МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЕЛЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. И.А.БУНИНА»

И. В. Пешков

ЛАБОРАТОРНЫЕ И ПРАКТИЧЕСКИЕ ЗАДАНИЯ ПО КУРСУ «ЦИФРОВЫЕ РАДИОПЕРЕДАЮЩИЕ УСТРОЙСТВА»

Методические указания

Елец – 2020

УДК 621.396.2
ББК 32.884.1
П 23

Печатается по решению редакционно-издательского совета
Елецкого государственного университета имени И.А. Бунина
от 28. 01. 2020 г., протокол № 1

Рецензенты:
Рощупкин С.А., к. ф.-м.н., доцент кафедры математического моделирования и компьютерных технологий
Елецкого государственного университета им. И.А. Бунина

Нечаев Ю.Б., д-р ф.-м.н., проф. каф. информационных систем
Воронежского государственного университета

И.В. Пешков

П 23 Лабораторные и практические задания по курсу «Цифровые радиопередающие устройства»: учебное пособие. – Елец: Елецкий государственный университет им. И.А. Бунина, 2020. – 43 с.
ISBN 978-5-00151-154-0

В методических указаниях представлены материалы практических и лабораторных заданий по дисциплине «Радиопередающие устройства». По каждой теме даются основные теоретические положения.

Методические указания предназначены для студентов направлений подготовки бакалавров: «Радиотехника».

Предназначено для студентов вышеуказанных направлений подготовки очной и заочной форм обучения.

УДК 621.396.2
ББК 32.884.1

ISBN 978-5-00151-154-0

ОГЛАВЛЕНИЕ

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Пропускная способность систем связи

В общем виде структура передачи-приёма цифровой информации по радиоканалу выглядит следующим образом [1].

Рис. 1. Структура передачи-приёма сигналов

Передатчик принимает последовательность бит (0 или 1) и создаёт физический сигнал или радиоволну (т.е. изменяющееся во времени напряжение), которая переносится по радиоканалу [1].

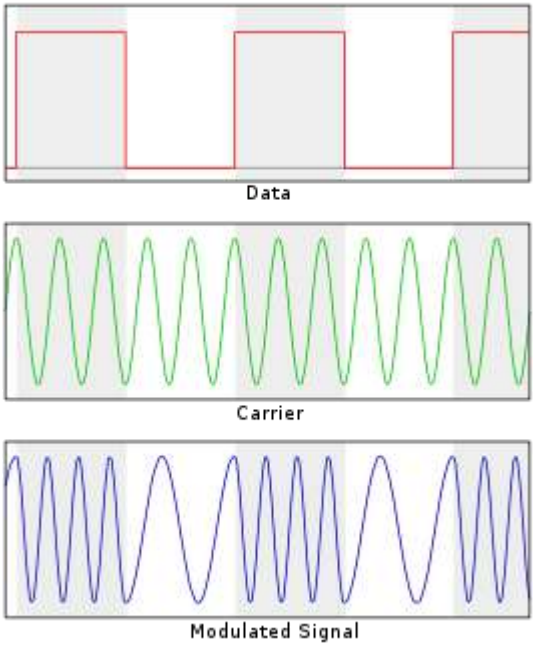


Рис. 2. Формирование радиоволны из бит данных

Задание:

Что такое биты?

• Бит – это основная единица информации, используемая в современных компьютерах и системах связи.

• Бит – это переменная, которая может принимать только два возможных значения или «состояния», обычно обозначаемых 0 или 1.

• Интуитивно этот бит можно рассматривать как ответ на вопрос «да / нет».

• Более сложная информация может быть отправлена последовательностями бит.

Представление битов

Физически биты могут быть представлены как два различных состояния физической переменной [1].

Примеры:

• напряжение (1 = высокое / 0 = низкое)

• текущий (1 = положительный / 0 = отрицательный)

• свет (1 = включен / 0 = выключен)

Рис. 3. Представление бит как сигналов

Передатчик отправляет сигнал, представляющий последовательность битов, приемнику по каналу [1].

Примеры:

• Форма волны напряжения или тока может передаваться по проводу.

• Световой сигнал может быть отправлен по оптоволоконному каналу (Интернет) или по радио.

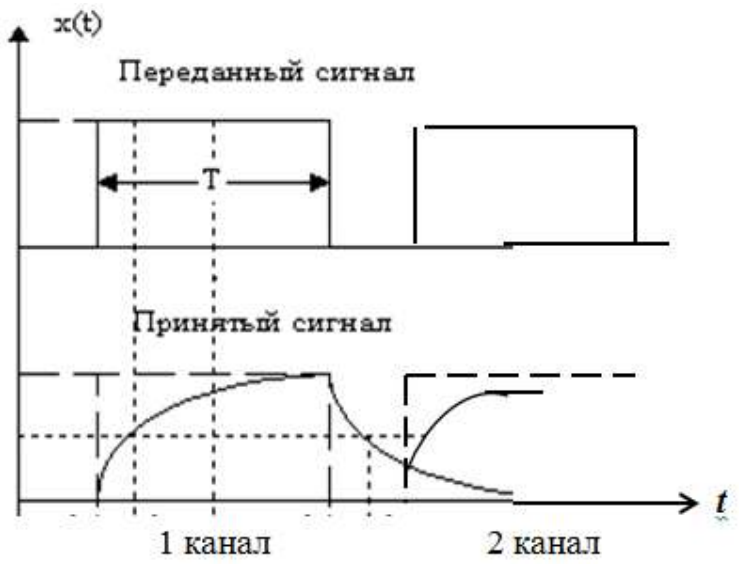


Рис. 4. Искажения сигнала в канале связи

Вопросы. Основы коммуникационных систем

Рассмотрим передачу двоичной последовательности, например «10010110». Для увеличения скорости передачи данных битовое время должно быть _____?

- Короче
- Длиннее

Какова основная роль передатчика в системе связи?

- Чтобы преобразовать информацию в физическую форму волны
- Для восстановления информации из физического сигнала
- Обеспечить среду, по которой распространяется физическая форма волны

Кодирование информации битами

В таблице ASCII ниже приведены коды ASCII для общих буквенно-цифровых символов и символов, перечисленных от MSB до LSB. Какая битовая последовательность кодирует сообщение «Hi»? Предположим, что мы передаем коды каждого символа последовательно, начиная с LSB.

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	i	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Введите битовую последовательность как последовательность единиц и нулей без пробелов между ними, например 1010 ....

Ответ: 0001001010010110

Каково десятичное значение битовой последовательности «1000»? Предположим, что MSB указан первым.

Ответ: 8

Непрерывные и дискретные временные формы сигналов

Представление битовых последовательностей в виде сигналов

Последовательность битов можно закодировать, изменяя значение физической переменной с течением времени.

• Каждый бит кодируется путем сохранения постоянного состояния в течение определенного периода времени,

известное как бит времени.

• Чем короче битовое время, тем быстрее мы можем передавать информацию (биты)

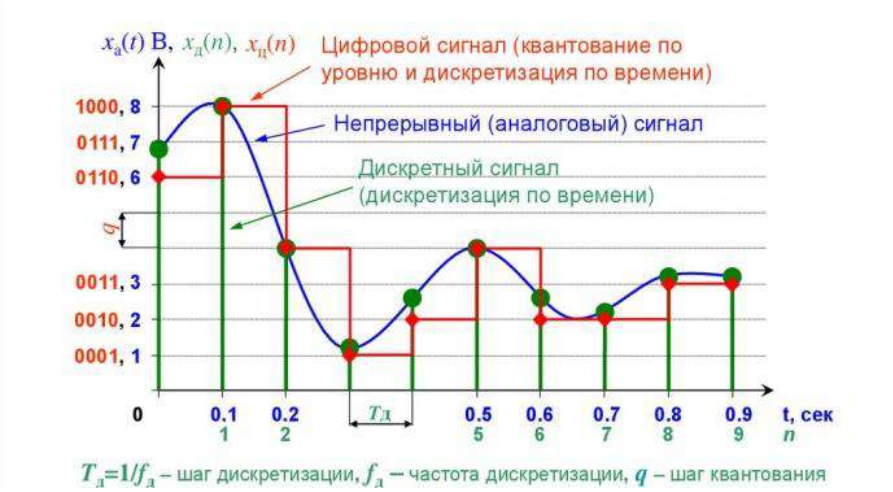


Рис. 5. Непрерывные и дискретные сигналы времени

Сигнал непрерывного времени (CT) имеет известное значение для всех точек временного интервала.

Сигнал дискретного времени (DT) имеет известное значение только в дискретном (прерывистом) наборе моментов времени.

Выборка: непрерывный сигнал в дискретный

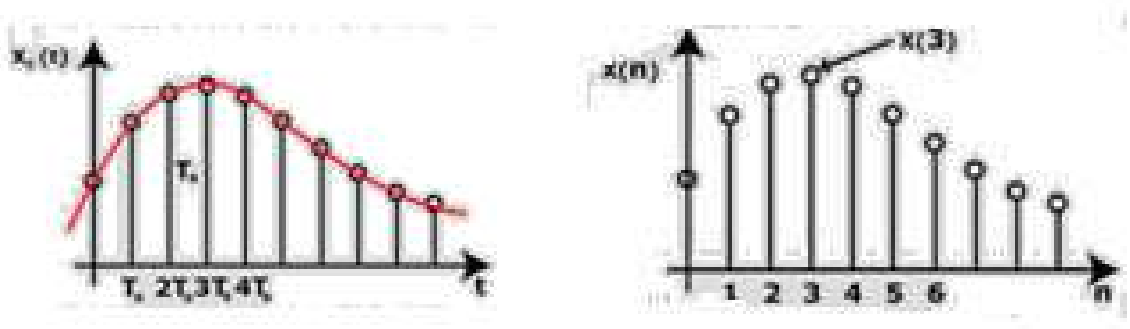
• Получите дискретную временную форму сигнала путем дискретизации непрерывного временного сигнала x c (t) через равные промежутки времени.

Ts = период выборки.

• Проиндексируйте каждую выборку по целочисленному номеру выборки n.

• n-й образец соответствует форма волны в момент временит = nTs

Пример: x(n) =xc(nTs)



От дискретного к непрерывному времени

Учитывая выборку x (n), мы можем получить непрерывную временную форму волны xh (t), удерживая форму волны в x (n) между временами nTs и (n + 1) Ts

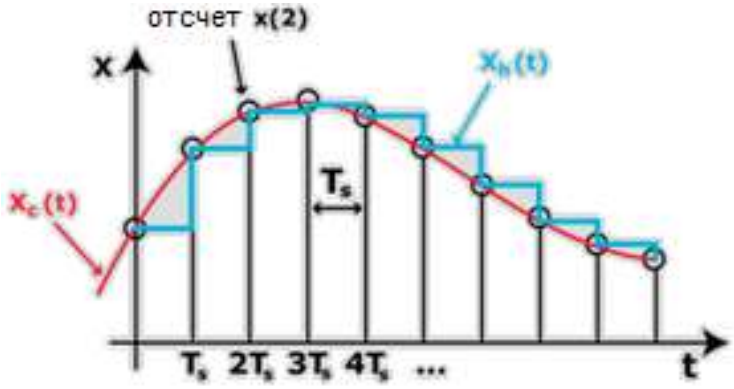


Рис. 6. Непрерывные и дискретные сигналы времени

Период выборки в зависимости от частоты

Ts = период выборки (временной интервал между выборками)

Типичная единица измерения: секунды (с, с)

Fs = частота или частота дискретизации (количество выборок за фиксированный период времени)

Типичная единица измерения: Герц (Гц, выборок в секунду) $F_s = \frac{1}{T_s}$

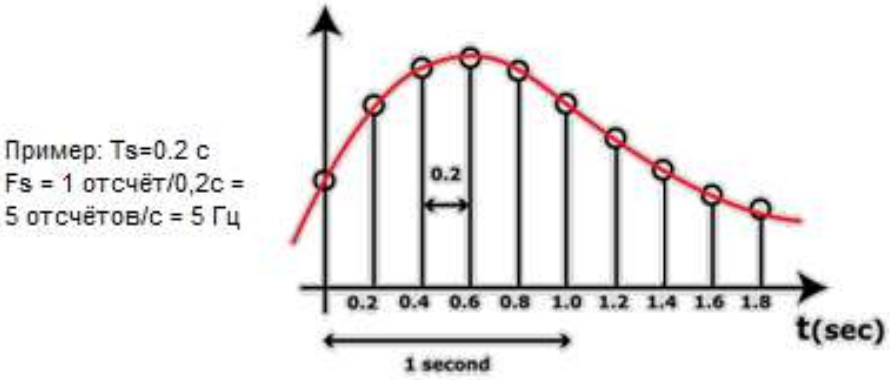


Рис. 7. Процесс дискретизации

Количество выборок

Выборка сигнала длиной Tw с периодом выборки Ts приводит к N отсчётам, где

$$N = \frac{T_w}{T_s} = T_w \cdot F_s$$

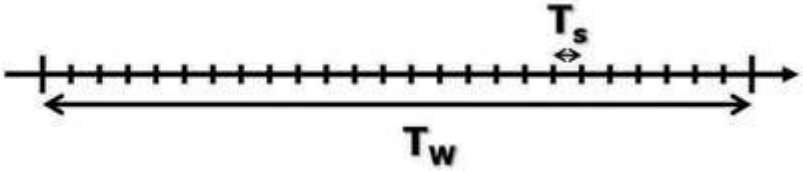


Рис. 8. Выборка сигнала

Компромисс: более высокая частота дискретизации

• Хорошо: меньше информации теряется благодаря меньшему времени между выборками.

• Плохо: требуется больше места для хранения, поскольку за заданный промежуток времени требуется больше образцов.

Предположим, мы дискретизируем сигнал на частоте Fs. Если мы соберем 1500 отсчётов или семплов за 5 секунд, чему равно F в Гц?

Ответ 300

Компакт-диски записывают два канала (левый и правый) музыки с частотой дискретизации Fs = 44,1 кГц. Если каждый отсчёт или сэмпл закодирован 16 битами, а один байт равен 8 битам, сколько байтов требуется для хранения одной минуты музыки?

Ответ 10584000

Битовые сигналы дискретного времени

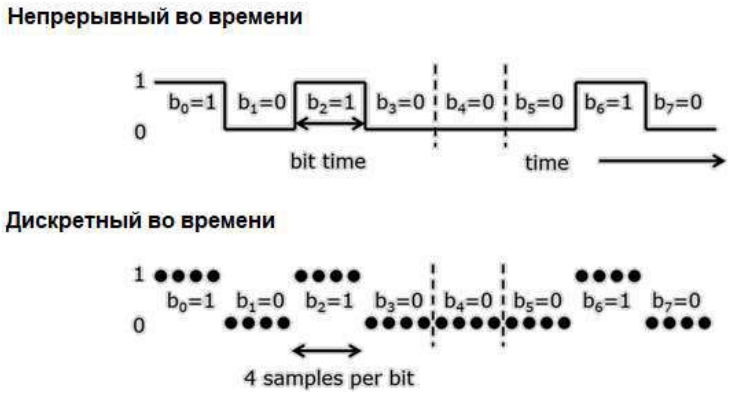


Рис. 9. Битовые сигналы

Битовое время измеряет время, необходимое для отправки одного бита.

$$\text{Длит. бита} = \text{SPB} \cdot T_s = \frac{\text{SPB}}{F_s}$$

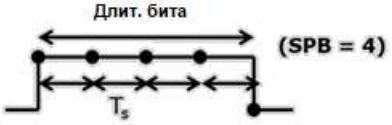
Битрейт измеряет количество бит, которое мы можем отправить за заданную единицу времени.

$$\textbf{Битрейт} = \frac{1}{\text{Длит. бита}} = \frac{1}{\text{SPB} \cdot T_s} = \frac{F_s}{\text{SPB}}$$

Обычно мы хотим:

– битрейт должен быть большим

– время бит быть маленьким



Пример расчета скорости передачи данных

Частота дискретизации

Fs = 1МГц = 1 Мега Герц =1,000,000 отсчётов / с $=10^6$ отсчётов / с

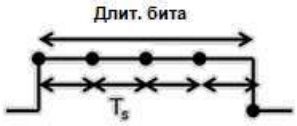
Если используется 4 отсчёта за выборку (SPB = 4), тогда

Ts = $(Fs)^{-1} = 10^{-6}$ с =1 мкс микросекунд

Длительность бита = SPB Ts = 4 мкс

Битовая скорость $= \frac{F_s}{SPB}$

$= \frac{1,000,000}{4}$ Hz = 250kHz



Рассмотрим систему, которая использует 8-битные коды ASCII для кодирования букв. Сколько времени потребуется, чтобы передать последовательность битов с кодировкой «Hello» (без кавычек), если мы используем время передачи данных в 5 отсчетов на бит и передаем отсчеты с частотой 1 МГц?

Ответ: 200.

Рассмотрим систему, которая использует 8-битные коды ASCII для кодирования букв. Сколько времени потребуется для передачи битовой последовательности с кодировкой «Good Morning» (без кавычек), если мы используем битовое время 5 отсчетов на бит и передаем отсчеты с частотой 1 МГц?

Рассмотрим систему связи, в которой передатчик использует 0 В для представления бита «0» и 1 В для представления бита «1». Пример формы передаваемого сигнала приведен на следующем рисунке.

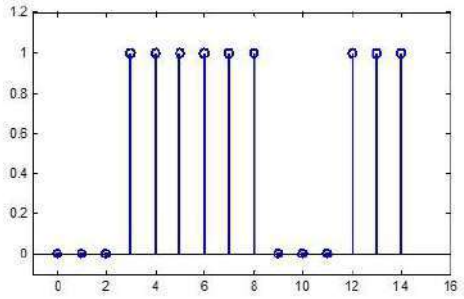


Рис. 10. Пример передаваемого сигнала

Предположим, что первый бит начинается с отсчета 0, каково максимально возможное время передачи (в SPB), используемое при передаче?

Ответ: 3.

ЛАБОРАТОРНАЯ РАБОТА

Передача данных в канале с ограниченной полосой

Моделирование аналоговых, импульсных сигналов и цифровых сигналов.

Цель работы: Ознакомиться с возможностями программы Microcap 9.0. Научиться моделировать аналоговые и цифровые сигналы [2].

1. Моделирование аналогового сигнала.

Для моделирования аналогового сигнала в программе Microcap предназначен компонент Sine Source (Component → Analog Primitives → Waveform Sources → Sine Source). Компонент Sine Source имеет следующие модели:

1MHZ – синусоидальный сигнал с амплитудой 1В и частотой 1МГц.

3PHASEA – синусоидальный сигнал с амплитудой 169.7В и частотой 60Гц, предназначенный для промышленных сетей.

3PHASEB – синусоидальный сигнал с амплитудой 169.7В, частотой 60Гц и начальной фазой 120° предназначенный для промышленных сетей.

3PHASEC – синусоидальный сигнал с амплитудой 169.7В, частотой 60Гц и начальной фазой 240° предназначенный для промышленных сетей.

60HZ – синусоидальный сигнал с амплитудой 169.7В и частотой 60Гц, предназначенный для промышленных сетей.

GENERAL – синусоидальный сигнал с амплитудой 1В и частотой 10МГц. Каждая модель компонента Sine Source имеет следующие изменяемые параметры:

A – амплитуда сигнала.

DC – постоянная составляющая сигнала.

F – частота сигнала.

PH – начальная фаза сигнала.

RS – внутреннее сопротивление источника сигнала.

TAU – постоянная времени цепи.

Для получения модели аналогового синусоидального сигнала необходимо собрать схему представленную на рисунке 1. Параметры генератора представленного на рисунке: A=1m, DC=0, F=1MEG, PH=0, RS=1m, TAU=0.

Рис. 11. Аналоговый генератор

Для проведения моделирования необходимо выбрать пункт Analysis → Transient. На экране появится окно параметров временного анализа (рис. 12).

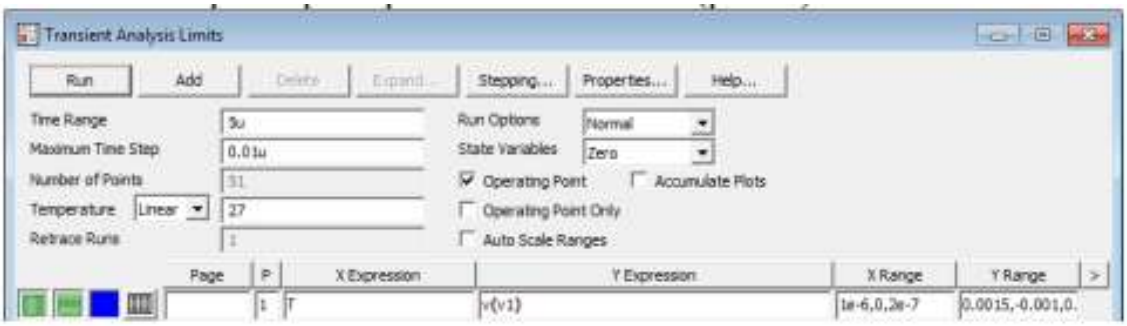


Рис. 12. Окно параметров для временного анализа

Основные параметры Transient:

Time range – временной интервал для моделирования.

Maximum Time Step – интервал дискретизации.

Temperature – температурный диапазон.

Необходимо при помощи Transient-анализа получить временные характеристики на выходе генератора.

2. Моделирование импульсного сигнала.

Для моделирования цифрового сигнала в программе Microcap предназначен компонент Pulse Source (Component → Analog Primitives → Waveform Sources → Pulse Source).

Компонент **Pulse Source** имеет следующие модели:

IMPUSLE – одиночный импульс длительностью 1нс и амплитудой 1 ГВ.

PULSE – трапецеидальные импульсы.

SAWTOOTH – импульсы пилообразного напряжения.

SQUARE – прямоугольные импульсы.

TRIANGLE – треугольные импульсы.

Компонент **Pulse Source** имеет следующие основные параметры: **P1, P2, P3, P4, P5** – временные задержки до определенных уровней сигнала (см. рис. 13) [2].

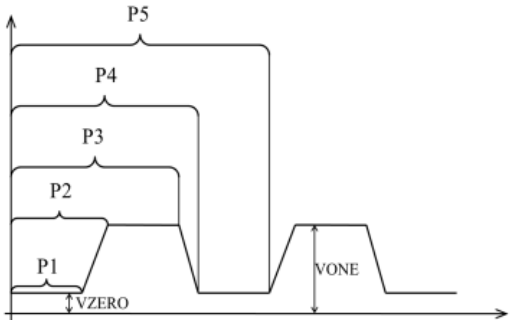


Рис. 13. Параметры компонента Pulse Source

VONE – уровень логической единицы.

VZERO – уровень логического нуля.

Для получения модели аналогового синусоидального сигнала необходимо собрать схему представленную на рисунке 14. Параметры генератора представленного на рисунке: P1=100n, P2=110n, P3=500n, P4=510n, P5=1u, VZERO=0, VONE=5.

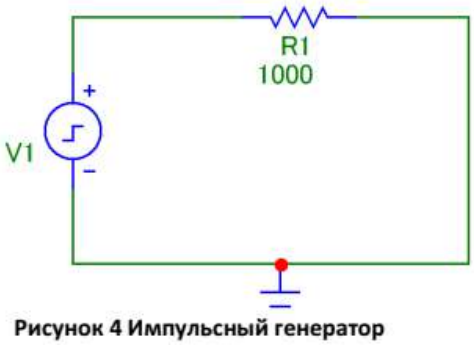


Рисунок 4 Импульсный генератор

Рис. 14. Импульсный генератор

Для проведения моделирования необходимо выбрать пункт Analysis → Transient.

Необходимо при помощи Transient-анализа получить временные характеристики на выходе генератора.

3. Моделирование цифрового сигнала.

Для моделирования аналогового сигнала в программе Microcap предназначены компоненты раздела **Stimulus Generators** (Component → Digital Primitives → Stimulus Generator).

В разделе Stimulus Generator содержатся следующие основные компоненты:

Dclock – генератор тактовых импульсов.

Stim1 – программируемый генератор цифровых сигналов, с одним выходом.

Stim2 – программируемый генератор цифровых сигналов, с двумя выходами.

Stim4 – программируемый генератор цифровых сигналов, с четырьмя выходами.

Stim8 – программируемый генератор цифровых сигналов, с восемью выходами.

Stim16 – программируемый генератор цифровых сигналов, с шестнадцатью выходами.

Программирование компонентов Stim ведется путем задания временных интервалов для уровней логического «0» и логической «1» [2].

Для создания сигнала переключения с логического нуля на логическую единицу (рис. 15) необходимо выполнить следующие действия [2].

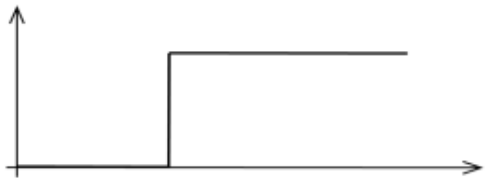


Рис. 15. Переключение с уровня лог. 0 на уровень лог. 1.

1. Создать новую схему.

2. Поставить элемент **Stim1** (Component → Digital Primitives → Stimulus Generator → Stim1).

3. В появившемся окне параметров найти поле ввода программного текста. В поле ввода программного текста стоит начальная директива .define _.

4. В поле ввода программного текста вписать следующий текст:

.define Switch

+0ns 0 (с момента 0нс от начала – уровень логического «0»)

+100ns 1 (с момента 100нс от начала – уровень логической «1»)

5. Закрыть окно параметров.

6. Запустить Transient-анализ. Временной диапазон поставить до 1мкс. По оси X поставить время (T). По оси Y – d(1) – цифровой сигнал на узле №1 [2].

Для создания одной цифровой единицы (рис. 2.) программный текст будет выглядеть следующим образом:

.define D1

+0ns 0 (с момента 0нс от начала – уровень логического «0»)

+100ns 1 (с момента 100нс от начала – уровень логической «1»)

+200ns 0 (с момента 200нс от начала – уровень логического «0»)

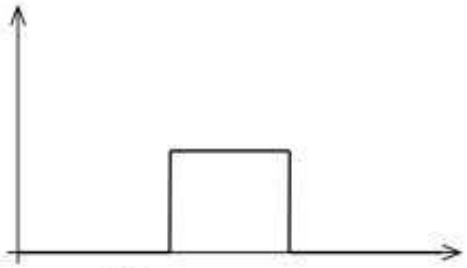


Рисунок 6 Логическая единица

Рис. 16. Логическая единица

Временные задержки можно задавать не только от 0, но и от последнего изменения.

.define D1

+0ns 0 (с момента 0нс от начала – уровень логического «0»)

++100ns 1 (через 100нс – уровень логической «1»)

++100ns 0 (еще через +100нс – уровень логического «0»)

Для создания нескольких повторяющихся цифровых последовательностей используются циклические конструкции [2].

Для создания последовательности 010101 (рис. 3) программный текст будет выглядеть следующим образом [2]:

.define D01

+0ns 0

+label=start (начало цикла)

++100ns 1 (через 100нс переключение в 1)

++100ns 0 (еще через 100нс переключение в 0)

++100ns goto start 3 times (повторить тело цикла еще три раза)

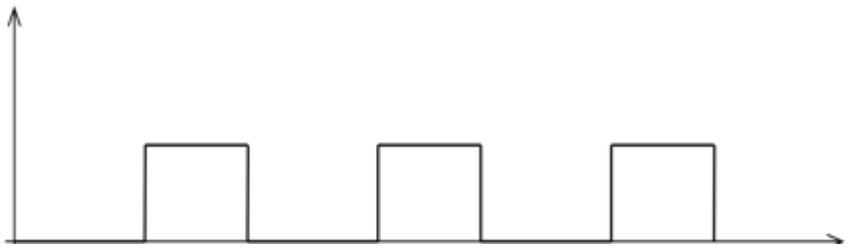


Рис. 17. Цифровая последовательность 010101

Если необходимо воздать бесконечную последовательность, то программный текст модифицируется следующим образом [2]:

.define D01

+0ns 0

+label=start (начало цикла)

++100ns 1 (через 100нс переключение в 1)

++100ns 0 (еще через 100нс переключение в 0)

++100ns goto start -1 times (повторить тело цикла бесконечное число раз)

1. Задания на лабораторную работу.

1. Создать модель генератора синусоидального сигнала представленного на рисунке 8.

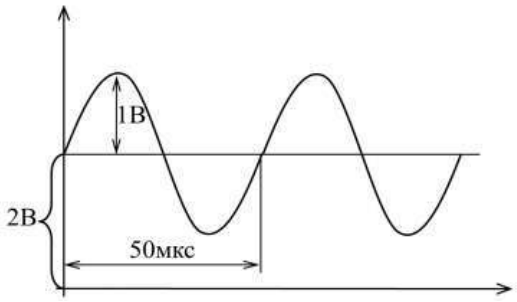


Рис. 18. Синусоидальный сигнал для моделирования

2. Создать два генератора цифровых сигналов, реализующих бесконечные цифровые последовательности, указанные на рис. 19.

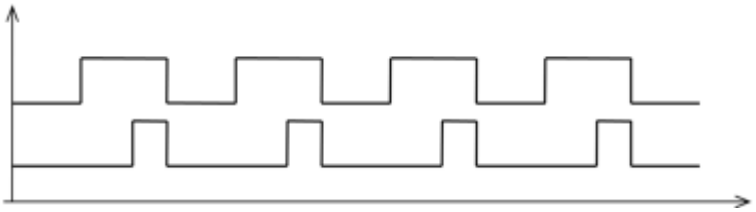


Рис. 19. Цифровые последовательности генераторов цифровых сигналов

2. Задания на лабораторную работу.

Цель: посмотреть, каким образом изменится форма прямоугольного импульса после прохождения фильтров.

Micro-Cap включает встроенную программу проектирования фильтров, которая непосредственно создает схемы фильтров. Разработчик активных фильтров поддерживает фильтры нижних и верхних частот, полосовой фильтр, режекторные фильтры и фильтры задержки, которые могут быть настроены с помощью ответа Баттерворта, Чебышева, Бесселя, эллиптического или обратного-чебышевского ответа. Созданный фильтр

может быть отправлен в новую схему, в текущую загруженную схему или может быть преобразован в макрокомпонент, к которому легко получить доступ через панель или меню «Компонент». Для начала проектирования необходимо запустить программу Microcap 9. Затем найти меню Design (Проектирование), в котором имеется пункт Passive Filters (Пассивные фильтры) [3].

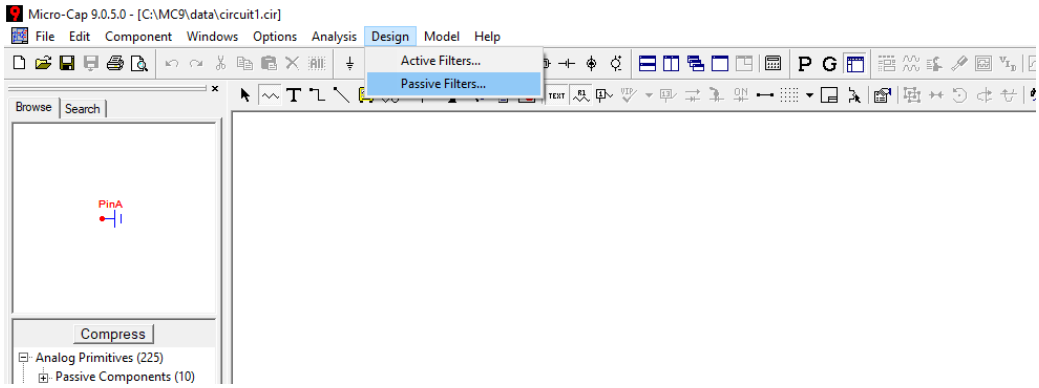


Рис. 20. Окно программы меню пассивные фильтры.

Затем появится окно настроек для проектирования пассивных аналоговых фильтров.

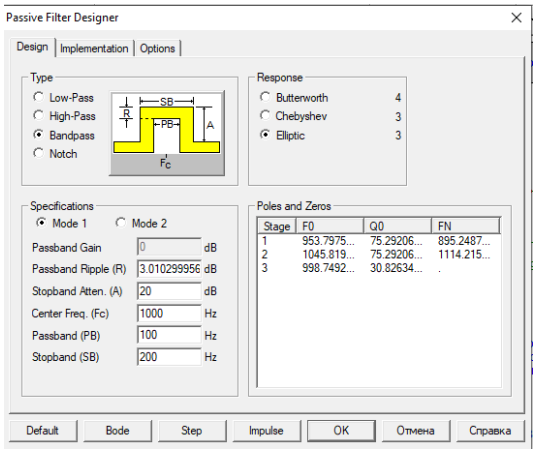


Рис. 21. Окно настроек

В данной вкладке имеется множество настроек. На данном этапе на понадобится только Type (т.е. Тип фильтра), Response (частотная характеристика) и Specifications (т.е. характеристики фильтра). Другие настройки нам здесь в этой лабораторной работе не понадобятся. В дальнейшем, конечно же, они полезны и весьма необходимы для тонкой настройки того или иного фильтра.

Тип фильтра позволяет выбрать один из следующих: Low pass (Фильтр нижних частот); High pass (Фильтр высоких частот); Bandpass (Полосовой фильтр); Notch (Фильтр-пробка); Delay (Устройство задержки).

Response (частотная характеристика) – позволяет выбирать математическую аппроксимацию идеального фильтра: Butterworth (Баттерворта); Chebyshev (Чебышева); Bessel (Бесселя); Elliptic (Элиптический); Inverse-Chebyshev (Обратный Чебышева). В нашем случае подойдет фильтр нижних частот Баттерворта.

Specifications (технические требования) – сюда вводят технические требования к фильтру. Есть два способа определить фильтр: Mode 1 (режим 1) и Mode 2 (режим 2). В Mode 1 определяются функциональные характеристики фильтра, подобные Passband Gain (усиление полосы пропускания в дБ), Passband (ширина полосы пропускания в Гц) (Fc), Stopband (ширина полосы подавления в Гц) (Fs) и Stopband Atten. (подавление полосы пропускания в дБ) (A). Здесь определяется то, что необходимо, и программа вычисляет число звеньев, требуемых для достижения этого, используя указанную аппроксимацию частотной характеристики. Режим Mode 2 позволяет непосредственно определить основные значения параметров проекта и число звеньев фильтра [4].

Для выполнения данной работы необходимо выбрать фильтр нижних частот, Response (частотная характеристика) – фильтр Баттерворта, Passband Ripple (пульсации полосы пропускания, т.к. АЧХ фильтра неидеальна) – 3 дБ (т.е. оставить по умолчанию), Stopband Atten. (подавление полосы пропускания в дБ) – 20 дБ (оставить по умолчанию), Passband (ширина полосы пропускания в Гц) равна величине обратной ширине импульса, т.е. $1/\tau_и$, Stopband (ширина полосы подавления в Гц) (Fs), несколько сотен герц.

В качестве источника необходимо использовать единичный прямоугольный импульс, подобный рис. 6. При этом длительность импульса составляет 100 мкс, время до начала импульса 0 с. Затем необходимо вывести форму импульса после прохождения спроектированного фильтра.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Сжатие информации. Кодирование Хаффмана

Классический алгоритм Хаффмана на входе получает таблицу частот встречаемости символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево) [4].

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение.
2. Выбираются два свободных узла дерева с наименьшими весами.
3. Создается их родитель с весом, равным их суммарному весу.
4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.
5. Одной дуге, выходящей из родителя, ставится в соответствие бит 1, другой – бит 0. Битовые значения ветвей, исходящих от корня, не зависят от весов потомков.
6. Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

Допустим, у нас есть следующая таблица частот [4]:

Символ	**А**	**Б**	**В**	**Г**	**Д**
Частота	15	7	6	6	5

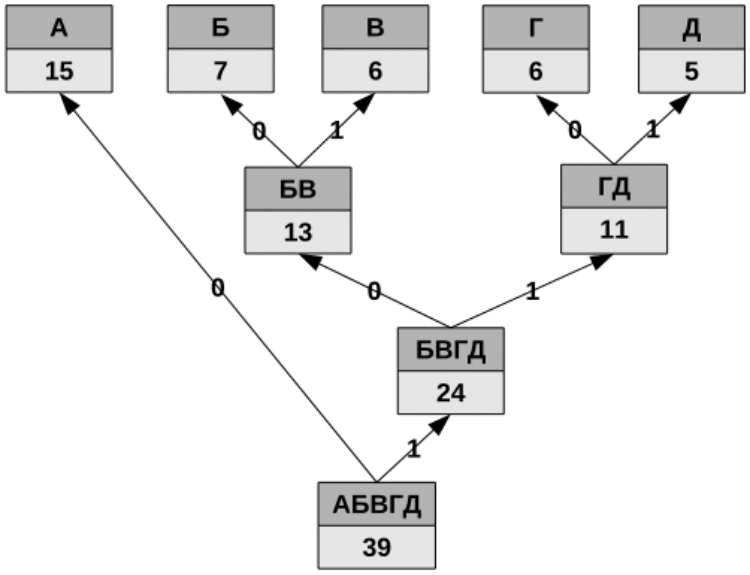
Для данной таблицы символов коды Хаффмана будут выглядеть следующим образом [4].

Символ	**А**	**Б**	**В**	**Г**	**Д**
Код	0	100	101	110	111

Кроме того, наиболее частый символ сообщения А закодирован наименьшим количеством бит, а наиболее редкий символ Д – наибольшим.

При этом общая длина сообщения, состоящего из приведённых в таблице символов, составит 87 бит (в среднем 2,2308 бита на символ). При использовании равномерного кодирования общая длина сообщения составила бы 117 бит (ровно 3 бита на символ). Заметим, что энтропия источника, независимым образом порождающего символы с указанными частотами, составляет ~2,1858 бита на символ, то есть избыточность

построенного для такого источника кода Хаффмана, понимаемая как отличие среднего числа бит на символ от энтропии, составляет менее 0,05 бит на символ [4].



Итого:

А	Б	В	Г	Д
0	100	101	110	111

Рис. 22. Дерево Хаффмана

1. Построить дерева Хаффмана, если символ А встречается 15 раз, Б – 7 раз, В – 6 раз, Г – 6 раз, Д – 5 раз.
2. Построить дерева Хаффмана, если символ А встречается 5 раз, Б – 7 раз, В – 6 раз, Г – 8 раз, Д – 1 раз.
3. Построить дерева Хаффмана, если символ А встречается 1 раз, Б – 17 раз, В – 6 раз, Г – 16 раз, Д – 5 раз.
4. Предположим, что приведенная ниже строка должна быть отправлена по сети.. BCAADDDCCACACAC

Каждый символ занимает 8 бит. В приведенной выше строке всего 15 символов. Таким образом, для отправки этой строки требуется всего бит.

Используя технику кодирования Хаффмана, мы можем сжать строку до меньшего размера. После кодирования размер

Символы от a до h имеют набор частот.

a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21

Для представления символов используется код Хаффмана. Какая последовательность символов соответствует следующему коду?

5. В кодировке ASCII (8 бит на символ) 13-символьная строка "go go gophers" занимает бит.

Строка "go go gophers" будет записано (закодировано численно) как …….. Хотя это не так легко читается людьми, это будет записано как следующий поток битов (пробелы не будут записаны, только нули и единицы)

Перевести данную запись "go go gophers" согласно алгоритму Хаффмана. Привести все получившие двоичные последовательности. Показать последовательность 0 и 1 после кодирования.

6. Рассмотрим следующий короткий текст**:** Eerie eyes seen near lake. Подсчитайте количество вхождений всех символов в тексте и построить дерево Хаффмана.

ЛАБОРАТОРНАЯ РАБОТА

Канальное кодирование. Множественный доступ

Схемы модуляции, которые мы рассмотрели, позволяют посылать один сигнал для передачи данных по проводу или беспроводному каналу. Однако экономия ресурсов играет важную роль в сетях передачи данных. Стоимость прокладки и обслуживания магистрали с высокой пропускной способностью и низкокачественной линии практически одна и та же (то есть львиная доля этой стоимости уходит на рытье траншей, а не на сам медный или оптоволоконный кабель). Вследствие этого были развиты схемы мультиплексирования для совместно использования линий многими сигналами [6].

FDM (Frequency Division Multiplexing, мультиплексирование с разделением частоты, частотное уплотнение) использует передачу в полосе пропускания, чтобы совместно использовать канал. Спектр делится на диапазоны частот, каждый пользователь получает исключительное владение некоторой полосой, в которой он может послать свой сигнал. AM-радиовещание иллюстрирует FDM. Выделенный спектр составляет приблизительно 1 МГц, примерно от 500 до 1500 кГц. Другие частоты выделены другим логическим каналам (станциям), каждая станция действует в части спектра, с межканальным разделением, достаточно большим, чтобы предотвратить помехи [6].

Более подробный пример на рис. 2.21 показывает, как три речевых телефонных канала могут объединяться в одну линию с использованием частотного уплотнения. Фильтры ограничивают используемую полосу частот примерно 3100 Гц на каждый речевой канал. Когда одновременно мультиплексируется множество каналов, на каждый выделяется полоса 4000 Гц. Избыток называют **защитной полосой**. Она сохраняет каналы хорошо отделенными. Для начала сигналы повышаются по частоте, причем для разных каналов величины сдвигов разные. После этого их можно суммировать, поскольку каждый канал теперь сдвинут в свою область спектра. Заметьте, что даже при том, что между смежными каналами благодаря защитным полосам есть промежутки, имеется некоторое наложение. Это связано с тем, что у реальных фильтров нет идеального резкого края. Это означает, что сильный всплеск в одном канале может ощущаться как нетермальный шум в соседнем канале [5-6].

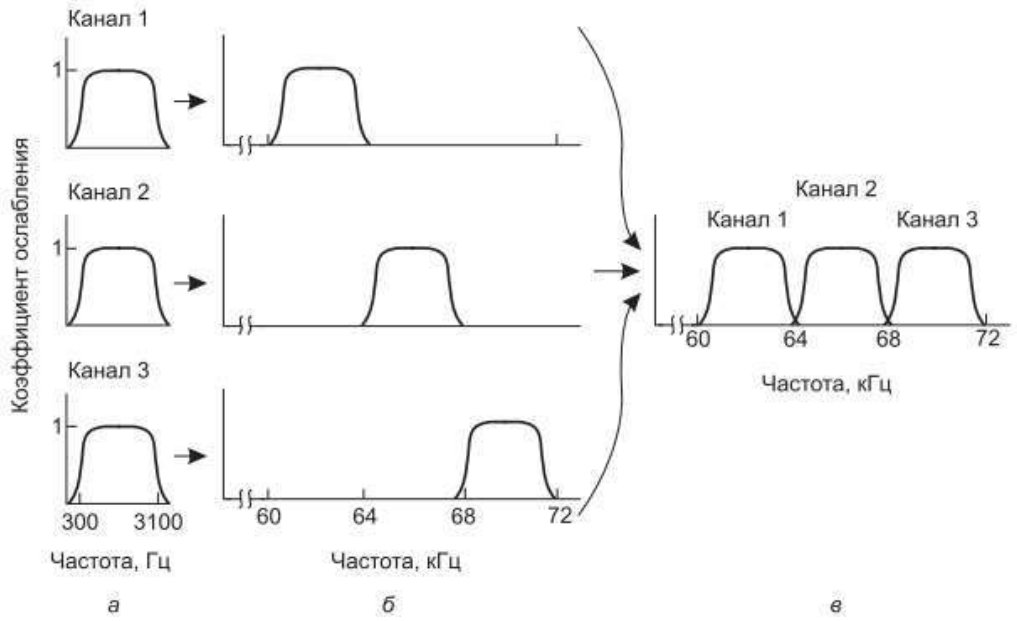


Рис. 23. Частотное уплотнение: а – исходные спектры сигналов, б – спектры сдвинутые по частоте, в – спектр уплотненного канала

Эта схема много лет использовалась для мультиплексирования звонков в телефонной сети, но теперь предпочтение отдается мультиплексированию по времени. Однако FDM продолжает использоваться в телефонных сетях, а также в сотовых, наземных радио и спутниковых сетях на более высоком уровне разбиения [5-6].

Мультиплексирование с разделением времени

Альтернатива FDM — **TDM** (**Time Division Multiplexing**, **Мультиплексирование с разделением времени, временное уплотнение**). Здесь пользователи сменяются (по кругу), каждый периодически получая всю полосу пропускания на небольшой отрезок времени. Пример трех потоков, мультиплексированных с помощью TDM, показан на рис. 2.23. Биты от каждого входного потока взяты в фиксированный **временной слот** и выведены в совокупный поток. Этот поток движется с суммарной скоростью отдельных потоков. Для этого потоки должны быть синхронизированы по времени. Маленькие **защитные интервалы**, аналогичные защитной полосе частот, могут быть добавлены, чтобы компенсировать небольшие отклонения синхронизации [6].

Рис. 24. Мультиплексирование с разделение времени (TDM)

TDM широко используется в телефонных сетях и сетях сотовой связи. Чтобы избежать путаницы, давайте отметим, что это очень отличается от альтернативного метода **STDM** (**Мультиплексирование со статистическим временным разделением**). Слово «статистическое» указывает, что отдельные потоки поступают в мультиплексный поток *не* по фиксированному распорядку, а согласно статистике их запросов. STDM — это другое название для пакетной коммутации [6].

Третий вид мультиплексирования, которое работает совершенно иначе, чем FDM и TDM, — это **CDM** (**Мультиплексирование с кодовым разделением, кодовое разделение каналов**). Оно является формой коммуникации **распределенного спектра**, в которой узкополосный сигнал распределяется по более широкому диапазону частот. Это делает его более терпимым к помехам, а также позволяет нескольким сигналам от различных пользователей совместно использовать общий диапазон частот. Поскольку мультиплексирование с кодовым разделением главным образом используется для этой последней цели, его обычно называют **CDMA** (**Code Division Multiple Access**, **множественный доступ с кодовым разделением**) [6].

Ч.1. Моделирование частотного мультиплексирования

Собрать схему, показанную на следующем рисунке.

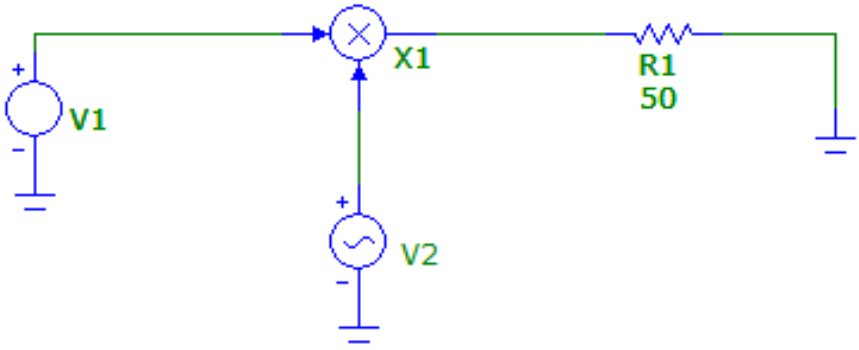


Рис. 25. Схема

V1 – Voltage Source. Представляет собой источник прямоугольных импульсов.

V1 – Sine Source. Представляет собой источник синусоидального сигнала для переноса сигнала на определенную частоту.

X1 – Mul. Представляет собой умножитель двух источников, на выходе которого произведение сигналов.

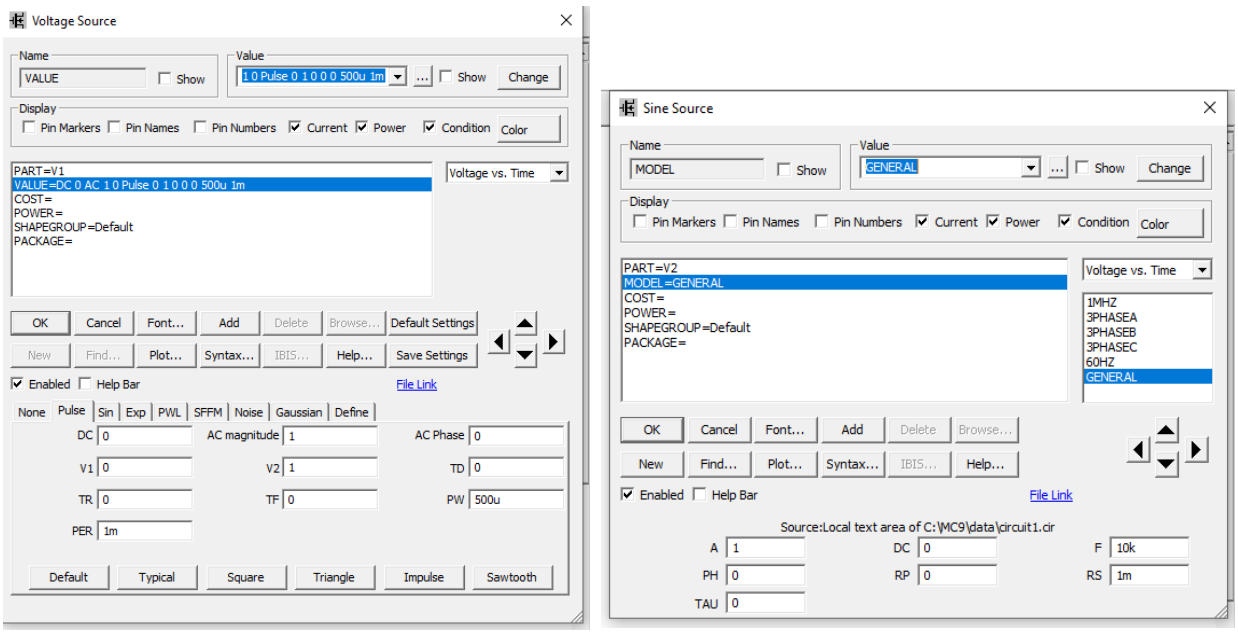
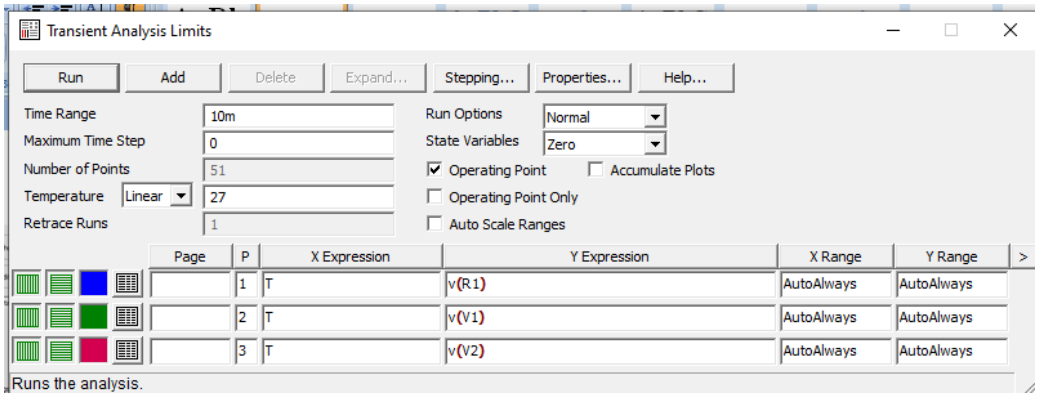


Рис. 26. Настройка источников а) полезного сигнала и б) несущего.

На рис. видно, что источник прямоугольных импульсов настроен для формирования последовательности амплитудой 1В, период 1мс, длительность единичного импульса 500 мкс. Источник несущего синусоидального сигнала настроен для формирования амплитуды также 1В, однако частоты 10 кГц. Настройки анализа переходных процессов на выходе схемы, а также напряжение на резисторе R1, показаны на следующем рисунке.



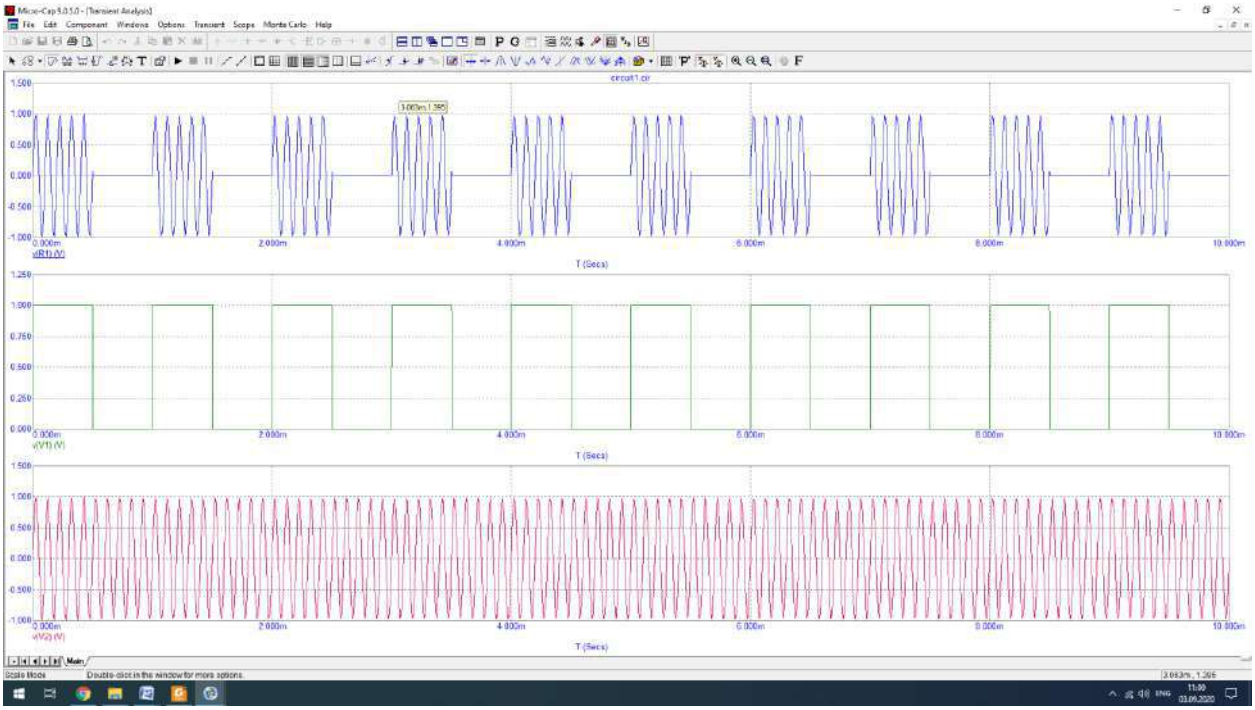
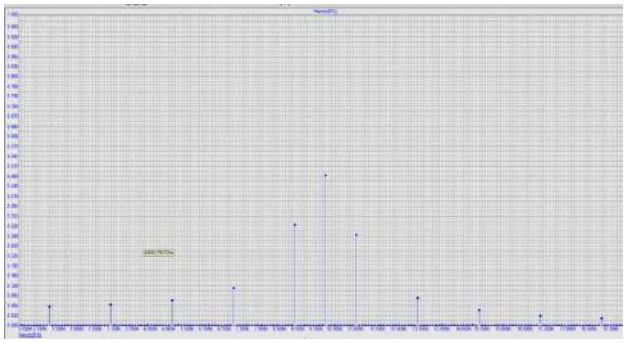
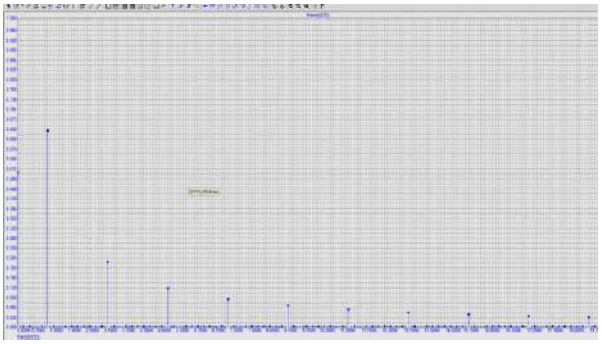


Рис. 27.

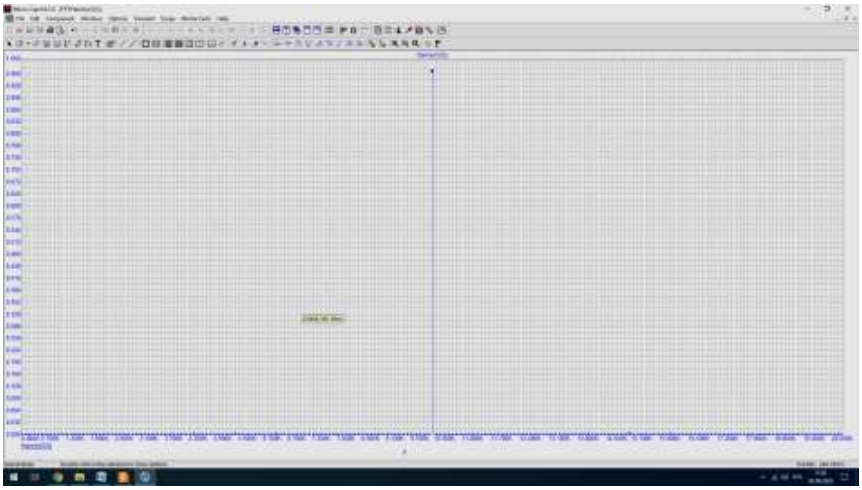
Из рис. виден модулированный сигнал. Далее приведены графики спектров.



На выходе R1



Источник прямоугольных импульсов



Генератор синусоидального сигнала

Рис. 28. Спектры

Для этого необходимо пройти в меню Transient->FFT Windows->Add FFT window. Во вкладке Plot в пункте Expression необходимо выбрать один из необходимых источников напряжения. Во вкладке Format для настроек оси X в качестве максимальной частоты указать в п. Range High 20000 (т.е. максимальная частота будет показана 20кГц), убрать галочку в п. Auto Scale. Лля настроек оси Y указать п. Range High 1 (т.е. максимальное напряжение будет показано 1В).

Ход работы

Построить две схемы из Рис. 25 Схема. Одна схема будет иметь несущую частоту 20 кГц, другая 50 кГц. Затем выходы с данных передатчиков будут суммироваться с помощью элемента Sum (Analog Primitives->Macros).

Затем после сумматора необходимо установить полосовой фильтр Баттерворта. Такой фильтр формируется автоматически с помощью меню Design->Passive Filters. Полоса пропускания данного фильтра должна быть настроена на центральную частоту одного из фильтров. Приблизительный вариант итоговой схемы приведен ниже на рис.

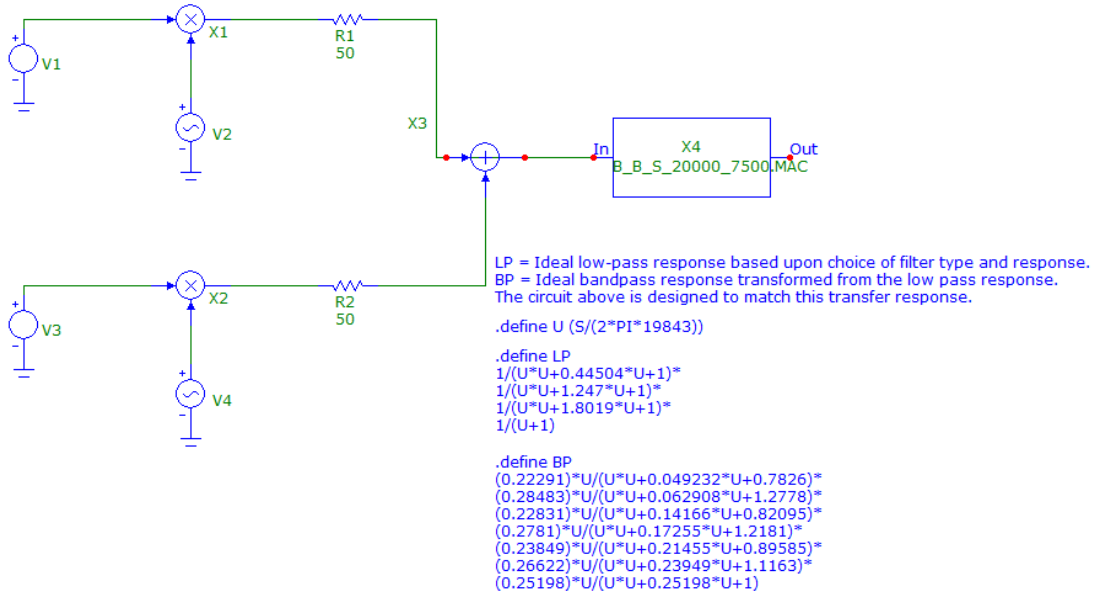


Рис. 29. Схема двух передатчиков

В отчёте необходимо привести эпюры напряжений на входе и выходе фильтра. Кроме того, привести спектр также в точках In и Out.

Ч.1. Моделирование кодового мультиплексирования

Мультиплексирование с кодовым разделением главным образом можно реализовать с помощью арифметической операции сложение по модулю 2 (XOR).

a	b	$a \oplus b$
0	0	0
1	0	1
0	1	1
1	1	0

В данной работе необходимо с помощью элемента **Stim1** (Component → Digital Primitives → Stimulus Generator → Stim1) сформировать информационную последовательность передатчика данные0 = (1, 0, 1, 1). Длительность импульса 2 мкс, таким образом полная последовательность должна быть 8 мкс. Затем с помощью другого элемента **Stim1** сформировать кодовую последовательность код0 = (1, 0). Здесь длительность импульса должна быть 1 мкс, т.е. в одном информационной импульсе должно умещать два кодовых. Получившиеся схема должна приблизительно выглядеть следующим образом:

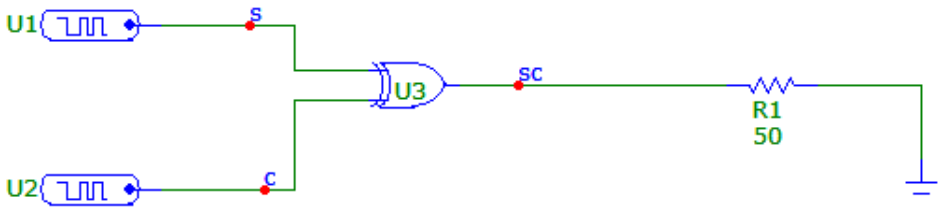


Рис. 30.

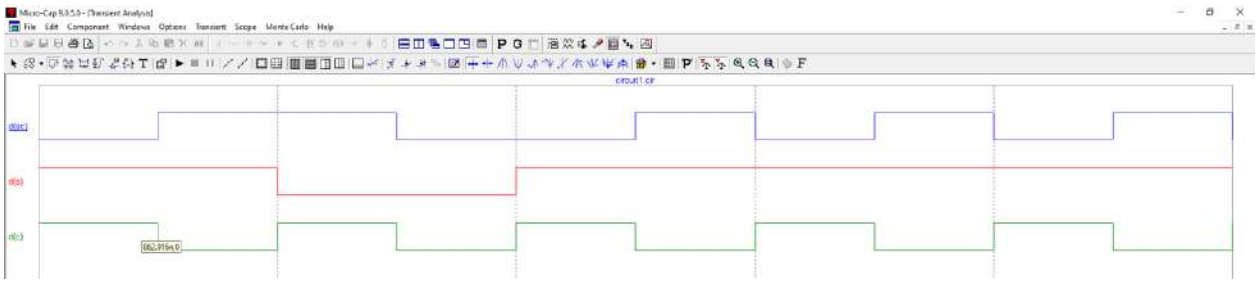


Рис. 31. Схема и напряжения на выходе передатчика.

Далее необходимо сформировать *Передатчик1,* который имеет код (1, 1) и данные (0, 0, 1, 1), и оба отправителя передают одновременно, т.е. суммируются. Привести эпюры напряжений после второго передатчика, а также суммарный сигнал данных двух отправителей.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Типы защиты от ошибок

Степень кодирования и избыточность

При использовании блочных кодов исходные данные делятся на блоки из *k* бит, которые иногда называют информационными битами, или битами сообщения; каждый блок может представлять любое из 2^k отдельных сообщений. В процессе кодирования каждый k-битовый блок данных преобразуется в больший блок из *n* бит, который называется кодовым битом, или канальным символом. К каждому блоку данных кодирующее устройство прибавляет (n - *k)* бит, которые называются *избыточными битами* (redundant bits), *битами четности* (parity bits), или *контрольными битами* (check bits); новой информации они не несут. Для обозначения описанного кода используется запись (n, *k).* Отношение числа избыточных бит к числу информационных бит, (n - *k)/k*, называется *избыточностью* (redundancy) кода; отношение числа бит данных к общему числу бит, k/n, именуется *степенью кодирования* (code rate). Под степенью кодирования подразумевается доля кода, которая приходится на полезную информацию. Например, в коде со степенью 1/2, каждый кодовый бит несет 1/2 бит информации [9].

Линейные блочные коды

Линейные блочные коды— это класс кодов с контролем четности, которые можно описать парой чисел *(n, k)* (объяснение этой формы записи приводилось выше). В процессе кодирования блок из *k* символов сообщения (вектор сообщения) преобразуется в больший блок из *n* символов кодового слова (кодовый вектор), образованного с использованием элементов данного алфавита. Если алфавит состоит только из двух элементов (0 и 1), код является двоичным и включает двоичные разряды (биты). Если не будет оговорено противное, наше последующее обсуждение линейных блочных кодов будет подразумевать именно двоичные коды [9].

Одним из важнейших параметров кода является кодовое расстояние. *Кодовым расстоянием* или просто *расстоянием* кода *V* называется минимальное расстояние между двумя различными кодовыми словами, т. е . [9]

$\mathbf{d}_V = \mathbf{min}\ d(\mathbf{x}, \mathbf{y}), \mathbf{x} \neq \mathbf{y}$ и $\mathbf{x}, \mathbf{y}$ Î V.

Для двоичного кода *V* под расстоянием понимается расстояние Хэмминга.

Теорема 1.1. Код V исправляет все комбинации из t или менее ошибок, если и только если кодовое расстояние не меньше, чем $2t+1$, т. е. $\mathbf{d}_V \geq 2t+1$.

Доказательство теоремы основано на построении сфер с радиусом t вокруг каждого кодового слова. Для того чтобы каждая ошибка кратности t была исправлена, слово с такой ошибкой должно содержаться внутри сферы, описанной только вокруг одного кодового слова. Таким образом, расстояние между центрами двух сфер, т. е. различными кодовыми словами должно быть не меньше $2t+1$. Геометрическая иллюстрация приведена на рис. 20 [9].

Теорема 1.2. Код V обнаруживает все комбинации из t или менее ошибок, если и только если кодовое расстояние не меньше, чем $t+1$, т.е. $\mathbf{d}_V \geq t+1$.

Для того, чтобы каждая ошибка кратности t была обнаружена, слово с такой ошибкой не должно быть кодовым, т. е. расстояние между любыми различными кодовыми словами должно быть больше t. [9]

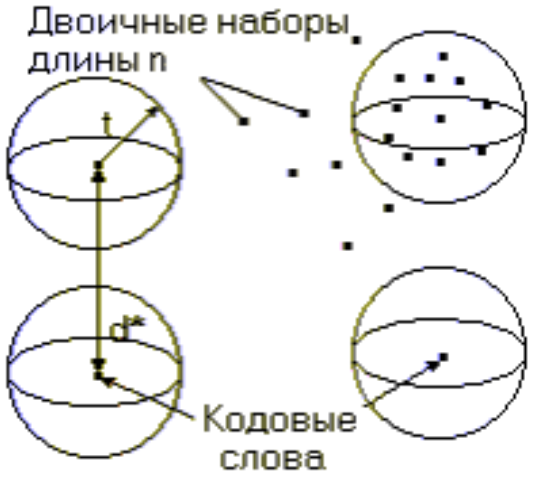


Рис 32. Сферы, центрами которых являются кодовые слова.

Например, код с расстоянием 3 исправляет все одиночные ошибки и обнаруживает все двукратные ошибки. Код с расстоянием 4 обнаруживает все ошибки кратности 3 и менее; однако такой код по-прежнему исправляет только одну ошибку [9].

Пример. В качестве примера попробуем построить код с кодовым словом длины 5, который имеет 3 информационных символа и исправляет одну ошибку. Так как информационных символов три, то различных кодовых слов должно быть $2^3 = 8$. Согласно теореме 1.1 расстояние между двумя различными кодовыми словами должно быть не меньше $2\times1 + 1 = 3$. Включим в код V слово 00000. Тогда все остальные кодовые слова имеют

вес не менее 3. После включения в код V любого слова веса 3, ни одно слова веса 3 в код добавлено быть не может, так как расстояние между любыми двумя словами веса 3 равно 2. По той же причине слово 11111 веса 5 также не принадлежит коду V. Остаются 5 слов веса 4, которых не достаточно, чтобы иметь 8 кодовых слов [9].

1. Определить способность исправлять и обнаруживать ошибки, если кодовое расстояние составляет 5.
2. Определить способность исправлять и обнаруживать ошибки, если кодовое расстояние составляет 7.
3. Определить способность исправлять и обнаруживать ошибки, если кодовое расстояние составляет 3.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Алгоритм Хэмминга

Кодирование

Предположим, мы хотим передать эти данные (1011) по шумному каналу связи. Мы берем произведение **G** и **p** с записями по модулю 2, чтобы определить переданное кодовое слово **x** [10]:

$$\mathbf{x} = \mathbf{Gp} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Это означает, что 0110011 будет передаваться вместо передачи 1011.

Проверка четности

Если при передаче не возникает ошибок, то полученное кодовое слово **r** идентично переданному кодовому слову **x** [10]:

r = x

Приемник умножает **H** и **r** для получения вектора синдрома **z**, который указывает, произошла ли ошибка и, если да, то для какого бита кодового слова. Выполняя это умножение (опять же, записи по модулю 2) [10]:

$$\mathbf{z} = \mathbf{Hr} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Коррекция ошибок

В противном случае предположим, что произошла ошибка в один бит. Математически мы можем написать

$\mathbf{r} = \mathbf{x} + \mathbf{e_i}$

по модуля 2, где $\mathbf{e_i}$ это i-й единичный вектор, т.е., нулевой вектор с 1 в i-й месте, считая от 1.

$$\mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Таким образом, вышеприведенное выражение означает ошибку с одним битом в i-м месте.

Теперь, если мы умножим этот вектор на **H**:

$$\mathbf{Hr} = \mathbf{H}\left(\mathbf{x} + \mathbf{e}_i\right) = \mathbf{Hx} + \mathbf{He}_i$$

Поскольку **x** переданные данные, они без ошибок, и, как результат, произведение **H** и **x** ноль. Таким образом

$$\mathbf{Hx} + \mathbf{He}_i = \mathbf{0} + \mathbf{He}_i = \mathbf{He}_i$$

Теперь произведение **H** на i-й стандартный базисный вектор выделяет этот столбец **H**, мы знаем, что ошибка происходит в том месте, где встречается этот столбец **H**.

Например, предположим, что мы ввели битную ошибку в бит № 5 [9]

$$\mathbf{r} = \mathbf{x} + \mathbf{e}_5 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

И тогда,

$$\mathbf{z} = \mathbf{Hr} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

который соответствует пятому столбцу H. Синдром 101 соответствует двоичному значению 5, которое указывает, что пятый бит был поврежден. Таким образом, ошибка была обнаружена в бите 5 и может быть исправлена (просто перевернуть или отменить ее значение):

$$\mathbf{r}_{\text{corrected}} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \overline{1} \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Это исправленное полученное значение действительно теперь соответствует переданному значению **x** сверху.

Дешифрование

Сначала определим матрицу **R**:

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Тогда полученное значение $\mathbf{p_r}$ равно **Rr**. Используя запущенный пример сверху

$$\mathbf{p_r} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

1. Построить дерева Хаффмана, если символ А встречается 15 раз, Б – 7 раз, В – 6 раз, Г – 6 раз, Д – 5 раз.
2. Коды, исправляющие ошибки при передачи сигналов
3. Построить дерева Хаффмана, если символ А встречается 5 раз, Б – 7 раз, В – 6 раз, Г – 8 раз, Д – 1 раз.
4. Детектирование ошибок. Кодовое расстояние.
5. Построить дерева Хаффмана, если символ А встречается 1 раз, Б – 17 раз, В – 6 раз, Г – 16 раз, Д – 5 раз.

ПРАКТИЧЕСКАЯ РАБОТА

Формирователь комплексной огибающей сигнала

Наибольшее распространение на сегодняшний день получили различные виды *квадратурной амплитудной манипуляции* (КАМ). Доказано, что квадратурная амплитудная манипуляция является наиболее эффективным видом модуляции. В современных системах подвижной связи, начиная с поколения 2.75G (EDGE), произошел переход к фазовым видам манипуляции и квадратурным видам амплитудной манипуляции высоких порядков. На рисунке 33 показаны сигнальные созвездия фазовых видов манипуляции BPSK, QPSK, 8PSK и квадратурной амплитудной манипуляции QAM16 на комплексной плоскости.

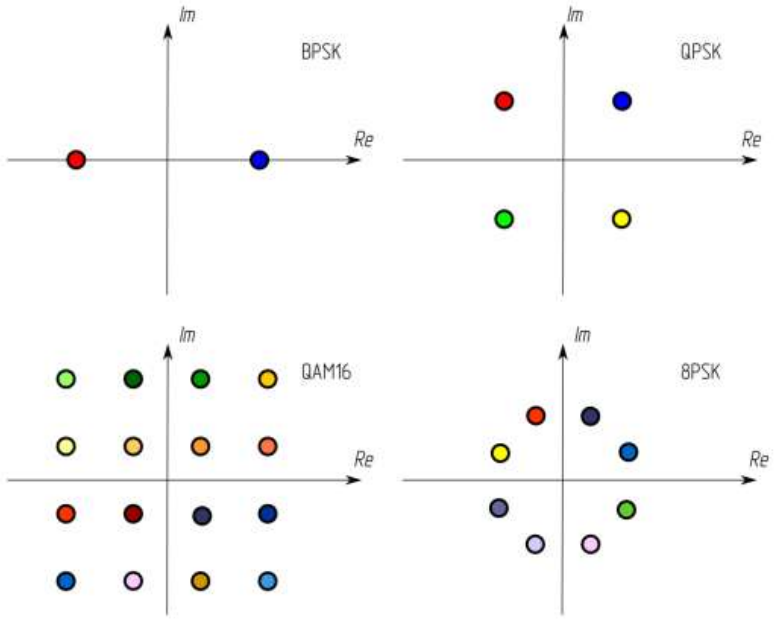


Рис. 33. Сигнальные созвездия фазовых видов модуляции BPSK, QPSK, 8PSK и квадратурной амплитудной манипуляции QAM16

Квадратурная амплитудная модуляция реализуется на практике при помощи *таблицы соответствий* (*таблицы истинности*), в которой для каждого входного символа определена одна точка на комплексной плоскости. Модулятор в таких системах называют *формирователем комплексной огибающей*. Выходной сигнал формирователя комплексной огибающей фильтруется и поступает на *квадратурный модулятор*, выполняющий перенос спектра сигнала на несущую или промежуточную частоту [3].

Модулятор современных цифровых систем связи выполняется при помощи простых таблиц соответствий, в которых для каждого символа данных соответствует одна точка на комплексной плоскости (рис. 34).

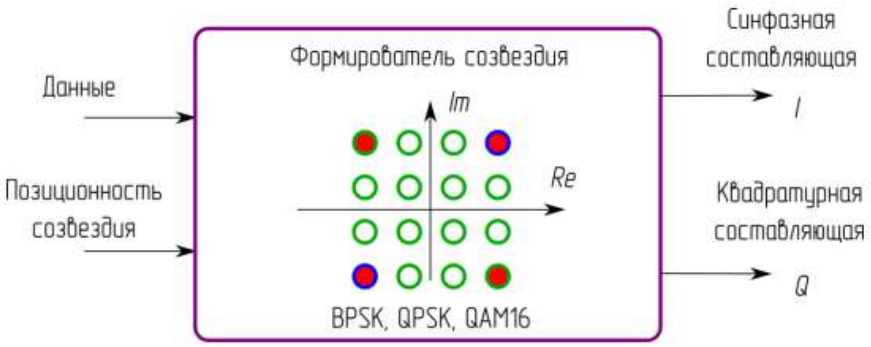


Рис. 34. Формирователь комплексной огибающей сигнала

В таблице 2 приведены примеры созвездий для манипуляций BPSK, QPSK и QAM16.

Таблица 1.2

Таблицы соответствий для различных видов манипуляции

Название манипуляции	Позиционность созвездия	Данные для передачи				Выход формирователя комплексной огибающей			
BPSK	2	0		1		-1-0j		+1+0j	
QPSK	4	0		1		-1+1j		+1+1j	
		2		3		+1-1j		-1-1j	
QAM16	16	0	1	2	3	-3+3j	-1+3j	+1+3j	+3+3j
		4	5	6	7	-3+1j	-1+1j	+1+1j	+3+1j
		8	9	10	11	-3-1j	-1-1j	+1-1j	+3-1j
		12	13	14	15	-3-3j	-1-3j	+1-3j	+3-3j

Приведенные в таблице 2 созвездия могут нормироваться для конкретных реализаций систем связи. Кроме того, иногда формирование созвездий совмещают с кодированием (например, кодирование Грея).

Столь простая реализация формирователя сигнала цифровой системы связи позволяет создавать устройства с адаптивно изменяемыми в процессе работы созвездиями, что дает возможность подстраиваться под изменение условий распространения радиосигналов в среде и использовать спектральный и энергетический ресурс наиболее эффективно [3].

Допустим, используется BPSK модуляция и передаются последовательно 0 и 1.

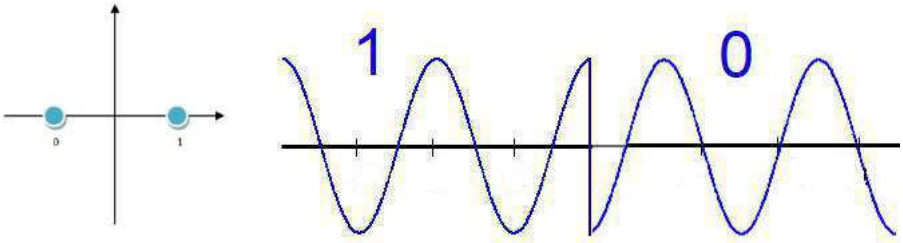


Рис. 35. Сигнальное созвездие и сигнал BPSK

Если взглянуть на таблицу и рисунок, то после формирователя будут -1+0j и +1+0j, т.е. косинусный сигнал единичной амплитуды со сдвигом фазы 0° и 180°.

Допустим, используется QPSK модуляция и передаются последовательно 1100. Тогда данная последовательность вначале разбивается на пачки по два бита каждая, т.е. первая посылка будет 11, а вторая 00.

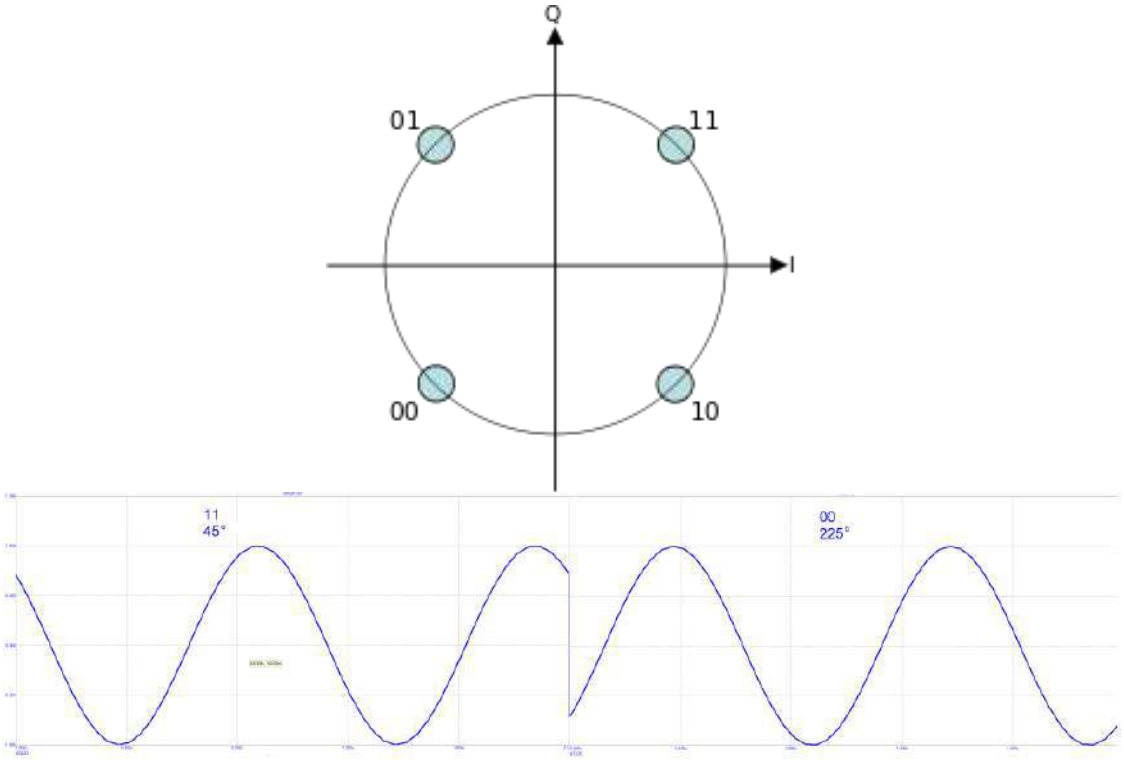


Рис. 36. Сигнальное созвездие и сигнал QPSK

Теперь если взглянуть на таблицу соответствий, то передаваться будут +1+1j и 1-1j, т.е. косинусный сигнал единичной амплитуды со сдвигом фазы 45° и 225°.

Практическое задание:

1. Пусть используется QPSK модуляция и необходимо передать следующую бинарную информационную последовательность 0111100100. Показать итоговый сигнал на выходе формирователя комплексной огибающей.
2. Пусть используется QPSK модуляция и необходимо передать следующую бинарную информационную последовательность 1010001010. Показать итоговый сигнал на выходе формирователя комплексной огибающей.
3. Пусть используется QPSK модуляция и необходимо передать следующую бинарную информационную последовательность

0000110101. Показать итоговый сигнал на выходе формирователя комплексной огибающей.

4. Пусть используется QAM16 модуляция и необходимо передать следующую бинарную информационную последовательность 00111101111000010110101010011010. Показать итоговый сигнал на выходе формирователя комплексной огибающей.
5. Пусть используется QAM16 модуляция и необходимо передать следующую бинарную информационную последовательность 00111111010010110011010010111000. Показать итоговый сигнал на выходе формирователя комплексной огибающей.
6. Пусть используется QAM16 модуляция и необходимо передать следующую бинарную информационную последовательность 00101010100101010111001111000010. Показать итоговый сигнал на выходе формирователя комплексной огибающей.

ЛАБОРАТОРНАЯ РАБОТА

Формирователь комплексной огибающей сигнала

Перенос спектра сигнала с нулевой промежуточной частоты осуществляется при помощи домножения сигнала на комплексную экспоненту. Практически такое домножение реализуется при помощи квадратурного гетеродина и двух умножителей (рис. 28). Перенос спектра может осуществляться как в цифровом виде, что более предпочтительно, так и в аналоговом [3].

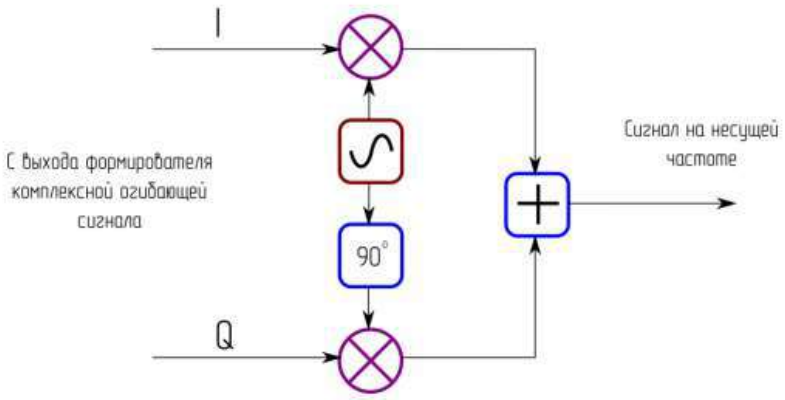


Рис. 37. Квадратурный модулятор

Некоторые наиболее дешевые современные системы связи используют аналоговую квадратурную обработку сигнала, что позволяет использовать недорогие низкочастотные ЦАП и дает возможность перенести спектр сигнала с нулевой промежуточной частоты сразу на несущую частоту. Благодаря этим преимуществам появляется возможность создавать компактные и универсальные системы связи, так как формирование сигнала происходит в цифровой форме и может изменяться на программном уровне. Аналоговое квадратурное преобразование имеет ряд недостатков, вызванных принципиальной невозможностью создания полностью одинаковых аналоговых каналов для синфазного и квадратурного компонентов комплексного сигнала. Требования к аналоговым трактам повышаются вместе с увеличением позиционности используемой манипуляции [3].

Цель работы: реализовать простую систему переноса спектра сигнала с нулевой частоты на промежуточную при помощи квадратурной схемы.

Задание на лабораторную работу.

В системе моделирования Microcap реализовать схему, изображенную на рис. 28. Для этого необходимо иметь два источника сигнала I и Q, которые бы генерировали постоянное напряжение согласно таблице соответствия.

Затем необходимы два умножителя сигналов для соответствующих каналов. На один вход подаётся постоянное напряжение I или Q, на другой – сигнал промежуточной частоты от блока гетеродина. Данный модуль должен состоять из источника косинусного напряжения, а также источника, генерирующего по форме –sin. Затем сигналы синфазного (I) и квадратурного (Q) каналов складываются.

Например, необходимо реализовать систему 4х-позиционной фазовой модуляции QPSK, которая бы передавала 01. Тогда на выходе формирователя комплексной огибающей сигнала будет -1+1j, на выходе I -1В (минус 1), на выходе Q 1В. Тогда на выходе должен быть сигнал косинуса со сдвигом фазы +135°. Зададим также частоту 1МГц. Общая схема, а также эпюра напряжений с выходов гетеродинов и источников постоянного напряжения I и Q показаны на следующем рисунке.

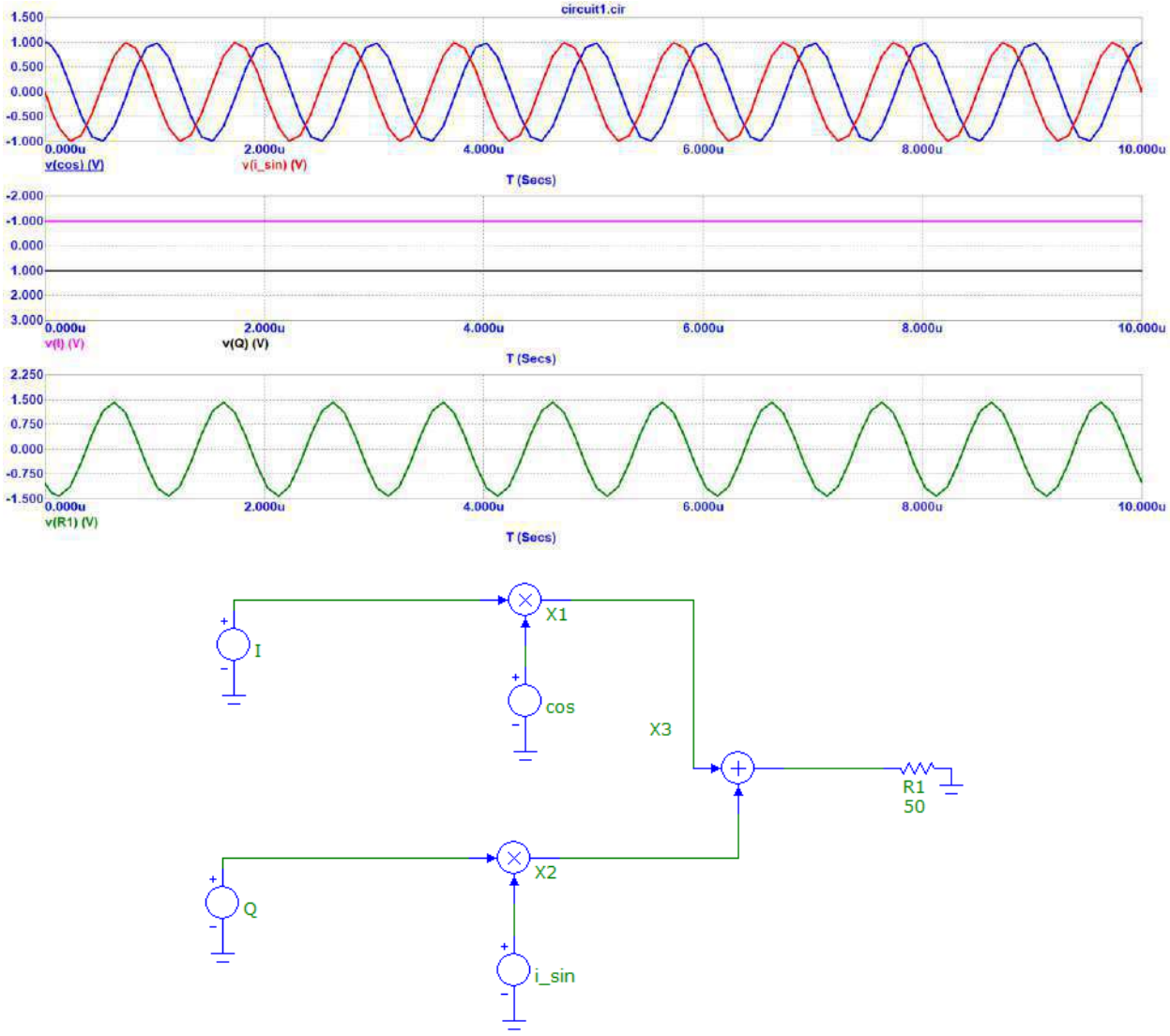


Рис. 38 Модель квадратурного модулятора

1. Пусть используется QPSK модуляция и необходимо передать следующую бинарную информационную последовательность 11 (или другие по заданию преподавателя). Показать итоговый сигнал на выходе формирователя комплексной огибающей.
2. Пусть используется QAM16 модуляция и необходимо передать следующую бинарную информационную последовательность 0010 (или другие по заданию преподавателя). Показать итоговый сигнал на выходе формирователя комплексной огибающей.

Список литературы

1. Пешков, И. В. Адаптивные алгоритмы пространственной обработки сигналов, эффективные при случайных дестабилизирующих воздействиях: дис. ... канд. физ.-мат. наук : 01.04.03 / И. В. Пешков. – Воронеж, 2012. – 182 с.

2. Компьютерное моделирование радиоэлектронных устройств: Методические указания по выполнению лабораторных работ / сост. Смирнов М.С. - Электрон, текстовые дан. (1,4 Мб). – Муром: МИ (филиал) ВлГУ, 2017.

3. Боев, Н.М. Системы связи. Подвижные системы связи: учебно-методическое пособие [Электронный ресурс] / сост. Н. М. Боев. – Красноярск: Сиб. федер. ун-т, 2012. – URL: http://diss.seluk.ru/m-radiotehnika/1087216-1-sistemi-svyazi-podvizhnie-sistemi-svyazi-lekcii-uchebno-metodicheskoe-posobie-elektronnoe-izdanie-krasnoyarsk-sfu-2013-udk-621396930.php

4. Сжатие данных. – URL: https://en.wikipedia.org/wiki/Data_compression.

5. Таненбаум, Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл; [пер. с англ. А. Гребеньков]. – 5-е изд. – М. [и др.] : Питер, 2014. – 955 с.

6. Кодовое разделение пользователей. – URL: https://en.wikipedia.org/wiki/Code-division_multiple_access.

7. Ташатов Н. Н. Использование избыточности для защиты от ошибок // Известия НАН РК. Серия физ.-мат. наук. – 2007. – №5. – С.63-68.

8. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение. 2-е изд., испр. ; пер. с англ. – М. : Издательский дом «Вильямс», 2003. – 1104 с.

9. Коды, исправляющие ошибки. – URL: https://ido.tsu.ru/iop_res1/kodi/index.php-mod=article&id=169.htm.

10. Коды Хэмминга. URL: https://en.wikipedia.org/wiki/Hamming(7,4).

Учебное издание

Илья Владимирович Пешков

ЛАБОРАТОРНЫЕ И ПРАКТИЧЕСКИЕ ЗАДАНИЯ ПО КУРСУ «ЦИФРОВЫЕ РАДИОПЕРЕДАЮЩИЕ УСТРОЙСТВА»

Методические указания

Технический редактор – О. А. Ядыкина
Техническое исполнение – В. М. Гришин
Формат 60 x 84 $^{1}/_{16}$. Гарнитура Times. Печать трафаретная.
Печ.л. 2,8 Уч.-изд.л. 2,6
Тираж 300 экз. Заказ 79

Отпечатано с готового оригинал-макета на участке оперативной полиграфии
Елецкого государственного университета им. И. А. Бунина

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Елецкий государственный университет им. И. А. Бунина»
399770, г. Елец, ул. Коммунаров, 28,1