

МОДЕЛИРОВАНИЕ ЛОГИЧЕСКОГО ЭЛЕМЕНТА ПЛИС FPGA ДЛЯ РЕАЛИЗАЦИИ СИСТЕМ ФУНКЦИЙ

С. Ф. Тюрин, А. С. Прохоров

Пермский национальный исследовательский политехнический университет

Поступила в редакцию 11.02.2017 г.

Аннотация. Моделируется логический элемент для реализации систем функций в ПЛИС типа FPGA (field-programmable gate array) – на основе модифицированного дерева транзисторов – дешифратора – DC LUT (Look Up Table). Такое дерево дешифратора DC LUT позволяет реализовать систему логических функций, и в результате существенно сокращаются аппаратные затраты в количестве транзисторов при незначительном увеличении задержки. Моделирование выполняется в системе схемотехнического моделирования NI Multisim 10 фирмы National Instruments Electronics Workbench Group и подтверждает работоспособность модифицированного технического решения.

Ключевые слова: логический элемент, LUT, ПЛИС, дерево транзисторов, логический элемент на основе дешифратора – DC LUT, реализация систем логических функций, NI Multisim 10.

Annotation. The article describes proposed gate for the realization of systems of functions in the FPGA (field-programmable gate array). The inverse tree to implement the decoder DC, which allows for using additional blocks or custom built to also transmit transistors system logic functions that can significantly reduce the cost of hardware in an amount transistors, a slight increase in latency. The simulation DC LUT is executed in the system NI Multisim 10 by National Instruments Electronics Workbench Group. Simulation confirms the efficiency of the proposed scheme DC LUT with a new block of disjunctions.

Keywords: transistor tree, LUT FPGA, Systems of Logic Functions, DC LUT FPGA, NI Multisim 10.

ВВЕДЕНИЕ

Логический элемент (ЛЭ) программируемых логических схем (ПЛИС) FPGA (field-programmable gate array) имеет своей основой постоянное запоминающее устройство ПЗУ, называемое Logic Cell, а чаще – LUT – Look Up Table (просмотровая таблица – имеется в виду таблица истинности). LUT выполнен на основе мультиплексора, который строится в виде дерева из элементарных мультиплексоров 2-1 на базе передающих МОП транзисторов, входы данных которого настраиваются так называемыми конфигурируемыми ячейками статической памяти SRAM [1]. На рис. 1 представлен пример LUT на два информационных (адресных) входа – дерево 4-1

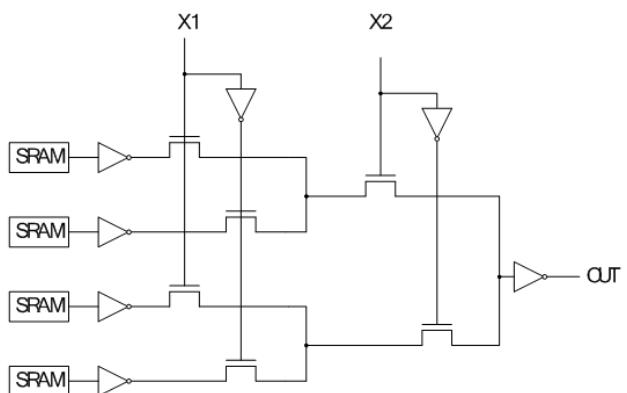


Рис. 1. LUT на два информационных (адресных) входа – дерево 4-1

10 фирмы National Instruments Electronics следующим образом – рис. 2.

Загружая в конфигурационную память SRAM значения таблицы истинности логической функции двух переменных можно реализовать 16 функций двух переменных $F(X_2X_1)$. На рис. 3 представлено моделирова-

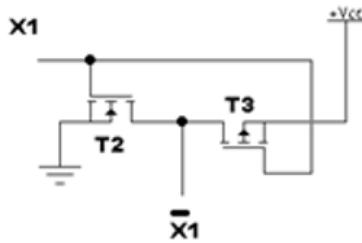


Рис. 2. Моделирование инвертора входа X_1

ние LUT на два информационных (адресных) входа – дерево 4-1, настроенного на реализацию логической функции $X_1 \oplus X_2$ (ячейки SRAM моделируются ключами d_0-d_3) рис. 3.

Стандартное число входов (адресных входов ПЗУ) как правило, равно четырём, однако современные «продвинутые» ПЛИС имеют сложные, перестраиваемые LUT с числом входов 6 и 7 [2–4]. Имеются сообщения о LUT с числом входов равным 8. Тем не менее, всегда реализуется только одна из возможных логических функций, информации о LUT, реализующей системы функций, автором найдено не было.

ЛОГИЧЕСКИЙ ЭЛЕМЕНТ DC – LUT

Транзисторная схема, «обратная» структуре LUT (рис. 1, 2) может быть получена путём «разворота» LUT на 180 градусов (рис. 2).

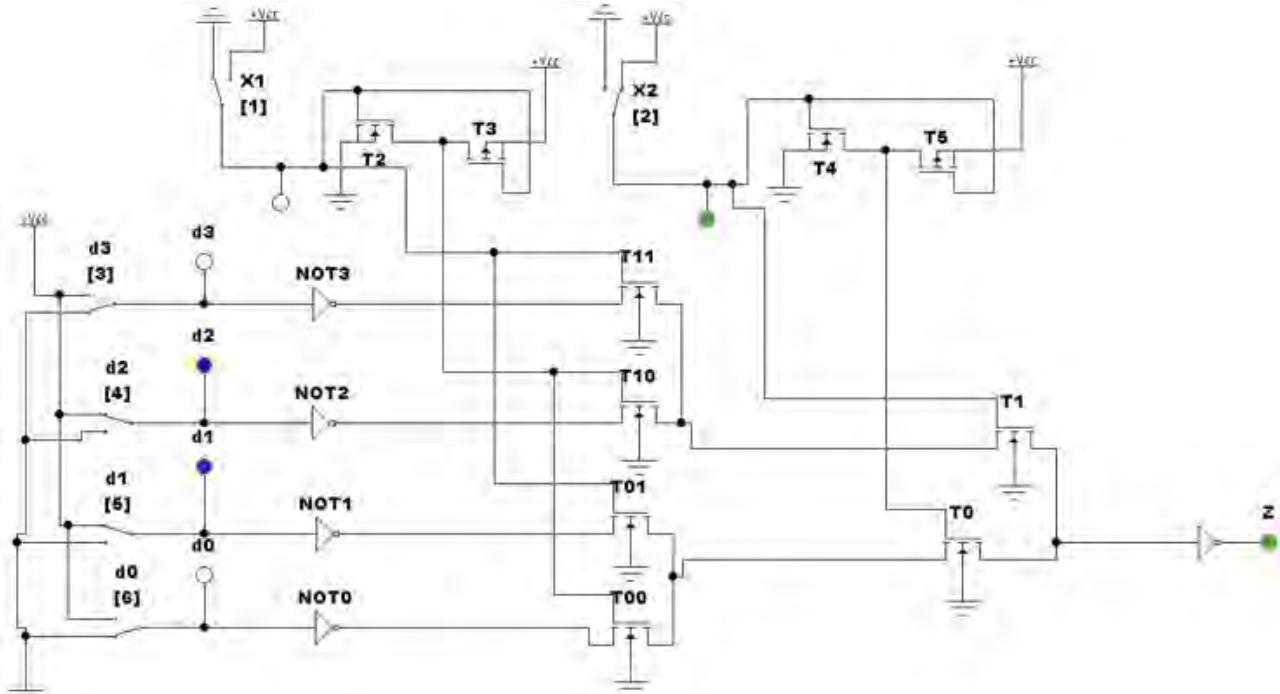


Рис. 3. Моделирование LUT на два информационных (адресных) входа – дерево 4-1, настроенного на реализацию логической функции $X_1 \oplus X_2$

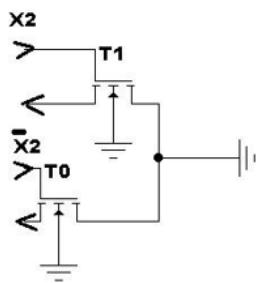


Рис. 4. Моделирование DC LUT на один информационный (адресный) вход X_2 – дерево 2-1

[5, 6]. Выполним её моделирование. Пусть по обратному дереву передаётся по ветви X_2 или \bar{X}_2 логический ноль – рис. 4.

Транзистор T_0 активируется по своему затвору инверсией X_2 , транзистор T_1 – сигналом X_2 . Далее строим второй уровень дерева на транзисторах T_{00} , T_{01} , T_{10} , T_{11} , получаем – рис. 5.

Инверторы по выходам необходимы для усиления сигнала, прошедшего передающие транзисторы. Непосредственная проверка подтверждает правильность дешифрации адреса на входах X_1X_2 . На рис. 5 видно, что при наборе 00 активируется нулевой выход Z_0 . Аналогично можно убедиться в правильности дешифрации набора 01 – активируется Z_1 , при наборе 10 – Z_2 , при наборе 11 – Z_3 .

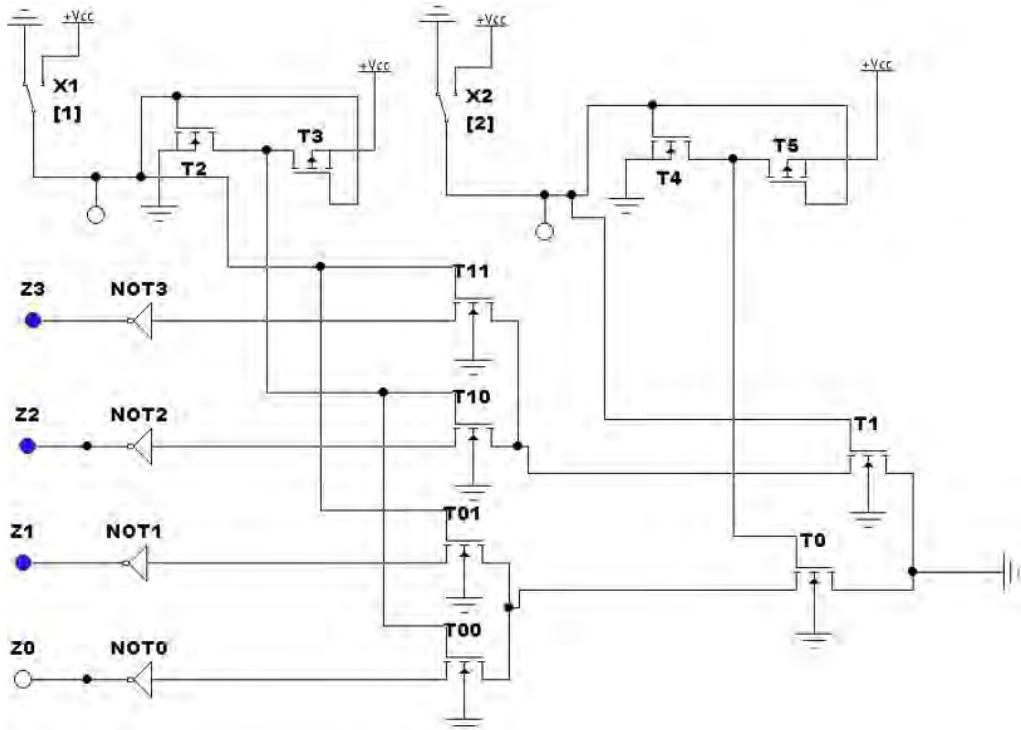


Рис. 5. Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 – обратное дерево 4-1

ЛОГИЧЕСКИЙ ЭЛЕМЕНТ DC – LUT С АЛЬТЕРНАТИВНЫМИ ТРАНЗИСТОРАМИ

В соответствие с правилами проектирования цепей из передающих транзисторов [7] необходимо для стока каждого транзистора T_{00} , T_{01} , T_{10} , T_{11} (рис. 5) создать альтернативную цепочку, гарантированно переводящую его выход, например, в единицу. Для этого используется выражение, двойственное конъюнкции, описывающей соответствующую цепочку в обратном дереве (рис. 5).

Однако лучшим вариантом является формирование альтернативного транзистора для

каждого отдельного передающего транзистора. Получаем уточнённую схему – рис. 7.

Получаем дешифратор с выходными функциями z (без учёта альтернативных цепочек):

$$\bar{z}_i = \& x_j^{\sigma_i^m},$$

где $i = 0 \dots 2^n - 1$, $j = 1 \dots n$.

Далее, m раз объединяя по логическому «ИЛИ», соответствующие выходы из 2^n выходов получим реализацию системы из m n -разрядных логических функций на основе совершенных дизъюнктивных нормальных форм (СДНФ) [5, 6].

ЛОГИЧЕСКИЙ ЭЛЕМЕНТ DC – LUT С АЛЬТЕРНАТИВНЫМИ ТРАНЗИСТОРАМИ И БЛОКОМ ДИЗЪЮНКЦИЙ

Для структуры, изображённой на рис. 2 могут программироваться связи выходов 0, 1, 2, 3 с соответствующими элементами «ИЛИ» с целью реализации систем логических функций. Такое программирование может быть реализовано аналогично программированию межсоединений [1].

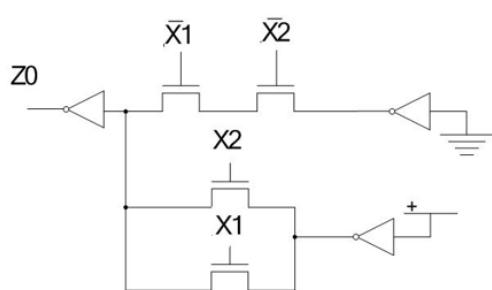


Рис. 6. Нулевой разряд DC LUT2 с альтернативной цепочкой, приведённой к выходу нулевой ветви обратного дерева

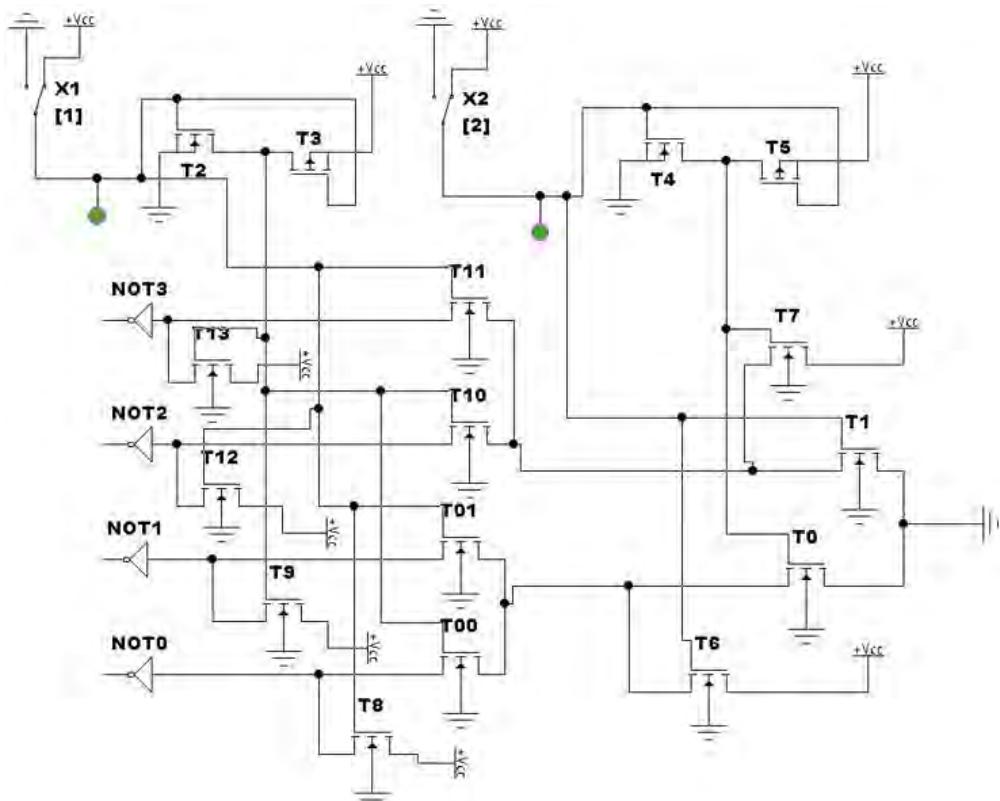


Рис. 7. Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 – обратное дерево 4-1 с альтернативными транзисторами для каждого передающего транзистора

На рис. 8 представлен один блок программируемых дизъюнкций, настроенный подключениями шин «Нуль вольт» и питания:

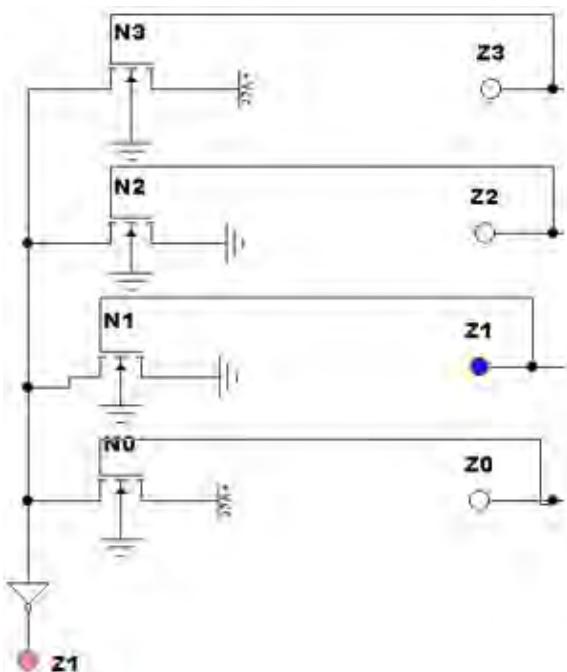


Рис. 8. Блок программируемых дизъюнкций для DC LUT на два информационных (адресных) входа

Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 с альтернативными транзисторами для каждого передающего транзистора и блоком программируемых дизъюнкций, настроенным на реализацию логической функции $X_1 \oplus X_2$ для набора 01 представлено на рис. 9.

Видим, что $X_1 \oplus X_2 = 1$. Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 с альтернативными транзисторами для каждого передающего транзистора и блоком программируемых дизъюнкций, настроенным на реализацию логической функции $X_1 \oplus X_2$ для набора 11 представлено на рис. 10.

Видим, что $X_1 \oplus X_2 = 0$.

ОЦЕНКА СЛОЖНОСТИ DC LUT ПЛИС FPGA

По каждой входной переменной оставим один инвертор, всего $2n$, на каждый элементарный дешифратор необходимо не 2, а 4 транзистора с учётом необходимости обеспече-

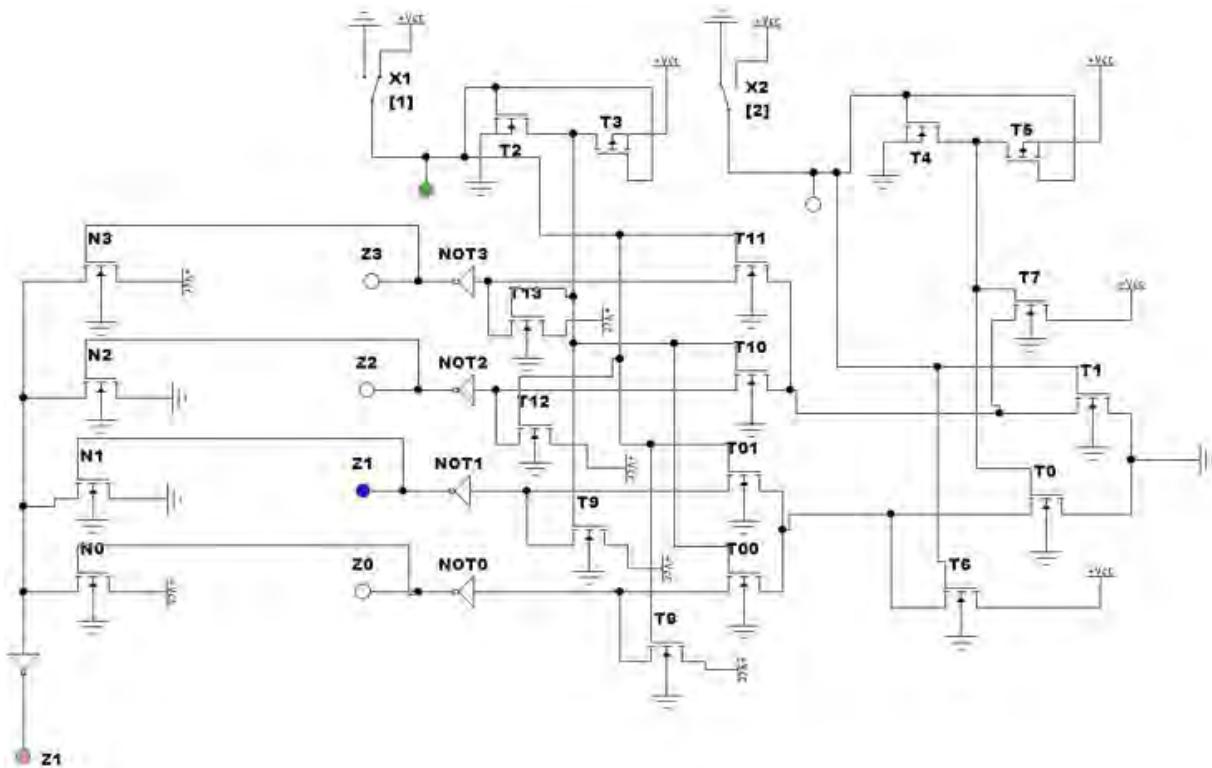


Рис. 9. Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 – обратного дерева 4-1 с альтернативными транзисторами для каждого передающего транзистора и блоком программируемых дизъюнкций, настроенным на реализацию логической функции $X_1 \oplus X_2$ для набора 01

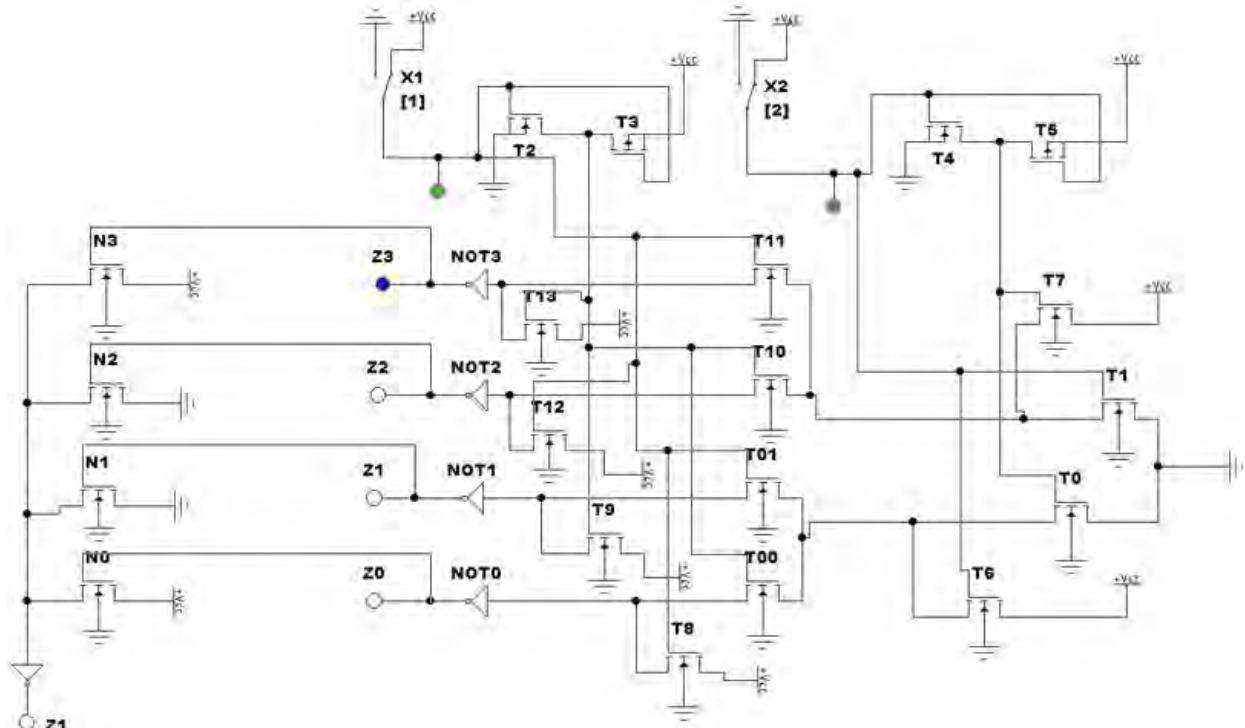


Рис. 10. Моделирование DC LUT на два информационных (адресных) входа – X_1X_2 – обратного дерева 4-1 с альтернативными транзисторами для каждого передающего транзистора и блоком программируемых дизъюнкций, настроенным на реализацию логической функции $X_1 \oplus X_2$ для набора 11

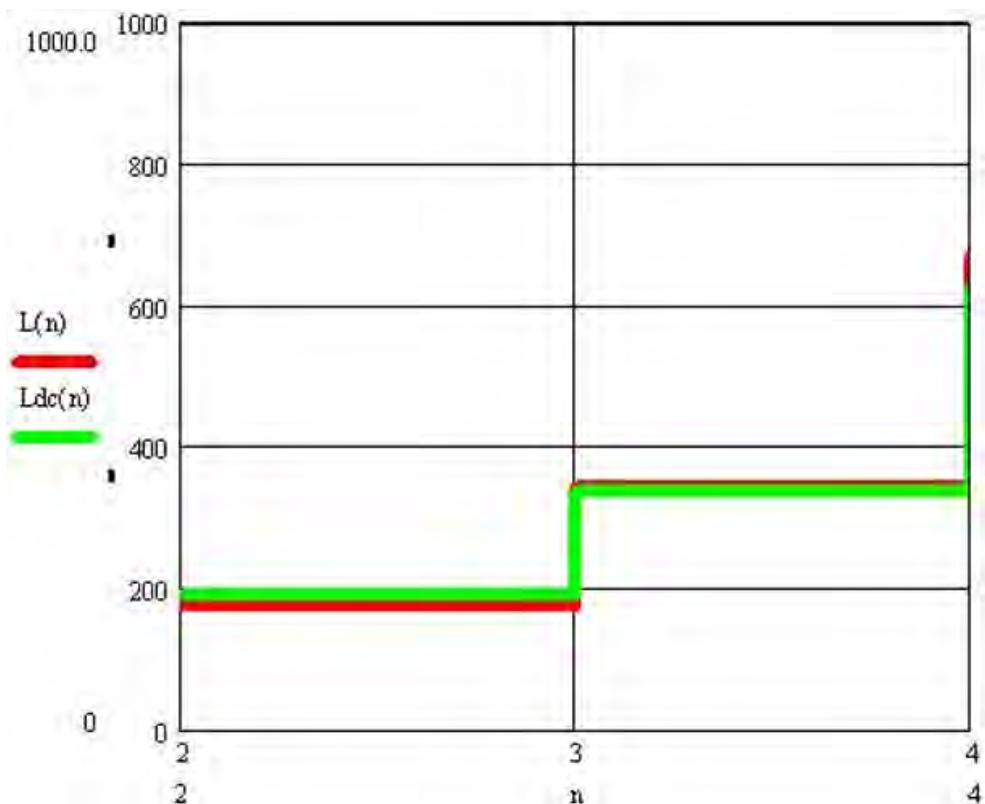


Рис. 11. Сравнение сложности логических элементов 2, 3, 4 LUT – $L(n)$ и 2, 3 ,4 DC LUT – $L_{dc}(n)$ при $m=4$

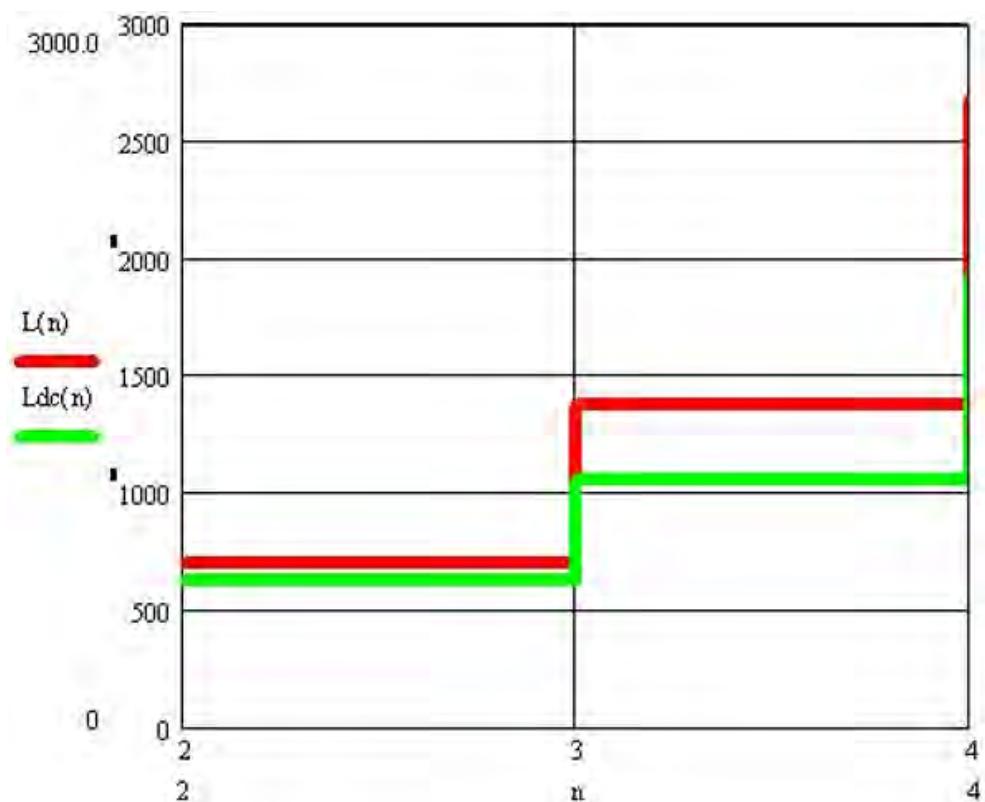


Рис. 12. Сравнение сложности логических элементов 2, 3, 4 LUT – $L(n)$ и 2, 3, 4 DC LUT – $L_{dc}(n)$ при $m=16$

чения ортогональности. Всего таких элементарных дешифраторов для n переменных необходимо $2^n - 1$, получаем сложность в транзисторах всех дешифраторов равную $(2^n - 1) \cdot 4$. На каждом выходе – инвертор (два транзистора), поэтому сложность вычисляется по формуле:

$$L = 2^{n+1} + (2^n - 1) \cdot 4 + 2n.$$

Для задания m логических функций в СДНФ необходима соответствующая настройка и блоки дизъюнкций конституент, итого для всего DC n -LUT:

$$L_{DC-LUT} = 2^{n+1} + (2^n - 1) \cdot 4 + 2n + 6 \cdot m \cdot (2^n + 2),$$

где +2 – это инвертор в блоке дизъюнкций конституент на рис. 9 – сложность одного бита SRAM.

Обычный n -LUT, $n < 5$ для задания m логических функций:

$$L_n = m(2^{n+1} + 8 \cdot 2^n + 2n).$$

Сравнение сложности реализации m логических функций в n -LUT и в n -DC LUT показано на рис. 11, 12.

Выигрывает m LUT при $n \geq 3$, проигрывая при $n = 2$. Выигрывает при увеличении m рис. 12.

Таким образом, получаем ориентировочно на 30 % меньше затрат при 4 переменных.

ВЫВОДЫ

Таким образом, модифицирована структура предложенного логического элемента дешифратора – DC LUT для реализации систем функций в ПЛИС типа FPGA, который основан на схеме в виде дерева передающих МОП транзисторов. Уточняется реализация альтернативных цепочек: для каждого отдельного передающего транзистора устанавливается альтернативный транзистор, обеспечивающий формирование на его стоке уровня логической единицы. В обычном LUT с одним выходом это не требуется, так как всегда активируется только одна ветвь прямого дерева передающих транзисторов. В DC LUT сигналы с обратного дерева транзисторов DC поступают на затворы МОП транзистора блока дизъюнкций, а настроенная информация

из конфигурационной памяти поступает на стоки этих транзисторов. Выполненное моделирование в системе схемотехнического моделирования NI Multisim 10 фирмы National Instruments Electronics Workbench Group подтверждает работоспособность предлагаемого нового технического решения DC-LUT. Для стандартного количества переменных LUT, равного четырём при реализации большого числа функций (16), DC-LUT позволяет получить выигрыш в сложности порядка 30 %. Выигрыш увеличивается при реализации LUT более 4-х переменных.

СПИСОК ЛИТЕРАТУРЫ

1. Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри. – Режим доступа: http://www.kit-e.ru/articles/plis/2010_11_56.php (дата обращения 16.12.2014)
2. Самкова Е. Stratix IV против Virtex-5. Точка не поставлена. – Режим доступа: <http://www.russianelectronics.ru/leader-r/review/2189/doc/46576/> (дата обращения: 16.01.2015)
3. Altera Arria II GX – Режим доступа: <http://icgamma.ru/linecard/altera/arria2gx/> (дата обращения: 16.01.2015)
4. An Ultra-Low-Energy, Variation-Tolerant FPGA Architecture Using Component-Specific Mapping. – Режим доступа: <http://thesis.library.caltech.edu/7226/> (дата обращения 11.11.2014)
5. Тюрин С. Ф. Логические элементы для реализации систем функций в ПЛИС FPGA // Проектирование и технология электронных средств. – 2013. – № 4. – С. 33–37.
6. Тюрин С. Ф., Вихорев Р. В. Программируемое логическое устройство: патент РФ №2573732; опубл. 27.01.2016, Бюл. № 3.
7. Ульман Дж. Д. Вычислительные аспекты СБИС. Пер. с англ.: А. В. Неймана. Под ред. П.П. Пархоменко. – М. : Радио и связь, 1990. – 480 с.

Тюрин Сергей Феофентович – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор кафедры автоматики и телемеханики, электротехнический факультет, Пермский национальный исследовательский политехнический университет.
E-mail: tyurinsergfeo@yandex.ru

Прохоров Андрей Сергеевич – аспирант кафедры автоматики и телемеханики, электротехнический факультет, Пермский национальный исследовательский политехнический университет.
E-mail: proxop007@yandex.ru

Tyurin Sergey Feofentovich – Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor at the Department of Automation and Telemechanics, Electrical Engineering Faculty, Perm National Research Polytechnic University.
E-mail: tyurinsergfeo@yandex.ru

Prokhorov Andrey Sergeevich – postgraduate student at the Department of Automation and Telemechanics, Electrical Engineering Faculty, Perm National Research Polytechnic University.
E-mail: proxop007@yandex.ru