Лабораторная работа 12

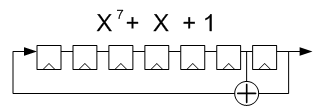
Генератор псевдослучайных чисел

**Ход мыслей**

Задача состоит в генерации псевдослучайных чисел. Т.к. уже доводилось использовать полиномы “prbs” для генерации псевдослучайного сигнала, было решено адаптировать данный вид полиномов под текущие нужды.

Есть несколько полиномов вида “prbs”. Отличаются они размерностью – “prbs7”, “prbs23”, “prbs31”. Цифра в конце означает количество бит – это значение на 1 меньше, чем значение кратное байтам (8). Это объясняется установкой результата операции xor в недостающую ячейку, а также увеличением случайности (т.к. потом информация будет преобразовываться в байты). Второе отличие – это какая именно пара бит будет взята для операции xor.

Пример: полином «prbs7»



В данном случае размер массива бит равен 7. Выбираются первый и последний бит.

* Производится операция xor между выбранными битами.
* Выполняется побитовый сдвиг на 1 вправо (Т.е. самый правый бит будет «отброшен», для самого левого не будет значения. Все остальные биты просто сдвинуться на 1 вправо).
* На место самого левого бита устанавливается результат операции xor.

**Реализация**

Т.к. предстоит генерировать числа, а не битовый сигнал – размерность массива бит была увеличена до целого числа байт (а именно – 4 (вместимость int и uint)).

Класс имеет 1 поле “Seed” – хранит значение из которого будет получено следующее число.

Есть 2 конструктора. Первый вариант получает стартовое число «Seed» из текущего времени системы. Второй конструктор даёт явно задать значение «Seed»

Метод «Next(uint)» генерирует из «Seed» следующее число, сохраняет полученное значение в «Seed» и возвращает остаток от деления на число, переданное в параметрах (т.е. будет возвращено число от 0 до заданного - 1). Сама генерация преобразовывает uint «Seed» в массив бит, выполняет битовый сдвиг влево на 1 и на 0 место устанавливает результат исключающего или (xor) между 31 и 28 значением.

Код:

public class Generator

{

private uint Seed;

public Generator()

{

Seed = (uint)DateTime.Now.TimeOfDay.Ticks;

}

public Generator(uint seed)

{

Seed = seed;

}

//prbs31 = X31+X28+1

public uint Next(uint mod)

{

//Seed >>= 1;

BitArray bits = new(BitConverter.GetBytes(Seed));//1

bits.LeftShift(1);//1

bits[0] = bits[31] ^ bits[28];

uint[] uints = new uint[1];

bits.CopyTo(uints, 0);

Seed = uints[0];

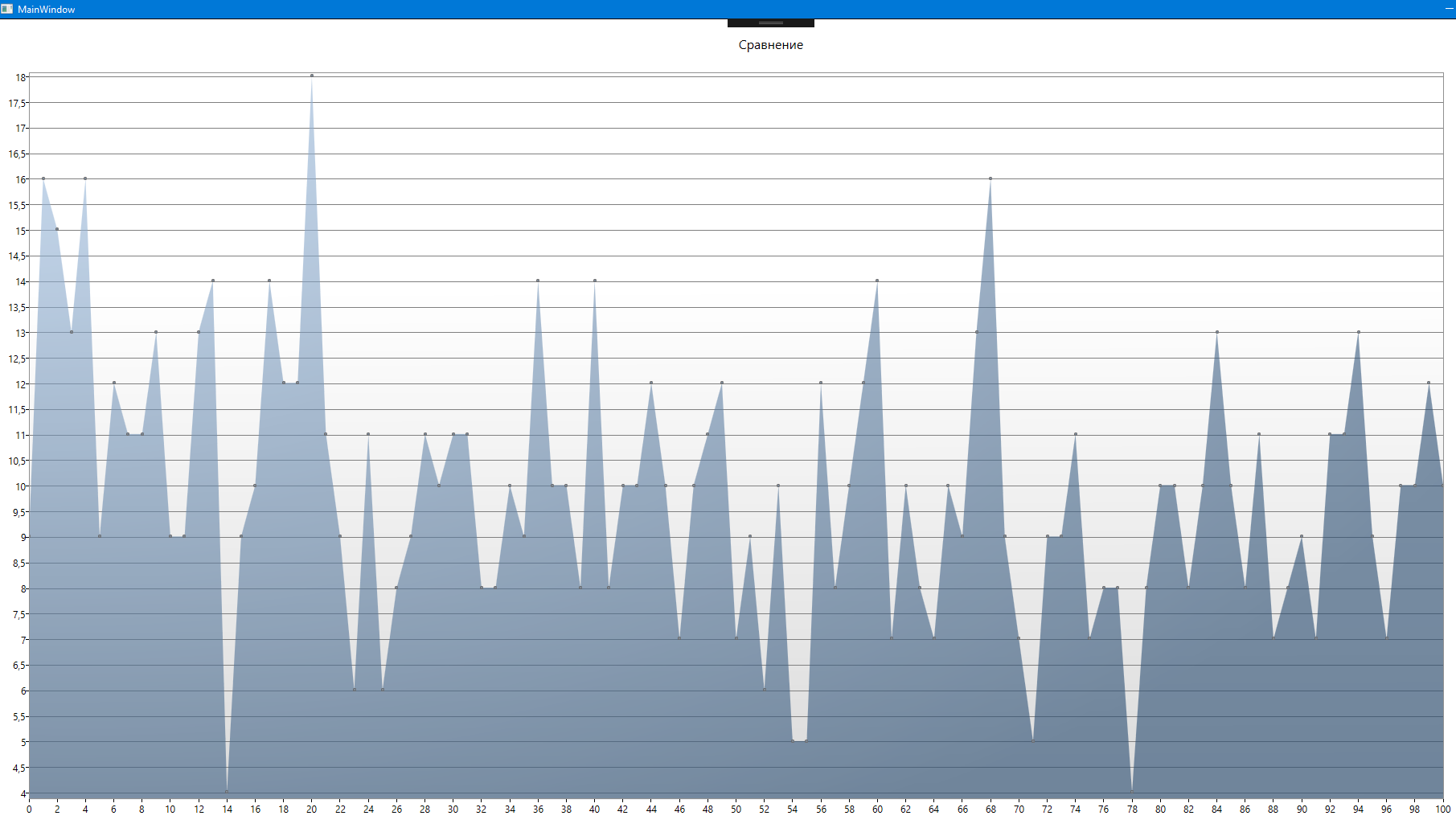
return Seed % mod;

}

}

**Визуальная интерпретация:**

Для демонстрации была сгенерирована 1000 значений в диапазоне [0; 100]. Подсчитано количество получений каждого числа:



Полученные значения:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дес\ед | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Сумма |
| 0 | 9 | 16 | 15 | 13 | 16 | 9 | 12 | 11 | 11 | 13 | 125 |
| 1 | 9 | 16 | 15 | 13 | 16 | 9 | 10 | 14 | 12 | 12 | 106 |
| 2 | 18 | 11 | 9 | 6 | 11 | 6 | 8 | 9 | 11 | 10 | 99 |
| 3 | 11 | 11 | 8 | 8 | 10 | 9 | 14 | 10 | 10 | 8 | 99 |
| 4 | 14 | 8 | 10 | 10 | 12 | 10 | 7 | 10 | 11 | 12 | 104 |
| 5 | 7 | 9 | 6 | 10 | 5 | 5 | 12 | 8 | 10 | 12 | 84 |
| 6 | 14 | 7 | 10 | 8 | 7 | 10 | 9 | 13 | 16 | 9 | 103 |
| 7 | 7 | 5 | 9 | 9 | 11 | 7 | 8 | 8 | 4 | 8 | 76 |
| 8 | 10 | 10 | 8 | 10 | 13 | 10 | 8 | 11 | 7 | 8 | 95 |
| 9 | 9 | 7 | 11 | 11 | 13 | 9 | 7 | 10 | 10 | 12 | 99 |
| 10 | 10 |  |  |  |  |  |  |  |  |  | 10 |

**Математическое обоснование:**

Для подтверждения случайности получения чисел используем критерий Пирсона – Х2.

Для удобства будут взяты интервалы: [0;9], [10;19], [20;29], [30;39], [40;49], [50;59], [60;69], [70;79], [80;89], [90;100].

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | Наблюдаемая частота | Ожидаемая частота | Мера отклонения |
| [0;9] | 125 | 100 | 6.25 |
| [10;19] | 106 | 100 | 0.36 |
| [20;29] | 99 | 100 | 0.01 |
| [30;39] | 99 | 100 | 0.01 |
| [40;49] | 104 | 100 | 0.16 |
| [50;59] | 84 | 100 | 2.56 |
| [60;69] | 103 | 100 | 0.09 |
| [70;79] | 76 | 100 | 5.76 |
| [80;89] | 95 | 100 | 0.25 |
| [90;100] | 109 | 100 | 0.81 |
| Всего: | 1000 | 1000 | 16.26 |

Табличные параметры выбраны наиболее распространенные: p = 0.05. Количество степеней свободы: k = 10 – 1 = 9.

X2табл. = 16,919 > X2получ. = 16.26

Т.к. расчётное значение отклонения меньше табличного – наблюдаемые частоты вписываются в пределы естественных случайных отклонений. (Если быть наиболее точным – данный генератор можно считать «случайным» с вероятностью 0,95)