

## Занятие №5. Циклический избыточный код. CRC

### 1. Цель работы

Получить представление о том, как осуществляется проверка на наличие ошибок в пакетах с данными в современных системах связи (Error detection) посредством использования циклического избыточного кода CRC (Cyclic Redundancy Check).

### 2. Краткие теоретические сведения

#### Псевдослучайные двоичные последовательности

**CRC — циклический избыточный код, иногда называемый также контрольным кодом или контрольной суммой.** CRC – это добавочная порция избыточных бит, вычисляемых по заранее известному алгоритму на основе исходного передаваемого пакета данных (информационной битовой последовательности), которое передаётся вместе с самим пакетом по каналам связи (добавляется после информационных битов) и служит для контроля его безошибочной передачи.

Простыми словами, CRC – это остаток от двоичного деления оригинального пакета с данными на какое-то двоичное  $n$ -разрядное число (порождающий полином), и его длина будет равна  $n-1$  бит. Рассмотрим пример, где имеется 7 бит данных: 100100 и 4-битный порождающий полином 1101. Требуется определить CRC. Для того, чтобы выполнить деление этих битовых последовательностей нужно в конце последовательности с данными добавить  $n-1$  нулей, как показано ниже, где  $n=4$ , для нашего случая.

Делитель - 1 1 0 1 | 1 0 0 1 0 0 0 0 0 - Делимое (данные+ $n-1$  нулей).

Основной операцией, используемой при делении бинарных чисел, является исключающее ИЛИ (XOR). Ниже показана таблица истинности для данной операции.

$x$	$y$	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Пошаговое вычисление CRC (*на стороне передатчика*):

1) 1 1 0 1 | 1 0 0 1 0 0 0 0  
       1 1 0 1 (*операция XOR*)

-----  
       1 0 0 0

2) 1 1 0 1 | 1 0 0 1 0 0 0 0  
       1 1 0 1

-----  
       1 0 0 0  
       1 1 0 1  
       -----  
       1 0 1 0

3) 1 1 0 1 | 1 0 0 1 0 0 0 0  
       1 1 0 1

-----  
       1 0 0 0  
       1 1 0 1  
       -----  
       1 0 1 0  
       1 1 0 1  
       -----  
       1 1 1 0

4) 1 1 0 1 | 1 0 0 1 0 0 0 0  
       1 1 0 1

-----  
       1 0 0 0  
       1 1 0 1  
       -----  
       1 0 1 0  
       1 1 0 1  
       -----  
       1 1 1 0  
       1 1 0 1  
       -----  
       0 1 1 0

При появлении 0, на следующем шаге делим на 0000.

$$5) \begin{array}{r} 1101 \mid 1001000000 \\ 1101 \\ \hline \end{array}$$

$$\begin{array}{r} 1000 \\ 1101 \\ \hline 1010 \\ 1101 \\ \hline 1110 \\ 1101 \\ \hline 0110 \\ 0000 \\ \hline 1100 \end{array}$$

$$6) \begin{array}{r} 1101 \mid 1001000000 \\ 1101 \\ \hline \end{array}$$

$$\begin{array}{r} 1000 \\ 1101 \\ \hline 1010 \\ 1101 \\ \hline 1110 \\ 1101 \\ \hline 0110 \\ 0000 \\ \hline 1100 \\ 1101 \\ \hline \end{array}$$

**001 – это и есть CRC, остаток от деления.**

Делитель принято записывать в виде полинома. Если считать, что каждый разряд делителя — это коэффициент полинома, то этот полином будет иметь вид:

$$x^{n-1} + x^{n-2} + \dots + x^2 + x^1 + x^0$$

Таким образом, делитель из примера выше можно записать в виде полинома как:  $1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$ , или сокращенно как:  $x^3 + x^2 + 1 = 1101$ .

Полученный остаток от деления CRC добавляется на передающей стороне к исходным данным и уже эта битовая последовательность, преобразованная в радиосигнал, передается в канал связи: 1 0 0 1 0 0 0 0 1.

На приемной стороне для обнаружения ошибки (или ее отсутствия) с полученным пакетом осуществляется ровно такая же процедура – деление на порождающий CRC полином. Если полученный в результате данного деления остаток будет *ненулевым*, то фиксируется факт *наличия ошибки*.

Пошаговое вычисление CRC (*на стороне приемника*):

$$\begin{array}{r}
 1) \quad 1101 | 100100001 \\
 \quad \quad 1101 \text{ (операция XOR)} \\
 \hline
 \quad \quad 1000
 \end{array}$$

$$\begin{array}{r}
 2) \quad 1101 | 100100001 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1000 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1010
 \end{array}$$

$$\begin{array}{r}
 3) \quad 1101 | 100100001 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1000 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1010 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1110
 \end{array}$$

$$\begin{array}{r}
 4) \quad 1101 | 100100001 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1000 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1010 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1110 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 0110
 \end{array}$$

При появлении 0, на следующем шаге делим на 0000.

$$\begin{array}{r}
 5) \quad 1101 \mid 100100001 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1000 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1010 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1110 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 0110 \\
 \quad \quad 0000 \\
 \hline
 \quad \quad 0101
 \end{array}$$

$$\begin{array}{r}
 6) \quad 1101 \mid 100100001 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1000 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1010 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 1110 \\
 \quad \quad 1101 \\
 \hline
 \quad \quad 0110 \\
 \quad \quad 0000 \\
 \hline
 \quad \quad 1101 \\
 \quad \quad 1101 \\
 \hline
 \end{array}$$

**000 – то есть, пакет передан без ошибок.**

### 3. Задание для выполнения практической работы

В рамках данной работы студенты должны научиться вычислять CRC-последовательности, а также на их основании детектировать ошибки.

#### Порядок выполнения работы:

- 1) Напишите программу на языке C/C++ для вычисления CRC для пакета данных длиной  $N$  бит ( $N = 20 + \text{порядковый номер в журнале}$ ) и определения факта наличия ошибки при передаче пакета по каналу связи.

- 2) Порождающий полином  $G$  для делителя выберите в соответствии с вариантом. Номер варианта – порядковый номер в журнале группы.

Табл. 5.1 Варианты заданий.

№ варианта	Непрерывная периодическая функция	№ варианта	Непрерывная периодическая функция
1	$G=x^7+x^5+x^4+x^3+x^2+x+1$	15	$G=x^7+x^5+x^2+x+1$
2	$G=x^7+x^6+x^4+x^3+x^2+x+1$	16	$G=x^7+x^2+x+1$
3	$G=x^7+x^6+x^5+x^3+x^2+x+1$	17	$G=x^7+x^6+x^2+x$
4	$G=x^7+x^6+x^5+x^4+x^2+x+1$	18	$G=x^7+x^6+x^5+x^4+x^3+x^2+x+1$
5	$G=x^7+x^6+x^5+x^4+x^3+x+1$	19	$G=x^7+x^6+x+1$
6	$G=x^7+x^6+x^5+x^4+x^3+x^2+1$	20	$G=x^7+x^6+x^5+x^3+x^2+x$
7	$G=x^7+x^6+x^5+x^4+x^3+x^2+x$	21	$G=x^7+x^4+x^3+x^2+x+1$
8	$G=x^6+x^5+x^4+x^3+x^2+x+1$	22	$G=x^7+x^6+x^5+x^4$
9	$G=x^7+x^5+x^3+x^2+x+1$	23	$G=x^7+x^6$
10	$G=x^7+x^6+x^4+x^3+x^2+x$	24	$G=x^7+x^5+x^4+x^3+x^2$
11	$G=x^7+x^5+x^4+x^3+x+1$	25	$G=x^7+x^3+x^2+x+1$
12	$G=x^7+x^6+x^4+x^3+x^2+x$	26	$G=x^7+x^6+x^4+x^3+1$
13	$G=x^7+x^5+x^4+x^3+x^2+1$	27	$G=x^7+x^6+x^4+x^3+x^2$
14	$G=x^7+x^6+x^3+x^2+x$	28	$G=x^7+1$

- 3) Добавьте полученный остаток от деления на  $G$  к пакету исходными данными и на приемной стороне вычислите повторно остаток от деления пакета с данными+CRC на полином  $G$ . Определите есть ли ошибка в принятом пакете. Выведите в окно терминала полученное значение CRC и отчет об ошибках в принятом пакете.
- 4) Возьмите  $N$ , равное 250 битам. Прodelайте п.1-3
- 5) Сделайте цикл из  $250+CRC \text{ length}$  итераций и в этом цикле по очереди искажайте по одному биту – с 0-го до  $250+CRC-1$ , проверьте в соответствии с п.3 обнаружена ли ошибка на приемной стороне и выполните подсчет того сколько раз за этот цикл приемник обнаружил и не обнаружил ошибки. Результат выведите в окно терминала.
- 6) Составьте отчет. Отчет должен содержать титульный лист, содержание, цель и задачи работы, теоретические сведения, исходные данные, этапы выполнения работы, сопровождаемые скриншотами и графиками, демонстрирующими успешность выполнения, и промежуточными выводами, результирующими таблицами, ответы на контрольные вопросы, и заключение и ссылка в виде QR-кода на репозиторий с кодом (git).

#### 4. Контрольные вопросы

- 1) Для чего в мобильных сетях используются CRC-проверки?
- 2) Что такое порождающий полином?
- 3) Как вычислить CRC для пакета с данными?

