

Я. Р. Совин, Ю. М. Наконечний, М. Ю. Стахів
Національний університет "Львівська політехніка"
кафедра захисту інформації

ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ВБУДОВАНОГО ГЕНЕРАТОРА ВИПАДКОВИХ ЧИСЕЛ МІКРОКОНТРОЛЕРІВ РОДИНИ STM32F4XX ЗГІДНО МЕТОДИКИ NIST STS

© Совин Я. Р., Наконечний Ю. М., Стахів М. Ю., 2013

Проведено тестування вбудованого генератора випадкових чисел мікроконтролерів родини STM32F4XX з ядром ARM Cortex-M4F згідно методики NIST STS при різних значеннях тактової частоти. Показано, що за результатами тестів NIST STS дані генератори задовольняють вимоги, які ставляться до генераторів випадкових чисел у криптографічних додатках.

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

Testing of microcontroller's hardware random number generator of family STM32F4XX with the kernel ARM Cortex-M4F is conducted in obedience to the method of NIST STS at the different values of clock rate. It is shown that as a result of NIST STS tests these generators satisfy requirements which behave to the random number generators in cryptographic applications.

Keywords: random number generators, NIST STS tests, STM32F4xx.

Вступ

Генератори випадкових чисел (ГВЧ) є важливими компонентами більшості криптографічних систем. Функції ГВЧ зводяться до генерації ключів в симетричних і асиметричних криптоалгоритмах, вироблення випадкових повідомлень в протоколах автентифікації побудованих за схемою "запит-відповідь", формування бітів доповнення до потрібного розміру блоку, утворення векторів ініціалізації у блокових шифрах та масок для протидії атакам через сторонні канали [1, 2]. Вразливість в алгоритмі роботи або реалізації ГВЧ може скомпрометувати всю криптосистему, або значно ослабити її криптостійкість, тому в криптографічних аплікаціях вимоги до якості ГВЧ є найвищі.

Ідеальний ГВЧ здатний генерувати випадкові послідовності чисел, які є статистично рівномірно розподілені, незалежні, непередбачувані та невідтворювані. Реальні ГВЧ, що використовуються в криптографії, лише певною мірою відповідають вказаним вимогам і згідно 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, при однакових значеннях зародку ГПВЧ завжди генерують однакові послідовності. Для забезпечення високого рівня захищеності ГПВЧ повинні періодично оновлювати значення зародку.

- Криптографічно захищені ГПВЧ (КЗГПВЧ). Базуються на ГПВЧ, але алгоритм, що служить для утворення випадкових чисел, робить неможливим 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 джиттер або певні випадкові явища – дії користувача, вміст ОЗП, сигнал від мікрофонного входу і т.п.

Оскільки ГІВЧ служать для вироблення зародків у ГПІВЧ та КЗГПІВЧ, то можна стверджувати, що вони відіграють фундаментальну роль в захищеності всієї криптосистеми.

Фізичні джерела випадкових процесів мають аналогову природу. Інтеграція аналогових джерел шуму в цифрові обчислювальні засоби – мікроконтролери чи FPGA, збільшує розміри кристалу 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 значно вищого рівня.

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

Таким чином, створення ефективного (з точки зору потрібних ресурсів, швидкодії та споживаної потужності) і якісного ГІВЧ для вбудованих систем є непростою задачею.

Аналіз останніх досліджень і публікацій

В загальному випадку ГІВЧ складається з фізичного джерела випадкового сигналу, дискретизатора та блоку детермінованої постобробки. Джерело випадкового сигналу генерує неперервний аналоговий сигнал (шум), який бінарно оцифровується (наприклад, компаратором). У багатьох випадках отримана випадкова послідовність піддається алгоритмічній постобробці, з метою маскування потенційних статистичних дефектів, що виникають внаслідок обмеженої смуги пропускання, технологічного розкиду параметрів, температурного дрейфу, дій зловмисника і т.п. Цілком очевидно, що алгоритмічна постобробка приводить 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 до цифрових логічних рівнів та дискретизується з допомогою компаратора та тригера. Оскільки кола підсилення споживають достатньо багато потужності, мають аналоговий характер та вносять спотворення в сигнал інтеграція подібних ГІВЧ у більшість мікроконтролерів чи FPGA є проблематичною.

Тому в роботі [4], ГІВЧ пропонується побудувати на PSoC-мікроконтролері, який має розміщені на кристалі програмно конфігуровані аналогові кола (резистори, конденсатори, підсилювачі, компаратори). Проте високий рівень внутрішніх шумів вбудованих підсилювачів МК вимагає використання зовнішніх широкосмугових прецизійних операційних підсилювачів. 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 джиттер фази t_j ,

(короточасні зміщення фронтів в часі), що і є джерелом випадковості [2]. Вихід високочастотного генератора (ВЧГ) дискретизується за зростаючим фронтом сигналу низькочастотного генератора (НЧГ) з допомогою D-тригера (рис. 1.а).

Недоліком такого методу є необхідність накопичення джиттеру фази впродовж тривалого часу (оскільки джиттер цифрових генераторів є достатньо малий), щоб отримати якісні випадкові дані, а це обмежує продуктивність ГІВЧ на рівні 1 Мбіт/секунда, що може бути недостатнім для високопродуктивних криптосистем.

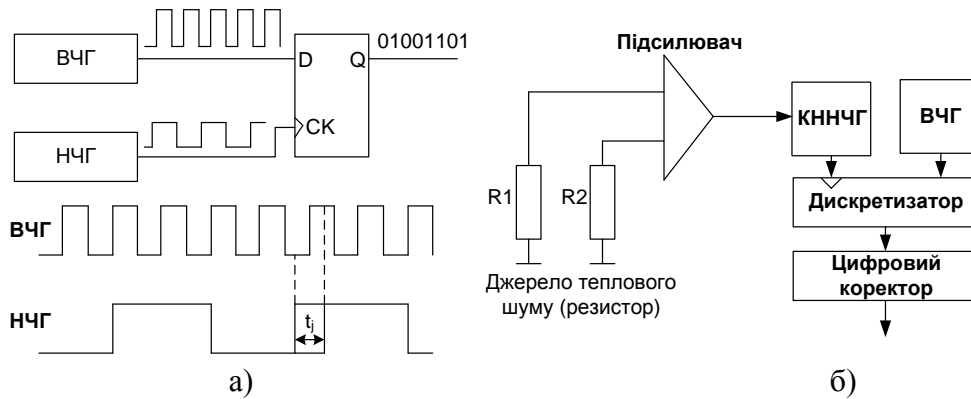


Рис. 1. ГІВЧ на основі незалежних генераторів (а) та його вдосконалений варіант (б)

Відомі декілька спроб зменшити час накопичення джиттеру. Наприклад, запропонований фірмою Intel ГІВЧ (рис. 1.б) в якості додаткового джерела ентропії використовує два диференційно включених резистори, чий тепловий шум підсилюється і здійснює модуляцію частоти керованого напругою низькочастотного генератора (КННЧГ), сигнал якого служить для дискретизації виходу незалежного ВЧГ. Вихідна послідовність піддається постобробці з використанням коректора фон-Неймана та алгоритму гешування SHA-1 [5]. Проте така архітектура не є повністю цифрова, що ускладнює її реалізацію.

Інший підхід до вдосконалення схеми незалежних генераторів полягає в їх використанні для тактування лінійного регістру 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].

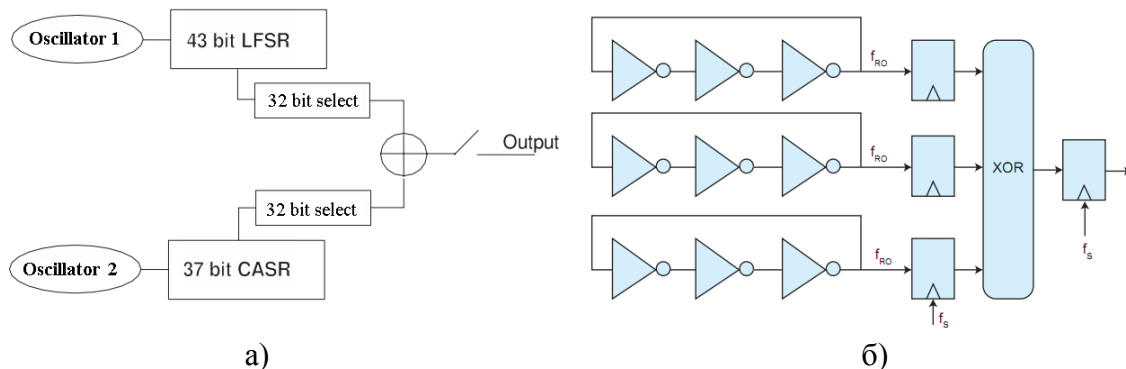


Рис. 2. ГІВЧ на основі лінійних і клітинних автоматів (а) та лінії затримки (б)

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 стали включати 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.

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 згідно стандарту FIPS 140-2 [8]. Слід відзначити, що хоча рання версія стандарту FIPS 140-2 і передбачала виконання чотирьох тестів над випадковими послідовностями довжиною 20000 біт, проте в останній діючій версії стандарту ця вимога відсутня, а сам набір тестів за сучасними міркам є надто примітивним.

Мета статті

Метою даної роботи є дослідити статистичні характеристики вбудованих ГІВЧ мікроконтролерів загального призначення родини STM32F4xx з 32-бітним ядром ARM Cortex-M4F в різних режимах роботи з метою виявлення та усунення можливих недоліків.

1. Архітектурні особливості вбудованих ГІВЧ мікроконтролерів родини STM32F4xx

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

Структурна схема апаратного ГІВЧ МК родини STM32F4xx представлена на рис. 3 [9].

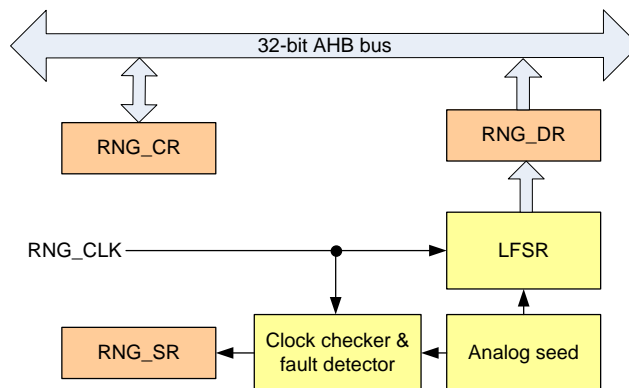


Рис. 3. Структурна схема модуля ГІВЧ в мікроконтролерах родини STM32F4xx

Аналогові кола генерують зародок (*Analog seed*), що поступає на лінійний регістр зсуву зі зворотними зв'язками (*LFSR*). Аналогові кола побудовані на незалежних генераторах, чий виходи об'єднуються операцією XOR. Для тактування *LFSR* використовується окремий тактовий сигнал (*RNG_CLK*), який формується спеціальною схемою 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 в регістр даних (*RNG_DR*) та встановлюється відповідний прапорець в регістрі статусу (*RNG_SR*).

Паралельно здійснюється моніторинг тактового сигналу *RNG_CLK* та зародку. Регістр

статусу містить спеціальні прапорці, які сигналізують про атипову послідовність зародків (прапорець *SECS*) або про те, що тактова частота є заниженою (прапорець *CECS*). За збір приймаються дві ситуації: коли згенеровано 64 і більше послідовних біт з однаковим значенням (0 або 1), або 32 послідовні пари 0 і 1 (0101010101...01). При виявленні збою ГІВЧ слід перезапустити з допомогою відповідних бітів регістра управління (*RNG_CR*).

Таким чином, єдиним програмно керованим параметром, що може впливати на роботу ГІВЧ є значення тактової частоти *RNG_CLK*, проте в технічній документації характер цього впливу не описаний, а тому це питання потребує додаткового вивчення.

2. Вибір та налаштування апаратних засобів для експериментальних досліджень

Для експериментальних досліджень вбудованого ГІВЧ мікроконтролерів родини STM32F4xx нами було обрано плату STM32F4DISCOVERY на якій встановлено МК STM32F407VG з 32-бітним ядром ARM Cortex-M4F. Згенеровані послідовності передавалися в USB-порт ПК для подальшого аналізу. Генерація випадкових послідовностей здійснювалася при таких параметрах:

- тактова частота ГІВЧ (*RNG_CLK*) – 22.4/28.0/37.3/48.0 МГц;
- тактова частота ядра мікроконтролера – 168 МГц;
- напруга живлення мікроконтролера – 3.3 В.

Генерація послідовностей виконувалася за вимогами [8], згідно яких перше вироблене 32-бітне випадкове число не використовується, а кожен наступний вироблений 32-бітний блок даних порівнюється з попереднім і якщо вони співпадають, то це трактується як збір в роботі ГІВЧ. Додатково в процесі формування випадкових послідовностей відстежувалася частота появи збоїв, що виявляються колами ГІВЧ (прапорці *SECS* та *CECS*). Під час відладки програми та генерації послідовностей нами не було виявлено жодного з вищевказаних збоїв, що свідчить про надійну і стабільну роботу вбудованого ГІВЧ.

3. Результати дослідження статистичних характеристик ГІВЧ

Для перевірки якості ГІВЧ застосовуються 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]. У порівнянні 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 дана послідовність є випадковою. Результатом виконання кожного тесту є так зване *P-value*, яке знаходиться в діапазоні [0, 1]. Рівень значущості α задає імовірність того, що випадкова послідовність буде прийнята як не випадкова. Якщо $P\text{-value} \geq \alpha$, то приймається, що послідовність, яка тестується, пройшла перевірку і є випадковою.

Тестування проводилося при рівні значущості $\alpha = 0.01$, який рекомендований в [10]. Враховуючи, що мінімальна кількість випадкових послідовностей m повинна бути обернено пропорційною до α ($m \geq 100$) в даній роботі обрано значення $m = 1000$. Оскільки деякі тести для отримання коректного результату вимагають, щоб розмір випадкової послідовності n був не менше 10^6 біт, нами було прийнято $n = 10^6$. Таким чином, сумарний

об'єм вибірки становив 10^9 біт.

Для дослідження статистичних характеристик випадкових послідовностей згенерованих вбудованим ГІВЧ було створене ПЗ в середовищі MatLab, що реалізує тести NIST STS, а також виконує протоколювання та інтерпретацію результатів. У середовищі IAR 6.30 Embedded Workbench for ARM було написано програму для МК STM32F407VG, яка здійснює генерацію вбудованим ГІВЧ 1000 послідовностей, розміром 10^6 біт кожна.

Особливістю тестів Random Excursions та Random Excursions Variant є те, що їх результати (*P-value*) достовірні лише тоді, коли для даної послідовності параметр J , який розраховується за наведеним в [10] алгоритмом, буде мати значення не менше 500 ($J \geq 500$). Тому для коректного виконання цих тестів окремо генерувалися і відбиралися 1000 послідовностей з $J \geq 500$.

До кожної з 1000 послідовностей застосовувався набір тестів з параметрами вказаними у табл. 1 при рівні значущості $\alpha = 0.01$.

Для оцінки якості вбудованого ГІВЧ мікроконтролерів STM32F4xx ми дотримувалися рекомендацій щодо інтерпретації результатів, описаних в [10]. Документ [10] передбачає дві стратегії ухвалення рішення про те, чи даний ГІВЧ 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 пороговим значенням $P1_{THR}$.

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

Якщо хоча б для одного з 15 тестів значення $P1$ є меншим за поріг ($P1 < P1_{THR}$), то приймається рішення, що ГІВЧ тест на випадковість не пройшов.

Як слідує з даних представлених у табл. 1 та на рис. 4 вбудований ГІВЧ пройшов перевірку на випадковість у всіх чотирьох режимах формування вихідних бітів.

Таблиця 1

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

№ тесту	Статистичний тест і його параметри	<i>P1</i>			
		22.4 МГц	28.0 МГц	37.3 МГц	48.0 МГц
1	Frequency	0.994000	0.989000	0.988000	0.996000
2	Block Frequency ($M=128$)	0.987000	0.990000	0.993000	0.984000
3	Runs	0.987000	0.993000	0.986000	0.990000
4	Longest Run ($M=10000$)	0.986000	0.987000	0.986000	0.988000
5	Rank	0.988000	0.988000	0.995000	0.994000
6	DFT	0.989000	0.992000	0.985000	0.987000
7...154	Non-Overlapping Template ($M=9$, 148 шаблонів)	0.990068 (mean)	0.989649 (mean)	0.989966 (mean)	0.990041 (mean)
155	Overlapping Template ($M=9$)	0.984000	0.989000	0.992000	0.985000
156	Linear Complexity ($M=500$)	0.987000	0.985000	0.986000	0.985000
157	Universal ($L=8$, $Q=2356$)	0.982000	0.988000	0.983000	0.986000
158	Serial ($M=16$, $\nabla \psi_m^2$)	0.989000	0.990000	0.987000	0.996000
159	Serial ($M=16$, $\nabla^2 \psi_m^2$)	0.996000	0.993000	0.987000	0.995000
160	Approximate Entropy ($M=10$)	0.992000	0.985000	0.993000	0.993000

161	Cumulative Sums (Forward)	0.990000	0.990000	0.990000	0.999000
162	Cumulative Sums (Reverse)	0.992000	0.991000	0.987000	0.996000
163...170	Random Excursions ($x = -4, \dots, -1, 1, \dots, 4$)	0.991500 (mean)	0.989000 (mean)	0.989625 (mean)	0.992000 (mean)
171...188	Random Excursions Variant ($x = -9, \dots, -1, 1, \dots, 9$)	0.991000 (mean)	0.990500 (mean)	0.990667 (mean)	0.990722 (mean)

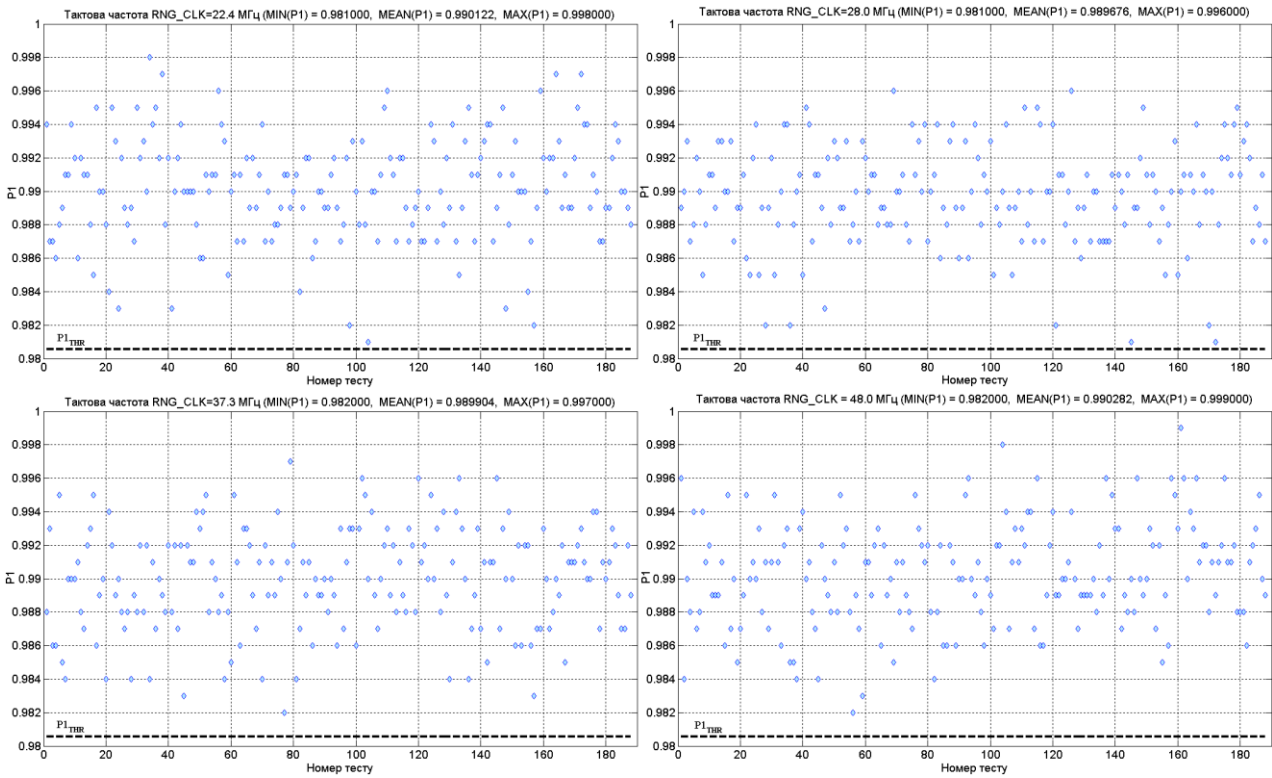


Рис. 4. Результати проходження тестів NIST STS згідно стратегії 1

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

$$\chi^2 = \sum_{i=1}^{10} \frac{(C_i - m/10)^2}{m/10}, \quad P2 = P(\chi^2) = \text{igamc}(9/2, \chi^2/2).$$

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

Згідно даних представлених у табл. 2 та на рис. 5 вбудований ГІВЧ пройшов перевірку на випадковість для всіх 15-ти тестів у всіх чотирьох режимах формування вихідних бітів.

Таблиця 2

Результати тестування ГІВЧ згідно стратегії 2 (при $m = 1000$ і $n = 10^6$)

№ тесту	Статистичний тест і його параметри	$P2$			
		22.4 МГц	28.0 МГц	37.3 МГц	48.0 МГц
1	Frequency	0.040901	0.340858	0.132640	0.419021
2	Block Frequency ($M=128$)	0.227180	0.002236	0.169044	0.992084
3	Runs	0.340858	0.715679	0.207730	0.056426

4	Longest Run ($M=10000$)	0.196920	0.298282	0.433590	0.467322
5	Rank	0.378705	0.668321	0.188601	0.373625
6	DFT	0.123038	0.703417	0.921624	0.313041
7...154	Non-Overlapping Template ($M=9$, 148 шаблонів)	0.503867 (mean)	0.545013 (mean)	0.537733 (mean)	0.484438 (mean)
155	Overlapping Template ($M=9$)	0.142062	0.459717	0.591409	0.007975
156	Linear Complexity ($M=500$)	0.721777	0.512137	0.872425	0.916599
157	Universal ($L=8$, $Q=2356$)	0.239266	0.465415	0.651693	0.542228
158	Serial ($M=16$, $\nabla \psi_m^2$)	0.401199	0.796268	0.599693	0.747898
159	Serial ($M=16$, $\nabla^2 \psi_m^2$)	0.325206	0.122325	0.486588	0.473064
160	Approximate Entropy ($M=10$)	0.326749	0.334538	0.727851	0.221317
161	Cumulative Sums (Forward)	0.805569	0.719747	0.859637	0.947308
162	Cumulative Sums (Reverse)	0.144504	0.605916	0.839507	0.057146
163...170	Random Excursions ($x = -4, \dots, -1, 1, \dots, 4$)	0.558054 (mean)	0.693370 (mean)	0.585901 (mean)	0.476444 (mean)
171...188	Random Excursions Variant ($x = -9, \dots, -1, 1, \dots, 9$)	0.403106 (mean)	0.447080 (mean)	0.517400 (mean)	0.570220 (mean)

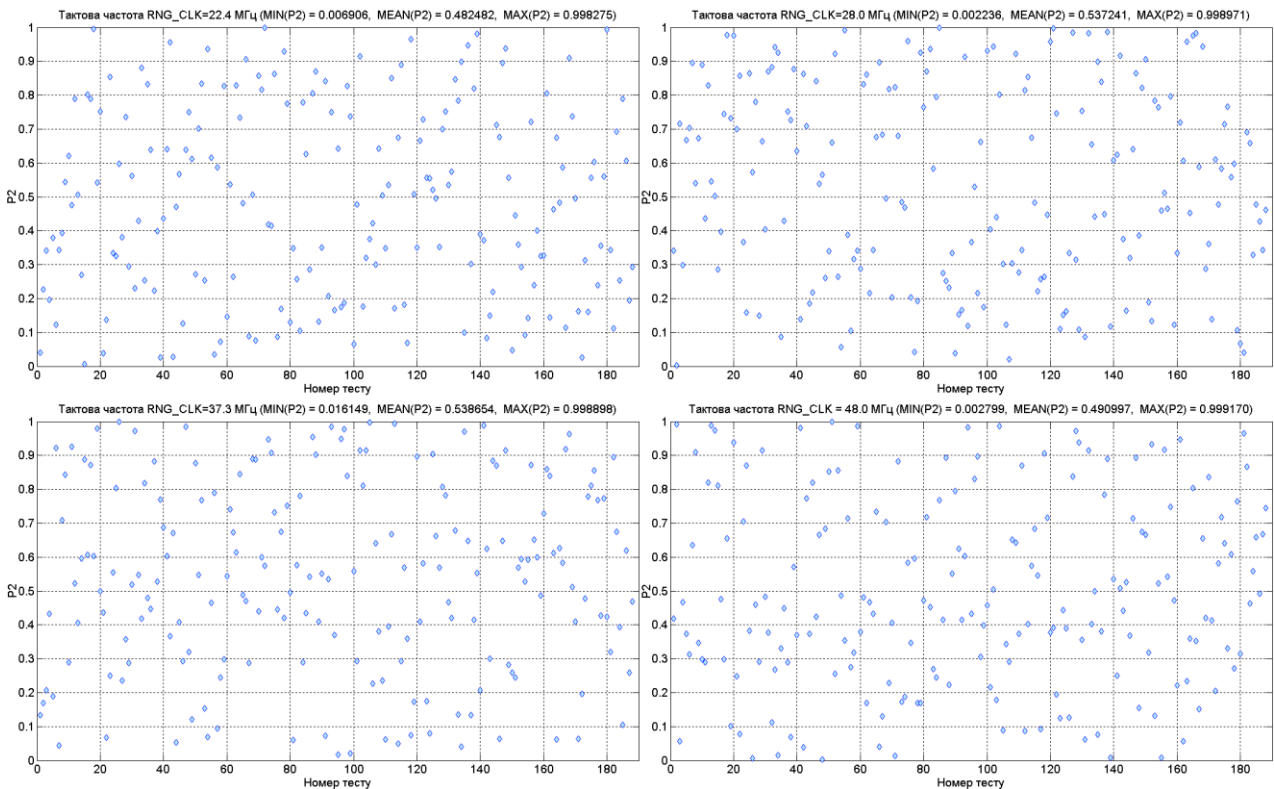


Рис. 5. Результати проходження тестів NIST STS згідно стратегії 2

Висновки

За даними тестування NIST STS вбудовані ГІВЧ мікроконтролерів родини STM32F4xx пройшли всі перевірки на випадковість і задовольняють вимоги, які ставляться до ГВЧ в криптографічних додатках. Висока криптостійкість досягається при різних значеннях тактової частоти генератора, що дає змогу розробнику вибирати її в залежності від вимог

конкретного застосування.

Висока продуктивність та стабільність роботи вбудованого ГІВЧ у поєднанні з апаратним шифруванням і гешуванням робить мікроконтролери родини STM32F4xx перспективною платформою для широкого спектру пристроїв та систем, які потребують використання криптографічних операцій та протоколів.

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.