

УДК 681.32

Особенности моделирования и оптимизации комплекта новых логических элементов ПЛИС

Р. В. Вихорев

ПАО "ПНППК", Россия, 614990, г. Пермь, ул. 25 Октября, д. 106
 Пермский национальный исследовательский политехнический университет
 Россия, 614990, г. Пермь, Комсомольский пр., д. 29
 vihrusvla@gmail.com; +7-919-706-52-41

Функциональные возможности и области применения программируемых логических интегральных схем (ПЛИС) постоянно расширяются, в том числе в специальной аппаратуре. Имеются два основных вида ПЛИС: FPGA (Field-Programmable Gate Array) и CPLD (Complex Programmable Logic Device), отличающиеся принципами реализации логики и хранения конфигурационной информации. Тем не менее, различия между ними постепенно нивелируются: так, в продуктах фирмы Интел ПЛИС FPGA, имеющие конфигурационную флэш-память отнесены к CPLD, хотя логические элементы типичны для FPGA, в которых используется оперативная память, требующая загрузки при включении питания. На этом фоне отмечаются попытки комплексирования двух подходов к реализации логики и создания гибридных ПЛИС. В статье анализируются особенности моделирования и оптимизации комплекта новых логических элементов ПЛИС для реализации систем функций.

Ключевые слова: моделирование; оптимизация; логический элемент; транзистор.

DOI: 10.17072/1993-0550-2018-3-111-116

Введение

В ПЛИС [1] логические функции реализуются двумя основными путями. В ПЛИС FPGA используются генераторы функций, называемые Look Up Table (LUT) [2], которые реализуют одну произвольную функцию z от n переменных x ($n = 3, 4$; в современных адаптивных логических модулях – АЛМ n может быть $= 8$) в совершенной дизъюнктивной нормальной форме в виде дерева:

$$z(x_n, x_{n-1}, \dots, x_2, x_1) = d_{out} \cdot \left(\begin{array}{c} \overline{x_{n-1}} \leftarrow \overline{x_{n-2}} \leftarrow \dots \leftarrow \overline{x_2} \leftarrow \overline{x_1} \leftarrow d_0 \\ \overline{x_{n-1}} \leftarrow \overline{x_{n-2}} \leftarrow \dots \leftarrow \overline{x_2} \leftarrow \overline{x_1} \leftarrow d_1 \\ \dots \\ \overline{x_{n-1}} \leftarrow \overline{x_{n-2}} \leftarrow \dots \leftarrow \overline{x_2} \leftarrow \overline{x_1} \leftarrow d_{2^n-2} \\ \overline{x_{n-1}} \leftarrow \overline{x_{n-2}} \leftarrow \dots \leftarrow \overline{x_2} \leftarrow \overline{x_1} \leftarrow d_{2^n-1} \end{array} \right), \quad (1)$$

где d – биты настройки 2^{2^n} функций в совершенной дизъюнктивной нормальной форме (СДФ), всего 2^n бит.

Поэтому реализация функций большого числа переменных n проблематична, а для получения системы из m функций необходимо m деревьев (1). В ПЛИС CPLD реализуется иная технология – программируемых логических матриц (ПЛМ), которые ориентированы на ДНФ-реализацию систем функций большого числа переменных [3, 4]. Так, матрица конъюнкций (матрица И) может быть описана выражением:

$$\begin{aligned} y_1(x_n, x_{n-1}, \dots, x_2, x_1) &= \frac{\overline{x_n}}{s_{1,n}} \cdot \frac{\overline{x_n}}{s_{1,n}} \cdot \frac{\overline{x_{n-1}}}{s_{1,n-1}} \cdot \frac{\overline{x_{n-1}}}{s_{1,n-1}} \cdot \dots \cdot \frac{\overline{x_2}}{s_{1,2}} \cdot \frac{\overline{x_2}}{s_{1,2}} \cdot \frac{\overline{x_1}}{s_{1,1}} \cdot \frac{\overline{x_1}}{s_{1,1}} \\ y_2(x_n, x_{n-1}, \dots, x_2, x_1) &= \frac{\overline{x_n}}{s_{2,n}} \cdot \frac{\overline{x_n}}{s_{2,n}} \cdot \frac{\overline{x_{n-1}}}{s_{2,n-1}} \cdot \frac{\overline{x_{n-1}}}{s_{2,n-1}} \cdot \dots \cdot \frac{\overline{x_2}}{s_{2,2}} \cdot \frac{\overline{x_2}}{s_{2,2}} \cdot \frac{\overline{x_1}}{s_{2,1}} \cdot \frac{\overline{x_1}}{s_{2,1}} \\ &\dots \\ y_j(x_n, x_{n-1}, \dots, x_2, x_1) &= \frac{\overline{x_n}}{s_{j,n}} \cdot \frac{\overline{x_n}}{s_{j,n}} \cdot \frac{\overline{x_{n-1}}}{s_{j,n-1}} \cdot \frac{\overline{x_{n-1}}}{s_{j,n-1}} \cdot \dots \cdot \frac{\overline{x_2}}{s_{j,2}} \cdot \frac{\overline{x_2}}{s_{j,2}} \cdot \frac{\overline{x_1}}{s_{j,1}} \cdot \frac{\overline{x_1}}{s_{j,1}} \\ &\dots \\ y_k(x_n, x_{n-1}, \dots, x_2, x_1) &= \frac{\overline{x_n}}{s_{k,n}} \cdot \frac{\overline{x_n}}{s_{k,n}} \cdot \frac{\overline{x_{n-1}}}{s_{k,n-1}} \cdot \frac{\overline{x_{n-1}}}{s_{k,n-1}} \cdot \dots \cdot \frac{\overline{x_2}}{s_{k,2}} \cdot \frac{\overline{x_2}}{s_{k,2}} \cdot \frac{\overline{x_1}}{s_{k,1}} \cdot \frac{\overline{x_1}}{s_{k,1}}, \end{aligned} \quad (2)$$

а матрица дизъюнкций (матрица ИЛИ)

Моделирование [13] показало, что гипотеза о существенном снижении энергопотребления при отказе от подтягивающего резистора в логических элементах (5)-DC-LUT-R, реализующих системы функций в СДНФ и ДНФ, полностью подтвердилась. Кроме того, энергопотребление варианта (7)-DC-LUT-BKN оказалось существенно меньше, чем у варианта (6) – DC-LUT-O и, соответственно, меньше, чем у t блоков LUT – рис. 3.

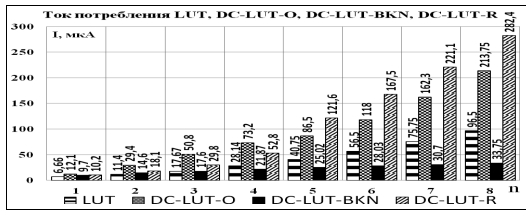


Рис. 3. Ток потребления логических элементов для разного количества входных переменных при напряжении питания 0,8 В на частоте 300МГц; количество реализуемых логических функций $t=3$

Кроме того, предлагается адаптивный логический элемент [14], который может реализовывать либо LUT, либо DC-LUT:

$$\begin{aligned}
 & \bar{x}_{n-2} \dots \bar{x}_1 (\bar{s} \cdot d_{2n-1} \vee s \{=z_0\}) \\
 & x_{n-1} x_{n-2} \dots x_2 x_1 (\bar{s} \cdot d_{2n-1+1} \vee s \{=z_1\}) \\
 & \dots \\
 & x_n \cdot s \vee z_{out} (x_n \dots x_1) \cdot \bar{s} \\
 & \dots \\
 & x_n \bar{x}_{n-2} \dots \bar{x}_1 (\bar{s} \cdot d_{2n-2} \vee s \{=z_{2n-2}\}) \\
 & x_{n-1} x_{n-2} \dots x_2 x_1 (\bar{s} \cdot d_{2n-1-1} \vee s \{=z_{2n-1}\})
 \end{aligned} \quad (9)$$

В выражении (9) с целью упрощения не указаны средства обеспечения ортогональности. Настройка на реализацию дешифратора осуществляется конфигурационной константой s . Это позволяет реализовать дешифратор набора n переменных на одном LUT вместо 2^n LUT. При этом возможная требуемая реализация дизъюнкций конъюнктов может быть возложена на штатные коммутационные средства ПЛИС.

2. Усовершенствованный метод реализации систем функций в ДНФ и моделирование соответствующих элементов

Существующая в ПЛИС типа CPLD реализация конъюнкций (2) может быть описана следующим образом по j -й переменной с использованием конфигурационной константы s :

$$y_j(x_n \dots x_2 x_1) = (x_i \vee s_{j,i})(\bar{x}_i \vee \bar{s}_{j,i}) \quad (10)$$

При этом для обеспечения ортогональности также используются подтягивающие транзисторы R в матрице И:

$$\begin{aligned}
 y_1(x_n x_{n-1} \dots x_2 x_1) &= (R_1) \frac{x_n}{s_{1,n}} \cdot \frac{x_n}{s_{1,n}} \cdot \frac{x_{n-1}}{s_{1,n-1}} \cdot \frac{x_{n-1}}{s_{1,n-1}} \cdot \dots \cdot \frac{x_2}{s_{1,2}} \cdot \frac{x_2}{s_{1,2}} \cdot \frac{x_1}{s_{1,1}} \cdot \frac{x_1}{s_{1,1}} \\
 y_2(x_n x_{n-1} \dots x_2 x_1) &= (R_2) \frac{x_n}{s_{2,n}} \cdot \frac{x_n}{s_{2,n}} \cdot \frac{x_{n-1}}{s_{2,n-1}} \cdot \frac{x_{n-1}}{s_{2,n-1}} \cdot \dots \cdot \frac{x_2}{s_{2,2}} \cdot \frac{x_2}{s_{2,2}} \cdot \frac{x_1}{s_{2,1}} \cdot \frac{x_1}{s_{2,1}} \\
 &\dots \\
 y_j(x_n x_{n-1} \dots x_2 x_1) &= (R_j) \frac{x_n}{s_{j,n}} \cdot \frac{x_n}{s_{j,n}} \cdot \frac{x_{n-1}}{s_{j,n-1}} \cdot \frac{x_{n-1}}{s_{j,n-1}} \cdot \dots \cdot \frac{x_2}{s_{j,2}} \cdot \frac{x_2}{s_{j,2}} \cdot \frac{x_1}{s_{j,1}} \cdot \frac{x_1}{s_{j,1}} \\
 &\dots \\
 y_k(x_n x_{n-1} \dots x_2 x_1) &= (R_k) \frac{x_n}{s_{k,n}} \cdot \frac{x_n}{s_{k,n}} \cdot \frac{x_{n-1}}{s_{k,n-1}} \cdot \frac{x_{n-1}}{s_{k,n-1}} \cdot \dots \cdot \frac{x_2}{s_{k,2}} \cdot \frac{x_2}{s_{k,2}} \cdot \frac{x_1}{s_{k,1}} \cdot \frac{x_1}{s_{k,1}}
 \end{aligned} \quad (11)$$

Предложено новое выражение для обеспечения ортогональности:

$$\begin{aligned}
 s_i &= (d_{SRAM,i} \cdot d_{in} \vee \bar{d}_{SRAM,i} \cdot d_0) \cdot x_i \vee \\
 &\vee (d_{SRAM,i} \cdot d_{in} \vee \bar{d}_{SRAM,i} \cdot d_0) \cdot \bar{x}_i; i = n;
 \end{aligned} \quad (12)$$

где $d_{SRAM,i}$, $\bar{d}_{SRAM,i}$ – настройки соответствующих ячеек конфигурационной памяти.

Соответствующий блок одной переменной приведен на рис. 4.

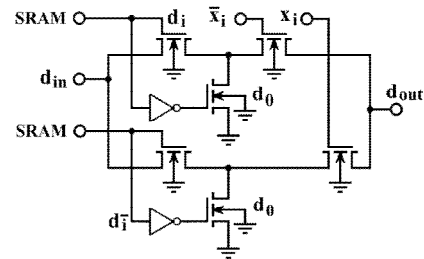


Рис. 4. Блок одной переменной по выражению (12)

Это позволяет обеспечить передачу заданного логического уровня при "правильном" значении переменной и противоположного уровня при "неправильном" значении для последующего формирования значения запрограммированной конъюнкции. В случае несущественности переменной обеспечивается передача заданного логического уровня. При этом в известном методе DNF-R образуется обрыв цепи, как раз и парируемый подтягивающим транзистором. Предложено два варианта формирования значения конъюнкции по значениям (12) – параллельный DNF-S, DNF-P.

При моделировании оказалось, что логический элемент, реализующий ДНФ с последовательным соединением блоков настройки переменных DNF-S, имеет существенно меньший ток потребления, чем с параллельным DNF-P и, конечно, DNF-R (рис. 5).

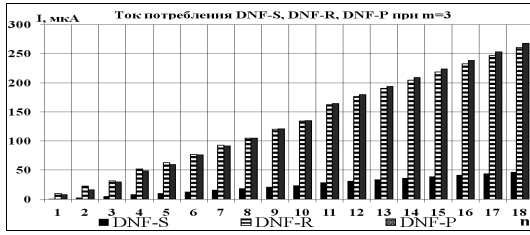


Рис. 5. Ток потребления логических элементов ДНФ для разного количества входных переменных n , при напряжении питания $0,8 В$ на частоте $300 МГц$; количество реализуемых логических функций $m=3$

3. Оптимизация комплекта новых логических элементов ПЛИС

Элементы оцениваются по сложности относительно заданного набора параметров n, m, w, k, r , где n – число переменных m – число функций, w – число конъюнкций для ДНФ, k – параметр декомпозиции дерева LUT (как правило $k=1$, т. е. дерево для большого числа переменных строится из деревьев на одну переменную), r – ограничения Мида-Конвей (максимально возможное число последовательно соединенных передающих транзисторов, $r=4$, но в ПЛИС по оценкам специалистов это параметр равен 3; после трех транзисторов обязательно устанавливается так называемый восстановитель уровня сигнала).

Дано: архитектура существующего адаптивного логического модуля – АЛМ,

множество элементов АЛМ: $\bigcup_{i=1}^e \Psi_i$, их коли-

чество: $= \sum_{i=1}^e |\Psi_i|$, (сейчас, например, как в

StratixIII $e=8$; LUT3 – 4 штуки, LUT3 – 2 штуки); исходные значения сложности существующих логических элементов FPGA и CPLD:

$L - \sum_{i=1}^e \sum_{\xi=1}^{|\Psi_\xi|} L(n, m, w, k, r)_{i\xi}$; энергопотребле-

ния $E - \sum_{i=1}^e \sum_{\mu=1}^{|\Psi_\mu|} E_\mu(n, m, w, k, r)_{i\mu}$ и макси-

мальной задержки $\max \tau_{\Psi_i}$.

Минимизируется сложность набора элементов в новом предлагаемом АЛМ на основе получения оценок сложности и модифицирования венгерского метода (алгоритма) [15–17] с целью получения оптимального набора (вектора, комплекта) элементов $[z_1 z_2 \dots z_\pi]$:

$$L_{z_1 z_2 \dots z_\pi}(n, m, w, k, r) \rightarrow \min; \quad (13)$$

$$[z_1 z_2 \dots z_\pi](n, m, w, k, r); z_\lambda = \sum_{\sigma=1}^v \Psi_\sigma; \Psi_\sigma \in \Psi_i^*; v \leq j; \sum_{\kappa=1}^{\pi} z_\kappa = \sum_{i=1}^e |\Psi_i| \quad (14)$$

С использованием полученных наборов $[z_1 z_2 \dots z_\pi](n, m, w, k, r)$ определить множество Парето, учитывающее полученные в результате моделирования оценки энергопотребления и задержки и их интерполяцию для реальных значений параметров систем логических функций.

В результате получить требуемое количество новых элементов в одном АЛМ*:

$$\sum_{j=1}^q |\Psi_j^*| \quad \text{так, чтобы с меньшими затратами в}$$

количестве транзисторов реализовать заданные системы логических функций

$$\sum_{j=1}^q \sum_{\gamma=1}^{|\Psi_\gamma^*|} L^*(n, m, w, k, r)_{j\gamma} \ll \sum_{i=1}^e \sum_{\xi=1}^{|\Psi_\xi|} L(n, m, w, k, r)_{i\xi}; \quad (15)$$

при этом не ухудшить энергопотребление и задержку:

$$\sum_{j=1}^q \sum_{\gamma=1}^{|\Psi_\gamma^*|} E(n, m, w, k, r)_{j\gamma} \leq \sum_{i=1}^e \sum_{\xi=1}^{|\Psi_\xi|} E(n, m, w, k, r)_{i\xi}; \quad (16)$$

$$\forall \tau_{\Psi_j^*} \forall \tau_{\Psi_i} (\max \tau_{\Psi_j^*} \leq \max \tau_{\Psi_i}).$$

Результаты оптимизации представлены на рис. 6.

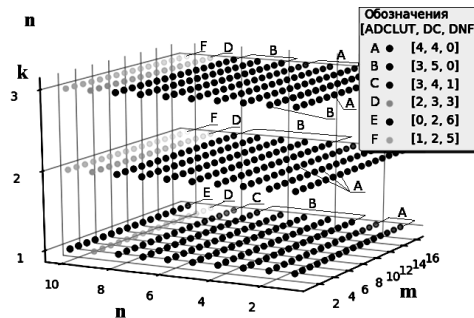


Рис. 6. Результаты оптимизации логических элементов

Выводы

Таким образом, при большом числе переменных целесообразно введение большего числа блоков ДНФ; такие АЛМ подходят для ПЛИС высокой ценовой категории. При среднем значении числа переменных и функций достаточно введение блоков ADC-LUT.

Для бюджетных ПЛИС достаточно введения примерно одинакового количества ADC-LUT и DC-LUT.

Задача решалась в общем виде относительно параметров, указанных в открытых источниках. Разработанный алгоритм может быть использован для получения решений с учетом других доступных параметров существующих ПЛИС.

В дальнейшем целесообразно оценить и площадь кристалла для уточнения результатов оптимизации.

Список литературы

1. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2007. 782 с.
2. Строгонов А., Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри. URL: http://www.kite.ru/articles/plis/2010_11_56.php (дата обращения: 11.06.2017).
3. PLA. URL: <https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/42-programmable/10-pla/pla.html> (дата обращения: 27.06.2018).
4. FPGA/CPLD – ПЛИС (Программируемые Логические Интегральные Схемы). URL: <http://www.fpga-cpld.ru/> (дата обращения 02.07.2018).
5. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices. URL: https://www.altera.com.cn/content/dam/altera-www/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf (дата обращения: 29.06.2018).
6. Обзор архитектуры ПЛИС семейства Virtex-5. URL: <http://elektors.ru/radioelektronika/mikroshemy/2759-obzor-arhitektury-plis-semeystva-virtex-.html> (дата обращения: 27.06.2018).
7. Chi Wai, Yu. Hybrid FPGA: Architecture and Interface. URL: <https://sydney.edu.au/engineering/electrical/people/philip.leong/UserFiles/File/theses/cwyu10.pdf> (дата обращения: 28.06.2018).
8. Alireza Kaviani and Stephen Brown. HYBRID FPGA ARCHITECTURE. URL: <https://sydney.edu.au/engineering/electrical/people/philip.leong/UserFiles/File/theses/cwyu10.pdf> (дата обращения: 21.06.2018).
9. Программируемое логическое устройство: пат. Рос. Федерация / Тюрин С.Ф., Вихорев Р.В.; – № 2573732; Оpubл. 27.01.2016, Бюл. № 3.
10. Тюрин С.Ф., Вихорев Р.В. Усовершенствованный метод реализации в FPGA систем логических функций, заданных в СДНФ // Инженерный вестник Дона. 2017. Т. 44, № 1(44). С. 39.
11. Вихорев Р.В., Тюрин С.Ф. Программируемые логические элементы ПЛИС FPGA для реализации систем логических функций // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2017. № 3(23). С. 133–145.
12. Tyurin S., Grekov A., Vikhorev R., Prokhorov A. Advanced FPGA Look up tables. International Journal of Pure and Applied Mathematics. 2017. Vol. 117, № 22. P. 143–147.
13. Вихорев Р.В., Прохоров А.С., Тюрин С.Ф., Никитин А.С. Моделирование и оптимизация инновационных логических элементов ПЛИС // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2017, № 24. С. 192–208.
14. Тюрин С.Ф., Вихорев Р.В. Адаптивный логический модуль ПЛИС с архитектурой FPGA // Вестник Рязанского государственного радиотехнического университета. 2018, № 63. С. 69–76.
15. Вихорев Р.В., Прохоров А.С., Скорнякова А.Ю., Тюрин С.Ф. Усовершенствованные методы реализации программируемой логики / в сб.: Управление большими системами. УБС–2017, матер. XIV Всерос. школы-конф. молодых ученых. 2017. С. 306–315.
16. Тюрин С.Ф., Никитин А.С., Вихорев Р.В., Скорнякова А.Ю. Выбор набора конфигурируемых логических элементов с использованием венгерского метода // Вестник Пермского университета. Математика. Механика. Информатика. 2017. № 2(37). С. 65–68.
17. Свидетельство о государственной регистрации программы для ЭВМ № 2017663289 Программа оптимизации набора логических элементов модифицированным венгерским методом "ВЕННИТ" / заявитель и правообладатель Тюрин С.Ф., Никитин А.С., Вихорев Р.В., Скорнякова А.Ю., Прохоров А.С.; – № 2017619911. Заявл. 04.10.2017; Оpubл. 28.11.2017.

Simulation and optimization of the FPGA's novel logic elements and their features

R. V. Vikhorev

Public joint-stock company "PSPIMC"; 106, ul. 25 Oktyabrya, Perm, 614990, Russia

Perm National Research Polytechnic University; 29, Komsomol'skiy prospekt, Perm, 614990, Russia

vihrusvla@gmail.com, +7-919-706-52-41

Functional capabilities and applications of programmable logic devices are constantly expanding, including in special equipment. There are two main types of PLDs: FPGA (Field-Programmable Gate Array) and CPLD (Complex Programmable Logic Device), which differ in the principles of implementing logic and storing configuration information. Nevertheless, the differences between them are gradually leveled, so in Intel's FPGA products, having a configuration flash memory are categorized as CPLD, although the logic elements are typical for FPGAs that use RAM that requires power-up loads. Against this backdrop, attempts are made to integrate two approaches to implementing logic and creating hybrid FPGAs. In the article features of simulation and optimization of a set of new logic elements of FPGA for the implementation of systems of functions are analyzed.

Keywords: *simulation; optimization; logic element; transistor.*