

УДК 004.3

ПРИМЕНЕНИЕ КЛЕТОЧНЫХ АВТОМАТОВ С РАСШИРЕННЫМ НАБОРОМ ПРАВИЛ ДЛЯ ГЕНЕРИРОВАНИЯ ПСЕВДОСЛУЧАЙНЫХ ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

И.А. Мурашко, Д.Е. Храбров

Гомельский государственный технический университет им. П.О. Сухого, Гомель, Беларусь

APPLICATION OF CELLULAR AUTOMATA WITH AN EXPANDED SET OF RULES TO GENERATE PSEUDO-RANDOM TEST SEQUENCES

I.A. Murashko, D.E. Hrabrov

P.O. Sukhoi Gomel State Technical University, Gomel, Belarus

В статье рассматриваются вопросы классификации правил функционирования клеточных автоматов в контексте генерирования псевдослучайной последовательности максимальной возможной длины. Проведен анализ возможных наборов правил, выявлены заведомо неудачные конфигурации. Предложена методика выбора конкретного набора правил для различных ситуаций.

Ключевые слова: псевдослучайная последовательность максимальной длины; генератор псевдослучайной тестовой последовательности; клеточный автомат; циклические граничные условия.

The article deals with the cellular automata rules classification in the context of generating pseudo-random sequence of the maximum possible length. The analysis of the possible sets of rules was successful and wittingly failed configurations were identified. The technique for selecting a specific set of rules for different situations has been proposed.

Keywords: maximum length pseudorandom sequence, pseudorandom test sequence, cellular automata, cyclic boundary conditions.

Введение

Важнейшим элементом встроенного самотестирования (англ. Built-in Self-test, BIST) является генератор псевдослучайных тестовых воздействий [1]. Самым используемым методом генерации тестовых воздействий максимальной длины является регистр сдвига с линейной обратной связью (англ. Linear feedback shift register, LFSR). Основным достоинством LFSR является его изученность и простота аппаратной реализации, для которой требуется лишь регистр сдвига и многовходовой сумматор по модулю два [2]. Однако использование LFSR не всегда оправдано для схем встроенного самотестирования ввиду сильной корреляции между последовательностями, формируемыми на различных разрядах генератора. Так что в последнее время внимание учёных направлено на использование альтернативных методов генерации псевдослучайных последовательностей максимальной длины, так же называемых M-последовательностями. В частности, в качестве генераторов M-последовательностей рассматриваются генераторы на клеточных автоматах (КА) [3], [4].

Наиболее полно исследованы клеточные автоматы на основании правил 90 и 150 с нулевыми граничными условиями [5]. Для них созданы таблицы конфигураций, позволяющих формировать M-последовательности [6].

В последнее время большое внимания уделяется клеточным автоматам с циклическими граничными условиями (ЦГУ). Например, в [7] рассмотрена возможность отражения результатов моделирования КА относительно временной оси. Проблема обратимости линейных клеточных автоматов с нулевыми граничными условиями была подробно рассмотрена в [4], однако только в двоичном поле Галуа GF(2). Возможность обращения генерирования последовательности важна, так как иначе предыдущую последовательность можно узнать только прибегнув к полному моделированию работы клеточного автомата. Подробно описание работ по обратимым клеточным автоматам можно найти в [8].

В статье рассматриваются вопросы классификации правил функционирования клеточных автоматов в контексте генерирования псевдослучайной последовательности максимальной возможной длины. Проведен анализ возможных наборов правил, выявлены заведомо неудачные конфигурации. Предложена методика выбора конкретного набора правил для различных ситуаций.

1 Клеточные автоматы

В общем случае клеточный автомат может быть рассмотрен как простая модель пространственно протяжённого устройства, состоящего из ряда ячеек. Связи между ячейками ограничены

локальным взаимодействием, то есть каждая ячейка находится в каком-либо состоянии, которое изменяется с течением времени в зависимости от предыдущего значения самой ячейки и значений её ближайших соседей. В данной работе рассматриваются только одномерные клеточные автоматы. Если на крайние ячейки автомата постоянно подаётся логический ноль, то такие граничные условия называются нулевыми. Если автомат «завёрнут» в кольцо (то есть последняя ячейка связана с первой и наоборот), то такие граничные условия называются циклическими.

В данной работе использован набор правил: 0, 170, 204, 102, 240, 90, 60, 150. Эти 8 правил представляют из себя все вариации одной клетки и двух соседей при использовании только сумматоров по модулю два. Особое значение здесь играют правила 0 и 204, при их использовании клетка никогда не поменяет своё значение, что фактически означает превращение клеточного автомата из циклического в линейный автомат с фиксированными граничными условиями.

Расширенный набор правил можно записать в виде битовой таблицы (таблица 1.1). В этой таблице первая строка – используемые правилом значения. Например, 011 в первой строке обозначает, что следующее значение будет вычисляться при помощи суммирования предыдущего значения клетки и значения правого соседа. Первый столбец – предыдущие значения левого соседа, ячейки и правого соседа. Крайний правый столбец – «вес» текущего значения. Последняя строка – название правила, которое является суммой значений (основная часть таблицы), умноженных на вес.

Таблица 1.1 – Битовая таблица расширенного набора правил

	000	001	010	011	100	101	110	111	
000	0	0	0	0	0	0	0	0	1
001	0	1	0	1	0	1	0	1	2
010	0	0	1	1	0	0	1	1	4
011	0	1	1	0	0	1	1	0	8
100	0	0	0	0	1	1	1	1	16
101	0	1	0	1	1	0	1	0	32
110	0	0	1	1	1	1	0	0	64
111	0	1	1	0	1	0	0	1	128
Правило:	0	170	204	102	240	90	60	150	

Таблица 1.3 – Примеры обращения наборов правил

Полином	Набор правил	Обратный набор	Полином
$1+x^3+x^5$	[240, 240, 240, 60, 60]	[170, 170, 170, 102, 102]	$1+x^3+x^5$
$1+x^3+x^5$	[240, 60, 60, 90, 90]	[170, 102, 102, 90, 90]	$1+x+x^2+x^3+x^5$
$1+x^3+x^5$	[240, 150, 60, 60, 60]	[170, 150, 102, 102, 102]	$x^2+x^3+x^5$
$1+x^2+x^5$	[240, 150, 60, 60, 150]	[170, 150, 102, 102, 150]	$1+x^2+x^5$
$1+x^2+x^5$	[90, 60, 60, 150, 60]	[90, 102, 102, 150, 102]	$1+x^2+x^5$
$1+x^3+x^5$	[240, 90, 150, 240, 60]	[170, 90, 150, 170, 102]	$1+x+x^2+x^3+x^5$
$1+x^3+x^5$	[240, 240, 150, 60, 90]	[170, 170, 150, 102, 90]	$1+x+x^3+x^5$

В первой строке таблицы 1.1 можно увидеть закономерность: если есть правило 100, то будет правило 001. Назовём это инвертированием правила, или обращением. На рисунке 1.1 показана общая идея обращения правил клеточного автомата: если использовалось значение только правого соседа, то после инвертирования будет использоваться значение только левого.

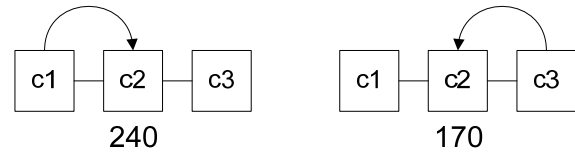


Рисунок 1.1 – Иллюстрация процесса инвертирования (обращения)

В таблице 1.2 показан полный расширенный набор и его обращение. Цветом в таблице отмечены правила 0 и 204, так как используют только значение самой клетки и по сути сводят циклический клеточный автомат к автомату с нулевыми граничными условиями, но меньшей размерности. Столбцы «зависимость» и «обратная» обозначают, используются ли в данном правиле значения левого соседа, самой клетки, и правого соседа.

Таблица 1.2 – Расширенный набор правил и его обращение

Правило	0	170	204	102	240	90	60	150
Прямая зависимость	000	001	010	011	100	101	110	111
Обратная зависимость	000	100	010	110	001	101	011	111
Обратное правило	0	240	204	60	170	90	102	150

В таблице 1.3 можно увидеть практическое применения обращения правил. Как видно из первых трёх строк, при обращении правил одного и того же примитивного полинома может получиться тот же самый полином, другой примитивный или другой не примитивный. Причём, как видно из строк 3 и 4, замена всего одной ячейки может существенно изменить результат.

В таблице 1.4 представлены примеры корректных векторов правил, построенных на двух правилах. Вектора собраны по парам – прямой и инвертированный. Инвертированный набор правил может быть построен в соответствии с таблицей 1.2. Следует заметить, что всего было найдено 3 пары правил, на которых можно построить удачные конфигурации: 60 и 240, 60 и 150, 90 и 240. Также корректные вектора можно построить на обратных к ним правилам: 102 и 170, 102 и 150, 90 и 170.

Таблица 1.4 – Корректные вектора, построенные на 2 правилах

Полином	Вектор правил								П1	П2
	240	240	240	240	240	240	60	60		
$1+x^6+x^7$	240	240	240	240	240	240	60	60	240	
$1+x^6+x^7$	170	170	170	170	170	170	102	102	170	
$1+x^6+x^7$	60	150	60	150	150	150	150	60	150	
$1+x^6+x^7$	102	150	102	150	150	150	150	102	150	
$1+x^3+x^7$	240	90	240	90	240	90	90	90	240	
$1+x^3+x^7$	170	90	170	90	170	90	90	90	170	

В таблице 1.4 приведены примеры конфигураций, при обращении дающих тот же самый порождающий полином.

2 Правила функционирования клеточных автоматов

В работе [9] в генераторах на клеточных автоматах с циклическими граничными условиями предложено использовать 8 правил: 0, 170, 204, 102, 240, 90, 60, 150. Там же предложено разделить правила на два класса: сдвигающие и суммирующие. Сдвигающие правила имеют направленность, т. е. кроме значения ячеек они могут брать значения либо из левого, либо из правого соседа. Одновременное использование или неиспользование значений соседей является отличительной особенностью суммирующих правил: 0, 204, 90 и 150. Остальные правила являются сдвигающими, что рассмотрено в таблице 2.1.

Таблица 2.1 – Виды правил

	Суммирующие				Сдвигающие			
Правило:	0	204	90	150	60	102	240	170
Значения:	000	010	101	111	110	011	100	001
τ :	0	0	1	2	1	1	0	0

В строке «значения» таблицы 2.1 показано бинарное представление используемых правил значений: левого соседа, предыдущее значение ячейки и правого соседа. Строка « τ » означает количество двухходовых сумматоров, необходимых для функционирования данного правила.

Отдельное место в этом наборе занимают правила 0 и 204, так как по сути сводят циклический клеточный автомат к автомату с нулевыми граничными условиями, но меньшей размерности. Это показано на рисунке 2.1.

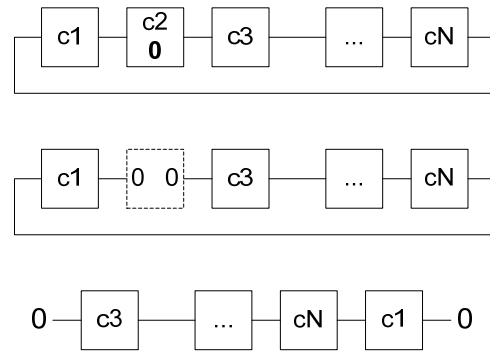


Рисунок 2.1 – Использование правила 0 в клеточном автомате с циклическими граничными условиями

В данной работе исследуется проблема классификации наборов правил КА, на которых генератор сможет выдавать последовательность максимальной длины. По рисунку 2.1 видно, что при использовании правила 0 клеточный автомат фактически будет уменьшен на степень (то есть иметь размерность $n-1$), соответственно такой автомат не сможет генерировать последовательность длины 2^n-1 , только $2^{n-1}-1$. Аналогичные результаты получаются для правила 204, которое формирует единичные граничные условия. Как следствие, для рассмотрения остались 6 правил: 170, 102, 240, 90, 60, 150.

Для классификации было проведено моделирование всех конфигураций клеточных автоматов на различных наборах правил. Для каждой из степеней были найдены все удачные конфигурации. В таблице 2.2 рассмотрены все возможные конфигурации при использовании только двух правил.

По аналогии были найдены конфигурации из трёх и четырёх правил, которые позволяют генерировать M-последовательность: 60, 90, 240 / 102, 90, 170; 60, 90, 150 / 102, 90, 150; 90, 150, 240 / 90, 150, 170; 60, 150, 240 / 102, 150, 170; 60, 90, 150, 240 / 90, 102, 150, 170.

Для пар правил авторами предлагается следующая классификация:

– Дающие корректный примитивный характеристический полином.

1. Прямые: 60–240 (110–100), 60–150 (101–111), 90–240 (101–100).

2. Инвертированные: 102–170 (011–001), 102–150 (011–111), 90–170 (101–001).

– Не дающие корректный полином.

1. Симметричные сами себе: 60–102 (110–011), 90–150 (101–111), 170–240 (001–100)

2. Сдвигающие, противоположно направленные: 60–170 (110–001), 102–240 (011–100)

3. Другие: 60–90 (110–101), 90–102 (101–011), 150–170 (111–001), 150–240 (111–100)

Таблица 2.2 – Возможные пары правил

№	Правила	Битовые	Комментарий
1	60–90	(110–101)	удачных конфигураций найдено не было
2	60–102	(110–011)	удачных нет, правила симметричны
3	60–150	(110–111)	примеры удачной конфигурации: $1+x^2+x^3$ [60, 60, 150]
4	60–170	(110–001)	удачных конфигураций найдено не было
5	60–240	(110–100)	примеры: $1+x^3+x^4$ [240, 240, 240, 60], $1+x^4+x^5+x^6+x^7$ [240, 240, 60, 240, 60, 240, 60]
6	90–102	(101–011)	удачных нет, набор обратный к (60–90)
7	90–150	(101–111)	удачных нет, правила симметричны
8	90–170	(101–001)	$1+x+x^3$ [170, 170, 90], $1+x^3+x^7$ [170, 90, 170, 90, 170, 90, 90]
9	90–240	(101–100)	$1+x^3+x^5$ [240, 90, 240, 90, 90], $1+x^3+x^7$ [240, 90, 240, 90, 240, 90, 90]
10	102–150	(011–111)	$1+x^2+x^3+x^4+x^5$ [102, 150, 102, 150, 150], $1+x^6+x^7$ [102, 102, 150, 102, 150, 102, 150]
11	102–170	(011–001)	$1+x^2+x^3$ [170, 170, 102], $1+x^5+x^6$ [170, 170, 170, 170, 170, 102]
12	102–240	(011–100)	(60–170), удачных нет
13	150–170	(111–001)	удачных нет
14	150–240	(111–100)	(150–170)
15	170–240	(001–100)	удачных нет, правила симметричны

Как видно из классификации, корректные порождающие вектора могут давать наборы как чисто сдвигающих правил (60–240), так и комбинации сдвигающих с суммирующими (60–150). Инвертировав правила из корректного набора так же получим правила, дающие корректные порождающие вектора: 60–240 и 102–170, 60–150 и 102–150.

В ходе исследования было выявлено, что при наличии в наборе противоположно направленных правил (рисунок 2.2) невозможно получить корректную конфигурацию. На рисунке знаком «>» обозначены правила, сдвигающие значение вправо, «<» – влево, «+» – суммирующие правила, «0» – ячейка, в которую собираются все значения.

>>>>0<<<

Рисунок 2.2 – Противоположная направленность правил

Из рисунка видно, что левая и правая части автомата по сути сдвигают все свои значения в ячейку «0». Так как значения оттуда не берутся, то получается «дыра», место куда значения уходят. В то же время крайняя левая и крайняя правая ячейки новых значений ниоткуда не берут, в не зависимости от граничных условий. Даже если на начальном этапе в этих ячейках было значение 1, то уже через 2 хода это значение будет перемещено далее и более никогда не появится.

3 Классификация наборов правил

При построении генератора на клеточных автоматах могут быть использованы различные

комбинации правил, однако далеко не все полученные генераторы смогут выдавать последовательность максимальной длины. Классификацию генераторов на клеточных автоматах по количеству используемых правил можно увидеть на рисунке 3.1.



Рисунок 3.1 – Классификация генераторов на клеточных автоматах по количеству используемых правил

В таблице 3.1 собраны сведения о неинвертированных наборах правил, генераторы на которых могут выдавать последовательность максимальной длины.

Исследования показали, что примитивные полиномы конкретной степени существуют не для всех пар правил. Например, столбец «E» отвечает за ошибку (Error) – то есть, если для определенной степени не было найдено ни одного вектора правил на данном наборе и дающего M-последовательность, то такое состояние называется ошибкой. «Ошибка в» показывает, для каких именно степеней нет векторов на данном наборе правил.

Таблица 3.1 – Выбор набора правил

Набор	E	Ошибка в	I	P	A_{sp}	A_{min}	A_{max}	L_{min}	L_{max}	L_{avg}
60, 150	6	4, 6, 8, 10, 12, 14	12	12	1,5	1,13	1,8	0,3	0,85	0,57
90, 240	6	4, 6, 8, 10, 12, 14	12	12	0,5	0,13	0,86			
60, 240	4	8, 12, 13, 14	11	11	0,5	0,06	0,93			
60, 90, 240	0	–	6	10	0,6	0,21	0,92			
60, 90, 150	0	–	6	10	1,3	1,07	1,76			
60, 150, 240	0	–	9	9	1	0,3	1,61			
90, 150, 240	0	–	6	8	1	0,23	1,61			
60, 90, 150, 240	0	–	0	0	1	0,38	1,61			

Столбец « I » – количество не пройденных тестов на неприводимость (Irreducibility). Тест на неприводимость считается не пройденным, если были найдены не все неприводимые характеристические полиномы для данной степени. « P » – количество проваленных тестов на примитивность (Primitiveness). Все примитивные полиномы неприводимы, обратное неверно [10]. Однако тест на неприводимость имеет гораздо меньшую сложность, чем тест на примитивность. Поэтому сначала используется достаточно быстрый тест на неприводимость [11] и в случае его прохождения выполняется тест на примитивность. Столбец « A_{sp} » отвечает за удельные аппаратные затраты, считая, что правила в наборе равновероятны. Известно, что аппаратная реализация генераторов на клеточных автоматах выполняется с использованием только триггеров и сумматоров по модулю два [12]. Например, значение «0,5» в столбце « A_{sp} » означает, что удельные аппаратные затраты для данного набора равны половине сумматора по модулю два на разряд. В столбцах « A_{min} » и « A_{max} » приведены минимальные и максимальные аппаратные затраты для генератора на различных конфигурациях правил.

Также для аналогии рассмотрим все LFSR тринадцатой степени, генерирующие последовательность максимальной длины. Пример порождающего полинома с минимальными аппаратными затратами для LFSR: $x^{13} + x^4 + x^3 + x + 1$. Аппаратные затраты в этом случае составляют один пятиходовой сумматор по модулю два, который можно представить как 4 двухходовых сумматора, то есть $LFSR_{min} = L_{min} = 4/13 = 0,3$. В наихудшем с точки зрения аппаратных затрат полиноме содержится 12 членов, то есть аппаратные затраты составляют 11 двухходовых сумматоров по модулю два: $LFSR_{max} = 11/13 = 0,85$. Можно предположить, что в среднем аппаратные затраты будут равны:

$$LFSR_{avg} = (LFSR_{max} + LFSR_{min}) / 2 = 0,58.$$

Общий вид LFSR и клеточного автомата с циклическими граничными условиями представлен на рисунке 3.2. Видно, что рисунки 3.2, а) и 3.2, б) отличаются только наличием у LFSR обратной связи с пятой ячейки. Однако именно эта обратная связь не позволяет представить LFSR в

видах, представленных на рисунках 3.2, в) и 3.2, г). Рисунок 3.2, в) представляет минимальное представление клеточного автомата с циклическими граничными условиями: все связи локальные и имеют малую длину. Рисунок 3.2, г) показывает представление клеточного автомата в виде кольца, аналогично можно расположить ячейки по любому другому пути, необходимому для данной конкретной реализации. У LFSR данное преимущество отсутствует.

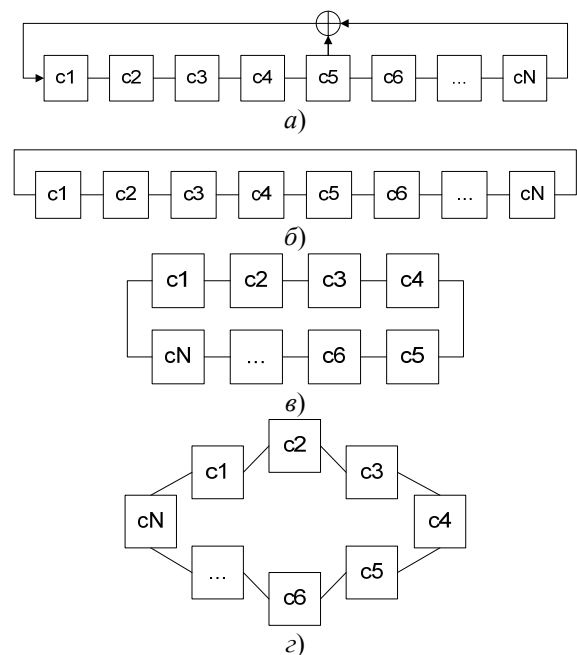


Рисунок 3.2 – Конечные автоматы:
 а) – LFSR; б) – КА с ЦГУ; в) – «минимальное» представление КА с ЦГУ;
 г) – КА «свёрнутый» в кольцо

Рассмотрим задачу: необходимо найти порождающий вектор размерности N , построенный на двух правилах и генератор на котором должен выдавать последовательность максимальной длины. Если нужно найти вектор размерности строго N , причём N нечётное, то лучше использовать пару правил 90, 240. В противном случае – 60, 240. Пару правил 60, 150 использовать не рекомендуется, так как она выдаёт такие статистические результаты, как и 90, 240, но имеет гораздо

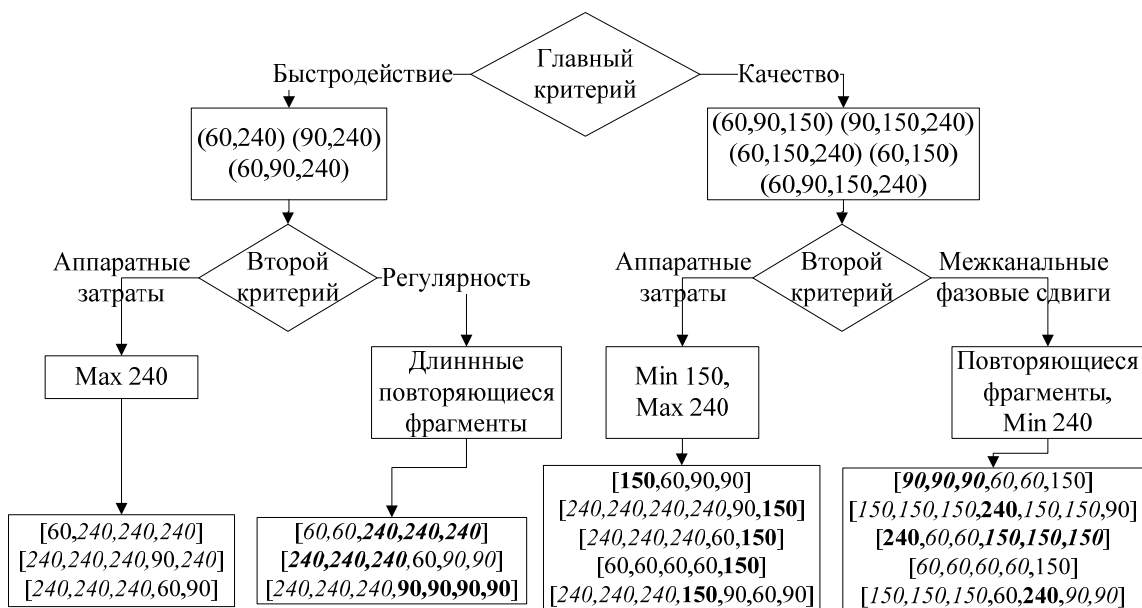


Рисунок 3.3 – Схема выбора набора правил

большие удельные аппаратные затраты. В таблице 3.1 видно, что пара 60, 240 имеет наилучшие статистические результаты среди пар правил, однако ошибка трудно предсказуема, в то время как пары 60, 150 и 90, 240 стабильно имеют ошибки на чётных степенях.

Тройки правил не имеют ошибок вообще, то есть высока вероятность, что при использовании набора из трёх правил будет получен вектор нужной размерности. Если важны удельные аппаратные затраты, то лучше использовать 60, 90, 240. В противном случае наилучшим выбором является 90, 150, 240, так как этот набор имеет лучшие статистические характеристики и относительно малые аппаратные затраты.

Если есть возможность использовать набор из 4 правил (60, 90, 150, 240) – то лучше использовать его ввиду отличных статистических характеристик и малых аппаратных затрат полученного генератора. Однако задача поиска вектора на 4 правилах значительно сложнее, чем поиск векторов на двух или трёх правилах.

Методика выбора конкретного набора правил для генератора в зависимости от критерия приведена на рисунке 3.3.

Минимальная реализация может быть выполнена на правилах 240 (0 сумматоров), 60 и 90 (по одному сумматору). Соответственно, генератор на этих правилах содержит максимум один сумматор по модулю два на разряд. В то же время, если используется большое количество ячеек с правилом 240, то аппаратные затраты снижаются (таблица 3.1). Однако их недостаток в том, что в этом случае фрагменты генератора повторяют работу LFSR. Чтобы улучшить качество, мы можем использовать конфигурации, содержащие

правила 60 и 90. Эти правила формируют достаточно большие фазовые сдвиги M-последовательности между соседними каналами, что повышает качество генерируемой последовательности.

С другой стороны, если главным критерием является качество, то тогда необходимо минимизировать число ячеек с правилом 240 (межканальный сдвиг для этого разряда равен единице). Заметим, что расчёт фазовых сдвигов упрощается при регулярной структуре генераторов, то есть наличии фрагментов с одинаковыми правилами функционирования. Соответственно, можно выбирать конфигурации, включающие правила и 150, и 60, и 240. Можно набирать любую доступную конфигурацию, при этом минимизируя количество ячеек с правилом 240.

Заключение

В работе исследована проблема проектирования генераторов псевдослучайных тестовых наборов на клеточных автоматах с расширенным набором правил – 0, 170, 204, 102, 240, 90, 60, 150. Рассмотрена проблема выбора наилучшего поднабора из приведенных правил. Выяснено, что генераторы далеко не на всех выборках способны генерировать последовательность максимальной длины. Предложен метод выбора наилучшего набора правил для каждой конкретной ситуации.

ЛИТЕРАТУРА

1. Agrawal, V. Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits / V. Agrawal, M. Bushnell. – Springer, 2000. – 712 p.

2. *Golomb, S.W.* Shift register sequences / S.W. Golomb. – San Francisco : Holden-Day. – 1967. – 224 p.
3. *Hortensius, P.D.* Parallel random number generation for VLSI systems using cellular automata / P.D. Hortensius // IEEE Transactions on Computers. – 1989. – Vol. 38, № 10. – P. 1466–1473.
4. *del Reya, A.M.* Reversibility of linear cellular automata / A.M. del Reya, G.R. Sanchez // Applied Mathematics and Computation. – 2011. – Vol. 217. – P. 8360–8366.
5. *Ярмолик, В.Н.* Реализация генератора псевдослучайной последовательности на клеточных автоматах / В.Н. Ярмолик, И.А. Мурашко // Автоматика и вычислительная техника. – 1993. – № 3. – С. 9–13.
6. *Cattell, K.* Minimal cost one-dimensional linear hybrid cellular automata of degree through 500 / K. Cattell, S. Zhang // Journal of Electronic Testing : Theory and Applications. – 1995. – Vol. 6. – P. 255–258.
7. *Cinki, Z.* Reversibility of 1D cellular automata with periodic boundary over finite fields Z_p / Z. Cinki, H. Akin, I. Siap // Journal of Statistical Physics. – 2011. – Vol. 143, № 4. – P. 807–823.
8. *Kari, J.* Reversible cellular automata // Lecture Notes in Computer Science. – 2005. – Vol. 3572. – P. 57–68.
9. *Voorhees, B.* Nearest neighbor cellular automata over Z_2 with periodical boundary conditions // Physica D. – 1990. – Vol. 45. – P. 26–35.
10. *Лидл, Р.* Конечные поля : В 2-х т., пер. с англ. / Р. Лидл, Г. Нидеррайтер // М. : Мир. – 1988. – Т. 1. – 820 с.
11. *Brent, R.P.* Three ways to test irreducibility // INRIA, Nancy France. – 2008. – P. 1–20.
12. *Мурашко, И.А.* Встроенное самотестирование. Методы минимизации энергопотребления / И.А. Мурашко, В.Н. Ярмолик // Saarbrücken : LAP Lambert Academic Publishing. – 2012. – 348 с.

Поступила в редакцию 14.11.13.