

УДК 004.3

## **МЕТОДИКА ПРОЕКТИРОВАНИЯ СИГНАТУРНОГО АНАЛИЗАТОРА НА КЛЕТОЧНЫХ АВТОМАТАХ ДЛЯ ВСТРОЕННОГО САМОТЕСТИРОВАНИЯ**

**Д. Е. ХРАБРОВ, И. А. МУРАШКО**

*Учреждение образования «Гомельский государственный  
технический университет имени П. О. Сухого»,  
Республика Беларусь*

### **Введение**

Распространенной методикой тестопригодного проектирования электронных устройств является встроенное самотестирование (*BIST – Built-in Self-Test*), которое позволяет устройству тестировать самого себя без применения дополнительного оборудования [1]. Для реализации встроенного самотестирования необходимы как минимум два компонента: генератор псевдослучайных тестовых наборов и схема сжатия [2]. Часто оба этих компонента реализуют на базе регистров сдвига с линейной обратной связью (*LFSR – Linear feedback shift register*). В работах [3]–[5] показано, что применение клеточных автоматов позволяет значительно улучшить качество генерируемого псевдослучайного тестового воздействия.

Главный аргумент в пользу использования *BIST* – отказ от использования дорогостоящей тестовой аппаратуры. Снижение инвестиций на тестирующее оборудование делает *BIST* очень привлекательной технологией. Даже нет необходимости полностью исключать тестирующее оборудование из процесса производства. Ведь если длительность тестирования может быть снижена путем вынесения части тестирования непосредственно на кристалл, то можно достичь той же пропускной способности с менее быстрым и менее дорогим тестовым оборудованием. Кроме того, если производство будет перенесено на более быстрые кристаллы, то *BIST* будет использовать появившиеся резервы и время тестирования станет еще меньше. В классическом подходе используется *LFSR* и многоканальный сигнатурный анализатор для реализации *BIST*. Недостатком данного подхода является длинная обратная связь с выхода последнего триггера на вход первого, которая зависит от порождающего полинома [6].

Целью данной работы является снижение вероятности необнаружения неисправностей за счет использования клеточного автомата с расширенным набором правил в качестве многоканального сигнатурного анализатора.

### **Встроенное самотестирование**

Ключевыми элементами любой системы встроенного самотестирования являются генератор тестовых воздействий и сигнатурный анализатор. Ко входу устройства подключен генератор псевдослучайных тестовых наборов. Наборы не обязательно должны быть псевдослучайными или генерируемыми. Например, тестовые воздействия можно брать из микросхемы памяти. Однако именно псевдослучайные тестовые наборы получили наибольшую популярность. Далее тестируемая схема (*CUT – Circuit Under Test*) обрабатывает каждый поступающий тестовый набор и отдает результат своего функционирования на сигнатурный анализатор, который накапливает в себе ответ [7]. После завершения тестирования получившееся на сигнатурном ана-

лизаторе накопленное число (сигнатура) сравнивается с эталонным числом. Если обе сигнатуры совпадают, то устройство функционирует корректно [8].

Рассмотрим эффективность реализации классических сигнатурных анализаторов и реализации на клеточных автоматах. Каждая ячейка генератора псевдослучайных последовательностей на клеточных автоматах включает в себя  $D$ -триггер и сумматор по модулю два [9]. Друг от друга ячейки правил 150 и 90 отличаются только наличием обратной связи с выхода триггера на его вход. Для создания сигнатурного анализатора на клеточных автоматах необходимо на каждый триггер добавлять по одному элементу  $XOR$ . Например, клеточный автомат [150, 150, 150, 150, 90] может быть использован в качестве сигнатурного анализатора, схема которого показана на рис. 1.

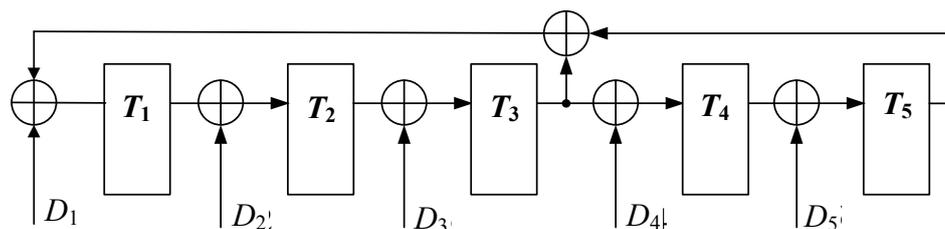


Рис. 1. Многоканальный сигнатурный анализатор на клеточных автоматах

Как известно, клеточные автоматы проигрывают  $LFSR$  с точки зрения занимаемой площади кристалла, так как каждая ячейка клеточного автомата требует как минимум одного элемента  $XOR$  [10]. Однако у клеточных автоматов есть преимущества перед  $LFSR$ . Пусть на многоканальный сигнатурный анализатор с порождающим полиномом  $1 \oplus x^5 \oplus x^6$  поступает следующая последовательность (табл. 1, строка «нет»). Такой сигнатурный анализатор не определяет ошибки при инверсии бит 2 и 9, 3 и 10, 4 и 11, 5 и 12. Это можно объяснить тем, что 4 триггера соединены последовательно и передают ошибку дальше. Далее ошибка опять появляется при инверсии 8 и 15, 9 и 16, и т. д. Похожие неопределимые ошибки можно увидеть и в случае трехкратного инвертирования бит. Например, если инвертировать биты (2, 3, 32) или (3, 4, 33), то с помощью сигнатурного анализатора на  $LFSR$  такая ошибка найдена не будет.

Таблица 1

### Сравнение пропущенных ошибок

Ошибка	Последовательность
нет	110010000011111110101001001001101010111011011011101001111110
2, 9	100010001011111110101001001001101010111011011011101001111110
3, 10	111010000111111110101001001001101010111011011011101001111110
4, 11	1101100000011111110101001001001101010111011011011101001111110
5, 12	110000000101111110101001001001101010111011011011101001111110
8, 15	1100100100111110110101001001001101010111011011011101001111110
9, 16	1100100010111111010101001001001101010111011011011101001111110
2, 3, 32	101010000011111110101001001001111010111011011011101001111110
3, 4, 33	111110000011111110101001001001100010111011011011101001111110

В табл. 1 в качестве входной последовательности может быть любая последовательность бит, так как необнаруживаемые ошибки определяются не входной последовательностью, а конфигурацией сигнатурного анализатора.

**Сравнение эффективности работы регистра сдвига с клеточными автоматами**

Основным преимуществом клеточных автоматов перед *LFSR* является меньшая вероятность необнаружения ошибок.

Рассмотрим пример, приведенный на рис. 2. На нем рассмотрена часть сигнатурного анализатора, состоящая из двух последовательно соединенных триггеров и двух элементов *XOR*, необходимых для функционирования.

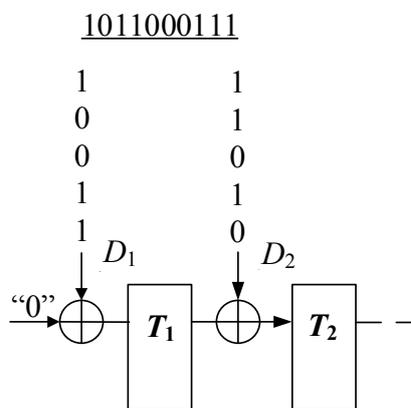


Рис. 2. Часть LFSR

На эту часть сигнатурного анализатора подается последовательность «1011000111», на каждом такте из начала последовательности берутся два бита и подаются на входы триггеров. При подаче правильной последовательности на триггерах будет значение «1 0». Но если произойдет двухкратная ошибка и последовательность изменится на «1110000111», то сигнатура также будет «1 0» и ошибка не будет обнаружена.

Теперь рассмотрим реализацию на клеточных автоматах. Общая схема части сигнатурного анализатора приведена на рис. 3. На рисунке приведены две ячейки клеточного автомата. Значение ячейки под номером *i* может зависеть только от значений левого соседа (ключ  $\alpha_i$ ), от собственного значения (ключ  $\lambda_i$ ) или от значения правого соседа (ключ  $\beta_i$ ).

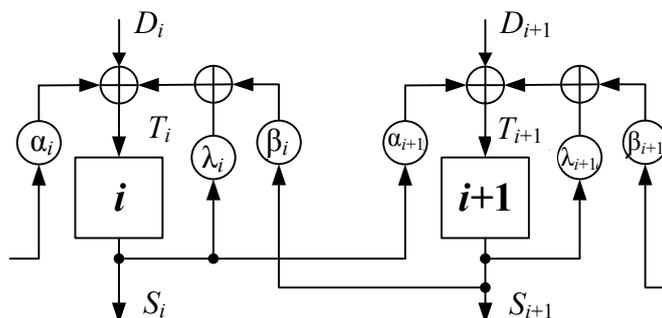


Рис. 3. Часть LHCA, состоящая из двух ячеек

Значение триггера ячейки клеточного автомата можно описать формулой

$$T_i = D_i \oplus \alpha_i S_{i-1} \oplus \lambda_i S_i \oplus \beta_i S_{i+1}.$$

Фрагмент сигнатурного анализатора с рис. 2 можно представить как клеточный автомат, работающий на паре правил (240, 240):

$$240: T_i = D_i \oplus S_{i-1},$$

$$240: T_i = D_i \oplus S_{i-1}.$$

Можно заменить первое правило 240 на 60:

$$60: T_i = D_i \oplus S_{i-1} \oplus S_i,$$

$$240: T_i = D_i \oplus S_{i-1}.$$

Как не сложно заметить, такой сигнатурный анализатор также подвержен описанной ошибке. Ведь по сути правило 240 означает просто взятие значения из предыдущего триггера.

Чтобы избавиться от ошибки, заменим правило 240 на правило 60. Тогда при тех же самых входных условиях ошибка будет обнаружена.

$$60: T_i = D_i \oplus S_{i-1} \oplus S_i,$$

$$60: T_i = D_i \oplus S_{i-1} \oplus S_i.$$

Рассмотрим возможные комбинации пар правил подробнее. При моделировании на часть клеточного автомата подается последовательность длиной 10 символов. Далее перебираются все двукратные ошибки, находится процент необнаружения. Для эксперимента слева всегда приходят нули, справа – предыдущее значение ячейки. Для комбинации (240, 60) из 45 возможных неисправностей не были обнаружены 5, т. е. 11 %.

Наиболее плохой результат дают пары правил (240, 240), (240, 90), (60, 90) и (60, 240) – вероятность необнаружения 22 %. Кроме уже предложенной ошибки данные пары правил не обнаруживают инвертирование пары бит, подаваемых на второй триггер, не зависимо от номера итерации. На рис. 2 изображен фрагмент *LFSR*, но такую же схематику имеет и клеточный автомат, работающий на правилах (240, 240) – происходит только сдвиг значений.

Один из лучших результатов дала пара правил (90, 60):

$$90: T_i = D_i \oplus S_{i-1} \oplus S_{i+1},$$

$$60: T_i = D_i \oplus S_{i-1} \oplus S_i.$$

Парой (90, 60) были обнаружены все возможные неисправности. Аналогичный результат выдали пары (60, 60), (60, 150) и (90, 150).

Вероятность необнаружения 8 % дают пары правил: (150, 90), (150, 240), (90, 90) и (90, 240):

$$90: T_i = D_i \oplus S_{i-1} \oplus S_{i+1},$$

$$240: T_i = D_i \oplus S_{i-1}.$$

Эти пары правил не находят только представленную ранее неисправность. Поскольку второе правило выполняет только сдвигающую функцию, то становится ясным, что такой сигнатурный анализатор не сможет находить все ошибки.

Следующий результат, 11 % необнаружения, дают пары правил: (240, 60), (240, 150), (150, 60), (150, 150). Эти пары правил не обнаруживают только инвертирование двух бит, одновременно подаваемых на два триггера. Если ошибки разнесены по времени, то они будут обнаружены. Подробнее результаты сравнения показаны в табл. 2.

Таблица 2

## Сравнение пар правил

Пары правил	Не найдено	Необнаружение	Шаблоны ошибок
(240, 240), (60, 90), (240, 90), (60, 240)	10	22 %	$(i, j), i = 1, 3, \dots,$ $j = 3, 5, \dots$
(90, 150), (60, 150), (90, 60), (60, 60)	5	11 %	$(i, i+1), i = 0, 2, \dots$
(150, 90), (150, 240), (90, 90), (90, 240)	4	8 %	$(i, i+1), i = 1, 3, \dots$
(60, 150), (90, 150), (60, 60), (90, 60)	0	0 %	–

Для реализации *LFSR* всегда нужна обратная связь, ячейка же клеточного автомата зависит только от ближайших соседей. Это означает, что сигнатурные анализаторы на клеточных автоматах имеют более гибкую структуру, при необходимости такой сигнатурный анализатор значительно проще модифицировать.

Сигнатурный анализатор и на клеточных автоматах, и на *LFSR* имеют 100%-е покрытие единичных неисправностей. Это означает, что если будет инвертирован один любой бит, то сигнатурный анализатор гарантированно предупредит об ошибке. Рассмотрим подробнее двукратные ошибки, т. е. когда в последовательности инвертированы два бита.

Для сравнения сигнатурных анализаторов на клеточных автоматах и *LFSR* был проведен эксперимент. Входная последовательность длиной 60 бит сжималась на сигнатурных анализаторах. Если при наличии ошибки сигнатура не отличалась от эталонной, то ошибка считается не найденной. Смоделированы все однократные, двукратные, трехкратные и четырехкратные ошибки. Результаты приведены в табл. 3.

Таблица 3

## Сравнение пропущенных ошибок

Ошибки	1	2	3	4	Аппаратные затраты			
Всего	60	1170	100 %	34220	100 %	487635	100 %	XOR
LFSR(1)	0	115	9,8 %	790	2,3 %	9972	2,0 %	2
LFSR(2)	0	56	4,8 %	518	1,5 %	7718	1,58 %	4
LHCA	0	39	3,3 %	471	1,3 %	7592	1,55 %	5

В строке «Ошибки» табл. 3 показано, сколько бит были инвертированы в изначальной битовой последовательности. Строка «всего» – сколько всего ошибочных ситуаций создано. Далее в каждой строке показано, сколько ошибок каждого типа было не найдено тем или иным сигнатурным анализатором. Характеристическим полиномом для «LFSR(1)» является  $1 \oplus x^5 \oplus x^6$ , для «LFSR(2)» и «LHCA» –  $1 \oplus x \oplus x^4 \oplus x^5 \oplus x^6$ , клеточный автомат имеет порождающий вектор [102 90 90 90 90 240] и нулевые граничные условия.

Исследования показали, что в общем случае аппаратные затраты сигнатурного анализатора на *LFSR* и клеточных автоматах отличаются незначительно. С другой стороны, на клеточных автоматах сигнатурный анализатор обнаруживает гораздо больше многократных неисправностей.

### Методика проектирования сигнатурного анализатора по заданным критериям

Методика выбора конкретного набора правил для генератора в зависимости от критерия была предложена авторами в статье [11]. Кроме того, в работе [12] предложен эвристический алгоритм построения генератора на клеточных автоматах. Полученные в этих работах выводы можно использовать и для получения сигнатурных анализаторов. Методику в общем виде можно записать следующим образом:

1. Выбор приоритета согласно схеме из [11]: быстродействие, качество, аппаратные затраты, регулярность, межканальные фазовые сдвиги.
2. Из выбора приоритета обычно следуют несколько наборов правил. Необходимо выбрать один набор правил.
3. Выбор старшей степени порождающего полинома – это необходимая размерность результирующего сигнатурного анализатора.
4. Синтез комбинации правил клеточного автомата. Подробнее рассмотрен в [12].
5. Синтез сигнатурного анализатора по клеточному автомату.

Рассмотрим пример использования методики. Пускай, главным критерием является быстродействие, а второстепенным регулярность. Под этот приоритет подходит пара правил (90, 240). Пускай старшая степень будет 27. Синтезированная комбинация правил для клеточного автомата с циклическими граничными условиями:  $[240\ 240\ (90)^{25}]$ . Характеристический полином у этой последовательности  $1 \oplus x \oplus x^3 \oplus x^5 \oplus x^{17} \oplus x^{19} \oplus x^{21} \oplus x^{25} \oplus x^{27}$  является примитивным.

Синтез сигнатурного анализатора по клеточному автомату происходит просто – на каждый триггер добавляется по одному двухвходовому элементу *XOR* для ввода данных. Сигнатурный анализатор  $[240\ 240\ (90)^{25}]$  показан на рис. 4.

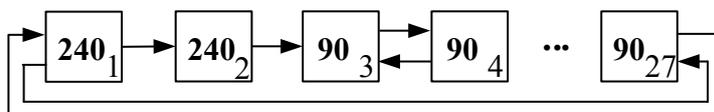


Рис. 4. Сигнатурный анализатор  $[240\ 240\ (90)^{25}]$

Оценим аппаратные затраты. На использование одного правила 240 в клеточном автомате нужен один двухвходовой сумматор по модулю 2 (*XOR*). Для правила 90 – два элемента *XOR*. Кроме того, для создания сигнатурного анализатора нужно дополнительно по одному *XOR* на каждый триггер. Итого для сигнатурного анализатора с рис. 4 получается  $2 \cdot 1 + 25 \cdot 2 + 27 = 79$  элементов *XOR*.

Для *LFSR* обратные связи строятся по порождающему полиному. Рассмотрим многовходовой сигнатурный анализатор с характеристическим полиномом  $1 \oplus x \oplus x^3 \oplus x^5 \oplus x^{17} \oplus x^{19} \oplus x^{21} \oplus x^{25} \oplus x^{27}$ . Аппаратные затраты:  $8 + 27 = 35$  элементов *XOR*.

### Заключение

В работе рассмотрено проектирование сигнатурных анализаторов на клеточных автоматах. Проведено сравнение сигнатурных анализаторов на клеточных автоматах и *LFSR*, рассмотрены преимущества и недостатки каждого. В случае необходимости минимизации занятой площади кристалла использовать *LFSR*. Для улучшения результатов тестирования использовались клеточные автоматы. Предложены конкретные наборы правил для синтеза сигнатурных анализаторов. В результате работы показано, что применение клеточного автомата с расширенным набором правил в качестве многоканального сигнатурного анализатора позволило снизить вероятность необнаружения неисправностей.

## Литература

1. Agrawal, V. Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits / V. Agrawal, M. Bushnell. – Springer, 2000. – P. 712.
2. Stroud, Charles E. A Designer's Guide to Built-In Self-Test / Charles E. Stroud. – Springer US, 2002. – Vol. 19. – P. 320.
3. Ярмолик, В. Н. Реализация генератора псевдослучайной последовательности на клеточных автоматах / В. Н. Ярмолик, И. А. Мурашко // Автоматика и вычисл. техника. – 1993. – № 3. – С. 9–13.
4. Patil, H. "Cellular automata" for built in test pattern generation and test response analyzer / H. Patil, S. S. Padhy // International Journal of Electrical and Electronics Research. – 2014. – Vol. 2, № 3. – P. 44–50.
5. Jeon, Jun-Cheol. Analysis of hash functions and cellular automata based schemes / Jun-Cheol Jeon // International Journal of Security and Its Applications. – May, 2013. – Vol. 7, № 3. – P. 303–316.
6. Храбров, Д. Е. Автоматизированное проектирование генератора псевдослучайной последовательности на клеточных автоматах / Д. Е. Храбров // Исследования и разработки в области машиностроения, энергетики и управления : материалы XI Междунар. межвуз. науч.-техн. конф. студентов, магистрантов и аспирантов / М-во образования Респ. Беларусь, Гомел. гос. техн. ун-т им. П. О. Сухого. – 2011. – С. 483–486.
7. Сагалович, Ю. Л. Синтез сигнатурного анализатора для двухуровневой комбинационной схемы / Ю. Л. Сагалович, В. Н. Ярмолик // Автоматика и телемеханика. – 1990. – Т. 50, № 4. – С. 155–160.
8. Miczo, A. Digital Logic Testing and Simulation / A. Miczo; Ed. by dm Miller. – Wiley Interscience, 2003. – P. 696.
9. Храбров, Д. Е. Методика синтеза генератора псевдослучайных последовательностей по заданному полиному на клеточных автоматах / Д. Е. Храбров, И. А. Мурашко / М-во образования Респ. Беларусь, Бел. гос. ун-т информатики и радиоэлектроники. – Минск, 2011. – С. 280–281.
10. Estimating aliasing in ca and lfsr based signature registers / D.M. Miller, S. Zhang, W. Pries, R.D. McLeod // Computer Design: VLSI in Computers and Processors, 1990. ICCD '90. Proceedings, 1990 IEEE International Conference on. – Sep. 1990. – P. 157–160.
11. Храбров, Д. Е. Применение клеточных автоматов с расширенным набором правил для генерирования псевдослучайных тестовых последовательностей / Д. Е. Храбров, И. А. Мурашко // Проблемы физики, математики и техники. – 2014. – № 1 (18). – С. 98–104.
12. Мурашко, И. А. Применение клеточных автоматов с расширенным набором правил для генерирования псевдослучайных тестовых последовательностей / И. А. Мурашко, Д. Е. Храбров // Вестн. Моск. гос. ун-та приборостроения и информатики. – 2013. – Т. 47. – С. 78–93.

Получено 23.06.2015 г.