

# Структурное решение тестового генератора для подсистем встроенного самотестирования цифровых схем

Н.В. Быханова<sup>1</sup>, С.Г. Мосин<sup>2</sup>

<sup>1</sup>ООО «ФС Сервис», [nvc\\_8@list.ru](mailto:nvc_8@list.ru)

<sup>2</sup>Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых, [smosin@ieee.org](mailto:smosin@ieee.org)

**Аннотация** — В данной статье предложено структурное решение для тестового генератора, входящего в состав подсистем встроенного самотестирования цифровых схем. Решение направлено на минимизацию аппаратных затрат и автоматизацию процесса формирования тестового генератора. Его идея заключается в использовании одного генератора тестов для двух способов тестирования: псевдослучайного и детерминированного. Представлены результаты экспериментальных исследований.

**Ключевые слова** — встроенное самотестирование, тестовый генератор, LFSR, покрытие неисправностей.

## I. ВВЕДЕНИЕ

С развитием интегральных технологий и увеличением сложности проектируемых электронных устройств возрастает и сложность процесса тестирования.

Тестирование обеспечивает надёжность и требуемое качество продукции. Вместе с тем оно повышает стоимость изделия и время его выпуска на рынок. Использование методов тестопригодного проектирования и, в частности, встроенного самотестирования позволяет уменьшить эти затраты. Встроенное самотестирование предполагает включение в оригинальный проект специальных тестирующих подсхем. Поиск вариантов реализации таких подсхем, направленных на уменьшение как аппаратных, так и временных затрат при тестировании, является на данный момент актуальной задачей.

В общем случае при построении тестовых подсистем для встроенного самотестирования преследуются цели построения тестового генератора (ТГ) с простой структурой, минимизации площади кристалла, занимаемой тестирующей подсхемой, обеспечение максимального покрытия неисправностей.

Большинство применений сдвиговых регистров с линейными обратными связями (LFSR – Linear Feedback Shift Register) в качестве генератора тестовых наборов предназначено для псевдослучайного тестирования легко определяемых неисправностей [1]-[2]. Для выявления трудно определяемых неисправностей используют детерминированные последовательности

тестовых наборов. Детерминированные тестовые последовательности можно хранить во внутренней памяти схемы, что требует существенного увеличения площади кристалла, или во внешней, что приводит к увеличению числа выводов схемы. Существуют также решения, основанные на использовании LFSR в качестве декомпрессора тестовых последовательностей [1]-[3]. Другие решения предлагают объединить подходы встроенного самотестирования и граничного сканирования – вначале с помощью псевдослучайных последовательностей выявляется часть неисправностей, после этого при граничном сканировании выявляются остальные неисправности [6].

В данной работе ставится задача построения тестового генератора на основе LFSR, совмещающего тестирование детерминированными и псевдослучайными последовательностями, представлен принцип формирования и варианты структуры такого ТГ. Также приводится описание экспериментов, направленных на определение покрытия неисправностей при использовании таких генераторов и определение характеристик ТГ.

## II. ТЕСТОВЫЙ ГЕНЕРАТОР

Реализация тестового генератора основана на применении алгоритма Берликемпа-Мессе для построения линейного сдвигового регистра, генерирующего заданную последовательность. Алгоритм выполняет  $n$  итераций для последовательности длиной  $n$ . Сложность алгоритма составляет  $O(n^2)$  битовых операций [4].

### A. Свойства LFSR

Полученный в результате выполнения алгоритма линейный сдвиговый регистр обладает минимальной длиной.

Разрядность LFSR зависит от двоичной последовательности, применяемой для его построения. Так для последовательности 0000...001 длиной  $n$  разрядов разрядность LFSR равна  $n$ . Для произвольной последова-

тельности длиной  $n$  разрядность LFSR в среднем равна  $\frac{n}{2}$  [4].

Для того, чтобы LFSR генерировал заданную последовательность, необходимо инициализировать его значениями первой части исходной последовательности.

Если степень полинома LFSR равна  $L$ , то данный регистр может генерировать до  $2^L$  наборов разрядностью  $L$ . Длина последовательности, по которой был сформирован LFSR, в среднем равна  $2L$ . Следовательно, оставшиеся  $2^L - 1 - 2L$  наборов могут быть использованы для псевдослучайного тестирования.

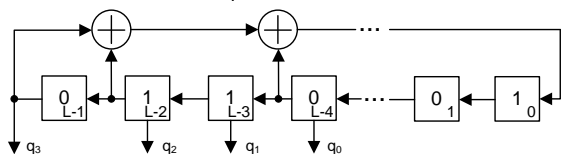
Таким образом, полученный в результате выполнения алгоритма Берликемпа-Мессе LFSR может быть использован для смешанного тестирования, которое совмещает использование детерминированных и псевдослучайных генерируемых тестовых последовательностей без введения дополнительной логики.

### В. Структура тестового генератора

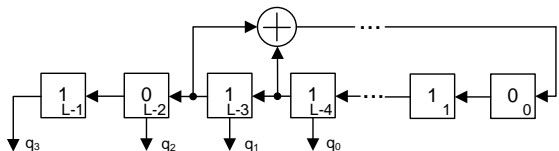
Структура тестового генератора определяется не только сформированным по последовательности линейным сдвиговым регистром, но и способом формирования тестовых наборов. Можно выделить последовательный и параллельный способы [7]-[8].

*Последовательный* – тестовый генератор состоит из одного LFSR, очередной тестовый набор считывается с нескольких старших разрядов регистра (рис. 1). Если разрядность тестового набора равна  $p$ , то для формирования каждого набора требуется  $p$  тактов.

Степень полинома LFSR равна его длине



Степень полинома LFSR меньше его длины на 2



**Рис. 1. Структура последовательного ТГ. Разрядность тестового набора равна 4**

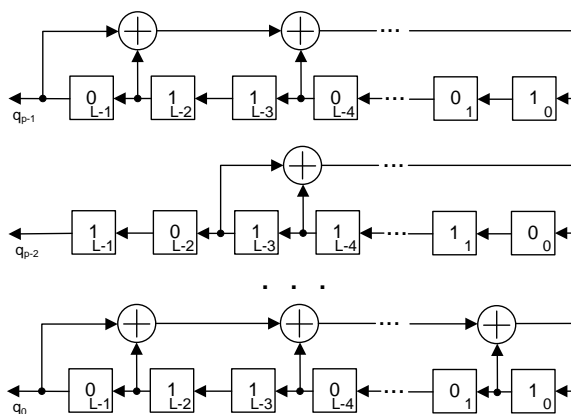
*Параллельный* – тестовый генератор состоит из нескольких LFSR, количество которых равно разрядности тестового набора. В этом случае тестовый набор считывается со старших разрядов регистров и формируется за один такт (рис. 2).

Тестовые генераторы, имеющие любую структуру из указанных, используют в среднем одинаковое коли-

чество элементов памяти. Если разрядность тестового набора равна  $p$ , а длина тестовой последовательности равна  $n$ , то разрядность LFSR будет равна  $\frac{(n \cdot p)}{2}$ . Раз-

рядность каждого LFSR параллельного генератора будет равна  $\frac{n}{2}$ , используется  $p$  модулей LFSR, следова-

тельно, параллельный тестовый генератор требует также  $\frac{(n \cdot p)}{2}$  элементов памяти.



**Рис. 2. Структура параллельного ТГ**

Однако кроме количества тактов, необходимых на формирование тестового набора, указанные решения для одной и той же исходной последовательности тестовых наборов различаются последовательностями, которые используются для построения LFSR тестового генератора и, соответственно, псевдослучайной частью последовательности, генерируемой во время выполнения тестов. Это различие может повлиять на результат тестирования.

Таким образом, оба структурных решения имеют право на существование и могут применяться в зависимости от задачи.

### С. Программная реализация

По алгоритму Берликемпа-Мессе реализована программа на C++, которая позволяет получить конфигурацию тестового генератора по последовательности, а также, используя полученный генератор, сформировать последовательность любой длины.

Входные данные программы представлены длиной входной последовательности, по которой будет сформирована конфигурация тестового генератора, текстовым файлом, содержащим входные последовательности символов '0' и '1' указанной длины, и длиной генерируемой последовательности. Количество строк файла соответствует количеству LFSR в тестовом генераторе. Если задана длина генерируемой последовательности, то после вычисления полинома для каждого LFSR генерируется выходная последовательность, первая часть которой равна входной, а оставшаяся является псевдослучайной.

В зависимости от вида конфигурации тестового генератора (последовательный или параллельный) меняется формат входных последовательностей.

В случае последовательной конфигурации тестовые наборы записывают в строку, начиная со старшего разряда (табл. 1).

Таблица 1

*Линеаризованная тестовая последовательность*

Набор	1	2	...	$n$
Разряды	$p-1, p-2..0$	$p-1, p-2..0$	...	$p-1, p-2..0$
Значения	10..1	11..1	...	00..1

В случае параллельной конфигурации в каждой строке входного файла содержатся значения соответствующих разрядов тестовых наборов (табл. 2).

Таблица 2

*Тестовая последовательность для параллельного ТГ*

Разряды	Номер тестового набора			
	1	2	...	$n$
$p-1$	1	1	...	0
$p-2$	0	1	...	0
...	...	...	...	...
0	1	1	...	1

Выходные данные программы представлены двумя текстовыми файлами. В первом содержатся коэффициенты полинома для всех LFSR, сформированных по входным последовательностям. Каждая строка содержит коэффициенты полинома для одного LFSR. Старший коэффициент полинома находится справа. Младший коэффициент всегда равен единице и означает входение обратной связи в первый разряд LFSR. Во втором файле содержатся сгенерированные с помощью полученных LFSR последовательности, представленные в том же формате, что и входные последовательности.

### III. ЭКСПЕРИМЕНТЫ

В данном разделе описаны эксперименты по определению эффективности смешанного тестирования при использовании разрабатываемого тестового генератора.

#### A. Цель эксперимента

Эксперимент имеет цель показать, что добиться полного покрытия неисправностей можно, построив по части тестовой последовательности генератор, который воспроизводит сначала заданную часть последовательности, а затем псевдослучайную последовательность. Также в рамках эксперимента ставится задача получения значений характеристик подсхемы тестового генератора и по результатам их анализа формирования вывода о пригодности предлагаемого ТГ для тестирования цифровых схем.

Анализируемые характеристики тестового генератора:

- 1) Занимаемая площадь кристалла, которую можно оценить по количеству элементов памяти LFSR и элементов XOR.
- 2) Время тестирования – время выполнения теста до достижения заданного покрытия неисправностей, которое можно оценить по произведению количества приложенных тестовых наборов на количество тактов, необходимых на получение одного набора.

Предполагается, что заданной частью последовательности будут покрыты нужные неисправности (или заданный процент неисправностей).

#### B. Исходные данные эксперимента

Имеется справочник неисправностей для цифровой комбинационной схемы, содержащей 12 входов.

Справочник представляет собой таблицу, где строки соответствуют присутствующим в схеме неисправностям, а столбцы – состояниям входов схемы.

Предполагается, что уже выявлены и удалены избыточные неисправности и каждой двойственной группе или одиночной неисправности соответствует одна строка справочника. Количество строк в справочнике равно 10000. Значения выходов схемы определены для всех 4096 состояний входов.

Каждая ячейка таблицы содержит 0 в случае полного совпадения значений выходов неисправной и исправной схемы и 1 в случае, если отличие присутствует хотя бы в одном разряде.

По данному справочнику неисправностей получена последовательность из 116 тестовых наборов, покрывающих все неисправности. График зависимости покрытия неисправностей от количества применённых тестовых наборов для исходной последовательности представлен на рис. 3.

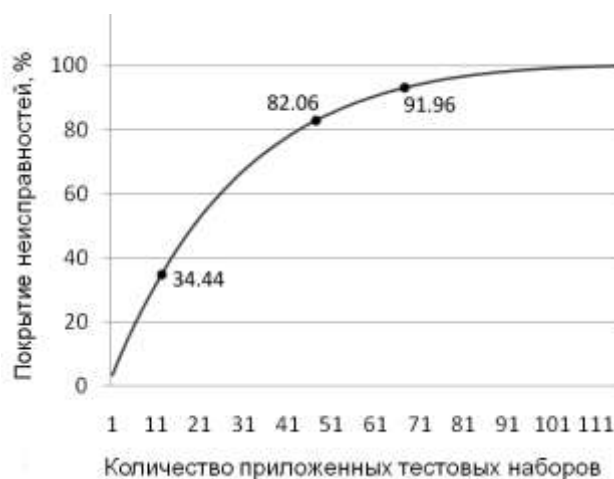


Рис. 3. Покрытие неисправностей для исходной последовательности

Для экспериментов взяты части последовательности, соответствующие первым 12, 46 и 64 наборам. Начальные значения покрытия неисправностей, которые предоставляет детерминированная часть последовательности, отмечены на графике.

Эксперименты были проведены для параллельного тестового генератора.

### С. Результаты эксперимента

Характеристики сформированных по исходной последовательности и её отрезкам тестовых генераторов представлены в табл. 3.

Таблица 3

#### Характеристики тестовых генераторов

Длина последовательности для формирования ТГ	Количество элементов памяти	Количество элементов XOR
Исходная	699	371
12 наборов	77	33
46 наборов	281	131
64 набора	387	207

В табл. 4 приведено количество тактов тестирования, необходимых для достижения полного покрытия неисправностей для тестового генератора, сформированного по исходной последовательности, и тестовых генераторов, сформированных по отрезкам исходной последовательности.

Таблица 4

#### Количество тактов тестирования для достижения полного покрытия неисправностей

Длина последовательности для формирования ТГ	Количество тактов тестирования
Исходная	116
12 наборов	295
46 наборов	352
64 набора	331

Таблица 5

#### Относительные значения характеристик ТГ

Длина последовательности для формирования ТГ	Относительное количество элементов памяти	Относительное количество тактов тестирования
12 наборов	0,1	2,5
46 наборов	0,4	3,0
64 набора	0,6	2,9

В табл. 5 приведены отношения значений количества элементов памяти и количества тактов выполнения теста для тестового генератора, сформированного по исходной последовательности, к значениям количества элементов памяти и количества тактов выполнения теста для тестовых генераторов, сформированных по отрезкам исходной последовательности.

## IV. ВЫВОДЫ

Предложенный тестовый генератор позволяет достичь полной покрытия неисправностей при небольшом объеме тестирующей подсистемы и приемлемом времени выполнения теста.

Эксперименты показали, что при использовании смешанного тестирования увеличивается время выполнения теста по сравнению с временем, необходимым на тестирование детерминированной последовательностью. Однако, возможно определить конфигурации тестового генератора, для которых выигрыш в объеме тестирующей подсистемы существенно превосходит проигрыш во времени выполнения теста. Этот факт становится особенно значимым при больших длинах исходных тестовых последовательностей.

Дальнейшие исследования будут направлены на анализ характеристик двоичных последовательностей с целью выявления зависимостей, применимых для определения оптимальной конфигурации тестового генератора.

## ЛИТЕРАТУРА

- [1] Novak O., Gramatova E., Ubar R. and collective. Handbook of testing electronic systems. Czech Technical University Publishing House. 2005. 395 p.
- [2] Rajski J., Tyszer J. Arithmetic Built-in Self-Test for Embedded Systems. Prentice Hall PTR. 1997. 268 p.
- [3] Wang L.-T., Stroud C.E., Touba N.A. System-on-chip Test Architectures: Nanometer Design for Testability. Morgan Kaufmann, 2008. 856 p.
- [4] Menezes A., P. van Oorschot, Vanstone S. Handbook of Applied Cryptography. CRC Press, 1996. P. 191-212.
- [5] Уколов И.В. Синтез псевдослучайных контролирующих тестов для дискретного устройства // Изв. Саратовского университета. Сер. Математика. Механика. Информатика. 2008. Вып. 1. С. 64-70.
- [6] Lin C.-J., Zorian Y., Bhawmik S. Integration of Partial Scan and Built-In Self-Test // Journal of Electronic Testing: Theory and Applications. 1995. № 7. P. 125-137.
- [7] Mosin S., Chebykina N. A Technique of Optimal Built-In Self-Test Circuitries Generation // Proc. of IEEE East-West Design and Test Symposium (EWDTS'2010). St.-Peterburg, Russia. 2010. P. 145-148.
- [8] Mosin S.G., Chebykina N.V., Serina M.S. Technique of LFSR Based Test Generator Synthesis for Deterministic and Pseudorandom Testing // Proc. of 11th Conference the Experience of Designing and Application of CAD System in Microelectronics (CADSM'11). Polyana-Svalyava, Ukraine. 2011. P. 128-131.