

Министерство науки и высшего образования Российской Федерации  
Томский государственный университет  
систем управления и радиоэлектроники

Д. А. Покаместов  
Ж. Т. Эрдынеев

**МНОГОКАНАЛЬНЫЕ ЦИФРОВЫЕ СИСТЕМЫ ПЕРЕДАЧИ / СИСТЕМЫ И  
УСТРОЙСТВА РАДИОСВЯЗИ И РАДИОДОСТУПА**

Методические указания по проведению практических занятий и организации  
самостоятельной работы для студентов направления 11.03.01 Радиотехника и 11.03.02  
Инфокоммуникационные технологии и системы связи

Томск  
2022

**УДК** 621.396  
**ББК** 32.884  
П 48

**Рецензент:**

**Крюков Я. В.**, доцент кафедры телекоммуникаций и основ радиотехники ТУСУР, канд техн. наук,

**Покаместов, Дмитрий Алексеевич**

П 48 Многоканальные цифровые системы передачи / Системы и устройства радиосвязи и радиодоступа: Методические указания по проведению практических занятий и организации самостоятельной работы для студентов направления 11.03.01 Радиотехника и 11.03.02 Инфокоммуникационные технологии и системы связи/ Д. А. Покаместов, Ж. Т. Эрдынеев. – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2022. – 18 с.

Настоящее пособие предназначено для студентов радиотехнического факультета по специальности 11.03.01 Радиотехника и 11.03.02 Инфокоммуникационные технологии и системы связи. Материалы пособия призваны закрепить и углубить полученные навыки, а также показать возможности применения ПЛИС в задачах реализации компонентов многоканальных цифровых систем передачи.

Одобрено на заседании кафедры ТОР, протокол № 5 от 17 февраля 2022 г.

УДК 621.396  
ББК 32.884

© Покаместов Д. А., Эрдынеев Ж. Т., 2022  
© Томск. гос. ун-т систем управления и радиоэлектроники, 2022

## Оглавление

Введение .....	4
1. Генератор случайных бит.....	5
2. Сверточный кодер.....	6
3. QPSK модулятор .....	7
4. 16-APSK модулятор.....	9
5. Создание OFDM модулятора. Объединение модулей в итоговый проект .....	11
Список литературы .....	18

## ВВЕДЕНИЕ

В задачах формирования цифровых сигналов сложной формы в современных системах связи нашли широкое применение программируемые логические интегральные схемы (ПЛИС). Благодаря гибкой архитектуре и высокой скорости обработки, ПЛИСы выигрывают у простых микросхем и микроконтролеров. Процесс проектирования на ПЛИС сводится к созданию программной прошивки в среде проектирования (для плис Altera это пакет Quartus II).

Основы проектирования и верификации были изучены в прошлом семестре. Данные практические работы призваны закрепить и углубить полученные навыки, а также показать возможности применения ПЛИС в задачах реализации компонентов многоканальных цифровых систем передачи.

Практические работы будут выполняться на отладочных платах SoCKit (рисунок В.1), включающих в себя процессор, периферию и интерфейс памяти, соединенные с ПЛИС широкополосными соединительными магистралями. В рамках данного курса возможности интеграции процессора и ПЛИС рассматриваться не будут.

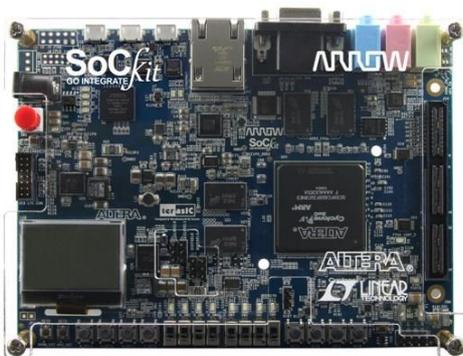


Рисунок В.1 – Отладочная плата SoCKit Altera

К отладочной плате должна быть подключена плата THDB-ADA (рисунок В.2) – АЦП-ЦАП.



Рисунок В.2 – THDB-ADA

В ходе практических занятий сигналы, формируемые в ПЛИС будут подаваться на ЦАП THDB-ADA и должны наблюдаться на экране осциллографа Agilent.

## 1. ГЕНЕРАТОР СЛУЧАЙНЫХ БИТ

Генератор ПСП (псевдослучайных последовательностей) – важный элемент цифровой техники, получивший применение в огромном количестве прикладных задач. Способов реализации таких устройств несколько, один из них – формирование на основе генератора М-последовательностей. Пример такого генератора приведен на рисунке 1.1а.

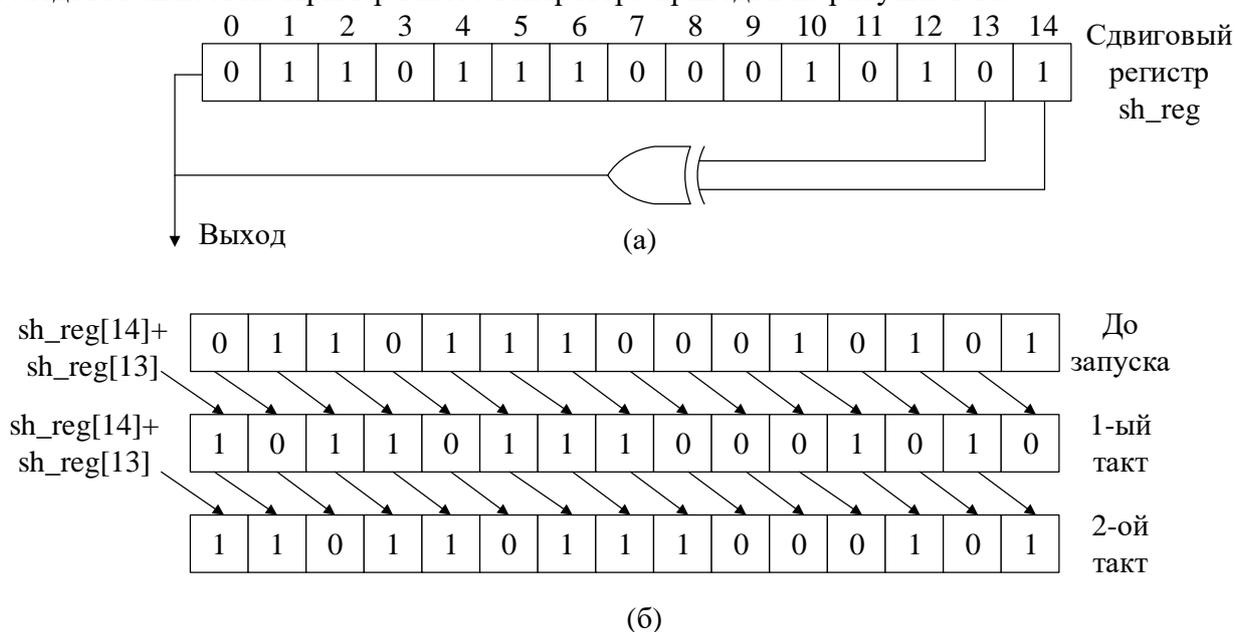


Рисунок 1.1 – структура генератора случайных бит (а); состояния сдвигового регистра внутри генератора в первые такты его работы

### Задание на практику

Вам необходимо реализовать модуль генератора случайных бит. Модуль имеет следующие порты:

clk – тактовая частота;

rst – сигнал сброса;

en – вход разрешения начала работы;

bit – выходной порт, на который каждый такт поступает сформированный бит.

Также необходимо объявить внутренний 15-разрядный регистр, присвоить ему инициализирующие значения (рисунок 1.1 (а)) и на его основе в блоке always реализовать конструкцию сдвигового регистра. Каждый такт первый разряд сдвигового регистра получает значение, равное сумме по модулю два 13 и 14 разрядов. Значения остальных разрядов сдвигаются на один вправо (если реализована схема, приведенная на рисунке 1). Выход модуля подключен к первому разряду сдвигового регистра.

## 2. СВЕРТОЧНЫЙ КОДЕР

Помехоустойчивое кодирование – тип кодирования, позволяющий обнаружить и исправить ошибки, возникающие при передаче информации по линиям связи. При этом в передаваемое сообщение добавляется избыточность. Сверточный кодер представляет собой сдвиговый регистр длиной  $N-1$ , к определенным разрядам которого подключены сумматоры по модулю два. Устройство имеет следующую логику работы, рисунок 2.1.

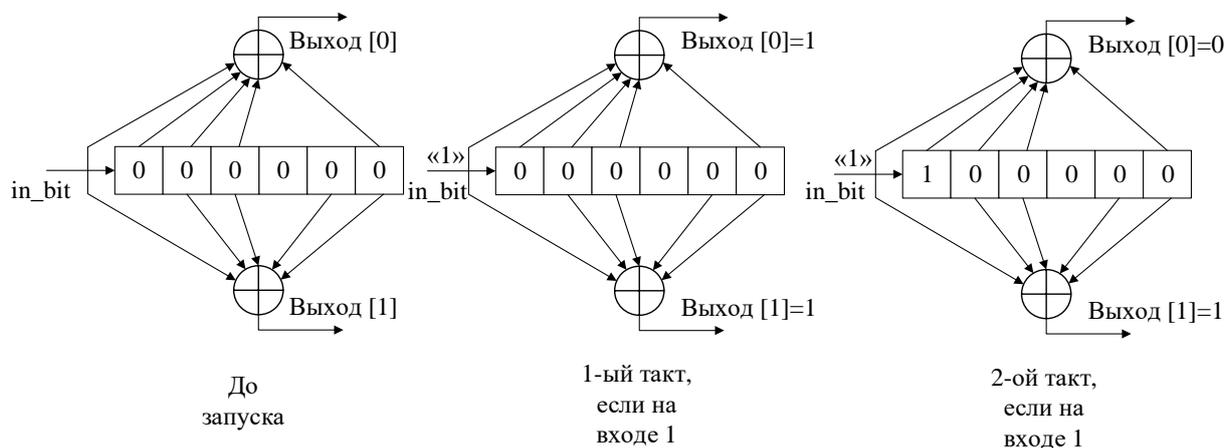


Рисунок 2.1 – Схема сверточного кодера и значения на его выходах в первые такты работы

### Задание на практику

Вам необходимо реализовать модуль сверточного кодера. Модуль имеет следующие порты:

clk – тактовая частота;

rst – сигнал сброса;

en – вход разрешения начала работы;

in\_bit – информационный вход

out\_bit [1:0] – выходной порт, на который каждый такт поступает два бита с выходов сумматоров.

Также необходимо объявить внутренний 6-разрядный регистр и на его основе в блоке always реализовать конструкцию сдвигового регистра. Каждый такт первому разряду сдвиговому регистру присваивается значение со входа модуля (in\_bit). Значения остальных разрядов сдвигаются на один вправо (если реализована схема, приведенная на рисунке 2.1). Выход модуля подключен к двум сумматорам, на которые подаются значения соответствующих разрядов сдвигового регистра.

### 3. QPSK МОДУЛЯТОР

Квадратурная фазовая манипуляция QPSK – один из видов модуляции, используемой в цифровой радиосвязи. Структурная схема QPSK модулятора приведена на рисунке 3.1.

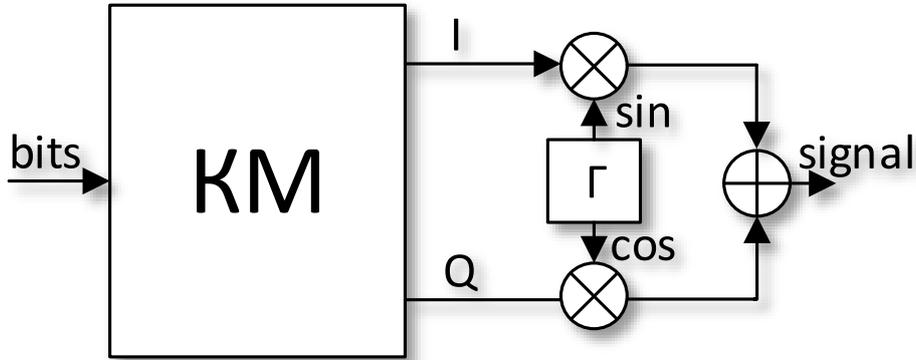


Рисунок 3.1 – Структурная схема QPSK модулятора

Формирования сигнала QPSK описывает формула:

$$E(t) = I(t) \cos(2\pi ft) + Q(t) \sin(2\pi ft)$$

Сам процесс QPSK модуляции сводится к выбору I и Q компонент в зависимости от поступающей комбинации пары бит. Правило такого отображения можно описать таблицей 3.1.

Таблица 3.1 – QPSK

QPSK		
Биты ( $b_0b_1$ )	Q	I
00	-1	-1
01	-1	1
10	1	-1
11	1	1

Или диаграммой «созвездие, рисунок 3.2.

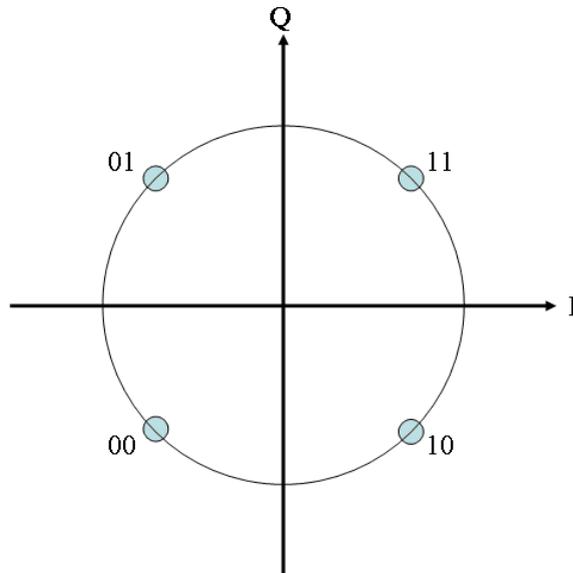


Рисунок 3.2 – Диаграмма «созвездие» QPSK модуляции

### *Задание на практику*

Модуль должен реализовать логику работы QPSK модулятора в соответствии с таблицей. При этом значение выходного символа «-1» соответствует минимальному без знаковому числу, которое можно подать на вход 14-битного ЦАП, а «1» - максимальному без знаковому числу.

Порты разрабатываемого модуля:

Входы:

clk – тактовая частота;

rst – сигнал сброса;

en – вход разрешения начала работы;

input\_bits [1:0] – входные биты.

Выходы:

output\_symbol\_real [13:0] – синфазная составляющая модулированного сигнала;

output\_symbol\_imag [13:0] – квадратурная составляющая модулированного сигнала.

Ход работы:

- 1) Самостоятельно реализуйте модуль в отдельном модуле Modelsim.
- 2) Напишите тестовый модуль –testbench, корректно задайте изменение управляющих сигналов.
- 3) Проведите временную симуляцию модуля в Modelsim.
- 4) Сохраните код модуля и результаты симуляции.

#### 4. 16-APSK МОДУЛЯТОР

Модуляция APSK нашла широкое применение в современных системах передачи информации, например в системах спутникового телевидения DVB. Стандартами DVB определены модуляции QPSK, 8PSK, 16APSK и 32APSK.

Семейство модуляций APSK имеет схожую природу с QAM модуляциями, манипулирование фазой комплексного сигнала можно перевести в определенные уровни амплитуд синфазной и квадратурной составляющих. Схема APSK модулятора совпадает с другими видами квадратурных модуляций, разница в таблице соответствия (блок КМ на рисунке 4.1).

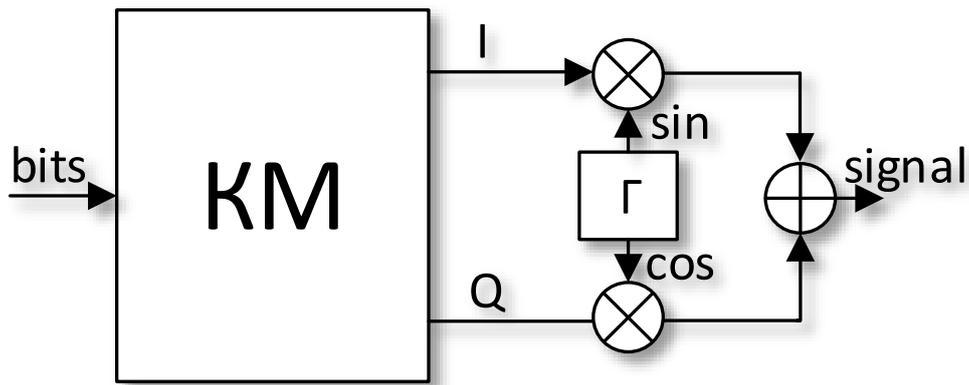


Рисунок 4.1 – Структурная схема квадратурного модулятора

Формирования сигнала описывает формула:

$$E(t) = I(t) \cos(2\pi ft) + Q(t) \sin(2\pi ft)$$

Реализация APSK похожа на реализацию QPSK модулятора, его модуль можно взять за основу. Вам необходимо самостоятельно рассчитать значение точек созвездия, исходя из угла и отношения радиусов  $R_1/R_2 \gamma=2.7$ .

Диаграмма «созвездие», приведена на рисунке 4.2.

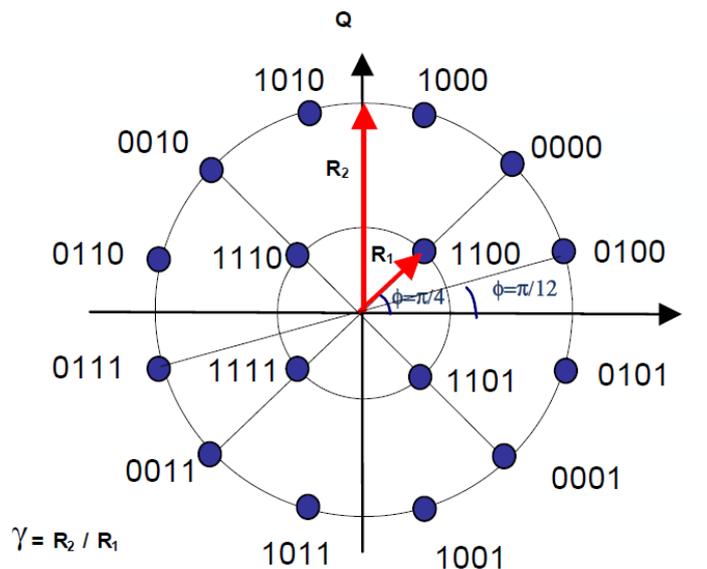


Рисунок 4.2 – Диаграмма «созвездие» 16-APSK модуляции

### *Задание на практику*

Модуль должен реализовать логику работы 16-APSK модулятора в соответствии с таблицей. При этом значение выходного символа «-1» соответствует минимальному знаковому числу, которое можно подать на вход 14-битного ЦАП, а «1» - максимальному знаковому числу.

Порты разрабатываемого модуля:

Входы:

clk – тактовая частота;

rst – сигнал сброса;

en – вход разрешения начала работы;

input\_bits [4:0] – входные биты.

Выходы:

output\_symbol\_real [13:0] – синфазная составляющая модулированного сигнала;

output\_symbol\_imag [13:0] – квадратурная составляющая модулированного сигнала.

Ход работы:

- 1) Самостоятельно реализуйте модуль в отдельном проекте.
- 2) Проведите временную симуляцию модуля.
- 3) Сохраните код модуля и результаты симуляции.

## 5. СОЗДАНИЕ OFDM МОДУЛЯТОРА. ОБЪЕДИНЕНИЕ МОДУЛЕЙ В ИТОГОВЫЙ ПРОЕКТ

Ортогональное частотное мультиплексирование данных (OFDM — Orthogonal Frequency Division Multiplexing) представляет собой специальный случай одновременной передачи потока цифровых данных по многим частотным каналам. Одним из привлекательных свойств данной технологии считается относительно высокая устойчивость по отношению к частотно-селективным замираниям и узкополосным помехам. Главный принцип OFDM заключается в том, чтобы разделить основной поток бит на ряд параллельных подпотоков с низкой скоростью передачи и затем использовать их для модуляции нескольких несущих (поднесущих). При этом, вообще говоря, к каждой из поднесущих может быть применена любая техника модуляции.

При формировании OFDM-сигнала необходимо обеспечить ортогональность поднесущих. Поэтому сначала, исходя из характера входных данных, определяются требуемый частотный спектр и необходимая схема модуляции. Каждая поднесущая связывается со своим подпотоком данных. Амплитуда и фаза поднесущей вычисляются на основе выбранной схемы модуляции. В качестве такой схемы может выступать бинарная фазовая манипуляция (Binary Phase Shift Keying, BPSK), квадратурная фазовая манипуляция (Quadrature Phase Shift Keying, QPSK) или квадратурная амплитудная модуляция (Quadrature Amplitude Modulation, QAM). Затем с помощью обратного преобразования Фурье (IFT) амплитуда как функция частоты преобразуется в функцию от времени (преимущественно используется вариант обратного быстрого преобразования Фурье - IFFT). Принимающая аппаратура с помощью прямого быстрого преобразования Фурье (FFT) преобразует амплитуду сигналов как функцию от времени в функцию от частоты, формируя при этом набор ортогональных синусоид.

За счёт применения преобразований Фурье частотный диапазон делится на поднесущие, спектры которых перекрываются (рисунок 5.1б), но остаются ортогональными.

При распространении радиочастотного сигнала в радиоканале происходит рассеивание и переотражение сигнала от неоднородностей канала. В системах связи для борьбы с эффектами, вносимыми многолучевым каналом распространения, применяется защитный интервал, называемый циклическим префиксом.

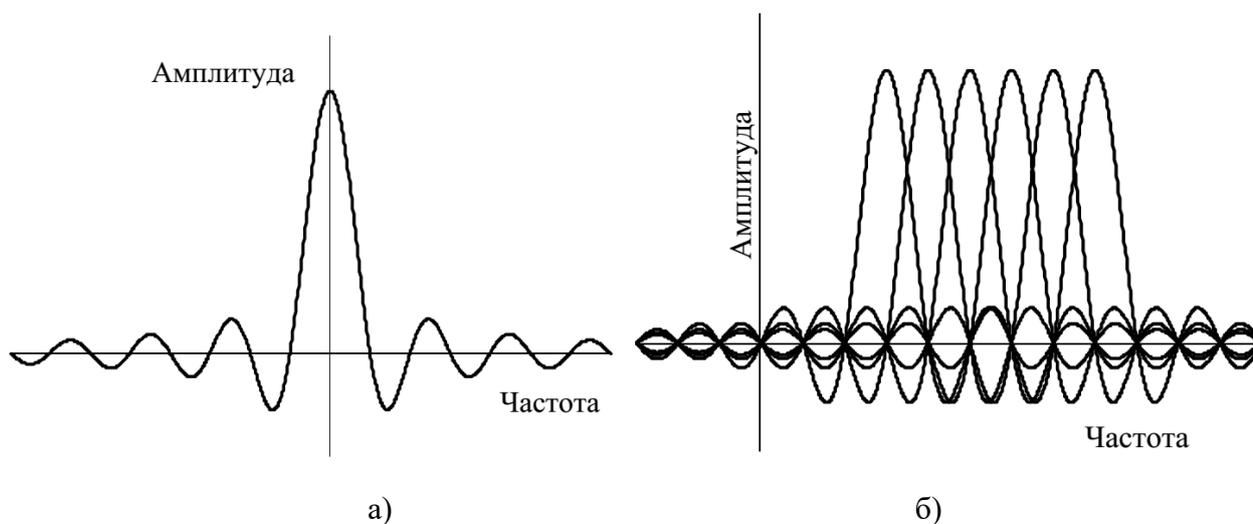


Рисунок 5.1 – Частотное представление OFDM символа

Он представляет собой копию отсчетов сигнала из окончания OFDM символа. Длина циклического префикса выбирается в соответствии с максимально возможной задержкой сигнала в канале РРВ. Циклический префикс является избыточной информацией и в этом

смысле снижает полезную (информационную) скорость передачи, но именно он служит защитой от возникновения межсимвольной интерференции, появляющейся из-за нарушения ортогональности между символами. Указанная избыточная информация добавляется к передаваемому символу в передатчике и отбрасывается при приеме символа в приемнике. На рисунке 2 приведена временная структура OFDM символа.

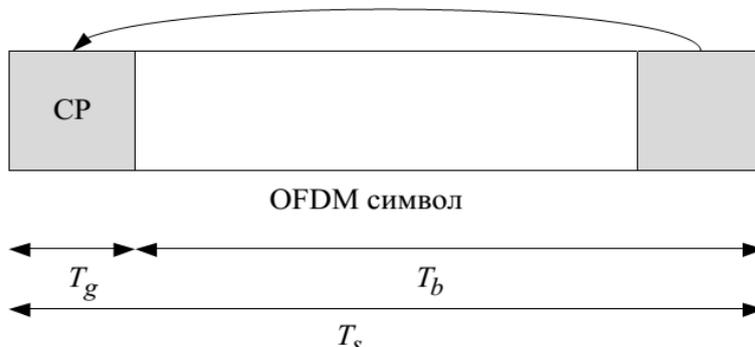


Рисунок 5.2 – Временная структура OFDM символа

Обратимся теперь к вопросу о практической реализации ансамбля ортогональных многочастотных сигналов. На рисунке 5.3 схематично показана архитектура типового приёмника и передатчика OFDM.

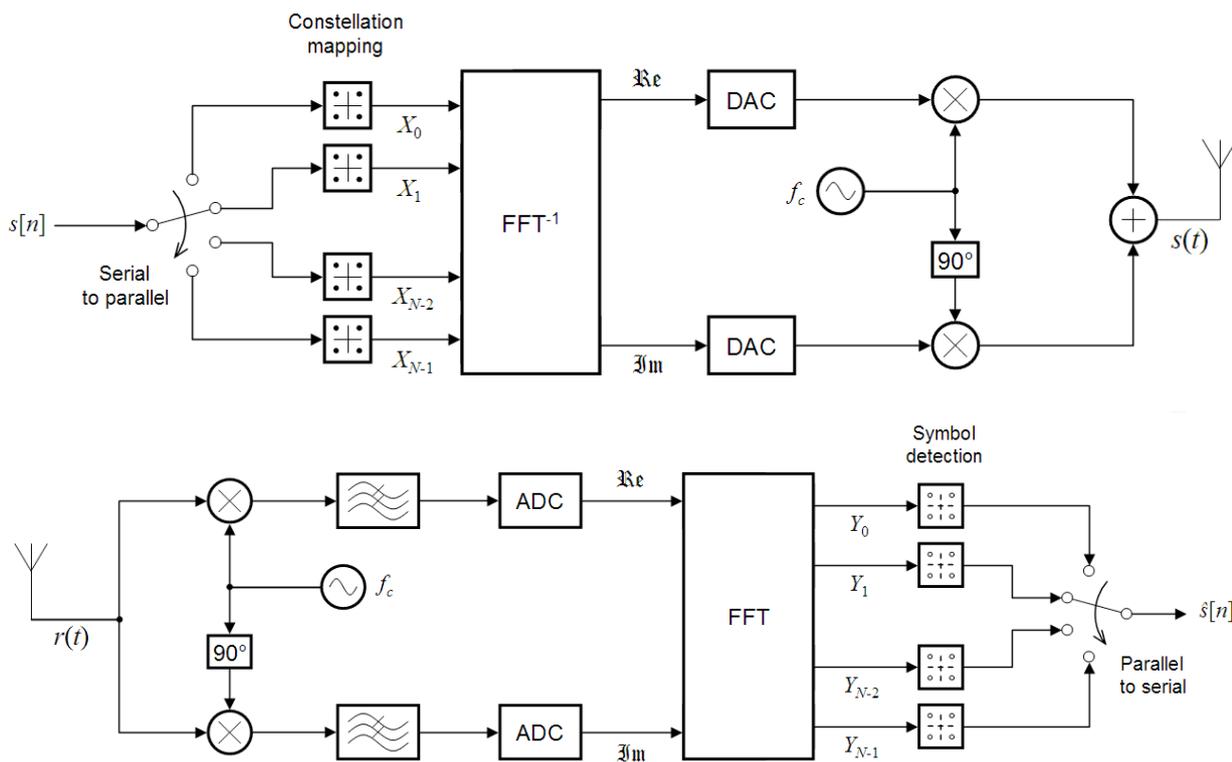


Рисунок 5.3 – Структурная схема передатчика и приёмника OFDM

Технической реализации OFDM не было долгое время, поскольку решение задачи аналоговыми методами весьма проблематично. С появлением быстрых вычислительных систем задача была реализована с помощью цифровых методов обработки сигналов. В основе подхода лежит преобразование Фурье, а точнее алгоритм быстрого преобразования Фурье. Синтетическим методом создаётся спектр сигнала, из которого обратным быстрым преобразованием Фурье (IFFT) получается аналоговый сигнал. Спектр такого сигнала уже состоит из ортогональных поднесущих, этот факт получается по определению преобразования

Фурье. Об этом также упоминалось выше. Непосредственное формирование сигнала после цифрового синтеза, который затем передаётся в антенну для излучения, происходит аналогично схеме QAM модуляции. В отдельности формируются квадратурные сигналы как мнимая и реальная часть синтезируемого сложного сигнала, а затем происходит его «сборка» и передача в антенну. Подробности функционирования можно прочитать в дополнительной литературе.

#### *Задание на практику*

В рамках данной лабораторной работы необходимо реализовать модуль OFDM модулятора, на вход которого следует подать биты идущие со выхода квадратурного модулятора.

- 1) Откройте Quartus II. Пуск => Все Программы => Altera => Quartus II.
- 2) Откройте шаблон проекта, расположение выясните у преподавателя. File => Open => Выберите тип файла «Project Files» => SoCKit\_ADA.qpf. Откройте модуль верхнего уровня SoCKit\_ADA.v, это файл описания, сгенерированный на языке Verilog, включающий список портов – кнопок, переключателей, светодиодов, портов АЦП-ЦАП, клоков.

Таблица 5.1 – Описание некоторых портов, используемых в проекте

Имя порта	Описание
OSC_50_B5B	Тактовая частота 50 МГц
LED [3:0]	Светодиоды
KEY [3:0]	Кнопки
SW [3:0]	Переключатели
DAC_DA [13:0]	ЦАП 1
DAC_DB [13:0]	ЦАП 2

- 3) На основе предыдущих лабораторных работ реализовать модуль нижнего уровня для QAM модулятора.

Порты разрабатываемого модуля:

#### Входы:

clk – тактовая частота, подключите в модуле верхнего уровня к шине OSC\_50\_B5B;  
 rst – сигнал сброса, подключите к одному из переключателей;  
 en – вход разрешения начала работы, подключите к одному из переключателей;  
 input\_bits [n:0] – входные биты. Размерность входных данных выбрать самостоятельно.

#### Выходы:

output\_simbol\_real [13:0] – синфазная составляющая модулированного сигнала;  
 output\_simbol\_imag [13:0] – квадратурная составляющая модулированного сигнала;

**ВАЖНО!** Нужно учесть, что старший бит на входе FFT берется как знаковый, поэтому значения должны меняться от -8192 до 8191 или от 10000000000000 до 01111111111111.

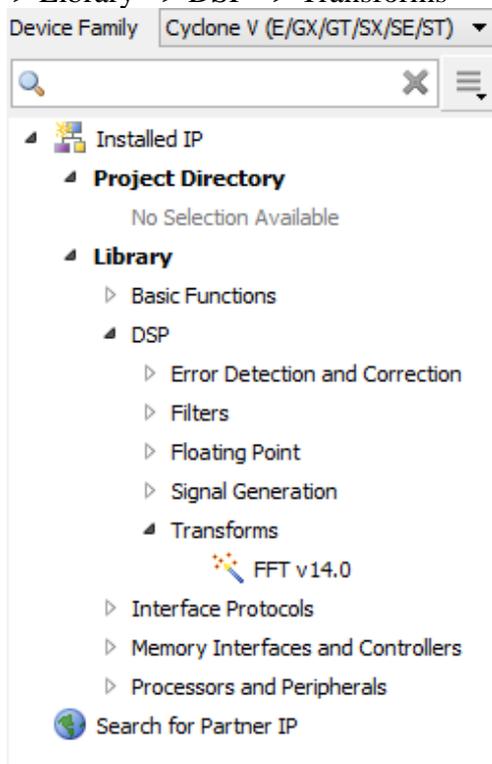
Таблица 5.2 – Варианты заданий:

Вариант	Модуляция
1	BPSK
2	QAM-4
3	QAM-16
4	8-APSK

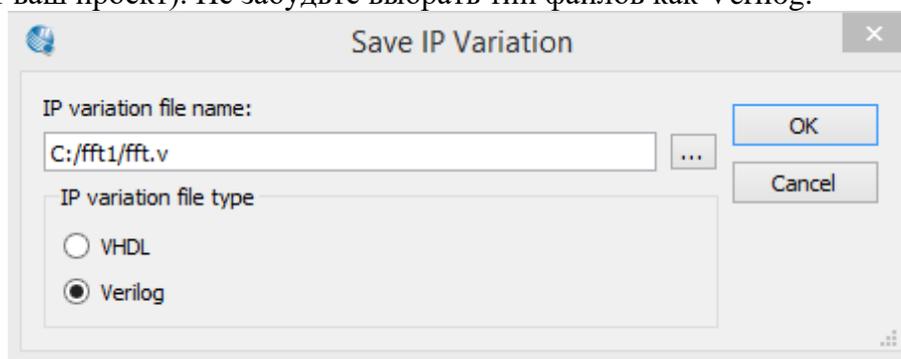
Вариант задания уточнить у преподавателя!

4) Сгенерировать модуль, выполняющий обратное преобразование Фурье из библиотеки IP-ядер программы Quartus. Данный модуль генерируется в следующей последовательности:

1. В начале в левом крае программы из каталога библиотеки выберете модуль FFT v14.0 через путь Installed IP => Library => DSP => Transforms



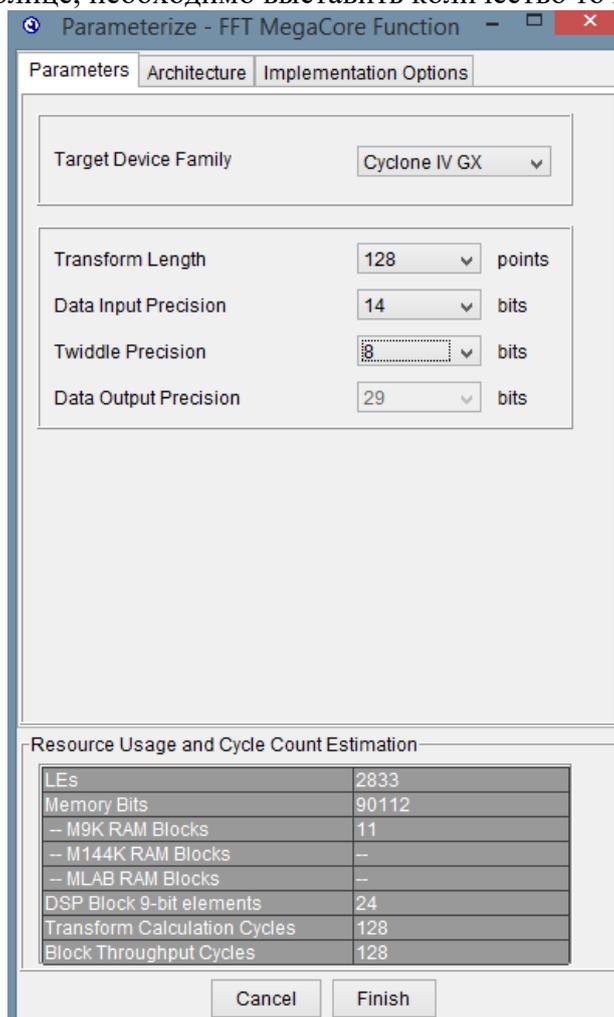
2. В открывшемся окне введите название модуля и расположение (Желательно там же, где лежит ваш проект). Не забудьте выбрать тип файлов как Verilog.



3. После этого должно открыться меню для установки параметров при генерации модуля. Последовательно установим параметры в пунктах Parameterize и Set Up Simulation.



4. Здесь необходимо установить основные параметры: количество точек преобразования, размерность данных, точность данных. В соответствии с вариантом, приведенные ниже в таблице, необходимо выставить количество точек преобразования.

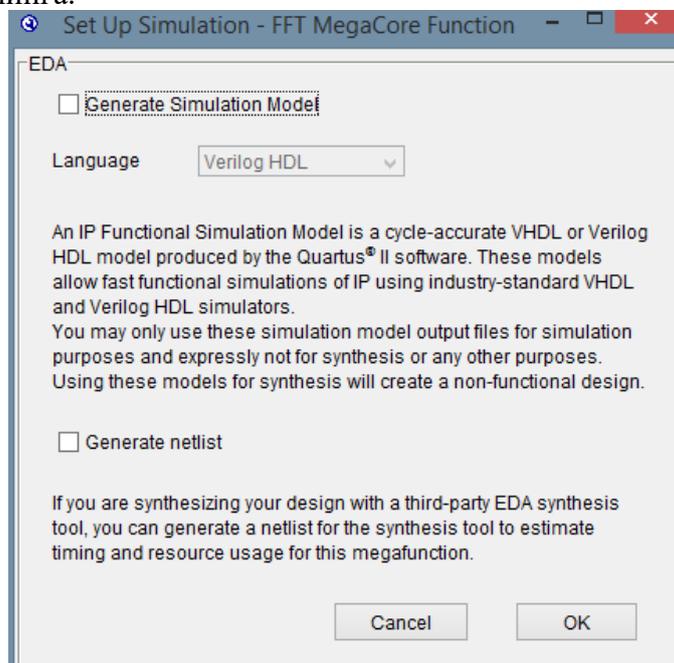


Для всех случаев нужно выставить семейство кристалла (Target Device Family) как Cyclone V, и битовый поток (Data flow) как Streaming, размерность входных данных (Data Input Precision) и точность преобразования (Twiddle Precision) - 14 бит. Остальные параметры оставить без изменений.

Таблица 5.3 – Варианты заданий:

Варианты	Количество точек преобразования (Transform Length)
1	64
2	128
3	256
4	512

5. В окне Set Up Simulation нужно убрать обе галочки, так как нам не нужна модель для симуляции и тайминга.



6. После выставления всех параметров осталось только сгенерировать модуль нажав в меню на Generate

7. В сгенерированном модуле верхнего уровня который к примеру называется, как fft.v (ЕСЛИ в пункте 2 вы дали имя fft.v) будут иметь следующие входы и выходы:

```
input      clk      – вход для тактовой частоты(подключите к OSC_50_B5B);
input      reset_n – вход для сброса (сбрасывается при нуле);
input      inverse - 1 – если IFFT, 0 – если FFT (так как у нас модулятор, то нам нужен IFFT);
input      sink_valid – вход разрешения на обработку (ниже будет указание как им управлять);
input      sink_sop – вход указывающий старт новой входной пачки для преобразования (ниже будет указание как им управлять);
input      sink_eop – вход указывающий конец входной пачки для преобразования (ниже будет указание как им управлять);
input [13:0] sink_real – вход для реальной части с выхода квадратурного модулятора;
```

input [13:0] sink\_imag – вход для мнимой части с выхода квадратурного модулятора;

input [1:0] sink\_error – вход для случаев возникновения ошибок в входном сигнале (для простоты реализации можно приравнять к значению 2'b00)

input source\_ready – вход указывающий, что на выходе готовый принять результат обработки (так как у нас выход подключен к осциллографу, то можно привязать значение 1'b1, т.е. всегда готов);

output sink\_ready - выход указывающий, что модуль готов принять данные;

output [1:0] source\_error - выход показывающий о произошедших ошибках (не обязательно указывать в модуле верхнего уровня);

output source\_sop - выход показывающий начало пачки обработанного сигнала (не обязательно указывать в модуле верхнего уровня);

output source\_eop - выход показывающий конец пачки обработанного сигнала (не обязательно указывать в модуле верхнего уровня);

output source\_valid – выход показывающий, что данные доступны (не обязательно указывать в модуле верхнего уровня);

output [5:0] source\_exp – экспонента выходных данных;

output [13:0] source\_real – мантисса реальной части выходных данных (необходимо подключить к DAC\_DA);

output [13:0] source\_imag – мантисса мнимой части выходных данных (необходимо подключить к DAC\_DB);

5) Последним шагом остается настроить сигналы на входе *sink\_valid*, *sink\_eop* и *sink\_sop*.

Вход *sink\_valid* соединяется ко входу одnorазрядного регистра, который сбрасывается в ноль по входу *reset\_n* и выставляется в единицу по входу *en* с квадратурного модулятора. В качестве тактовой частоты выставить OSC\_50\_B5B.

Входы *sink\_sop* и *sink\_eop* управляются следующим образом.

```
assign sink_sop = (cnt == 0) ? 1'b1 : 1'b0;
```

```
assign sink_eop = (cnt == <максимальное значение регистра>) ? 1'b1 : 1'b0;
```

где *cnt* – синхронный счетчик, с тактовой частотой с OSC\_50\_B5B, со сбросом *reset\_n* и разрешением на запись с выражения:  $(sink\_valid == 1'b1) \& (sink\_ready == 1'b1)$ . Разрядность регистра *cnt* выбрать в зависимости от количества точек преобразования Фурье (для 64 соответственно 6-разрядный).

б) Согласовать все модули между собой. Пронаблюдать сигналы с помощью осциллографа. Подготовить отчет о проделанной работе. Сделать выводы.

## СПИСОК ЛИТЕРАТУРЫ

1. Каршенбойм, И. Г. Краткий курс HDL. Часть 1 [Электронный ресурс] / И. Г. Каршенбойм // Компоненты и Технологии. – 2008. – № 3. – Режим доступа: [http://iosifk.narod.ru/hdl\\_coding/verilog.htm](http://iosifk.narod.ru/hdl_coding/verilog.htm) (дата обращения: 12.01.2022).
2. Скляр Б. Цифровая связь: Теоретические основы и практическое применение. – Издательский дом Вильямс, 2004.
3. Дворкович А., Дворкович В. Цифровые видеоинформационные системы (теория и практика). – Litres, 2021.
4. Сергиенко А. Б. Цифровая обработка сигналов. – БХВ-Петербург, 2011.
5. Тарасов, И. Е. Программируемые логические схемы и их применение в схемотехнических решениях : учеб. пособие / И. Е. Тарасов, Е. Ф. Певцов / ФБГОУ ВПО «Московский государственный технический университет радиотехники, электроники и автоматики». – М., 2012. – 184 с.
6. Quartus II Handbook. Vol. 3. Verification [Electronic resource]. – URL: [http://www.altera.com/literature/hb/qts/quartusii\\_handbook.pdf](http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf) (accessed 12.01.2022).
7. Кондратенко, Ю. П. Verilog-HDL для моделирования и синтеза цифровых электронных схем / Ю. П. Кондратенко, В. В. Мохор, С. А. Сидоренко. – Новосибирск : НГТУ, 2002. – 221 с.
8. Наваби, З. Проектирование встраиваемых систем на ПЛИС / З. Наваби. – М. : ДМК Пресс, 2016. – 464 с.