УДК 004.056:061.68

Я.Р. Совин, Ю.М. Наконечний, В.М. Чінка, І.Я. Тишик
Національний університет "Львівська політехніка",
кафедра захисту інформації

ТЕСТУВАННЯ ВБУДОВАНОГО ГЕНЕРАТОРА ВИПАДКОВИХ ЧИСЕЛ МІКРОКОНТРОЛЕРІВ РОДИНИ STM32F4XX ЗГІДНО З МЕТОДИКОЮ NIST STS

Проведено тестування вбудованого генератора випадкових чисел мікроконтролерів родини STM32F4XX з ядром ARM Cortex-M4F згідно з методикою NIST STS. Показано, що за результатами тестів NIST STS ці генератори задовольняють вимоги, які ставляться до генераторів випадкових чисел у криптографічних додатках.

Ключові слова: генератори випадкових чисел, тести NIST STS, STM32F4xx.

Testing of hardware random number generator of microcontrollers STM32F4XX family with the kernel ARM Cortex-M4F is conducted in obedience to the method of NIST STS. It is shown that as a result of NIST STS tests these generators satisfy requirements which behave to the random number generators for cryptographic applications.

Key words: random number generators, NIST STS tests, STM32F4xx.

Вступ

Генератори випадкових чисел (ГВЧ) є складовою частиною більшості криптографічних систем. Зокрема, ГВЧ застосовують для генерації ключів у симетричних і асиметричних криптоалгоритмах, вироблення випадкових повідомлень у протоколах автентифікації, побудованих за схемою "запит-відповідь", формування бітів доповнення до потрібного розміру блока, утворення векторів ініціалізації у блокових шифрах та масок для протидії атакам через сторонні канали [1, 2].

Вразливість ГВЧ може скомпрометувати всю криптосистему або значно ослабити її криптостійкість, тому в криптографічних аплікаціях вимоги до якості ГВЧ найвищі.

Ідеальний ГВЧ здатний генерувати випадкові послідовності чисел, які статистично рівномірно розподілені, незалежні, непередбачувані та невідтворювані. Реальні ГВЧ, які використовуються в криптографії, лише певною мірою відповідають вказаним вимогам і відповідно до них поділяються на три базові класи [2]:

• Генератори псевдовипадкових чисел (ГПВЧ). Побудовані на певному детермінованому алгоритмі, який ініціалізується зовнішньо згенерованим випадковим числом – так званим зародком (seed). Відповідно, за однакових значень зародка ГПВЧ завжди генерують однакові послідовності. Для забезпечення високого рівня захищеності ГПВЧ повинні періодично оновлювати значення зародка.

• Криптографічно захищені ГПВЧ (КЗГПВЧ). Основані на ГПВЧ, але алгоритм, призначений для утворення випадкових чисел, унеможливлює в обчислювальному сенсі передбачення наступного значення, навіть якщо відомі сам алгоритм і попередні вихідні дані. З цією метою можуть використовуватися, наприклад, алгоритми гешування або симетричного шифрування.

• Генератори істинно випадкових чисел (ГІВЧ). ГІВЧ використовують або певний фізичний випадковий процес – тепловий шум, фазовий джитер, або певні випадкові явища – дії користувача, вміст ОЗП, сигнал від мікрофонного входу тощо. Враховуючи, що ГІВЧ слугує для генерації зародків у ГПВЧ та КЗГПВЧ, то можна стверджувати, що вони відіграють фундаментальну роль у забезпеченні захищеності всієї криптосистеми.

Фізичні джерела випадкових процесів мають аналогову природу. Проте інтеграція аналогових джерел шуму в цифрові обчислювальні засоби – мікроконтролери чи FPGA збільшує розміри кристала та споживану потужність. Проблемою також є те, що всередині мікросхеми на аналогові кола ГІВЧ впливають близько розташовані цифрові кола та кола живлення, які генерують квазіперіодичні завади набагато вищого рівня. Перенесення аналогової частини ГІВЧ на нову платформу чи технологічний процес потребує значних витрат коштів та часу на редизайн мікро-

схеми. Використання зовнішніх джерел випадкових процесів небажане з погляду захищеності від сторонніх впливів і складності організації інтерфейсу з ними.

З огляду на це у вбудованих системах, особливо на базі мікроконтролерів (МК), переважно використовуються криптографічно слабкі ГПВЧ: різні варіанти лінійного конгруентного методу чи регістрів зсуву зі зворотними зв'язками.

Отже, створення ефективного (з погляду потрібних ресурсів, швидкодії та споживаної потужності) і якісного ГВЧ для вбудованих систем є непростим завданням.

Аналіз останніх досліджень і публікацій

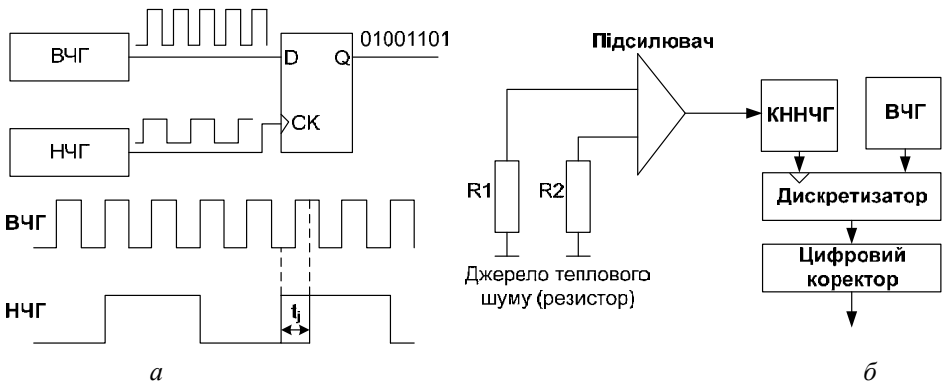
У загальному випадку ГІВЧ складається з фізичного джерела випадкового сигналу, дискретизатора та блока детермінованої постобробки. Джерело випадкового сигналу генерує неперервний аналоговий сигнал (шум), який бінарно оцифровується (наприклад, компаратором). У багатьох випадках отримана випадкова послідовність потребує алгоритмічної постобробки, з метою маскування потенційних статистичних дефектів, що виникають внаслідок обмеженої смуги пропускання, технологічного розкиду параметрів, температурного дрейфу, дій зловмисника тощо. Цілком очевидно, що алгоритмічна постобробка призводить до зменшення продуктивності.

Двома найпоширенішими методами побудови ГІВЧ є підсилення і дискретизація внутрішніх шумів електронних компонентів (резисторів, діодів, стабілітронів) та використання частотної нестабільності несинхронізованих генераторів.

У статті [3] описано типовий ГІВЧ, що використовує комбінацію аналогових і цифрових компонентів. Він складається з двох стабілітронів, які є джерелом білого шуму. Шумовий сигнал підсилюється диференційним підсилювачем та дискретизується за допомогою компаратора та тригера. Оскільки кола підсилення, які повинні підсилити рівень шуму до цифрових логічних рівнів, споживають достатньо багато потужності, мають аналоговий характер та вносять спотворення в сигнал, інтеграція таких ГІВЧ у більшість мікроконтролерів чи FPGA є проблематичною.

Як спробу вирішення цієї проблеми можна вказати роботу [4], де ГІВЧ побудовано на PSoC-мікроконтролері CY8C24x23A, який має розміщені на кристалі програмно конфігуровані аналогові кола (резистори, конденсатори, підсилювачі, компаратори). Проте через високий рівень внутрішніх шумів вбудованих підсилювачів мікроконтролера потрібно використовувати зовнішні широкосмугові прецизійні операційні підсилювачі. Крім збільшення розмірів, вартості та енергоспоживання генератора, це додатково робить його дуже вразливим до зовнішніх впливів.

ГІВЧ на основі цифрових генераторів використовують два незалежні генератори (без стабілізації частоти), відмінні внутрішні шуми яких спричиняють джитер-фази t_J (короткочасні зміщення фронтів у часі), що і є джерелом випадковості [2]. Вихід високочастотного генератора (ВЧГ) дискретизується за зростаючим фронтом сигналу низькочастотного генератора (НЧГ) за допомогою D-тригера (рис. 1, а).



а *б*

Рис. 1. ГІВЧ на основі незалежних генераторів (а) та його вдосконалений варіант (б)

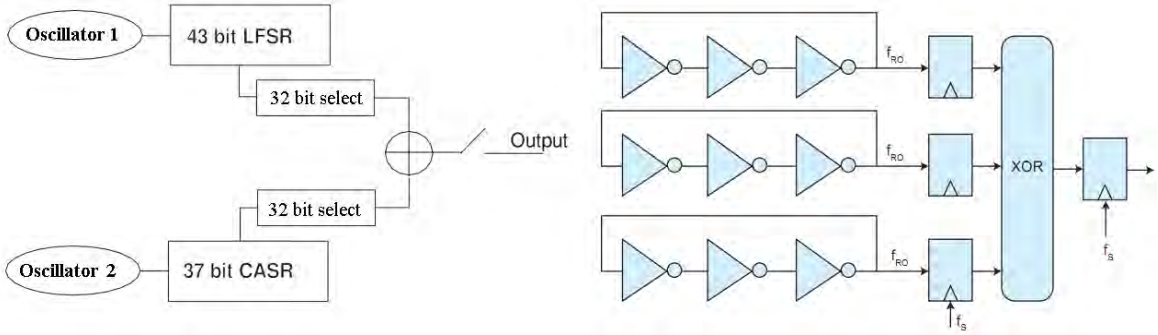
Недоліком такого методу є необхідність для джитера накопичення фази впродовж тривалого часу (оскільки джитер цифрових генераторів достатньо малий), щоб отримати якісні випадкові дані, а це обмежує продуктивність ГІВЧ на рівні 1 Мбіт/с, чого іноді недостатньо для високопродуктивних криптосистем.

Відомо декілька спроб зменшити час накопичення джитера. Наприклад, запропонований фірмою Intel ГІВЧ (рис. 1, б) як додаткове джерело ентропії використовує два диференційно ввімкнених резистори, чий тепловий шум підсилюється і здійснює модуляцію частоти керованого напругою низькочастотного генератора (КННЧГ), сигнал якого слугує для дискретизації виходу незалежного високочастотного генератора (ВЧГ). Вихідна послідовність проходить постобробку з використанням коректора фон Неймана та алгоритму гешування SHA-1 [5]. Проте така архітектура не повністю цифрова, що ускладнює її реалізацію.

Інший підхід до вдосконалення схеми незалежних генераторів полягає в їх використанні для тактування або лінійного регістра зсуву зі зворотними зв'язками (LFSR) та клітинного автомата (CASR) – рис. 2, а [6], або декількох ліній затримок на логічних вентилях, виходи яких об'єднуються за допомогою операції XOR – рис. 2, б [7].

Описані ГІВЧ орієнтовані на FPGA, ASIC, SoC, тобто обчислювальні засоби з програмованою внутрішньою структурою і не придатні для реалізації в готовому мікроконтролері з жорсткою архітектурою.

Для вирішення цієї проблеми деякі виробники останнім часом почали вводити до складу мікроконтролерів апаратні ГІВЧ. Зокрема, у високопродуктивних мікроконтролерах родини STM32F4xx (фірма STMicroelectronics), побудованих на базі універсального ядра ARM Cortex-M4F, з'явився вбудований модуль ГІВЧ, що у поєднанні з апаратною підтримкою основних криптографічних операцій (шифрування – AES/DES/TDES, гешування – MD5/SHA-1/HMAC) робить їх зручною платформою для криптоаплікацій.



а *б*

Рис. 2. ГІВЧ на основі лінійних і клітинних автоматів (а) та лінії затримки (б)

У технічній документації на мікроконтролери родини STM32F4xx відсутня детальна кількісна оцінка якості вбудованого генератора, а лише зазначено, що він забезпечує у 85 % успішне проходження тестів відповідно до стандарту FIPS 140-2 [8], без вказання результатів кожного з тестів та об'єму вибірки.

Відзначимо, що хоча рання версія стандарту FIPS 140-2 і передбачала виконання чотирьох простих тестів (Monobit test, Poker test, Runs test, Long runs test) над випадковими послідовностями довжиною 20000 бітів, проте цієї вимоги в останній чинній версії стандарту не було, а сам набір тестів за сучасними міркам надто примітивний.

Мета статті

Мета роботи – дослідити статистичні характеристики вбудованих ГІВЧ мікроконтролерів загального призначення родини STM32F4xx із 32-бітним ядром ARM Cortex-M4F в різних режимах формування випадкових послідовностей з метою виявлення та усунення можливих недоліків, оцінити перспективи використання генератора у вбудованих криптографічних аплікаціях.

1. Архітектурні особливості вбудованих ГІВЧ мікроконтролерів родини STM32F4xx

Модуль ГІВЧ у мікроконтролерах STM32F4xx побудований на джерелі аналогового шуму і забезпечує генерацію випадкових 32-бітних чисел. Також у ньому передбачені спеціальні кола, які здійснюють онлайн-контроль роботи ГІВЧ та сигналізують про можливі збої, такі як генерація постійних значень або постійної послідовності значень.

Структурну схему апаратного ГІВЧ МК родини STM32F4xx подано на рис. 3 [9].

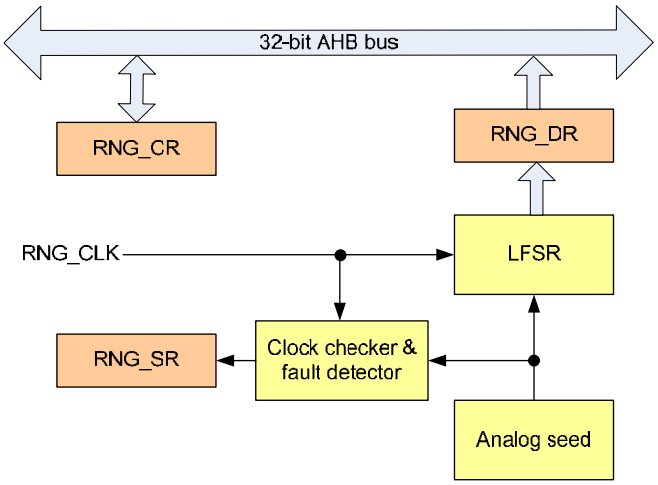


Рис. 3. Структурна схема модуля ГІВЧ у мікроконтролерах родини STM32F4xx

Аналогові кола генерують зародок (Analog seed), що надходить на лінійний регістр зсуву зі зворотними зв'язками (LFSR). Аналогові кола побудовані на незалежних генераторах, чиї виходи об'єднуються операцією XOR. Для тактування LFSR використовується окремий тактовий сигнал (RNG_CLK), який формує спеціальна схема ФАПЧ, тому якість ГІВЧ не залежить від значення основної тактової частоти МК. Генерація одного 32-бітного значення потребує до 40 тактів сигналу RNG_CLK. Максимальне значення RNG_CLK становить 48 МГц.

Коли 32-бітне випадкове число сформоване – воно пересилається у регістр даних (RNG_DR) та встановлюється відповідний прапорець у регістрі статусу (RNG_SR).

Паралельно здійснюється моніторинг тактового сигналу RNG_CLK та зародка. Регістр статусу містить спеціальні прапорці, які сигналізують про атипову послідовність зародків (прапорець SECS) або про те, що тактова частота є занизькою (прапорець CECS). Збоєм вважають дві ситуації: коли згенеровано 64 і більше послідовних бітів з однаковим значенням (0 або 1), або 32 послідовні пари 0 і 1 (0101010101…01). Виявивши збій, ГІВЧ треба перезапустити за допомогою відповідних бітів регістра управління (RNG_CR).

2. Вибір та налаштування апаратних засобів для експериментальних досліджень

Для експериментальних досліджень вбудованого ГІВЧ мікроконтролерів родини STM32F4xx ми вибрали плату STM32F4DISCOVERY, на якій встановлено МК STM32F407VG із 32-бітним ядром ARM Cortex-M4F. Згенеровані послідовності передавалися в USB-порт ПК за допомогою інтерфейсної плати. Генерація випадкових послідовностей здійснювалася за таких параметрів:

– тактова частота ядра мікроконтролера – 168 МГц;
– тактова частота ГІВЧ (RNG_CLK) – 48 МГц;
– напруга живлення мікроконтролера – 3.3 В.

Формування випадкових послідовностей здійснювалося у чотирьох режимах:

– без додаткової постобробки (NONE);
– з об'єднанням двох послідовно згенерованих 32-бітних випадкових значень в одне операцією додавання (ADD);
– з об'єднанням двох послідовно згенерованих 32-бітних випадкових значень в одне операцією сума за модулем 2 (XOR);
– з використанням коректора фон Неймана (NEUM), який функціонує відповідно до табл. 1.

Таблиця 1

Коректор фон Неймана

Вхідні біти	Вихідні біти
00	Відсутні
01	1
10	0
11	Відсутні

Генерацію послідовностей виконано за вимогами стандарту FIPS 140-2, згідно з якими перше вироблене 32-бітне випадкове число не використовується, а кожен наступний вироблений 32-бітний блок даних порівнюється з попереднім, і якщо вони збігаються, то це трактується як збій у роботі ГІВЧ. Додатково в процесі формування випадкових послідовностей відстежувалася частота появи збоїв, що виявляються колами ГІВЧ (прапорці SECS та CECS). Під час налагодження програми та генерації послідовностей ми не виявили жодного з вищевказаних збоїв, що свідчить про надійну і стабільну роботу вбудованого ГІВЧ.

3. Результати тестування криптографічної якості ГІВЧ

Для перевірки якості ГВЧ застосовують різні набори статистичних тестів, серед яких стандартом де-факто є набір тестів (Statistical Test Suite, STS) розроблений National Institute of Standards and Technology (NIST) [10]. Порівняно з іншими відомими наборами тестів тести NIST STS використовують відкриті, детально специфіковані алгоритми, містять контрольні послідовності для перевірки правильності їх реалізації, а також забезпечують однозначну інтерпретацію результатів тестування.

Набір NIST STS складається з 15 окремих статистичних тестів, кожен з яких здійснює перевірку бінарної послідовності на одну з можливих ознак відхилення від випадковості. На підставі результатів тестів приймається (або відхиляється) гіпотеза про те, що ця послідовність є випадковою.

Результатом виконання кожного тесту є так зване *P - value*, яке міститься в діапазоні [0, 1]. Рівень значущості α задає імовірність того, що випадкова послідовність буде сприйнята як невипадкова. Якщо *P - value* $\geq \alpha$, то вважають, що послідовність, яка тестується, пройшла перевірку і є випадковою.

Тестування проводилося за рівня значущості $\alpha = 0.01$, який рекомендований в [10]. Враховуючи, що мінімальна кількість випадкових послідовностей m повинна бути обернено пропорційною до α ($m \geq 100$), у цій роботі вибрано значення $m = 100$. Оскільки деякі тести для отримання коректного результату вимагають, щоб розмір випадкової послідовності n був не меншим за 10^6 біт, ми прийняли $n = 10^6$. Отже, сумарний об'єм вибірки становив 10^9 біт.

Для дослідження статистичних властивостей випадкових послідовностей, згенерованих вбудованим ГІВЧ, ми створили програмне забезпечення в середовищі MatLab, що реалізує набір тестів NIST STS, а також виконує протоколювання та інтерпретацію результатів.

У середовищі IAR 6.30 Embedded Workbench for ARM написано програму для МК STM32F407VG, яка здійснює генерацію вбудованим ГІВЧ 1000 послідовностей, розміром 10^6 біт кожна.

Особливістю тестів № 14–15 (Random Excursions та Random Excursions Variant) є те, що їх результати (*P - value*) будуть достовірними лише тоді, коли для цієї послідовності певний параметр *J*, розрахований за наведеним в [10] алгоритмом, матиме значення, не менше за 500 ($J \geq 500$). Якщо $J < 500$, то це не є ознакою дефекту ГІВЧ, а лише вказує на те, що результати тесту можуть бути недійсними. Тому для коректного виконання тестів № 14-15 окремо генерувалися і відбиралися 1000 послідовностей з $J \geq 500$.

До кожної з 1000 послідовностей застосовувався набір тестів з параметрами, вказаними у табл. 2, при рівні значущості $\alpha = 0.01$.

Для оцінки якості вбудованого ГІВЧ мікроконтролерів STM32F4xx ми дотримувалися рекомендацій щодо інтерпретації результатів, описаних у [10]. Документ [10] передбачає дві стратегії ухвалення рішення про те, чи ГВЧ пройшов тест на випадковість.

Стратегія 1. Ця стратегія для кожного тесту визначає частку послідовностей $P1$, що пройшли перевірку (*P - value* $\geq \alpha$), та порівнює її з нижньою межею довірчого інтервалу $P1_{THR}$.

$$P1 = \frac{\sum_{i=1}^{1000}(P\text{-}value(i) \geq \alpha)}{m}, \quad P1_{THR} = (1-\alpha) - \sqrt{\frac{(1-\alpha)\alpha}{m}} = 0.980561.$$

Якщо хоча б для одного з 15 тестів значення $P1$ виходить за нижню межу довірчого інтервалу ($P1 < P1_{THR}$), то приймають рішення, що ГІВЧ тест на випадковість не пройшов.

Як випливає з даних, наведених у табл. 2 та на рис. 4, вбудований ГІВЧ пройшов перевірку на випадковість для всіх 15 тестів у всіх чотирьох режимах формування вихідних бітів. У режимах формування XOR і NEUM отримано мінімальне значення $P1$, що дорівнює 0.981000, у режимах NONE та

ADD мінімальне значення *P*1 становить 0.982000. Аналізуючи значення *P*1, можна стверджувати, що додаткова постобробка не приводить до помітного покращення результатів проходження тестів.

Таблиця 2

Результати тестування ГІВЧ згідно зі стратегією 1 (при m = 1000 і $n = 10^6$)

№ тесту	Статистичний тест і його параметри	*P1*			
		NONE	*ADD*	*XOR*	*NEUM*
1	**Frequency**	0.996000	0.988000	0.991000	0.984000
2	**Block Frequency** (*M=128)*	0.984000	0.988000	0.987000	0.991000
3	**Runs**	0.990000	0.989000	0.994000	0.994000
4	**Longest Run** (*M=10000)*	0.988000	0.987000	0.990000	0.997000
5	**Rank**	0.994000	0.992000	0.996000	0.993000
6	**DFT**	0.987000	0.987000	0.985000	0.988000
7…154	**Non-Overlapping Template** (*M=9, 148 шаблонів)*	0.990041 (mean)	0.990068 (mean)	0.989581 (mean)	0.990500 (mean)
155	**Overlapping Template** (*M=9*)	0.985000	0.985000	0.988000	0.990000
156	**Linear Complexity** (*M=500*)	0.985000	0.990000	0.989000	0.991000
157	**Universal** (*L=8, Q=2356)*	0.986000	0.986000	0.988000	0.988000
158	**Serial** (*M=16,* $\nabla\psi_m^2$*)*	0.996000	0.986000	0.995000	0.998000
159	**Serial** (*M=16,* $\nabla^2\psi_m^2$*)*	0.995000	0.993000	0.990000	0.991000
160	**Approximate Entropy** (*M=10*)	0.993000	0.986000	0.993000	0.987000
161	**Cumulative Sums** (*Forward*)	0.999000	0.988000	0.988000	0.991000
162	**Cumulative Sums** (*Reverse*)	0.996000	0.989000	0.992000	0.986000
163…170	**Random Excursions** *(x = –4,…,–1, 1,…,4)*	0.992000 (mean)	0.988000 (mean)	0.990250 (mean)	0.990375 (mean)
171…188	**Random Excursions Variant** *(x = –9,…,–1, 1,…,9)*	0.990722 (mean)	0.989833 (mean)	0.991222 (mean)	0.988778 (mean)

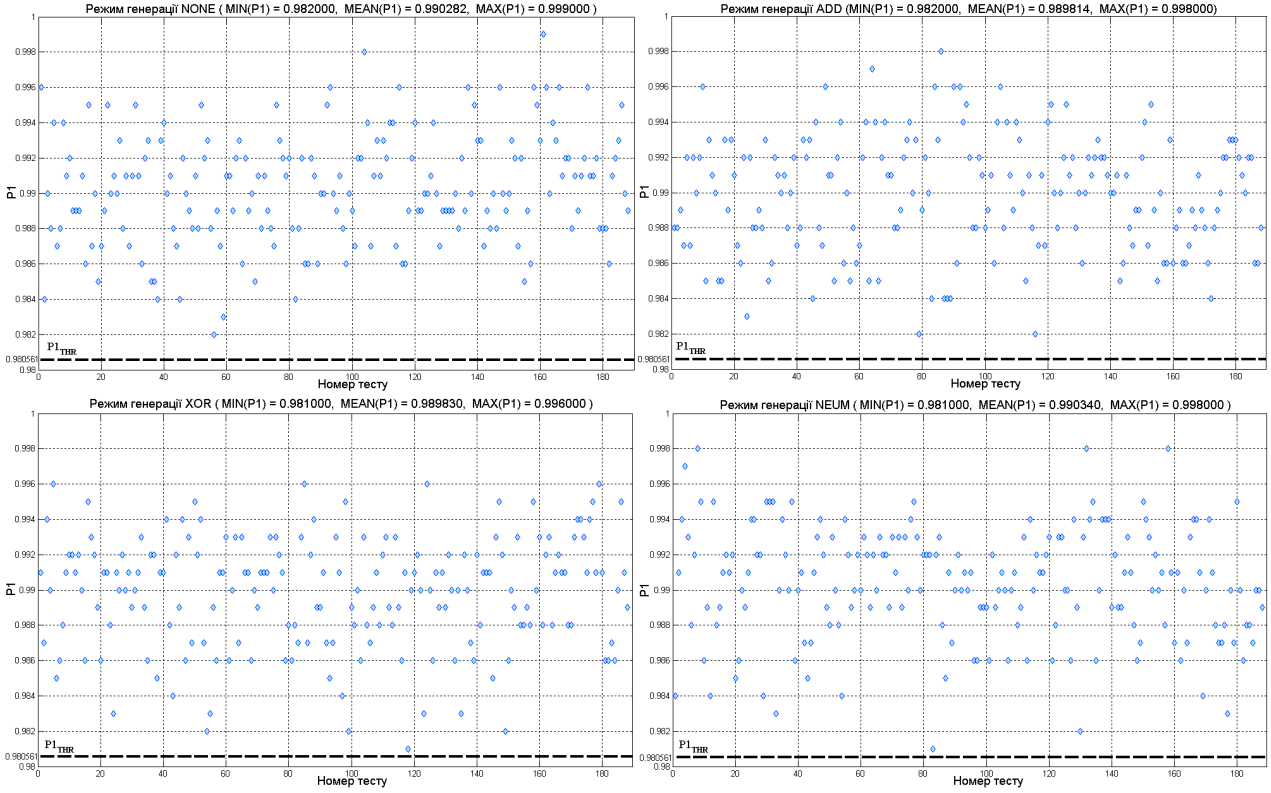


Рис. 4. Результати проходження тестів NIST STS згідно зі стратегією 1

Стратегія 2. Ця стратегія базується на тому, що в якісного ГІВЧ розподіл *P – value* для кожного тесту є рівномірним на інтервалі [0, 1]. Для перевірки цієї гіпотези використано тест χ^2 значень *P – value*, розбитих на 10 підінтервалів *C1-C10*, з кроком 0.1:

$$\chi^2 = \sum_{i=1}^{10} \frac{(Ci - m/10)^2}{m/10}, \qquad P2 = P(\chi^2) = igamc(9/2,\ \chi^2/2)\cdot$$

Якщо отримане в результаті перевірки гіпотези значення $P2 < 0.0001$, то приймається рішення, що ГІВЧ тест не пройшов.

Згідно з даними, наведеними у табл. 3 та на рис. 5, вбудований ГІВЧ пройшов перевірку на випадковість для всіх 15 тестів у всіх чотирьох режимах формування вихідних бітів.

Таблиця 3

Результати тестування ГІВЧ згідно зі стратегією 2 (якщо m = 1000 і $n = 10^6$)

№ тесту	Статистичний тест і його параметри	*P2*			
		NONE	*ADD*	*XOR*	*NEUM*
1	**Frequency**	0.419021	0.100709	0.316052	0.933472
2	**Block Frequency** (*M=128)*	0.992084	0.402962	0.585209	0.552383
3	**Runs**	0.056426	0.904708	0.713641	0.316052
4	**Longest Run** (*M=10000)*	0.467322	0.088226	0.856359	0.053627
5	**Rank**	0.373625	0.620465	0.170922	0.854708
6	**DFT**	0.313041	0.767582	0.536163	0.678686
7…154	**Non-Overlapping Template** (*M=9, 148 шаблонів)*	0.484438 (mean)	0.503837 (mean)	0.511683 (mean)	0.468894 (mean)
155	**Overlapping Template** (*M=9*)	0.007975	0.134944	0.881662	0.263572
156	**Linear Complexity** (*M=500*)	0.916599	0.803720	0.655854	0.512137
157	**Universal** (*L=8, Q=2356)*	0.542228	0.684890	0.715679	0.433590
158	**Serial** (*M=16,* $\nabla\psi_m^2$*)*	0.747898	0.433590	0.159910	0.319084
159	**Serial** (*M=16,* $\nabla^2\psi_m^2$*)*	0.473064	0.581082	0.142872	0.570792
160	**Approximate Entropy** (*M=10*)	0.221317	0.488534	0.546283	0.589341
161	**Cumulative Sums** (*Forward*)	0.947308	0.305599	0.682823	0.257004
162	**Cumulative Sums** (*Reverse*)	0.057146	0.800005	0.402962	0.912724
163…170	**Random Excursions** *(x = –4,…,–1, 1,…,4)*	0.476444 (mean)	0.649775 (mean)	0.393689 (mean)	0.570766 (mean)
171…188	**Random Excursions Variant** *(x = –9,…,–1, 1,…,9)*	0.570220 (mean)	0.475034 (mean)	0.474497 (mean)	0.527368 (mean)

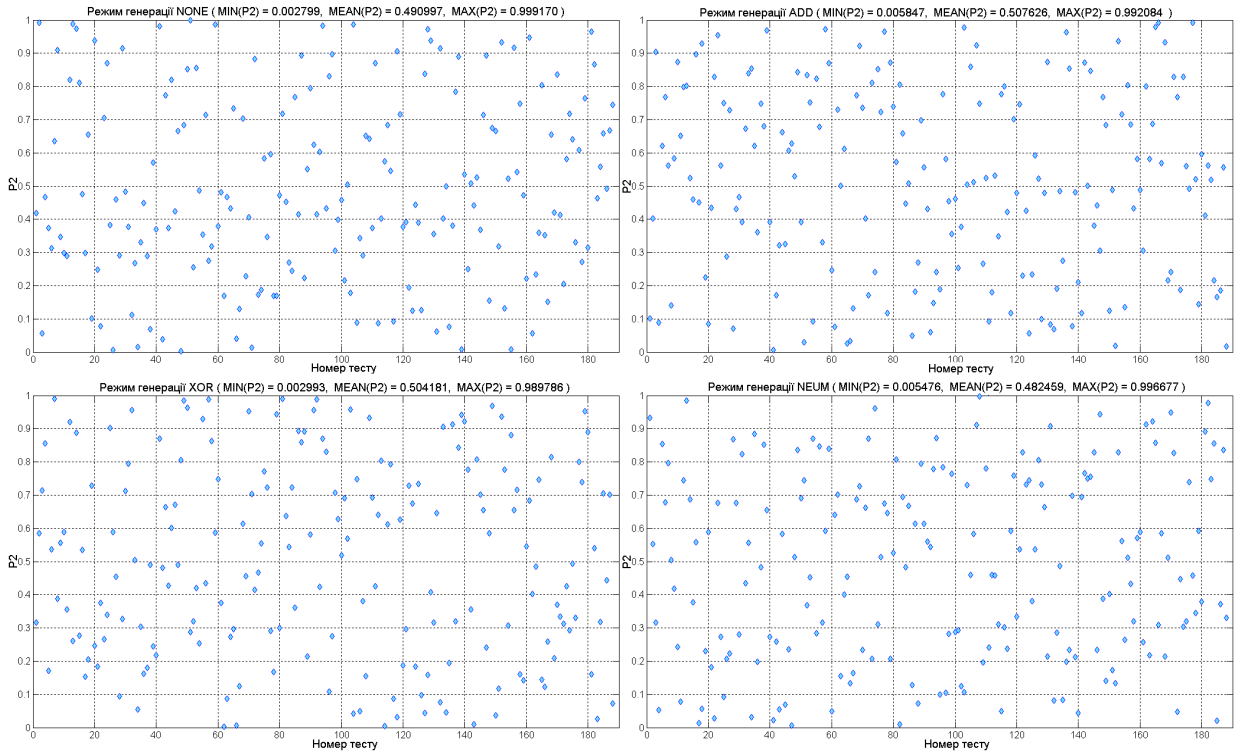


Рис. 5. Результати проходження тестів NIST STS згідно зі стратегією 2

Висновки

За даними тестування NIST STS вбудовані ГІВЧ мікроконтролерів родини STM32F4xx пройшли всі перевірки на випадковість і задовольняють вимоги, які ставляться до ГВЧ у криптографічних додатках. Висока криптостійкість досягається навіть без використання додаткової постобробки випадкових послідовностей.

Висока продуктивність та стабільність роботи вбудованого ГІВЧ у поєднанні з апаратним шифруванням і гешуванням робить мікроконтролери родини STM32F4xx перспективною платформою для широкого спектра криптографічних пристроїв.

1. Secure Integrated Circuits and Systems // Ed. Ingrid M.R. Verbauwhede. – Springer-Verlag, 2010. – 246 p. – ISBN 978-0-387-71827-9. 2. Cryptographic Engineering // Ed. Koc C.-K. – New York: Springer Science+Business Media, 2009. – 522 p. – ISBN: 978-0-387-71816-3. 3. Killmann W., Schindler W.A Design for a Physical RNG with Robust Entropy Estimators // Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'08), 2008, Washington, USA, LNCS, Vol. 5154, pp. 146-163, Springer, Heidelberg (2008). 4. Application Note AN2307. Consumer /Industrial Hardware Random Number Generator // Cypress Semiconductor, 2006, 12 p. 5. Jun B., Kocher P. The Intel Random Number Generator. Cryptography Research, Inc., White Paper prepared for Intel Corporation, 1999, 8 p. 6. Tkacik T. A Hardware Random Number Generator // Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'02), 2002, Redwood Shores, USA, LNCS, Vol. 2523, pp. 450-453, Springer, Heidelberg (2002). 7. Schaumont P. True Random Number Generation // Circuit Cellar, Issue 268, November 2012, pp. 52-58. 8. FIPS PUB 140-2. Security Requirements for Cryptographic Modules // Federal Information Processing Standards Publication 140-2, 2001, 69 p. 9. Reference manual. STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs (RM0090) // STMicroelectronics, 2011, 1316 p. 10. NIST SP 800-22rev1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications // National Institute of Standards and Technology Special Publication 800-22rev1a, 2010, 131 p.