Санкт-Петербургский государственный политехнический университет
Кафедра компьютерных систем и программных технологий

Надежность систем и устройств

Моисеев Михаил Юрьевич

Лекция №6

Информационное резервирование
Линейные коды

2011

Способы повышения надежности

- Использование резервирования
  - Структурное
  - **Информационное**
  - Временное
  - Алгоритмическое
  - Функциональное
- Другие способы

Информационное резервирование

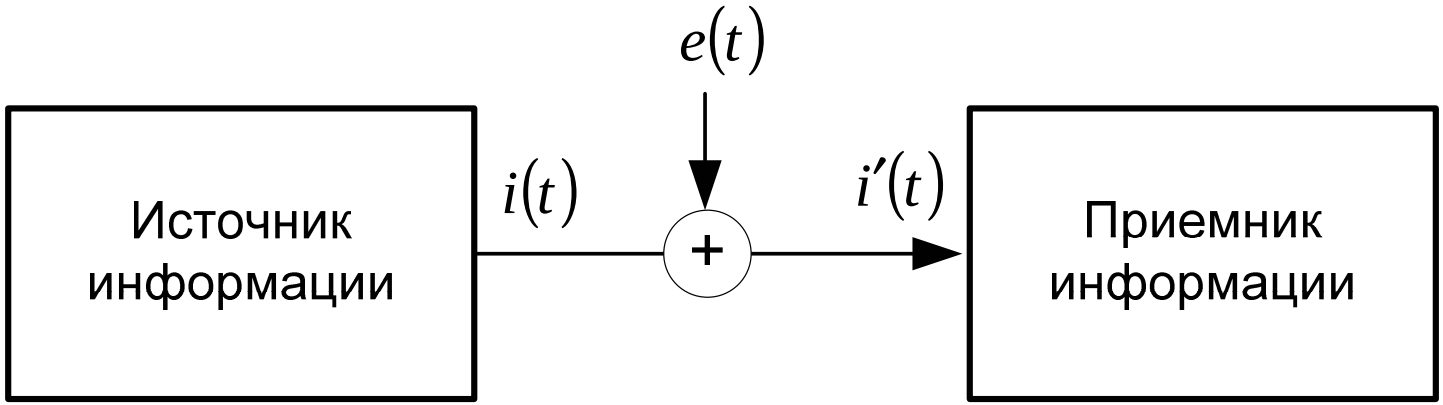
- **Информационное резервирование** – способ повышения надежности основанный на введении избыточности в информационные потоки данных, управляющих команд и вычислительных операций
- Повышение надежности достигается за счет обнаружение искаженных данных и результатов вычислений, устранения ошибок в них, восстановления синхронизации при передаче данных

Особенности информационного резервирования

- Информационное резервирование используется в аппаратных и программных системах для защиты информации всех видов
- При информационном резервировании применяются различные методы введения избыточности основанные на использовании **помехоустойчивых кодов**
- Информационное резервирование применяется
  - в каналах передачи информации
  - в системах хранения информации
  - в системах обработки информации

Модель канала связи с помехами

- Информационный вектор, вектор ошибок



$$i'(t) = i(t) \oplus e(t)$$

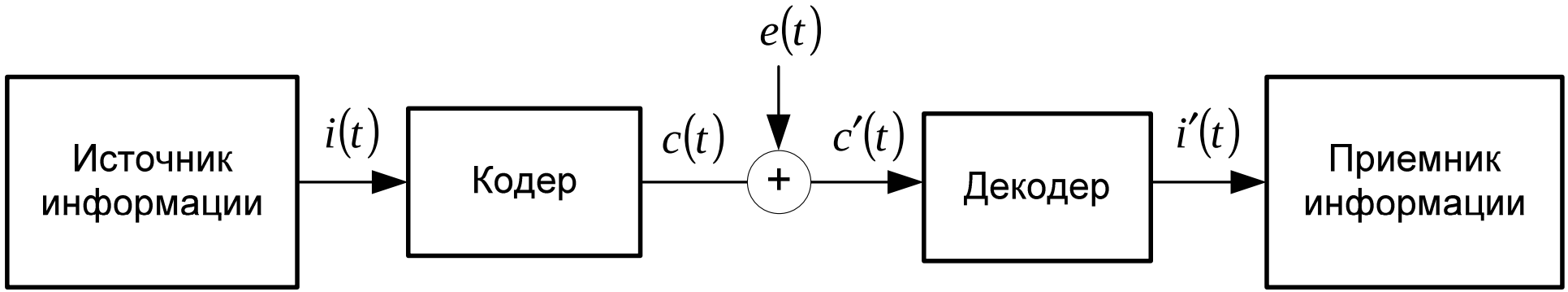
- Двоичный канал с аддитивными независимыми ошибками
- Обнаружение и исправление ошибок на стороне приемника

Определение кода

- **Код** – способ представления информации
- **Кодирование** – процесс преобразования информации из одного представления в другое по некоторому набору правил
- **Кодовая комбинация** – комбинация символов допустимая с точки зрения кода, может быть получена в результате кодирования
- Набор всех кодовых комбинаций – **алфавит кода**

Определение помехоустойчивого кода

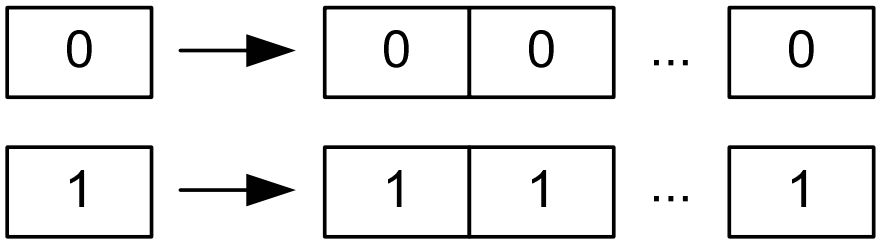
- Оптимальный код
- **Помехоустойчивый код** – код, использующий избыточные символы (не оптимальный)
- Кодовый вектор



$$c(t) = F(i(t)) \qquad c'(t) = c(t) \oplus e(t) \qquad i'(t) = F^{-1}(c'(t))$$

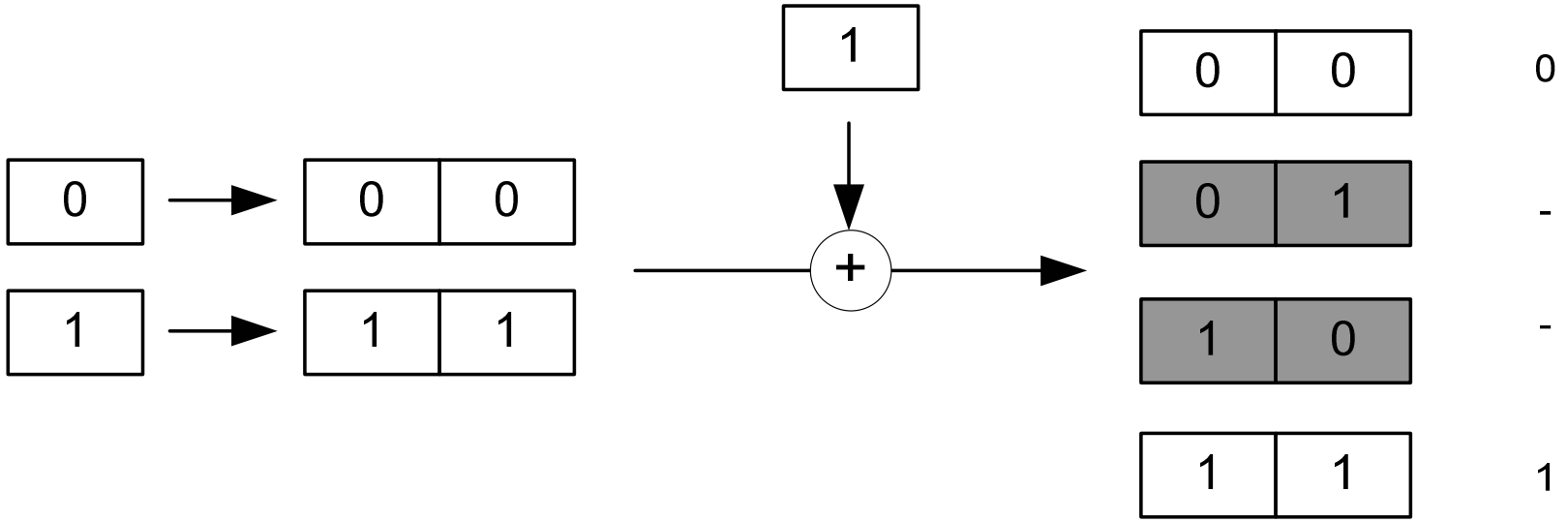
Мажоритарный код

- **Мажоритарный код** – избыточный код, использующий повторение символов
- Каждый символ передается *n* раз
- Обнаружение ошибок – полученные копии символа различаются
- Исправление ошибок – полученные копии символа содержат большее количество переданных (неискаженных) значений



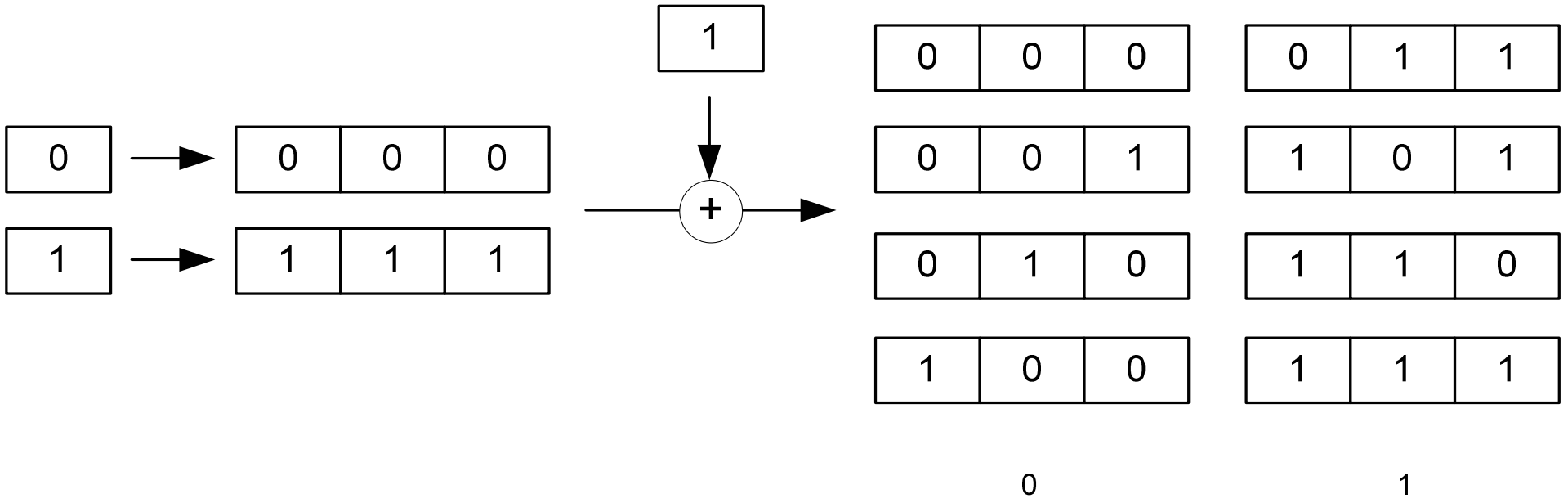
Применение мажоритарного кода

- В двоичном канале вероятность искажения символа 0,1
  - Какова вероятность необнаружения ошибки если *n=2*?



Применение мажоритарного кода

- В двоичном канале вероятность искажения символа 0,1
  - Какова вероятность неправильного приема при исправлении ошибок если *n=3*?



Кодовое расстояние

- Вес Хэмминга – число ненулевых элементов

$$X = (01101001),\ w(X) = 4$$

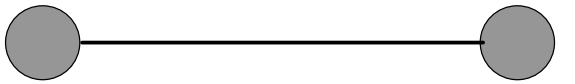
- Расстояние Хэмминга – вес разности векторов

$$X_1 = (0100111),\ X_2 = (1100100)$$
$$w(X_1 - X_2) = (0100111) - (1100100) = (1000011)$$

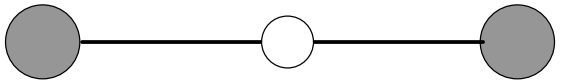
- Кодовым расстоянием блокового кода называют минимальное расстояние Хэмминга между двумя любыми кодовыми векторами

$$X_1 = (1111), X_2 = (0100), X_3 = (0111)$$
$$d = 1$$

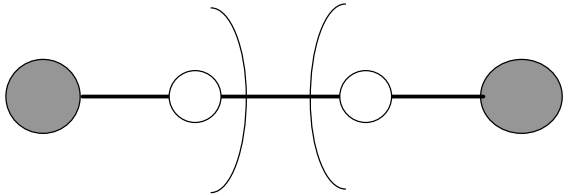
Кодовое расстояние



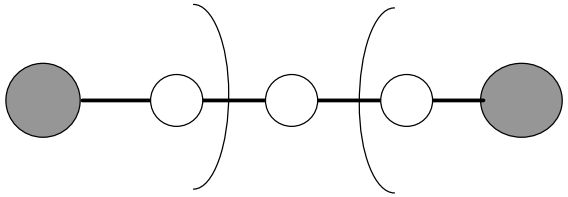
$d = 1$ $t_{\text{исп}} = 0, t_{\text{обн}} = 0$



$d = 2$ $t_{\text{исп}} = 0, t_{\text{обн}} = 1$



$d = 3$ $t_{\text{исп}} = 1, t_{\text{обн}} = 2$



$d = 4$ $t_{\text{исп}} = 1, t_{\text{обн}} = 3$

Кодовое расстояние

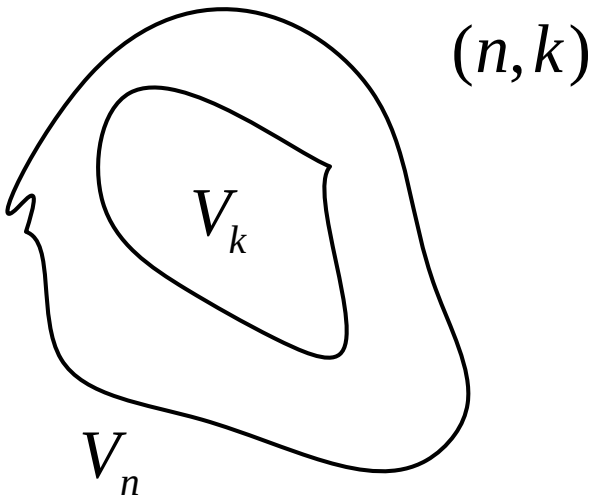
- Связь между кодовым расстоянием и свойством кода исправлять и обнаруживать ошибки

$$d = 2t_{испр} + 1$$

$$d = t_{обн} + 1$$

Определение линейного кода

- Множество векторов образующих подпространство V_k пространства V_n называют **линейным (n,k)-кодом** (k < n)
- Нулевое подпространство дополняет V_k до V_n, любой вектор из V_{n-k} ортогонален любому вектору из V_k



$$V_k \cup V_{n-k} = V_n$$

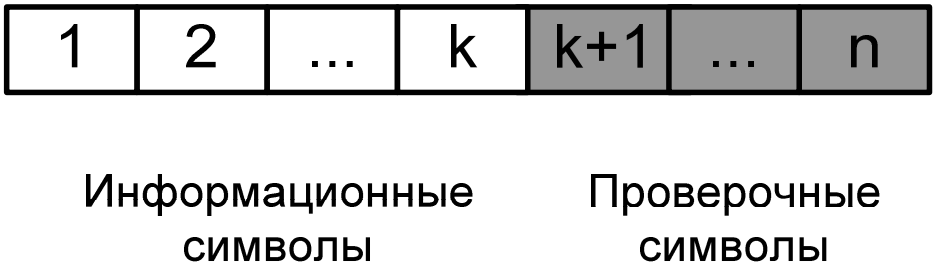
$$V_k \perp V_{n-k}$$

Определение линейного кода

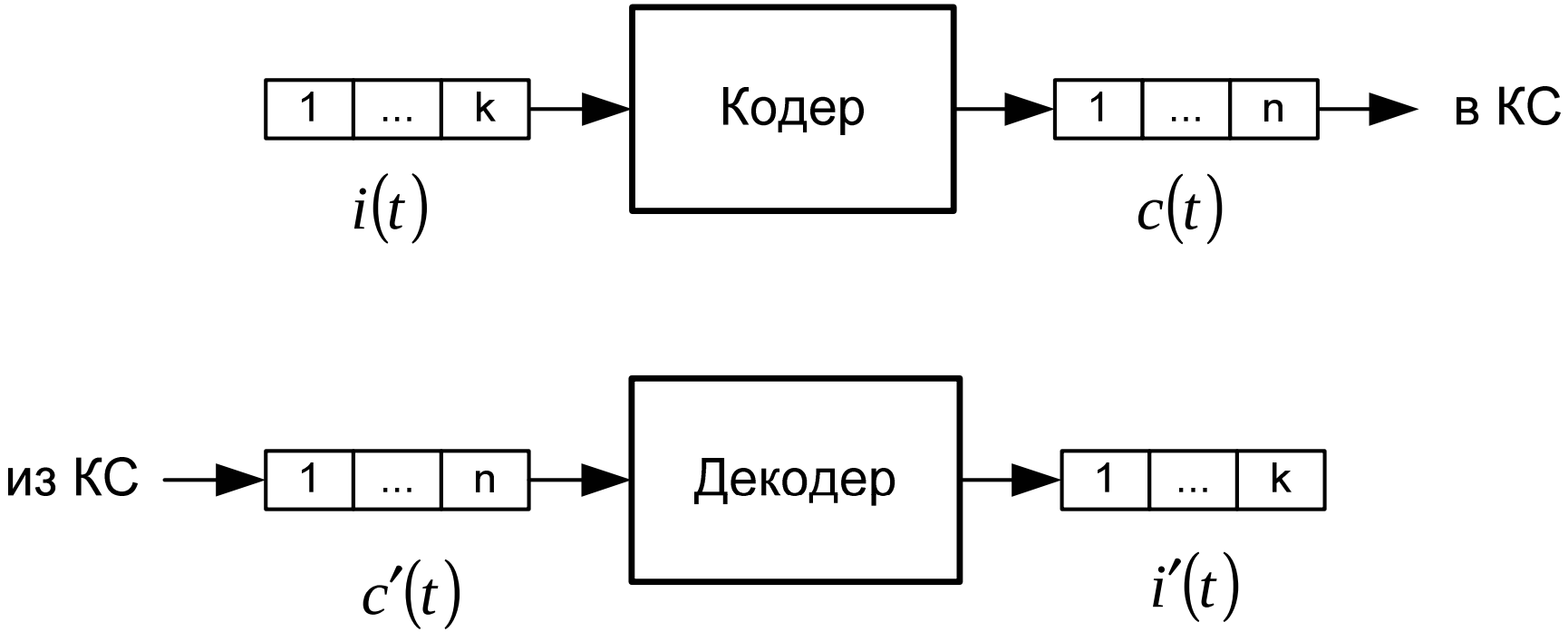
- Из всех векторов длиной n (всего 2^n), определенным образом выбираются 2^k векторов
- Подпространство V_k имеет базис из k векторов, все кодовые вектора получаются с помощью линейных комбинаций базисных векторов (всего 2^k комбинаций)
- Коды отличаются друг от друга параметрами n и k, а также способом выбора базисных векторов

Кодовый вектор

- Кодовый вектор включает
  - информационные символы
  - проверочные символы
- В систематическом коде первые ***k*** символов – информационные, последние ***(n-k)*** – проверочные

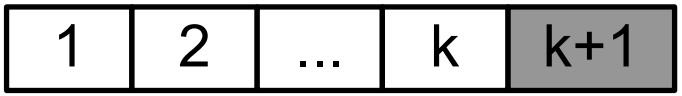


Кодер и декодер линейного кода



Код с проверкой на четность

- Код с проверкой на четность - линейный **(k+1,k)**-код
- Кодовое расстояние $d = 2$
- Обнаружение ошибок – сравнение проверочного символа с суммой информационных символов



Информационные символы

Проверочный символ

$$i_{k+1} = \bigoplus_{j=1..k} i_j$$

Порождающая и проверочная матрицы

- $G_{k\times n}$ – порождающая матрица кода, состоит из базисных векторов кодового подпространства V_k
- $H_{(n-k)\times n}$ – проверочная матрица кода, состоит из базисных векторов нулевого подпространства V_{n-k}

$$G_{k\times n} = \overset{n}{\boxed{g_{ij}}}\ k \qquad H_{(n-k)\times n} = \overset{n}{\boxed{h_{ij}}}\ n-k$$

Порождающая и проверочная матрицы

- Свойство порождающей и проверочной матриц $G \times H^T = 0$ следствие – $c \times H^T = 0$
- Вектор x – синдром принятого кодового вектора, используется для обнаружения и исправления ошибок

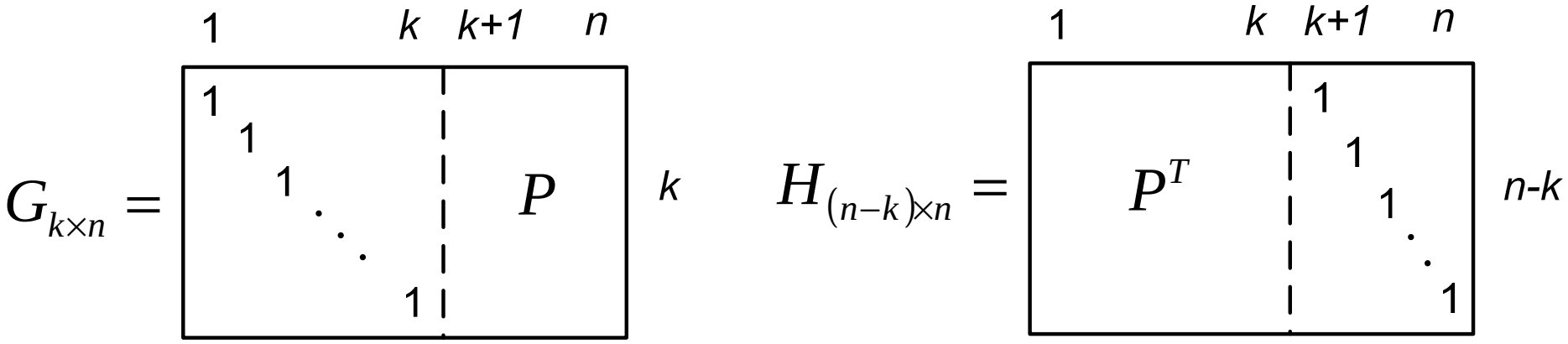
$$c = i \times G \qquad c' = c + e \qquad x = c' \times H^T$$

- Нулевой синдром свидетельствует об отсутствии ошибок

$$x = c' \times H^T = (c + e) \times H^T = e \times H^T$$

СКФ матриц

- Ступенчато-каноническая форма (СКФ) порождающей и проверочной матриц
- Матрицы любого линейного кода могут быть приведены к СКФ
- СКФ обеспечивает получение систематического кода



$$G = [I \vdots P] \qquad H = [P^T \vdots I]$$

Код Хэмминга

- Код Хэмминга – линейный код с $d = 3$
- Алгоритм построения кода Хэмминга
  - по заданному ***k*** находим минимальное ***n***, такое что $2^{n-k} \geq n+1$
  - строим проверочную матрицу с ненулевыми и различными вектор-столбцами
  - по проверочной матрице строим порождающую матрицу
- Построение проверочной матрицы в СКФ обеспечивает простое построение порождающей матрицы

Пример построения кода Хэмминга

- Построим код Хэмминга для ***k*** = 4
- Для $2^{n-k} \geq n+1$ минимальное ***n*** = 7
- Размер проверочной матрицы – 3*7, заполним столбцы
- Получили (7,4)-код

$$H = \begin{array}{|c|ccccccc|} \hline & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ \hline x_1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ x_3 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

$$G = \begin{array}{|ccccccc|} \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline \end{array}$$

Задача 1

- Построить порождающую и проверочную матрицы мажоритарного кода, $n = 3$
- Построить порождающую и проверочную матрицы кода с проверкой на четность, $k = 3$

Уравнения проверки

- Синдром принятого кодового вектора $x = c' \times H^T$
- Уравнения проверки строятся по строкам проверочной матрицы

$$x = (x_1 \ldots x_{n-k}),\ x_j = \sum_i a_i \cdot h_{ij}$$

$H =$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	1	1	1	0	1	0	0
x_2	1	0	1	1	0	1	0
x_3	0	1	1	1	0	0	1

$$x_1 = a_1 + a_2 + a_3 + a_5$$
$$x_2 = a_1 + a_3 + a_4 + a_6$$
$$x_3 = a_2 + a_3 + a_4 + a_7$$

Уравнения контроля

- Если синдром вектора равен нулю то этот вектор кодовый
- Уравнения контроля позволяют определить значения проверочных символов

$H =$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	1	1	1	0	1	0	0
x_2	1	0	1	1	0	1	0
x_3	0	1	1	1	0	0	1

$$0 = a_1 + a_2 + a_3 + a_5 \qquad a_5 = a_1 + a_2 + a_3$$

$$0 = a_1 + a_3 + a_4 + a_6 \quad \Rightarrow \quad a_6 = a_1 + a_3 + a_4$$

$$0 = a_2 + a_3 + a_4 + a_7 \qquad a_7 = a_2 + a_3 + a_4$$

Задача 2

- Построить порождающую и проверочную матрицы кода Хэмминга для $k = 5$, построить уравнения проверки и контроля, выполнить кодирование информационного вектора 01011, внести ошибку в 3-й разряд полученного кодового вектора и найти синдром вектора с внесенной ошибкой

Декодирование кода Хэмминга

- Обнаружение ошибок
  - обнаруживаются ошибки $t = 2$
  - признак ошибки – ненулевой синдром
- Исправление ошибок
  - исправляются ошибки $t = 1$
  - синдром совпадает со столбцом, в символе которого произошла ошибка

Пример исправления ошибки

- Значение синдрома зависит только от ошибки и не зависит от информационного вектора $x = e \times H^T$

$$H = \begin{array}{|c|ccccccc|} \hline & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ \hline x_1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ x_3 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

$$x_1 = a_1 + a_2 + a_3 + a_5$$
$$x_2 = a_1 + a_3 + a_4 + a_6$$
$$x_3 = a_2 + a_3 + a_4 + a_7$$

- Пусть в 4-ом разряде кодового вектора (7,4)-кода Хэмминга произошла ошибка $x = (011)$

ПТС

- Полная таблица соответствия (ПТС) – содержит отображение множества всех векторов ошибок на множество синдромов

$H =$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x_1	1	1	1	0	1	0	0
x_2	1	0	1	1	0	1	0
x_3	0	1	1	1	0	0	1

X			
000	0	236	345
001	7	46	
010	6	15	
011	4	12	
100	5		
101	2		
110	1		
111	3		

Свойства ПТС

- Каждая ошибка в ПТС однозначно отображается на один синдром, на каждый синдром отображается несколько ошибок
- Число комбинаций ошибок 2^n
  число синдромов 2^{n-k}
- Множество обнаруживаемых ошибок
- Множество исправляемых ошибок
- Одновременное обнаружение и исправление ошибок

X			
000	0	236	345
001	7	46	
010	6	15	
011	4	12	
100	5		
101	2		
110	1		
111	3		

Расширенный код Хэмминга

- Для одновременного обнаружения двукратных ошибок и исправления однократных ошибок расширим код Хэмминга с помощью проверки на четность
- Строим проверочную матрицу кода Хэмминга
- К матрице добавляем одну строку, содержащую все единицы и столбец содержащий все нули и одну единицу
- Кодовое расстояние $d = 4$

$$H = \left[\begin{array}{ccccccc:c} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Декодирование расширенного кода Хэмминга

- Алгоритм декодирования
  - если $x = (x_1 x_2 x_3) = (000)$, $x_4 = 0$, то ошибок нет
  - если $x = (x_1 x_2 x_3) \neq (000)$, $x_4 = 1$, то была однократная ошибка, которая исправляется
  - если $x = (x_1 x_2 x_3) \neq (000)$, $x_4 = 0$, то обнаружили двукратную ошибку
  - если $x = (x_1 x_2 x_3) = (000)$, $x_4 = 1$, то обнаружили ошибку более высокой кратности (нечетной)

Связь параметров кода с кодовым расстоянием

- Как изменится размерность кода Хэмминга если
  - добавить в проверочную матрицу одну строку
  - добавить в проверочную матрицу одну строку и один столбец
  - удалить из проверочной матрицы одну строку
  - удалить из проверочной матрицы один столбец

$$H = \begin{array}{c|ccccccc} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ \hline x_1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ x_2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ x_3 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}$$

Вопросы

- На чем основано информационное резервирование ?
- Что такое помехоустойчивых код ?
- Перечислите наиболее важные свойства кода.
- Что такое кодовое расстояние, как оно связано с корректирующими свойствами кода?
- Что такое порождающая и проверочная матрицы кода?
- Чем удобно использованием СКФ?
- Как строится код Хэмминга, какими свойствами он обладает?
- Для чего нужны уравнения проверки и контроля?
- Как выполняется обнаружение и исправление ошибок с помощью кода Хэмминга?
- Что такое ПТС?