УДК 004.27+004.383

И.А. Жуков, Н.А. Ковалев

Один из подходов к уменьшению аппаратурных затрат в цифровых интегрирующих структурах

Предложен метод, позволяющий снизить аппаратурную сложность цифровых интегрирующих структур путем определенной организации перехода от сложных функций многих переменных к порождающим уравнениям Шеннона, а также способ их построения. Метод создает предпосылки к повышению точности и улучшению технико-экономических характеристик интегральных реализаций цифровых интегрирующих структур на базе программируемых логических интегральных схем.

The method of reduction of hardware costs in digital integrating structures due to the certain organization of transition from the many variables complex functions to the generating Shannon equations is proposed. The method creates preconditions for enhancement of accuracy and technical and economic characteristics of integrated implementations of digital integrating structures on FPGA bases.

Запропоновано метод, що дозволяє знизити апаратурну складність цифрових інтегруючих структур шляхом певної організації переходу від складних функцій багатьох змінних до породжувальних рівнянь Шеннона, а також спосіб їх побудови. Метод створює передумови до підвищення точності й поліпшення техніко-економічних характеристик інтегральних реалізацій цифрових інтегруючих структур на базі програмувальних логічних інтегральних схем.

Введение. Необходимость повышения эффективности вычислительных средств в системах управления и моделирования в реальном масштабе времени связана с постоянным расширением сфер их применения и ростом сложности решаемых ими задач. К ним можно отнести решение алгебраических, дифференциальных, трансцендентных уравнений и их систем, вычисление сложных интегралов, функциональных зависимостей, цифровую обработку сигналов и др. [1, 2].

Один из подходов к решению указанных задач предусматривает применение цифровых интегрирующих структур (ЦИС) [3], реализующих эквивалентные системы уравнений Шеннона (СУШ). Последние содержат только операции сложения и умножения, что упрощает структуру ЦИС, в состав которых нет необходимости вводить сложные арифметические блоки. Для реализации ЦИС целесообразно использовать программируемые логические интегральные схемы (ПЛИС), например, типа *FPGA* (*Field Programmable Gate Array*) [4–6].

Основу ЦИС составляют цифровые интеграторы (ЦИ), количество которых определяет как их функциональные возможности, так и аппаратную сложность и, как следствие, ресурсоемкость интегральных реализаций ЦИС на базе *FPGA*. Поэтому для ее уменьшения нужно решать задачи с помощью минимально возможного количества ЦИ. При построении ЦИС необходимо перейти от исходных функциональных зависимостей к эквивалентным им СУШ. Организация такого перехода непосредственно определяет количество ЦИ в составе ЦИС. В [3] показана возможность представления элементарных математических функций и их систем с помощью порождающих СУШ. Однако обоснование такой возможности не равносильно указанию рационального с учетом аппаратной сложности ЦИС формирования СУШ.

Широко используемый метод, предложенный А.В. Каляевым [3], предполагает получение конечной симметричной СУШ. Он состоит в расчленении исходной функции с учетом принципа суперпозиции на более простые, последовательном их дифференцировании (при введении новых переменных) до момента, когда высшие производные вырождаются либо содержат ранее введенные переменные. В случае сложных математических функций многих переменных описанный процесс может предполагать интенсивное наращивание количества новых переменных [7] и, как следствие, дифференциальных уравнений, число которых будет соответствовать числу ЦИ в составе ЦИС. Оценим такой прирост на примере целого рационального выражения

$$Z=\sum_{i=1}^{m}\prod_{j=1}^{s(i)} z_j\,, \qquad (1)$$

к которому можно свести многие элементарные функции и их производные, получаемые в процессе последовательного дифференцирования. Для реализации интегрирования в соответствии им в состав ЦИС потребуется дополнительно ввести $\left(2\sum_{i=1}^{m} s(i)\right)$ ЦИ. Ввиду возможности получения большого числа подобных *промежуточных* функций при формировании конечной симметричной СУШ очевидно значительное наращивание числа ЦИ. При высоких порядках точности $(N = 4,...,7)$ ЦИ предполагает реализацию от 3 до 13 умножений соответственно и прямо пропорционального числа сложений, поэтому интегральная реализация ЦИС потребует значительных расходов ресурсов, особенно блоков умножения или *DSP*-блоков (*Digital Signal Processing blocks*) [8], кристалла *FPGA*. Таким образом, задача дальнейшего снижения аппаратурной сложности ЦИС остается актуальной.

Постановка задачи

Необходимо разработать метод перехода от сложных функций многих переменных к порождающим СУШ, позволяющий уменьшить число дифференциальных уравнений, а следовательно, и количество ЦИ в составе ЦИС, а также ресурсоемкость их интегральной реализации с учетом архитектурных особенностей *FPGA*.

Формирование порождающей СУШ

В [3] высказано предположение, что введение в ЦИС наряду с интегрированием, сложением, экстраполяцией и квантованием более сложных операций можно сократить объем их оборудования, повысить быстродействие и точность. Современные семейства *FPGA* позволяют эффективно реализовывать мультипликативные, аддитивные операции и их комбинации [9], представляющие собой целые рациональные выражения. Поэтому если в процессе формирования СУШ исходная, введенная или подынтегральная функция примет вид такого выражения, то исключив дальнейшее ее упрощение и дифференцирование, можно сократить число ЦИ при высокой эффективности его реализации внутри микросхемы *FPGA*. На этой идее основан предложенный ниже метод получения порождающей СУШ.

Пусть необходимо представить в форме уравнений Шеннона некоторую элементарную функцию многих переменных

$$\omega = F\left\{x_1, x_2, ..., x_{i-1}, f\left[y_1, y_2, ..., y_{k-1}, \varphi\left(z_1, z_2, ..., z_l\right),\right.\right. \\ \left.\left. y_{k+1}, ..., y_m\right], x_{i+1}, ..., x_{j-1}, \psi\left(\xi_1, \xi_{2,...,}\xi_p\right), x_{j+1}, ..., x_n\right\}, \quad (2)$$

составленную в виде композиции конечного числа арифметических действий и операций взятия функций от функций.

В соответствии с принципом суперпозиции выделим и обозначим новыми переменными в (2) более простые элементарные функции так, чтобы исходная функция ω стала подобной целому рациональному выражению. Тогда (2) можно представить системой

$$\begin{cases} \omega = F\left\{x_1, x_2, ..., x_{i-1}, f\left[f_2, y_{k+1}, ..., y_m\right], f_3\right\} \\ f_2 = F\left[y_1, y_2, ..., y_{k-1}, \varphi\left(z_1, z_2, ..., z_l\right)\right] \\ f_3 = F\left\{x_{i+1}, ..., x_{j-1}, \psi\left(\xi_1, \xi_{2,...,}\xi_p\right), x_{j+1}, ..., x_n\right\} \end{cases}. \quad (3)$$

К введенным функциям f_2 и f_3 также применим подобный подход, пока с учетом ввода новых переменных исходная функция не представится системой уравнений, правые части которых будут представлять собой целые рациональные выражения либо элементарные функции (в основном трансцендентные), не поддающиеся дальнейшей декомпозиции и упрощению на основе принципа суперпозиции. Тогда (3) запишем таким образом:

$$\begin{cases} \omega = F\left\{x_1, x_2, ..., x_i, f_3\right\} \\ f_2 = F\left[y_1, y_2, ..., y_k\right] \\ f_3 = F\left\{x_{i+1}, x_{i+2}, ..., x_n\right\} \\ x_i = f\left[f_2, y_{k+1}, ..., y_m\right] \\ x_j = \psi\left(\xi_1, \xi_{2,...,}\xi_p\right) \\ y_k = \varphi\left(z_1, z_2, ..., z_l\right) \end{cases}. \quad (4)$$

Используя известный метод [3], рассмотрим процесс перехода от одной из вновь введенных функций x_j к эквивалентной ей СУШ (для функций x_i и y_k процесс аналогичный). Продифференцировав x_j по переменным интегрирования $\xi_1, \xi_{2,...,}\xi_p$ и введя обозначения $u_1, u_2, ..., u_p$ для образующихся подынтегральных функций, запишем: $dx_j = \sum_{i=1}^{p} u_i d\xi_i$.

Продолжая дифференцирование подынтегральных функций $u_1, u_2, ..., u_p$ и вводя соответствующие обозначения для возникающих при этом новых функций, начнем формировать систему дифференциальных уравнений вида

$$\begin{cases} dx_j = \sum_{i=1}^{p} u_i d\xi_i, \\ du_i = \sum_{k=1}^{p} u_{ik} d\xi_k, \\ du_{ik} = \sum_{m=1}^{p} u_{ikm} d\xi_m, \\ .... \end{cases} \qquad (5)$$

где вводимые подынтегральные функции u_{ik}, u_{ikm} и так далее есть производными высших порядков. Предложены следующие условия завершения процесса последовательного дифференцирования с некоторого его шага в зависимости от того, что наступит раньше:

– получение подынтегральной функции целого рационального вида, которое обозначим новой переменной. Все входящие в нее переменные в правой части уже введены ранее. В этом случае для функции x_j будет получена несимметричная порождающая СУШ;

– получение высших производных, содержащих ноль либо ранее введенные переменные. Это следует из того, что функция $x_j = \psi\left(\xi_1, \xi_{2,...,}\xi_p\right)$ негипертрансцендентна и, следовательно, удовлетворяет конечной СУШ. Тогда, используя известный метод [3], необходимо получить для нее конечную симметричную СУШ.

Получив подобным образом СУШ для функций x_i и y_k и расширив систему (4) всеми полученными уравнениями и их системами вида (5), получим эквивалентную порождающую СУШ, представляющую исходную функцию (2). В нее будут входить как конечные симметричные подсистемы уравнений Шеннона, так и целые рациональные функции.

В разработанном методе можно выделить следующие этапы:

• получение для исходной функции уравнений и их систем, содержащих целые рациональные выражения либо элементарные функции, не поддающиеся дальнейшей декомпозиции и упрощению;

• получение конечных симметричных либо несимметричных порождающих СУШ для упомянутых выше элементарных функций;

• сведение всех полученных целых рациональных выражений и СУШ в единую порождающую СУШ, представляющую исходную функцию;

• определение начальных значений всех зависимых переменных.

Таким образом, суть предлагаемого метода заключается в том, что декомпозиция исходной функции и последовательное дифференцирование выделенных функций (при одновременном введении новых переменных) проводится в общем случае до момента, когда подынтегральные и введенные функции примут вид целых рациональных выражений. Сокращая процесс последовательного дифференцирования, приведенная схема, в отличие от известного метода, создает предпосылки для существенного уменьшения количества дифференциальных уравнений в СУШ и, как следствие, числа ЦИ в составе ЦИС. Например, при каждом получении в процессе формирования СУШ подынтегральной функции вида (1) в сравнении с известной схемой разработанный метод позволяет сокращать состав ЦИС на $\left(2\sum_{i=1}^{m} s(i)\right)$ ЦИ.

Построение и функционирование ЦИС

Оптимальное соотношение между точностью, быстродействием и сложностью оборудования имеют экстраполяционные параллельные ЦИС, использующие формулы интегрирования высоких порядков точности $\left(N \geq 4\right)$ и рассчитанные на работу с фиксированной точкой [3, 7]. Поэтому для построения и функционирования ЦИС будем использовать разностно-квантованную схему представления СУШ [7], основанную на экстраполированных приращениях интегральных переменных первого порядка, которыми и обмениваются ЦИ между собой в процессе интегрирования. Предложенный метод предусматривает расширение этой схемы:

– численное интегрирование дифференциальных уравнений в результирующей СУШ реализуют ЦИ, в основу построения которых положим упомянутую выше схему. Однако приращения интегральных переменных в правых ее частях ее уравнений могут формироваться введенными в ЦИС операционными блоками (ОБ), реализующими вычисления целых рациональных функций;

– в качестве основы для построения ОБ дополнительно предложен следующий экстраполяционный разностно-квантованный алгоритм:

$$\begin{cases} x_{j,i+1} = x_{j,i} + \nabla x_{j,i+1} \left(j = \overline{1,l}\right), \\ \bar{z}_{j,i+1} = \bar{z}_{j,i} + \nabla \bar{z}^{э}_{j,i+1} \left(j = \overline{1,k}\right), \\ Z_{i+1} = F\left(x_{1,i+1},...,x_{k,i+1}, \bar{z}_{1,i+1},...,\bar{z}_{k,i+1}\right), \\ \nabla Z_{i+1} = Z_{i+1} - Z_i , \\ \nabla \bar{Z}_{i+1} = P^{0}_{-n}\left[\nabla Z_{i+1} + s_i\right], \\ s_{i+1} = P^{-n-1}_{-n-m}\left[\nabla Z_{i+1} + s_i\right], \\ \nabla \bar{Z}^{э}_{i+2} = \sum_{\alpha=1}^{N} (-1)^{\alpha-1} \binom{N}{\alpha} \nabla \bar{Z}_{i+2-\alpha} , \end{cases} \qquad (6)$$

где $F\left(x_{1,i+1},...,x_{k,i+1}, \bar{z}_{1,i+1},...,\bar{z}_{k,i+1}\right)$ – целая рациональная функция, реализуемая данным ОБ; Z, $\bar{Z}$, $\nabla\bar{Z}$ – неквантованное, квантованное значения функции F и ее приращения; x_j, ∇x_j $\left(j = \overline{1,l}\right)$ – значения независимых переменных и их приращений; $\bar{z}_j$, $\nabla \bar{z}_j$ $\left(j = \overline{1,k}\right)$ – квантованные значения зависимых интегральных переменных и их приращений; P^{0}_{-n}, P^{-n-1}_{-n-m} – операторы выделения старшего и младшего диапазонов разрядов при квантовании функции F; s_{i+1} – остаток от квантования.

Согласно схеме (6) ОБ выполняет следующие действия:

- вычисляет новые значения интегральных переменных $x_{j,i+1}$ и $\bar{z}_{j,i+1}$ $\left(j = \overline{1,l}\right)$;
- вычисляет новое значение целой рациональной функции Z_{i+1};
- вычисляет приращение этой функции ∇Z_{i+1};
- квантует это приращение, получая значение $\nabla \bar{Z}_{i+1}$;
- экстраполирует квантованное приращение, получая значение $\nabla \bar{Z}^{э}_{i+2}$.

На вход ОБ с выходов ЦИ поступают экстраполированные приращения интегральных приращений, исходя из которых он вычисляет очередное квантованное приращение подынтегральной функции и/или переменной интегрирования. В свою очередь оно используется другими ЦИ, для согласования с которыми ОБ должен проводить экстраполяцию этого приращения. Она может выполняться с помощью алгоритма экстраполяции на основе первых разностей [3].

Использование только экстраполированных приращений позволит избежать усложнения структуры как ОБ, так и коммутационной сети ЦИС. Покажем, что при этом порядок точности вычислений внутри ОБ не изменится, учитывая, что независимые переменные не вносят погрешность в вычисления. Экстраполированное и фактическое значения зависимой интегральной переменной, как следует из [3], связаны как:

$$z_i = z^{э}_i + \left(\nabla^n z_i - \nabla^n z_0\right), \qquad (7)$$

где z_i, $z^{э}_i$ – фактическое и экстраполированное значение интегральной переменной на i-м шаге интегрирования; $\nabla^n z$ – приращения интегральной переменной n-о порядка (n – порядок точности интегральных вычислений) на соответствующих шагах интегрирования. Рассмотрим полином вида (1), реализуемый введенными ОБ, и представляющего собой сумму произведений значений зависимых интегральных переменных. Подставляя (7) в (1), получим:

$$Z = \sum_{i=1}^{m} \prod_{j=1}^{s(i)} \left[z^{э}_i + \left(\nabla^n z_i - \nabla^n z_0\right)\right].$$

При упрощении этого выражения можно пренебречь произведениями, содержащими множители вида $\nabla^n z$, как имеющими порядок малости, превышающий порядок точности соответствующей экстраполяционной формулы интегрирования по Стилтьесу. В результате получим исходный полином (1), что свидетельствует о сохранении порядка точности вычислений.

В результате ЦИС будет включать ЦИ и ОБ, реализующие вычисления в соответствии с введенными в результирующую СУШ целыми рациональными функциями.

Пример. Рассмотрим синтез ЦИС для вычисления функции:

$$z = \frac{x^3 + \frac{1}{y}}{\sin(x)}, \tag{8}$$

Составим согласно разработанному методу порождающую эквивалентную ей СУШ. В соответствии с первым его этапом выделим в (8) функции

$$z_1 = \frac{1}{y},\ z_2 = \frac{1}{\sin(x)}.$$

Тогда (8) можно представить целым рациональным выражением:

$$z = \left(x^3 + z_1\right)z_2. \tag{9}$$

Неделимыми функциями есть z_1, z_2, а также

$$z_3 = \sin(x).$$

Теперь (8) можно представить системой

$$\begin{cases} z = \left(x^3 + z_1\right)z_2 \\ z_1 = \dfrac{1}{y} \\ z_2 = \dfrac{1}{z_3} \\ z_3 = \sin(x) \end{cases}. \tag{10}$$

Дальше выполняем действия в соответствии со вторым этапом метода. Составим для z_3 симметричную СУШ:

$$\begin{cases} dz_3 = z_4 dx \\ dz_4 = -z_3 dx \end{cases}, \tag{11}$$

где $z_4 = \cos(x)$.

Дифференцируя z_2, получим:

$$dz_2 = -\frac{1}{(z_3)^2}dz_3 = -(z_2)^2 dz_3,$$

Что позволяет, введя новую переменную, записать следующую порождающую СУШ для функции z_2:

$$\begin{cases} dz_2 = -z_5 dz_3 \\ z_5 = (z_2)^2 \end{cases}. \tag{12}$$

Для функции z_1 запишем аналогичную порождающую СУШ:

$$\begin{cases} dz_1 = -z_6 dy \\ z_6 = (z_1)^2 \end{cases}. \tag{13}$$

В соответствии с третьим этапом, преобразуя (10) введением в нее СУШ (11) – (13), получим общую порождающую СУШ для функции (8):

$$\begin{cases} z = \left(x^3 + z_1\right)z_2 \\ z_5 = (z_2)^2 \\ z_6 = (z_1)^2 \\ dz_3 = z_4 dx \\ dz_4 = -z_3 dx \\ dz_2 = -z_5 dz_3 \\ dz_1 = -z_6 dy \end{cases}. \tag{14}$$

Зададим в соответствии с последним этапом начальные значения всех зависимых переменных в (14) с учетом начальных значений независимых переменных x_0 и y_0:

$$z_1(y_0) = \frac{1}{y_0},\ z_2(x_0) = \frac{1}{\sin(x_0)},$$

$$z_3(x_0) = \sin(x_0),\ z_4(x_0) = \cos(x_0),$$

$$z_5(x_0) = \left(\frac{1}{\sin(x_0)}\right)^2,\ z_6(y_0) = \left(\frac{1}{y_0}\right)^2,$$

$$z(x_0, y_0) = \frac{x_0^3 + z_1(y_0)}{z_3(x_0)}.$$

На рис. 1 показана построенная в соответствии с порождающей СУШ (14) интегрирующая структура, вычисляющая функцию (8). Здесь ОБ$_1$, используя схему (6), реализует вычисления в соответствии с первым целым рациональным выражением в (14), ОБ$_2$ – со вторым, ОБ$_3$ – с третьим соответственно. ЦИ$_1$ – ЦИ$_4$ проводят численное интегрирование в соответствии с последним, шестым, четвертым и пятым дифференциальными уравнениями соответственно.

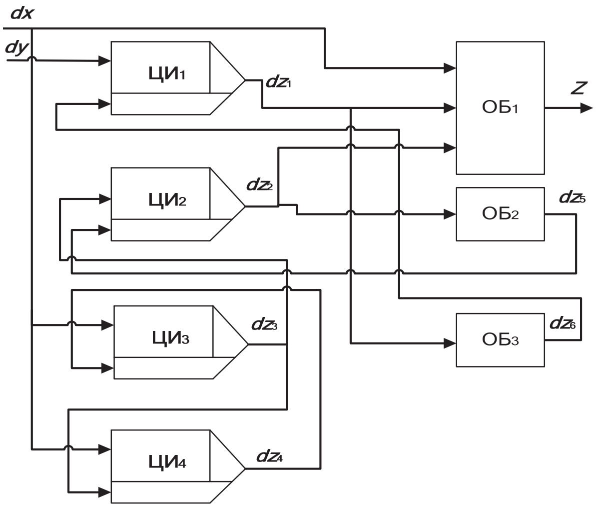

Рис. 1. Структурная схема ЦИС

Исследование характеристик аппаратной сложности и точности ЦИС

Применение известного метода [3] приводит к получению следующей конечной симметричной СУШ для функции (8)

$$\begin{cases} dz = d\left[z_3\left(z_8 + z_2\right)\right] \\ dz_2 = z_5 dy \\ dz_5 = -z_5 dz_2 \\ dz_3 = z_6 dz_4 \\ dz_6 = -z_6 dz_3 \\ dz_4 = z_7 dx \\ dz_7 = -z_4 dx \\ dz_8 = z_9 dx \\ dz_9 = z_{10} dx \\ dz_{10} = 6dx \end{cases}, \qquad (15)$$

которая при построении ЦИС предусматривает использование девяти ЦИ. Для реализации СУШ (14), полученной в соответствии с разработанным методом, достаточно четырех ЦИ. На рис. 2 приведена гистограмма зависимостей общего количества операций умножения от порядка точности формулы интегрирования по Стилтьесу с применением при синтезе ЦИС разработанного (светлый цвет) и известного (темный цвет) методов формирования СУШ для функции (8). Из нее следует, что аппаратурная сложность ЦИС, построенной в соответствии с разработанным методом, может быть уменьшена также практически вдвое при всех рассмотренных порядках точности.

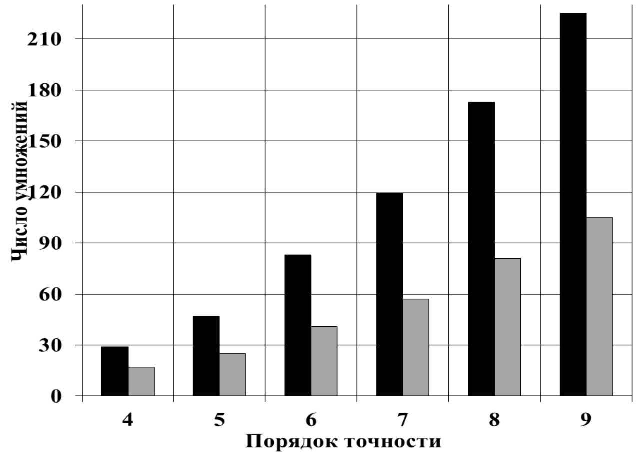

Рис. 2. Суммарное число умножений в ЦИС в зависимости от порядка точности формулы численного интегрирования по Стилтьесу

При увеличении порядка точности вычислений практически прямо пропорционально возрастает число реализуемых в одном ЦИ умножений в отличие от постоянного их числа в (1). Поэтому в общем случае структура реализующего ее ОБ оказывается значительно проще структуры ЦИ. Даже если реализацию произведения (первое дифференциальное уравнение в СУШ (15)) проводить с использованием известной формулы вычисления его приращения без использования ЦИ [3], реализация полиномов вида (1) потребует удвоенного числа умножений в сравнении с непосредственным их вычислением.

В соответствии со структурной схемой (рис. 1) и конечной симметричной СУШ (15) на базе микросхемы *FPGA Altera Stratix* IV *EP*4*SE*230*F*29*C*4 в системах автоматизированного проектирования *Altera Quartus* II v11.0 и *Mentor Graphics ModelSim AE v*6.6*d* получены интегральные реализации ЦИС. Описание схем проводилось на языке *VHDL*, при синтезе минимизировалась задействованная площадь кристалла *FPGA*. Гистограммы зависимостей характеристик ресурсоемкости полученных реализаций ЦИС (число *DSP*- и логических *ALUT*-

блоков (*adaptive look-up table*)) от разрядности и порядка точности ($n = 4,5,6$) интегральных вычислений представлены на рис. 3 (зависимо-

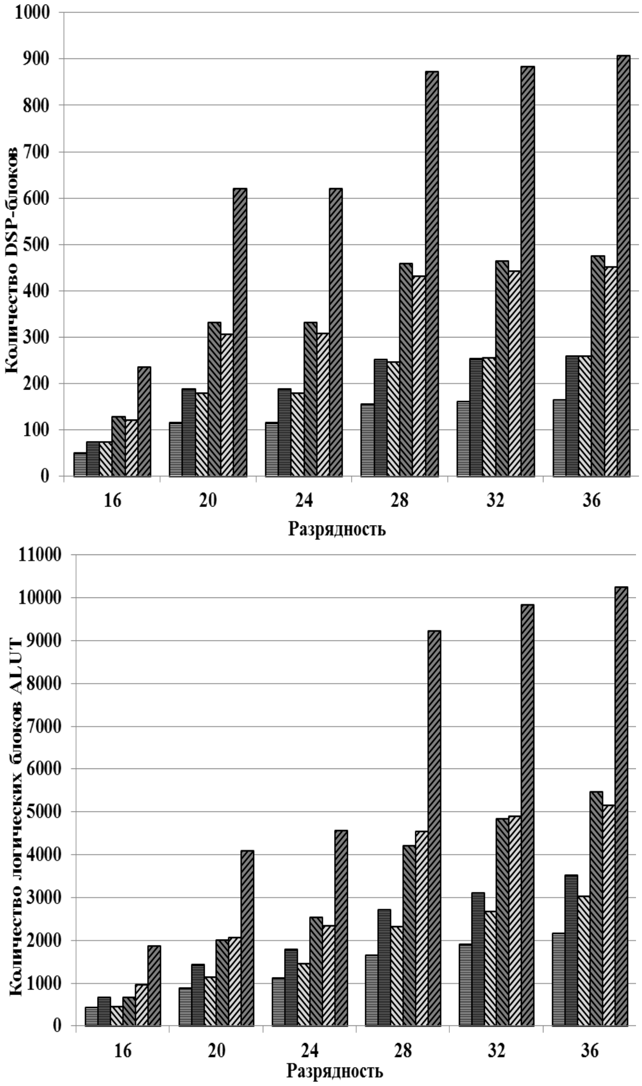

Рис. 3. Зависимости характеристик ресурсоемкости интегральных реализаций ЦИС от разрядности и порядка точности интегральных вычислений

сти одного порядка точности сгруппированы парами и обозначены одинаковым узором, более светлый их фон соответствует разработанному методу). При возрастании точности вычислений в сравнении с известным [3] предложенный метод способствует сокращению расходов арифметических и логических ресурсов микросхем *FPGA* от полутора раз (формула квадратичных парабол, $n = 4$) до двух раз (четвертичных парабол, $n = 6$) на рассматриваемых разрядностях вычислений.

В [9] показано, что при достаточно большом числе шагов численного интегрирования внутри ЦИС может наблюдаться накопление погрешностей, которые могут выйти за допустимые пределы. Разработанный метод формирования эквивалентной СУШ в сравнении с известным создает предпосылки к более высокой точности интегральных вычислений, что объясняется следующим:

– бóльшая часть зависимых интегральных переменных, представленных как целые рациональные функции, вычисляются непосредственно, а не через эквивалентные СУШ;

– меньшее количество дифференциальных уравнений предполагает участие в интегральных вычислениях меньшего числа экстраполированных приращений, что снижает суммарный вклад их погрешностей в процесс интегрирования.

В таблице приведены результаты вычислений функции $Z = \dfrac{1}{x}$. В одном случае для ее вычисления применялась модель ЦИС, реализующая СУШ вида (12) и синтезированная с применением разработанного метода. В другом – применялась модель ЦИС, реализующая симметричную конечную СУШ вида

$$\begin{cases} dz = -zdu \\ du = zdx \end{cases}.$$

В обоих случаях вычисления проводились на одинаковых диапазонах изменения независимой переменной при неизменном ее шаге, использовалась формула интегрирования по Стилтьесу квадратичных парабол. Получены значения абсолютной и относительных погрешностей на первом и последнем шагах интегрирования. Из таблицы видно, что предложенный метод способен обеспечить, по крайней мере, на порядок меньшую абсолютную погрешность. В сравнении с известным методом перехода относительная погрешность в процессе интегрирования уменьшается также почти на порядок, при использовании которого она на столько же увеличивается.

Заключение. Разработан метод, позволяющий снизить аппаратурные затраты при построении

Погрешности вычислений в ЦИС, синтезированных в соответствии с разработанным и известным методом получения эквивалентной СУШ

	$x \in [1;3]$; $\nabla x = 0{,}01$			$x \in [1;10]$; $\nabla x = 0{,}01$			$x \in [1;3]$; $\nabla x = 0{,}001$			$x \in [1;10]$; $\nabla x = 0{,}001$		
	∇Z	Относ. погр. на 1-м шаге	Относ. погр. на посл. шаге	∇Z	Относ. погр. на 1-м шаге	Относ. погр. на посл. шаге	∇Z	Относ. погр. на 1-м шаге	Относ. погр. на посл. шаге	∇Z	Относ. погр. на 1-м шаге	Относ. погр. на посл. шаге
Извест. метод [3]	9,9396E-4	1,0001E-4	2,973E-3	3,5372E-3	1,0001E-4	3,4163E-2	1,0227E-5	1E-6	3,0681E-5	3,6498E-5	1E-6	3,6485E-4
Разраб,метод	3,2505E-5	1,0001E-4	9,7526E-5	2,925E-6	1,0001E-4	2,9251E-5	3,3255E-7	1E-6	9,9766E-7	2,9929E-8	1E-6	2,9929E-7

ЦИС за счет определенной организации перехода от сложных функций многих переменных к порождающим СУШ. Новизна метода заключается в объединении допустимости введения в состав СУШ более сложных, чем интеграторы и сумматоры, *функциональных устройств* (по Шеннону), в частности вычисляющих целые рациональные функции, и высокой эффективности их реализации на базе современных семейств *FPGA*. При этом процесс декомпозиции и последовательного дифференцирования при формировании СУШ проводится до получения подынтегральных и введенных функций как целых рациональных выражений. В сравнении с известным методом это создает предпосылки для существенного уменьшения количества дифференциальных уравнений в СУШ и, как следствие, количества ЦИ в составе ЦИС.

Предложен способ построения ЦИС на основе расширенной в сравнении с известной экстраполяционной разностно-квантованной схемы. Он предусматривает введение в состав ЦИС ОБ реализации целых рациональных функций с меньшей в общем случае, чем у ЦИ, аппаратной сложностью. В то же время возможно в полтора–два раза уменьшить количество ЦИ, а значит практически во столько же снизить аппаратную сложность ЦИС, что подтверждается сравнением основных характеристик ресурсоемкости интегральных реализаций ЦИС на базе ПЛИС типа *FPGA*.

Показана возможность повышения точности ЦИС (в некоторых случаях на порядки), что позволяет не прибегать к методам коррекции промежуточных результатов интегральных вычислений [8] и, как следствие, усложнению ее архитектуры. С другой стороны, становится возможным увеличить шаг интегрирования при сохранении точности, увеличив быстродействие ЦИС, либо использовать формулы интегрирования меньшей точности, уменьшив объем оборудования и сложность ЦИС.

Практическая значимость метода заключается в том, что интегральные реализации ЦИС на базе *FPGA* значительно экономнее расходуют ресурсы кристалла, особенно блоки умножения (*DSP*-блоки). Это способствует удешевлению, снижению энергетических параметров (энергопотребления, тепловой рассеиваемой мощности и т.д.) и других технико-экономических характеристик подобных реализаций. Предложенный метод создает предпосылки к упрощению и ускорению процесса перехода к порождающим СУШ и сокращению сроков и удешевлению этапа разработки ЦИС.

1. *Torrellas J.* Architectures for Extreme-Scale Computing // Computer. – 2009. – **42**, N 11. – P. 28–35.
2. *Greenfield G.I.*, *Szalay P.*, *Williams Roy A.* Data-Intensive Computing in the 21st Century. – Computer. – 2008. – **41**, N 4. – P. 30–32.
3. *Каляев А.В.* Теория цифровых интегрирующих машин и структур. – М.: Сов. радио, 1970. – 472 с.
4. *Опанасенко В.М.*, *Лісовий О.М.* Реалізація проблемно-орієнтованих цифрових пристроїв на кристалах FPGA // Радіоелектронні і комп'ютерні системи. – 2009. – № 5(39). – С. 176–183.
5. *Проектирование* реконфигурируемых цифровых систем / А.В. Палагин, А.А. Баркалов, В.Н. Опанасенко и др. – Луганск: Изд-во ВНУ им. В. Даля, 2011. – 432 с.
6. *Palagin A.V., Opanasenko V.N.* Design and application of the PLD-based reconfigurable devices // Design of Digital Systems and Devices. – Berlin, Heidelberg: Springer, Verlag, 2011. – **79**. – P. 59–91.
7. *Буланкин В.Б.* Параллельные вычислительные системы. – Владимир: Владим. гос. ун-т, 2003. − 32 с.
8. *Опанасенко В.Н., Лисовый А.Н., Сорока Е.В.* Проектирование и физическая верификация цифровых устройств на ПЛИС // Комп'ютерні засоби, мережі та системи. – 2008. – № 7. – С. 76–85.
9. *Каляев А.В.* Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.

Поступила 12.12.2013

Тел. для справок: +38 067 680-2009 (Киев)
E-mail: *nikolay@onet.in.ua*