УДК 004.3.001.57

МОДЕЛЬ ГЕНЕРАТОРА ПЕРЕСТАНОВОК НА ОСНОВЕ УПРАВЛЯЕМОГО ЦИКЛИЧЕСКОГО СДВИГА

С. С. Соболев

Воронежская государственная лесотехническая академия

Поступила в редакцию 22.09.2011 г.

Аннотаця. В работе описан алгоритм работы устройство генератора комбинаторны перестановок *n* целочисленны элементов. Дя исследованя работы генератора разработан системная модель на SystemC RTL-модель на языке описаня аппаратуры VHDL. Генератор отличаетя простото аппаратной реализации гибкостью настройк благодаря использованию алгоритм управляемог циклического сдвига. Предложенны данной статье модели генератора комбинаторны перестановок могу использоватья проектировании систем аппаратног формированя бинарных стро, представяющи данные ЭВМ.
Ключевы слов: генерато перестановок, комбинаторны перестановки, моделировани на системно уровне, моделировани на RTL-уровне, SystemC, HDL-описание, VHDL, регистры сдвига.
Annotation. The algorithm and structure of the permutations generator of n integer-valued elements are described. For the investigation of its work, the system-level model and the transaction-level model are implemented for the generator (in SystemC and VHDL, respectively). The generator of permutations is notable for the simplicity of its hardware implementation and for the flexibility of its configuration. It is achievable due to using of algorithm of the manageable circular shift. The models offered in this paper can be used in the design of the systems of hardware formatting of binary lines representing data in the computers.
Keywords: permutations generator, combinatorial permutations, system-level design, transaction-level modeling, SystemC, hardware description, VHDL, shift registers.

ВВЕДЕНИЕ

Генерация полных комбинаторных перестановок целочисленных элементов является весьма распространенной задачей во многих областях электроники, в частности, в сфере защиты информации. Примером использования алгоритмов генерации может служить формирование множества уникальных многосимвольных ключей для различных криптографических алгоритмов, а также их применение в механизмах шифрования данных и аутентификации пользователей; кроме того, существуют различные шифры, основанные на перестановках [1, 2].

Проведено много исследований, посвященных изучению генераторов и устройств формирования комбинаторных перестановок, в частности, глубокое исследование по систематизации и классификации комбинаторных алгоритмов и их аппаратных моделей проведено в монографии [3].

Основными критериями, характеризующими генераторы комбинаторных перестановок, являются простота аппаратной реализации, производительность, простота управления и настройки. Оптимизация по данным параметрам противоречива, поэтому для конкретных задач определяется компромиссное решение.

Отличительными особенностями рассматриваемого в данной статье генератора являются простота реализации, гибкость управления, возможность формирования исходного множества элементов, на основе которого генерируются перестановки. Кроме того, все перестановки генерируются в строгом порядке, а значит, каждую перестановку можно получить по её порядковому номеру. Простота реализации обеспечивается благодаря использованию алгоритма управляемого циклического сдвига, легко реализуемого на аппаратном уровне с помощью регистров сдвига.

Для исследования работы генератора перестановок разработаны системная и RTL (англ. Register transfer level) модели, которые могут использоваться при проектировании специализированных устройств, применяемых в системах динамического преобразования форматов

представления данных в ЭВМ [4]. Обе модели могут быть встроены в маршрут проектирования таких устройств в виде программных IP блоков (англ. soft IP cores).

АЛГОРИТМ ФОРМИРОВАНИЯ ПЕРЕСТАНОВОК

Опишем применяемый алгоритм генерации комбинаторных перестановок мощности n. В качестве множества упорядоченных элементов X возьмем исходную строку S длины n, заполненную уникальными символами, т.е.

$$X = \bigcup_{i=1}^{n} S_i,$$

где S_i – символ, находящийся на i-ой позиции в строке S, причем $\forall\, i, j\ i \neq j\ S_i \neq S_j$. Эта строка и будет начальной точкой алгоритма [5].

Первоначально полезная длина строки k равна 2. Запоминаем перестановку мощности k (для $k = 2$ ей соответствуют первые два символа в исходной строке в той же последовательности) и применяем циклический сдвиг строки следующим образом: символы, начиная со второй позиции и заканчивая k-ой, сдвигаем на одну позицию влево, а символ, находящийся на первой позиции, перемещаем на k-ую позицию. Запоминаем текущую перестановку, и далее уже к ней применяем циклический сдвиг строки описанным ранее способом. Продолжаем такие сдвиги до тех пор, пока следующая перестановка не станет повторением первой для этого шага (всего $(k - 1)$ циклических сдвигов). Множество перестановок, образованных из исходной строки с помощью одного или нескольких последовательных циклических сдвигов, назовем группой циклического сдвига. Исходная перестановка также входит в группу. Для исходной перестановки длины k, количество перестановок в соответствующей ей группе циклического сдвига равно k.

Переходим к следующему шагу, увеличивая на 1 полезную длину строки k. Для этого к каждой из $(k - 1)!$ сгенерированных на предыдущем этапе перестановок добавляем очередной символ, стоящий на k-ой позиции в исходной строке, и применяем циклические сдвиги для всех перестановок. Очевидно, что общее число циклических сдвигов равно $(k - 1)!\cdot(k - 1)$, а число перестановок для этого шага – $k!$.

Продолжаем процесс до тех пор, пока полезная длина строки k не станет равной n. Сформированные на последнем этапе перестановки будут искомыми комбинаторными перестановками мощности n. Множество всех перестановок обозначим следующим образом:

$$P = \bigcup_{i=1}^{n!} \pi_i.$$

Примечание: в качестве уникальных символов строки использовались символы латинского алфавита в соответствующем порядке.

Схема алгоритма для $n = 3$ изображена на рис. 1.

Докажем, что данный алгоритм генерации позволяет получить все возможные перестановки множества мощности n, т.е. результатом его работы является $n!$ уникальных перестановок. Очевидно, что общее число перестановок, генерируемых на каждом k-ом шаге, равно $k!$, а значит, общее число перестановок, полученных в результате работы n-шагового алгоритма, рав-

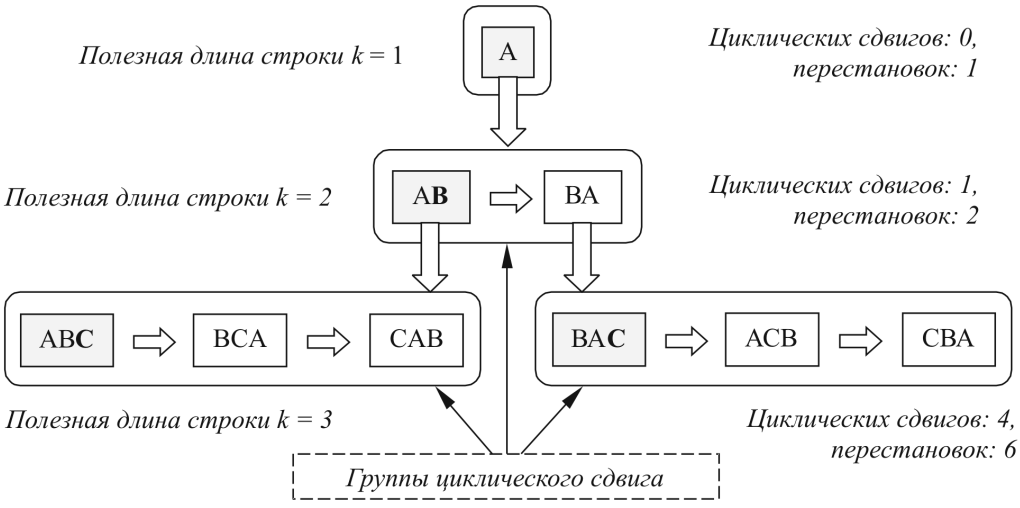

Рис. 1. Схема алгоритма генерации комбинаторных перестановок для $n = 3$

но $n!$. Применим метод доказательства от противного и предположим, что среди итоговых $n!$ перестановок нашлись две одинаковые перестановки. Здесь возможны два варианта:

Одна из двух найденных перестановок образована из другой путем одного или нескольких последовательных циклических сдвигов, т.е. обе эти перестановки находятся в одной из $(n-1)!$ групп по n перестановок в каждой, состоящих из перестановок, полученных с помощью циклических сдвигов (эти группы обозначены на рис. 1).

Найденные перестановки находятся в разных вышеупомянутых группах.

Первая ситуация невозможна, так как, согласно алгоритму, циклические сдвиги в рамках одной группы продолжаются до тех пор, пока следующая перестановка не станет повторением первой для этого шага.

Для анализа второй ситуации рассмотрим две группы, в состав которых входят обе найденные перестановки. В этом случае все перестановки, входящие в состав обеих групп, попарно бы совпадали, а значит, на каком-либо из предыдущих шагов (выше по дереву алгоритма) обязательно нашлись бы две перестановки, которые, являясь предками этих групп, сами в свою очередь входили в состав одной группы циклического сдвига и совпадали, т.е. попадали бы под условия ситуации № 1, которая невозможна.

Таким образом, исходное предположение оказалось неверным, следовательно, предложенный алгоритм действительно генерирует набор из $n!$ уникальных перестановок для множества мощности n.

Заметим, что рассматриваемый алгоритм представляет собой комбинацию следующих последовательно применяемых элементарных операций: добавление нового символа в строку (увеличение полезной длины строки) и циклический сдвиг строки. Подобная структура делает данный алгоритм очень удобным для реализации при разработке специализированных аппаратных устройств, т.к. на аппаратном уровне управляемый циклический сдвиг может быть реализован с помощью регистров сдвига.

УСТРОЙСТВО И ФУНКЦИОНАЛЬНОСТЬ ГЕНЕРАТОРА ПЕРЕСТАНОВОК

В соответствии с вышеописанным алгоритмом, блок генерации полных комбинаторных перестановок должен:

сформировать упорядоченное множество X_0 элементов символьной строки, на основе которого формируются перестановки;

каждый такт синхронизации таймера осуществлять генерацию уникальной перестановки, состоящей из элементов символьной строки наперед заданной длины n;

осуществлять прием очередной сгенерированной перестановки;

Общая структура блока генерации представлена на рис. 2. Блок генерации включает в себя несколько подблоков, каждый из которых имеет свою функциональность.

Задача формирователя исходной символьной строки – сформировать символьную строку S_0 фиксированной длины n, множество элементов которой представляет собой упорядоченный набор целых чисел от 0 до $n-1$. Таким образом, элементами исходной символьной строки S_0 являются элементы множества

$$X_0 = \{x_i \in \mathbb{Z} : x_i = i,\ i = 0, n-1\},$$

где n – наперед заданное количество элементов.

Формирователь очередной уникальной перестановки принимает на вход сформированную предыдущим подблоком символьную строку S_0 и на её основе формирует выходную символьную строку S_i, элементы которой образуют уникальную перестановку π_i. Для формирования перестановки π_i используется алгоритм генерации на основе циклического сдвига. Процедура формирования очередной перестановки происходит каждый такт синхронизации внутреннего таймера.

Подблок приема перестановок принимает на вход очередную сформированную перестановку. На каждом тактовом импульсе текущая перестановка включается в результирующее множество сгенерированных перестановок.

СИСТЕМНАЯ МОДЕЛЬ ГЕНЕРАТОРА ПЕРЕСТАНОВОК

Системная модель генератора перестановок разработана в среде Microsoft Visual Studio с использованием языка SystemC [6]. SystemC

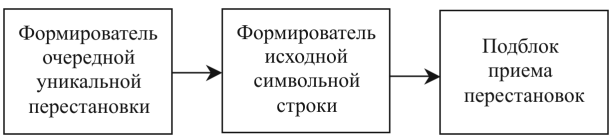

Рис. 2. Общая структура блока генерации перестановок

представляет собой надстройку стандартного языка программирования C++, реализованную в виде отдельных библиотек специальных классов. Данные библиотеки SystemC содержат в себе конструкции, позволяющие создавать эффективные и точные модели программных алгоритмов, аппаратных архитектур, интерфейсов и схем на системном уровне, т.е. практически всех компонентов встроенных систем. Такой подход имеет значительный потенциал, так как основан на однородном описании C++ и легко позволяет моделировать, тестировать системы, рассматривать альтернативные архитектуры. Кроме того, SystemC предоставляет развёрнутое описание процесса работы всей системы, представляющее собой программу C++, которая при исполнении ведёт себя так же, как и система.

Остановимся подробнее на внутреннем устройстве каждого подблока анализируемой модели, общая структура которой представлена на рис. 2. Очевидно, что любому подблоку на схеме соответствует отдельный модуль на SystemC.

Формирователь исходной символьной строки.

Модуль формирования исходной символьной строки включает в себя следующие элементы:

Входной порт тактирующего сигнала clk.

Множество выходных портов, каждый из которых передаёт на выход один символ сформированной строки. Общее количество портов n, причем каждый из них несет $[\log_2 n]$ бит информации, представляющих собой целое число из множества X0.

Функция инициализации, формирующая множество X0 на основе количества элементов n. Множество X0 задается на основе формулы. В общем случае, для обеспечения универсальности, множество X0 может задаваться любым способом, но должно отвечать следующим требованиям:

$$\forall x \in X_0 \; x \in Z;$$

$$\forall x, y \in X_0 \; x \neq y\,.$$

Функция процесса передачи элементов сформированного множества X0 на выходные порты. Процесс запускается при появлении переднего фронта на входе clk.

Формирователь очередной уникальной перестановки.

Модуль формирования очередной уникальной перестановки включает в себя несколько модулей:

$n - 1$ модулей циклического сдвига;

$n - 1$ вспомогательных модулей управления.

Каждый из $n - 1$ модулей циклического сдвига соединены последовательно и имеют уровень от 2 до n в соответствии с увеличением полезной длины символьной строки согласно вышеописанному алгоритму циклического сдвига. Общая схема модуля формирования уникальной перестановки для $n = 4$ изображена на рис. 3.

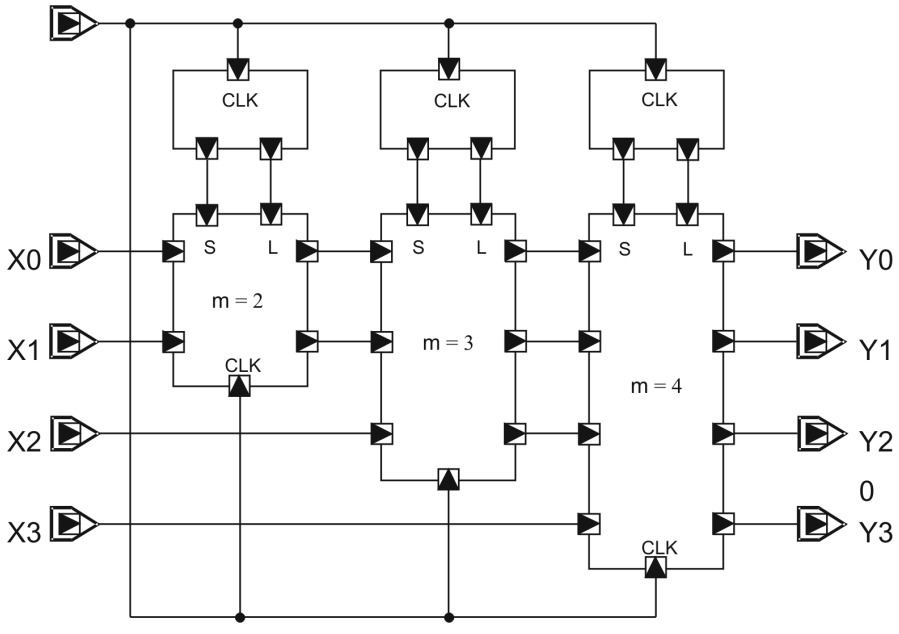

Рис. 3. Структура модуля формирования очередной перестановки для $n = 4$

На рис. 3 значения элементов исходной символьной строки подаются на порты X0, X1, X2, X3. Элементы очередной сформированной перестановки выводятся на порты Y0, Y1, Y2, Y3.

В состав модуля циклического сдвига уровня m входят:

Входной порт тактирующего сигнала clk.

Множества входных и выходных портов мощности m, каждый из которых принимает на вход или передаёт на выход один элемент текущей перестановки.

Внутренняя структура данных, хранящая текущее значение символьной строки на уровне m. Представляет собой целочисленный массив.

Входные порты управления L и S. При высоком логическом уровне сигнала на входе L значения элементов символьной строки считываются с входных портов и тут же передаются на выходные порты (либо на следующий уровень $m + 1$, либо в качестве результирующей перестановки), без циклического сдвига. При высоком логическом уровне сигнала на входе S, происходит циклический сдвиг строки, хранящейся во внутреннем массиве модуля, и её передача на выходные порты.

Функция считывания элементов строки с входных портов и запись во внутренний массив.

Функция чтения элементов из внутреннего массива и передача на выходные порты.

Функция инициализации внутреннего целочисленного массива.

Функция процесса, запускаемая при появлении переднего фронта на входе clk. В соответствии со значениями сигналов на портах L и S, либо происходит циклический сдвиг, либо зачитывание строки с входных портов, либо ни одно из этих действий. Сигналы на L и S не могут иметь высокий логический уровень одновременно.

Каждый из вспомогательных модулей управления представляет собой счетчик, который в соответствии с номером тактового импульса передает сигналы управления на порты L и S соответствующих модулей циклического сдвига. Модуль управления уровня уровня m содержит:

Входной порт тактирующего сигнала clk.

Выходные порты управления модулем циклического сдвига L и S.

Внутренняя переменная-счетчик c, увеличивающаяся на 1 каждый тактовый импульс.

Функция процесса, запускаемая при появлении переднего фронта на входе clk. В соответствии с уровнем m, количеством элементов n и текущим значением внутренней переменной c, процесс вычисляет значения функций $L = L(c, m, n)$ и $S = S(c, m, n)$ и передает полученные значения на выходные порты L и S соответственно.

Каждая из функций $L(c, m, n)$ и $S(c, m, n)$ возвращает либо логическую единицу, либо ноль. Если функция $L(c, m, n)$ возвращает единицу, то на порт L подается высокий уровень сигнала, если ноль – то низкий уровень. Аналогичное поведение для функции $S(c, m, n)$ и порта S.

Значения функций высчитываются по формулам:

$$L(c,m,n) = \left[c \bmod \left(\prod_{i=m}^{n} i \right) = 0 \right],$$

$$S(c,m,n) = \begin{cases} \overline{L(c,m,n)}, \; m = n \\ \overline{L(c,m,n)} \wedge \left[\left(c \bmod \left(\prod_{i=m}^{n} i \right) \right) \times \right. \\ \left. \times \bmod \left(\prod_{i=m+1}^{n} i \right) = 0 \right], \; m < n, \end{cases}$$

где mod – оператор вычисления остатка от деления.

Первые $n - 2$ тактов не дают на выход очередную перестановку. В связи с этим начальное значение счетчика c для каждого модуля управления циклическим сдвигом различно и вычисляется как $c = 2 - m$, где m – уровень текущего модуля управления.

Подблок приема перестановок.

В состав модуля приема перестановок входят:

Входной порт тактирующего сигнала clk.

Множество входных портов мощности n, каждый из которых принимает на вход один элемент текущей перестановки.

Функция процесса, вызывающаяся при появлении заднего фронта на порте clk. Функция считывает значения элементов текущей перестановки.

Общая схема системной модели блока генерации полных комбинаторных перестановок изображена на рис. 4 (при $n = 4$).

На приведенной схеме X0, X1, X2, X3 – элементы исходной символьной строки S_0; Y0, Y1, Y2, Y3 – элементы очередной сформированной перестановки на каждом такте.

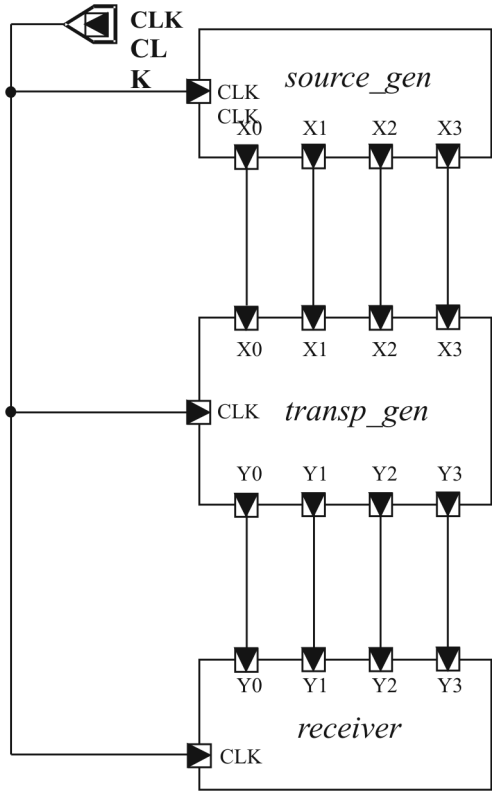

Рис. 4. Общая схема системной модели блока генерации полных комбинаторных перестановок для $n = 4$

Соответствия модулей SystemC элементам на схеме:

source_gen: модуль формирования исходной символьной строки;

transp_gen: модуль формирования очередной перестановки;

receiver: модуль приема перестановок;

CLK: источник тактовых импульсов.

Укрупненная схема модуля формирования очередной перестановки (*transp_gen*) изображена на рис. 3.

RTL-МОДЕЛЬ ГЕНЕРАТОРА ПЕРЕСТАНОВОК

Моделирование на RTL-уровне предоставляет возможность описания блока генерации перестановок на более низком уровне абстракции, нежели системный, – уровне регистровых передач. На данном уровне поведение цифровой микросхемы рассматривается в терминах потоков сигналов (или передачи данных) между аппаратными регистрами и логических операций над этими сигналами. Разработка RTL-описания блока генерации осуществлялась с применением языка VHDL, который является общепринятым международным стандартом [7]. Исследование RTL-модели проводилось в среде разработки ModelSimSE от Mentor Graphics [8].

Блок генерации перестановок представляет собой совокупность моделируемых объектов VHDL, каждый из которых соответствует отдельному подблоку на общей схеме, изображенной на рис. 2. При этом каждый объект может включать другие объекты в своем структурном описании.

Соединяющие объекты сигналы могут быть единичными или множественными (векторами). Сигналы первого типа в любой момент времени может принимать единственное двоичное значение (0 или 1). Векторы, в свою очередь, передают информацию в виде комбинации N двоичных значений, где N – ширина диапазона вектора. В случае генерации перестановок на множестве $X_0 = \{0,..., n - 1\}$, каждый элемент этого множества имеет разрядность $d = \log_2 n$. Таким образом, каждый передаваемый элемент перестановки моделируется на RTL-уровне в виде d-разрядного вектора, а вся перестановка – в виде $(n \cdot d)$-разрядного вектора.

Формирователь исходной символьной строки представляет собой $(n \cdot d)$-разрядный регистр с параллельным вводом и выводом. Регистр хранит в себе элементы исходного упорядоченного множества X_0, на основе которого осуществляется генерация перестановок. Запись и выдача данных происходит по высокому уровню синхросигнала. Запись данных в регистр осуществляется только при высоком уровне сигнала разрешения записи WE, что позволяет конфигурировать исходное множество извне перед процессом генерации.

Формирователь очередной уникальной перестановки представляет собой набор регистров циклического сдвига с параллельным вводом и выводом, работа которых координируется вспомогательными объектами управления. Всего объект формирования уникальной перестановки содержит $d \cdot (n - 1)$ регистров ($n - 1$ групп по d регистров) сдвига и $n - 1$ вспомогательных объектов управления. Ширина регистров в каждой группе ступенчато увеличивается от 2 до n бит с шагом 1 в соответствии с увеличением полезной длины символьной строки согласно алгоритму циклического сдвига. Каждый вспомогательный объект управления координирует работу своей группы

из d регистров с помощью сигналов управления, поступающих на входные управляющие порты регистров. Отметим, что отдельно взятый регистр в своей группе отвечает за определенный бит в элементе перестановки. Таким образом, моделируется управляемый циклический сдвиг на один элемент в рамках текущей перестановки (что на самом деле является сдвигом на d бит).

Для управления элементами перестановки, всякий регистр, помимо информационных портов, также обладает и двумя входными портами управления L и S. По переднему фронту сигнала на порте L регистр зачитывает соответствующие биты элементов перестановки с входных информационных портов. По переднему фронту сигнала на порте L в регистре осуществляется сдвиг на 1 бит. При низком уровне сигнала на обоих портах текущее состояние регистра остается неизменным.

Вспомогательный объект управления включает в себя два счетчика: счетчик по модулю M_1 и счетчик по модулю M_2. Счетчики устроены таким образом, что каждый из них по достижении своего максимального значения подает на выход высокий уровень сигнала, в противном случае – низкий уровень. Внутреннее состояние счетчика увеличивается при подаче очередного тактирующего импульса на входной порт тактирующего сигнала и сбрасывается по достижении максимального значения. Согласно значениям сигналов на выходе обоих счетчиков рассчитываются уровни управляющих сигналов L и S. Значения M_1 и M_2 вычисляются по формулам:

$$M_1 = \prod_{i=l}^{n} i,$$

$$M_1 = \begin{cases} 1, l = n, \\ \prod_{i=l}^{n} i, l < n, \end{cases}$$

где l – разрядность соответствующего управляемого регистра. Все счетчики обладают входным портом сброса RESET. Общая логическая схема формирователя очередной уникальной перестановки изображена на рис. 5.

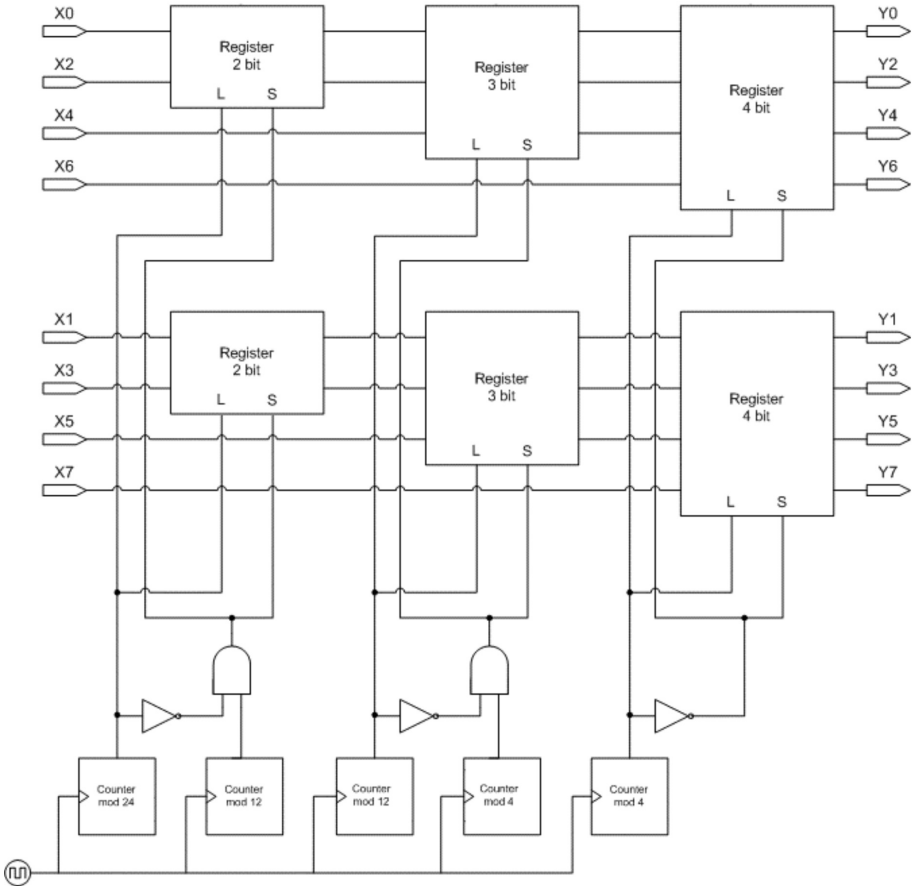

Рис. 5. Логическая схема формирователя очередной уникальной перестановки ($n = 4$, $d = 2$)

Объект приема перестановок представляет собой $(n \cdot d)$-разрядный регистр с параллельным вводом и выводом. Запись данных в регистр осуществляется по переднему фронту тактирующего сигнала и только при высоком уровне сигнала разрешения записи WE.

Для анализа, мониторинга и верификации блока генерации перестановок на RTL-уровне, использовалась специальная тестовая среда окружения. Для исходного множества $X_0 = \{0,1,2,3\}$ $(n = 4, d = 2)$ временная диаграмма процесса работы блока генерации изображена на рис. 6.

Помимо тактирующего сигнала и вектора элементов сгенерированной перестановки на диаграмме указаны значения управляющих сигналов Sm и Lm , где m – разрядность соответствующих регистров сдвига.

На диаграмме также указывается значения элементов сгенерированной перестановки на каждом тактовом импульсе в десятичной системе счисления. Звездочкой указаны перестановки, имеющие хотя бы одну неподвижную точку. Необходимо отметить, что в первые два тактовых импульса происходит заполнение регистров сдвига, и значения элементов выходного вектора не определены.

ЗАКЛЮЧЕНИЕ

В данной работе было предложены системная и RTL модели функционального генератора комбинаторных перестановок на основе заданного множества исходных целочисленных элементов мощности n. Генератор обладает довольно простой реализацией и гибкостью управления. Разработанные модели могут использоваться как программные IP блоки для автоматизации проектирования специализированных устройств в области обеспечения информационной безопасности, в частности, для систем динамического преобразования форматов представления данных в ЭВМ. Целевыми устройствами могут быть программируемые вентильные матрицы (FPGA), проблемно-ориентированные интегральные микросхемы (ASIC) и системы на кристалле (SoC).

СПИСОК ЛИТЕРАТУРЫ

1. *Pat U. S.* No. 5,734,721 Bl H04L9/00 Anti-spoof without error extension (ANSWER)/Clark James Monroe . – March 31, 1998.

2. *Ritter T.* Transposition Cipher with Pseudo-Random Shuffling: The Dynamic Transposition Combiner / T.Ritter // Cryptologia. – 1991. V. 15 (1). – P. 1–17.

3. *Курейчик В. М.* Комбинаторные аппаратные модели и алгоритмы в САПР / В. М. Курейчик, В. М. Глушань, Л. И. Щербаков; М.: Радио и связь. – 1990.

4. *Молодченко Ж. А.* Динамическое форматирование представлений объектов реляционных СУБД на основе кластерных транспозиций / Ж. А. Молодченко, Л. С. Сотов, В. Н. Харин // Естественные и технические науки. – М.: Изд-во компании “Спутник +”. – 2007, № 6 (32). – С. 224–226.

5. *Соболев С. С.* Алгоритм локализации полных комбинаторных перестановок с применением циклического сдвига / С. С. Соболев, В. Н. Харин, Л. С. Сотов; Фед. агентство по образованию, Гос. образоват. учреждение высш. проф. образования,

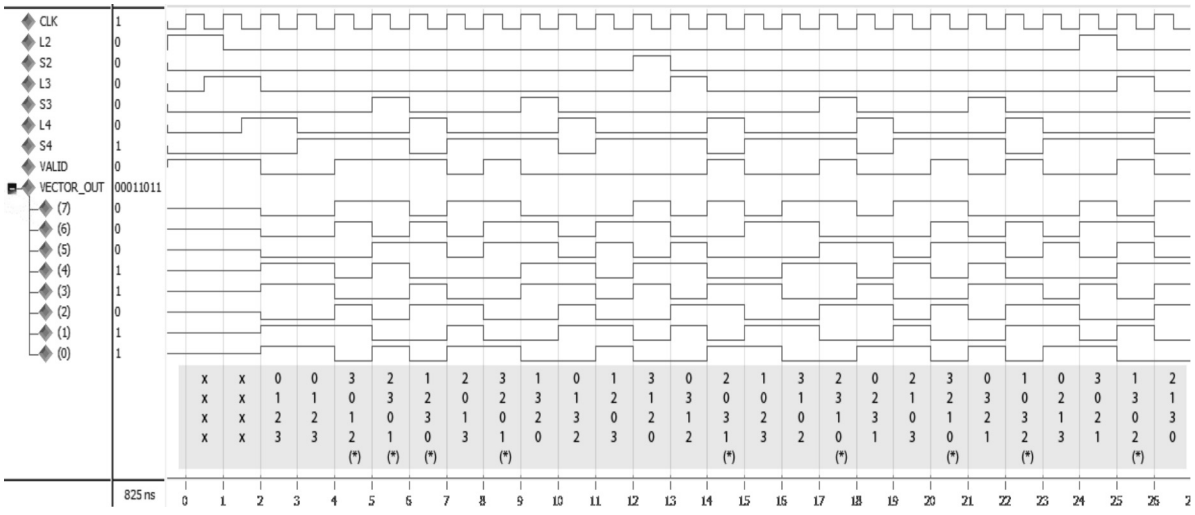

Рис. 6 Временная диаграмма процесса работы блока генерации перестановок $[\![(X]\!]_0 = \{0,1,2,3\}, n = 4, d = 2)$

Воронеж. гос. лесотехн. акад. – Воронеж, 2010. – 6 с. : ил. – Библиогр.: с. 6. – Деп. в ВИНИТИ.

6. IEEE Std 1666-2005 SystemC Language Reference Manual / The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA. – 2008.

7. IEEE Std 1076-2008 VHDL Language Reference Manual / The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA. – 2009.

8. *Бибило П.Н.* Системы моделирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum [Текст] / П.Н. Бибило. – М.: СОЛОН Пресс, 2005. – 384 с.

Соболев Сергей Сергеевич – аспирант , Воронежская государственная лесотехническая академия. Тел.: 8 (920) 426-53-62, 8 (473) 272-60-44. E-mail: roygbiv86@gmail.com

Sobolev S. S. – Post-Graduate Student. Voronezh State Academy of Forestry and Technologies. Tel.: 8 (920) 426-53-62, 8 (473) 272-60-44. E-mail: roygbiv86@gmail.com