

Министерство образования Российской Федерации

Государственное образовательное учреждение  
высшего профессионального образования

Ульяновский государственный технический университет

## ЦИФРОВЫЕ СИСТЕМЫ ПЕРЕДАЧИ

Методические указания к лабораторным работам  
для студентов специальности 200900  
Часть 1

Составитель В. Н. Горохин

Ульяновск 2015

УДК 654 (076)  
ББК 32.88я7  
С 41

Рецензент канд. техн. наук,

Одобрено секцией методических пособий научно-методического совета университета

**Цифровые** системы передачи: Методические указания к лабораторным работам для студентов специальности 200900. Часть 1/  
Сост. В.Н. Горохин. – Ульяновск: УлГТУ, 2015. – 24 с.

Указания разработаны в соответствии с программой курса «Цифровые системы передачи» и предназначены для студентов радиотехнического факультета. Рассмотрены вопросы линейного и помехоустойчивого кодирования в цифровых системах передачи, проектирования и моделирования схем кодера и декодера, правила построения и исследование цифрового канала передачи.

Цикл лабораторных работ направлен на закрепление знаний по курсу «Цифровые системы передачи», читаемого студентам специальности «Сети связи и системы коммутации». Сборник подготовлен на кафедре «Телекоммуникации».

УДК 654 (076)  
ББК 32.88я7

СОДЕРЖАНИЕ

<b>1. Лабораторная работа №1</b>	
ИССЛЕДОВАНИЕ ЛИНЕЙНОГО КОДИРОВАНИЯ .....	4
<b>2. Лабораторная работа №2</b>	
ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ..	8
<b>3. Лабораторная работа №3</b>	
МОДЕЛИРОВАНИЕ СХЕМЫ КОДЕРА И ДЕКОДЕРА .....	16
<b>4. Лабораторная работа №4</b>	
РАЗРАБОТКА И МОДЕЛИРОВАНИЕ УСТРОЙСТВА ТАКТОВОЙ СИНХРОНИЗАЦИИ .....	32
<b>5. Лабораторная работа №5</b>	
РАЗРАБОТКА ЦИФРОВОЙ СИСТЕМЫ ПЕРЕДАЧИ.....	32

## Лабораторная работа №1

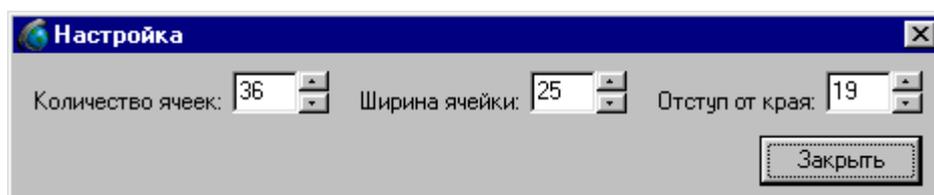
### *Исследование линейного кодирования*

#### 1. Общие сведения о линейных кодах и работе в среде программы CODE.EXE

##### 1.1. Программная среда CODE.EXE

Данная программа предназначена для проверки знаний и навыков студентов по линейному кодированию двоичных сигналов и вычислению параметров двоичных последовательностей.

Настройка данной программы производится с помощью пункта меню Опции | Настройка (Options | Configure).



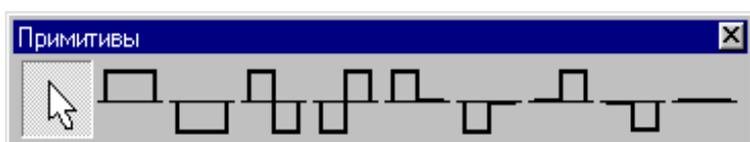
В качестве параметров настройки используются характеристики главного окна программы, а именно количество ячеек кодовой последовательности, ширина одной ячейки и отступ от края области изображения кода.

Все параметры настройки автоматически сохраняются при выходе из программы и восстанавливаются при следующем запуске.

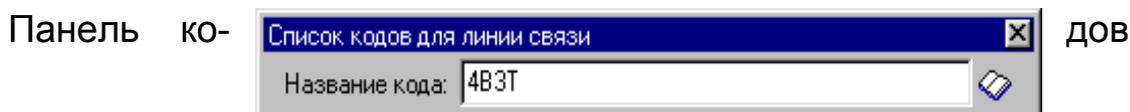
В качестве панелей инструментов используются панель примитивов, панель кодов, панель инструментов, а также строка состояния.

Управлять видимостью этих панелей можно с помощью пункта меню Вид. Панели элементов, кодов и инструментов можно легко перетащить в любое нужное Вам место на экране или вообще закрыть.

На панели примитивов



содержатся кнопки, с помощью которых можно выбрать любой из элементов и сделать его активным для построения кодовой последовательности. Набор данных элементов кода является исчерпывающим и с их помощью можно построить любой из предлагаемых Вам кодов.

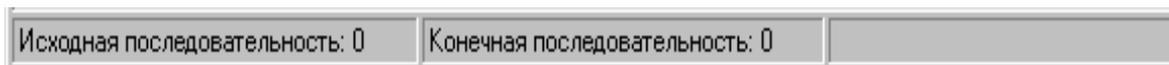


содержит список доступных в данный момент кодов и позволяет выбрать любой из них.



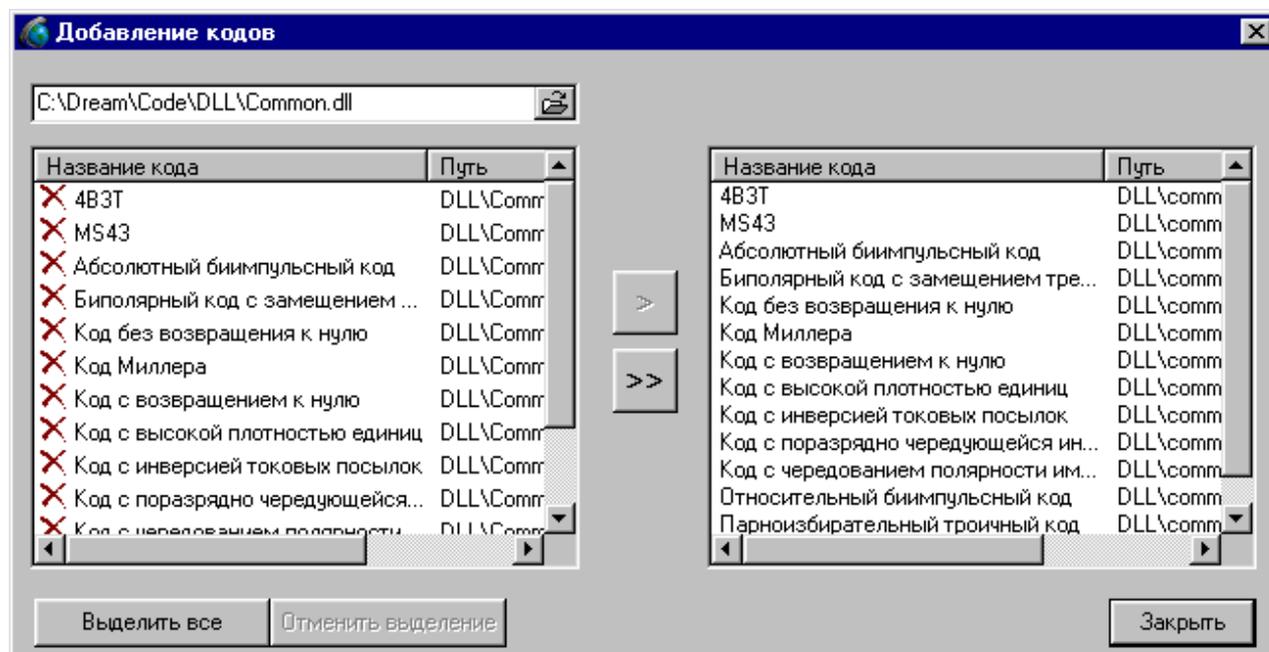
позволяет использовать средства инструментальной среды, такие как расчет параметров кодов, построение автокорреляционной и спектральной характеристик.

Строка состояния



отображает порядковый номер активной в данный момент ячейки.

Добавить кодовую последовательность в список доступных в данный момент кодов можно с помощью пункта меню Опции | Добавить код. После выбора данного пункта на экране отобразится окно



В правой части данного окна представлен список доступных в данный момент кодов, а в левой - коды, содержащиеся в открытой в данный момент библиотеке.

При совпадении названий доступных кодов и кодов из открытой библиотеки, они помечаются значком , показывая, что коды с такими названиями добавлять нельзя. Перед тем как все-таки добавить такие коды необходимо сменить название, для чего достаточно щелкнуть левой кнопкой мыши на данном коде и ввести новое название.

Добавлять можно только коды, помеченные значком .

Для того, чтобы добавить код необходимо открыть библиотеку кодов, представленную в виде DLL – модуля. Открываемая библиотека должна быть написана по определенному шаблону.

Библиотека, содержащая описание кодов, представляет собой DLL-модуль. Библиотека может быть написана на любом языке программирования при условии соблюдения соответствия типов данных.

Представленная библиотека написана на языке программирования Паскаль в среде Delphi 4.0.

Описание алгоритма формирования кода оформляется в виде процедуры, в которую в качестве параметра передается запись следующего вида:

```
TProcParam = record
  Alphabit   : boolean;
  ElemCount  : integer;
  SourceArr  : TArray;
  EndArr     : TArray;
  Cells      : TCellsRec;
  UsesElem0  : TUsesElem;
  UsesElem1  : TUsesElem;
end;
```

Данная запись описывает формат данных, получаемых из библиотеки.

Поля данной записи имеют следующие значения:

Alphabit – переменная логического типа, определяющая тип кода (алфавитный – значение TRUE, или неалфавитный – значение FALSE);

ElemCount – переменная целого типа, передающая в процедуру значение, указывающее на количество элементов (тактовых интервалов) в кодовой последовательности;

SourceArr – массив элементов исходной кодовой последовательности;

EndArr – массив элементов результирующей кодовой последовательности (заполняется согласно алгоритму и является результатом кодирования);

Cells – запись, определяющая соответствие элементов исходной и результирующей кодовой последовательности;

UsesElem0 – массив, определяющий элементы, соответствующие двоичному нулю исходной кодовой последовательности (необходим для расчета параметров кода);

UsesElem1 – массив, определяющий элементы, соответствующие двоичной единице исходной кодовой последовательности (необходим для расчета параметров кода).

Массив элементов кодовой последовательности определяется следующим образом:

TArray = array of integer;

Во всех рассмотренных случаях под элементом понимается сочетание видеоимпульсов и пауз внутри тактового интервала, отведенного для передачи одного символа информации. Возможное число элементов составляет девять (рис. 1) и при кодировании данные элементы представляются целыми числами от одного до девяти.

Соответствие между элементами (примитивами) и элементами массива

Элементы (примитивы)	Элементы массива
$S_1(t)$	1
$S_2(t)$	2
$S_3(t)$	3
$S_4(t)$	4
$S_5(t)$	5
$S_6(t)$	6
$S_7(t)$	7
$S_8(t)$	8
$S_9(t)$	9

Запись, определяющая соответствие элементов исходной и результирующей кодовой последовательности, определяется следующим образом:

TCellsRec = record

Source : integer;

Result : integer;

end;

Данная запись показывает сколько элементов исходной последовательности (поле Source) соответствует элементам результирующей последовательности (поле Result). Определение данного соответствия полезно для алфавитных кодов, для остальных кодов следует принимать значения данных полей, равными единице.

Массивы, определяющие элементы, соответствующие двоичному нулю и единице исходной кодовой последовательности, определяются следующим образом:

```
TUsesElem = array [0..1] of integer;
```

Определение таких элементов необходимо для расчета параметров кода. При этом массив является двумерным на тот случай если код формируется несколькими наборами элементов. В случае когда используется только один элемент, соответствующий либо нулю, либо единице, второй элемент приравнивается к нулю.

При передаче в процедуру данной записи указывается с помощью ключевого слова var, что данная запись передается как переменная. После описания процедуры указывается, что она экспортируется (ключевое слово export).

Ниже приведен пример процедуры, описывающей алгоритм формирования кода без возвращения к нулю.

```
procedure NRZ (var Paramtrs : TProcParam); export;  
var  
  i : integer;  
  
begin  
  Paramtrs.Alphabit:=false;  
  Paramtrs.Cells.Source:=1;  
  Paramtrs.Cells.Result:=1;  
  Paramtrs.UsesElem0[0]:=2;  
  Paramtrs.UsesElem0[1]:=0;  
  Paramtrs.UsesElem1[0]:=1;  
  Paramtrs.UsesElem1[1]:=0;  
  
  for i:=0 to Paramtrs.ElemCount-1 do  
    if Paramtrs.SourceArr[i]=9 then  
      Paramtrs.EndArr[i]:=2
```

```
else Parametr.EndArr[i]:=Parametr.SourceArr[i];  
end;
```

Для обеспечения автоматического поиска процедур, описывающих алгоритмы кодирования, в библиотеке должна присутствовать процедура, хранящая информацию такого рода. Название такой процедуры фиксировано и она должна называться "Info". Эта процедура также должна экспортироваться и ей в качестве параметра должен передаваться объект TList.

Предварительно в библиотеке должен быть описан класс TCodeDesc:

```
TCodeDesc = class(TObject)  
public  
  ProcName      : string;  
  CodeName      : string;  
  EngName : string;  
  RusName : string;  
  LibName  : string;  
  constructor Create(_ProcName, _CodeName, _EngName,  
_RusName, _LibName : string);  
end;
```

А также его конструктор:

```
constructor TCodeDesc.Create;  
begin  
  inherited Create;  
  ProcName:=_ProcName;  
  CodeName:=_CodeName;  
  EngName:=_EngName;  
  RusName:=_RusName;  
  LibName:=_LibName;  
end;
```

В этом классе:

ProcName – строка, содержащая название процедуры, описывающей алгоритм кодирования;

CodeName – строка, содержащая название кода;

EngName – строка, содержащая английскую аббревиатуру кода;

RusName – строка, содержащая русскую аббревиатуру кода;

LibName – строка, содержащая название библиотеки, в которой находится данный код (данная строка оставляется пустой и заполняется только после подключения данного кода к инструментальной среде).

Ниже приведен пример процедуры Info.

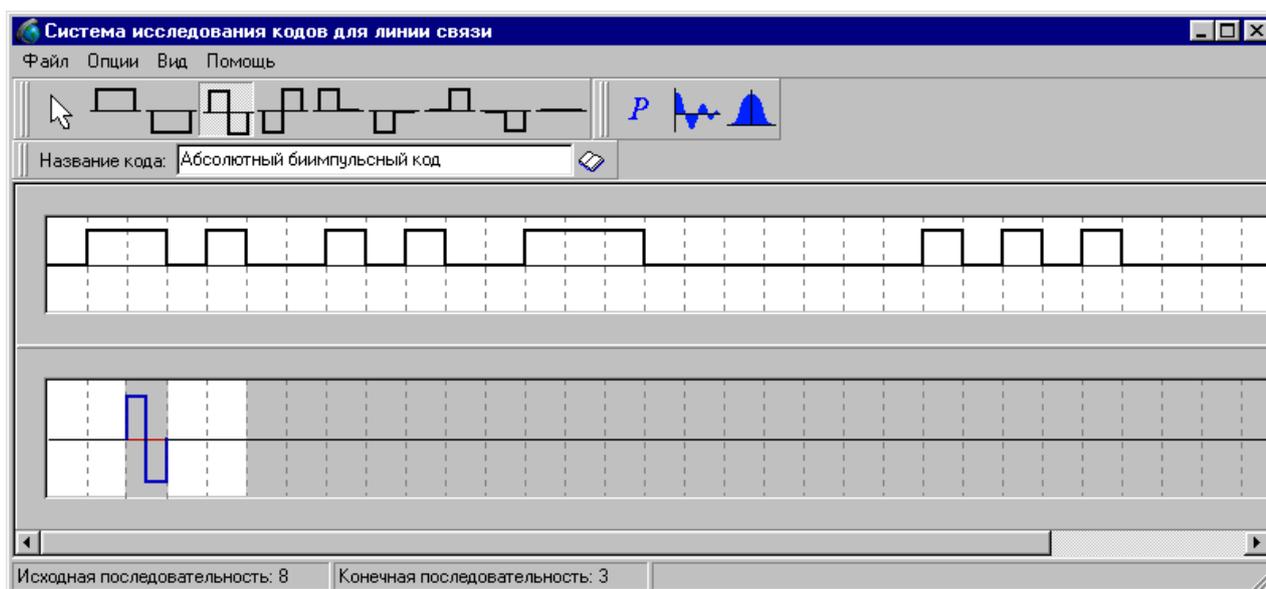
```
procedure Info(CodeList : TList); export;  
begin  
  CodeList.Add(TCodeDesc.Create('NRZ', 'Код без возвращения к нулю',  
'NRZ', 'БВН' ,));  
  CodeList.Add(TCodeDesc.Create('RZ', 'Код с возвращением к нулю',  
'RZ', ", "));  
end;
```

В конце модуля после ключевого слова exports располагаются названия процедур, которые в дальнейшем могут быть экспортированы, а также имена, по которым будет обеспечиваться доступ к таким процедурам:

```
exports  
  Info name'Info',  
  RZ name'RZ',  
  NRZ name'NRZ';  
  
begin  
end.
```

Пользоваться данной программой очень просто, интерфейс интуитивно понятен, попробуйте и у Вас получится. Но, если все-таки что-то непонятно, то данный раздел помощи поможет Вам разрешить любую из проблем, возникших при работе с данным пакетом.

Построение результирующего кода осуществляется последовательно по пять элементов. Активные элементы выделяются белым фоном. Поместить элемент кода в одну из активных ячеек можно следующим образом. Выбрав на панели элементов нужный элемент кода (щелкнув по соответствующей кнопке левой клавишей мыши), необходимо подвести курсор к той ячейке результирующего кода, в которую Вы намереваетесь поместить данный элемент и, убедившись в правильности своего выбора (выбранный незакрепленный в ячейке элемент будет нарисован в текущей ячейке), щелкнуть левой клавишей мыши.



Если Вы все же ошиблись и хотите исправить ошибку, то просто поместите правильный элемент поверх ошибочного. Заполнив все активные ячейки элементами и проверив правильность такого заполнения нажмите на пе-



риодически мигающую кнопку

В случае правильности формирования данного участка кода Вы получаете возможность сформировать следующую часть кодовой последовательности. В случае если Вы допустили ошибку, о чем Вас проинформирует сообщение соответствующего содержания, Вы не сможете перейти на следующий этап пока не исправите все ошибки. Если Вы допустите **три** ошибки при построении кода, то после вывода соответ-

вующего сообщения, Ваше задание будет заменено на другое. Если же Вы умудрились не допустить ни одной ошибки при построении трех участков кода, то Вы получаете бонус в виде автоматического построения оставшейся части кодовой последовательности.

Сменить код можно с помощью панели элементов. Все действия, производимые Вами в процессе работы с программой, заносятся в отчет, который вы можете просмотреть и сохранить с помощью соответствующих пунктов меню.

Выбрав пункт меню Опции | Параметры кода на экране отображается окно

Параметры кода

Текущий код: Абсолютный биимпульсный код

Тип кода: Алфавитный

Эквивалентная мощность: 0.000

Коэффициент, характеризующий среднее значение тактовой частоты: 0.000

Коэффициент, характеризующий минимальное значение тактовой частоты: 0.000

Коэффициент, характеризующий максимальное значение тактовой частоты: 0.000

Коэффициент, характеризующий реальное значение тактовой частоты: 0.000

Коэффициент устойчивости признаков тактовой частоты: 0.000

Число групп двоичных символов: 0.000

Число групп символов кода: 0.000

Коэффициент изменения тактовой частоты: 0.000

Избыточность кода (%): 0.000

Предельный коэффициент снижения тактовой частоты: 0.000

Сведения Ok Cancel

С помощью этого окна Вам предоставляется возможность проверить свои знания по расчету параметров кодов. Введя все параметры кода и нажав на кнопку Ok на экране появится сообщение, в котором будет содержаться заключение о правильности Ваших расчетов.

Заключение о правильности введенных Вами параметров автоматически занесется в отчет, который Вы в последствии сможете сохранить в файл.

Расчет параметров кода производится по следующим формулам:

– эквивалентная мощность элементов

$$P_{\text{э}} = \frac{1}{T} [S_i(t) - S_j(t)]^2 dt,$$

где  $S_i(t)$  и  $S_j(t)$  – элементы сигнала, которые используются при передаче "0" и "1" исходной информации;

– коэффициент, характеризующий среднее значение тактовой частоты

– коэффициент устойчивости признаков тактовой частоты

$$K_T = \frac{P_{T \min}}{P_{T \max}}$$

Сравнение алфавитных кодов (преобразование кода  $2^n$  в код  $M^k$ ) проводится по следующим дополнительным параметрам:

– число групп двоичных символов  $2^n$ ;

– число групп символов кода  $M^k$ ;

– коэффициент изменения тактовой частоты

$$K_M = n/k;$$

– избыточность кода

$$r = (H_{\max} - H)/H_{\max} \leq 1,$$

где  $H$  - энтропия источника сообщений (сигналов), равная  $H = \log_2 N$ , если все сообщения (сигналы) равновероятны и объем алфавита  $N$ ;

– предельный коэффициент снижения тактовой частоты (при  $r = 0$ )

$$K_{M \max} = \lim_{r \rightarrow 0} (n/k) = \log_2 M.$$

## 1.2. Линейное кодирование

Для передачи по линейному тракту информационный двоичный сигнал, формируемый на выходе каналообразующего оборудования, преобразуется в некоторый специальный цифровой сигнал, параметры которого должны быть согласованы с характеристиками используемых направляющих систем и отвечать ряду специфических требований:

1. Энергетический спектр цифровых сигналов, передаваемых по линии, должен быть сосредоточен в относительно узкой полосе частот при отсутствии постоянной составляющей (вне зависимости от статистических свойств исходной двоичной информации). Это позволяет уменьшить межсимвольные искажения, возникающие вследствие ограничения полосы частот линейного тракта в области как нижних (например, за счет ограничения полосы частот с целью оптимизации отношения сигнал - тепловой шум и увеличения затухания в линии) частот, повысить взаимозащищенность цепей симметричных кабелей, которая снижается с увеличением частоты, обеспечить возможность совместной работы с аналоговыми системами передачи на параллельных цепях и т.п. В итоге выполнение указанного требования позволяет либо увеличить длину участка регенерации при заданной вероятности ошибки, либо повысить верность передачи при заданной длине участка регенерации.

2. Структура цифрового сигнала в линии должна быть такой, чтобы можно было просто и надежно выделять тактовую частоту в каждом линейном регенераторе.

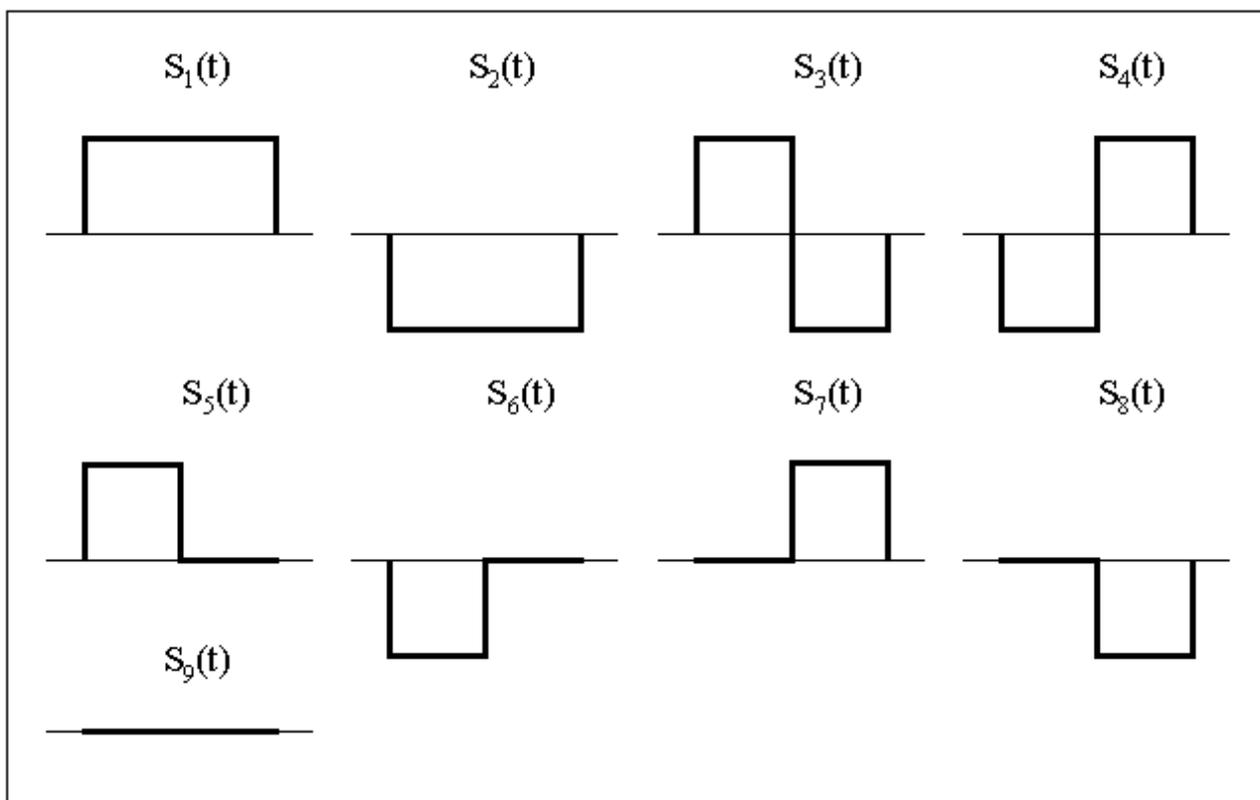
3. Должна обеспечиваться возможность постоянного достаточно простого контроля за коэффициентом ошибок в линейном тракте без перерыва связи.

Кроме того, используемые в линии коды должны иметь достаточно простую аппаратную реализацию, не должны приводить к существенному размножению ошибок и т.п.

Формирование требуемого энергетического спектра может быть осуществлено соответствующим изменением структуры импульсной последовательности и выбором импульсов нужной формы.

Учитывая, что на вероятности появления символов исходного двоичного сигнала не могут быть наложены ограничивающие условия, код должен обладать некоторой избыточностью, позволяющей удовлетворить дополнительные требования.

Будем считать элементом видеоимпульсного сигнала любое возможное сочетание видеоимпульсов и пауз внутри тактового интервала  $T$ , отведенного для передачи одного символа информации. Если принять, что импульсы имеют прямоугольную форму, их длительность равна  $T$  или  $T/2$ , передний фронт совпадает с началом или серединой тактового интервала, а амплитуда равна  $A/2$ , то возможное число разнотипных элементов видеоимпульсных сигналов  $S_k(t)$  составит девять.



При передаче 1 и 0 исходной информации может использоваться комбинация двух любых элементов видеоимпульсного сигнала  $S_i(t)$  и  $S_j(t)$ . В этом случае число двоичных видеоимпульсных сигналов  $C_9^2=36$ . Кроме того в цифровых системах передачи могут применяться многоэлементные видеоимпульсные сигналы (передача одного символа информации осуществляется с помощью нескольких элементов), а также многопозиционная (многоуровневая) модуляция отдельных элементов (например, для каждого элемента предусматривается несколько градаций амплитуды). Т.о. видеоимпульсные сигналы отличаются весьма большим многообразием.

Однако следует учесть, что далеко не все видеоимпульсные сигналы отвечают требованиям, предъявляемым к цифровым сигналам, которые предназначены для передачи по линии.

С целью повышения стабильности признаков тактовой частоты для большинства видеоимпульсных сигналов требуется дополнительное преобразование цифровой информации путем изменения ее статистических свойств. Если изменение статистических свойств исходной двоичной информации происходит при некоторых определенных условиях (например, заданном количестве подряд следующих 0), то в результате формируются так называемые неалфавитные коды. Если же статистические свойства исходной двоичной последовательности изменяются путем ее деления на группы с постоянным числом тактовых интервалов

и последующего преобразования этих групп по определенному алфавиту в группы символов кода с другим основанием счисления (больше двух) и, как правило, с новым числом тактовых интервалов, то в результате формируются алфавитные коды. Использование алфавитных кодов позволяет не только повысить стабильность признаков тактовой частоты, но и увеличить пропускную способность системы передачи за счет соответствующего снижения значения тактовой частоты. Однако, при этом требуется передача признаков, достаточных для правильного восстановления границ групп символов кода при декодировании.

Среди неалфавитных кодов наибольшее распространение получили коды типов, в которых последовательность двоичных 0 длиной  $(n+1)$  символов заменяется определенным сочетанием импульсов и пауз.

Алфавитное кодирование характеризуется избыточностью преобразования двоичной информации ( $2^n < M^k$ ), что обеспечивает выполнение условия независимой передачи групп двоичных символов сочетаниями (группами)  $M$ -ичного кода.

Наибольшее распространение на практике получили троичные и квазитроичные коды. Простейшим из них является код ЧПИ. Существенным недостатком кода ЧПИ является то, что с целью обеспечения устойчивой работы регенераторов (цепей выделения тактовой частоты) необходимо накладывать ограничения на допустимое число подряд следующих нулей в исходной двоичной последовательности, что вызывает определенные неудобства.

Код В3ZS является модифицированным квазитроичным и называется биполярным кодом с замещением трех нулей. Символы логической 1 имеют длительность, равную половине тактового интервала, и попеременно изменяющуюся полярность (т.е. передаются элементы  $S_5(t)$  или  $S_6(t)$ ), а логический 0 передается элементом  $S_9(t)$  (пассивная пауза). Исключение правила чередования полярностей имеет место только в том случае, если в цифровом сигнале появляются три (или более) подряд следующих символов 0. В коде В3ZS каждая группа из трех последовательных нулей заменяется комбинацией В0V или 00V, где В - импульс, отвечающий правилу чередования полярности логических 1, V - импульс, нарушающий правило чередования логических единиц. Выбор одной из указанных замещающих комбинаций (В0V или 00V) производится с таким расчетом, чтобы число импульсов вида В между соседними импульсами вида V было нечетным.

На практике находит применение также код В6ZS, в котором аналогичным образом замещаются шесть подряд следующих символов 0 (используется комбинация 0BV0BV).

В отличие от кодов типов HDB $n$  и В $n$ ZS в парноизбирательном троичном (ПИТ) и почти разностном квазитроичном (ПРКК) кодах осуществляется не замещение серии пробелов, а преобразование символов всей двоичной последовательности. Так, в модифицированном ПИТ ко-

де символы передаваемой двоичной последовательности группируются попарно и преобразуются в кодовые группы троичного сигнала в соответствии с таблицей.

Двоичный сигнал	Сигнал модифицированного кода ПИТ	
	вариант "+"	вариант "-"
1 1	+1 0	0 -1
1 0	+1 -1	+1 -1
0 1	-1 +1	-1 +1
0 0	0 +1	-1 0

Для передачи каждой из групп двоичных символов (кроме 10 и 01) предусматриваются два набора (алфавита) троичных символов. Выбор одного из них в каждой конкретной ситуации зависит от текущего значения цифровой суммы символов кода, что обеспечивает устранение постоянной составляющей из спектра сигнала и сокращение возможного числа подряд следующих нулей до двух.

Весьма перспективным следует признать использование алфавитных кодов типа  $nV_kM$ . В процессе преобразования исходного двоичного сигнала используется не один, а несколько алфавитов, а выбор в каждом конкретном случае определенного алфавита (моды) осуществляется на основе анализа некоторого числа предшествующих символов с учетом текущего значения цифровой суммы символов кодовой последовательности. Т.о., одна и та же группа исходных двоичных символов может быть передана различными группами кода  $nV_kM$ . Алфавиты подбираются так, чтобы различные моды, используемые для передачи одной и той же двоичной группы, могли отклонять цифровую сумму в сторону как увеличения, так и уменьшения. Выбирая соответствующие моды, добиваются того, чтобы цифровая сумма изменялась в достаточно узких пределах, по возможности близких к нулю. Это обеспечивает уменьшение мощности низкочастотных составляющих в спектре передаваемой последовательности.

Наибольшее распространение получили абсолютный (АБС) и относительный (ОБС) биимпульсные сигналы. При формировании АБС вместо каждой единичной посылки исходной двоичной последовательности передается элемент  $S_3(t)$ , а вместо каждой нулевой посылки исходной двоичной последовательности - элемент  $S_4(t)$ . При формировании ОБС осуществляется передача биимпульсного элемента  $S_3(t)$  или  $S_4(t)$ , одинакового с предыдущим, при появлении двоичной единицы, и биимпульса, противоположного предыдущему, при появлении двоичного нуля.

Весьма перспективными для передачи сигналов по трактам магистральных цифровых систем передачи являются коды с числом уровней больше трех, т.е. многоуровневые коды.

### 2.4. Многоуровневые коды

В случае *многоуровневых кодов*, последовательность входных битов группируется в блоки по  $\log_2 M$  бит. В результате длительность символа на выходе кодера в  $\log_2 M$  раз превышает длительность одного бита.

Каждому из  $M$  возможных блоков присваивается значение  $a_k$ . Табл. 2.2 демонстрируют возможные значения  $a_k$  для 4-х уровневый ( $M=4$ ) и 8-ми ( $M=8$ ) уровневый кодирования. 4-х уровневый код принято называть *2B1Q* кодом.

Таблица 2.2

4-х уровневый код (2B1Q)		8-ми уровневый код	
Двухбитовый блок	$a_k$	Трехбитовый блок	$a_k$
00	1	000	1
01	$1/3$	001	$5/7$
11	$-1/3$	011	$3/7$
10	-1	010	$1/7$
		110	$-1/7$
		111	$-3/7$
		101	$-5/7$
		100	-1

На рис. 2.6 показана форма сигнала на выходе 4-х уровневый кодера.

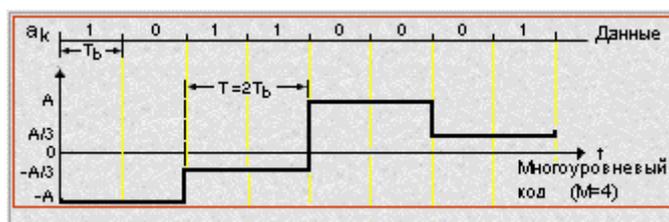


Рис. 2.6

При 4-х уровневом кодировании длительность символа в 2 раза больше длительности бита:

$$T=2T_b,$$

(2.8)

а число возможных символов составляет  $M=4$ .

При 8-ми уровневом кодировании число возможных символов  $M=8$ , а длительность каждого символа в 3 раза больше длительности бита:

$$T=3T_b.$$

(2.9)

Спектр многоуровневых сигналов совпадает со спектром, приведенным на [рис. 2.2](#) для биполярного NRZ кода. Однако, с учетом того, что в случае многоуровневых кодов длительность символа превышает длительность бита, спектр многоуровневых сигналов более компактен. Так 4-х уровневый сигнал занимает в 2 раза более узкую полосу, чем NRZ сигнал. Полоса 8-ми уровневого сигнала в 3 раза уже полосы NRZ сигнала.

Узкий спектр - основное достоинство многоуровневых кодов. Именно поэтому их широко используют на линиях связи с ограниченной полосой пропускания. В частности, в системах радиосвязи многоуровневые коды применяют при формировании радиосигналов с многопозиционными видами манипуляции (ФМ-4, ФМ-8, АФМ-16, АФМ-64 и др.).

## 2.5. Манчестерский код

В манчестерском коде, как и в NRZ/RZ кодах, длительность символа равна длительности одного бита. Однако, логическим единице и нулю соответствуют не прямоугольные, а двухполярные сигналы.

При передаче логической "1" первую половину символа формируется импульс положительной полярности, а вторую - отрицательной. При передаче логического "0", наоборот, первая половина символа представляет импульс отрицательной полярности, а вторая - положительной.

Форма сигнала на выходе кодера имеет вид, показанный на рис. 2.7.

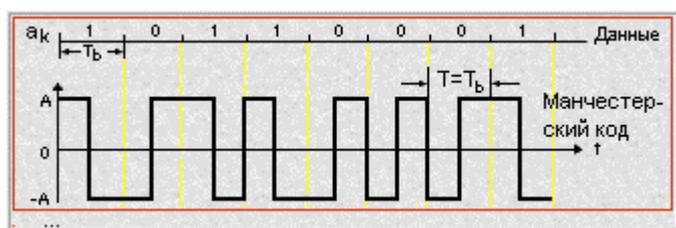


Рис. 2.7

Основное достоинство манчестерского кода - отсутствие постоянной составляющей: код имеет нулевую спектральную плотность при  $f=0$  и максимум в окрестности

$f=0.7/T$  (рис. 2.8). Такой спектр обусловлен формой элементарных сигналов, каждый из которых не содержит постоянной составляющей.

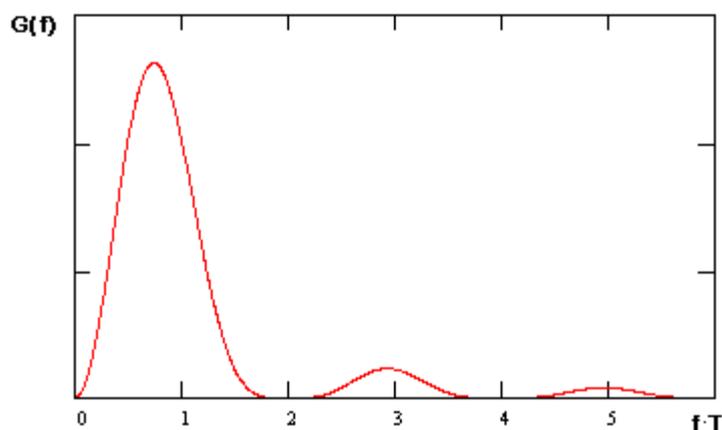


Рис. 2.8

Отсутствие постоянной составляющей облегчает построение линий связи с трансформаторами и разделительными емкостями.

К сожалению, ширина спектра манчестерского кода в 2 раза шире по сравнению с NRZ кодом. Широкий спектр - главный недостаток манчестерского кода, ограничивающий его применение системами, в которых проблема ширины полосы несущественна.

## 2.6. Код Миллера

В отличие от ранее рассмотренных видов кодирования, *код Миллера* использует запоминание значений сигнала. Его применяют в системах магнитной записи и в некоторых системах передачи данных с фазовой манипуляцией.

Для передачи данных служат два элементарных сигнала  $S_1(t)$  и  $S_2(t)$ , а так же их инвертированные варианты  $S_4(t) = -S_1(t)$  и  $S_3(t) = -S_2(t)$ . Форма этих сигналов показана на рис. 2.9.

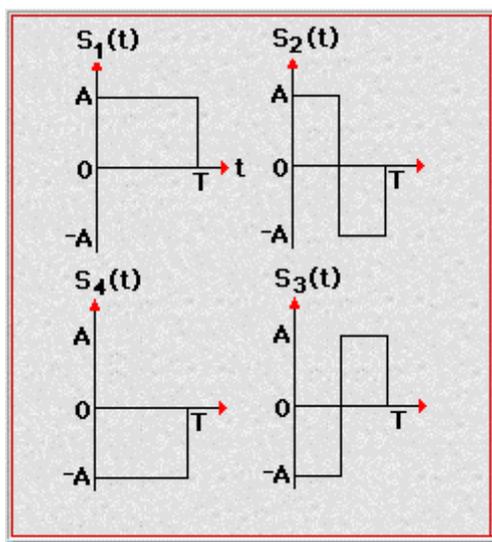


Рис. 2.9

Выбор одного из четырех сигналов зависит от того, передается в данный момент "1" или "0", а также от того, какой сигнал передавался на предыдущем этапе.

Для пояснения формирования кода Миллера воспользуемся диаграммой состояний, приведенной на рис. 2.10.

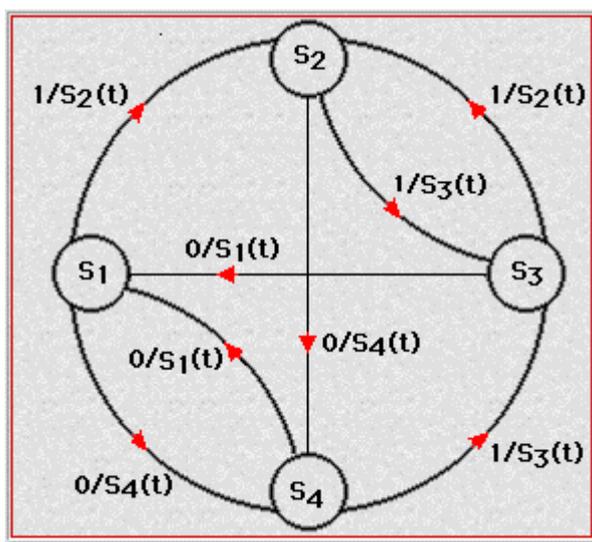


Рис. 2.10

Предположим, что на предыдущем этапе был сформирован сигнал  $S_1$ . Если теперь на вход кодера поступит логическая "1", то будет сформирован сигнал  $S_2$ . Если же поступит логический "0", то на выходе кодера появится сигнал  $S_4$ .

Если предшествующим был сигнал  $S_4$ , то за ним последует  $S_1$  при передаче "0" и  $S_3$  при передаче "1" и т. д.

Форма сигнала на выходе формирователя кода Миллера имеет вид, приведенный на рис. 2.11.

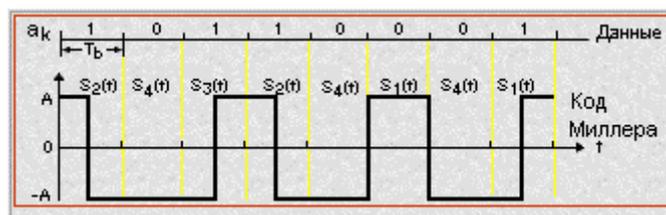


Рис. 2.11

Хотя в коде Миллера длительность символа равна длительности бита ( $T=T_b$ ), благодаря корреляции между соседними символами средняя длительность положительных и отрицательных импульсов на выходе кодера несколько больше длительности символа. Это определяет хорошие спектральные свойства кода Миллера (рис. 2.12).

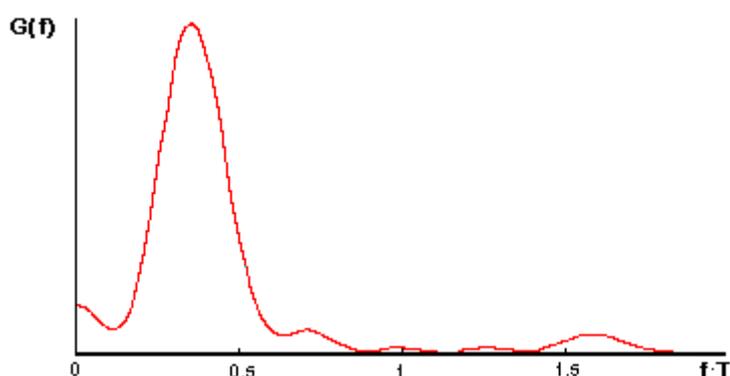


Рис. 2.12

Код Миллера имеет узкий спектр и относительно небольшую постоянную составляющую. Занимаемая им полоса частот меньше, чем полоса частот NRZ кода. Эти два параметра делают код Миллера привлекательным для использования в узкополосных каналах, которые не пропускают постоянную составляющую.

## 2. Линейное кодирование

### 2.1. Принципы линейного кодирования

В своей основе цифровое сообщение представляет собой упорядоченную последовательность символов (обычно в двоичной форме), вырабатываемых источником сообщения или кодером источника.

Сообщение преобразуется в электрический сигнал с помощью канального кодера. При этом кодер устанавливает однозначное соответствие, называемое кодом, между элементами сообщения и элементами сигнала на его выходе (кодowymi символами).

Кодер выбирает различные символы из алфавита, содержащего  $M \geq 2$  различных символов, и "выдает" символы со скоростью  $I$ .

Ограниченная полоса пропускания канала связи, помехи в нем, возможная многолучевость, а также замирания накладывают ограничения на максимальную скорость передачи информации.

При передаче в основной полосе частот, то есть в низкочастотной области (без модуляции высокочастотной несущей) канальный кодер называют линейным кодером, а сигнал на его выходе - линейным кодом.

Цифровой сигнал, передаваемый в основной полосе частот, может быть представлен как:

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g(t - kT)$$

(2.1)

Коэффициент  $a_k$  определяет  $k$ -ый символ в последовательности передаваемых по каналу связи символов. При этом значения коэффициентов принадлежат множеству из  $M$  дискретных величин. Элементарный сигнал  $g(t)$  может быть прямоугольной или иной формы в зависимости от параметров канала связи. Длительность символа  $T$  определяет символьную скорость передачи

$$I_{[\text{Бод}]} = 1/T,$$

(2.2)

оцениваемую в Бодах (числе передаваемых символов за 1 секунду).

Длительность символа  $T$  связана с длительностью одного бита передаваемой информации  $T_b$  соотношением

$$T = T_b \text{Log}_2 M.$$

(2.3)

В свою очередь, длительность бита определяет скорость передачи информации (битовую скорость)

$$I_{[\text{бит/с}]} = 1/T_b.$$

(2.4)

Данную скорость принято оценивать в бит/с.

Скорости передачи, определяемые выражениями (2.2) и (2.4), связаны соотношением

$$I_{[\text{бит/с}]} = I_{[\text{Бод}]} \text{Log}_2 M.$$

Рассматриваемые ниже *RZ* и *NRZ* коды, *многоуровневые коды*, *манчестерский код* описываются выражением (2.1) и не требуют запоминания предшествующих значений сигнала. В *коде Миллера*, также как в *TBC* и *AMI* кодах значение передаваемого символа определяется не только символом на входе в данный момент, но и предшествующим состоянием.

В табл. 2.1 представлен вид элементарных сигналов  $g(t)$  и возможные значения коэффициентов  $a_k$  для основных способов линейного кодирования, не использующих запоминание.

**Таблица 2.1**

Код	$a_k$	$g(t)$	$T$
Униполярный NRZ	0/1		$T_b$
Биполярный NRZ	-1/1		$T_b$
Униполярный RZ	0/1		$T_b$
Биполярный RZ	-1/1		$T_b$
Многоуровневый	См. табл. 2.2		$T_b \log_2 M$
Манчестерский	-1/1		$T_b$

## 2.2. NRZ код

*NRZ* (Non Return to Zero) - невозвращающийся в нуль код. NRZ код является простейшим *линейным кодом*, широко применяемым на практике. Существуют две разновидности этого кода - униполярный и биполярный NRZ коды.

В биполярном NRZ коде логической единице соответствует прямоугольный импульс положительной полярности, а логическому нулю - прямоугольный импульс отрицательной полярности. Длительность импульсов равна длительности одного бита.

Положительное или отрицательное напряжение на выходе кодера сохраняется неизменным в течении длительности символа, что и определяет термин "невозвращающийся в нуль" код.

Униполярный NRZ код отличается от биполярного, тем что логическому нулю соответствует не отрицательный импульс, а нулевое напряжение.

Форма сигнала на выходе биполярного и униполярного NRZ кодера имеет вид, приведенный на рис. 2.1.



Рис. 2.1

NRZ код - это двухпозиционный код ( $M=2$ ) и длительность символа совпадает с длительностью бита ( $T=T_b$ ).

Энергетический спектр биполярного NRZ сигнала определяется соотношением

$$G(f) = A^2 T \left( \frac{\sin \frac{2\pi f T}{2}}{\frac{2\pi f T}{2}} \right)^2,$$

(2.5)

где  $A$  - амплитуда импульса.

Зависимость спектральной плотности биполярного NRZ сигнала от частоты представлена на рис. 2.2. На оси абсцисс отложены нормированные значения частоты, причем, как уже отмечалось, в случае NRZ кода  $T=T_b$ .

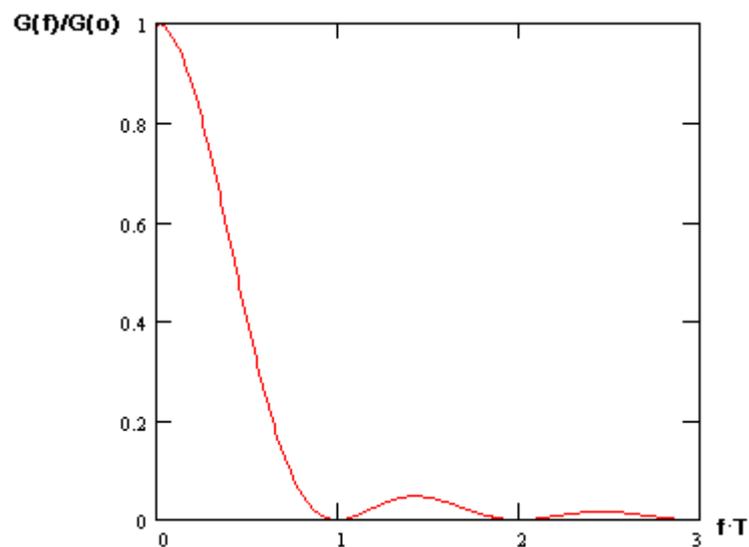


Рис. 2.2

Теоретически спектр NRZ сигнала бесконечен. Ограничение полосы при передаче по реальным каналам связи приводит к взаимному наложению сигналов различных символов, называемому межсимвольной интерференцией.

Ширину спектра NRZ, как и других цифровых сигналов, принято оценивать по ширине главного лепестка. В случае NRZ ширина главного лепестка спектра

$$B=1/T_b.$$

(2.6)

Важной особенностью спектра NRZ сигнала является конечное значение спектральной плотности на нулевой частоте. Следовательно, биполярный NRZ сигнал имеет постоянную составляющую.

Спектр униполярного NRZ кода отличается от спектра биполярного NRZ кода наличием дискретной спектральной линии на нулевой частоте.

### 2.3. RZ код

RZ (Return to Zero) - код с возвратом к нулю. Как и у NRZ кода длительность символа RZ кода равна длительности одного бита. Однако прямоугольный импульс положительной или отрицательной полярности занимает лишь половину длительности символа. Вторую половину напряжение на выходе кодера равно нулю. Именно поэтому код и назван "кодом с возвратом к нулю".

Код имеет две разновидности - биполярный RZ код и униполярный RZ код.

Униполярный RZ код отличается от биполярного тем, что логическому нулю соответствует не отрицательный импульс, а нулевое напряжение.

Подобно NRZ коду RZ код является двухпозиционным ( $M=2$ ) и имеет длительность символа, равную длительности бита ( $T=T_b$ ).

Форма сигнала на выходе биполярного и униполярного RZ кодера приведена на рис. 2.3.

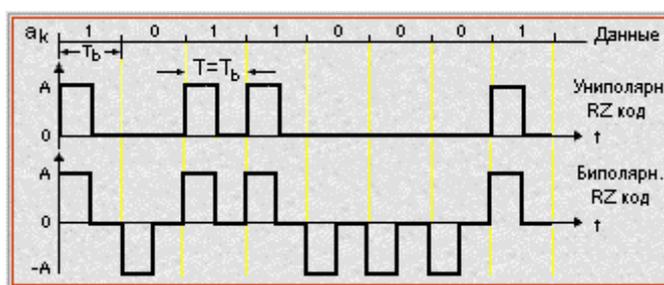


Рис. 2.3

Спектр биполярного RZ сигнала (рис. 2.4) подобен спектру биполярного NRZ и также имеет постоянную составляющую.

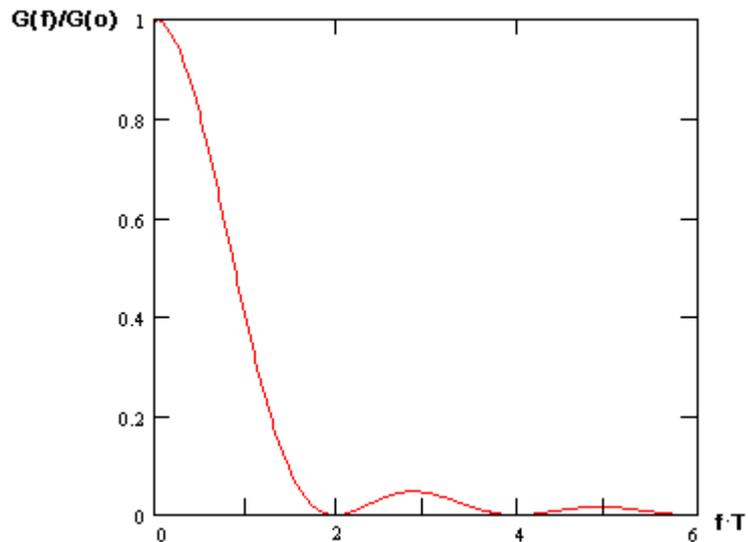


Рис. 2.4

Основное отличие спектра RZ сигнала - в два раза большая, чем у NRZ сигнала, ширина спектра при тех же значениях скорости передачи:

$$B=2/T_b.$$

(2.7)

Спектр униполярного RZ кода (рис. 2.5) содержит дискретные спектральные линии, причем не только на нулевой частоте, но и на нечетных гармониках тактовой частоты. Наличие в спектре составляющей с тактовой частотой облегчает построение системы тактовой синхронизации в приемнике.

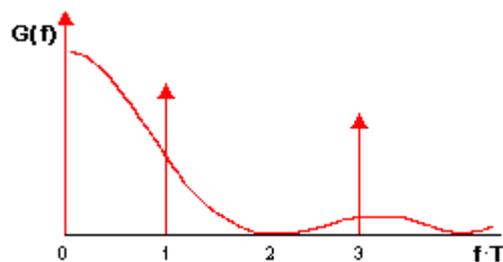


Рис. 2.5

## 2.7. Псевдотрочный код TBC

Энергетический спектр NRZ, RZ и 2B1Q кодов содержит постоянную составляющую. Следовательно, такие линейные коды нельзя использовать в каналах, содержащих трансформаторы или разделительные емкости, т. е. в каналах со связью по перемен-

ному току. Использование таких кодов возможно только в каналах со связью по постоянному току.

Обеспечить отсутствие постоянной составляющей можно путем соответствующего выбора формы импульсов. Примером такого кода является манчестерский код. Он, однако, обладает очень широким спектром.

Другой метод обнуления постоянной составляющей заключается в введении корреляции между передаваемыми символами. Для того, чтобы при введении избыточности не увеличить символьную скорость, увеличивают количество уровней сигнала.

Примером такого кода может служить псевдотроичный код. Этот код принято обозначать латинскими буквами **TBC** (Twinned Binary Code).

Хотя этот код практически не используют, он важен для понимания идеи. Построенный на основе TBC перемежающийся биполярный код, который будет рассмотрен ниже, широко применяют в системах электросвязи.

Символы на выходе TBC кодера ( $a_k$ ) образуют по следующему правилу

$$a_k = b_k - b_{k-1}$$

(2.10)

где биты  $b_k$  передаваемой информационной последовательности принимают значения "0" и "1".

Легко удостовериться, что символы  $a_k$  имеют алфавит, состоящий из трех символов, которые в дальнейшем будем обозначать как "+1", "0" и "-1", непосредственно ассоциируя их с действительными амплитудами импульсов. Правило образования символов на выходе TBC кодера можно ввести к следующему: при положительном переходе от "0" к "1" формируется "+1"; при отрицательном переходе от "1" к "0" формируется "-1"; и при отсутствии перехода - "0". Форма сигнала на выходе TBC кодера имеет вид, приведенный на рис. 2.13.

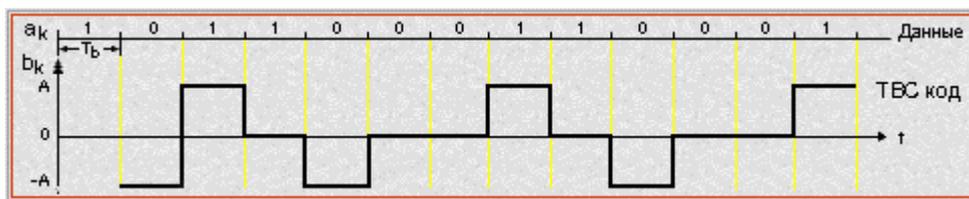


Рис. 2.13

Поскольку символ на выходе TBC кодера передает только один бит информации, код является *псевдотроичным*.

Структура TBC кодера и декодера показана на рис. 2.14.

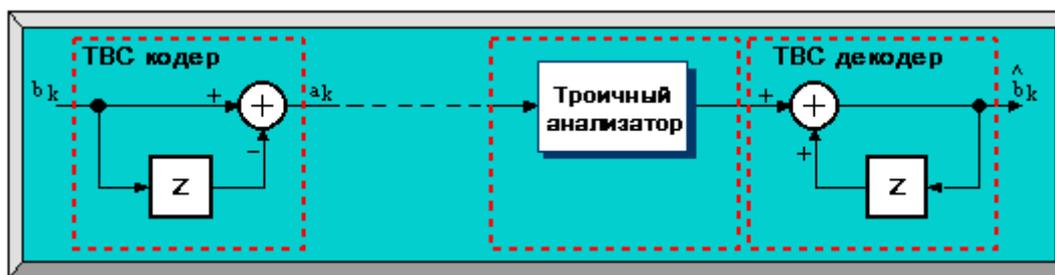


Рис. 2.14

На этом рисунке модуль "z" представляет оператор задержки на длительность одного бита. Приемник содержит троичный анализатор с порогами принятия решения  $-1/2$  и  $+1/2$ .

Можно показать, что если среднее количество нулей в потоке  $b_k$  на входе ТВС кодера равно среднему количеству единиц, то равны средние количества символов "-1" и "+1" в потоке  $a_k$  на выходе кодера. Кроме того, если во входном потоке  $b_k$  присутствуют длинные цепочки из "0" или "1", то на выходе кодера они будут преобразованы в длинные цепочки "0". В результате, постоянная составляющая сигнала на выходе ТВС кодера отсутствует.

При ТВС кодировании возможно размножение ошибок при приеме длинных последовательностей нулей или единиц. Пусть, как показано в табл. 2.3, длинная последовательность нулей на входе сменяется последовательностью единиц и наоборот

Таблица 2.3

$b_k$	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
$a_k$	-	0	0	0	+1	0	0	0	0	0	0	0	-1	0	0	0	0	0	+1	0	0

Видно, что единственное различие в выходном сигнале при приеме последовательностей нулей и единиц - это полярность сигнала в момент перехода от одной последовательности к другой. Ошибка при приеме в этот момент приведет к ошибочному приему всей последовательности одинаковых символов.

Размножение ошибок - существенный недостаток псевдотроичного кодирования и именно поэтому ТВС код в чистом виде не используют.

## 2.8. Линейный код АМІ

Для устранения размножения ошибок при использовании псевдотроичного кода можно использовать предварительное дифференциальное кодирование, получая в результате линейный код, известный как перемежающийся биполярный код, обозначаемый также латинскими буквами АМІ (Alternate Mark Inversion).

Как было отмечено выше, размножение ошибок при ТВС кодировании возникают из-за неоднозначности при передаче длинных последовательностей нулей и единиц. При этом вместо последовательности нулей на выходе декодера приемника может появиться последовательность единиц, и наоборот. Дифференциальное предкодирование, выполняемое перед линейным кодированием устраняет эту неоднозначность.

Построение предкодера и посткодера иллюстрирует рис. 2.15.

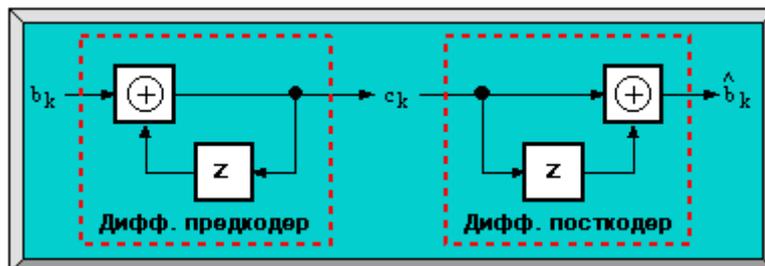


Рис. 2.15

Блок "+" на рисунке выполняет операцию сложения по модулю два. Если на входах действуют двоичные сигналы, сумматор по модулю два эквивалентен схеме "исключающее ИЛИ", и его работу можно описать с помощью следующей таблицы истинности.

Таблица 2.4

Входы		Выход
0	0	0
0	1	1
1	0	1
1	1	0

Работа предкодера описывается соотношением

$$c_k = b_k \oplus c_{k-1}$$

(2.11)

где  $c_k$  - символ на выходе предкодера, принимающий, подобно  $b_k$ , значения "0" и "1".

Предкодер называют дифференциальным, поскольку его можно описать следующим образом: нулевой бит на входе не вызывает изменения выходного бита, а единичный бит на входе ведет к изменению бита на выходе (либо с единицы на нуль, либо с нуля на единицу).

Дифференциальный посткодер восстанавливает исходный бит, поскольку в соответствии с рис. 2.15, сигнал на выходе

$$b_k \oplus c_{k-1} \oplus c_{k-1} = b_k$$

(2.12)

В выражении (2.12) использован тот факт, что

$$c_{k-1} \oplus c_{k-1} = 0$$

Полный АМІ кодер показан на рис. 2.16. Он представляет собой комбинацию дифференциального предкодера и псевдотроичного кодера. В свою очередь, АМІ декодер - это комбинация псевдотроичного декодера и дифференциального посткодера.



Рис. 2.16

Обратите внимание, что в кодере и посткодере использованы два вида сумматоров - обычный сумматор и сумматор по модулю два. Для ясности использованы два элемента задержки, тогда как, фактически, достаточно одного.

Использование дифференциального предкодирования перед ТВС кодированием приводит к тому, что при действии на входе кодера длинной последовательности единиц, на выходе кодера будет получена последовательность чередующихся "+1" и "-1". При действии же на входе кодера длинной последовательности нулей, на выходе кодера будет также последовательность "0". Это иллюстрирует табл. 2.5.

Таблица 2.5

$b_k$	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
$c_k$	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1
$a_k$	-	0	0	0	+1	-1	+1	-1	+1	-1	+1	-1	0	0	0	0	0	0	+1	-1	+1

Таким образом, в отличие от ТВС кодирования, сигналы на выходе АМІ кодера при действии длинных последовательностей нулей и единиц существенно различны. Это исключает размножение ошибок при АМІ кодировании.

При вычислении энергетического спектра АМІ кода используют аппарат теории марковских цепей. Для прямоугольного импульса  $g(t)$  с амплитудой  $A$  имеем

$$G(f) = A^2 T \left( \frac{\sin \frac{2\pi f T}{2}}{\frac{2\pi f T}{2}} \right)^2 2p(1-p) \frac{1 - \cos 2\pi f T}{1 + (1-2p)^2 - 2(1-2p)\cos 2\pi f T}$$

(2.13)

где  $p$  - вероятность того, что бит единичный.

На рис. 2.17 показана частотная зависимость энергетической плотности АМІ сигнала  $G(f)$  при равновероятном появлении нулей и единиц на входе кодера ( $p=0.5$ ).

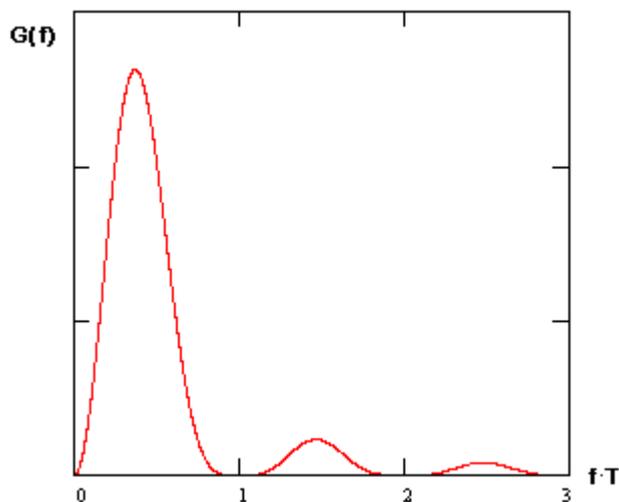
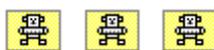


Рис. 2.17



## 2.9. Сравнение свойств линейных кодов NRZ, АМІ и 2В1Q

При сравнении биполярного NRZ, АМІ и 2В1Q кодов обратим внимание на следующие их свойства:

- ширина полосы;
- наличие постоянной составляющей;
- чувствительность к полярности сигнала;
- помехоустойчивость.

Для удобства сравнения спектров рассматриваемых сигналов представим их на одном рисунке.

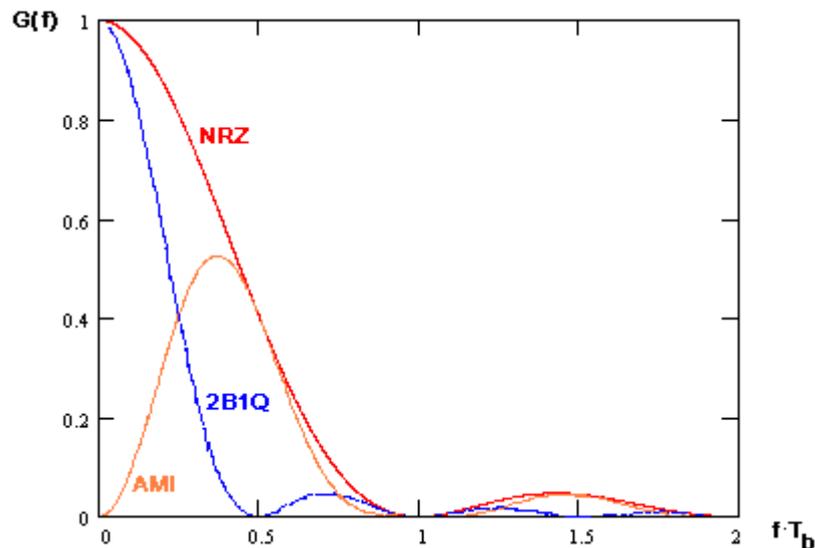


Рис. 2.18

При сравнении кодов по ширине полосы принято оценивать их спектральную эффективность, под которой понимают отношение

$$\gamma = \frac{I}{B}$$

(2.14)

где  $I$  - скорость передачи информации, измеряемая в бит/с, а  $B$  - ширина главного лепестка спектра, измеряемая в Гц.

Битовая скорость зависит, в свою очередь, от символьной скорости и от числа бит ( $k$ ), приходящихся на один символ.

Ширина спектра зависит от символьной скорости. В зависимости от формы импульса ее значения лежат в пределах от  $0.5/T$  до  $1/T$ , где  $T$  - длительность символа. При прямоугольной форме импульсов  $g(t)$  ширина главного лепестка  $B=1/T$ .

При NRZ и AMI кодировании длительность символа совпадает с длительность бита ( $k=1$ ) и, следовательно,  $I=1/T$ . При прямоугольной форме импульса

$$\gamma_{NRZ} = \gamma_{AMI} = \frac{1/T}{1/T} = 1$$

(2.15)

При 2B1Q кодировании один символ содержит информацию о двух битах ( $k = 2$ ) и поэтому  $I=2/T$ . При прямоугольной форме импульса  $B=1/T$ , и, следовательно,

$$\gamma_{2B1Q} = \frac{2/T}{1/T} = 2$$

(2.16)

Из сказанного следует, что с помощью кода 2B1Q можно передавать информацию в два раза быстрее, чем кодами AMI и NRZ в одной и той же полосе частот.

Это обстоятельство говорит в пользу кода 2B1Q. Однако, как показано на рис. 2.18, спектр сигнала 2B1Q (также как и NRZ) содержит постоянную составляющую и частоты близкие к ней. Наличие постоянной составляющей затрудняет использование 2B1Q и NRZ кодов на линиях связи с трансформаторами и разделительными емкостями.

Спектр сигнала AMI не содержит постоянной составляющей и это является существенным преимуществом этого вида кодирования.

Кроме того, коды NRZ и 2B1Q чувствительны к полярности сигнала. При неправильном подключении линии сигналы будут приняты ошибочно. Код AMI не чувствителен к полярности сигнала, что связано с дифференциальным предкодированием, используемым при AMI.

Отсутствие постоянной составляющей и нечувствительность к полярности сигнала определяют широкое использование AMI кода в качестве линейного кода.

Сравним помехоустойчивость основных видов кодирования. При этом будем считать, что канал не вносит искажений и принимаемый сигнал свободен от межсимвольной интерференции.

Будем также считать, что в канале действует аддитивный белый шум с нулевым средним.

При использовании согласованного фильтра в приемнике вероятность ошибочного приема одного бита переданной информации в случае биполярного NRZ кода составит

$$P_e = 1 - F(\sqrt{2SNR})$$

(2.17)

где

$$SNR \triangleq \frac{E}{N_0}$$

- отношение средней энергии  $E$ , приходящейся на один бит информации, к односторонней спектральной плотности шума  $N_0$ . При приеме цифровых сигналов это отношение принято называть отношением сигнал/шум.

В выражение (2.17) входит функция Лапласа

$$F(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\alpha} \exp\left\{-\frac{\lambda^2}{2}\right\} d\lambda$$

(2.18)

При 2B1Q коде расстояние между уровнями сигнала в 3 раза меньше, чем при NRZ. В результате вероятность ошибочного приема символа 2B1Q сигнала выше и определяется приближенным соотношением

$$P_s \approx 1.5 \left(1 - F(\sqrt{0.5SNR})\right)$$

(2.19)

в котором под  $SNR$  понимается средняя энергия одного символа 2B1Q сигнала, которая в два раза больше средней энергии, приходящейся на один бит.

С некоторым приближением можно считать, что средняя вероятность ошибочного приема одного бита информации при 2B1Q составляет

$$P_e \cong 0.5 P_s$$

(2.20)

На рис. 2.19 показана зависимость вероятности ошибки на бит  $P_e$  от отношения сигнал/шум  $SNR$  при оптимальном приеме сигналов NRZ и 2B1Q. Вероятность ошибки при АМІ лежит между соответствующими значениями вероятности для NRZ и 2B1Q.

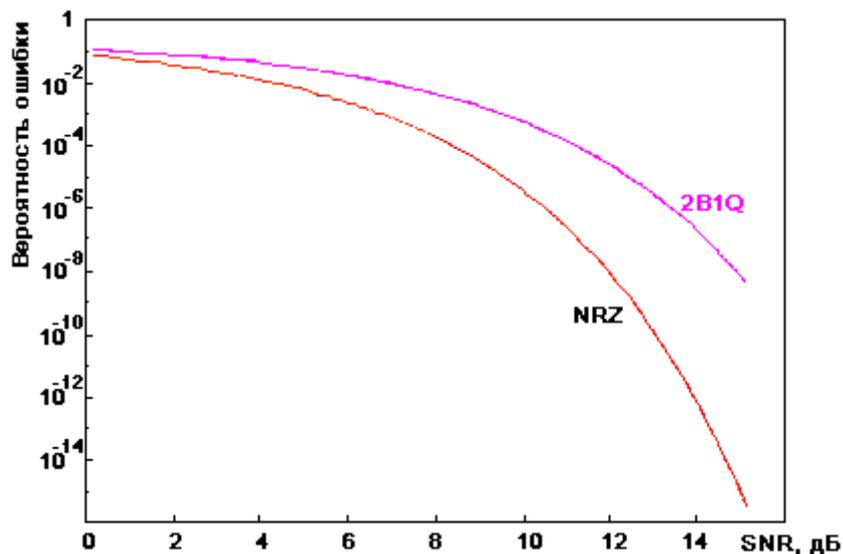


Рис. 2.19

Физически наибольшая помехоустойчивость NRZ кодирования объясняется максимальным расстоянием между возможными уровнями сигнала, равным удвоенному значению амплитуды. При AMI кодировании это расстояние в 2 раза, а при 2B1Q кодировании в 3 раза меньше.

### ***1.3. Задание к работе и порядок выполнения***

- 1.3.1. Запустить программу Code.exe.
- 1.3.2. Выбрать тип кодовой последовательности.
- 1.3.3. Построить кодовую последовательность.
- 1.3.4. Вычислить параметры кодовой последовательности.
- 1.3.5. Произвести оценку корреляционных и спектральных свойств.
- 1.3.6. Выбрать другой тип кодовой последовательности. Повторить п.1.3.3-1.3.5. не менее чем для 6 типов.
- 1.3.7. Повторить п. 1.3.2.- 1.3.6. совместно с преподавателем, после чего при правильных построениях работа считается сданной.

### ***1.4. Контрольные вопросы***

1. Почему необходимо линейное кодирование.
2. Перечислите и объясните параметры кодовой последовательности.
3. Корреляционные свойства кодовых последовательностей.
4. Спектральные свойства кодовых последовательностей.

## Лабораторная работа №2

### ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

**Цель работы:** Изучение и исследование свойств полиномиального кодирования и кодов Хэмминга.

#### *2.1. Задание к работе и порядок выполнения*

2.1.1. Запустить и изучить программу esclabs.exe.

2.1.2. Построить полиномиальный код на основе задания  $n$ ,  $k$  и порождающего полинома.

2.1.3. Произвести кодирование информационных слов умножением на порождающий полином.

2.1.4. Произвести наложение ошибок в кодовых словах.

2.1.5. Произвести декодирование информационных слов без ошибок и при наложении ошибок.

2.1.6. Построить код Хэмминга на основе задания  $n$ ,  $k$ ,  $m$  и матрицы  $H$ .

2.1.7. Произвести кодирование информационных слов умножением на порождающий полином.

2.1.8. Произвести наложение ошибок в кодовых словах.

2.1.9. Произвести декодирование информационных слов без ошибок и при наложении ошибок.

2.1.10. Произвести действия по п. 2.1.2. – 2.1.9 без программы esclabs.exe (вручную) и сравнить полученные результаты, записать результаты в отчет.

#### *2.2. Контрольные вопросы*

1. Дать определение полиномиальных кодов.
2. Пояснить получение кодовой последовательности по известному полиному.
3. Что такое синдром и как он определяется.
4. Основные характеристики кодов Хэмминга.
5. Как определяется порождающая матрица.
6. Как определяется проверочная матрица.
7. Как определяется синдром при кодах Хэмминга.

## ЛАБОРАТОРНАЯ РАБОТА №3

### МОДЕЛИРОВАНИЕ СХЕМЫ КОДЕРА И ДЕКОДЕРА

**Цель работы:** Изучение устройства и работы кодеров и декодеров при полиномиальном помехоустойчивом кодировании.

#### *3.1. Тип порождающего (образующего) полинома*

Порождающий (образующий) полином степени  $r$  находится по таблице неприводимых полиномов из программы esclabs.exe лабораторной работы №2, например:

$$g(x) = x^3 + x + 1$$

#### *3.2. Структурная схема кодера*

Схема кодера – декодера состоит из следующих элементов:

1. Генератор тактовых импульсов.
2. Устройство формирования информационной группа на сдвиговом регистре и логическом элемента ИЛИ.
3. Кодер на сдвиговом регистре и логических элементах Искл-ИЛИ (сложение по модулю 2).

Кодер осуществляет умножение информационной группы на полином.

3. Декодер на сдвиговом регистре и логических элементах Искл-ИЛИ (сложение по модулю 2).

Декодер осуществляет деление на полином (множение на проверочный полином).

Пример моделирования показан на рис.3.1.

Информационная группа состоит из 5 двоичных разрядов и равна 00010011. Три старших разряда являются контрольными и заполняются декодером. Код на выходе декодера после умножения на полином:

$$g(x) = x^3 + x + 1$$

равен 10110101. Код на выходе декодера после умножения на проверочный полином :

$$f(x) = x^7 + x^4 + x^2 + x + 1$$

равен 00010011 исходной информационной группе.

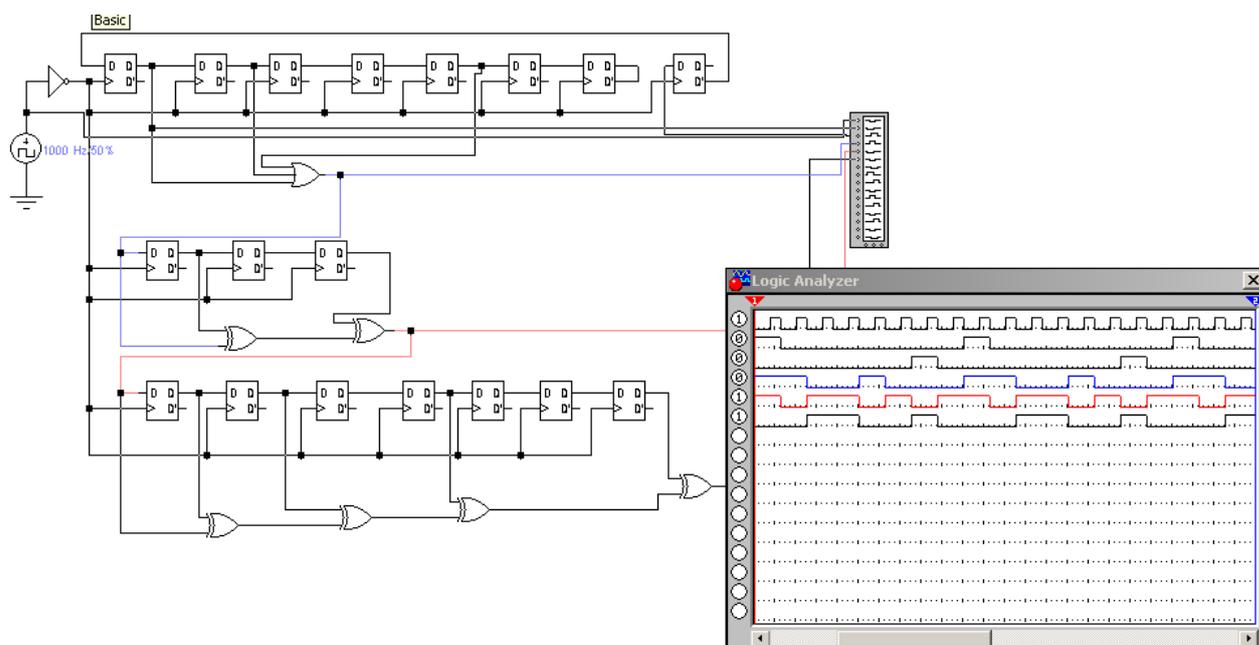


Рис.3.1.Схема моделирования кодера – декодера.

Моделирование работы кодера и декодера производится в среде Work-Bench. Степень полинома рекомендуется брать 3 – 7, не более.

Отчет должен содержать схемы устройств и результаты моделирования.

### 3.3. Задание на лабораторную работу

Задание состоит из разрядности информационной группы, выдается преподавателем. Полином выбирается студентом самостоятельно.

После получения результатов моделирования необходимо проверить работу декодера при задании ошибки в принятом коде.

#### 3.3. Контрольные вопросы

1. Объясните назначение и принцип работы каждого из рассмотренных цифровых устройств по их схемам.
2. Перечислите и поясните основные функции кодера – декодера.



## СПИСОК ЛИТЕРАТУРЫ

1. Лебедько Е.Г. Теоретические основы передачи информации [Электронный ресурс]: монография/ Е. Г.Лебедько.- Электрон. Дан. - Санкт.-Петербург : Лань, 2011. - 352 с — Режим доступа: <https://e.lanbook/book/1543>
2. Крухмалев, Владимир Васильевич. Цифровые системы передачи: учебное пособие / Крухмалев В. В., Гордиенко В. Н., Моченов А. Д.; под ред. А. Д. Моченова. - 2-е изд., перераб. и доп. - Москва: Горячая линия-Телеком, 2012. - (Специальность). - 376 с. –(10)
3. Крухмалев, Владимир Васильевич. Цифровые системы передачи: учебное пособие для вузов / Крухмалев В. В., Гордиенко В. Н., Моченов А. Д. - Москва: Горячая линия-Телеком, 2007. - (Специальность). - 351 с.: ил.(2)
- 4.Иванов В.И. Цифровые и аналоговые системы передачи. Учебник для вузов. 2005, -232с. (35)
5. Иванов В.И. Цифровые и аналоговые системы передачи. Учебник для вузов. 2003, -232с. (5)
6. Крухмалев В.В. Цифровые системы передачи.: Учебн. пособие для вузов.-М. Горячая линия, Телеком, Пит., 2007, - 351с.(2)

---

Учебное издание

### ЦИФРОВЫЕ СИСТЕМЫ ПЕРЕДАЧИ

Сборник лабораторных работ

Составитель **Горохин** Валерий Николаевич

Корректор

Подписано в печать 30.09.2003. Формат 60x84/16.

Бумага писчая. Печать трафаретная.

Усл. печ. л. 1,40. Уч.-изд. л. 1,10. Тираж 50 экз. Заказ

Ульяновский государственный технический университет,  
432027, Ульяновск, Сев. Венец, 32.

Типография УлГТУ, 432027, Ульяновск, Сев. Венец, 32.