Занятие № 5. “Циклические коды”

1. Принципы формирования циклических кодов

Циклические коды получили широкое распространение благодаря их эффективности при обнаружении и исправлении ошибок. Эти коды получили такое название потому, что основной операцией кодирования является циклическая перестановка.

Сущность циклической перестановки заключается в том, что последний символ кодовой комбинации занимает место первого, первый – второго, и т.д.

Если циклической перестановке подвергалась разрешенная кодовая комбинация, то в результате этой операции появляется новая разрешенная комбинация, что является основным свойством циклических кодов.

Для построения циклические коды достаточно знать порождающую матрицу. Но можно указать и другой способ построения циклических кодов, базирующийся на использовании неприводимых многочленов, в этом случае процесс кодирования сводится к отысканию многочлена (из известных неприводимых многочленов) соответствующего информационной последовательности.

Неприводимым называется многочлен, который не может быть представлен в виде произведения многочленов низших степеней, т. е. такой многочлен делится только на самого себя или на единицу и не делится ни на какой другой многочлен. На такой многочлен делится без остатка двучлен x^n+1.

Над многочленами можно производить все алгебраические действия.

При этом сложение двоичных многочленов сводится к сложению по модулю два коэффициентов при равных степенях переменной $\boldsymbol{x}$.

Умножение производится по обычному правилу перемножения степенных функций, однако полученные в этом случае коэффициенты при данной степени складываются по модулю два.

Деление осуществляется по правилам деления степенных функций, при этом операции вычитания заменяются операциями суммирования по модулю два.

При рассмотрении циклического кода, можно представить комбинацию двоичного кода не в виде последовательностей нулей и единиц, а в виде полинома некоторой степени:

$$V(x)=b_0\cdot x^0+b_1\cdot x^1+b_2\cdot x^2+b_3\cdot x^3+\ldots+b_{n-2}\cdot x^{n-2}+b_{n-1}\cdot x^{n-1}.$$

В этом случае циклическая перестановка есть результат умножения данного полинома на x:

$$x\cdot V(x)=x\cdot(b_0\cdot x^0+b_1\cdot x^1+b_2\cdot x^2+b_3\cdot x^3+\ldots+b_{n-2}\cdot x^{n-2}+b_{n-1}\cdot x^{n-1})=$$
$$=b_0\cdot x^1+b_1\cdot x^2+b_2\cdot x^3+b_3\cdot x^4+\ldots+b_{n-2}\cdot x^{n-1}+b_{n-1}\cdot x^n.$$

Однако в последнем члене необходимо заменить x^n на 1 (в противном случае длина кодовой комбинации превысит n).

Учитывая, замену x^n на 1 в произведении $b_{n-1}\cdot x^n$ получаем новый полином, коэффициенты которого образуют новую разрешенную комбинацию:

$$V'(x)=b_{n-1}\cdot x^0+b_0\cdot x^1+b_1\cdot x^2+b_2\cdot x^3+\ldots+b_{n-3}\cdot x^{n-2}+b_{n-2}\cdot x^{n-1}.$$

Полином $g(x)$ степени $r=n-k$, на который делится без остатка двучлен x^n+1, называют порождающим полиномом циклического кода.

Проверочный полином, образуется как результат деления двучлена на порождающий полином: $h(x)=\dfrac{x^n+1}{g(x)}$.

Произведение $g(x)\cdot h(x)=x^n\oplus 1=0$, поэтому полиномы $h(x)$ и $g(x)$ рассматривают как ортогональные и данная операция лежит в основе построения алгоритмов декодирования.

Рассмотрим как строится циклический код (7,4) взяв в качестве порождающего полинома $g(x)=x^3+x+1$.

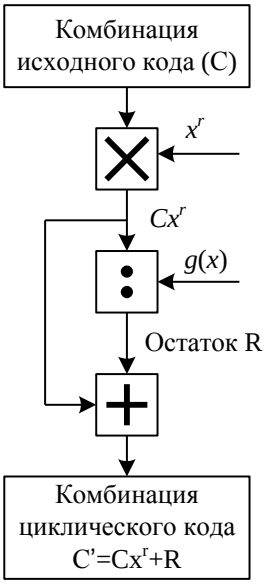
Докажем, что выбранный полином является порождающим, для этого произведем деление двучлена на выбранный полином:

$$
\begin{array}{l|l}
x^7+1 & x^3+x+1 \\
\cline{2-2}
x^7+x^5+x^4 & x^4+x^2+x+1 \\
\cline{1-1}
x^5+x^4+1 & \\
x^5+x^3+x^2 & \\
\cline{1-1}
x^4+x^3+x^2+1 & \\
x^4+x^2+x & \\
\cline{1-1}
x^3+x+1 & \\
x^3+x+1 & \\
\cline{1-1}
0 &
\end{array}
$$

т.к. результат деления получили без остатка, то $g(x)$ действительно порождающий полином.

Полученный полином $h(x)=x^4+x^2+x+1$ является проверочным.

Принцип обнаружения ошибок при помощи циклического кода заключается в том, что в качестве разрешенных принимаются только те комбинации, которые *без остатка* делятся на заранее выбранный ***образующий*** многочлен ***g(x)***. Если принимаемая комбинация искажена, то это условие на приемной стороне не будет выполнено. В результате этого формируется сигнал, указывающий на наличие ошибки.



Задача кодирования состоит в том, чтобы сформировать кодовые комбинации на передаче, удовлетворяющие указанному условию.

Метод построения кодовых комбинаций.

В процессе кодирования кодовая комбинация (***C***) отображающая двоичный код передаваемого сообщения (примитивный код) умножаются на x^r. При этом длина кодовой комбинации увеличивается на ***r*** разрядов. Эти дополнительные разряды будут проверочными. Полученное произведение $C\cdot x^r$ делят на специально подобранный образующий многочлен ***g(x)***. При этом получают остаток ***R***. Данный остаток ***R*** суммируют с произведением $C\cdot x^r$. Получают кодовую комбинацию ***C"***= $C\cdot x^r + R$, которая будет без остатка делиться на ***g(x)***. Алгоритм формирования комбинаций циклического кода показан на рис. 1.

Рис. 1. Алгоритм формирования циклического кода

Рассмотрим использование описанного алгоритма формирования комбинаций циклического кода на примере.

Пример. Требуется закодировать сообщение 1001. Дано: порождающий полином $g(x)=x^3+x+1$, общее число разрядов n = 7, число информационных разрядов k =4, число избыточных разрядов r = 3.

Решение:

для кодирования сообщения 1001 определим, какому многочлену оно соответствует:

$$C=1\cdot x^3+0\cdot x^2+0\cdot x^1+1\cdot x^0=x^3+1.$$

Умножаем кодовую комбинацию ***C*** на x^r.

$$C\cdot x^r=C\cdot x^3=(x^3+1)\cdot x^3=x^6+x^3.$$

Разделим полином $C\cdot x^r$ на порождающий полином ***g(x)*** для определения остатка ***R***:

$$
\begin{array}{l|l}
x^6+x^3 & x^3+x+1 \\
\cline{2-2}
\oplus\ x^6+x^4+x^3 & x^3+x \\
\cline{1-1}
x^4 & \\
\oplus\ x^4+x^2+x & \\
\cline{1-1}
x^2+x \text{ - остаток} & R=x^2+x
\end{array}
$$

Суммируем произведение $C\cdot x^r$ с полученным остатком $R=x^2+x$ получим кодовый многочлен:

$C' = x^6 + x^3 + x^2 + x^1$

В двоичном коде этому многочлену соответствует кодовая комбинация (a_6 = 1, a_5 = 0, a_4 = 0, a_3 = 1, a_2 = 1, a_1 = 1, a_0 = 0) = **1001_*110***. В этой кодовой комбинации последние три позиции занимают проверочные разряды (выделены курсивом).

2. Алгоритм декодирования циклического кода

Фундаментальным свойством любого циклического кода является то, что каждая разрешенная кодовая комбинация делится без остатка на порождающий полином.

Это свойство лежит в основе обнаружения и исправления ошибок циклическими кодами.

Рассмотрим деление кодовой комбинации на порождающий полином в алгебраической форме записи и в двоичных числах.

Разделим кодовую комбинацию, которая была принята без ошибок $C'' = 1001\_110$, т.е. $x^6 + x^3 + x^2 + x$ на порождающий полином $g(x) = x^3 + x + 1$:

$$
\begin{array}{l|l}
x^6 + x^3 + x^2 + x & x^3 + x + 1 \\
\underline{x^6 + x^4 + x^3} & \underline{} \\
x^4 + x^2 + x & x^3 + x \\
\underline{x^4 + x^2 + x} & \\
0 \quad \text{(деление произведено без остатка)} &
\end{array}
$$

Т.к. принятая кодовая комбинация после деления на $g(x)$ дала результат без остатка, это подтверждает, что в принятой кодовой комбинации ошибок нет.

Рассмотрим пример, когда произошла ошибка во втором разряде, т.е. принята комбинация $C'' = 1101\_110$, тогда используя $g(x) = x^3 + x + 1$ получаем:

$$
\begin{array}{l|l}
x^6 + x^5 + x^3 + x^2 + x & x^3 + x + 1 \\
\underline{x^6 + x^4 + x^3} & \underline{} \\
x^5 + x^4 + x^2 + x & x^3 + x^2 + \mathrm{x} + 1 \\
\underline{x^5 + x^3 + x^2} & \\
x^4 + x^3 + x & \\
\underline{x^4 + x^2 + x} & \\
x^3 + x^2 & \\
\underline{x^3 + x + 1} & \\
x^2 + x + 1 \quad \text{остаток} &
\end{array}
$$

При делении принятой кодовой комбинации циклического кода на порождающий многочлен был получен остаток, значит, имеет место ошибка. Таким образом, остатки от деления принятой комбинации на порождающий полином являются опознавателями ошибок циклических кодов. Но данные остатки еще не указывают непосредственно на место ошибки в кодовой комбинации.

В циклических кодах алгоритм исправления ошибок основан на следующем. Ошибочная комбинация после определенного числа циклических сдвигов в сумме с остатком дает исправленную комбинацию. Остаток при этом представляет собой разницу между искаженными и правильными символами, а единицы в остатке стоят на местах искаженных разрядов в «подогнанной» циклическими сдвигами комбинации. Преобразования с искаженной комбинацией выполняются до тех пор, пока число единиц в остатке не будет равно числу ошибок в коде. При этом, естественно; число единиц может быть равно числу ошибок $\boldsymbol{t}$, исправляемых данным кодом или меньше.

Таким образом, для обнаружения и исправления ошибочного разряда производят следующие операции:

принятую комбинацию делят на порождающий полином;

подсчитывают количество единиц в остатке (вес остатка ω):

если $\omega \le t$, принятую комбинацию складывают по модулю два с полученным остатком, сумма дает исправленную комбинацию;

если $\omega > t$, то производят циклический сдвиг принятой комбинации влево на один разряд. Комбинацию, полученную в результате циклического сдвига, делят на образующий полином $g(x)$. Если в результате повторного деления $\omega \le t$, то делимое суммируют с остатком;

затем производят циклический сдвиг вправо на один разряд комбинации, полученной в результате суммирования последнего делимого с последним остатком. Полученная комбинация уже не содержит ошибок. Если после первого циклического сдвига и последующего деления остаток получается таким, что его вес $\omega > t$ повторяют предыдущую операцию до тех пор, пока не будет достигнуто $\omega \le t$. В этом случае комбинацию, полученную в результате последнего циклического сдвига, суммируют с остатком от деления этой комбинации на образующий многочлен;

производят циклический сдвиг вправо ровно настолько разрядов, на сколько сдвинута суммируемая с последним остатком комбинация, относительно принятой комбинации. В результате получим исправленную комбинацию.

Пример 2.5. При передаче комбинации $C' = 1001\_110$ циклического кода, исправляющего одиночные ошибки $(t = 1)$, полученного с помощью образующего полинома $g(x) = x^3 + x + 1$, произошла ошибка во втором разряде. Принятая комбинация имеет вид $C'' = 1101\_110$. Процесс исправления ошибки следующий.

1. Делим принятую комбинацию на образующий полином:

```
1101110 | 1011
1011    |
 1101
 1011
  1101
  1011
   1100
   1011
    111   остаток
```

2. Сравниваем вес полученного остатка $\omega = 3$ с числом исправляемых ошибок $t = 1$, т.е. $\omega > t$.

3. Производим циклический сдвиг принятой комбинации на один разряд влево и деление на $g(x)$:

```
1011101 | 1011
1011    |
   101   остаток
```

Сравниваем вес полученного остатка $\omega = 2$ с числом исправляемых ошибок $t = 1$, т.к. $\omega > t$.

4. Повторяем п. 3 до тех пор, пока не будет получено $\omega \le t$:

```
0111011 | 1011
 1011   |
  1011
  1011
     1   остаток
```

Сравниваем вес полученного остатка $\omega = 1$ с числом исправляемых ошибок $t = 1$, полученные значения равны $\omega = t$.

5. Складываем по модулю два последнее делимое с последним остатком:

```
  0111011
⊕ 0000001
---------
  0111010
```

6. Производим циклический сдвиг комбинации, полученной в результате суммирования последнего делимого с последним остатком, вправо на два разряда (так как перед этим мы дважды сдвигали принятую комбинацию влево): $0111010 \to 0011101 \to 1001110$. Как видим, последняя комбинация соответствует переданной, т.е, не содержит ошибки.