ЛАБОРАТОРНАЯ РАБОТА № 12

РЕГИСТРЫ СДВИГА С ЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ КАК ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Цель работы: изучение принципа работы генератора псевдослучайных последовательностей, основанного на регистре сдвига с линейной обратной связью.

Описание лабораторной работы

Общие сведения о регистрах сдвига с линейной обратной связью. Последовательности регистров сдвига используются как в криптографии, так и в теории кодирования. Их теория прекрасно проработана, потоковые шифры на базе регистров сдвига являлись рабочей лошадкой военной криптографии задолго до появления электроники.

Регистр сдвига с обратной связью (*далее РгСсОС*) *состоит из двух частей*: *регистра сдвига и функции обратной связи* (рис. 3.8). Регистр сдвига представляет собой последовательность битов. Количество битов определяется ***длиной сдвигового регистра***, если длина равна ***n*** битам, то регистр называется ***n-битовым сдвиговым регистром***. Всякий раз, когда нужно извлечь бит, все биты сдвигового регистра сдвигаются вправо на одну позицию. Новый крайний левый бит является функцией всех остальных битов регистра. На выходе сдвигового регистра оказывается один, обычно младший значащий, бит. ***Периодом сдвигового регистра*** называется длина получаемой последовательности до начала ее повторения.

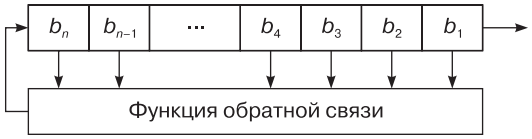

Рис. 3.8. Регистр сдвига с обратной связью

Регистры сдвига очень быстро нашли применение в потоковых шифрах, так как они легко реализовывались с помощью цифровой аппаратуры. В 1965 году Эрнст Селмер (*Ernst Selmer*), главный криптограф норвежского правительства, разработал теорию последовательности регистров сдвига. Соломон Голомб (*Solomon Golomb*), математик NSA, написал книгу, излагающие некоторые свои результаты и результаты Селмера. Простейшим видом регистра сдвига с обратной связью является *регистр сдвига с линейной обратной связью* (*linear feedback shift register*, *далее LFSR или РгСсЛОС*). Обратная связь таких регистров

представляет собой просто XOR (сложение по модулю два) некоторых битов регистра, перечень этих битов называется отводной последовательностью (*tap sequence*). Иногда такой регистр называется конфигурацией Фиббоначи. Из-за простоты последовательности обратной связи для анализа РгССЛОС можно использовать довольно развитую математическую теорию. Проанализировав получаемые выходные последовательности, можно убедиться в том, что эти последовательности достаточно случайны, чтобы быть безопасными. РгССЛОС чаще других сдвиговых регистров используются в криптографии.

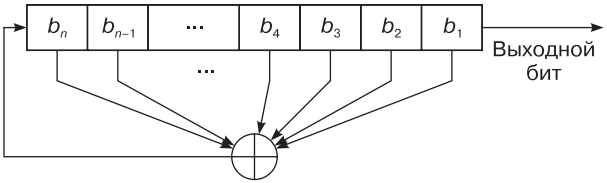

Рис. 3.9. РгССЛОС Фиббоначи

В общем случае n-битовый РгССЛОС может находиться в одном из $N = 2^n - 1$ внутренних состояний. Это означает, что теоретически такой регистр может генерировать псевдослучайную последовательность с периодом $T = 2^n - 1$ битов. (Число внутренних состояний и период равны $N = T_{\max} = 2^n - 1$, потому что заполнение РгССЛОС нулями приведет к тому, что сдвиговый регистр будет выдавать бесконечную последовательность нулей, что абсолютно бесполезно.) Только при определенных отводных последовательностях РгССЛОС циклически пройдет через все $2^n - 1$ внутренних состояний, такие РгССЛОС являются ***РгССЛОС с максимальным периодом***. Получившийся результат называется ***М-последовательностью***.

Пример. На рисунке 3.10 показан четырехбитовый РгССЛОС с отводом от первого и четвертого битов. Если его проинициализировать значением 1111, то до повторения регистр будет принимать следующие внутренние состояния (табл. 3.4).

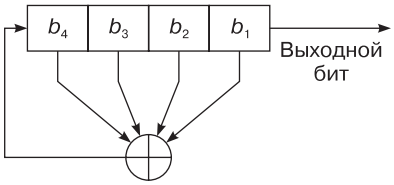

Рис. 3.10. Четырехбитовый РгССЛОС с отводом от первого и четвертого битов

Таблица 3.4

Номер такта сдвига (внутреннего состояния)	Состояние регистров				Выходной бит
	T_1	T_2	T_3	T_4	
Инициальное значение	**1**	**1**	**1**	**1**	—
1	0	1	1	1	1
2	1	0	1	1	1
3	0	1	0	1	1
4	1	0	1	0	0
5	1	1	0	1	1
6	0	1	1	0	0
7	0	0	1	1	1
8	1	0	0	1	1
9	0	1	0	0	0
10	0	0	1	0	0
11	0	0	0	1	1
12	1	0	0	0	0
13	1	1	0	0	0
14	1	1	1	0	0
15 (возврат в инициальное состояние)	**1**	**1**	**1**	**1**	**1**
16 (повтор состояний)	**0**	**1**	**1**	**1**	**1**

Выходной последовательностью будет строка младших значащих битов: 111101011001000 с периодом $T = 15$, общее число возможных внутренних состояний (кроме нулевого) $N = 2^4 - 1 = 16 - 1 = 15 = T_{\max}$, следовательно, выходная последовательность — M-последовательность.

Для того чтобы конкретный РгССЛОС имел максимальный период, многочлен, образованный из отводной последовательности и константы 1, должен быть примитивным по модулю 2. Многочлен представляется в виде суммы степеней, например многочлен степени n представляется так:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0 x^0 = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0,$$

где $a_i = \{0,1\}$ для $i = 1 \ldots n$, а x^i — указывает разряд.

Степень многочлена является длиной сдвигового регистра. Примитивный многочлен степени n — это неприводимый многочлен, который является делителем $x^{2n-1} + 1$, но не является делителем $x^d + 1$ для всех d, являющихся делителями $2^n - 1$.

В общем случае не существует простого способа генерировать примитивные многочлены данной степени по модулю 2. Проще всего вы-

бирать многочлен случайным образом и проверять, не является ли он примитивным. Это нелегко и чем-то похоже на проверку, не является ли простым случайно выбранное число — но многие математические пакеты программ умеют решать такую задачу.

Некоторые, но, конечно же не все, многочлены различных степеней, примитивные по модулю 2, приведены далее. Например, запись (32, 7, 5, 3, 2, 1, 0) означает, что следующий многочлен примитивен по модулю 2: $x^{32} + x^7 + x^5 + x^3 + x^2 + x + 1$.

Это можно легко обобщить для РгССЛОС с максимальным периодом. Первым числом является длина РгССЛОС. Последнее число всегда равно 0, и его можно опустить. Все числа, за исключением 0, задают отводную последовательность, отсчитываемую от левого края сдвигового регистра. Другими словами, члены многочлена с меньшей степенью соответствуют позициям ближе к правому краю регистра.

Продолжая пример, запись (32, 7, 5, 3, 2, 1, 0) означает, что для взятого 32-битового сдвигового регистра новый бит генерируется с помощью XOR тридцать второго, седьмого, пятого, третьего, второго и первого битов, получающийся РгССЛОС будет иметь максимальную длину, циклически проходя до повторения через $2^{32} - 1$ значений.

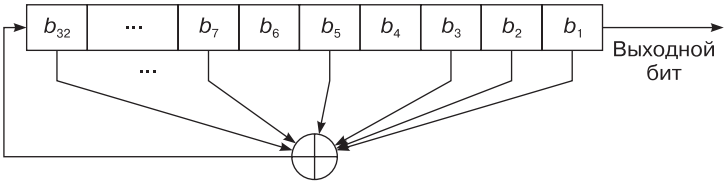

Рис. 3.11. 32-битовый РгССЛОС с максимальной длиной

Рассмотрим программный код РгССЛОС, у которого отводная последовательность характеризуется многочленом (32, 7, 5, 3, 2, 1, 0). На языке *C* это выглядит следующим образом:

```
int LFSR ()
{
     static unsigned long ShiftRegister = 1;
     /* Все, кроме 0. */
     ShiftRegister = ((((ShiftRegister >> 31)
     ^(ShiftRegister >> 6)
     ^(ShiftRegister >> 4)
     ^(ShiftRegister >> 2)
     ^(ShiftRegister >> 1)
```

```
      ^ShiftRegister))
      &0x00000001)
      <<31)
      |(ShiftRegister >> 1);
      return ShiftRegister & 0x00000001;
}
```

Если сдвиговый регистр длиннее компьютерного слова, код усложняется, но не намного. В приложении В приведена таблица некоторых примитивных многочленов по модулю 2, будем использовать ее в дальнейшем для выявления некоторых свойств этих многочленов, а также в программной реализации для задания отводной последовательности.

Следует обратить внимание, что у всех элементов таблицы нечетное число коэффициентов. Такая длинная таблица приведена для дальнейшей работы с РгССЛОС, так как РгССЛОС часто используются для работы с потоковыми шифрами и в генераторах псевдослучайных чисел. В нашем случае можно использовать многочлены со старшей степенью не более семи.

Если $p(x)$ примитивен, то примитивен и $x^n p(1/x)$, поэтому каждый элемент таблицы на самом деле определяет два примитивных многочлена. Например, если (a, b, 0) примитивен, то примитивен и (a, a-b, 0). Если примитивен (a, b, c, d, 0), то примитивен и (a, a-d, a-c, a-b, 0). Математически:

если примитивен $x^a + x^b + 1$, *то примитивен и* $x^a + x^{a-b} + 1$,
если примитивен $x^a + x^b + x^c + x^d + 1$,
то примитивен и $x^a + x^{a-d} + x^{a-c} + x^{a-b} + 1$.

Наиболее просто программно реализуются примитивные трехчлены, так как для генерации нового бита нужно выполнять XOR только двух битов сдвигового регистра (нулевой член не учитывается, т.е. $x^0 = 1$, см. пример выше). Действительно, все многочлены обратной связи, приведенные в таблице, являются разреженными, т.е., у них немного коэффициентов. Разреженность всегда представляет собой источник слабости, которой иногда достаточно для вскрытия алгоритма. Для криптографических алгоритмов гораздо лучше использовать плотные примитивные многочлены, у которых много коэффициентов. Применяя плотные многочлены, особенно в качестве части ключа, можно использовать значительно более короткие РгССЛОС.

Генерировать плотные примитивные многочлены по модулю 2 нелегко. В общем случае для генерации примитивных многочленов степени k нужно знать разложение на множители числа $2^k - 1$.

Сами по себе РгССЛОС являются хорошими генераторами псевдослучайных последовательностей, но они обладают некоторыми нежелательными неслучайными (детерминированными) свойствами. Последовательные биты линейны, что делает их бесполезными для шифрования. Для РгССЛОС длины n внутреннее состояние представляет собой предыдущие n выходных битов генератора. Даже если схема обратной связи хранится в секрете, она может быть определена по $2n$ выходным битам генератора с помощью высокоэффективного алгоритма *Berlekamp — Massey*.

Кроме того, большие случайные числа, генерируемые с использованием идущих подряд битов этой последовательности, сильно коррелированны и для некоторых типов приложений вовсе не являются случайными. Несмотря на это, РгССЛОС часто используются в качестве составных частей систем и алгоритмов шифрования.

О потоковых шифрах на базе РгССЛОС. Основной подход при проектировании генератора потока ключей на базе РгССЛОС прост. Сначала берется один или несколько РгССЛОС обычно с различными длинами и различными многочленами обратной связи. Если длины взаимно просты, а все многочлены обратной связи примитивны, то у образованного генератора будет максимальная длина. Ключ является начальным состоянием регистров РгССЛОС. Каждый раз для получения нового бита, достаточно сдвинуть на бит регистры РгССЛОС (это иногда называют тактированием (*clocking*)). Бит выхода представляет собой функцию, желательно нелинейную, некоторых битов регистров РгССЛОС. Эта функция называется комбинирующей функцией, а генератор в целом — комбинационным генератором. Если бит выхода является функцией единственного РгССЛОС, то генератор называется фильтрующим генератором. Большая часть теории подобного рода устройств разработана Селмером (*Selmer*) и Нилом Цирлером (*Neal Zierler*). Можно ввести ряд усложнений. В некоторых генераторах для различных РгССЛОС используется различная тактовая частота, иногда частота одного генератора зависит от выхода другого. Все это электронные версии идей шифровальных машин, появившихся до Второй мировой войны, которые называются генераторами с управлением тактовой частотой (*clock-controlled generators*). Управление тактовой частотой может быть с прямой связью, когда выход одного РгССЛОС управляет тактовой частотой другого РгССЛОС, или с обратной связью, когда выход одного РгССЛОС управляет его собственной тактовой частотой. Хотя все эти генераторы чувствительны, по крайней мере теоретически, к вскрытиям вложением и вероятной корреляцией, многие из них безопасны до сих пор.

Ян Касселлс (*Ian Cassells*), ранее возглавлявший кафедру чистой математики в Кембридже и работавший криптоаналитиком в Блетчли Парк (*Bletchly Park*), сказал, что «криптография — это смесь математики и путаницы, и без путаницы математика может быть использована против вас». Он имел в виду, что в потоковых шифрах для обеспечения максимальной длины и других свойств необходимы определенные математические структуры, такие как РгСсЛОС, но, чтобы помешать кому-либо получить содержание регистра и вскрыть алгоритм, необходимо внести некоторый сложный нелинейный беспорядок. Этот совет справедлив и для блочных алгоритмов.

Большинство реальных потоковых шифров основаны на РгСсЛОС. Даже в первые дни электроники построить их было несложно. Сдвиговый регистр не представляет собой ничего большего, чем массив битов, а последовательность обратной связи — набор вентилей XOR. Даже при использовании современных интегральных схем потоковый шифр на базе РгСсЛОС обеспечивает немалую безопасность с помощью нескольких логических вентилей. Проблема РгСсЛОС состоит в том, что их программная реализация очень неэффективна. Вам приходится избегать разреженных многочленов обратной связи — они облегчают корреляционные вскрытия — а плотные многочлены обратной связи неэффективны.

Эта отрасль криптографии быстро развивается и находится под зорким государственным контролем со стороны NSA. Большинство разработок засекречены — множество используемых сегодня военных систем шифрования основаны на РгСсЛОС. Действительно, у большинства компьютеров *Cray* (*Cray* 1, *Cray X-MP*, *Cray Y-MP*) есть весьма любопытная инструкция, обычно называемая как «счетчик совокупности» (*population count*). Она подсчитывает количество единиц в регистре и может быть использована как для эффективного вычисления расстояния Хэмминга между двумя двоичными словами так и для реализации векторизированной версии РгСсЛОС. Эта инструкция считается канонической инструкцией NSA, обязательно фигурирующей почти во всех контрактах, касающихся компьютеров.

С другой стороны, было взломано удивительно большое число казавшихся сложными генераторов на базе сдвиговых регистров.

О линейной сложности генерируемой последовательности псевдослучайных чисел РгСсЛОС. Анализировать потоковые шифры часто проще, чем блочные. Например, важным параметром, используемым для анализа генераторов на базе РгСсЛОС, является линейная сложность (*linear complexity*), или линейный интервал. Она определяется как длина *n* самого короткого РгСсЛОС, который может имитировать выход

генератора. Любая последовательность, сгенерированная конечным автоматом над конечным полем, имеет конечную линейную сложность. Линейная сложность важна, потому что с помощью простого алгоритма, называемого алгоритмом *Berlekamp-Massey*, можно определить этот РгСсЛОС, проверив только 2*n* битов потока ключей. Воссоздавая нужный РгСсЛОС, вы взламываете потоковый шифр.

Эта идею можно расширить с полей на кольца и на случаи, когда выходная последовательность рассматривается как числа в поле нечетной характеристики. Дальнейшее расширение приводит к вводу понятия профиля линейной сложности, который определяет линейную сложность последовательности по мере ее удлинения. Существуют также понятия сферической и квадратичной сложности. В любом случае следует помнить, что высокая линейная сложность не обязательно гарантирует безопасность генератора, но низкая линейная сложность указывает на недостаточную безопасность генератора.

О корреляционной независимости генерируемой последовательности псевдослучайных чисел РгСсЛОС. Криптографы пытаются получить высокую линейную сложность, нелинейно объединяя результаты некоторых выходных последовательностей. При этом опасность состоит в том, что одна или несколько внутренних выходных последовательностей — часто просто выходы отдельных РгСсЛОС — могут быть связаны общим ключевым потоком и вскрыты при помощи линейной алгебры. Часто такое вскрытие называют корреляционным вскрытием, или вскрытием разделяй-и-властвуй. Томас Сигенталер (*Thomas Siegenthaler*) показал, что можно точно определить корреляционную независимость и что существует компромисс между корреляционной независимостью и линейной сложностью.

Основной идеей корреляционного вскрытия является обнаружение некоторой корреляции между выходом генератора и выходом одной из его составных частей. Тогда, наблюдая выходную последовательность, можно получить информацию об этом промежуточном выходе. Используя эту информацию и другие корреляции, можно собирать данные о других промежуточных выходах до тех пор, пока генератор не будет взломан.

Против многих генераторов потоков ключей на базе РгСсЛОС успешно использовались корреляционные вскрытия и их вариации, такие как быстрые корреляционные вскрытия, предлагающие компромисс между вычислительной сложностью и эффективностью.

О других способах вскрытия генерируемой последовательности псевдослучайных чисел РгСсЛОС. Существуют и другие способы вскрытия генераторов потоков ключей. Тест на линейную корректность (*linear*

consistency) — это попытка найти некоторое подмножество ключа шифрования с помощью матричной техники. Существует и вскрытие корректности «встречей посередине» (*meet-in-the-middle consistency attack*). Алгоритм линейного синдрома (*linear syndrome algorithm*) основан на возможности записать фрагмент выходной последовательности в виде линейного уравнения. Существует вскрытие лучшим аффинным приближением (*best affine approximation attack*) и вскрытие выведенным предложением (*derived sequence attack*). К потоковым шифрам можно применить также методы дифференциального и линейного криптоанализа.

На рисунке 3.12 приведена обобщенная схема алгоритма РгССЛОС, рассматриваемого в лабораторной работе.

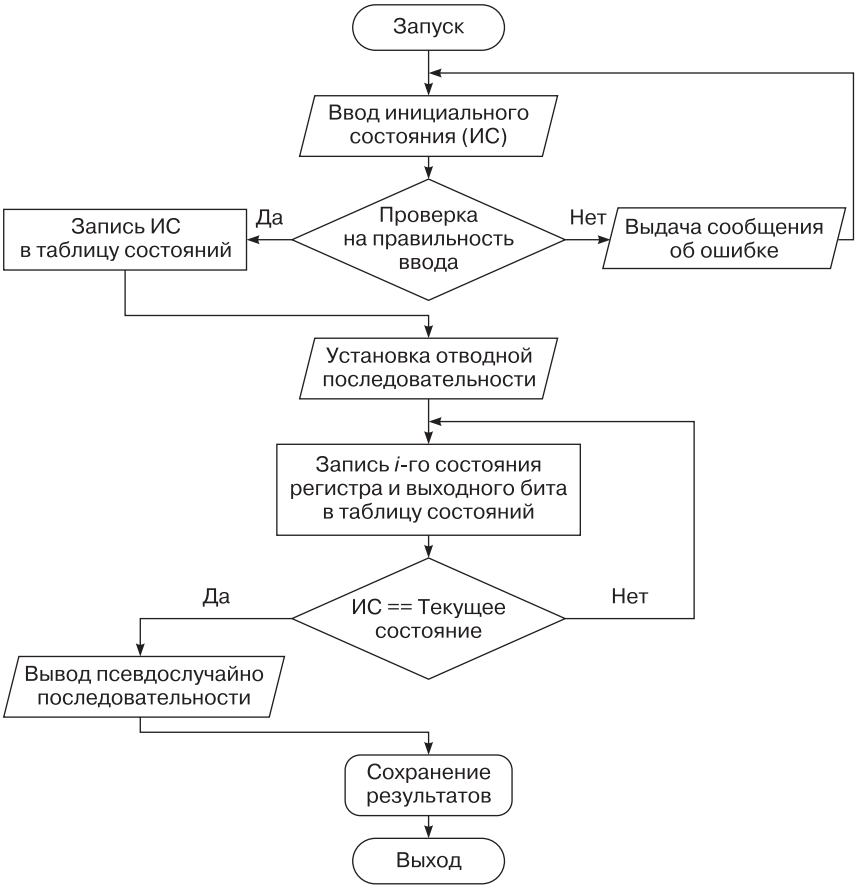

Рис. 3.12. Обобщенная схема алгоритма РгССЛОС

Описание программного интерфейса. Ниже на рис. 3.13 представлено главное окно программы.

Рис. 3.13. Главное окно программы

В меню есть следующие функции:

- *Файл->Сохранить отчет*

Эта функция осуществляет создание файла отчета, куда записывается инициальное состояние РгСсЛОС, отводная последовательность, период полученной псевдослучайной последовательности бит, сама последовательность и таблица состояний. Если файл успешно был сохранен, то выдается сообщение об успешном сохранении (рис. 3.14). Рекомендуемое расширение файла отчета *.txt.

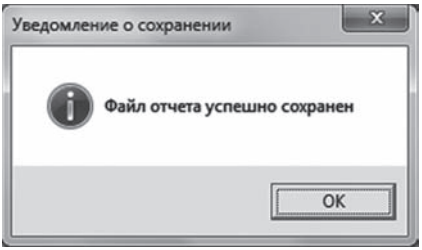

Рис. 3.14. Уведомление о сохранении файла

- *Файл->Выход*

Эта функция обеспечивает закрытие приложения.

- *Задать отводную последовательность*

Эта функция формирует отводную последовательность, считывая значения из клеток, которые пользователь отметил галочкой на экранной

форме. После чего она уведомляет пользователя о создании отводной последовательности (рис. 3.15). Отводная последовательность определяет, от каких триггеров регистра сдвига будут идти обратные связи XOR для формирования бита смещения. По умолчанию обратная связь от первого триггера стоит всегда, при отсутствии других связей будет осуществляться сдвиг влево с перестановкой младшего бита на позицию старшего.

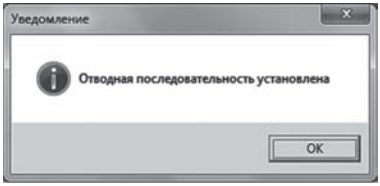

Рис. 3.15. Уведомление об установке отводной последовательности

■ *Установить инициальное значение*

Эта функция считывает введенное пользователем инициальное значение регистра из окна Edit и осуществляет проверку введенного значения согласно заданным условиям: введенная строка непустая (рис. 3.16), введенная строка имеет длину равную восьми (8 бит = 1 байт, рис. 3.17), введенная строка содержит только нули и/или единицы (рис. 3.18), введенная строка ненулевая (рис. 3.19). В противном случае выдаются сообщения об ошибке, пользователь должен их исправить и снова нажать на кнопку. В случае успешной проверки инициальное значение будет записано в таблицу состояний в строке «Инициальное» и выдано уведомление (рис. 3.20).

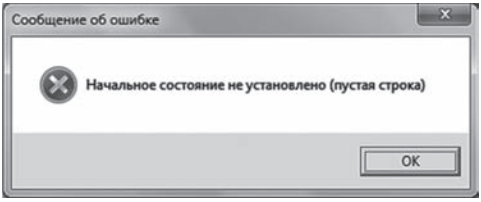

Рис. 3.16. Сообщение о том, что начальное состояние не установлено

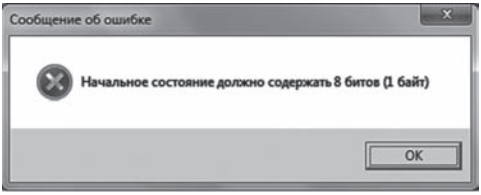

Рис. 3.17. Сообщение о том, что длина начального состояния недопустима

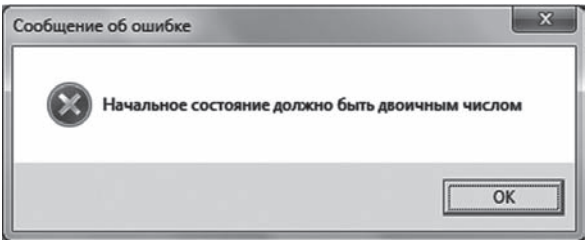

Рис. 3.18. Сообщение о том, что система счисления для указания начального состояния выбрана неверно

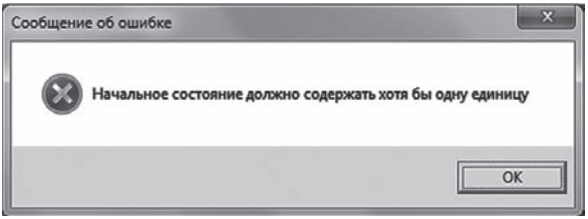

Рис. 3.19. Сообщение о том, что строка начального состояния не содержит единиц

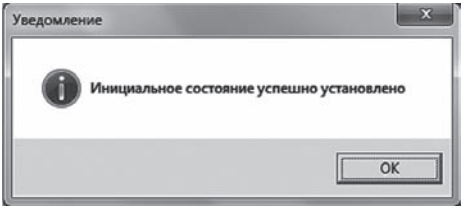

Рис. 3.20. Сообщение об успешном установлении инициального состояния

- *Произвести сдвиг регистра*

Эта функция эмулирует работу регистра сдвига. Последовательно производя 256 сдвигов, каждый сдвиг формирует выходной бит псевдослучайной последовательности и новое состояние регистра. Как только находится состояние регистра равное инициальному, вычисляется период и выводит в поле Memo полученную псевдослучайную последовательность.

- *Помощь->О программе*

Эта функция выводит на экран краткое описание программы и инструкцию (рис. 3.21).

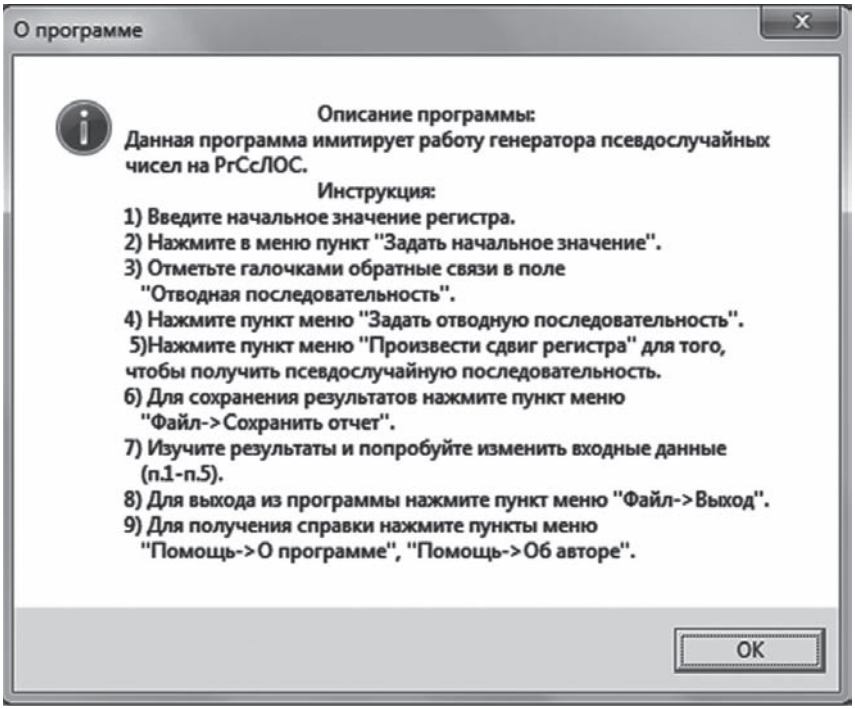

Рис. 3.21. О программе

■ *Помощь->Об авторе*

Эта функция выводит на экран информацию об авторе программы (рис. 3.22).

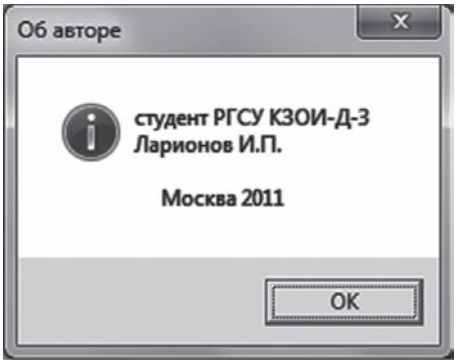

Рис. 3.22. Информация об авторе

Задание

1. Запустите программу 8*bit-LFSR.exe*. Ознакомьтесь со справкой по программе для выполнения работы (см. рис. 3.21).

2. Введите в поле «**Инициальное значение**» любое восьмиразрядное двоичное число.

3. Установите отводную последовательность (по умолчанию обязательно один триггер должен входить в отводную последовательность и должна присутствовать хотя бы одна обратная связь, иначе регистр будет неисправен).

4. Нажмите пункт меню «**Произвести сдвиг регистра**».

5. В отчете по лабораторной работе отразите результаты, полученные в п. 4 (**Файл — Сохранить отчет**):

- введенное в п. 2 инициальное значение;
- многочлен, характеризующий отводную последовательность;
- период полученной псевдослучайной последовательности;
- полученную псевдослучайную последовательность.

Ответьте на вопрос: является ли полученная в п. 5 последовательность максимальной? Почему?

6. Выполните п. 3—5 с теми же параметрами, но в отводной последовательности укажите только один триггер.

Ответьте на вопрос: какое значение поступает на вход восьмого триггера при каждом его сдвиге и как оно вычисляется в случае если присутствует одна обратная связь? Если присутствует две или более (до восьми) обратных связей?

7. Выполните п. 3—5 с теми же параметрами, но в качестве инициальной последовательности укажите «11111111».

Ответьте на вопрос: влияет ли инициальное значение на период псевдослучайной последовательности? Влияет ли инициальное значение на псевдослучайную последовательность?

- Измените отводную последовательность (должно быть минимум две связи), а инициальное значение оставьте тем же. Произведите сдвиг регистра. Что изменилось по сравнению с предыдущим пунктом?
- Измените отводную последовательность, установив только одну обратную связь T1. Произведите сдвиг регистра. Что изменилось по сравнению с предыдущим пунктом? Почему период последовательности отличается от аналогичных в предыдущих пунктах (п. 6—7)?

8. Выполните п. 3—5 с теми же параметрами, но в качестве инициальной последовательности укажите «00000000».

Ответьте на вопрос: почему это значение недопустимо при работе РгССЛОС?

9. Выполните п. 3—5, но в отводной последовательности отметьте галочками 1, 5, 6 и 7 триггеры или 1, 4, 6, 8 триггеры.

- Измените несколько раз инициальное значение и произведите сдвиг регистра.

Ответьте на вопросы: Каков период псевдослучайной последовательности? Почему он принимает это значение? Какой многочлен по модулю два задает такую последовательность? Является ли он приводимым? Что меняется при изменении инициального значения?

10. Предоставьте электронный отчет преподавателю. Отчет должен содержать ответы на вопросы п. 5—10, сведения для п. 5—10, описанные в п. 5, 1, и скриншот главной формы и формы «**Справка — О программе**» (п. 1).

11. Включите в отчет о лабораторной работе ответы на вопросы, выбранные в соответствии с номером варианта из табл. 3.5.

Таблица 3.5

Номер варианта	Контрольные вопросы
1, 5, 7, 3, 9, 18, 28	Что такое М-последовательность? Каковы ее свойства? Какая отводная последовательность позволяет ее получить? Опишите процесс работы четырехбитового РгССЛОС: откуда берется выходной бит и как формируется псевдослучайная последовательность, как происходит сдвиг регистра, как меняется его состояние, как образуется бит функции обратной связи (приведите таблицу истинности для операции XOR от 1, 2, 3, 4 переменных)
2, 4, 6, 8, 20, 22, 24, 26, 30	Что определяет свойство периодичности РгССЛОС? Отчего зависит период РгССЛОС? Какие многочлены являются неприводимыми по модулю 2? Где и для чего их можно применять на практике?
11, 13, 15, 10, 17, 19, 27	Что входит в понятие «Линейная сложность бинарной последовательности»? Как ее можно использовать для оценки псевдослучайно бинарной последовательности? Что определяет свойство периодичности РгССЛОС? Отчего зависит период РгССЛОС?
12, 14, 16, 21, 23, 25, 29	Что такое отводная последовательность? Для чего она нужна в РгССЛОС и на какие его параметры влияет? Что такое инициальное состояние РгССЛОС? Какие есть ограничения на значение инициального состояния? Почему? Что такое ГСПЧ? На чем они могут быть основаны? Какие у них преимущества и недостатки?

Список литературы

1. *Бабаш А.В.*, *Шанкин Г.П.* Криптография / под ред. В.П. Шерстюка, Э.А. Применко. М. : СОЛОН-Р, 2002.

2. *Башлы П.Н.*, *Бабаш А.В.*, *Баранова Е.К.* Информационная безопасность : учеб.-практ. пособие. М. : Изд. центр ЕАОИ, 2010.

3. *Шнайер Б.* Прикладная криптография. 2-е изд. Протоколы, алгоритмы и исходные тексты на языке С. М. : Триумф, 2002.