Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

ЛАБОРАТОРНА РОБОТА №4

З дисципліни«Криптографія»

«Побудова регістрів зсуву з лінійним зворотним зв’язком та дослідження їх властивостей»

Виконали:
студенти 3 курсу ФТІ
групи ФБ-73
АбкерімовЕрвін
Білоконь Богдан

Перевірив:
Чорний О.

Мета роботи:

Ознайомлення з принципами побудови регістрів зсуву з лінійним зворотним зв'язком; практичне освоєння їх програмної реалізації; дослідження властивостей лінійних рекурентних послідовностей та їх залежності від властивостей характеристичного полінома регістра.

Порядок виконання роботи:

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
1. Вибрати свій варіант завдання згідно зі списком. Варіанти завдань містяться у файлі Crypto_CP4 LFSR_Var.
2. За даними характеристичними многочленами p1(x), p2(x) скласти лінійні рекурентні співвідношення для ЛРЗ, що задаються цими характеристичними многочленами.
3. Написати програми роботи кожного з ЛРЗ L1 , L2 .
4. За допомогою цих програм згенерувати імпульсні функції для кожного з ЛРЗ і підрахувати їх періоди.
5. За отриманими результатами зробити висновки щодо влавстивостей кожного з характеристичних многочленів p1(x), p2(x): многочлен примітивний над F2 ; не примітивний, але може бути незвідним; звідний.
6. Для кожної з двох імпульсних функцій обчислити розподіл k-грам на періоді, k≤ni, де ni - степінь полінома fi(x), i=1,2 а також значення функції автокореляції A(d) для 0≤ d ≤ 10. За результатами зробити висновки.

Варіант 1

P1(X)= X20+X16+X14+X12+X10+X7+X6+X+1

P2(X)= X24 + X21 + X12 + X11 + X10 + X7 + X2 + X + 1

Довжини періодів імпульсних функцій

L_1 1048575

L_2 3355443

Значення функцій автокореляції Ad(s) для 0≤ d ≤ 10, для відповідних імпульсних функцій:

Значенняd	*Ad(s) дляімп.ф-ції L_1*	*Ad(s) дляімп.ф-ції L_2*
0	0	0
1	524288	1677312
2	524288	1677312

3	524288	1677312
4	524288	1677312
5	524288	1677312
6	524288	1679360
7	524288	1677312
8	524288	1677312
9	524288	1677312
10	524288	1677312

КграмиL1

"11" f : 25% | c : 262144
"01" f : 25% | c : 262144
"10" f : 25% | c : 262143
"00" f : 25% | c : 262143
"101" f : 12.5% | c : 131072
"010" f : 12.5% | c : 131072
"001" f : 12.5% | c : 131072
"111" f : 12.5% | c : 131072
"011" f : 12.5% | c : 131072
"110" f : 12.4999% | c : 131071
"100" f : 12.4999% | c : 131071
"000" f : 12.4999% | c : 131071
"1011" f : 6.25002% | c : 65536
"1101" f : 6.25002% | c : 65536
"0110" f : 6.25002% | c : 65536
"1111" f : 6.25002% | c : 65536
"0111" f : 6.25002% | c : 65536
"0001" f : 6.25002% | c : 65536
"0010" f : 6.25002% | c : 65536
"0100" f : 6.25002% | c : 65536
"0101" f : 6.25002% | c : 65536
"1010" f : 6.25002% | c : 65536
"1001" f : 6.25002% | c : 65536
"0011" f : 6.25002% | c : 65536
"1100" f : 6.24993% | c : 65535
"1110" f : 6.24993% | c : 65535
"0000" f : 6.24993% | c : 65535
"1000" f : 6.24993% | c : 65535
"11111" f : 3.12501% | c : 32768
"01100" f : 3.12501% | c : 32768
"11101" f : 3.12501% | c : 32768
"10110" f : 3.12501% | c : 32768
"11011" f : 3.12501% | c : 32768
"10111" f : 3.12501% | c : 32768
"01011" f : 3.12501% | c : 32768
"11010" f : 3.12501% | c : 32768
"01101" f : 3.12501% | c : 32768
"00110" f : 3.12501% | c : 32768
"10101" f : 3.12501% | c : 32768
"01111" f : 3.12501% | c : 32768
"10011" f : 3.12501% | c : 32768
"11001" f : 3.12501% | c : 32768
"01110" f : 3.12501% | c : 32768
"00111" f : 3.12501% | c : 32768
"00011" f : 3.12501% | c : 32768
"10010" f : 3.12501% | c : 32768
"01001" f : 3.12501% | c : 32768
"00001" f : 3.12501% | c : 32768
"00010" f : 3.12501% | c : 32768
"00100" f : 3.12501% | c : 32768
"01000" f : 3.12501% | c : 32768
"10001" f : 3.12501% | c : 32768
"00101" f : 3.12501% | c : 32768
"01010" f : 3.12501% | c : 32768
"10100" f : 3.12501% | c : 32768
"11000" f : 3.12492% | c : 32767
"11110" f : 3.12492% | c : 32767
"11100" f : 3.12492% | c : 32767
"10000" f : 3.12492% | c : 32767
"00000" f : 3.12492% | c : 32767

КграмиL2

"00" f : 25.0183% | c : 839475
"11" f : 24.9939% | c : 838656
"01" f : 24.9939% | c : 838656
"10" f : 24.9939% | c : 838655
"000" f : 12.5214% | c : 420147
"101" f : 12.497% | c : 419328
"010" f : 12.497% | c : 419328
"001" f : 12.497% | c : 419328
"111" f : 12.497% | c : 419328
"011" f : 12.497% | c : 419328
"110" f : 12.4969% | c : 419327
"100" f : 12.4969% | c : 419327
"0000" f : 6.27289% | c : 210483
"1111" f : 6.24848% | c : 209664
"1011" f : 6.24848% | c : 209664

"0101" f : 6.24848% | c : 209664
"1010" f : 6.24848% | c : 209664
"1101" f : 6.24848% | c : 209664
"0001" f : 6.24848% | c : 209664
"0010" f : 6.24848% | c : 209664
"0100" f : 6.24848% | c : 209664
"1001" f : 6.24848% | c : 209664
"0011" f : 6.24848% | c : 209664
"0111" f : 6.24848% | c : 209664
"1110" f : 6.24848% | c : 209664
"0110" f : 6.24845% | c : 209663
"1100" f : 6.24845% | c : 209663
"1000" f : 6.24845% | c : 209663
"00000" f : 3.1372% | c : 105267
"11111" f : 3.13569% | c : 105216
"01110" f : 3.13569% | c : 105216
"00001" f : 3.13569% | c : 105216
"10000" f : 3.13566% | c : 105215
"00101" f : 3.12806% | c : 104960
"10100" f : 3.12806% | c : 104960
"01010" f : 3.12806% | c : 104960
"10011" f : 3.12806% | c : 104960
"11011" f : 3.12806% | c : 104960
"01101" f : 3.12806% | c : 104960
"11001" f : 3.12806% | c : 104960
"11100" f : 3.12806% | c : 104960
"00111" f : 3.12806% | c : 104960
"00010" f : 3.12806% | c : 104960
"01000" f : 3.12806% | c : 104960
"10110" f : 3.12803% | c : 104959
"00110" f : 3.12043% | c : 104704
"10111" f : 3.12043% | c : 104704
"01011" f : 3.12043% | c : 104704
"10101" f : 3.12043% | c : 104704
"11010" f : 3.12043% | c : 104704
"11101" f : 3.12043% | c : 104704
"00100" f : 3.12043% | c : 104704
"01001" f : 3.12043% | c : 104704
"10010" f : 3.12043% | c : 104704
"00011" f : 3.12043% | c : 104704
"11000" f : 3.1204% | c : 104703
"01100" f : 3.1204% | c : 104703
"11110" f : 3.1128% | c : 104448
"01111" f : 3.1128% | c : 104448
"10001" f : 3.1128% | c : 104448

L_1 - примітивне, незвідне

L_2 - не примітивне, може бути незвідним

Код

```
#include<iostream>
using namespace std;

int main()
{
int j;
cin>> j;
if (j == 1) {
int a[20];
a[0] = 1;a[1] = 1;a[2] = 0;a[3] = 0;
a[4] = 0;a[5] = 0;a[6] = 1;a[7] = 1;
a[8] = 0;a[9] = 0;a[10] = 1;a[11] = 0;
a[12] = 1;a[13] = 0;a[14] = 1;a[15] = 0;
a[16] = 1;a[17] = 0;a[18] = 0;a[19] = 0;

int* s;
s = new int[2000000];
for (inti = 0; i< 19; i++) {
s[i] = 0;
}
s[19] = 1;
for (inti = 0; i< 1999980; i++) {
```

```
s[i + 20] = ((a[19] * s[i + 19]) + (a[18] * s[i + 18]) + (a[17] * s[i + 17]) + (a[16] * s[i + 16]) + (a[15] * s[i
+ 15]) + (a[14] * s[i + 14]) + (a[13] * s[i + 13]) + (a[12] * s[i + 12]) + (a[11] * s[i + 11]) + (a[10] * s[i +
10]) + (a[9] * s[i + 9]) + (a[8] * s[i + 8]) + (a[7] * s[i + 7]) + (a[6] * s[i + 6]) + (a[5] * s[i + 5]) + (a[4] *
s[i + 4]) + (a[3] * s[i + 3]) + (a[2] * s[i + 2]) + (a[1] * s[i + 1]) + (a[0] * s[i + 0])) % 2;
}

for (int k = 20; k < 1999980; k++) {
if (s[k] == 0) {
if (s[k + 1] == 0) {
if (s[k + 2] == 0) {
if (s[k + 3] == 0) {
if (s[k + 4] == 0) {
if (s[k + 5] == 0) {
if (s[k + 6] == 0) {
if (s[k + 7] == 0) {
if (s[k + 8] == 0) {
if (s[k + 9] == 0) {
if (s[k + 10] == 0) {
if (s[k + 11] == 0) {
if (s[k + 12] == 0) {
if (s[k + 13] == 0) {
if (s[k + 14] == 0) {
if (s[k + 15] == 0) {
if (s[k + 16] == 0) {
if (s[k + 17] == 0) {
if (s[k + 18] == 0) {
if (s[k + 19] == 1) {
cout<< k<<endl;
}}}}}}}}}}}}}}}}}}}}
int b = 0;
for (int d = 0; d < 11; d++) {
for (inti = 0; i<= 1048574; i++) {
b = b + (s[i] + s[(i + d) % 1048575]) % 2;
}
cout<<endl<< b <<endl;
}
delete[] s;
}
else if (j == 2) {
int a[24];
a[0] = 1;a[1] = 1;a[2] = 1;a[3] = 0;
a[4] = 0;a[5] = 0;a[6] = 0;a[7] = 1;
a[8] = 0;a[9] = 0;a[10] = 1;a[11] = 1;
a[12] = 1;a[13] = 0;a[14] = 0;a[15] = 0;
a[16] = 0;a[17] = 0;a[18] = 0;a[19] = 0;
a[20] = 0;a[21] = 1;a[22] = 0;a[23] = 0;

int* s;
```

```
s = new int[4000000];
for (inti = 0; i< 23; i++) {
s[i] = 0;
}
s[23] = 1;

for (inti = 0; i< 3999976; i++) {

s[i + 24] = ((a[23] * s[i + 23]) + (a[22] * s[i + 22]) + (a[21] * s[i + 21]) + (a[20] * s[i + 20]) + (a[19] * s[i
+ 19]) + (a[18] * s[i + 18]) + (a[17] * s[i + 17]) + (a[16] * s[i + 16]) + (a[15] * s[i + 15]) + (a[14] * s[i +
14]) + (a[13] * s[i + 13]) + (a[12] * s[i + 12]) + (a[11] * s[i + 11]) + (a[10] * s[i + 10]) + (a[9] * s[i + 9])
+ (a[8] * s[i + 8]) + (a[7] * s[i + 7]) + (a[6] * s[i + 6]) + (a[5] * s[i + 5]) + (a[4] * s[i + 4]) + (a[3] * s[i +
3]) + (a[2] * s[i + 2]) + (a[1] * s[i + 1]) + (a[0] * s[i + 0])) % 2;
if (s[i + 20] < 0) {
s[i + 20] = -s[i + 20];
}

}
for (int k = 24; k < 3999976; k++) {
if (s[k] == 0) {
if (s[k + 1] == 0) {
if (s[k + 2] == 0) {
if (s[k + 3] == 0) {
if (s[k + 4] == 0) {
if (s[k + 5] == 0) {
if (s[k + 6] == 0) {
if (s[k + 7] == 0) {
if (s[k + 8] == 0) {
if (s[k + 9] == 0) {
if (s[k + 10] == 0) {
if (s[k + 11] == 0) {
if (s[k + 12] == 0) {
if (s[k + 13] == 0) {
if (s[k + 14] == 0) {
if (s[k + 15] == 0) {
if (s[k + 16] == 0) {
if (s[k + 17] == 0) {
if (s[k + 18] == 0) {
if (s[k + 19] == 0) {
if (s[k + 20] == 0) {
if (s[k + 21] == 0) {
if (s[k + 22] == 0) {
if (s[k + 23] == 1) {
cout<< k <<endl;
}}}}}}}}}}}}}}}}}}}}}}}}
int b = 0;
for (int d = 0; d < 11; d++) {
for (inti = 0; i<= 3355442; i++) {
b = b + (s[i] + s[(i + d) % 3355443]) % 2;
```

```
}
cout<<endl<< b <<endl;
}
delete[] s;
}
else cout<< "error";
return 0;
}
```

Висновок:

На цій лабораторній роботі ми:

- ознайомлсь з принципами побудови регістрів зсуву з лінійним зворотним зв’язком;
- на практиці освоєл їх програмної реалізації;
- дослідли властивості
  лінійних рекурентних послідовностей та їх залежності від властивостей характеристичного полінома регістра.