUT  std_logic_vector(9 DOWNT0 0);
END Obnar2_Channel_handler;

```

ARCHITECTURE rtl OF Obnar2_Channel_handler IS

-- Component Declarations

COMPONENT Obnar2_CW_HANDLER

```

    GENERIC (
        gMISS_LIM      : integer := 6094;
        gPW_LIM        : integer := 12188);

```

PORT(

```

    clk          : IN  std_logic;
    reset        : IN  std_logic;
    clk_en       : IN  std_logic;
    DETECT       : IN  std_logic;
    DETECT_VALID : IN  std_logic;
    AMPL_IN      : IN  std_logic_vector(11 DOWNT0 0);
    BLANK        : IN  std_logic;
    MILLISEC_HIT : IN  std_logic;
    DETECT_START : IN  std_logic;
    FINE_TIME    : IN  std_logic_vector(17 DOWNT0 0);
    COARSE_TIME  : IN  std_logic_vector(15 DOWNT0 0);
    CW_enable    : OUT std_logic;
    PA_max       : OUT std_logic_vector(11 DOWNT0 0);
    interrupt_out : OUT std_logic;
    FINE_TIME_OUT : OUT std_logic_vector(17 DOWNT0 0);
    COARSE_TIME_OUT : OUT std_logic_vector(15 DOWNT0 0);

```

END COMPONENT;

COMPONENT Obnar2_PS_HANDLER IS

```

    GENERIC (
        gPW_LIM          : integer := 12188;
        gPACKET_LIM      : integer := 3;
        gPERIOD_WINDOW   : integer := 4;
        gEND_PACKET_LIM  : integer := 60939);

```

PORT(

```

    clk          : IN  std_logic;
    reset        : IN  std_logic;
    clk_enable   : IN  std_logic;

```

```

DETECT          : IN  std_logic;
DETECT_VALID    : IN  std_logic;
AMPL_INT        : IN  std_logic_vector(11 DOWNT0 0);
DETECT_END      : IN  std_logic;
BLANK           : IN  std_logic;
MILISEC_HIT     : IN  std_logic;
CW_EN           : IN  std_logic;
DETECT_START    : IN  std_logic;
FINE_TIME       : IN  std_logic_vector(17 DOWNT0 0);
COARSE_TIME     : IN  std_logic_vector(15 DOWNT0 0);
interrupt_out   : OUT  std_logic;
PA_max          : OUT  std_logic_vector(11 DOWNT0 0);
PW              : OUT  std_logic_vector(13 DOWNT0 0);
FINE_TIME_OUT   : OUT  std_logic_vector(17 DOWNT0 0);
COARSE_TIME_OUT : OUT  std_logic_vector(15 DOWNT0 0);
PACKET_WIDTH_OUT : OUT  std_logic_vector(9 DOWNT0 0);
PERIOD_OUT      : OUT  std_logic_vector(15 DOWNT0 0));
END COMPONENT;
```

COMPONENT Obnar2_PHASE_HANDLER

```

PORT(
  clk          : IN  std_logic;
  reset        : IN  std_logic;
  clk_en       : IN  std_logic;
  CH1_PHASE    : IN  std_logic_vector(9 DOWNT0 0);
  PHASE        : IN  std_logic;
  BASE_NUM     : IN  std_logic;
  DETECT       : IN  std_logic;
  DETECT_VALID : IN  std_logic;
  DETECT_END   : IN  std_logic;
  PDW_EN       : IN  std_logic;
  CH1_PHASE_DIFF : OUT  std_logic_vector(9 DOWNT0 0);
  PHASE1       : OUT  std_logic_vector(9 DOWNT0 0));
END COMPONENT;
```

-- Signals

```

SIGNAL sAMPL_INT_signed    : unsigned(11 DOWNT0 0);
SIGNAL sLVL1_signed        : unsigned(11 DOWNT0 0);
SIGNAL sLVL2_signed        : unsigned(11 DOWNT0 0);
signal sTHLD_HYST_UP       : unsigned(11 downto 0);
signal sTHLD_HYST_DOWN     : unsigned(11 downto 0);
SIGNAL sTHLD_HYST_MID      : unsigned(11 DOWNT0 0);
signal sHYST_div2          : unsigned(11 downto 0);
SIGNAL sAmpl_compare_relop1 : std_logic;
```

```

signal sDETECT_1           : std_logic;
signal sDETECT_2           : std_logic;
signal sDETECT_3           : std_logic;
SIGNAL sDetect_reg         : std_logic;
SIGNAL sDetect_reg1        : std_logic;
SIGNAL sCH_Valid_reg       : std_logic;
SIGNAL sAmpl_reg_out1      : unsigned(11 DOWNT0 0);
SIGNAL sMilisec_reg_out1   : std_logic;
SIGNAL sDetect_start       : std_logic;
SIGNAL sDetect_end         : std_logic;
SIGNAL sFINECOUNT_unsigned : unsigned(17 DOWNT0 0);
SIGNAL sFineTime_reg_out1  : unsigned(17 DOWNT0 0);
SIGNAL sCW_enable          : std_logic;
SIGNAL sCW_PA_max          : std_logic_vector(11 DOWNT0 0);
SIGNAL sCW_inter_out       : std_logic;
SIGNAL sCW_FT_out          : std_logic_vector(17 DOWNT0 0);
SIGNAL sCW_CT_out          : std_logic_vector(15 DOWNT0 0);
SIGNAL sPs_inter_out       : std_logic;
SIGNAL sPS_PA_max          : std_logic_vector(11 DOWNT0 0);
SIGNAL sPS_PW              : std_logic_vector(13 DOWNT0 0);
SIGNAL sPS_FT_out          : std_logic_vector(17 DOWNT0 0);
SIGNAL sPS_CT_out          : std_logic_vector(15 DOWNT0 0);
SIGNAL sPS_PACKET_out      : std_logic_vector(9 DOWNT0 0);
SIGNAL sPS_PERIOD          : std_logic_vector(15 DOWNT0 0);
SIGNAL sPS_PA_max_sign     : unsigned(11 DOWNT0 0);
SIGNAL sCW_PA_max_sign     : unsigned(11 DOWNT0 0);
SIGNAL PA_switch_out1      : unsigned(11 DOWNT0 0);
SIGNAL inter_switch_out1   : std_logic;
SIGNAL sPS_PW_unsign       : unsigned(13 DOWNT0 0);
SIGNAL PW_switch_out1      : unsigned(13 DOWNT0 0);
SIGNAL sPS_FT_out_unsign   : unsigned(17 DOWNT0 0);
SIGNAL sCW_FT_out_unsign   : unsigned(17 DOWNT0 0);
SIGNAL FT_switch_out1      : unsigned(17 DOWNT0 0);
SIGNAL sPS_CT_out_unsign   : unsigned(15 DOWNT0 0);
SIGNAL sCW_CT_out_unsign   : unsigned(15 DOWNT0 0);
SIGNAL CT_switch_out1      : unsigned(15 DOWNT0 0);
SIGNAL sPS_PACKET_out_unsign : unsigned(9 DOWNT0 0);
SIGNAL PACKET_WIDTH_switch_out1 : unsigned(9 DOWNT0 0);
SIGNAL sPS_PERIOD_unsign   : unsigned(15 DOWNT0 0);
SIGNAL PERIOD_switch_out1  : unsigned(15 DOWNT0 0);
SIGNAL sCH1_PHASE_INT_signed : signed(9 DOWNT0 0);
SIGNAL sPM_NUM_IN_unsigned : unsigned(1 DOWNT0 0);
SIGNAL sPM_NUM_IN_reg      : unsigned(1 DOWNT0 0);
SIGNAL sCH1_phase_reg_out1 : signed(9 DOWNT0 0);

```

```

SIGNAL phase_dif : std_logic_vector(9 DOWNT0 0);
SIGNAL sPHASE1 : std_logic_vector(9 DOWNT0 0);
signal sDETECT_EN : std_logic := '0';
signal sTHLD_INIT : std_logic := '0';
signal sDETECT_out : std_logic;
signal sEND_PACKET_dbg : std_logic;
signal sEND_PACKET_reg_db : std_logic;
signal sPDW_EN_dbg : std_logic;

```

```
BEGIN
```

```
--%%%%%%%%%
```

```
-- Блок определения параметров непрерывного сигнала
```

```
--%%%%%%%%%
```

```
Obnar2_CW_HANDLER_1 : Obnar2_CW_HANDLER
```

```
  GENERIC MAP ( gMISS_LIM => gCW_MISS_LIM,
               gPW_LIM => gPW_LIM)
```

```
  PORT MAP( clk => clk,
```

```
            reset => reset,
```

```
            clk_en => clk_en,
```

```
            DETECT => sDETECT_3,
```

```
            DETECT_VALID => sCH_Valid_reg,
```

```
            AMPL_INT => std_logic_vector(sAmpl_reg_out1),
```

```
            BLANK => BLANK,
```

```
            MILLISEC_HIT => sMilisec_reg_out1,
```

```
            DETECT_START => sDetect_start,
```

```
            FINE_TIME => std_logic_vector(sFineTime_reg_out1),
```

```
            COARSE_TIME => COARSECOUNT,
```

```
            CW_enable => sCW_enable,
```

```
            PA_max => sCW_PA_max,
```

```
            interrupt_out => sCW_inter_out,
```

```
            FINE_TIME_OUT => sCW_FT_out,
```

```
            COARSE_TIME_OUT => sCW_CT_out
```

```
          );
```

```
--%%%%%%%%%
```

```
-- Блок определения параметров импульсного сигнала
```

```
--%%%%%%%%%
```

```
Obnar2_PS_HANDLER_1: Obnar2_PS_HANDLER
```

```
  PORT MAP(
```

```
    clk => clk,
```

```
    reset => reset,
```

```
    clk_enable => clk_en,
```

```
    DETECT => sDETECT_3,
```

```
    DETECT_VALID => sCH_Valid_reg,
```

```
    AMPL_INT => std_logic_vector(sAmpl_reg_out1),
```

```

DETECT_END      => sDetect_end,
BLANK           => BLANK,
MILISEC_HIT    => sMilisec_reg_out1,
CW_EN          => sCW_enable,
DETECT_START   => sDetect_start,
FINE_TIME      => std_logic_vector(sFineTime_reg_out1),
COARSE_TIME    => COARSECOUNT,
interrupt_out  => sPS_inter_out,
PA_max         => sPS_PA_max,
PW            => sPS_PW,
FINE_TIME_OUT  => sPS_FT_out,
COARSE_TIME_OUT => sPS_CT_out,
PACKET_WIDTH_OUT => sPS_PACKET_out,
PERIOD_OUT     => sPS_PERIOD);
--%%%%%%%%
-- Блок измерения разности фаз
--%%%%%%%%
Obnar2_PHASE_HANDLER_1 : Obnar2_PHASE_HANDLER
PORT MAP(
  clk => clk,
  reset => reset,
  clk_en => clk_en,
  CH1_PHASE => std_logic_vector(sCH1_phase_reg_out1), -- sfix10
  PHASE => PHASE,
  DETECT => sDETECT_3,
  DETECT_VALID => sCH_Valid_reg,
  DETECT_END => sDetect_end,
  PDW_EN => inter_switch_out1,
  CH1_PHASE_DIFF => phase_dif,
  PHASE1 => sPHASE1);

--%%%%%%%%
-- Схема управления гистерезисом
--%%%%%%%%
sAMPL_INT_signed <= unsigned(AMPL_INT);
sLVL1_signed     <= unsigned(LVL1);
sLVL2_signed     <= unsigned(LVL2);
sHYST_div2      <= SHIFT_RIGHT(unsigned(HYST),1);

process(clk, reset)
begin
  if reset = '0' then
    sTHLD_HYST_MID <= to_unsigned(0,12);
    sTHLD_HYST_DOWN <= to_unsigned(0,12);

```

```

    sTHLD_HYST_UP  <= to_unsigned(0,12);
elseif clk'event and clk = '1' then
    if BMCH = '0' then
        sTHLD_HYST_MID <= sLVL1_signed;
    else
        sTHLD_HYST_MID <= sLVL2_signed;
    end if;
    if sHYST_div2 < sLVL1_signed then
        sTHLD_HYST_UP  <= sTHLD_HYST_MID + sHYST_div2;
        sTHLD_HYST_DOWN <= sTHLD_HYST_MID - sHYST_div2;
    else
        sTHLD_HYST_UP  <= sTHLD_HYST_MID;
        sTHLD_HYST_DOWN <= sTHLD_HYST_MID;
    end if;
end if;
end process;
-- Схема разрешения гистерезиса
THLD_init_process : process(clk, reset, clk_en)
begin
    if reset = '0' then
        sTHLD_INIT <= '0';
    elseif clk'event and clk = '1' then
        if sTHLD_HYST_MID = to_unsigned(0,12) then
            sTHLD_INIT <= '0';
        else
            sTHLD_INIT <= '1';
        end if;
    end if;
end process;

process(channel_data_valid)
begin
    if (channel_data_valid = '1')then
        sDETECT_EN <= '1';
    else
        sDETECT_EN <= '0';
    end if;
end process;

```

```
--%%%%%%%%%
-- Схема амплитудного компаратора с гистерезисом, схема учета результатов
-- измерителя частота
--%%%%%%%%%
```

```
sDetect_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
  IF reset = '0' THEN
    sDetect_reg <= '0';
    sDetect_reg1 <= '0';
    sDETECT_1 <= '0';
    sDETECT_2 <= '0';
  ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
    IF (BLANK = '1') or (sTHLD_INIT = '0') then
      sDetect_reg <= '0';
      sDetect_reg1 <= sDETECT_3;
      sDETECT_1 <= '0';
      sDETECT_2 <= '0';
    elsif sDETECT_EN = '1' then
      if (PEAK_in = '1') THEN
        if sAMPL_INT_signed > sTHLD_HYST_UP then
          sDetect_reg <= '1';
        elsif sAMPL_INT_signed < sTHLD_HYST_DOWN then
          sDetect_reg <= '0';
        end if;
        sDETECT_1 <= sDetect_reg;
        sDETECT_2 <= sDETECT_1;
      else
        sDetect_reg <= '0';
        sDETECT_1 <= '0';
        sDETECT_2 <= '0';
      end if;
      sDetect_reg1 <= sDETECT_3;
    END IF;
  END IF;
END PROCESS sDetect_reg_process;
```

```
sDETECT_3 <= sDETECT_1 and sDETECT_2 and sDetect_reg;
```

```
sCH_Valid_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
  IF reset = '0' THEN
    sCH_Valid_reg <= '0';
  ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
    sCH_Valid_reg <= channel_data_valid;
```

```

    END IF;
END PROCESS sCH_Valid_reg_process;

sAmpl_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
    IF reset = '0' THEN
        sAmpl_reg_out1 <= to_unsigned(16#000#, 12);
    ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
        IF channel_data_valid = '1' THEN
            sAmpl_reg_out1 <= sAMPL_INT_signed;
        END IF;
    END IF;
END PROCESS sAmpl_reg_process;

sMilisec_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
    IF reset = '0' THEN
        sMilisec_reg_out1 <= '0';
    ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
        sMilisec_reg_out1 <= milisec_hit;
    END IF;
END PROCESS sMilisec_reg_process;

Detect_start_process : PROCESS (reset,sDETECT_3,sDetect_reg1,sDETECT_EN)
BEGIN
    IF reset = '0' THEN
        sDetect_start <= '0';
    ELSE
        if (sDETECT_3 = '1') and (sDetect_reg1 = '0') and (sDETECT_EN = '1') then
            sDetect_start <= '1';
        else
            sDetect_start <= '0';
        end if;
    END IF;
END PROCESS Detect_start_process;

Detect_end_process : PROCESS (reset,sDETECT_3,sDetect_reg1,sDETECT_EN)
BEGIN
    IF reset = '0' THEN
        sDetect_end <= '0';
    ELSE
        if (sDETECT_3 = '0') and (sDetect_reg1 = '1') and (sDETECT_EN = '1') then
            sDetect_end <= '1';
        else

```



```

    sDetect_end <= '0';
  end if;
END IF;
END PROCESS Detect_end_process;

```

```

sFINECOUNT_unsigned <= unsigned(FINECOUNT);

```

```

sFineTime_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
  IF reset = '0' THEN
    sFineTime_reg_out1 <= to_unsigned(16#00000#, 18);
  ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
    sFineTime_reg_out1 <= sFINECOUNT_unsigned;
  END IF;
END PROCESS sFineTime_reg_process;

```

```

sPS_PA_max_sign      <= unsigned(sPS_PA_max);
sCW_PA_max_sign      <= unsigned(sCW_PA_max);
sPS_PW_unsign        <= unsigned(sPS_PW);
sPS_FT_out_unsign    <= unsigned(sPS_FT_out);
sCW_FT_out_unsign    <= unsigned(sCW_FT_out);
sPS_CT_out_unsign    <= unsigned(sPS_CT_out);
sCW_CT_out_unsign    <= unsigned(sCW_CT_out);
sPS_PACKET_out_unsign <= unsigned(sPS_PACKET_out);
sPS_PERIOD_unsign    <= unsigned(sPS_PERIOD);

```

```

--%%%%%%%%%%
-- Мультиплексор данных от блоков определения параметров сигналов. Выбор
-- типа формируемого дескриптора

```

```

--%%%%%%%%%%

```

```

Handler_switch_process :

```

```

  process(sCW_enable, reset, sPS_inter_out, sCW_inter_out, sCW_PA_max_sign,
sPS_PA_max_sign, sPS_PW_unsign, sCW_FT_out_unsign, sPS_FT_out_unsign,
sCW_CT_out_unsign, sPS_CT_out_unsign,
sPS_PACKET_out_unsign, sPS_PERIOD_unsign)

```

```

  begin

```

```

    if reset = '0' then

```

```

      PA_switch_out1      <= (others=> '0');
      inter_switch_out1   <= '0';
      PW_switch_out1      <= (others => '0');
      FT_switch_out1      <= (others => '0');
      CT_switch_out1      <= (others => '0');
      PACKET_WIDTH_switch_out1 <= (others => '0');
      PERIOD_switch_out1  <= (others => '0');

```

```

    else

```

```

if sCW_enable = '1' then
    PA_switch_out1      <= sCW_PA_max_sign;
    inter_switch_out1   <= sCW_inter_out;
    PW_switch_out1      <= (others => '0');
    FT_switch_out1      <= sCW_FT_out_unsign;
    CT_switch_out1      <= sCW_CT_out_unsign;
    PACKET_WIDTH_switch_out1 <= (others => '0');
    PERIOD_switch_out1  <= (others => '0');
else
    PA_switch_out1      <= sPS_PA_max_sign;
    inter_switch_out1   <= sPS_inter_out;
    PW_switch_out1      <= sPS_PW_unsign;
    FT_switch_out1      <= sPS_FT_out_unsign;
    CT_switch_out1      <= sPS_CT_out_unsign;
    PACKET_WIDTH_switch_out1 <= sPS_PACKET_out_unsign;
    PERIOD_switch_out1  <= sPS_PERIOD_unsign;
end if;
end if;
end process;

```

```

sCH1_PHASE_INT_signed <= signed(CH1_PHASE_INT);
sCH1_phase_reg_process : PROCESS (clk, reset, clk_en)
BEGIN
    IF reset = '0' THEN
        sCH1_phase_reg_out1 <= to_signed(16#000#, 10);
    ELSIF (clk'EVENT AND clk = '1') and (clk_en = '1') THEN
        IF channel_data_valid = '1' THEN
            sCH1_phase_reg_out1 <= sCH1_PHASE_INT_signed;
        END IF;
    END IF;
END PROCESS sCH1_phase_reg_process;

```

```

process(clk,reset,clk_en)
begin
    if reset = '0' then
        PDW_EN          <= '0';
        FREQ             <= (others => '0');
        PA_OUT           <= (others => '0');
        PW_OUT           <= (others => '0');
        FINETIME_OUT     <= (others => '0');
        COARSETIME_OUT   <= (others => '0');
        PACKET_WIDTH_OUT <= (others => '0');
    end if;
end process;

```

```

    PERIOD_OUT          <= (others => '0');
    PHASE1              <= (others => '1');
elseif (clk'event and clk = '1') and (clk_en = '1') then
    if inter_switch_out1 = '1' then
        FREQ <= std_logic_vector(unsigned(gCHANNEL_NUM)+to_unsigned(1,8));
    else
        FREQ <= (others => '0');
    end if;
    PDW_EN              <= inter_switch_out1;
    PA_OUT              <= std_logic_vector(PA_switch_out1);
    PW_OUT              <= std_logic_vector(PW_switch_out1);
    FINETIME_OUT       <= std_logic_vector(FT_switch_out1);
    COARSETIME_OUT     <= std_logic_vector(CT_switch_out1);
    PACKET_WIDTH_OUT   <= std_logic_vector(PACKET_WIDTH_swit-
ch_out1);
    PERIOD_OUT         <= std_logic_vector(PERIOD_switch_out1);
    PHASE1             <= sPHASE1;
    end if;
end process;

END rtl;

```