КАНАЛЬНОЕ КОДИРОВАНИЕ

4.0.1.1

Кодирование на канальном уровне (канальное кодирование) призвано решать две фундаментальные задачи:

1. Адаптировать битовые последовательности к возможностям физического уровня с целью обеспечения или улучшения требующихся технических характеристик. Лучше всего это назвать *линейным кодированием* (line encoding), где слово «линейное» происходит от понятия физической линии.

2. Обеспечить проверку целостности данных и, по возможности, восстановление ошибочных битов. Лучше всего это назвать *помехоустойчивым кодированием* (antinoise encoding).

4.0.1.2

Следует отметить, что терминологически линейное кодирование имеет и альтернативный смысл, происходящий от математического понятия линейных функций.

В первом подразделе в этот термин будет вкладываться физический смысл, во втором -- математический.

4.0.1.3

Не следует путать линейное кодирование с модуляцией, выполняемой на физическом уровне. Несмотря на то, что часто эти процессы связаны неразрывно, линейное кодирование все-таки следует рассматривать как надстройку над модуляцией.

4.0.1.4

Фактор помех, если под помехами понимать различные электромагнитные наводки, в КС безусловно учитывают. Но борьба с помехами -- это лишь частные случаи обеих задач (даже вторая задача порождена не только воздействием на канал наводок).

ЛИНЕЙНОЕ КОДИРОВАНИЕ

4.1.1.1

Одной из основных предпосылок для разработки линейных кодов, является проблема, проявляющаяся во многих системах передачи цифровой (не только) информации, известная как *девиацией несущей* (carrier deviation).

Очевидно, передатчик и приемник должны работать на одной частоте. В большинстве случаев, передатчик и приемник имеют разные источники синхронизации. При этом тактовые генераторы далеко не идентичны.

Если состояние линии очень долго не изменяется, что происходит при передаче очень длинных нулевых либо единичных последовательностей с использованием классической амплитудной модуляции цифровых цепей (логический ноль соответствует земле, а логическая единица некоторому положительному потенциалу относительно земли), то приемнику «цепляться не за что». В результате накапливаются фазовые сдвиги, что в конце концов приводит к возникновению ошибок.

Современная схемотехническая база для борьбы с девиацией несущей имеет в распоряжении блок ФАПЧ (фазовой автоподстройки частоты), позволяющий автоматически подстраивать тактовый генератор приемника к тактовому генератору передатчика. Наиболее близкий англоязычный термин -- PLL (Phased-Locked Loop).

4.1.1.2

Все линейные коды, в той или иной степени, направлены на преобразование битовых последовательностей, чтобы в линии постоянно происходили изменения. В том числе, за счет равномерного распределения нулей и единиц.

4.1.2.1

Шесть факторов, влияющих на классификацию линейных кодов:

1. Кодирование уровнями либо переходами.
2. Наличие инвертирования.
3. Однополярность либо многополярность.
4. Наличие так называемого «возврата к нулю».
5. Наличие самосинхронизации.
6. Наличие перестановки или подмены битов.

4.1.2.2

Что такое самосинхронизация?

4.1.2.3

Для изучения в рамках данной дисциплины выбраны следующие основные группы кодов:

1. NRZ (Non-Return-to-Zero) codes -- коды без возврата к нулю.
2. RZ (Return-to-Zero) codes -- коды с возвратом к нулю.
3. Manchester codes -- манчестерские коды.
4. MLT (Multi-Level Transmit) codes -- многоуровневые коды.
5. Block codes -- блочные коды.

NRZ-коды выражаются в изменении уровней между тактами.

В простейших случаях, логические уровни в исходной последовательности не преобразуются совсем либо инвертируются.

Более сложными случаями являются space и mark. При space-варианте ноль во входной последовательности кодируется сменой текущего уровня в выходной, а единица -- сохранением текущего уровня. При mark-варианте, наоборот, единицы в исходной последовательности приводят к переключению уровней. Начальное состояние значения не имеет.

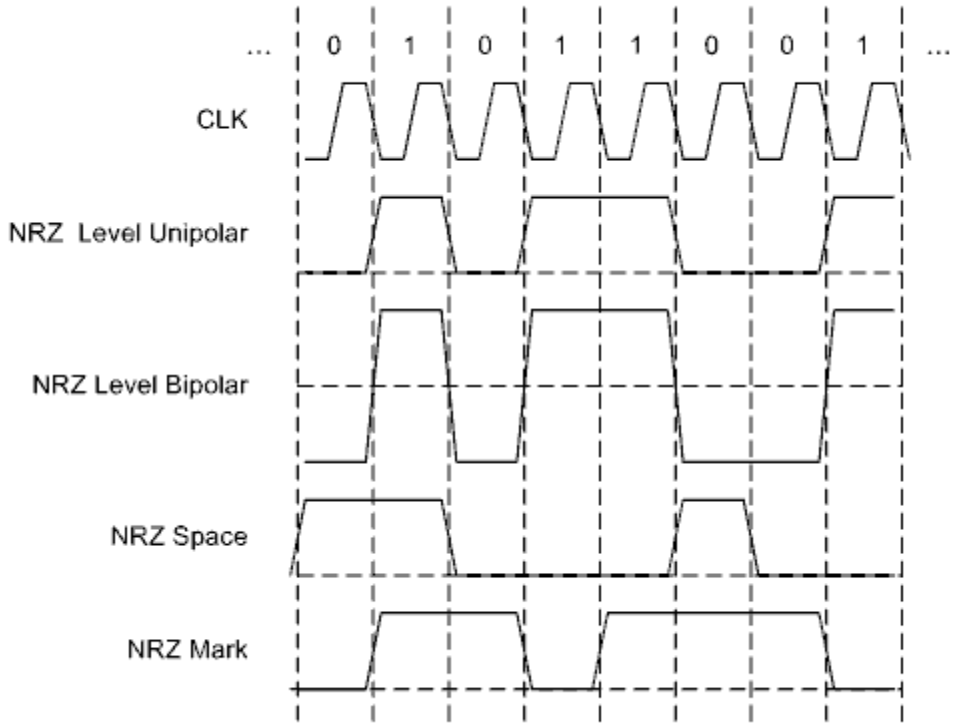
Space и mark инверсны друг относительно друга.

NRZ-коды могут быть однополярными и двухполярными.

Требуется наличие дополнительной цепи для тактирования.

Примеры технологий с применением NRZ-кодов: RS-232, USB, HDLC.

4.1.3.2



NRZ-коды

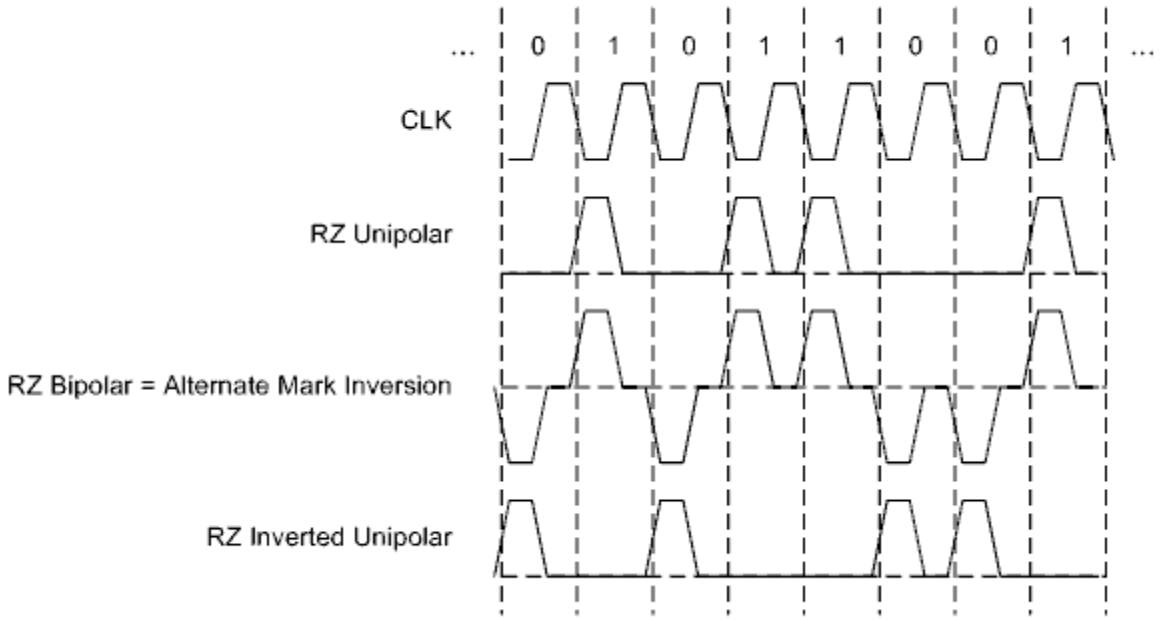
4.1.3.3

Закодируйте байт `10100110` кодами NRZ Level Inverted Unipolar, NRZ Space Bipolar.

RZ-коды так же выражаются в изменении уровней между тактами, но на половине каждого такта всегда происходит возврат к нулю (земле).

Двухполярные RZ-коды обладают свойством самосинхронизации.

Пример технологии с применением RZ-кода: IrDA.


...
0
1
0
1
1
0
0
1
...
CLK
RZ Unipolar
RZ Bipolar = Alternate Mark Inversion
RZ Inverted Unipolar

RZ-коды

4.1.4.3

Закодируйте байт `10100110` кодом RZ Inverted Bipolar.

4.1.5.1a

Манчестерские коды выражаются в переходах между уровнями во время тактов, поэтому их иногда называют фазовыми кодами.

Есть два «равноправных» варианта собственно манчестерского кода. Ноль во входной последовательности заменяется на переход от единицы к нулю, а единица заменяется на переход от нуля к единице. Либо наоборот.

Манчестерские коды обладают свойством самосинхронизации.

4.1.5.1b

Еще несколько кодов близки к манчестерскому.

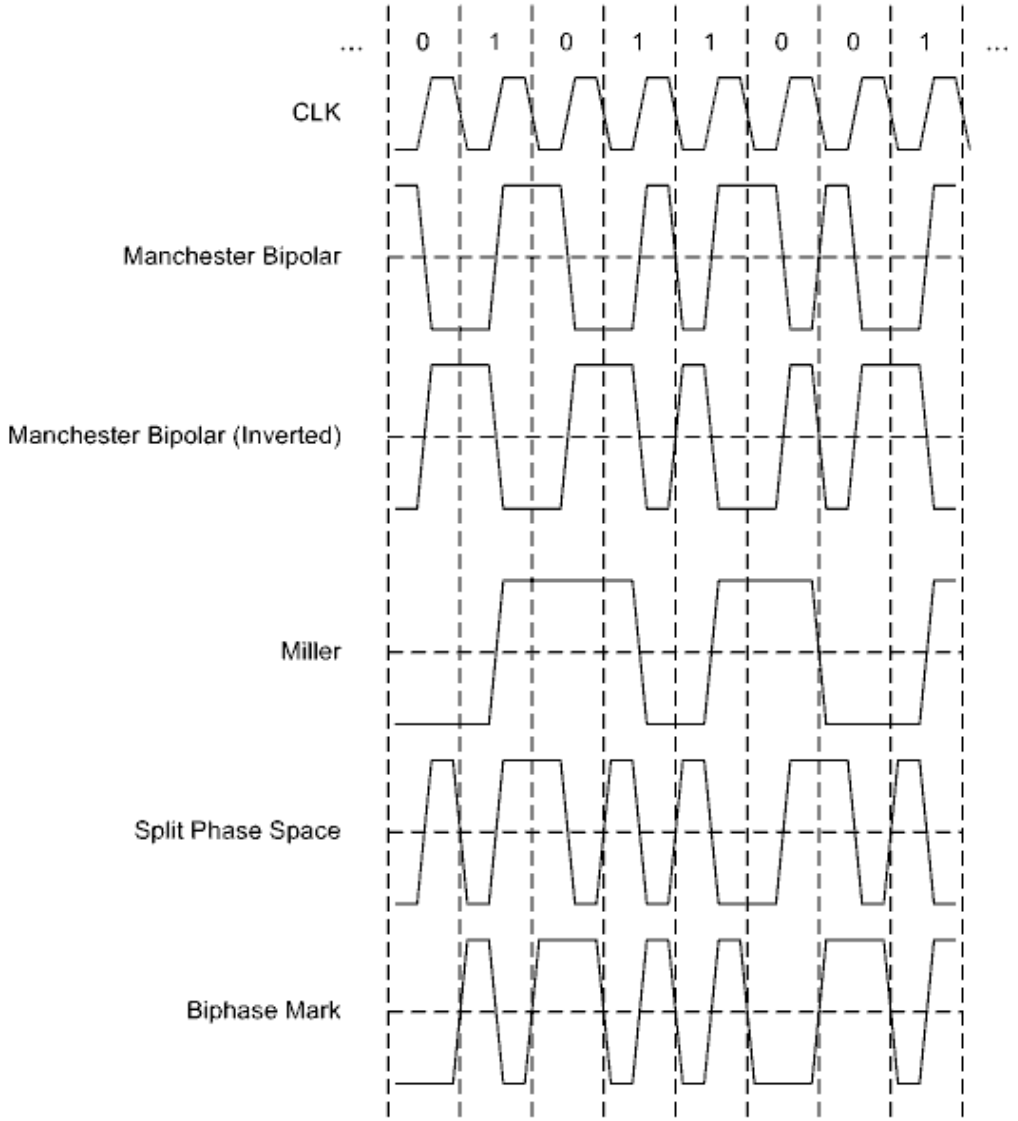
Согласно коду Миллера (Miller), ноль соответствует отсутствию перехода во время такта, единица соответствует переходу во время такта, плюс между двумя нулями всегда выполняется смена уровня.

Согласно коду Split Phase учитывается направление предыдущего перехода. При space-варианте ноль соответствует переходу во время такта в направлении, противоположном направлению предыдущего перехода, единица соответствуют переходу во время такта в направлении, совпадающем с направлением предыдущего перехода. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.

Согласно коду Biphase, кроме возможных переходов во время тактов, всегда выполняется смена уровня между тактами. При space-варианте ноль соответствуют переходу во время такта, единица соответствуют отсутствию перехода во время такта. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.

Примеры технологий с применением манчестерских кодов: Ethernet, Token Ring, некоторые IR-технологии.

4.1.5.2



Манчестерские коды

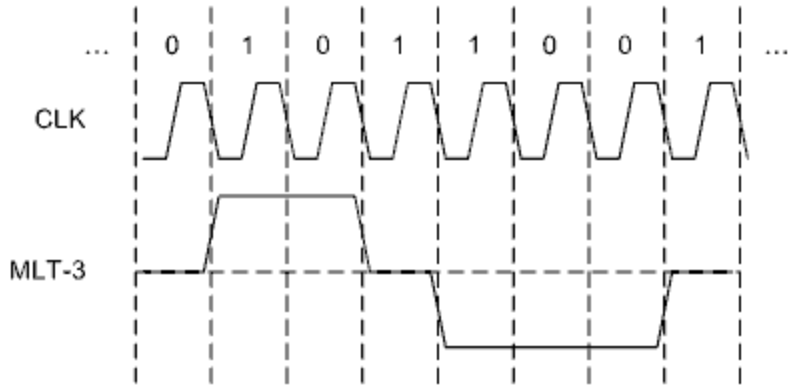
4.1.5.3

Закодируйте байт `10100110` кодами Manchester Bipolar, Split Phase (Mark), Biphase Space.

MLT-коды выражаются в переключении между несколькими уровнями между тактами.

Например, код MLT-3 имеет три уровня: -1, 0, +1. Кодирование может начинаться с нуля, ноль в исходной последовательности кодируется сохранением текущего уровня, а единица -- переходом к соседнему уровню (с сохранением направления, если это возможно).

Примеры технологий с применением MLT-кодов: Fast Ethernet, FDDI.



MLT-коды

4.1.6.3

Закодируйте байт `10100110` кодом MLT-3.

4.1.6.4

Что можно достичь увеличивая число уровней при кодировании?

4.1.7.1

Блочные коды выражаются в замене блоков битов из входной последовательности на бо'льшие (как правило) по размеру блоки битов в выходной последовательности.

Блочные коды могут комбинироваться с вышеперечисленными кодами.

В связи с избыточностью блочных кодов, во многих из них предусмотрены контрольные последовательности, которые, по сути, являются управляющими символами.

Первым примером может служить код 4b/5b, применяемый в Fast Ethernet и CDDI.

4b	5b
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Основная таблица кода 4b/5b

4.1.7.3

Более сложным примером может служить код 8b/10b, применяемый в оптических вариантах Gigabit Ethernet.

Биты входного блока обозначают как *ABCDEFGH* -- от младшего к старшему, выходного *abcdefghij* -- так же от младшего к старшему.

Входной блок разбивается на два подблока: *x* из пяти битов и *y* из трех битов. Поэтому выходной код представляет собой конкатенацию двух кодов: 5b/6b и 3b/4b.

Кроме собственно блоков данных *D*, имеются контрольные блоки *K*, которые кодируют альтернативно.

Таким образом, входной блок обозначают как *Dx.y* либо *Kx.y*.

Наконец, в код 8b/10b заложена гибкая система уравнивания количества нулей и количества единиц, заключающаяся в динамическом выборе блока для замены (одного из двух) исходя из текущего значения так называемого RD (Running Disparity). Предусмотрено два значение RD: -1 и +1. При выборе текущего значения RD учитывается предыдущее значение RD и соотношение нулей и единиц во входном блоке (плюс есть исключения).

5b	6b		5b	6b	
EDCBA	*abcdei*		*EDCBA*	*abcdei*	
D	RD = -1	RD = +1	*D*	RD = -1	RD = +1
00000	100111	011000	10000	011011	100100
00001	011101	100010	10001	100011	
00010	101101	010010	10010	010011	
00011	110001		10011	110010	
00100	110101	001010	10100	001011	
00101	101001		10101	101010	
00110	011001		10110	011010	
00111	111000	000111	10111	111010	000101
01000	111001	000110	11000	110011	001100
01001	100101		11001	100110	
01010	010101		11010	010110	
01011	110100		11011	110110	001001
01100	001101		11100	001110	
01101	101100		11101	101110	010001
01110	011100		11110	011110	100001
01111	010111	101000	11111	101011	010100

3b	4b	
HGF	*fghj*	
D	RD = -1	RD = +1
000	1011	0100
001	1001	
010	0101	
011	1100	0011
100	1101	0010
101	1010	
110	0110	
111	1110	0001
111	0111	1000

4.1.8.1

Линейные коды, применяемые в оптических каналах имеют особенности в сравнении с кодами для проводниковых каналов.

Примеры: TS-FO (Three of Six -- Fiber Optical), RZ carrier-suppressed, RZ alternate-phase.

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Очевидно, что неправильная интерпретация принятых данных чревата непредсказуемыми последствиями.

Серьезное изучение помехоустойчивого кодирования предполагает «погружение» в математику. При изучении же данной дисциплины больше важны прикладные аспекты, но без некоторых алгебраических основ не обойтись.

В теории помехоустойчивого кодирования очень важное место занимают поля Галуа, но чтобы к ним «подойти» нужно сделать ряд шагов.

4.2.1.2

Как в математике задают множество?

Приведите примеры множеств.

4.2.1.3

Некоторую операцию * называют бинарной если после ее применения к двум любым элементам *a* и *b* некоторого множества получают элемент *c*, принадлежащий тому же множеству: *a* * *b* = *c*.

А соответствующее непустое множество *S* называют *замкнутым* относительно бинарной операции *.

Элемент *e* множества называют *нейтральным* если, после бинарной операции над этим элементом и некоторым другим, другой участвовавший в операции элемент не изменяется: *a* * *e* = *a*.

Два элемента множества называют *обратными* (относительно друг друга) если в результате бинарной операции над ними получают нейтральный элемент: *a* * *b* = *e*.

4.2.1.4

Множество G называют группой если для него определена бинарная операция * и:

1. Операция * является ассоциативной: $(a * b) * c = a * (b * c)$ -- соответствует умножению.
2. Существует нейтральный элемент -- соответствует единице.
3. Имеется унарная операция, позволяющая получить обратный элементу a элемент -- соответствует a^{-1}.

Группу называют *абелевой* если операция * коммутативна: $a * b = b * a$.

Если для группы определена операция умножения ($a * b = ab$), то группу называют *мультипликативной*.

Мультипликативную группу называют *циклической* если в ней существует такой элемент, что все остальные элементы являются степенями этого элемента: $b = a^k$. А сам элемент a называют *образующим* группу.

4.2.1.5

Классом вычетов по модулю *n*, принадлежащему множеству натуральных чисел **N**, называют подмножество элементов из множества целых чисел **Z**, имеющих одинаковый остаток от деления на *n*.

[*a*] -- класс вычетов, одним из элементов которого является *a*.

Группу, образованную множеством классов вычетов по модулю *n*, называют *группой классов вычетов* по модулю *n*.

4.2.1.6

Приведите пример других чисел из класса вычетов по модулю 100, в котором содержится число 205.

4.2.1.7

Группу называют *конечной* если группа состоит из конечного числа элементов.

Число элементов $|G|$ конечной группы называют ее *порядком*.

4.2.1.8

Два целых числа *сравнимы* (*эквивалентно равны*) по модулю натурального числа n если при делении на n они дают одинаковые остатки: $a \equiv b \pmod{n}$.

4.2.1.9

Отображение *f*: *G* -> *H* группы *G* в группу *H* называют *гомоморфным* если оно сохраняет операцию группы *G*. Отображение *изоморфно* если оно взаимно однозначно.

Отображение *f*: *G* -> *G* называют *эпиморфным*, изоморфное отображение *f*: *G* -> *G* называют *автоморфным*.

4.2.1.10

Множество R называют *кольцом* если для множества определены две бинарные операции # и * такие что:

1. Множество R является абелевой группой относительно операции # -- соответствует сложению.
2. Операция * является ассоциативной.
3. Выполняется закон дистрибутивности: $a * (b \# c) = a * b \# a * c$.

Если для группы определена операция *сложения* ($a \# b = a + b$), то группу называют *аддитивной*. Единичный элемент аддитивной группы соответствует нулю. Обратный элементу a элемент аддитивной группы соответствует $-a$.

На операцию * можно накладывать дополнительные ограничения. Если в кольце присутствует единица, то кольцо называют *кольцом с единицей*.

При выполнении закона коммутативности кольцо называют *коммутативным*.

Коммутативное кольцо называют *целостным* если его единица не равна нулю и $a * b = 0$ только при $a = 0$ или $b = 0$.

4.2.1.11

Кольцо называют *телом* если кроме нуля в кольце существуют другие элементы и эти элементы образуют группу относительно операции *.

Наконец, коммутативное тело *F* называют *полем*.

Подгруппой, *подкольцом*, *подполем* называют подмножества сохраняющие соответствующие свойства.

Поле, не содержащее подполей, называют *простым*. Простым будет поле, порядок которого равен простому числу.

4.2.1.12

Подкольцо *I* кольца *R* называют его *идеалом* (*двухсторонним идеалом*) если для любой пары элементов *a* из *I* и *r* из *R* их произведение принадлежит *I*.

Подкольцо *R*/*I* классов вычетов по модулю идеала *I* из кольца *R* называют *факторкольцом* кольца *R* по идеалу *I*.

Наименьшее из натуральных чисел *n*, такое что для любого элемента *r* из кольца *R* выполняется равенство $n * r = 0$, называют *характеристикой* кольца *R*.

Согласно теореме, каждое конечное целостное кольцо образует поле.

Согласно другой теореме, характеристикой конечного поля является простое число.

Поле *GF*(*p*) из целых чисел 0, 1 ... *p* - 1, порожденное в результате отображения *f*: **Z**/*p* -> *GF*(*p*), где **Z**/*p* -- факторкольцо множества целых чисел, в котором роль идеала играет простое число *p*, и *f*([*a*]) = *a*, называют *полем Галуа* (Galois field) порядка *p*.

При вычислениях с элементами поля Галуа используют целочисленную арифметику с приведением по соответствующему модулю.

4.2.1.14

Задача у доски.

Из каких элементов будет состоять поле $GF(5)$?

Задайте операции сложения и умножения для поля $GF(5)$?

Одно и то же число можно записать самыми разными способами. Число можно рассматривать и как значение полинома.

Полиномом (*многочленом*) одной переменной называют выражение:

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 \ .$$

В случае *полинома над кольцом*, коэффициенты a_i соответствуют элементам кольца, а переменная x кольцу не принадлежит.

Два полинома $f(x)$ и $g(x)$ называют *равными* если равны их степени и равны коэффициенты при одинаковых степенях переменной.

Если старший коэффициент полинома равен единице, то полином называют *приведенным*.

4.2.1.16

Что такое НОД и НОК?

Для чего применяют алгоритм Евклида?

4.2.1.17

Тривиальными делителями полинома называют сам полином и полином равный 1.

Полином над полем *F* называют *неприводимым* над этим полем если полином допускает только тривиальное разложение, то есть у полинома нет нетривиальных делителей из *F* (с точностью до домножения на ненулевую константу).

4.2.1.18

Запишите все неприводимые полиномы третьей степени.

Что такое взаимно простые числа?

4.2.1.19

Что такое корень полинома?

Для практического применения полей Галуа в компьютерных системах необходимо перейти от скалярного представления к векторному.

Расширенное поле Галуа $GF(p^n)$ можно рассматривать как векторное пространство, где простое число p является характеристикой поля и соответствует количеству состояний разряда вектора, а n является степенью поля над его простым подполем и соответствует количеству разрядов вектора.

Поскольку в обычных компьютерных системах разряды регистров бинарные, то наибольший интерес представляют поля $GF(2^n)$.

Сложение бинарных векторов (совпадает с вычитанием) проблему не представляет и соответствует поразрядной операции xor.

А вот с умножением и делением дела обстоят значительно сложнее.

Скалярное произведение не подходит, так как его результат может «выйти» за пределы поля.

Векторное произведение определено только для трехразрядных векторов.

Полиномиальное представление так же с ходу не решает проблему, так как произведение полиномов опять же «выводит» за пределы поля.

Для обеспечения конечности поля Галуа, полученный в результате произведения полином нужно привести. Это достигают путем деления на некий выбранный полином степени n. Ясно, что выбирать можно разные полиномы. Выбор другого полинома приведет к другим результатам умножения и, соответственно, к другому полю $GF(p^n)$.

Выбранный для построения поля Галуа полином называют *порождающим* (*образующим*).

4.2.1.21b

Деление векторов в математике не известно.

После перехода на язык полиномов, опять же для обеспечения конечности поля Галуа, деление всегда должно быть безостаточным. Деление можно представить как умножение полинома-делимого на полином, обратный делителю. При этом для достижения цели на основании математических выкладок, необходимо ввести еще одно ограничение: порождающий полином должен быть неприводимым по модулю p (например, если $p = 2$ и $n = 4$, то полином $x^4 + 1$ (число 17) не подходит, так как $x^4 + 1 \equiv (x^2 + 1)^2 \pmod 2$).

Возведение в степень обладает цикличностью.

4.2.1.22

Задача у доски.

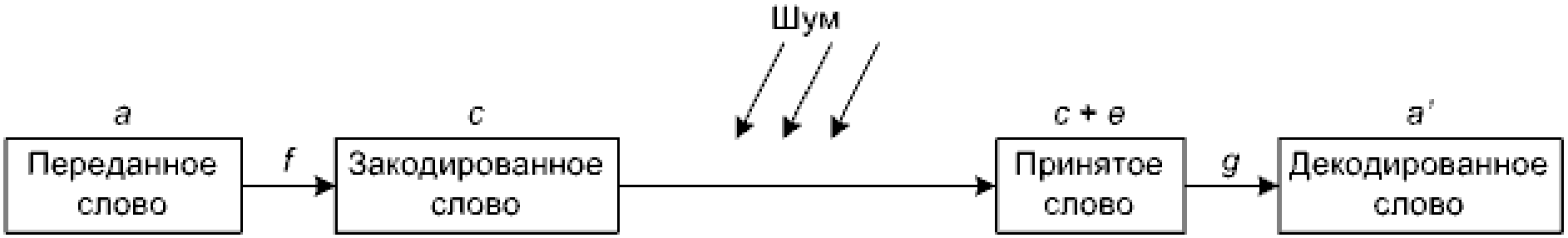
Как будут выглядеть элементы поля $GF(2^3)$ с порождающим полиномом $x^3 + x^2 + 1$?

Определите операции сложения и умножения.

4.2.2.1

Считается, что начало помехоустойчивому кодированию положила теорема Шеннона, утверждающая что любой дискретный канал связи имеет конечную пропускную способность и этот канал может быть задействован для передачи информации со сколь угодно большой степенью достоверности, не смотря на наличие помех.

4.2.2.2a



Модель канала связи

4.2.2.2b

Передаваемое сообщение разбивается на блоки фиксированного размера a из k битов a_1, a_2 ... a_k.

Кодер выполняет функцию f, называемую *схемой кодирования*, и тем самым преобразует вектор a в вектор c из $n > k$ битов c_1, c_2 ... c_n, называемый *кодовым словом*.

В процессе пересылки кодового слова по каналу связи на него накладывается *вектор ошибок* e, в котором единичные биты соответствуют искажениям.

После применения декодером *схемы декодирования* g получается вектор a', в идеале совпадающий с исходным вектором a.

Подобная схема кодирования является избыточной. На практике всегда ищут компромисс между степенью обеспечения достоверности при передаче и вычислительной сложностью кодов (что в первую очередь отражается на скорости декодирования).

В КС множество кодовых слов получается из множества исходных слов как отображение из конечного поля $GF(2^k)$ в конечное поле $GF(2^n)$.

При более простых схемах кодирования, в кодовом слове сначала располагаются биты входного сообщения, называемые *информационными*, а за ними дополнительные биты, называемые *проверочными*: a_1, a_2 ... a_k, c_{k+1}, c_{k+2} ... c_n.

В более сложных случаях проверочные биты чередуются с информационными.

Схему кодирования удобно представлять в матричном виде.

Схема кодирования:

f: $GF(2^3)$ -> $GF(2^6)$ = a_1,a_2,a_3 -> a_1,a_2,a_3,c_4,c_5,c_6

Проверочные уравнения:

$c_4 = a_1 \wedge a_2$
$c_5 = a_2 \wedge a_3$
$c_6 = a_1 \wedge a_3$

Переход к матричному представлению:

$$\begin{bmatrix} c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Проверочные уравнения по-другому:

$a_1 \wedge a_2 \wedge c_4 = 0$
$a_2 \wedge a_3 \wedge c_5 = 0$
$a_1 \wedge a_3 \wedge c_6 = 0$

В матричном виде (^{T} означает транспонирование):

$H\,c^T = 0$

4.2.2.4b

Видно, что проверочные уравнения образуют систему линейных уравнений. Следовательно, отображение f (схема кодирования) является линейным.

Если H -- матрица размером $(n - k) \times n$ ранга $n - k$ и $H c^{T} = 0$, то множество всех n-разрядных векторов, входящих в поле $GF(2^n)$ (в общем случае $GF(p^n)$), называют *линейным (n,k)-кодом* (в математическом смысле) длины n и размерности k. А матрицу H называют *проверочной*.

Линейный код по-другому называют *групповым*, так как множество кодовых слов можно рассматривать как подгруппу в отношении поля $GF(2^n)$.

Линейный код называют *систематическим* (*разделенным*) если расположение проверочных битов известно (не важно где они находятся), то есть если $H = [A \quad I_{n-k}]$, где A -- матрица размером $(n - k) \times k$, а I_{n-k} -- единичная матрица ранга $n - k$.

4.2.2.6

Матрицу $G = [I_k \quad -A^T]$ размером $k \times n$ называют *кодирующей* (*порождающей*) матрицей систематического кода.

Кодирующая и проверочная матрицы связаны следующим образом: $GH^T = 0$.

4.2.2.7

Самыми примитивными из линейных кодов являются подсчет контрольной суммы и дублирование информационных символов.

4.2.2.8

Перед выбором того либо иного помехоустойчивого кода всегда нужно определиться, что требуется от кода. Если перефразировать, то нужно ответить на два вопроса:

1. Сколько бинарных ошибок код должен обнаруживать.
2. Сколько бинарных ошибок код должен исправлять.

Исправлять ошибки значительно сложнее, чем обнаруживать. Применительно ко многим кодам, исправление ошибки подразумевает нахождение ее позиции.

4.2.2.9

В общем случае ошибки носят случайный характер. Множественные ошибки могут быть взаимозависимыми, то есть образовывать *модули ошибок*. Если ошибки расположены рядом, то они образуют *пакет ошибок* (частный случай модуля).

4.2.2.10

Число координат (позиций), которыми два вектора *x* и *y* различаются называют *расстоянием Хэмминга* -- *d*(*x*,*y*).

Число ненулевых позиций вектора x называют *весом Хэмминга* -- *w*(*x*).

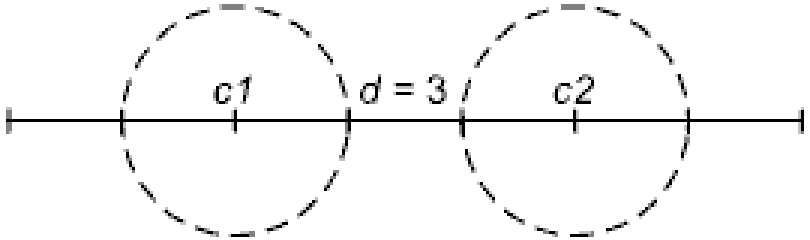
Видно, что расстояние Хэмминга показывает количество возникших ошибок.

4.2.2.11

Для увеличения корректирующей способности кода следует стремиться увеличивать расстояния между кодовыми словами. При этом минимальное расстояние d_{min} называют *кодовым* и оно является очень важной характеристикой помехоустойчивого кода.

Согласно теореме, для того чтобы линейный код исправлял t ошибок должно выполняться условие: $d_{min} \geq 2t + 1$.

Для того, чтобы линейный код обнаруживал t ошибок должно выполняться условие: $d_{min} \geq t + 1$.



Для того чтобы линейный код имел $d_{min} \geq s + 1$, необходимо и достаточно, чтобы любые s столбцов его проверочной матрицы были линейно независимы.

4.2.2.12

Способность того или иного кода сохранять свои характеристики зависит и от количественного соотношения информационных и проверочных символов. В теории помехоустойчивого кодирования определяют так называемые верхние и нижние границы кодов.

Существуют три основных способа декодирования:

1. По минимуму расстояния.
2. По синдрому (по лидеру смежного класса).
3. Мажоритарное:

-- с разделенными проверками;

-- с λ-проверками;

-- с квазиразделенными проверками.

4.2.3.1

За достаточно длительную историю развития прикладной теории кодирования, как науки, было придумано очень много помехоустойчивых кодов.

Основные группы помехоустойчивых кодов:

1. Линейные коды, в том числе: коды Хэмминга, циклические коды, БЧХ-коды (коды Боуза-Чоудхури-Хоквингема), РМ-коды (коды Рида-Маллера), итеративные коды, коды на основе матриц Адамара, симплексные коды и некоторые другие.

2. Коды для контроля модульных и пакетных ошибок, в том числе: РС-коды (коды Рида-Соломона), низкоплотные модульные коды, векторные модульные коды, итеративные модульные коды и некоторые другие.

3. Сверточные коды.

4. Арифметические коды.

5. Низкоскоростные коды, в том числе: коды максимальной длины, нелинейные коды, D-коды и некоторые другие.

4.2.3.2

Для изучения в рамках данной дисциплины выбраны два кода:

1. Код Хэмминга -- Hamming code.
2. Циклический код -- CRC (Cyclic Redundancy Code).

4.2.4.1

Бинарным кодом Хэмминга называют код длины $n = 2m - 1$, $m \geq 2$ с проверочной матрицей H размером $m \times (2m - 1)$, в которой столбцы соответствуют записи 1, 2 ... $2^m - 1$ в двоичной системе счисления.

Код Хэмминга позволяет исправлять одиночную ошибку и обнаруживать множественные ошибки.

4.2.4.2

Задача у доски.

Запишите проверочную матрицу кода Хэмминга (7,4).

Циклические коды являются особо выделяемой подгруппой линейных кодов.

Циклическим кодом называют линейный код, удовлетворяющий дополнительному условию: если вектор a_0, a_1 ... a_{n-1} является кодовым словом, то и его циклический сдвиг a_{n-1}, a_0 ... a_{n-2} так же является кодовым словом.

Циклический код позволяет исправлять одну и более ошибок и обнаруживать множественные ошибки (зависит от параметров).

4.2.5.2

Базовая идея циклического кодирования состоит в том, чтобы в качестве проверочных битов передавать остаток от деления информационных битов на некоторое выбранное число.

После приема снова выполняется деление уже возможно искаженных информационных битов на то же самое число и сравниваются остатки.

Если остатки совпадают, то данные с определенной вероятностью приняты без ошибок.

На практике же деление выполняется по правилам арифметики полей Галуа, то есть без учета переносов.

Информационные биты, то есть делимое, соответствуют информационному полиному.

Делитель соответствует порождающему (образующему) полиному.

Частное в процессе кодирования не используется и поэтому «отбрасывается».

Для того чтобы максимально разнообразить остатки в качестве порождающего полинома должен выбираться неприводимый полином.

4.2.5.4

Существуют два подхода к реализации циклического кода на стороне приемника:

1. Согласно базовой идее, описанной выше.

2. На порождающий полином делится все принятое кодовое слово. Если ошибок не произошло, то остаток будет нулевым.

Оба подхода равноценны.

4.2.5.5

Задача у доски.

Сформируйте циклический код для информационного вектора `10100110` (требуется обнаружить и исправить одиночную ошибку).