

Научная статья

DOI: 10.15593/2224-9397/2025.1.06

УДК 004.89

И.А. Васенин¹, О.В. Гончаровский¹, С.Ф. Тюрин^{1,2}¹Пермский национальный исследовательский политехнический университет, Пермь, Российская Федерация²Пермский государственный национальный исследовательский университет, Пермь, Российская Федерация

СИНТЕЗ ЭЛЕМЕНТОВ ПЛИС, ИСПОЛЬЗУЮЩИХ КОМБИНИРОВАННОЕ КОДИРОВАНИЕ

Повышение быстродействия программируемой логики возможно путем уменьшения количества передающих транзисторов n -MOS в цепи от ячейки памяти, хранящей соответствующее значение логической функции, до выхода, на котором формируется сигнал для данного набора переменных. Минимально возможное значение без учета входных и выходных инверторов-восстановителей логического уровня сигнала равно единице. Но в этом случае необходим так называемый унитарный код, фиксирующий истинность данного набора совершенной дизъюнктивной нормальной формы (СДНФ), что требует $2n$ связей вместо n для существующего элемента LUT (Look Up Table), где n – количество переменных. В то же время при реализации коммутатора связей (мультиплексоров маршрутизации – Routing Multiplexers) на основе LUT уменьшается количество ячеек памяти с $2n$ до n , но зато путь сигнала содержит n транзисторов вместо одного. В статье рассматривается синтез «среднего» варианта элемента с комбинированным кодированием, позволяющим уменьшить число транзисторов в пути сигнала, не слишком увеличивая число линий связи переменных или объем памяти настроек мультиплексоров маршрутизации. Целью исследования является разработка алгоритма и программы синтеза элемента ПЛИС с использованием комбинированного кодирования переменных для вычисления логических функций или коммутации связей. Методы исследования базируются на научно-методическом аппарате дискретной математики, математической логики, цифровой схемотехники, алгоритмизации и программирования. В результате исследования разработаны алгоритм и программа синтеза элемента ПЛИС с использованием комбинированного кодирования переменных для вычисления логических функций или коммутации связей по заданным параметрам. Разработан алгоритм синтеза предлагаемого логического элемента. Приведены примеры работы алгоритма при построении дерева передающих транзисторов существующего LUT, мультиплексора маршрутизации и предлагаемого элемента с комбинированным кодированием. Практическая значимость: использование элементов с комбинированным кодированием позволяет повысить быстродействие при вычислении логических функций за счет некоторого увеличения числа связей переменных или уменьшить число ячеек памяти конфигураций за счет некоторого снижения быстродействия. Подана заявка на регистрацию программы.

Ключевые слова: логическая функция, ПЛИС, LUT, унитарный код.

I.A. Vasenin¹, O.V. Goncharovsky¹, S.F. Tyurin^{1,2}

¹Perm National Research Polytechnic University, Perm, Russian Federation

²Perm State National Research University, Perm, Russian Federation

DESIGN OF FPGA ELEMENTS USING COMBINED CODING

The performance of programmable logic can be increased by reducing the number of n-MOS transmitting transistors in the chain from the memory cell storing the corresponding value of the logical function to the output, where the signal for the given set of variables is formed. The minimum possible value without taking into account the input and output inverters-recoverers of the logical level of the signal is equal to one. But in this case, the so-called unitary code is required, which fixes the truth of the given set of perfect disjunctive normal form (PDNF), which requires $2n$ connections instead of n for the existing LUT (Look Up Table) element, where n is the number of variables. At the same time, when implementing a connection switch (Routing Multiplexers) based on LUT, the number of memory cells is reduced from $2n$ to n , but the signal path contains n transistors instead of one. The article discusses the synthesis of an "average" version of an element with combined coding, which allows reducing the number of transistors in the signal path without significantly increasing the number of variable connection lines or the memory capacity of the routing multiplexers settings. **The aim of the study** is the development of an algorithm and program for synthesizing an FPGA element using combined coding of variables for calculating logical functions or switching connections. **The research methods** are based on the scientific and methodological apparatus of discrete mathematics, mathematical logic, digital circuitry, algorithmization and programming. As a result of the study, an algorithm and program for synthesizing an FPGA element using combined coding of variables for calculating logical functions or switching connections according to specified parameters have been developed. An algorithm for synthesizing the proposed logical element has been developed. Examples of the algorithm's operation are given for constructing a tree of transmitting transistors of an existing LUT, a routing multiplexer, and the proposed element with combined coding. Discussion: the use of elements with combined coding allows to increase the speed of action when calculating logical functions due to some increase in the number of variable connections or to decrease the number of configuration memory cells due to some decrease in speed. An application for registration of the program has been submitted.

Keywords: Logic Functions, FPGA, LUT, One-Hot Coding.

Введение

Программируемые пользователем вентильные матрицы (ППВМ) или программируемые логические интегральные схемы (ПЛИС, «гибкая логика») [1–7] позволяют получать более быстродействующие аппаратные решения, чем программные («алгоритмическая логика»), но они как универсальные устройства с программируемыми функциями и связями, конечно, проигрывают узкоспециализированным заказным микросхемам ASIC (Application-Specific Integrated Circuit, «жесткая логика»), безусловно, выигрывая у них в стоимости за счет большего объема выпуска. «Полугибкая логика» – это полузаказные микросхемы, так называемые базовые матричные кристаллы (БМК, ULA – Uncommitted Logic Array), программируются только на заводском оборудовании.

Области применения ПЛИС постоянно расширяются [8–11] вслед за совершенствованием технологий их производства и характеристик [12, 13], поэтому актуальны и исследования в области реализации логических функций и автоматов в ПЛИС. Совершенствованию логики ПЛИС посвящены работы [14–20].

Совершенствованию элементов ПЛИС типа FPGA (Field-Programmable Gate Array), известных как LUT (Look Up Table) и маршрутизатор связей (Routing Multiplexers), посвящены работы [14, 21]. Разработаны модель и метод синтеза такого элемента с использованием комбинированного кодирования, сочетающего обычное бинарное (позиционное) кодирование и унитарное кодирование, когда активен (равен нулю – One-Cold Coding или единице – One-Hot Coding) только один бит. Схемотехническое и топологическое виды моделирования [22, 23] подтвердили работоспособность технических решений.

Однако в ранних работах алгоритм синтеза разработан не в полной мере, а также не получена программа синтеза, что затрудняет использование новых элементов в технологиях создания перспективных ПЛИС. Следовательно, необходимо детализировать научно-методический аппарат синтеза таких новых предложенных элементов.

Целями представленной работы являются разработка и исследование предлагаемого алгоритма и программы синтеза элемента ПЛИС с использованием комбинированного кодирования – для вычисления логических функций, позволяющего повысить быстродействие при вычислении логической функции, увеличивая до заданного значения число линий связи переменных или снизить объем памяти настроек – для мультиплексоров маршрутизации связей, снижая быстродействие не ниже заданного значения количества транзисторов в пути сигнала.

1. Разработка алгоритма синтеза элементов ПЛИС типа FPGA, использующих комбинированное кодирование

Дано выражение (1) элементарных мультиплексоров 2-1 из двух передающих транзисторов n -МОП (или двух КМОП ключей [1, 2]) $j.k.x, j.k.x'$ k -уровень дерева от n до 1 (старшая переменная – уровень 1, но переменная n):

$$\begin{matrix} j.k.s_0 - j.k.x' \\ j.k.s_1 - j.k.x \end{matrix}](\vee \bullet) j.k.f. \quad (1)$$

В (1) « $\leftarrow$ » означает соответствующую связь, « $\vee \bullet$ » означает корень дерева, реализующий операцию «Монтажное ИЛИ». В этом случае сигналы $j.k.x, j.k.x'$ инверсны (ортогональны), то есть на выходе $i.f$ всегда имеется некоторый логический уровень. На рис. 1 показана реализация (1) в виде мультиплексора 2–1 (1-LUT): а) на базе n -МОП (n -MOS) транзисторов; б) на базе КМОП (n -MOS, p -MOS: Complementary MOS, CMOS) транзисторов.

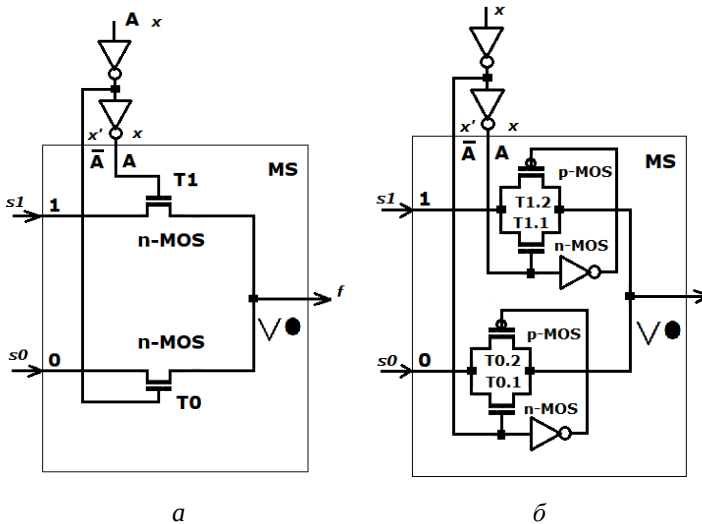


Рис. 1. Мультиплексор 2 – 1 (1-LUT): а – на базе n -МОП транзисторов; б – на базе КМОП транзисторов

Транзистор n -МОП T_0 (рис. 1, а) получает на исток некоторый сигнал $i.s.0$, который в случае $i.x'=1$ передает на выход $i.f$ сигнал $i.s.1$. Транзистор n -МОП T_1 получает на исток некоторый сигнал $i.s.1$, который в случае $i.x=1$ передает на выход $i.f$ сигнал $i.s.1$. Аналогично работают КМОП-ключи, показанные на рис. 1, б, более предпочтительные при передаче сигналов уровня логического нуля, но и более сложные.

Также имеется выражение (2) элементарных коммутаторов 2-1 из двух передающих транзисторов n -МОП $i.x, i.x''$, где i – номер мультиплексора (или итерации):

$$\begin{matrix} i.s_0 - i.x'' \\ i.s_1 - i.x \end{matrix}](\vee \bullet) i.f. \quad (2)$$

Сигналы $i.x$, $i.x''$ таковы, что если один равен единице, то другой равен нулю, иначе оба нуля. То есть на выходе $i.f$ может и не быть логического уровня. На рис. 2 показана реализация (2) в виде коммутатора 2 – 1 (1-LUT): а) на базе n -МОП транзисторов; б) на базе КМОП транзисторов.

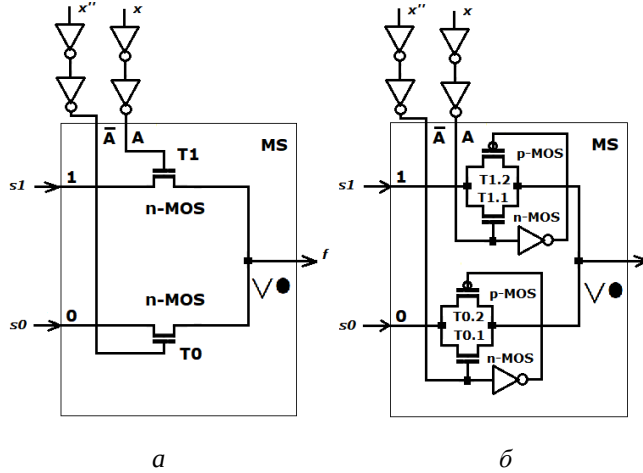


Рис. 2. Коммутатор (унитарный 1-LUT) 2 – 1: а – на базе n -МОП транзисторов; б – на базе КМОП транзисторов

Требуется синтезировать комбинированный элемент LUT с использованием комбинированного кодирования n переменных, $n = n_2 + n_1$, где n_1 – реализуется обычным LUT по выражениям (1), а n_2 – унитарным LUT по выражениям (2). При этом n_1 и n_2 могут быть в двух вариантах: а) (n_2, n_1) ; б) (n_1, n_2) .

Для этого предлагается выполнить следующие этапы (рассмотрим вариант а).

1. Синтез унитарных блоков.

1.1. Синтез одного унитарного блока:

1.1.1. $i = 1$. Соединить $i.f$ с $i + 1.f$ – выражение (3) (рис. 3).

$$\left[\begin{array}{l} i.s_0 - i.x'' \\ i.s_1 - i.x \\ i+1.s_0 - i+1.x'' \\ i+1.s_1 - i+1.x \end{array} \right] (\vee \bullet) i.f \quad (\vee \bullet). \quad (3)$$

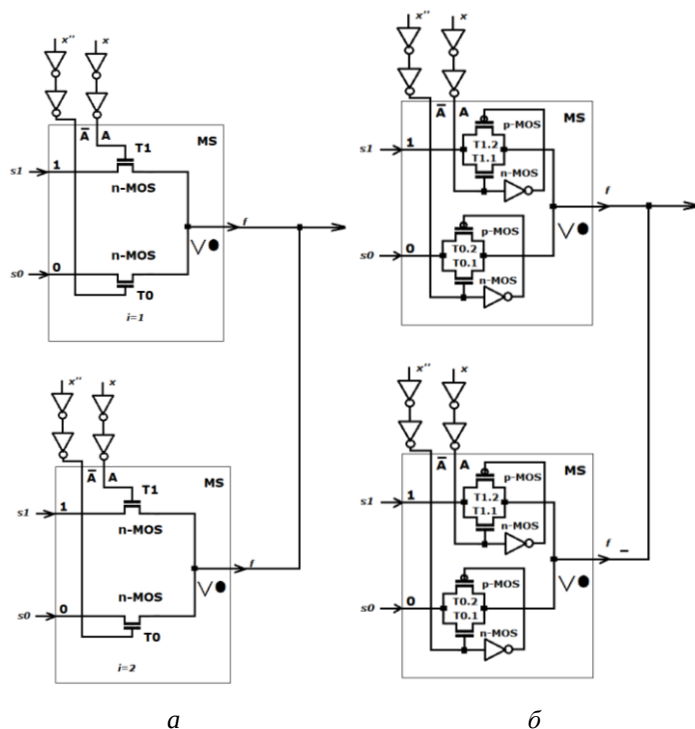


Рис. 3. Коммутатор (унитарный 2-LUT) 4 – 1: *а* – на базе *n*-МОП транзисторов; *б* – на базе КМОП транзисторов

1.1.2. Повторить пункт 1.1.1 $2^{n_2-1} - 1$ раз, n_2 – размерность унитарного блока.

В результате получают выражение:

$$\left. \begin{array}{l} i.s_0 - i.x'' \\ i.s_1 - i.x \\ i+1.s_0 - i+1.x'' \\ i+1.s_1 - i+1.x \\ \vdots \\ i+2^{n_2}.s_0 - i+2^{n_2}.x'' \\ i+2^{n_2}.s_1 - i+2^{n_2}.x \end{array} \right\} \left. \begin{array}{l}](\vee \bullet) i.f \\](\vee \bullet) i.f \\](\vee \bullet) i.f \\](\vee \bullet) i.f \\](\vee \bullet) i.f \end{array} \right\} (\vee \bullet) \dots \left. \right\} (\vee \bullet). \quad (4)$$

Далее, если производится вычисление логической функции $f(dx)$, выполняется следующее:

1.1.3. Подключить $i.s$ к соответствующим выходам конфигурационной памяти d ,

1.1.4. Подключить $i.x$, $i.x''$ к соответствующим входам переменных унитарного блока (унитарной части кода переменных) для реализации логической функции $f(dx)$:

$$\left[\begin{array}{l} d_0(i.s_0)-i.x'' \\ \quad](\vee\bullet)i.f(dx) \\ d_1(i.s_1)-i.x \\ d_2(i+1.s_0)-i+1.x'' \\ \quad](\vee\bullet)i.f(dx) \\ d_3(i+1.s_1)-i+1.x \\ \cdot \\ \cdot \\ d_{2^{n_2}-2}(i+2^{n_2}-2.s_0)-i+2^{n_2}-2.x'' \\ \quad](\vee\bullet)i.f(dx) \\ d_{2^{n_2}-1}(i+2^{n_2}-1.s_1)-i+2^{n_2}-1.x \end{array} \right] \begin{array}{l} (\vee\bullet)... \\ \\ \\ (\vee\bullet) \end{array} \quad (5)$$

Если реализуется коммутатор, то необходимо подключить $i.s$ ко входам матрицы межсоединений q для реализации $f(qd)$, по входам $i.x$, $i.x''$ осуществляется настройка соответствующими выходами конфигурационной памяти d :

$$\left[\begin{array}{l} q_0(i.s_0)-i.x''(d_0) \\ \quad](\vee\bullet)i.f(qd) \\ q_1(i.s_1)-i.x(d_1) \\ q_2(i+1.s_0)-i+1.x''(d_2) \\ \quad](\vee\bullet)i.f(qd) \\ q_3(i+1.s_1)-i+1.x(d_3) \\ \cdot \\ \cdot \\ q_{2^{n_2}-2}(i+2^{n_2}-2.s_0)-i+2^{n_2}-2.x''(d_{2^{n_2}-2}) \\ \quad](\vee\bullet)i.f(qd) \\ q_{2^{n_2}-1}(i+2^{n_2}-1.s_1)-i+2^{n_2}-1.x(d_{2^{n_2}-1}) \end{array} \right] \begin{array}{l} (\vee\bullet)... \\ \\ \\ (\vee\bullet). \end{array} \quad (6)$$

Выражения (5), (6) содержат один транзистор в пути сигнала от d до $f(dx)$ или от q до $f(qd)$, поэтому своего рода «внутренние» восстановители не требуются, однако они устанавливаются на входах и выходах. По входам настройки (или переменных) – это пара инверторов, по входам связей – это инверторы, а по выходу – тоже инвертор NOT или восстановитель уровня.

2. *Синтез j позиционных блоков.* Для синтеза многоуровневого ($k = 1 \dots n_1$) дерева j -го блока необходимо соединить один блок $j.1.f_1$ с выходным инвертором NOT (восстановителем), $j.1.x, j.1.x'$ с соответствующими входами x_{n1}, x'_{n1} :

$$\begin{array}{l} j.1.s_0 - (j.1.x')x_{n1} \\ j.1.s_1 - (j.1.x)x_{n1} \end{array}](\vee \bullet) j.1.f_1(NOT). \quad (7)$$

Далее соединяются два блока $(j+1.2.f_2, j+1.2.f_2), (j+2.2.f_2, j+2.2.f_2)$ с соответствующими входами $j.n_1.s1; j.n_1.s2; j.1.x, j.1.x'$ и соответствующими входами переменных x_{n1-1}, x'_{n1-1} :

$$\begin{array}{l} j+3.2.s_0 - (j+3.2.x')x_{n1-1} \\ j+3.2.s_1 - (j+3.2.x)x_{n1-1} \end{array}](\vee \bullet) j+1.2.f_2 \bullet j.1.s_1 - (j.1.x')x_{n1} \\ \begin{array}{l} j+4.2.s_0 - (j+2.2.x')x_{n1-1} \\ j+4.2.s_1 - (j+2.2.x)x_{n1-1} \end{array}](\vee \bullet) j+2.2.f_3 \bullet j.1.s_2 - (j.1.x)x_{n1} \end{array}](\vee \bullet) j.1.f_1(NOT). \quad (8)$$

Выполняются вышеуказанные действия $2^{n_1} - 2$ раз, n_1 – размерность позиционного блока до подключения всех позиционных переменных, если не задано ограничение Мида–Конвей.

Получаем одно дерево. В случае задания ограничения Мида–Конвей [24] необходимо строить несколько деревьев, соединяя их для получения общего дерева.

Так, для ограничения (≤ 3) требуемое n дерево строится как композиция деревьев $n = 1$ (7), $n = 2$ (8), $n = 3$.

Такие деревья генерируются, исходя из требуемого n , а затем соединяются в одно дерево. Так, для $n = 5$ получают $(3 + 3 + 1 = 4)$,

$(3 + 3 + 1 = 4)$, $(4 + 4 + 1 = 5)$, здесь знак «+» означает соединение деревьев. Но возможна и другая декомпозиция: $(3 + 3 + 3 + 3) + 2 = 5$.

3. *Соединение позиционных и унитарных блоков.* Соединение требуемых типов позиционных блоков и унитарных в двух вариантах.

Первый вариант: соединить $i.f$ всех i унитарных блоков с соответствующими входами $j.n_2.s1$; $j.n_2.s2$ всех $j.n_2$ позиционных блоков (рассматриваем многоуровневое дерево для вычисления логической функции). Так, для унитарного на $2^{n^2} = 4$ переменных и позиционного на одну $n_1 = 1$ получаем выражение (9). Здесь позиционный блок справа.

$$\left[\begin{array}{l} i.s_0 - i.x^n \\ \vdots \\ i.s_1 - i.x \\ i+1.s_0 - i+1.x^n \\ \vdots \\ i+1.s_1 - i+1.x \end{array} \right] \begin{array}{l}](\vee \bullet) i.f \\ \\](\vee \bullet) i.f \end{array} \left(\vee \bullet \right) [(\vee \bullet) \cdot j.k.s_0 - (j.k.x^n) x_{n_1} \\ \\](\vee \bullet) i.f \left(\vee \bullet \right) j.k.f(dx_{n_1+n_2})(NOT). \quad (9)$$

$$\left[\begin{array}{l} i.s_0 - i.x^n \\ \vdots \\ i.s_1 - i.x \\ i+1.s_0 - i+1.x^n \\ \vdots \\ i+1.s_1 - i+1.x \end{array} \right] \begin{array}{l}](\vee \bullet) i.f \\ \\](\vee \bullet) i.f \end{array} \left(\vee \bullet \right) [(\vee \bullet) \cdot j.k.s_1 - (j.k.x) x_{n_1}$$

Второй вариант: соединить $i.f$ всех i позиционных блоков с соответствующими входами унитарных блоков, получаем выражение (10), здесь унитарный блок на 2^{n^2} переменных справа, позиционный на одну переменную $n_1 = 1$:

$$\left[\begin{array}{l} (i.s_0) - (i.x^n) x_0 \\ \vdots \\ (i.s_1) - (i.x) x_1 \\ \vdots \\ (i+1.s_0) - (i+1.x^n) x_2 \\ \vdots \\ (i+1.s_1) - (i+1.x) x_3 \\ \vdots \\ \vdots \\ \vdots \\ (i+2^{n^2}-2.s_0) - (i+2^{n^2}-2.x^n) x_{2^{n^2}-2} \\ \vdots \\ (i+2^{n^2}-1.s_1) - (i+2^{n^2}-1.x) x_{2^{n^2}-1} \end{array} \right] \begin{array}{l}](\vee \bullet) i.f(dx) \\ \\](\vee \bullet) i.f(dx) \\ \\](\vee \bullet) i.f(dx) \end{array} \left(\vee \bullet \right) \dots \left(\vee \bullet \right) . \quad (10)$$

2. Разработка алгоритма и программы синтеза элемента с комбинированным кодированием

Предлагаемая схема алгоритма синтеза элемента с комбинированным кодированием показана на рис. 4.

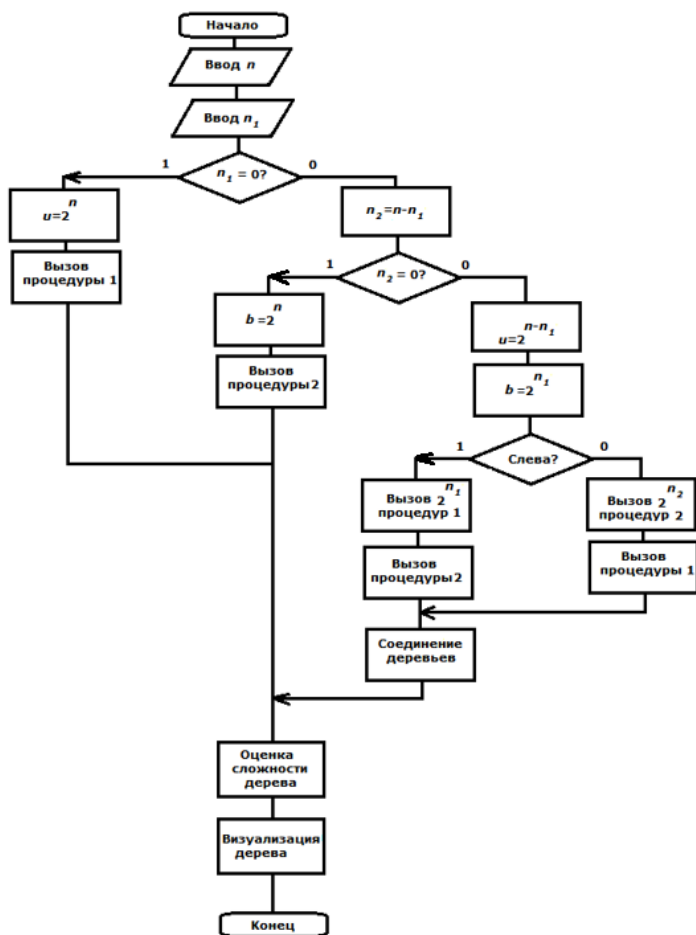


Рис. 4. Схема алгоритма синтеза элемента с комбинированным кодированием

Разработанная программа синтеза на языке Python строит унитарное дерево по параметру n_2 , если он не равен нулю. Далее строится полное бинарное дерево по параметру n_1 , если он не равен нулю. У бинарного дерева 2^{n_1} информационных входов s , которые подключаются к выходам соответствующих 2^{n_1} унитарных деревьев 2^{n_2} . К управляющим входам унитарных деревьев подключаются

сдвоенные инверторы унитарных переменных. К управляющим входам полного бинарного дерева сдвоенные инверторы бинарных переменных. При необходимости выполняется обратное подключение – слева позиционные полные бинарные деревья, справа только одно унитарное. Визуализируется итоговое комбинированное дерево.

Заключение

Предлагаемый алгоритм и программа на языке Python позволяют разработчикам ПЛИС проектировать новые элементы, которые могут повысить эффективность вычисления логических функций и реализацию коммутаторов локальных и глобальных связей. Проведено тестирование на типовых вариантах размерностей элемента. Результаты работы программы проверены с помощью систем схемотехнического и топологического моделирования [22, 23]. Интеллектуальная собственность защищена патентом РФ (патентообладатель – ПНИПУ) [21]. Авторами подана заявка на регистрацию программы (патентообладатель – ПНИПУ). Использование нового элемента в дальнейшем позволит доработать соответствующие САПР ПЛИС в плане введения дополнительных функций выбора параметров кодирования. Полученные результаты также возможно использовать при разработке новых вариантов БМК с универсальными элементами.

Библиографический список

1. Строгонов, А. Современные тенденции развития ПЛИС: от системной интеграции к искусственному интеллекту / А. Строгонов, П. Городков // *Электроника: Наука, технология, бизнес*. – 2020. – № 4 (195). – С. 46–56.
2. Строгонов, А. Проектирование конечных автоматов в приложении STATEFLOW системы MatLab / Simulink с последующей реализацией в базисе ПЛИС / А. Строгонов // *Электроника: Наука, технология, бизнес*. – 2023. – № 3 (224). – С. 134–147.
3. Development of a device for multiplying numbers by means of FPGA / N.M. Berezin, I.E. Chernetskaya, V.S. Panishchev, A.M. Shabarov // Published under licence by IOP Publishing Ltd. *Journal of Physics: Conference Series*. Vol. 2142, XI International Conference on High-performance computing systems and technologies in scientific research, automation of

control and production (HPCST 2021); 21–22 May 2021. – Barnaul, Russia, 2021. DOI: 10.1088/1742-6596/2142/1/012001

4. Методика проектирования преобразователя кода Грея на ПЛИС / А.А. Пирогов, Ю.А. Пирогова, А.В. Башкиров, М.Ю. Чепелев, Б.И. Жилин // Вестник Воронеж. ин-та ФСИН России. – 2020. – № 3. – С. 9–14.

5. Строгонов, А. Обзор программных средств с открытым исходным кодом для исследования современных архитектур ПЛИС XILINX / А. Строгонов, М. Кривчун, П. Городков // Электроника: Наука, технология, бизнес. – 2020. – № 1 (192). – С. 100–107.

6. Арбузов, И. Пример разработки проекта в базисе ПЛИС 5578TC024 / И. Арбузов, А. Строгонов, П. Городков // Компоненты и технологии. – 2019. – № 7 (216). – С. 66–69.

7. Строгонов, А. Обзор ПЛИС китайских производителей [Электронный ресурс] / А. Строгонов, П. Городков. – URL: https://www.elibrary.ru/download/elibrary_48565021_33092934.pdf (дата обращения: 17.11.2023).

8. Enhancing blockchain security and efficiency through FPGA-based consensus mechanisms and post-quantum cryptography / Jalel Ktari, Tarek Frikha, Monia Hamdi, Habib Hamam // IEEE Access. – May 2024. DOI: 10.2174/0123520965288815240424054237

9. FPGA accelerated post-quantum cryptography / He Li, Yongming Tang, Zhiqiang Que, Jiliang Zhang // IEEE Transactions on Nanotechnolog. – January 2022. – 99. – P. 1–7. DOI: 10.1109/TNANO.2022.3217802

10. Levental, Maksim. Tensor networks for simulating quantum circuits on FPGAs [Электронный ресурс] / Maksim Levental. – August 2021. – URL: https://www.researchgate.net/publication/353941749_Tensor_Networks_for_Simulating_Quantum_Circuits_on_FPGAs (дата обращения: 17.11.2023).

11. Hemin, Rahimi. Optimum implementation of digital logic circuits on 3D FPGAs / Hemin Rahimi // Thesis for: MasterAdvisor: Dr. Hadi jahannirad. – November 2023. DOI: 10.13140/RG.2.2.24347.85283

12. Monther, Abusultan. A comparison of FinFET based FPGA LUT / Monther Abusultan, Sunil P. Khatri // Texas A&M University, College Station, TX, USA. – Published in ACM Great Lakes Symposium on VLSI, 2014. DOI: 10.1145/2591513.2591596

13. Развитие гибридной многоядерной рекуррентной архитектуры на ПЛИС / Ю.А. Степченко, Н.В. Морозов, Ю.Г. Дьяченко, Д.В. Хилько, Д.Ю. Степченко // Системы и средства информатики. – 2020. – Т. 30, № 4. – С. 95–101.

14. Тюрин, С.Ф. Логические элементы ПЛИС FPGA на основе комбинированного кодирования переменных / С.Ф. Тюрин, И.А. Васенин, С.И. Советов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2023. – № 46. – С. 83–107. DOI: 10.15593/2224-9397/2023.2.04

15. Тюрин, С.Ф. Логический элемент ПЛИС FPGA, реализующий функцию и дешифрацию набора переменных / С.Ф. Тюрин, С.И. Советов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2023. – № 47. – С. 5–31. DOI: 10.15593/2224-9397/2023.3.01

16. Советов, С.И. Разработка топологии многофункционального логического элемента ПЛИС / С.И. Советов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2023. – № 48. – С. 30–49. DOI: 10.15593/2224-9397/2023.4.02

17. Тюрин, С.Ф. Логический элемент программируемых логических интегральных схем FPGA, вычисляющий функцию одновременно с дешифрацией входных переменных / С.Ф. Тюрин, С.И. Советов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2024. – № 50. – С. 216–234. DOI: 10.15593/2224-9397/2024.2.11

18. Vikhorev, R. Universal logic cells to implement systems functions / R. Vikhorev // Conference of Russian Young Researchers in Electrical and Electronic Engineering. – IEEE, 2016. – P. 404–406. DOI: 10.1109/EIConRusNW.2016.7448197

19. Vikhorev, R. Improved FPGA logic elements and their simulation / R. Vikhorev // Conference of Russian Young Researchers in Electrical and Electronic Engineering. – IEEE, 2018. – P. 275–280. DOI: 10.1109/EIConRus.2018.8317080

20. Skornyakova, A.Yu. Self-timed LUT layout simulation / A.Yu. Skornyakova, R.V. Vikhorev // Conference of Russian Young Researchers in Electrical and Electronic Engineering. – IEEE, 2020. – P. 176–179. DOI: 10.1109/EIConRus49466.2020.9039374

21. Программируемое логическое устройство: пат. Рос. Федерация / Тюрин С.Ф., Васенин И.А., Степченков Ю.А. Дьяченко Ю.Г., Советов С.И. – 2811404; заявл. 02.08.2023; опубл. 11.01.2024.

22. National Instruments. Multisim [Электронный ресурс]. – URL: <http://www.ni.com/multisim/> (дата обращения: 12.07.2023).

23. Microwind & Dsch Version 3.5 [Электронный ресурс]. – URL: <https://www.yumpu.com/en/document/view/40386405/microwind-manual-lite-v35pdf-moodle> (дата обращения: 21.12.2024).

24. Mead, C.A. Introduction to VLSI Systems [Электронный ресурс] / C.A. Mead, L. Conway. – URL: https://www.researchgate.net/publication/234388249_Introduction_to_VLSI_systems (дата обращения: 12.07.2023).

References

1. Strogonov A., Gorodkov P. Sovremennye tendentsii razvitiia PLIS: ot sistemnoi integratsii k iskusstvennomu intellektu [Modern trends in the development of FPGAs: from system integration to artificial intelligence]. *Elektronika: Nauka, tekhnologiya, biznes*, 2020, no. 4 (195), pp. 46-56.

2. Strogonov A. Proektirovanie konechnykh avtomatov v prilozhenii STATEFLOW sistemy MatLab Simulink s posleduiushchei realizatsiei v bazise PLIS [Review of FPGAs from Chinese manufacturers]. *Elektronika: Nauka, tekhnologiya, biznes*, 2023, no. 3 (224), pp. 134-147.

3. Berezin N.M., Chernetskaya I.E., Panishchev V.S., Shabarov A.M. Development of a device for multiplying numbers by means of FPGA. Published under licence by IOP Publishing Ltd. *Journal of Physics: Conference Series*. Vol. 2142, XI International Conference on High-performance computing systems and technologies in scientific research, automation of control and production (HPCST 2021); 21-22 May 2021. Barnaul, Russia. DOI: 10.1088/1742-6596/2142/1/012001

4. Pirogov A.A., Pirogova Iu.A., Bashkirov A.V., Chepelev M.Iu., Zhilin B.I. Metodika proektirovaniia preobrazovatel'ia koda Greia na PLIS

[Gray Code Converter design methodology for FPGAs]. Vestnik Voronezhskogo instituta FSIN Rossii, 2020, no. 3, pp. 9-14.

5. Stroganov A., Krivchun M., Gorodkov P. Obzor programmnykh sredstv s otkrytym iskhodnym kodom dlia issledovaniia sovremennykh arkhitektur PLIS XILINX [Design of finite state machines in the STATEFLOW application of the MatLab Simulink system with subsequent implementation in the FPGA basis]. *Elektronika: Nauka, tekhnologiia, biznes*, 2020, no. 1 (192), pp. 100-107.

6. Arbuzov I., Stroganov A., Gorodkov P. Primer razrabotki proekta v bazise PLIS 5578TS024 [Review of open source software tools for researching modern FPGA architectures XILINX]. *Komponenty i tekhnologii*, 2019, no. 7 (216), pp. 66-69.

7. Stroganov A., Gorodkov P. Obzor PLIS kitaiskikh proizvoditelei [An example of project development based on FPGA 5578TS024], available at: https://www.elibrary.ru/download/elibrary_48565021_33092934.pdf (accessed 17 November 2023).

8. Ktari Jalel, Frikha Tarek, Hamdi Monia, Hamam Habib. Enhancing blockchain security and efficiency through FPGA-based consensus mechanisms and post-quantum cryptography. May 2024. DOI: 10.2174/0123520965288815240424054237

9. Li He, Tang Yongming, Que Zhiqiang, Zhang Jiliang FPGA accelerated post-quantum cryptography. IEEE Transactions on Nanotechnolog. January 2022, 99, pp. 1-7. DOI: 10.1109/TNANO.2022.3217802

10. Levental Maksim. Tensor networks for simulating quantum circuits on FPGAs. August 2021, available at: https://www.researchgate.net/publication/353941749_Tensor_Networks_for_Simulating_Quantum_Circuits_on_FPGAs (accessed 17 November 2023).

11. Hemin Rahimi. Optimum implementation of digital logic circuits on 3D FPGAs. Thesis for: MasterAdvisor: Dr. Hadi jahanirad. November 2023. DOI: 10.13140/RG.2.2.24347.85283

12. Monther Abusultan, Sunil P. Khatri. A comparison of FinFET based FPGA LUT. Texas A&M University, College Station, TX, USA. Published in ACM Great Lakes Symposium on VLSI, 2014. DOI: 10.1145/2591513.2591596

13. Stepchenkov Iu.A., Morozov N.V., D'iachenko Iu.G., Khil'ko D.V., Stepchenkov D.Iu. Razvitie gibridnoi mnogoiadernoi rekurrentnoi

arkhitektury na PLIS [Development of a hybrid multi-core recurrent architecture on FPGA]. *Sistemy i sredstva informatiki*, 2020, vol. 30, no. 4, pp. 95-101.

14. Tiurin S.F., Vasenin I.A., Sovetov S.I. Logicheskie elementy PLIS FPGA na osnove kombinirovannogo kodirovaniia peremennykh [FPGA gates using combined variables coding]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2023, no. 46, pp. 83-107. DOI: 10.15593/2224-9397/2023.2.04

15. Tiurin S.F., Sovetov S.I. Logicheskii element PLIS FPGA, realizuiushchii funktsiiu i deshifratsiiu nabora peremennykh [FPGA LUT implementing the function and decoding of a set of variables]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2023, no. 47, pp. 5-31. DOI: 10.15593/2224-9397/2023.3.01

16. Sovetov S.I. Razrabotka topologii mnogofunktsional'nogo logicheskogo elementa PLIS [Lay-out development of multifunctional logic element for FPGA]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2023, no. 48, pp. 30-49. DOI: 10.15593/2224-9397/2023.4.02

17. Tiurin S.F., Sovetov S.I. Logicheskii element programmiruemyykh logicheskikh integral'nykh skhem FPGA, vychisliaiushchii funktsiiu odnovremenno s deshifratsiei vkhodnykh peremennykh [FPGA logic element that calculates a function simultaneously with decoding input variables]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2024, no. 50, pp. 216-234. DOI: 10.15593/2224-9397/2024.2.11

18. Vikhorev R. Universal logic cells to implement systems functions. *Conference of Russian Young Researchers in Electrical and Electronic Engineering*. IEEE, 2016, pp. 404-406. DOI: 10.1109/EIConRusNW.2016.7448197

19. Vikhorev R. Improved FPGA logic elements and their simulation. *Conference of Russian Young Researchers in Electrical and Electronic Engineering*. IEEE, 2018, pp. 275-280. DOI: 10.1109/EIConRus.2018.8317080

20. Skornyakova A.Yu., Vikhorev R.V. Self-Timed LUT Layout Simulation. *Conference of Russian Young Researchers in Electrical and Electronic Engineering*. IEEE, 2020, pp. 176-179. DOI: 10.1109/EIConRus49466.2020.9039374
21. Tiurin S.F., Vasenin I.A., Stepchenkov Iu.A., D'iachenko Iu.G., Sovetov S.I. Programmiruемое logicheskoe ustroistvo [Programmable logic device]. Patent Rossiiskaia Federatsiia no. 2811404 (2024).
22. National Instruments. Multisim, available at: <http://www.ni.com/multisim/> (accessed 17 November 2023).
23. Microwind & Dsch Version 3.5, available at: <https://www.yumpu.com/en/document/view/40386405/microwind-manual-lite-v35pdf-moodle> (accessed 21 December 2024).
24. Mead C.A., Conway L. Introduction to VLSI Systems, available at: https://www.researchgate.net/publication/234388249_Introduction_to_VLSI_systems (accessed 12 Juli 2023).

Сведения об авторах

Васенин Иван Андреевич (Пермь, Российская Федерация) – аспирант кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: Vasenin.Ioann@yandex.ru).

Гончаровский Олег Владленович (Пермь, Российская Федерация) – кандидат технических наук, доцент, доцент кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: 35911953@mail.ru).

Тюрин Сергей Феофентович (Пермь, Российская Федерация) – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: tyurinsergfeo@yandex.ru); профессор кафедры «Математическое обеспечение вычислительных систем» Пермского государственного национального исследовательского университета (614990, Пермь, ул. Букирева, 15).

About the authors

Ivan A. Vasenin (Perm, Russian Federation) – Postgraduate Student of the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: Vasenin.Ioann@yandex.ru).

Oleg V. Goncharovsky (Perm, Russian Federation) – Ph. D. in Technical Sciences, Associate Professor, Associate Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: 35911953@mail.ru).

Sergey F. Tyurin (Perm, Russian Federation) – Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: tyurinsergfeo@yandex.ru); Professor at the Department of Software Computing Systems Perm State University (614990, Perm, 15, Bukireva str.).

Поступила: 06.02.2025. Одобрена: 19.02.2025. Принята к публикации: 15.03.2025.

Финансирование. Исследование не имело спонсорской поддержки.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов по отношению к статье.

Вклад авторов. Авторы сделали равноценный вклад в подготовку статьи.

Просьба ссылаться на эту статью в русскоязычных источниках следующим образом:

Васенин, И.А. Синтез элементов ПЛИС, использующих комбинированное кодирование / И.А. Васенин, О.В. Гончаровский, С.Ф. Тюрин // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2025. – № 53. – С. 111–129. DOI: 10.15593/2224-9397/2025.1.06

Please cite this article in English as:

Vasenin I.A., Goncharovsky O.V., Tyurin S.F. Design of FPGA elements using combined coding. *Perm National Research Polytechnic University Bulletin. Electrotechnics, information technologies, control systems*, 2025, no. 53, pp. 111-128. DOI: 10.15593/2224-9397/2025.1.06