

ИНФОРМАТИКА ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 681.32

Выбор набора конфигурируемых логических элементов с использованием венгерского метода

С. Ф. Тюрин, А. С. Никитин, Р. В. Вихорев, А. Ю. Скорнякова

Пермский национальный исследовательский политехнический университет
Россия, 614990, г. Пермь, ул. Комсомольский пр., 29
tyurinsergfeoyandex.ru; +7-952-32-02-510

Получаются оценки сложности конфигурируемых логических элементов, реализующих системы функций. Анализируются и сравниваются варианты реализации элементов. Для выбора оптимального набора элементов для различных параметров систем логических функций предлагается модификация венгерского метода реализации.

Ключевые слова: логические элементы; системы логических функций; оценки сложности; оптимизация; венгерский метод.

DOI: 10.17072/1993-0550-2017-2-65-68

Введение

Как известно, в программируемых логических интегральных схемах (ПЛИС, FPGA) широко используются конфигурируемые логические элементы [1]. Основой таких элементов является устройство, называемое в англоязычной литературе LUT (Look Up Table), так как оно реализует задаваемую (загружаемую) таблицу истинности логической функции. Классическое значение количества переменных $n=4$.

В настоящее время в так называемых адаптивных логических модулях (АЛМ) реализованы LUT с изменяемой разрядностью до 6 переменных, в том числе имеется возможность реализации некоторых логических функций 7 и даже 8 переменных [2]. Функции реализуются в совершенной дизъюнктивной нормальной форме (СДНФ). Существующие принципы реализации систем m логических функций от одних n переменных предполагают использование m LUT.

В статье рассматриваются оценки сложности предложенных конфигурируемых логических элементов, реализующих системы функций в СДНФ и в ДНФ, в том числе ориентированных на самосинхронные схемы (ССС) [3]. Предлагается подход к выбору оптимального набора логических элементов с использованием венгерского метода.

1. Оценка сложности LUT

Сложность LUT [1, 2] в количестве транзисторов без декомпозиции (до $n=4$) имеет вид выражения

$$L_n = 2^n \cdot 8 + 2^{n+1} + 6n, \quad (1)$$

Однако с учетом ограничений Мида–Конвей [4] при декомпозиции сложного дерева по k -LUT, $k \in \{1,2,3,4\}$, $n > k$, n :

$$L_{n,k} = 2^{\lfloor n \rfloor} \cdot 8 + (2^{\lfloor k \rfloor + 1} + 6 \lfloor k \rfloor) \cdot \sum_{i=1}^{\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \rfloor} 2^{\lfloor n \rfloor - i \cdot \lfloor k \rfloor} + \\ + (2^{\lfloor n \rfloor - \lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \rfloor \cdot \lfloor k \rfloor + 1} + 6 \cdot \left(\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor \right)) + 6 \lfloor n \rfloor, \quad (2)$$

© Тюрин С. Ф., Никитин А. С., Вихорев Р. В., Скорнякова А. Ю., 2017

где $2^{\lfloor k \rfloor + 1} + 6\lfloor k \rfloor$ – сложность одного k -дерева $\lfloor \dots \rfloor$ – округление в нижнюю сторону (floor); таких деревьев (k -LUT) необходимо в первом слое $2^{\lfloor n \rfloor - \lfloor k \rfloor}$, затем нужно провести декомпозицию k -LUT этого первого слоя, получаем $2^{\lfloor n \rfloor - \lfloor k \rfloor - \lfloor k \rfloor}$. Всего необходимо i k -LUT, где i определяется из соотношения: $\lfloor i \rfloor = \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor$ всего

$\sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor}$ и последний LUT на $\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor$ переменных.

Временная задержка в количестве транзисторов определяется выражением

$$T_{n,k} = n + 2 \left\lfloor \frac{n}{k} \right\rfloor + 2 \cdot \left[\left\lfloor \frac{n}{k} \right\rfloor - \left\lfloor \frac{n}{k} \right\rfloor \right]. \quad (3)$$

2. Оценка сложности LUT-ST

Для использования в ССС предложен LUT-ST [5]. В этом случае сложность LUT возрастает:

$$L_{n,k,ST} = 2 \cdot \{2^{\lfloor n \rfloor} \cdot 8 + (2^{\lfloor k \rfloor + 1} + 6\lfloor k \rfloor) \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + (2^{\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor + 1} + 6 \cdot \left(\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor \right) + 6\lfloor n \rfloor + 2 \cdot \lfloor n \rfloor + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + 2 \cdot \left[\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \right], \quad (4)$$

где $2 \cdot \lfloor n \rfloor$ – сложность дополнительных цепочек спейсера, $2 \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor}$ – сложность индикаторов k -LUT + индикатор последнего

LUT $2 \cdot \left[\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \right]$.

Выражение (4) не учитывает затраты на фиксацию переходного процесса Г-триггерами [3]. Учесть двухвходные Г-триггеры сложностью 12 транзисторов можно, положив $k=2$, а n

$$n = \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right] - \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right]. \quad (5)$$

Таким образом, получаем сложность Г-триггеров:

$$12 \cdot \left\{ \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right] - \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right] \right\} + \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right] - \left[\frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right]. \quad (6)$$

Временная задержка LUT-ST в количестве транзисторов (без учета задержки Г-триггеров) увеличивается на задержку индикаторов:

$$T_{n,k,ST} = n + 2 \left\lfloor \frac{n}{k} \right\rfloor + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + 2 \cdot \left[\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \right]. \quad (7)$$

3. Оценка сложности DC-LUT-ST, ДНФ-LUT-ST

Для реализации систем функций в СДНФ предложен DC-LUT [6]. Для ССС с учетом сложности m блоков реализации функций с соответствующей настройкой, получаем сложность DC-LUT-ST:

$$L_{dc,n,k,ST}^m = 2 \cdot \{2^{\lfloor n \rfloor} \cdot 8 + (2^{\lfloor k \rfloor + 1} + 6\lfloor k \rfloor) \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + (2^{\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor + 1} + 6 \cdot \left(\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor \right) + 6\lfloor n \rfloor + 2 \cdot \lfloor n \rfloor + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + 2 \cdot \left[\left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \right] + \sum_{j=1}^{\lfloor n \rfloor} 2^j + 6m(2^n + 2)\}, \quad (8)$$

где $6m(2^n + 2)$ – сложности m блоков реализации функций.

Для реализации систем функций в ДНФ предложен ДНФ-LUT [7]. При использовании ДНФ-LUT получаем сложность в количестве транзисторов:

$$L_{dnf} = \lfloor k \rfloor \cdot (20\lfloor n \rfloor + 2 \cdot \left\lfloor \frac{\lfloor n \rfloor}{2} \right\rfloor) + 6 \cdot \lfloor m \rfloor (\lfloor k \rfloor + 2) + 6\lfloor n \rfloor, \quad (9)$$

где $\lfloor k \rfloor \cdot (20\lfloor n \rfloor + 2 \cdot \left\lfloor \frac{\lfloor n \rfloor}{2} \right\rfloor)$ учитывает сложность реализации k настраиваемых конъюнкций; $2 \cdot \left\lfloor \frac{\lfloor n \rfloor}{2} \right\rfloor$ – учитывает сложность инверторов в

блоке конъюнкций, в том числе для удовлетворения ограничения Мида–Конвей [4] в блоках конъюнкций; $6 \cdot \lfloor m \rfloor (\lfloor k \rfloor + 2)$ – сложность m блоков функций от k конъюнкций (реализация монтажного И); $6 \lfloor n \rfloor$ – сложность инверторов по n переменным (два на неинверсный вход, один на инверсный).

Тогда сложность ДНФ-LUT-ST без Г-триггеров определяется выражением:

$$L_{dnf-ST} = 2 \cdot \{ \lfloor k \rfloor \cdot (20 \lfloor n \rfloor + 2 \cdot \lfloor \frac{n}{2} \rfloor) + 6 \cdot \lfloor m \rfloor (\lfloor k \rfloor + 2) + 4 \lfloor n \rfloor \}, \quad (10)$$

где $4 \lfloor n \rfloor$ учитывает инверторы по n парафазным переменным (один на каждый вход, $2 \lfloor n \rfloor$ и цепочку спейсера – один транзистор на каждый парафазный вход, всего $2 \lfloor n \rfloor$).

Сравнение выражений сложности предлагаемых технических решений изображено на рис. 1.

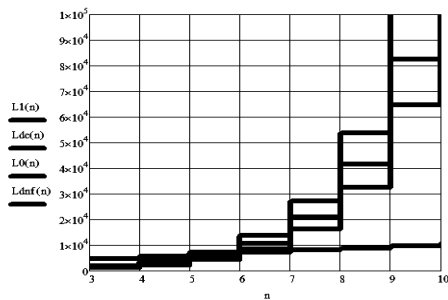


Рис. 1. Сравнение $L0(n)$, $L1(n)$, $Ldc(n)$, $Ldnf(n)$ при $m=8$; $k=3$; $r=20$

Таким образом, ДНФ-LUT-ST выигрывает по сложности реализации при большом количестве переменных системы логических функций. При среднем количестве переменных целесообразно использование DC-LUT-ST.

2. Оптимизация набора логических элементов с использованием венгерского метода

Предлагается использовать известный венгерский метод оптимизации (Hungarian algorithm, Hungarian method) [1, 2] итеративно – для получения нескольких назначений по параметрам логических элементов и систем логических функций так, чтобы из частичных решений в дальнейшем построить глобальную таблицу назначений и получить глобальные назначения, которые и описывают требуемый результат выбора. Модификация метода предполагает вначале получение оценок по

выражениям сложности (10, 9, 8, 4, 3, 2) и/или времени (3, 7) для заданных параметров системы функций. Для учета возможности покрытия одним типом устройства нескольких систем возможно использовать повторение строки матрицы.

Предлагаемый алгоритм выбора конфигурируемых логических элементов, реализующих системы функций с использованием венгерского метода, изображен на рис. 2.

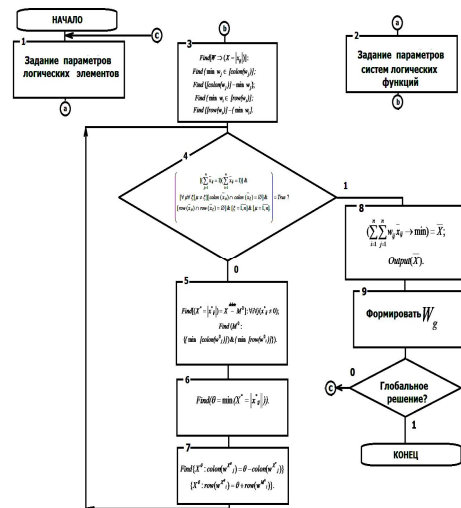


Рис. 2. Алгоритм выбора конфигурируемых логических элементов, реализующих системы функций с использованием венгерского метода

В качестве вариантов реализации систем логических функций можно рассмотреть:

- 1) LUT по числу требуемых функций в системе;
- 2) DC LUT на заданное максимальное число функций;
- 3) ДНФ-LUT на заданное максимальное число конъюнкций и функций;
- 4) варианты комбинирования 1–3. Получим матрицу назначений W следующим образом: каждой строке соответствует вектор, отображающий количество каждой из формул, принимающих участие в расчете.

Так, вектору для первой строки (10,0,0,0) соответствует сумма из десяти формул 1, вектору для второй строки (0,10,0,0) – сумма десяти вторых формул, вектору для третьей строки (0,0,0,10) – сумма десяти третьих формул, вектору для четвертой строки (3,3,4,0) – сумма трех первых, трех вторых и четырех третьих формул. При этом каждому столбцу матрицы назначений W соответствует аргументы m , k и n .

Пример результатов расчета представлен на рис. 3.

	m = 6 k = 4 n = 2	m = 2 k = 3 n = 3	m = 4 k = 2 n = 5	m = 10 k = 1 n = 6
(10,0,0,0)				
(0,10,0,0)				
(0,0,10,0)				
(3,3,4,0)				

Рис. 3. Пример результатов расчета

Выводы

Расчеты показывают, что предлагаемый DC LUT FPGA предпочтительней по аппаратным затратам, чем известный LUT уже при количестве функций $m=8$ для числа переменных $n=4$. Предлагаемый логический элемент ПЛИС – ДНФ FPGA на основе ДНФ по сравнению с ЛЕ-СДНФ выигрывает при переходе к восьмиразрядным функциям (для $n=k=m$).

При этом существующий ЛЕ не может реализовать даже 32 разрядные функции, а предлагаемый имеет приемлемые затраты даже для 64 разрядных функций. Причем быстроедействие предлагаемого варианта так же, как и известного, определяемого в основном длиной цепочки передающих транзисторов – n , определяется цепочкой транзисторов в блоках программируемых конъюнкций – это тоже n , а цепочки в блоках программируемых функций содержат всего один транзистор.

Полагаем, в дальнейшем целесообразно использовать средние характеристики затрат w на реализацию различных систем логических функций, которые были получены путем анализа типовых проектов, загружаемых в ПЛИС.

Список литературы

1. Строгонов А., Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри. URL: http://www.kite.ru/articles/plis/2010_11_56.php (дата обращения: 13.03.2017).
2. Золотуха Р., Комолов Д. Stratix III – новое семейство FPGA фирмы Altera. URL: http://kit-e.ru/assets/files/pdf/2006_12_30.pdf (дата обращения: 14.03.2017).
3. Степченко Ю.А., Денисов А.Н., Дьяченко Ю.Г. и др. Библиотека элементов для проектирования самосинхронных полузаказных микросхем серий 5503/5507 и 5508/5509. М.: ИПИ РАН, 2014. С. 150–151.
4. Ульман Дж. Д. Вычислительные аспекты СБИС / пер. с англ. А.В. Неймана; под ред. П.П. Пархоменко. М.: Радио и связь, 1990. 480 с.
5. Тюрин С.Ф., Каменских А.Н., Плотникова А.Ю. Программируемое логическое устройство. Патент РФ № 2601145. Оpubл. БИ № 30 27.10.2016.
6. Тюрин С.Ф., Вихорев Р.В. Программируемое логическое устройство. Патент РФ № 2573732. Оpubл. БИ № 3 27.01.2016.
7. Тюрин С.Ф. Программируемое логическое устройство. Патент РФ № 2544750. Оpubл. БИ № 8 20.03.2015.
8. Harold W. Kuhn. "The Hungarian Method for the assignment problem", *Naval Research Logistics Quarterly*, 2: 83–97, 1955; doi:10.1002/nav.3800020109.
9. Hungarian algorithm. URL: <http://www.hungarianalgorithm.com/solve.php> (дата обращения: 02.02.2017).

Selection of a configurable logic elements set using the Hungarian method

S. F. Tyurin, A. S. Nikitin, R. V. Vikhorev, A. Yu. Skornyakova

Perm National Research Polytechnic University; 29, Komsomolsky prospekt, Perm, 614990, Russia

Perm State University; 15, Bukireva st., Perm, 614990, Russia

tyurinsergfeoyandex.ru; +7 952-320-02-510

Estimates of the complexity of configurable logic elements that implement systems of functions are determined. The variants of the implementation of elements are analyzed and compared. To select the optimal set of elements for different parameters of logical function systems, a modification of the Hungarian implementation method is proposed.

Keywords: logical elements; systems of logical functions; complexity estimates; optimization; Hungarian method.