Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження генераторів випадкових послідовностей на мікроконтролерах

(тема)

Виконав:

студент II курсу, групи СПзм-20-1

Пушкар О.О.
(прізвище, ініціали)

Спеціальність

123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: проф. Торба А.А.

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет	навчально-науковий центр заочної форми навчання
Кафедра	електронних обчислювальних машин
Рівень освіти	другий (магістерський)
Спкціальність	123 – Комп'ютерна інженерія
Тип програми	освітньо-наукова
Освітня програма	системне програмування

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

"____" _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентці Пушкар Ользі Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження генераторів випадкових послідовностей на мікроконтролерах

затверджена наказом по університету від " 25 " березня 2022 р. № 33 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022

3. Вхідні дані до роботи – Використання мікроконтролерів для формування випадкових ключових даних,
– На основі аналізу сучасних датчиків шуму обрати джерело ентропії,
– Швидкість формування випадкових послідовностей – не менш 30 кбіт/с,
– Тестування випадкових послідовностей, що генеруються відомими тестами,
– Дослідження методів покращення статистичних параметрів випадкових
– Реалізація генератора випадкових послідовностей у вигляді макета для лабораторних досліджень

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області,
2. Вибір датчиків шуму – джерел невизначеності (джерел ентропії),
3. Розробка апаратної частини генератора випадкових послідовностей,
4. Розробка програмного забезпечення генератора випадкових послідовностей,
5. Тестування випадкових послідовностей, що генеруються,
6. Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Слайд-презентація – 17 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	**Назва етапів роботи**	**Термін виконання етапів роботи**	Примітка
1	Аналіз предметної області	30.03.2022-05.04.2022	
2	Розробка апаратного та програмного забеспечення	06.04.2022-16.04.2022	
3	Проведення експериментів	17.04.2022-01.05.2022	
4	Оформлення матеріалів кваліфікаційної роботи	23.04.2022-12.05.2022	
5	Подання кваліфікаційної роботи керівникові та	13.05.2022-14.05.2022	
	її попередній захист		
6	Подання кваліфікаційної роботи на рецензування	15.05.2022-15.05.2022	

Дата видачі завдання 28 березня 2022 р..

Студент _____
(підпис)

Керівник роботи _____ проф. Торба А.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 72 с., 18 рис., 5 табл., 1 дод., 9 джерел.

ГЕНЕРАТОР ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ, ДАТЧИКИ ШУМУ, АЛГОРИТМИ ВИРІВНЮВАННЯ ІМОВІРНОСТЕЙ, ТЕСТУВАННЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ, МІКРОКОНТРОЛЕР, ARDUINO.

Метою кваліфікаційної роботи є аналіз сучасних генераторів випадкових послідовностей та методів покращення їх статистичних параметрів, розробка апаратної та програмної частини лабораторного макету генератора випадкових послідовностей

У ході виконання кваліфікаційної роботи досліджувалися випадкові послідовності, що генеруються лабораторним макетом. Обґрунтовані методи покращення статистичних параметрів випадкових послідовностей та підвищення надійності генератора.

ABSTRACT

Master’s thesis: 72 pages, 18 figures, 5 tables, 1 appendices, 9 sources.

RANDOM SEQUENCE GENERATOR, NOISE SENSORS, PROBABILITY ALIGNMENT ALGORITHMS, RANDOM SEQUENCE TESTING, MICROCONTROLLER, ARDUINO.

The purpose of the certification work is to analyze modern random sequence generators and methods for improving their statistical parameters, develop the hardware and software part of the laboratory layout of the random sequence generator

During the certification work, random sequences generated by the laboratory layout were studied. Methods for improving the statistical parameters of random sequences and improving the reliability of the generator are substantiated.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EEPROM Electrically Erasable Programmable Read-Only Memory) – постійний запам’ятовуючий пристрій, що електрично стирається та перепрограмовується

ISP In System Programming – програмування в системі

MIPS Millions of Instructions Per Second, мільйон команд в секунду

RISC Reduced Instruction Set Computing – концепція обчислень зі скороченим набором команд

SRAM Static Random Access Memory – статична пам’ять з довільною вибіркою

SSD Solid-state drive – твердотільний накопичувач

UART Universal asynchronous receiver/transmitter – універсальний асинхронний приймач/передавач

USB Universal Serial Bus – універсальна послідовна шина

АГВП Апаратний генератор випадкових послідовностей

АЦП Аналого-цифровий перетворювач

ГВБ Генератор випадкових послідовностей

ЕОМ Електронна обчилювальна машина

КМОН Інтегральні мікросхеми за технологією компліментарний (додатковий) метал-оксид-напівпровідник

МК Мікроконтролер

РЗП Регістр загального призначення

ТТЛШ Інтегральні мікросхеми за технологією транзісторно-транзісторної логіки з діодами Шотки

ФЕП	Фотоелектронний ПОМНОЖУВАЧ
ЦАП	Цифро-аналоговий перетворювач

ВСТУП

Граничні характеристики стійкості криптографічних систем захисту інформації досягаються у разі, якщо для формування ключів, параметрів і синхромаркерів використовується генератор випадкових послідовностей на основі фізичних датчиків шуму із статистично обґрунтованими параметрами рівноймовірності, незалежності і некорельованості [1,2,3,4].

При реалізації державних і комерційних криптографічних систем необхідною умовою являється застосування вітчизняних розробок, що виключають наявність програмних і апаратних "закладок" і, як наслідок, не допускають маніпуляцію або злом систем захисту інформації.

Аналіз вітчизняних і закордонних криптографічних систем і, зокрема, систем генерації випадкових послідовностей вказує напрями досліджень для підвищення швидкодії, надійності, скритності, поліпшення статистичних параметрів і економічних показників вітчизняних засобів криптографії.

Опублікований в 2005-му році Міжнародний стандарт ISO/IEC 18031 : 2005 – Information technology – Security techniques – Random bit generation (Інформаційні технології – Методи захисту – Генерація випадкових бітів) [1] узагальнює величезний міжнародний досвід практичних і теоретичних досліджень попередніх років і встановлює спеціальні вимоги, яких необхідно дотримуватися при розробці генераторів випадкових біт, які використовуватимуться для криптографічних застосувань.

Сучасна елементна база обчислювальних систем на основі мікроконтролерів дозволяє реалізовувати генерацію та тестування випадкових послідовностей на одному кристалі з крипто шифраторами та крипто дешифраторами, разом з алгоритмами обмеження доступу до апаратних та програмних засобів.

Мета, основні завдання і напрями досліджень, відбиті в кваліфікаційній роботі.

Мета:

визначити ефективні механізми поліпшення статистичних параметрів випадкових послідовностей, що генеруються, та підвищення надійності апаратних засобів генерації випадкових послідовностей.

Завдання:

– аналіз існуючих систем генерації випадкових і псевдовипадкових послідовностей для криптографічних систем,

– вибір та обґрунтування джерел ентропії (датчиків шуму, джерел невизначеності),

– розробка апаратної та програмної складової недетермінованих генераторів випадкових послідовностей на елементній базі сучасних мікроконтролерів,

– дослідження недетермінованих генераторів випадкових послідовностей з метою досягнення максимальної швидкості формування випадкових бітових послідовностей.

1 АНАЛІЗ АПАРАТНИХ ГЕНЕРАТОРІВ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Захист багатьох криптографічних систем залежить від генерування непрогнозованих параметрів. Прикладами цього можуть бути [2,3,4]:

– випадкові ключі і значення ініціалізації для шифрування;

– випадкові ключі для MAC алгоритмів;

– випадкові особисті ключі для алгоритмів цифрового підпису;

– випадкові значення, що використовуються в механізмах автентифікації об’єктів;

– випадкові значення, що використовуються в протоколах встановлення ключів;

– генерація випадкових PIN і паролів;

– випадковий початковий стан.

У усіх цих випадках непередбачувані параметри (числа, послідовності), які генеруються, мають бути достатнього розміру і бути "випадковими" в тому сенсі, що імовірність будь-якого вибираного окремого значення має бути досить малою для відвертання досягнення супротивником переваг за допомогою оптимізації стратегії пошуку на базі такої імовірності.

Генератор випадкових бітів – це пристрій або алгоритм, який виводить послідовність статистично незалежних і безпомилкових двійкових розрядів [1,2]. Ідеальним генератором випадкових бітів можна вважати підкидання монети і реєстрацію випадання "орла" – одиниці, або "решки" – нуля [1].

Застосовують також узагальнене поняття – генератор випадкових послідовностей. Цим терміном можливо називати пристрій або алгоритм генерації випадкових бітів, з яких складаються випадкові числа.

Псевдовипадкові послідовності генеруються на основі відомого (детермінованого) алгоритму – апаратно або програмно. Тому псевдовипадкові послідовності можливо повторити в тому ж порядку в подальших дослідах, знаючи алгоритм і початкові значення, можливо

передбачити (прорахувати) наперед будь-яку кількість псевдовипадкових бітів. Псевдовипадкові послідовності мають період «*T*», тобто генерована послідовність після «*T*» бітів починає повторюватися. У сучасних генераторах псевдовипадкових послідовностей, що складаються з 100-розрядного регістра і працюючих на частоті 10 Мгц або більше, період повторення в мільйони разів перевищує вік Всесвіту. Тому в сучасних генераторах псевдовипадкових послідовностей поняття періоду не має практичного сенсу [2].

Генератор випадкових послідовностей обов'язково включає джерело випадковості (чи джерело невизначеності) – фізичний датчик шуму [1,2].

Проектування апаратного пристрою або програми для використання такої випадковості і отримання бітової послідовності, що не має помилок і кореляцій, є важким завданням. Крім того, для більшості криптографічних застосувань, генератор не повинен піддаватися спостереженню або маніпуляції супротивником.

1.1 Вибір і обґрунтування фізичних датчиків шуму

Прості фізичні датчики випадковості, реалізовані на основі випадкових механічних переміщень: підкидання монети, кидання «гральних кісток», оптоелектронні системи спостереження броунівського руху та ін., мають недостатню швидкодію і вимагають для своєї реалізації оптичні облаштування введення результатів дослідів в ЕОМ.

Прикладом вдалої реалізації устаткування введення в ЕОМ випадкових механічних переміщень є зчитування станів швидкодіючого детермінованого лічильника у випадкові моменти натиснення клавіш. Обов'язковою умовою для генерації випадкових рівноімовірних послідовностей є багатократне переповнювання лічильника між зчитуваннями (натисненнями клавіш).

Відомий приклад генерації випадкових чисел при зчитуванні станів лічильника в таймері IBM PC (кількість переповнювань таймера - більше 18

разів в секунду) в моменти натиснень довільних клавіш (швидкість натиснення клавіш – не більше 2 - 3 в секунду). Для набору випадкового двійкового числа завдовжки 512 біт необхідно натиснути 32 клавіші. Це займає багато часу (до однієї хвилини), проте такий метод може бути реалізований на програмному рівні і не вимагає додаткового устаткування [2].

У сучасних криптографічних системах швидкість генерації випадкових послідовностей перевищує 1 Мбіт/с. Такі параметри можуть бути реалізовані тільки на основі електронних датчиків шуму з широким спектром частот.

Випадкові зміни параметрів (так званий, тепловий шум) спостерігаються в усіх електронних компонентів при температурах вище за абсолютний нуль по Кельвіну. Тому в якості фізичних датчиків шуму можуть бути використані будь-які електронні компоненти [2].

Порівняльний аналіз електронних датчиків шуму робиться за основними показниками:

- смуга частот випадкового сигналу;
- вихідна напруга (амплітуда шуму);
- споживана потужність;
- напруга живлення;
- масо-габаритні параметри;
- надійність роботи при зміні умов експлуатації;
- економічні показники (вартість) [2,4].

Датчики шуму на основі РЕЗИСТОРІВ генерують випадковий сигнал в смузі частот від одиниць Гц до сотень Мгц з амплітудами вихідної напруги менше 1 мВ. Тому для сполучення з цифровими пристроями необхідно застосовувати підсилювачі з коефіцієнтом посилення по напрузі в декілька тисяч разів.

Відомі обмеження по «площі посилення» (тобто твору коефіцієнта посилення на смугу посилюваних частот) призводять до того, що вихідний сигнал підсилювача має обмежену смугу частот – не більше за одиниці Мгц [2,3].

Споживана потужність такого датчика визначається, в основному, потужністю підсилювача і складає від десятків до сотень мВт при напрузі живлення від 5 до 15 В. Можливість реалізації резистивного датчика шуму, підсилювача і аналого-цифрового перетворювача на одному кристалі в єдиному технологічному процесі виготовлення інтегральних схем визначає малі масо-габаритні параметри і невеликі економічні витрати на виготовлення.

Надійність роботи при зміні умов експлуатації визначається, в основному, не резистивним датчиком, а електронною схемою. Враховуючи сферу застосування датчиків шуму у складі обчислювальних систем, умови експлуатації останніх визначають допустимі зміни кліматичних умов і є прийнятними для електронної схеми датчика.

Датчики шуму на основі p-n- ПЕРЕХОДІВ (діоди, транзистори) генерують випадковий сигнал (так званий, шум рекомбінації) в діапазоні частот від одиниць Гц до сотень Мгц з амплітудою в декілька мкВ [2,3]. Обов'язкове застосування підсилювачів вносить в сигнал додаткові шуми самих підсилювачів, які виконані на транзисторах і мають аналогічну природу.

Амплітуди спектральних складових шуму убувають із зростанням частоти, тому застосування таких датчиків на частотах вище за один Мгц недоцільно. Споживана потужність, напруга живлення, масо-габаритні і економічні показники визначаються параметрами підсилювачів і за цими показниками аналогічні датчикам на основі резисторів.

Датчики шуму на основі КРЕМНІЄВИХ ДІОДІВ Із ЗЕНЕРІВСЬКИМ ПРОБОЄМ (СТАБІЛІТРОНИ) генерують випадковий сигнал з рівномірним спектром від одиниць Гц до десятків Мгц і амплітудами в десятки або сотні мВ. Розроблені і випускаються великими серіями спеціалізовані шумові діоди із Зенерівським пробоєм (наприклад, КГ401), які при постійній напрузі 8 - 9 В і струмі від 50 до 100 мкА генерують широкий спектр (до десятків Мгц) випадкових імпульсів з амплітудами від 0,1 до 1 В [2,3,4].

Зенерівський пробій базо-емітерного переходу кремнієвих високочастотних транзисторів також можливо використовувати як датчик шуму.

Споживана потужність у таких датчиків шуму складає одиниці мВт при напрузі живлення від 10 до 20 В. Це дозволяє легко вбудовувати їх в обчислювальні системи. Масо-габаритні і економічні показники датчиків шуму на основі шумових діодів із Зенерівським пробоєм (при реалізації в інтегральному виконанні на одному кристалі з обчислювальною системою) – найкращі з усіх даних датчиків.

Датчики шуму на основі ЕЛЕКТРОННИХ ЛАМП (діоди, тріоди і так далі) генерують випадковий сигнал (так званий, дробовий шум) з рівномірним спектром від одиниць Гц до десятків Мгц і амплітудами менше 1 мВ. Тому обов'язковим є застосування підсилювачів на лампах або транзисторах.

Споживана потужність (з урахуванням потужності розжарення катоду) складає одиниці Вт. Живляча напруга – від 50 до 200 В. Масо-габаритні показники значно перевищують аналогічні показники сучасних інтегральних обчислювальних систем. Необхідно також враховувати зміну параметрів в процесі експлуатації електронних ламп за рахунок погіршення емісійних властивостей катодів. Тому застосування таких датчиків шуму в сучасних обчислювальних системах – недоцільно.

Датчики шуму на основі ГАЗОРОЗРЯДНИХ ЛАМП (тиратрони, стабілітрони, неонові лампи та ін.) генерують випадковий сигнал в смузі частот від одиниць Гц до сотень кГц з амплітудами до 1 В. Напруга живлення – від 70 до 200 В, споживана потужність – від декількох мВт до одного Вата. Враховуючи великі масо-габаритні показники (як у лампових датчиків) і високу напругу живлення, застосування газорозрядних ламп в сучасних обчислювальних системах – недоцільно [3].

ФОТОЕЛЕКТРОННІ ПОМНОЖУВАЧІ (ФЕП), які працюють в одноелектроному режимі, формують випадкові імпульси амплітудою менше 1

мА з частотою, яка визначається як тепловими процесами на поверхні фотокатода, так і рівнем зовнішнього засвічення. Це дозволяє легко регулювати середню частоту випадкових імпульсів в інтервалі від одиниць кГц до десятків Мгц зміною оптичного сигналу підсвічування фотокатода. Необхідно також враховувати залежність формованого сигналу від живлячої напруги і температури довкілля [2,3].

Напруга живлення – від 500 В до 3 кВ, споживана потужність – одиниці Вт. Великі масо-габаритні показники (як у лампових датчиків) і висока живляча напруга, визначають недоцільність застосування ФЕП в якості датчиків шуму.

Серед екзотичних датчиків шуму можливо відмітити застосування вимірників радіоактивного випромінювання (лічильник Гейгера та ін.) спільно з радіоактивними ізотопами. Використання таких датчиків в сучасних обчислювальних системах обмежене з міркувань екології.

На основі аналізу сучасних електронних датчиків шуму можливо говорити про доцільність застосування в сучасних обчислювальних системах фізичних датчиків шуму на основі кремнієвих діодів із Зенерівським пробоєм.

1.2 Модель генератора випадкових послідовностей

Властивості випадковості можуть бути продемонстровані на прикладі підкидання монети і спостереження, яка сторона опиниться зверху після її падіння, де одна сторона називається «орлом» (двійкова одиниця – 1), а протилежна – «решкою» (двійковий нуль – 0). Монета також має обідок, але імовірність падіння монети на свій обідок настільки маловірогідна, що в цілях цієї демонстрації її можливо ігнорувати.

Необхідні властивості випадковості можливо досліджувати, використовуючи описаний вище приклад підкидання монети. Результатом кожного підкидання монети є:

– непередбачуваність: перед підкиданням невідомо, якою стороною приземлиться монета, «орлом» чи «решкою». Також, якщо підкидання монети тримається в таємниці, то не можливо визначити результат підкидання, навіть при умові, якщо відомі наступні результати підкидання. Непередбачуваність після підкидання залежить від можливості спостереження спостерігачем за підкиданням монети. Поняття ентропії визначає кількість непередбачуваності або невпевненості спостерігача;

– незміщеність: кожен потенційний результат має одну і ту ж можливість (імовірність) появи;

– незалежність: підкидання монети вважається некорельованим без пам’яті або без хронології; що б не трапилося перед процесом підкидання монети, це не вплине на процес.

Якщо генератор випадкових бітів правильно реалізований і правильно працює, то не можливо передбачити вихідні дані. Неможливість передбачення майбутніх вихідних даних визначається як пряма секретність. Неможливість визначення попередніх вихідних даних генератора випадкових послідовностей (ГВП) при відомих існуючих або будь-яких майбутніх вихідних даних ГВП визначається як зворотна секретність.

Для генерації випадкових послідовностей достатнє внесення одного якого-небудь випадкового (непередбачуваного) параметра в детермінований процес (чи алгоритм).

Фізичні датчики шуму на основі діодів із Зенерівським пробоєм типу КГ401 при середній постійній напрузі 8 - 9 В і струмі 50...100 мкА формують випадкові імпульси амплітудою 0,1...1 В з максимальною середньою частотою до 3 - 8 Мгц (графік *Uшд* на рисунку 1.1 а) (середня частота вимірюється електронно-рахунковим цифровим частотоміром) [2,3].

Перетворення цих імпульсів в логічні рівні цифрових мікросхем (графік *Uтш* на рисунку 1.1 б) реалізується підсилювачем-обмежувачем (компаратором) з невеликим гістерезисом на вході – тригером Шмітта (TS) [2].

У вихідному сигналі тригера Шмітта (графік *Uтш* на рисунку 1.1 б) переважає рівень логічного нуля. При зчитуванні цього сигналу в довільні моменти часу випадкова послідовність, яка формується, міститиме значно більше нульових бітів, чим одиничних.

Для вирівнювання імовірності «0» і «1» вихідний сигнал тригера Шмітта подається на лічильний тригер (рисунок 1.2)

Вихідний сигнал лічильного тригера (графік *Uлт* на рисуноку 1.1 в) з рівною імовірністю приймає значення «логічного нуля» і «логічної одиниці» в довільні моменти часу. Зчитування випадкових бітів можливо робити в детерміновані моменти часу.

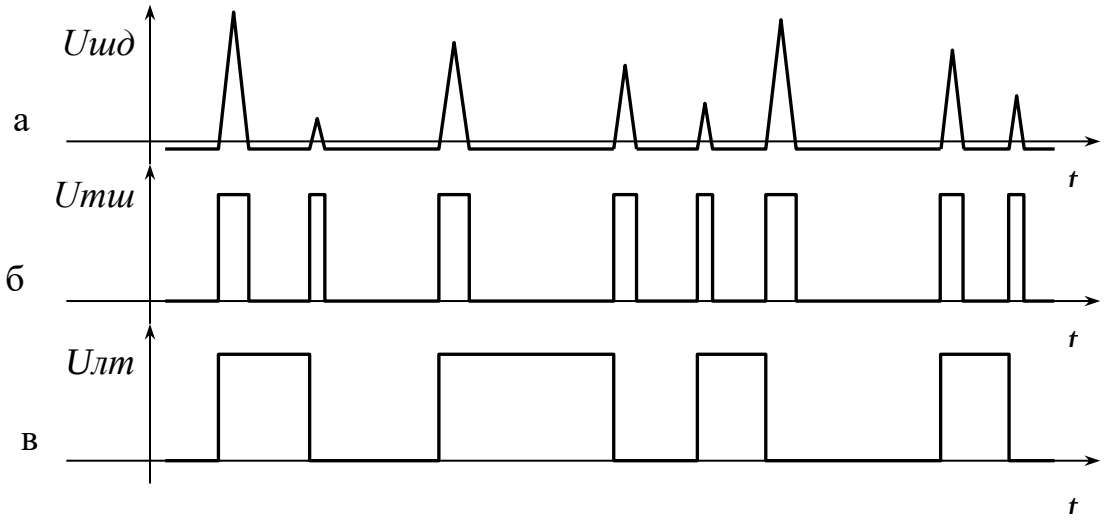

Рисунок 1.1 – Часові діаграми формувача випадкових бітів

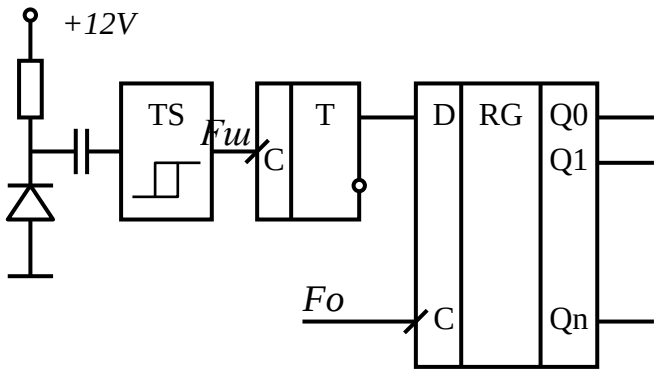

Рисунок 1.2 – Генератор випадкових послідовностей

Обов'язковою умовою незалежності елементів згенерованих випадкових послідовностей є багатократне спрацьовування лічильного тригера впродовж інтервалу часу між зчитуваннями [2].

Стани лічильного тригера зчитуються з частотою *Fo* в регістр зсуву RG (рисунок 1.2). Частота *Fo* вибирається в 5...10 разів менше, ніж середня частота шумових імпульсів *Fш* на виході тригера Шмітта. Це необхідно для багатократного спрацьовування лічильного тригера між сусідніми зчитуваннями випадкових бітів. При цьому виключається взаємний вплив імовірності появи чергового біта від стану попереднього біта.

Одноканальна схема формування випадкових бітів, що включає шумовий діод, підсилювач-обмежувач (тригер Шмітта) і лічильний тригер (рисунок 1.2), не забезпечує необхідну експлуатаційну надійність генерації рівноімовірних бітів у разі зміни параметрів джерела шуму або підсилювача-обмежувача на основі тригера Шмітта.

Підвищення експлуатаційної надійності каналу формування випадкових бітів досягається гарячим резервуванням, тобто паралельною роботою декількох каналів. На рисунку 1.3 приведена схема генератора випадкових послідовностей з двома каналами формування випадкових бітів (перший канал виділений на рисунку 1.3 пунктиром). Можливе застосування трьох і

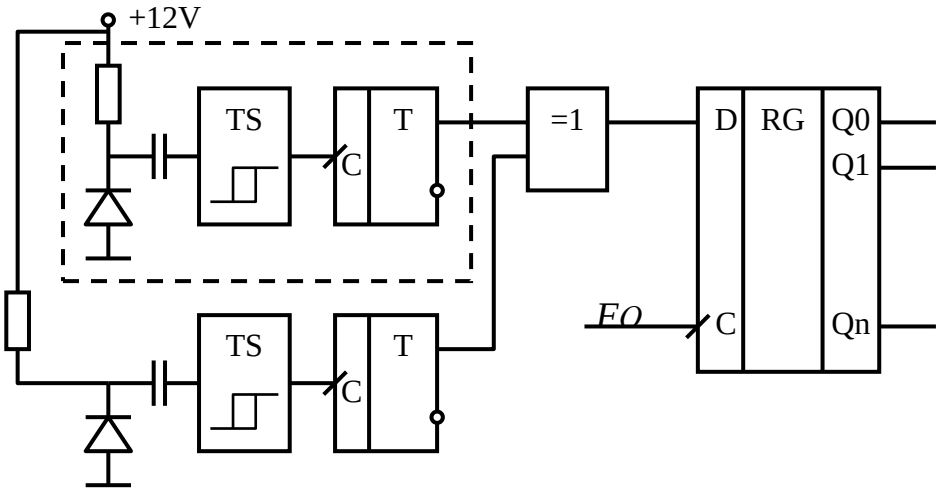

Рисунок 1.3 – Генератор випадкових послідовностей з гарячим резервуванням каналів формування випадкових бітів

більше аналогічних каналів формування випадкових бітів [2].

Вихідні рівноімовірні випадкові логічні сигнали усіх каналів об'єднуються елементом «ВИКЛЮЧНЕ АБО» (схемою «додавання за модулем 2» схемою «XOR») і зчитуються в регістр зсуву з частотою *Fo* (рисунок 1.3).

Запропонований метод гарячого резервування генераторів випадкових бітових послідовностей, тобто введення декількох каналів генерації випадкових бітів (див. рисунок 1.3), дозволяє також пропорційно збільшити еквівалентну частоту шумового сигналу *Fш* на виході елементу «ВИКЛЮЧНЕ АБО». Це підтверджується вимірами частот випадкових сигналів на виходах лічильних тригерів в першому і другому каналах з частотою випадкового сигналу на виході елементу «ВИКЛЮЧНЕ АБО».

Порівняйте частоти випадкових сигналів на виходах лічильних тригерів в першому і другому каналах (осцилограми *в* і *д* на рисунку 1.4) з частотою випадкового сигналу на виході елементу «ВИКЛЮЧНЕ АБО» (осцилограма *е* на рисунку 1.4).

Експериментально перевірялося, що введення чотирьох каналів генераторів шуму збільшує середню частоту *Fш* на виході елементу «ВИКЛЮЧНЕ АБО» (рисунок 1.4.е) приблизно в 4 рази (середня частота *Fш* вимірювалася електронно-рахунковим цифровим частотоміром).

Експериментально також перевірялася надійність формування випадкових бітових послідовностей для двоканальної схеми (рисунок 1.3). Спочатку статистичними тестами перевірялися випадкові бітові послідовності для двоканальної схеми.

Після проходження усіх тестів в другому каналі моделювалися три види несправностей :

– на виході другого каналу встановлювався «логічний нуль» – при цьому випадкові бітові сигнали першого каналу проходили на вихід елемента «ВИКЛЮЧНЕ АБО» без інверсії;

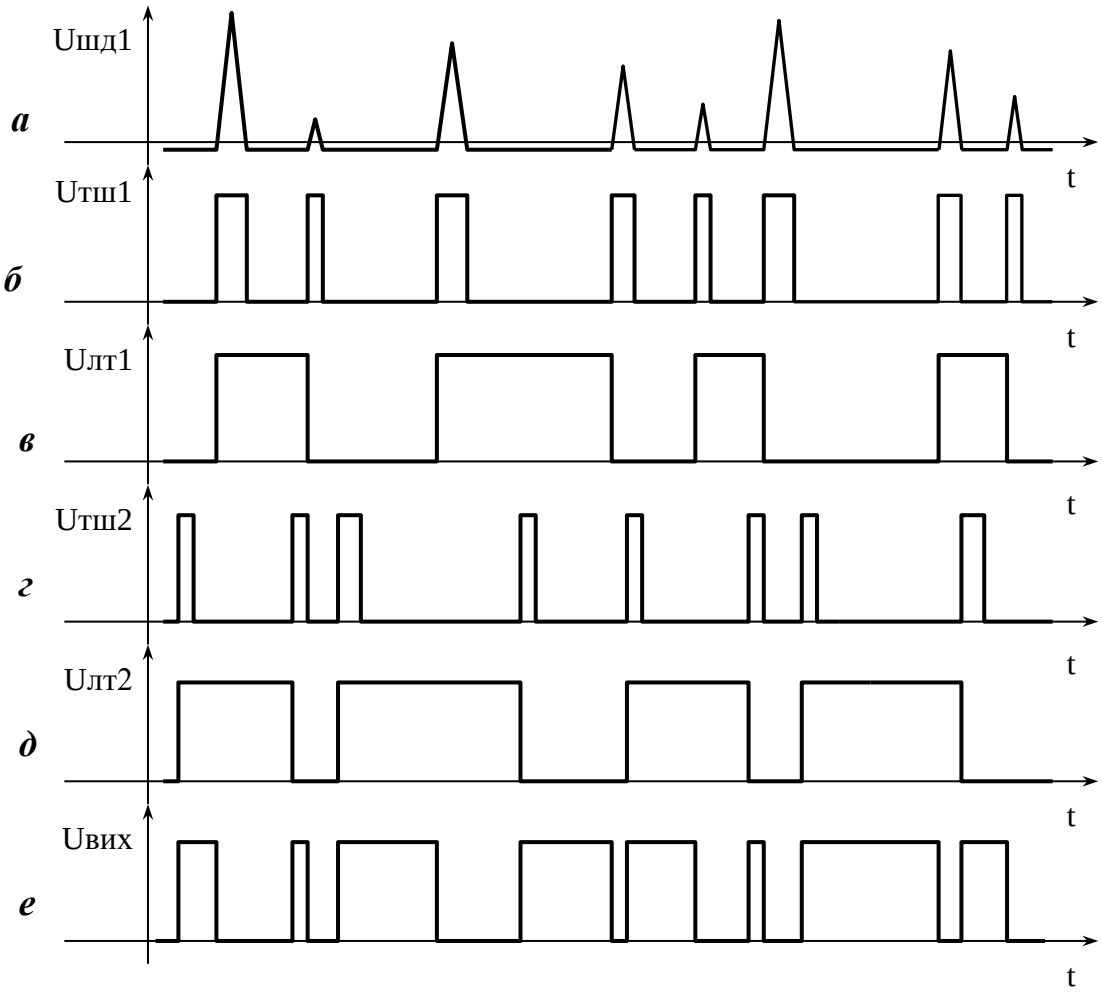

Рисунок 1.4 – Осцилограми випадкових сигналів

– на виході другого каналу встановлювалася «логічна одиниця» – при цьому випадкові бітові сигнали першого каналу проходили на вихід елемента «ВИКЛЮЧНЕ АБО» з інверсією;

– на вихід другого каналу подавалися прямокутні імпульси детермінованого генератора, тобто випадкові бітові сигнали першого каналу на виході елемента «ВИКЛЮЧНЕ АБО» першу половину періоду детермінованого генератора проходили без інверсії, а другу половину періоду – проходили з інверсією.

Для усіх трьох випадків проводилося повне статистичне тестування випадкових бітових послідовностей, що формуються. Експериментально встановлено, що при нормально працюючому одному каналі формування випадкових бітових послідовностей і будь-якому типі несправності другого

каналу – випадкові бітові послідовності, що формуються, проходять усі статистичні тести.

При експериментальних дослідженнях генераторів випадкових послідовностей (рисунок 1.2 або рисунок 1.3), реалізованих на логічних елементах ТТЛШ або КМОН, було виявлено, що при частоті вхідних імпульсів шумового генератора *Fш* близько 6 МГц імовірність «нульових» бітів перевищує імовірність «одиничних» бітів на величину приблизно:

$\Delta P = P(0) - P(1) = 0{,}005$.

При збільшенні вхідної частоти шумових імпульсів – різниця імовірностей ΔP також збільшується. Це пояснюється особливостями схемотехніки вихідного каскаду логічних мікросхем ТТЛШ і КМОН. Вихідний опір каскаду в стані «логічна одиниця» значно більше вихідного опору каскаду в режимі «логічний нуль» [2]. Тому час перезарядки «паразитних ємностей» навантаження елементу ТТЛШ або КМОН через вихідний опір каскаду буде різним. В результаті: рахунковий тригер довше переходить із стану «0→1», чим «1→0». Тому у вихідній послідовності генераторів випадкових бітів в середньому на 1000 «нулів» формується приблизно 995 «одиниць».

Існує декілька алгоритмів вирівнювання імовірностей випадкових бітових послідовностей. Перший алгоритм дозволяє значно зменшити різницю імовірностей випадкових бітів, що генеруються [2]. Для цього з двох послідовних випадкових бітів формується їх логічна функція «ВИКЛЮЧНЕ АБО». Проміжний регістр RG1 запам'ятовує два останніх випадкових біта, що генеруються (рисунок 1.5). У вихідний регістр RG2 записується логічна функція «ВИКЛЮЧНЕ АБО» цих бітів, але з частотою в два рази менше, ніж *Fo* (рахунковий тригер Т2 ділить частоту зсуву *Fo* на два).

Імовірність одиничного біта на вході регістра RG1 позначимо $P(1)$, а імовірність нульового – $P(0) = P(1) + \Delta$. Сума імовірностей $P(0) + P(1)$ завжди дорівнює одиниці. Запишемо усі комбінації бітів на виході проміжного регістра RG1 і імовірність цих комбінацій (з урахуванням повної

статистичної незалежності сусідніх випадкових бітів, що генеруються) (див. таблицю. 1.1).

На виході елемента «ВИКЛЮЧНЕ АБО» (див. рисунок 1.5) формується логічний нуль при комбінаціях, що відповідають першому і останньому рядкам таблиці. 1.1. Тому імовірність «нулів» *P(0)'* на виході елемента «ВИКЛЮЧНЕ АБО» дорівнює :

$$P(0)' = [\, P(1) + \Delta \,] * [\, P(1) + \Delta \,] + P(1) * P(1).$$

Таблиця 1.1 – Імовірності бітів на виході регістра RG1

Q1	Q2	Імовірності
0	0	$[\, P(1) + \Delta \,] * [\, P(1) + \Delta \,]$
0	1	$[\, P(1) + \Delta \,] * P(1)$
1	0	$P(1) * [\, P(1) + \Delta \,]$
1	1	$P(1) * P(1)$

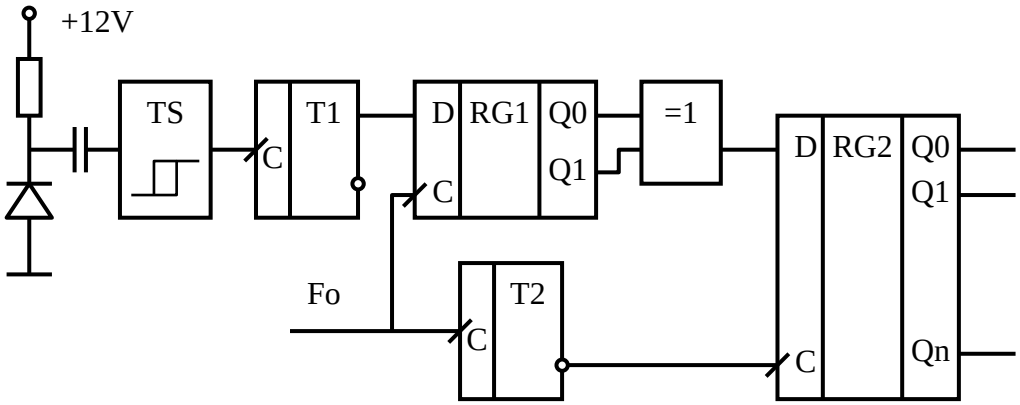

Рисунок 1.5 – Схема вирівнювання імовірностей по алгоритму «Дельта-квадрат»

Логічній одиниці на виході елемента «ВИКЛЮЧНЕ АБО» відповідатимуть два середні рядки в таблиці 1.1, тому імовірність «одиниць» *P(1)'* дорівнює :

$$P(1)' = [\, P(1) + \Delta \,] * P(1) + P(1) * [\, P(1) + \Delta \,].$$

Різниця імовірностей на виході елемента «ВИКЛЮЧНЕ АБО» Δ' дорівнює:

$$\Delta' = P(0)' - P(1)' = \Delta^2.$$

Враховуючи малу величину різниці імовірностей Δ (приблизно 0,005), можливо стверджувати, що її квадрат буде значно менше. До недоліків цього методу (умовно назвемо його: метод «Дельта-квадрат») можливо віднести в два рази меншу швидкість формування випадкових бітів і, хоча і маленьку, але не нульову, різницю імовірностей «0» і «1». Багаторазове застосування операції «ВИКЛЮЧНЕ АБО» при формуванні випадкових послідовностей дозволяє генерувати рівноімовірні випадкові послідовності із заданою різницею імовірності (наприклад, $\Delta = \left| P(0) - P(1) \right| < 10^{-6}$).

Другий метод (запропонований Джоном фон Нейманом) дозволяє вирівняти імовірності «0» і «1». Ідея цього методу зрозуміла з аналізу таблиці. 1.1. Імовірність другого і третього рядків рівні. Тому при комбінації сигналів на виходах проміжного регістра RG1, якій відповідає другий рядок, у вихідний регістр RG2 записується нульовий біт, а при комбінації, якій відповідає третій рядок, – одиничний біт. Комбінації сигналів, що відповідають першому і останньому рядкам, не використовуються.

Для реалізації цього методу нульовий логічний сигнал з виходу логічного елемента «ВИКЛЮЧНЕ АБО» (контролюючою виходи проміжного регістра RG1) забороняє запис випадкових бітів у вихідний регістр RG2

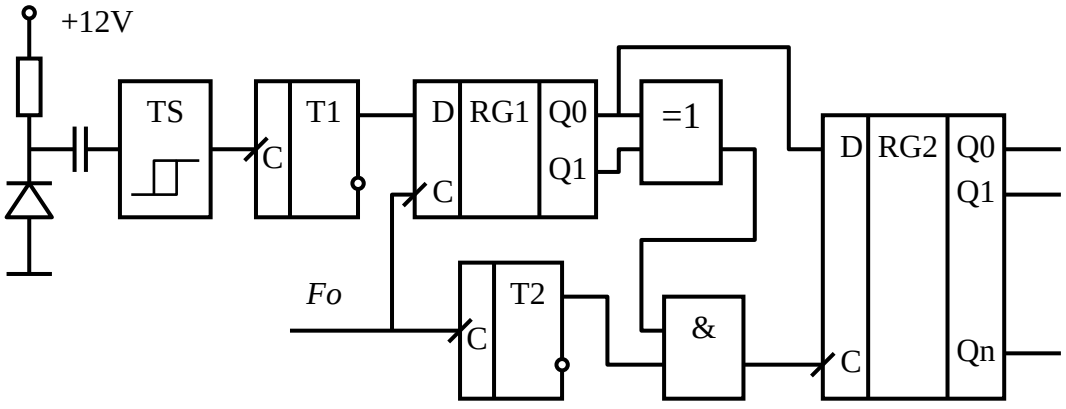

Рисунок 1.6 – Схема повного вирівнювання імовірностей

(рисунок 1.6) при комбінаціях, що відповідають першому і останньому рядкам таблиці. 1.1.

Функціонування генераторів випадкових бітових послідовностей, які наведені на рисунку 1.2, рисунку 1.3, рисунку 1.5 або рисунку 1.6 можливо тільки у складі програмно-апаратного комплексу, що включає:

– власне генератор рівномірно розподілених випадкових бітів,

– програмний драйвер зчитування випадкових чисел в ПЕОМ, або мікроконтролер,

– програм тестування випадкових послідовностей.

2 РОЗРОБКА АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ ГЕНЕРАТОРА ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

2.1 Вибір апаратної платформи генератора випадкових послідовностей

Переваги використання мікроконтролерів (МК) для побудови повністю функціонуючого обчислювального пристрою полягають в тому, що для цього досить однієї мікросхеми МК з приєднаними до неї пристроями вводу/виведення. Сучасні моделі однокришталевих МК перевищують обчислювальні можливості IBM PC AT на 286-му процесорі зразка другої половини 1980-х років.

Є динамічні області, де кордон між мікропроцесорами (МП) і МК провести важко — такі, наприклад, процесори для мобільних пристроїв, від звичайних телефонів до смартфонів і планшетів, в яких процесорний вузол повинен мати розвинені обчислювальні функції і управляти чисельними зовнішніми компонентами.

У сучасних AVR-контролерів є дві особливості, які відрізняють це сімейство від інших 8-розрядних МК. По-перше, це наявність конвеєра, завдяки чому для AVR не існує поняття машинного циклу: більшість команд виконуються за один такт. Для порівняння відзначимо, що МК сімейства PIC, які користуються великою популярністю, виконують команду за 4 такту, а класичні 8051 – взагалі за 12 або навіть 24 такту.

Друга особливість AVR-контролерів – наявність 32-х оперативних регістрів, які не зовсім рівноправні, але дозволяють в ряді випадків взагалі не звертатися до оперативної пам'яті і не використовувати в явному вигляді стек.

Особливості мікроконтролерів Atmel AVR:

– продуктивність порядку 1 MIPS (Millions of Instructions Per Second, мільйон команд в секунду). Обчислювальне ядро AVR на ряді завдань по продуктивності перевершує 16-розрядний процесор 80286;

– удосконалена RISC-архітектура (Reduced Instruction Set Computing) концепція обчислення зі скороченим набором команд передбачає наявність набору команд, що складається з мінімуму компактних інструкцій, які швидко виконуються. Концепція RISC спрощує пристрій ядра (в типовому ядрі AVR міститься лише 32 тис. транзисторів);

– в AVR є найпростіший двоступеневий конвеєр, коли команда виконується в одному такті з вибіркою наступної;

– роздільні шини пам'яті команд і даних – AVR (як і більшість інших мікроконтролерів) має так звану Гарвардську архітектуру, де області пам'яті програм і даних розділені (на відміну від класичної архітектури фон Неймана в звичайних комп'ютерах, де пам'ять загальна). Роздільні шини для цих областей пам'яті значно прискорюють виконання програми – дані і команди можуть вибиратися одночасно;

– 32 регістра загального призначення (РЗП);

– велика кількість різноманітних команд (інструкцій), номенклатура яких (приблизно від 90 до 130, в залежності від моделі контролера) для AVR більше, ніж в інших RISC-сімействах. Протиріччя з концепцією RISC тут немає, оскільки значна частина цих інструкцій-псевдоніми, і введені вони виключно для зручності програмування;

– Flash-пам'ять програм (10 000 циклів стирання/запис) – з можливістю внутрисистемного перепрограмування, тобто завантаження програм прямо в готовій схемі (In System Programming, ISP);

– окрема область енергонезалежної пам'яті (EEPROM, 100 000 циклів стирання/запис) — для зберігання даних з можливістю запису програмним шляхом або зовнішнього завантаження подібно програмам;

– вбудовані пристрої для обробки аналогових сигналів: Аналоговий компаратор і багатоканальний 10-розрядний АЦП.

Сімейство мікроконтролерів AVR має ряд переваг в силу універсальності пристрою, легкості завантаження програм, наступності структури для різних типів контролерів, різноманітності типів корпусів,

простоти схемотехніки, практично позбавленої будь-яких специфічних особливостей, що ускладнюють освоєння новачками. Окремо слід відзначити наявність відмінної бази для початку роботи з цими МК у вигляді платформи ARDUINO і всього, що з нею пов'язано.

Платформа виникла в середовищі співробітників Interaction Design Institute (що можна перекласти як «Інститут конструювання взаємодій» з італійського містечка Івреа і отримала свою майже толкієнівську назву по імені реально існуючого короля Ардуїна, який правив цією місцевістю на початку минулого тисячоліття.

ARDUINO виросла з завдання навчити студентів непрофільних спеціальностей створювати електронні пристрої, причому швидко і, бажано, без опори на поглиблене вивчення електроніки, електротехніки та програмування.

Зрештою група, керована програмістом Массімо Банці, створила універсальну апаратну платформу на основі дешевих, зручних і доступних мікроконтролерів Atmel AVR і вирішила її поширювати на принципах open source. Ліцензія ARDUINO забороняє використання цієї торгової марки для будь-яких сторонніх продуктів, крім розширень основного проекту.

Філософія ARDUINO полягає в тому, що якщо ви бажаєте навчитися електроніці, ви зможете вивчати її з першого дня, а не займатися пошуком додаткового обладнання та програм.

Своїм успіхом ARDUINO багато в чому зобов'язаний існуючому до нього «софту» Processing і Wiring. Від цих проектів ARDUINO успадкував одну сильну рису – зручне для користувачів середовище розробки.

До появи ARDUINO програмування мікроконтролерів супроводжувалося складним і рутинним навчанням. З ARDUINO навіть ті, хто не мав досвіду роботи з електронними пристроями, тепер можуть доторкнутися до раніше недосяжного для них світу електроніки.

Платформа ARDUINO постійно розвивається і тепер ARDUINO представлена не однією платою, а цілим сімейством. На додаток до

оригінального проекту, названому ARDUINO UNO, нові проекти, що мають більш потужні засоби на платі, носять назву ARDUINO MEGA, більш компактні називаються ARDUINO NANO, у водонепроникному виконанні – LilyPad ARDUINO.

Плата з більш потужним 32-розрядним процесором Cortex M3 ARM називається ARDUINO DUO.

Остання модель ARDUINO UNO Rev3 (рисунок 2.1) виконана на базі процесора ATmega328P з тактовою частотою 16 МГц, має 20 контрольованих контактів вводу/виведення (Input/Output) для взаємодії із зовнішніми пристроями.

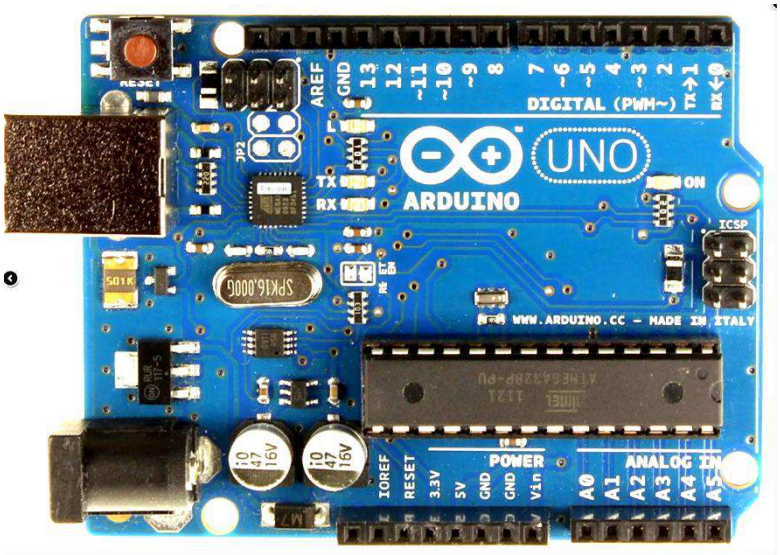

Рисунок 2.1 – ARDUINO UNO Rev3

ARDUINO UNO може живитися як від USB підключення, так і від зовнішнього джерела: батарейки або мережевого джерела живлення з напругою від 6 до 15 В.

Плата ARDUINO UNO оснащена 32 кБ flash-пам'яті програм, 2 кБ з яких відведено під так званий bootloader (завантажувач). Він дозволяє завантажувати програми ARDUINO (скетчі) зі звичайного комп'ютера через рознімання USB.

Також є 2 кБ SRAM-пам'яті, які використовуються для зберігання тимчасових даних змінних програми. Ще є 1 кБ EEPROM-пам'яті для довготривалого зберігання даних (навіть після виключення напруги живлення).

Кожен з 14 цифрових контактів може працювати в якості входу або виходу. Рівень напруги на контактах обмежений 5В. Максимальний струм, який може віддавати або споживати один контакт, становить 40 мА.

У ARDUINO UNO є 6 аналогових входів (A0 - A5), кожен з яких може представити аналогову напругу у вигляді 10-бітового числа (1024 різних аналогових значень). За замовчуванням, вимірювання напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можливо змінити, використовуючи контакт AREF або функцію analogReference (). Крім цього, деякі з аналогових входів мають додаткові функції.

У пакет програмного забезпечення ARDUINO входить спеціальна програма, що дозволяє зчитувати в комп’ютер, а також відправляти на ARDUINO прості текстові дані.

ARDUINO NANO (рисунок 2.2, рисунок 2.3) – компактний варіант плати ARDUINO UNO за рахунок розміщення компонентів з двох сторін плати.

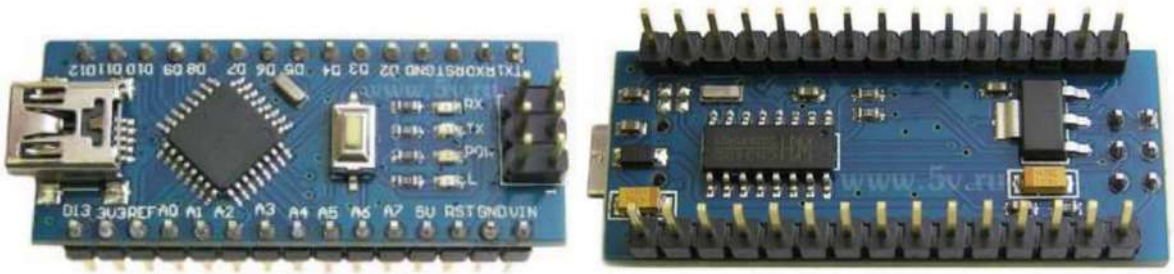

Рисунок 2.2 – ARDUINO NANO

ARDUINO MEGA 2560 (рисунок 2.4) виконана таким чином, щоб бути максимально сумісною зі своїми молодшими побратимами і модулями

розширення. Ліва частина плати по конфігурації контактів ідентична ARDUINO UNO, як по розташуванню, так і за призначенням.

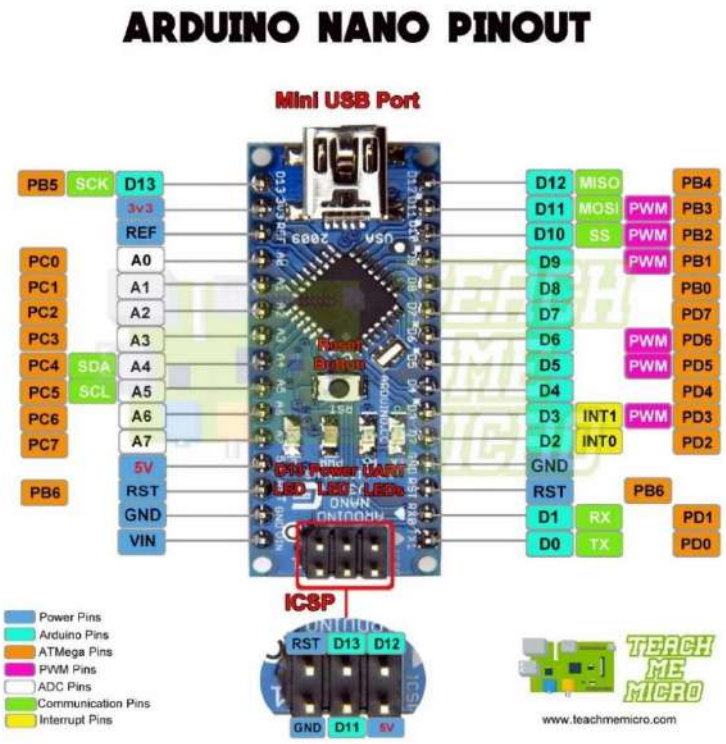

Рисунок 2.3 – Розпіновка плати ARDUINO NANO

Це означає, що ARDUINO MEGA 2560 може просто підмінити ARDUINO UNO, якщо її можливостей перестало вистачати. Живлення, розподіл напруги, захист USB і принципи взаємодії аналогічні базовій моделі.

Рисунок 2.4 – Апаратний модуль ARDUINO MEGA

В таблиці 2.1 наведені порівняльні характеристики апаратних модулів ARDUINO UNO (ARDUINO NANO) та ARDUINO MEGA.

Таблиця 2.1 – Порівняльна таблиця процесорних плат ARDUINO

Процесорна плата ARDUINO	UNO, NANO	MEGA
Мікроконтролер	ATMEGA-328P	ATMEGA-2560
Кількість цифрових входів/виходів	14	54
з них ШІМ підтримують	6	15
Кількість аналогових входів	6	16
Кількість контактів апаратного переривання	2	6
Кількість апаратних serial-портів	1	4
Тактова частота, МГц	16	16
Об'єм Flash-пам’яті програм, кБ	32	256
Об'єм SRAM-пам’яті даних, кБ	2	8
Об'єм EEPROM-пам’яті даних, кБ	1	4

ARDUINO – це не спеціалізований продукт або технологія, це екосистема апаратного і програмного забезпечення, інструментів і людських ресурсів. Поряд з основною апаратною платформою – процесорною платою, важливу роль відіграють плати розширення – ARDUINO SHIELDS.

Завдяки цим платам і модулям значно розширюється галузь застосування ARDUINO, а також відкриваються нові можливості для простих радіо-аматорів і комерційних розробників.

Практично всі процесорні плати ARDUINO завдяки універсальному форм-фактору підтримують підключення модулів розширення і додаткових плат за допомогою спеціальних штирьових роз'ємів по краю плати (тобто без застосування паяльника).

Саме тому з'явилося безліч спеціалізованих плат розширення: плати реле для комутації потужних навантажень, плати АЦП і ЦАП, плати контроля електричної мережі, плати драйверів електродвигунів ін.

Через плати розширення модулі ARDUINO легко підключаються до дротових і бездротових комп'ютерних мереж.

Апаратна платформа ARDUINO підтримується пакетом програмного забезпечення з відкритим вихідним кодом, в який входить інтегроване середовище розробки (ARDUINO IDE), стандартна мова програмування з компілятором і завантажувач. Редактор коду включає в себе такі функції, як підсвічування синтаксису, підсвічування дужок і автоматичний відступ, а також мається функції компіляції і завантаження програми в мікроконтролер одним клацанням миші.

Компілятор Сі ARDUINO дещо простіше, ніж професійні Сі-компілятори, але вельми ефективний. З компілятором ARDUINO не потрібно піклуватися про програмування складних апаратних засобів, оскільки в середовищі розробки є відповідні вбудовані команди.

2.2 Лабораторний макет генератора випадкових послідовностей

Лабораторний макет генератора випадкових послідовностей реалізовано на процесорній платі ARDUINO_NANO (рис. 2.5). Це дозволяє з мінімальними додатковими витратами створити лабораторний макет для генерації та дослідження випадкових послідовностей.

Два датчики шуму (джерела невизначеності, джерела ентропії) реалізовані на шумових діодах КГ-401Б (можливо використання база-емітерних переходів кремнієвих високочастотних мезапланарних транзисторів).

Напруга живлення датчиків шуму +15В формується перетворювачем постійної напруги (DC/DC) з напруги живлення плати ARDUINO NANO +5 В.

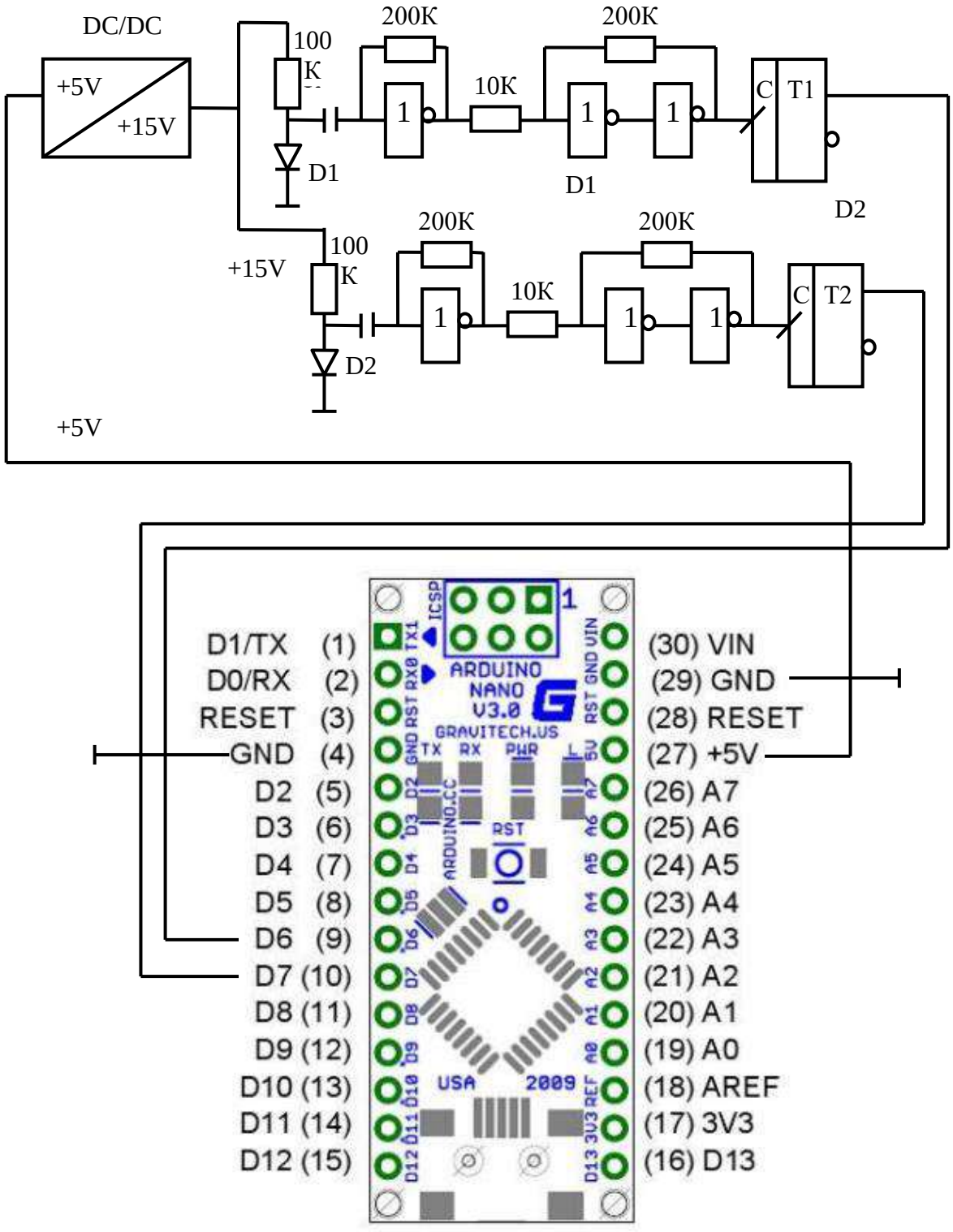

Рисунок 2.5 – Лабораторний макет на основі ARDUINO–NANO

2 резистори по 100кОм, які включені послідовно з датчиками шуму, обмежують струм через діоди до 70 мА.

2 обмежувача-підсилювача аналогових сигналів з коефіцієнтом підсилення 7…10 на логічних КМОП елементах мікросхеми D1 збільшують амплітуду шумових імпульсів до 2…5 В.

Тригери Шмітта реалізовані на логічних елементах мікросхеми D1 з позитивним зворотнім зв'язком. Вихідні сигнали тригерів Шмітта подаються до входів лічильних тригерів, які реалізовані на мікросхемі D2.

Вихідні сигнали лічильних тригерів надходять до цифрових входів D6, D7 процесорної плати ARLUINO NANO. Ці входи відповідають пінам PIND 6, PIND 7 мікроконтролера ATMega 328P.

Подальша обробка випадкоих сигналів відбувається на програмному рівні у мікроконтролері ATMega 328P.

2.3 Програмне забезпечення генератора випадкових послідовностей

Програма (scetch) написана на мові програмування процесорних модулів ARDUINO (аналог мови Сі) з асемблерними вставками для досягнення максимальної пришвидкості формування випадкових байтів.

Програма має обов'язкові модулі: void setup (), який виконується один раз і void loop (), який виконується у нескінченому циклі до вимкнення живлення плати ARDUINO.

Директиви .equ задають адреси портів в просторі регістрів вводу/ виведення.

В модулі void setup () запускається бібліотека Serial.begin(38400) послідовного порта UART та налаштовується швидкість обміну даними між процесорною платою ARDUINO та персональним комп'ютером.

В цьому ж модулі порт D програмується на введення даних.

В модулі void loop() у нескінченому циклі формуються випадкові байти з цифрових сигналів, які зчитуються з бітів 7 і 6 порта PIND у регістр R18. Регістр R18 копіюється у регістр R19. Цей регістр зсувається ліворуч та об'єднується операцією XOR (ВИКЛЮЧНЕ АБО) з регістром R18. Таким чином у старшому біті регістра R18 формується перший випадковий біт

(тобто об’єднуються випадкові біти двох каналів). Цей біт використовується для вирівнювання імовірностей за алгоритмом Неймана.

Перед зчитуванням наступних сигналів з бітів 7 і 6 порта PIND формується часова затримка для уникнення статистичних зав’язків між сусідніми зчитаними бітами. Цю затримку можливо програмно змінювати записом константи у регістр R17.

Другий випадковий біт формується аналогічно у регістрі R20.

Два регістри R18 та R20 об’єднуються операцією XOR (ВИКЛЮЧНЕ АБО). Якщо старший біт регістра R20 дорівнює «0», – це означає, що сформовані у старших бітах регістрів R18 та R20 випадкові біти були однакові. За алгоритмом Неймана така комбінація бітів відкидається (не використовується). У іншому разі сформований у регістрі R18 старший біт зсувається (праворуч) у флаг C, після чого послідовно заштовхується у регістр R22.

Такий цикл повторюється 8 разів для формування випадкового байта у регістрі R22. Цей випадковий байт передається у регістр даних послідовного порту UDR.

З цього регістру байти апаратно переписуються у регістр зсуву і у послідовному форматі виводяться на контакт TxD мікроконтролера ATMega 328P. Мікросхема CH340G на платі ARDUINO_NANO перетворює сигнали TxD послідовного порта UART в сигнали інтерфейса USB 1.0.

Лістинг 1.1 – Формування випадкових послідовностей

```
.equ   DDRD,  0x0A;          // регістр напрямку  порта D
.equ   PIND,  0x09;          // вхідний регістр порта D
.equ   UDR,  0xC6;           // регістр даних послідовного порта

void setup () {
    Serial.begin(57600);     // налаштування UART на 57600 бод
```

```
        asm volatile        {
        ldi     R16,0 n
        out     DDRD,R16 n
        }
}
void loop()    {

        asm volatile
        {
L4:     ldi         R16,8       // лічильник бітів у кожному байті
L5:     in          R18, PIND   // завантаження порта D у регістр
        mov         R19,R18     // копіювання регістрів
        lsl         R19         // логічний зсув ліворуч
        eor         R18,R19     // XOR двох регістрів
L6      ldi         R17,4       //
        dec         R17         //   часова затримка
        brNz        L6          //
        in          R20, PIND   //      формування наступного біта
        mov         R21,R20     // копіювання регістрів
        lsl         R21         // логічний зсув ліворуч
        eor         R20,R21     // XOR двох регістрів
        eor         R20,R18     // сформований випадковий біт
        sbrs        R20,7       // пропустити, якщо біт дорівнює “1”  –
                    // вирівнювання імовірностей за алгоритмом Неймана
        rjmp        L5
        lsl         R18         // сформований випадковий біт -> флаг  C
        ror         R22         // випадковий біт зсувається у R22
L7      ldi         R17,3       //
        dec         R17         //   часова затримка
        brNz        L7          //
```

```
dec         R16
brNe        L4          // сформований випадковий байт
out         UDR, R22    // передавання випадкового байта через
                        // послідовний порт UART
}
}
```

3 ТЕСТУВАННЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Обов'язковою умовою при формуванні ключових даних є використання джерел ентропії (невизначеності) – джерел шуму. Це забезпечує пряму і зворотну секретність, тобто непередбачуваність формованих ключових даних [1,2].

Тестом на непередбачуваність випадкових послідовностей є «тест наступного біта». Цей тест вважається пройденим, якщо після спостереження і аналізу як завгодно довгою бітової послідовності можливо передбачити наступний біт або кілька бітів з ймовірністю приблизно 0,5 [2].

На різних етапах життєвого циклу апаратних генераторів випадкових послідовностей (АГВП) параметри фізичних електронних датчиків шуму, аналого-цифрових перетворювачів і логічних схем формування випадкових чисел, зміни напруги живлення або зовнішніх кліматичних факторів – деградують [2,4].

Тому в системах з високою надійністю формування випадкових бітових послідовностей застосовують постійний апаратний контроль головних параметрів фізичних датчиків шуму [1, 2], логічних схем формування випадкових бітів (tot-test), а також статистичний контроль вихідних випадкових бітових послідовностей [1, 2, 5, 6, 7, ,8 ,9].

Розрізняють види тестувань:

– перевірка статистичних властивостей випадкових бітових послідовностей малої довжини [2, 7];

– постійне (оперативне) тестування (Online-test) – в процесі генерації випадкових послідовностей перевіряються всі вихідні випадкові біти [2, 5, 8];

– технологічне тестування – проводиться при включенні напруги живлення генераторів випадкових бітових послідовностей, або через фіксовані часові інтервали (зазвичай – не більше 24-х годин) і по запиту користувача при виникненні підозр про вплив зовнішніх факторів [2];

– повне тестування – здійснюється при приймально-здавальних випробуваннях апаратних генераторів випадкових бітових послідовностей і/або при проведенні регламентних робіт [2, 9].

3.1 Перевірка статистичних властивостей випадкових послідовностей малої довжини

Дана перевірка призначена для тестування статистичних властивостей випадкових і псевдовипадкових бітових послідовностей малої довжини, до яких відносяться послідовності з довжинами n = {64, 128, 192, 256, 512, 768, 1024, 2048, 4096, 8192, 16384} біт. Саме такі довжини ключових даних використовують сучасні криптографічні алгоритми [2, 4, 7].

Необхідність тестування визначається тим, що не всі ключі і таблиці замін (сформовані навіть ідеальним генератором випадкових бітів) забезпечують максимальну стійкість шифрів. Так, для алгоритму AES відомо існування так званих «слабких ключів», при використанні яких зв'язок між відкритими і зашифрованими даними не маскується достатнім чином, і шифр порівняно просто розкривається [2, 4].

Невтішний висновок про те, що навіть ідеальний генератор випадкових біт (до нього наближається алгоритм підкидання монети) може виробляти «слабкі» ключі (оскільки вони, в силу рівноімовірності, мають таку ж імовірність, як і всі інші «хороші» ключі), визначає необхідність тестування сформованих випадкових ключових даних.

Перевірка статистичних властивостей випадкових бітових послідовностей малої довжини здійснюється з урахуванням рекомендацій AIC 20 [2, 7].

Перевірка включає виконання таких статистичних тестів.

1 Монобітний тест:

Метою монобітного тесту є визначення – чи буде кількість нулів і одиниць в бітовій послідовності такою ж, як це можна очікувати в ідеальній випадковій послідовності.

2 Тест серій:

Метою тесту серій є визначення – чи буде кількість серій одиниць (або нулів) різної довжини в бітовій послідовності такою ж, як можна очікувати у ідеальній випадковій послідовності.

3 Покер-тест:

Нехай m позитивне ціле число таке, що $\left\lfloor \frac{n}{m} \right\rfloor \geq 5 \times \left(2^m\right)$, і нехай $k = \left\lfloor \frac{n}{m} \right\rfloor$

Розіб'ємо послідовність s на k блоків, які не перекриваються, довжиною m. Нехай n_i – кількість блоків i-го типу довжиною m, $1 \leq i \leq 2^m$.. Метою тесту є визначення: чи будуть блоки довжиною m зустрічатися в бітовій послідовності s з такою ж частотою, як у випадковій послідовності. Під час тестування розраховується статистика:

$$X_1 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k,$$

яка приблизно підпорядковується χ^2-розподілу з 2^m ступенями свободи.

В таблиці 3.3 представлені припустимі значення статистики X_1.

Таблиця 3.3 – припустимі значення статистики X1

n	m	X_1
64	2	25. 9017
128	3	35. 2585
192	3	35. 2585
256	3	35. 2585
512	4	50. 4927
768	4	50. 4927
1024	5	76. 5625
2048	6	122. 7266
4096	6	122. 7266
8192	7	206. 7220
16384	8	362. 9867

4 Аавтокореляційний тест:

Мета тесту – перевірка співпадіння (кореляції) між послідовністю s і її циклічними зсувами. Нехай d фіксоване ціле число,. Кількість біт в послідовності s, що не збігаються з їх d-зсувом обчислюється як

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d} . \qquad (3.6)$$

Статистика

$$X_2 = 2\left(A(d) - \frac{n-d}{2} \right) \Big/ \sqrt{n-d} \qquad (3.7)$$

має розподіл близький до стандартного нормального розподілу $N(0,1)$. Значення статистики X_2 повинні знаходитися в інтервалі {-4,7534; 4,7534}.

При відхиленні гіпотези про випадковий розподіл вибірки малої довжини – ця вибірка не використовується (відкидається, бракується). Але це не є проблемою – нова вибірка може бути згенерована і протестована за припустимо малий час (тисячні частки секунди).

3.2 Постійний (оперативний) контроль (ON-LINE-test)

Обсяг вибірок випадкових бітових послідовностей і кількість тестів при такому контролі впливають на швидкість роботи апаратного генератора, тому тест AIS 31 (Application Notes and Inter-pretation of the Scheme) [8], який застосовується для цих цілей перевіряє бітові послідовності s довжиною n = 20000 біт шістьма основними тестами [5, 8].

1 Блочний тест – (тест Покера) – тест на рівномірність появи сусідніх блоків по 4 біта.

2 Монобітний тест – тест на рівноймовірність випадкових бітів, що генеруються, (цей тест можливо розглядати, як тест Покера з довжиною блоку 1 біт).

Мета цього тесту полягає в тому, щоб визначити, чи дійсно число одиничних і нульових біт в послідовності приблизно є таким, як можливо

очікувати для випадкової послідовності. Нехай число одиниць позначено $n1$. Тоді число $n1$ має задовольняти умові: $9654 < n1 < 10346$.

3 Тест серій.

Мета тесту серій полягає в тому, щоб визначити, чи дійсно число серій різних довжин в послідовності s таке ж, як можливо очікувати для істинної випадкової послідовності.

4 Автокореляційний тест перевіряє незалежність появи випадкових бітів від результатів генерації попередніх бітів;

Цей тест розраховує кореляцію між бітами послідовності s і її зсувами:

$$Z_\tau = \sum_{j=1}^{5000} \left(b_j \oplus b_{j+\tau} \right).$$

Тест вважається пройденим, якщо для всіх τ виконується нерівність:

$$2326 < Z_\tau < 2674.$$

5 Тест довгих серій – контролює довжину серій і видає повідомлення про появу дуже малоймовірних довгих серій.

Мета тесту довгих серій полягає в тому, щоб визначити, чи дійсно максимальна довжина деякої серії в послідовності s така ж, як можливо очікувати для випадкової послідовності. Тест довгих серій пройдено, якщо довжина найдовшої серій не перевищує 34

6 Диз'юнктивний тест – порівнює сусідні бітові блоки по 32 біта.

Здійснюється перевірка, яка в безлічі, сформованому зі слів w1, w2, ... w_m, де $w_i=\{0,1\}^{32}$, що не перетинаються, порівнює сусідні слова, тобто мета перевірки – встановити, що не існує жодного послідовного однакового 32-х бітового слова.

У всіх тестах для задовільних значень статистичних параметрів задаються межі (довірчі інтервали) [2, 5,8]. Попадання результатів тестування в ці довірчі інтервали повинно походити з довірчою ймовірністю, яка залежить від розмірів довірчих інтервалів і зазвичай знаходиться в діапазоні від 0,95 ... 0,99.

Якщо один з тестів не пройдено, то бракується (відкидається) вся бітова послідовність s=20 000 біт.

Тест AIS 31 оцінює конкретний бітовий блок s обмеженої довжини (20000 біт). Тому результат тестування відноситься тільки до цього блоку і не може бути поширений на АГВП (апаратний генератор випадкових послідовностей) в цілому.

Сучасні ПЕОМ дозволяють проводити від 30 до 100 тестувань в секунду за алгоритмом AIS 31.

3.3 Технологічне тестування

Це тестування може тривати від однієї секунди до однієї хвилини, тому довжина бітових послідовностей, що перевіряються, може досягати 1 Мбайт і більше із застосуванням основних статистичних тестів [2]. Результати тестування оцінюють не тільки блок бітів, що перевіряється, але також свідчать про працездатність всього апаратного генератора випадкових послідовностей (АГВП) або відповідності випадкових послідовностей, що генеруються, моделі ідеального генератора випадкових бітів (ГВБ).

Довжина послідовності, що тестується, повинна бути не менше: $N \geq 10^6$ біт.

Кількість тестів визначається компромісом між бажанням проведення всебічного тестування та обмеженим часом тестування [2]. Тому обрані 5 основних тестів:

- монобітний тест (тест на рівноймовірність випадкових бітів);
- тест Покеру (тест на рівномірність появи блоків по 8 біт);
- автокореляційний тест і
- тест серій (тести на незалежність);
- тест довжин серій.

Для зменшення часу тестування поєднуються в часі процес генерації випадкової вибірки довжиною 1 Мбайт (8 Мбіт) і розрахунок теста Покеру, а

також поєднуються у часі обчислення тестів серій і монобітного тесту. При обчисленні автокореляційної функції використовуються перший мільйон випадкових бітів.

Для кожного тесту розраховано допустимі межі (довірчі інтервали) на основі критерію χ2 (хі-квадрат) [2].

Для експериментальних досліджень апаратних генераторів випадкових і псевдовипадкових бітових послідовностей програма технологічного тестування була доповнена графічною оболонкою, що дозволяє виводити на екран монітора результати досліджень по кожному тесту в зручній формі.

Необхідно враховувати, що представлення результатів тестувань в графічній формі призводить до збільшення часу технологічного тестування, тому така можливість використовується тільки при експериментальних дослідженнях нових генераторів і в навчальних цілях.

У програму додана також можливість збереження на жорсткому диску сформованої випадкової бітової послідовності 1 Мбайт і результатів кожного тесту в текстовому форматі у вигляді таблиць [2].

У програму технологічного тестування доданий датчик псевдовипадкових послідовностей, наявний в програмному середовищі C++. Це дозволяє порівнювати результати тестувань реальних генераторів випадкових послідовностей з майже ідеальним, з точки зору статистичних характеристик, генератором псевдовипадкових послідовностей [2].

3.4 Повне тестування

Це тестування не обмежується в часі (воно може тривати кілька годин), тому обсяг вибірок може досягати 100 Мбіт і більше із застосуванням всіх відомих методів тестування.

Тестування здійснюється відповідно до рекомендацій NIST SP 800-22 (NIST STS) [2, 9].

Набір тестів NIST STS був запропонований в ході проведення конкурсу на новий національний стандарт США блочного шифрування – AES. Цей набір застосовувався для досліджень статистичних властивостей кандидатів на новий блочний шифр. На сьогодні методика тестування, яка запропонована NIST, є найбільш поширеною у розробників криптографічних засобів захисту інформації.

На комп’ютері з частотою двоядерного процесора 3 ГГц тестування NIST SP 800-22 (NIST STS) триває 15 хвилин.

4 ДОСЛІДЖЕННЯ ЛАБОРАТОРНОГО МАКЕТУ ГЕНЕРАТОРА ВИПАДКОВИХ ПОЛІДОВНОСТЕЙ

4.1 Дослідження статистичних параметрів випадкових послідовностей

Лабораторний макет у нескінченому циклі (доки включено джерело живлення) генерує рівноімовірні випадкові байти та передає їх через послідовний інтерфейс USB у комп'ютер. У комп'ютері програма «Термінал», яка входить до складу середовища програмування ARDUINO, відображає ці байти на екрані монітора, запам'ятовує у оперативній пам'яті, та після вводу вказаної кількості байтів – записує їх з визначеним попередньо ім'ям на носій інформації (HDD або SSD).

Живлення лабораторного макета відбувається від комп'ютера через кабель USB.

Згенеровані випадкові послідовності довжиною 1 МБайт (8 Мбіт) тестуються програмою оперативного тестування з відображенням результатів кожного тестування в графічному або текстовому вигляді [2].

При першому включені лабораторного макета часові затримки між сусідніми випадковими сигналами, які зчитуються з 6-го та 7-го піна порта PIND, – мінімальні. Для цього у програмі закоментовані команди формування часових затримок.

Статистична залежність між сусідніми бітами, що генеруються, неприпустимо велика. Це визначається не проходженням теста Покера (рисунок 4.1), автокореляційного тесту (рисунок 4.2) і тесту серій (рисунок 4.3).

Максимально припустимий результат для проходження теста Покера за критеріям χ^2 не повинен перевищувати 293 (на рисунку. 4.1 цей показник – 26756).

На рисунку 4.2 показані результати автокореляційного тесту для часових затримок τ від 1 до 100. Паралельно горизонтальній осі нанесені три

лінії, які відповідають середньоквадратичній похибці +1σ, +2σ, +3σ (вище горизонтальної осі *Tau*), а також три лінії, які відповідають середньоквадратичній похибці –1σ, –2σ, –3σ (нижче горизонтальної осі).

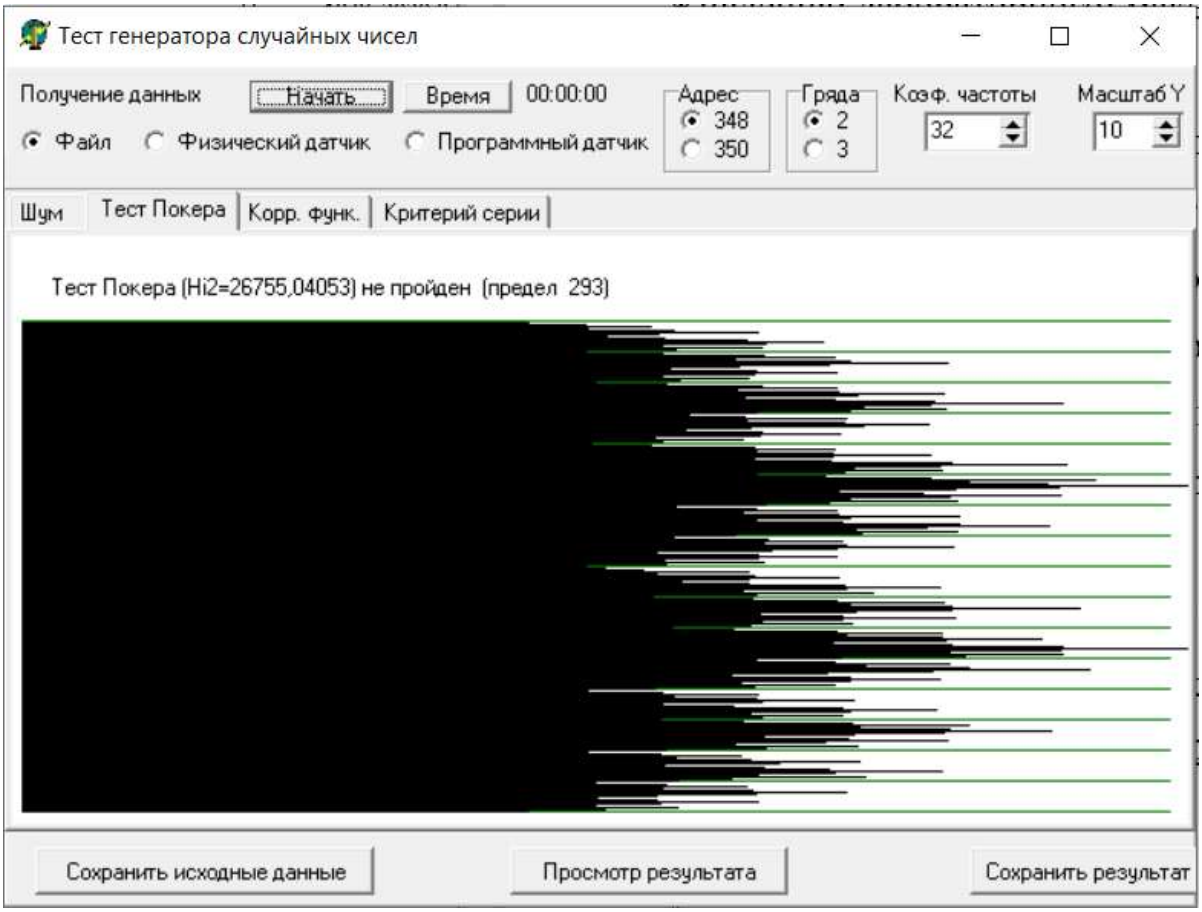

Рисунок 4.1 – Тест Покера не пройдено

Автокореляційна функція має нормальне (Гаусовське) розподілення з параметрами N(0,1). Для проходження автокореляційного теста необхідно, щоб не менш 67 значень (зі 100) автокореляційної функції попадали в інтервал +–1σ, не менш 95 значень – в інтервал +–2σ та не менш 99 значень – в інтервал +–3σ.

За цими показниками автокореляційний тест не пройдено.

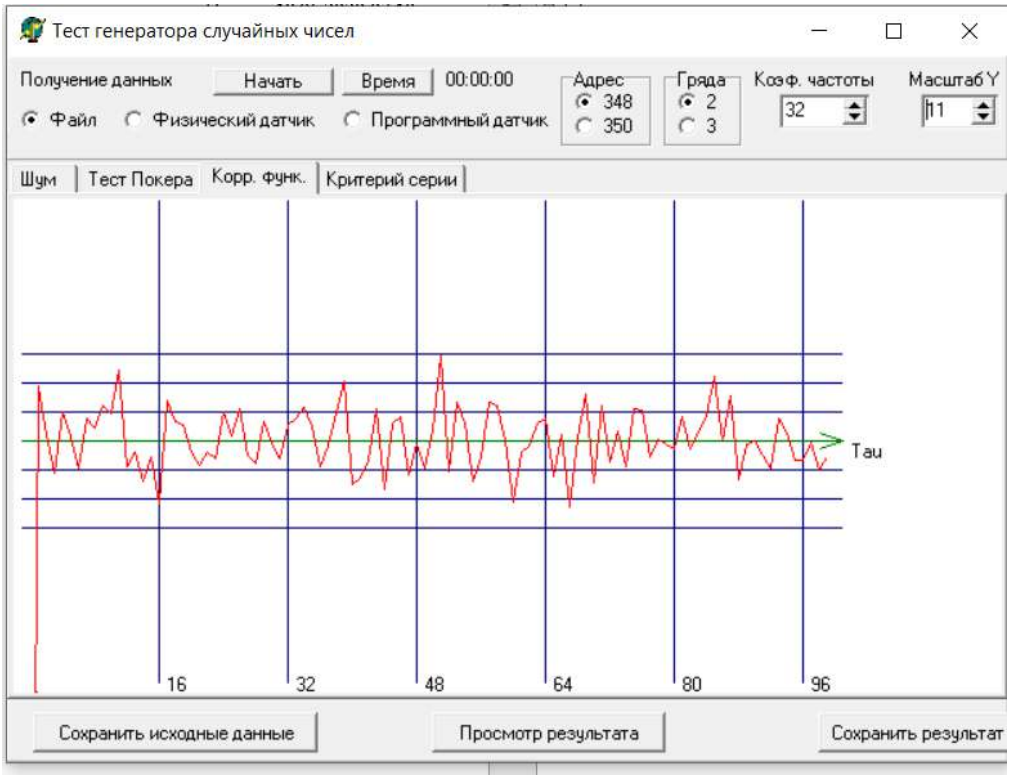

Рисунок 4.2 – Автокореляційний тест не пройдено

На рисунку 4.3 вказано, що монобітний тест пройдено. Це є свідченням ефективної роботи алгоритму Неймана вирівнювання імовірностей, який реалізовано на програмному рівні у мікроконтролері.

Тест серій – не пройдено. Це свідчить про статистичні зв'язки між сусідніми випадковими бітами.

На рисунку 4.3 вказано, що найдовша серія – 20 біт. Враховуючи на те, що найдовша серія не повинна перевищувати 34 біта, тому тест довгих серій пройдено.

Для зменшення статистичних зв'язків між сусідніми випадковими бітами треба збільшити часовий інтервал між зчитуваннями випадкових бітів. Для цього в програму додані дві часові затримки з можливістю програмного збільшення часового інтервалу.

Константа, яка записується в регістр R17 пропорційна часовому інтервалу, що формується. Збільшення константи в регістрі R17 на одиницю збільшує часову затримку на 3 такти процесора, тобто на t = 3 / 16 МГц = 0,188 мкс.

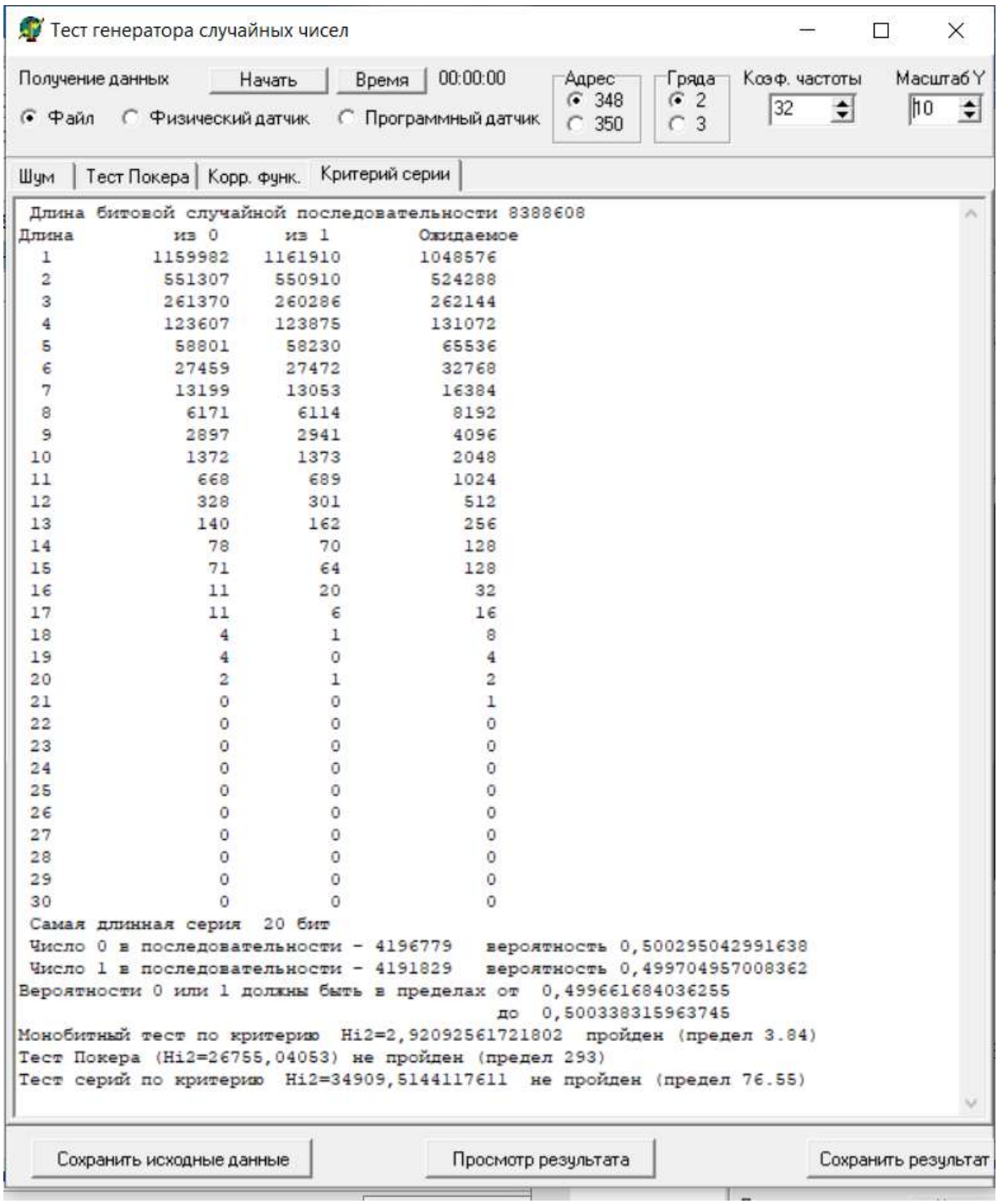

Рисунок 4.3 – Результати монобітного тесту та тесту серій

Треба також враховувати час виконання інших команд в циклі при формуванні випадкових бітів. Саме тому константа, яка записується у регістр R17 після формування другого біта, менше, ніш після формування першого біта. Можливо записувати однакові константи, обираючи найбільшу константу.

Після введення часової затримки проводиться генерація нового файла випадкових послідовностей розміром 1 МБайт (8 Мбіт) та його тестування.

На рисунках 4.4, 4.5, 4.6 наведені результати тестування випадкових послідовностей після введення в програму часових затримок (константи в регістрі R17 наведені в тексті програми в главі 2).

Тест Покера (рисунок 4.4) при параметрі $\chi^2 = 286$ пройдено (тому, що максимально припустимий результат не повинен перевищувати 293).

Автокореляційний тест також пройдено (рисунок 4.5) тому, що тільки 3 значення автокореляційної функції перевищують значення +–2σ, а за довірчий інтервал +–3σ не вийшло жодне значення автокореляційної функції.

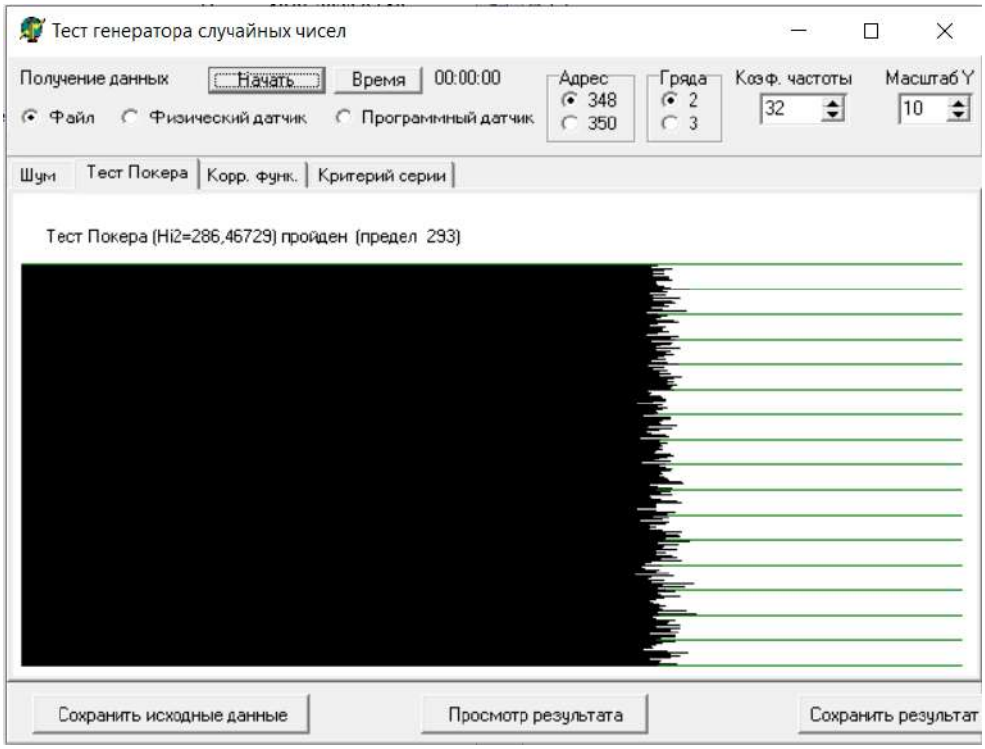

Рисунок 4.4 – Результати теста Покера

Монобітний тест також пройдено – параметр $\chi^2 = 1{,}76$ (при максимально припустимому значені 3,84).

Зі значним запасом пройдено тест серій $\chi^2 = 49$ (при максимально припустимому значені 76,55).

Найдовша серія 24 (при максимально припустимому значені 34).

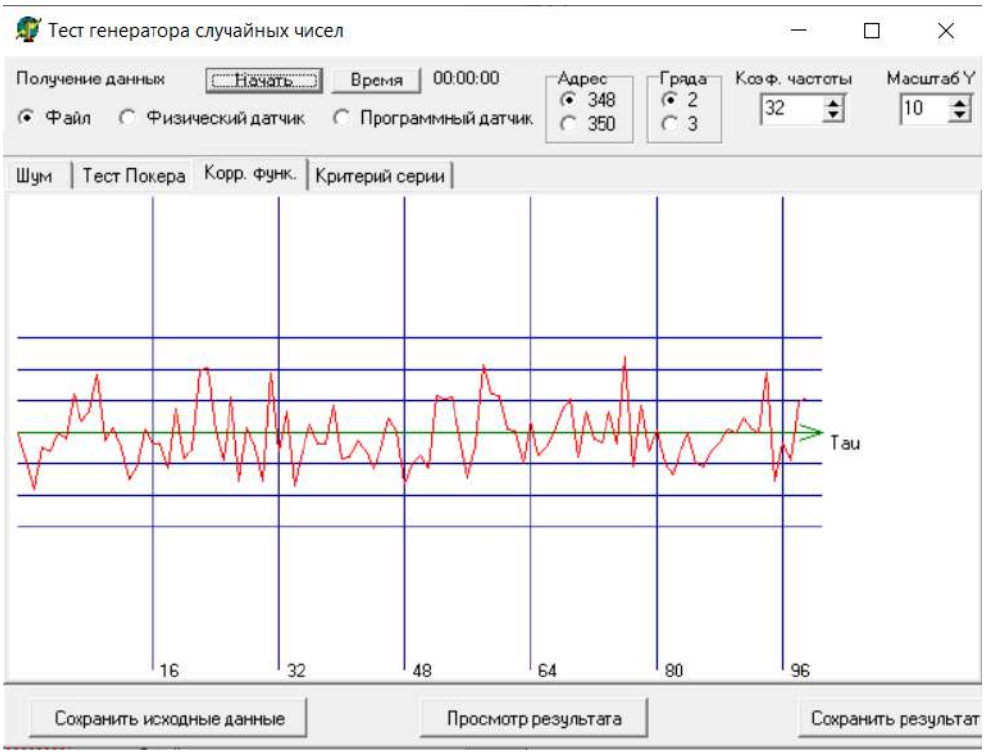

Рисунок 4.5 – Результати автокореляційного теста

1.1 Дослідження системи гарячого резервування датчиків шуму

Використання двох каналів формування випадкових імпульсів (див рис. 1.3) дозволяє збільшити надійність роботи генератора випадкових послідовностей. При непрацездатності одного з каналів формування випадкових рівноімовірних бітових сигналів на входах D6 або D7 плати ARDUINO, тобто при нормальній роботі хоча б одного каналу, – генератор буде нормально працювати, а вихідні бітові послідовності будуть проходити усі статистичні тести.

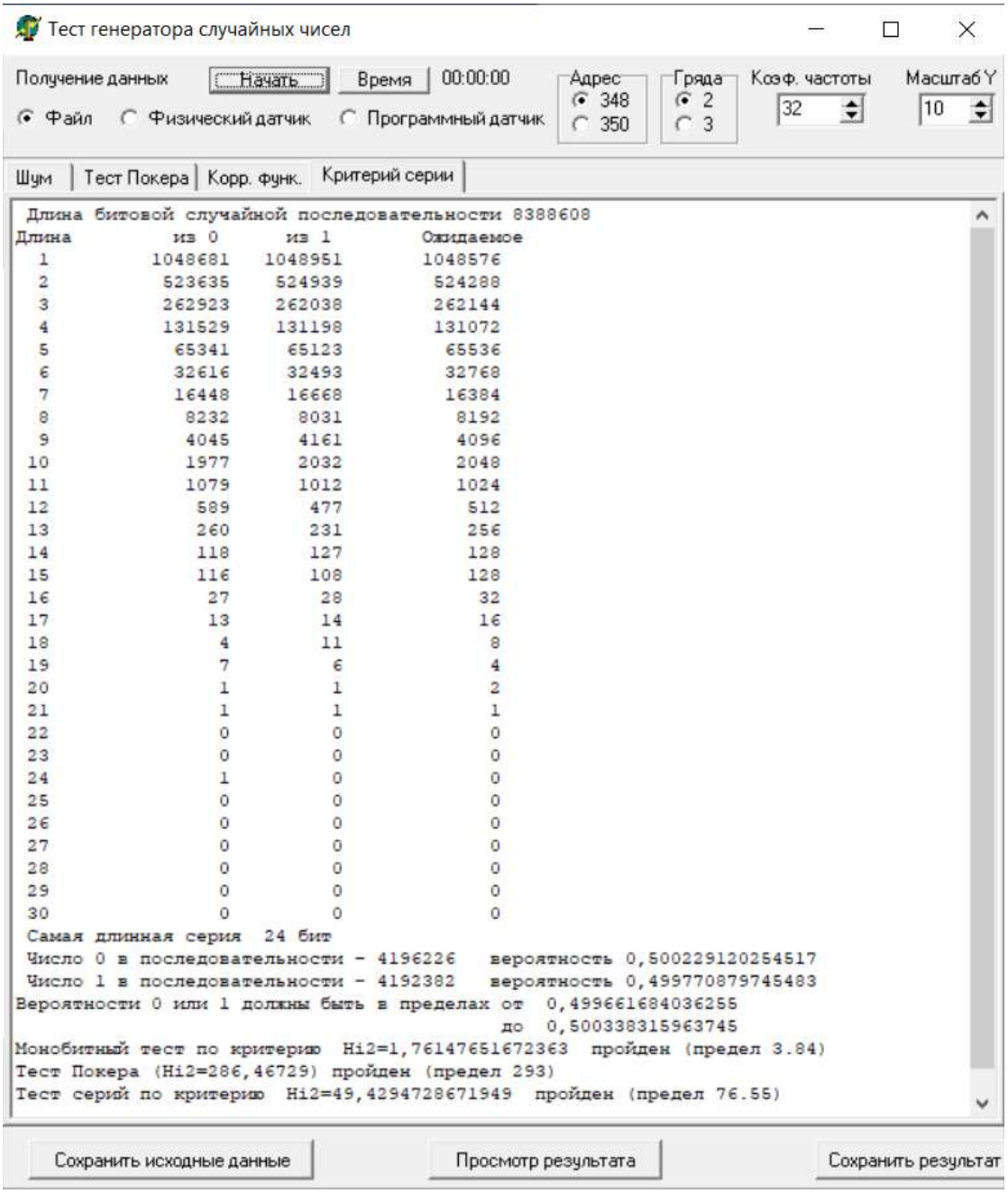

Рисунок 4.6 – Результати монобітного теста і теста серій

Для моделювання несправності в одному з каналів перемикався на коротко один з шумових діодів. На виході цього каналу, тобто на вході D6 або D7 плати ARDUINO (дивись рисунок 2.5) встановлюється постійний логічний рівень – логічний «0», або логічна «1».

Результати тестування вихідних випадкових послідовностей наведені на рисунках 4.7, 4.8, 4.9.

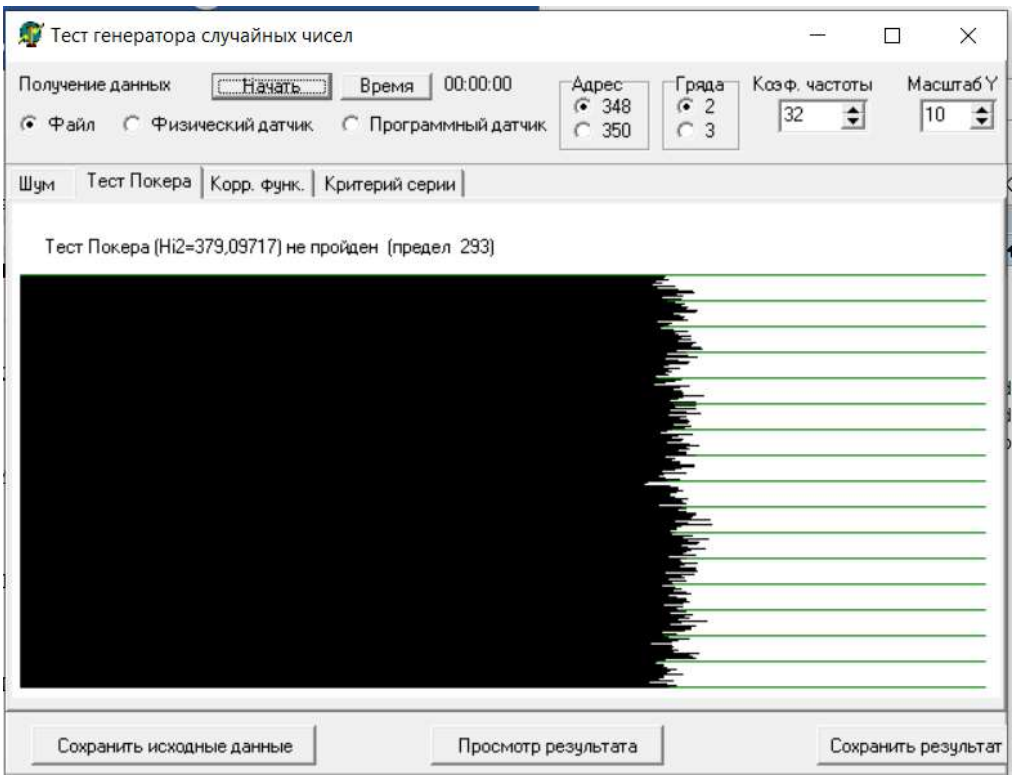

Рисунок 4.7 – Результати теста Покера

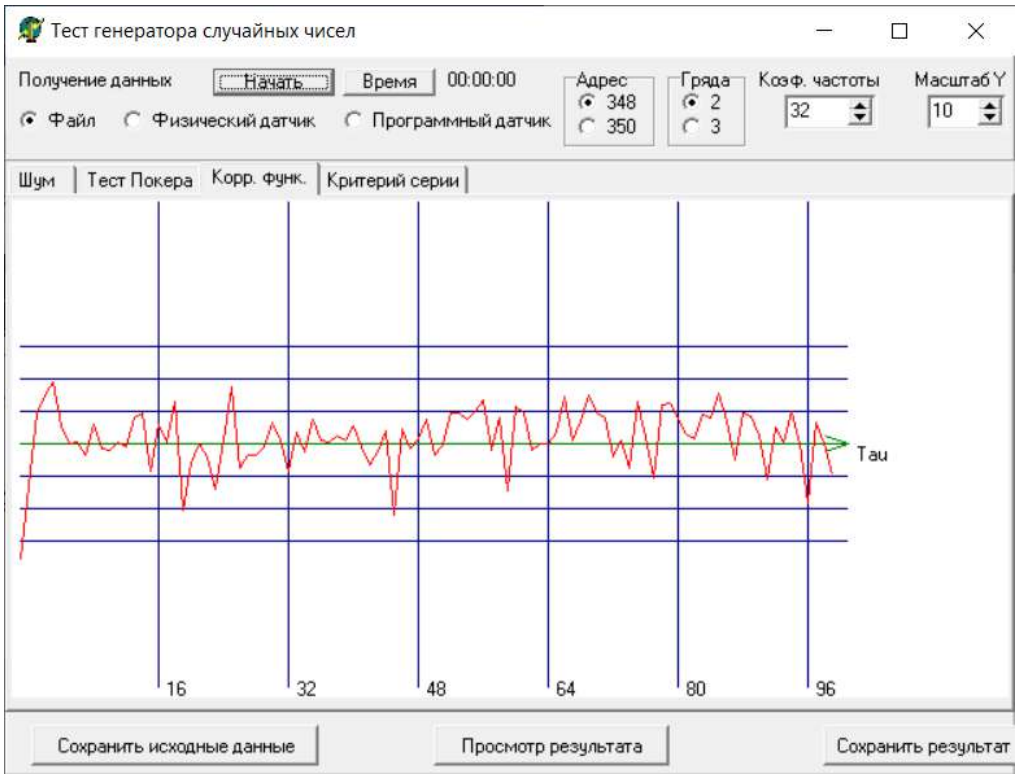

Рисунок 4.8 – Результати автокореляційного теста

Тест Покера (рисунок 4.7) не пройдено, тому що параметр тесту χ^2 = 379, що перевищує максимально припустиме значення 293.

Автокореляційний тест (рисунок 4.8) також не пройдено, тому що 2 значення автокореляційної функції перевищують значення –2σ.

Монобітний тест (рисунок 4.9) пройдено. Тест серій не пройдено. Найдовша серія 24 біта вказує на проходження тесту довгих серій.

Непроходження тесту Покера, автокореляційного теста і теста серій вказують на неприпустимі статистичні зв'язки між сусідніми випадковими бітами, що генеруються.

Це можливо пояснити, аналізуючи рисунок 1.4. Гаряче резервування двох каналів датчиків шуму (рисунок 1.3) збільшує еквіваленту шумову частоту майже вдвічі. При роботі з одним каналом датчика шуму частота *Fш* буде менша. Тому треба зменшувати частоту зчитування випадкових бітів в регістр зсуву F_0.

Для цього треба збільшувати часові затримки, тобто збільшувати константи у регістрі R17. При збільшені констант до 6 і 5 випадкові послідовності, що генеруються проходять усі тести. Можливо обирати ці дві константи однаковими – від 6 до 8.

Аналогічно перевірялась робота другого каналу формування шумових імпульсів (при перемиканні іншого шумового діоду). Генератор випадкових послідовностей проходив усі тести.

Після відновлення роботи обох каналів формування шумових імпульсів генератор працює з гарячим резервуванням (з підвищеною надійністю), а вихідні випадкові послідовності проходять усі тести.

4.2 Оперативне тестування

Ці бітові послідовності також проходять оперативне тестування FIPS-140-1 [5] (або європейський тест AIS-31 [8]):

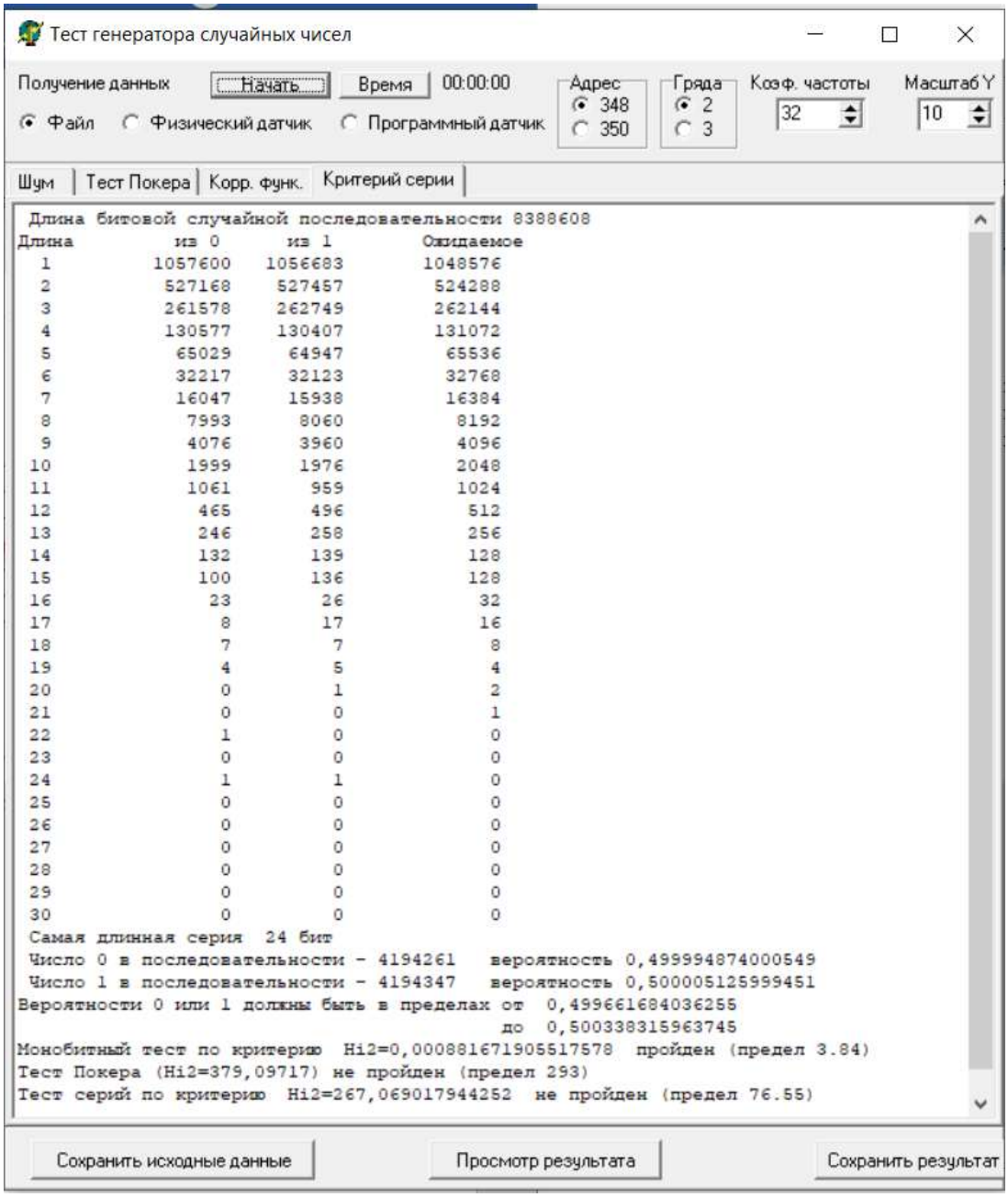

Рисунок 4.9 – Результати монобітного теста і теста серій

Результати тестування статистичного теста FIPS-140-1:

Довжина послідовності, що тестується 20000 біт

Статистистика.

1 Монобітний тест : кількість біт (біт 1) 9889 біт, отримане значення попадає в інтервал 9654...10346 біт, тест пройдено.

2 Покер тест : значення статистики 13,6, отримане значення попадає в інтервал 1.03...57.4,

тест пройдено.

3 Поточний тест : статистика бітові серій (кількість серій), число серій (біт 0 и біт 1):

довжина серії	серій 0	серій 1	теоретичні інтервали
1	2398	2416	2267 - 2733
2	1228	1273	1079 - 1421
3	632	602	502 - 748
4	311	307	223 - 402
5	170	174	90 - 223
6	177	145	90 - 223

число задовільних бітових серій 6 (біт 0),

число задовільних бітових серій 6 (біт 1),

тест пройдено.

4 Поточний тест довжини : значення максимальної довжини серії 14 біт,

отримане значення не перевищує значення 34 біт,

тест пройдено.

Висновок : бітова послідовність прийнята

ВИСНОВКИ

В кваліфікаційній роботі було проведено аналіз існуючих систем генерації випадкових і псевдовипадкових послідовностей для криптографічних систем, вітчизняних та світових стандартів для побудови недермінованих генераторів випадкових послідовностей.

Вибрані та обґрунтовані джерела ентропії (датчики шуму, джерела невизначеності) на основі кремнієвих діодів з Зенеровським пробоєм (стабілітронів).

Розроблена апаратна частина генератора випадкових послідовностей на основі мікроконтролера ATMega 328P. Розроблені та досліджені методи поліпшення статистичних параметрів випадкових послідовностей, що генеруються.

Запропоновано метод підвищення експлуатаційної надійності генератора випадкових послідовностей за рахунок гарячого резервування датчиків шуму (датчиків ентропії) та апаратних каналів формування випадкових імпульсних сигналів

Запропоновано реалізувати генератор випадкових послідовностей у вигляді лабораторного макета на основі апаратного процесорного модуля ARDUINO_NANO. Більшість алгоритмів обробки випадкових імпульсних сигналів реалізовано на програмному рівні.

Проаналізовані методи тестування випадкових послідовностей, що генеруються, малої довжини на основі міжнародних алгоритмів і стандартів. Випадкові послідовності, що не пройшли тестування відкидаються і за припустимо малий час генеруються та тестуються наступні випадкові послідовності

Проаналізовані методи тестування генераторів випадкових послідовностей на основі вітчизняних та міжнародних алгоритмів і стандартів. Не проходження цих тестів вказує на неприпустимі помилки апаратного або програмного забезпечення.

Проведені дослідження лабораторного макета генератора випадкових послідовностей з метою визначення фізичних обмежень на максимальну швидкість формування випадкових бітових послідовностей

Протестовані вихідні випадкові послідовності генератора, запропоновані параметри та коефіцієнти програмної обробки сигналів з метою досягнення максимальної швидкості формування та підвищення надійності генератора.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ISO/IEC 18031:2005. Information technology – Security techniques – Random bit generation (Информационные технологии – Методы защиты – Генерация случайных битов).

2. Методы и средства генерации случайных битовых последовательностей [Текст] : Под ред. д.т.н., проф. Горбенко И.Д. / А.А.Торба, А.А. Бобкова, Ю.И. Горбенко, В.А. Бобух. – Харьков: Изд-во «Форт», 2012.– 232 с.

3. Менезис А., Руководство по прикладной криптографии [Текст] / А. Менезис, П. ван Оршоа, С. Ватсон – CRC: Press, 1996. – 816с.

4. Корт С.С. Теоретические основы защиты информации [Текст] / С.С. Корт – М.: Гелиос АРВ, 2004. – 240 с.

5. FIPS PUB 140-1 Federal Information Processing Standards Publication**.** Cryptographic modules security requirements // NIST, 1993.

6. FIPS PUB 140-2 Federal Information Processing Standards Publication**.** Security requirements for cryptographic modules. NIST, 2001.

7. AIS 20 Application Notes and Interpretation of the Scheme, version 1. Functionality classes and evaluation methodology for deterministic random number generators. Bundesamt fur Sicherheit in der Informationstechnik (BSI). 2001 (Функциональные классы и методология оценки детерминированных генераторов случайных бит) .– Режим доступа : www/ URL: http://www.bsi.bund.de/zertifiz/zert/interpr/ais20e.pdf

8. AIS 31 Application Notes and Interpretation of the Scheme, version 1. Functionality classes and evaluation methodology for physical random number generators. Bundesamt fur Sicherheit in der Informationstechnik (BSI). 2001 (Функциональные классы и методология оценки физических генераторов случайных бит) Режим доступа : www/ URL: http://www.bsi.bund.de/zertifiz/zert/interpr/ais31e.pdf

9. NIST SP 800-22. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. NIST, 2000.

10. Торба А., Дяченко В., Партика С., Пушкар О. Недерміновані генератори випадкових бітів на основі стандарту ISO/IEC 18031:2005 // Improvement of scientific approaches to the development of engineering: collective monograph – Boston.– 2022, p. 232 - 241