Лабораторна робота №4

з дисципліни «Теорія інформації та

кодування»

«Циклічні коди»

Виконав:

студент групи ДА-51

Факультету «ІПСА»

Болобан Олег

#### Мета роботи:

вивчити методи побудови циклічного коду та його програмну реалізацію.

#### Формуючий многочлен:

|  |
| --- |
| 110111 |

,

#### Методика побудови циклічних кодів діленням на формуючий многочлен

Інформаційний многочлен P(x) множиться на xr, де r - старша степінь формуючого многочлена g(x), і отриманий вираз ділимо на g(x). В результаті вийде часткове Q(x) і залишок R(x) степені менше r:

Перенесем R(x) в ліву частину:

Права частина останньої рівності кратна g(x) і, таким чином, є кодовим многочленом F(x):

Саме так отримуються кодові многочлени. Перший доданок кодового многочлена має нульові коефіцієнти в r молодших членах. Цей многочлен відповідає здвигнутій вліво на r розрядів інформаційної частини P(x). Степінь многочлена R(x) менше r, тому ці коефіцієнти не мають нульові коефіцієнти першого многочлена при їхньому додаванні. Таким чином, інформаційні елементи в кодовій комбінації зберігаються, а перевірочними елементами є коефіцієнти залишку R(x), число яких дорівнюватиме степеню формуючого многочлена.

#### Властивості циклічних кодів:

1. Мінімальна кодова відстань d циклічного коду, не перевищує число членів формуючого многочлена.
2. Циклічний код з формуючим многочленом степеню r>1 виявляє будь-яку одиничну і будь-яку подвійну помилку, тобто має d>3.
3. Код з формуючим многочленом x + 1 є кодом з кратним числом одиниць.
4. Циклічний код з формуючим многочленом g(x)(x+1) має d>4.
5. Код з формуючим многочленом g(x) степеню r виявляє всі пакети помилок довжини r чи менше.

#### Алгоритм визначення помилки

Нехай маємо n-елементарних комбінації тоді:

1. Отримаємо остачу від ділення Е(х) що відповідає помилці в старшому розряді [1000000000], на формуючий поліном Pr(x):
2. Ділимо отриманий поліном Н(х) на Pr(x) і отримаємо остачу R(x).
3. Порівнюємо R0(x) і R(x).
   * Якщо вони рівні, то помилка виникла в старшому розряді.
   * Якщо ні, то збільшуємо степінь прийнятого полінома на х і знову проводимо ділення:
4. Знову порівняємо отриманий залишок з R0(x).
   * Якщо вони рівні, то помилка виникла в другому розряді.
   * Якщо ні, то множимо Н(х) • х2 і повторюємо ці операції поки R(x) не буде рівний R0(x).

Помилка буде в розряді, що відповідає відповідному числу, на яке підвищена степінь Н(х) плюс один.

Наприклад:

#### Опис програмної реалізації

Програма реалізує кодування методом ділення на формуючий многочлен. А також виявляє й виправляє одну помилку за вказаним алгоритмом. А також будує формуючу та перевірочну матриці.

#### Лістинг

//При несистематическом кодировании кодовое слово получается в виде произведения информационного полинома на порождающий

#include <iostream>

#include <vector>

#include <conio.h>

using namespace std;

vector <bool> Resault;

vector <bool> Polynom;

vector <bool> Rest;

vector <bool> BoolTransform (string str){ //запись строки полинома

vector <bool> v;

for(int i=0; i< str.size(); i++)

if(str[i]=='1')

v.push\_back(1);

else

v.push\_back(0);

return v;

}

vector <bool> multiple(vector <bool> v)

{

vector <bool> s;

for(int i=0; i<v.size(); i++)

s.push\_back(v[i]);

for(int i=0; i<Polynom.size()-1; i++)

s.push\_back(0);

return s;

}

void divide(vector <bool> str, vector <bool> bgn=Polynom, int tmp=0)//деление на производный многочлен

{

if(!str[tmp])

{

Resault.push\_back(0);

if(tmp+bgn.size()<str.size())

divide(str, bgn, tmp+1);

else

Rest=str;

}

else

{

Resault.push\_back(1);

for(int i=0; i<bgn.size(); i++)

{

str[tmp+i]=str[tmp+i] ^ bgn[i];

}

if(tmp+bgn.size()<str.size())

divide(str, bgn, tmp+1);

else

Rest=str;

}

}

void repair(vector <bool> str){ //нахождение ошибки кода

int count=0;

for(int i=0; i<Rest.size(); i++)

if(Rest[i])

count++;

if(!count){

for(int i=0; i<str.size(); i++)

cout<<str[i];

cout<<endl;

}

else{

count=0;

vector <bool> b;

b.push\_back(1);

for(int i=0; i<str.size()-1; i++)

b.push\_back(0);

divide(b);

vector <bool> Ro=Rest;

while(1){

divide(str);

Rest.erase(Rest.begin(), Rest.begin()+count);

if(Rest==Ro){

str[count]=str[count]^1;

break;

}

count++;

str.push\_back(0);

}

}

str.erase(str.end()-count, str.end());

for(int i=0; i<str.size(); i++)

cout<<str[i];

cout<<endl;

}

vector <bool> go\_left(vector <bool> vector){//перемещение по вектору

bool b=vector[0];

for(int i=0; i<vector.size()-1; i++)

vector[i]=vector[i+1];

vector[vector.size()-1]=b;

return vector;

}

vector <bool> go\_right(vector <bool> vector){ //перемещение по вектору

bool b=vector[vector.size()-1];

for(int i=vector.size()-1; i>0; i--)

vector[i]=vector[i-1];

vector[0]=b;

return vector;

}

int main(){

Polynom=BoolTransform("110111");//Порождающий полином

string cmd;

cin>>cmd;

vector <bool> Comm=BoolTransform(cmd);

int n;

cout<<endl<<"1 Form the Code;"<<endl<<"2 Repair broken Code;"

<<endl<<"3 Form the matrix; "<<endl;

cin>>n;

switch(n)

{ case 1:

Comm=multiple(Comm);

divide(Comm);

for(int i=0; i<Comm.size(); i++)

{ Comm[i]=Comm[i] || Rest[i];

cout<<Comm[i];

}

cout<<endl;

break;

case 2://Алгоритм выявления ошибки

divide(Comm);

repair(Comm);

break;

case 3:

Comm.clear();

Comm.push\_back(1);

for(int i=0; i<11; i++)

Comm.push\_back(0);

divide(Comm);

cout<<"Forming"<<endl;//Порождающая матрица

for(int i=0;i<Polynom.size();i++)

{

cout<<" "<<i+1<<"\t";

for(int j=0;j<Resault.size();j++)

cout<<Resault[j];

Resault=go\_right(Resault);

cout<<endl;

}

cout<<endl;

Resault=go\_left(Resault);

cout<<"Verify"<<endl;//Проверочная матрица

for(int i=0;i<Polynom.size();i++)

{

cout<<" "<<i+1<<"\t";

for(int j=0;j<Resault.size();j++)

cout<<Resault[j];

Resault=go\_left(Resault);

cout<<endl;

}

break;

}

\_getch();

}

1101110111010110111101010

1

*1101010101010110110111010101010*

1001010111010110110101010101010

2

*1001010101010110110101010101010*

*Forming*

*1 111011*

*2 111101*

*3 111110*

*4 011111*

*5 101111*

*6 110111*

*7 111011*

*Verify*

*1 111011*

*2 110111*

*3 101111*

*4 011111*

*5 111110*

*6 111101*

*7 111011*

#### Висновок

При виконанні даної лабораторної роботи, було розкрите поняття циклічних кодів. Було розроблено програму, що реалізує кодування бінарного повідомлення методом циклічних кодів. Окрім цього, програма має можливість виправлення однієї помилки, допущеної при передачі кодованого повідомлення; побудови формуючої та перевірочної матриць.