Автоматизированное проектирование генераторов псевдослучайных последовательностей с использованием аппарата клеточных автоматов

Мурашко И.А.; Храбров Д.Е.
Кафедра «Информационные технологии», ФАИС
Гомельский государственный технический университет им. П. О. Сухого
Гомель, Республика Беларусь
e-mail: science@dexp.in

***Аннотация*—В работе исследуется проблема синтеза генераторов псевдослучайных последовательностей на клеточных автоматах по заданному полиному. Получен алгоритм синтеза генераторов для порождающих полиномов определённой степени, реализованный программно. Проанализированы циклические клеточные автоматы, использующие более одного правила для своего функционирования.**

Ключевые слова: псевдослучайная последовательность; генератор последовательности; последовательность максимальной длины; циклические граничные условия

I. Введение

Самым распространённым методом генерации псевдослучайных последовательностей (ПСП) является регистр сдвига с линейной обратной связью (англ. *Linear feedback shift register*, *LFSR*) [1]. Его распространённость объясняется изученностью и простотой аппаратной реализации, для которой требуется лишь регистр сдвига и многовходовой сумматор по модулю два. Но в последнее время внимание учёных направлено на использование и альтернативных методов генерации псевдослучайных последовательностей максимальной длины, т.е. на использование клеточных автоматов [2].

Причина популярности клеточных автоматов кроется в их простоте и существенном потенциале при моделировании сложных систем. Клеточный автомат в общем может быть рассмотрен как простая модель пространственно протяжённой системы, состоящей из ряда компонентов (ячеек). Связи между ячейками ограничены локальным взаимодействием.

Пример клеточного автомата показан на рисунке 1. Здесь правило 90 означает что следующее значение ячейки зависит только от значений соседних ячеек. Правило 150 означает что для вычисления последующего значения необходимо также просуммировать предыдущее значение самой ячейки.

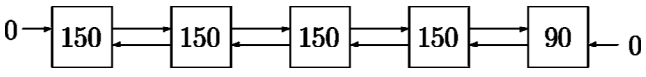

Рис. 1. Пример клеточного автомата

На рисунке 2 представлена аппаратная реализация клеточного автомата, показанного на рисунке 1. Реализация выполнена на *D*-триггерах.

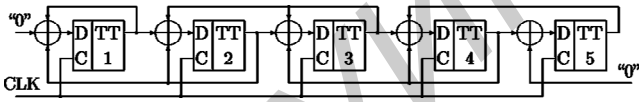

Рис. 2. Аппаратная реализация клеточного автомата

Аппаратная реализация клеточного автомата не сложна, однако требует N триггеров и столько же элементов *XOR*. На рисунке 2 имеются 4 трёхвходовых и 1 двухвходовой элемент *XOR*. Это объясняется тем, что правило 150 использует для вычисления 3 значения (значения двух соседей и значение самой клетки), а правило 90 использует лишь значения соседей.

Для вычисления значения ячейки клеточного автомата можно использовать (1)

$$y'[i] = f(y[i-1], y[i], y[i+1]), \qquad (1)$$

где f – функция переходов клетки;

$y'[i]$ – состояние i-й клетки в следующий момент времени;

$y[i]$ – состояние i-й клетки в данный момент времени.

Например: *правило 90* – $y'[i] = y[i-1] \oplus y[i+1]$; *правило 240* – $y'[i] = y[i-1]$; *правило 170* – $y'[i] = y[i+1]$; *правило 150* – $y'[i] = y[i-1] \oplus y[i] \oplus y[i+1]$; *правило 60* – $y'[i] = y[i-1] \oplus y[i]$; *правило 102* – $y[i] \oplus y[i+1]$.

II. Постановка задачи

Наиболее полно исследованы клеточные автоматы на основании правил 90 и 150 с нулевыми граничными условиями (ГУ) [2]. Для них созданы таблицы конфигураций, позволяющих формировать ПСП максимальной длины. Например, таблица всех таких полиномов седьмой степени опубликована авторами в [3]. Однако использование только правил 90 и 150 ограничивает свободу разработчиков цифровых систем.

В ходе проведённых исследований было показано, что расширение количества используемых правил позволяет достаточно просто находить конфигурации генераторов ПСП максимальной длины. Кроме нулевых ГУ могут использоваться и циклические ГУ, для которых комбинации правил 90 и 150 не дают

решения. В данной работе предложена методика проектирования генераторов ПСП на клеточных автоматах с циклическими ГУ на основании заданного полинома с расширенным набором правил.

III. Исследование

Правила 90 и 150 используют значения одновременно обоих соседей. Можно получить 8 различных правил за счёт использования или не использования каждого из значений: левого, правого соседа и самой клетки. В качестве функции получения значения используется только сумматор по модулю два.

Граничные условия также могут варьироваться. Например, в [2] циклические ГУ лишь упоминаются, так как считается [4] что самыми перспективными являются именно нулевые ГУ. В данной работе сделана попытка опровергнуть данное утверждение. На рисунке 3 показан общий вид клеточного автомата с циклическими граничными условиями.

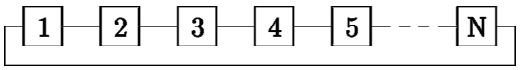

Рис. 3. Циклические граничные условия

Был создан программный комплекс, перебирающий все возможные порождающие вектора определённой степени. Каждому вектору находится соответствующий полином. Серьёзной оптимизацией является использование свойств трёхдиагональной матрицы и *LU*-разложения. Далее полученный полином проверяется на неприводимость, так как только генератор, построенный на неприводимом полиноме, может выдавать последовательность максимальной длины. Если старшая степень полинома соответствует размерности порождающего вектора, то данный порождающий вектор является корректным и запоминается, как выдающий последовательность максимальной длины.

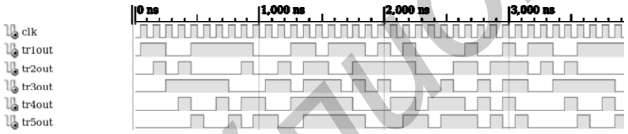

Рис. 4. Моделирование работы клеточного автомата

Проверка корректности производилась на различных уровнях описания: функциональном и структурном. При моделировании любого получается график с рисунка 4, так как моделируется одна и та же физическая модель. Это свойство использовалось для верификации.

Функциональный *VHDL* хорош для быстрого прототипирования и отладки. Однако генерируемые принципиальные схемы избыточны. Как следствие, менее надёжны и имеют большее энергопотребление.

Полученное при помощи *Schematic* структурное *VHDL*-описание было проанализировано и конвертировано в шаблон, с помощью которого в программном комплексе реализуются КА любой любой размерности. Структурное описание оптимально с точки зрения надёжности устройства и энергопотребления, однако без автоматизации использовать этот уровень сложно. Программный комплекс генерирует структурное *VHDL*-описание в автоматическом режиме.

IV. Анализ результатов

Для примитивного полинома пятой степени $1 \oplus x^3 \oplus x^5$ было найдено 150 различных конфигураций КА с циклическими граничными условиями. Примеры конфигураций: [60 60 240 240 240], [90 240 90 240 90], [240 60 60 150 60], [150 60 240 90 240].

Первым в списке примеров идёт порождающий вектор [60 60 240 240 240]. Так как комплекс перебирает все порождающие вектора, а ГУ циклические, то были найдены ещё 4 эквивалентных порождающих вектора: [240 60 60 240 240], [240 240 60 60 240], [240 240 240 60 60], [60 240 240 240 60]. Правило 60 определяет работу *T*-триггера, а правило 240 – *D*-триггера. Такой генератор имеет простую аппаратную реализацию, так как используются только стандартные компоненты вычислительной техники, что позволяет обойтись вообще без сумматоров.

Вектор [240 60 60 150 60] можно было бы отнести к автомату на *D*- и *T*-триггерах, если бы не использование также правила 150. С одной стороны отличие минимально (всего одна ячейка), с другой – получен новый генератор, отличающийся от других.

Вектор [150 60 240 90 240] показателен тем, что используется 4 правила. Векторов, использующих 5 и более правил (из заданного расширенного набора) не существует. Это объясняется направленностью правил.

Также интересен эффект симметричности правил. Например, *правило 60* – $y[i] = y[i-1] \oplus y[i]$; *правило 102* – $y[i] \oplus y[i+1]$. Для полинома $1 \oplus x^3 \oplus x^5$ есть порождающий вектор [60 90 90 240 60]. Если заменить в данном векторе правила на симметричные, то получим: [102 90 90 170 102], порождающий вектор для полинома $1 \oplus x \oplus x^2 \oplus x^3 \oplus x^5$.

V. Вывод

Программный комплекс позволяет находить конфигурации КА с нулевыми и циклическими ГУ для расширенного набора правил для примитивных полиномов до 50 степени. Также формируется VHDL-описание генератора на клеточных автоматах.

[1] Golomb, S. W. Shift Register Sequences – San Francisco: Holden-Day, 1967. – 224 с.

[2] Hortensius, P. D. Parallel Random Number Generation for VLSI Systems Using Cellular Automata /P. D. Hortensius, R. D. McLeod, H. C. Card – IEEE Trans. Computers 38(10), 1989 – 1466-1473 с.

[3] Мурашко, И. А. Методика синтеза генератора псевдослучайных последовательностей по заданному полиному на клеточных автоматах / И. А. Мурашко, Д. Е. Храбров // Материалы Международной научной конференции «ИТС 2011». – Минск: БГУИР, 2011 г. / М-во образования Респ. Беларусь, Бел. гос. ун-т. инф-ки. и рад-ки – Минск: БГУИР, 2011. – 280-281 с.

[4] K. Cattell, J.C. Muzio. Synthesis of one-dimensional linear hybrid cellular automata. IEEE Transactions on Computer-Aided Design, 1996. – 325–335 pp.