МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Кафедра телекоммуникаций и информационных технологий

ПУПКО
Максим Сергеевич

РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОЛУЧЕНИЯ СЛУЧАЙНОЙ ЧИСЛОВОЙ ПОСЛЕДОВАТЕЛЬНОСТИ С ФИЗИЧЕСКОГО ГЕНЕРАТОРА СЧП

Аннотация к дипломной работе

Научный руководитель – старший преподаватель А.Л. Труханович

Минск, 2023

РЕФЕРАТ

Дипломная работа: 61 с., 24 рис.,2 табл., 32 источника

ГЕНЕРАТОРЫ СЧП, КРИТЕРИИ КАЧЕСТВА, КОМПЛЕКСЫ ГЕНЕРАЦИИ СЧП, NIST, FIST

Объект исследования – генераторы СЧП.

Цель работы – создание собственного решения для получения и проверки случайных числовых последовательностей, сгенерированных комплексами генерации СЧП.

В ходе проведения работы были исследованы способы генерации случайных последовательностей с использованием физических явлений, обозрены критерии для оценки качества СЧП. Также были рассмотрены пакеты тестов FIPS, NIST и Diehard с объяснениями проверяемых свойств.

В качестве наиболее подходящего инструмента для создания приложения был выбран фреймворк QT. Была описана структура проекта и показаны основные свойства модулей, использованных в разработке собственного решения.

В результате выполнения работы было разработано приложение для персональных компьютеров на языке программирования C++, предоставляющее возможности выгрузки и тестирования СЧП в соответствии с методологиями.

РЭФЕРАТ

Дыпломная праца: 61 с., 24 мал.,2 табл., 32 крыніцы

ГЕНЕРАТАРЫ ВЛП, КРЫТЭРЫЯ ЯКАСЦІ, КОМПЛЕКСЫ ГЕНЕРАЦЫІ ВЛП, NIST, FIST

Аб'ект даследавання – генератары ВЛП.

Мэта працы – стварэнне ўласнага рашэння для атрымання і праверкі выпадковых лікавых паслядоўнасцяў, згенераваных комплексамі генерацыі ВЛП.

У ходзе правядзення работы былі даследаваны спосабы генерацыі выпадковых паслядоўнасцей з выкарыстаннем фізічных з'яў, агледжаны крытэры для ацэнкі якасці ВЛП. Таксама былі разгледжаны пакеты тэстаў FIPS , NIST і Diehard з тлумачэннямі правяраных уласцівасцяў.

У якасці найбольш прыдатнай прылады для стварэння прыкладання быў абраны фрэймворк QT. Была апісана структура праекта і паказаны асноўныя ўласцівасці модуляў, скарыстаных у распрацоўцы ўласнага рашэння.

У выніку выканання працы было распрацавана прыкладанне для персанальных кампутараў на мове праграмавання C ++, якое прадстаўляе магчымасці выгрузкі і тэставанні ВЛП у адпаведнасці з метадалогіямі.

ABSTRACT

Thesis: 61 p., 24 pics, 2 tab., 32 sources

RNS GENERATORS, QUALITY CRITERIA, RNS GENERATION COMPLEXES, NIST , FIST

The object of the study is the generators of the RNS.

The purpose of the work is to create our own solution for obtaining and checking random numerical sequences generated by the complexes for generating RNS.

In the course of the work, methods for generating random sequences using physical phenomena were investigated, and criteria for assessing the quality of the RNS were reviewed. The FIPS, NIST, and Diehard test suites were also reviewed, with explanations of the properties being tested.

The framework was chosen as the most suitable tool for creating an application. Qt. The structure of the project was described and the main properties of the modules used in the development of their own solution were shown.

As a result of the work, an application for personal computers in the C ++ programming language was developed, which provides the ability to download and test the RNS in accordance with the methodologies.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ТЕРМИНОВ

ГСП – генератор случайных последовательностей
ГСЧП – генератор случайных числовых последовательностей
СЧП – случайная числовая последовательность
ППО – прикладное программное обеспечение
FIPS – федеральные стандарты обработки информации
NIST – национальный институт стандартов и технологий
ФГСЧП – физический генератор случайной числовой последовательности
СП – случайная последовательность

ВВЕДЕНИЕ

С развитием информационных технологий и возникновением все новых направлений применений стремительно развивающихся технологий, появилась необходимость получать истинно случайные числовые последовательности, которые можно применять для широкого диапазона задач, начиная от задач обеспечения криптобезопасности и заканчивая технологиями моделирования и тестирования.

Со временем соответственно расширенному кругу задач, стало необходимо находить более качественные «случайности», среди которых было бы невозможно или почти невозможно проследить закономерностей в генерации, или периодов генерации чисел.

Задачами для таких качественных случайностей, становятся криптографические методы, тестирование алгоритмов, математическое и имитационное моделирование и математическая статистика. Каждая из этих задач может предъявить свои, особые требования к случайным числам, но все они в той или иной мере опираются на то, что эти числа будут по-настоящему случайны и для генерации такого рода СЧП желательно использовать генераторы, основанные на некоторых физических процессах, таких как шумы.

Из вытекающих из этих задач требований возникает неотложная потребность в разработке комплексов для СЧП и программного обеспечения, которое способно провести тестирование генератора, собрать сгенерированные СЧП в файл и осуществить их проверку. Это приводит к концепции разработки приложения, позволяющего проводить тестирование полученных последовательностей в соответствии с методологиями FIPS и NIST, а также проводить анализ статистической равномерности выборки для более точного определения пригодности созданной выборки для решения поставленной задачи.

Таким образом, для решения указанных задач предлагается создание специализированного приложения, которое включает в себя функциональность для тестирования генерируемых последовательностей в соответствии с признанными методологиями FIPS и NIST. Данное приложение также должно обеспечивать возможность сбора сгенерированных СЧП в файл и в последующем их проверять с использованием различных статистических метрик, включая анализ равномерности выборки.

Целью данного приложения является создание среды, в которой можно проводить тщательное и всестороннее тестирование полученных СЧП, анализировать их свойства и статистическую пригодность для решения конкретных задач. Это позволяет обеспечить более надежную оценку и выбор оптимальных генераторов и последовательностей, соответствующих требуемым

стандартам и критериям, и обеспечить эффективное решение поставленных задач.

ГЛАВА 1
ФИЗИЧЕСКИЕ ИСТОЧНИКИ СЧП

1.1 Виды источников генерации случайных последовательностей

Аппаратный генератор случайных чисел – это устройство, которое генерирует последовательность случайных чисел на основе измеряемых, хаотически изменяющихся параметров протекающего физического процесса. При аппаратном способе генерации случайные числа являются прямым или побочным продуктом измерений некоторой физической величины, служащей надежным источником энтропии. Обычно это процессы, протекающие в неживой природе. Теоретически такие процессы абсолютно непредсказуемы, однако на практике полученные таким образом случайные числа приходится подвергать проверке с помощью специальных статистических тестов.

В аппаратных генераторах случайных последовательностей для генерации случайных чисел используются такие источники энтропии, как:

1) тепловой и электрический шум;
2) квантовые процессы;
3) радиоактивный распад;
4) космическое излучение;
5) различные механические, оптические и фотоэлектрические явления.

Конкретными примерами источников энтропии, подходящих для генерации истинно случайных последовательностей, могут быть:

1) временные интервалы между выбросами частиц при радиоактивном распаде;
2) тепловой шум полупроводникового диода или резистора;
3) состояния спутанности фотонов;
4) квантовый шум лазеров;
5) дробовой шум;
6) нестабильности частоты осцилляторов [1].

Далее будет рассказано о некоторых видах источников энтропии.

1.1.1 Физические источники шума

Под шумом понимают флуктуации одного или нескольких параметров системы, отличающиеся сложной временной и спектральной структурой. Наиболее важной энергетической характеристикой шума является спектральная плотность шума.

Примеров «шумящих» систем существует бесчисленное множество. К ним можно отнести абсолютное большинство природных явлений и систем – как наиболее сложных для аналитического описания. К примеру, акустический шум может вызывать дерево, листья которого беспорядочно колеблются под воздействием потока воздуха.

В качества источника шума для ФГСЧП будем рассматривать только электронные системы. Характеристиками такого шума достаточно легко управлять, легко преобразовывать флуктуирующий параметр в цифровой сигнал для дальнейшей его обработки.

Рассмотрение электронных систем позволяет выделить следующие основные источники шума [5]:

Тепловой шум. Данному виду шума подвержены любые электронные системы. Причиной его возникновения является хаотическое движение носителей заряда. Вследствие броуновского движения носителей спонтанные флуктуации напряжения на сопротивлении имеют белый спектр до частот, определяемых постоянством сопротивления резистора.

Дробовый шум. Является результатом корпускулярной природы электричества и случайной эмиссии носителей заряда через потенциальный барьер. Поэтому этот тип шума всегда присутствует в таких устройствах как диоды. Спектр флуктуаций тока является белым (до частот, много меньше обратной величины времени пролёта носителей заряда через активную область) и пропорционален элементарному заряду носителей q_e и среднему току $\bar{I}$. На низких частотах флуктуации RTS шума, генерационно-рекомбинационного шума и $1/f$ шума могут значительно превышать флуктуации дробового шума. При этом, чем меньше носителей заряда в системе, тем больше будут относительные флуктуации сопротивления и напряжения.

RTS шум (случайный телеграфный сигнал). Данный шум является индикатором активности одиночной ловушки в системе с малым числом носителей заряда. Шум наблюдается в приборах, которые имеют малые размеры и работают при высоких полях и плотностях тока. Шумовые импульсы тока имеют приблизительно фиксированную амплитуду, равную нескольким процентам от среднего уровня, с длительностью от 1 с до 10^{-5} с, при заданном режиме работы. RTS шум доминирует над $1/f$ шумом только в случае малого количества носителей (менее 10^5) [5].

Генерационно-рекомбинационный шум (ГР-шум). ГР-шум обусловлен флуктуациями числа носителей ΔN вследствие наличия ловушек заряда. Плотность распределения амплитуды шума имеет гауссову форму (4)

Полностью устранить наличие ловушек невозможно, однако уменьшить данный шум можно, изменив температурный диапазон работы прибора, тем самым, увеличив «расстояние» между уровнем Ферми и уровнем ловушки. ГР-шум может перекрываться 1/f шумом.

1/f шум. Данный шум представляет собой спонтанные флуктуации сопротивления. Причём плотность данных флуктуаций S_R обратно пропорциональна частоте f в широком диапазоне. Однозначной модели, описывающей данный шум, на сегодняшний день не существует. Однако наиболее распространённым подходом к объяснению 1/f шума является предположение, что в образцах, обнаруживающих этот шум, происходят разнообразные релаксационные процессы с широким спектром времён релаксации [5].

Микроплазменный шум. Данный вид шума наблюдается внутри p-n перехода полупроводниковых приборов в виде ступенчатого сигнала с амплитудой порядка 10^{-5} А. При этом математическая модель изменения тока во времени представляет собой непрерывный во времени Марковский процесс с двумя дискретными состояниями: «включено» и «выключено» лавинное умножение носителей зарядов [5]. В данном случае количество шумовых импульсов на интервале времени подчиняется распределению Пуассона.

При лавинном пробое полупроводниковых приборов с микроплазменными образованиями преобладают два механизма образования шума: флуктуации лавинного умножения носителей заряда и бистабильные флуктуации «включения» и «выключения» ударной ионизации.

1.1.2 Генераторы свободных колебаний

Кольцевые генераторы В том случае, когда выходной сигнал инвертора подается на его вход, цепь превращается в так называемый свободный осциллятор, или генератор свободных 20 колебаний. Частота его колебаний определяется внутренними запаздываниями и паразитными емкостями. Инвертирующий вентиль представляет собой инверсный усилитель высокой мощности. Особенность возникающих в такой схеме колебаний заключается в том, что они возникают в цепи с отрицательной обратной связью (сдвиг фазы 180°), в то время как обычно отрицательная обратная связь приводит к стабилизации. Причина заключается в том, что при анализе системы предполагается, что усиление является бесконечным.

Однако, поскольку в реальных условиях такое усиление недостижимо, цепь может перейти в некоторое промежуточное состояние, с нулевыми колебаниями или с колебаниями очень малой амплитуды. Для поддержания колебаний можно добавить некоторое реактивное сопротивление в петлю обратной связи, чтобы произвести сдвиг фазы, отличный от ±180°. Такая же функция может быть обеспечена с рассеянным реактивным сопротивлением. Из-за сложного механизма свободных колебаний их частота, как правило, весьма чувствительна к изменению напряжения питания и температуры, но эти изменения в сравнении с частотой колебаний происходят медленно.

С другой стороны, электрический шум, присутствующий на входе, прибавляется к сигналу, поданному назад от выхода, и после того, как он подвергается серьезному усилению, вызывает очень сильное случайное колебание (дрожание) частоты и фазы колебаний. В этом смысле генераторы случайных чисел, основанные на генераторах свободных колебаний, можно рассматривать как частный случай генераторов на основе шума. Поскольку шум каждого такого контура является индивидуальным, разумно предположить, что несколько генераторов даже на одной микросхеме имеют разные частоты и что их взаимные фазы случайны по времени. Если несколько таких генераторов расположены близко друг к другу (например, на одной микросхеме), они имеют тенденцию синхронизироваться через электромагнитное взаимодействие, чему способствует высокий коэффициент усиления свободных осцилляторов.

Этот эффект является основной проблемой, присущей генераторам свободного хода, и может отрицательно сказаться на надежности генератора. Другой важной проблемой генераторов на свободных осцилляторах является то, что выходная амплитуда генератора зависит от паразитных реактивностей и задержек в цепи. Отмечается, что шумовой сигнал диода достаточно велик вследствие эффекта лавинного нарастания заряда.

Названы такие методы преобразования шумового сигнала в цифровую форму, как:

1) Аналогово-цифровое преобразование.

2) Наблюдение последовательности импульсов с определением количества импульсов в единицу времени.

3) Наблюдение последовательности импульсов с определением интервала времени между последовательными импульсами [1].

1.1.3 Квантовые генераторы случайных последовательностей

Основу физического описания нашего мира составляют законы квантовой физики. С точки зрения поисков надежных источников энтропии важнейшей особенностью квантовых процессов является не вполне детерминированное

математическое описание движения частиц. Случайность внутренне присуща квантовым процессам, которые естественно рассматривать как хороший источник случайных чисел. Действительно, все рассмотренные выше системы, используемые как ГСП, подчиняются законам классической физики. Случайность их поведения вызвана лишь неопределенностью начальных состояний, что теоретически делает эти системы предсказуемыми. Поэтому только генераторы, работающие на квантовых принципах, можно строго обоснованно считать производящими истинно случайные значения. Квантовыми генераторами принято называть такие устройства, которые используют один действительно случайный квантовый эффект, который возможно воспроизводить многократно для получения случайных значений таким образом, чтобы перед каждым измерением система возвращалась к тем же начальным условиям.

Важно отметить, что при одинаковых начальных значениях и одном и том же способе измерения в соответствии с принципами квантовой физики будут получены различные результаты. Приведем основные сведения для объяснения принципов работы квантовых генераторов. Базовым понятием является кубит (qubit) [18-20], квантовый аналог обычного бита. Кубит определяют как квантовую систему, которая может находиться в двух состояниях: 0 и 1. В области элементарных частиц пример такой системы – электрон (имеет два возможных направления спина) или фотон (две возможные поляризации). Идея квантовых вычислений основана на таких эффектах квантовой механики, как квантовая суперпозиция и квантовый параллелизм, и использовании квантовых систем из двухуровневых квантовых элементов (кубитов). Система из L кубитов имеет 2L линейно независимых состояний и может выполнять параллельно 2L операций.

В рамках квантовой теории поля можно доказать, что при измерении состояния кубита его можно обнаружить в одном из двух указанных возможных состояний, а это означает, что из кубита можно извлечь один бит информации. В квантовых генераторах случайных чисел часто используются фотоны, т.к. их легко создавать, обнаруживать и манипулировать ими. В качестве иллюстрации можно привести схему ГСП, основанного на прохождении фотона, имеющего круговую поляризацию, через светоделительную пластину.

Такая система с равной вероятностью обеспечивает выход фотона как в вертикальном, так и в горизонтальном направлении. При неизменной системе и ее начальном состоянии результат каждый раз может быть иным, т.е. является случайным. Следует также добавить, что качество квантовых ГСП не снижается из-за существующих недостатков, таких, как неидеальная поляризация, многофазное излучение, время простоя детектора и т.д. Это объясняется тем, что вносимые ими отклонения могут измеряться и оцениваться независимо от

процесса генерации случайных значений. Основная проблема практической реализации генераторов случайных чисел, базирующихся на расщеплении пучка света, состоит в том, что для этого требуются два детектора. Их изначальные различия и последующий выход из строя из-за старения или температурных эффектов оказывают непосредственное влияние на качество сгенерированных случайных чисел.

Например, если эффективность детектирования фотонов датчиками не совсем одинакова, или если светоделительная пластина не идеально делит пучок 23 на две равные части, то при использовании такого генератора вероятность появления единиц не будет равна вероятности появления нулей. Эту проблему можно свести к минимуму с помощью схемы расщепления пучка, в которой используется только один фотонный детектор, но при этом коэффициент расщепления пучка должен быть точно отрегулирован механически. Другие проблемы возникают из-за времени простоя детектора и последующего импульса, что приводит к корреляциям, которые невозможно полностью устранить, но которые могут быть уменьшены ниже любого желаемого (допустимого) уровня путем применения целевой постобработки. Генераторы случайных чисел, основанные на светоделителях, являются примером так называемого «пространственного принципа», в котором значение, принимаемое случайным битом (0 или 1), определяется местом, в котором фотон заканчивается. Дополняющий его «временной принцип» использует информацию о времени выбросов случайных фотонов, например, в прямой квантовой (или атомной) релаксации, от хорошо насыщенных лазеров и т.д. [1].

1.2 Оценка «истинности» получаемой последовательности

ФГСЧП применяются в криптографических системах, для которых особую важность имеет решение задачи обеспечения гарантированного качества используемой СЧП.

Если для задач моделирования каких-либо процессов, явлений в подавляющем большинстве случаев вполне приемлемым является использование генераторов псевдослучайных чисел, то для криптографических систем такое решение обычно не годится. Основная причина состоит в том, что в генераторах псевдослучайных чисел все элементы последовательности всегда имеют функциональную зависимость между собой. В идеальном случае случайные числа должны быть равномерно распределены на всем диапазоне и независимы. Однако равномерное распределение характерно для идеального ГСЧП, который является математической конструкцией.

СЧП, применяемая в криптографических системах, должна иметь хорошие статистические свойства, близкие к свойствам независимых одинаково распределенных случайных величин Бернулли.

Для проверки качества СЧП используются наборы (пакеты) статистических тестов, специально подобранных для конкретного генератора СЧП, исходя из заданных (выбранных) критериев качества СЧП.

1.2.1 Критерии качества СЧП

Следует отметить, что критериев качества СЧП разработано очень много. При этом СЧП, прошедшая определённую группу тестов, может не пройти другую, что, однако, не всегда является основанием для того, чтобы не использовать данную СЧП в приложении. В зависимости от области применения одна и та же СЧП может являться как качественной, так и нет. К примеру, СЧП может являться равномерно распределённой, согласно соответствующим статистическим тестам, но ключи, полученные из данной СЧП, могут не являться достаточно стойкими для какого-либо криптографического алгоритма. Ниже кратко рассмотрим наиболее распространённые критерии качества СЧП.

Критерий «хи-квадрат» (χ^2-критерий). Данный критерий является наиболее распространенным и базовым методом, используемым в сочетании с другими критериями. В общем случае данный критерий выглядит следующим образом [10].

Предположим, что каждое наблюдение может принадлежать одной из k категорий. Проводим n независимых наблюдений (т. е. исход одного наблюдения не влияет на исход других наблюдений). Пусть p_s – вероятность того, что каждое наблюдение относится к категории s, и пусть Y_s – число наблюдений, которые действительно относятся к категории s. Образуем статистику:

$$V = \sum_{s=1}^{k} \frac{(Y_s - np_s)^2}{np_s}, \tag{1}$$

Возводя в квадрат $(Y_s - np_s)^2 = Y_s^2 - 2np_sY_s + n^2p_s^2$ и учитывая:

$$\begin{aligned} Y_1 + Y_2 + \ldots + Y_k &= n, \\ p_1 + p_2 + \ldots + p_k &= 1, \end{aligned} \tag{2}$$

получаем:

$$V = \frac{1}{n}\sum_{s=1}^{k}\left(\frac{Y_s^2}{p_s}\right) - n, \tag{3}$$

Теперь качество тестируемой СЧП будет определяться тем, насколько приемлемо значение V, что можно определить по специальным таблицам.

Данные таблицы предназначены для определения значений χ^2-распределения с ν степенями свободы, где $\nu = k - 1$ [10]. Если в таблице выбрать число x, стоящее в строке ν и в столбце p, то вероятность того, что значение V в (19) будет меньше либо равно x, приближённо равна p, если n достаточно велико.

Адаптированной для целей криптографии версией данного критерия можно назвать *тест четырехграмм*, представленный в стандарте FIPS 140-1 [11]. В данном тесте 20000-битная последовательность разбивается на 5000 последовательно идущих 4-битовых сегментов, что соответствует количеству независимых наблюдений $n = 5000$, количеству категорий $k = 16$ (количество всех значений, которые может принимать 4-битовая последовательность), вероятности появления категорий в равновероятном распределении считаем равными $p_s = 1/16$. Итого имеем:

$$V_{чг} = \frac{16}{5000}\sum_{s=0}^{15} Y_s^2 - 5000, \qquad (4)$$

По стандарту FIPS 140-1 пределы допустимых значений для $V_{чг}$, при которых тест считается пройденным, равны $1{,}03 < V_{чг} < 57{,}4$ [11].

Критерий Колмогорова-Смирнова. По сути, критерий Колмогорова-Смирнова является χ^2-критерием, экстраполированным на случай непрерывно распределённых случайных величин.

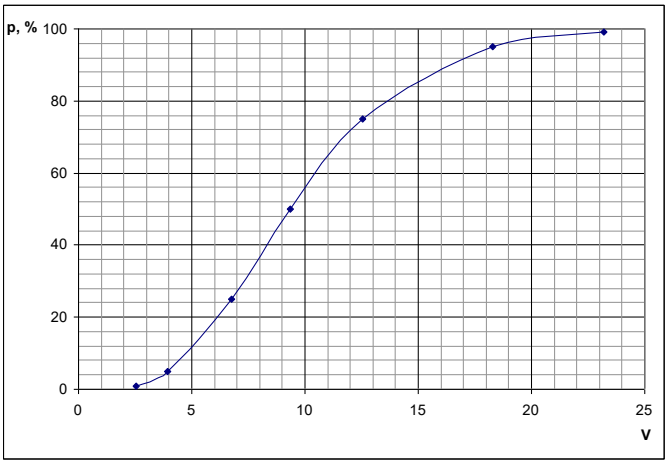

Рисунок 1 – Предельное распределение

Если взять значения таблицы для χ^2-критерия для числа степеней свободы 10, и представить их графически, то полученный результат представляет собой предельное распределение случайной величины *V*, которое показано на рисунке 1 (назовём это функцией $F(x)$). Тогда для любой случайной величины *X* можно выполнить *n* независимых наблюдений и построить эмпирическую функцию распределения $F_n(x)$:

$$F_n(x) = \frac{\textit{число } X_1, X_2, \ldots, X_n, \textit{таких, что они} \le x}{n}, \qquad (5)$$

Суть критерия Колмогорова-Смирнова (КС-критерий) сводится к определению разности между функциями $F(x)$ и $F_n(x)$. Чтобы построить КС-критерий, образуем следующие статистики:

$$K_n^+ = \sqrt{n} \max_{-\infty<x<+\infty} (F_n(x) - F(x)),$$
$$K_n^- = \sqrt{n} \max_{-\infty<x<+\infty} (F(x) - F_n(x)), \quad (6)$$

где K_n^+ – наибольшее из отклонений, когда F_n больше F;

K_n^- – наибольшее из отклонений, когда F_n меньше F.

Для того чтобы проверить, насколько значения K_n^+ и K_n^- удовлетворяют требованиям, можно воспользоваться специальной таблицей (аналогично χ^2-критерию).

Следует отметить безусловное преимущество данного метода над χ^2-критерием. Оно заключается в том, что КС-критерий можно использовать для любого количества наблюдений n, что никак не отразится на точности [10].

Описание следующих критериев ведётся применительно к последовательности:

$$\langle U_n \rangle = U_0, U_1, U_2, \ldots, \quad (7)$$

действительных чисел, которые предположительно независимы и равномерно распределены между 0 и 1. Либо к последовательности:

$$\langle Y_n \rangle = Y_0, Y_1, Y_2, \ldots$$
$$\text{где} \quad Y_n = \lfloor dU_n \rfloor, \quad (8)$$

целых чисел, которые предположительно независимы и одинаково распределены между 0 и $d-1$.

Критерий равномерности (*критерий частот*). Простейший тест на соответствие равномерному распределению. Тест может быть проведён используя критерий Колмогорова-Смирнова с функцией $F(x) = x$ для $0 \le x \le 1$.

Критерий серий. Суть данного критерия заключается в том, чтобы пары (тройки, четвёрки и т. д.) последовательных чисел были равномерно распределены независимым образом. Для пар чисел критерий выглядит следующим образом. Подсчитывается количество случаев, когда пара $(Y_{2j}, Y_{2j+1}) = (q, r)$, таких, что $0 \le j < n$. Такая операция осуществляется для каждой пары целых чисел (q, r), таких, что $0 \le q, r < d$. Затем применяется χ^2-критерий для $k = d^2$ категорий, где $1/d^2$ – вероятность отнесения пары чисел к каждой из категорий. Также не следует забывать, что для χ^2-критерия d должно быть значительно меньше n (по крайней мере $n \ge 5d^2$).

Критерий интервалов. Для некоторых действительных α и β, таких, что $0 \le \alpha < \beta \le 1$ рассматриваются подпоследовательности $U_j, U_{j+1}, \ldots, U_{j+r}$, в которых U_{j+r} лежит между α и β, а другие U_s не лежат между этими числами. Эту

последовательность, состоящую из $r+1$ чисел, называют интервалом длиной r. Основу критерия интервалов составляет подсчёт количества интервалов разной длины, и анализ их распределения.

Покер-критерий (критерий разбиений). Покер-критерий рассматривает n групп по пять последовательных целых чисел $\{Y_{5j}, Y_{5j+1}, Y_{5j+2}, Y_{5j+3}, Y_{5j+4}\}$ для $0 \le j < n$ и проверяет, какие из следующих семи комбинаций соответствуют таким пятёркам чисел (порядок не имеет значения):

Все числа разные:	*abcde*
Одна пара:	*aabcd*
Две пары:	*aabbc*
Тройка:	*aaabc*
Полный набор:	*aabb*
Четвёрка:	*aaaab*
Покер:	*aaaaa*

Использование здесь χ^2-критерия заключается в подсчёте числа пятёрок в каждой категории.

Критерий собирания купонов. Данный критерий выглядит следующим образом. Используется последовательность $Y_0, Y_1, \ldots$, находятся длины отрезков $Y_{j+1}, Y_{j+2}, \ldots, Y_{j+r}$, содержащие «полный набор» целых чисел от 0 до $d-1$.

Критерий перестановок. Тестируемая последовательность разделяется на n групп по t элементов в каждой, т. е. $(U_{jt}, U_{jt+1}, \ldots, U_{jt+t-1})$ для $0 \le j < n$. Элементы в каждой группе можно упорядочивать $t!$ различными способами. Подсчитывается число групп с любым возможным порядком и применяется χ^2-критерий с $k = t!$ возможными категориями и вероятностью $1/t!$ для каждой категории.

Критерий монотонности. В последовательности проверяется распределение восходящих и нисходящих серий, то есть *монотонных* частей заданной последовательности (отрезки возрастания или убывания). К примеру, для последовательности цифр «3475208916», восходящие серии: |347|5|2|089|16|, и нисходящие серии: |3|4|7520|8|91|6|.

Здесь следует обратить внимание на то, что применять для анализа серий χ^2-критерий нельзя, поскольку отсутствует требуемая независимость событий (длинные серии имеют тенденцию следовать за короткими и наоборот). Более подробное описание статистики, которую можно применить для данного критерия приведено в [10].

Критерий «максимум-t». Обозначим $V_j = \max(U_{tj}, U_{tj+1}, \ldots, U_{tj+t-1})$ для $0 \le j < n$. Затем полученная последовательность $V_0, V_1, \ldots, V_{n-1}$ проверяется по критерию Колмогорова-Смирнова с функцией распределения $F(x) = x^t$, $0 \le x \le 1$. Либо, как

вариант, применить критерий Колмогорова-Смирнова к последовательности $V_0^t, V_1^t, \ldots, V_{n-1}^t$, проверяя гипотезу о равномерном распределении.

Критерий конфликтов. χ^2-критерий имеет серьёзное ограничение по применимости. Его можно применять лишь в случаях, когда в каждой категории ожидается ненулевое число элементов. Данный же критерий можно использовать, когда число категорий намного больше числа наблюдений.

Суть критерия конфликтов легко пояснить на следующем примере. Пусть имеется *m* урн. Поместим в них наудачу *n* шаров, причём *m* намного больше *n*. Большинство шаров попадёт в пустые урны, но если шар попадёт в урну, в которой уже содержится хотя бы один шар, то будем говорить, что произошёл «конфликт». Данный критерий подсчитывает число конфликтов. Тестируемая СЧП будет удовлетворять этому критерию, если не возникает слишком много или слишком мало конфликтов.

Критерий промежутков между днями рождений. Поместим *n* шаров в *m* урн, как и в критерии конфликтов, но под урнами будем подразумевать дни года, а под шарами – дни рождения. Предположим, что $(Y_1, \ldots, Y_n)$ – это дни рождения, $0 \le Y_k < m$. Расположим их в порядке неубывания $Y_{(1)} \le \ldots \le Y_{(n)}$, определим *n* «промежутков» $S_1 = Y_{(2)} - Y_{(1)}$, $\ldots, S_{n-1} = Y_{(n)} - Y_{(n-1)}$, $S_n = Y_{(1)} + m - Y_{(n)}$ и расположим промежутки в порядке неубывания $S_{(1)} \le \ldots \le S_{(n)}$. Теперь подсчитываем количество равных промежутков *R*, а именно – число индексов *j*, таких, что $1 < j \le n$ и $S_{(j)} = S_{(j-1)}$. Таким образом, значение *R* и является критерием. В таблице 1 приведён пример вероятностей для величины *R* в случае следующих параметров: $m = 2^{25}$ и $n = 512$.

Таблица 1 – Соотношение величины и вероятности

Наименование показателя	Значение			
Величина *R*	0	1	2	≥ 3
Вероятность	0,368801577	0,369035243	0,183471182	0,078691997

Для подсчёта процентных точек существует точный алгоритм. Также следует отметить тот интересный факт, что данный критерий является первым, на котором провалилась последовательность Фибоначчи с запаздыванием (всем предыдущим критериям последовательность удовлетворяет).

Критерий сериальной корреляции. Данный критерий призван обнаружить корреляцию между элементами последовательности U_{j+1} от U_j для $0 \le j < n$. Для этого, формула для коэффициента корреляции между величинами $U_0, U_1, \ldots, U_{n-1}$ и $V_0, V_1, \ldots, V_{n-1}$:

$$C = \frac{n\sum_j (U_j V_j) - \left(\sum_j U_j\right)\left(\sum_j V_j\right)}{\sqrt{\left(n\sum_j U_j^2 - \left(\sum_j U_j\right)^2\right)\left(n\sum_j V_j^2 - \left(\sum_j V_j\right)^2\right)}}, \qquad (25)$$

преобразуется к виду:

$$C = \frac{n(U_0U_1 + U_1U_2 + \ldots + U_{n-2}U_{n-1} + U_{n-1}U_0) - (U_0 + U_1 + \ldots + U_{n-1})^2}{n(U_0^2 + U_1^2 + \ldots + U_{n-1}^2) - (U_0 + U_1 + \ldots + U_{n-1})^2}, \qquad (26)$$

посредством замены $V_j = U_{(j+1) \bmod n}$.

Критерием «качественности» будет являться принадлежность коэффициента корреляции *C* интервалу $\mu_n - 2\sigma_n < C < \mu_n + 2\sigma_n$, где:

$$\mu_n = \frac{-1}{n-1}, \qquad \sigma_n^2 = \frac{n^2}{(n-1)^2(n-2)}, \qquad n > 2. \qquad (27)$$

Ожидается, что *C* находится между этими значениями в 95 % случаев.

1.2.2 Пакеты статистических тестов

На сегодняшний день разработано достаточно большое количество различных типов генераторов СЧП. Для демонстрации и возможности сравнения их статистических свойств принято использовать некоторые стандартные наборы (пакеты) статистических тестов, объединенные единой методикой расчета необходимых показателей эффективности ГСЧП и принятия решения о случайности формируемых последовательностей.

Наиболее известным набором статистических тестов является набор из пяти тестов, предложенный Кнутом в его классической работе «Искусство программирования для ЭВМ» [10]. Однако в [10] были предложены только тесты, тогда как вопросы методики их применения рассматриваются недостаточно полно.

Первым шагом к стандартизации набора статистических тестов принято считать [12] принятие в 1994 году национального стандарта США *FIPS 140-1 «Требования безопасности для криптографических модулей»* [11].

В стандарте FIPS 140-1 определен набор из четырех тестов (тесты монобит, четырехграмм, серий и длинных серий) для статистического контроля качества СЧП, применяемой в криптографических модулях [11]. Для каждого теста в FIPS 140-1 приведены рассчитываемые показатели и их пороговые (предельные) значения.

В 1999 году специалистами NIST в рамках проекта AES (Advanced Encryption Standard) был разработан набор статистических тестов *NIST STS (NIST Statistical Test Suite)* и предложена методика проведения статистического

тестирования ГСЧП. В 2000 году опубликован стандарт NIST – SP 800-22, определяющий набор статистических тестов *NIST*. В 2010 году вышла используемая в работе версия пакета статистических тестов NIST – SP 800-22 Revision 1a [13].

Пакет статистических тестов NIST включает в себя 15 статистических тестов, которые разработаны для проверки гипотезы о случайности двоичных последовательностей произвольной длины. Все тесты направлены на выявление различных дефектов случайности. Основным принципом тестирования является проверка нулевой гипотезы H_0, заключающейся в том, что тестируемая последовательность является случайной. Альтернативной гипотезой H_1 является гипотеза о том, что тестируемая последовательность не случайна. По результатам применения каждого теста нулевая гипотеза либо принимается, либо отвергается. Решение о случайности заданной последовательности нулей и единиц принимается по совокупности результатов всех тестов.

Порядок тестирования отдельной двоичной последовательности S выглядит следующим образом:

1) Выдвигается нулевая гипотеза H_0 – предположение о том, что данная двоичная последовательность S случайна.

2) По последовательности S вычисляется статистика теста $c(S)$.

3) С использованием специальной функции и статистики теста вычисляется значение вероятности $P = f(c(S))$, $P \in [0, 1]$.

4) Значение вероятности P сравнивается с уровнем значимости α, $\alpha \in [0{,}001, 0{,}01]$. Если $P \geq \alpha$, то гипотеза H_0 принимается. В противном случае принимается альтернативная гипотеза.

В таблице 2 приводятся сводные данные по всем тестам с указанием физического смысла статистики теста и дефекта, на выявление которого направлен тест.

Таблица 2 – Описание статистических тестов

№ п/п	Статистический тест	Статистика теста $c(S)$	Выявляемый дефект
1	Частотный (монобит) тест (Frequency (Monobit) Test)	Нормализованная абсолютная сумма значений элементов последовательности	Слишком много нулей или единиц в последовательности
2	Частотный тест внутри блока (Frequency Test within a Block)	Мера согласования наблюдаемого количества единиц внутри блока с теоретически ожидаемым	Локализованные отклонения частоты появления единиц в блоке от идеального значения ½
3	Тест серий (Runs Test)	Общее количество серий на всей длине последовательности	Слишком быстрая или слишком медленная перемена знака в ходе генерации последовательности
4	Тест максимальной длины серии из единиц в блоке (Test for the Longest Run of Ones in a Block)	Мера согласования наблюдаемого значения максимальной длины единичной серии с теоретически ожидаемым значением	Отклонение от теоретического закона распределения максимальных длин серий единиц
5	Тест рангов бинарных матриц (Binary Matrix Rank Test)	Мера согласования наблюдаемого значения рангов различного порядка с теоретически ожидаемым	Отклонение эмпирического закона распределения значений рангов матрицы от теоретического, что указывает на зависимость символов в последовательности
6	Спектральный тест на основе дискретного преобразования Фурье (Discrete Fourier Transform (Spectral) Test)	Нормализованная разница между наблюдаемым и ожидаемым количеством частотных компонент, которые превышают 95 % пороговый уровень	Выявление периодических составляющих (трендов) в двоичной последовательности
7	Проверка неперекрывающихся шаблонов (Non-overlapping Template Matching Test)	Мера согласования наблюдаемого количества непериодических шаблонов в последовательности с теоретическим значением	Большое количество заданных непериодических шаблонов в последовательности
8	Проверка перекрывающихся шаблонов (Overlapping Template Matching Test)	Мера согласования наблюдаемого количества перекрывающихся шаблонов в последовательности с теоретическим значением	Большое количество m-битных серий из единиц в последовательности
9	Универсальный статистический тест Маурера	Сумма логарифма расстояния между l-битными шаблонами	Сжимаемость последовательности без потери информации

Продолжение таблицы 2

№ п/п	Статистический тест	Статистика теста $c(S)$	Выявляемый дефект
10	Тест на линейную сложность (Linear Complexity Test)	Мера согласования наблюдаемого количества событий, заключающихся в появлении фиксированной длины эквивалентного линейного регистра сдвига с обратной связью для заданного блока с теоретически ожидаемым	Отклонение эмпирического распределения длин эквивалентных линейных регистров сдвига с обратной связью для последовательности фиксированной длины от теоретического закона распределения для случайной последовательности, что указывает на недостаточную сложность тестируемой последовательности
11	Последовательный тест (Serial Test)	Мера согласования наблюдаемого количества всех встретившихся вариантов m-битных шаблонов с теоретически ожидаемым	Неравномерность распределения m-битных слов в последовательности
12	Тест приблизительной энтропии (Approximate Entropy Test)	Мера согласования наблюдаемого значения энтропии источника с теоретически ожидаемым для случайного источника	Неравномерность распределения *m*-битных слов в последовательности (регулярность свойств источника)
13	Тест кумулятивных сумм (Cumulative Sums (Cusum) Test)	Максимальное отклонение значения кумулятивной суммы элементов последовательности от начальной точки отсчета (точка 0)	Большое значение единиц или нулей вначале или в конце двоичной последовательности
14	Тест на произвольные отклонения (Random Excursions Test)	Мера согласования наблюдаемого количества визитов при случайном блуждании в заданное состояние внутри цикла с теоретически ожидаемым	Отклонение от теоретического закона распределения визитов в конкретное состояние при случайном блуждании
15	Вариант теста на произвольные отклонения (Random Excursions Variant Test)	Общее количество визитов в заданное состояние при случайном блуждании	Отклонение от теоретического ожидаемого общего количества визитов при случайном блуждании в заданное состояние

Тесты пакета NIST доступны в исходных текстах на языке программирования C на сайте NIST [14].

Еще одним популярным набором статистических тестов проверки качества СЧП является *пакет статистических тестов Diehard* [14]. Они были разработаны Джорджем Марсальей (англ. George Marsaglia) в течение нескольких лет и представлены для общего доступа. Вместе они рассматриваются как один из наиболее строгих известных наборов тестов [15].

Пакет Diehard включает следующие основные тесты:

Дни рождения (Birthday Spacings) – выбираются случайные точки на большом интервале. Расстояния между точками должны быть асимптотически распределены по Пуассону. Название этот тест получил на основе парадокса дней рождения.

Пересекающиеся перестановки (Overlapping Permutations) – анализируются последовательности пяти последовательных случайных чисел. 120 возможных перестановок должны получаться со статистически эквивалентной вероятностью.

Ранги матриц (Ranks of matrices) – выбирается некоторое количество бит из некоторого количества случайных чисел для формирования бинарной матрицы определенной размерности, затем определяется ранг матрицы. Считаются ранги.

Обезьяньи тесты (Monkey Tests) – последовательности некоторого количества бит интерпретируются как слова. Считаются пересекающиеся слова в потоке. Количество «слов», которые не появляются, должны удовлетворять известному распределению. Название этот тест получил на основе теоремы о бесконечном количестве обезьян.

Подсчёт единичек (Count the 1’s) – считаются единичные биты в каждом из последующих или выбранных байт. Эти счётчики преобразуется в «буквы», и считаются случаи пятибуквенных «слов».

Тест на парковку (Parking Lot Test) – единичные окружности случайно размещаются в квадрате 100×100. Если окружность пересекает уже существующую, попытаться ещё. После 12 000 попыток, количество успешно «припаркованных» окружностей должно быть нормально распределено.

Тест на минимальное расстояние (Minimum Distance Test) – 8000 точек случайно размещаются в квадрате 10 000×10 000, затем находится минимальное расстояние между любыми парами. Квадрат этого расстояния должен быть экспоненциально распределён с некоторой медианой.

Тест случайных сфер (Random Spheres Test) – случайно выбираются 4000 точек в кубе с ребром 1000. В каждой точке помещается сфера, чей радиус является минимальным расстоянием до другой точки. Минимальный объём сферы должен быть экспоненциально распределён с некоторой медианой.

Тест сжатия (The Squeeze Test) – 231 умножается на случайные вещественные числа в диапазоне [0,1) до тех пор, пока не получится 1.

Повторяется 100 000 раз. Количество вещественных чисел необходимых для достижения 1 должно быть распределено определённым образом.

Тест пересекающихся сумм (Overlapping Sums Test) – генерируется длинная последовательность на [0,1). Добавляются последовательности из 100 последовательных вещественных чисел. Суммы должны быть нормально распределены с характерной медианой и сигмой.

Тест последовательностей (Runs Test) – генерируется длинная последовательность на [0,1). Подсчитываются восходящие и нисходящие последовательности. Числа должны удовлетворять некоторому распределению.

Тест игры в кости (The Craps Test) – играется 200 000 игр в кости, подсчитываются победы и количество бросков в каждой игре. Каждое число должно удовлетворять некоторому распределению.

Тесты пакета Diehard доступны в исходных текстах на языке программирования C на сайте автора [14].

Кроме общедоступных пакетов тестов NIST и Diehard имеются и другие наборы статистических тестов. Например, пакеты Crypt-X – коммерческий программный пакет, разработанный Центром исследований информационной безопасности Квинслендского технологического университета (Information Security Research Centre at Queensland University of Technology) в Австралии) [16], и TestU01 – программная библиотека статистических тестов ГСЧП, разработанная Pierre L'Ecuyer и Richard Simard в Университете Монреаля, Канада, и доступная в исходных текстах на языке программирования C [17].

1.2.3 Критерии принятия решения о прохождении теста

Для принятия решения о прохождении последовательностью случайных (псевдослучайных) чисел статистического теста используются следующие три основных вычислительных подхода [11].

Пусть дана двоичная последовательность $S = \{s_1, s_2, \ldots, s_n\}$, $s_i \in \{0,1\}$ длиной n бит. Необходимо принять решение, проходит данная последовательность статистический тест или нет. Возможны следующие подходы решения этой задачи.

1) Критерий принятия решения на основе задания порогового уровня. Данный подход основан на вычислении по данной последовательности S статистики теста $c(S)$, с её последующим сравнением с некоторым пороговым уровнем $c_{\text{пор}}(S)$. Критерий принятия решения формулируется следующим образом: *считается, что двоичная последовательность S не проходит статистический тест всякий раз, когда статистика теста $c(S)$ принимает значение меньшее, чем пороговый уровень $c_{\text{пор}}(S)$.*

2) Критерий принятия решения на основе задания фиксированного доверительного интервала. При данном подходе критерий принятия решения формулируется следующим образом: *считается, что двоичная последовательность S не проходит статистический тест, если значение статистики теста c*(*S*) *находится вне пределов доверительного интервала значений статистики, вычисленного для заданного уровня значимости* α. Например, пусть к двоичной последовательности *S* длиной $n = 800$ бит применяется частотный тест. Значение статистики теста *c*(*S*) есть число единиц в последовательности *S*, причём ожидается, что в последовательности будет приблизительно 400 единиц и 400 нулей. Если зафиксировать уровень значимости на уровне 5 % (α = 0,05), то последовательность *S* не пройдет частотный тест, если число единиц будет находиться вне доверительного интервала $400 \pm 1{,}96/2 \times \sqrt{800} = [373{,}427]$.

Данный критерий является более надежным по сравнению с первым. Необходимо только учитывать, что различным уровням значимости будут соответствовать различные доверительные интервалы.

3) Третий подход построения критерия принятия решения опирается на вычисление для статистики теста *c*(*S*) соответствующего значения вероятности *P*. Здесь статистика теста рассматривается как реализация случайной величины, которая подчиняется известному закону распределения. Статистка теста строится таким образом, чтобы её большие значения указывали на какой-либо дефект случайности последовательности. Значение вероятности *P* есть вероятность того, что статистика теста примет значение большее, чем наблюдаемое при опыте в предположении случайности последовательности. Следовательно малые значения *P* (*P* < 0,05 или *P* < 0,01) интерпретируются как доказательство того, что последовательность не случайна. Решающее правило формулируется так: *для фиксированного уровня значимости* α, *двоичная последовательности S не проходит статистический тест если значение вероятности P* < α. Значения α рекомендуется выбирать из интервала [0,001, 0,01].

Например, пусть последовательность *S* содержит 10^6 бит. Применим к последовательности тест серий, статистикой которого является общее количество серий *V*, которое в данном случае должно быть близко к значению 500 000. Предположим, что в ходе тестирования мы получили $V = 499996$. Тогда

$$P = erfc\left(\left|\frac{V - 2n\rho(1-\rho)}{2\sqrt{2n}\rho(1-\rho)}\right|\right) = 0{,}994876$$

где n – длина последовательности;

ρ – общее количество единиц деленное на n;

$erfc$ – дополнительная функция ошибок.

Поскольку $P > 0{,}01$, то последовательность S тест прошла.

Использование данного подхода имеет дополнительное преимущество по сравнению с предыдущим, которое заключается в том, что однажды рассчитанное значение вероятности P может сравниваться с произвольно выбранным уровнем значимости α без проведения дополнительных расчетов. Обычно значение вероятности P определяется с использованием

функции стандартного нормального распределения $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{\frac{-u^2}{2}} du$;

дополнительной функции ошибок $erfc = \frac{2}{\sqrt{\pi}} \int_{-z}^{\infty} e^{-u^2} du$;

неполной гамма-функции $Q(a,x) \equiv 1 - P(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_{x}^{\infty} e^{-t} t^{a-1} dt$,

где $Q(a,0) = 1$ и $Q(a,\infty) = 0$.

В основу рассмотренных выше пакетов статистических тестов Diehard [14] и NIST [12] заложен именно третий критерий принятия решения.

Важным аспектом оценки качества СЧП является оценка ее независимости или непредсказуемости, что близко соотносится с энтропией.

В отличие от генераторов псевдослучайных чисел от ФГСЧП потребитель вправе ожидать, что полученные с их использованием случайные числа остаются инвариантными по времени. Следовательно, разумно использовать ФГСЧП по крайней мере для генерации случайных чисел, которые должны быть защищены в долгосрочной перспективе. При этом для гарантии качества СЧП необходима оценка качества, основанная на расчете теоретически безопасных границ, определяемых вероятностью предсказания числа для нахождения СЧП.

Важнейшим критерием оценки ФГСЧП является оценка величины энтропии в случайном бите. Решение данной задачи может быть целесообразным, когда необходимы максимальные гарантии подтверждения качества используемой СЧП. Например, данная задача актуальна для ГСЧП, используемых в ИТ-системах, обрабатывающих информацию с грифом секретности.

Энтропия не может быть измерена как напряжение или температура. Энтропия является свойством случайных переменных, а не последовательностей наблюдаемых значений, которые обобщены этими случайными переменными. Как первое следствие, энтропия не может быть простым гарантом, вне зависимости от количества и состава, применяемых для оценки качества статистических тестов черного ящика (более того, обычно даже псевдослучайные последовательности проходят эти тестовые наборы черного ящика). Поэтому, чтобы определить количество энтропии на случайный бит,

вначале необходимо изучить распределение случайных чисел, или более точно, распределение основных случайных переменных.

Случайные переменные, например, $R_1,R_2,\ldots$ и $Y_1,Y_2,\ldots$ квалифицируют стохастическое поведение случайных чисел и внутренних случайных чисел, соответственно. Эти распределения зависят от источника шума и механизма дискретизации (оцифровывания) для Y_j, а также от последующей алгоритмической постобработки. Обычно сложно точно определить эти распределения. Более того, точное распределение зависит от компонент для частного источника шума, которые могут отличаться на некоторую степень даже для ФГСЧП из той же самой серии. Вместо этого следует специфицировать семейство распределений и представить доказательство, что истинное распределение случайных чисел (также как и специфицированных вспомогательных случайных переменных) всегда содержится (т.е., для всех копий этого ФГСЧП и для всех условий использования) в этом семействе.

Конечной целью оценки является определение, по меньшей мере, нижней границы для средней энтропии на внутреннее случайное число. В идеальном случае стохастическая модель источника шума включает распределение для внутренних случайных чисел или распределение вспомогательных случайных переменных, которое позволяет вычислять энтропию для внутренних случайных чисел. Необходимо особо отметить, что стохастическая модель не является физической моделью источника шума и его механизма дискретизации.

В общем случае для решения задачи оценки энтропии необходимо [18]:

– сформулировать стохастическую модель для конкретного ФГСЧП. Настроить эту модель. Попытаться подтвердить свои предположения экспериментально, если не имеется четкого математического доказательства;

– использовать этот ФГСЧП для генерирования случайных чисел. Оценить неизвестные параметры, которые принадлежат этому ФГСЧП на основе этих случайных чисел;

– использовать эти оценки параметров для получения оценки энтропии на случайный бит.

1.3 Возможности обработки СП для получения СЧП

При считывании сигналов на выходе физического источника шума наиболее часто используются два метода генерации бинарных СЧП:

– дискретизация непрерывного шумового сигнала по двум уровням;

– подсчёт импульсов шумового сигнала в течение фиксированного интервала времени.

Каждый из этих способов имеет как преимущества, так и недостатки, что и является причиной использования обоих методов.

Для оценки преимуществ и недостатков данных способов введём меру неопределённости случайной величины, с использованием которой формируется элемент выходной бинарной СЧП.

Мерой неопределённости случайной величины, принимающей конечное множество значений $\{a_i\}$ ($i = 1,...,n$), определяемых распределением вероятностей $p_i = p(a_i)$ ($0 \le p_i \le 1$, $\sum_i p_i = 1$), является энтропия [6]

$$H = -\sum_{i=1}^{n} (p_i \log_K p_i), \qquad (9)$$

В двоичной системе счисления ($K = 2$) равномерно распределённая случайная последовательность представляет собой ряд из «0» и «1», вероятности появления которых в идеальном случае $p\{"0"\} = p\{"1"\} = 0.5$. В этом случае энтропия равна единице.

Для дальнейшего анализа введём вероятностную модель ГСЧП, в рамках которой генератор будет рассматриваться как источник сообщений [7].

Каждый символ сообщения ξ может принимать значения из алфавита $A = \{a_1 = "0", a_2 = "1"\}$ с мощностью $n = 2$. Для описания неопределённости, содержащейся в зарегистрированном в результате проведения эксперимента символе ξ, введём вероятностное пространство (Ω, F, P), где Ω – пространство элементарных событий, F – множество случайных событий, P – вероятностная мера. Тогда каждому элементарному исходу $\omega \in \Omega$ этого эксперимента ставится в соответствие значение символа $\xi = \xi(\omega) \in A$, и $\mathrm{P}\{\xi = a_i\} = p_i$.

Для рассмотренного выше случая, в предположении идеальности источника, пространство элементарных событий Ω должно делиться на два непересекающихся подпространства $\Omega = \Omega_1 \cup \Omega_2$ таких, что $\forall \omega \in \Omega$ будет выполняться

$$\mathrm{P}(\omega \in \Omega_1) = \mathrm{P}(\omega \in \Omega_2), \qquad (10)$$

Отсюда можно ввести отклонение распределения СЧП от равномерного распределения (ε), как разность вероятностей $p\{"1"\}$ и $p\{"0"\}$.

Суть метода *дискретизации непрерывного шумового сигнала по двум уровням* заключается в том, что значения элементов выходной бинарной последовательности формируются на основании того, является ли в определённый момент времени *T* случайный шумовой параметр выше или ниже заданного уровня.

На рисунке 2 приведён пример шумового сигнала (слева) – равномерно распределённая (в диапазоне от 0,0(0) до 0,9(9)) случайная величина, а также ее

бинарный вариант, в котором значение «1» присваивалось при условии превышения шумом порогового значения 0,49(9), и 0 в обратном случае.

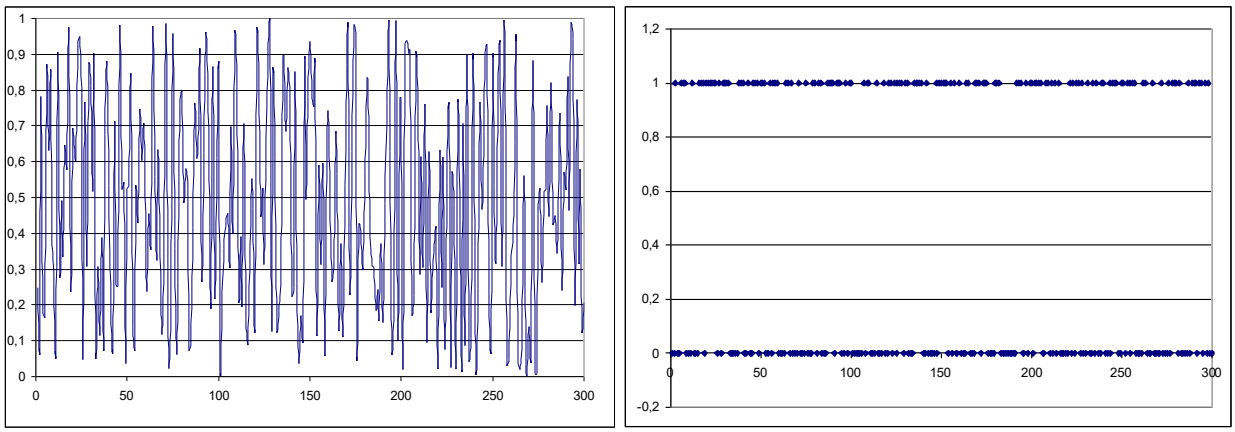

Рисунок 2 – Пример дискретизации шумового сигнала по двум уровням (слева – шумовой сигнал, справа – бинарный вариант сигнала после дискретизации)

Проанализируем статистические характеристики данного метода. Пусть в момент времени T наступило событие A, если имеет место превышение заданного уровня. Индикатором события A называется случайная величина $I_A(\omega)$ такая, что

$$I_A(\omega)=\begin{cases}1, & \omega\in\Omega_1;\\ 0, & \omega\in\Omega_2.\end{cases}, \qquad (11)$$

Таким образом, I_A является бернуллиевской случайной величиной [7]. Обычно данный метод реализуется с использованием теплового шума резистора, который имеет равномерный спектр до частот, где сохраняется постоянным сопротивление источника шума. Распределение амплитуды такого шума описывается нормальным законом, а значит, изменение соотношения объёмов подпространств Ω_1 и Ω_2 может быть существенным в зависимости от соотношения величин дисперсии напряжения шума и дрейфа его среднего уровня. Тогда энтропия (9) будет равна

$$H=-\left(0{,}5+\frac{\varepsilon}{2}\right)\log_2\left(0{,}5+\frac{\varepsilon}{2}\right)-\left(0{,}5-\frac{\varepsilon}{2}\right)\log_2\left(0{,}5-\frac{\varepsilon}{2}\right), \qquad (12)$$

Самыми существенными недостатками генерации СЧП на основе дискретизации сигнала является низкая стабильность основных вероятностных характеристик, объясняемая нестабильностью первичных источников флуктуаций, дрейфом параметров преобразующих схем, источников питания и т.д., что требует постоянного контроля качества СЧП. Более того – значительно усложняется реализация таких генераторов, поскольку необходимо использовать несколько источников питания (для физических источников шума и для

цифровых схем), а также необходимостью фильтрации и стабилизации напряжения источников, развязки линий питания от линий устройства.

Основным преимуществом данного метода является высокое быстродействие.

В методе *подсчёта импульсов СЧП за фиксированный интервал времени* элементы выходной бинарной последовательности формируются посредством подсчёта количества событий за фиксированный интервал времени и, если количество событий чётное, то элемент принимается равным 0, а если нечётное, то 1.

Обычно предполагается, что количество импульсов $N(T)$, зарегистрированных на интервале времени $(0,T]$ подчиняется распределению Пуассона

$$P\{N(T)=i\}=\frac{(\lambda T)^i}{i!}e^{-\lambda T}, \quad i=0,1,\ldots, \tag{13}$$

где λ – интенсивность потока.

Вследствие конечности интервала $(0,T]$ разность между суммами вероятностей выпадения чётного и нечётного количества импульсов равна

$$\varepsilon_p=\sum_{i=0}^{\infty}(-1)^i P(i)=e^{-2\lambda T}, \tag{14}$$

Например, для среднего числа импульсов $M\{N(T)\}=\lambda T=10$ имеем $\varepsilon_p \approx 2\cdot 10^{-9}$, откуда значение энтропии $H > (1 - 10^{-17})$.

Достоинством данного способа является высокая устойчивость к различным возмущениям (дрейф порогового напряжения устройства селекции импульсов, дрейф постоянного тока через источник шума).

На практике используются различные методы постобработки. Ниже приведены наиболее популярные из них.

Криптографические хэш-функции. Возможно, наиболее популярной и устойчивой методикой постобработки является применение криптографически сильной хэш-функции. Реализация полной хэш-функции для ФГСЧП существенно снижает производительность. Но обеспечивает дополнительный барьер для исключения возможности анализа источника случайности по выходу. Кроме того, нелинейность хэш-функции становится полезной, если найдена слабость в механизме сбора выборки СЧП. Хорошей стратегией является реализация односторонней хэш-функции как последнего шага в работе программного обеспечения.

Корректор фон Неймана. Корректор фон Неймана – один из наиболее ранних и хорошо известных методов постобработки. Он используется для ограничения локальных смещений. Корректор берет очередную пару бит из случайного битового потока. Если имеются идентичные значения (т.е., оба бита

равны 0 или оба бита равны 1), то они удаляются из случайного битового потока. Если они различны, то используется только один бит, например, первый бит. В среднем битовая скорость будет уменьшена приблизительно до 1/4 от входной битовой скорости. Существенным преимуществом корректора фон Неймана является его очень простая реализация.

Функции извлечения (*экстрактор*). Функции извлечения были предложены Бараком, Шалтиелом и Томером в [3] с целью создания ФГСЧП, устойчивых к изменениям условий среды. Функции извлечения – мощные функции без запоминания с измеримыми свойствами, первоначально разработанными как инструмент в теории сложности. Авторы разработали математическую модель для извлечения влияния противника на источник случайности и предложили механизм, основанный на универсальных хэш-функциях, который обеспечивает приемлемые свойства выхода, даже если существуют нелокальные корреляции во входном источнике. Ниже дается ряд определений относительно функций извлечения.

Статистическое расстояние между двумя распределениями X и Y определяется как:

$$\varepsilon = \frac{1}{2}\sum_{a} | P(X = a) - P(Y = a) | . \qquad (15)$$

На практике говорят, что X является ε-близким к Y и наоборот.

Распределение X на $\{0, 1\}^n$ имеет минимум энтропии k, если для всех $x \in \{0, 1\}^n$ P $(X = x) \le 2^{-k}$.

В общем, экстрактор является функцией, характеризуемой относительно его поведения по входу-выходу. Экстрактор рассматривается как преобразователь, на который поступает входная последовательность с определенным уровнем минимальной энтропии k, и который гарантирует выходное распределение, ε-близкое к равномерному распределению.

В [3] авторы дают расширение этого определения. Они определяют функцию E: $\{0, 1\}^n \rightarrow \{0, 1\}^m$, которая устанавливается выбором публичного параметра π. Они позволяют противнику выбирать из 2^t распределений $D_1, D_2, ..., D_{2^t}$ над $\{0, 1\}^n$ такие, что минимальная энтропия каждого из D_i выше, чем k для всех $i = 1, 2, \ldots, 2^t$. Публичный параметр π выбирается случайно и независимо для выбранных D_i. Противник выбирает одно из распределений D_u. Пользователь оценивает функцию извлечения, используя публичный параметр π и значение, выведенное из выбранного распределения D_u.

В [3] t-*эластичная функция извлечения* определяется следующим образом. Задавая m, k, ε и t, экстрактор E: $\{0, 1\}^n \rightarrow \{0, 1\}^m$ является t-эластичным, если с вероятностью максимум 1 - ε, при выбранном публичном параметре π, статистическое расстояние (относительно равномерного распределения) для выходного распределения $E^{\pi}(X)$ составляет максимум ε.

На практике это означает, что противник предполагает иметь контроль над t бинарными значениями (или 2^t внутренними состояниями) ФГСЧП через управление напряжением, температурой, тактовой частотой и т.п. Несмотря на способности противника, выходное распределение смещается относительно равномерного самое большее на ε. Этот механизм предоставляет большие возможности для разработчиков ФГСЧП, так как позволяет зафиксировать любой вид влияния противника. С другой стороны, с точки зрения проекта, не ясно, как квалифицировать способности противника, и поэтому сложно выбрать параметры проекта для экстрактора (или для универсального семейства хэш-функций).

Эластичные функции. Эластичные функции были предложены Санером, Мартином и Стинсоном в [8] как шаг постобработки для проекта колец. Целью было отфильтровать любые детерминированные биты путем использования эластичной функции. Обработка битов, которые под влиянием противника становятся детерминированными, дает возможность изучать свойств допуска эластичных функций против активных противников. В [8] рекомендуется использовать высокую степень эластичности для удаления детерминированных бит. Разность между степенью эластичной функции и числом детерминированных бит, ожидаемых в окне выборки, определяет величину допуска (в битах) для ФГСЧП по отношению к активным противникам.

Эластичные функции формально определяются следующим образом: (n, m, t) - эластичной функцией является функция $F(x_1, x_2, ..., x_n) = (y_1, y_2, ..., y_m)$ из Z_2^n в Z_2^m, которая обладает таким свойством, что для любых t координат i_1, …, i_t, для любых констант a_1, …, a_t из Z_2 и любого элемента y области значений:

$$P\{F(x) = y \mid x_{i_1} = a_1, ..., x_{i_l} = a_l\} = \frac{1}{2^m}, \qquad (16)$$

При вычислении этой вероятности все x_i для $i \notin \{ i_1, …, i_t\}$ рассматриваются как независимые случайные переменные, каждая из которых принимает значение 0 или 1 с вероятностью 0.5.

В более неформальных терминах, если до t любых входных бит детерминированы и оставшиеся – случайны, выход эластичной функции будет совершенно случайным (или непредсказуемым). С криптографической точки зрения, знание любых t значений на входе функции не позволяет увеличить знание относительно выхода. Эластичные функции используются в ряде криптографических приложений, где предполагается, что противник имеет возможность извлекать или определять некоторое число ключевых бит.

Простой метод для конструирования эластичных функций задается следующим образом [9]. Пусть G будет генерирующей матрицей для [n, m, d] линейного кода C. Определим функцию f : $\{0, 1\}^n \rightarrow \{0, 1\}^m$ по правилу f(x) = xG^T. Тогда f – (n, m, d - 1)-эластичная функция.

ГЛАВА 2
КОМПЛЕКСЫ ГЕНЕРАЦИИ СЧП

2.1 Основные способы проектирования генераторов СЧП

Проект ФГСЧП Баггини и Бусси. Ранняя разработка, проведенная Баггини и Бусси [23], как показано на рисунке 3, использует комбинацию аналоговых и цифровых компонент для усиления и выборки белого шума.

Проект построен для сопротивления вариациям условий эксплуатации и поведения компонентов. В [23] дается аналитическая модель для ФГСЧП, которая фиксирует отношения между максимальной битовой корреляцией и выходной битовой скоростью, и поэтому полагает, что нет необходимости в статистическом тестировании. В [23] не даются какие-либо сведения о результатах реализации.

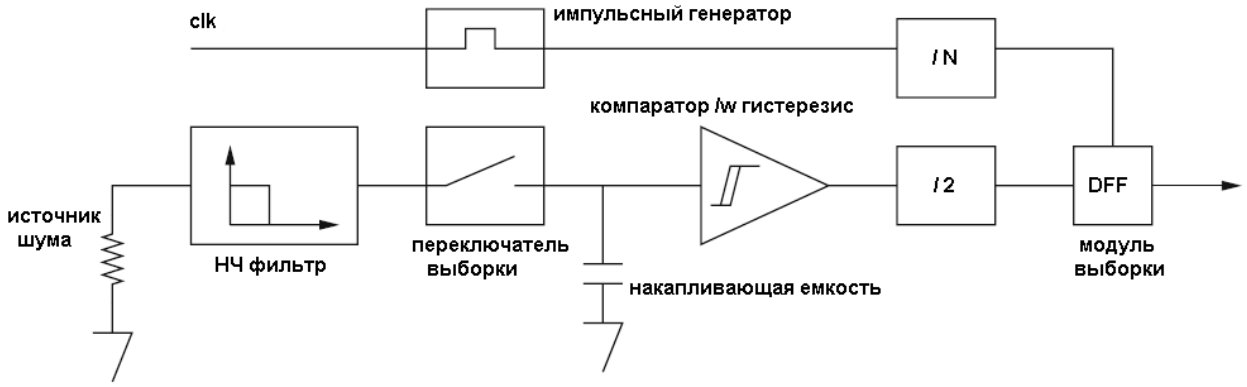

Рисунок 3 – Проект ФГСЧП Баггини и Бусси

Проект ФГСЧП Intel. Проект, показанный на рисунке 4, был описан в [4]. Источником энтропии в проекте является тепловой шум p-n перехода. При этом используются два резистора в различной конфигурации для обеспечения функционирования, более устойчивого к изменениям по питанию и условиям внешней среды. Дифференциальный тепловой шум усиливается и используется на входе управляемого напряжением осциллятора (VCO). VCO затем выбирается другим осциллятором. Выходная последовательность подвергается постобработке, используя корректор фон Неймана, и затем хэшируется, используя SHA-1. В качестве дополнительной меры безопасности применяется программный драйвер, который связывается с ФГСЧП и осуществляет монобитный тест случайности, а также тест четырехграмм согласно FIPS 140-1.

Jun и Kocher [4], которые анализировали ФГСЧП выход, использовали 16 специализированных тестов и набор тестов FIPS 140-1. В результате не были обнаружены слабости в ФГСЧП выходе перед обработкой SHA-1. Отмечается, однако, что метод постобработки фон Неймана является необходимым для устранения смещений в выходном потоке.

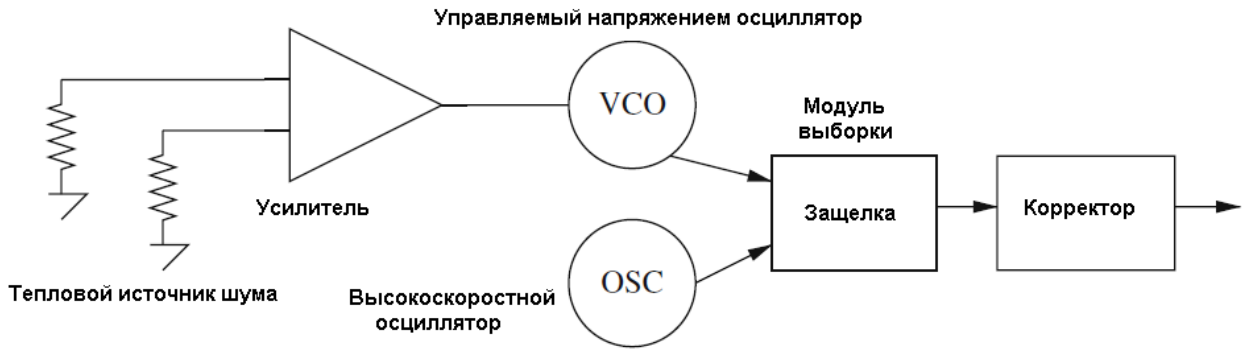

Рисунок 4 – Проект ФГСЧП Intel

Так как в [4] даются только подробности анализа защиты, ничего не известно относительно эффективности проекта, т.е. производительности и потребляемой мощности. Можно только предполагать, что затраты будут низкими из-за простоты проекта, так как SHA-1 реализован программно. Относительно безопасности имеется только анализ черного ящика, проведенный Jun и Kocher. С другой стороны, так как проект относительно простой, посредством моделирования шума перехода и флуктуации генератора можно анализировать качество и эффективность ФГСЧП выхода. В заключение необходимо отметить, что проект имеет аналоговые компоненты, т.е. усилитель шума и управляемый напряжением осциллятор, и поэтому не представляется возможным реализовать его на реконфигурируемой платформе.

Проект ФГСЧП Ткасика. Инновационный проект, представленный в [24], основывается на XOR случайных выборок бит, полученных на выходах сдвигового регистра с линейной обратной связью (LFSR) и сдвигового регистра на основе клеточного автомата (CASR). Случайность формируется из флуктуаций двух свободно осциллирующих генераторных схем, которые используются для тактирования двух детерминированных схем. Проект показан на рисунке 5.

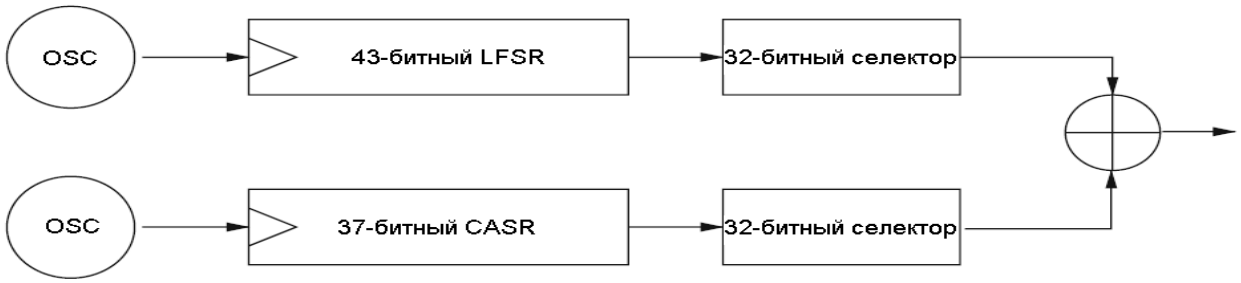

Рисунок 5 – Проект ФГСЧП Ткасика

ФГСЧП выводит 32 бита одновременно. Автор заявляет, что этот проект с небольшими вариациями используется в фирме Motorola в течение ряда лет.

Выходной поток верифицируется, используя наборы тестов Diehard [15], FIPS 140-1 [11] и Crypt-X [17]. Автор показывает, что выход полного проекта имеет гораздо лучшее статистическое поведение по сравнению с отдельным выходом LFSR или CASR. Не имеется деталей относительно аспектов производительности проекта.

В [26] Дихтл выделил атаку на эту частную ФГСЧП конструкцию, основанную на двух слабостях проекта:

– источник энтропии довольно ограничен (используются только два осциллятора). Фактически, LFSR и CASR действуют как генераторы псевдослучайных чисел, выбранные только двумя низко энтропийными осцилляторами;

– проект использует линейные компоненты (например, LFSR), и поэтому атакующий может построить линейную модель и решить ее.

Атака позволяет противнику предсказать выходные биты, имея доступ к более ранним битам. Возможность атаки оценивалась теоретически, проверка на практической реализации не проводилась. Также предположение, что нападающий знает некоторые из предварительно сгенерированных бит, делает эту атаку непрактичной для многих приложений. С другой стороны, атака указывает на зависимость между выходными битами, и это приводит к серьезным сомнениям относительно надежности ФГСЧП Ткасика. В заключение следует отметить, что проект может быть сделан устойчивым, при значительном снижении выходной скорости и/или включении нелинейных компонентов. В [26] Шиндлер проанализировал проект Ткасика с помощью сформированной стохастической модели и разработал нижние и верхние границы энтропии для случайных выходных бит. Шиндлер также показал, что выходные биты несут достаточную энтропию, когда вывод выбирается в 60000 раз медленнее, чем предложено в [24].

Проект ФГСЧП Эпштейна и др. В [27] была предложена простая архитектура, основанная на схемах с двумя устойчивыми состояниями. На рисунке 6 показана базовая компонента ФГСЧП проекта, которая входит в состав многих других подобных модулей, и которая вычисляет XOR своих выходных бит.

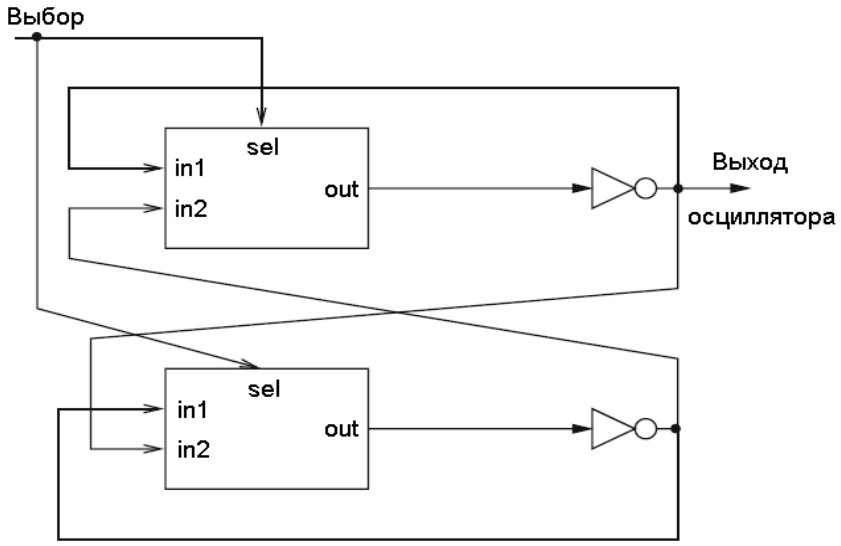

Рисунок 6 – Бистабильный компонент памяти проекта ФГСЧП Эпштейна и др.

Модуль состоит из двух мультиплексоров и двух инверторов, представляя, таким образом, конфигурацию, которая задает метастабильную схему. Если выбранный вход равен логическому 0, тогда схема переходит к двум отдельным одиночным кольцам инвертирующего осциллятора. Альтернативно, если выбранный вход равен логической 1, схема становится функционально идентичной двум каскадным инверторам. В первом режиме имеются два свободно запущенных осциллятора, а во втором – стабильная схема с не переключаемой активностью. В случае, когда выбранный вход переходит из логического 0 в 1, два свободно запущенных осциллятора могут не быть в одной и той же фазе, и получается бистабильная схема с неопределенностью в выходном сигнале до тех пор, пока не завершится переход.

ФГСЧП, который был составлен из 15 образцов компонент, показанных на рисунке 10, и дополнительных 14 XOR-схем, был изготовлен с использованием 0.18 мкм CMOS-технологии. Все выходные последовательности проходили Diehard тесты после постобработки корректором фон Неймана. Будучи созданным только из цифровых компонент, проект был также реализован на реконфигурируемой логике. Проект довольно компактен и эффективен по мощности. К сожалению, в [27] не дается информация о производительности

проекта. Выход верифицируется, используя статистические тесты. Анализ безопасности не обеспечивается.

Проект ФГСЧП Фишера-Друтаровски. Данный проект [28] выбирает флуктуацию в системе фазовой автоподстройки частоты (ФАПЧ) на платформе специализированной реконфигурируемой логики. Проект является уникальным в том смысле, что он был первым ФГСЧП, предложенным на базе FPGA. В [28] в качестве целевого устройства выбрано семейство программируемой логики фирмы Altera, которое содержит в своем составе PLL (ФАПЧ) (например, APEX E и APEX II семейства), как изображено на рисунке 7.

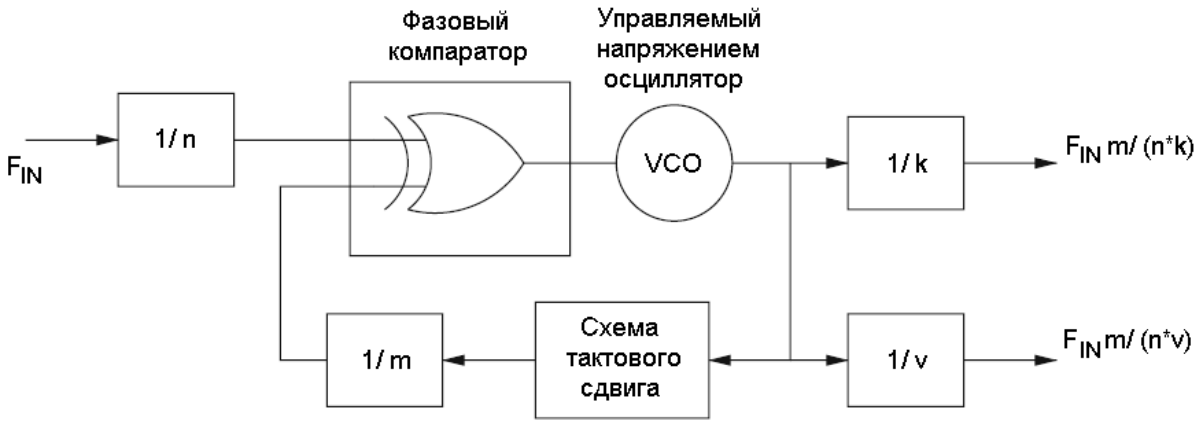

Рисунок 7 – Архитектура программируемой ФАПЧ (PLL), используемой как источник энтропии в проекте ФГСЧП Фишера-Друтаровски

Флуктуация тактового сигнала, сгенерированного модулем PLL чипа, выбирается через задержку каскадируемых модулей выборки, организованных в конфигурации, показанной на рисунке 8.

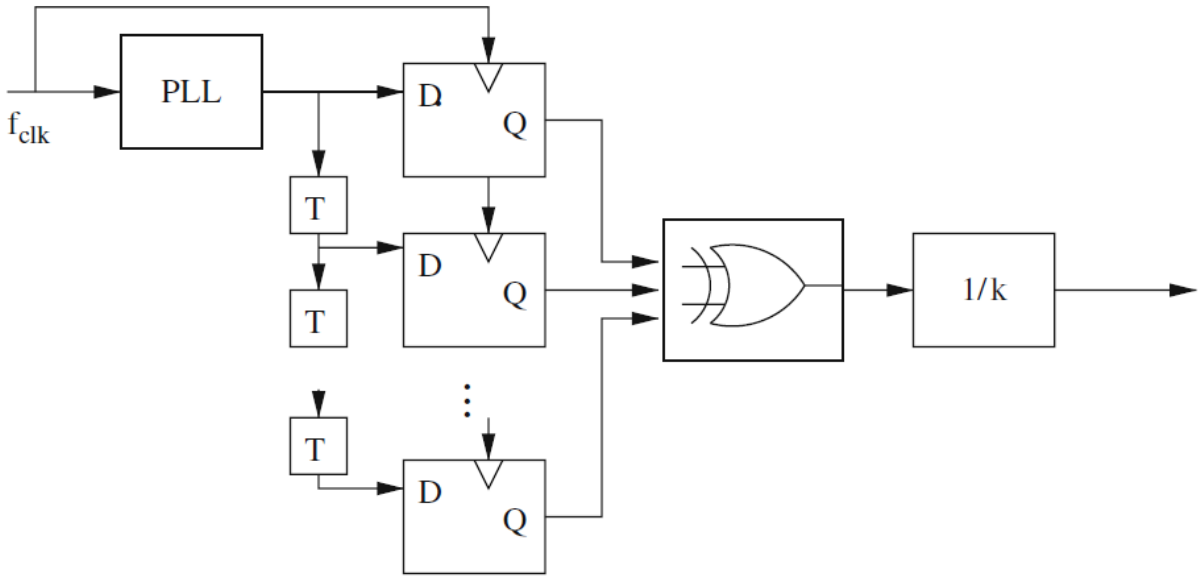

Рисунок 8 – Проект ФГСЧП Фишера-Друтаровски

Ключевой идеей является использование множественной выборки, чтобы иметь возможность выбирать ближнюю переходную зону, которая является флуктуирующей и которая, согласно [28], имеет порядок только несколько десятков пикосекунд. Множественные выборки, взятые через регулярные интервалы с последующей XOR операцией, дают выборку от формы волны, которая имеет желаемую неопределенность. В дальнейшем выход XOR операции подвергается понижающей дискретизации, используя прореживатель.

Авторы [28] дают довольно пространное резюме относительно фактической реализации проекта. Например, они отмечают, что ресурсы должны быть зафиксированы по месту в FPLD для получения желаемой конфигурации маршрутизации. Это особенно важно для задерживаемых модулей выборки. В приведенной в сообщении реализации битовая скорость составляла около 70 Кбит/с. Однако не ясно, был ли это верхний предел надежной операции, или это просто выбор проекта. Сгенерированная случайная битовая последовательность была проверена с использованием пакета статистических тестов NIST [12].

В целом проект Фишера-Друтаровски важен в том смысле, что показывает возможность реализации ФГСЧП на реконфигурируемой платформе. Авторы вводят новый каскадируемый задерживающий модуль выборки, а также обеспечивают математическую модель, которая позволяет им выбирать рабочие точки, чтобы увеличить вероятность выборки коллекции бит вблизи переходных зон.

Проект ФГСЧП Голича FIGARO. На рисунке 19 показан осциллятор Фибоначчи [29].

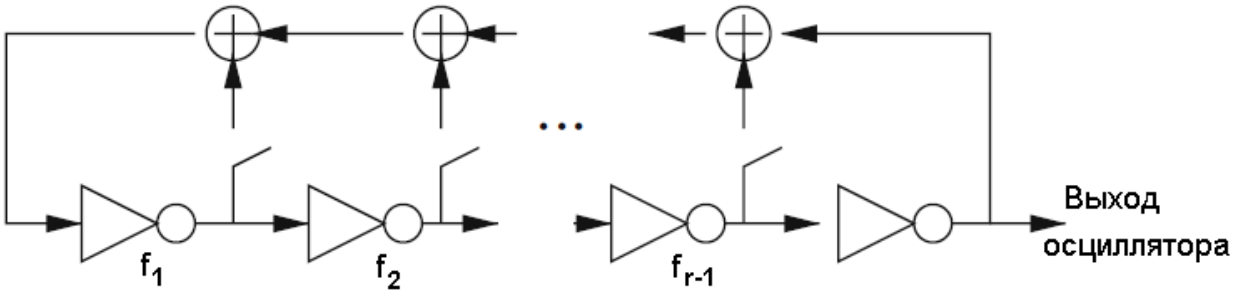

Рисунок 9 – Проект осциллятора Фибоначчи

В основном структура идентична LFSR, исключая элементы задержки, замещенные инверторами. Позиции обратной связи отмечены переключаемыми значениями f_i. Если $f_i = 1$, то переключатель закрыт, а иначе – открыт. Значения переключателей могут быть представлены более удобно в терминах полинома обратной связи, который задается следующим образом:

$$f(x) = \sum_{i=0} f_i x^i, \text{где } f_0 = f_r = 1.$$

Важно, что осциллятор не погружается в одиночное фиксированное состояние. Необходимые и достаточные условия задаются следующими двумя утверждениями [29]:

– кольцевой осциллятор Фибоначчи не имеет фиксированного состояния, если и только если $f(x) = (1+x)h(x)$ *u* $h(1) = 1$;

– кольцевой осциллятор Галуа не имеет фиксированного состояния, если и только если $f(1) = 1$ и r – нечетно.

Кроме того, для обоих видов осцилляторов, если $h(x)$ выбран как примитивный полином, то гарантированно имеем два цикла: короткий цикл на два состояния и длинный цикл, который включает оставшиеся $2^r - 2$ состояний.

Конфигурация Галуа кольцевого осциллятора показана на рисунке 10.

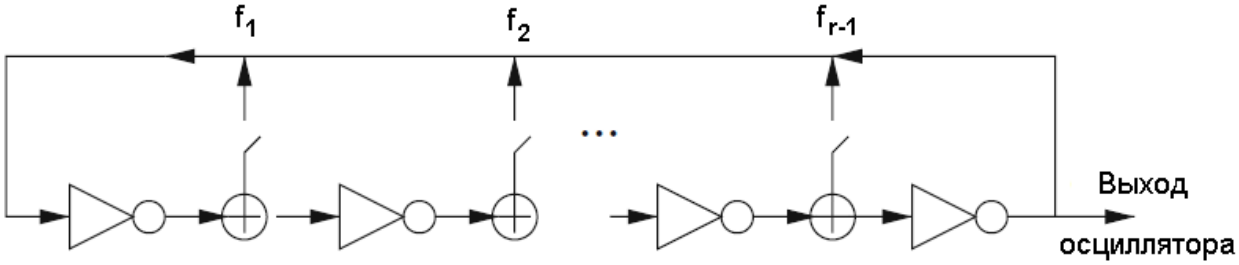

Рисунок 10 – Проект осциллятора Галуа

ФГСЧП проект FIGARO (FIbonacci-GAlois-Ring-Oscillator) – это просто XOR выход для Figaro осциллятора. Для оценки локальных корреляций и смещений автор также предложил использовать самоуправляемый LFSR для постобработки выхода. Была также проанализирована производительность реализации проекта.

Дихтл и Голич исследовали кольцевые осцилляторы Фибоначчи и Галуа в [30]. Их анализ первоначально основывался на методе рестарта. Через рестарт осцилляторов из некоторых исходных условий они измеряли время, которое требуется для наблюдаемого перехода из псевдослучайного битового потока в случайный. Это время используется для определения скорости выборки и производительности генератора. В некоторых отчетах авторы реализации кольца Фибоначчи для FPGA приводят данные, согласно которым достигнутая производительность составляет 6.25 Мб/с.

Основываясь на этих экспериментах, Дихтл и Голич заявляют о более высокой энтропийной скорости, чем та, которая свойственна традиционным кольцевым осцилляторам. Авторы также отмечают, что техника рестарта может быть использована как режим операции для генератора, и что подход на основе рестарта позволяет реализовать контролируемость. Другим важным результатом данной работы является новый двухуровневый модуль выборки, который уменьшает смещение, вносимое триггером выборки. Проект, с его малыми требованиями по объему схемной реализации, представляется идеальным решением для встраиваемых систем. Авторы приводят некоторые предварительные настроечные данные по увеличению эффективности.

Проект ФГСЧП Колбреннера-Гая. В проекте Колбреннера-Гая [31] используется флуктуация в кольцевых осцилляторах как источник шума. Эта разработка предназначена для CLB, соответствующих архитектуре Xilinx Virtex-II FPGA. Осциллятор, например, встроен в CLB. Сигнал осциллятора дважды проходит через CLB структуру и зеркально отражается только в одном из проходов (в LUT1), как показано на рисунке 11.

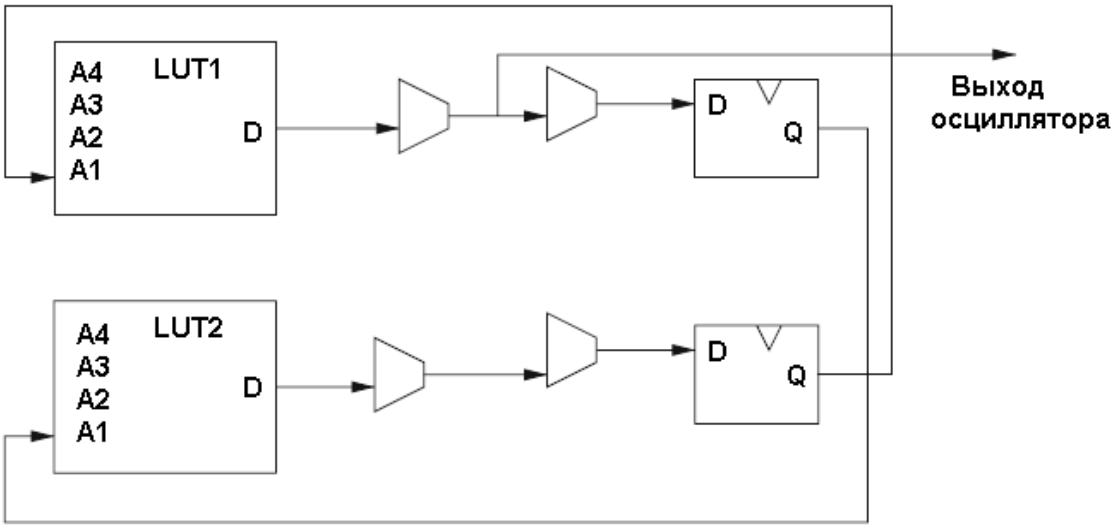

Рисунок 11 – Структура CLB осциллятора для проекта Колбреннера-Гая

Для ясности clk и reset сигналы не показаны на рисунке. Частота осциллятора определяется через задержку элементов на пути осциллятора, т.е., двумя поисковыми таблицами, четырьмя мультиплексорами и двумя ячейками памяти. Колбреннер отметил, что эта частная конфигурация дает достаточно устойчивый 130 МГц сигнал осциллятора. ФГСЧП производит выборку одного такого осциллятора с помощью другого.

Выход ФГСЧП подвергается постобработке с помощью простой последовательной XOR-схемы для ограничения смещений. Сообщенная битовая скорость имеет порядка нескольких сотен килобайт в секунду. Точная скорость зависит от эффективности XOR-схемы постобработки. Хотя скорость относительно низкая, проект является достаточно компактным, и его битовая скорость была приемлемой во многих приложениях. Выходная последовательность была проверена с использованием набора статистических тестов FIPS 140-1 [10].

Тестируемая ФГСЧП структура проекта Бусси-Луззи. Бусси и Луззи [29] отмечают, что бывает весьма трудно проверить качество выходов ФГСЧП после сложных методов постобработки. Авторы предлагают расширить проекты схемами сброса, которые очищают состояние ФГСЧП. Это делается для поддержки так называемого *сертификационного режима*, который устанавливает, является ли ФГСЧП заслуживающим доверия.

В сертификационном режиме ФГСЧП перестартовывает перед получением коллекции каждого выходного бита. Целью рестарта является устранение любой зависимости между собранными битами. Тогда выход ФГСЧП или зацикливается на фиксированном бите и генерируется нулевая энтропия, или генерируются независимые биты. Формирователь может быть проверен через схему, которая просто вычисляет переходы. Если скорость перехода является

приемлемой (ожидаемой), тогда смещения на выходе могут быть устранены постобработкой без запоминания. Постпроцессор без запоминания сохраняет независимость среди блоков вывода. На практике предложенный рестартовый подход приемлем к любому источнику энтропии, который разрешает рестарт. Ключевой особенностью является то, что вывод быстро уходит от стартового состояния в непредсказуемое состояние. Следовательно, время, необходимое энтропийному источнику для формирования расходящихся выводов после сброса, может быть использовано в качестве критерия оценки.

Проект колец. Данный проект, показанный на рисунке 12, был предложен Санером, Мартином и Стинсоном в [29]. В основе данного проекта лежит то, что выходы свободно запущенных кольцевых осцилляторов комбинируют вместе через XOR-операцию и затем фиксируются.

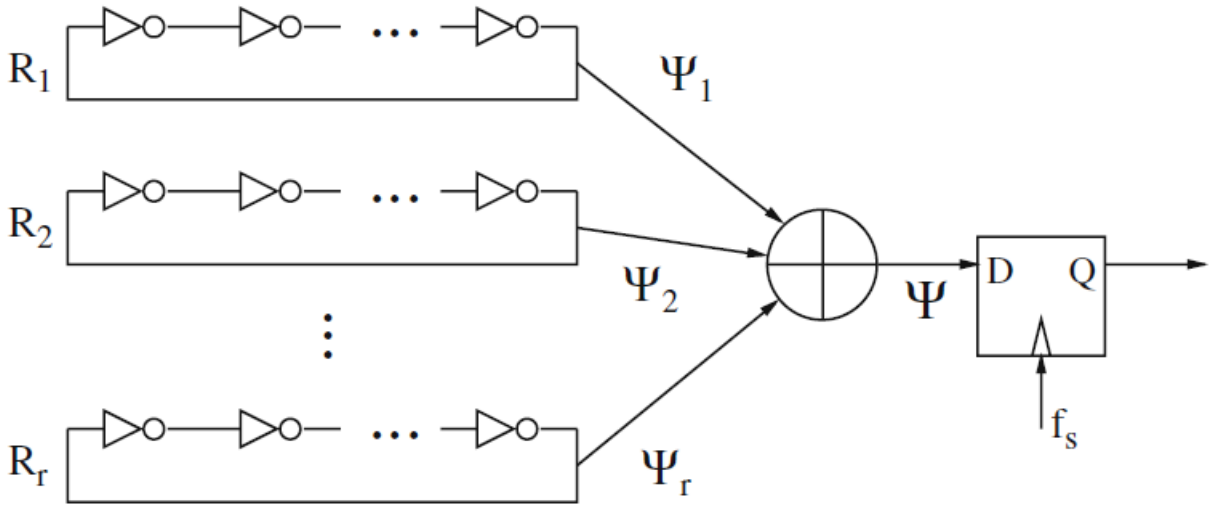

Рисунок 12 – Проект кольцевых осцилляторов

Главной идеей является вывод формы волны на переходные зоны с последующей случайной выборкой. Авторы дают математическую структуру и строгий анализ качества выхода ФГСЧП, основываясь на ряде предположений по входу. Кроме того, для уменьшения числа колец, авторы предлагают использовать эластичную функцию для последующей обработки ФГСЧП выхода. При удержании высокой степени эластичности функции, ФГСЧП можно разработать со значительным допуском против активных нападений. Проект колец имеет два главных вклада: структуру анализа и введенные эластичные функции для последующей обработки. С помощью анализа строится простая флуктуационная модель, а также вычисляется минимальное число колец, которые необходимо включить в проект для достижения определенной степени заполнения в окне выборки на некотором доверительном уровне.

Детерминированные биты, собранные из незаполненной части окна выборки, устраняются эластичной функцией соответствующей мощности.

Первоначальная реализация проекта колец была предложена Шеллекенсом и др. [30] на Xilinx Virtex-II FPGA. Реализация производит поток со скоростью 2.5 Мбит/с при частоте выборки 40 МГц, с использованием 110 колец, 13 инверторов и эластичной функции, сконструированной на основе линейного циклического кода (256, 16, 113). Выходная последовательность была проверена пакетами тестов Diehard [14] и NIST [12]. Шеллекенс и др. также отметили, что проект колец не содержит элементов запоминания и использует линейный метод постобработки без запоминания (эластичная функция сконструирована из линейного кода), и поэтому удовлетворяет критерию контролируемости, введенному ранее Бусси и Луззи [31].

Следует отметить, что проект колец получил критические замечания по ряду аспектов от Дихтла и Голича [30]. Среди замечаний – предположение независимости кольцевых осцилляторов и скорость выборки. В то время как скорость выборки может быть легко сокращена, то более сложно проверить независимость кольцевых осцилляторов, когда используется большое число колец. Для меньшего числа колец, тщательные компоновка и трассировка могут существенно изолировать кольца от взаимодействия одного с другим. Намного более простым решением является считывание только одной выборки на одном периоде осцилляции. В этом случае независимость колец не требуется. Это является удовлетворительным решением для проверки взаимной фазовой блокировки, которая уменьшает степень заполнения.

Проект PUF-RNG. Проект ФГСЧП, основанный на физически неклонируемых функциях (PUFs), был предложен О'Доннелом и др. в [29]. Проект строится в среде PUF схем, как показано на рисунке 13.

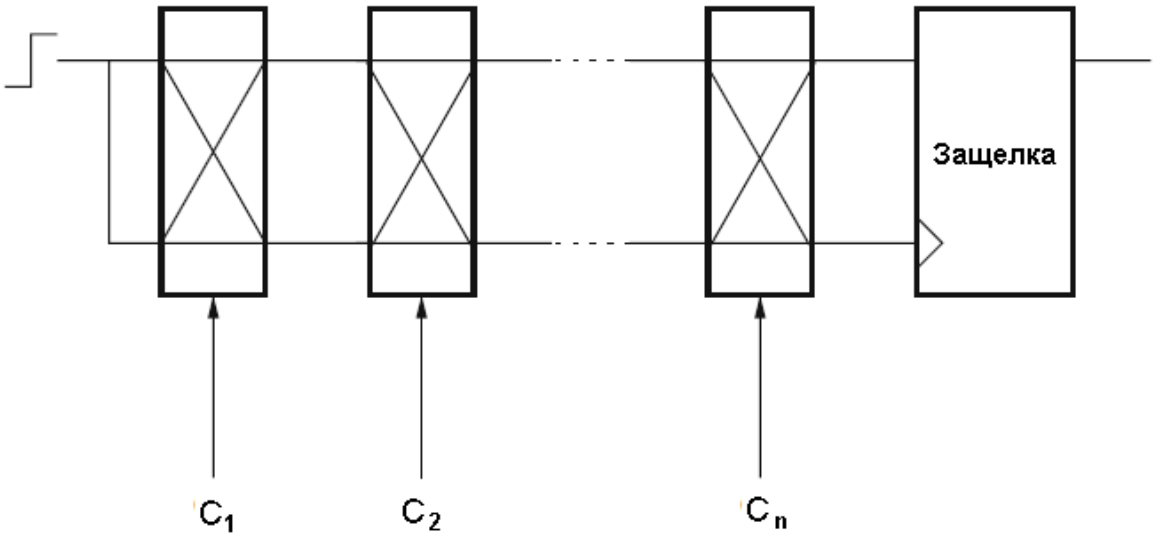

Рисунок 13 – Проект, основанный на PUF задержках

При нормальной операции выход PUF схемы определяется тонкими неточностями в путях задержки, созданных во время производственного процесса, наряду с приобретенными значениями на пути распространения. Альтернативно, для частного набора условий задержки будут сравнимыми, и схема выборки будет входить в метастабильное состояние. Следовательно, выход устройства будет непредсказуемым. В то время как это является хорошим обстоятельством, условие, которое вызывает метастабильность, является изменчивым из-за вариаций температуры и напряжения.

Следовательно, при проектировании PUF-RNG ищутся метастабильные условия, неоднократно применяя вызовы и повторения, при этом проверяется, получен ли достаточно нестабильный выход.

В общем, при том же самом начальном значении, поданным на PUF схему фиксированное число раз с фиксированным размером окна, имеется возможность получить почти равномерное распределение на PUF выходе в одном окне. Если это не достигается после проб фиксированного числа окон, генерируется новое начальное значение с помощью генератора псевдослучайных чисел. Когда метастабильное значение найдено, оно используется для генерирования выходной строки, которая в дальнейшем обрабатывается, используя корректор фон Неймана.

В отчетах по работе сообщается, что реализация, основанная на PUF, была интегрирована в безопасный процессор AEGIS [30]. Метод поиска начального значения метастабильности реализован в программном обеспечении. Выход ФГСЧП проверяется с использованием набора тестов NIST [12]. К сожалению, в отчетах не приводятся показатели производительности. Существенным преимуществом этого проекта является использование существующего PUF компонента. PUF схемы становятся популярным инструментарием для микросхем идентификации и для достижения сопротивляемости внешнему вмешательству. Поэтому весьма вероятно, что устройства защиты будут интегрированы с PUF модулями.

Проект Yoo и др. Практические аспекты проекта колец, эффекты IC трассировки, а также эффекты мощности питания и вариации температуры, были исследованы в [8]. Вначале авторы обращают внимание, что если сигнал предвыбран, то имеется шанс, особенно при низкой степени заполнения, что схема выборки зациклится на детерминированном участке выборочного окна. Поэтому авторы рекомендуют производить выборку на частоте, которая является взаимно простой с частотой осцилляции. Авторы отмечают, что эффекты IC уровня, такие как взаимная фазовая блокировка, незначительное отклонение сигнала в XOR дереве и воздействие некоторого ослабления сигнала будут ограничивать масштабируемость и производительность проекта RNG колец. Кроме того, в экспериментах, выполненных на FPGA реализации, было

показано, что при изменении температуры и питающего напряжения частота осцилляции может быть сдвинута на неприемлемую относительно основного условия величину. Следовательно, кольцевой ФГСЧП может быть уязвим по отношению к неагрессивным атакам вариаций температуры и питающего напряжения. Поэтому, чтобы сделать проект устойчивым против подобных атак, авторы предлагают использовать более чем одну линейку колец в проекте. Предложен проект из двух линеек колец. Проект прошел тесты Diehard [14] и NIST [12] и реализует производительность в 67 Мб/с. При этом потребление мощности составляет менее чем 300 мВт, а занимаемый объем на схемную реализацию - менее чем 1000 LUTs.

2.2 Комплекс генерации СЧП «Ключ-ВС»

Комплекс генерации случайных числовых последовательностей - это набор методов и алгоритмов, используемых для создания последовательности чисел, которые кажутся случайными. Генерация случайных чисел играет важную роль в различных областях, таких как криптография, статистика, моделирование случайных процессов, игры и многие другие.

В комплексе генерации случайных числовых последовательностей обычно включаются два основных компонента: источник случайности (энтропия) и генератор случайных чисел.

1) Источник случайности (энтропия) представляет собой источник, который поставляет некоторую неопределенность или случайность. Это может быть физическое явление, такое как термальный шум, атомный распад или временные различия в пользовательском вводе. Источник случайности предоставляет начальные данные для генератора псевдослучайных чисел.

2) Генератор псевдослучайных чисел (- это алгоритм, который использует входные данные от источника случайности и создает последовательность чисел, которые являются статистически случайными. ГСЧ обычно используется для создания псевдослучайных чисел, то есть чисел, которые выглядят случайными, но на самом деле получены с помощью детерминированного алгоритма. ГСЧ должен быть способен генерировать числа с высокой степенью случайности, обладать равномерным распределением и быть непредсказуемым.

Комплекс генерации случайных числовых последовательностей может включать в себя дополнительные компоненты, такие как алгоритмы шифрования, хэширования или статистические тесты для проверки качества генерируемых чисел. Это помогает обеспечить надежность и безопасность в различных приложениях, где требуется случайность.

На рисунке 14 показан «Ключ-ВС»

Рисунок 14 – «Ключ-ВС»

2.2.1 Структурная схема

ГСЧП «Ключ-ВС» выполнен в виде аппаратно-программного устройства, в состав которого входят следующие основные модули:

– схема обработки шумового сигнала;
– программируемая логическая интегральная схема (ПЛИС);
– USB-мост;
– блоки стабилизации и преобразования напряжения;
– датчик температуры;
– блок индикации.
–

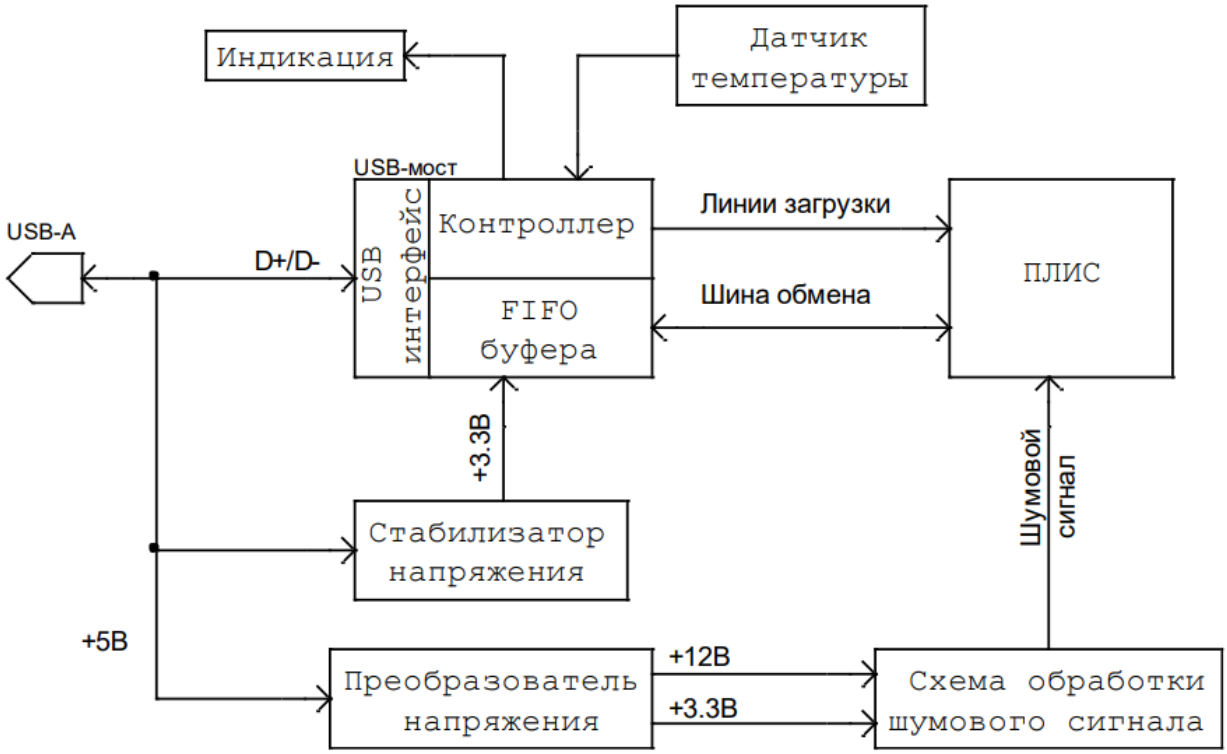

Рисунок 15 – Структурная схема

Схема обработки шумового сигнала обеспечивает получение аналогового шумового сигнала с физического источника шума (полупроводникового шумового диода) и его преобразование в цифровую форму.

ПЛИС выполняет обработку оцифрованного аналогового шумового сигнала, обеспечивая получение первичной СЧП. Она также обеспечивает функции интерфейса обмена с USB- мостом.

USB-мост обеспечивает реализацию внешнего интерфейса изделия с целевой ПЭВМ. Для обмена данными между ПЛИС и ПЭВМ используются FIFO-буфера моста USB. Кроме того, USB-мост включает контроллер, который обеспечивает выполнение функций управления режимами работы моста USB, а также загрузку в ПЛИС конфигурационных данных.

Модули стабилизации и преобразования напряжения обеспечивают электропитание ГСЧП «Ключ-ВС» от USB-порта целевой ПЭВМ, включая организацию всех режимов электропитания (плюс 3,3 В, плюс 5 В, плюс 12 В), необходимых для функционирования аппаратных и аппаратно-программных компонент устройства.

Датчик температуры обеспечивает измерение температуры и во взаимодействии с контроллером моста USB позволяет обеспечить контроль температурного режима эксплуатации ГСЧП «Ключ-ВС».

Блок индикации предназначен для визуального отображения основных режимов функционирования устройства в целях обеспечения удобства его эксплуатации. Его наличие позволяет оператору визуально определять состояние устройства (работоспособно, не функционирует, находится в состоянии ошибки или обмена данными и т.д.). На аппаратном уровне индикация обеспечивается применением светодиода.

ГЛАВА 3
РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Проектирование программного обеспечения для работы с АПК «Ключ-ВС»

3.1.1 Функциональные требования

Для разработки приложения для получения СЧП с физического генератора СЧП были сформулированы следующие требования:

- Приложение должно давать пользователю использовать несколько генераторов для генерации СЧП;
- Приложение должно давать возможность сохранить СЧП в файл регулируемого значения в выбранную пользователем директорию с определенным пользователем названием;
- Приложение должно предоставлять возможность тестирования сгенерированной СЧП;
- Приложение должно иметь сохраняемый журнал событий;
-Приложение должно иметь возможность тестирования генератора.

3.1.2 Нефункциональные требования

В результате формирования функциональных требований приложения были сформированы следующие нефункциональные требования:

- Для доступа к СЧП будем использовать библиотеки функций комплекса;
- Основой приложения станет язык программирования C++ и фреймворк QT;
- Разработка будет вестись под операционную систему Windows.

3.1.3 Варианты использования приложения

Диаграмма вариантов использования представлена на рисунке 16.

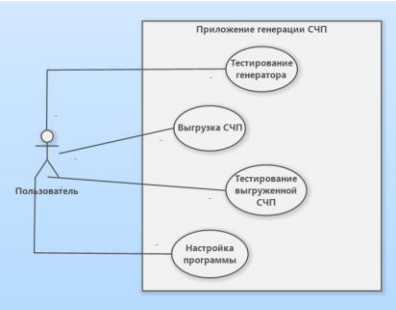

Рисунок 16 – Диаграмма использования

3.2 Архитектура ПО для работы с АПК «Ключ-ВС»

Приложение представляет из себя программу на C++.Архитектура приложения представляет из себя набор модулей для решения определенных задач.

3.2.1 Модуль Control

Модуль Control представляет собой набор классов, ответственных за реализацию различных методов контроля качества СЧП.

Класс BaseTest представляет собой класс для абстрактного базового теста, позволяющий установить новые параметры для теста или протестировать последовательность СЧП.

Класс ControlDialog предназначен для создания абстрактного диалога, для использования пакета тестов FIPS и NIST, т.к. у них есть своя библиотека

Класс ControlWidget имеет значение абстрактного виджета для работы с пакетами СЧП.

На рисунке 17 показаны связи с классом BaseTest

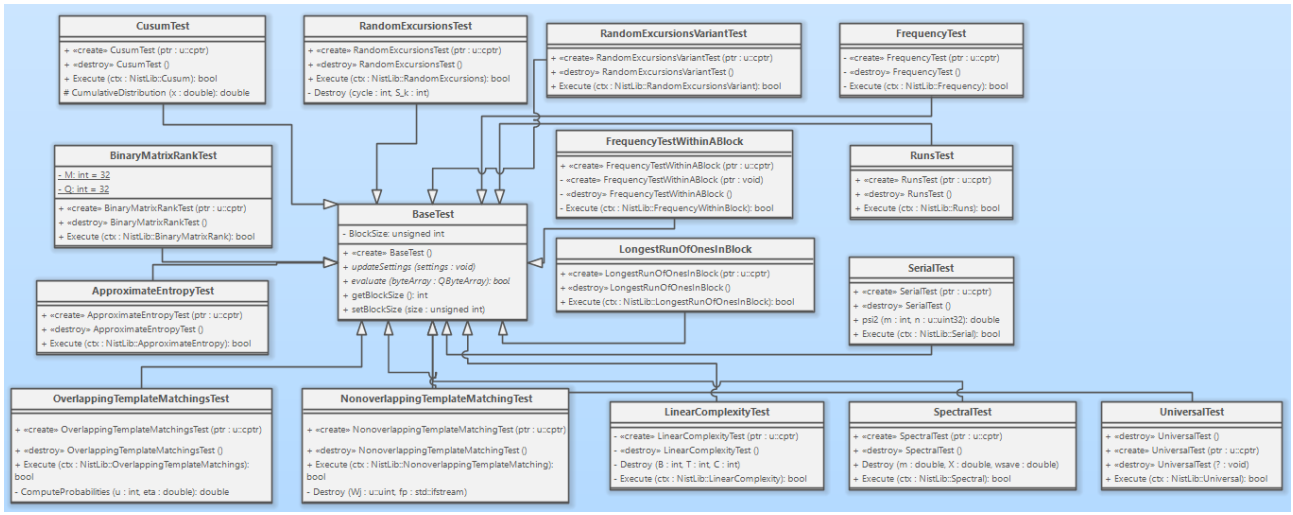

Рисунок 17 – Связи класса BaseTest

На рисунке 18 показаны связи класса ControlWidget

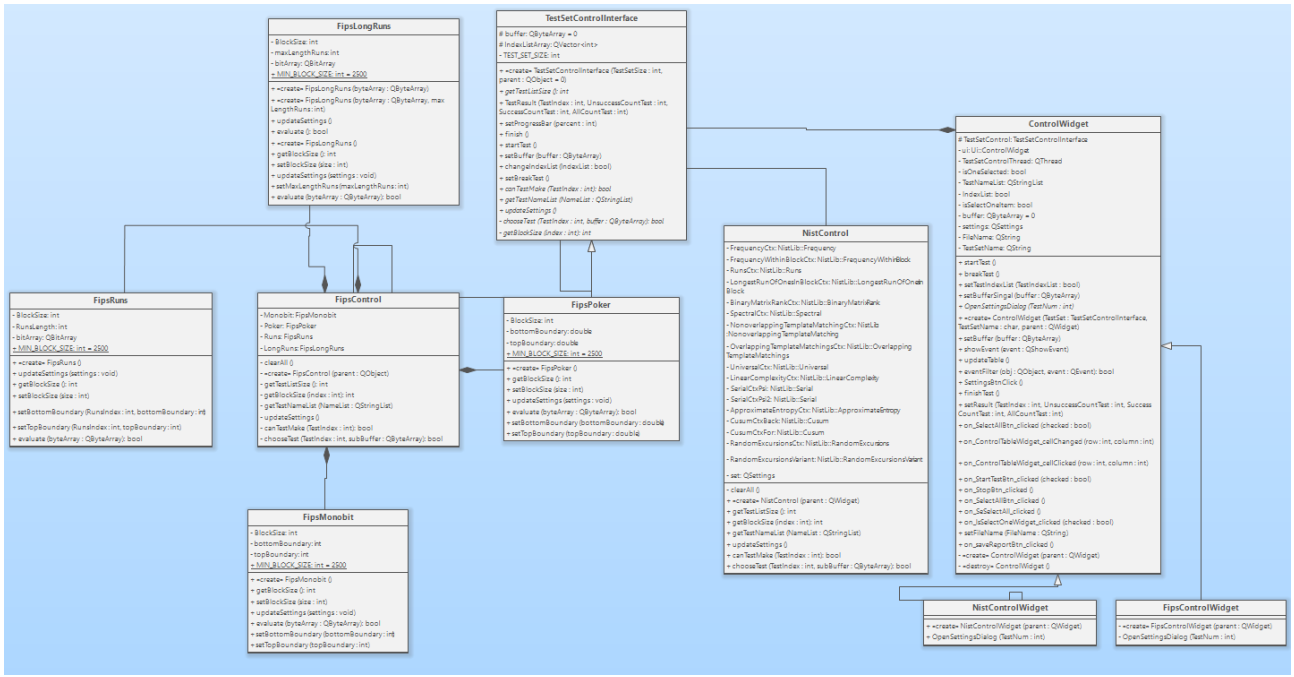

Рисунок 18 – Связи класса ControlWidget

На рисунке 19 показаны связи ControlDialog

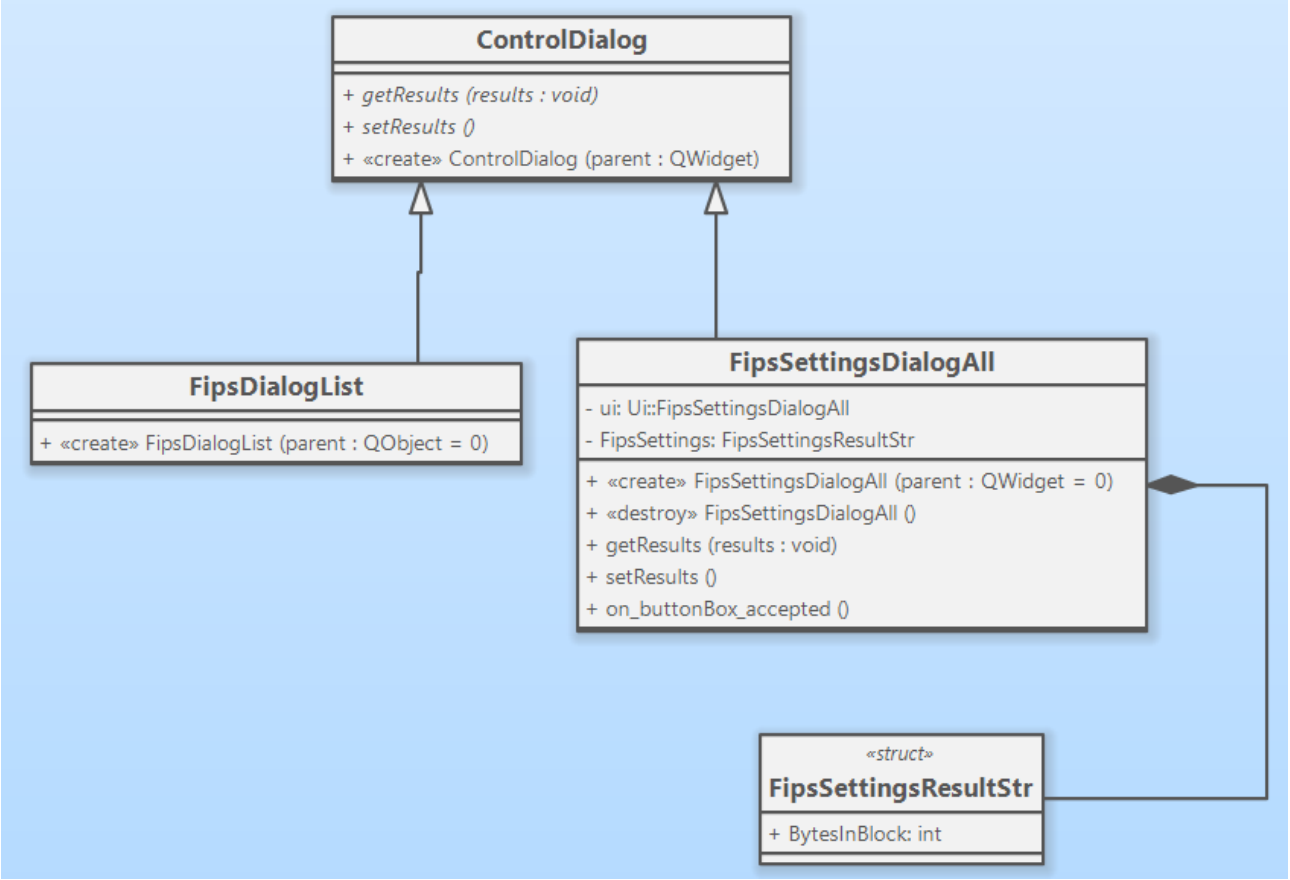

Рисунок 19 – Связи класса ControlDialog

3.2.2 Модуль Generators

Модуль Generators включает в себя классы отвечающие за работу с генераторами СЧП.

Класс GeneratorListWidgetэто класс отвечающий за работу с генераторами, и представляющий собой список доступных устройств и реализующий интерфейс взаимодействия с ними.

Класс TestModule является объектом-циклом для тестирования СЧП.

На рисунке 20 показаны связи TestModule

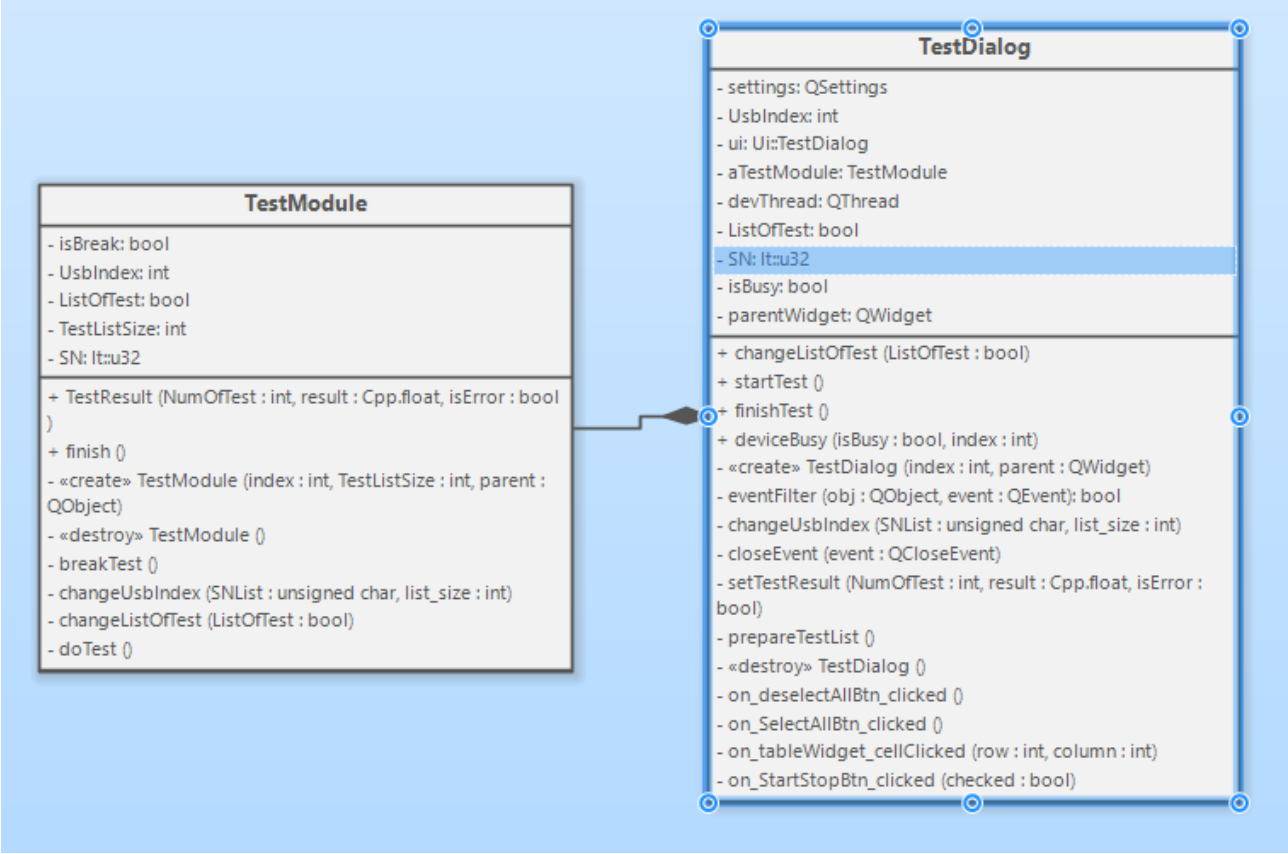

Рисунок 20 – Связи класса TestModule

3.2.3 Модуль Settings

Модуль Settings является ответственным за управление окном настроек приложения и содержит один класс, SettingsWidget, который представляет собой класс, предназначенный для работы с окном настроек.

SettingsWidget обладает функциональностью, необходимой для создания, отображения и взаимодействия с окном настроек приложения. Он предоставляет возможности для настройки различных параметров, которые могут быть изменены пользователем.

3.3 Интерфейс ППО для работы с АПК «Ключ-ВС»

В результате описанной выше архитектуры был разработан интерфейс, который состоит из следующих окон:

Окно работы с генераторами СЧП позволяет определить количество устройств генерации СЧП подключенных к компьютеру, начать процесс выгрузки СЧП заданного объема в файл и провести тестирование генератора.

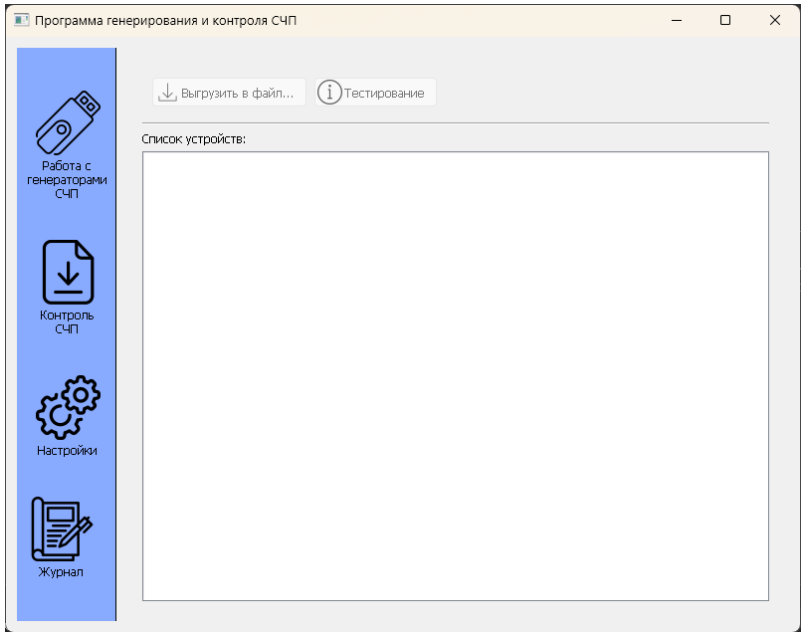

Рисунок 21 – Окно работы с генераторами СЧП

Окно контроля СЧП позволяет произвести выбор файла СЧП, и начать его проверку в соответствии с тестами FIPS и NIST, а также получить визуальную статистику равномерности распределения значений выборки.

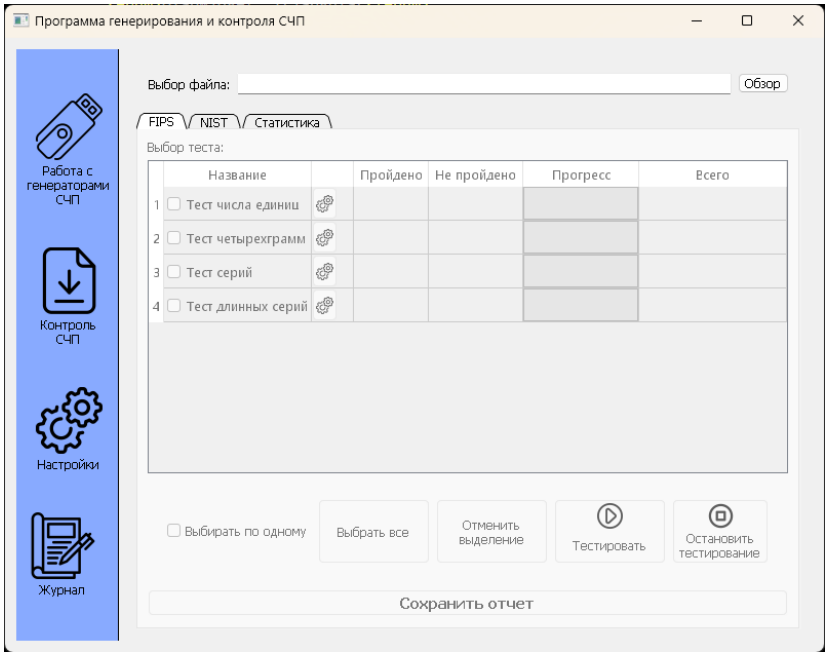

Рисунок 22 – Окно контроля СЧП

Окно настроек предоставляет возможности по изменению определенных настроек приложения, наподобие подтверждать ли выход из приложения, сохранять ли отчеты по статистике и тестам и т.п.

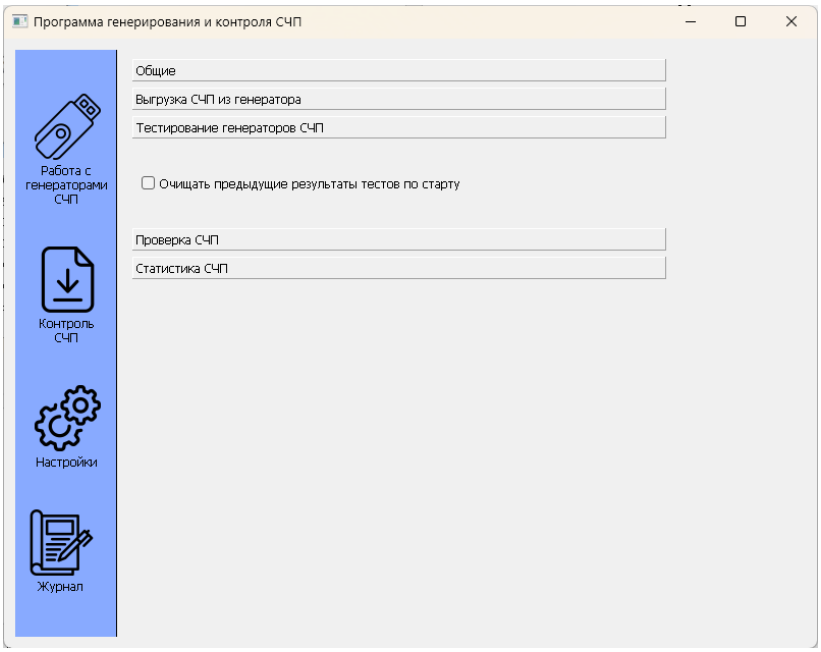

Рисунок 23 – Окно настроек

Окно журнала предоставляет возможность получить информацию о дате последней очистки журнала, возможных ошибках и предупреждениях.

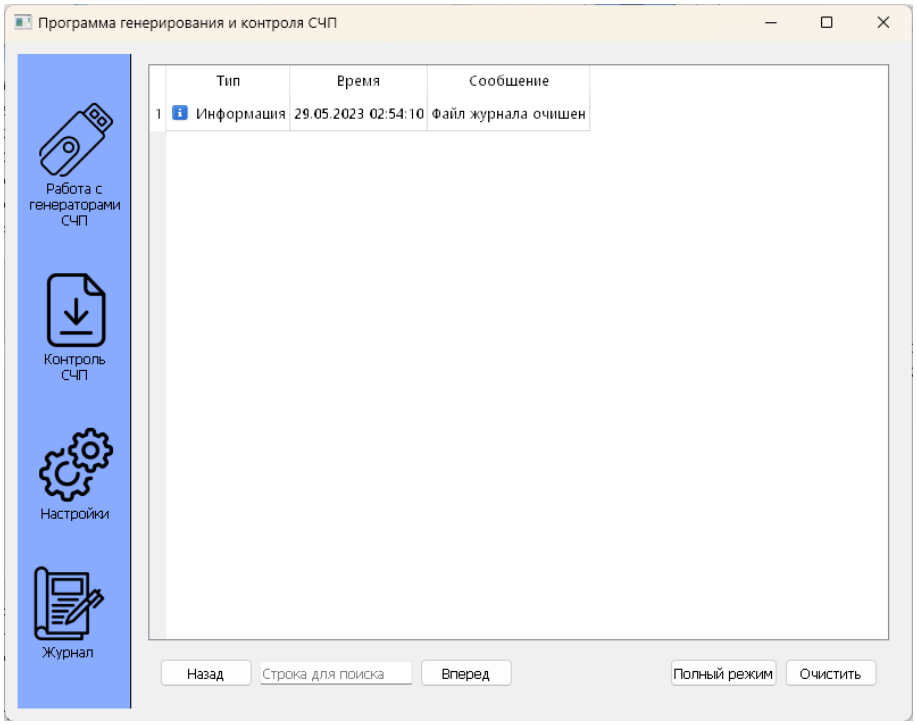

Рисунок 24 – Окно журнала

ЗАКЛЮЧЕНИЕ

В результате работы было произведено проектирование ППО, соответственно проектированию разработана архитектура, а также разработан интерфейс ППО. Разработанное программное обеспечение позволяет выполнять следующие функции:

- генерировать заданный объем СЧП с комплекса генерации СЧП «Ключ-ВС»;
- производить выбор генератора;
- производить тестирование качества сгенерированной СЧП в соответствии с методологиями FIPS и NIST;
- получать визуальное представление о равномерности выборки, в виде графика;
- сохранять отчеты о результатах пройденных выборкой тестов;
- вести журнал ошибок и предупреждений;
- настроить работу приложения для удобства пользователя.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Будько М.Б.,Будько М.Ю.,Гирик А.В.,Грозов В.А Методы генерации и тестирования случайных последовательностей. – СПб., Редакционно-издательский отдел университета ИТМО, 2019 – 73 с.
2. B. Barak, R. Shaltiel, and E. Tomer. True Random Number Generators Secure in a Changing Environment. In C. K. Koc. and C. Paar, editors, Workshop on Cryptographic Hardware and Embedded Systems—CHES 2003, pp. 166–180, Berlin, Germany, Lecture Notes in Computer Science, Vol. 2779 2003. Springer-Verlag, 2003.
3. B. Jun and P. Kocher. The Intel random number generator, White Paper Prepared for Intel Corporation, April 1999.
4. Букингем М. Шумы в электронных приборах. – М.: Изд. ин. лит., 1986. – 399 с.
5. Шеннон К. Работы по теории информации и кибернетике. – М.: Изд. ин. лит., 1963.
6. Королюк В.С., Портенко Н.И., Скороход А.В., Турбин А.Ф. Справочник по теории вероятностей и математической статистике. – М.: Наука. Главная редакция физико-математической литературы, 1985. – 640 с.
7. S.-K. Yoo, B. Sunar, D. Karakoyunlu, and B. Birand. Practical Aspects of the Rings Design, [Электронный ресурс] / Сайт http://ece.wpi.edu/ ~sunar/preprints/rings.pdf.
8. B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions, 26th IEEE Symposium on Foundations of Computer Science, pp. 396–407, 1985.
9. Кнут Д. Искусство программирования. Том 2. Получисленные алгоритмы. 3-е издание.: Пер. с англ. – М.; СПб.; К.: Изд. дом «Вильямс», 2005. – 832 с.
10. Security requirements for Cryptographic Modules. FIPS 140-1. – U.S. Department of Commerse. 1994.
11. Потий А.В., Орлова С.Ю., Гриненко Т.А. Статистическое тестирование генераторов случайных и псевдослучайных чисел с использованием набора статистических тестов NIST STS // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2001. – Вып. 2. – С. 206–214.
12. NIST SP 800-22. A Statistical Test Suite for Random and Pseudorandom Numbers, revision 1a.
13. NIST, [Электронный ресурс] / Сайт http://csrc.nist.gov/rng/.
14. G. Marsaglia. Diehard: A Battery of Tests of Randomness, [Электронный ресурс] / Сайт http://stat.fsu.edu/~geo, 1996.

15. [Электронный ресурс] / Сайт http://ru.wikipedia.org/wiki/Тесты_diehard/.

16. Crypt-X. [Электронный ресурс] / Сайт http://www.isi.qut.edu.au/ resources/cryptx/.

17. TestU01. A Software Library in ANSI C for Empirical Testing of Random Number Generators, [Электронный ресурс] / Сайт http://www.iro.umontreal.ca/ ~simardr/ testu01/tu01.html.

18. Koc C.K. Cryptographic Engineering. – Springer, 2009, 517p.

19. V. Bagini and M. Bucci. A Design of Reliable True RandomNumberGenerator for CryptographicApplications. In C. .K.Koc. and C. Paar, editors,Workshop on Cryptographic Hardware and Embedded Systems—CHES 1999, pp. 204–218, Berlin, Germany, Lecture Notes in Computer Science, Vol. 1717. Springer-Verlag, 1999.

20. T. E. Tkacik. A Hardware Random Number Generator In B. S. Kaliski Jr., C. K. Koc., C. Paar, editors, Workshop on Cryptographic Hardware and Embedded Systems—CHES 2002, pp. 450–453, Berlin, Germany, Lecture Notes in Computer Science, Vol. 2523. Springer-Verlag Berlin Heidelberg, 2003.

21. M. Dichtl. How to Predict the Output of a Hardware Random Number Generator, In C. D.Walter, C.K. Koc, C. Paar, editors, Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems – CHES 2003, Lecture Notes in Computer Science, Vol. 2779, pp. 181–188, Springer-Verlag Berlin Heidelberg, 2003.

22. W. Schindler. A Stochastical Model and Its Analysis for a Physical Random Number Generator In K. G. Paterson editor, Cryptography and Coding—IMA 2003, Springer, Lecture Notes in Computer Science, vol. 2898, 276–289, Berlin, 2003.

23. M. Epstein, L. Hars, R. Krasinski,M. Rosner and H. Zheng. Design and Implementation of a True Random Number Generator Based on Digital Circuit Artifacts. In C. D.Walter, C. K. Koc, C. Paar, editors, Workshop on Cryptographic Hardware and Embedded Systems—CHES 2003, Lecture Notes in Computer Science, Vol. 2779, pp. 152–165. Springer-Verlag Berlin Heidelberg, 2003.

24. V. Fischer and M. Drutarovsky. True Random Number Generator Embedded in Reconfigurable Hardware In B. S. Kaliski Jr., C. K. Koc, C. Paar, editors, Workshop on Cryptographic Hardware and Embedded Systems—CHES 2002, pp. 415–430, Berlin, Germany, Lecture Notes in Computer Science, Vol. 2523. Springer-Verlag Berlin Heidelberg, 2003.

25. J. Dj. Golic. New methods for digital generation and postprocessing of random data. IEEE Transactions on Computers 55(10): 1217–1229, 2006.

26. M. Dichtl and J. Dj. Golic. High-Speed True Random Number Generation with Logic Gates Only. Pascal Paillier, Ingrid verbauwhede, editors, Proceedings of the Cryptographic Hardware and Embedded Systems – CHES 2007, 9th

InternationalWorkshop, Vienna, Austria, September 10–13, 2007. Lecture Notes in Computer Science, vol. 4727, pp. 45-62, Springer Verlag, 2007.

27. P. Kohlbrenner and K. Gaj. An embedded true random number generator for FPGAs International Symposiumon Field Programmable Gate Arrays. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays, PP. 71–78, ACM Press, New York, NY, 2004.

28. B. Sunar, W. J. Martin, and D. R. Stinson. A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks, IEEE Transactions on Computers, vol. 58, no 1, p. 109–119, January 2007.

29. D. Schellekens, B. Preneel, and I. Verbauwhede. FPGA Vendor Agnostic True Random Number Generator. In Proceedings of the 16th International Conference on Field Programmable Logic and Applications. pp. 1–6, August, 2006.

30. M. Bucci and R. Luzzi. Design of Testable Random Bit Generators, In J. R. Rao and B. Sunar, editors, Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems – CHES 2005, Lecture Notes in Computer Science, Vol. 3659, pp. 131–146, Springer-Verlag Berlin Heidelberg, August 2005.

31. C. W. O'Donnell, G. E. Suh, and S. Devadas. PUF-Based Random Number Generation. Technical Report 481, MIT CSAIL, November 2004. [Электронный ресурс] / Сайт http://www.csg.csail.mit.edu/pubs/publications.html.

32. G. E. Suh, C. W. ODonnell, I. Sachdev, and S. Devadas. Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. Technical report, MIT CSAIL CSG Technical Memo 483, November 2004.