

LAB WORK 03. СХЕМЫ ЛИНЕЙНОГО И БЛОКОВОГО КОДИРОВАНИЯ.

1. ЦЕЛЬ РАБОТЫ.

Получить представление о различных способах представления информации в линиях связи при телекоммуникации данных, о проблемах при передаче данных в линиях и о решении этих проблем.

2. ЗАДАНИЕ.

2.1. Сформируйте индивидуальный вариант исходного сигнала.

a) Запишите свою фамилию буквами английского алфавита. Должно быть не менее 5 букв, если не хватает, то добавьте нужное количество букв из имени.

Например, для Ли Юрий будет LIYURIY.

b) Замените первые 5 буквы их порядковыми номерами в алфавите, записав номера как двухразрядные десятичные числа.

Например, 12 09 25 21 18.

c) Переведите каждое десятичное число в 5 разрядное бинарное число отбросьте последнюю бинарную цифру

Например, 01100 01001 11001 10101 10010.

d) Полученная последовательность из 24 битов будет вашим вариантом исходной последовательности для кодирования (исходным потоком бинарных данных).

Например, исходный сигнал: 011000100111001101011001.

2.2. Создайте 10 диаграмм иллюстрирующих 10 схем линейного и блокового кодирования.

3. ОТЧЁТ.

3.1. Работа сдаётся в электронном виде на почту преподавателя (см. бланк отчета). Для графического представления диаграмм рекомендуется использовать электронные таблицы MS Excel (см. бланк отчета).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	Report for Lab Work 04: Схемы линейного и блочного кодирования.																						
2																							
3	Student Name Surname	Student ID	Date																				
4	Valentin Kim	gr.3461	20.10.2019																				
5												0	1	0	1	1	0						
6																							
7																							
8	2.1. Create an individual variant of the original signal																						
9	a) Kim Valentin = KIMVA																						
10	b) 11 09 13 22 01																						
11	c) 01011 01001 01101 10110 00001																						
12	d) 010110100101101101100000																						
13																							
14	2.2. Create 10 diagrams illustrating various																						
15	line coding schemes for your original signal.																						
16																							
17																							
18	2.2.1. NRZ-I																						
19												0	1	0	1	1	0						
20	Polar Coding Non-Return-to-Zero Inverse																						
21																							
22	1 forces a low level;																						
23	0 forces a high level.																						
24																							
25																							
26																							
27																							
28																							
29	2.2.2. NRZ-M																						
30												0	1	0	1	1	0						

3.2. Оценка за выполнение на 100%: необходимо иметь правильно сформированную исходную 24-битовую последовательность и правильно сделанные все 10 вариантов схем кодирования.

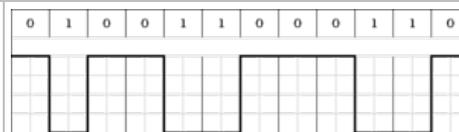
4. РЕКОМЕНДАЦИИ.

4.2. Рекомендации по созданию диаграмм иллюстрирующих различные схемы линейного и блокового кодирования бинарного потока в телекоммуникации данных.

4.2.1. NRZ-I. Unipolar None-Return-to-Zero Inverse Coding

Для передачи единиц и нулей используются два устойчиво различаемых потенциала:

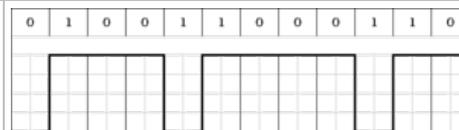
- биты 0 представляются значением $U(V)$,
- биты 1 представляются нулевым напряжением $0(V)$.



4.2.2. NRZ-M. Потенциальное кодирование без возврата к нулю Mark.

В коде NRZ-M также используется 2 потенциала, но его текущее значение зависит от предыдущего:

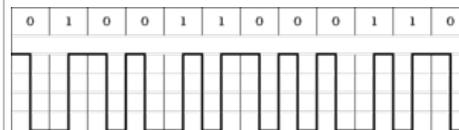
- при передаче единицы происходит смена потенциала сигнала,
- при передаче нуля изменение напряжения не производится.



4.2.3. Biphasic-I. Манчестерское кодирование по стандарту IEEE 802.3.

При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала в середине каждого такта:

- 1 кодируется перепадом от низкого уровня к высокому (вверх),
- 0 кодируется обратным перепадом (вниз).

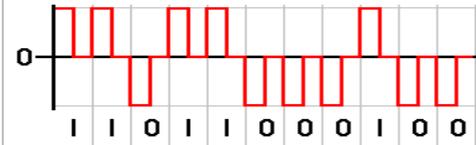


4.2.4. RZ-3. Трёхуровневое кодирование с возвратом к нулю RZ.

В этом трехуровневом коде после значащего уровня сигнала в первой половине передаваемого бита информации следует возврат к нулевому уровню.

Переход к нему происходит в середине бита:

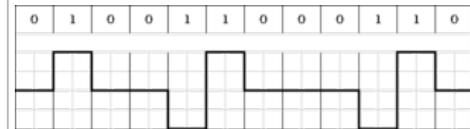
- логическому 0 соответствует положительный импульс,
- логической 1 — отрицательный.



4.2.5. AMI. Биполярный код AMI.

AMI-код (Alternate Mark Inversion) использует следующие представления битов:

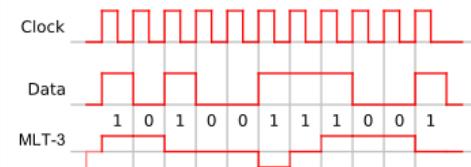
- 0 представляются нулевым напряжением (0 В);
- 1 представляются поочередно значениями -U или +U (В).



4.2.6. MLT-3. Трёхуровневое кодирование.

MLT-3 (Multi-Level Transition - 3) — схож с кодом NRZ-M, но имеет три уровня сигнала.

- Единице соответствует переход с одного уровня сигнала на другой, причем изменение уровня сигнала происходит последовательно с учетом предыдущего перехода.
- При передаче «нуля» сигнал не меняется.

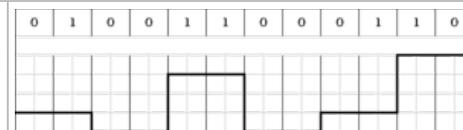


4.2.7. 2B/1Q. Четырёхуровневое кодирование.

Передаёт пару бит за один тактовый интервал.

Каждой возможной паре сигналов в соответствие ставится свой уровень из четырёх возможных уровней потенциала:

00 это $-U$; 01 это $-U/2$; 11 это $+U/2$; 10 это $+U$.



4.2.8. B8ZS - Bipolar with 8-Zeros Substitution (скремблирующий код).

B8ZS исправляет последовательности из 8 нулей. Показан пример для кодов AMI, B8ZS, HDB3.

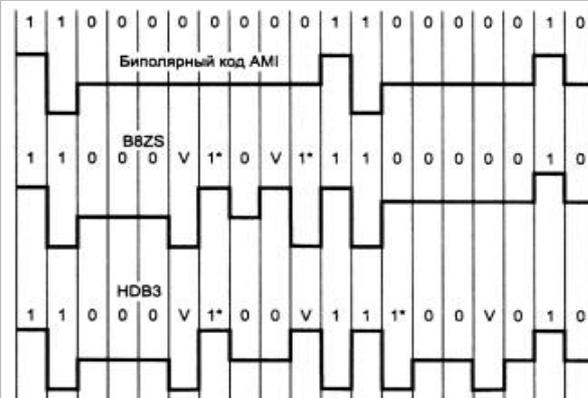
Если есть подряд 8 нулей, то вместо них вставляется следующие 8 сигналов: 0-0-0-V-1*-0-V-1*.

- V - сигнал единицы, запрещенной для данного такта полярности, то есть сигнал, не изменяющий полярность предыдущей единицы;
- 1* - сигнал единицы корректной полярности, но, в исходном коде в этом такте была не 1, а 0.

В результате на 8 тактах приемник наблюдает 2 искажения – считается маловероятным, что это случилось из-за шума на линии или других сбоев передачи.

Поэтому приемник считает такие нарушения кодировкой 8 последовательных нулей и после приема заменяет их на исходные 8 нулей.

Код B8ZS построен так, что его постоянная составляющая равна нулю при любых последовательностях двоичных цифр.



Если в вашей исходной последовательности не оказалось восьми нулей подряд, то замените биты с 10 по 17 нулями и проведите B8ZS кодирование.

Например,

011000100**11100110**1011001
011000100**00000000**1011001

4.2.9. HDB3 - High-Density Bipolar 3-Zeros (скремблирующий код).

HDB3 исправляет любые четыре подряд идущих нуля в исходной последовательности, см. рис.

Правила формирования кода HDB3 более сложные, чем кода B8ZS.

Каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал V .

Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах.

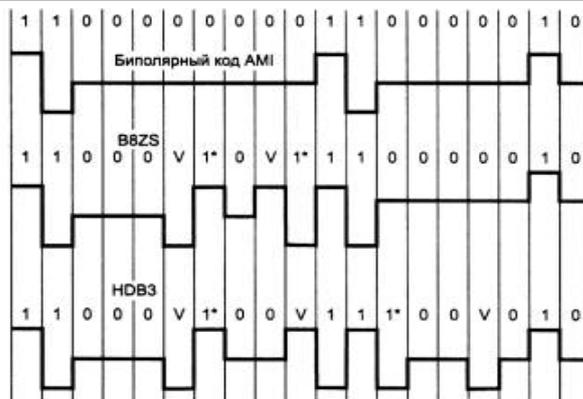
Кроме того, для замены используются два образца четырех-тактовых кодов:

- если перед заменой исходный код содержал нечетное число единиц, то используется последовательность $000V$;
- если число единиц было четным - последовательность 1^*00V .

Здесь

V - сигнал единицы запрещенной полярности;

1^* -сигнал единицы корректной полярности, но заменившей 0 в исходном коде.



Если в вашей исходной последовательности не оказалось четырех нулей подряд по 2 раза, то замените биты с 5 по 8 и с 14 по 17 нулями и проведите HDB3 кодирование.

Например, это

0110**0010**01110**0110**1011001

заменяем на это

0110**0000**01110**0000**1011001

4.2.10. 4B/5B. Избыточный код

Пять проблем решаемых схемой кодирования в линии:

- широкая полоса частот,
- наличие составляющей постоянного тока,
- потеря синхронизации,
- контроль/исправление ошибок,
- сложность реализации.

Избыточные коды основаны на разбиении исходной последовательности бит на порции, которые называют символами. Затем каждый исходный символ заменяется на новый, который имеет большее количество бит, чем исходный.

Например, логический код 4B/5B (используемый в технологиях FDDI и Fast Ethernet) заменяет исходные символы длиной в 4 бита на символы длиной в 5 бит.

Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы - только 16.

Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат большого количества нулей, а остальные считать запрещенными кодами (code violation).

Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Если приемник принимает запрещенный код, значит, на линии произошло искажение сигнала.

Соответствие исходных и результирующих кодов 4B/5B представлено ниже.

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Код 4B/5B затем передается по линии с помощью физического кодирования по одному из методов потенциального кодирования, чувствительному только к длинным последовательностям нулей, например, AMI.

Символы кода 4B/5B длиной 5 бит гарантируют, что при любом их сочетании на линии не могут встретиться более трех нулей подряд.